

INTERDISZIPLINÄRE FINANZDATENANALYSE IM GYMNASIALUNTERRICHT



GymInf | Informatikausbildung für
Gymnasiallehrkräfte
Formation en informatique destinée
aux enseignant·e·s au gymnase

GymInf Individuelles Projekt – Frühlingssemester 2025

Eingereicht von Patrick Suter

GymInf-Student & Gymnasiallehrperson an der Kantonsschule Zug

patrick.suter@unifr.ch, patrick.suter@ksz.ch

Betreut von Prof. Dr. Dennis Komm, Prof. Dr. Juraj Hromkovič und Regula Lacher

dennis.komm@inf.ethz.ch, juraj.hromkovic@inf.ethz.ch, regula.lacher@inf.ethz.ch

Abstract

Mein Individuelles Projekt im GymInf-Studiengang dokumentiert die Konzeption, Durchführung und Evaluation einer Unterrichtsreihe zur Analyse von Finanzdaten im gymnasialen Unterricht. Ziel war es, ökonomisches Denken mit datenanalytischer Kompetenz zu verknüpfen und den Schüler:innen die praktische Arbeit mit realen Kursdaten näherzubringen. Die Arbeit integriert dabei Inhalte aus Wirtschaft, Informatik und Statistik auf fortgeschrittenem Niveau.

In der ersten Unterrichtsreihe werden Grundlagen der Datenbeschaffung, -aufbereitung, Visualisierung und Risikoanalyse vermittelt. Die methodische Struktur besteht aus Impulsfragen, Niveaudifferenzierung (Basic/Advanced/Expert), Reflexionselementen sowie begleitender Lernstandsdiagnostik. Die zweite Unterrichtsreihe ist projektbasiert angelegt: Die Schüler:innen analysierten Aktienkurse mithilfe von Prognosemodellen (Prophet, ARIMA, LSTM) und reflektierten deren Stärken und Schwächen im Hinblick auf Prognosegüte und Modellverständnis.

Die Ergebnisse zeigen, dass Schüler:innen sowohl technische Kompetenzen als auch metakognitive Fähigkeiten (Fehlertoleranz, strukturierte Problemlösung, Modellreflexion) aufbauen konnten. Besonders positiv wirkte die Arbeit mit echten Daten und die Freiheit in der Projektausgestaltung. Gleichzeitig offenbarten sich Herausforderungen in der technischen Umsetzung, insbesondere bei komplexeren Modellen und im Zeitmanagement.

Im Ausblick wird diskutiert, wie sich projektorientierte Datenanalyse im Unterricht langfristig besser verankern lässt. Die Empfehlungen betreffen unter anderem die didaktische Struktur, die gezielte Unterstützung bei offenen Projektaufgaben sowie die begleitende Leistungsrückmeldung während des Lernprozesses. Die Arbeit leistet damit einen Beitrag zur Weiterentwicklung einer datenbezogenen ökonomischen Bildung und bietet ein gut übertragbares Konzept für ähnliche Unterrichtsvorhaben.

Als Lehrperson Informatik und Wirtschaft & Recht möchte einen Beitrag zur Interdisziplinarität dieser beiden Fächer als Abschlussarbeit des GymInf Programms leisten.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Persönliche Motivation und Begründung der Themenwahl	5
1.2	Ziel, Aufbau und Methoden der Unterrichtsreihe	5
1.3	Pilotklasse: Klassendynamik und relevantes Vorwissen	7
2	Entwicklung der Unterrichtsreihe.....	9
2.1	Übersicht	9
2.2	Lernziele: Basic, Advanced und Expert	10
2.3	Lernzielüberprüfung.....	12
2.4	Lektionsaufbau.....	13
2.4.1	Klassischer Lektionsaufbau in Unterrichtsreihe 1.....	13
2.4.2	Hybrider Lektionsaufbau in Unterrichtsreihe 2	13
2.5	Lern-/Aufgaben-Journale	13
2.5.1	Aufgaben-Journal der Unterrichtsreihe 1.....	13
2.5.2	Lern-Journal der Unterrichtsreihe 2	14
3	Unterrichtsreihe Teil 1: Finanzdatenanalyse & Visualisierung	15
3.1	Kompetenzblock 1: Datenimport und -aufbereitung.....	15
3.1.1	Lektion 1: Datenimport mit yfinance und -strukturanalyse mit .head(), .tail(), .columns.....	15
3.1.2	Lektion 2: Datenimport mit pandas, -untersuchung mit .info(), -bereinigung .ffill(), .dropna()	19
3.1.3	Lektion 3: Datenanalyse mit .shape() und .describe(), -filterung mit .loc []	23
3.1.4	Formativer Mini-Check am Ende des Kompetenzblockes 1	28
3.1.5	Repetitionslektion: Börsenweisheit „Sell in May and go away“ überprüfen	29
3.2	Kompetenzblock 2: Datenvisualisierung und -interpretation.....	36
3.2.1	Lektion 4: Datenvisualisierung Liniendiagramme/matplotlib, Candlesticks/plotly.....	36
3.2.2	Lektion 5: Datenglättung mittels gleitender Durchschnitte zur Trendanalyse.....	42
3.2.3	Lektion 6: Mittelwert und SMA als Funktion programmieren und plotten.....	48
3.2.4	Formativer Mini-Check am Ende des Kompetenzblockes 2	53
3.3	Kompetenzblock 3: Datenbasierte Risiko- und Marktanalyse	54
3.3.1	Lektion 7: Datenvolatilität (Standardabweichung, Beta) gegenüber Börsenindizes	54
3.3.2	Lektion 8: Standardabweichung als Funktion programmieren und plotten	59
3.3.3	Lektion 9: Korrelation zwischen Einzelaktien und Gesamtmarkt visualisieren	64
3.3.4	Formativer Mini-Check am Ende des Kompetenzblockes 3	69
3.4	Kompetenzblock 4: Repetition, Überprüfung und Reflexion.....	70
3.4.1	Lektion 10: Erstellung Mini-Dashboard zur Repetition (Partnerarbeit)	70
3.4.2	Lektion 11: Summative Prüfung der Unterrichtsreihe 1.....	74
3.4.3	Lektion 12: Umfrage nach Unterrichtsreihe 1	84

4	Unterrichtsreihe Teil 2: KI-gestützte Finanzmarktprognosen	86
4.1	Kompetenzblock 5: Grundlagen der Zeitreihenprognose.....	88
4.1.1	Lektion 13: Was bedeutet Prognose?.....	88
4.1.2	Lektion 14: SMA-/EMA-basierte naive Vorhersagen	89
4.2	Kompetenzblock 6: Zeitreihenmodelle und KI-gestützte Prognose.....	91
4.2.1	Lektion 15: Prophet (Modellaufbau & Anwendung)	91
4.2.2	Lektion 16: ARIMA mit statsmodels	95
4.2.3	Lektion 17: Modellvergleich und Gütekriterien.....	100
4.3	Kompetenzblock 7: Deep Learning-Modell (optional)	103
4.3.1	Lektion 18: LSTM-Modelle mit Keras (für Expert-Niveau).....	103
4.4	Kompetenzblock 8: Abschlussprojekt	107
4.4.1	Projektphasen mit Arbeitsschritten für Schüler:innen	107
4.4.2	Lernjournal und Reflexionsfragen	108
4.4.3	Bewertungskriterien der Projektarbeit.....	109
4.4.4	Musterlösungen.....	110
4.4.5	Umfrage am Ende der Unterrichtsreihe 2	113
5	Evaluation der Unterrichtsreihe	116
5.1	Lernumfangsanalyse gemäss Aufgaben-Lernjournal aus Schüler:innen-Perspektive	116
5.2	Lernerfolgsanalyse aus Lehrer-Perspektive	119
5.2.1	Auswertung formative Mini-Checks	119
5.2.2	Auswertung summative Prüfung	122
5.2.3	Auswertung Projektarbeiten	125
5.3	Wirksamkeit der Unterrichtsreihe	129
5.3.1	Auswertung Kurzumfragen Temperaturmessung nach Kompetenzblöcken 1-3.....	129
5.3.2	Auswertung Umfrage nach Unterrichtsreihe 1	132
5.3.3	Auswertung Umfrage nach Unterrichtsreihe 2	138
5.4	Gelungene Aspekte und Herausforderungen	144
5.4.1	Gelungene Aspekte und Herausforderungen während der Vorbereitung.....	144
5.4.2	Gelungene Aspekte und Herausforderungen der Unterrichtsreihe 1	145
5.4.3	Gelungene Aspekte und Herausforderungen der Unterrichtsreihe 2	147
5.5	Verbesserungsvorschläge zur Unterrichtsgestaltung und -Struktur.....	148
6	Fazit.....	151
6.1	Zusammenfassung der Erkenntnisse	151
6.2	Persönlicher Rück- und Ausblick	151
7	Quellen und Glossar: Verwendete Bibliotheken, Funktionen und Methoden	154

1 Einleitung

1.1 Persönliche Motivation und Begründung der Themenwahl

Als Lehrperson Informatik und Wirtschaft & Recht möchte einen Beitrag zur Interdisziplinarität dieser beiden Fächer als Abschlussarbeit des Gym Inf Programms leisten.

Als Input für die Weiterentwicklung der Maturität WEGM habe ich im September 2024 zusammen mit einem Team aus Informatik- und Wirtschaftslehrpersonen das Grobkonzept eines interdisziplinären Schwerpunktfachs Informatik und Wirtschaft bei der Schulleitung der Kantonsschule Zug eingereicht. Trotz der zunehmenden Bedeutung von Data Science und Finanzmärkten sind diese Themen bisher in den meisten Lehrplänen kaum vertreten. Ein integrativer Bestandteil des Grobkonzeptes ist die Finanzdatenanalyse mit Python. Ich möchte eine entsprechende Unterrichtsreihe entwickeln, mit meiner Pilotklasse im 4. Gymnasium durchführen und anschliessend evaluieren. So verbindet die Unterrichtsreihe Informatik mit wirtschaftlichen Anwendungsfällen und schafft einen praxisnahen Unterrichtsansatz. Statt rein theoretischer Modelle wird der Fokus auf praxisnahe Finanzdaten und deren Analyse gelegt. Die Schüler:innen arbeiten mit echten Marktdaten und lernen, diese eigenständig zu interpretieren.

Die Finanzwelt mit Aktien, Optionen und Kryptowährungen fasziniert mich seit meinem Wirtschaftstudium an der Universität St. Gallen. Diese Faszination widerspiegelt sich in meiner persönlichen Handelsaktivität auf der Plattform Swissquote und im Durchführen von Börsensimulationen mit den Schüler:innen auf der virtuellen Plattform <https://virtualalpha.io>. An der Kantonsschule Zug verzeichnen wir ein hohes Interesse an Finanzmarktthemen. In Abschlussarbeiten sind Aktienanalyse, Tradingstrategien, Optionshandel bis hin zu automatisierten Trading Bots von Kryptowährungen beliebte Themen. Einige Maturaarbeiten durfte ich in diesem Themenbereich bisher als Maturaarbeitsexperte begleiten. In vielen Bereichen, insbesondere in der Finanzwelt, ist datengetriebene Entscheidungsfindung essenziell. Das Verstehen und Anwenden von Datenanalysen ist eine Schlüsselkompetenz der Zukunft.

1.2 Ziel, Aufbau und Methoden der Unterrichtsreihe

Das Hauptziel der Unterrichtsreihe ist die Vermittlung der Fähigkeit zum Arbeiten mit Finanzmarktdaten. So sollen Schüler:innen im ersten Teil der Unterrichtsreihe reale Finanzzeitreihen eigenständig importieren, bereinigen, visualisieren und interpretieren. Im zweiten Teil der Unterrichtsreihe werden die Schüler:innen einfache bis fortgeschrittene Prognosemodelle anwenden und kritisch reflektieren.

Die Unterrichtsreihe wird zuerst didaktisch-konzeptionell in Kapitel 2 entwickelt. Danach führe ich diese mit meiner Pilotklasse im 4. Gymnasium (siehe Kapitel 1.3) nach den Frühlingsferien gemäss Kapiteln 3 und 4 durch. Abschliessend findet die Validierung gemäss Kapitel 5 statt.

Die gesamte Unterrichtsreihe ist in zwei thematisch und kompetenzorientiert gegliederte Hauptteile unterteilt:

- Der erste Teil (Unterrichtsreihe 1) vermittelt die Grundlagen der Finanzdatenanalyse: Datenimport, -aufbereitung, Visualisierung sowie einfache analytische Verfahren wie gleitende Durchschnitte, Volatilität und Korrelation.
- Der zweite Teil (Unterrichtsreihe 2) führt in die Modellierung und Bewertung von KI-gestützten Zeitreihenprognosen ein. Dabei werden Prognosemodelle (Prophet, ARIMA, LSTM) angewendet und mit geeigneten Gütekriterien (MAPE, RMSE) beurteilt.

Die Reihe umfasst insgesamt 18 Lektionen. Unterrichtsreihe 1 wird als geleiteter Fachunterricht mit klarer Struktur. Die hybride Unterrichtsreihe 2 beinhaltet zusätzlich noch das selbstgesteuerte Projekt mit hoher Individualisierung und Eigenverantwortung.

Jede Lektion folgt einem klaren methodisch-didaktischen Aufbau, der sich im Unterricht bewährt hat:

- Problemorientierter Einstieg mittels Impulsfragen z.B. „Wie volatil war Nestlé im Vergleich zum SMI im März 2024?“
- Lehrerzentrierter Theorie-Input zur Einführung neuer Begriffe und Methoden mit Bezügen zu Wirtschaft, Mathematik und Informatik, z.B. Standardabweichung, Korrelation, Prognoseunsicherheit
- Schülerzentrierte Anwendung in Form von differenzierten Aufgabenstellungen auf den Niveaus Basic, Advanced und Expert; gezielt abgestimmt auf die heterogene Leistungsstruktur der Klasse
- Reflexionsfragen zum Lektionsabschluss; Förderung von Reflexion, Fehlerkultur und Transferdenken

In der Unterrichtsreihe werden verschiedene informatikbezogene Werkzeuge gezielt eingesetzt, die sich in der Praxis bewährt haben und auf Grundlagen des obligatorischen Informatikunterrichts aufbauen:

- Python als Programmiersprache: verständlich, vielseitig und für datenanalytische Anwendungen hervorragend geeignet
- Thonny als leichtgewichtige Entwicklungsumgebung für die meisten Schüler:innen; Visual Studio Code für fortgeschrittene Schüler:innen
- Fachspezifische Bibliotheken zur Datenanalyse, Visualisierung und Modellierung:
 - yfinance zur automatisierten Beschaffung historischer Finanzdaten
 - pandas zur Datenstrukturierung und -verarbeitung
 - matplotlib und plotly zur grafischen Darstellung von Zeitreihen und Analyseergebnissen
 - statsmodels und prophet für Zeitreihenprognosen
 - tensorflow/keras zur Umsetzung neuronaler Netze im Expert-Modul

Die Erläuterungen der in dieser Arbeit verwendeten Python-Methoden und Funktionen basieren überwiegend auf der offiziellen Dokumentation der jeweiligen Bibliotheken: pandas, matplotlib, prophet, statsmodels, keras und yfinance:

- <https://pandas.pydata.org/docs>
- <https://matplotlib.org/stable/contents.html>
- <https://facebook.github.io/prophet/docs>
- <https://www.statsmodels.org/stable/index.html>
- <https://keras.io/api>
- <https://pypi.org/project/yfinance>

1.3 Pilotklasse: Klassendynamik und relevantes Vorwissen

Die Unterrichtsreihe wird in der 4. Klasse im Gymnasium nach den Frühlingsferien 2025 durchgeführt. Die Klasse wird sowohl in Informatik als auch in Wirtschaft und Recht seit dem Schuljahr 2023/24 von Patrick Suter unterrichtet.

Die Klasse besteht aktuell aus 16 Schüler:innen, darunter sowohl ruhigere als auch selbständige und kommunikative Persönlichkeiten. Das Profil der Klasse zeichnet sich durch ein insgesamt ausgewogenes Mass an Leistungsbereitschaft und ein ruhiges, konzentriertes Klassenverhalten aus. Die Schüler:innen arbeiten engagiert und unter klarer Anleitung zuverlässig. Viele bringen sich aktiv in den Unterricht ein und zeigen Eigeninitiative bei Projekten und Arbeitsaufträgen.

Die technische Affinität und der Umgang mit digitalen Werkzeugen sind in der Klasse insgesamt gut. Ein signifikanter Teil der männlichen Schüler zeigt eine ausgeprägte Gaming-Affinität. Dies widerspiegelt sich in Gesprächsthemen, Interessen und technischen Fähigkeiten. Diese informelle IT-Kompetenz lässt sich im Informatikunterricht oft positiv nutzen, z.B. bei Visualisierungen, Simulationen oder Gamification-Elementen.

Die Schüler:innen zeigen eine für unsere Schule normal zu erwartende Bandbreite an Lernfortschritten und -geschwindigkeiten. Die Unterschiede resultieren aus individuellen Begabungen, Vorkenntnissen, Lernstilen, Interessen und dem allgemeinen Verständnis der Unterrichtsinhalte. Einige Schüler verstehen neue Themen sehr schnell, während andere mehr Zeit benötigen, um dieselben Konzepte zu erfassen. Dies führt zu mehreren Herausforderungen: Unter- / Überforderung, ungleichmässige Lernfortschritte und somit reduzierte Unterrichtseffizienz. Aus diesem Grund wurde in der hier vorliegenden Unterrichtsreihe die Inhalte und Lernziele in die drei Niveaus Basic, Advanced und Expert differenziert. Die Einteilung orientiert sich daran, welche Kompetenzen für alle Schüler:innen zugänglich sein sollen und welche vertiefend oder anspruchsvoll sind.

Insgesamt ist die Klasse gut geeignet für das projekt- und datenorientierte Arbeiten. Sie bringt sowohl das ökonomische Vorwissen aus dem Schwerpunkt als auch die technischen Grundlagen aus dem Informatikunterricht mit. Die ruhige Arbeitsweise in Kombination mit der hohen Leistungsbereitschaft ermöglicht tiefgehende Analyseprojekte und macht die Klasse zu einer passenden Zielgruppe für die geplante Unterrichtsreihe zur Finanzdatenanalyse mit Python.

Die Klasse befindet sich aktuell im 4. Jahr des Langzeitgymnasiums an der Kantonsschule Zug und hat ab der 3. Klasse den Schwerpunkt Wirtschaft und Recht gewählt. Die Analyse und Strukturierung des Vorwissens ist eine wichtige Voraussetzung zur Erstellung der Unterrichtsreihe. Das Vorwissen bringen die Schüler:innen aus dem Grundlagen Mathematik, obligatorischen Fach Informatik und dem Schwerpunktfach Wirtschaft & Recht mit. Die untenstehende Tabelle zeigt die für diese Unterrichtsreihe relevanten Themenbereiche. Es ist ersichtlich, dass das notwendige Vorwissen aus den drei Fächern wie folgt vorhanden ist.

- Aus mathematischer Sicht bringen die Schüler:innen solide Grundlagen mit, um Zeitreihendaten mathematisch zu interpretieren, Wachstumsprozesse zu verstehen und statistische Kennwerte zu berechnen.
- Aus Informatik-Sicht sind die Schüler:innen fähig, Python-Code zu verstehen und zu schreiben. Dies ist eine zentrale Voraussetzung für alle praktischen Übungsanteile der Unterrichtsreihe.
- Aus Wirtschaftssicht kennen die Schüler:innen zentrale Kapitalanlagen, die Börse und wirtschaftliche Konjunkturdaten. Somit können sie Finanzdaten nicht nur technisch, sondern auch ökonomisch einordnen.

Fach	Themenbereich	Themen
Grundlagenfach Mathematik	Funktionen und Algebra	• Lineare, quadratische und einfache Exponentialfunktionen • Verständnis von Funktionsgraphen, Schnittpunkten, Steigungen • Rechnen mit Potenzen und Logarithmen (Einführung, noch nicht vertieft) • Grundverständnis von Gleichungen und deren Lösungen
	Statistik Grundlagen	• Deskriptive Statistik: Mittelwert, Median, • Diagrammarten: Balken-, Kreis- und Liniendiagramme • Erste Erfahrungen mit Datenlisten, Häufigkeiten, Klassenbildung
	Modellierung	• Erste Anwendungen: Wachstumsprozesse linear/exponentiell • Übersetzen einfacher Alltagssituationen in mathematische Darstellungen
Obligatorisches Fach Informatik	Python Grundkenntnisse	• Variablen, Datentypen (int, float, str, list, dict) • Kontrollstrukturen (if, for, while) • Funktionen
	Algorithmen	• Algorithmisches Denken (Biber Wettbewerb) • Such- und Sortieralgorithmen (Insertion-, Quicksort) • Teilprobleme strukturieren (Teile und herrsche Prinzip)
	Fehlersuche und Debugging	• Erstellen eigenständige Programmcode Blöcke • Lesen und Interpretieren von Fehlermeldungen
Schwerpunktfach Wirtschaft & Recht	Finanzmärkte und Börse	• Funktionsweise von Börsen, Aktien, Obligationen • Kapitalbeschaffung und Anlageformen
	Rechnungswesen	• Erfolgsrechnung, Bilanzanalyse, Kennzahlen (Liquidität, Rentabilität)
	Makroökonomische Indikatoren	• Bruttoinlandprodukt (BIP), Inflation, Konjunktur • Bedeutung von Finanzstatistiken

Abbildung 1: Relevantes Vorwissen in Mathematik, Informatik und Wirtschaft

2 Entwicklung der Unterrichtsreihe

2.1 Übersicht

Die Unterrichtsreihe ist in die Basisreihe «Finanzdatenanalyse & Visualisierung» und die Fortsetzungsreihe «KI-gestützte Finanzmarktprognosen» unterteilt. Beide Reihen behandeln je vier Kompetenzblöcke.

Diese klare Struktur wurde gewählt, um den progressiven didaktischen Aufbau aufzuzeigen. Dies bildet eine Grundlage für die Ausgestaltung der Lernziele, Lektionsablaufplanung, Differenzierung der Schwierigkeiten und Evaluation der Unterrichtsreihe.

Unterrichtsreihe	Kompetenzblock	Ziel
Unterrichtsreihe 1: Finanzdatenanalyse und Visualisierung	1. Datenimport und -aufbereitung	Schüler:innen können Zeitreihendaten finden, importieren und bereinigen.
	2. Datenvisualisierung und -interpretation	Schüler:innen können Daten visuell aufbereiten und Muster erkennen.
	3. Datenbasierte Risiko- und Marktanalyse	Schüler:innen können Risiken, Korrelationen und Marktvergleiche analysieren.
	4. Repetition, Prüfung und Umfrage	Schüler:innen können Aktienanalyse erstellen und Erlerntes reflektieren.
Unterrichtsreihe 2: KI-gestützte Finanzmarktprognosen	5. Grundlagen der Zeitreihenprognose	Schüler:innen verstehen die Logik hinter Prognosemodellen.
	6. Zeitreihenmodelle und KI-gestützte Prognose	Schüler:innen wenden Prophet und ARIMA an.
	7. Deep Learning-Modell (optional)	Schüler:innen erlernen Prognosen mit neuronalen Netzen.
	8. Projektarbeit / Abschluss parallel zu 5-7	Schüler:innen erstellen eigene Prognosen für selbstgewählte Finanzdaten.

Abbildung 2: Übersicht Unterrichtsreihe mit Kompetenzblöcken

In einem nächsten Schritt wurden die 8 Kompetenzblöcke auf Lektionen aufgeteilt. Dabei wurde darauf geachtet, dass jeder Kompetenzblock 1-3 Lektionen beinhaltet. Repetitions- und Prüfungslektionen werden auch eingebaut.

Unterrichtsreihe 1: Kompetenzblöcke und Lektionen
1. Datenimport und -aufbereitung • Lektion 1: Datenimport mit yfinance und -strukturanalyse mit .head(), .tail() und .columns • Lektion 2: Datenimport mit pandas, -untersuchung mit .info(), -bereinigung .ffill(), .dropna() • Lektion 3: Datenanalyse mit .shape() und .describe(), -filterung mit .loc []
2. Datenvisualisierung und -interpretation • Lektion 4: Datenvisualisierung mit Linien- und Candlestick-Charts (matplotlib, plotly) • Lektion 5: Datenglättung mittels gleitender Durchschnitte zur Trendanalyse

• Lektion 6: Mittelwert und SMA als Funktion programmieren und plotten	
3. Datenbasierte Risiko- und Marktanalyse • Lektion 7: Datenvolatilität (Standardabweichung, Beta) im Vergleich zu Börsen-Indizes • Lektion 8: Standardabweichung als Funktion programmieren und plotten • Lektion 9: Korrelation zwischen Einzelaktien und Gesamtmarkt visualisieren	
4. Repetition, summative Prüfung und Umfrage zur Unterrichtsreihe 1 • Lektion 10: Erstellung Mini-Dashboard zur Repetition (Partnerarbeit) • Lektion 11: Summative Prüfung • Lektion 12: Umfrage zu Unterrichtsreihe 1 bezüglich Unterrichtsaufbau und Methodik, Inhalt und Kompetenzerwerb, Wirkung und persönlicher Lernerfolg	
Unterrichtsreihe 2: Kompetenzblöcke und Lektionen	
5. Grundlagen der Zeitreihenprognose • Lektion 13: Was bedeutet Prognose? • Lektion 14: SMA-/EMA-basierte naive Vorhersagen	8. Abschlussprojekt parallel zu 5-7
6. Zeitreihenmodelle und KI-gestützte Prognose • Lektion 15: Prophet (Modellaufbau & Anwendung) • Lektion 16: ARIMA mit statsmodels • Lektion 17: Modellvergleich & Gütekriterien	
7. Deep Learning-Modell (optional) • Lektion 18: LSTM-Modelle mit Keras	

Abbildung 3: Übersicht Lektionen Einteilung je Kompetenzblock

2.2 Lernziele: Basic, Advanced und Expert

Die Unterrichtsreihe „Finanzdatenanalyse mit Python“ hat zum Ziel, 6 Kompetenzblöcke abzudecken. Dazu wurde ein Kompetenzraster als Übersicht der Lernziele mit entsprechender Differenzierung in die drei Niveaus Basic, Advanced und Expert erstellt. Die Einteilung orientiert sich daran, welche Kompetenzen für alle Schüler:innen zugänglich sein sollen und welche vertiefend oder anspruchsvoll sind.

- **Basic Ziele** bilden die Grundlage und werden im gemeinsamen Unterricht vermittelt. Alle Schüler:innen sollen diese Lernziele erfüllen können.
- **Advanced Ziele** können in Phasen individueller Vertiefung oder in Partnerarbeit bearbeitet werden.
- **Expert-Ziele** dienen der Förderung besonders leistungsstarker Schüler:innen im Sinne der Binnendifferenzierung.

Kompetenzblock	Basic	Advanced	Expert
1. Datenimport und -aufbereitung	Ich kann Kursdaten aus verschiedenen Quellen laden, die Struktur des Datensatzes beschreiben und einfache Eigenschaften anzeigen.	Ich kann Kursdaten bereinigen, fehlende Werte behandeln. Auch kann ich gezielt nach Zeiträumen und Kursinformationen filtern.	Ich kann mehrere Kursdatenquellen kombinieren, strukturierte Vergleiche erstellen und komplexe statistische Kennwerte analysieren.

2. Datenvisualisierung und -interpretation	Ich kann Kursverläufe mit Liniendiagrammen darstellen und einfache Visualisierungen mit matplotlib erstellen.	Ich kann Diagramme individuell gestalten und SMA/EMA zur Interpretation von Trends grafisch vergleichen.	Ich kann komplexe Zeitreihenanalysen mit mehreren SMA-Fenstern und interaktiven Candlestick-Charts visualisieren und interpretieren.
3. Datenbasierte Risiko- und Marktanalyse	Ich kann absolute und relative Volatilität eines Kursverlaufs berechnen und deren Aussagekraft erklären.	Ich kann Kursdaten mit einem Index vergleichen, Korrelationen berechnen und diese wie ein Beta interpretieren.	Ich kann relative Volatilitäten über Zeiträume hinweg analysieren, visuell darstellen und daraus Marktverhalten ableiten.
4. Repetition, summative Prüfung und Umfrage zur Unterrichtsreihe 1	Ich kann die wichtigsten Begriffe und Methoden aus der Unterrichtsreihe (z.B. Datenimport, Mittelwert, Liniendiagramm) wiederholen und in einer Prüfung anwenden.	Ich kann typische Aufgaben aus der Unterrichtsreihe selbstständig lösen und die Ergebnisse korrekt interpretieren, auch wenn die Fragestellung leicht variiert wurde.	Ich kann mein eigenes Lernen reflektieren, den Schwierigkeitsgrad der Inhalte einschätzen und Vorschläge zur Verbesserung der Unterrichtsreihe auf Basis der Umfrageergebnisse formulieren.
5. Grundlagen der Zeitreihenprognose	Ich kann einfache Begriffe wie Prognose, Trend und Unsicherheit erklären.	Ich kann einfache Methoden wie SMA/EMA zur Vorhersage einsetzen.	Ich kann naive Prognosen mit SMA/EMA reflektieren und mit realen Kursverläufen vergleichen.
6. Zeitreihenmodelle und KI-gestützte Prognose	Ich kann ein KI-gestütztes Prognosemodell (Prophet) mit einer Vorlage anwenden.	Ich kann verschiedene Modelle (Prophet, ARIMA) vergleichen und mit MAPE bewerten.	Ich kann die Prognosefähigkeit komplexer Modelle kritisch interpretieren und mit RMSE begründen.
7. Deep Learning-Modell (optional)	-	Ich verstehe, dass neuronale Netze für komplexe Daten geeignet sind.	Ich kann ein LSTM-Netz trainieren und interpretieren.
8. Abschlussprojekt parallel zu 5-7	Ich kann mit Prophet eine Prognose zu einer Zeitreihe erstellen und darstellen.	Ich kann Prophet und ARIMA anwenden, vergleichen und die Prognosegüte interpretieren.	Ich kann Prophet, ARIMA und LSTM kombinieren, die Güte mit MAPE & RMSE bewerten und kritisch reflektieren.

Abbildung 4: Übersicht Lernziele Basic, Advanced und Expert je Kompetenzblock

2.3 Lernzielüberprüfung

Die Lernzielüberprüfung soll sowohl formativ während den Lektionen als auch summativ am Ende der Unterrichtsreihe 1 und 2 stattfinden.

Formativ wird der Lernstand mittels Aufgaben-Journal und formativen Mini-Checks durchgeführt:

- **Aufgaben-Lernjournal:** Schüler:innen bearbeiten Aufgaben, die klar den drei Niveaus Basic, Advanced und Expert zugeordnet sind. Sie führen das Lernjournal ihrer Selbsteinschätzung gemäss folgendem Ampelsystem. Das Lernjournal gibt Hinweise, welche Aufgaben mit welcher Güte verstanden worden sind.
 - Grau = Ich habe die Aufgabe nicht angeschaut.
 - Rot = Ich brauche noch Hilfe, um die Aufgabe zu verstehen.
 - Orange = Ich verstehe die Aufgabe, kann sie aber nicht komplett selbstständig lösen.
 - Grün = Ich kann die Aufgabe komplett selbstständig lösen.
- **Formative Mini-Checks:** Am Ende eines Kompetenzblocks erfolgt eine formative Lernkontrolle mit 5 kurzen Fragen pro Niveau-Stufe Basic, Advanced und Expert. Diese werden digital auf Classtime zur gezielten Lernstandserfassung und Selbstreflexion durchgeführt. In kurzer Zeit kann überprüft werden, ob die zentralen Kompetenzen eines Kompetenzblockes verstanden und angewendet werden können. Im Gegensatz zur summativen Prüfung werden die Mini-Checks nicht benotet. Sie sollen die Schüler:innen unterstützen, ihre individuellen Lernfortschritte sichtbar zu machen. Gleichzeitig geben sie der Lehrperson wertvolle Hinweise darauf, welche Inhalte gefestigt sind und wo gezielte Wiederholung nötig ist.

Summativ werden die Lernstände mittels Prüfung und Projektarbeitsauswertung durchgeführt.

- **Summative Prüfung zur Unterrichtsreihe 1:** Die digitale Prüfung auf Classtime ist eine Kombination aus Fachbegriffen, Codeverständnis, Fehlersuche, Diagramminterpretation und einfacher Code-Generierung. Sie prüft die Basic- und Advanced-Kompetenzen in Datenimport, Bereinigung, Visualisierung und Statistik.
- **Projektarbeit der Unterrichtsreihe 2:** Die Schüler:innen erstellen und präsentieren als Projektarbeit ein Dashboard mit Daten, Visualisierung und Prognose. Dabei soll die Anwendung von Prognoseverfahren (SMA, Prophet, ARIMA, LSTM) überprüft und reflektiert werden.

Zudem werden die Schüler:innen nach der Unterrichtsreihe 1 und 2 eine Umfrage zur Wirksamkeit und Verbesserungsvorschlägen ausfüllen.

- **Umfrage nach Unterrichtsreihe 1:** Die Schüler:innen geben eine Rückmeldung zur eigenen Einschätzung und zum Unterricht auf Google Forms.
 - Quantitative Einschätzung der unterschiedlichen Komponenten der Unterrichtsreihe: Lernziele, Übungsmöglichkeiten, Niveau-Differenzierung, Mini-Checks, Lernjournal, summative Prüfung, Lernbegleitung
 - Qualitative Einschätzungen / offene Fragen:
 - Was war für dich das Beste an dieser Unterrichtsreihe?
 - Was war für dich besonders herausfordernd oder unklar?
 - Welche Verbesserungsvorschläge hast du für eine nächste Durchführung?
 - Was hast du persönlich aus der Reihe für dich mitgenommen?
- **Umfrage nach Unterrichtsreihe 2:** Die Schüler:innen geben eine Rückmeldung zur eigenen Einschätzung und zum Projekt auf Google Forms.

- Quantitative Einschätzung bezügl. subjektiver Erfahrungseinschätzung und persönlichem Projektfortschritt
- Qualitative Einschätzungen / offene Fragen:
 - Was war für dich die wichtigste Erkenntnis beim Arbeiten mit einem Prognosemodell?
 - Was fiel dir besonders schwer? warum?
 - Welches Prognosemodell hat dich am meisten überzeugt? warum?
 - Was würdest du deiner nächsten Projektarbeit anders machen?

2.4 Lektionsaufbau

2.4.1 Klassischer Lektionsaufbau in Unterrichtsreihe 1

Die meisten Lektionen der Unterrichtsreihe 1 folgen grundsätzlich einem ähnlichen didaktischen Ablauf.

Repetition / Warm-up (5 min)

Kurze Wiederholung der letzten Inhalte / Aktivierung von Vorwissen mittels Impulsfragen

Wissenserarbeitung: Theorie-Input / Gemeinsames Coding (10 min)

Einführung in das fachliche Konzept (z.B. Mittelwert, gleitender Durchschnitt, Beta). Lehrperson zeigt exemplarisch eine Umsetzung (z.B. Liniendiagramm, Berechnung mit pandas) und Schüler:innen arbeiten mit.

Wissenssicherung: Üben und Anwenden (20-25 min)

Schüler:innen bearbeiten differenzierte Aufgaben (Basic, Advanced, Expert) individuell oder in Partnerarbeit. Bei Lektionen mit erhöhter Komplexität kann die selbständige Übungsphase auch verlängert werden.

Zusammenfassung / Reflexion / Austausch (5-10 min)

Kurze Diskussion der Lösungen zu den Aufgaben und Beantwortungen der Reflexionsfragen

2.4.2 Hybrider Lektionsaufbau in Unterrichtsreihe 2

In der Unterrichtsreihe 2 wird ein hybrider Aufbau verfolgt. Somit wird der klassische Lektionsaufbau mit einem selbstgesteuerten Projektauftrag verbunden. Die Lektionen 13 bis 17 führen modellbasiert und theoriegestützt in das Thema ein. Die Projektarbeit wird parallel selbstorganisiert umgesetzt. So erhalten die Schüler:innen sowohl fachliche Inputs als auch Anwendungsfreiheit. Auch bringt der hybride Lektionsaufbau eine Abwechslung nach der Unterrichtsreihe 1 und unterstützt Differenzierung und kompetenzorientiertes Arbeiten. So können Schüler:innen mit unterschiedlichem Vorwissen und Tempo flexibel arbeiten, erhalten aber dennoch die fachliche und technische Grundlage zur Umsetzung.

2.5 Lern-/Aufgaben-Journale

2.5.1 Aufgaben-Journal der Unterrichtsreihe 1

Die Schüler:innen führen während den Lektionen 1-10 der Unterrichtsreihe 1 ein Lernjournal mit Google Forms unter <https://forms.gle/Y5T8vz2K8NSRhbj27> digital.

In diesem Lernjournal haben sie festgehalten, welche Aufgaben sie wie gut lösen konnten. Dabei wurde folgende Einschätzungsskala bereitgestellt.

- 0% : Ich habe die Aufgabe nicht angeschaut
- 33% : Ich brauche noch Hilfe, um die Aufgabe zu verstehen (ROT)
- 66% : Ich verstehe die Aufgabe, kann sie aber nicht komplett selbstständig lösen (ORANGE)
- 100% : Ich kann die Aufgabe komplett selbstständig lösen (GRÜN)

Lektion 1 *

Datenimport mit yfinance und -strukturanalyse mit .head(), .tail() und .columns

	0%	33%	66%	100%
Basic	☐	☐	☐	☐
Advanced	☐	☐	☐	☐
Expert	☐	☐	☐	☐

Abbildung 5: Lernjournal in Google Forms

Die Auswertung der Aufgaben-Journale befindet sich in Kapitel 5.1

2.5.2 Lern-Journal der Unterrichtsreihe 2

Die Schüler:innen führen während den Lektionen der Unterrichtsreihe 2 ein Lernjournal, das sie zusammen mit dem Projektauftrag am Ende der Unterrichtsreihe einreichen.

Lernjournal

Datum	Modell / Thema	Was habe ich gemacht?	Schwierigkeit	Nächstes Ziel	Reflexion
16.5.2025	Prophet / Bitcoin	Prognose mit Prophet durchgeführt	Mittel	Vergleich mit ARIMA	Prophet ist sehr einfach anwendbar

Abbildung 6: Unterrichtsreihe 2 - Lernjournal

3 Unterrichtsreihe Teil 1: Finanzdatenanalyse & Visualisierung

3.1 Kompetenzblock 1: Datenimport und -aufbereitung

In diesem Kompetenzblock lernen die Schüler:innen, wie man Finanzdaten aus verschiedenen Quellen (API und CSV) importiert, strukturell überprüft und bereinigt. Dabei werden zentrale pandas-Funktionen wie `.head()`, `.info()`, `.ffill()` und `.dropna()` eingeführt. Ziel ist es, eine saubere, korrekt formatierte Datengrundlage für weiterführende Analysen zu schaffen. Das methodische Fundament für Datenverständnis und digitale Lesekompetenz wird gelegt.

3.1.1 Lektion 1: Datenimport mit `yfinance` und -strukturanalyse mit `.head()`, `.tail()`, `.columns`

Lektion 1 bildet die Einführung in die Finanzdatenanalyse mit Python und legt den Fokus auf Verständnis von Zeitreihen und erste Erfahrungen mit Datenimport. Der niedrighschwellige Einstieg über `.head()` und `.columns` ist bewusst reduziert gehalten, um Überforderung zu vermeiden. Die differenzierte Aufgabenstellung (Basic/Advanced/Expert) ermöglicht es, alle Schüler:innen entsprechend ihrem Können aktiv einzubinden. Durch Austauschphasen und Reflexion wird das technische Tun zudem sprachlich und konzeptuell verankert.

Lernziele

Niveau	Lernziele
Basic	• Ich kann Kursdaten mit <code>yfinance</code> laden. • Ich kann die ersten Zeilen eines DataFrames mit <code>.head()</code> anzeigen. • Ich kann mir mit <code>.columns</code> die Spaltenüberschriften anzeigen lassen.
Advanced	• Ich kann Kursdaten eines bestimmten Zeitraums gezielt laden. • Ich kann nur eine bestimmte Spalte (z.B. Close) ausgeben. • Ich kann den ersten und letzten Wert einer Zeitreihe bestimmen.
Expert	• Ich kann mehrere Aktienkurse (z.B. Apple und Google) gleichzeitig laden. • Ich kann gezielt einzelne Zeilen aus mehreren DataFrames anzeigen. • Ich kann die Anzahl Zeilen pro Kursdatensatz ausgeben und vergleichen.

Abbildung 7: Lektion 1 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	• Fokus der Lektion (Lehrgespräch) • Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	• Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) • Basic Aufgabe gemeinsam lösen (Lehrgespräch) • Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	• Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)

5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	• Diskussion der Lösungen zu den Aufgaben (Plenum) • Beantwortungen der Reflexionsfragen (Lehrgespräch)
-------	---	--

Abbildung 8: Lektion 1 – Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Was könnten Gründe sein, den Verlauf eines Aktienkurses anzuschauen?	Um zu sehen, ob sich ein Investment lohnt oder wie sich der Markt entwickelt hat.
2.	Was braucht man, um mit Computern Finanzdaten zu analysieren?	Eine Datenquelle (z.B. Tabelle oder Website) und ein Programm, das die Daten verarbeitet.
3.	Was unterscheidet tägliche, wöchentliche und monatliche Kursdaten?	Je kürzer das Intervall, desto detaillierter, aber auch schwankungsanfälliger sind die Daten.

Abbildung 9: Lektion 1 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
Ticker-Symbol	Ein Kürzel, das eine Aktie eindeutig identifiziert (z. B. AAPL für Apple, NESN.SW für Nestlé).
DataFrame	Eine strukturierte Tabelle in Python (pandas), mit Zeilen und Spalten, in der Daten gespeichert werden.
.columns	Zeigt Spaltennamen der Datentabelle
.head()	Zeigt die ersten fünf Zeilen eines DataFrames (also die „Kopfzeile“ der Daten).
.tail()	Zeigt die letzten fünf Zeilen eines DataFrames.
yfinance.download()	Lädt historische Kursdaten direkt von Yahoo Finance.
Yfinance Installation	«Where Thonny» in Eingabeaufforderung eingeben. In Pfad reingehen, z.B. C:\Users\suter\AppData\Local\Programs\Thonny «python -m pip install yfinance» eingeben

Abbildung 10: Lektion 1 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

Übungen

Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade die Aktienkurse von Nvidia („NVDA“) aus dem Jahr 2024 mit yfinance
- Zeige die ersten fünf Zeilen mit .head(5)
- Zeige die Spaltennamen mit .columns an

Ausgabe

```
Kommandozeile x
>>> %Run 'Lektion 1.py'

YF.download() has changed argument auto_adjust default to True
[*****100%*****] 1 of 1 completed
Price      Close      High      Low      Open      Volume
Ticker      NVDA      NVDA      NVDA      NVDA      NVDA
Date
2024-01-02  48.149918  49.276493  47.577135  49.225514  411254000
2024-01-03  47.551144  48.165911  47.302237  47.467176  320896000
2024-01-04  47.979984  48.481795  47.490167  47.749068  306535000
2024-01-05  49.078568  49.528399  48.287864  48.443808  415039000
2024-01-08  52.233383  52.255378  49.460426  49.493415  642510000
MultiIndex([( 'Close', 'NVDA'),
              ( 'High', 'NVDA'),
              ( 'Low', 'NVDA'),
              ( 'Open', 'NVDA'),
              ('Volume', 'NVDA')],
            names=['Price', 'Ticker'])

>>>
```

Musterlösung

```
1 import yfinance as yf
2
3 # Aufgabe Basic
4 nvda = yf.download("NVDA", start="2024-01-01", end="2024-12-31", interval="1d")
5 print(nvda.head(5))
6 print(nvda.columns)
7
```

Advanced Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade die Aktienkurse von Nestlé vom 4. Quartal 2024
- Extrahiere nur die Schlusskurse
- Zeige den ersten und letzten Kurs

Ausgabe

```
Kommandozeile x
>>> %Run 'Lektion 1.py'

YF.download() has changed argument auto_adjust default to True
[*****100%*****] 1 of 1 completed
Ticker      NESN.SW
Date
2024-10-01  84.919998
Ticker      NESN.SW
Date
2024-12-30  74.879997

>>>
```

Musterlösung

```

8 # Aufgabe Advanced
9 nesn = yf.download("NESN.SW", start="2024-10-01", end="2024-12-31", interval="1d")
10 nesn_close = nesn["Close"]
11 print(nesn_close.head(1))
12 print(nesn_close.tail(1))

```

Expert Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Lade die Aktienkurse von Apple und Google aus dem Jahr 2024
- Gib die erste Zeile der beiden Datentabellen aus
- Gib die Anzahl Zeilen der beiden Datensätze aus

Ausgabe

Kommandozeile x

```

>>> %Run 'Lektion 1.py'

YF.download() has changed argument auto_adjust default to True
[*****100%*****] 2 of 2 completed
Price      Open      High      Low      Close      Volume
Date
2024-01-02  186.033072  187.315382  182.792533  184.532089  82488700
Price      Open      High      Low      Close      Volume
Date
2024-01-02  137.890755  138.786466  135.830597  137.512558  23711200
Index(['Open', 'High', 'Low', 'Close', 'Volume'], dtype='object', name='Price')
AAPL: 251 Zeilen, GOOGL: 251 Zeilen

>>>

```

Musterlösung

```

14 # Aufgabe Expert
15 tickers = ["AAPL", "GOOGL"]
16 data = yf.download(tickers, start="2024-01-01", end="2024-12-31", interval="1d", group_by='ticker')
17 print(data["AAPL"].head(1))
18 print(data["GOOGL"].head(1))
19 print(data["AAPL"].columns)
20 print(f"AAPL: {len(data['AAPL'])} Zeilen, GOOGL: {len(data['GOOGL'])} Zeilen")

```

Abbildung 11: Lektion 1 - Niveau-differenzierte Übungen inkl. Musterlösungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage	Antwort
1. Was ist eine Zeitreihe und wo begegnet sie uns im Alltag oder in der Wirtschaft?	Eine Zeitreihe ist eine geordnete Folge von Messwerten, die in regelmäßigen Abständen über die Zeit hinweg erfasst wurden, z.B. Im Alltag: Temperaturverlauf, Stromverbrauch, Schrittzähler z.B. In der Wirtschaft: Aktienkurse, BIP-Entwicklung, Arbeitslosenzahlen
2. Was macht die Funktion .head() in Python mit einem DataFrame?	.head() zeigt die ersten Zeilen einer Datentabelle an. Optional: df.head(10) zeigt die ersten 10 Zeilen.
3. Was zeigt dir .columns bei einem Datensatz?	.columns gibt eine Liste mit den Spaltennamen der Datentabelle zurück. So sieht man, ob es Spalten wie Open, Close, Volume gibt.

Abbildung 12: Lektion 1 – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

3.1.2 Lektion 2: Datenimport mit pandas, -untersuchung mit .info(), -bereinigung .ffill(), .dropna()

Die Methode `.read_csv()` ermöglicht zusätzlich zu `yfinance` einen alternativen Datenimport. Mit der Einführung von `.info()`, `.shape` und `.isnull()` werden wichtige Werkzeuge zur Strukturanalyse und Fehlererkennung vermittelt.

Durch die Einbindung absichtlich fehlerhafter Datentabellen lernen die Schüler:innen, dass Daten in der Praxis selten sauber sind. Dieses Problembewusstsein trägt wesentlich zur Entwicklung von Datenkompetenz bei.

Lernziele

Niveau	Lernziele
Basic	- Ich kann einen CSV-Datensatz mit <code>read_csv()</code> einlesen. - Ich kann die Datenstruktur mit <code>.info()</code> überprüfen und interpretieren. - Ich kann die Eigenschaften eines DataFrames in der Konsole darstellen.
Advanced	- Ich kann fehlende Werte mit <code>.isnull().sum()</code> erkennen. - Ich kann fehlende Werte mit dem vorherigen Wert (<code>.ffill()</code>) ersetzen. - Ich kann überprüfen, ob nach dem Ersetzen noch fehlende Werte vorhanden sind.
Expert	- Ich kann alle Zeilen mit fehlenden Werten mit <code>.dropna()</code> entfernen. - Ich kann die Zeilenanzahl vor und nach dem Entfernen von NaNs vergleichen. - Ich kann qualitativ bewerten, ob das Entfernen von Daten die Analyse beeinflusst.

Abbildung 13: Lektion 2 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 14: Lektion 2 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Was könnten Fehlerquellen in grossen Datentabellen sein?	Leere Zellen, Tippfehler oder doppelte Einträge.
2.	Wie würdest du mit fehlenden Daten umgehen?	Löschen oder ersetzen.
3.	Was bedeutet für dich eine gute Datenbasis für eine Analyse?	Vollständige, verständliche, aktuelle und saubere Daten.

Abbildung 15: Lektion 2 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
<code>read_csv()</code>	Liest / importiert ein csv-File
<code>.info()</code>	Gibt Informationen zu Spalten, Datentypen und wie viele Werte fehlen (z.B. NaN).
<code>.isnull()</code>	Prüft, ob ein Feld leer ist (gibt True für fehlende Werte zurück).
<code>.sum()</code>	Rechnet die Summe oder zählt z.B. die Anzahl der NaN-Werte, wenn mit <code>.isnull()</code> kombiniert.
<code>.dropna()</code>	Entfernt Zeilen, in denen Werte fehlen.
<code>.ffill()</code>	Füllt fehlende Werte mit dem zuletzt bekannten Wert (forward fill).

Abbildung 16: Lektion 2 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Was zeigt dir <code>.info()</code> einer Datentabelle?	<code>.info()</code> zeigt Datentypen, Spaltenanzahl, Nicht-Null-Werte je Spalte.
2.	Wie erkennst du, ob eine Spalte fehlende Werte enthält?	<code>.isnull().sum()</code> zeigt die Anzahl fehlender Werte pro Spalte.
3.	Welche Möglichkeiten gibt es, mit NaN umzugehen?	Entweder Vorwerte mit <code>.ffill()</code> übernehmen oder NaN mit <code>.dropna()</code> entfernen.

Abbildung 17: Lektion 2 – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Übungen

Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_2_NOVN.csv» mit read_csv()
- Überprüfe die Datenstruktur mit .info()
- Zeige die Datenstrukturanalyse in der Konsole an.

Ausgabe

```
Kommandozeile x
>>> %Run 'Lektion 2.py'

-- Strukturanalyse mit .info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 312 entries, 0 to 311
Data columns (total 6 columns):
#   Column  Non-Null Count  Dtype
---  ---
0   Date    311 non-null      object
1   Close   297 non-null      object
2   High    312 non-null      object
3   Low     312 non-null      object
4   Open    297 non-null      object
5   Volume  297 non-null      object
dtypes: object(6)
memory usage: 14.8+ KB
None
```

Musterlösung

```
1 import pandas as pd
2
3 # Aufgabe Basic
4 df = pd.read_csv("Lektion_2_NOVN.csv")
5 print("-- Strukturanalyse mit .info() ")
6 print(df.info())
7
```

Advanced Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_2_NOVN.csv» mit read_csv()
- Zeige die Anzahl der fehlenden Werte pro Spalte mit .isnull().sum()
- Ersetze sie mit den Vorwerten mit .ffill().
- Zeige die Anzahl der fehlenden Werte pro Spalte nach dem Ersetzen.

Ausgabe

```
>>> %Run 'Lektion 2.py'

Fehlende Werte pro Spalte vor .ffill():
Date      1
Close     15
High      0
Low       0
Open      15
Volume    15
dtype: int64

Fehlende Werte pro Spalte nach .ffill():
Date      1
Close     0
High      0
Low       0
Open      0
Volume    0
dtype: int64
```

Musterlösung

```

7
8 # Aufgabe Advanced
9 df = pd.read_csv("Lektion_2_NOVN.csv")
10 print("Fehlende Werte pro Spalte vor .ffill():")
11 print(df.isnull().sum()) # Zeigt Spalten mit fehlenden Werten
12 df_filled = df.ffill() # Vorherigen Wert übernehmen
13 print("")
14 print("Fehlende Werte pro Spalte nach .ffill():")
15 print(df_filled.isnull().sum())
16

```

Expert Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Lade «Lektion_2_NOVN.csv» mit read_csv()
- Zeige die Anzahl der fehlenden Werte pro Spalte mit .isnull().sum()
- Entferne alle Zeilen mit fehlenden Werten mit .dropna().
- Vergleiche die Anzahl Zeilen vor und nach dem Entfernen.

Ausgabe

```

>>> %Run 'Lektion 2.py'

Fehlende Werte pro Spalte vor .dropna():
Date      1
Close     15
High      0
Low       0
Open      15
Volume    15
dtype: int64
Zeilen vor dropna(): 312
Fehlende Werte pro Spalte nach .dropna():
Date      0
Close     0
High      0
Low       0
Open      0
Volume    0
dtype: int64
Zeilen nach dropna(): 267

```

Musterlösung

```

17 # Aufgabe Expert
18 df = pd.read_csv("Lektion_2_NOVN.csv")
19 print("Fehlende Werte pro Spalte vor .dropna():")
20 print(df.isnull().sum()) # Zeigt Spalten mit fehlenden Werten
21 print(f"Zeilen vor dropna(): {len(df)}") # Zeilenanzahl vorher
22 df_clean = df.dropna() # Entfernen der Zeilen mit NaN
23 print("Fehlende Werte pro Spalte nach .dropna():")
24 print(df_clean.isnull().sum()) # Zeigt Spalten mit fehlenden Werten
25 print(f"Zeilen nach dropna(): {len(df_clean)}")

```

Abbildung 18: Lektion 2 - Niveau-differenzierte Übungen inkl. Musterlösungen

3.1.3 Lektion 3: Datenanalyse mit .shape() und .describe(), -filterung mit .loc []

Lektion 3 stellt die Verbindung zwischen strukturierter Datenaufbereitung (Lektion 2) und der inhaltlichen Analyse & Visualisierung (Lektion 4) her. Die Schüler:innen lernen, umfangreiche Datensätze zu charakterisieren und zu filtern.

Lernziele

Niveau	Lernziele
Basic	- Ich kann die Struktur eines DataFrames mit .shape analysieren. - Ich kann Kursdaten mit .to_numeric() in Float-Zahlen umwandeln. - Ich kann mit .describe() statistische Basiswerte eines Kurses anzeigen lassen.
Advanced	- Ich kann Daten mit .loc[] auf bestimmte Zeiträume (Monat, Jahr) filtern. - Ich kann Monats- und Jahresdurchschnitte einer Kursreihe berechnen und ausgeben.
Expert	- Ich kann zwei Aktienkurse für das gleiche Jahr mit yfinance laden. - Ich kann zwei Quartale (Q1, Q4) extrahieren und mit .describe() vergleichen. - Ich kann Marktvolatilitäten zwischen Quartalen interpretieren (Standardabweichung, Min/Max, Quartile).

Abbildung 19: Lektion 3 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 20: Lektion 3 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Was bedeutet für dich Durchschnitt im Alltag?	Ein typischer Mittelwert z.B. der Notenschnitt.
2.	Wie kann man herausfinden, ob ein Aktienkurs stark schwankt?	Indem man sich den Unterschied zwischen Hochs und Tiefs anschaut.
3.	Warum könnten Extremwerte (Ausreisser) problematisch sein?	Weil sie den Eindruck verfälschen und die Analyse verzerren.

Abbildung 21: Lektion 3 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
.shape	Gibt die Grösse der Tabelle an: Anzahl Zeilen und Spalten
.describe()	Zeigt statistische Kennzahlen wie Mittelwert, Min, Max, Quartile usw.
.to_numeric()	Wandelt Textdaten in Zahlen um Nützlich, wenn Zahlen als Text importiert wurden.
.loc[]	Dient zum Filtern von Zeilen z.B. nach Datum oder Bedingungen.
Standardabweichung	Ein Mass dafür, wie stark die Werte im Durchschnitt vom Mittelwert abweichen.

Abbildung 22: Lektion 3 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Wie hilft .describe() beim ersten Überblick über deine Kursdaten?	.describe() zeigt wichtige Kennzahlen wie den Mittelwert, die Standardabweichung und die Extremwerte. Man sieht schnell, ob es Ausreisser gibt und wie stark die Werte schwanken. So kann man besser einschätzen, ob die Daten ruhig oder volatil sind.
2.	Was lernt man, wenn z.B. der März 2025 herausgefiltert und analysiert wird?	Der Monatsdurchschnitt zeigte, ob es in diesem Monat einen Trend gab oder ob er eher durchschnittlich verlief.
3.	Was bedeutet es, wenn der Monatsdurchschnitt stark vom Jahresdurchschnitt abweicht?	Es kann ein Hinweis auf ein Ereignis oder einen Trendwechsel sein. Der Vergleich zeigt, ob der Monat auffällig war oder nicht.

Abbildung 23: Lektion 3 – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Übungen

Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_3_ROGN.csv»
- Gib .shape der Datenstruktur aus
- Wandle die Close-Kurse mit .to_numeric() in float Zahlen um
- Gib .describe() für Close aus.

Ausgabe

```
>>> %Run 'Lektion 3.py'

Shape zeigt :
(1262, 6)
Describe zeigt:
count      1261.000000
mean       269.427469
std        30.101442
min        207.245987
25%        249.272614
50%        267.003784
75%        289.681427
max        358.954010
Name: Close, dtype: float64
```

Musterlösung

```
1 import pandas as pd
2 import yfinance as yf
3
4 # Aufgabe Basic
5 df = pd.read_csv("Lektion_3_ROGN.csv")
6 print("Shape zeigt :")
7 print(df.shape)
8 df["Close"] = pd.to_numeric(df["Close"], errors="coerce") # Umwandlung in float
9 print("Describe zeigt: ")
10 print(df["Close"].describe())
```

Advanced Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Filtere die Daten mit `.loc[]` auf März 2024 und das Jahr 2024
- Gib den Monatsdurchschnitt und den Jahresdurchschnitt der Close-Kurse auf der Konsole aus

Ausgabe

```
>>> %Run 'Lektion 3.py'

März-Mittelwert: 221.09822769165038
Jahres-Mittelwert: 240.92779766845703
```

Musterlösung

```
11
12 # Aufgabe Advanced
13 df = pd.read_csv("Lektion_3_ROGN.csv", parse_dates=["Date"])
14 df["Close"] = pd.to_numeric(df["Close"], errors="coerce") # Umwandlung in float
15 df.set_index("Date", inplace=True) # Index auf das Datum setzen
16 month = df.loc["2024-03"] #März 2024 auswählen
17 print("März-Mittelwert:", month["Close"].mean())
18 year = df.loc["2024"] #Jahr 2024 auswählen
19 print("Jahres-Mittelwert:", year["Close"].mean())
20
```

Expert Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade zwei verschiedene Aktienkurse (z.B. Novartis und Apple) für das Jahr 2024 mit yfinance
- Filtere Q1 2024 und Q4 2024 jeweils mit .loc[]
- Berechne mit .describe() die statistischen Kennwerte der Close-Werte.
- Vergleiche die Marktvolatilität beider Quartale – wo schwankte der Kurs stärker? Begründe deine Aussage anhand von Standardabweichung, Min/Max und Quartilen.

Ausgabe

```
>>> %Run 'Lektion 3.py'

YF.download() has changed argument auto_adjust default to True
[*****100%*****] 1 of 1 completed
Q1 2024:
Ticker      NOVN.SW
count      62.000000
mean       84.280211
std        1.533452
min        81.585159
25%        83.357559
50%        84.185875
75%        85.072901
max        87.838036

Q4 2024:
Ticker      NOVN.SW
count      62.000000
mean       90.557793
std        4.110525
min        83.726112
25%        87.406986
50%        89.900330
75%        94.305323
max        97.493813
[*****100%*****] 1 of 1 completed
Q1 2024:
Ticker      AAPL
count      61.000000
mean       180.870374
std        7.291873
min        168.205765
25%        172.903580
50%        181.771698
75%        185.744812
max        194.015152

Q4 2024:
Ticker      AAPL
count      63.000000
mean       235.071950
std        10.629499
min        221.203156
25%        226.845589
50%        231.270996
75%        242.758087
max        258.735504
```

Musterlösung

```
21 # Aufgabe Expert
22 # Novartis-Aktie
23 df = yf.download("NOVN.SW", start="2024-01-01", end="2024-12-31") # Daten holen
24 q1 = df["Close"].loc["2024-01":"2024-03"] # Quartal1 filtern
25 q4 = df["Close"].loc["2024-10":"2024-12"] # Quartal4 filtern
26 # Kennzahlen vergleichen
27 print("Q1 2024:")
28 print(q1.describe())
29 print("\n Q4 2024:")
30 print(q4.describe())
31
32 # Apple-Aktie
33 df = yf.download("AAPL", start="2024-01-01", end="2024-12-31") # Daten holen
34 q1 = df["Close"].loc["2024-01":"2024-03"] # Quartal1 filtern
35 q4 = df["Close"].loc["2024-10":"2024-12"] # Quartal4 filtern
36 # Kennzahlen vergleichen
37 print("Q1 2024:")
38 print(q1.describe())
39 print("\n Q4 2024:")
40 print(q4.describe())
```

Abbildung 24: Lektion 3 - Niveau-differenzierte Übungen inkl. Musterlösungen

3.1.4 Formativer Mini-Check am Ende des Kompetenzblockes 1

Am Ende des Kompetenzblockes wird ein formativer Mini-Check anhand von je 5 Kurzfragen pro Niveau Basic, Advanced und Expert durchgeführt. Diese formative Lernzielkontrolle fand auf Classtime digital statt. Sie kann mittels folgendem Link angeschaut werden <https://www.classtime.com/student/login/N983RWK>. Die Schüler:innen lösen den Mini-Check in einer sicheren Umgebung im Safe Exam Browser. Die Auswertung des Mini-Checks ist in Kapitel 5.2.1 Auswertung formative Mini-Checks. Der Mini-Check umfasst folgende 15 Fragen:

The screenshot shows the Classtime interface for a Mini-Check. At the top, there are navigation links: MEINE FRAGEN, SESSIONS, KLASSEN, and a search bar. A user profile for Patrick Suter is visible. The main heading is 'Meine Fragen > Mini-Check Kompetenzblock 1: Datenimport & -aufbereitung'. Below this, a list of 15 questions is displayed, categorized by difficulty (Basic, Advanced, Expert). Each question has a point value of 1 Pkte. On the right side, there are icons for 'Neue Frage', 'Vorschau / Übungslink', 'Veröffentlichen', 'Teilen', 'Kopieren nach ...', 'Verschieben nach ...', 'Ausdrucken', and 'Papierkorb'.

Frage	Punkte
1 Basic 1: Welche Methode zeigt dir die Anzahl Zeilen und Spalten in einem DataFrame?	1 Pkte.
2 Basic 2: Welche Funktion verwendest du, um mit yfinance die Kursdaten von Apple (AAPL) zu laden?	1 Pkte.
3 Basic 3: Was gibt df.columns zurück?	1 Pkte.
4 Basic 4: Wie setzt man die Datumsspalte als Index?	1 Pkte.
5 Basic 5: Welche Methode zeigt die ersten fünf Zeilen eines importierten DataFrames?	1 Pkte.
6 Advanced 1: Was zeigt df["Spalte"].isnull().sum() an?	1 Pkte.
7 Advanced 2: Welche Methode ersetzt fehlende Werte mit dem jeweils vorherigen Wert?	1 Pkte.
8 Advanced 3: Was bewirkt .dropna()?	1 Pkte.
9 Advanced 4: Was macht .loc["2025-03"]?	1 Pkte.
10 Advanced 5: Wie ersetzt man alle NaN mit dem Mittelwert der Spalte?	1 Pkte.
11 Expert 1: Was ist der Unterschied zwischen .shape[0] und .shape[1]?	1 Pkte.
12 Expert 2: Was macht pd.to_numeric(df["Close"], errors="coerce")?	1 Pkte.
13 Expert 3: Was macht .describe() bei einer Spalte?	1 Pkte.
14 Expert 4: Wie zeigt man die Anzahl Zeilen eines DataFrames df1?	1 Pkte.
15 Expert 5: Was bewirkt .resample("M") bei Zeitreihen?	1 Pkte.

Abbildung 25: Formativer Mini-Check nach Kompetenzblock 1

3.1.5 Repetitionslektion: Börsenweisheit „Sell in May and go away“ überprüfen

In dieser zusätzlichen Lektion untersuchen die Schüler:innen die im Wirtschaftsunterricht diskutierte Börsenweisheit: „Sell in May and go away – but remember to come back in September.“ Ziel ist es, auf Basis realer Kursdaten zu überprüfen, ob sich zwischen Mitte Mai und Mitte September tatsächlich ein überdurchschnittliches Kursrisiko oder eine negative Entwicklung beobachten lässt. Die Lektion fördert den Transfer von Datenanalyse, Visualisierung und Interpretation im wirtschaftlichen Kontext.

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Klärungsfragen gemäss Temperaturmessung (Plenum)
5'	Wissenserarbeitung: Repetition	- Überblick behandelte Themen und verwendete Methoden (Repetition) (Lehrgespräch) - Erklärung Auftrag und Bewertung (Lehrgespräch)
30'	Wissenssicherung: Üben und Anwenden	Erarbeitung Auftrag (Partnerarbeit)
5'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Beantwortung offener Fragen (Plenum) - Würdigung Rangliste (Plenum)

Abbildung 26: Repetitionslektion Börsenweisheit - Lektionsablauf

Klärungsfragen gemäss Temperaturmessung inkl. Musterlösungen

Frage	Antwort
1. Was ist der Unterschied zwischen <code>len(df)</code> und <code>df.shape[0]</code> ?	- Keiner. Beide Methoden sind gleichwertig. <code>len(df)</code> und <code>df.shape[0]</code> : gibt Anzahl Zeilen zurück <code>df.shape[1]</code> : gibt Anzahl Spalten zurück <code>df.shape</code> : gibt (Anzahl Zeilen, Anzahl Spalten) zurück
2. Was bedeutet <code>.describe()</code> ?	gibt statistische Kennzahlen über eine Spalte oder den ganzen DataFrame zurück
3. Was ist <code>.close()</code> ?	<code>.close()</code> ist keine offizielle pandas-Funktion. Es ist wahrscheinlich ein Missverständnis oder Tippfehler. - Schüler:in meint <code>df["Close"]</code>. Das ist der Zugriff auf die Spalte mit Schlusskursen (Close).
4. Was bedeutet <code>error = coerce</code> ?	<code>coerce</code> = erzwingen - Python erzwingt die Umwandlung und ersetzt Problemwerte mit NaN. - Problemwerte könnten Wörter wie «Affe», «Banane», etc. sein - Es wird somit kein Fehler ausgelöst - Häufige Verwendung bei <code>pd.to_numeric()</code>

5.	Was bedeutet inplace=True ?	• Änderungen werden direkt im aktuellen Dataframe durchgeführt. • Es wird keine neue Variable angelegt • Beispiel: df.dropna(inplace=True)
6.	Was bedeutet parse_dates=["Spaltenname"]?	• Angegebene Spalte wird direkt beim Einlesen automatisch in Datumsformat (datetime) umgewandelt. • Alternative Umwandlung mit pd.to_datetime() • Ohne diese Angabe wäre die Datumsspalte nur ein Textfeld. Zeitfilterungen würden nicht funktionieren.

Abbildung 27: Repetitionslektion Börsenweisheit – Klärungsfragen gemäss Temperaturmessung

Überblick behandelte Themen und verwendete Methoden (Repetition)

Behandelte Themen	Verwendete Methoden / generische Form
Datenimport	• yfinance.download ("TICKER", start="YYYY-MM-DD", end="YYYY-MM-DD", interval="1d"/"1wk"/"1mo") • pd.read_csv ("dateiname.csv", parse_dates=["Datum-Spalte"])
Datenuntersuchung	• df.head(n) : zeigt die ersten n Zeilen • df.tail(n) : zeigt die letzten n Zeilen • df.columns : gibt die Spaltennamen zurück • df.info() : Überblick über Struktur und Datentypen • df.shape : Anzahl Zeilen und Spalten
Datenbereinigung	• df.dropna() : entfernt Zeilen mit fehlenden Werten • df.ffill() : füllt fehlende Werte mit dem vorherigen Wert • df.fillna(df["Spalte"].mean()) : ersetzt NaN mit Spaltenmittelwert • pd.to_numeric(df["Spalte"], errors="coerce") : wandelt Text in Zahlen um
Datenfilterung	• df.loc["YYYY-MM"] : filtert nach Jahr und Monat (Index = datetime) • df.loc[start_date:end_date] : filtert nach Zeitintervall • df.iloc[Zahl] : positionsbasierter Zugriff auf Zeile (neu!) • df[df["Spalte"] > wert] : filtert nach Bedingung • df.resample ("Y" / "Q" / "M") : Zeitreihe in neue Zeitintervalle gruppieren
Datenanalyse	• df.describe() : statistische Kennzahlen • df["Spalte"].mean() : Mittelwert
Datenvisualisierung	• plt.plot(df["Spalte"]) : Liniendiagramm mit matplotlib • plt.title(), plt.xlabel(), plt.ylabel() : Achsen und Titel • plt.grid(True), plt.legend() : Zusatzfunktionen für Lesbarkeit • go.Figure(data=go.Ohlc(...)) : Candlestick-Chart mit plotly.graph_objects

Abbildung 28: Repetitionslektion Börsenweisheit – Überblick behandelte Themen und verwendete Methoden (Repetition)

Aufgabe

Untersuche für Dir bekannte Aktien über mehrere Jahre hinweg, ob die Börsenweisheit «Sell in May and go away; but remember to come back in September» zutrifft. Befolge dazu folgende Schritte:

1. Lade Kursdaten entweder über yfinance oder aus bisher bereitgestellten .csv-Dateien.
2. Filtere den Zeitraum vom 15. Mai bis 15. September eines Kalenderjahres.
3. Gib folgende Daten aus: den Kurs am 15. Mai, den Kurs am 15. September, die prozentuale Veränderung
4. Entscheide, ob die Börsenweisheit in diesem Jahr für diese Aktie zutrifft.
5. Visualisiere den Verlauf mit einem Liniendiagramm oder einem OHLC-Diagramm
6. Kommentiere deine Ergebnisse in 1–2 Sätzen.

Bewertung (max. 3 Punkte pro Aktie/Jahr)

- +1 Punkt für jede korrekt berechnete Konsolenausgabe
- +1 Punkt für ein korrektes Diagramm
- +1 Punkt für eine wirtschaftlich sinnvolle Interpretation

Aktie	Jahr	Börsen- weisheit zutreffend?	Nummerisch mit- tels Konsolenaus- gabe	Graphisch in Dia- gramm	Erklärung in Text
SIKA	2024				

Abbildung 29: Repetitionslektion Börsenweisheit – Übungsdokumentation

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Fühlst du dich sicherer im Umgang mit Kursdaten?	Individuelle Antworten
2.	Was war dein spezielles aha-Erlebnis heute?	Individuelle Antworten
3.	Was musst du dir noch- mals genauer anschauen?	Individuelle Antworten

Abbildung 30: Repetitionslektion Börsenweisheit – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Musterlösung Lehrperson

Aktie	Jahr	Börsen- weisheit zutref- fend?	Numerisch mittels Konsolenausgabe	Graphisch in Diagramm	Erklä- rung in Text
SIKA	2024	Ja	<pre>>>> %Run 'Lektion Börsenweisheit.py' YF.download() has changed argument auto_adjust default to True [*****100%*****] 1 of 1 completed Startkurs (15. Mai): Ticker SIKA.SW 283.399994 Name: 2024-05-15 00:00:00, dtype: float64 Endkurs (15. Sept): Ticker SIKA.SW 267.0 Name: 2024-09-13 00:00:00, dtype: float64 Veränderung in %: Ticker SIKA.SW -5.79 dtype: float64</pre>		Sika ist vom 15.5. bis 15.9. um 5.79% gesunken.

```
>>> %Run 'Lektion Börsenweisheit.py'
YF.download() has changed argument auto_adjust default to True
[*****100%*****] 1 of 1 completed
Startkurs (15. Mai): Ticker
SIKA.SW      283.399994
Name: 2024-05-15 00:00:00, dtype: float64
Endkurs (15. Sept): Ticker
SIKA.SW      267.0
Name: 2024-09-13 00:00:00, dtype: float64
Veränderung in %: Ticker
SIKA.SW      -5.79
dtype: float64
```

Sika OHLC – Mai bis Sept 2024



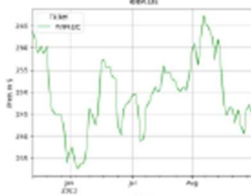
```
1 import pandas as pd
2 import yfinance as yf
3 import matplotlib.pyplot as plt
4 import plotly.graph_objects as go
5
6 # Daten laden
7 ticker = "SIKA.SW"
8 df = yf.download(ticker, start="2024-04-01", end="2024-10-01")
9 # oder df = pd.read_csv("Pfad / Dateiname.csv")
10
11 # Datum als Index setzen
12 df.reset_index(inplace=True)
13 df["Date"] = pd.to_datetime(df["Date"])
14 df.set_index("Date", inplace=True)
15
16 # Zeitraum filtern
17 df_period = df.loc["2024-05-15":"2024-09-15"]
18
19 # Konsolenausgabe
20 start_price = df_period["Close"].iloc[0]
21 end_price = df_period["Close"].iloc[-1]
22 veraenderung = round(((end_price - start_price) / start_price) * 100, 2)
23 print("Startkurs (15. Mai): ", start_price)
24 print("Endkurs (15. Sept): ", end_price)
25 print("Veränderung in %: ", veraenderung)
26
27 # Liniendiagramm
28 plt.plot(df_period["Close"])
29 plt.title("Sika - Kursverlauf (Mai-Sept 2024)")
30 plt.xlabel("Datum")
31 plt.ylabel("CHF")
32 plt.grid(True)
33 plt.show()
34
35 # OHLC-Diagramm
36 fig = go.Figure(data=go.Candlestick(
37     x=df_period.index,
38     open=df_period["Open", "SIKA.SW"],
39     high=df_period["High", "SIKA.SW"],
40     low=df_period["Low", "SIKA.SW"],
41     close=df_period["Close", "SIKA.SW"]
42 ))
43 fig.update_layout(title="Sika OHLC - Mai bis Sept 2024")
44 fig.show()
```

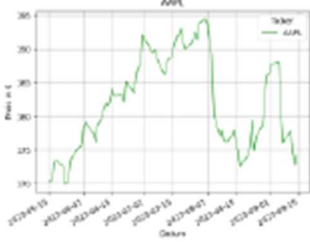
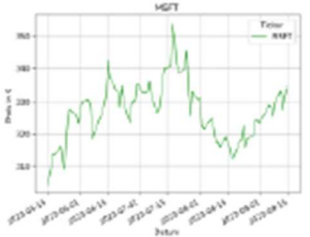
Lösung Siegerteam

```

1 import yfinance as yf
2 import pandas as pd
3 import plotly.graph_objects as go
4 import matplotlib.pyplot as plt
5
6 ticker = input("Bitte gib den Aktien-Ticker ein (z.B. GOOGL): ").upper()
7
8 # Herunterladen der Aktienkurse
9 df = yf.download(ticker, start="2023-05-15", end="2023-09-15", interval="1d")
10
11 # Zeigen des Close werts am anfang und am Ende des DataFrames
12 print(df["Close"].head(1))
13 print(df["Close"].tail(1))
14
15 # Diagramm für Die Aktien
16 df["Close"].plot(title="ticker", xlabel="Datum", ylabel="Preis in $", color="green", grid=True, linestyle='-', linewidth=1)
17
18 plt.show()

```

Aktie	Jahr	Börsenweisheit zutreffend?	Numerisch mittels Konsolenausgabe	Graphisch in Diagramm	Erklärung. in Text
NVDA	2023	Nein	Ticker NVDA Date 2023-05-15 28.93421 Ticker NVDA Date 2023-09-14 45.559887		Wert am 15.5.2023= 28.9 Wert am 15.9.2023= 45.6 Prozentuale Veränderung= + 57.78% Hier trifft die Börsenwahrheit auf jeden Fall nicht zu. Über den Sommer hinweg hat die Aktie fast um 50% an Wert gewonnen. Wer im Mai
			Date 2023-05-15 28.93421 Ticker NVDA Date 2023-09-14 45.559887		Wert am 15.5.2023= 45.6 Prozentuale Veränderung= + 57.78% Hier trifft die Börsenwahrheit auf jeden Fall nicht zu. Über den Sommer hinweg hat die Aktie fast um 50% an Wert gewonnen. Wer im Mai verkauft hätte und im September zurückgekommen wäre, wäre wahrscheinlich sehr enttäuscht gewesen.
RHM.DE	2023	Ja	Ticker RHM.DE Date 2023-05-15 264.181152 Ticker RHM.DE Date 2023-09-14 253.993698		Wert am 15.5.2023= 264.1 Wert am 15.9.2023=254 Prozentuale Veränderung=-3.82% Hier trifft die Börsenwahrheit zu da die Aktien im Sommer stark an Wert verlieren. Im September beginnen sie dann wieder zu steigen.
MSTR	2023	Nein	Ticker MSTR Date 2023-05-15 28.25 Ticker MSTR Date 2023-09-14 34.639999		Wert am 15.5.2023=28.5 Wert am 15.9.2023= 34.6 Prozentuale Veränderung=+ 22.5% Auch hier trifft die Wahrheit nicht zu. Die Aktie gewinnt über den Sommer einiges an Wert. Mitte Juli erreicht die Aktie sogar einen Höchstwert von 45. Hier wäre es schlauer, die Aktie zu verkaufen.
PLTR	2023	Nein	Ticker PLTR Date 2023-05-15 9.52 Ticker PLTR Date 2023-09-14 15.83		Wert am 15.5.2023=9.5 Wert am 15.9.2023=15.8 Prozentuale Veränderung=+ 66.31% Die Wahrheit stimmt hier auch nicht. Kurz nach dem 15.5 beginnt die Aktie rasant zu steigen. Einem wäre ein riesiger Gewinn entgangen, wenn man im Mai verkauft hätte.

AAPL	2023	Nein	Ticker AAPL Date 2023-05-15 170.364258 Ticker AAPL Date 2023-09-14 174.232834		Wert am 15.5.2023=170.4 Wert am 15.9.2023=174.2 Prozentuale Veränderung=+ 2.23% Hier trifft es nur knapp nicht zu. Die Aktie gewinnt über den Sommer nur 2.23% an Wert. Im August erreicht die Aktie aber einen Peak von fast 195. Da wäre es schlaue gewesen, zu verkaufen.
MSFT	2023	Nein	Ticker MSFT Date 2023-05-15 304.057007 Ticker MSFT Date 2023-09-14 334.220093		Wert am 15.5.2023=304.1 Wert am 15.9.2023=334.2 Prozentuale Veränderung=+ 9.89% Börsenwahrheit trifft nicht zu. Aktie gewinnt leicht an Wert über den Sommer.
AMZN	2023	Nein	Ticker AMZN Date 2023-05-15 111.199997 Ticker AMZN Date 2023-09-14 144.720001		Wert am 15.5.2023=111.2 Wert am 15.9.2023=144.7 Prozentuale Veränderung=+ 30.12% Börsenwahrheit trifft nicht zu. Aktie gewinnt stark an Wert über den Sommer.
META	2023	Nein	Ticker META Date 2023-05-15 237.739532 Ticker META Date 2023-09-14 310.257751		Wert am 15.5.2023=237.7 Wert am 15.9.2023=310.3 Prozentuale Veränderung=+ 30.54% Börsenwahrheit trifft nicht zu. Aktie gewinnt stark an Wert über den Sommer.
GOOGL	2023	Nein	Ticker GOOGL Date 2023-05-15 115.955627 Ticker GOOGL Date 2023-09-14 137.442886		Wert am 15.5.2023=116 Wert am 15.9.2023=137.4 Prozentuale Veränderung=+ 18.45% Börsenwahrheit trifft nicht zu. Aktie gewinnt stark an Wert über den Sommer

3.2 Kompetenzblock 2: Datenvisualisierung und -interpretation

Im Kompetenzblock 2 wird der Fokus auf die visuelle Darstellung und Interpretation von Zeitreihendaten gelegt. Die Schüler:innen nutzen matplotlib und plotly, um Kursverläufe als Linien- und Candlestick-Charts darzustellen. Sie erkennen, wie sich Trends mit gleitenden Durchschnitten (SMA, EMA) sichtbar machen lassen. Der Block fördert das Verständnis für Datenmuster und unterstützt die Lesekompetenz von Diagrammen.

3.2.1 Lektion 4: Datenvisualisierung Liniendiagramme/matplotlib, Candlesticks/plotly

Diese Lektion eröffnet einen intuitiven, visuell orientierten Zugang zur Datenanalyse. Während matplotlib schnell zu ersten Ergebnissen führt, erlaubt plotly das Eintauchen in interaktive Analysen.

Die Differenzierung ermöglicht es, sowohl grundlegende als auch fortgeschrittene Visualisierungskompetenzen aufzubauen. Durch die Deutung realer Kursverläufe entstehen automatisch Verbindungen zur Wirtschaft, zur Marktbeobachtung und zum späteren Prognoseverständnis.

Diese Lektion bildet den Übergang von der numerischen Analyse in Lektion 3 zur Interpretationskompetenz, die in Lektion 5 weiter vertieft wird.

Lernziele

Niveau	Lernziele
Basic	Ich kann Kursdaten laden und ein einfaches Liniendiagramm mit matplotlib erstellen.
Advanced	- Ich kann Liniendiagramme gestalten: Titel, Farbe, Linienstil, Achsenbeschriftung. - Ich kann verschiedene Kurven innerhalb eines Diagramms visuell anpassen.
Expert	- Ich kann interaktive Candlestick-Charts mit plotly.graph_objects erstellen (OHLC). - Ich kann die vier Kurspunkte (Open, High, Low, Close) visuell darstellen

Abbildung 31: Lektion 4 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 32: Lektion 4 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Warum kann ein Bild manchmal mehr sagen als eine Tabelle?	Weil Muster, Trends oder Veränderungen schneller sichtbar sind.
2.	Wie würdest du einem Laien einen Börsenverlauf erklären?	Mit einem Diagramm z.B. einem Kurschart.
3.	Was ist der Vorteil einer Zeitachse in einem Diagramm?	Man erkennt, wann sich etwas verändert hat.

Abbildung 33: Lektion 4 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
matplotlib.pyplot	Ein Python-Modul zur grafischen Darstellung von Daten.
plt.plot()	Zeichnet ein einfaches Liniendiagramm.
plt.xlabel(), plt.ylabel()	Beschriftet die x- bzw. y-Achse.
plt.title()	Gibt dem Diagramm einen Titel.
Candlestick-Chart	Eine Darstellungsform zur Anzeigen von Open, High, Low und Close
plotly.graph_objects	Ein Python-Modul für interaktive Candlestick-Charts

Abbildung 34: Lektion 4 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

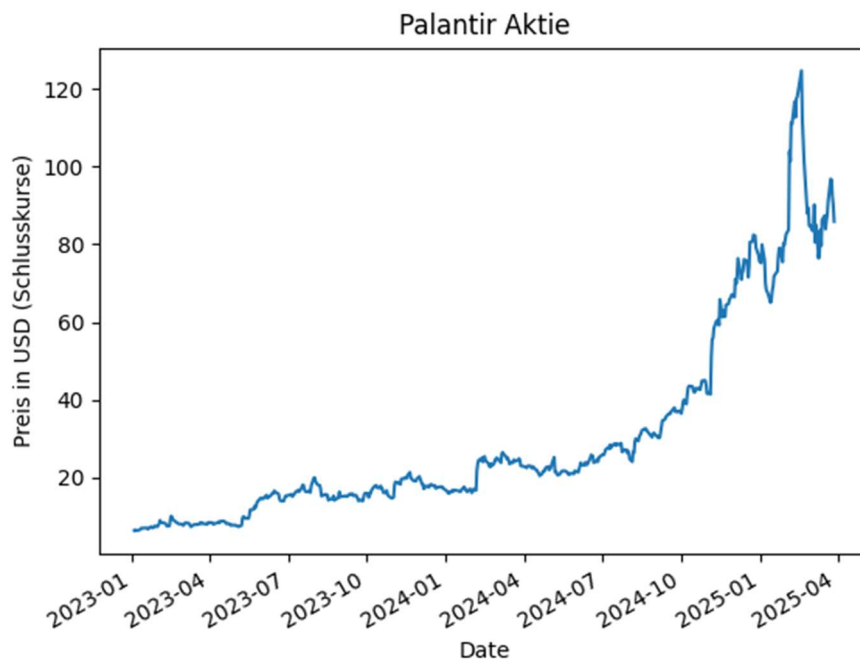
Übungen

Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_4_PLTR.csv»
- Plote die Schlusskurse mit matplotlib

Ausgabe



Musterlösung

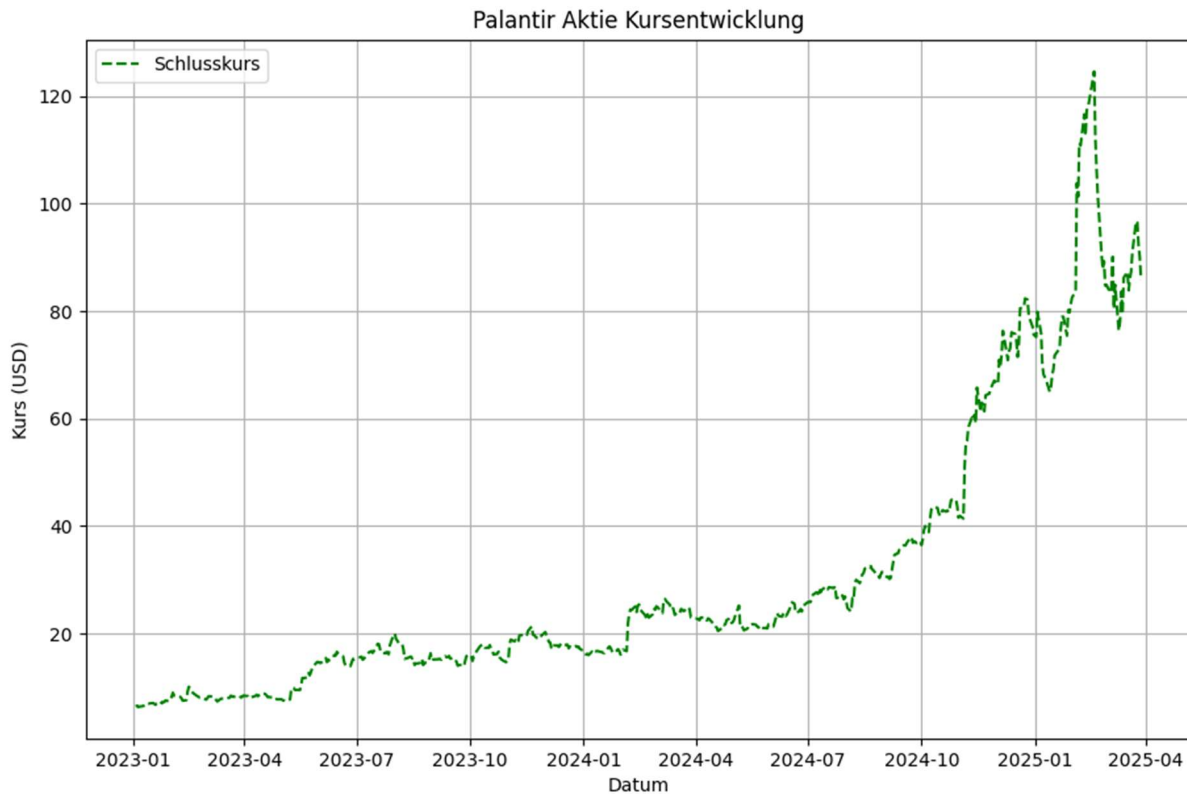
```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import plotly.graph_objects as go #für Expert Aufgabe
4
5 # Aufgabe Basic
6 df = pd.read_csv("Lektion_4_PLTR.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
7 df.set_index("Date", inplace=True) # Datum als Index setzen
8 df["Close"] = pd.to_numeric(df["Close"], errors="coerce") # Umwandlung in float
9 df["Close"].plot(title="Palantir Aktie", ylabel="Preis in USD (Schlusskurse)") # Liniendiagramm erstellen
10 plt.show() # Liniendiagramm für Schlusskurs anzeigen
11
```

Advanced Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_4_PLTR.csv»
- Plote die Schlusskurse mit matplotlib
- Gestalte dein Diagramm individuell (Titel, Farben, Linienstil, Achsenbeschriftung)

Ausgabe



Musterlösung

```

12 # Aufgabe Advanced
13 df = pd.read_csv("Lektion_4_PLTR.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
14 df.set_index("Date", inplace=True) # Datum als Index setzen
15 df["Close"] = pd.to_numeric(df["Close"], errors="coerce") # Umwandlung in float
16 plt.plot(df["Close"], color="green", linestyle="--", label="Schlusskurs") # Liniendiagramm manuell gestalten
17 plt.title("Palantir Aktie Kursentwicklung")
18 plt.xlabel("Datum") # X-Achse: Datum
19 plt.ylabel("Kurs (USD)") # Y-Achse: Preis in USD
20 plt.legend() # Legende anzeigen
21 plt.grid(True) # Gitternetz anzeigen
22 plt.show()

```

Expert Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_4_PLTR.csv»
- Erstelle ein interaktives Candlestick-Chart mit `plotly.graph_objects` (OHLC).

Ausgabe



Musterlösung

```

24 # Aufgabe Expert
25 df = pd.read_csv("Lektion_4_PLTR.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
26 fig = go.Figure(data=go.Candlestick( # Interaktives Candlestick-Diagramm mit OHLC-Werten erstellen
27     x=df.index,                # X-Achse: Datum
28     open=df["Open"],            # Eröffnungskurs
29     high=df["High"],            # Tageshöchstkurs
30     low=df["Low"],              # Tagestiefstkurs
31     close=df["Close"]          # Schlusskurs
32 ))
33 fig.update_layout( # Layout mit Achsentiteln ergänzen
34     title="Palantir Candlestick Chart",
35     xaxis_title="Datum",
36     yaxis_title="Preis (USD)"
37 )
38 fig.show() # Diagramm anzeigen

```

Abbildung 35: Lektion 4 - Niveau-differenzierte Übungen inkl. Musterlösungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Was ist der Vorteil eines Liniendiagramms?	Ein Liniendiagramm ist einfach zu erstellen und gut lesbar. Es zeigt den Verlauf des Schlusskurses über die Zeit und eignet sich besonders gut, um Trends oder Kursentwicklungen schnell zu erkennen. Es hilft beim Vergleich verschiedener Zeiträume oder Aktien, da es übersichtlich bleibt.
2.	Welche Information bietet dir ein Candlestick zusätzlich?	Ein Candlestick-Chart zeigt mehr Informationen als ein Liniendiagramm, nämlich: • Eröffnungskurs (Open) • Tageshöchstkurs (High) • Tageriefstkurs (Low) • Schlusskurs (Close) Damit sehe ich, wie stark der Kurs an einem Tag geschwankt hat, ob es Ausreisser oder Wendepunkte gab und ob der Kurs im Laufe des Tages gestiegen oder gefallen ist. Diese Details sehe ich im Liniendiagramm nicht.
3.	Wann würde man welche Darstellung verwenden?	Ein Liniendiagramm verwende ich, wenn ich einen einfachen Überblick über den Kursverlauf möchte – z.B. für langfristige Trends. Ein Candlestick-Chart ist sinnvoll, wenn ich Intraday-Bewegungen oder Schwankungen innerhalb eines Tages analysieren will – z.B. beim Trading oder zur Detailanalyse.

Abbildung 36: Lektion 4 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

3.2.2 Lektion 5: Datenglättung mittels gleitender Durchschnitte zur Trendanalyse

In dieser Lektion wird der Simple Moving Average (SMA) und der Exponential Moving Average (EMA) eingeführt. Beide Durchschnitte dienen der Trendglättung in Zeitreihen. Der SMA berechnet den Durchschnitt über ein festes Zeitfenster und reagiert gleichmässig auf neue Daten. Der EMA gewichtet neuere Daten stärker und reagiert dadurch schneller auf Kursänderungen.

Die Auswahl dieser beiden Methoden erlaubt es, mathematisch einfache, aber in der Finanzanalyse stark genutzte Werkzeuge einzuführen. Sie sind anschaulich, gut visualisierbar und leicht programmierbar mit `.rolling().mean()` und `.ewm().mean()` in pandas.

Didaktisch bietet diese Lektion einen idealen Übergang von der reinen Darstellung (Lektion 4) zur bewussten Interpretation von Trends. Die Schüler:innen lernen, wie Daten geglättet und analysiert werden können, um Schwankungen besser zu verstehen. Durch verschiedene Fenstergrössen erkennen sie, wie sich kurzfristige und langfristige Trends unterscheiden – eine Kompetenz, die später für Prognosemodelle und Markteinschätzung wichtig wird.

Lernziele

Niveau	Lernziele
Basic	- Ich kann einen einfachen SMA (z.B. 5 Tage) mit <code>.rolling().mean()</code> berechnen. - Ich kann einen SMA zusammen mit dem Schlusskurs grafisch darstellen.
Advanced	- Ich kann den Unterschied zwischen SMA und EMA erkennen, berechnen und grafisch vergleichen. - Ich kann Aussagen zur Glättung und Reaktionsgeschwindigkeit treffen.
Expert	- Ich kann mehrere SMA-Werte mit verschiedenen Fenstergrössen berechnen. - Ich kann mögliche Crossovers erkennen und als Signal für Trendwechsel interpretieren.

Abbildung 37: Lektion 5 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 38: Lektion 5 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Warum interessiert uns nicht nur der Tageskurs, sondern der Verlauf über Zeit?	Weil Trends erst über längere Zeit sichtbar werden.
2.	Was könnte ein „geglätteter“ Kursverlauf zeigen?	Langfristige Entwicklungen ohne kurzfristige Störungen.
3.	Warum könnte ein Investor auf einen gleitenden Durchschnitt schauen?	Um zu erkennen, ob sich ein Kauf oder Verkauf lohnt.

Abbildung 39: Lektion 5 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
SMA	Simple Moving Average; z.B. 5-Tages-Durchschnitt Gleitender Durchschnitt mit gleichgewichteten Werten, Python: <code>df.rolling().mean()</code>
<code>.rolling()</code>	Erzeugt ein gleitendes Fenster zur Berechnung von Durchschnittswerten über mehrere Zeilen hinweg.
<code>.mean()</code>	Berechnet den Durchschnitt innerhalb eines solchen Fensters oder allgemein über Daten.
EMA	Exponential Moving Average Gleitender Durchschnitt mit stärkerer Gewichtung der aktuelleren Werte. Python: <code>df.ewm().mean()</code>
<code>.ewm()</code>	Erzeugt ein exponentiell gewichtetes Fenster
<code>.ewm(span = 5, adjust = False))</code>	Je höher der span, desto glatter der EMA adjust=False = nutzt rekursive Berechnung (praxisnäher)

Abbildung 40: Lektion 5 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

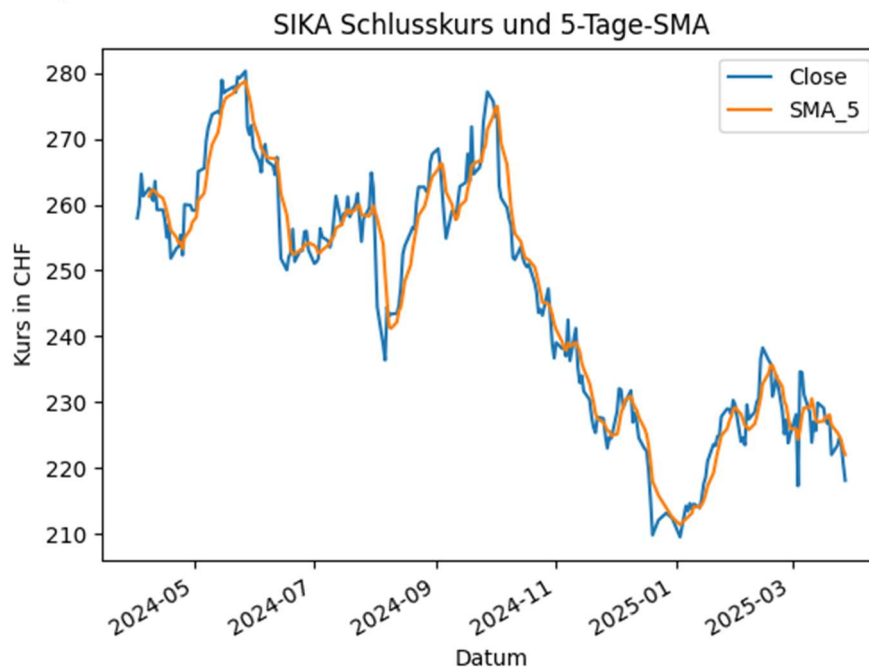
Übungen

Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_5_SIKA.csv»
- Berechne einen 5-Tage-SMA
- Zeige den 5-Tage-SMA zusammen mit dem Close-Kurs im Liniendiagramm.

Ausgabe

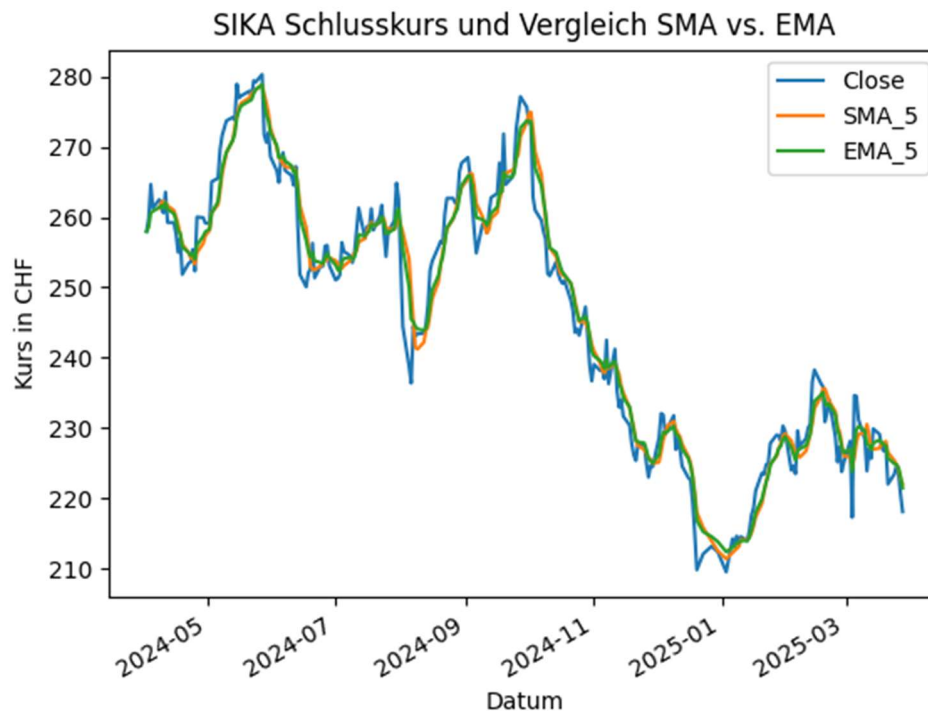


Musterlösung

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Aufgabe Basic
5 df = pd.read_csv("Lektion_5_SIKA.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
6 df.set_index("Date", inplace=True)
7 df["Close"] = pd.to_numeric(df["Close"], errors="coerce") # Umwandlung in float
8 df["SMA_5"] = df["Close"].rolling(window=5).mean() # Berechnung des 5-Tage Simple Moving Average (SMA)
9 df[["Close", "SMA_5"]].plot(title="SIKA Schlusskurs und 5-Tage-SMA") # Kursverlauf und SMA gemeinsam darstellen
10 plt.ylabel("Kurs in CHF")
11 plt.xlabel("Datum")
12 plt.show()
```

Advanced Aufgabe / Ausgabe / Musterlösung**Aufgabe**

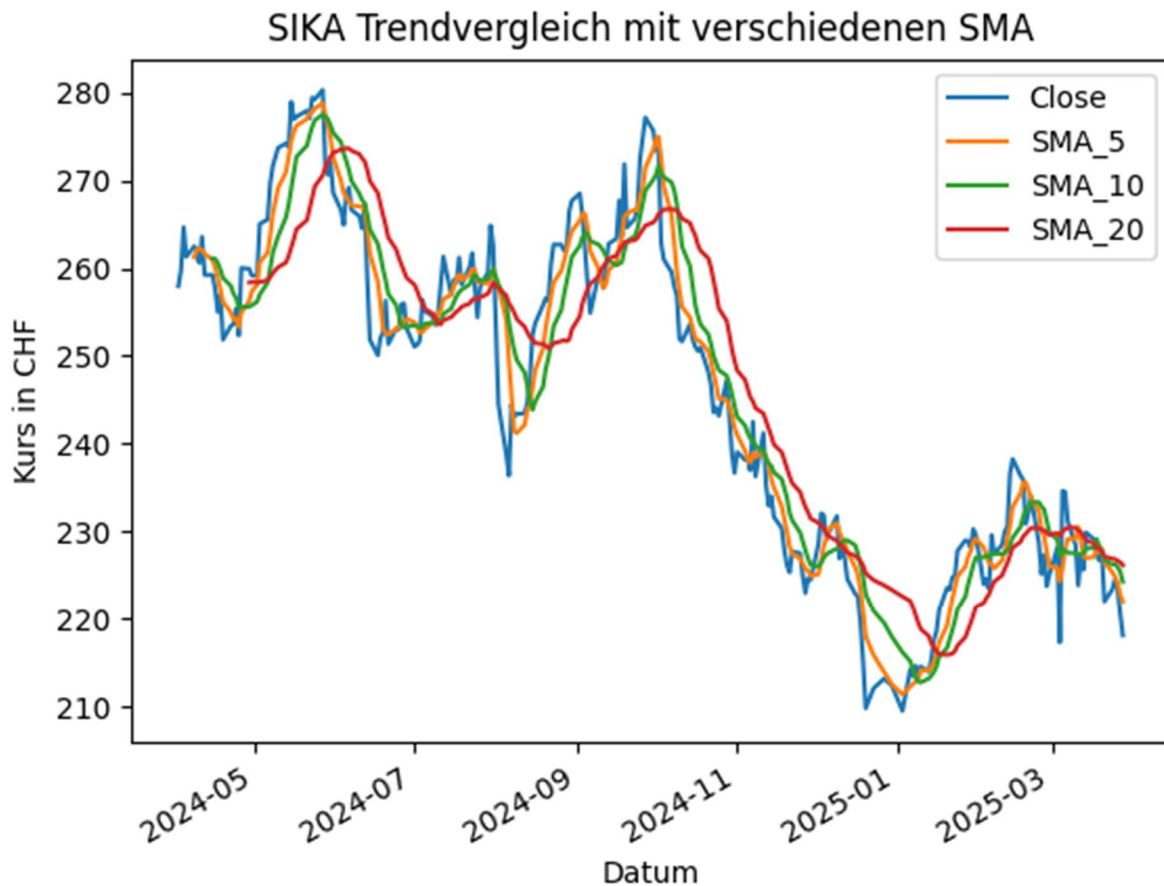
- Lade «Lektion_5_SIKA.csv»
- Vergleiche SMA (5) mit EMA (5), stelle beide grafisch dar und diskutiere den Unterschied.

Ausgabe**Musterlösung**

```
13  
14 # Aufgabe Advanced  
15 # Berechnung des Exponential Moving Average (EMA) mit gleichem Fenster  
16 df["EMA_5"] = df["Close"].ewm(span=5, adjust=False).mean()  
17 # Darstellung von Kurs, SMA und EMA  
18 df[["Close", "SMA_5", "EMA_5"]].plot(title="SIKA Schlusskurs und Vergleich SMA vs. EMA")  
19 plt.ylabel("Kurs in CHF")  
20 plt.xlabel("Datum")  
21 plt.show()
```

Expert Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Lade «Lektion_5_SIKA.csv»
- Berechne SMA mit 3 Fenstergrößen (z.B. 5, 10, 20) und analysiere Trendlinien und mögliche „Crossovers“.

Ausgabe**Musterlösung**

```

23 # Aufgabe Expert
24 # Weitere SMA-Linien mit unterschiedlichen Fenstergrößen berechnen
25 df["SMA_5"] = df["Close"].rolling(window=5).mean()
26 df["SMA_10"] = df["Close"].rolling(window=10).mean()
27 df["SMA_20"] = df["Close"].rolling(window=20).mean()
28 # Darstellung aller SMA-Linien im Vergleich zum Kursverlauf
29 df[["Close", "SMA_5", "SMA_10", "SMA_20"]].plot(title="SIKA Trendvergleich mit verschiedenen SMA")
30 plt.ylabel("Kurs in CHF")
31 plt.xlabel("Datum")
32 plt.show()

```

Abbildung 41: Lektion 5 - Niveau-differenzierte Übungen inkl. Musterlösungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Was ist der Unterschied zwischen SMA und EMA?	Der SMA berechnet den Durchschnitt gleichmässig über ein Zeitfenster, z.B. die letzten 5 Tage. Der EMA gewichtet die jüngeren Werte stärker, reagiert also schneller auf Kursveränderungen. Somit ist der EMA sensibler und der SMA stabiler.
2.	Welche Rolle spielt die Fenstergrösse bei der Trendanalyse?	Je grösser das Fenster, desto glatter wird die Durchschnittslinie und desto weniger empfindlich ist sie gegenüber kurzfristigen Schwankungen. Kleine Fenster zeigen kurzfristige Trends, grosse Fenster zeigen langfristige Entwicklungen.
3.	Was sagt dir eine steigende, fallende oder kreuzende Trendlinie?	Eine steigende Trendlinie zeigt einen möglichen Aufwärtstrend. Eine fallende Linie deutet auf einen Abwärtstrend hin. Wenn zwei Trendlinien sich kreuzen (z.B. SMA kurz / SMA lang), kann das ein Signal für eine Trendwende sein.

Abbildung 42: Lektion 5 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

3.2.3 Lektion 6: Mittelwert und SMA als Funktion programmieren und plotten

Diese Lektion erweitert Lektion 5. Sie vermittelt den Mittelwert und den gleitenden Durchschnitt SMA nicht nur rechnerisch, sondern auch visuell verständlich. Ziel dieser Lektion ist ein verbessertes Verständnis für geglättete Daten im Vergleich zu Rohdaten.

Lernziele

Niveau	Lernziele
Basic	Ich kann den Mittelwert berechnen und als horizontale Linie im Liniendiagramm darstellen.
Advanced	Ich kann den SMA(3) mit einer Schleife berechnen und grafisch darstellen.
Expert	Ich kann eine eigene SMA-Funktion mit Fensterparameter programmieren und interpretieren.

Abbildung 43: Lektion 6 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 44: Lektion 6 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage	Antwort
1. Warum ist es sinnvoll, den Mittelwert von Aktienkursen über mehrere Tage zu berechnen?	Weil einzelne Tageswerte schwanken. Der Mittelwert zeigt, wo der Kurs insgesamt steht.
2. Was könnten Gründe für starke Schwankungen bei einem Kurs sein?	Nachrichten, Ereignisse, Gerüchte oder Wirtschaftsdaten.
3. Warum könnten manche Anleger:innen bewusst in schwankende Werte investieren?	Weil dort hohe Gewinne möglich sind – aber auch Verluste.

Abbildung 45: Lektion 6 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
sum()	Addiert alle Elemente einer Liste oder Spalte.
len()	Gibt die Anzahl Elemente einer Liste zurück z.B. Anzahl Kurswerte.
Mittelwert (arithmetisches Mittel)	Summe aller Werte geteilt durch deren Anzahl Zeigt den typischen Durchschnitt.
SMA (einfacher gleitender Durchschnitt)	Durchschnitt über ein Zeitfenster, das sich von Tag zu Tag weiterbewegt.
Fenstergrösse (bei SMA)	Die Anzahl der Tage, über die jeweils der Durchschnitt berechnet wird (z. B. SMA(3) = 3 Tage).

Abbildung 46: Lektion 6 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

Übungen

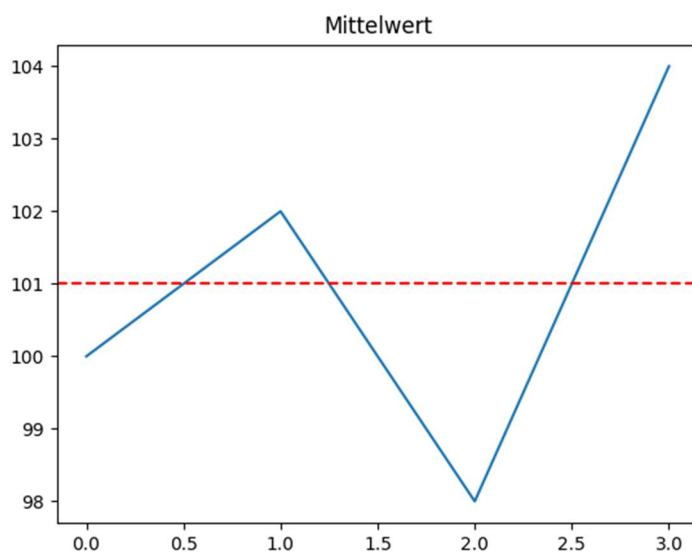
Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Berechne den Mittelwert der Liste [100, 102, 98, 104]
- Stelle die Werte als Liniendiagramm mit matplotlib.pyplot dar
- Füge den Mittelwert als rote gestrichelte Linie mittels axhline ein.

Ausgabe

```
Kommandozeile x  
>>> %Run 'Lektion 6.py'  
Mittelwert: 101.0
```



Musterlösung

```
1 import matplotlib.pyplot as plt  
2  
3 # Aufgabe Basic  
4 werte = [100, 102, 98, 104]  
5 mittelwert = sum(werte) / len(werte) # Berechnung Mittelwert  
6 print("Mittelwert:", mittelwert)  
7  
8 plt.plot(werte) # Kursreihe darstellen  
9 plt.axhline(mittelwert, color="red", linestyle="--") # Linie einzeichnen  
10 plt.title("Mittelwert")  
11 plt.show()
```

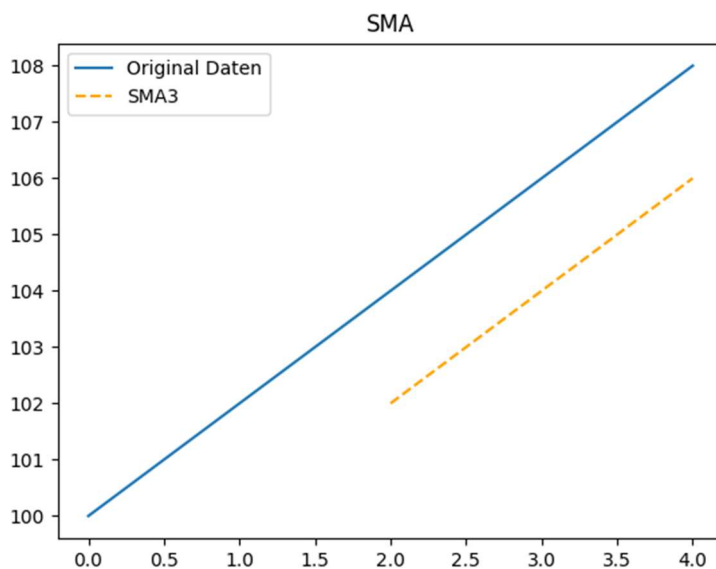
Advanced Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Berechne den 3-Tage-SMA der Liste [100, 102, 104, 106, 108] mit Hilfe einer for-Schleife.
- Stelle die Werte als Liniendiagramm mit matplotlib.pyplot dar
- Füge den SMA graphisch als orange gestrichelte Linie ein.

Ausgabe

```
>>> %Run 'Lektion 6.py'
```

```
102.0  
104.0  
106.0
```

**Musterlösung**

```
13  
14 # Aufgabe Advanced  
15 daten = [100, 102, 104, 106, 108]  
16 sma = []  
17 for i in range(len(daten) - 2):  
18     fenster = daten[i:i+3] # Slicing (i, i+1, i+2, ohne i+3)  
19     sma.append(sum(fenster)/3) # Mittelwert über drei Werte  
20     print((sum(fenster)/3))  
21 plt.plot(daten, label="Original Daten")  
22 plt.plot([2, 3, 4], sma, label="SMA3", color = "orange", linestyle="--")  
23 plt.title("SMA")  
24 plt.legend()  
25 plt.show()  
26
```

Expert Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Programmiere eine Funktion sma(liste, fenster) zur Berechnung der gleitenden Durchschnitte mit Hilfe einer for-Schleife.
- Rufe deine Funktion mit sma([100, 102, 104, 106, 108], 3) auf und gib sie auf der Konsole aus.

Ausgabe

Kommandozeile ×

```

>>> %Run 'Lektion 6.py'
[102.0, 104.0, 106.0]
```

Musterlösung

```

27 # Aufgabe Expert
28 def sma(liste, f):
29     resultate = [] # Leere Liste für Ergebnisse
30     for i in range(len(liste) - f + 1):
31         fenster = liste[i:i+f] # Aktuelles Fenster Slicing
32         durchschnitt = sum(fenster) / f # Mittelwert berechnen
33         resultate.append(durchschnitt) # Ergebnis zur Liste hinzufügen
34     return resultate
35
36 print(sma([100, 102, 104, 106, 108], 3)) # → [102.0, 104.0, 106.0]
```

Abbildung 47: Lektion 6 - Niveau-differenzierte Übungen inkl. Musterlösungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Was zeigt dir ein gleitender Durchschnitt besser als Rohdaten?	Der SMA zeigt den allgemeinen Trend, ohne dass einzelne Ausreisser zu stark ins Gewicht fallen. Beispiel: Ein plötzlicher Kurseinbruch beeinflusst den SMA weniger als den Tageskurs.
2.	Warum wäre eine Börsenentscheidung auf Einzelwerten gefährlich?	Einzelwerte können stark schwanken und irreführend sein. Wer nur auf Tagesbewegungen achtet, könnte impulsiv und ohne langfristigen Überblick handeln.
3.	Welche Rolle spielen Fenstergrößen bei der Glättung?	Je grösser das Fenster, desto stärker wird geglättet. Dafür sind aber kurzfristige Entwicklungen weniger sichtbar. Beispiel: SMA(3) reagiert schneller. SMA(20) ist stabiler aber träger.

Abbildung 48: Lektion 6 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

3.2.4 Formativer Mini-Check am Ende des Kompetenzblockes 2

Am Ende des Kompetenzblockes wird ein formativer Mini-Check anhand von je 5 Kurzfragen pro Niveau Basic, Advanced und Expert durchgeführt. Diese formative Lernzielkontrolle fand auf Classtime digital statt. Sie kann mittels folgendem Link angeschaut werden: <https://www.classtime.com/student/login/ZJ3QD2R>. Die Schüler:innen lösen den Mini-Check in einer sicheren Umgebung im Safe Exam Browser. Die Auswertung des Mini-Checks ist in Kapitel 5.2.1 Auswertung formative Mini-Checks. Der Mini-Check umfasst folgende 15 Fragen:

MEINE FRAGEN SESSIONS KLASSEN

Was ist neu? Hilfe Patrick Suter Institutionelle Lizenz

Meine Fragen > Mini-Check Kompetenzblock 2: Datenvisualisierung und -interpretation [Neue Session aktivieren](#)

Dieses Fragenset ist nur für Sie sichtbar. Leisten Sie einen Beitrag, indem Sie es veröffentlichen. [Veröffentlichen](#)

Frage	Punkte	Aktionen
1 Basic 1: Welche Methode erzeugt ein einfaches Liniendiagramm?	1 Pkte.	Zur Session hinzufügen
2 Basic 2: Wofür steht plt.ylabel("Preis")	1 Pkte.	
3 Basic 3: Wie erstellt man ein Diagramm mit matplotlib?	1 Pkte.	
4 Basic 4: Was bewirkt plt.title("Titel")?	1 Pkte.	
5 Basic 5: Welche Bibliothek nutzt man für einfache Diagramme?	1 Pkte.	
6 Advanced 1: Was ist ein SMA(5)?	1 Pkte.	
7 Advanced 2: Welche Aussage über EMA stimmt?	1 Pkte.	
8 Advanced 3: Wie stellt man SMA und EMA gemeinsam dar?	1 Pkte.	
9 Advanced 4: Was zeigt ein Crossover von SMA(5) und SMA(20)?	1 Pkte.	
10 Advanced 5: Was bewirkt df["SMA"] = df["Close"].rolling(5).mean()?	1 Pkte.	
11 Expert 1: Wofür steht Candlestick in plotly.graph_objects?	1 Pkte.	
12 Expert 2: Was zeigt ein Candlestick zusätzlich zum Liniendiagramm?	1 Pkte.	
13 Expert 3: Was ist ein Vorteil von plotly gegenüber matplotlib?	1 Pkte.	
14 Expert 4: Was bedeutet „OHLC“?	1 Pkte.	
15 Expert 5: Wann ist ein SMA-Crossover relevant?	1 Pkte.	

+ Neue Frage
Vorschau / Übungslink
Veröffentlichen
Teilen
Kopieren nach ...
Verschieben nach ...
Ausdrucken
Papierkorb

Abbildung 49: Formativer Mini-Check nach Kompetenzblock 2

3.3 Kompetenzblock 3: Datenbasierte Risiko- und Marktanalyse

In Kompetenzblock 3 analysieren die Schüler:innen Kursschwankungen, Volatilität und Korrelation. Sie lernen die Konzepte Standardabweichung und relative Volatilität kennen. Sie setzen diese ins Verhältnis zu Marktindizes und programmieren eigene Analysefunktionen. Im Fokus steht die Entwicklung eines Verständnisses für finanzielle Risiken, Marktabhängigkeiten und Streuungsmasse – sowohl rechnerisch als auch visuell.

3.3.1 Lektion 7: Datenvolatilität (Standardabweichung, Beta) gegenüber Börsenindizes

Diese Lektion verknüpft Statistik, Wirtschaft und Datenanalyse auf verständliche Weise. Reale Daten und mathematische Grössen helfen den Schüler:innen bei der Risikoabschätzung einer Einzelaktie. Das Risiko soll nicht nur gefühlsmässig, sondern auch rechnerisch bewertet werden. Es werden die zwei Methoden `pandas.std()` für die Standardabweichung und `pandas.corr()` für die Korrelation eingeführt. Die Konzepte sind mathematisch gemäss Vorwissen zugänglich. Sie helfen für die finanzielle Bildung und das Marktverständnis.

Durch die Kombination aus Zahlen, Vergleich und Interpretation wird ein differenziertes Risikobewusstsein gefördert. Besonders die Expert-Aufgabe schafft eine Brücke zur Kapitalmarkttheorie ($\text{Beta} \approx \text{Sensitivität zur Markttrendite}$), ohne formal zu werden.

Lernziele

Niveau	Lernziele
Basic	- Ich kann den Mittelwert und die Standardabweichung von Kursdaten berechnen. - Ich kann die relative Volatilität als Verhältnis von Standardabweichung zu Mittelwert berechnen. - Ich kann Schwankungen einer Aktie quantifizieren.
Advanced	- Ich kann zwei Kursreihen kombinieren und NaN-Werte bereinigen. - Ich kann die Korrelation zwischen einer Aktie und einem Index berechnen. - Ich kann die Korrelation inhaltlich als Beta-ähnliches Mass interpretieren.
Expert	- Ich kann Zeitreihen mit <code>.resample("M")</code> monatsweise gruppieren. - Ich kann die relative Volatilität für jeden Monat berechnen und grafisch darstellen. - Ich kann Unterschiede in der Volatilität über Zeit hinweg analysieren und deuten.

Abbildung 50: Lektion 7 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)

20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 51: Lektion 7 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Was bedeutet für dich ein sicherer Kursverlauf?	Wenige Schwankungen, eher stabil.
2.	Was könnten Gründe für starke Schwankungen bei einem Kurs sein?	Nachrichten, Ereignisse, Gerüchte oder Wirtschaftsdaten.
3.	Warum könnten manche Anleger:innen bewusst in schwankende Werte investieren?	Weil dort hohe Gewinne möglich sind – aber auch Verluste.

Abbildung 52: Lektion 7 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
.std()	Berechnet die Standardabweichung eines Datenfeldes.
Volatilität (Konzept)	Mass für die Schwankungsintensität eines Kurses: je höher, desto riskanter.
Relative Volatilität (Konzept)	Verhältnis von Standardabweichung zu Mittelwert Erlaubt den Vergleich zwischen Werten mit unterschiedlichen Niveaus.
Kursrisiko	Die Unsicherheit über die zukünftige Entwicklung eines Kurswertes.
Streuung grafisch interpretieren	Veranschaulichung, wie stark Werte um einen Mittelwert „verteilt“ sind.

Abbildung 53: Lektion 7 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Was zeigt dir die Standardabweichung eines Kurses?	Sie misst, wie stark der Kurs um seinen Mittelwert schwankt. Je höher, desto volatil. Allerdings sollte immer auch das Niveau des Mittelwertes berücksichtigt werden. Deshalb ist die relative Standardabweichung aussagekräftiger.
2.	Was bedeutet es, wenn eine Aktie ein hohes Beta	Die Aktie schwankt stärker als der Markt. Sie ist risikoreicher und reagiert empfindlicher auf Marktveränderungen.

	gegenüber dem Markt hat?	
3.	Was sagt dir eine Korrelation nahe 1 oder -1?	Nahe 1 = beide Kurse verlaufen sehr ähnlich. Nahe -1 = entgegengesetzter Verlauf. Nahe 0 = kein Zusammenhang.

Abbildung 54: Lektion 7 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Übungen

Basic Aufgabe / Ausgabe / Musterlösung
Aufgabe - Lade «Lektion_7_NVDA.csv» für die Nvidia-Aktie - Berechne die Standardabweichung als absolute Volatilität und der Mittelwert der Close-Werte für die Nvidia-Aktie. - Berechne die relative Volatilität als Standardabweichung / Mittelwert anhand der Close-Werte.
Ausgabe <pre> Nvidia-Aktienanalyse Durchschnittlicher Kurs: 119.8 Standardabweichung: 17.76 Relative Volatilität: 14.82 % >>> </pre>
Musterlösung <pre> 1 import pandas as pd 2 3 Aufgabe Basic 4 df = pd.read_csv("Lektion_7_NVDA.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen 5 df.set_index("Date", inplace=True) 6 df["Close"] = pd.to_numeric(df["Close"], errors="coerce") # Umwandlung in float 7 vola = df["Close"].std() #Standardabweichung (Volatilität) berechnen 8 mean = df["Close"].mean() # Mittelwert des Kurses berechnen 9 rel_vola = (vola / mean) * 100 # Relative Volatilität in Prozent 10 print("Nvidia-Aktienanalyse") 11 print("Durchschnittlicher Kurs:", round(mean, 2)) 12 print("Standardabweichung:", round(vola, 2)) 13 print("Relative Volatilität:", round(rel_vola, 2), "%") 14 </pre>

Advanced Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Lade «Lektion_7_ZURN.csv» und «Lektion_7_SMI.csv»
- Kombiniere die Daten zu einem gemeinsamen DataFrame
- Lösche NaN-Zeilen
- Berechne die Korrelation zwischen der Aktie und dem Index.
- Interpretiere die Korrelation ähnlich wie «Beta»

Ausgabe

```
Korrelation Zurich - SMI: 0.537
```

```
>>>
```

Interpretation: Zurich Aktie schwankt in gleicher Richtung wie der Markt, jedoch nur etwa halb so stark.

Musterlösung

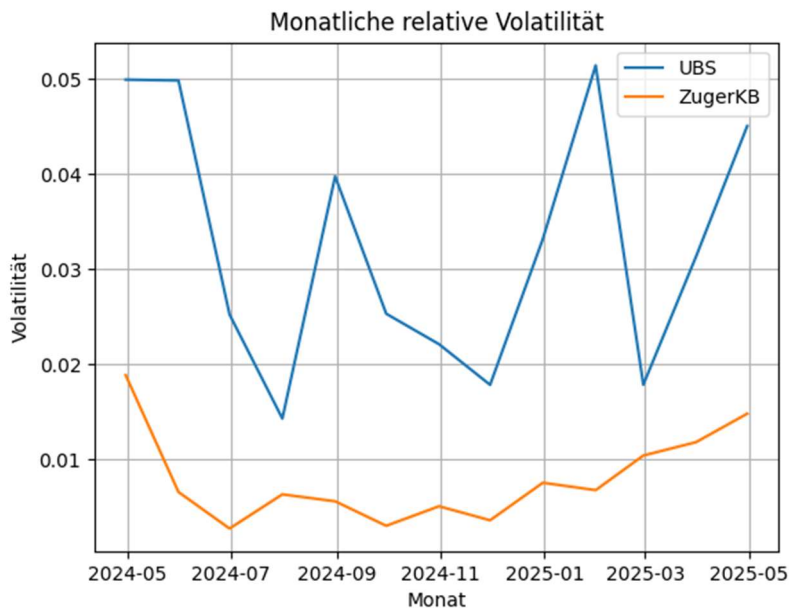
```
15
16 # Aufgabe Advanced
17 zurn = pd.read_csv("Lektion_7_ZURN.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
18 zurn.set_index("Date", inplace=True)
19 zurn["Close"] = pd.to_numeric(zurn["Close"], errors="coerce") # Umwandlung in float
20 smi = pd.read_csv("Lektion_7_SMI.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
21 smi.set_index("Date", inplace=True)
22 smi["Close"] = pd.to_numeric(smi["Close"], errors="coerce") # Umwandlung in float
23
24 # Daten zu einem gemeinsamen DataFrame kombinieren und NaN-Zeilen entfernen
25 df_vergleich = pd.DataFrame({ "Zurich": zurn["Close"], "SMI": smi["Close"] }).dropna()
26
27 # Korrelation berechnen: Wie ähnlich verlaufen Apple und SMI?
28 korrelation = df_vergleich.corr().loc["Zurich", "SMI"]
29 print("Korrelation Zurich - SMI:", round(korrelation, 3))
30
```

Expert Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Lade «Lektion_7_UBS.csv» und «Lektion_7_ZUGERKB.csv»
- Gruppiere die Daten monatsweise mit `resample(«M»)`
- Berechne für zwei Aktien die relative Volatilität pro Monat
- Stelle diese in einem Diagramm dar.
- Interpretation: Welche Aktie ist wann volatiler?

Ausgabe



Musterlösung

```

31
32 # Aufgabe Expert
33 ubs = pd.read_csv("Lektion_7_UBS.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
34 ubs.set_index("Date", inplace=True)
35 ubs["Close"] = pd.to_numeric(ubs["Close"], errors="coerce") # Umwandlung in float
36 zugerkb = pd.read_csv("Lektion_7_ZUGERKB.csv", parse_dates=["Date"]) # CSV-Datei laden und Datum parsen
37 zugerkb.set_index("Date", inplace=True)
38 zugerkb["Close"] = pd.to_numeric(zugerkb["Close"], errors="coerce") # Umwandlung in float
39
40 # Monatsweise Gruppierung und Berechnung der relativen Volatilität
41 ubs_vola = ubs["Close"].resample("ME").std()/ubs["Close"].resample("ME").mean()
42 zugerkb_vola = zugerkb["Close"].resample("ME").std()/zugerkb["Close"].resample("ME").mean()
43
44 # Plot des monatlichen Risikoverlaufs
45 import matplotlib.pyplot as plt
46 plt.plot(ubs_vola, label="UBS")
47 plt.plot(zugerkb_vola, label="ZugerKB")
48 plt.title("Monatliche relative Volatilität")
49 plt.ylabel("Volatilität")
50 plt.xlabel("Monat")
51 plt.legend()
52 plt.grid(True)
53 plt.xticks()
54 plt.show()

```

Abbildung 55: Lektion 7 - Niveau-differenzierte Übungen inkl. Musterlösungen

3.3.2 Lektion 8: Standardabweichung als Funktion programmieren und plotten

In dieser Lektion lernen die Schüler:innen, wie man Schwankungen in Daten analysiert. Die Standardabweichung wird als Mass für Streuung erklärt, grafisch dargestellt und per Hand berechnet. Ziel ist es, ein Gefühl für Volatilität als Risikoindikator zu entwickeln.

Lernziele

Niveau	Lernziele
Basic	Ich kann grafisch zeigen, wie weit Werte vom Mittelwert abweichen.
Advanced	Ich kann die Standardabweichung mittels Programmcode berechnen.
Expert	Ich kann eine Standardabweichungsfunktion selbst schreiben.

Abbildung 56: Lektion 8 – Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 57: Lektion 8 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage	Antwort
1. Was bedeutet es, wenn ein Kurs ruhig verläuft?	Die Werte sind nahe beieinander. Sie schwanken wenig.
2. Was könnte dir helfen, die Stabilität eines Kurses zu messen?	Eine Streuungsmasszahl wie die Standardabweichung.
3. Wie kannst du im Diagramm erkennen, ob ein Kurs stark streut?	Die Punkte liegen weit vom Mittelwert entfernt oder schwanken stark auf und ab.

Abbildung 58: Lektion 8 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
Abweichung vom Mittelwert	Differenz eines Wertes zum Mittelwert: $x - \mu$.
Quadratische Abweichung	Die Abweichung im Quadrat: $(x - \mu)^2$, betont starke Ausreisser.
Varianz	Der Durchschnitt der quadratischen Abweichungen als Schritt zur Standardabweichung.
Standardabweichung (σ)	Die Wurzel der Varianz; misst die durchschnittliche Streuung vom Mittelwert.
Streuung im Diagramm	Grafische Darstellung der Abweichungen z.B. durch farbliche Markierung um den Mittelwert.

Abbildung 59: Lektion 8 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

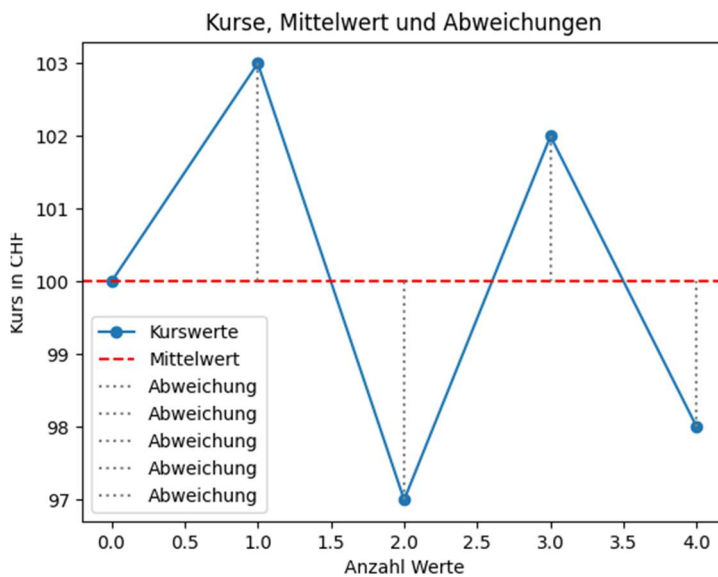
Übungen

Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Berechne den Mittelwert der Kurswerte [100, 103, 97, 102, 98].
- Stelle die Werte als Liniendiagramm mit matplotlib.pyplot dar
- Füge den Mittelwert als rote gestrichelte Linie mittels axhline ein.
- Füge die Abweichungen der Werte vom Mittelwert als vertikale Linien ein.

Ausgabe



Musterlösung

```

1 import matplotlib.pyplot as plt
2
3 # Aufgabe Basic
4 werte = [100, 103, 97, 102, 98]
5 mittelwert = sum(werte) / len(werte) # Mittelwert berechnen
6 plt.plot(werte, marker="o", label = "Kurswerte")
7 plt.axhline(mittelwert, color="red", linestyle="--", label = "Mittelwert") # Linie einzeichnen
8 for wert in range(len(werte)): # Vertikale Abweichungen zum Mittelwert darstellen
9     plt.plot([wert, wert], [werte[wert], mittelwert], color="gray", linestyle=":", label = "Abweichung")
10 plt.title("Kurse, Mittelwert und Abweichungen")
11 plt.ylabel("Kurs in CHF")
12 plt.xlabel("Anzahl Werte")
13 plt.legend()
14 plt.show()

```

Advanced Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Berechne in Python die Standardabweichung der Kurswerte [100, 103, 97, 102, 98] ohne Benützung der Funktion .std().
- Gib mindestens die Varianz und Standardabweichung auf der Konsole aus.

Ausgabe

```
>>> %Run 'Lektion 8.py'

Absolute Abweichungen: [0.0, 3.0, 3.0, 2.0, 2.0]
Summe aller Abweichungen: 10.0
Quadrierte Abweichungen: [0.0, 9.0, 9.0, 4.0, 4.0]
Summe aller quadrierten Abweichungen: 26.0
Varianz: 5.2
Standardabweichung: 2.28
```

Musterlösung

```
16 # Aufgabe Advanced
17 werte = [100, 103, 97, 102, 98]
18 mittelwert = sum(werte) / len(werte) # Mittelwert berechnen
19
20 abweichung_absolut = []
21 for wert in werte:
22     abweichung_absolut.append(abs(wert - mittelwert)) # Absolute Abweichungen berechnen
23
24 abweichung_quadriert = []
25 for wert in werte:
26     abweichung_quadriert.append((wert - mittelwert) ** 2) # Quadratische Abweichungen berechnen
27
28 varianz = sum(abweichung_quadriert) / len(werte) # Varianz = Durchschnitt der quadratischen Abweichungen
29 std = varianz ** 0.5 # Standardabweichung = Wurzel aus Varianz
30
31 print("Absolute Abweichungen: ", abweichung_absolut)
32 print("Summe aller Abweichungen: ", sum(abweichung_absolut))
33 print("Quadrierte Abweichungen: ", abweichung_quadriert)
34 print("Summe aller quadrierten Abweichungen: ", sum(abweichung_quadriert))
35 print("Varianz:", round(varianz, 2))
36 print("Standardabweichung:", round(std, 2))
```

Expert Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Programmiere eine eigene Funktion std(liste) zur Standardabweichung.
- Rufe die Funktion mit std([100, 103, 97, 102, 98]) auf und überprüfe das Resultat.

Ausgabe

```
>>> %Run 'Lektion 8.py'
2.280350850198276
```

Musterlösung

```
38 # Aufgabe Expert
39 werte = [100, 103, 97, 102, 98]
40 def std(liste):
41     mittelwert = sum(liste) / len(liste) # Mittelwert berechnen
42     abweich_quadriert = [] # Liste für quadratische Abweichungen
43     for wert in liste:
44         abweichung = (wert - mittelwert) ** 2
45         abweich_quadriert.append(abweichung)
46     varianz = sum(abweich_quadriert) / len(liste) # Durchschnitt der quadrierten Abweichungen
47     standardabweichung = varianz ** 0.5 # Wurzel ziehen
48     return standardabweichung
49 print(std([100, 103, 97, 102, 98]))
```

Abbildung 60: Lektion 8 - Niveau-differenzierte Übungen inkl. Musterlösungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage		Antwort
1.	Warum ist Standardabweichung ein besseres Risikomass als nur die Preisspanne?	Die Standardabweichung berücksichtigt alle Schwankungen und nicht nur die Extremwerte. Sie gibt an, wie stabil sich ein Kurs insgesamt verhält. Es zeigt nicht nur, wie weit oben und unten er war.
2.	Wann ist hohe Volatilität gewünscht (z.B. im Trading)?	Bei kurzfristigem Trading kann hohe Volatilität zu Gewinnchancen führen, da schnelle Kursveränderungen ausgenutzt werden können. Für langfristige Anleger:innen ist sie hingegen oft unerwünscht.
3.	Woran erkennt man Ausreisser in einer Grafik?	Wenn einzelne Werte deutlich über/unter dem Mittelwert liegen. Sogenannte Ausreisser fallen in der Visualisierung auf. Diese Abweichungen sollten hinterfragt werden: Fehler, News, externe Einflüsse?

Abbildung 61: Lektion 8 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

3.3.3 Lektion 9: Korrelation zwischen Einzelaktien und Gesamtmarkt visualisieren

Diese Lektion erweitert das Verständnis von Korrelation. So lernen die Schüler:innen die grafische Interpretation von Zusammenhängen zwischen Einzelaktien und dem Gesamtmarkt. Diese Interpretation soll funktionieren, ohne auf Formeln zurückzugreifen.

Lernziele

Niveau	Lernziele
Basic	Ich kann zwei Kursreihen grafisch als Punkte darstellen.
Advanced	Ich kann erkennen, ob eine Einzelaktie dem Markt folgt. Und den Zusammenhang beschreiben.
Expert	Ich kann reale Kursdaten einer Aktie und eines Index importieren, vergleichen und die Korrelation interpretieren.

Abbildung 62: Lektion 9 – Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
10'	Wissenserarbeitung: Theorie-Input / Gemeinsames Anwenden / Schritt-für-Schritt-Coding	- Neue Begriffe / Konzepte / Funktionen (Lehrgespräch) - Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
20-25'	Wissenssicherung: Üben und Anwenden	Aufgabenbearbeitung Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5-10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Diskussion der Lösungen zu den Aufgaben (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 63: Lektion 9 – Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage	Antwort
1. Was könnte es bedeuten, wenn zwei Aktien fast immer gleichzeitig steigen?	Dass sie miteinander verbunden sind. Sie korrelieren positiv.
2. Was wäre ein Beispiel für zwei Aktien mit schwacher Korrelation?	Eine Technologiefirma und ein Energiekonzern. Sie reagieren oft unterschiedlich.
3. Was bedeutet: „Korrelation ist nicht Kausalität“?	Nur weil zwei Kurse sich ähnlich entwickeln, heisst das nicht, dass einer den anderen beeinflusst.

Abbildung 64: Lektion 9 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Neue Begriffe / Konzepte / Funktionen

	Erklärung
Korrelation (visuell)	Grafisch erkennbare Beziehung zwischen zwei Kursen, z. B. durch ein Streudiagramm.
plt.scatter()	Erzeugt ein Punktdiagramm (Scatterplot), das zwei Datenreihen gegenüberstellt.
Interpretation von Punktwolken	Enge Punktwolke in steigender Linie = positive Korrelation; Streuung = schwache Korrelation.
Korrelation $\neq$ Kausalität (Konzept)	Nur weil zwei Werte gemeinsam steigen, bedeutet das nicht, dass einer den anderen verursacht.

Abbildung 65: Lektion 9 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen

Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Frage	Antwort
1. Wie stark hängen Wirtschaft und Börse wirklich zusammen?	Häufig ist ein solcher Zusammenhang erkennbar. Wenn das BIP steigt, steigt auch das Vertrauen in Unternehmen. So ziehen die Aktienkurse an. Bei dieser Aussage ist jedoch auch Vorsicht geboten, denn die Börse antizipiert. Sie ist also oft schneller als das BIP.
2. Wo kann Korrelation trügen?	Wenn ein Drittfaktor beide beeinflusst. Eine Zinssenkung kann sowohl das Wirtschaftswachstum anregen als auch Aktienkurse steigen lassen. Andere Drittvariable könnten politische Ereignisse, Inflation und Rohstoffpreise sein.
3. Was fehlt einem Scatterplot?	Es fehlt die Erklärung, warum etwas zusammenhängt. Korrelation $\neq$ Kausalität.

Abbildung 66: Lektion 9 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen

Übungen

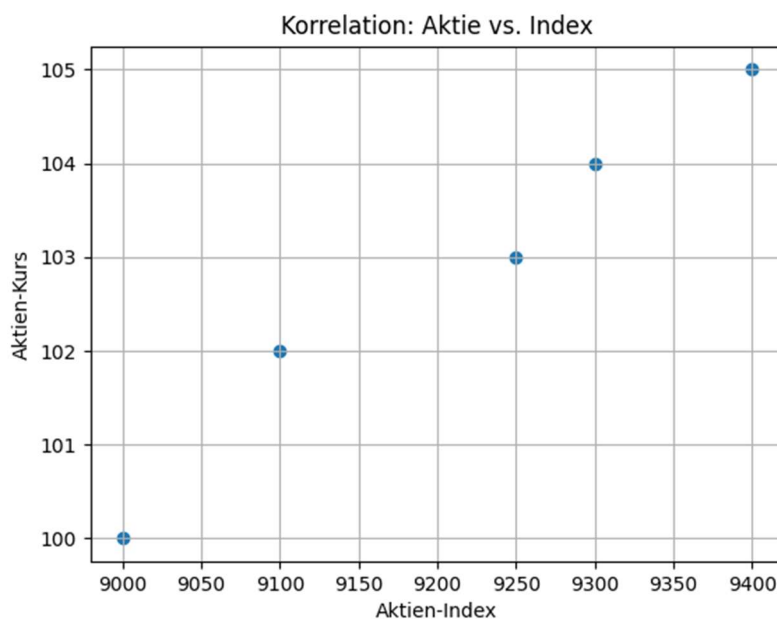
Basic Aufgabe / Ausgabe / Musterlösung

Aufgabe

- Erstelle ein Scatterplot mit zwei Listen von fiktiven Kurswerten einer Aktie und eines Index. Schätze den Zusammenhang.
- Benütze dazu den `scatter(x,y)` aus der Bibliothek `matplotlib.pyplot`.
- Plote den Index auf die x-Achse und die Aktie auf die y-Achse.

	2020	2021	2022	2023	2024
Aktie	100	102	104	103	105
Index	9000	9100	9300	9250	9400

Ausgabe



Interpretation: Die Punkte steigen gemeinsam → positive Korrelation.

Musterlösung

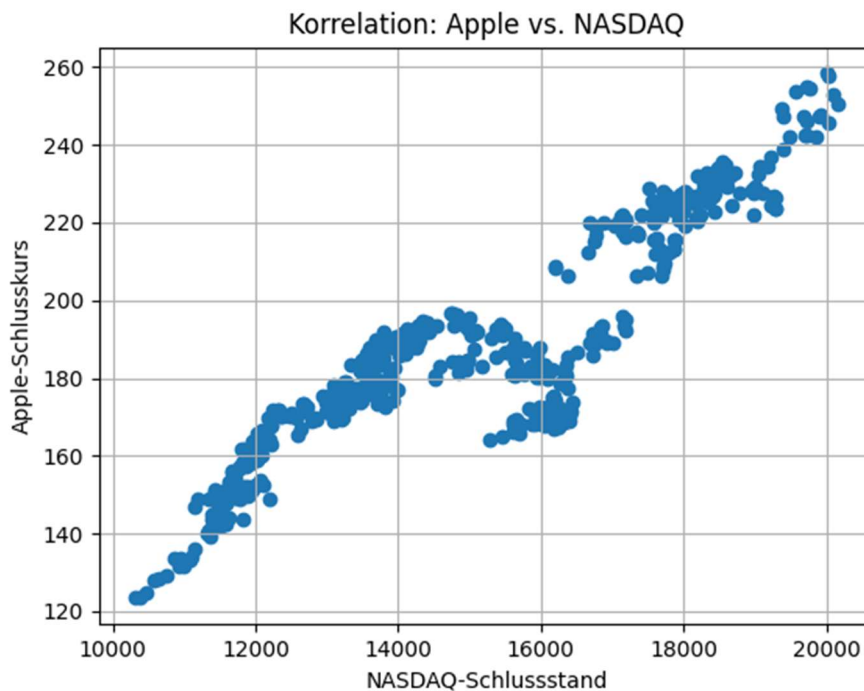
```

1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 # Aufgabe Basic
5 y = [100, 102, 104, 103, 105] # Aktie
6 x = [9000, 9100, 9300, 9250, 9400] # Index
7
8 plt.scatter(x, y)
9 plt.ylabel("Aktien-Kurs")
10 plt.xlabel("Aktien-Index")
11 plt.title("Korrelation: Aktie vs. Index")
12 plt.grid(True)
13 plt.show()
14

```

Advanced Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Lade «Lektion_9_AAPL-NASDAQ.csv» mit Tagesendkursen von Apple und dem NASDAQ Index aus dem Jahre 2023 und 2024.
- Stelle sie als Punktdiagramm dar
- Beschreibe die Korrelation

Ausgabe**Interpretation:**

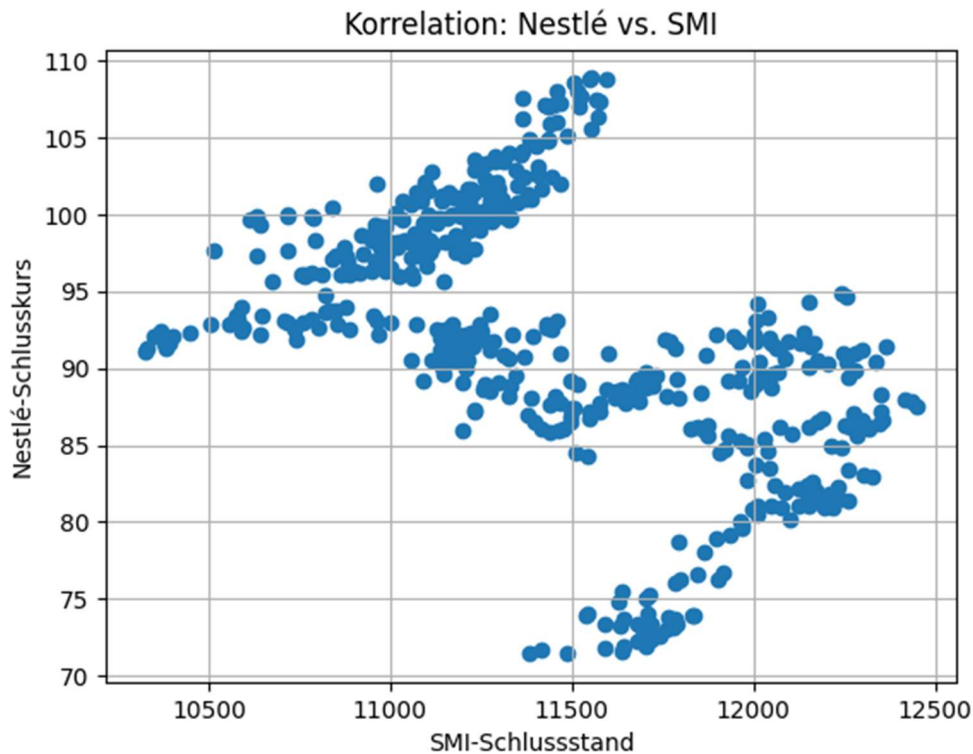
- Die Punkte liegen eng entlang einer aufsteigenden Linie. Wenn der NASDAQ steigt, steigt auch der Apple-Kurs (fast proportional). Das ist ein klassisches Beispiel für starke positive Korrelation
- Grund: Apple ist eine der grössten Aktien im NASDAQ. Apple beeinflusst den NASDAQ aktiv. Die gegenseitige Bewegung ist deshalb besonders ausgeprägt.

Musterlösung

```
14
15 # Aufgabe Advanced
16 import matplotlib.pyplot as plt
17 import pandas as pd
18 df = pd.read_csv("Lektion_9_AAPL-NASDAQ.csv") # enthält Spalten: Apple, Nasdaq
19 plt.scatter(df["NASDAQ"], df["Apple"])
20 plt.ylabel("Apple-Schlusskurs")
21 plt.xlabel("NASDAQ-Schlussstand")
22 plt.title("Korrelation: Apple vs. NASDAQ")
23 plt.grid(True)
24 plt.show()
25
```

Expert Aufgabe / Ausgabe / Musterlösung
Aufgabe

- Lade «Lektion_9_NESN-SMI.csv» mit Tagesendkursen von Nestlé und dem SMI-Index aus dem Jahre 2023 und 2024.
- Stelle sie als Punktdiagramm dar
- Beschreibe die Korrelation

Ausgabe

Interpretation:

- Die Punkte bilden keine klar erkennbare Linie, sondern zeigen mehrere verdichtete Cluster. Das deutet auf phasenweise Korrelation, aber nicht durchgehend lineare Abhängigkeit hin.
- Die Ansammlungen von Punkten deuten darauf hin, dass sowohl Nestlé als auch der SMI sich eine Zeit lang auf bestimmten Kursniveaus bewegt haben, bevor sie in einen anderen Bereich gewechselt sind. Gründe sind Marktkonsolidierung und externe Schocks (v.a. Wahlen USA und Ukraine-Krieg).

Musterlösung

```

26 # Aufgabe Expert
27 import pandas as pd
28 import matplotlib.pyplot as plt
29
30 df = pd.read_csv("Lektion_9_NESN-SMI.csv") # enthält Spalten: Nestle, SMI
31 plt.scatter(df["SMI"], df["Nestle"])
32 plt.ylabel("Nestlé-Schlusskurs")
33 plt.xlabel("SMI-Schlussstand")
34 plt.title("Korrelation: Nestlé vs. SMI")
35 plt.grid(True)
36 plt.show()

```

Abbildung 67: Lektion 9 – Niveau-differenzierte Übungen inkl. Musterlösungen

3.3.4 Formativer Mini-Check am Ende des Kompetenzblockes 3

Am Ende des Kompetenzblockes wird ein formativer Mini-Check anhand von je 5 Kurzfragen pro Niveau Basic, Advanced und Expert durchgeführt. Diese formative Lernzielkontrolle findet auf Classtime digital statt. Sie kann mittels folgendem Link angeschaut werden: <https://www.classtime.com/student/login/D6RKPJ3>. Die Schüler:innen selbst lösen den Mini-Check in einer sicheren Umgebung im Safe Exam Browser. Der Mini-Check umfasst folgende 15 Fragen:

MEINE FRAGEN SESSIONS KLASSEN Was ist neu? Hilfe Patrick Suter Institutionelle Lizenz

Meine Fragen > ☒ Mini-Check Kompetenzblock 3: Datenbasierte Risiko- und Marktanalyse [Neue Session aktivieren](#)

Dieses Fragenset ist nur für Sie sichtbar. Leisten Sie einen Beitrag, indem Sie es veröffentlichen. [Veröffentlichen](#)

Frage	Punkte
15 Pkte.	
1 Basic 1: Was zeigt die Standardabweichung eines Kurses?	1 Pkte.
2 Basic 2: Was bedeutet eine hohe Standardabweichung?	1 Pkte.
3 Basic 3: Was misst die relative Volatilität?	1 Pkte.
4 Basic 4: Welche Formel ergibt relative Volatilität in %?	1 Pkte.
5 Basic 5: Was bewirkt .std() bei einer Kursreihe?	1 Pkte.
6 Advanced 1: Was bedeutet eine Korrelation von 0.85?	1 Pkte.
7 Advanced 2: Was bedeutet eine Korrelation von -1?	1 Pkte.
8 Advanced 3: Warum entfernt man NaN-Werte vor der Korrelation?	1 Pkte.
9 Advanced 4: Wie kombinierst du zwei Kursreihen nach Datum, um eine Korrelation zwischen Aktie und Index zu berechnen?	1 Pkte.
10 Advanced 5: Wie berechnet man die Korrelation zwischen zwei Kursreihen?	1 Pkte.
11 Expert 1: Welche Frequenz steht bei .resample() für monatlich?	1 Pkte.
12 Expert 2: Was zeigt die monatliche relative Volatilität zweier Aktien?	1 Pkte.
13 Expert 3: Was ist eine typische Anwendung von .resample("M").std()?	1 Pkte.
14 Expert 4: Wie kann man Volatilität visuell am besten darstellen?	1 Pkte.
15 Expert 5: Worauf deutet eine steigende Volatilität hin?	1 Pkte.

Neue Frage Vorschau / Übungslink Veröffentlichen Teilen Kopieren nach ... Verschieben nach ... Ausdrucken Papierkorb

Abbildung 68: Formativer Mini-Check nach Kompetenzblock 3

3.4 Kompetenzblock 4: Repetition, Überprüfung und Reflexion

Der Kompetenzblock 4 dient der Konsolidierung und Überprüfung des Gelernten. Durch das Erstellen eines Mini-Dashboards werden die zentralen Elemente der Unterrichtsreihe nochmals angewendet. In einer summativen Prüfung zeigen die Schüler:innen, wie gut sie Daten importieren, analysieren und visualisieren können. Abschliessend reflektieren sie im Rahmen einer Umfrage die Wirksamkeit der Unterrichtsreihe und geben Feedback zur eigenen Lernentwicklung.

3.4.1 Lektion 10: Erstellung Mini-Dashboard zur Repetition (Partnerarbeit)

Lektion 10 bietet die Gelegenheit, das Erlernte aus den vorherigen Lektionen zu verknüpfen und in einem eigenen Kontext anzuwenden. Diese Lektion eignet als Vorbereitung auf die summativ Prüfung.

Lernziele

Niveau	Lernziele
Basic	- Ich kann eine Aktie oder Kryptowährung mit yfinance laden. - Ich kann den Schlusskurs und SMA(5) visualisieren.
Advanced	- Ich kann Mittelwert, Standardabweichung und relative Volatilität berechnen. - Ich kann die Volatilität grafisch darstellen und inhaltlich interpretieren.
Expert	- Ich kann Wochenvolatilitäten berechnen, vergleichen und grafisch analysieren. - Ich kann aus den Schwankungen auf die Stabilität oder Risikostruktur eines Wertes schliessen.

Abbildung 69: Lektion 10 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
5'	Wissenserarbeitung	- Basic Aufgabe gemeinsam lösen (Lehrgespräch) - Aufgabenstellung Advanced / Expert (Lehrgespräch)
25'	Wissenssicherung	Bearbeitung Projektauftrag Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
10'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Präsentation einzelner Projektbeispiele (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 70: Lektion 10 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Wie würdest du ein Thema für deine Analyse auswählen?	Ich wähle eine Branche / ein Land, die / das mich interessiert oder relevant erscheint.
2.	Was möchtest du mit deiner Analyse zeigen oder herausfinden?	Ob es einen Trend gibt, wie stark die Schwankung ist, ob etwas auffällt.
3.	Was brauchst du, um eine sinnvolle Analyse durchzuführen? Wie kannst du anderen deine Ergebnisse verständlich machen?	Gute Daten, ein klares Ziel und eine verständliche Darstellung. Mit einer klaren Visualisierung, Beschreibung und Interpretation.

Abbildung 71: Lektion 10 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Übungen

Basic Aufgabe / Ausgabe / Musterlösung	
Aufgabe	- Lade eine Aktie oder Kryptowährung (z.B. Bitcoin) mit yfinance - Visualisiere Schlusskurs und SMA(5).
Ausgabe	BTC Kursverlauf mit SMA_5 für das Jahr 2024

Musterlösung

```

1 import yfinance as yf
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 # Aufgabe Basic
6 df = yf.download("BTC-USD", start="2024-01-01", end="2024-12-31") # Daten laden
7 df.index = pd.to_datetime(df.index) # Index auf Datum setzen
8 df["SMA_5"] = df["Close"].rolling(window=5).mean() # Gleitenden Durchschnitt berechnen
9
10 # Liniendiagramm mit Kurs und SMA anzeigen
11 df[["Close", "SMA_5"]].plot(title="BTC Kursverlauf mit SMA_5 für das Jahr 2024")
12 plt.ylabel("Preis (USD)")
13 plt.grid(True)
14 plt.show()
15

```

Advanced Aufgabe / Ausgabe / Musterlösung**Aufgabe**

- Berechne die Standardabweichung, den Mittelwert und schlussendlich die relative Volatilität.
- Interpretiere die relative Volatilität.

Ausgabe

```

YF.download() has changed argument auto_adjust default to True
[*****100%*****] 1 of 1 completed
Durchschnittlicher Kurs: Ticker
BTC-USD      65888.87
dtype: float64
Standardabweichung: Ticker
BTC-USD      14632.13
dtype: float64
Relative Volatilität: Ticker
BTC-USD       22.21
dtype: float64 %
>>>

```

Interpretation: Volatilität von 22% ist hoch für eine Währung. Es ist also eine risikoreiche Kapitalanlage.

Musterlösung

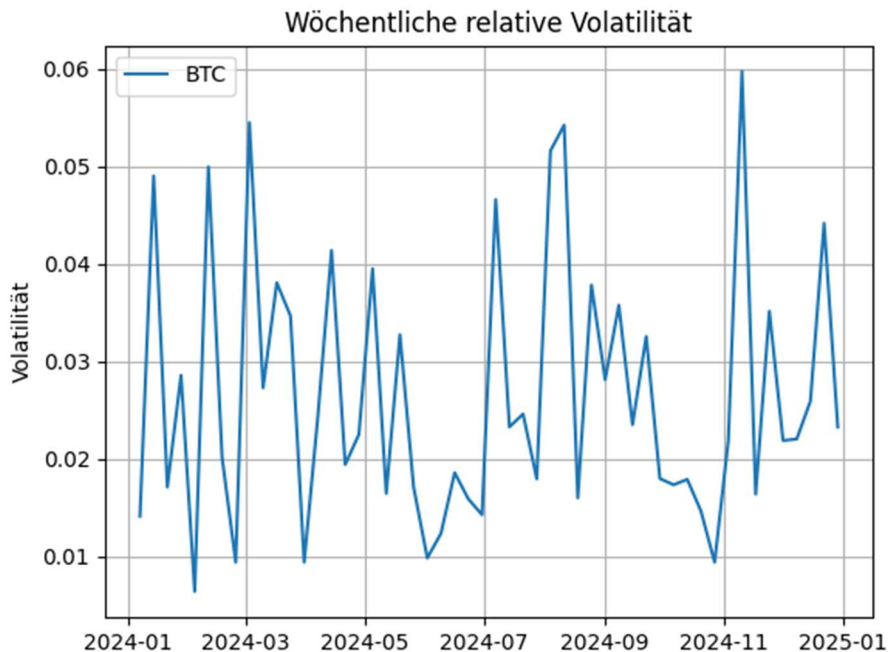
```

16
17 # Aufgabe Advanced
18 # Volatilität berechnen (Standardabweichung der Schlusskurse)
19 vola = df["Close"].std()
20 mean = df["Close"].mean()
21 rel_vola = (vola / mean) * 100
22
23 # Ergebnis ausgeben
24 print("Durchschnittlicher Kurs:", round(mean, 2))
25 print("Standardabweichung:", round(vola, 2))
26 print("Relative Volatilität:", round(rel_vola, 2), "%")
27

```

Expert Aufgabe / Ausgabe / Musterlösung
Aufgabe

- Berechne die wöchentlichen relativen Volatilitäten.
- Stelle die wöchentliche Volatilität graphisch dar.

Ausgabe

Musterlösung

```

28
29 # Aufgabe Expert
30 df_vola_week = df["Close"].resample("W").std()/df["Close"].resample("W").mean()
31 plt.plot(df_vola_week, label="BTC") # Plot des monatlichen Risikoverlaufs
32 plt.title("Wöchentliche relative Volatilität")
33 plt.ylabel("Volatilität")
34 plt.legend()
35 plt.grid(True)
36 plt.xticks()
37 plt.show()
38

```

Abbildung 72: Lektion 10 – Übungen

Reflexionsfragen zum Lektionsabschluss

Frage		Antwort
1.	Was habe ich gelernt?	Individuelle Antworten
2.	Wo sehe ich Potenzial zur Vertiefung?	Individuelle Antworten
3.	Was muss ich mir für die Prüfung genauer anschauen?	Individuelle Antworten

Abbildung 73: Lektion 10 - Reflexionsfragen zum Lektionsabschluss

3.4.2 Lektion 11: Summative Prüfung der Unterrichtsreihe 1

Prüfungsgestaltung

Die Prüfung ist auf Classtime geschrieben und kann mittels folgendem Link aufgerufen werden:
<https://www.classtime.com/student/login/4K28GNK> (Schüler:innen-Ansicht)

Die Prüfung prüft die Kernziele der Lektionen 1–7:

- Daten importieren und vorbereiten (Lektion 1–2)
- Zeitreihen filtern und untersuchen (Lektion 3)
- Daten visualisieren mit matplotlib (Lektion 4)
- Trends erkennen mit SMA und EMA (Lektion 5-6)
- Volatilität und Risiko berechnen (Lektion 7-9)
- Anwenden von Wissen in einem kleinen Projekt (Lektion 10)

Die Prüfung dauert 45min und besteht aus 8 Aufgaben mit unterschiedlichen Aufgabentypen. So können die Schüler:innen zeigen, was sie verstanden haben. Auswendig lernen wird nicht belohnt. Die Prüfung verwendet folgende Aufgabentypen:

- kurze Definitionen
- Richtig/Falsch-Fragen
- Verständnisfragen
- Kategorisierung von Funktionen
- Code erklären
- Fehlersuche
- Code ergänzen

Total können 30 Punkte erzielt werden. Die Aufgaben sind so gewichtet, dass eine durchschnittlicher Schüler:in ca. 70 % der Punkte erreichen kann. Jede Aufgabe trägt sinnvoll zur Gesamtpunktzahl bei. Alle Bewertungen sind transparent und auf nachvollziehbaren Kriterien aufgebaut.

			30 Pkte.
1	 1. Erklärung von Fachbegriffen	 Zur Session hinzufügen	5 Pkte.
2	 2. Verständnisfragen		5 Pkte.
3	 3. Richtig / Falsch Aussagen		3 Pkte.
4	 4. Kategorisierungsaufgabe		3 Pkte.
5	 5. Code Erklärung		3 Pkte.
6	 6. Code Fehler		3 Pkte.
7	 7. Konsolenausgabe		4 Pkte.
8	 8. Code Lücken		4 Pkte.

Abbildung 74: Übersicht Aufgaben der summativen Prüfung

Prüfungsaufgaben

Nachfolgend sind die einzelnen Prüfungsfragen jeweils mit entsprechender Instruktion, Prüfungstyp, Bewertungsmaßstab und Musterlösung dargestellt.

Frage Titel

1. Erklärung von Fachbegriffen

30/170

Erkläre in je einem Satz:

- Was ist eine Zeitreihe? (1P)
- Was bedeutet „parsen“ in der Programmierung? (1P)
- Was misst die Standardabweichung bei Kapitalanlagen? (1P)
- Was zeigt `.describe()` in einem DataFrame? (1P)
- Was bedeutet „Visualisierung“ in der Datenanalyse? (1P)

Bewertung: Jede korrekte Erklärung gibt einen Punkt. Teilpunkte können durch teilweise korrekte Erklärungen erzielt werden.

Bild hinzufügen

YouTube hinzufügen

Audio hinzufügen

Freitext (Kurzantwort)

5.00 Pkte.

Freitext

- ✓ Für kurze und knappe Antworten
- ✓ KI-Autoprüfung verfügbar
 - Keine Textformatierung
 - Kein Anti-Betrug (Text kann eingefügt werden)

- Zeitreihe: Eine geordnete Folge von Messwerten über die Zeit (z.B. täglicher Aktienkurs).
- Parsen: Umwandeln von Text in strukturierte Datentypen (z.B. Datum als Zeitobjekt).
- Standardabweichung: Ein Maß dafür, wie stark Werte um ihren Mittelwert schwanken.
- `.describe()`: Gibt statistische Kennwerte wie Mittelwert, Minimum, Maximum und Quartile zurück.
- Visualisierung: Daten in Form von Diagrammen / Grafiken darstellen. Einfachere Interpretation.

Abbildung 75: Aufgabe 1 der summativen Prüfung

Frage Titel

2. Verständnisfragen

20/170

Beantworte kurz:

- a) Warum verwendet man einen gleitenden Durchschnitt (SMA)? (1P)
- b) Was ist der Unterschied zwischen SMA und EMA? (1P)
- c) Was bedeutet es, wenn eine Aktie ein Beta < 1 , aber > 0 hat? (1P)
- d) Warum ist es sinnvoll, die Standardabweichung relativ zum Mittelwert zu betrachten (z.B. als Prozentsatz)? (1P)
- e) Was sagt eine Korrelation von -0.8 zwischen zwei Aktien? (1P)

Bewertung: Jede korrekte Antwort gibt einen Punkt. Teilpunkte können durch teilweise korrekte Antworten erzielt werden.

Bild hinzufügen

YouTube hinzufügen

Audio hinzufügen

Freitext (Kurzantwort) ▼

5.00 Pkte.

Freitext

- ✓ Für kurze und knappe Antworten
- ✓ KI-Autoprüfung verfügbar
 - Keine Textformatierung
 - Kein Anti-Betrug (Text kann eingefügt werden)

- a) Gleitender Durchschnitt: Glättet Kurse, um den Trend besser erkennbar zu machen.
- b) SMA vs. EMA: SMA gewichtet alle Tage gleich, EMA gibt neueren Werten mehr Gewicht.
- c) Beta < 1 und > 0 : Die Aktie ist weniger volatil als der Markt (defensives Verhalten).
- d) Weil die reine Standardabweichung vom Kursniveau abhängig ist. Eine Aktie mit hohem Kurs kann eine hohe Schwankung haben, die relativ unbedeutend ist. Die relative Volatilität zeigt, wie stark der Kurs im Verhältnis zum Durchschnitt schwankt.
- e) Die beiden Aktien bewegen sich tendenziell entgegengesetzt. Wenn die eine steigt, fällt die andere. Die Werte entwickeln sich negativ verbunden, aber mit starkem Zusammenhang.

Abbildung 76: Aufgabe 2 der summativen Prüfung

Frage Titel

3. Richtig / Falsch Aussagen

28/170

Kreuze an, ob die Aussagen richtig oder falsch sind.

Bewertung: Jede korrekte Antwort gibt einen halben Punkt.

Bild hinzufügen

YouTube hinzufügen

Audio hinzufügen

Kategorisieren ▼

3.00 Pkte.

	richtig	falsch	+ Neue Spalte
.shape gibt den Mittelwert der Kurse zurück.	☐	☒	×
.dropna() entfernt Zeilen mit fehlenden Werten.	☒	☐	×
plotly erlaubt interaktive Diagramme.	☒	☐	×
rolling(5).mean() berechnet die höchste Kursspitze der letzten 5 Tage.	☐	☒	×
.describe() zeigt Minimum, Maximum und Durchschnitt einer Spalte.	☒	☐	×
Eine hohe Standardabweichung bedeutet wenig Volatilität.	☐	☒	×
+ Neue Reihe	×	×	

Falsche Aussagen:

.shape gibt den Mittelwert der Kurse zurück. Falsch, denn .shape gibt die Zeilen- und Spaltenanzahl zurück.
 rolling(5).mean() berechnet die höchste Kursspitze der letzten 5 Tage. Falsch, denn damit wird der Durchschnitt berechnet.
 Eine hohe Standardabweichung bedeutet wenig Volatilität. Falsch, denn das bedeutet hohe Schwankungen.

Richtige Aussagen:

.dropna() entfernt Zeilen mit fehlenden Werten. Korrekt, da es sogenannte NaN-Zeilen entfernt
 plotly erlaubt interaktive Diagramme. Korrekt, da man auf Diagramme klicken und zoomen kann.
 .describe() zeigt Minimum, Maximum und Durchschnitt einer Spalte. Korrekt, da diese zentralen statistischen Kennwerte ausgegeben werden.

Abbildung 77: Aufgabe 3 der summativen Prüfung

Frage Titel

4. Kategorisierungsaufgabe

26/170

Ordne jede Funktion einer der drei Kategorien zu:

Bewertung: Jede korrekte Antwort gibt einen halben Punkt.



Bild hinzufügen



YouTube hinzufügen



Audio hinzufügen



Kategorisieren ▼

3.00 Pkte.

	Datenaufbereitung	Datenanalyse	Datenvisualisierung	+ Neue Spalte
.dropna()	☒	☐	☐	×
.rolling()	☐	☒	☐	×
.describe()	☐	☒	☐	×
.corr()	☐	☒	☐	×
.plot()	☐	☐	☒	×
.set_index()	☒	☐	☐	×
.loc[]	☒	☐	☐	×
.to_numeric()	☒	☐	☐	×
.isnull()	☒	☐	☐	×
.resample()	☐	☒	☐	×
+ Neue Reihe	×	×	×	

Datenaufbereitung: .dropna(), .set_index(), .loc[], .to_numeric(), .isnull()

Datenanalyse: .rolling(), .describe(), .corr(), .resample()

Datenvisualisierung: .plot()

Abbildung 78: Aufgabe 4 der summativen Prüfung

Frage Titel

5. Code Erklärung

17/170

Beantworte folgende 3 Fragen für untenstehenden Code

- a) Was bewirkt rolling(window=10)? (1P)
- b) Was zeigt der Plot? (1P)
- c) Warum ist ein SMA hilfreich für die Interpretation? (1P)

```
df["SMA_10"] = df["Close"].rolling(window=10).mean()  
df[["Close", "SMA_10"]].plot()
```

Bewertung: Jede korrekte Antwort gibt einen Punkt. Teilpunkte können durch teilweise korrekte Antworten erzielt werden.

Bild hinzufügen

YouTube hinzufügen

Audio hinzufügen

Freitext (Kurzantwort) ▼

3.00 Pkte.

Freitext

- ✓ Für kurze und knappe Antworten
- ✓ KI-Autoprüfung verfügbar
 - Keine Textformatierung
 - Kein Anti-Betrug (Text kann eingefügt werden)

- a) Berechnet den 10-Tage-Durchschnitt des Schlusskurses (gleitend).
- b) Zeigt den Originalkurs und den SMA(10) im Liniendiagramm.
- c) Der SMA macht langfristige Trends sichtbar und reduziert kurzfristige Schwankungen.

Abbildung 79: Aufgabe 5 der summativen Prüfung

Frage Titel

6. Code Fehler

14/170

- a) Beschreibe den Fehler in folgendem Code (1P)
b) Erkläre, wie man den Fehler korrigiert (als erklärende Worte oder als korrigierter Code) (2P)

```
df = pd.read_csv("daten.csv")  
df.plot("Datum", "Preis")
```

*Bewertung: Bei Teilaufgabe a) gibt die korrekte Antwort einen Punkt, bei Teilaufgabe b) zwei Punkte.
Teilpunkte können durch teilweise korrekte Antworten erzielt werden.*

Bild hinzufügen

YouTube hinzufügen

Audio hinzufügen

Freitext (Kurzantwort) ▼

3.00 Pkte.

Freitext

- ✓ Für kurze und knappe Antworten
- ✓ KI-Autoprüfung verfügbar
 - Keine Textformatierung
 - Kein Anti-Betrug (Text kann eingefügt werden)

a) Fehler:

- 1) Die Spalte "Datum" wurde nicht indexiert. Dies ist aber für eine Zeitreihe unabdingbar. (1P)

b) Korrektur:

- 1) Datum parsen und indexieren (1P)
2) Nur Preis beim Plot angeben. (1P)

Korrektur Code:

```
df = pd.read_csv("daten.csv", parse_dates=["Datum"])  
df.set_index("Datum", inplace=True)  
df["Preis"].plot()
```

Abbildung 80: Aufgabe 6 der summativen Prüfung

Frage Titel

7. Konsolenausgabe

18/170

Gegeben ist folgender Code. Was genau gibt jede der vier Printfunktionen auf der Konsole aus?



- a) print(df["SMA_3"][1]) (1P)
- b) print(df["SMA_3"][2]) (1P)
- c) print(df["Close"].mean()) (1P)
- d) print(df["SMA_3"].mean()) (1P)

Bewertung: Jede korrekte Antwort gibt einen Punkt. Teilpunkte können durch teilweise korrekte Antworten erzielt werden.

```

1 import pandas as pd
2
3 df = pd.DataFrame({
4     "Close": [100, 102, 104, 106, 108]
5 })
6
7 df["SMA_3"] = df["Close"].rolling(3).mean()
8
9 print(df["SMA_3"][1])
10 print(df["SMA_3"][2])
11 print(df["Close"].mean())
12 print(df["SMA_3"].mean())

```

 Freitext (Kurzantwort) ▼

4.00 Pkte.

Freitext

- ✓ Für kurze und knappe Antworten
- ✓ KI-Autoprüfung verfügbar
 - Keine Textformatierung
 - Kein Anti-Betrug (Text kann eingefügt werden)

- a) print(df["SMA_3"][1]) # → NaN
- b) print(df["SMA_3"][2]) # → (100+102+104)/3 = 102.0
- c) print(df["Close"].mean()) # → (100+102+104+106+108)/5 = 104.0
- d) print(df["SMA_3"].mean()) # → Mittelwert von SMA_3[2] = 102.0, SMA_3[3] = 104.0 und SMA_3[4] = 106.0, also 105

Abbildung 81: Aufgabe 7 der summativen Prüfung

Frage Titel

8. Code Lücken

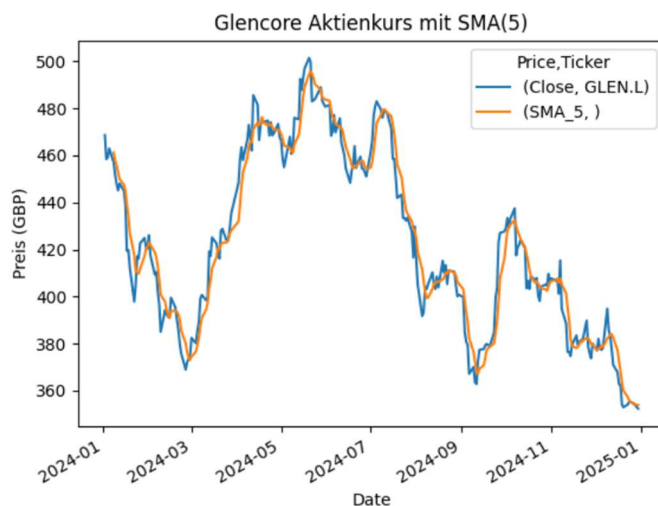
14/170

Bitte fülle die Lücken in folgendem Code, damit er die Graphik korrekt anzeigt. Der Aktienkurs der Glencore soll als „GLEN.L“ mit yfinance für das Jahr 2024 geladen werden. Der zu berechnene 5-Tages-SMA soll zusammen mit dem Schlusskurs in einem Liniendiagramm dargestellt werden.

```
import _Lücke1_ as yf
import pandas as _Lücke2_
import _Lücke3_ . _Lücke4_ as plt
```

```
df = yf . download("GLEN.L", _Lücke5_="2024-01-01", end="2024-12-31")
df[["SMA_5"]] = df["Close"].rolling(5).mean()
df[["Close", "SMA_5"]]._Lücke6_(title="Glencore Aktienkurs mit SMA(5)")
plt.ylabel("Preis (_Lücke7_)")
plt._Lücke8_()
```

Bewertung: Jede korrekt gefüllte Lücke gibt einen halben Punkt.



Lückentext

4.00 Pkte.

Geben Sie den Text in das Feld ein und markieren Sie Wörter, um die Lücken zu erstellen.

Lücke1:	1	yfinance	
Lücke2:	2	pd	
Lücke3:	3	matplotlib	
Lücke4:	4	pyplot	
Lücke5:	5	start	
Lücke6:	6	plot	
Lücke7:	7	GBP	
Lücke8:	8	show	

Abbildung 82: Aufgabe 8 der summativen Prüfung

Prüfungsfairness und -sicherheit

Zur Durchführung der Prüfung sind folgende Session-Einstellungen vorgenommen worden. Um eine möglichst hohe Fairness zu erzielen, werden die Fragen und Antwortmöglichkeiten gemischt. Zudem kann die Prüfung nur mittels Safe Exam Browser gestartet werden, was de facto jegliche Art der digitalen Kommunikation unter den Schüler:innen unterbindet.

Allgemeine Einstellungen ^

- Antwortoptionen mischen** ☒ ?
Antwortoptionen werden in zufälliger Reihenfolge angezeigt.
- Fragen mischen** ☒ ?
Mischen Sie die Reihenfolge der Fragen für die Lernenden.
- Teilpunkte aktivieren** ☒ ?
Wählen Sie die Auswertungsmethode:
Teilpunkte (mit Abzug) ▾
- Erweiterte Schüler-Login-Kontrolle** ☒ ?
Wenn ein Lernender versucht, sich von einem anderen Browser aus in die Session einzuloggen, müssen Sie dies zunächst genehmigen.
- Abmelden, wenn Lernende die Session einreichen** ☐
Wenn die Lernenden ihre Session einreichen, werden sie sofort abgemeldet.

Erlauben Sie nur einen Versuch pro Frage ☐
SchülerInnen können ihre Antwort nicht mehr ändern.

Reflexion zur Session ☐
Aktivieren Sie diese Option, um Ihre Schüler nach Abschluss aller Fragen zu kurzer Reflexion aufzufordern.

Session-Planer ☐
Begrenzen Sie das Zeitfenster, in welchem Lernende beitreten können.
Die Session beginnt um Jederzeit
Die Session endet um Jederzeit

Safe Exam Browser (sichere digitale Prüfungen) ^ ☒

Safe Exam Browser (SEB) ist ein kostenloser Webbrowser zur Durchführung von sicheren, anspruchsvollen Prüfungen auf Windows, MacOS und iOS. Alle Lernenden müssen den SEB auf ihrem Gerät installiert haben.
[Erfahren Sie mehr über diese Integration](#)

Legen Sie Passwörter für das Starten sowie das Beenden des Safe Exam Browsers (SEB) fest:

Start Passwort:

Beenden / Entsperren Passwort:

Abbildung 83: Sicherheitseinstellungen der summativen Prüfung

3.4.3 Lektion 12: Umfrage nach Unterrichtsreihe 1

Die Schüler:innen geben eine quantitative und qualitative Rückmeldung zur eigenen Einschätzung und zum Unterricht. Diese Umfrage wird digital in Google Forms unter <https://forms.gle/aLerL5CG1zaM9E9G8> durchgeführt. Sie ist wie folgt aufgebaut:

Teil A: Quantitative Skalenfragen
Beantworte bitte folgende quantitative Fragen, um messbare Ergebnisse zu erhalten.

1. Unterrichtsaufbau und Methodik *

	Trifft gar nicht zu	Trifft knapp nicht zu	Trifft knapp zu	Trifft zu
Die Unterrichtsreihe war klar strukturiert.	☐	☐	☐	☐
Die Differenzierung in Basic / Advanced / Expert war für mich sinnvoll.	☐	☐	☐	☐
Die Arbeit mit echten Daten war sinnvoll.	☐	☐	☐	☐
Die Projektaufgabe in Lektion 7 hat mein Wissen zusammengefasst.	☐	☐	☐	☐
Die Mini-Checks am Ende der Blöcke haben mir gezeigt, wo ich stehe.	☐	☐	☐	☐
Das Lernjournal hat mir beim Reflektieren geholfen.	☐	☐	☐	☐
Die summative Prüfung war inhaltlich fair und angemessen.	☐	☐	☐	☐

Abbildung 84: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 1

2. Inhalt und Kompetenzerwerb *

	Trifft gar nicht zu	Trifft knapp nicht zu	Trifft knapp zu	Trifft zu
Ich habe verstanden, wie man mit Python Finanzdaten importiert und bereinigt.	☐	☐	☐	☐
Ich konnte mit den gezeigten Methoden Daten analysieren und interpretieren.	☐	☐	☐	☐
Ich habe den Unterschied zwischen rohen Kursdaten und gleitenden Durchschnitten verstanden.	☐	☐	☐	☐
Ich konnte anhand der Übungen mein Wissen direkt anwenden.	☐	☐	☐	☐
Die Visualisierungen (Diagramme, Charts) haben mir beim Verstehen geholfen.	☐	☐	☐	☐
Die Inhalte waren insgesamt gut strukturiert und nachvollziehbar.	☐	☐	☐	☐
Ich konnte durch die Aufgaben Schritt für Schritt mehr Fachwissen aufbauen.	☐	☐	☐	☐

Abbildung 85: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 2

3. Wirkung und persönlicher Lernerfolg *

	Trifft gar nicht zu	Trifft knapp nicht zu	Trifft knapp zu	Trifft zu
Ich habe das Gefühl, in dieser Reihe wichtige Kompetenzen gelernt zu haben.	☐	☐	☐	☐
Ich kann nun einfache Kursdaten selbstständig analysieren und darstellen.	☐	☐	☐	☐
Ich bin mit meinem Lernerfolg in dieser Unterrichtsreihe zufrieden.	☐	☐	☐	☐
Ich verstehe nun wirtschaftliche Begriffe wie Volatilität, Trend oder Korrelation besser.	☐	☐	☐	☐
Die Unterrichtsreihe hat mein Interesse an Programmieren oder Wirtschaft geweckt.	☐	☐	☐	☐
Ich habe gemerkt, dass ich durch Übung sicherer im Umgang mit Daten geworden bin.	☐	☐	☐	☐
Ich traue mir zu, das Gelernte anzuwenden.	☐	☐	☐	☐

Abbildung 86: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 3

Teil B: Qualitative Reflexionsfragen
Bitte beantworte folgende 4 Fragen für dich individuell.

1. Was war für dich das Beste an dieser Unterrichtsreihe? *

Meine Antwort _____

2. Was war für dich besonders herausfordernd oder unklar? *

Meine Antwort _____

3. Welche Verbesserungsvorschläge gehen Dir durch den Kopf? *

Meine Antwort _____

4. Was hast du persönlich aus der Reihe für dich mitgenommen? *

Meine Antwort _____

Abbildung 87: Umfrage nach Unterrichtsreihe 1, Teil B, Fragen 1-4

4 Unterrichtsreihe Teil 2: KI-gestützte Finanzmarktprognosen

Im Anschluss an die Unterrichtsreihe 1 analysieren die Schüler:innen nun Zeitreihen mit dem Ziel, eigenständig Prognosen zu erstellen. Dabei kommen verschiedene Prognosemodelle wie Prophet, ARIMA und LSTM zum Einsatz. Die Modelle werden anhand ihrer Prognosegüte verglichen und kritisch reflektiert. Der Fokus liegt auf dem Verständnis der Modellfunktion, der Evaluation der Vorhersagen und der eigenständigen Durchführung eines datenbasierten Projekts. Die Projektreihe inkl. Abschlussarbeit kann in ca. 10 Lektionen individuell oder in Partnerarbeit durchgeführt werden.

Lernziele

Wie aus der Unterrichtsreihe 1 bekannt, sind nachfolgend die Lernziele gemäss Niveau dargestellt.

Niveau	Lernziele
Basic	- Ich kann ein einfaches Prognosemodell (z.B. Prophet) mit Hilfe einer Vorlage anwenden. - Ich kann eine Prognose grafisch darstellen und verständlich beschreiben.
Advanced	- Ich kann verschiedene Prognosemodelle (z.B. Prophet und ARIMA) anwenden und vergleichen. - Ich kann die Prognosegüte mit Kennzahlen wie RMSE und MAPE bewerten.
Expert	- Ich kann ein neuronales Netz (z.B. LSTM) zur Prognose implementieren. - Ich kann begründet reflektieren, welches Modell bei welchem Datentyp am sinnvollsten ist.

Abbildung 88: Unterrichtsreihe 2 - Lernziele

Prognosemodelle und Gütemasse

Die drei Prognosemodelle und zwei Gütemasse sind hier kurz erklärt:

	Erklärung
Prophet (Prognosemodell)	- Intuitiv nutzbares Prognosemodell, entwickelt von Meta / Facebook) - Erkennt automatisch Trend, Saisonalität und Feiertagseffekte - Robust gegenüber Ausreissern und fehlenden Daten - Eignet sich besonders für Zeitreihen mit klarer Struktur (z. B. Tagesdaten über mehrere Jahre) - Ideal für Einsteiger:innen, da wenig manuelle Parametrierung nötig ist. - Wird direkt über das eigene prophet-Paket genutzt (nicht Teil von sklearn/statsmodels).
ARIMA (Prognosemodell)	- ARIMA = Autoregressive Integrated Moving Average - Ein klassisches statistisches Modell zur Zeitreihenanalyse. - Besteht aus drei Komponenten: - AR (Autoregression): Nutzung vergangener Werte - I (Integrated): Differenzierung zur Stabilisierung der Zeitreihe - MA (Moving Average): Nutzung vergangener Fehler

	• Gut geeignet für lineare, stationäre Zeitreihen ohne stark ausgeprägte Saisonalität. • Modell muss oft manuell konfiguriert werden (z. B. Parameterwahl). • Wird mit dem statsmodels-Paket in Python verwendet.
LSTM (Prognosemodell)	• LSTM = Long Short-Term Memory • Ein spezieller Typ von rekurrenten neuronalen Netzwerken (RNNs). • Entwickelt zur Verarbeitung von sequenziellen Daten, wie z. B. Sprache oder Zeitreihen. • Erkennt komplexe, nicht-lineare Muster in langen Datenfolgen. • Eignet sich für hochkomplexe Zeitreihen mit vielen Einflussfaktoren. • Erfordert mehr Rechenleistung, mehr Daten und besseres Modellverständnis. • Wird mit dem Deep-Learning-Framework Keras innerhalb von TensorFlow umgesetzt.
MAPE (Gütekriterium)	• MAPE = Mean Absolute Percentage Error • Misst den durchschnittlichen prozentualen Fehler zwischen Prognose und realem Wert. • Leicht verständlich – Fehler wird in Prozent angegeben. • Nachteil: Sehr empfindlich bei kleinen Ist-Werten (Division durch kleine Zahlen). • Gut geeignet für den vergleichenden Einsatz bei mehreren Modellen.
RMSE (Gütekriterium)	• RMSE = Root Mean Squared Error • Misst den durchschnittlichen quadratischen Fehler, wobei grosse Abweichungen stärker gewichtet werden. • Aussagekräftig für metrische Vorhersagen (z. B. Preis, Temperatur). • Weniger intuitiv interpretierbar als MAPE, aber mathematisch robuster. • Besonders geeignet, wenn man grosse Abweichungen vermeiden möchte.

Abbildung 89: Unterrichtsreihe 2 – Erklärung Prognosemodelle und Gütemasse

4.1 Kompetenzblock 5: Grundlagen der Zeitreihenprognose

In diesem Block lernen die Schüler:innen den Unterschied zwischen Rückblick und Vorhersage kennen. Sie setzen sich mit den Grenzen der Vorhersagbarkeit auseinander und wenden erste naive Modelle (SMA, EMA) zur Prognose an.

4.1.1 Lektion 13: Was bedeutet Prognose?

In dieser Einstiegseinheit setzen sich die Schüler:innen mit der Grundfrage auseinander, was eine Prognose ist, wozu sie dient und wo ihre Grenzen liegen. Der Fokus liegt auf einem kritischen und realitätsnahen Verständnis von Vorhersagbarkeit in Finanzmärkten.

Lernziele

Niveau	Lernziele
Basic	Ich kann erklären, was eine Prognose ist.
Advanced	Ich kann erläutern, warum Prognosen unsicher sind.
Expert	Ich kann externe Faktoren analysieren, die Prognosen beeinflussen oder unbrauchbar machen.

Abbildung 90: Lektion 13 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 2 Impulsfragen (Plenum)
5'	Wissenserarbeitung	Diskussionsinput anhand eines echten Kursverlauf mit unterschiedlichen Ereignissen (Lehrgespräch)
30'	Wissenssicherung	- Erklärung Abschluss Projekt gemäss Kompetenzblock 8 - Bearbeitung Abschlussprojekt Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Klärung offener Fragen (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 91: Lektion 13 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage	Antwort
1. Was unterscheidet eine Beobachtung von einer Prognose?	Eine Beobachtung beschreibt etwas, das bereits passiert ist. Eine Prognose ist ein Ausblick in die Zukunft.
2. Kann man Börsenkurse exakt vorhersagen? Warum (nicht)?	Nein, nur Näherungen auf Basis vorhandener Daten. Es gibt zu viele unvorhersehbare Einflüsse wie politische Ereignisse, Marktstimmungen oder Krisen.

3.	In welchen Bereichen sind Prognosen besonders wichtig?	In der Wirtschaft, Meteorologie, Epidemiologie, beim Investieren oder bei politischen Entscheidungen.
----	--	---

Abbildung 92: Lektion 13 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Reflexionsfragen zum Lektionsabschluss

Frage		Antwort
1.	Was sind Chancen und Risiken, wenn man sich auf Prognosen verlässt?	Chancen: Bessere Planung, Vorbereitung. Risiken: Fehlentscheidungen, Scheinsicherheit.
2.	Wie sollte man mit Unsicherheiten in Prognosen umgehen?	Man sollte Prognosen als Orientierung und nicht als Wahrheit nutzen. Szenarien-Denkweise ist hilfreich.
3.	Was unterscheidet eine gute von einer schlechten Prognose?	Eine gute Prognose zeigt Unsicherheiten offen, ist datenbasiert und wird regelmässig angepasst.

Abbildung 93: Lektion 13 – Reflexionsfragen zum Lektionsabschluss

4.1.2 Lektion 14: SMA-/EMA-basierte naive Vorhersagen

Diese Lektion dient als Brücke zwischen den Grundlagen aus Unterrichtsreihe 1 und der Prognoselogik. Es werden einfache Mittelwert-Modelle (SMA, EMA) wiederholt und grafisch interpretiert. Anhand dieser Modelle könne erste Prognosen erstellt werden.

Lernziele

Niveau	Lernziele
Basic	Ich kann einen gleitenden Durchschnitt (SMA) berechnen und interpretieren.
Advanced	Ich kann einen EMA berechnen und Unterschiede zum SMA erklären.
Expert	Ich kann Crossovers verschiedener SMAs analysieren und interpretieren.

Abbildung 94: Lektion 14 – Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum)
5'	Wissenserarbeitung	Diskussionsinput anhand eines echten Kursverlauf mit SMA und EMA (Lehrgespräch)

30'	Wissenssicherung	Bearbeitung Abschlussprojekt Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Klärung offener Fragen (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 95: Lektion 14 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Was zeigt ein SMA im Diagramm?	Den Durchschnitt der letzten X Tage. Er glättet kurzfristige Schwankungen.
2.	Wieso reagiert ein EMA schneller als ein SMA?	Weil er die neueren Werte stärker gewichtet. Er verzögert weniger.
3.	Was bedeutet es, wenn ein kurzfristiger SMA einen langfristigen SMA von unten kreuzt?	Ein potenzielles Kaufsignal. Es könnte ein Aufwärtstrend beginnen.

Abbildung 96: Lektion 14 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Reflexionsfragen zum Lektionsabschluss

Frage		Antwort
1.	Was ist die Stärke eines SMA als Prognosemodell?	Einfach zu berechnen, zeigt Trends gut.
2.	Welche Schwächen hat ein SMA bei der Vorhersage?	Er berücksichtigt keine externen Faktoren, reagiert träge.
3.	Wie zuverlässig findest du SMA/EMA zur Prognose von Kursen?	Eingeschränkt. Sie zeigen Tendenzen, aber keine Ereignisse oder Wendepunkte.

Abbildung 97: Lektion 14 - Reflexionsfragen zum Lektionsabschluss

4.2 Kompetenzblock 6: Zeitreihenmodelle und KI-gestützte Prognose

Die Schüler:innen wenden zunehmend komplexe Prognosemodelle an. Prophet (intuitiv), ARIMA (manuell konfigurierbar) und deren Vergleich stehen im Zentrum.

4.2.1 Lektion 15: Prophet (Modellaufbau & Anwendung)

Die Schüler:innen lernen Prophet als benutzerfreundliches Prognosemodell kennen. Sie wenden es auf echte Daten an. So werden sie verstehen, wie Trends, Saisonalitäten und Ausreisser automatisch von Prophet modelliert werden. Der Fokus liegt auf Anwendung, Visualisierung und Interpretation.

Lernziele

Niveau	Lernziele
Basic	Ich kann Prophet nutzen, um eine Zeitreihe zu prognostizieren.
Advanced	Ich kann Prophet anpassen (z.B. Zeitraum, Saisonalität, Feiertage)
Expert	Ich kann die Modellstruktur von Prophet nachvollziehen und interpretieren.

Abbildung 98: Lektion 15 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum) - Erklärung Prophet-Modell (Lehrgespräch)
5'	Wissenserarbeitung	Erklärung Prophet-Modell (Live-Demo)
30'	Wissenssicherung	Bearbeitung Abschlussprojekt Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Klärung offener Fragen (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 99: Lektion 15 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage		Antwort
1.	Was erwartest du von einem Prognosemodell für Zeitreihen?	Dass es Trends und Muster erkennt und möglichst realistische Zukunftswerte ausgibt.
2.	Warum ist Saisonalität in Kursdaten manchmal wichtig?	Weil z.B. das Weihnachtsgeschäft oder Quartalsberichte regelmässig Kurse beeinflussen.
3.	Was könnte ein Vorteil eines intelligenten Modells gegenüber SMA sein?	Es erkennt automatisch Muster und ist robuster bei Ausreissern. Die Prognose wird flexibler.

Abbildung 100: Lektion 15 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Verwendete Funktionen aus prophet

Funktion	Bedeutung
Prophet()	Erstellt ein Prophet-Modell
.fit(df)	Trainiert das Modell mit Daten
.make_future_dataframe(periods=30)	Erzeugt zukünftigen Zeitrahmen, hier für 30 Tage
.predict()	Berechnet Prognosewerte
.plot()	Visualisiert Prognose
df.rename (columns = { 'Datum' : 'ds', 'Preis': 'y'}, inplace=True)	Spalten in Prophet-Modell sollen ds (für Datastamp) und y (für Werte) heissen

Abbildung 101: Lektion 15 – Verwendete Funktionen aus prophet

Reflexionsfragen zum Lektionsabschluss

Frage		Antwort
1.	Wie einfach war die Anwendung von Prophet für dich?	Individuelle Antwort, z.B. Die Prognose lässt sich mit wenigen Zeilen erzeugen.
2.	Was war hilfreich an der automatischen Trend- und Saisonalitätserkennung?	Individuelle Antwort, z.B. Man muss die Struktur nicht selbst kennen. Das Modell findet sie.
3.	Wo würdest du Prophet eher nicht verwenden?	Individuelle Antwort, z.B. Bei sehr kurzen oder chaotischen Datenreihen. Prophet braucht Struktur.

Abbildung 102: Lektion 15 – Reflexionsfragen zum Lektionsabschluss

Live-Demo Übung

Prophet

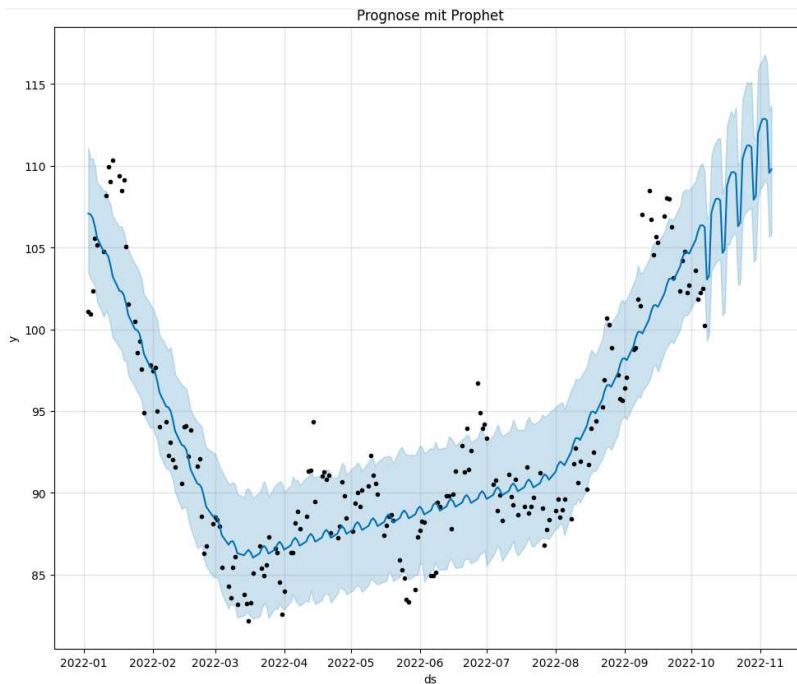
Aufgabe

- Schritt 1: Importiere die Bibliotheken pandas, prophet und pyplot
- Schritt 2: Lade die Datei «Lektion 13-18 kursdaten fiktiv». Überprüfe die Spaltennamen.
Achtung: Prophet erwartet ganz bestimmte Spaltennamen:
 - Die Spalte «ds» für Datestamp. Diese Spalte muss das Datum oder den Zeitstempel enthalten.
 - Die Spalte «y» für Value. Diese Spalte enthält die zu prognostizierenden Werte, z.B. Aktienpreise, Temperaturen, Umsätze etc.
 - Bitte Spalten umbenennen, bevor sie an Prophet übergeben werden:
 - `df.rename (columns = { 'Datum' : 'ds', 'Preis': 'y'}, inplace=True)`

	A	B	C
1	ds,y		
2	2022-01-03,101.09342830602246		
3	2022-01-04,100.91496948997386		
4	2022-01-05,102.32311384069257		
5	2022-01-06,105.5422601021335		
6	2022-01-07,105.15354083465877		
7	2022-01-10,104.76628777422425		
8	2022-01-11,108.18001934772228		
9	2022-01-12,109.94862144402752		
10	2022-01-13,109.02620883473497		
11	2022-01-14,110.31830033591743		
12	2022-01-17,109.40614959192007		
13	2022-01-18,108.4964817597417		
14	2022-01-19,109.13001934517163		
15	2022-01-20,105.063223162272		
16	2022-01-21,101.54377784195627		

- Schritt 3: Erstelle ein Prophet-Modell mit `Prophet()` und trainiere das Prophet-Modell mit `fit(df)`
- Schritt 4: Erstelle den Zukunftsdatenrahmen (z.B. 30 Tage in die Zukunft) mit `make_future_dataframe(30)`
- Schritt 5: Berechne die Prognose mit `predict()`
- Schritt 6: Visualisiere die Prognose mit `plot()`
- Schritt 7: Zeige die folgenden Prognosewerte der letzten 5 Zeilen zusammen mit dem Datum auf der Konsole aus:
 - `ds`: das Datum der Prognose (datetime).
 - `yhat`: die vorhergesagten Werte.
 - `yhat_lower`: die untere Grenze des Prognoseintervalls (Unsicherheitsbereich).
 - `yhat_upper`: die obere Grenze des Prognoseintervalls.

Ausgabe



```
>>> %Run 'Lektion 15 Übung.py'
```

```
10:43:44 - cmdstanpy - INFO - Chain [1] start processing
10:43:44 - cmdstanpy - INFO - Chain [1] done processing
      ds      yhat  yhat_lower  yhat_upper
225 2022-11-02  112.851578  109.116775  116.806025
226 2022-11-03  112.879296  109.040467  116.934310
227 2022-11-04  112.764516  108.856168  116.721579
228 2022-11-05  109.552324  105.422720  113.394644
229 2022-11-06  109.784894  105.833339  113.611758
```

Musterlösung

```
1 # Schritt 1: Bibliotheken importieren
2 import pandas as pd
3 from prophet import Prophet
4 import matplotlib.pyplot as plt
5
6 # Schritt 2: CSV-Datei einlesen (z.B. vorbereitete Kursdaten mit 'ds' und 'y')
7 df = pd.read_csv("Lektion 13-18 kursdaten fiktiv.csv") # Spalten: 'ds' = Datum, 'y' = Preis
8
9 # Schritt 3: Prophet-Modell initialisieren und trainieren
10 model = Prophet()
11 model.fit(df)
12
13 # Schritt 4: Zukunftsdatenrahmen erstellen (30 Tage in die Zukunft)
14 future = model.make_future_dataframe(periods=30)
15
16 # Schritt 5: Prognose berechnen
17 forecast = model.predict(future)
18
19 # Schritt 6: Visualisierung der Prognose
20 model.plot(forecast)
21 plt.title("Prognose mit Prophet")
22 plt.grid(True)
23 plt.show()
24
25 # Schritt 7: Prognoseergebnisse anzeigen
26 # ds: datastamp
27 # yhat: die vorhergesagten Werte.
28 # yhat_lower: die untere Grenze des Prognoseintervalls (Unsicherheitsbereich).
29 # yhat_upper: die obere Grenze des Prognoseintervalls.
30
31 print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail())
```

Abbildung 103: Lektion 15 – Übung als Live-Demo

4.2.2 Lektion 16: ARIMA mit statsmodels

Die Schüler:innen lernen ARIMA als klassisches Modell kennen. Es wird manuell konfiguriert und erfordert stationäre Daten. Somit bietet es eine wichtige Ergänzung zur Prophet-Anwendung. Der Fokus liegt auf Modellverständnis und Parametereinschätzung.

Lernziele

Niveau	Lernziele
Basic	Ich kann ein ARIMA-Modell mit Beispielwerten berechnen.
Advanced	Ich kann Parameter setzen und begründen.
Expert	Ich kann die Modellannahmen analysieren und bewerten.

Abbildung 104: Lektion 16 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum) - Erklärung Arima-Modell (Lehrgespräch)
5'	Wissenserarbeitung	Erklärung Arima-Modell (Live-Demo)
30'	Wissenssicherung	Bearbeitung Abschlussprojekt Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Klärung offener Fragen (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 105: Lektion 16 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage	Antwort
1. Wofür steht AR in Arima?	Autoregressiv: Es verwendet vergangene Werte als Grundlage für die Vorhersage.
2. Wofür steht I in Arima?	Integrated = Differenzierung. Bei der Differenzierung wird der Unterschied zwischen zwei aufeinanderfolgenden Werten gebildet. Das Ergebnis ist eine neue Zeitreihe der Veränderungen, nicht mehr der absoluten Werte.
3. Wofür steht MA in Arima?	Moving Average: Fehler aus vergangenen Zeitpunkten glätten

Abbildung 106: Lektion 16 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Verwendete Funktionen aus statsmodels

Funktion	Bedeutung
ARIMA(df, order=(p,d,q))	Erstellt ARIMA-Modell mit festgelegten Parametern p, d und q: - p steht für Autoregression und nimmt die Anzahl früherer Zeitpunkte, die in die Vorhersage eingehen. Bei Tagesendkursen sind dies die Anzahl Tage. - d steht für Differenzierung, also wie oft die Daten differenziert werden, um sie stationär (stabil über die Zeit) zu machen. Stationarität erhöht die Prognosequalität. Bei d = 1 arbeitet das Modell nicht direkt mit den Originalwerten, sondern mit deren Unterschieden (z.B. heutiger Kurs minus gestriger Kurs). - q steht für die Fehler aus vorherigen Zeitpunkten, die in die Prognose einfließen.
.fit()	Trainiert das Modell
.forecast(steps = 30)	Erzeugt Vorhersage, hier mit 30 Tagen

Abbildung 107: Lektion 16 - Verwendete Funktionen aus statsmodels

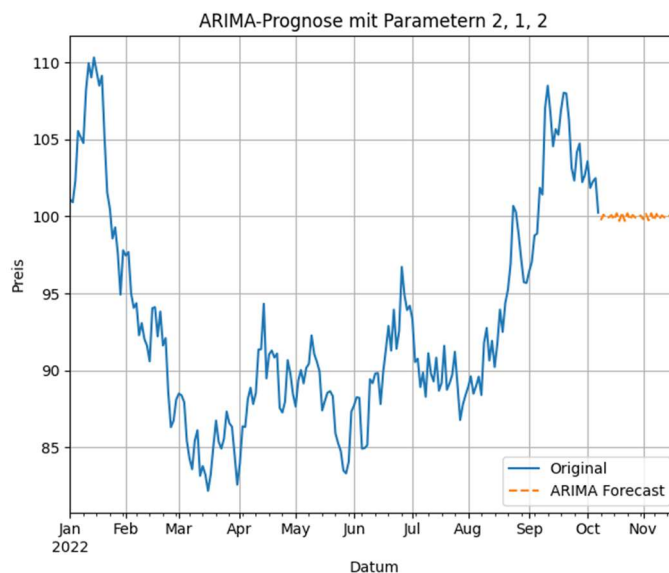
Reflexionsfragen zum Lektionsabschluss

Frage	Antwort
1. Was war schwieriger: Prophet oder ARIMA? Warum?	Individuelle Antwort, z.B. ARIMA, weil man die Parameter verstehen und selbst wählen muss.
2. Welche Vorteile hat ARIMA?	Sehr präzise bei linearen, stationären Datenreihen.
3. Wann würdest du ARIMA Prophet vorziehen?	Individuelle Antwort, z.B. Wenn ich die Daten gut kenne und gezielt modellieren will.

Abbildung 108: Lektion 16 - Reflexionsfragen zum Lektionsabschluss

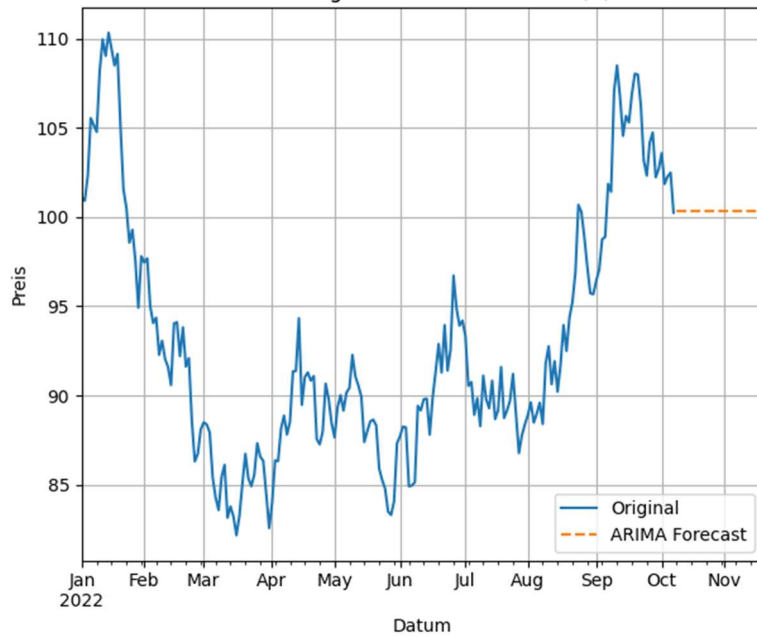
Live-Demo Übung**Arima****Aufgabe**

- Schritt 1: Importiere die Bibliotheken pandas, statsmodels und pyplot
- Schritt 2: Lade die Datei «Lektion 13-18 kursdaten fiktiv».
- Schritt 3: Konvertiere das Datum in ein Datumsformat und setze es als Index
- Schritt 4: Erzeuge die Variable y mit der Spalte «y» des Dataframes
- Schritt 5: Erstelle ein einfaches ARIMA-Modell (z.B. ARIMA(2,1,2))
- Schritt 6: Trainiere das ARIMA-Modell mit model.fit()
- Schritt 7: Erstelle eine Prognose mit forecast() für 30 Tage in die Zukunft
- Schritt 8: Visualisiere Originaldaten und Prognose mit plot()
- Schritt 9: Ändere die Parameter im ARIMA-Modell, damit die Prognose besser funktioniert

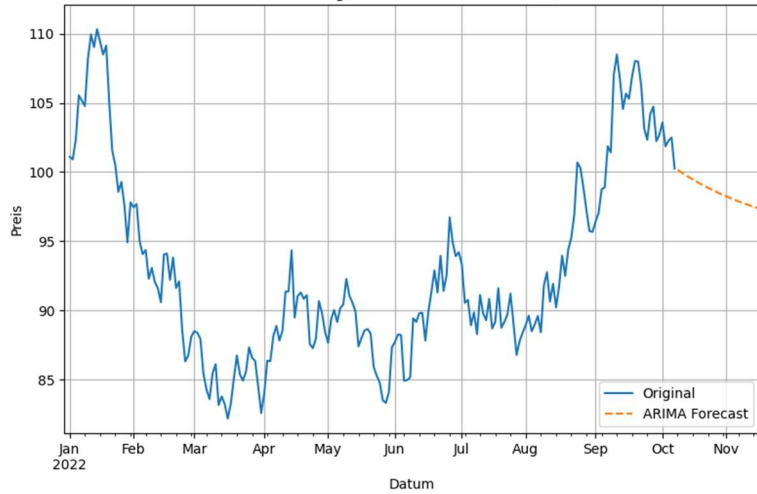
Ausgabe

Die Parameter 2,1,2 scheinen nicht gut zur Zeitreihe zu passen.
Deshalb wurden gemäss Schritt 9 noch weitere Parameter getestet

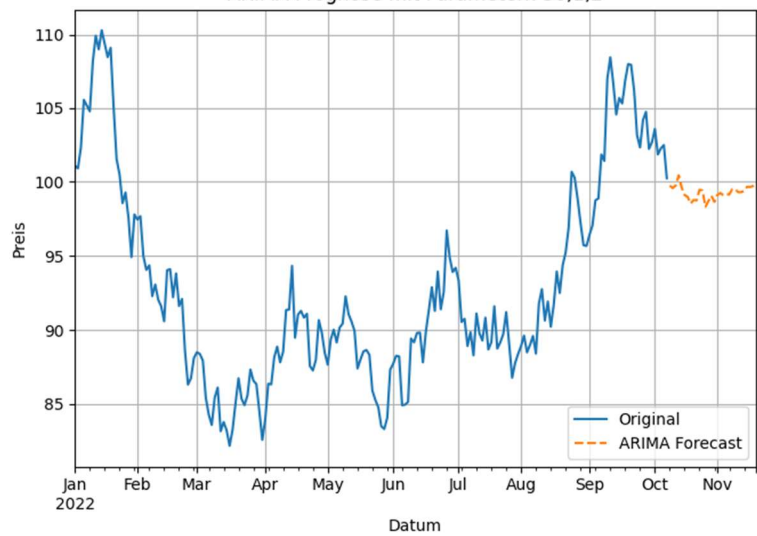
ARIMA-Prognose mit Parametern 1,1,0



ARIMA-Prognose mit Parametern 1, 0, 1



ARIMA-Prognose mit Parametern 30,1,2



Musterlösung

```
1 # Schritt 1: Importiere die Bibliotheken pandas, statsmodels und pyplot
2 import pandas as pd
3 from statsmodels.tsa.arima.model import ARIMA
4 import matplotlib.pyplot as plt
5
6 # Schritt 2: Lade die Datei «Lektion 13-18 kursdaten fiktiv»
7 df = pd.read_csv("Lektion 13-18 kursdaten fiktiv.csv")
8
9 # Schritt 3: Konvertiere das Datum in ein Datumsformat und setze es als Index
10 df["ds"] = pd.to_datetime(df["ds"])
11 df.set_index("ds", inplace=True)
12
13 # Schritt 4: Erzeuge die Variable y mit der Spalte «y» des Dataframes
14 y = df["y"]
15
16 # Schritt 5: Erstelle ein einfaches ARIMA-Modell
17 model = ARIMA(y, order=(2,1,2))
18
19 # Schritt 6: Trainiere das ARIMA-Modell mit model.fit()
20 model_fit = model.fit()
21
22 # Schritt 7: Erstelle eine Prognose für 30 Tage in die Zukunft:
23 forecast = model_fit.forecast(steps=30)
24
25 # Schritt 8: Visualisiere Originaldaten und Prognose mit plot()
26 y.plot(label="Original")
27 forecast.plot(label="ARIMA Forecast", linestyle="--")
28 plt.title("ARIMA-Prognose mit Parametern 2,1,2")
29 plt.ylabel("Preis")
30 plt.xlabel("Datum")
31 plt.legend()
32 plt.grid(True)
33 plt.show()
```

Abbildung 109: Lektion 16 – Übung als Live-Demo

4.2.3 Lektion 17: Modellvergleich und Gütekriterien

In dieser Lektion vergleichen die Schüler:innen zwei Prognosemodelle anhand ihrer Vorhersagegenauigkeit. Sie wenden dazu die Fehlerkennzahlen MAPE (Advanced) und RMSE (Expert) an und reflektieren, welches Modell für welche Daten sinnvoller ist.

Lernziele

Niveau	Lernziele
Basic	Ich kann Prognosekurven visuell vergleichen.
Advanced	Ich kann MAPE berechnen und interpretieren.
Expert	Ich kann RMSE berechnen und begründet einsetzen.

Abbildung 110: Lektion 17 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Warm-up mit 3 Impulsfragen (Plenum) - Erklärung MAPE und RMSE (Lehrgespräch)
5'	Wissenserarbeitung	Erklärung MAPE und RMSE (Live-Demo)
30'	Wissenssicherung	Bearbeitung Abschlussprojekt Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	- Klärung offener Fragen (Plenum) - Beantwortungen der Reflexionsfragen (Lehrgespräch)

Abbildung 111: Lektion 17 - Lektionsablauf

Impulsfragen zum Lektionsstart inkl. Musterlösungen

Frage	Antwort
1. Wie erkennt man visuell, welches Modell besser ist?	Wenn die Prognosekurve näher an den echten Werten liegt.
2. Was bedeutet ein hoher MAPE-Wert?	Die Prognose hat im Schnitt hohe prozentuale Abweichungen.
3. Was zeigt ein hoher RMSE-Wert?	Grosse Einzelabweichungen. Gewähltes Prognosemodell ist bei Ausreissern ungenau.

Abbildung 112: Lektion 17 – Impulsfragen zum Lektionsstart inkl. Musterlösungen

Übungen

MAPE

Aufgabe

- Berechne das Gütekriterium MAPE für folgende Daten

Tatsächliche Werte	100	200	300	400
Prognostizierte Werte	110	190	310	390

Ausgabe

```
>>> %Run 'MAPE Beispiel.py'

MAPE: 5.21 %

>>>
```

Erklärung:

MAPE misst die durchschnittliche Abweichung in Prozent. Bei diesem Beispiel sind die Prognosefehler klein, deshalb ist auch der MAPE niedrig. Je kleiner, desto besser.

Musterlösung

```
1 from sklearn.metrics import mean_absolute_percentage_error
2
3 # Beispielwerte
4 y_true = [100, 200, 300, 400]      # Tatsächliche Werte
5 y_pred = [110, 190, 310, 390]     # Prognostizierte Werte
6
7 # Berechnung
8 mape = mean_absolute_percentage_error(y_true, y_pred)
9
10 # Ausgabe
11 print("MAPE:", round(mape * 100, 2), "%")
```

Abbildung 113: Lektion 17 – Übung MAPE

RMSE

Aufgabe

- Berechne das Gütekriterium RMSE für folgende Daten

Tatsächliche Werte	100	200	300	400
Prognostizierte Werte	110	190	310	390

Ausgabe

```
>>> %Run 'RMSE Beispiel.py'

RMSE: 10.0
```

Erklärung:

RMSE misst die mittlere quadratische Abweichung in absoluten Einheiten. Fehler mit grossem Ausmass werden stärker bestraft. Deshalb eignet sich RMSE besonders bei metrischen Daten, wie z.B. Aktienkursen.

Musterlösung

```

1 from sklearn.metrics import mean_squared_error
2 import numpy as np
3
4 # Beispielwerte
5 y_true = [100, 200, 300, 400]
6 y_pred = [110, 190, 310, 390]
7
8 # Berechnung
9 rmse = np.sqrt(mean_squared_error(y_true, y_pred))
10
11 # Ausgabe
12 print("RMSE:", round(rmse, 2))

```

Abbildung 114: Lektion 17 – Übung RMSE

Reflexionsfragen zum Lektionsabschluss

Frage		Antwort
1.	Was sagt dir die Modellgüte über die Zuverlässigkeit aus?	Je niedriger der Fehler, desto verlässlicher das Modell.
2.	Würdest du MAPE oder RMSE bevorzugen? Warum?	MAPE ist intuitiver (Prozent). RMSE ist präziser bei grossen Fehlern.
3.	Wie beeinflusst der Testzeitraum die Bewertung?	Ein zu kurzer Zeitraum kann Ausreisser überbewerten. Besser mehrere Phasen testen.

Abbildung 115: Lektion 17 - Reflexionsfragen zum Lektionsabschluss

4.3 Kompetenzblock 7: Deep Learning-Modell (optional)

4.3.1 Lektion 18: LSTM-Modelle mit Keras (für Expert-Niveau)

Diese Lektion führt LSTM als modernes Deep-Learning-Modell ein. Sie ist optional und richtet sich an Expert:innen, die sich tiefer mit neuronalen Netzen und komplexen Zeitreihenmodellen befassen möchten.

Lernziele

Niveau	Lernziele
Expert	Ich kann ein LSTM-Modell aufbereiten, trainieren und seine Vorhersage interpretieren.

Abbildung 116: Lektion 18 - Lernziele

Lektionsablauf

Zeit	Phase	Inhalt / Methode
5'	Warm-up / Intro	- Fokus der Lektion (Lehrgespräch) - Erklärung LSTM (Lehrgespräch)
5'	Wissenserarbeitung	Erklärung LSTM (Live-Demo)
30'	Wissenssicherung	Bearbeitung Abschlussprojekt Basic / Advanced / Expert (Einzel- oder Partnerarbeit)
5'	Zusammenfassung / Reflexion / Austausch / Lernkontrolle	Klärung offener Fragen (Plenum)

Abbildung 117: Lektion 18 – Lektionsablauf

Verwendete Funktionen aus tensorflow.keras und numpy

Funktion	Bedeutung
<code>values.reshape(-1, 1)</code>	<code>reshape(-1, 1)</code> formatiert die Daten in ein 2D-Array mit einer Spalte. Dies ist notwendig zur Anwendung eines Machine-Learning-Modells
<code>MinMaxScaler()</code>	<code>MinMaxScaler()</code> skaliert alle Werte auf einen Bereich zwischen 0 und 1, da LSTM-Netzwerke sensibel für Skalen sind. Rohdaten wie Aktienkurse führen ohne Skalierung zu schlechter Performance.
<code>.fit.transform(data)</code>	<code>.fit</code> berechnet die Minimal- und Maximalwerte des Datensatzes <code>.transform</code> Skaliert alle Werte zwischen 0 und 1
<code>Sequential()</code>	<code>Sequential()</code> initialisiert ein lineares, schichtbasiertes neuronales Netzwerk-Modell. Dabei werden die Schichten (Layers) nacheinander hinzugefügt.

LSTM(units=50, activation='relu', input_shape=(X.shape[1], 1))	LSTM() Erstellt eine LSTM-Schicht mit 50 Einheiten. activation='relu': Aktivierungsfunktion input_shape=(X.shape[1], 1): Form der Eingabedaten (z.B. 30 Tage mit je 1 Feature)
model.compile(optimizer='adam', loss='mse')	.compile() bereitet das LSTM für das Training vor. optimizer='adam' ist ein Optimisierungsalgorithmus loss='mse' ist die zu minimisierende Fehlerquote, hier «mean squared root»
Dense(1)	Fügt eine voll verbundene Schicht mit 1 Neuron hinzu. Gibt einen einzelnen numerischen Vorhersagewert zurück (hier z.B. Aktienkurs am nächsten Tag).
model.fit(X, y, epochs=20, verbose=0)	Trainiert das neuronale Netz mit Inputdaten X und Zielwerten y. epochs=20: Der gesamte Trainingsdatensatz wird 20× durchlaufen. Je mehr Durchläufe, desto genauer das Modell. verbose=0: Anzahl angezeigt Trainingsläufe in Konsole. Hier: Kein Trainingsfortschritt im Terminal angezeigt.
.predict()	Erstellt Prognose mit trainiertem Netz

Abbildung 118: Verwendete Funktionen aus tensorflow.keras und numpy

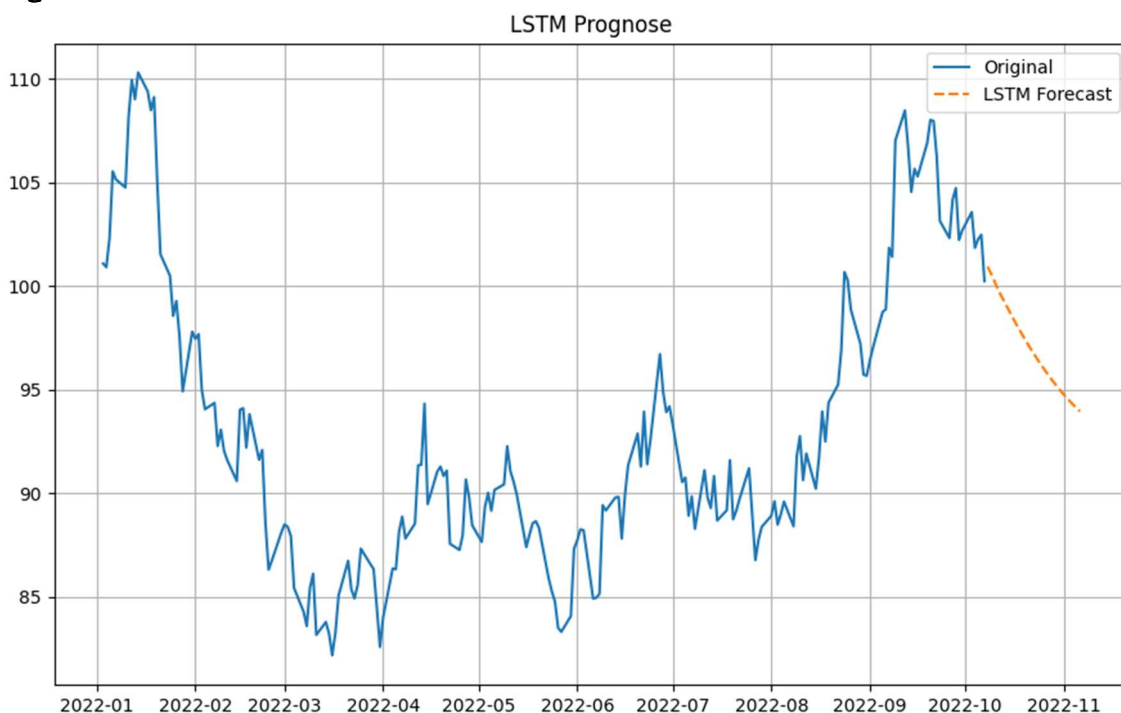
Live-Demo Übung

LSTM

Aufgabe

- Schritt 1: Importiere die Bibliotheken pandas, numpy, tensorflow, sklearn und pyplot
- Schritt 2: Lade und bereite die Daten für Machine-Learning vor
 - Spezifisch: `data = df["y"].values.reshape(-1, 1)`
 - `.values` extrahiert die Werte der Spalte Preis als NumPy-Array
 - `.reshape(-1, 1)` formatiert die Daten in ein 2D-Array mit einer Spalte
- Schritt 3: Skaliere die Daten mit `MinMaxScaler()` auf einen Bereich von 0 bis 1
 - `MinMaxScaler()` skaliert alle Werte auf einen Bereich zwischen 0 und 1, da LSTM-Netzwerke sensibel für Skalen sind. Rohdaten wie Aktienkurse führen ohne Skalierung zu schlechter Performance.
 - `.fit` berechnet die Minimal- und Maximalwerte des Datensatzes
 - `.transform(data)` Skaliert alle Werte zwischen 0 und 1
- Schritt 4: Erstelle Input- und Zielwerte für das LSTM (z.B. 10 Zeitschritte) mit einer for-Schleife
- Schritt 5: Baue das LSTM-Modell mit `Sequential()`, `LSTM()`, `Dense()` und `compile()`
 - `Sequential()` initialisiert ein lineares, schichtbasiertes neuronales Netzwerk-Modell. Dabei werden die Schichten (Layers) nacheinander hinzugefügt.
 - `LSTM()` Erstellt eine LSTM-Schicht mit einer Anzahl Einheiten.
 - `.compile()` bereitet das LSTM für das Training vor.
 - `Dense()` fügt eine voll verbundene Schicht mit Anzahl Neuronen hinzu.
- Schritt 6: Trainiere das Modell mit `.fit(X, y)`
 - Trainiert das neuronale Netz mit Inputdaten X und Zielwerten y.
- Schritt 7: Erstelle Prognosen mit `.predict()` für die nächsten 30 Tage, z.B.
 - Nimm die letzten 10 Werte aus `data_scaled`
 - prognostiziere Schritt für Schritt
- Schritt 8: Visualisiere Originaldaten und Prognose

Ausgabe



Musterlösung

```

1 # Schritt 1: Importiere die Bibliotheken pandas, numpy, tensorflow, sklearn und pyplot
2 import pandas as pd
3 import numpy as np
4 from tensorflow.keras.models import Sequential
5 from tensorflow.keras.layers import LSTM, Dense
6 from sklearn.preprocessing import MinMaxScaler
7 import matplotlib.pyplot as plt
8
9 # Schritt 2: Lade und bereite die Daten vor
10 df = pd.read_csv("Lektion 13-18 kursdaten fiktiv.csv")
11 df["ds"] = pd.to_datetime(df["ds"])
12 df.set_index("ds", inplace=True)
13 data = df["y"].values.reshape(-1, 1)
14
15 # Schritt 3: Skaliere die Daten mit MinMaxScaler auf einen Bereich von 0 bis 1
16 scaler = MinMaxScaler()
17 data_scaled = scaler.fit_transform(data)
18
19 # Schritt 4: Erstelle Input- und Zielwerte für das LSTM (z.B. 10 Zeitschritte)
20 X, y = [], []
21 for i in range(10, len(data_scaled)):
22     X.append(data_scaled[i-10:i])
23     y.append(data_scaled[i])
24 X, y = np.array(X), np.array(y)
25
26 # Schritt 5: Baue das LSTM-Modell
27 model = Sequential()
28 model.add(LSTM(units=50, activation='relu', input_shape=(X.shape[1], 1)))
29 model.add(Dense(1))
30 model.compile(optimizer='adam', loss='mse')
31
32 # Schritt 6: Trainiere das Modell
33 model.fit(X, y, epochs=20, verbose=0)
34
35 # Schritt 7: Erstelle Prognosen für die nächsten 30 Tage
36 # z.B. Nimm die letzten 10 Werte aus data_scaled und prognostiziere Schritt für Schritt
37 forecast_scaled = []
38 input_seq = data_scaled[-10:]
39
40 for _ in range(30):
41     pred = model.predict(input_seq.reshape(1, 10, 1), verbose=0)
42     forecast_scaled.append(pred[0,0])
43     input_seq = np.append(input_seq[1:], [[pred[0,0]]], axis=0)
44
45 forecast = scaler.inverse_transform(np.array(forecast_scaled).reshape(-1, 1))
46
47 # Schritt 8: Visualisiere Originaldaten und Prognose
48 plt.plot(df.index, df["y"], label="Original")
49 future_index = pd.date_range(start=df.index[-1], periods=31, freq='D')[1:]
50 plt.plot(future_index, forecast, label="LSTM Forecast", linestyle="--")
51 plt.title("LSTM Prognose")
52 plt.legend()
53 plt.grid(True)
54 plt.show()

```

Abbildung 119: Lektion 18 – Übung

4.4 Kompetenzblock 8: Abschlussprojekt

Projektauftrag für Schüler:innen

Du analysierst den Verlauf einer Zeitreihe (z.B. einer Aktie oder Kryptowährung) und erstellst mit Hilfe eines geeigneten Modells (Prophet, ARIMA oder LSTM) eine eigene Prognose. Du reflektierst dabei, wie gut das Modell zu deinen Daten passt, und präsentierst deine Ergebnisse grafisch und schriftlich. Du arbeitest allein oder zu zweit.

4.4.1 Projektphasen mit Arbeitsschritten für Schüler:innen

Als Hilfestellung für die Schüler:innen ist das Abschlussprojekt in fünf klare, nachvollziehbare Phasen gegliedert. Die Schritte helfen, eine Zeitreihe auszuwählen, ein Prognosemodell anzuwenden, die Prognose zu visualisieren und zu bewerten.

Phase 1: Datensatz und Modell wählen
• Wähle eine Aktie, Kryptowährung oder einen Index, der dich interessiert. • Wähle ein Modell, das deinem Lernniveau entspricht: ○ Basic: Prophet mit Vorlage ○ Advanced: Prophet & ARIMA <i>mit Vergleich</i> ○ Expert: Prophet, ARIMA & LSTM (eigenständig, tiefgehend)
Phase 2: Daten beschaffen und vorbereiten
• Lade deine Daten mit yfinance • ... oder nutze bereitgestellte CSV-Dateien von Kaggle, z.B. ○ https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset ○ https://www.kaggle.com/datasets/prasoonkottarathil/btcinUSD • Achte auf mindestens 200 Zeitpunkte (z.B. tägliche Daten über 1 Jahr). • Bereinige die Daten (z.B. NaN, falsche Spaltennamen, Zeitformat).
Phase 3: Prognose mit Modell erstellen
• Basic ○ Verwende die Prophet-Vorlage und führe eine 30-Tage-Prognose aus. ○ Stelle die Vorhersage grafisch dar. ○ Beurteile visuell: Wirkt die Prognose sinnvoll und realistisch? • Advanced ○ Wende sowohl Prophet als auch ARIMA an. ○ Berechne MAPE oder RMSE für beide Modelle im gleichen Testzeitraum. ○ Vergleiche die Prognosegüte und begründe, welches Modell besser passt. • Expert ○ Wende Prophet, ARIMA und LSTM (mit Vorlage) an. ○ Vergleiche alle Modelle systematisch. ○ Interpretiere Modellgüte und Grenzen selbstständig. ○ Diskutiere die Eignung je nach Datentyp (z.B. saisonal, volatil, stabil).
Phase 5: Visualisieren
• Zeige Originalverlauf und Prognose im Diagramm. • Stelle Unsicherheitsbereiche, Unterschiede und Trends deutlich dar. • Achte auf korrekte Titel, Achsen und Legende.

Phase 6: Dokumentieren und reflektieren

- Erstelle eine schriftliche Projektzusammenfassung (ca. 5-10 A4-Seiten) mit folgenden Inhalten:
 - Dokumentation der Prognosen: Analyse, Umsetzung, Interpretation
 - Diagramme, Visualisierungen
 - Klar benannte Python-Dateien
 - Sauber geführtes Lernjournal
 - Antworten zu den Reflexionsfragen
 - Quantitative Selbstbewertung gemäss 5 Bewertungskriterien mit Begründung

Abbildung 120: Unterrichtsreihe 2 – Projektphasen und Anleitung für Schüler:innen

4.4.2 Lernjournal und Reflexionsfragen

Das Lernjournal unterstützt die Schüler:innen dabei, ihren individuellen Lernprozess zu dokumentieren. In strukturierter Form erfassen sie ihre Modellwahl, Arbeitsschritte, Herausforderungen und Fortschritte. Die ergänzenden Fragen regen zur Reflexion an. Das Lernjournal dient zugleich als Instrument zur formativen Begleitung und als Teil der Leistungsdokumentation.

Lernjournal

Datum	Modell / Thema	Was habe ich gemacht?	Schwierigkeit	Nächstes Ziel	Reflexion
16.5.2025	Prophet / Bitcoin	Prognose mit Prophet durchgeführt	Mittel	Vergleich mit ARIMA	Prophet ist sehr einfach anwendbar

Abbildung 121: Unterrichtsreihe 2 - Lernjournal

Reflexionsfragen

1.	Welches Modell habe ich gewählt? warum?
2.	Wie beurteile ich die Prognose? (visuell, rechnerisch oder beides)
3.	Was war für mich am schwierigsten?
4.	Was habe ich über Prognosen gelernt?

Abbildung 122: Unterrichtsreihe - Reflexionsfragen

4.4.3 Bewertungskriterien der Projektarbeit

Die Projektarbeit wird anhand von fünf Kriterien bewertet. Die Bewertung ist auf 30 Punkte skaliert und ist niveauübergreifend einsetzbar. Die Kriterien sind so formuliert, dass sie klare Rückmeldung über Stärken und Entwicklungspotenzial ermöglichen.

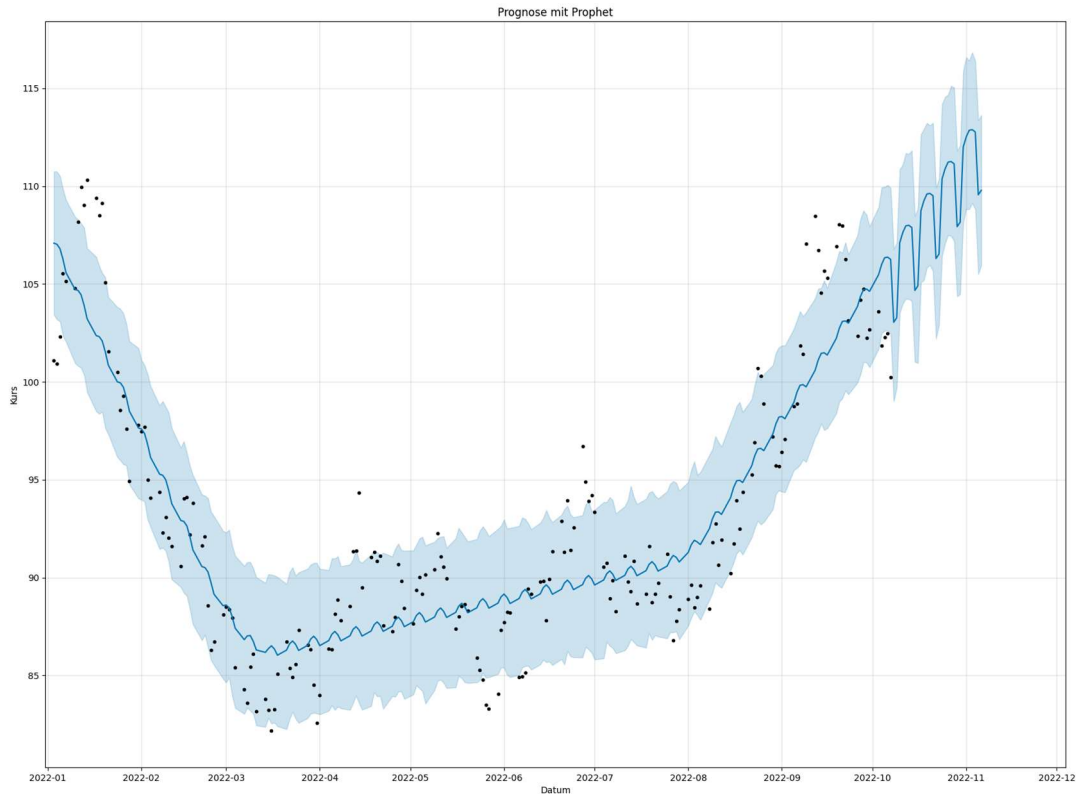
Kriterium	Punkte	Erklärung
1. Datenwahl / -vorbereitung	4	Werden reale, relevante Daten genutzt? Ist die Datenstruktur korrekt aufbereitet? Bewertung der Datenquelle, Sauberkeit der Daten, Umgang mit fehlenden Werten, sinnvolle Zeitreihenwahl.
2. Modellanwendung / Umsetzung	6	Wird der Code verstanden, angepasst, reflektiert eingesetzt? Korrekte Ausführung des Modells je nach Niveau: ○ Prophet (Basic) ○ Prophet+ARIMA (Advanced) ○ LSTM (Expert)
3. Visualisierung / Interpretation	6	Wird die Prognose richtig interpretiert? Werden Trends, Unsicherheiten und allenfalls Fehler benannt? Darstellung der Prognose in einem klar beschrifteten Diagramm mit Titel, Achsen, Legende.
4. Modellgüte / Reflexion	6	Je nach Niveau: visuelle Einschätzung (Basic), MAPE-Berechnung (Advanced), RMSE-Bewertung (Expert). Begründung, welches Modell besser geeignet war, und Reflexion über Aussagekraft.
5. Dokumentation / Lernjournal	8	Ist der Projektverlauf vollständig dokumentiert? Wurden Reflexionsfragen ernsthaft beantwortet? Wird der Arbeitsprozess sichtbar gemacht?
Total	30	

Abbildung 123: Unterrichtsreihe 2 – Bewertungskriterien

4.4.4 Musterlösungen

Basic: Prophet

Ausgabe



Python-Code

```

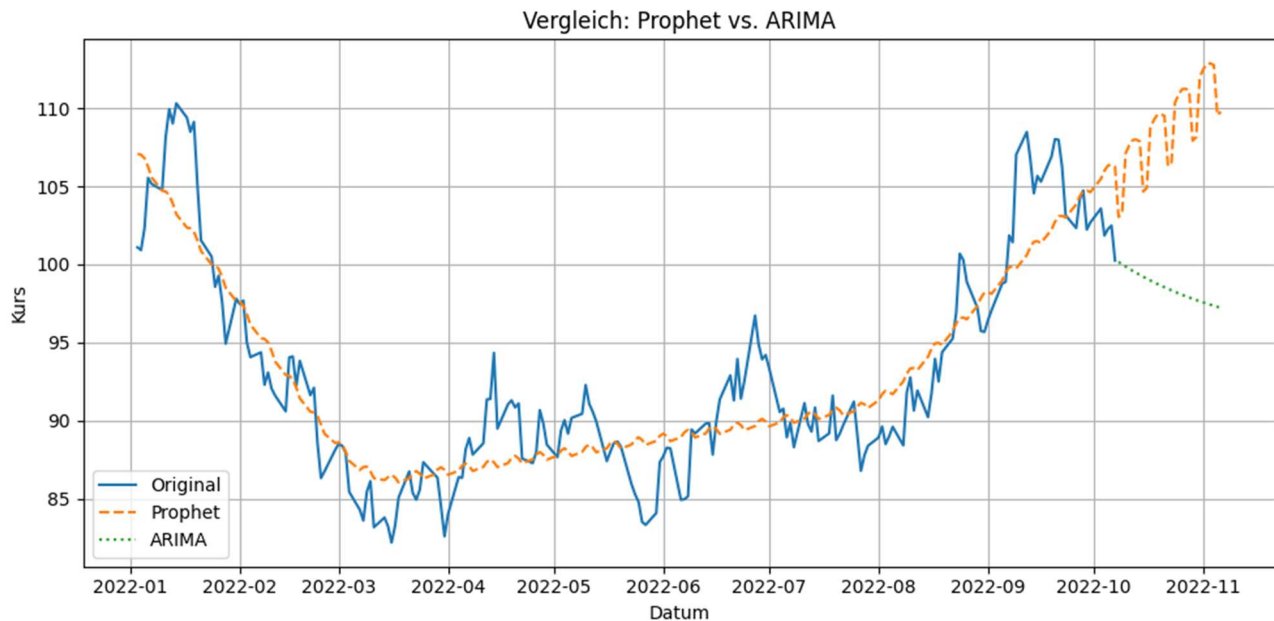
1 # Schritt 1: Bibliotheken importieren
2 import pandas as pd
3 from prophet import Prophet
4 import matplotlib.pyplot as plt
5
6 # Schritt 2: CSV-Datei einlesen (z.B. vorbereitete Kursdaten mit 'ds' und 'y')
7 df = pd.read_csv("Lektion 13-18 kursdaten fiktiv.csv") # Spalten: 'ds' = Datum, 'y' = Preis
8
9 # Schritt 3: Prophet-Modell initialisieren und trainieren
10 model = Prophet()
11 model.fit(df)
12
13 # Schritt 4: Zukunftsdatenrahmen erstellen (30 Tage in die Zukunft)
14 future = model.make_future_dataframe(periods=30)
15
16 # Schritt 5: Prognose berechnen
17 forecast = model.predict(future)
18
19 # Schritt 6: Visualisierung der Prognose
20 model.plot(forecast)
21 plt.title("Prognose mit Prophet")
22 plt.grid(True)
23 plt.ylabel("Kurs")
24 plt.xlabel("Datum")
25 plt.show()

```

Abbildung 124: Unterrichtsreihe 2 – Musterlösung Basic

Advanced: Prophet vs. ARIMA

Ausgabe



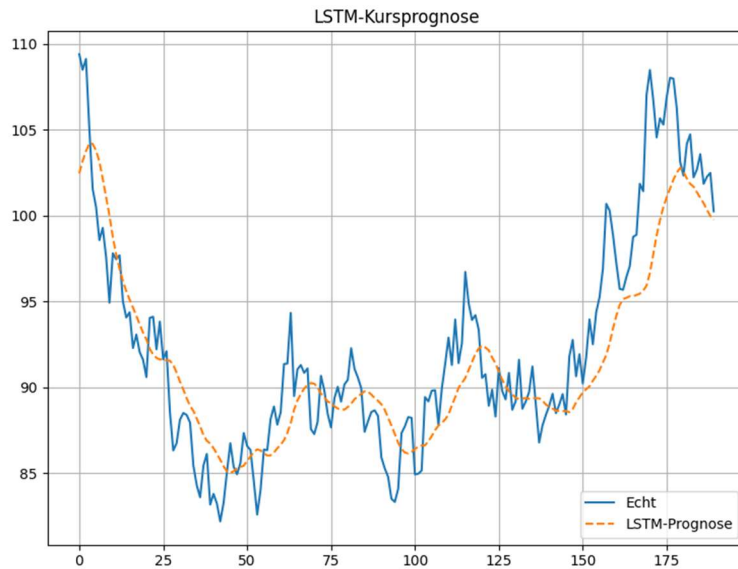
Python-Code

```

1 # Schritt 1: Bibliotheken importieren
2 import pandas as pd
3 from prophet import Prophet
4 from statsmodels.tsa.arima.model import ARIMA
5 from sklearn.metrics import mean_absolute_percentage_error
6 import matplotlib.pyplot as plt
7
8 # Schritt 2: Daten einlesen
9 df = pd.read_csv("Lektion 13-18 kursdaten fiktiv.csv") # Spalten: 'ds', 'y'
10 df["ds"] = pd.to_datetime(df["ds"])
11
12 # Schritt 3: Prophet-Modell erstellen & trainieren
13 model_prophet = Prophet()
14 model_prophet.fit(df)
15 future = model_prophet.make_future_dataframe(periods=30)
16 forecast_prophet = model_prophet.predict(future)
17
18 # Schritt 4: Prophet-Prognosen mit Originaldaten abgleichen
19 # nur bestehende Werte mergen, um NaN-Risiken zu vermeiden
20 merged = df.merge(forecast_prophet[["ds", "yhat"]], on="ds", how="left")
21 df_test = merged.tail(30).dropna(subset=["yhat"]) # Nur letzte 30 bekannte Werte mit Vorhersage
22
23 # Schritt 5: MAPE berechnen
24 mape = mean_absolute_percentage_error(df_test["y"], df_test["yhat"])
25 print("MAPE Prophet:", round(mape * 100, 2), "%")
26
27 # Schritt 6: ARIMA-Modell
28 arima_model = ARIMA(df["y"], order=(1, 0, 1))
29 arima_fit = arima_model.fit()
30 forecast_arima = arima_fit.forecast(steps=30)
31
32 # Schritt 7: Plot: Prophet + ARIMA + Originaldaten
33 plt.figure(figsize=(10, 5))
34 plt.plot(df["ds"], df["y"], label="Original", linewidth=1.5)
35 plt.plot(forecast_prophet["ds"], forecast_prophet["yhat"], label="Prophet", linestyle="--")
36 plt.plot(forecast_prophet["ds"].tail(30), forecast_arima, label="ARIMA", linestyle=":")
37 plt.title("Vergleich: Prophet vs. ARIMA")
38 plt.xlabel("Datum")
39 plt.ylabel("Kurs")
40 plt.grid(True)
41 plt.legend()
42 plt.tight_layout()
43 plt.show()

```

Abbildung 125: Unterrichtsreihe 2 – Musterlösung Advanced

Expert: LSTM**Ausgabe****Python-Code**

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import MinMaxScaler
5 from tensorflow.keras.models import Sequential
6 from tensorflow.keras.layers import LSTM, Dense
7
8 # Daten vorbereiten
9 df = pd.read_csv("Lektion 13-18 kursdaten fiktiv.csv") # Spalten: 'ds', 'y'
10 df["y"] = df["y"].astype(float)
11 scaler = MinMaxScaler()
12 scaled_y = scaler.fit_transform(df["y"].values.reshape(-1, 1))
13
14 # Zeitfenster erzeugen
15 X, y = [], []
16 window_size = 10
17 for i in range(len(scaled_y) - window_size):
18     X.append(scaled_y[i:i+window_size])
19     y.append(scaled_y[i+window_size])
20 X, y = np.array(X), np.array(y)
21
22 # Modell bauen
23 model = Sequential()
24 model.add(LSTM(units=50, return_sequences=True, input_shape=(X.shape[1], 1)))
25 model.add(LSTM(units=50))
26 model.add(Dense(1))
27 model.compile(optimizer='adam', loss='mean_squared_error')
28
29 # Modell trainieren
30 model.fit(X, y, epochs=10, batch_size=16)
31
32 # Prognose erstellen
33 predicted = model.predict(X)
34 predicted = scaler.inverse_transform(predicted)
35
36 # Plot
37 plt.plot(df["y"].values[window_size:], label="Echt")
38 plt.plot(predicted, label="LSTM-Prognose", linestyle="--")
39 plt.title("LSTM-Kursprognose")
40 plt.legend()
41 plt.grid(True)
42 plt.show()

```

Abbildung 126: Unterrichtsreihe 2 – Musterlösung Expert

4.4.5 Umfrage am Ende der Unterrichtsreihe 2

Die Schüler:innen geben eine quantitative und qualitative Rückmeldung zur eigenen Einschätzung und zum Projekt. Diese Umfrage wird digital in Google Forms unter <https://forms.gle/A4D4Hsog35nFUEbk7> durchgeführt. Sie ist wie folgt aufgebaut:

Teil A: Quantitative Skalenfragen
Beantworte bitte folgende quantitative Fragen, um messbare Ergebnisse zu erhalten.

1. Subjektive Erfahrungseinschätzung *

	Trifft nicht zu	Trifft knapp nicht zu	Trifft knapp zu	Trifft zu
Ich habe mich sicher genug gefühlt, um mit dem Projekt zu starten.	☐	☐	☐	☐
Ich konnte ein Modell wählen, das zu meinem Niveau passte.	☐	☐	☐	☐
Ich habe verstanden, wie eine Zeitreihenprognose grundsätzlich funktioniert.	☐	☐	☐	☐
Ich konnte meine Ideen und Entscheidungen selbst einbringen.	☐	☐	☐	☐
Ich konnte meine Ergebnisse sinnvoll visualisieren.	☐	☐	☐	☐
Ich hatte genug Zeit für das Projekt.	☐	☐	☐	☐
Ich habe durch das Projekt Neues über Datenanalyse gelernt.	☐	☐	☐	☐
Ich fand das Projekt realitätsnah und relevant.	☐	☐	☐	☐
Ich hatte Freude am selbständigen Arbeiten.	☐	☐	☐	☐
Ich würde gerne nochmals ein datenbasiertes Projekt machen.	☐	☐	☐	☐

Abbildung 127: Umfrage nach Unterrichtsreihe 2, Teil A, Frage 1

2. Persönlicher Projektfortschritt *

	Trifft gar nicht zu	Trifft knapp nicht zu	Trifft knapp zu	Trifft zu
Ich habe einen geeigneten Datensatz ausgewählt.	☐	☐	☐	☐
Ich habe die Daten bereinigt (z. B. NaN entfernt, Format angepasst).	☐	☐	☐	☐
Ich habe ein Modell (z. B. Prophet, ARIMA oder LSTM) erfolgreich angewendet.	☐	☐	☐	☐
Ich habe eine Prognose für einen sinnvollen Zeitraum erstellt.	☐	☐	☐	☐
Ich habe meine Prognose grafisch dargestellt.	☐	☐	☐	☐
Ich habe meine Prognose mit echten Werten verglichen (z. B. im Testzeitraum).	☐	☐	☐	☐
Ich habe MAPE oder RMSE berechnet (nur bei Advanced / Expert).	☐	☐	☐	☐
Ich habe die Prognosegüte begründet interpretiert.	☐	☐	☐	☐
Ich habe mein Projekt nachvollziehbar dokumentiert.	☐	☐	☐	☐
Ich habe mein Projekt im Lernjournal festgehalten.	☐	☐	☐	☐
Ich habe vollständig im Team zusammengearbeitet.	☐	☐	☐	☐
Ich habe bewusst über mein gewähltes Modell reflektiert.	☐	☐	☐	☐
Ich habe etwas über meine eigenen Fähigkeiten gelernt.	☐	☐	☐	☐

Abbildung 128: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 2

Teil B: Qualitative Reflexionsfragen

Bitte beantworte folgende 4 Fragen für dich individuell.

1. Was war für dich die wichtigste Erkenntnis beim Arbeiten mit einem Prognosemodell? *

Meine Antwort

2. Was fiel dir besonders schwer? warum? *

Meine Antwort

2. Was fiel dir besonders schwer? warum? *

Meine Antwort

3. Welches Prognosemodell hat dich am meisten überzeugt? warum? *

Meine Antwort

4. Was würdest du beim nächsten Mal anders machen? *

Meine Antwort

Abbildung 129: Umfrage nach Unterrichtsreihe 2, Teil B

5 Evaluation der Unterrichtsreihe

In diesem Kapitel wird die Wirkung der Unterrichtsreihe „Finanzdatenanalyse mit Python“ systematisch aus verschiedenen Perspektiven analysiert. Ziel der Evaluation ist es, die Effektivität der Lernstruktur und den Kompetenzerwerb der Schüler:innen zu überprüfen. Auch sollen Schwachstellen erkannt und daraus mögliche Weiterentwicklungen abgeleitet werden.

Zunächst wird in Kapitel 5.1 der tatsächliche Lernumfang aus Sicht der Schüler:innen anhand ihrer Einträge im Lernjournal erfasst. In Kapitel 5.2 folgt die Einschätzung des Lernerfolgs aus Lehrersicht. Diese stützen sich auf die Ergebnisse der Mini-Checks, der summativen Prüfung sowie der Projektarbeiten. Die Ergebnisse werden quantitativ und qualitativ ausgewertet. In Kapitel 5.3 wird die Wirksamkeit der Unterrichtsreihe auf Grundlage strukturierter Umfragen reflektiert. Kapitel 5.4 benennt zentrale Herausforderungen, die sich bei der Umsetzung auf methodischer und organisatorischer Ebene ergaben. Abschliessend enthält Kapitel 5.5 konkrete Verbesserungsvorschläge für eine künftige Anwendung der Unterrichtsreihe.

5.1 Lernumfangsanalyse gemäss Aufgaben-Lernjournal aus Schüler:innen-Perspektive

Die Lernjournale zeigen auf, wie gut die Aufgaben in den einzelnen Lektionen gemäss den drei Schwierigkeitsgraden Basic, Advanced und Expert erfüllt worden sind. Basierend auf der grafischen und tabellarischen Auswertung können folgende Beobachtungen festgehalten werden:

- Basic-Aufgaben wurden durchschnittlich zu 74 % erfüllt, Advanced zu 55 %, Expert zu 31 %.
- Die Differenzierung nach Schwierigkeitsstufen hat grundsätzlich funktioniert. Besonders die hohe Beteiligung im Basic-Bereich zeigt, dass alle Schüler:innen erreicht werden konnten.
- Die niedrigeren Werte bei Advanced und Expert machen deutlich, dass für anspruchsvollere Aufgaben mehr Wiederholung, mehr Zeit und zusätzliche Erklärungen notwendig sind.

Erfüllungsgrad Aufgaben gemäss Lernjournal (Mittelwert)

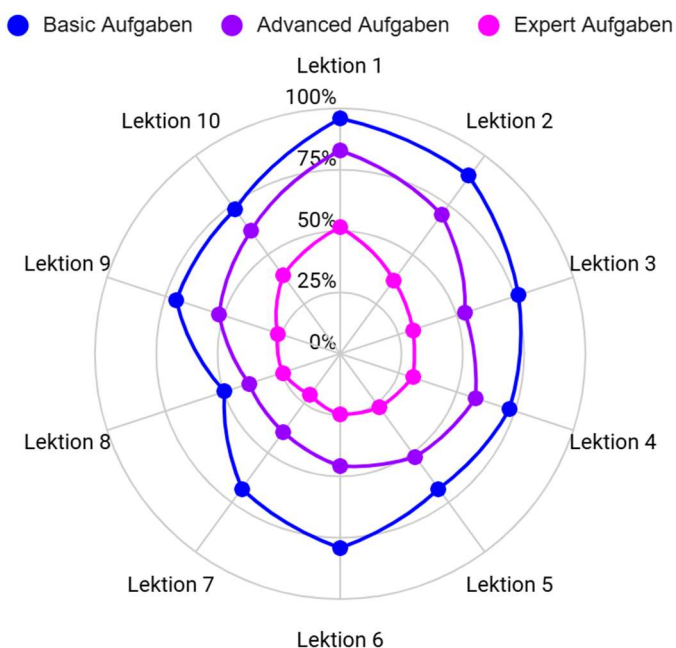


Abbildung 130: Durchschnittlicher Erfüllungsgrad aller Aufgaben Lektionen 1-7 gemäss Lernjournal

Die Basic-Aufgaben wurden mit durchschnittlich 74 % sehr zuverlässig bearbeitet. Besonders hoch war der Erfüllungsgrad in den ersten beiden Lektionen (L1: 96 %, L2: 89 %). In Lektion 8 fiel der Wert deutlich ab (50 %), was auf das inhaltlich anspruchsvolle Konzept der Standardabweichung hindeutet.

Erfüllungsgrad Basic Aufgaben gemäss Lernjournal

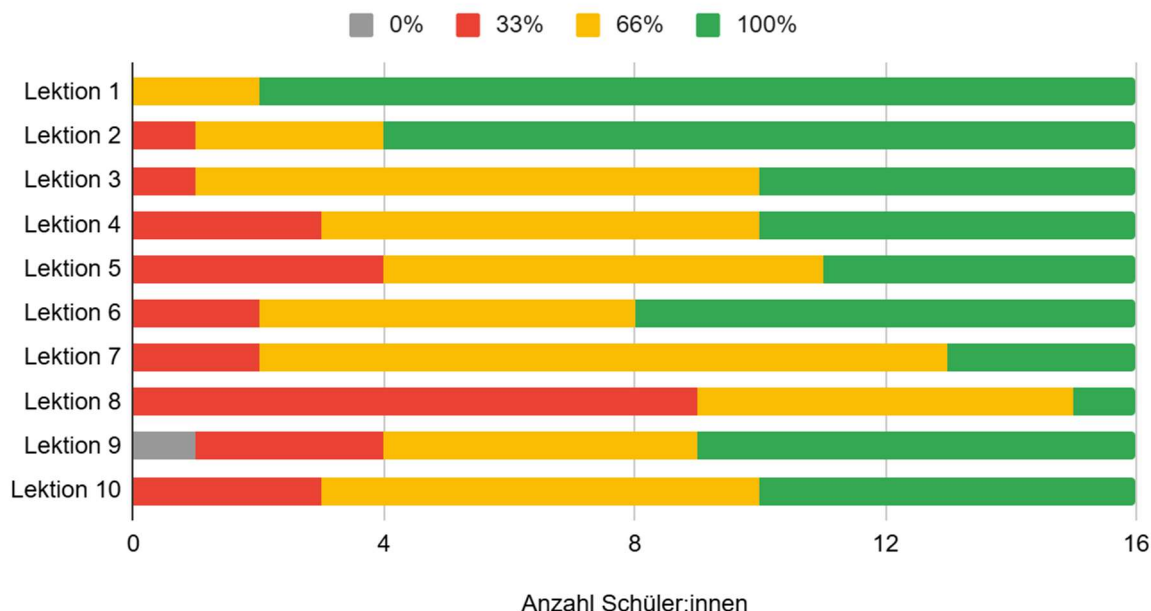


Abbildung 131: Erfüllungsgrad Basic Aufgaben gemäss Lernjournal

Die Advanced-Aufgaben wurden im Schnitt zu 55 % gelöst. Hier zeigten sich grössere Unterschiede zwischen den Lektionen. Während L1 noch 83 % erreichte, lagen mehrere Lektionen (z.B. L7, L8) unter 40 %. Dies deutet auf unterschiedliche Schwierigkeitsgrade hin und benötigt zusätzliche Übungszeit und Erklärungen.

Erfüllungsgrad Advanced Aufgaben gemäss Lernjournal

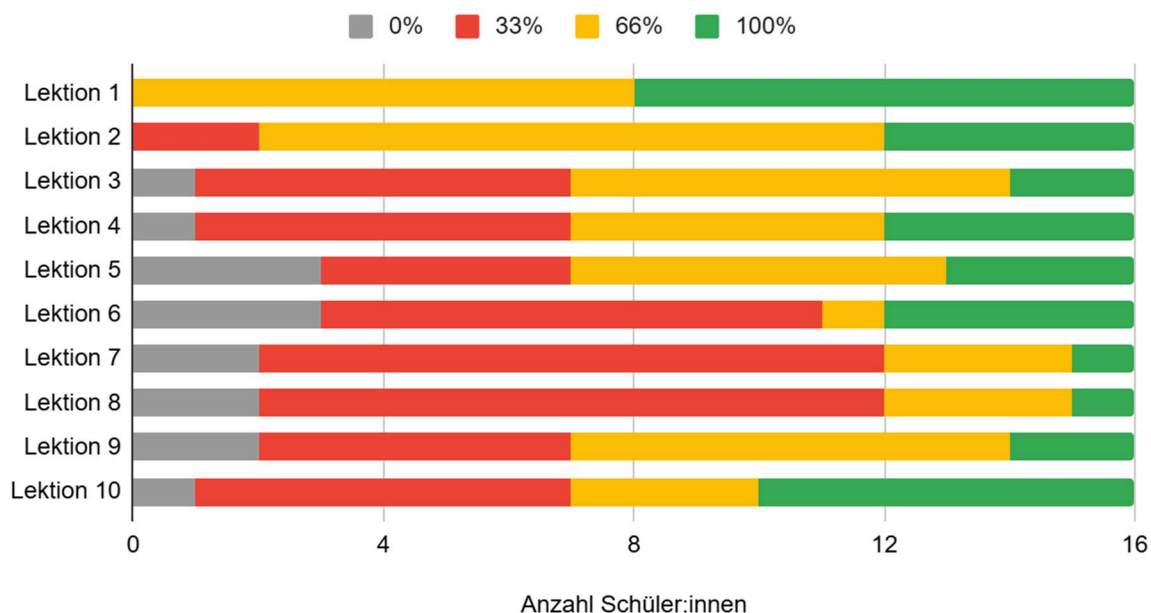


Abbildung 132: Erfüllungsgrad Advanced Aufgaben gemäss Lernjournal

Expert-Aufgaben wurden durchschnittlich zu 31 % gelöst. Dies zeigt, dass das Niveau bewusst fordernd angesetzt wurde. Trotzdem konnten in Lektion 1 über 50 % erreicht werden. In den Lektionen 6 bis 8 lagen die Werte mit unter 25 % teils deutlich tiefer, was dem gestiegenen Anspruchsniveau entspricht.

Erfüllungsgrad Expert Aufgaben gemäss Lernjournal

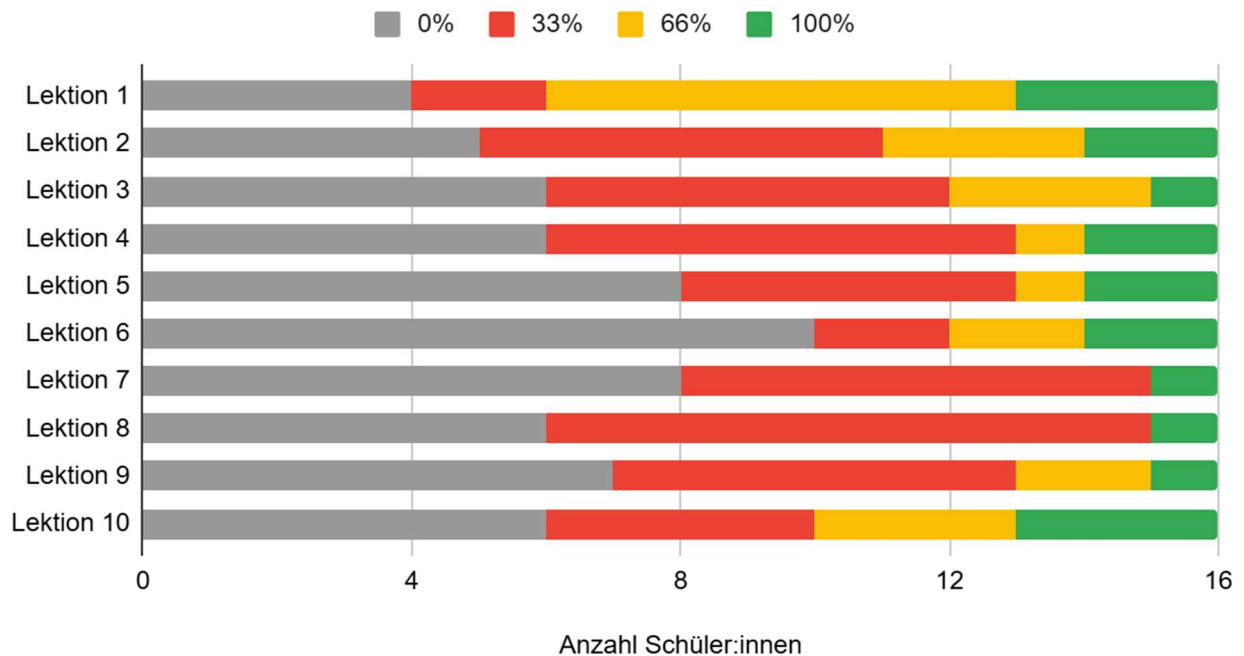


Abbildung 133: Erfüllungsgrad Expert Aufgaben gemäss Lernjournal

5.2 Lernerfolgsanalyse aus Lehrer-Perspektive

5.2.1 Auswertung formative Mini-Checks

In diesem Abschnitt sind die formativen Mini-Checks je Kompetenzblock analysiert. Die Mini-Checks zeigen auf, wie qualitativ gut die Schüler:innen die Lernziele eines bestimmten Kompetenzblocks verstanden haben und das Erlernte anwenden können.

Kompetenzblock 1: Datenimport und -aufbereitung

Die Auswertung zeigt ein solides Leistungsbild mit erkennbarer Differenzierung nach Aufgabenniveau. Die maximale Punktzahl betrug 15 Punkte. Der höchste erreichte Wert lag bei 13 Punkten, erzielt von zwei Schüler:innen. Die niedrigste Punktzahl betrug 7 Punkte. Der Durchschnitt über alle Schüler:innen lag bei 10.75 Punkten, was einem Erfüllungsgrad von 65 % entspricht.

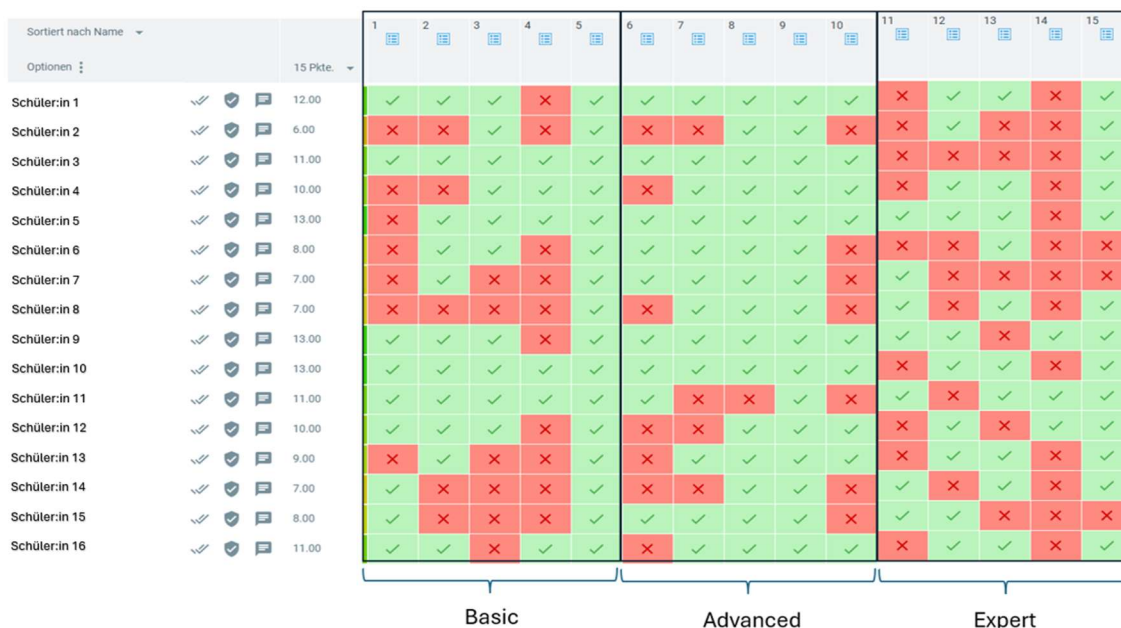


Abbildung 134: Mini-Check nach Kompetenzblock 1 - Resultate

Niveau	Aufgaben	Erfüllung in %	Analyse
Basic	1-5	65%	Die Schüler:innen konnten grundlegende Funktionen wie <code>.head()</code> , <code>.dropna()</code> und <code>.fillna()</code> grösstenteils korrekt anwenden. Schwächen zeigten sich jedoch bei <code>.shape</code> und <code>.columns</code> , was auf ein unsicheres Verständnis der Datenstruktur hindeutet.
Advanced	6-10	76%	Mit einem Erfüllungsgrad von 76 % schnitten die Schüler:innen hier überraschend stark ab. Besonders <code>.resample("M")</code> und <code>.to_numeric()</code> wurden von der Mehrheit korrekt angewendet. Das deutet darauf hin, dass praxisnahe, strukturierte Aufgaben mit erkennbarem Zweck besonders gut verstanden werden.
Expert	11-15	54%	Einzelne Schüler:innen konnten anspruchsvollere Konzepte wie <code>.describe()</code> , <code>df.shape[0]</code> und <code>.fillna(df["Spalte"].mean())</code> umsetzen. Gleichzeitig war hier auch die grösste Streuung zu beobachten, was die Heterogenität im Leistungsniveau bestätigt.

Abbildung 135: Mini-Check nach Kompetenzblock 1 - Auswertung

Kompetenzblock 2: Datenvisualisierung und -interpretation

Die Auswertung zeigt ein Leistungsbild mit hoher Gesamtleistung und relativ homogener Verteilung. Die maximale Punktzahl betrug 15 Punkte. Der höchste erreichte Wert lag bei 15 Punkten, erzielt von mehreren Schüler:innen. Die niedrigste Punktzahl betrug 11 Punkte. Der Durchschnitt über alle Schüler:innen lag bei 12.99 Punkten, was einem Erfüllungsgrad von 86.6 % entspricht.

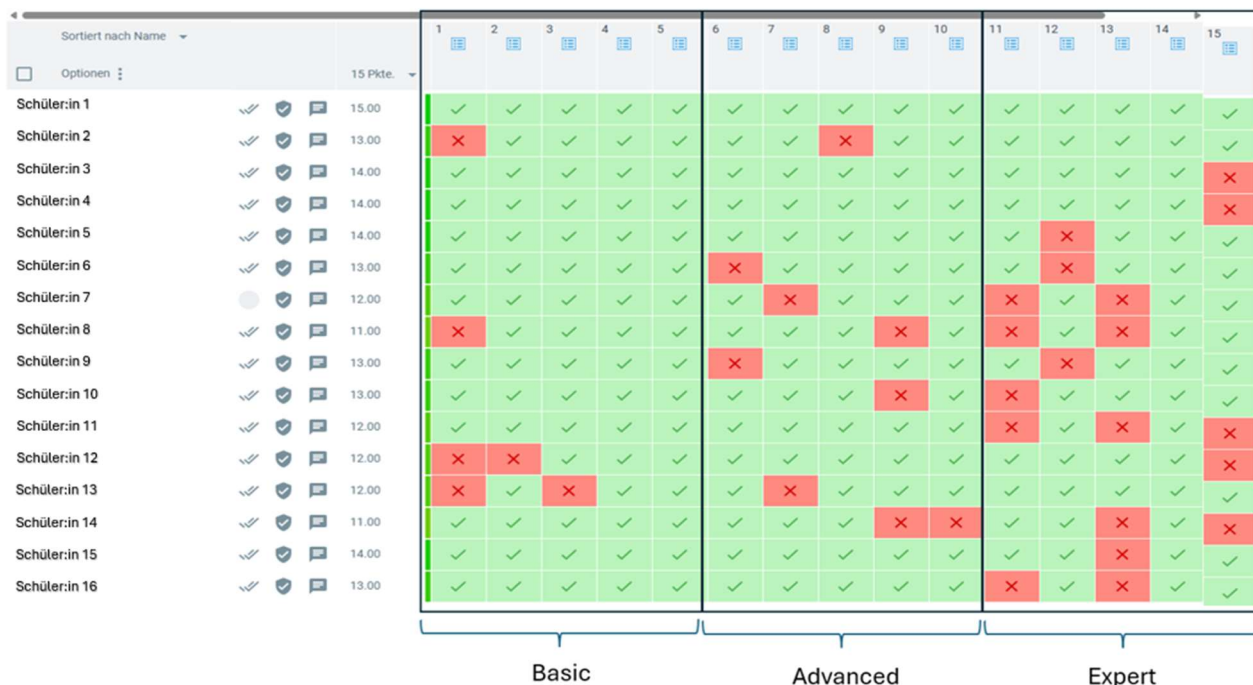


Abbildung 136: Mini-Check nach Kompetenzblock 2 - Resultate

Eine differenzierte Analyse nach Aufgaben-Niveau ergibt folgendes Bild:

Niveau	Aufgaben	Erfüllung in %	Analyse
Basic	1-5	93%	Die grundlegenden Visualisierungsfunktionen wie <code>plt.plot()</code> , <code>title</code> , <code>xlabel</code> , <code>ylabel</code> sowie einfache Diagrammgestaltung wurden von den Schüler:innen sehr sicher beherrscht. Fast alle konnten diese Aufgaben korrekt umsetzen.
Advanced	6-10	89%	Auch die weiterführenden Aufgaben (Candlestick-Charts oder Mehrfachplots) wurden gut gelöst. Die Schüler:innen zeigten ein solides Verständnis für die grafische Aufbereitung von Zeitreihendaten.
Expert	11-15	77%	Komplexere Aufgaben, wie interaktive plotly-Darstellungen oder kombinierte Darstellungen mit Gleitdurchschnitten, wurden von einer Mehrheit gut umgesetzt. Einzelne Schüler:innen zeigten hier besonders starke Leistungen, allerdings war die Streuung höher, was auf unterschiedliche Erfahrungsniveaus schließen lässt.

Abbildung 137: Mini-Check nach Kompetenzblock 2 – Auswertung

Kompetenzblock 3: Datenbasierte Risiko- und Marktanalyse

Die Ergebnisse zeigen ein insgesamt gutes Leistungsbild mit leicht erhöhter Streuung. Die maximale Punktzahl betrug 15 Punkte. Der höchste erreichte Wert lag bei 15 Punkten, der tiefste bei 9 Punkten. Der Durchschnitt über alle Schüler:innen lag bei 12.44 Punkten, was einem Erfüllungsgrad von 82.92 % entspricht.

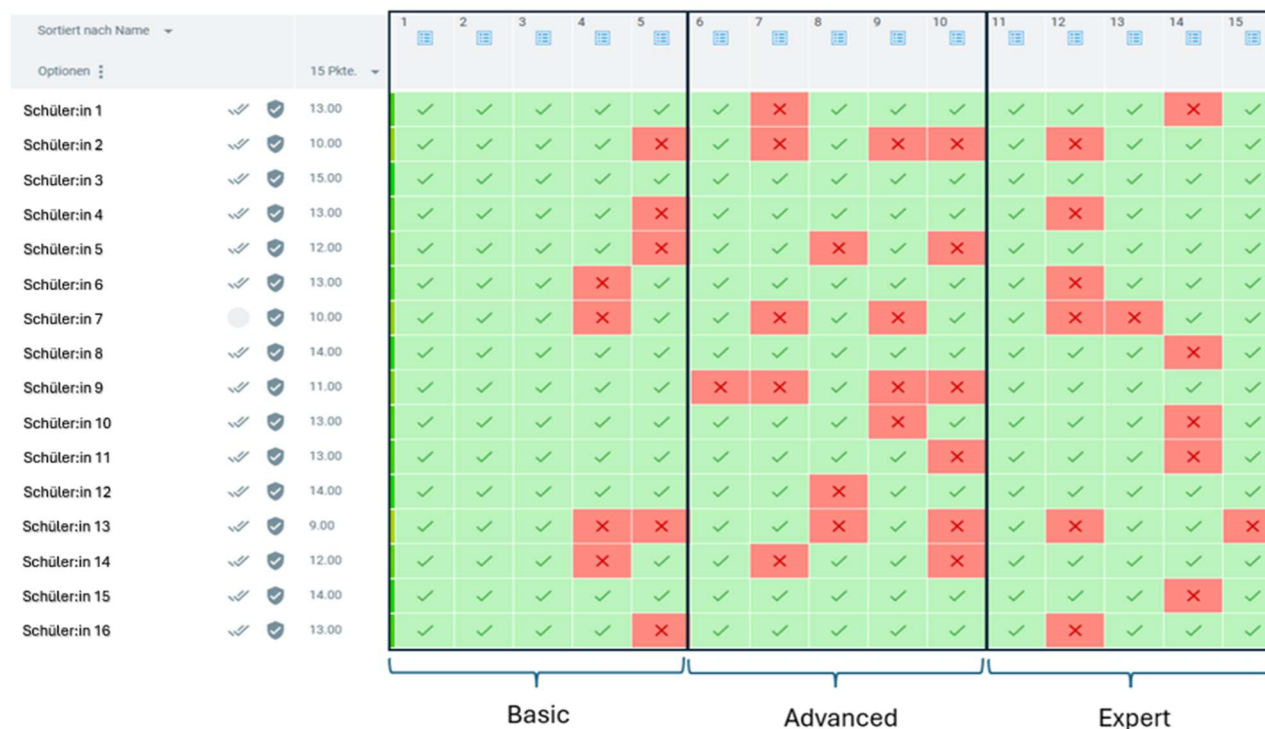


Abbildung 138: Mini-Check nach Kompetenzblock 3 - Resultate

Eine differenzierte Analyse nach Aufgaben-Niveau ergibt folgendes Bild:

Niveau	Aufgaben	Erfüllung in %	Analyse
Basic	1-5	89%	Die Schüler:innen konnten grundlegende Berechnungen zur relativen Volatilität sowie einfache Visualisierungen mit Matplotlib gut umsetzen. Konzepte wie Mittelwert und Standardabweichung sind grösstenteils verstanden.
Advanced	6-10	76%	Die Verknüpfung von Aktie mit Index und die Berechnung von Korrelationen wurden von vielen korrekt angewandt. Einige zeigten Unsicherheiten beim Umgang mit <code>resample()</code> und <code>dropna()</code> .
Expert	11-15	84%	Erfreulich hohe Erfüllung im Expert-Bereich: Monatsweise Volatilitätsanalysen und deren Visualisierung wurden häufig korrekt gelöst. Die Analyse deutet darauf hin, dass besonders motivierte Schüler:innen mit datengetriebenen Aufgaben gut umgehen können.

Abbildung 139: Mini-Check nach Kompetenzblock 3 – Auswertung

5.2.2 Auswertung summative Prüfung

Prüfungsgestaltung

Die summative Prüfung bestand aus acht Aufgaben. Sie wurde am 6.6.2025 bei der Pilotklasse auf Classtime. Ziel war die Überprüfung zentraler Kompetenzen aus den Lektionen 1–10 der Unterrichtsreihe zur Finanzdatenanalyse. Es kamen folgende Aufgabentypen zum Einsatz:

- Fachbegriffe und Verständnis (Fragen 1 und 2)
- Richtig/Falsch & Kategorisierung (Fragen 3 und 4)
- Fehlersuche, Code-Erklärung, Konsolenausgabe (Fragen 5, 6 und 7)
- Code-Vervollständigung (Frage 8)

Leistungsüberblick

Die Prüfungsergebnisse zeigen ein gutes Leistungsbild. Die maximale Punktzahl betrug 30 Punkte. Der höchste erreichte Wert lag bei 26.1 Punkten, der tiefste bei 17.9 Punkten. Der Durchschnitt über alle Schüler:innen lag bei 22 Punkten, was einem Erfüllungsgrad von 73 % entspricht. Die grössten Herausforderungen lagen bei den Fragen 6 (Fehleranalyse) und 7 (SMA-Konsolenausgabe).

	1	2	3	4	5	6	7	8
30 Pkte. ▾								
21.90								
21.80								
22.10								
22.50								
22.30								
22.60								
21.90								
21.30								
24.40								
23.60								
22.00								
18.80								
21.60								
21.40								
26.10								
17.90								

Abbildung 140: Summative Prüfung in Kompetenzblock 4 - Resultate

Notentechnisch liegt der Klassen-Durchschnitt bei 4.7. Alle Schüler:innen haben die Prüfung bestanden. Es wurden Noten zwischen 4.0 und 5.5 erreicht.

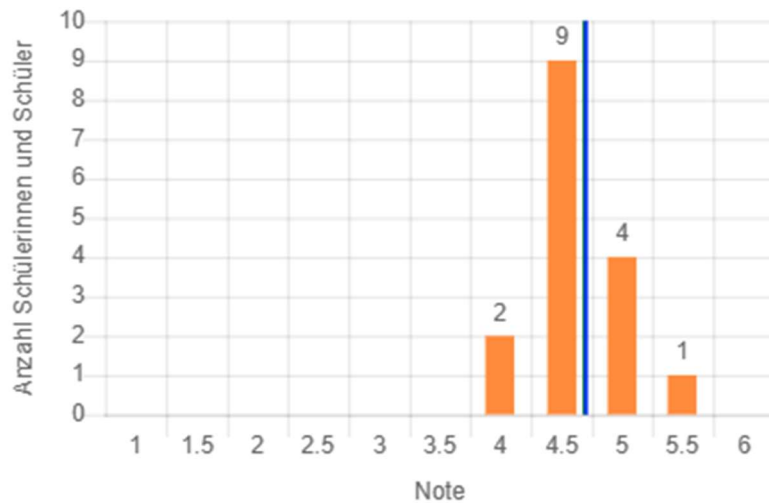


Abbildung 141: Summative Prüfung in Kompetenzblock 4 - Noten

Auswertung nach Aufgaben

Aufgabe	Erfüllung in %	Analyse
1: Fachbegriffe (Zeitreihe, Parsen, Std, Describe)	86%	Begriffe wurden oft gut erklärt, aber gelegentlich unpräzise oder zu oberflächlich
2: Verständnisfragen zu SMA/EMA, Beta, Volatilität	79%	Gute Leistung; Unterschiede zwischen SMA und EMA wurden grösstenteils verstanden
3: Richtig/Falsch Aussagen zu Funktionen	90%	Sehr gute Leistung; .describe() und .dropna() gut erkannt, .rolling() oft ungenau verstanden
4: Kategorisierung von Methoden	68%	Differenzierung zwischen Visualisierung, Analyse, Aufbereitung teils noch unsicher
5: Code-Erklärung von .rolling().mean() + .plot()	86%	Sehr solide, allerdings bei Funktion von .plot() und Bedeutung des Vergleichs Verbesserungspotenzial
6: Fehlersuche im Python-Code	51%	Strukturelle Fehlererkennung teils mangelhaft, oft fehlendes Debugging-Verständnis
7: Konsolenausgabe zu SMA	35%	Schwächstes Ergebnis; deutliche Schwierigkeiten bei Listenlogik und Interpretation
8: Code Lücken (import, plot, show etc.)	89%	Sehr gute Ergebnisse; geringfügige Fehler bei Librarynamen

Abbildung 142: Summative Prüfung in Kompetenzblock 4 – Auswertung nach Aufgaben

Interpretation der Ergebnisse

Stärken	• Solides Verständnis von Fachbegriffen, gleitenden Durchschnitten und Visualisierungsfunktionen • Die Mehrheit konnte Schlüsselmethoden wie <code>.dropna()</code>, <code>.describe()</code> oder <code>plot()</code> korrekt einordnen und anwenden • Der Umgang mit bekannten Python-Strukturen (Imports, Diagramme, <code>.show()</code>) funktionierte sehr zuverlässig.
Herausforderungen	• Einige Schüler:innen zeigten Unsicherheit im analytischen Debugging (Frage 6), oft wurde nur geraten statt systematisch geprüft • Konsolenausgabe (Frage 7) war klar unter Niveau: Einfache Logikfehler bei Listen, unpräzise Rechenwege, keine strukturierte Herleitung • Begriffskategorisierung (Frage 4) war für einige zu abstrakt, deutet auf Defizite in der Transferleistung von Theorie auf Methodenanwendung
Didaktische Folgerungen	• Mehr Übungen zu Fehleranalyse und Code-Logik (Debugging-Trainings, Fehlerdiagnose-Sheets) • Noch mehr Hinweise auf Konsolenausgabe mit <code>print()</code>-Checks pro Zeile und Aufgaben zum Rückwärtsdenken: „Was kommt raus?“ • Verzahnung von Code und Theorie: Die Schüler:innen müssen explizit lernen, warum ein Code funktioniert, nicht nur wie

Abbildung 143: Summative Prüfung in Kompetenzblock 4 – Interpretation der Ergebnisse

5.2.3 Auswertung Projektarbeiten

Die 8 Projektarbeiten im Rahmen der Unterrichtsreihe 2 zeigen insgesamt ein erfreuliches Bild. Die Schüler:innen haben sich mit anspruchsvollen Konzepten der Datenanalyse, Modellierung und Reflexion auseinandergesetzt. Die Ergebnisse verdeutlichen, wie engagiert und kreativ viele Gruppen an ihre Aufgaben herangegangen sind. In vielen Gruppen zeigte sich ein hohes Mass an Kooperation und gegenseitiger Unterstützung.

Basierend auf den 5 Bewertungskriterien wurden die Projektarbeiten wie folgt bewertet:

- Note 5.75 für 2 Teams
- Note 5.50 für 1 Team
- Note 5.25 für 1 Team
- Note 4.75 für 2 Teams
- Note 4.50 für 2 Teams

Der Klassen-Durchschnitt liegt bei 5.0. Alle Schüler:innen haben somit die Projektarbeit erfolgreich abgeschlossen. Es wurden Noten zwischen 4.5 und 5.75 erreicht. In der nachfolgenden Graphik sind die erfassten Noten in Schulnetz eingeblendet.

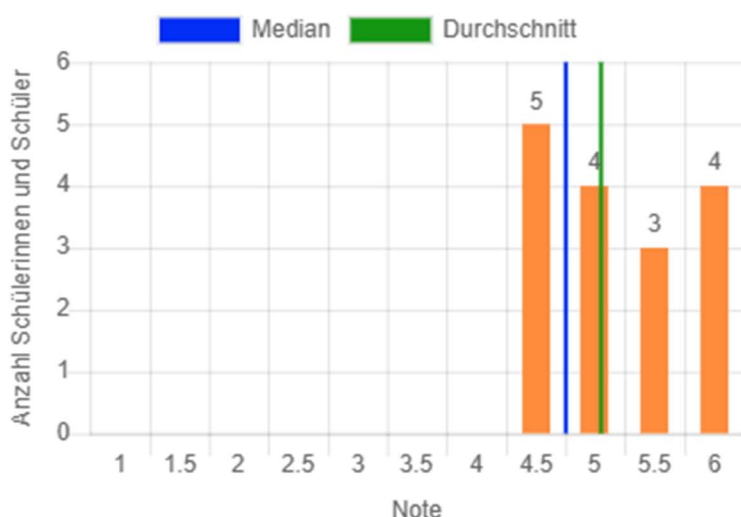


Abbildung 144: Auswertung Projektarbeiten - Noten

Würdigung der Projektarbeiten

Die Projektarbeiten zeigen, dass Schüler:innen auch komplexe Aufgaben wie Finanzzeitreihenprognosen mit KI-gestützten Modellen bewältigen können. Voraussetzung dafür sind die Unterstützung bei Strukturierung, Reflexion und technischer Umsetzung. Während Visualisierung und grundlegende Modellanwendung gut gemeistert wurden, bietet die Reflexion der Modellgüte und die tiefergehende Parametrierung der Modelle Potenzial für weitere Entwicklung.

Stärken	• Nahezu alle Gruppen haben Prophet korrekt implementiert, teils kombiniert mit ARIMA. Zwei Teams haben darüber hinaus auch das LSTM-Modell erfolgreich implementiert und reflektiert. • Die Visualisierungen waren bei fast allen Gruppen sorgfältig umgesetzt. Titel, Achsen und Legenden wurden überwiegend korrekt verwendet, sodass die Diagramme den Erkenntnisgewinn unterstützten.
----------------	---

	• Einige Teams zeigten ein bemerkenswertes Verständnis für die Modellgüte. Besonders hervorzuheben sind hier zwei Gruppen. Sie haben v.a. MAPE in ihre Reflexion inkludiert. • Die Mehrzahl der Gruppen dokumentierte den Arbeitsprozess nachvollziehbar und mit ernsthafter Auseinandersetzung mit den Reflexionsfragen.
Herausforderungen	• Viele Arbeiten blieben bei der Reflexion der Modellgüte oberflächlich. Bei der Berechnung und Aussagekraft zeigten sich deutliche Unterschiede im Niveau. • Manche Gruppen wählten ihre Zeitreihen wenig begründet oder gaben die Quelle nicht präzise an. Dies betraf vor allem Projekte mit nur einer Aktie ohne Begründung der Relevanz. • Die Qualität der Projektdokumentationen war heterogen. Während einige Arbeiten sehr leserfreundlich strukturiert waren, wirkten andere eher wie ein unkommentiertes Lernjournal ohne klaren Aufbau. • Die Parametervariation bei ARIMA und Prophet wurde bei einigen ausgeschöpft. Die Gruppen mit einer Note unter 5 haben jedoch die Möglichkeiten unterschiedlicher Parameter nicht / zu wenig ausgenutzt.

Abbildung 145: Projektarbeiten der Unterrichtsreihe 2 – Würdigung

Highlights der Projektarbeiten

Team1: Herausragend waren die umfassende Reflexion mit MAPE-Berechnungen und die Umsetzung mehrerer Modellvarianten. Die Visualisierung war klar und die Ergebnisse wurden kritisch eingeordnet.

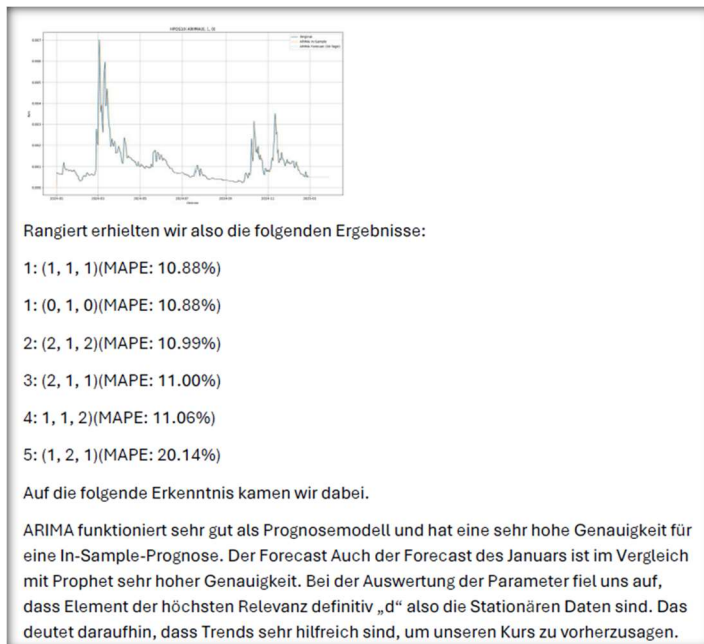


Abbildung 146: Projektarbeiten der Unterrichtsreihe 2 – Screenshot von Team1

Team2: Sie überzeugten durch eine sehr gute Kombination aus ARIMA und LSTM-Umsetzung, gepaart mit einer reflektierten Diskussion der Parameterwahl.

Was haben wir gemacht?

```
def lstm_forecast(model, last_sequence, steps):
    forecasts = []
    current_seq = last_sequence.copy()
    for _ in range(steps):
        next_pred = model.predict(current_seq.reshape(1, SEQ_LEN, 1), verbose=0)
        forecasts.append(next_pred[0,0])
        current_seq = np.append(current_seq[1:], next_pred)
    return scaler.inverse_transform(np.array(forecasts).reshape(-1, 1))

last_seq = train_scaled[-SEQ_LEN:]
lstm_test_pred = lstm_forecast(model_lstm, last_seq, len(test))
future_lstm = lstm_forecast(model_lstm, test_scaled[-SEQ_LEN:], PRED_LEN)

# Evaluation
def evaluate_model(name, true, pred):
    mae = mean_absolute_error(true, pred)
    rmse = np.sqrt(mean_squared_error(true, pred))
    print(f"({name}) - MAE: {mae:.2f}, RMSE: {rmse:.2f}")
    return mae, rmse

print("\nModel Evaluation on Test Data:")
prophet_mae, prophet_rmse = evaluate_model("Prophet", test["y"], prophet_test_pred[-len(test):])
arima_mae, arima_rmse = evaluate_model("ARIMA", test["y"], arima_test_pred[-len(test):])
lstm_mae, lstm_rmse = evaluate_model("LSTM", test["y"][-SEQ_LEN:], lstm_test_pred[-len(test)-SEQ_LEN:])
```

Schwierigkeiten:

Die größte Herausforderung war die Interpretation der LSTM-Ergebnisse. Das Modell zeigte starke Overfitting-Tendenzen – die Trainingsgenauigkeit war hoch, aber die Testdaten wurden schlecht vorhergesagt.

Heute haben wir intensiv am LSTM-Modell gearbeitet. Nach mehreren Anpassungen an der Modellarchitektur und Datenvorverarbeitung ist es uns gelungen, das Training erfolgreich abzuschließen. Anschließend haben wir die Vorhersagen aller drei Modelle (Prophet, ARIMA und LSTM) systematisch verglichen. Dazu haben wir Metriken wie MAE und RMSE berechnet und die Ergebnisse in gemeinsamen Plots visualisiert.

Abbildung 147: Projektarbeiten der Unterrichtsreihe 2 – Screenshot von Team2

Team3: Positiv fiel die strukturierte und leserfreundliche Dokumentation auf, die den Arbeitsprozess sehr gut nachvollziehbar machte.

Weniger Daten, gleicher Zeitraum – mehr Effizienz

Aufgrund begrenzter Rechenleistung müssen wir die zu verarbeitende Datenmenge reduzieren. Bisher lagen die Daten in einer täglichen Auflösung über einen Zeitraum von ca. 220 Tagen vor. **Um die Datenmenge zu verringern und eine effizientere Modellierung zu ermöglichen, verdichten wir die Zeitreihe auf eine wöchentliche Auflösung (z. B. 52 Wochen).** Damit bleibt der Gesamtzeitraum erhalten, allerdings mit weniger Einzelwerten.



Abbildung 148: Projektarbeiten der Unterrichtsreihe 2 – Screenshot von Team3

5.3 Wirksamkeit der Unterrichtsreihe

In diesem Abschnitt möchte ich die Schüler:innen-Umfrage nach der Unterrichtsreihe 1 und 2 auswerten. Damit sollen Unterrichtselemente/-methoden nach Wirksamkeit und Effektivität eingeschätzt werden.

5.3.1 Auswertung Kurzumfragen Temperaturmessung nach Kompetenzblöcken 1-3

In diesem Abschnitt werden die Rückmeldungen der Schüler:innen im Rahmen informeller Kurzumfragen nach den drei Kompetenzblöcken ausgewertet. Die offenen Rückmeldungen bieten Einblick in die subjektive Wahrnehmung der Unterrichtsgeschwindigkeit und des inhaltlichen Verständnisses. Die Aussagen wurden jeweils in drei Blöcke strukturiert: positive Rückmeldungen, spezifische Herausforderungen und konkrete Verbesserungsvorschläge. Daraus habe ich gezielte didaktische Anpassungen für die Lektionsplanung abgeleitet. Die Auswertung dient somit als qualitative Ergänzung zu den quantitativen Resultaten der Mini-Checks.

Temperaturmessung nach Kompetenzblock 1: Datenimport und -aufbereitung

Nach dem Mini-Check zu Kompetenzblock 1 wurde am 23.5.2025 eine informelle Kurzumfrage im Sinne einer Temperaturmessung durchgeführt. Die Schüler:innen konnten ihr Feedback per direkter Nachricht an die Lehrperson via MS Teams senden. Die Wortmeldungen wurden in Rückmeldungen, spezifische Herausforderungen und Verbesserungsvorschläge zusammengefasst:

Positive Rückmeldungen	• Grundidee des interdisziplinären Unterrichts Wirtschaft und Informatik wird mehrfach als sehr gut empfunden. • Mini-Checks werden fast durchgehend positiv bewertet. Sie seien hilfreich beim Wiederholen und Einordnen des eigenen Lernstands. • Der strukturelle Aufbau der Lektionen wird gelobt. • Einige Schüler:innen schätzen die schnellere Arbeitsweise und kommen damit gut zurecht. • Interesse am Thema ist grundsätzlich vorhanden, auch wenn nicht alle Informatik lieben.
Spezifische Herausforderungen	• Probleme bei Fehlern: Wer einen Fehler macht, braucht oft sehr lange, um ihn zu finden und kommt dadurch in Verzug. • Tempo ist für einige Schüler:innen (zu) hoch. • Geringe Wiederholung der Grundlagen aus früherem Informatikunterricht (z.B. Thonny, Python-Basics) • Wunsch nach kompakteren, erklärenden Materialien. • Viel Informatik / wenig Wirtschaft: Die Gewichtung wird als einseitig wahrgenommen.
Verbesserungsvorschläge	• Mehr Zeit pro Lektion / Thema • Mehr WR-Inhalte sichtbar machen • Begriffe und Funktionen explizit erklären • Mehr Übungsaufgaben und Wiederholungen • Klare Dokumentation der verwendeten Funktionen

Abbildung 149: Temperaturmessung nach Kompetenzblock 1

Basierend auf obigem Feedback wurde die Repetitionslektion «Börsenweisheit Sell in May and go Away» im Kompetenzblock 1 in Kapitel 3.1.5 erarbeitet, welche den Schüler:innen zusätzliche Übungsmöglichkeiten und Sicherheit mit dem Datenimport, -bereinigung und -visualisierung geben soll. Sie findet zwischen Lektion 4 und 5 statt.

Temperaturmessung nach Kompetenzblock 2: Datenvisualisierung und -interpretation

Nach dem Mini-Check zu Kompetenzblock 2 wurde am 28.5.2025 eine informelle Kurzumfrage im Sinne einer Temperaturmessung durchgeführt. Die Schüler:innen konnten ihr Feedback per direkter Nachricht an die Lehrperson via MS Teams senden. Die Wortmeldungen wurden in positive Rückmeldungen, spezifische Herausforderungen und Verbesserungsvorschläge zusammengefasst:

Positive Rückmeldungen	• Passendes Lerntempo: Mehrere Schüler:innen empfinden das aktuelle Tempo als ideal • Interesse am Thema: Mehrere Schüler:innen melden, dass der Unterricht interessant sei und sie kaum Kritikpunkte hätten. • Strukturierung der Lektion: Wird als hilfreich empfunden. • Zusammenfassung vor Mini-Check schreiben: Wird als positives Lernerlebnis wahrgenommen. • Mini-Check als Lernhilfe und Erfolgserlebnis: Der Test wird als fair, passend und lösbar empfunden. Mehrere Schüler:innen betonen, dass die Mini-Checks ihnen helfen, Begriffe besser zu verstehen und gezielt zu repetieren.
Spezifische Herausforderungen	• Probleme bei eigenständigem Programmieren: Einige geben an, dass sie zwar bestehenden Code verstehen, ihn aber nicht selbst schreiben könnten • Anwendung von neuen Methoden braucht Zeit und Übungsmöglichkeiten. Schüler:innen sind teils geneigt, Code von schnellen Schüler:innen zu kopieren, anstatt selbst zu erstellen.
Verbesserungsvorschläge	• Repetition / Q&A vor Prüfung: Der Wunsch nach einer strukturierten Repetition vor der Prüfung wurde mehrfach geäußert. • Mehr Coaching und Hilfe bei Aufgaben und Wiederholung zu schwierigen Methoden: Insbesondere zu <code>.rolling()</code>, <code>.ewm()</code>, Candlestick-Charts und <code>fig</code>-Struktur.

Abbildung 150: Temperaturmessung nach Kompetenzblock 2

Die Rückmeldungen der Schüler:innen bestätigen, dass die gewählte Struktur mit einem klaren Aufbau, praxisnahen Aufgaben und formativen Lernkontrollen als hilfreich wahrgenommen wird. Besonders positiv hervorgehoben wurden die Mini-Checks. Diese helfen den Schüler:innen beim Wiederholen der Inhalte und fördern ein tieferes Verständnis der zentralen Begriffe und Funktionen. Die Schüler:innen empfinden das Schreiben der Zusammenfassungen als sinnvolle Vorbereitung, da dabei das Gelernte nochmals bewusst durchdacht werden muss.

Gleichzeitig zeigt sich, dass das Tempo des Unterrichts für die Klasse teils gut, teil zu hoch ist. Einige Schüler:innen berichten, dass sie aufgrund von Verständnisproblemen und geringen Programmierinteresse den Anschluss verlieren. Dieses Gefühl tritt insbesondere die Programmieranteile auf fortgeschrittenem Niveau auf. Die Differenzierung in Basic-, Advanced- und Expert-Aufgaben wird grundsätzlich verstanden. Um das mittlere Niveau zu stärken, werden die Lektionen 8, 9 und 10 jeweils von einer Einzellektion auf eine Doppellektion erweitert. Diese Entlastung soll Raum und Zeit schaffen, damit die Mehrheit der Schüler:innen zusätzliche Erfolgserlebnisse und Vertrauen in der Python-Programmierung erzielen können. Der Lehrperson ermöglicht es auch mehr Zeit für Fehlersuche und individuelles Coaching.

Temperaturmessung nach Kompetenzblock 3: Datenbasierte Risiko- und Marktanalyse

Nach dem Mini-Check zu Kompetenzblock 3 wurde am 4.6.2025 eine informelle Kurzumfrage im Sinne einer Temperaturmessung durchgeführt. Die Schüler:innen konnten ihr Feedback per direkter Nachricht an die Lehrperson via MS Teams senden. Die Wortmeldungen wurden wiederum in positive Rückmeldungen, spezifische Herausforderungen und Verbesserungsvorschläge zusammengefasst:

Positive Rückmeldungen	- Die Unterrichtsgestaltung mit schrittweisem Aufbau wird positiv wahrgenommen. - Mehrere Schüler:innen berichten, dass sie den Code im Nachhinein verstehen, auch wenn es beim ersten Mal noch nicht ganz klar ist. - Einzelne Rückmeldungen zeigen, dass der Kompetenzblock 3 zeitlich gut getaktet war und sich die Schüler:innen besser eingearbeitet fühlten.
Spezifische Herausforderungen	- Die Programmierlogik („warum passiert was?“) ist oft nicht direkt / sofort nachvollziehbar. Dieses Verständnis braucht Zeit, Übung und Repetition. Dies trifft besonders auf fortgeschrittenere Aufgaben zu. - Einige Schüler:innen haben grundsätzliche Python Programmierungsdefizite (for-Schleifen, Listen, etc.), die logischerweise auch bei schwierigeren Themen wie Volatilität oder Korrelation auftreten. So bleiben neue Konzepte trotz Erklärung, Übungen und Repetition vereinzelt unklar. So äussern gewisse Schüler:innen ein Gefühl der Überforderung durch die Menge an neuem Inhalt in kurzer Zeit, da bei diesen Schüler:innen die informatische Basis nicht auf Verständnis, sondern auf Auswendiglernen beruht.
Verbesserungsvorschläge	- Einige Schüler:innen wünschen sich mehr Zeit zur Wiederholung und Verknüpfung der Themenblöcke. Dies findet im Kompetenzblock in der Lektion 10 mit Partnerarbeit statt. Diese Lektion wird auf eine Doppellektion erweitert. - Der Wunsch nach einer klareren Erklärung der Bedeutung einzelner Codezeilen ist mehrfach geäußert worden. Da die Fragen individuell unterschiedlich sind, wird die Lehrperson während der Lektion 10 individuelles Lerncoaching für die betroffenen Schüler:innen anbieten.

Abbildung 151: Temperaturmessung nach Kompetenzblock 3

Die Rückmeldungen zeigen, dass sich die Schüler:innen grundsätzlich engagieren und das Unterrichtskonzept mit Differenzierung sowie Visualisierung schätzen. Gleichzeitig offenbart sich bei einzelnen komplexeren Themenbereichen ein erhöhtes Bedürfnis nach Entschleunigung, intensiverer Begriffseinführung und stärkeren Praxiserklärungen. Die Kombination aus Repetition, Reflexionsfragen und gestuften Aufgaben scheint hilfreich. Neue Konzepte wie Volatilität oder Korrelation brauchen zusätzliche Übungs- und Repetitionsmöglichkeiten, was in Lektion 10 stattfindet. Zur Entschleunigung wird diese Lektion als Doppellektion durchgeführt.

5.3.2 Auswertung Umfrage nach Unterrichtsreihe 1

Die Umfrage wurde am 6.6.2025 zum Ende der Unterrichtsreihe 1 durchgeführt. Die Umfrage beinhaltet quantitative und qualitative Einschätzung bezüglich der Unterrichtswirkung aus Sicht der Schüler:innen.

Quantitative Einschätzung: Gesamtfazit

Die quantitativen Rückmeldungen der Schüler:innen zeigen ein insgesamt positives Bild der Unterrichtsreihe. Positiv werden die klare Struktur, die Verwendung echter Daten und die grafische Darstellung von Finanzinformationen aufgenommen. Besonders gut wurden der Unterrichtsaufbau, Visualisierungen als Verständnisbrücke und die fachliche Kompetenzentwicklung im Bereich Datenimport und -bearbeitung bewertet. Allerdings zeigen sich in den Bereichen Anwendungskompetenz, Selbstwirksamkeit und Interesse gewisse Schwächen. Einige Schüler:innen fühlen sich noch nicht sicher genug, das Gelernte eigenständig anzuwenden. Die Mini-Checks, das Lernjournal und die summative Prüfung werden mehrheitlich als nützlich wahrgenommen. Der konkrete Nutzen variiert je nach Schüler:in. Insgesamt sprechen die Ergebnisse für eine wirksame und strukturierte Unterrichtsreihe, die jedoch an mehreren Stellen zielgerichtet nachgebessert werden kann.

Quantitative Einschätzung: Unterrichtsaufbau und Methodik

Die Struktur und Methodik der Unterrichtsreihe wurden sehr positiv wahrgenommen. Die Differenzierung in Niveaus sowie der Einsatz echter Marktdaten fanden breite Zustimmung. Schwächer schnitten das Lernjournal und die Projektaufgabe in Lektion 10 ab, die als weniger hilfreich wahrgenommen wurden. Die Mini-Checks wurden grundsätzlich als unterstützend empfunden.

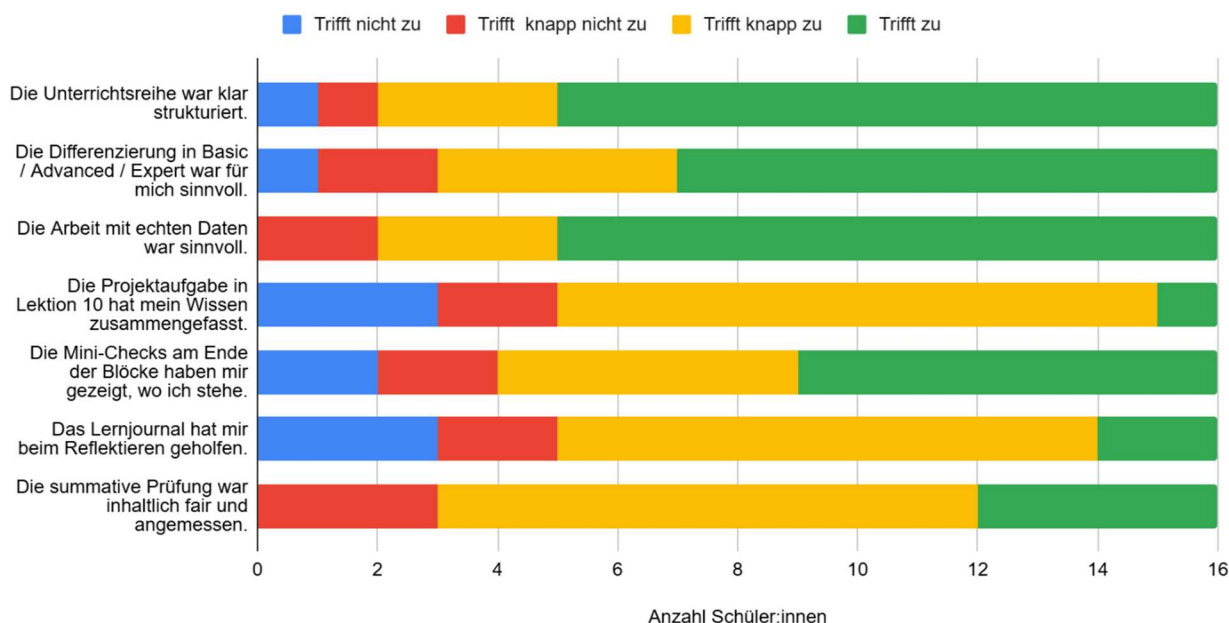


Abbildung 152: Umfrageresultate zu Unterrichtsaufbau und Methodik

Quantitative Einschätzung: Inhalt und Kompetenzerwerb

Die Schüler:innen fühlten sich inhaltlich gut angeleitet und konnten zentrale Begriffe und Methoden nachvollziehen. Besonders stark war das Verständnis beim Datenimport, der Visualisierung und dem Aufbau von Fachwissen durch gestufte Aufgaben. Weniger überzeugend war das selbstständige Anwenden des Wissens. Hier zeigte sich Unsicherheit.

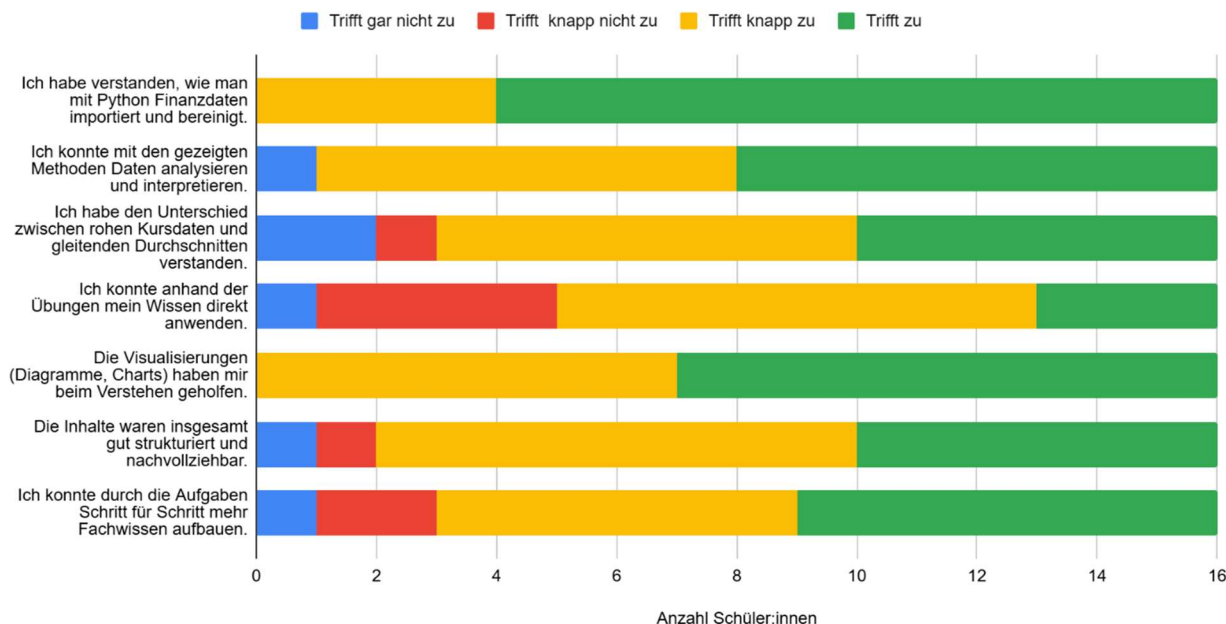


Abbildung 153: Umfrageresultate zu Inhalt und Kompetenzerwerb

Quantitative Einschätzung: Wirkung und persönlicher Lernerfolg

Die subjektive Wirkung der Unterrichtsreihe wurde verhaltener bewertet. Zwar wurde ein grundlegender Kompetenzerwerb bestätigt. Jedoch fielen die Sicherheit bei der Anwendung, das Interesse und das Vertrauen in den eigenen Lernerfolg weniger deutlich aus.

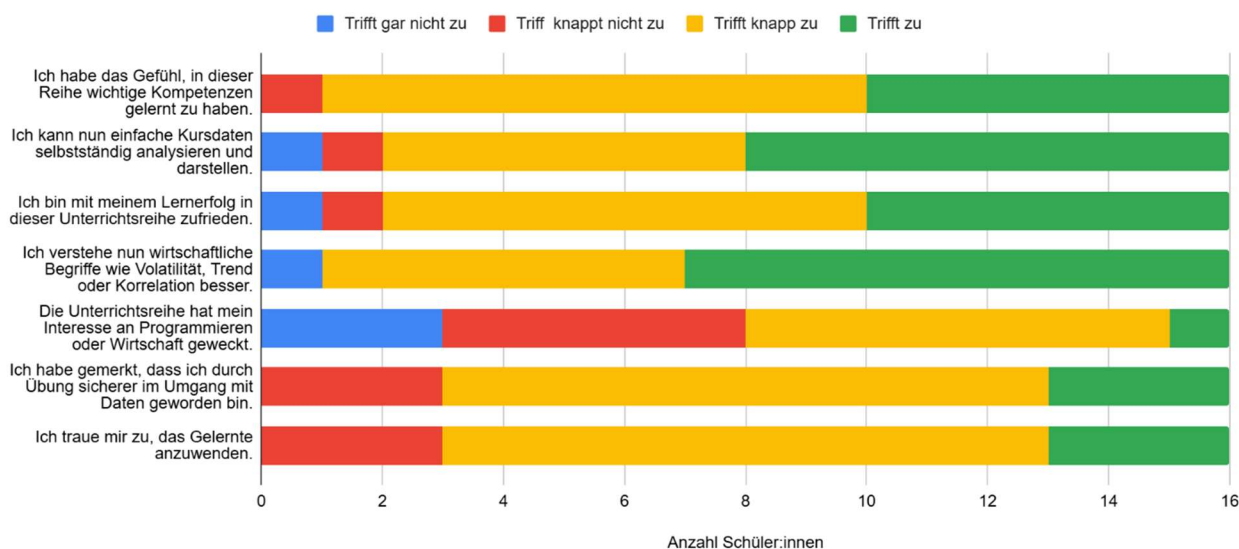


Abbildung 154: Umfrageresultate zur Wirkung und persönlichem Lernerfolg

Qualitative Einschätzung: Gesamtfazit

Die qualitativen Rückmeldungen bestätigen den Gesamteindruck aus den quantitativen Ergebnissen. Die Schüler:innen würdigen die strukturierte Gestaltung und differenzierte Aufgabenstellung der Unterrichtsreihe. Besonders positiv hervorgehoben wurden die Visualisierungen, Mini-Checks, die Einführung wirtschaftlicher Fachbegriffe und die nachvollziehbare Aufbereitung über mehrere Schwierigkeitsstufen hinweg. Gleichzeitig zeigen die Rückmeldungen aber auch deutlich, dass die Programmieranteile für viele herausfordernd bis teils überfordernd waren. Ein Grund dafür ist grundsätzliches Interesse, teils fehlende Grundkenntnisse und längere Pausen zum vorherigen Programmierblock. Trotz dieser Schwierigkeiten benennen viele Schüler:innen konkrete Lernfortschritte. Dies spricht für einen nachhaltigen Kompetenzzuwachs, auch wenn der Weg dahin für einige holprig war.

Qualitative Einschätzung:

Frage 1 «Was war für dich das Beste an dieser Unterrichtsreihe?»

Struktur, visuelle Lernhilfen und Praxisbezug werden als besonders gelungen wahrgenommen. Die Kombination aus Theorie, Übung und konkretem Codebezug motiviert viele. Die Schüler:innen schätzen besonders:

- Strukturierte Gliederung in Niveaustufen
- Grafische Visualisierung mit Python
- Mini-Checks und die klare Aufgabenabfolge
- Möglichkeit, mit echten Aktienbeispielen zu arbeiten
- gemeinsames Lösen der Aufgaben mit der Lehrperson

1. Was war für dich das Beste an dieser Unterrichtsreihe?

Dass die Lektionen in 3 Schwierigkeitsgraden aufgeteilt waren und man vor der Lektion schon die neuen Definitionen anschauen konnte.

Analysieren und Virtual Alpha

die gliederung mit den funktionen direkt erklärt, und erst dann die nach schwierigkeit aufgebauten aufgaben.

dass es mehr informatik war als wirtschaft

Wirtschaftliche Teil

Das wir gelernt haben wie man solche Liniendiagramme in python darstellt

Partnerarbeit Lektion 10

Die Kurztests

das arbeiten mit den aktien

Für mich waren die Mini-Checks hilfreich, um das gelernte zu prüfen, auch wenn ich sagen muss, dass ich mir die ähnlicher gewünscht hätte wie die eigentliche Prüfung. Weil die Mini-Checks waren wesentlich einfacher.

Wie man diagramme in python erstellt

Die Strukturierung des Unterrichts hat mir gut gefallen und so wusste ich wo wir was gelernt haben.

Die Wissenschecks, welche mir meinen Lernstand gezeigt haben.

Ich fand es gut, wenn sie jeweils die Aufgaben vorne mitgelöst haben, denn so habe ich es besser verstanden und bin auch besser mitgekommen. Ich fand es auch gut, dass sie am Anfang immer alle Begriffe kurz erklärt haben, das hat mir sehr geholfen.

die Repetitionen und Minichecks

Wie man Aktien analysieren kann und wann man kaufen soll und wann nicht.

Abbildung 155: Umfrageresultate zur 1. qualitativen Reflexionsfrage

Qualitative Einschätzung:

Frage 2 «Was war für dich besonders herausfordernd oder unklar?»

Die technischen und konzeptuellen Anforderungen waren für viele Schüler:innen hoch, teils auch zu hoch. Es fehlt teils an aktivem Vorwissenstraining und mehrmalige Wiederholungen. Als besonders herausfordernd empfinden die Schüler:innen:

- Das eigenständige Programmieren
- Funktionen wie Candlestick, resample() oder Expert-Aufgaben
- Das hohe Tempo, insbesondere bei der Einführung neuer Begriffe
- Unsicherheit im Syntaxverständnis, z.B. bei Klammern oder Importstrukturen
- Die längere Pause zur vorherigen Programmierphase

2. Was war für dich besonders herausfordernd oder unklar?

Selber auf die Codes zu kommen wäre für mich sehr schwierig

Programmieren

wo genau welche funktionen eingesetzt werden

Volatilität varianz

Programmieren von Expert/Advanced

Ganz Lektion 8

Candlestick Diagramme

Das Tempo, die Aufgaben selbst zu lösen

das manchmal in den expertaufgaben befehle vorkamen die wir noch nicht gehabt haben oder manchmal die befehle zu kurz erklärt wurden sind

Herausfordernd für mich bleibt weiterhin das programmieren, auch wenn es schon besser geworden ist, aber um dies wirklich gut zu können, haben wir zu wenig Informatik.

Die begriffe im code welche benutzt jedoch nie angeschaut wurden.

Für mich war das programmieren besonders herausfordernd sowie manchmal das Verständnis für Informatik.

Wie man die Expertaufgaben löst

Wenn ich etwas alleine programmieren musste, war das eher schwierig für mich.

viele neue Begriffe für das programmieren, auch z.B. wann man welche Klammern benutzt (beim programmieren)

Die python befehle zu lernen, weil wir fast ein ganzes Jahr ohne Python gearbeitet haben.

Abbildung 156: Umfrageresultate zur 2. qualitativen Reflexionsfrage

Qualitative Einschätzung:**Frage 3 «Welche Verbesserungsvorschläge gehen dir durch den Kopf?»**

Die Schüler:innen wünschen sich mehr Zeit, Klarheit und Übung von Programmiermethoden. Einzelne regen an, alternative Tools oder eine stärker gestufte Einführung zu verwenden. Vorgeschlagene Verbesserungen:

- Tempo bei komplexeren Lektionen noch mehr reduzieren
- Funktionen noch ausführlicher erklären, mit Beispielen und begleitenden Materialien
- Python-Basics mehr wiederholen am Anfang der Unterrichtsreihe
- Visual Studio Code statt Thonny für mehr Funktionalität benützen
- Begriffe im OneNote noch klarer dokumentieren
- Expert-Aufgaben mit Zwischenschritten zugänglicher gestalten
- Mini-Checks schwieriger gestalten, damit kleinere Diskrepanz zur Prüfung besteht. Alternativ Prüfung vereinfachen.

3. Welche Verbesserungsvorschläge gehen Dir durch den Kopf?

Je nachdem ein kleines bisschen mehr Zeit dafür dann einfach den Test um 1 Aufgabe länger z.B. Der Test war ein bisschen schwieriger als die Mini Checks aber ansonsten fand ich es ganz gut

Besser verteilen. Nicht mehrere Wochen durchgehend programmieren. Besser dies über längeren Zeitraum verteilen.

mehr üben

Anstatt thonny sollten alle schüler mit vsc ausgestattet werden da es um einiges hilfreicher ist als thonny und man neue sachen einfacher herunterladen kann

mehr Zeit für Lektionen am Ende (6,7,8,9)

Der Unterricht was sehr schnell und manche Funktionen wurden (für mich persönlich) noch unklar

keine

Die einzelnen Programmfunktionen genauer erklären.

an der prüfung war viel weniger programieren als an den wissens checks und im unterricht ich hatte sie dutlich anders erwartet

Vielleicht bisschen langsamer voran gehen, für die die nicht so geübt sind, weil es sind pro Lektion mind. 5 neue Definitionen dazu gekommen, was für mich persönlich teilweise zu viel war.

Jeden teil des codes zu erklären und zu definieren damit man auch versteht was es macht und damit man es anwenden kann nicht so wie z.B. resample und es dann auch noch in der Prüfung bringen

Das man evt. am Anfang nochmal das allg. Programmieren nochmals aufnimmt, weil für mich waren die ersten Lektionen sehr herausfordernd, da ich eine sehr lange Zeit nicht programmiert habe.

Den Schülern vielleicht etwas bessere Hilfe bieten.

keine

am Anfang Begriffe (Theorie) mehr anschauen, um in den Aufgaben besser nachzukommen

Begrifflichkeiten oder Befehle alle klar und besser zu definieren. Manche befehle haben sie bei der Prüfung abgefragt, aber sie waren in im Onenote nicht klar definiert. Ich finde auch, dass die Expertaufgaben für alle lösbar sein sollten , auch wenn es in der INF eine grosse "Schere " gibt. Ich finde auch dass er zu viel Informatik am Stück mit 8 Lektionen. Es ist unmöglich sich für so lange zu konzentrieren.

Abbildung 157: Umfrageresultate zur 3. qualitativen Reflexionsfrage

Qualitative Einschätzung:**Frage 4 «Was hast du persönlich aus der Reihe für dich mitgenommen?»**

Trotz Unsicherheit erkennen viele Schüler:innen ihren Lerngewinn. Sie können benennen, was sie fachlich und methodisch gelernt haben. Sie reflektieren offen über ihren Kompetenzzuwachs. Benannte Lernerträge:

- Verständnis für wirtschaftliche Fachbegriffe
- Grundlagen der Datenvisualisierung mit Python
- Sicherheit im Umgang mit Liniendiagrammen, axhline, matplotlib
- Kritisches Verständnis für Aktien und Marktanalysen
- Neue Einblicke in das Arbeiten mit Finanzdaten
- Erkenntnis: „Wenn man ins kalte Wasser geworfen wird, lernt man viel.“

4. Was hast du persönlich aus der Reihe für dich mitgenommen?

Die Fachbegriffe, wie man die Graphen liest und wie man die Codes versteht.

Wie man Daten analysiert und was für eine Arbeit Trader machen.

viele neue funktionen die aktien betreffend

matplotlib.pyplot verstanden

Fachbegriffe und Analysieren

Wie man (mit den Basics) solche Liniendiagramme darstellt mit axhline, etc

Viel Grundwissen von wirtschaftlichen Begriffen

Ich habe vieles gelernt, ich weiss jetzt wie man in Python Daten analysiert.

wenn man ein bisschen ins kalte wasser geworfen wird und schwierigkeiten hat lernt man viel

Wie man Daten einfach importiert, analysiert und in einem Liniendiagramm mit den wichtigsten Informationen darstellen kann.

wie man liniendiagramme erstellt

Neue Arten zu programmieren

Chat GPT kann wunder bewirken

Ich verstehe jetzt ein bisschen mehr zu Informatik und habe neue Begriffe gelernt.

viele neue Begriffe gelernt (auch Python-Begriffe)

Wie man Aktien analysieren kann

Abbildung 158: Umfrageresultate zur 4. qualitativen Reflexionsfrage

5.3.3 Auswertung Umfrage nach Unterrichtsreihe 2

Die Umfrage wurde am 25.6.2025 zum Ende der Unterrichtsreihe 2 durchgeführt. Die Umfrage nach der zweiten Unterrichtsreihe evaluiert die Wirkung des Projekts zur KI-gestützten Zeitreihenprognose aus Sicht der Schüler:innen. Dabei wurden sowohl quantitative Skalenfragen zur Selbsteinschätzung als auch qualitative Reflexionsfragen zur Projektwahrnehmung verwendet. Zusätzlich wurde erhoben, welche konkreten Arbeitsschritte im Projekt erfolgreich umgesetzt wurden. Die Umfrage wurde von 15 der insgesamt 16 Schüler:innen ausgefüllt, da ein Schüler aufgrund Krankheit am 25.6.2025 abwesend war.

Quantitative Einschätzung: Gesamtfazit

Die Umfrage zeigt einen positiven Gesamteindruck der Projektarbeit der Unterrichtsreihe 2. Die Schüler:innen konnten ihr Wissen anwenden und erweitern. Eigene Entscheidungen wurden getroffen, um die Projektarbeit umzusetzen. Das Arbeiten mit echten Daten und die Visualisierung der Ergebnisse wurden als besonders gelungen erlebt. Die grössten Herausforderungen lagen in der kritischen Reflexion der Modellgüte (MAPE, RMSE). Das selbständige Bearbeiten eines Projekts und die Erkenntnisgewinne zur Datenanalyse wurden oft als Erfolge dieser Phase genannt.

Quantitative Einschätzung: Subjektive Erfahrungseinschätzung

Die meisten Schüler:innen erlebten das Projekt als lehrreich, zugänglich und selbstwirksam. Eine deutliche Mehrheit fühlte sich sicher genug, um mit dem Projekt zu starten. Sie konnten ein Modell wählen, das ihrem individuellen Niveau entsprach. Ebenso wurde ein grundlegendes Verständnis für Zeitreihenprognosen erreicht. Besonders positiv wurde die Möglichkeit bewertet, eigene Ideen und Entscheidungen einzubringen. Ähnlich wie bei der Unterrichtsreihe 1 wurden die Visualisierungen hoch bewertet. Auch die zur Verfügung stehende Zeit wurde in der Regel als ausreichend empfunden. Der Lerneffekt war für viele greifbar, insbesondere im Hinblick auf die Anwendung von Prognosemodellen. Die Einschätzungen zur Relevanz des Projekts und zur Freude am selbstständigen Arbeiten fielen etwas differenzierter aus. Während ein Grossteil der Lernenden das Projekt als realitätsnah empfand, zeigte sich beim Wunsch nach einem weiteren datenbasierten Projekt eine gewisse Zurückhaltung. Dies kann als Hinweis auf die anspruchsvolle Natur der Aufgaben und den hohen Grad an Selbstverantwortung gedeutet werden.

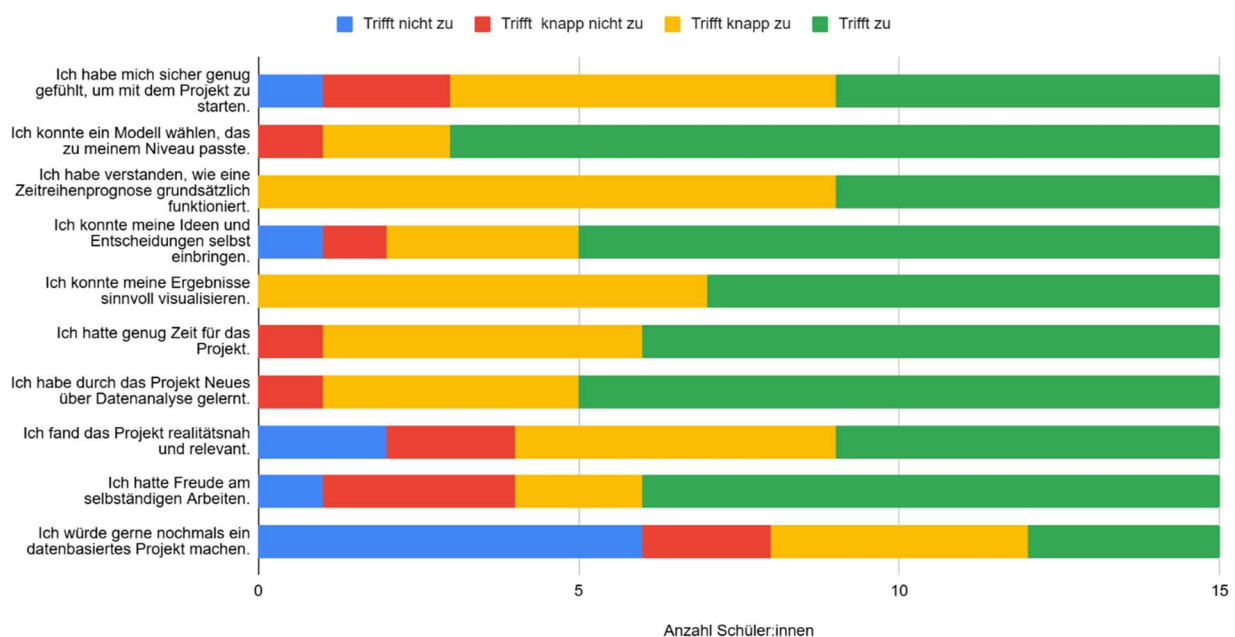


Abbildung 159: Umfrageresultate zur subjektiven Erfahrungseinschätzung

Quantitative Einschätzung: Persönlicher Projektfortschritt

Auch bei der Selbsteinschätzung zum persönlichen Projektfortschritt ergibt sich ein insgesamt positives Bild. Die meisten Schüler:innen konnten einen geeigneten Datensatz finden, diesen aufbereiten und ein Prognosemodell erfolgreich anwenden. In fast allen Fällen wurde eine sinnvolle Prognose erstellt und grafisch umgesetzt. Viele verglichen ihre Prognose mit realen Werten und hielten ihr Vorgehen nachvollziehbar fest. Besonders hervorzuheben ist die hohe Einschätzung der Zusammenarbeit im Team. Nahezu alle Gruppen erlebten diese als vollständig gelungen. Die Berechnung und Interpretation der Prognosegüte (MAPE/RMSE) hat einigen Teams Mühe bereitet. Insgesamt wird das Projektformat bestätigt, um selbstständiges Arbeiten in einem realitätsnahen Kontext zu fördern.

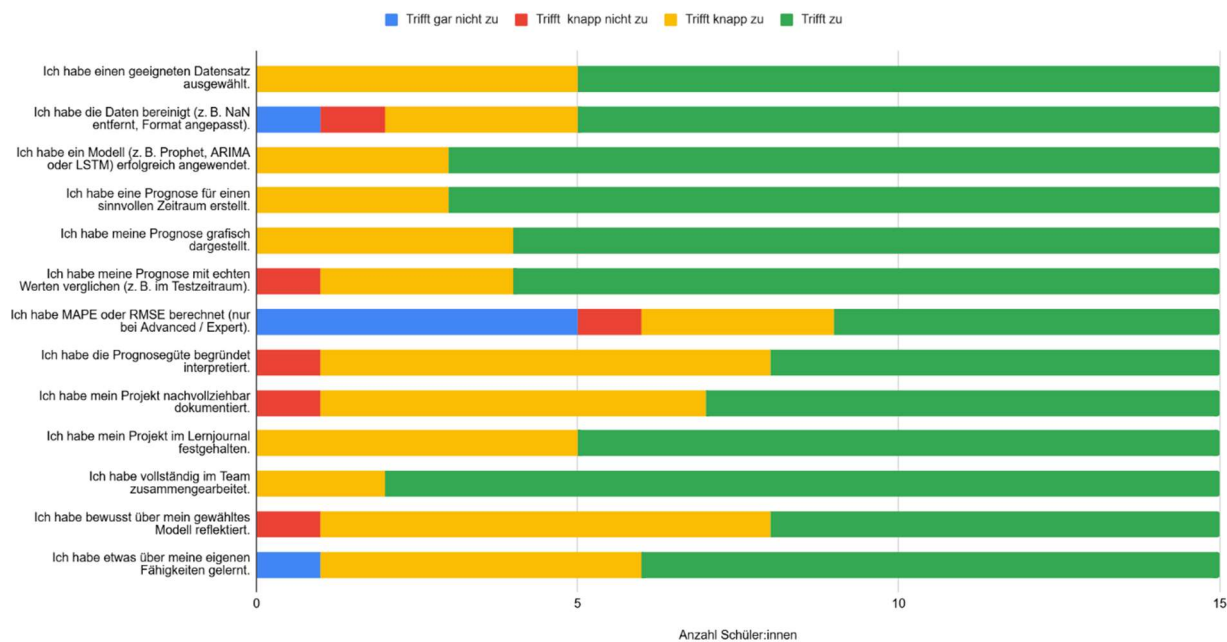


Abbildung 160: Umfrageresultate zum persönlichen Lernfortschritt

Qualitative Einschätzung:

Frage 1 «Was war für dich die wichtigste Erkenntnis beim Arbeiten mit einem Prognosemodell?»

Die Schüler:innen antworten heterogen bezüglich ihrer individuellen Lernerfahrungen im Umgang mit Prognosemodellen. Eine zentrale Erkenntnis war die Bedeutung von sorgfältigem und schrittweisem Arbeiten. Mehrere Rückmeldungen betonen die Wichtigkeit von Zwischenschritten beim Programmieren. Dies hilft ihnen beim Testen ihres Codes, um Fehler frühzeitig zu erkennen. Zudem wurde Geduld als entscheidender Erfolgsfaktor beim Programmieren erkannt. Besonders viel Geduld brauchte es bei komplexeren Modellen oder unerwarteten Fehlermeldungen. Ein weiterer wiederkehrender Punkt ist das Verständnis für die Grenzen von Prognosen. Einige Schüler:innen erkannten, dass Prognosemodelle unscharfe oder unzuverlässige Vorhersagen liefern können. In diesem Zusammenhang wurde reflektiert, dass ein komplexeres Modell nicht automatisch bessere Resultate liefert. Diese wertvolle Erkenntnis ist im Hinblick auf den bewussten und reflektierten Einsatz von Analysetools hilfreich. Auch wurden technisch-konzeptionelle Lernfortschritte genannt. Dabei wurden auch Konzepte aus der Unterrichtsreihe 1 wie z.B. Spaltenbenennung oder Indexsetzung gestärkt. Nebst der Vermittlung von technischem Wissen wurden auch bei den Schüler:innen auch Fehlertoleranz und Problemlöseverhalten gestärkt.

1. Was war für dich die wichtigste Erkenntnis beim Arbeiten mit einem Prognosemodell?

15 Antworten

Manchmal weichen Out of sample forecast etwas zu stark von der realen Kursdaten ab, weshalb man sich nicht zu sehr auf diese Prognosen verlassen sollten.

genau zu arbeiten, auf die Schreibfehler zu achten. Im allgemeinen immer wieder zu testen ob der Zwischenstand eines Codes funktioniert.

das es sehr kompliziert ist

Eine wichtige Erkenntnis war für mich zum Beispiel beim Index das man den zuerst ändern musste und dann mit dem neuen Index weitermachen musste, sowie war es wichtig das man das Projekt in kleinen Abschnitten testete sodass man so die Fehler schnell erkannte und verbessern konnte.

das Prophet die zeilen nicht einfach so lesen kann. Die Daten müssen als ds und y umgeschrieben werden müssen

Dass Programmieren viel Geduld braucht, aber sich lohnt

Prognosen können auch sehr ungenau sein.

Das man in der Lage ist, in die Zukunft zu schauen

Ich habe gelernt, dass man immer zuerst die kleinen Schritte testen sollte ob sie funktionieren und erst dann weiterprogrammieren soll.

Dass Tensorflow endlich ein update braucht.

Das es ist relativ zeitaufwändig zu programmieren

Dass das komplexere Modell nicht zwingend immer das genauere und effektivere ist.

Das Prognosen helfen, aber nicht immer 100% stimmen

gerne freie arbeit ins klate waasser geworfen und selbst rausfinden kniffeln

Viel probeln

Abbildung 161: Umfrageresultate zur 1. qualitativen Reflexionsfrage

Qualitative Einschätzung:

Frage 2 «Was fiel dir besonders schwer? Warum?»

Die grössten Herausforderungen der Schüler:innen lagen klar in der Programmierung. Viele Schüler:innen erlebten eine Vielzahl möglicher Fehlerquellen: Fehlende Klammern, falsch gesetzte Parameter oder unklare Fehlermeldungen. Dabei zeigte sich, dass kleine Änderungen im Code grosse Auswirkungen haben können und ein gutes Debugging-Verständnis nötig ist. Trotz dieser Schwierigkeiten zeigen viele Antworten, dass sich die Schüler:innen aktiv mit den Herausforderungen auseinandergesetzt haben. So konnten sie ihr Problemlösestrategien im Projektverlauf deutlich stärken.

2. Was fiel dir besonders schwer? warum?

15 Antworten

Der Beginn des Projekts war zwar etwas überfordernd, jedoch haben wir uns relativ schnell hineingefunden.
das programmieren-
fast alles
Mir fiel das Projekt an sich schwer, da ich nicht sehr geschickt in Informatik war. Aber ich habe mein Bestes gegeben.
30 Tage vom einem Datum abzuziehen (für Prophet)
LSTM zu sinnvollen Prognosen zu programmieren. Es hat viel Zeit gedauert.
Kleine Anpassungen ohne die der Code nicht funktioniert hat.
Ich habe bis am Schluss nicht ganz verstanden, was der Auftrag war.
Es fiel mir sehr schwer die Fehler zu finden, weshalb unser Programm nicht funktioniert hat.
LSTM programmieren war extrem schwierig
die daten richtig zu formatieren und länge festzusetzen vorallem für das RMSE
Das experimentieren mit Parametern war ziemlich kompliziert, wir schafften es aber schliesslich doch.
Am Anfang war für mich nicht ganz klar was die Aufgabe war.
der start
Programmieren, viele Befehle die ich nicht kannte.

Abbildung 162: Umfrageresultate zur 2. qualitativen Reflexionsfrage

Qualitative Einschätzung:

Frage 3 «Welches Prognosemodell hat dich am meisten überzeugt? Warum?»

Die Rückmeldungen zeigen ein differenziertes Bild der Wahrnehmung der drei Prognosemodelle Prophet, ARIMA und LSTM. Prophet überzeugte durch einfache Handhabung, ARIMA durch Genauigkeit und Verständlichkeit, LSTM durch hohe Prognosequalität (sofern die Implementierung gelang):

- Prophet wurde von mehreren Schüler:innen wegen seiner einfachen Anwendung und guten Grundgenauigkeit positiv hervorgehoben. Es wurde als „simpel“, „funktionierend“ und zugleich „leistungsfähig“ beschrieben. Somit wurde es als dasjenige Modell angewandt, das auch ohne tiefere Vorkenntnisse verständlich blieb und schnell Ergebnisse lieferte. Gerade für Schüler:innen auf Basic-Niveau war Prophet offenbar ein robustes und einsteigerfreundliches Werkzeug.
- ARIMA überzeugte bei einigen Schüler:innen durch seine Balance zwischen Genauigkeit und Verständlichkeit. Mehrere Rückmeldungen betonen, dass ARIMA präzise Prognosen liefert, gleichzeitig aber „relativ einfach anzuwenden“ und gut verständlich sei. Für Schüler:innen auf Advanced-Niveau stellte ARIMA daher ein gelungenes Modell dar, das analytische Tiefe mit praktischer Zugänglichkeit verbindet.
- LSTM wurde mehrfach als das genaueste Modell wahrgenommen. Dies war nur bei Top-Teams möglich, die es auch erfolgreich implementieren konnten. Sie lobten die Qualität der Prognose („sehr genau“, „realitätsnah“) und empfanden das Ergebnis als „sinnvoll“. Gleichzeitig war LSTM für viele Schüler:innen eine teils zu grosse Herausforderung, was aber auch aufgrund der Ausrichtung auf das Expert-Niveau nachvollziehbar ist.

3. Welches Prognosemodell hat dich am meisten überzeugt? warum?

15 Antworten

Arima, weil es am genauesten war und relativ simpel
Das für Basic, also Prophet
prophet simple und funktioniert
Wir haben Prophet und Arima ausprobiert und die beiden haben mich gleichfest überzeugt, jedoch sind die Prognosemodelle nicht immer korrekt.
Prophet, weil es simple ist, jedoch man kann viele damit machen
LSTM, da es eine sinnvolle Prognose angezeigt hat.
Ich habe nur eines benutzt.
LSTM war sehr genau.
Wir haben nur Advanced und Basic ausprobiert, deshalb würde ich sagen Advanced, weil es übersichtlicher dargestellt war als das Diagramm mit dem Prophet Modell.
LSTM es zeigte einen sinnvollen kurs an welcher auch in echt so passieren könnte
Arima, weil es ziemlich genau war und man es relativ einfach anwendet.
ARIMA, weil ich es einfach verstand und trotzdem die genauesten Prognosen ausführe
Ich habe nur mit Prophet gearbeitet, das war ganz okay
LSTM die prognose war sehr genau
Prophet

Abbildung 163: Umfrageresultate zur 3. qualitativen Reflexionsfrage

Qualitative Einschätzung:

Frage 4 «Was würdest du persönlich bei deiner nächsten Projektarbeit anders machen?»

Die Schülerantworten zeigen, dass viele aus der Projektarbeit konkrete Lernprozesse und Reflexionen für zukünftige Arbeiten mitgenommen haben. Mehrere Schüler:innen möchten künftig technisch tiefer einsteigen. Sie nennen Vorsätze wie „mehr Code zu verstehen“, „detaillierter analysieren“ oder auch „mutiger“ beim Programmieren sein. Andere formulieren, dass sie einen höheren Schwierigkeitsgrad anstreben würden oder mit mehr Datensätzen experimentieren. Einige Schüler:innen sehen bei sich Verbesserungspotential in der Aufgabenteilung im Team und Zeitplanung, um Zeitdruck am Ende zu vermeiden. Auch das schrittweise Testen des Codes wird als Methode genannt, um Probleme frühzeitig zu erkennen. Insgesamt zeigen die Antworten eine erfreuliche Selbstreflexion und vielfältige Ideen zur Verbesserung künftiger Projekte.

4. Was würdest du persönlich bei deiner nächsten Projektarbeit anders machen?

15 Antworten

Zeitplanung besser gestalten um am Ende weniger in Stress zu geraten.
Genauer zu arbeiten, dass man nicht zu vielen Fehler begegnet
-
Das weiss ich nicht
statt basic advanced
Ich würde bisschen mutiger sein zu Beginn da ich dachte ich würde garnichts können aber ich konnte gut mithelfen und eigene Ideen mit einbringen.
Vielleicht noch detaillierter den Code und Prognose Analysieren.
Ich würde es genau so machen
Ich würde nach jedem einzelnen Schritt den Programmcode testen zum schauen ob er auch wirklich funktioniert.
Nicht mit Datenanalyse arbeiten, da man dies in der Arbeitswelt schon genug machen muss und es nicht wirklich spass macht mit daten zu arbeiten. Es wäre um einiges lustiger geworden hätten wir ein game oder eine App programmiert(ich hätte einen Vorschlag: ein programm machen welches auf eine bibliothek zugreift in der sich musik dateien befinden und diese man abspielen kann(also eigentlich spotify programmieren))
Ich würde kaum was anderes machen
Von Anfang an besser koordinieren, wer was macht und die Planung etwas strukturierter machen.
Ich würde den Code detaillierter beschreiben
mehr datensätze mer probeln und probieren
Mehr Code verstehen

Abbildung 164: Umfrageresultate zur 4. qualitativen Reflexionsfrage

5.4 Gelungene Aspekte und Herausforderungen

In diesem Abschnitt möchte ich die gelungenen Aspekte und Herausforderungen erklären, die sich während der Vorbereitung, Durchführung und Nachbereitung der Unterrichtsreihe ergeben haben.

5.4.1 Gelungene Aspekte und Herausforderungen während der Vorbereitung

Die Vorbereitung der Unterrichtsreihe stellte in mehrfacher Hinsicht eine anspruchsvolle, aber auch lohnende Phase dar. Positiv hervorzuheben ist der systematische Aufbau der Unterrichtsreihe in klar gegliederte Kompetenzblöcke und Lektionen und Kompetenzblöcke. Die durchgängige Niveaudifferenzierung (Basic / Advanced / Expert) erwies sich in der Vorbereitungsphase als hilfreich. Sie hat eine übersichtliche Materialentwicklung für die Anwendung im Unterricht ermöglicht. Auch die Erstellung der Lektionen anhand eines standardisierten Lektionsablaufs (Impulsfragen, Begriffe/Funktionen, Aufgaben, Reflexion) sorgt für Klarheit. Besonders motivierend war das Ausprobieren und Testen erster Python-Codes mit Finanzmarktdaten. Ein funktionierender Code ist immer noch ein schönes direktes Erfolgserlebnis.

Während der Vorbereitung der Unterrichtsreihe gab es aber auch viele unterschiedliche Herausforderungen. Einige davon betrafen die fachliche Komplexität, andere den didaktischen Aufbau. Zuerst war es anspruchsvoll, Finanzdatenanalyse mit Python verständlich und motivierend aufzubereiten. Viele Methoden wie gleitende Durchschnitte und Standardabweichung sind technisch und mathematisch komplex. Ich musste diese Inhalte vereinfachen, ohne sie fachlich falsch darzustellen. Auch die Wahl der geeigneten Datenquellen war nicht immer einfach. Dienste wie yfinance funktionieren nicht immer zuverlässig, besonders in geschützten Schulnetzwerken. Deshalb habe ich mich entschieden, viele Daten in CSV-Dateien umzuwandeln. So sind sie unabhängig vom Internet nutzbar.

Eine weitere Knacknuss war die Niveaudifferenzierung. Die Klasse ist sehr heterogen. Nebst sehr leistungsstarken Schüler:innen gibt es auch eher unsichere Schüler:innen. Die Entwicklung sinnvoller Basic-, Advanced- und Expert-Aufgaben war eine grosse und zeitintensive Herausforderung. Zudem stellte sich heraus, dass einige Methoden wie z.B. die Korrelation nur schwer erklärbar sind. Hier half mir das Feedback meiner Betreuer, um den Fokus stärker auf visuelles Verstehen und Interpretation zu legen.

Schliesslich war es anspruchsvoll, alle Materialien didaktisch sinnvoll zu strukturieren und gleichzeitig technisch vollständig lauffähige Python-Beispiele bereitzustellen. Ich musste viel testen, korrigieren und vereinfachen, damit die Lernenden möglichst reibungslos mitarbeiten können. Trotz dieser Herausforderungen war die Vorbereitung sehr lehrreich. Ich habe fachlich und methodisch viel gelernt. Sie hat mir gezeigt, wie wichtig es ist, Inhalte schrittweise aufzubauen, visuell zu erklären und praxisnah zu gestalten.

5.4.2 Gelungene Aspekte und Herausforderungen der Unterrichtsreihe 1

In diesem Abschnitt möchte ich zentrale Herausforderungen sowie gelungene Aspekte der Umsetzung entlang der Kompetenzblöcke der Unterrichtsreihe 1 dokumentieren und reflektieren.

Kompetenzblock 1: Datenimport und -aufbereitung

Bereits die Vorbereitung bis zur Start der Aufgaben erforderte mehrfache technische und didaktische Anpassungen. Insbesondere das Einrichten der benötigten Programmierumgebung stellte sich als deutlich aufwendiger heraus als erwartet: Bei mehr als der Hälfte der Schüler:innen funktionierte die Installation der Bibliothek `yfinance` über die Windows-Eingabeaufforderung `cmd` nicht im ersten Versuch. Auch das Einlesen lokaler CSV-Dateien verursachte wiederholt Schwierigkeiten, was zu Unsicherheiten und Frustration führte. Eine Schritt-für-Schritt-Demonstration durch die Lehrperson war zwei Mal notwendig, um einen stabilen Start für alle zu gewährleisten.

Trotz dieser Startschwierigkeiten verlief die eigentliche Aufgabenbearbeitung sehr positiv: Alle Schüler:innen konnten die Basic-Aufgaben lösen. Ein Grossteil der Klasse konnte auch die Advanced-Aufgaben erledigen. Vier bis fünf Schüler:innen bearbeiteten sogar selbstständig die Expert-Stufe. Die Arbeitsatmosphäre war konzentriert. Es zeigte sich eine spürbare Motivation, insbesondere wenn ein Code funktionierte und die Ausgabe wie erwartet war. Viele Resultate konnten im Plenum gut interpretiert und diskutiert werden. Einzelne Schüler:innen fanden sogar schnellere oder elegantere Lösungen als die Lehrperson, was zu einem sehr lebendigen und produktiven Austausch führte.

Die Struktur der Lektionen (Impulsfragen, Begriffserklärung, praxisnahen Aufgaben auf drei Niveaus und abschliessenden Reflexionsfragen) wurde sowohl von den Schüler:innen als auch von mir als Lehrperson als stimmig und motivierend empfunden. Der wiederkehrende Aufbau half, anfängliche Nervosität abzulegen und Vertrauen in die eigene Kompetenz zu gewinnen.

Als grösste pädagogische Herausforderung zeigte sich eine verbreitete Grundhaltung des Typs „Ich kann das nicht“ oder „Informatik war noch nie mein Fach“. Hinzu kamen häufige Tippfehler oder Syntaxprobleme, die zu unmittelbaren Fehlern führten. In vielen Fällen wurde bei auftretender Fehlermeldung sofort die Lehrperson gerufen, anstatt sich selbst durch die Fehlermeldung oder Recherche zu arbeiten.

Kompetenzblock 2: Datenvisualisierung und -interpretation

Der zweite Kompetenzblock markierte einen Übergang von rein tabellarischer Datenbetrachtung hin zu graphischen Darstellungen. Inhaltlich lag der Fokus auf der Visualisierung von Zeitreihen mittels Liniendiagrammen mit `matplotlib` und interaktiven Candlestick-Charts mit `plotly`. Ergänzt wurde dies durch die Einführung gleitender Durchschnitte (SMA, EMA) zur Trendanalyse.

Ein gelungener Aspekt war, dass die Visualisierung den Schüler:innen ein greifbares Verständnis für Kursverläufe ermöglichte. Viele äusserten spontan, dass sie sich durch die Diagramme erstmals einen verständlichen Überblick über den Kursverlauf verschaffen konnten. Besonders die interaktive Komponente der `plotly`-Diagramme fand grossen Anklang.

Gleichzeitig offenbarten sich Herausforderungen bei der Einführung neuer Bibliotheken wie `plotly.graph_objects`. Begriffe wie `fig`, `go.Figure` oder OHLC waren für viele ungewohnt und führten zu Unsicherheit. Die meisten Schüler:innen konnten zwar den Code ausführen, hatten aber Mühe beim Verständnis der zugrundeliegenden Struktur. Auch Konzepte wie gleitende Durchschnitte waren zunächst abstrakt, obwohl sie rechnerisch lösbar waren. Hier zeigte sich ein Bedarf nach anschaulichen Erklärungen z.B. durch animierte Beispiele oder schrittweise Visualisierungen.

Didaktisch hilfreich erwies sich der wiederkehrende Aufbau der Lektionen mit Begriffseinführung, Niveaudifferenzierung und Reflexionsfragen. Auch die durch Schüler erstellten Zusammenfassungen wurden von den Schüler:innen aktiv genutzt und trugen zur Wissenskonsolidierung bei. Insgesamt konnte das Ziel, Daten visuell aufzubereiten und interpretieren zu können, auf breiter Basis erreicht werden.

Kompetenzblock 3: Datenbasierte Risiko- und Marktanalyse

Dieser dritte Block stellte die bisher höchsten kognitiven Anforderungen der ersten Unterrichtsreihe. Die Themen umfassten Berechnungen Volatilität und Korrelation. Die Anforderungen wurden deutlich spürbar höher, was sich auch in den Reaktionen der Schüler:innen zeigte.

Positiv aufgenommen wurde die wirtschaftliche Relevanz dieses Kompetenzblocks. Begriffe wie Volatilität und Marktrisiko wurden durch konkrete Beispiele greifbarer gemacht. Auch die Korrelation aus mehreren Datensätze führten zu spannenden Diskussionen im Plenum. Einige Schüler:innen zeigten grosse Eigeninitiative und erweiterten die Aufgaben kreativ. Die Schüler waren insgesamt aktiver in diesem Kompetenzblock als in den ersten zwei Kompetenzblöcken.

Jedoch offenbarte sich bei einigen Schülern auch die Schwierigkeit, abstrakte Konzepte wie Volatilität oder Korrelation mit den entsprechenden Python-Funktionen zu verknüpfen. Die Bedeutung der statistischen Masse wurde meist verstanden, ihre Anwendung blieb aber teils mechanisch. Eine noch stärkere Verbindung zwischen Theorie und Visualisierung würden den Schüler:innen helfen.

Insgesamt bewährte sich erneut die Differenzierung in Basic-, Advanced- und Expert-Niveau. Die Schüler:innen konnten sich entsprechend ihrer Fähigkeiten einbringen. Dennoch zeigte sich, dass der ursprüngliche Lektionsplan umfangmässig für einzelne zu viel auf einmal ist. Aufgrund dessen wurden die Lektion 7 und 8 von einer Einzellektion auf eine Doppellesson ausgeweitet.

Kompetenzblock 4: Repetition, summative Prüfung und Umfrage zur Unterrichtsreihe 1

Zur Repetition wurde die Peer-Teaching Methode eingesetzt. Dadurch konnten die Schüler:innen spezifische Unsicherheiten bei Programm-Codes, Konzepten oder Fehlermeldungen nochmals vertiefen. Schüler:innen, die sich in bestimmten Teilbereichen sicher fühlten, übernahmen die Rolle der Unterstützenden. Dies förderte sowohl das Selbstvertrauen der stärkeren Schüler:innen als auch das Verständnis bei denjenigen mit grösserem Unterstützungsbedarf.

Die summative Prüfung wurde mit Aufgaben aus allen Kompetenzblöcken aufgestellt. Die Resultate zeigten eine breite Kompetenzbasis. Die Mehrheit konnte die Aufgaben selbstständig lösen und alle Schüler:innen haben die Prüfung bei einem Klassenschnitt von 4.7 bestanden

Die Abschlussumfrage zur Unterrichtsreihe 1 lieferte wertvolle Einblicke. Viele Schüler:innen gaben an, dass sich ihre Einstellung zur Programmierung oder Datenanalyse durch die Reihe verbessert habe. Besonders hervorgehoben wurden die realitätsnahen Aufgaben, die klare Struktur der Lektionen sowie das Gefühl, „etwas wirklich verstanden zu haben“. Kritisch angemerkt wurden punktuell zu hohe Einstiegshürden bei neuen Bibliotheken sowie der Wunsch nach zusätzlichen visuellen Lernhilfen. Die Rückmeldungen bestätigten insgesamt, dass die Reihe zwar fordernd, aber auch motivierend und wirksam war.

5.4.3 Gelungene Aspekte und Herausforderungen der Unterrichtsreihe 2

Die zweite Unterrichtsreihe stellte die Anwendung datenbasierter Prognosemodelle in den Mittelpunkt. Inhaltlich wurde auf den technischen und analytischen Grundlagen der ersten Reihe aufgebaut. Der Fokus der Unterrichtsreihe lag auf den drei konkreten Modelltypen: Prophet, ARIMA und LSTM. Dabei rückten eigenständiges Arbeiten, kritisches Denken und der bewusste Umgang mit Modellen stärker in den Vordergrund.

Ein zentraler gelungener Aspekt war die Förderung eines reflektierten Umgangs mit Prognosemodellen. Gemäss Rückmeldungen haben die Schüler:innen mehrheitlich verstanden, dass Vorhersagen in der Finanzanalyse immer mit Unsicherheiten behaftet sind. Besonders wertvoll war die Erkenntnis, dass ein komplexeres Modell nicht automatisch zu besseren Ergebnissen führt. Diese Einsicht markiert einen wichtigen Schritt im Aufbau datenanalytischer Urteilskraft.

Auch methodisch-technisch zeigten sich Lernfortschritte: Viele Schüler:innen berichteten, dass sie die Bedeutung schrittweisen Arbeitens und systematischen Testens ihres Codes erkannt haben. Das Debugging-Verständnis konnte im Verlauf der Projektarbeit gestärkt werden. Besonders positiv fiel auf, dass Geduld und Fehlertoleranz zunehmend als Teil des Programmierprozesses akzeptiert wurden. Die Stärkung von Problemlöseverhalten war ein pädagogisch erfreuliches Ergebnis dieser Reihe.

Zu den Herausforderungen zählten vor allem technische Hürden bei der Programmierung. Typische Schwierigkeiten lagen in Details wie Klammerfehlern, unklaren Fehlermeldungen oder falsch gesetzten Parametern. Dies erforderte ein hohes Mass an Ausdauer und Sorgfalt. Dennoch zeigten die Antworten der Schüler:innen, dass sie sich aktiv mit diesen Problemen auseinandersetzten und daraus wichtige Kompetenzen entwickelten.

5.5 Verbesserungsvorschläge zur Unterrichtsgestaltung und -Struktur

Basierend auf den Erfahrungen aus der Vorbereitung und Durchführung der Unterrichtsreihen 1 und 2 ergeben sich mehrere konkrete Verbesserungsvorschläge gegenüber dem Plan. Diese strukturierten Verbesserungsvorschläge helfen, den Unterricht verständlicher zu gestalten. Die bisherige Struktur bleibt dabei erhalten, wird aber noch stärker schülerzentriert ergänzt.

Verbesserungsvorschläge zur Unterrichtsgestaltung

Gewisse Verbesserungen hat die Lehrperson bereits direkt in der Pilotklasse durchgeführt:

Verbesserungsvorschlag	Ausgestaltung / Erklärung
Technische Einstiegshürden reduzieren	- Vorbereitungslektion vor Modulstart zum Einrichten der Entwicklungsumgebung (Installation, Pfadüberprüfung, Import-Test) zusätzlich einplanen (de facto in Lektion 0 integriert) - Klare Installationsanleitung für yfinance, pandas, matplotlib etc.
Supportaufwand innerhalb der Klasse aufteilen	- Einführung einer Ersthilfe-Kaskade: 1) Selbst nachdenken und Fehlermeldung interpretieren, 2) Banknachbar:in fragen, 3) erst dann Lehrperson involvieren - Aktives Peer-Learning durch Umplatzierung: Schwächere Schüler:innen gezielt neben stärkere setzen - Erklärung von Debugging-Strategien mit print(), Thematisierung von Rechtschreibfehlern als häufigste Fehlerquelle und unterschiedliche Fehlermeldungen,
Selbstwirksamkeit gezielt fördern	- Zu Beginn jeder Lektion ein einfacher Funktionscode mit garantierter Funktionsfähigkeit zur Stärkung des Selbstvertrauens - Positive Rückmeldungskultur („Siehst du, das hast du selbst gelöst.“) - Mini-Erfolgserebnisse betonen
Mini-Checks als separate Lektion	Eigenständige Lektion zur Lernzielüberprüfung, gekoppelt mit Wiederholung und Temperaturmessung zur Motivation und Klarheit
Entschleunigung, mehr Lerncoaching	- Längere Bearbeitungszeit pro Lektion oder Splittung anspruchsvoller Inhalte auf mehrere Lektionen (z.B. Volatilität oder Plotly) - Aufteilen von Fachbegriffen in Etappen: zuerst konzeptuell, dann funktional, dann im Code
Glossar für alle Methoden	Aufbau eines begriffsorientierten Glossars (inkl. Beispielcodes), das sowohl zur Wiederholung als auch als Nachschlagewerk genutzt werden kann
Praxisbezug	- Verknüpfung zum Börsenspiel Virtual Alpha (läuft parallel zur Unterrichtsreihe) - Tips für Aktienselektion und Portfolio-Aufbau

Abbildung 165: Verbesserungsvorschläge (in aktueller Durchführung implementiert)

Folgende Verbesserungen können in zukünftigen Wiederholungen der Unterrichtsreihe geplant und durchgeführt werden:

Verbesserungsvorschlag	Ausgestaltung / Erklärung
Digitaler Projekt Kit	• Aufbau einer digitalen Werkzeugkiste mit : ○ Code-Snippets ○ CSVs Beispieldateien ○ Vorlagen zur Visualisierung ○ Python-Templates • aktuell sind obige Informationen auf OneNote, für Schüler:innen und Lehrperson zugreifbar
Theorie visualisieren und reduzieren	• Einführung schwieriger Begriffe wie Korrelation, Volatilität oder gleitender Durchschnitt zuerst anschaulich (Grafik, Animation), dann erst als Code • Einführung von Diagrammtypen wie Liniendiagramm, Candlestick und Scatterplot zuerst ohne Informatik erklären, dann erst mit Python programmieren. Erhöht Verständnis der Eignung von Diagrammtypen für entsprechende Daten • Einige Schüler:innen haben englische Begriffe nicht verstanden, z.B. „Mean“, „Index“, „resample“. Übersetzung in die deutsche Sprache und visuelle Begriffsdefinitionen
Umgang mit Unsicherheit explizit thematisieren	• Förderung von Fehlertoleranz und Debugging als explizite Unterrichtsziele • Thematisierung von „Scheitern gehört dazu“ oder „Fehlermeldung als Lernchance“ • Bisherige Reflexionsfragen waren teils zu abstrakt. Künftig sollten sie direkter auf die durchgeführten Aufgaben bezogen sein und konkrete „Aha“-Erlebnisse hervorrufen
Stärkere Einführung in komplexe Modelle	• Mehr Zeit für das Verständnis der mathematischen und konzeptionellen Grundlagen von ARIMA und LSTM einplanen. Dies kann mittels Visualisierungen oder einfachen Modellanwendungen geschehen.

Abbildung 166: Verbesserungsvorschläge (für zukünftige Durchführungen)

Verbesserungsvorschlag einer überarbeiteten Struktur der Unterrichtsreihe 1

Die ursprüngliche Struktur der Unterrichtsreihe 1 zur Finanzdatenanalyse mit Python erwies sich als inhaltlich wertvoll. Die praktische Durchführung war anspruchsvoll. Insbesondere gab es Hürden im Hinblick auf das Tempo, die Abfolge und den Einsatz der Programmierbefehle. Daher wurde der Ablauf der Kompetenzblöcke didaktisch überarbeitet.

Die überarbeitete Struktur der Unterrichtsreihe 1 beginnt mit motivierenden und visuell greifbaren Elementen. Technische und mathematisch anspruchsvolle Inhalte sollen später eingeführt werden. Erst nachdem Grundverständnis für Daten und deren Visualisierung aufgebaut wurde, werden komplexere Themen eingeführt. Die sind Datenbereinigung, Volatilitätsanalyse und gleitende Durchschnitte. Die gezielte Platzierung des CSV-Imports sowie die Verschiebung technischer Hürden in spätere Blöcke erhöht die Zugänglichkeit für alle.

Die schrittweise Progression reduziert die kognitive Überlastung. Sie fördert nachhaltiges Lernen und verbindet informatische Fähigkeiten mit ökonomischem Denken. Durch diesen Aufbau werden frühzeitig Erfolgserlebnisse geschaffen. Die Selbstwirksamkeit der Schüler:innen wird gestärkt.

Überarbeitete Kompetenzblöcke		Inhalt / Erklärung
1	Datenimport mit ersten Visualisierungen	Einstieg mit der Bibliothek yfinance, einfache Kursabfragen und direkte Visualisierung mit pyplot. So entsteht bereits in der ersten Lektion ein sichtbares Ergebnis. Technisch unkompliziert, motivierend und niedrigschwellig
2	Candlestick-Charts mit plotly	Chartdarstellung mit realen OHLC-Daten. Hier wird der Visualisierungsaspekt vertieft, um verschiedene Kursdarstellungen kennenzulernen
3	Korrelation verstehen und anwenden	Vergleich von Aktien mit einem Index, z.B. Nestlé vs. SMI. Lineare Zusammenhänge werden durch einfache Scatterplots und die Korrelation mathematisch wie visuell untersucht.
4	Datenimport & -bereinigung (CSV)	Erst jetzt erfolgen der Import und die technische Auseinandersetzung mit lokalen CSV-Daten. So sind die Schüler:innen bereits vertraut mit der Datenstruktur und sehen den Sinn hinter Befehlen wie <code>parse_dates</code>, <code>set_index</code>, <code>dropna()</code> oder <code>ffill()</code>.
5	Volatilität und Risikoanalyse	Die Streuung von Kursdaten wird quantitativ bewertet. Begriffe wie Standardabweichung oder relative Volatilität werden eingeübt und ökonomisch interpretiert.
6	Trendanalyse mit SMA, EMA und Crossovers	Technische Indikatoren wie gleitende Durchschnitte werden berechnet und analysiert. Diese Methoden werden in der Praxis zur Unterstützung von Investitionsentscheidungen verwendet
7	Prüfung und Reflexion	Abschliessend erstellen die Schüler:innen ein Mini-Dashboard, absolvieren die summative Prüfung und reflektieren ihren Lernprozess mittels Umfrage und Diskussion.

Abbildung 167: Verbesserungsvorschlag mit überarbeiteter Struktur der Unterrichtsreihe 1

Verbesserungsvorschlag einer überarbeiteten Struktur der Unterrichtsreihe 2

Im Rückblick hat sich gezeigt, dass die Gliederung der Unterrichtsreihe 2 entlang von Kompetenzblöcken weniger sinnvoll war als zunächst angenommen. Aufgrund des projektorientierten Charakters der Reihe eignet sich eine alternative Strukturierung deutlich besser. Statt Kompetenzblöcken bietet die Gliederung in Projektauftrag, Modellwahl (Prophet, ARIMA, LSTM) und Modellgüte eine deutlich schlüssigere und für die Schüler:innen besser greifbare Struktur. Sie spiegeln zugleich den tatsächlichen Verlauf und die Schwerpunkte der Unterrichtsreihe realitätsnäher wider.

6 Fazit

6.1 Zusammenfassung der Erkenntnisse

Erkenntnisse aus Unterrichtsreihe 1: Fachsystematik, Visualisierung, Diagnoseinstrumente

Die erste Unterrichtsreihe zeigte, dass ein systematischer und schrittweiser Aufbau im Bereich der Datenanalyse im schulischen Kontext sehr gut umsetzbar ist. Besonders wirksam erwies sich die klare Gliederung in Kompetenzblöcke, aufsteigende Schwierigkeitsstufen (Grundlagen, Fortgeschrittene, Expertengruppe) sowie ein wiederkehrender Ablauf mit Einstiegsfragen, Begriffsklärung, Aufgaben und abschliessender Reflexion. Diese strukturierte Gestaltung half den Schülerinnen und Schülern, Vertrauen in die eigenen Fähigkeiten aufzubauen und zentrale Konzepte schrittweise zu verankern.

Ein besonders motivierender Aspekt war der visuelle Zugang zu Daten. Die grafische Darstellung von Kursverläufen ermöglichte ein intuitiveres Verständnis. Didaktisch hilfreich war ausserdem der Einsatz von begleitenden Diagnoseinstrumenten wie kurzen Wissensüberprüfungen über Classtime oder Diskussionen im Plenum. So konnten Verständnisprobleme früh erkannt und gezielt aufgegriffen werden. Insgesamt legte die erste Unterrichtsreihe eine solide Grundlage für technische und wirtschaftliche Basiskompetenzen.

Erkenntnisse aus Unterrichtsreihe 2: Projektarbeit, Modellwahl, Reflexion

Die zweite Unterrichtsreihe verfolgte einen offeneren Ansatz mit starkem Fokus auf eigenständiger Projektarbeit. Die Schüler:innen konnten ihr Wissen in einem selbst gewählten Anwendungsbeispiel vertiefen. Im Zentrum stand die Auseinandersetzung mit Prognosemodellen wie Prophet, ARIMA und LSTM. Dabei entwickelten viele ein differenziertes Verständnis für die Stärken und Schwächen dieser Modelle, sowohl im Hinblick auf die technische Umsetzung als auch hinsichtlich ihrer Aussagekraft.

Ein zentrales Lernerlebnis war die Einsicht, dass komplexere Modelle nicht automatisch bessere Ergebnisse liefern. Dies förderte die kritische Auseinandersetzung mit der Aussagekraft von Prognosen und die bewusste Reflexion über den Einsatz von Analysetools. Die Projektform ermöglichte zudem echte Erfahrungen im Umgang mit Herausforderungen wie Fehlersuche, Zeitplanung und Zusammenarbeit im Team. Dabei wurde deutlich, dass Geduld, strukturierte Arbeitsweise und Reflexionsfähigkeit wesentliche Erfolgsfaktoren sind. Diese Fähigkeiten wirken über den Unterricht hinaus. Die in der zweiten Reihe gewählte Gliederung nach Projektauftrag, Modellwahl und Bewertung der Modellqualität erwies sich als deutlich passender als die bisherige Einteilung in Kompetenzblöcke. Sie entsprach stärker dem Arbeitsprozess und Denkweg der Schüler:innen

6.2 Persönlicher Rück- und Ausblick

Am Ende der Arbeit möchte ich einige übergeordnete Beobachtungen zu drei Fragen teilen, die sich aus der Vorbereitung und Durchführung der beiden Unterrichtsreihen ergeben haben und für andere Lehrpersonen und die Weiterentwicklung des Unterrichts relevant sein könnten.

Welche Bedingungen braucht interdisziplinärer Datenunterricht in der Schule?

Interdisziplinärer Unterricht mit datenanalytischem Fokus benötigt mehr als nur Zeit und Technik. Entscheidend sind klare Rahmenbedingungen, eine didaktisch durchdachte Struktur sowie eine Kultur des Umgangs mit Fehlern. Schüler:innen müssen lernen dürfen, dass Trial & Error zum Prozess dazugehört. Hilfreich sind kleinschrittige Vorlagen, strukturierte Reflexionsanlässe und transparente Bewertungskriterien.

Was würde ich einem Kollegen empfehlen, der ähnliche Themen unterrichten möchte?

Ich würde empfehlen, mit gut dokumentierten und realitätsnahen Beispielen zu beginnen und technische Hürden nicht zu unterschätzen. Vor allem aber lohnt es sich, mutig Projektformen zu erproben, auch wenn diese auf den ersten Blick herausfordernd wirken. Wichtig ist, den Schüler:innen ein Gefühl von Autonomie zu vermitteln, ohne sie dabei alleine zu lassen.

Welche Teile sind übertragbar, welche sind personengebunden?

Die Grundstruktur (Visualisierung, Analyse, Prognose) und der didaktische Aufbau sind sicher gut übertragbar. Auch Tools wie Classtime, Google Forms oder die Python Codes lassen sich leicht adaptieren. Personengebunden sind hingegen vermutlich gewisse technische Präferenzen (z.B. Modellwahl, Programmiersprache) und die Bereitschaft, Fehler im Unterricht produktiv aufzugreifen. Dennoch zeigt meine Erfahrung: Der Aufwand lohnt sich, sowohl für die Schüler:innen als auch für die Lehrperson.

Persönliche Einblicke und lernbezogene Beobachtungen

Mit dieser interdisziplinären Arbeit konnte ich ein Ziel verwirklichen, das mir von Beginn an wichtig war. Ich wollte eine praxisnahe Unterrichtsreihe zur Förderung von ökonomischem Denken und informatischer Analysekompetenz erschaffen. Besonders die Verbindung von Informatik und Wirtschaft hat mir persönlich grosse Freude bereitet. Dies hängt wohl mit meinem Interesse an Aktienmärkten und Datenanalysen zusammen. Die Arbeit an diesem Projekt war inhaltlich wie organisatorisch eine intensive, aber auch sehr bereichernde Erfahrung. Gleichzeitig wurde mir aber auch bewusst, wie herausfordernd die Aufbereitung fachlicher Inhalte ist. Dieser soll verständlich und motivierend sein. Oft zeigt sich der Erfolg erst beim Unterrichten der Inhalte.

Die Strukturierung der Unterrichtsreihen war eine erste grosse Hürde. Sie beinhaltete die top-down Einteilung in Kompetenzblöcke, Lektionen und Aufgabenniveaus. Nach einem ersten groben Entwurf der Strukturierung erstellte ich die ersten Lektionen, um die Struktur zu testen. Adjustierungen gemäss Feedback meines Betreuerteams habe ich so weit wie möglich eingearbeitet. Schnell wurde mir bewusst, dass die Schüler:innen wohl aus didaktischer Sicht mehr Zeit für Ein- / Ausstieg bei den verschiedenen Strukturelementen brauchten. So führte ich auf Lektionsebene die Impuls- und Reflexionsfragen nebst den Begriffserklärungen ein. Auf Stufe Kompetenzblock entwickelte ich die formativen Mini-Checks auf Classtime. Und zu guter Letzt erstellte ich auf Ebene Unterrichtsreihe je eine Umfrage auf Google. Nach der Erarbeitung der ersten Lektionen kam mir der Gedanke eines hybriden Aufbaus der Unterrichtsreihe 2. Dabei wollte ich einen klassische Lektionsaufbau mit einem selbstgesteuerten Projektauftrag verbinden. Ich traute mir dies und den Schüler:innen zu. Meine Idee stiess bei den Schüler:innen auf positiven Anklang. Dieser Projektauftrag hat mir wertvolle Erfahrungen aufgezeigt. Es war motivierend zu sehen, wie viel die Schüler:innen in den Projektdokumentationen geleistet haben. Dabei zeigte sich jedoch auch, dass die ursprünglich geplante Strukturierung der Unterrichtsreihe 2 entlang von Kompetenzblöcken nicht optimal zur Natur des Projekts passte. Stattdessen erwies sich eine Gliederung entlang der drei Kernelemente (Projektauftrag inklusive Bewertung und Dokumentation, drei Prognosemodellen sowie den Formen der Modellgüte) als deutlich sinnvoller. Diese Struktur entsprach besser dem tatsächlichen Arbeitsprozess der Schüler:innen und war für sie klarer nachvollziehbar. Besonders gefreut hat mich die Ernsthaftigkeit, mit der viele Schüler:innen ihre Projektarbeit am Ende des Schuljahres reflektiert haben. Diese Rückmeldungen haben mir gezeigt, dass datenbasierte Projektarbeit nicht nur fachliche, sondern auch metakognitive Lernprozesse in Gang setzt.

Danksagung

Mein herzlicher Dank gilt meinem Betreuerteam, Prof. Dr. Dennis Komm, Prof. Dr. Juraj Hromkovič und Regula Lacher. Ich danke für die engagierte und wertschätzende Begleitung. Ihr motivierendes Feedback und Rückfragen haben mich oft einen entscheidenden Schritt weitergebracht. Ebenso danke ich meinen Schüler:innen der Pilotklasse. Sie haben sich interessiert und kreativ auf ein für sie ungewohntes Projekt eingelassen. Ihre Ungeduld und gleichzeitig ihre Begeisterung für funktionierenden Code hat mir persönlich viel zurückgegeben. Ein Dank geht auch an meine Kolleg:innen in der Fachschaft Wirtschaft und Informatik, die mich in der Phase der Konzeptentwicklung unterstützt haben. Nicht zuletzt danke ich meiner Familie, die mir in arbeitsintensiven Phasen den Rücken freigehalten hat. Natürlich gäbe es immer noch Aspekte zu optimieren oder auszubauen. Doch gerade unter Berücksichtigung des Zeitrahmens, der Parallelbelastung durch Unterrichtsverpflichtungen und Maturakorrekturen bin ich froh, diese Arbeit abgeschlossen zu haben.

7 Quellen und Glossar: Verwendete Bibliotheken, Funktionen und Methoden

Die Erläuterungen der in dieser Arbeit verwendeten Python-Methoden und Funktionen basieren überwiegend auf der offiziellen Dokumentation der jeweiligen Bibliotheken: pandas, matplotlib, prophet, statsmodels, keras und yfinance:

- <https://pandas.pydata.org/docs>
- <https://matplotlib.org/stable/contents.html>
- <https://facebook.github.io/prophet/docs>
- <https://www.statsmodels.org/stable/index.html>
- <https://keras.io/api>
- <https://pypi.org/project/yfinance>

Verwendete Bibliotheken

Bibliotheken	Zweck
pandas	Datenstrukturen und -verarbeitung als DataFrame
matplotlib.pyplot	Diagramme und Visualisierungen
yfinance	Import historischer Börsendaten von Yahoo Finance
prophet	Zeitreihenprognose mit Trend- und Saisonalitätskomponenten
statsmodels	Statistische Modelle, v.a. ARIMA
keras	Deep Learning Framework zur Erstellung von LSTM-Netzen
sklearn.metrics	Fehlermasse wie MAPE, RMSE

Abbildung 168: Verwendete Bibliotheken in Python

Verwendete Methoden und Funktionen aus pandas

Methode/Funktion	Bedeutung
.read_csv()	Lädt CSV-Datei in ein DataFrame
.head() / .tail()	Zeigt die ersten / letzten Zeilen eines DataFrames
.columns	Listet alle Spaltennamen auf
.info()	Überblick über Spalten, Datentypen und fehlende Werte
.isnull() / .sum()	Prüft auf fehlende Werte, zählt sie
.dropna()	Entfernt Zeilen mit fehlenden Werten
.ffill()	Füllt fehlende Werte mit dem vorherigen Eintrag
.set_index()	Setzt eine Spalte als Index, z. B. Datum

<code>.to_numeric()</code>	Wandelt Strings in numerische Werte um
<code>.describe()</code>	Gibt statistische Kennwerte aus
<code>.rolling().mean()</code>	Gleitender Durchschnitt (SMA)
<code>.loc[]</code>	Filtert nach Indexwerten (z.B. Datum / Woche / Monat)

Abbildung 169: Verwendete Methoden und Funktionen aus pandas

Verwendete Funktionen aus matplotlib

Funktion	Bedeutung
<code>plt.plot()</code>	Liniendiagramm erstellen
<code>plt.scatter()</code>	Punktdiagramm erstellen (z. B. für Korrelationen)
<code>plt.title()</code>	Diagrammtitel setzen
<code>plt.xlabel()</code> / <code>.ylabel()</code>	Achsenbeschriftungen setzen
<code>plt.grid()</code> / <code>.legend()</code>	Gitterlinien und Legende einblenden

Abbildung 170: Verwendete Funktionen aus matplotlib

Verwendete Funktionen aus prophet

Funktion	Bedeutung
<code>Prophet()</code>	Erstellt ein Prophet-Modell
<code>.fit(df)</code>	Trainiert das Modell mit Daten
<code>.make_future_dataframe(periods=30)</code>	Erzeugt zukünftigen Zeitrahmen, hier für 30 Tage
<code>.predict()</code>	Berechnet Prognosewerte
<code>.plot()</code>	Visualisiert Prognose
<code>df.rename (columns = { 'Datum' : 'ds', 'Preis': 'y'}, inplace=True)</code>	Spalten in Prophet-Modell sollen ds (für Datastamp) und y (für Werte) heissen

Abbildung 171: Verwendete Funktionen aus prophet

Verwendete Funktionen aus statsmodels

Funktion	Bedeutung
<code>ARIMA(df, order=(p,d,q))</code>	Erstellt ARIMA-Modell mit festgelegten Parametern p, d und q: - p steht für Autoregression und nimmt die Anzahl früherer Zeitpunkte, die in die Vorhersage eingehen. Bei Tagesendkursen sind dies die Anzahl Tage. - d steht für Differenzierung, also wie oft die Daten differenziert werden, um sie stationär (stabil über die Zeit) zu machen. Stationarität erhöht die Prognosequalität. Bei d = 1 arbeitet das Modell nicht direkt mit den Originalwerten,

	sondern mit deren Unterschieden (z.B. heutiger Kurs minus gestriger Kurs). - q steht für die Fehler aus vorherigen Zeitpunkten, die in die Prognose einfließen.
<code>.fit()</code>	Trainiert das Modell
<code>.forecast(steps = 30)</code>	Erzeugt Vorhersage, hier mit 30 Tagen

Abbildung 172: Verwendete Funktionen aus statsmodels

Verwendete Funktionen aus tensorflow.keras und numpy

Funktion	Bedeutung
<code>values.reshape(-1, 1)</code>	<code>reshape(-1, 1)</code> formatiert die Daten in ein 2D-Array mit einer Spalte. Dies ist notwendig zur Anwendung eines Machine-Learning-Modells
<code>MinMaxScaler()</code>	<code>MinMaxScaler()</code> skaliert alle Werte auf einen Bereich zwischen 0 und 1, da LSTM-Netzwerke sensibel für Skalen sind. Rohdaten wie Aktienkurse führen ohne Skalierung zu schlechter Performance.
<code>.fit.transform(data)</code>	<code>.fit</code> berechnet die Minimal- und Maximalwerte des Datensatzes <code>.transform</code> Skaliert alle Werte zwischen 0 und 1
<code>Sequential()</code>	<code>Sequential()</code> initialisiert ein lineares, schichtbasiertes neuronales Netzwerk-Modell. Dabei werden die Schichten (Layers) nacheinander hinzugefügt.
<code>LSTM(units=50, activation='relu', input_shape=(X.shape[1], 1))</code>	<code>LSTM()</code> Erstellt eine LSTM-Schicht mit 50 Einheiten. <code>activation='relu'</code> : Aktivierungsfunktion <code>input_shape=(X.shape[1], 1)</code> : Form der Eingabedaten (z.B. 30 Tage mit je 1 Feature)
<code>model.compile(optimizer='adam', loss='mse')</code>	<code>.compile()</code> bereitet das LSTM für das Training vor. <code>optimizer='adam'</code> ist ein Optimisierungsalgorithmus <code>loss='mse'</code> ist die zu minimisierende Fehlerquote, hier «mean squared root»
<code>Dense(1)</code>	Fügt eine voll verbundene Schicht mit 1 Neuron hinzu. Gibt einen einzelnen numerischen Vorhersagewert zurück (hier z.B. Aktienkurs am nächsten Tag).
<code>model.fit(X, y, epochs=20,</code>	Trainiert das neuronale Netz mit Inputdaten X und Zielwerten y.

verbose=0)	epochs=20: Der gesamte Trainingsdatensatz wird 20× durchlaufen. Je mehr Durchläufe, desto genauer das Modell. verbose=0: Anzahl angezeigt Trainingsläufe in Konsole. Hier: Kein Trainingsfortschritt im Terminal angezeigt.
.predict()	Erstellt Prognose mit trainiertem Netz

Abbildung 173: Verwendete Funktionen aus tensorflow.keras und numpy

Abbildungsverzeichnis

Abbildung 1: Relevantes Vorwissen in Mathematik, Informatik und Wirtschaft.....	8
Abbildung 2: Übersicht Unterrichtsreihe mit Kompetenzblöcken	9
Abbildung 3: Übersicht Lektionen Einteilung je Kompetenzblock	10
Abbildung 4: Übersicht Lernziele Basic, Advanced und Expert je Kompetenzblock	11
Abbildung 5: Lernjournal in Google Forms.....	14
Abbildung 6: Unterrichtsreihe 2 - Lernjournal	14
Abbildung 7: Lektion 1 - Lernziele	15
Abbildung 8: Lektion 1 – Lektionsablauf	16
Abbildung 9: Lektion 1 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	16
Abbildung 10: Lektion 1 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	16
Abbildung 11: Lektion 1 - Niveau-differenzierte Übungen inkl. Musterlösungen	18
Abbildung 12: Lektion 1 – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	18
Abbildung 13: Lektion 2 - Lernziele	19
Abbildung 14: Lektion 2 - Lektionsablauf	19
Abbildung 15: Lektion 2 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	20
Abbildung 16: Lektion 2 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	20
Abbildung 17: Lektion 2 – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	20
Abbildung 18: Lektion 2 - Niveau-differenzierte Übungen inkl. Musterlösungen	22
Abbildung 19: Lektion 3 - Lernziele	23
Abbildung 20: Lektion 3 - Lektionsablauf	23
Abbildung 21: Lektion 3 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	24
Abbildung 22: Lektion 3 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	24
Abbildung 23: Lektion 3 – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	24
Abbildung 24: Lektion 3 - Niveau-differenzierte Übungen inkl. Musterlösungen	27
Abbildung 25: Formativer Mini-Check nach Kompetenzblock 1	28
Abbildung 26: Repetitionslektion Börsenweisheit - Lektionsablauf.....	29
Abbildung 27: Repetitionslektion Börsenweisheit – Klärungsfragen gemäss Temperaturmessung	30
Abbildung 28: Repetitionslektion Börsenweisheit – Überblick behandelte Themen und verwendete Methoden (Repetition).....	30
Abbildung 29: Repetitionslektion Börsenweisheit – Übungsdokumentation	31
Abbildung 30: Repetitionslektion Börsenweisheit – Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen.....	31
Abbildung 31: Lektion 4 - Lernziele	36
Abbildung 32: Lektion 4 - Lektionsablauf	36
Abbildung 33: Lektion 4 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	37
Abbildung 34: Lektion 4 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	37
Abbildung 35: Lektion 4 - Niveau-differenzierte Übungen inkl. Musterlösungen	40
Abbildung 36: Lektion 4 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	41
Abbildung 37: Lektion 5 - Lernziele	42
Abbildung 38: Lektion 5 - Lektionsablauf	42
Abbildung 39: Lektion 5 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	43
Abbildung 40: Lektion 5 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	43
Abbildung 41: Lektion 5 - Niveau-differenzierte Übungen inkl. Musterlösungen	46
Abbildung 42: Lektion 5 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	47
Abbildung 43: Lektion 6 - Lernziele	48
Abbildung 44: Lektion 6 - Lektionsablauf	48
Abbildung 45: Lektion 6 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	48

Abbildung 46: Lektion 6 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	49
Abbildung 47: Lektion 6 - Niveau-differenzierte Übungen inkl. Musterlösungen	52
Abbildung 48: Lektion 6 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	52
Abbildung 49: Formativer Mini-Check nach Kompetenzblock 2	53
Abbildung 50: Lektion 7 - Lernziele	54
Abbildung 51: Lektion 7 - Lektionsablauf	55
Abbildung 52: Lektion 7 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	55
Abbildung 53: Lektion 7 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	55
Abbildung 54: Lektion 7 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	56
Abbildung 55: Lektion 7 - Niveau-differenzierte Übungen inkl. Musterlösungen	58
Abbildung 56: Lektion 8 – Lernziele	59
Abbildung 57: Lektion 8 - Lektionsablauf	59
Abbildung 58: Lektion 8 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	59
Abbildung 59: Lektion 8 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	60
Abbildung 60: Lektion 8 - Niveau-differenzierte Übungen inkl. Musterlösungen	63
Abbildung 61: Lektion 8 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	63
Abbildung 62: Lektion 9 – Lernziele	64
Abbildung 63: Lektion 9 – Lektionsablauf	64
Abbildung 64: Lektion 9 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	64
Abbildung 65: Lektion 9 - Neue Begriffe / Konzepte / Funktionen inkl. Erklärungen	65
Abbildung 66: Lektion 9 - Reflexionsfragen zum Lektionsabschluss inkl. Musterlösungen	65
Abbildung 67: Lektion 9 – Niveau-differenzierte Übungen inkl. Musterlösungen.....	68
Abbildung 68: Formativer Mini-Check nach Kompetenzblock 3	69
Abbildung 69: Lektion 10 - Lernziele	70
Abbildung 70: Lektion 10 - Lektionsablauf	70
Abbildung 71: Lektion 10 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	71
Abbildung 72: Lektion 10 – Übungen	73
Abbildung 73: Lektion 10 - Reflexionsfragen zum Lektionsabschluss	73
Abbildung 74: Übersicht Aufgaben der summativen Prüfung	74
Abbildung 75: Aufgabe 1 der summativen Prüfung	75
Abbildung 76: Aufgabe 2 der summativen Prüfung	76
Abbildung 77: Aufgabe 3 der summativen Prüfung	77
Abbildung 78: Aufgabe 4 der summativen Prüfung	78
Abbildung 79: Aufgabe 5 der summativen Prüfung	79
Abbildung 80: Aufgabe 6 der summativen Prüfung	80
Abbildung 81: Aufgabe 7 der summativen Prüfung	81
Abbildung 82: Aufgabe 8 der summativen Prüfung	82
Abbildung 83: Sicherheitseinstellungen der summativen Prüfung	83
Abbildung 84: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 1.....	84
Abbildung 85: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 2.....	85
Abbildung 86: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 3.....	85
Abbildung 87: Umfrage nach Unterrichtsreihe 1, Teil B, Fragen 1-4	85
Abbildung 88: Unterrichtsreihe 2 - Lernziele	86
Abbildung 89: Unterrichtsreihe 2 – Erklärung Prognosemodelle und Gütemasse	87
Abbildung 90: Lektion 13 - Lernziele	88
Abbildung 91: Lektion 13 - Lektionsablauf	88
Abbildung 92: Lektion 13 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	89
Abbildung 93: Lektion 13 - Reflexionsfragen zum Lektionsabschluss	89

Abbildung 94: Lektion 14 – Lernziele	89
Abbildung 95: Lektion 14 - Lektionsablauf	90
Abbildung 96: Lektion 14 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	90
Abbildung 97: Lektion 14 - Reflexionsfragen zum Lektionsabschluss	90
Abbildung 98: Lektion 15 - Lernziele	91
Abbildung 99: Lektion 15 - Lektionsablauf	91
Abbildung 100: Lektion 15 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	92
Abbildung 101: Lektion 15 – Verwendete Funktionen aus prophet	92
Abbildung 102: Lektion 15 - Reflexionsfragen zum Lektionsabschluss	92
Abbildung 103: Lektion 15 – Übung als Live-Demo	94
Abbildung 104: Lektion 16 - Lernziele	95
Abbildung 105: Lektion 16 - Lektionsablauf	95
Abbildung 106: Lektion 16 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	95
Abbildung 107: Lektion 16 - Verwendete Funktionen aus statsmodels	96
Abbildung 108: Lektion 16 - Reflexionsfragen zum Lektionsabschluss	96
Abbildung 109: Lektion 16 – Übung als Live-Demo	99
Abbildung 110: Lektion 17 - Lernziele	100
Abbildung 111: Lektion 17 - Lektionsablauf	100
Abbildung 112: Lektion 17 – Impulsfragen zum Lektionsstart inkl. Musterlösungen	100
Abbildung 113: Lektion 17 – Übung MAPE	101
Abbildung 114: Lektion 17 – Übung RMSE	102
Abbildung 115: Lektion 17 - Reflexionsfragen zum Lektionsabschluss	102
Abbildung 116: Lektion 18 - Lernziele	103
Abbildung 117: Lektion 18 – Lektionsablauf	103
Abbildung 118: Verwendete Funktionen aus tensorflow.keras und numpy	104
Abbildung 119: Lektion 18 – Übung	106
Abbildung 120: Unterrichtsreihe 2 – Projektphasen und Anleitung für Schüler:innen	108
Abbildung 121: Unterrichtsreihe 2 - Lernjournal	108
Abbildung 122: Unterrichtsreihe - Reflexionsfragen	108
Abbildung 123: Unterrichtsreihe 2 – Bewertungskriterien	109
Abbildung 124: Unterrichtsreihe 2 – Musterlösung Basic	110
Abbildung 125: Unterrichtsreihe 2 – Musterlösung Advanced	111
Abbildung 126: Unterrichtsreihe 2 – Musterlösung Expert	112
Abbildung 127: Umfrage nach Unterrichtsreihe 2, Teil A, Frage 1	113
Abbildung 128: Umfrage nach Unterrichtsreihe 1, Teil A, Frage 2	114
Abbildung 129: Umfrage nach Unterrichtsreihe 2, Teil B	115
Abbildung 130: Durchschnittlicher Erfüllungsgrad aller Aufgaben Lektionen 1-7 gemäss Lernjournal	116
Abbildung 131: Erfüllungsgrad Basic Aufgaben gemäss Lernjournal	117
Abbildung 132: Erfüllungsgrad Advanced Aufgaben gemäss Lernjournal	117
Abbildung 133: Erfüllungsgrad Expert Aufgaben gemäss Lernjournal	118
Abbildung 134: Mini-Check nach Kompetenzblock 1 - Resultate	119
Abbildung 135: Mini-Check nach Kompetenzblock 1 - Auswertung	119
Abbildung 136: Mini-Check nach Kompetenzblock 2 - Resultate	120
Abbildung 137: Mini-Check nach Kompetenzblock 2 – Auswertung	120
Abbildung 138: Mini-Check nach Kompetenzblock 3 - Resultate	121
Abbildung 139: Mini-Check nach Kompetenzblock 3 – Auswertung	121
Abbildung 140: Summative Prüfung in Kompetenzblock 4 - Resultate	122
Abbildung 141: Summative Prüfung in Kompetenzblock 4 - Noten	123

Abbildung 142: Summative Prüfung in Kompetenzblock 4 – Auswertung nach Aufgaben	123
Abbildung 143: Summative Prüfung in Kompetenzblock 4 – Interpretation der Ergebnisse	124
Abbildung 144: Auswertung Projektarbeiten - Noten.....	125
Abbildung 145: Projektarbeiten der Unterrichtsreihe 2 – Würdigung.....	126
Abbildung 146: Projektarbeiten der Unterrichtsreihe 2 – Screenshot von Team1.....	127
Abbildung 147: Projektarbeiten der Unterrichtsreihe 2 – Screenshot von Team2.....	127
Abbildung 148: Projektarbeiten der Unterrichtsreihe 2 – Screenshot von Team3.....	128
Abbildung 149: Temperaturmessung nach Kompetenzblock 1	129
Abbildung 150: Temperaturmessung nach Kompetenzblock 2	130
Abbildung 151: Temperaturmessung nach Kompetenzblock 3	131
Abbildung 152: Umfrageresultate zu Unterrichtsaufbau und Methodik.....	132
Abbildung 153: Umfrageresultate zu Inhalt und Kompetenzerwerb.....	133
Abbildung 154: Umfrageresultate zur Wirkung und persönlichem Lernerfolg.....	133
Abbildung 155: Umfrageresultate zur 1. qualitativen Reflexionsfrage.....	135
Abbildung 156: Umfrageresultate zur 2. qualitativen Reflexionsfrage.....	135
Abbildung 157: Umfrageresultate zur 3. qualitativen Reflexionsfrage.....	136
Abbildung 158: Umfrageresultate zur 4. qualitativen Reflexionsfrage.....	137
Abbildung 159: Umfrageresultate zur subjektiven Erfahrungseinschätzung.....	138
Abbildung 160: Umfrageresultate zum persönlichen Lernfortschritt.....	139
Abbildung 161: Umfrageresultate zur 1. qualitativen Reflexionsfrage.....	140
Abbildung 162: Umfrageresultate zur 2. qualitativen Reflexionsfrage.....	141
Abbildung 163: Umfrageresultate zur 3. qualitativen Reflexionsfrage.....	142
Abbildung 164: Umfrageresultate zur 4. qualitativen Reflexionsfrage.....	143
Abbildung 165: Verbesserungsvorschläge (in aktueller Durchführung implementiert)	148
Abbildung 166: Verbesserungsvorschläge (für zukünftige Durchführungen)	149
Abbildung 167: Verbesserungsvorschlag mit überarbeiteter Struktur der Unterrichtsreihe 1	150
Abbildung 168: Verwendete Bibliotheken in Python	154
Abbildung 169: Verwendete Methoden und Funktionen aus pandas	155
Abbildung 170: Verwendete Funktionen aus matplotlib	155
Abbildung 171: Verwendete Funktionen aus prophet.....	155
Abbildung 172: Verwendete Funktionen aus statsmodels	156
Abbildung 173: Verwendete Funktionen aus tensorflow.keras und numpy	157