

**MSc Wirtschaftsinformatik
DIFA-Projekt**

Horizontal Patterns in Currency Pairs

Eingereicht am 03.01.2025 von

Kuonen Noah, Business Analyst, noahkuonen@gmail.com

Bushi Pjeter, Technical Consultant, pjeter.bushi@stud.hslu.ch

Inhaltsverzeichnis

1	Abstract	3
2	Einleitung	4
2.1	Ziel des Projekts	5
3	Theoretischer Hintergrund	6
3.1	Definition und Bedeutung von Unterstützung und Widerstand	6
4	Projektplanung	7
5	Datenbeschreibung	8
5.1	Datenquelle	8
5.2	Datenformat und Inhalt	8
5.3	Verarbeitung der Daten	9
5.4	Bedeutung der Daten	9
5.5	Visualisierung der Daten und Ergebnisse	9
5.5.1	Plot 1 – Schlusskurse von EUR/USD	10
5.5.2	Plot 1 – Bedeutung	10
5.5.3	Plot 2 – Schlusskurse mit Unterstützungs- und Widerstandsniveaus	11
5.5.4	Plot 2 – Bedeutung	11
5.5.5	Plot 3 – Trades	12
5.5.6	Plot 3 – Bedeutung	13
5.5.7	Plot 4 – Gleitende Durchschnitte und Trades	13
5.5.8	Plot 4 – Bedeutung	14
5.5.9	Plot 5 – Bollinger Bands	15
5.5.10	Plot 5 – Begründung	16
5.5.11	Plot 6 – Relative Strength Index (RSI)	16
5.5.12	Plot 6 – Begründung	17
5.6	Fazit Ergebnisse	17
6	Reflektion	18
6.1	Fazit zum Projekt	20
6.2	Herausforderungen und Schwierigkeiten	20
6.3	Mögliche zukünftige Erweiterungen	21
7	A Anhang	22
7.1	Abbildungsverzeichnis	22
7.2	Tabellenverzeichnis	22
7.3	Code:	22

1 Abstract

Diese Arbeit untersucht die Analyse horizontaler Muster im Devisenhandel am Beispiel des Währungspaares EUR/USD. Ziel ist die Entwicklung einer algorithmischen Handelsstrategie basierend auf Unterstützungs- und Widerstandsniveaus. Mithilfe von Python wurden historische Marktdaten analysiert, technische Indikatoren wie RSI, Bollinger Bands und gleitende Durchschnitte integriert sowie Stop-Loss- und Take-Profit-Mechanismen implementiert.

Das Backtesting zeigte, dass die Strategie in Seitwärtsmärkten solide Ergebnisse liefert und die Buy-and-Hold-Strategie übertrifft. Limitierungen bestehen jedoch in volatilen Märkten. Die Arbeit verdeutlicht das Potenzial algorithmischer Ansätze für präzisere Handelsentscheidungen und liefert eine Grundlage für zukünftige Erweiterungen, wie Machine Learning und Echtzeit-Handel.

2 Einleitung

Im Zuge der digitalen Transformation des Finanzsektors gewinnt der Bereich Decentralized Finance (DeFi) zunehmend an Bedeutung. DeFi bietet die Möglichkeit, Finanzdienstleistungen unabhängig von traditionellen, zentralisierten Institutionen wie Banken oder Finanzintermediären anzubieten. Mithilfe neuer Technologien wie Blockchain und Kryptowährungen können Finanzdienstleistungen nicht nur schneller und effizienter, sondern vor allem auch für eine grössere Zielgruppe zugänglich gemacht werden. Das Ziel von DeFi ist es, eine vollständig dezentralisierte, transparente und vertrauenslose Infrastruktur für den globalen Finanzmarkt zu schaffen.

Ein wichtiger Aspekt des erfolgreichen Handels ist das Verständnis von Markttrends und der Fähigkeit, wiederkehrende Muster in Kursbewegungen zu erkennen. In diesem Zusammenhang spielen sogenannte Horizontal Patterns oder horizontale Muster eine wichtige Rolle. Diese Muster treten auf, wenn sich der Kurs eines Währungspaares über einen längeren Zeitraum innerhalb eines bestimmten Preisbereichs bewegt und weder signifikante Anstiege noch starke Abwärtstrends zeigt. Solche Bewegungen deuten häufig darauf hin, dass der Markt eine Phase der Konsolidierung durchläuft, in der sich Angebot und Nachfrage die Waage halten.

Horizontal Patterns bieten wertvolle Informationen, um Unterstützungs- und Widerstandsniveaus zu identifizieren. Diese Niveaus sind Preisschwellen, an denen der Kurs eines Währungspaares entweder häufig nach unten (Unterstützung) oder nach oben (Widerstand) abprallt. In der Praxis bedeutet dies, dass bei einem Unterstützungsniveau der Kurs dazu tendiert, nicht weiter zu fallen, da ein starkes Kaufinteresse besteht. Auf der anderen Seite signalisiert ein Widerstandsniveau, dass der Kurs Schwierigkeiten hat, weiter zu steigen, da in diesem Bereich verstärkt Verkaufsdruck auftritt. Im Rahmen dieser Arbeit wurde die technische Analyse durch die Integration fortschrittlicher Indikatoren wie RSI, Bollinger Bands und gleitender Durchschnitte ergänzt. Diese Ansätze ermöglichen es, präzisere Handelsentscheidungen zu treffen und Risiken durch Mechanismen wie Stop-Loss und Take-Profit gezielt zu steuern.

2.1 Ziel des Projekts

Im Rahmen unseres Projekts wollen wir diese Horizontal Patterns gezielt identifizieren und nutzen, um eine Handelsstrategie zu entwickeln. Konkret haben wir uns dazu entschlossen, ein Python-Skript zu programmieren, das historische Kursdaten von Währungspaaren (z. B. EUR/USD) analysiert. Das Skript soll in der Lage sein, horizontale Unterstützungs- und Widerstandsniveaus automatisch zu erkennen und diese grafisch in einem Kursdiagramm darzustellen. Diese Visualisierung erleichtert es, potenziell profitable Kurszonen zu identifizieren, in denen der Kurs wahrscheinlich drehen oder stagnieren wird.

Zusätzlich zur reinen Identifikation der Muster wollen wir unser System mithilfe von Backtesting-Strategien überprüfen. Beim Backtesting simulieren wir, wie sich unsere Handelsstrategie in der Vergangenheit entwickelt hätte, indem wir die identifizierten Unterstützungs- und Widerstandsniveaus auf historische Daten anwenden. Dadurch können wir die Rentabilität und Zuverlässigkeit des Systems testen, bevor wir es in einem realen Marktumfeld anwenden.

Die Strategie wurde durch die Integration fortschrittlicher technischer Indikatoren wie dem Relative Strength Index (RSI) und Bollinger Bands erweitert. Diese Indikatoren ermöglichen es, überkaufte oder überverkaufte Marktbedingungen sowie die Marktvolatilität zu identifizieren, was präzisere Handelsentscheidungen unterstützt. Darüber hinaus wurden Stop-Loss- und Take-Profit-Mechanismen eingeführt, um Risiken besser zu steuern und Gewinne gezielt zu sichern.

Unsere Arbeit legt einen klaren Fokus auf Risiko- und Ertragsbewertung. Langfristig könnte ein solches System auch in den Bereich des automatisierten Handels übergehen, wo es mithilfe von Echtzeitdaten selbstständig Entscheidungen über Kauf- und Verkaufsaktionen treffen könnte. In Zukunft könnte der Einsatz von Machine-Learning-Algorithmen die Erkennung horizontaler Muster und die Vorhersage von Kursbewegungen weiter optimieren.

3 Theoretischer Hintergrund

Horizontale Muster, insbesondere horizontale Unterstützungs- und Widerstandsniveaus (HSARs), sind zentrale Elemente der technischen Analyse und helfen, potenzielle Wendepunkte im Markt zu identifizieren. Unterstützungs-niveaus zeichnen sich durch erhöhten Kaufdruck aus, der den Kurs stabilisiert oder steigen lässt, während Widerstandsniveaus durch erhöhten Verkaufsdruck geprägt sind, der den Kurs sinken lässt.

Ergänzend wurden in diesem Projekt weitere technische Indikatoren wie der Relative Strength Index (RSI) und Bollinger Bands verwendet. Der RSI misst, ob ein Finanzinstrument überkauft (Werte > 70) oder überverkauft (Werte < 30) ist, und unterstützt so die Entscheidungsfindung. Bollinger Bands hingegen visualisieren die Volatilität und potenzielle Trendwenden anhand eines gleitenden Durchschnitts und zweier Bänder, die die Standardabweichung darstellen.

Zusätzlich wurden gleitende Durchschnitte (50 und 200 Tage) verwendet, um längerfristige Trends zu erkennen und Kauf- oder Verkaufssignale zu bestätigen. Diese Indikatoren ergänzen die horizontalen Unterstützungs- und Widerstandsniveaus und steigern die Präzision der Handelsstrategie.

3.1 Definition und Bedeutung von Unterstützung und Widerstand

Unterstützung ist ein Preisniveau unterhalb des aktuellen Kurses, bei dem die Nachfrage in der Regel zunimmt und den Preis stabilisiert oder ansteigen lässt. Händler betrachten diese Niveaus oft als Punkte, an denen das Asset unterbewertet ist, was den Kaufdruck erhöht. Widerstand hingegen ist ein Preisniveau oberhalb des aktuellen Kurses, bei dem der Verkaufsdruck den Kaufdruck übersteigt, wodurch der Preis fällt oder stagniert. Diese Niveaus sind von grosser Bedeutung, da sie als psychologische Barrieren für Händler fungieren und ihre Erwartungen an die zukünftige Preisentwicklung prägen. Sobald der Preis ein Unterstützungs- oder Widerstandsniveau durchbricht, kommt es oft zu erheblichen Marktbewegungen, was klare Kauf- oder Verkaufssignale liefert. In diesem Projekt wurden Unterstützungs- und

Widerstandsniveaus durch rollierende Höchst- und Tiefstwerte der letzten 20 Tage berechnet. Diese Methode ermöglicht eine dynamische Anpassung an aktuelle Marktbedingungen und unterstützt die Analyse horizontaler Muster.

4 Projektplanung

Nr.	Meilenstein	Eintrittszeitpunkt	Beschreibung
1	Start Projekt	01.10.2024	Beim ersten Termin haben wir uns über das Thema sowie unsere Methodik ausgetauscht.
2	Aufteilung Arbeiten	08.10.2024	Im zweiten Termin haben wir die Aufgaben untereinander aufgeteilt.
3	Erarbeitung des Codes	17.11.2024	Unser Ziel ist es, den Code bis Mitte November fertigzustellen, um ausreichend Zeit für die Überarbeitung zu haben und uns anschliessend auf den Bericht konzentrieren zu können.
4	Feinschliff Code	24.11.2024	Während dieser Woche werden wir den Code final überarbeiten und so weit wie möglich optimieren.
5	Erarbeitung Bericht	08.12.2024	Während diesen 2 Wochen werden wir den Bericht erarbeiten.
6	Finale Überarbeitung und Abgabe	15.12.2024	In der letzten Woche werden wir alle Arbeiten abschliessend überarbeiten und finalisieren, damit

			das Projekt erfolgreich abgeschlossen und termingerecht eingereicht werden kann.
--	--	--	--

Tabelle 1: Projektplanung

5 Datenbeschreibung

Die verwendeten Daten für die Analyse und Implementierung der Handelsstrategie stammen von der Alpha Vantage API, einer Plattform für Finanzdaten. Die API bietet eine breite Palette von Finanzdaten, einschliesslich Zeitreihen für Aktien, Währungen, Rohstoffe und Kryptowährungen. Im spezifischen Kontext dieser Arbeit wurden die täglichen Wechselkurse des Währungspaares EUR/USD abgerufen.

5.1 Datenquelle

Die Alpha Vantage API liefert Wechselkursdaten über ihre Funktion `FX_DAILY`. Diese Funktion stellt eine Zeitreihe bereit, die tägliche Informationen über den Devisenhandel enthält. Die API liefert strukturierte und standardisierte Daten, die für verschiedene Finanzanalysen geeignet sind. Der Abruf der Daten erfolgt über einen HTTP-Anfrage.

5.2 Datenformat und Inhalt

Die API-Antwort wird im JSON-Format bereitgestellt und enthält zwei Hauptabschnitte:

- Metadaten: Informationen über die Anfrage, wie Basis- und Zielwährung, Aktualisierungszeitpunkt und Datenquelle
- Zeitreihen-Daten (Time Series FX (Daily))
- Tägliche Wechselkursdaten mit folgenden Attributen:
 1. open: Eröffnungskurs des Tages
 2. high: Höchstkurs des Tages
 3. low: Tiefstkurs des Tages
 4. close: Schlusskurs des Tages

5.3 Verarbeitung der Daten

Für die Analyse wurden die Zeitreihen-Daten (Time Series FX (Daily)) extrahiert und in eine tabellarische Form umgewandelt. Jede Zeile repräsentiert die Wechselkursdaten eines Handelstags, während die Spalten die Kursarten (open, high, low, close) darstellen. Das Datum dient als Index, um eine zeitliche Zuordnung zu ermöglichen. In der folgenden Tabelle ist ein Beispiel einer solchen Tabelle dargestellt:

Datum	Open	High	Low	Close
2024-12-01	1.0976	1.1020	1.0955	1.0998
2024-11-30	1.1002	1.1035	1.0987	1.1001

Tabelle 2: Beispiel Verarbeitung der Daten

5.4 Bedeutung der Daten

Die bereitgestellten Wechselkursdaten sind für die Analyse der Preisbewegungen zwischen zwei Währungen unerlässlich. Sie bieten die Grundlage für:

- Identifikation von Trends (z. B. anhand von gleitenden Durchschnitten).
- Bestimmung von Unterstützungs- und Widerstandsniveaus.
- Entwicklung und Bewertung von Handelsstrategien.

Die Qualität und Verfügbarkeit der Daten über eine zuverlässige Quelle wie Alpha Vantage gewährleisten eine fundierte und reproduzierbare Analyse.

5.5 Visualisierung der Daten und Ergebnisse

Im nachfolgenden Kapitel werden die Ergebnisse der Datenauswertungen dargestellt und erklärt. Insgesamt hat der Code sechs Plots erstellt, welche verschiedene Aspekte der Analyse und Handelsstrategie für die Wechselkurse von EUR/USD repräsentieren. Jeder Plot zeigt zusätzliche Informationen oder Verfeinerungen, die zusammen ein umfassendes Bild der Strategie ergeben.

5.5.1 Plot 1 – Schlusskurse von EUR/USD

Der erste Plot zeigt die Entwicklung der Schlusskurse (Close) des Währungspaar EUR/USD über einen bestimmten Zeitraum. In unserem Fall von Juli bis Dezember (rund 4 Monate).

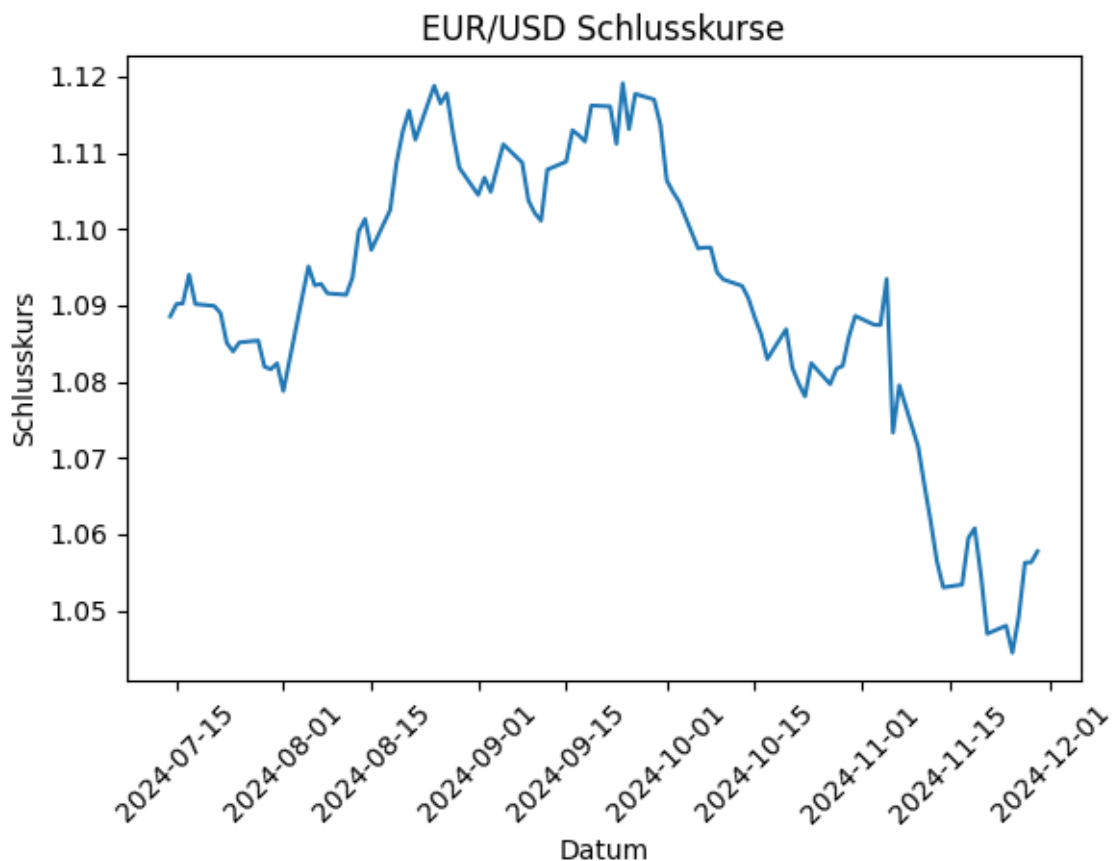


Abbildung 1: Plot 1 - Schlusskurse EUR/USD

5.5.2 Plot 1 – Bedeutung

Der Plot in Abbildung 1 dient als Ausgangspunkt der Analyse. Er gibt einen Überblick über die Preisbewegungen, wie zum Beispiel die Aufwärts- und Abwärtstrends sowie die Seitwärtsbewegungen (konsolidierende Märkte). Solche Informationen sind wichtig, um die Marktstruktur zu verstehen und mögliche Handelsmöglichkeiten zu identifizieren. Ohne jedoch unterstützende Niveaus, Indikatoren oder Handelsentscheidungen liefert dieser Plot allein jedoch wenig Handlungsempfehlungen. Auf dieser Basis wurde dann der zweite Plot erstellt.

5.5.3 Plot 2 – Schlusskurse mit Unterstützungs- und Widerstandsniveaus

Der zweite Plot erweitert die Visualisierung um zwei Schlüsselkonzepte der technischen Analyse:

- Widerstandsniveau (orange gestrichelte Linie): Höchstkurse der letzten 20 Tage (rollierendes Fenster).
- Unterstützungsniveau (grüne gestrichelte Linie): Tiefstkurse der letzten 20 Tage.

Diese Linien werden neben den Schlusskursen (blaue Linie) dargestellt

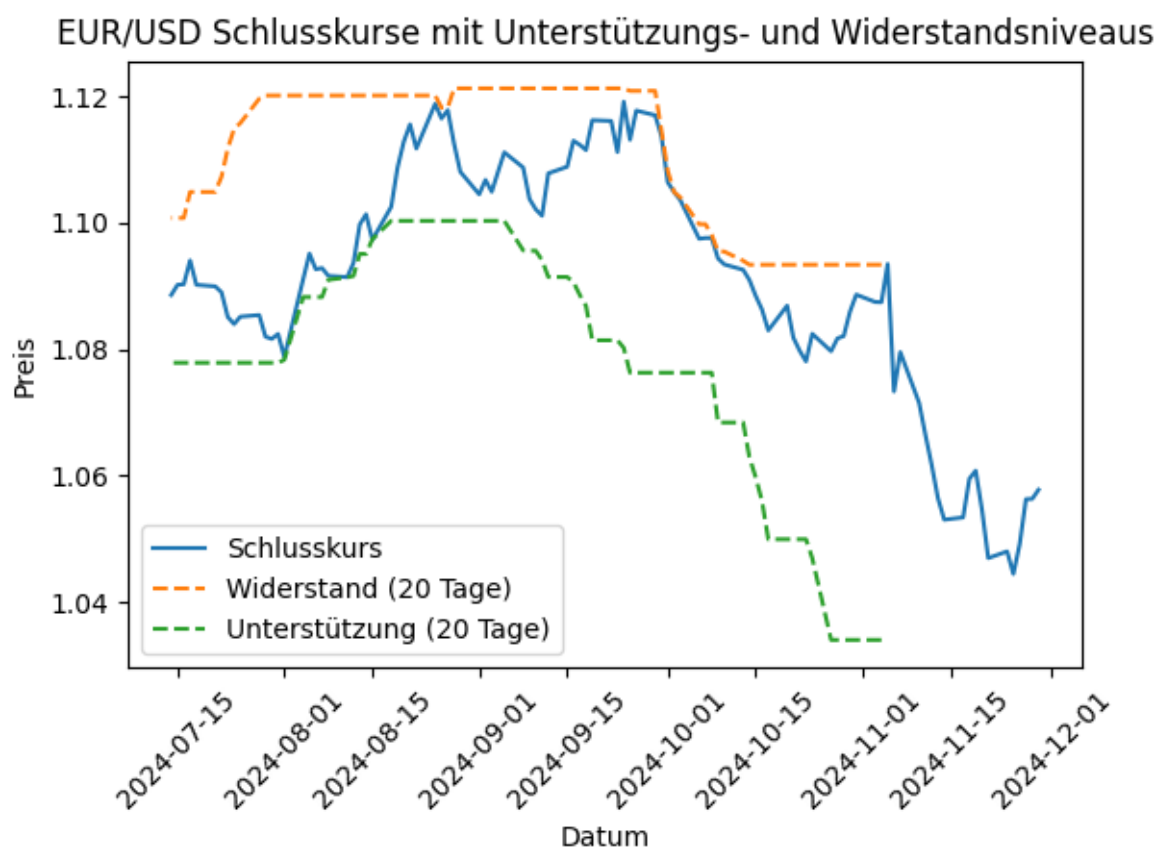


Abbildung 2: Plot 2 - Schlusskurse Unterstützungs/Widerstandsniveaus

5.5.4 Plot 2 – Bedeutung

Die Unterstützungsniveaus zeigen Preisbereiche, in denen Käufer typischerweise in den Markt eintreten, was den Kurs stabilisiert oder steigen lässt. Die Widerstandsniveaus markieren Preisbereiche, in denen Verkäufer typischerweise aktiv werden, was den Kurs drückt oder sinken lässt. Diese Niveaus helfen, Marktphasen zu

identifizieren, in denen Kauf- oder Verkaufsentscheidungen sinnvoll sein könnten. Solche Unterstützungs- und Widerstandsniveaus sind essenziell für die Entwicklung einer Handelsstrategie. Dieser Plot bereitet den Boden für die Visualisierung der tatsächlichen Handelsentscheidungen im dritten Plot (siehe Abbildung 3).

5.5.5 Plot 3 – Trades

Der dritte Plot enthält die folgenden Informationen:

- Schlusskurse (blaue Linie).
- Unterstützungs- (grüne gestrichelte Linie) und Widerstandsniveaus (rote gestrichelte Linie).
- Handelsmarkierungen:
 - Kaufpunkte (grünes Dreieck): Diese markiert den Zeitpunkt, an dem der Schlusskurs das Unterstützungsniveau erreicht oder unterschreitet. Dies signalisiert eine potenzielle Kaufgelegenheit, da der Markt auf ein Unterstützungsniveau trifft, an dem üblicherweise verstärkt Kaufdruck entsteht.

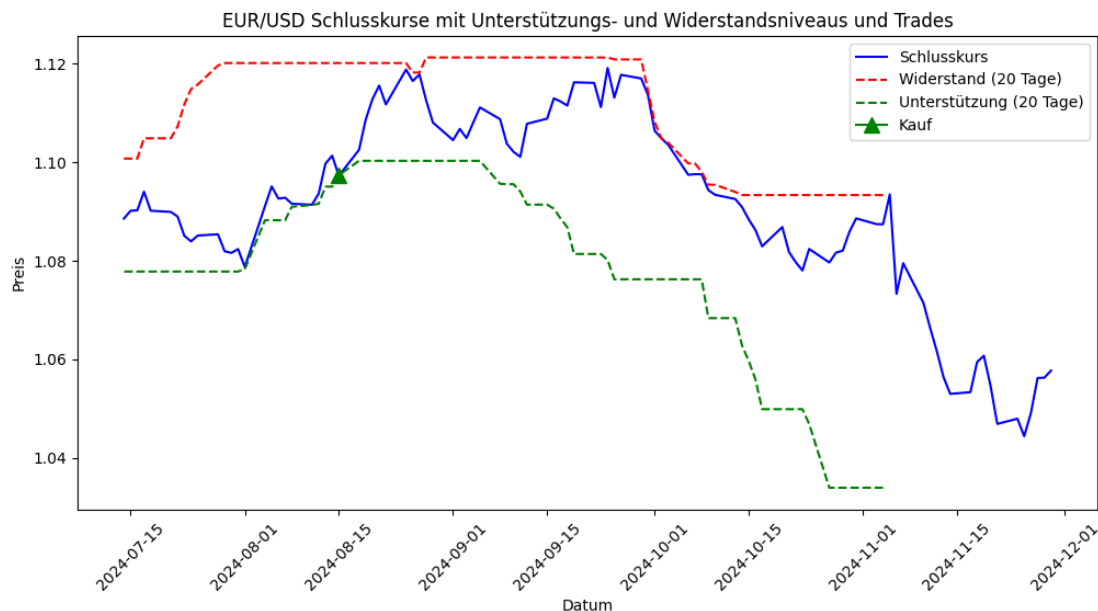


Abbildung 3: Plot 3 - Trades

5.5.6 Plot 3 – Bedeutung

Der Plot 3 veranschaulicht die praktische Umsetzung der Handelsstrategie. Im dargestellten Beispiel zeigt der Plot einen Kaufpunkt (grünes Dreieck), der entsteht, wenn der Schlusskurs das Unterstützungsniveau erreicht oder unterschreitet. Dies signalisiert eine potenzielle Kaufgelegenheit, da der Markt an diesem Punkt auf erhöhten Kaufdruck trifft. Aktuell zeigt der Plot keine Verkaufsmarkierungen, da der Kurs innerhalb des analysierten Zeitraums kein Widerstandsniveau erreicht hat. Sollte der Kurs jedoch ein Widerstandsniveau (rote gestrichelte Linie) berühren oder überschreiten, könnte dies als Signal für einen potenziellen Verkauf interpretiert werden.

Dieser Plot veranschaulicht, wie die Theorie der Unterstützungs- und Widerstandsniveaus auf praktische Handelsentscheidungen angewandt werden kann. Darüber hinaus bietet er Hinweise darauf, ob die Niveaus zuverlässig sind oder ob Anpassungen an der Strategie erforderlich sein könnten.

5.5.7 Plot 4 – Gleitende Durchschnitte und Trades

Der vierte Plot baut auf Plot 3 auf und fügt zwei weitere Indikatoren hinzu:

- 50-Tage gleitender Durchschnitt (orange Linie): Kurzfristiger Durchschnitt zur Erkennung von kurzfristigen Trends.
- Handelsmarkierungen (grüne und rote Dreiecke) und Unterstützungs- und Widerstandsniveaus bleiben erhalten.

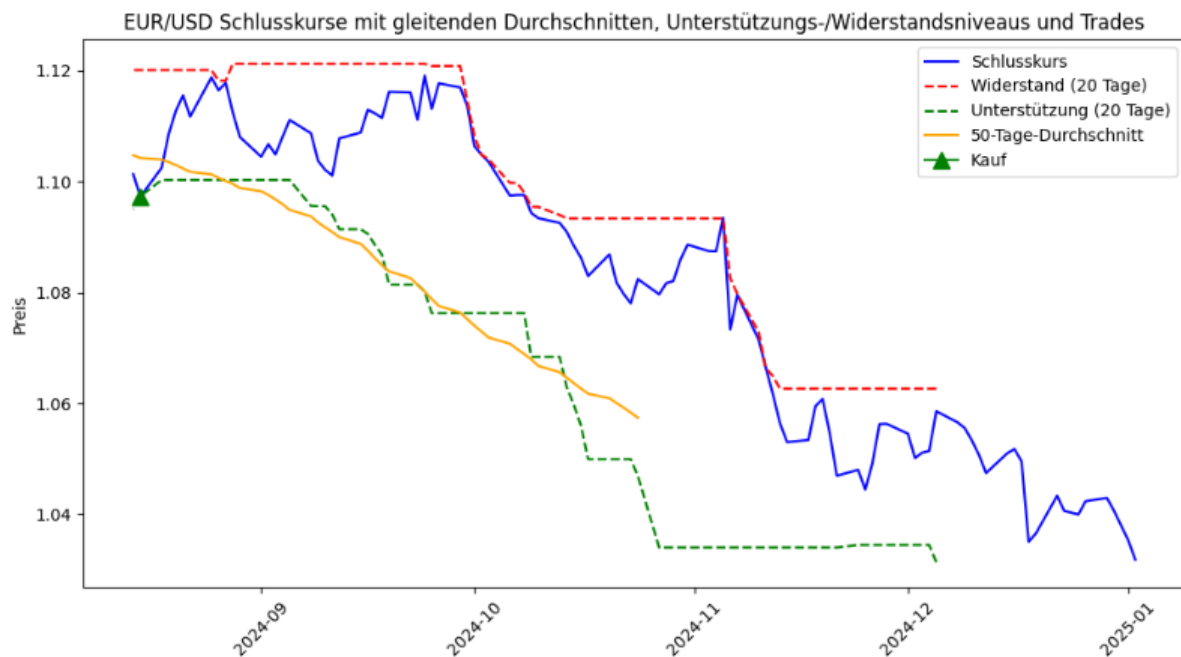


Abbildung 4: Plot 4 - Gleitende Durchschnitte & Trades

5.5.8 Plot 4 – Bedeutung

Die gleitenden Durchschnitte ergänzen die Unterstützungs- und Widerstandsniveaus:

- Wenn der Schlusskurs über den gleitenden Durchschnitt steigt, zeigt dies einen potenziellen Aufwärtstrend.
- Ein Fall unter den gleitenden Durchschnitt signalisiert einen möglichen Abwärtstrend.
- Aussagen zum 50-Tage-Durchschnitt: Der 50-Tage-Durchschnitt dient als Indikator für kurzfristige Trends und zeigt potenzielle Aufwärts- oder Abwärtstrends, basierend darauf, ob der Schlusskurs über oder unter diesem Durchschnitt liegt

Der Plot hilft insgesamt, um zu beurteilen, ob die Handelsentscheidungen (basierend auf Unterstützungs- und Widerstandsniveaus) auch mit längerfristigen Trends übereinstimmen. Somit kann die Robustheit der Strategie überprüft werden.

5.5.9 Plot 5 – Bollinger Bands

Die Bollinger Bands ergänzen die Analyse der Unterstützungs- und Widerstandsniveaus durch die Visualisierung der Marktvolatilität. Sie bestehen aus einem gleitenden Durchschnitt (Mittellinie) und zwei Bändern, die jeweils die obere und untere Standardabweichung des Kurses darstellen.

- Wenn der Schlusskurs die obere Grenze der Bollinger Bands erreicht oder überschreitet, deutet dies auf eine potenziell überkaufte Marktsituation hin, was häufig zu einer Kurskorrektur führt.
- Ein Schlusskurs unterhalb der unteren Bollinger-Band-Grenze signalisiert eine mögliche überverkaufte Marktsituation, die oft von einer Gegenbewegung nach oben begleitet wird.
- Die Mittellinie des Bollinger Bands dient als kurzfristiger Trendindikator und zeigt potenzielle Umkehrpunkte im Markt.

Dieser Plot bietet entscheidende Informationen über die Marktvolatilität und hilft, potenzielle Handelsentscheidungen mit zusätzlichen Sicherheitsmechanismen wie Stop-Loss-Strategien zu stützen. In Verbindung mit den vorherigen Analysen (z. B. Unterstützungs- und Widerstandsniveaus) liefert er eine umfassendere Grundlage für die Beurteilung von Handelsmöglichkeiten.

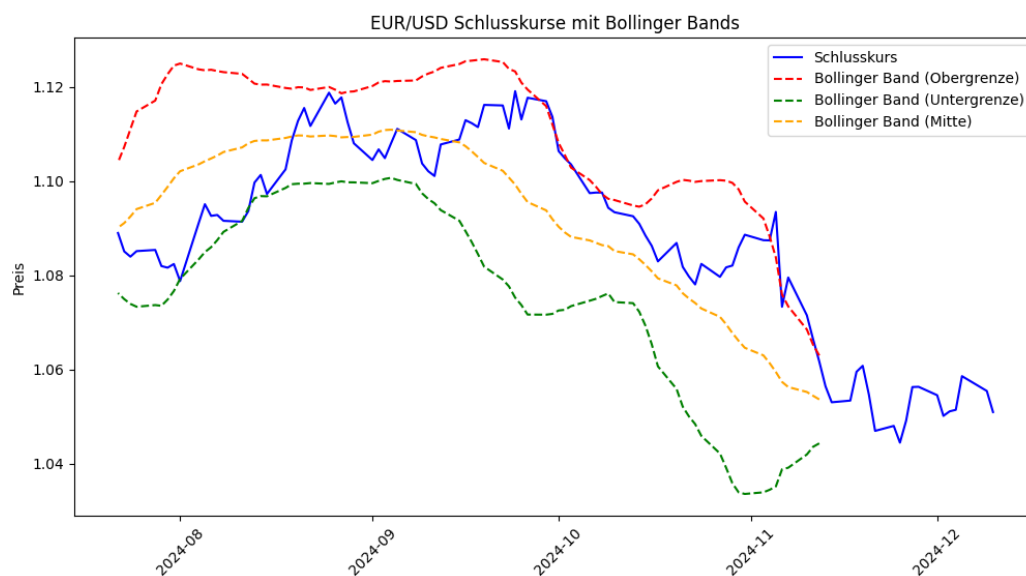


Abbildung 5: Plot 5

5.5.10 Plot 5 – Begründung

Die Bollinger Bands liefern zusätzliche Einblicke in die Dynamik der Marktbewegungen, insbesondere durch die Visualisierung der Volatilität.

- Ein Schlusskurs oberhalb der oberen Bollinger-Grenze deutet oft auf eine Phase hoher Volatilität und potenzieller Überbewertung hin, was zu einer Trendwende führen kann.
- Ein Schlusskurs unterhalb der unteren Bollinger-Grenze zeigt dagegen eine mögliche Unterbewertung, oft begleitet von einer bevorstehenden Erholung.
- Der Abstand zwischen den Bändern variiert je nach Marktvolatilität. Enge Bänder weisen auf eine Phase niedriger Volatilität hin, die oft einer starken Bewegung folgt (bekannt als „Bollinger-Squeeze“).

Dieser Plot hilft nicht nur, überkaufte oder überverkaufte Marktbedingungen zu identifizieren, sondern auch, die Risiken während volatiler Marktphasen zu steuern. Die Kombination mit den vorherigen Plots, insbesondere den Unterstützungs- und Widerstandsniveaus, ermöglicht eine präzisere Analyse von Einstiegs- und Ausstiegspunkten. Dies verbessert die Robustheit der entwickelten Handelsstrategie.

5.5.11 Plot 6 – Relative Strength Index (RSI)

Der Plot in Abbildung 6 visualisiert den Relative Strength Index (RSI) für das Währungspaar EUR/USD. Der RSI ist ein technischer Indikator, der misst, ob ein Markt überkauft oder überverkauft ist. Dabei wird der Wert zwischen 0 und 100 skaliert, mit zwei zentralen Schwellwerten:

- Ein RSI über 70 signalisiert eine überkaufte Marktsituation, in der eine Kurskorrektur wahrscheinlich ist.
- Ein RSI unter 30 deutet auf eine überverkaufte Marktsituation hin, in der eine Erholung des Kurses möglich ist.

Der RSI ergänzt die bestehende Analyse, indem er Momentum und relative Stärke des Marktes quantifiziert. Er liefert zusätzliche Hinweise für den Einstieg in oder Ausstieg

aus Positionen und ist ein entscheidendes Werkzeug für die Optimierung der Handelsstrategie.

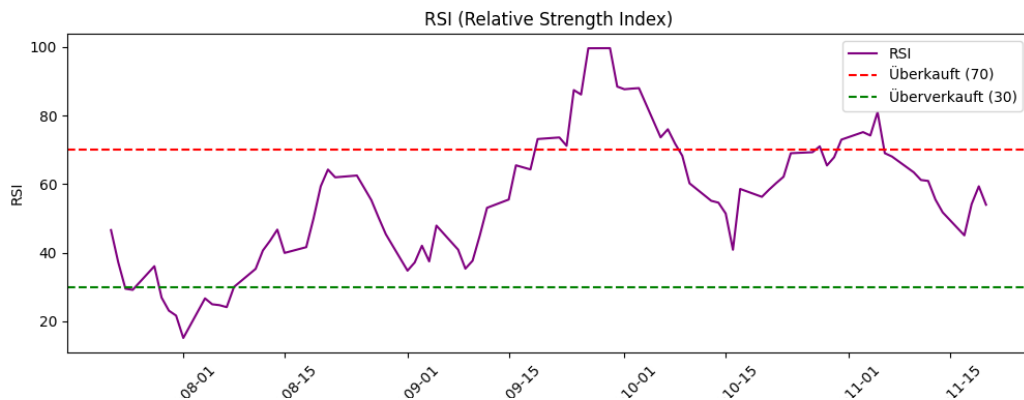


Abbildung 6: Plot 6

5.5.12 Plot 6 – Begründung

Der RSI bietet eine robuste Ergänzung zur Identifikation von Handelschancen:

- Ein RSI-Wert über 70 deutet oft auf eine bevorstehende Korrektur hin, da der Markt möglicherweise überkauft ist.
- Ein RSI-Wert unter 30 signalisiert potenzielle Kaufgelegenheiten, da der Markt überverkauft sein könnte.
- Der neutrale Bereich zwischen 30 und 70 zeigt konsolidierende Märkte, in denen Momentum-bedingte Entscheidungen nicht eindeutig sind.

Dieser Indikator erweitert die bestehende Strategie um eine weitere Dimension, die es ermöglicht, Trends und Wendepunkte präziser zu erkennen. In Kombination mit den Bollinger Bands und den horizontalen Unterstützungs- und Widerstandsniveaus kann der RSI das Timing für Handelsentscheidungen optimieren und das Risiko reduzieren.

5.6 Fazit Ergebnisse

Die sechs Plots bieten eine vollständige Visualisierungskette, die von einem ersten Überblick über die Kursbewegungen (siehe Abbildung 1) bis hin zur Integration komplexer Indikatoren reicht. Beginnend mit der Darstellung der Schlusskurse (siehe

Abbildung 1), über die Identifikation von Unterstützungs- und Widerstandsniveaus (siehe Abbildung 2), bis hin zur Analyse von Handelsentscheidungen (siehe Abbildung 3), einer langfristigen Validierung der Strategie durch gleitende Durchschnitte (siehe Abbildung 4) und der Visualisierung von Volatilität mit Bollinger Bands (siehe Abbildung 5) sowie der Einbindung des RSI zur Bestätigung von Trendwenden (siehe Abbildung 6).

Die Analyse der Handelsstrategie, insbesondere des Kaufpunkts auf Plot 3 (siehe Abbildung 3), zeigt die Effektivität der Methode. Unterstützungs- und Widerstandsniveaus bilden hierbei die Grundlage und helfen, potenzielle Wendepunkte im Markt zu identifizieren. Der Kaufpunkt wurde direkt am Unterstützungsniveau gesetzt, das durch die Tiefstkurse der letzten 20 Tage berechnet wurde. In diesem Fall führte der Kauf zu einem Kursanstieg, was die Zuverlässigkeit der Strategie unterstreicht. Dennoch zeigt die Analyse, dass die Integration weiterer Indikatoren, wie Bollinger Bands (siehe Abbildung 5) und RSI (siehe Abbildung 6), dazu beiträgt, Handelsentscheidungen zu validieren und Risiken zu minimieren.

Die langfristige Analyse mit gleitenden Durchschnitten (siehe Abbildung 4) validiert die Strategie über grössere Zeiträume hinweg und hilft, Trends auf verschiedenen Ebenen zu erkennen. Die Bollinger Bands erweitern die Analyse, indem sie Volatilität sichtbar machen, während der RSI die Stärke von Marktbewegungen quantifiziert.

Insgesamt zeigt die Visualisierungskette, dass die Methodik solide ist und in Märkten mit klaren Unterstützungs- und Widerstandszonen erfolgreich angewendet werden kann. Durch die Kombination mehrerer Indikatoren wird die Strategie robuster und erlaubt präzisere Entscheidungen.

6 Reflektion

Das vorliegende Projekt bietet im Vergleich zu den auf Moodle bereitgestellten Artikeln einen klaren Mehrwert, da es theoretische Grundlagen mit praktischen Anwendungen verbindet und innovative Ansätze verfolgt. Während die Moodle-Materialien Konzepte wie Python-Programmierung, Blockchain, Decentralized Finance (DeFi) und finanztechnische Indikatoren umfassend abdecken, zeigt das Projekt, wie diese theoretischen Inhalte genutzt werden können, um reale Herausforderungen zu lösen.

Ein zentraler Mehrwert des Projekts liegt in der praktischen Umsetzung. Während die Moodle-Materialien häufig grundlegende Informationen und statische Beispiele bieten, demonstriert das Projekt die Entwicklung einer algorithmischen Handelsstrategie anhand realer Marktdaten. Mit Hilfe der Alpha Vantage API wurden historische Kursdaten des Währungspaares EUR/USD analysiert und strategisch verarbeitet. Dieser Ansatz zeigt, wie theoretisches Wissen zu einem handlungsorientierten Ergebnis führen kann. Ein weiterer entscheidender Aspekt ist die Integration fortschrittlicher Analysewerkzeuge wie gleitender Durchschnitte, Bollinger Bands und des Relative Strength Index (RSI). Diese Indikatoren wurden nicht nur theoretisch behandelt, sondern aktiv in die Handelsstrategie eingebunden. Dies hebt das Projekt von den Moodle-Materialien ab, die diese Indikatoren meist isoliert betrachten. Das Projekt zeigt, wie sie zusammenwirken, um eine präzisere und robustere Entscheidungsgrundlage zu schaffen. Ein besonders wertvoller Beitrag ist die Validierung der Strategie durch Backtesting. Anhand historischer Daten wurde die Rentabilität der entwickelten Strategie getestet, was über die statischen Beispiele und Konzepte auf Moodle hinausgeht. Die Ergebnisse des Projekts zeigen, dass die Strategie insbesondere in Seitwärtsmärkten erfolgreich ist, während in volatilen Märkten Verbesserungen nötig sind. Diese praxisnahe Evaluierung liefert wichtige Einblicke in die Effektivität und Zuverlässigkeit der Strategie. Darüber hinaus setzt das Projekt auf zukunftsorientierte Technologien wie DeFi und Blockchain, die in den Moodle-Materialien oft nur oberflächlich behandelt werden. Das Projekt zeigt, wie dezentrale Finanzplattformen und Kryptowährungen in Handelsstrategien integriert werden können, wodurch es aktuelle Entwicklungen im Finanzsektor aufgreift und vertieft. Ein weiterer Vorteil des Projekts ist seine Flexibilität. Die entwickelten Methoden sind leicht an andere Finanzinstrumente oder Märkte anpassbar und können um Machine-Learning-Techniken oder Echtzeitanalysen erweitert werden. Diese Anpassungsfähigkeit hebt das Projekt deutlich von den oft starren Moodle-Beispielen ab und macht es zu einem wertvollen Werkzeug für weiterführende Forschung und Anwendung.

Zusätzlich wurde grosser Wert auf eine benutzerfreundliche Darstellung gelegt. Die Visualisierung der Ergebnisse – wie Plots zu Schlusskursen, Unterstützungs- und Widerstandsniveaus sowie Handelsmarkierungen – erleichtert die Interpretation

komplexer Daten und macht das Projekt sowohl für Experten als auch für Einsteiger verständlich.

Insgesamt erweitert das Projekt die auf Moodle verfügbaren Materialien, indem es praktische Anwendungen ermöglicht, fortgeschrittene Methoden integriert und durch seine Flexibilität und Benutzerfreundlichkeit überzeugt. Es liefert einen wertvollen Beitrag zur algorithmischen Analyse von Finanzmärkten und bietet eine Basis für zukünftige Entwicklungen.

6.1 Fazit zum Projekt

Das Projekt hat gezeigt, dass die Analyse von Horizontal Patterns ein wirkungsvolles Werkzeug zur Identifikation von Handelszonen ist. Die entwickelte Strategie demonstrierte durch das Backtesting eine solide Grundlage und lieferte vielversprechende Ergebnisse. Insbesondere die Visualisierung der Daten ermöglichte es, komplexe Muster einfach zu interpretieren und fundierte Handelsentscheidungen zu treffen.

6.2 Herausforderungen und Schwierigkeiten

Datenverarbeitung

Der Umgang mit den Zeitreihen-Daten der Alpha Vantage API erwies sich als anspruchsvoll, insbesondere die Datenbereinigung und die Transformation in ein geeignetes Format für die Analyse.

Code-Optimierung

Die Entwicklung eines performanten Codes zur automatischen Erkennung von Horizontal Patterns erforderte zahlreiche Iterationen und Tests.

Integration von Indikatoren:

Die Wahl geeigneter Indikatoren zur Ergänzung der Horizontal Patterns war eine Herausforderung. Insbesondere die Balance zwischen Komplexität und Aussagekraft musste sorgfältig abgewogen werden.

6.3 Mögliche zukünftige Erweiterungen

Echtzeit-Handel

Die Weiterentwicklung des Systems hin zu einer Echtzeit-Handelsanwendung könnte die Nützlichkeit deutlich steigern. Dazu wäre eine Integration von Live-Daten erforderlich.

Machine Learning

Der Einsatz von maschinellem Lernen zur Optimierung der Erkennung von Horizontal Patterns und zur Vorhersage von Kursbewegungen wäre eine vielversprechende Erweiterung.

Zusätzliche Indikatoren

Die Integration weiterer Indikatoren, wie beispielsweise Relative Strength Index (RSI) oder Bollinger Bands, könnte die Zuverlässigkeit der Handelsentscheidungen verbessern.

Cross-Asset-Analyse

Das System könnte auf andere Finanzinstrumente wie Aktien oder Rohstoffe ausgeweitet werden, um eine breitere Anwendbarkeit zu ermöglichen.

GUI-Entwicklung

Die Entwicklung einer grafischen Benutzeroberfläche würde die Bedienbarkeit für Anwender ohne technische Vorkenntnisse erheblich verbessern.

Insgesamt hat das Projekt einen wichtigen Beitrag zur praktischen Anwendung der technischen Analyse geleistet und bietet vielfältige Möglichkeiten zur Weiterentwicklung.

7 A Anhang

7.1 Abbildungsverzeichnis

Abbildung 1: Plot 1 - Schlusskurse EUR/USD	10
Abbildung 2: Plot 2 - Schlusskurse Unterstützungs/Widerstandsniveaus.....	11
Abbildung 3: Plot 3 - Trades.....	12
Abbildung 4: Plot 4 - Gleitende Durchschnitte & Trades	14
Abbildung 6: Plot 5.....	15
Abbildung 7: Plot 6.....	17

7.2 Tabellenverzeichnis

Tabelle 1: Projektplanung	8
Tabelle 2: Beispiel Verarbeitung der Daten	9

7.3 Code:

```
import requests
API_KEY = "DIFA"
url =
f"https://www.alphavantage.co/query?function=FX_DAILY&from_symbol=EUR&to_symbol=USD&ap
ikey={ API_KEY }"
response = requests.get(url)
data = response.json()

print(data) # Schau dir die Struktur der Daten an, um zu sehen, wie du sie weiterverarbeiten kannst.

import pandas as pd

# Extrahiere die Zeitreihen-Daten aus der API-Antwort
if "Time Series FX (Daily)" in data:
    time_series = data["Time Series FX (Daily)"]

# Wandle die Zeitreihen-Daten in einen DataFrame um
df = pd.DataFrame.from_dict(time_series, orient="index")
```

```
df = df.astype(float) # Konvertiere die Daten in den Typ float
df.index = pd.to_datetime(df.index) # Konvertiere das Index in Datumsformat

# Benenne die Spalten um für eine bessere Lesbarkeit
df.columns = ["Open", "High", "Low", "Close"]
print(df.head()) # Zeige die ersten Zeilen des DataFrames

else:
    print("Keine Zeitreihen-Daten in der API-Antwort gefunden.")

import matplotlib.pyplot as plt

plt.plot(df.index, df["Close"])
plt.title("EUR/USD Schlusskurse")
plt.xlabel("Datum")
plt.ylabel("Schlusskurs")
plt.xticks(rotation=45)
plt.show()

# Berechne den rollierenden Höchst- und Tiefstwert (z.B. 20-Tage-Fenster)
df["High_20"] = df["High"].rolling(window=20).max()
df["Low_20"] = df["Low"].rolling(window=20).min()

# Plot der Schlusskurse und der Unterstützungs- und Widerstandsniveaus
plt.plot(df.index, df["Close"], label="Schlusskurs")
plt.plot(df.index, df["High_20"], label="Widerstand (20 Tage)", linestyle="--")
plt.plot(df.index, df["Low_20"], label="Unterstützung (20 Tage)", linestyle="--")
plt.title("EUR/USD Schlusskurse mit Unterstützungs- und Widerstandsniveaus")
plt.xlabel("Datum")
plt.ylabel("Preis")
plt.xticks(rotation=45)
plt.legend()
plt.show()

# Simulierter Handelsansatz
initial_balance = 10000 # Startkapital in EUR
balance = initial_balance
position = 0 # Anzahl der Einheiten im Markt

for i in range(20, len(df)):
    # Kauf bei Unterstützungsniveau
    if df["Close"].iloc[i] <= df["Low_20"].iloc[i] and position == 0:
        position = balance / df["Close"].iloc[i]
        balance = 0
        print(f"Kauf am {df.index[i].date()}: Preis = {df['Close'].iloc[i]:.2f}, Position = {position:.2f} Einheiten")

    # Verkauf bei Widerstandsniveau
    elif df["Close"].iloc[i] >= df["High_20"].iloc[i] and position > 0:
        balance = position * df["Close"].iloc[i]
        print(f"Verkauf am {df.index[i].date()}: Preis = {df['Close'].iloc[i]:.2f}, Balance = {balance:.2f} EUR")
        position = 0
```

```
# Endwert berechnen
if position > 0:
    balance = position * df["Close"].iloc[-1]
print(f"Endwert der Strategie: {balance:.2f} EUR (Startkapital: {initial_balance} EUR)")

# Ergebnisanalyse der Handelsstrategie
initial_balance = 10000 # Startkapital in EUR
balance = initial_balance
position = 0 # Anzahl der Einheiten im Markt
trades = [] # Liste zur Speicherung der Trades (Käufe und Verkäufe)

for i in range(20, len(df)):
    # Kauf bei Unterstützungsniveau
    if df["Close"].iloc[i] <= df["Low_20"].iloc[i] and position == 0:
        position = balance / df["Close"].iloc[i]
        balance = 0
        trades.append(("Kauf", df.index[i], df["Close"].iloc[i]))

    # Verkauf bei Widerstandsniveau
    elif df["Close"].iloc[i] >= df["High_20"].iloc[i] and position > 0:
        balance = position * df["Close"].iloc[i]
        trades.append(("Verkauf", df.index[i], df["Close"].iloc[i]))
        position = 0

# Endwert berechnen
if position > 0:
    balance = position * df["Close"].iloc[-1]
final_value = balance
total_profit = final_value - initial_balance

# Ergebnisse anzeigen
print(f"Startkapital: {initial_balance:.2f} EUR")
print(f"Endwert der Strategie: {final_value:.2f} EUR")
print(f"Gesamter Gewinn/Verlust: {total_profit:.2f} EUR")
print(f"Anzahl der Trades: {len(trades)//2} (Käufe und Verkäufe)")

# Details der Trades anzeigen
for trade in trades:
    print(f"{trade[0]} am {trade[1].date()}: Preis = {trade[2]:.2f}")

# Plot der Schlusskurse und der Unterstützungs- und Widerstandsniveaus mit Markierungen für Käufe
und Verkäufe
plt.figure(figsize=(12, 6))
plt.plot(df.index, df["Close"], label="Schlusskurs", color="blue")
plt.plot(df.index, df["High_20"], label="Widerstand (20 Tage)", linestyle="--", color="red")
plt.plot(df.index, df["Low_20"], label="Unterstützung (20 Tage)", linestyle="--", color="green")

# Kauf- und Verkaufspunkte markieren
for trade in trades:
    if trade[0] == "Kauf":
        plt.plot(trade[1], trade[2], marker="^", color="green", markersize=10, label="Kauf")
    elif trade[0] == "Verkauf":
```



```
plt.plot(trade[1], trade[2], marker="v", color="red", markersize=10, label="Verkauf")

plt.title("EUR/USD Schlusskurse mit Unterstützungs- und Widerstandsniveaus und Trades")
plt.xlabel("Datum")
plt.ylabel("Preis")
plt.xticks(rotation=45)
plt.legend(loc="best")
plt.show()

# Performance-Analyse: Vergleich zur Buy-and-Hold-Strategie
buy_and_hold_profit = df["Close"].iloc[-1] - df["Close"].iloc[20]
buy_and_hold_return = (buy_and_hold_profit / df["Close"].iloc[20]) * 100

strategy_return = (total_profit / initial_balance) * 100

print(f"Buy-and-Hold-Rendite: {buy_and_hold_return:.2f}%")
print(f"Strategie-Rendite: {strategy_return:.2f}%")

# Berechnung des gleitenden Durchschnitts (50 und 200 Tage)
df["MA50"] = df["Close"].rolling(window=50).mean()
df["MA200"] = df["Close"].rolling(window=200).mean()

# Aktualisierte Visualisierung
# Aktualisierte Visualisierung
plt.figure(figsize=(12, 6))
plt.plot(df.index, df["Close"], label="Schlusskurs", color="blue")
plt.plot(df.index, df["High_20"], label="Widerstand (20 Tage)", linestyle="--", color="red")
plt.plot(df.index, df["Low_20"], label="Unterstützung (20 Tage)", linestyle="--", color="green")
plt.plot(df.index, df["MA50"], label="50-Tage-Durchschnitt", color="orange")
plt.plot(df.index, df["MA200"], label="_nolegend_", color="purple") # Legende entfernt

# Kauf- und Verkaufspunkte markieren
for trade in trades:
    if trade[0] == "Kauf":
        plt.plot(trade[1], trade[2], marker="^", color="green", markersize=10, label="Kauf")
    elif trade[0] == "Verkauf":
        plt.plot(trade[1], trade[2], marker="v", color="red", markersize=10, label="Verkauf")

plt.title("EUR/USD Schlusskurse mit gleitenden Durchschnitten, Unterstützungs-/Widerstandsniveaus
und Trades")
plt.xlabel("Datum")
plt.ylabel("Preis")
plt.xticks(rotation=45)
plt.legend(loc="best")
plt.show()

# Funktion zur Berechnung von Unterstützungs- und Widerstandsniveaus mit variablem Fenster
def calculate_levels(df, window=20):
    df["High_level"] = df["High"].rolling(window=window).max()
    df["Low_level"] = df["Low"].rolling(window=window).min()
    return df
```

```
# Beispiel: Berechnung mit einem 30-Tage-Fenster
df = calculate_levels(df, window=30)

# Drawdown-Berechnung
df["Balance"] = balance if position == 0 else position * df["Close"]
df["Peak"] = df["Balance"].cummax()
df["Drawdown"] = (df["Balance"] - df["Peak"]) / df["Peak"]

max_drawdown = df["Drawdown"].min() * 100
print(f"Maximaler Drawdown der Strategie: {max_drawdown:.2f}%")

# Berechnung des RSI (Relative Strength Index)
def calculate_rsi(data, window=14):
    delta = data['Close'].diff()
    gain = (delta.where(delta > 0, 0)).rolling(window=window).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=window).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
    return rsi

df['RSI'] = calculate_rsi(df)

# Berechnung der Bollinger Bands
def calculate_bollinger_bands(data, window=20):
    sma = data['Close'].rolling(window=window).mean()
    std = data['Close'].rolling(window=window).std()
    upper_band = sma + (std * 2)
    lower_band = sma - (std * 2)
    return sma, upper_band, lower_band

df['BB_Middle'], df['BB_Upper'], df['BB_Lower'] = calculate_bollinger_bands(df)

# Visualisierung von RSI und Bollinger Bands
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Close'], label='Schlusskurs', color='blue')
plt.plot(df.index, df['BB_Upper'], label='Bollinger Band (Obergrenze)', linestyle='--', color='red')
plt.plot(df.index, df['BB_Lower'], label='Bollinger Band (Untergrenze)', linestyle='--', color='green')
plt.plot(df.index, df['BB_Middle'], label='Bollinger Band (Mitte)', linestyle='--', color='orange')
plt.title("EUR/USD Schlusskurse mit Bollinger Bands")
plt.xlabel("Datum")
plt.ylabel("Preis")
plt.xticks(rotation=45)
plt.legend(loc='best')
plt.show()

plt.figure(figsize=(12, 4))
plt.plot(df.index, df['RSI'], label='RSI', color='purple')
plt.axhline(70, linestyle='--', color='red', label='Überkauft (70)')
plt.axhline(30, linestyle='--', color='green', label='Überverkauft (30)')
plt.title("RSI (Relative Strength Index)")
plt.xlabel("Datum")
plt.ylabel("RSI")
plt.xticks(rotation=45)
```

```
plt.legend(loc='best')
plt.show()
```

```
# Simulierter Handelsansatz mit Stop-Loss und Take-Profit
```

```
stop_loss_percent = 0.02 # 2% Stop-Loss
```

```
take_profit_percent = 0.05 # 5% Take-Profit
```

```
position_entry_price = 0
```

```
for i in range(20, len(df)):
```

```
    # Kauf bei Unterstützungsniveau
```

```
    if df["Close"].iloc[i] <= df["Low_20"].iloc[i] and position == 0:
```

```
        position = balance / df["Close"].iloc[i]
```

```
        position_entry_price = df["Close"].iloc[i]
```

```
        balance = 0
```

```
        print(f"Kauf am {df.index[i].date()}: Preis = {df['Close'].iloc[i]:.2f}")
```

```
# Verkauf bei Widerstandsniveau oder Take-Profit/Stop-Loss
```

```
elif position > 0:
```

```
    current_price = df["Close"].iloc[i]
```

```
    take_profit_price = position_entry_price * (1 + take_profit_percent)
```

```
    stop_loss_price = position_entry_price * (1 - stop_loss_percent)
```

```
    if current_price >= take_profit_price or current_price <= stop_loss_price:
```

```
        balance = position * current_price
```

```
        position = 0
```

```
        print(f"Verkauf am {df.index[i].date()}: Preis = {current_price:.2f}, Balance = {balance:.2f} EUR")
```