

Duale Hochschule Baden-Württemberg Mannheim

Fallstudie

StockTimes - Risikoanalyse von Aktien

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Verfasser(in):	Julia Pfützer, Annika Dackermann, Max Bernauer Julius Koenning, Zabiullah Salehi, Philipp Dingfelder
Matrikelnummer:	7411297, 5562028, 5763624, 8687786, 7305370, 7891497
Kurs:	WWI20DSB
Dozent:	Benjamin Jung
Bearbeitungszeitraum:	16.11.2021 - 28.02.2021

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
Abkürzungsverzeichnis	iii
1 Abstract	iv
2 Einleitung	1
2.1 Hintergrund	1
2.2 Aufgabe und Zielsetzung	2
3 Grundlegende Vorgehensweise	3
3.1 Vorgehensweise	3
3.2 Wirtschaftliche Analysen	6
3.2.1 Bestandsanalyse	6
3.2.2 Zielgruppenanalyse	7
3.2.3 Kosten-Nutzen-Analyse	8
3.2.4 Auswirkungen auf Marketing / Branding	9
3.2.5 SWOT-Analyse	9
3.3 Systementwurf	11
4 Umsetzung	13
4.1 Frontend	13
4.1.1 Technische Möglichkeiten	13
4.1.2 Produktpräsentation	14
4.2 Sentiment Analyse	15
4.2.1 Grundlagen Sentiment Analyse	15
4.2.2 Sentiment Analyse bei StockTimes	16
4.2.3 Reflektion Sentiment Analyse	18
4.3 Risikoklassifizierung	19
4.3.1 Erster Ansatz	19
4.3.2 Zweiter Ansatz	24
4.3.3 Resümee des zweiten Ansatzes	30

5	Fazit	31
6	Anhang	32
6.1	Entscheidungstabelle	32
6.2	Use-Case-Schablone	32

Abbildungsverzeichnis

3.1	Meilensteine unseres Projekts	4
3.2	Risikoanalyse unseres Projekts	5
3.3	Kosten- Nutzen-Analyse	8
3.4	SWOT-Analyse	10
3.5	Ereignis-Reaktions-Modell	11
3.6	Aktivitätsdiagramm	12
4.1	Frontend der App	15
4.2	Prozess der Sentiment Analyse	17
4.3	Vorbereitete Daten	21
4.4	Confusion Matrix für SVM Klassifizier	22
4.5	Korrelationsheatmap	23
4.6	Trainingsdaten	24
4.7	Betas der vergangenen Jahre von Apple	25
4.8	Train-Test-Prediction quarterly forecast	28
4.9	Test vs. Regression	28
4.10	Validierung LSTM mit unterschiedlichen Vorhersagezeiträumen	29
6.1	SWOT-Analyse	32
6.2	Use-Case-Schablone	32

Abkürzungsverzeichnis

SVM	Support Vector Machine
ABT	Abbott Laboratories
RSME	Root-Mean-Square-Error
KI	Künstliche Intelligenz
ML	Machine Learning
LSTM	Long Short-Term Memory

1 Abstract

Im heutigen politischen und wirtschaftlichen Leben sind große Aktiengesellschaften entscheidende Faktoren, die das öffentliche Geschehen maßgeblich beeinflussen. Als aufgeteiltes Kapital der Aktiengesellschaften, bilden Aktien eine wichtige Schnittstelle zwischen der Unternehmensfinanzierung (Kapitalnachfrage) und der Nutzung des Geldvermögens durch Investoren (Kapitalangebot).

Während in der Vergangenheit ein Großteil der Privatanleger in Deutschland ihr Geld in sichere Geldanlagen wie Bausparverträge, Bundeswertpapiere oder Sparkonten investierte, legen die Neuanleger zunehmend ihr Geld in spekulativere Anlageformen wie beispielsweise Aktien und Aktienfonds an.

Vor allem interessiert sich die Junge Wirtschaftsinteressente zunehmend für diese spekulativen Anlageformen. Eine Geldanlage in Aktien kann zwar große Rendite bringen, es ist jedoch auch zu beachten, dass große Risiken mit einer solchen Investition verbunden sind. Während manche Anleger bereit sind, viel Risiko einzugehen, wollen andere lieber weniger Risiko und sind dafür auch bereit, weniger Rendite zu erzielen.

Vor diesem Hintergrund wird in dieser Arbeit eine Applikation entwickelt, mithilfe dieser jeder Nutzer durch seine individuelle Risikoeinstellung und der aktuellen Marktstimmung entsprechende Aktien vorgeschlagen bekommt. Um diese Anwendung umzusetzen, werden nicht nur Nachrichten-Seiten, sondern auch verschiedene Kennzahlen und historische Daten über die einzelnen Aktien erhoben und mithilfe von Machine Learning-Methoden analysiert. Dadurch soll nicht nur das aktuelle Marktgeschehen sondern auch die individuelle Risikovorliebe der Nutzer für Aktienvorschläge berücksichtigt werden.

2 Einleitung

2.1 Hintergrund

Aufgrund mangelnder Absicherung durch staatliche Sicherungssysteme ist es heute wichtiger denn je, fundierte Anlageentscheidungen zu treffen. Durch die niedrigen Zinsen und immer mehr Online-Anbieter von Anlagemöglichkeiten gab es in den letzten Jahren einige Veränderungen bei der Kapitalanlage von Privatpersonen.

Heutzutage haben die Privatanleger eine nahezu unüberschaubare Menge an verschiedenen Investitionsmöglichkeiten. Während in der Vergangenheit ein Großteil der Privatanleger in Deutschland ihr Geld in sichere Geldanlagen wie Bausparverträge, Bundeswertpapiere oder Sparkonten investierte, legen die Neuanleger zunehmend ihr Geld in spekulative Anlageformen wie beispielsweise Aktien und Aktienfonds an.

Insbesondere wird diese Anlagemöglichkeit von den jüngeren Anlegern wahrgenommen. So zeigt eine Studie vom Forschungsinstitut DIW Econ zum Nutzerverhalten auf der Neobroker-Plattform Trade Republic, dass 70% aller Kunden von Trade Republic jünger als 35 Jahre alt sind. Davon sind wiederum rund die Hälfte zwischen 18 und 26 Jahren alt (Vgl. Antonia Mannweiler, n. d.).

Auch zu betrachten ist, dass sich immer mehr Menschen aus Haushalten mit geringerem Einkommen am Kapitalmarkt beteiligen. So hat die Studie von DIW Econ ergeben, dass etwa 30% der Neuanleger der unteren Hälfte der Einkommensverteilung angehören.

Eine Investition in Aktien ist mit Risiken verbunden und es ist möglich, dass Anleger durch ihre Investition viel Geld verlieren. Das Risiko ist daher ein wesentliches Merkmal vieler Investitionsentscheidungen. Dabei unterscheidet sich die Risikoeinstellung der Anleger. Während manche Anleger bereit sind, viel Risiko einzugehen, wollen andere lieber weniger Risiko und sind dafür auch bereit, weniger Rendite zu erzielen.

2.2 Aufgabe und Zielsetzung

Diese Fallstudie beschäftigt sich mit einer Möglichkeit Anlegern zu helfen, eine Investitionsentscheidung zu treffen.

Unsere Aufgabe war es, einen Weg zu finden, Kunden bestmöglich zu beraten. Dabei sollte zum einen die persönliche Risikofreudigkeit und zum anderen die aktuelle weltweite Entwicklung und die damit verbundene Marktstimmung berücksichtigt werden. Zusätzlich sollten diese beiden Faktoren dann noch von einem Machine Learning-Modell unterstützt werden um eine optimale Entscheidung zu ermöglichen. Die Marktstimmung soll auf Basis verschiedener Meinungen ermittelt werden. Für das Machine Learning-Modell sollen historische Daten sowie verschiedene Kennzahlen genutzt werden. Insgesamt war es also das Ziel dieser Fallstudie, Kunden Aktien auf Basis ihrer individuell einstellbaren Risikoneigung und der aktuellen Marktstimmung vorzuschlagen. Wie wir dabei vorgegangen sind wird im Laufe dieser Dokumentation erklärt.

3 Grundlegende Vorgehensweise

Im Kapitel der grundlegenden Vorgehensweise wird inhaltlich die Herangehensweise an das Projekt erklärt. Dazu wird eine wirtschaftliche Analyse durchgeführt. Diese besteht aus einer Bestandsanalyse, Zielgruppenanalyse, Kosten-Nutzen-Analyse, Auswirkungen auf Marketing/Branding und SWOT-Analyse. Zusätzlich wird ein Systementwurf beschrieben.

3.1 Vorgehensweise

Das Projektvorgehen war strukturell stark durch die Arbeitsweise der SCRUM-Methode geprägt. Es gab wöchentliche Updates in denen die verschiedenen Unterteams sich gegenseitig auf den neuesten Wissensstand brachten und neue Aufgaben definierten. Dabei gab es drei Unterteams: Das Frontend (Annika, Julius), das Backend (Julia, Philip, Zabiullah), und die Sentiment-Analyse (Max). Die verschiedenen Teams verteilten die Aufgaben unter sich und sorgten dafür, dass sie bis zum nächsten Update-Meeting die nötigen Schritte erledigt haben. Ein zentraler Orientierungspunkt bot hierbei unser Backlog in MS Teams. Dort konnten wir unsere Aufgaben definieren und den momentanen Status aufzeigen. Dies sorgte für eine Transparenz innerhalb des Teams und für eine klare Aufgabenverteilung.

Außerdem hat jede Person im Team eine Rolle zugewiesen bekommen, die diese Person sekundär ausführt. Dabei gab es den Product Owner (Annika), den Scrum-Master (Julius) und die Entwickler (Alle). Somit war immer klar, wer für welchen Bereich an Aufgaben zuständig ist. Bei der Planung des Projekts haben wir vorausgehend zu der konkreten Arbeit an den verschiedenen Bereichen eine Grafik zu unseren Meilensteinen aufgestellt.

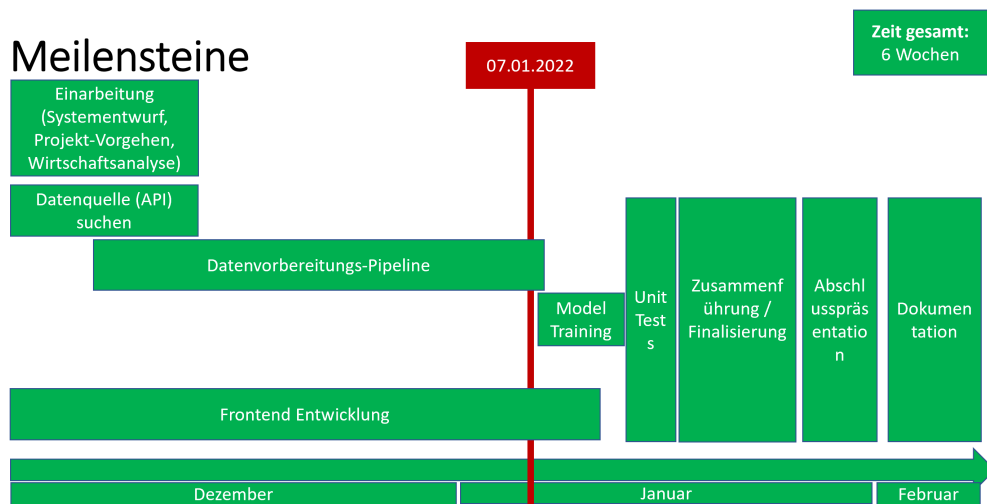


Abbildung 3.1: Meilensteine unseres Projekts
Quelle: Eigene Abbildung

Anfang Dezember planen wir unser Projekt einerseits mit der Einarbeitung zu beginnen. Diese Bestand aus Recherche und Präsentation eines Systementwurfs, des Projekt-Vorgehens und der Wirtschaftsanalyse. Andererseits planen wir die Datenquelle (API) zu suchen. Bis Anfang Januar wollten wir dann die Datenvorbereitungs-Pipeline und die Frontend-Entwicklung abgeschlossen haben um mit dem Model-Training zu beginnen. Die Planung sah vor darauf die Unit-Tests einzubauen und schließlich das gesamte Projekt zusammenzuführen und zu finalisieren. Darauf die Abschlusspräsentation und die Bearbeitung der Dokumentation bis Mitte Februar. Zusätzlich zu den Meilensteinen analysierten wir die wichtigsten Risikopunkte unseres Projekts.

Risikoanalyse

Art des Risikos	Lösungsansatz	Bewertung
Nicht-Einhaltung von Deadlines	<ul style="list-style-type: none"> • ordentliche Zeitplanung • Notfallplan erarbeiten • Reduktion des Projekts auf essentielle Bestandteile 	Hohes Risiko
Falscher Entwicklungsansatz	<ul style="list-style-type: none"> • Frühe Erkennung des Problems • Soll-Ist-Vergleich der Entwicklung 	Hohes Risiko
Fehlende Kommunikation	<ul style="list-style-type: none"> • Zentrale Kommunikation auf Teams • Regelmäßige Updates • Transparenz • Probleme direkt ansprechen 	Hohes Risiko
Abhängigkeiten der Aufgaben	<ul style="list-style-type: none"> • Vermeidung durch systematische Aufgabenplanung 	Mittleres Risiko
Technische Schwierigkeiten	<ul style="list-style-type: none"> • Fortschritt dezentral in Cloud speichern 	Geringes Risiko

Abbildung 3.2: Risikoanalyse unseres Projekts
Quelle: Eigene Abbildung

Das erste Risiko ist die Nicht-Einhaltung von Deadlines. Dieses ist bewertet mit einem hohen Risiko. Als Lösungsansätze kann man beispielsweise eine ordentliche Zeitplanung und einen Notfallplan erarbeiten. Der zweite Risikopunkt ist ein falscher Entwicklungsansatz, dieser wurde auch mit einem hohen Risiko bewertet. Lösungsansätze hierbei sind eine frühe Erkennung des Problems und ein Soll-Ist-Vergleich der Entwicklung. Die fehlende Kommunikation ist der dritte Risikopunkt, auch bewertet mit einem hohen Risiko. Als Lösung haben wir eine zentrale Kommunikation auf MS Teams und regelmäßige Updates. Das sorgt für eine allgemeine Transparenz im Team. Außerdem sollte man Probleme direkt ansprechen und sie nicht für sich behalten. Die Abhängigkeiten der Aufgaben sind ein weiterer Risikopunkt. Mit einem mittleren Risiko kann man diesen Punkt durch eine systematische Aufgabenplanung vermeiden. Der letzte Risikopunkt sind technische Schwierigkeiten. Diese haben ein vergleichsweise geringes Risiko. Als Lösung kann man den Fortschritt dezentral in der Cloud speichern.

3.2 Wirtschaftliche Analysen

Um die Wirtschaftlichkeit von StockTimes zu prüfen wurden verschiedene Analysen durchgeführt. Es wurden die Auswirkung auf Marketing und Branding, die Stärken und Schwächen des Unternehmens, die Konkurrenz, die Zielgruppe untersucht und eine Kosten-Nutzen-Analyse erstellt.

3.2.1 Bestandsanalyse

In diesem Abschnitt werden die Ergebnisse der Bestandsanalyse vorgestellt, indem auf die Produkte und Erfahrungen eingegangen wird, die bereits am Markt von der Konkurrenz angeboten werden. Ebenso werden aber auch die Merkmale vorgestellt, die unsere App im Vergleich zur Konkurrenz besonders macht. Bei unserer Konkurrenz handelt es sich um größere Unternehmen, die bereits viel Erfahrung am Finanzmarkt haben. Ein Beispiel hierfür sind Banken, die ihren Kunden empfehlen ein geringes Risiko beim Aktienwerb einzugehen. Des Weiteren verwenden Banken, Maschinelles Lernen um Prozesse zu beschleunigen und Kosten zu senken. Sie verwenden es aber auch um Betrugsversuche bei Kreditkartentransaktionen zu erkennen. Konkurrenzprodukte unserer entwickelten App sind beispielsweise der Robo-Advisor und Stimmungsanalysen. Der Nutzer erhält beim Robo-Advisor ein individuell optimiertes Portfoliomanagement, auf Basis von intelligenten Algorithmen bereitgestellt. Das zweite Konkurrenzprodukt ist die Stimmungsanalyse, die die Meinungen verschiedener Analysten zusammenfasst und einen Ausblick auf zukünftige Marktentwicklungen gibt. Wir bieten unseren Kunden Innovative Ideen und Technologisches Verständnis, besonders im Bereich Künstlicher Intelligenz, an. Des Weiteren haben wir viele Produktideen, die darauf warten umgesetzt zu werden, wie zum Beispiel unsere StockTimes App, die dem Kunden Aktienvorschläge durch einen Algorithmus des Maschinellen Lernens gibt. Doch was macht uns nun im Vergleich zu Konkurrenz-Produkten besonders oder was hebt uns von diesen ab. Zum einen ist unsere App, passend zu unserem Marketingslogan "Made for everyone", einfach zu bedienen. Dies drückt sich dadurch aus, dass der Nutzer nicht aktiv nach einer Aktie suchen muss, sondern die beste Auswahl passend zu seiner Risikoneigung und Marktstimmung vorgeschlagen bekommt. Ebenso wird bei unserer App keine pure Analyse der aktuellen oder Quartalsbericht-Daten durchgeführt, da die Vorhersagen des Risikos auf zukünftigen Zahlen beruht. Darüber hinaus, wird die Analyse nicht auf vergangenen Daten durchgeführt, da hierbei tagesaktuelle Kurse und Stimmungen am Markt

analysiert werden. Zusätzlich werden bei unserer App heterogene Faktoren wie die Marktstimmung und historische Daten der einzelnen Aktien mit einbezogen, bevor diese dem Anwender ausgegeben werden. Außerdem wird die Analyse nicht durch subjektive Analysenmeinungen verunreinigt, da die verwendete Künstliche Intelligenz für Neutralität bei der Analyse sorgt.

3.2.2 Zielgruppenanalyse

Für unsere StockTimes App haben wir auch eine Zielgruppenanalyse durchgeführt um feststellen zu können, welche Zielgruppe angesprochen und welchen Charakteristika diese entsprechen soll. Privatpersonen und Firmen, die Interesse am Aktienmarkt und deren Entwicklungen haben, sind die Zielgruppe unserer Anwendung. Ebenso ist diese an keinen bestimmten Standort gebunden und kann somit Überregional genutzt werden. Darüber hinaus, richtet sich diese nicht nur an bestehende Kunden sondern auch Interessenten und potentielle Neukunden. In einem zweiten Schritt haben wir unsere zuvor definierte Zielgruppe charakterisiert. Mit unserer App möchten wir Unternehmen im Bankensektor und am Finanzmarkt ansprechen. Darüber hinaus muss die Zielgruppe keine weiteren Eigenschaften in Bezug auf Bildung, Nationalität und Region erfüllen, da diese wie zuvor in der Bestandsanalyse erwähnt einfach und verständlich zu bedienen ist. Des Weiteren sollte der Nutzer offen für Neues sein, wie den Einbezug neuer nicht traditioneller Ideen und neuer Technologien wie zum Beispiel das Maschinelle Lernen. Außerdem sollte die Zielgruppe Spontanität und Risikobereitschaft beim Kauf einer Aktie mitbringen, da die Einstellung der Risikobereitschaft beim Kauf einer Aktie ein wichtiger Teil der App darstellt. Zum Schluss, wurde das Kaufverhalten der Zielgruppe untersucht, indem die Kaufmotive, die Zeitpunkte und die Inhalte des Kaufes untersucht wurden. Als Kaufmotive stellten wir fest, dass der Anwender Interesse am Finanzmarkt haben sollte, aber zusätzlich dem Risiko, welches durch die Investition eingegangen wird Beachtung schenken sollte. Die Zielgruppe investiert, wenn der Aktienkurs steigt und die Marktstimmung positiv ist und sich dadurch mehr Ertrag erhofft wird. Ebenso wird die Zielgruppe bei einer negativen Marktstimmung weniger investieren und Risiko eingehen wollen.

3.2.3 Kosten-Nutzen-Analyse

Darüber hinaus wurde eine Kosten-Nutzenanalyse erstellt, die einen Überblick über die auftretenden Kosten und Nutzen bei Erstellung unserer Anwendung geben. Jeder Bestandteil der Aufwendungen und Nutzen wurde pro Mitarbeiter und auf die gesamte Mitarbeiteranzahl bestimmt. Um die Applikation zu erstellen werden folgende Aufwendungen benötigt: die Frontendentwicklung, das Finden einer passenden Datenquelle, die Datenvorbereitung, das Modelltraining, die Durchführung von Unit-Tests und am Ende die Zusammenführung des Front- und Backends. Die Bereitstellung unserer Anwendung und die Erzielung positiver Ergebnisse durch die Investition, in die von uns vorgeschlagenen Aktien sorgen für eine positive Darstellung unserer Marke am Markt. Des weiteren, erhalten wir durch das positive Marketing weitere Aufträge und unsere Mitarbeiter können dadurch an Know-How gewinnen. Wie in der Abbildung zu erkennen ist, erhält man durch Summierung der Aufwendungen einen Gesamtaufwand von 36.400 EURO für die komplette Erstellung der StockTimes App. Als einmaligen Wert des Nutzens oder Einsparungen wurde der Wert 5.000 EURO geschätzt. Dieser kann aber dauerhaft nach Verwendung der App erzielt werden, da es zu weiteren Folgeaufträgen kommen könnte. Somit könnte sich der negative Saldo aus der Kosten-Nutzenanalyse nach einigen Monaten oder Jahren amortisieren und sich nicht negativ auf unser Unternehmen auswirken. Bei den ermittelten Werten handelt es sich um Schätzungen. Dabei wurde geschätzt, wie viel ein Arbeitnehmer innerhalb der Laufzeit des Projektes, hierbei drei Monate, und spezifisch auf die von ihm ausgeführte Arbeit erhalten würde.

Aufwand	Pro Mitarbeiter	Gesamt
Frontend Entwicklung	3500	7000
Datenquelle finden	2000	4000
Daten vorbereiten	3000	6000
Aktienvorschlag nach Risikoneigung	2500	10000
Modelle trainieren	2000	2000
Unit Test	2500	5000
Zusammenführung Front- & Backend	1200	2400
Gesamtaufwand	16700	36400
Nutzen/ Einsparungen		
Marketing für unsere Firma, Folgeaufträge, Know-How Gewinn der Mitarbeiter		5000
Saldo aus Nutzen und Kosten		31400

Abbildung 3.3: Kosten-Nutzen-Analyse
Quelle: Eigene Abbildung

3.2.4 Auswirkungen auf Marketing / Branding

Is neues und innovatives Unternehmen ragen wir aum den Markt hervor, weil wir intelligente Produkte mit Hilfe von neuen, modernen Technologien wie Machine Learning (ML) und künstlicher Intelligenz (KI) entwickeln.

Wir vermarkten unser Produkt als neu, innovativ und unverwechselbar, da wir mit unserem Produkt den Nutzern die Möglichkeit geben das potenzielle Aktienrisiko besser einschätzen zu können. Der Nutzer gibt selbst sein gewünschtes Risiko ein und erhält schnelle und für ihn optimierte Vorschläge zum Aktienkauf, die sein Risiko- und sein Verlustgrad besser optimieren. Die zugrunde liegenden Vorschlägen beruhen auf einer Vorhersage basierend auf vielen Daten, welche aus verschiedenen Informationsquellen genommen werden. Das Produkt trifft diese Vorhersage mit Hilfe der künstlichen Intelligenz und kann dadurch das mögliche Risiko besser einschätzen. Das Produkt ist einfach anzuwenden und steht allen Interessenten zur Verfügung.

Aufgrund der Unverwechselbarkeit und des Technologievorsprungs hat das Produkt die Möglichkeit sich als Marke zu etablieren. Die Markenbekanntschaft kann gesteigert werden, indem wir unser Produkt an andere Unternehmen bzw. anderen Kunden zur Verfügung stellen oder es verkaufen.

Zusammenfassend lässt sich sagen, dass unser neues Produkt eine gute Chance hat sich am Markt zu etablieren.

3.2.5 SWOT-Analyse

Um die Entwicklung unseres Unternehmens StockTimes zu analysieren, die weitere Unternehmensstrategie zu entwickeln und zu planen haben wir auf die SWOT-Analyse zurückgegriffen. Ein anderes Wort dafür ist die Stärken-Schwächen-Analyse. Das "S" in SWOT steht für "strength" (Stärke), "W" für "weaknesses" (Schwäche), "O" für "opportunities" (Chancen) und "T" für "threat" (Risiken)(Asana, 16.02.2022).

Eine SWOT-Analyse ist eine Kombination von Umweltanalyse (externe Analyse) und Unternehmensanalyse (interne Analyse). In der Unternehmensanalyse werden die Stärken und Schwächen des Unternehmens aufgezeigt und bei der Umweltanalyse die Chancen und Risiken(Asana, 16.02.2022).

Zu den Stärken unseres Unternehmens zählen, dass wir ein neues und innovatives Unternehmen sind. Wir haben kreative Produktideen, wie beispielsweise die Aktientipps auf Grund von Nachrichten und Risikoeinstellung. Auch haben wir das nötige Know-How, sowie qualifizierte Mitarbeiter, da unser Team aus Studenten besteht, die in den neusten Technologien ausgebildet werden. Unser Unternehmen ist darüber hinaus offen gegenüber Innovationen, Ideen, Produkten und Arbeitsweisen.

Als neues Unternehmen fehlt uns die langfristige Praxiserfahrung im Bereich Banken und Aktienmarkt. Ein weiteres Problem ist unsere Abhängigkeit zum Finanzmarkt und dessen Veränderungen. Eine schlechte Wirtschaftslage bedingt eine geringe Nachfrage nach unserem Produkt und unvorhersehbare Ereignisse die den Finanzmarkt stark beeinflussen können nicht prognostiziert werden.

Zu den Risiken unseres Unternehmens gehören die Gefahr von Kündigungen der Know-How Träger und die Datenabhängigkeit. So kann es z.B. zu Verfälschungen der Daten kommen und damit zu falschen Vorhersagen. Ein weiteres Risiko ist unser Mangel an Erfahrungen und die Frage, ob wir uns als neue Wettbewerber durchsetzen können.

Unsere Chancen sind, dass wir durch die Anwendung von ML neue Produkte erstellen und unsere Wettbewerbsfähigkeit dadurch steigern können. Außerdem erzielen wir anderen gegenüber Vorteile, weil wir Vorteile Trends verfolgen und auf sie eingehen können.

Aus den Ergebnissen lässt sich schlussfolgern, dass die Stärken die Schwächen und die Chancen die Risiken überwiegen, sodass es sich lohnt die Geschäftsidee weiter zu verfolgen.

Stärken <ul style="list-style-type: none"> • Neues, innovatives Unternehmen • Innovative Produktidee • Know-How • Qualifizierte Mitarbeiter • Offenes Unternehmen (Technologien, Ideen, Produkte & Arbeitsweisen) 	Chancen <ul style="list-style-type: none"> • Anwendung von ML -> Wettbewerbsfähig, Vorteil gegenüber anderen, Trends verfolgen & mitgehen • Technologien -> neue Produkte entstehen
Schwächen <ul style="list-style-type: none"> • Neues Unternehmen • ML noch unerfahren / neues Gebiet • Abhängig vom Finanzmarkt und dessen Veränderungen 	Risiken <ul style="list-style-type: none"> • Kündigung von Know-How Trägern • Ergebnisse abhängig von den Daten • (neue) Wettbewerber

Abbildung 3.4: SWOT-Analyse.

Quelle: Eigene Abbildung

3.3 Systementwurf

Für die nachfolgende Implementierung der Problemstellung dient ein Systementwurf als Grundlage. Im folgenden Kapitel wird dieser erläutert.

Einen schnellen Überblick über die Funktionalitäten und Möglichkeiten zeigt das ERM-Diagramm (Abbildung 3.5):

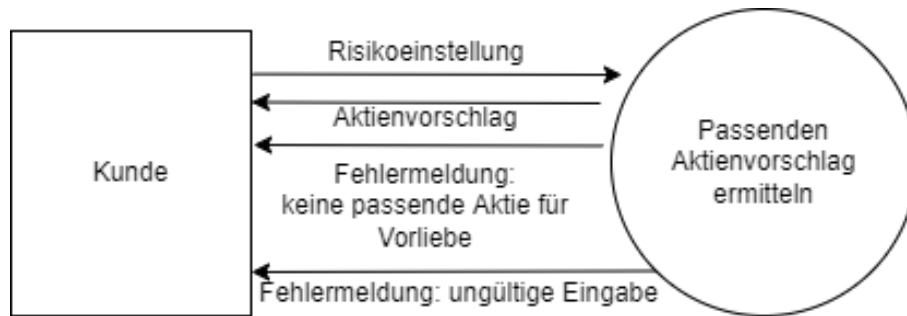


Abbildung 3.5: Ereignis-Reaktions-Modell
Quelle: Eigene Abbildung

Zuerst soll der Kunde, der als Nachbarsystem fungiert, das Risiko einstellen können. Dieser Datenfluss löst beim System die Funktion „passender Aktienvorschlag ermitteln“ aus. Anschließend gibt es drei mögliche Datenflüsse als Antwort: Wenn das System einen passenden Aktienvorschlag ermitteln kann, dann wird dieser zurückgegeben. Ansonsten erhält der Kunde vom System entweder die Fehlermeldung, dass keine passende Aktie für die Vorliebe existiert, oder den Hinweis, dass seine Eingabe ungültig ist.

Eine genauere Betrachtung vor allem in Bezug auf die Parallelität der Abläufe ermöglicht das Aktivitätsdiagramm in Abbildung 3.6.

Hier wird gezeigt, dass nach der Einstellung des Risikos noch eine Überprüfung mithilfe des Aktivitätsschrittes „Risikovorliebe“ erfolgt, ob das Risiko innerhalb des erlaubten Bereichs ist. Wenn das bei dem darauffolgenden Entscheidungsknoten nicht der Fall ist, wird die Fehlermeldung „ungültige Eingabe“ ausgegeben. Ansonsten wird versucht einen passenden Aktienvorschlag zu ermitteln. Diese Ermittlung kann wiederum entweder erfolgreich sein, wobei als Folgeschritt der ermittelte Aktienvorschlag ausgegeben wird, oder fehlschlagen, was zu der Ausgabe „keine passende Aktie gefunden“ führt.

Hiermit sind die grundlegenden Funktionalitäten und Ausgaben des Systems abgebildet. Für eine noch genauere Darstellung, welche Bedingung zu welchem Ergebnis führt, befindet sich im Anhang eine Entscheidungstabelle (Abbildung 6.1). Eine formale Beschreibung des Anwendungsfalls befindet sich mit der Use-Case-Schablone (Abbildung 6.2) ebenfalls im Anhang.

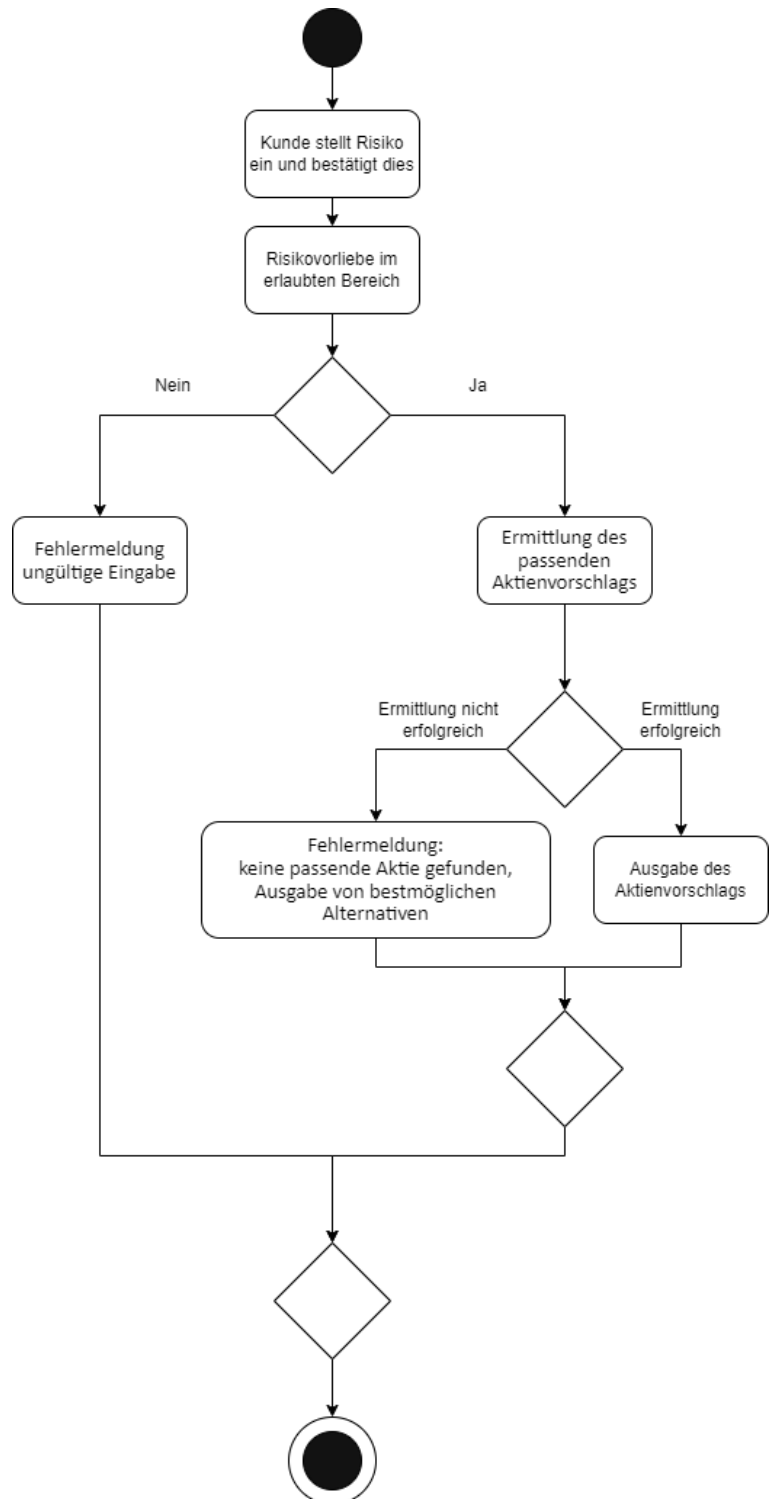


Abbildung 3.6: Aktivitätsdiagramm
Quelle: Eigene Abbildung

4 Umsetzung

4.1 Frontend

Die direkte Verbindung des Backends mit dem Kunden geschieht im Frontend. Hier werden die einzelnen Funktionalitäten graphisch dargestellt und für den Nutzer verwendbar. Es wird im Folgenden auf die technischen Möglichkeiten, sowie auf den Aufbau des Frontends eingegangen.

4.1.1 Technische Möglichkeiten

Bei den Techniken des Frontends war das GUI-Toolkit Tkinter für Python von zentraler Rolle. Mit diesem Modul war es uns möglich eine grafische Benutzeroberfläche zu erstellen. Diese wurde mithilfe eines sogenannten Layout Manager designt. Dabei hat man durch ein Grid die Möglichkeit alle Elemente des Frontends mit einem einheitlichen Layout darzustellen. Das Grid basiert auf Tabellen, heißt auf Zeilen und Spalten. Man legt fest in welcher Zeile bzw. Spalte das jeweilige Element angezeigt werden soll. Dies ermöglicht eine einfache Anordnung der Elemente auf gleicher Höhe oder Breite der Anwendung.

Bei der Anwendung gibt es mehrere Bedienelemente. Eins der wichtigsten Elemente sind die Knöpfe. Dabei gibt es einen Knopf um die jeweilige Risikoeingabe zu bestätigen. Dieser wird genutzt um die passenden Aktien zur gewählten Risikoneigung anzuzeigen. Außerdem gibt es einen Knopf um die Marktstimmung zu ermitteln. Dieser ist wichtig für die Sentiment Analyse. Dabei wird das Sentiment neu berechnet und somit eine potenziell neue Marktstimmung angezeigt. Außerdem gibt es einen Knopf um die StockTimes-App zu verlassen. Dieser Knopf schließt die App. Zusätzlich gibt es verschiedene Labels. Diese sind die Textfelder in der App. Außerdem wurde eine MessageBox eingebaut. Sollten für die ausgewählte Risikoneigung in Kombination mit der momentanen Marktstimmung keine Aktien vorhanden sein, dann zeigt die App durch die MessageBox eine Fehlermeldung an. Somit weiß der Kunde immer, wann ihm keine Aktien empfohlen werden können.

Die verschiedenen Aktien werden in einer Json-Datei ins Frontend geladen. Mithilfe dieser erhält man das Sentiment und die Sektoren einer Aktie. Nach Berechnung des Sentiments

auf dem Aktienmarkt wird ein Smiley angezeigt der die momentane Marktstimmung wiedergibt. Ein glücklicher steht für eine positive, ein trauriger für eine negative und ein neutraler für eine neutrale Marktstimmung. Durch einen Slider kann der Kunde die von ihm gewollte Risikoneigung wählen. Bei Betätigung des Eingabeknopfes erhält der Kunde die Aktien denen das gewählte Risiko zugeordnet ist.

4.1.2 Produktpräsentation

In diesem Abschnitt wird auf den Aufbau und die Funktionsweise unserer App eingegangen. Auf der linken Seite, kann die Risikoneigung des Nutzers durch einen Schieberegler ausgewählt werden, die beim Aktienerwerb eingegangen werden soll. Darüber hinaus wird die aktuelle Marktstimmung anhand eines Smileys dargestellt, der die aktuelle Marktentwicklung zeigt und dabei behilflich sein soll, das Risiko beim Aktienerwerb besser einschätzen zu können. Hierbei kann es sich entweder um eine positive, neutrale oder negative Marktstimmung handeln, die zuvor durch eine Sentiment Analyse ermittelt wurde. Unterhalb der Marktstimmung befindet sich ein Schieberegler, der dem Anwender die Möglichkeit gibt zwischen sechs Risikoklassen zu wählen. Die Risikoneigungen teilen sich in die sechs Risikoklassen, sehr geringe Risikoneigung, geringe Risikoneigung, mittlere Risikoneigung, hohe Risikoneigung, sehr hohe Risikoneigung und keine Risikoneigung, auf. Zusätzlich wird die aktuelle Risikoneigung am unteren Rand der App angezeigt, damit der Nutzer weiß welche Risikoneigung gerade ausgewählt ist. Wenn sich der Anwender für eine Risikoklasse entschlossen hat, kann diese durch eine Taste bestätigt werden. Durch die Taste, Eingabe Bestätigen, werden auf der rechten Seite fünf Aktienvorschläge ausgegeben. Diese wurden zuvor nach Sektoren gefiltert, die der aktuellen Marktstimmung entsprechen, die zuvor durch die Sentiment Analyse ermittelt wurden. Bei den Sektoren kann es sich zum Beispiel um den Energie- und den Kommunikationssektor handeln. Die Aktien, die auf der Anwendung zu sehen sind werden zufällig ausgegeben, damit der Nutzer eine Wahlmöglichkeit zwischen den einzelnen Aktien hat. Unterhalb der Aktienvorschläge befinden sich zwei Tasten mit denen die aktuelle Marktstimmung durch eine erneute Sentiment Analyse ermittelt und die App geschlossen werden kann. Des Weiteren können dem Nutzer auch Fehlermeldungen ausgegeben werden, falls es zur ausgewählten Risikoneigung sowie den Sektoren nicht ausreichend viele Aktienvorschläge gibt.

The screenshot shows the 'Herzlich willkommen bei StockTimes!' (Welcome to StockTimes!) screen. It is divided into two main sections. The left section, titled 'Wie viel Risiko wollen Sie eingehen?' (How much risk do you want to take?), features a 'Marktstimmung:' (Market Mood) slider set to 0, accompanied by a sad face icon and a 'Eingabe Bestätigen' (Confirm Input) button. The right section, titled 'Unsere Vorschläge für keine Risikoneigung' (Our suggestions for no risk aversion), lists five companies: Tenet Fintech Group Inc., Smith & Wesson Brands, Inc., Super League Gaming, Inc., Integrated Ventures, Inc., and Ecosciences, Inc. At the bottom, there are three buttons: 'Aktuelle Risikoneigung: Absicherung' (Current risk aversion: Hedging), 'Marktstimmung ermitteln' (Determine market mood), and 'StockTimes verlassen' (Leave StockTimes).

Abbildung 4.1: Frontend der App
Quelle: Eigene Abbildung

4.2 Sentiment Analyse

Ein weiterer wichtiger Bestandteil der gebauten App ist die Sentiment Analyse. Dieses Kapitel dient dazu, dem Leser einen kurzen Überblick darüber zu geben, wie eine Sentiment Analyse allgemein funktioniert und wie wir sie für StockTimes eingesetzt haben. Bei der Umsetzung für StockTimes wird auf die zugrundeliegende Datenbasis ebenso eingegangen, wie auf das genutzte Python-Modul.

4.2.1 Grundlagen Sentiment Analyse

Unter Sentiment Analyse versteht man grundsätzlich ein Klassifizierungsproblem, welches versucht durch ein allgemeines Stimmungsbild die Stimmung eines vorliegenden Textes zu analysieren. Um das zu tun wird TextMining, eine automatisierte Auswertung von Textdateien verwendet. Durch TextMining werden bestimmte Worte aus dem vorliegenden Text herausgefiltert. Diese werden anschließend analysiert. Dafür gibt es zwei Methoden. Zum einen gibt es eine einfache Methode, nämlich die herausgefilterten Worte mit einem vorgefertigten Wörterbuch zu vergleichen. In diesem Wörterbuch stehen vorher festgelegte Wörter, welche einer bestimmten Polarität zugeordnet sind. Die Polarität entscheidet, ob die Wörter als positiv, negativ oder neutral zu klassifizieren sind. Bei der einfachen Methode werden die herausgefilterten Worte nur mit dem Wörterbuch verglichen und am Ende gezählt, wie viele positive, negative oder neutrale Worte in einem Text vorkommen. Die Klasse mit der höchsten Wortanzahl ist die Klasse des Textes. Ist die Mehrheit der Wörter positiv, wird der gesamte Text als positiv gekennzeichnet. Ein Problem dabei ist, dass nur

die Wörter gezählt werden und dadurch keine Ironie erkannt werden kann und Sätze, die eigentlich eine negative Bedeutung haben auf Grund der Wörter als positiv klassifiziert werden können. Eine Möglichkeit, dieses Problem zu umgehen ist der Einsatz von Machine Learning. Durch Machine Learning wird versucht, Ironie, oder Sätze die nur auf Grund des Kontexts als positiv oder negativ identifiziert werden können zu erkennen und dann richtig zu klassifizieren. Der Unterschied zur einfachen Methode ist also, dass über das einfache Zählen von klassifizierten Wörtern hinausgegangen wird und auch der Kontext sowie der Aufbau des Textes mit einbezogen werden. Das Machine Learning stellt also eine Erweiterung zur einfachen Methode da. (Vgl. „Was ist Sentiment-Analyse?“, 2022) (Vgl. Geierhos, n. d.)

4.2.2 Sentiment Analyse bei StockTimes

Datenbasis

Als Datenbasis für StockTimes haben wir uns dafür entschieden Twitterdaten, also Tweets zu nutzen. Das hat zwei Gründe. Zum einen ist der Zugriff auf Twitterdaten durch die frei zugängliche Twitter-API sehr einfach. Zum anderen sind Tweets aber auch eine gute Möglichkeit, die Stimmung vieler Menschen auf einmal analysieren zu können. Das liegt daran, dass Twitter von vielen verschiedenen Personen unterschiedlichen Alters, unterschiedlicher Herkunft sowie unterschiedlicher Berufe genutzt wird. Kurz gesagt, man kann mit Hilfe von Tweets einen guten Überblick über die gesamte Gesellschaft bekommen, da Personen aus so gut wie allen gesellschaftlichen Gruppen Twitter nutzen. Nachdem wir uns also für Twitter als Datenlieferant entschieden haben mussten wir noch festlegen, welche Tweets wir überhaupt nutzen wollen. Dafür haben wir verschiedene Such-Querys an die Twitter-API weitergegeben. Die Twitter-API gibt uns dann nur Tweets zurück, welche die Wörter aus der Such-Query beinhalten. Unsere Such-Querys waren: "aktien OR aktie", um allgemeine Tweets über Aktien zu erhalten, "aktien gut OR aktie gut", um positive Tweets über Aktien zu erhalten und "aktien schlecht OR aktie schlecht", um negative Tweets über Aktien zu erhalten. Diese drei Suchanfragen haben wir zu einem Datensatz zusammengefügt, auf dem wir anschließend die Sentiment Analyse durchgeführt haben. Durch die Keywords aktie und aktien werden nur deutschsprachige Tweets geliefert, da nur die Tweets ausgewählt werden, die das Wort beinhalten. Unsere Datenbasis besteht also ausschließlich aus deutschsprachigen Tweets.

Sentiment Analyse in Python

Für StockTimes haben wir uns entschieden, die Sentiment Analyse mit Hilfe der Bibliothek TextBlobDE durchzuführen. TextBlobDE verwendet ein vorgefertigtes Wörterbuch mit ausgewählten Wörtern und einer zugeordneten Polarität. Diese Polarität gibt an, wie stark ein Wort zu einer der Gruppen (positiv, negativ, neutral) gehört. Zum Beispiel wird das Wort "gut" mit einer Polarität von 1.0 gewertet, wohingegen das Wort "schlecht" mit einer Polarität von -1.0 gewertet wird. TextBlobDE summiert alle Polaritätswerte eines Tweets auf und gibt das Ergebnis dann zurück.

Implementierung der Sentiment Analyse

Für die Sentiment Analyse haben wir die eben beschriebenen Bausteine genutzt. Die folgende Grafik zeigt, wie unser Vorgehen dabei war.



Abbildung 4.2: Prozess der Sentiment Analyse
Quelle: Eigene Abbildung

Zuerst haben wir uns wie oben beschrieben, Tweets der letzten sieben Tage, welche die geforderten Keywords enthalten, heruntergeladen. Diese Tweets wurden, wie beschrieben, in einen Datensatz zusammengefasst. Jeder Tweet aus dem Datensatz wurde anschließend einzeln von TextBlobDE bewertet. Auf Basis dieser Bewertung wurden die Tweets den Klassen "positiv", "negativ" und "neutral" zugewiesen. Dabei ist wichtig zu sagen, dass alle Tweets kleiner 0 negativ klassifiziert wurden, alle gleich der 0 neutral und alle größer als 0 positiv. Anschließend wurde gezählt, wie viele Tweets zu jeder Klasse gehören. Die Klasse mit den meisten Tweets entscheidet, aus welchen Sektoren wir dem Kunden

Aktien vorschlagen. Wenn als Klasse "negativ" rauskommt, werden dem Kunden Aktien aus den Sektoren "Healthcare", "Utilities", "Basic Materials" und "Consumer Defensive" vorgeschlagen. Für die Klasse "positiv" Aktien aus den Sektoren "Financial Services", "Technology" und "Real Estate". Die Klasse "neutral" wiederum liefert Aktien aus den Sektoren "Industrials", "Communication", "Consumer Cyclical" und "Energy". Nachdem also die Sektoren ausgewählt wurden werden Tweets mit den genannten Keywords für jeden einzelnen Sektor heruntergeladen und anschließend wieder von TextBlobDE bewertet und klassifiziert. Diese zweite Runde von Tweets herunterladen, bewerten und klassifizieren hat den Grund, dass sich dadurch die Sektoren in eine Reihenfolge einteilen lassen. Der Sektor mit der größten Zustimmung, sprich den meisten positiven Tweets, ist die Nummer 1. Danach geht es in absteigender Reihenfolge mit den anderen Sektoren weiter. Aus dem Sektor, der Nummer 1 ist werden dem Kunden mehr Aktien vorgeschlagen als aus dem Sektor auf dem letzten Platz. Nachdem die Sektoren also auf Basis der Marktstimmung ausgewählt wurden und entsprechend ihrer jeweiligen Zustimmung sortiert wurden werden sowohl die Stimmung, als auch die Reihenfolge der ausgewählten Sektoren an das Frontend übergeben. Dort werden abschließend final mit Hilfe der Liste aus dem Backend die passenden Aktien ausgewählt und dem Kunden präsentiert.

4.2.3 Reflektion Sentiment Analyse

Während der Überlegung, wie wir die Sentiment Analyse umsetzen können, sowie bei der Implementierung sind wir auf mehrere Hindernisse gestoßen. Diese Hindernisse und wie sie gelöst wurden wird im Folgenden näher erklärt.

Hindernisse bei der Implementierung

Das erste Hindernis bei der Implementierung war, dass die Tweets am Anfang mit einer überwiegenden Mehrheit (95%) "Neutral" waren, da zu wenige positive oder negative Adjektive enthalten waren. Die Mehrheit der Tweets war nur eine sachliche Beschreibung einer bestimmten Aktie, welche keine Aussage über die allgemeine Stimmung zulässt.

Ein zweites Hindernis war, dass wir zuerst alle Tweets mit einer Polarität zwischen -0.3 und 0.3 als neutral klassifiziert haben. Dadurch war selbst bei den Tweets mit mehr Adjektiven der Anteil noch bei ca. 90%.

Lösung der Hindernisse

Die Lösung des ersten Hindernisses war es, die ursprüngliche Suchanfrage "aktie OR aktien" mit den anderen beiden oben genannten zu kombinieren, und die dadurch erhaltenen einzelnen Datensätze wiederum zu einem großen. Durch die Nutzung eines großen Datensatzes hat sich die Menge der neutralen Tweets verkleinert und ein realistischeres Bild der aktuellen Marktstimmung ist entstanden.

Um das zweite Hindernis zu lösen haben wir die Schwellwerte, ab denen etwas als "Neutral" klassifiziert wird verändert. Es wurden nur noch Tweets mit einer Polarität von genau 0 als "Neutral" klassifiziert. Dadurch ist die Menge der neutralen Tweets geschrumpft und die anderen beiden gewachsen.

4.3 Risikoklassifizierung

Die im vorangegangenen Kapitel erläuterte Sentimentanalyse gibt Aufschlüsse über die Stimmung am Markt. Je nachdem wie diese ausfällt werden unterschiedliche Branchen ausgewählt. Hierdurch wird also das allgemeine Risiko des Marktes und einzelner Branchen berücksichtigt, was auch systematisches Risiko oder Marktrisiko genannt wird. Für die Auswahl und Ausgabe einzelner Aktien muss jedoch darüber hinaus noch das spezifische Risiko der Aktie ermittelt werden. Dieser Punkt ist Aufgabe und Ziel der Risikoklassifizierung, die im Folgenden erläutert wird.

4.3.1 Erster Ansatz

Die Grundidee des ersten Ansatzes besteht darin das spezifische Risiko einer Aktie, was durch den Betafaktor repräsentiert wird, anhand von Fundamentaldaten der Aktie und sonstigen Informationen des Unternehmens wie den Sektor vorherzusagen. Dafür werden zuerst im Unterkapital Datenauswahl passende Informationen über die Aktie abgefragt und verarbeitet. Anschließend wird das Modell trainiert und am Ende des Kapitels erfolgt die Modellevaluation.

Datenauswahl

Für die Datenauswahl sind vor allem die Kriterien Aktualität, Konsistenz, aussagekräftige Daten sowie ein geeignetes Label für die Klassifizierung wichtig. Diese Anforderungen und Lösungen dazu werden hier erläutert.

Die Aktualität der Daten ist vor allem für eine genaue Vorhersage existenziell, da starke Schwankungen vor allem bei den Aktienkursen der jeweiligen Unternehmen möglich sind. Damit dies nicht zu falschen Vorhersagen führt und die Aktualität gewährleistet ist, werden die Daten über die Yahoo Finance-API abgefragt. Hierdurch werden tagesaktuelle Kurse gewährleistet. Durch die Wahl von nur einer Datenquelle verbessert sich auch die Konsistenz.

Für möglichst aussagekräftige Features, also Eingabewerte, mit denen der Klassifikator eine Zielvariable oder auch Label vorhersagen soll, werden verschiedene Kennzahlen berechnet. Diese sollen das Unternehmen möglichst ganzheitlich abbilden, um somit die Klassifikation zu ermöglichen. Hierfür werden aus den Jahresbilanzen der Unternehmen Kennzahlen des Gewinns (Gewinnwachstum, Kurs-Gewinn-Verhältnis), des Umsatzes (Umsatzwachstum, Kurs-Umsatz-Verhältnis) und der Kapitallage (Veränderung des Cash-Flows, Eigenkapitalquote) verwendet. Diese Daten werden mit dem Sektor und dem Land, also allgemeinen Daten des Unternehmens, sowie Analystenempfehlungen komplementiert.

Als Label für die Klassifizierung dient wie anfangs erläutert der spezifische Risikofaktor Beta. Dieser gibt an, wie stark ein Unternehmen im Vergleich zum Gesamtmarkt für einen bestimmten Betrachtungszeitraum schwankt. Diese Kennzahl kann berechnet werden, indem die Kovarianz aus Markttrendite und Aktienrendite durch die Varianz der Markttrendite geteilt wird.

Der Betafaktor dient als Standardverfahren, um eine passende Risikobewertung für ein Unternehmen und eine nachvollziehbare Wertfindung zu ermöglichen. Dennoch gibt es einige Kritikpunkte: Zuerst ist das berechnete Beta vergangenheitsorientiert und sagt nicht direkt etwas über die zukünftige (Risiko-) Entwicklung des Unternehmens aus. Damit Aussagen über das zukünftige Risiko getroffen werden können, wird das Beta der Folgeperiode berechnet und soll anschließend für die Unternehmen mithilfe von Machine Learning vorhergesagt werden. Der Zweite Kritikpunkt ist, dass das Beta nicht das Risiko des Gesamtmarkts berücksichtigt. Dieser Punkt wird aber bereits bei der Sentiment-Analyse abgedeckt (DeltaValue.de, 08/02/2022, Vgl.).

Die Umsetzung der Datenklassifizierung erfolgt mithilfe des Skripts „data_prep.py“. Ein beispielhafter Ausschnitt des vorverarbeiteten Datensatzes kann der Abbildung 4.3 entnommen werden.

	name	sector	...	kuv	beta_following_period
2018-09-29	Apple Inc.	Technology	...	3.357676	NaN
2019-09-28	Apple Inc.	Technology	...	3.374122	1.109896
2020-09-26	Apple Inc.	Technology	...	6.628599	1.412091
2021-09-25	Apple Inc.	Technology	...	6.550687	1.173193

Abbildung 4.3: Vorbereitete Daten
Quelle: Eigene Darstellung

Modelltraining

Um ein Modell zu trainieren, welches die Betas der Folgeperiode vorhersagt, wurden zunächst die Daten entsprechend vorbereitet. Dafür wurden zum Beispiel die Betas in 6 Klassen eingeteilt, welche verschiedene Risikostufen repräsentieren.

Außerdem werden einige Spalten wie z.B. "Name der Firma", die für das Trainieren eines Modells nicht benötigt werden, von dem Datensatz entfernt.

Nachdem die Daten vorverarbeitet wurden, wird der Datensatz skaliert, damit alle Werte zwischen Null und Eins liegen. Dies verhindert, dass einige Features durch das Machine Learning Modell mehr gewichtet werden als andere (Vgl. Baijayanta Roy, n. d.).

Nachdem die Daten skaliert sind und entsprechend in Trainings- und Test-Datensätze eingeteilt sind, erfolgt das Trainieren der Machine Learning-Modellen für die Vorhersage der Betas bzw. die gewählten Risikoklassen.

Im Rahmen des Trainings wurden viele Klassifikatoren, also Machine Learning Algorithmen, die Daten klassifizieren, trainiert. Unter anderem wurden Logistic Regression, Decision Tree, Kneighbors, Linear Discrimantal Analysis, Gaussian Naive Bayes sowie Support Vector Machine (SVM) Klassifizier trainiert.

Evaluation

Als Grundlage der Evaluation wurden „Accuracy“ und Confusion Matrix, auch als Fehlermatrix bekannt, herangezogen. Confusion Matrix ist ein spezifisches Tabellenlayout, das die Visualisierung der Güte des trainierten Modells ermöglicht (Vgl. Jason Brownlee, 2020). „Accuracy“, auch Genauigkeit genannt, ist eine Metrik zur Bewertung von Klassifizierungsmodellen. Informell ist Genauigkeit der Bruchteil der Vorhersagen, die unser Modell richtig gemacht hat (Vgl. Aditya Mishra, 2020).

Die Meisten der oben genannten Klassifizierer haben eine ähnliche Performance aufgewiesen. Im Folgenden wird anhand der SVM-Klassifizierer die Güte des Modells analysiert.

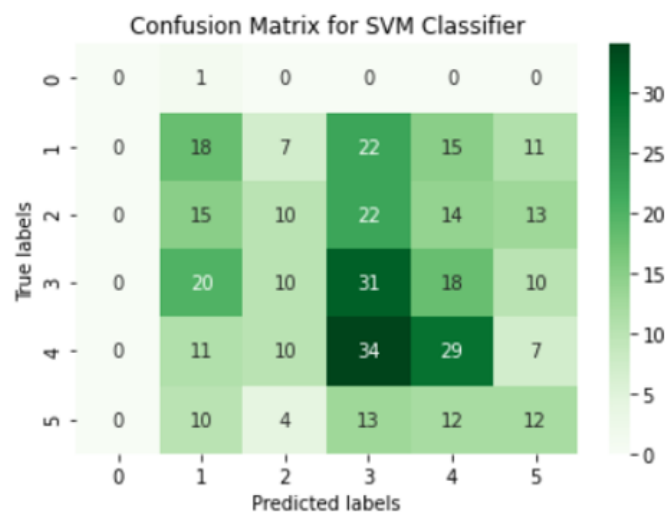


Abbildung 4.4: Confusion Matrix für SVM Klassifizier
Quelle: Eigene Darstellung

Auf der horizontalen Achse sind die vorhergesagten Klassen dargestellt. Auf der vertikalen Achse sind die echten Klassen dargestellt. Es ist zu sehen, dass die Klasse Null gar nicht vorhergesagt wird, was jedoch damit zusammenhängt, dass diese Klasse in dem Datensatz stark unterrepräsentiert ist.

Das Model performt jedoch auch nicht gut bei der Vorhersage anderer Klassen. Die relative schlechte Performance des Modells ist bei der ersten Klasse zum Beispiel besonders deutlich. Während die Klasse „1“ 18 mal richtig vorhergesagt wurde, wurde diese Klasse 56 mal (15+20+11+10) falsch vorhergesagt.

Bleibt man bei der Häufigkeit der Vorhersage der Klasse „1“, merkt man, dass sehr oft die vorhergesagte Klasse „1“ eigentlich einer viel höheren Klasse zugehörig ist.

So wurde zum Beispiel 20 mal die Klasse „1“ vorhergesagt, obwohl die echte Klasse „3“ wäre.

Dies stellt ein großes Problem für die Anwendung dar, denn um eine möglichst passende Aktie anhand der Risikoeinstellung der Nutzer vorschlagen zu können, braucht man ein Modell, das die Risikoklasse der einzelnen Aktien zumindest annähernd richtig vorhersagt. Eine große Abweichung zwischen der vorhergesagten Klasse und der richtigen Klasse würde zu einem falschen Aktienvorschlag und somit einer fehlerhaften Investitionsentscheidung führen.

Um die Ursache für die schlechte Performance von den Modellen zu finden, wurde die Korrelation zwischen den Input-Features und der Zielvariable, auch Label genannt, analysiert. In dem Fall wird die Zielvariable durch die Betas der Folgeperiode repräsentiert. In der folgenden Tabelle ist diese Korrelation in einer Korrelationsheatmap dargestellt.

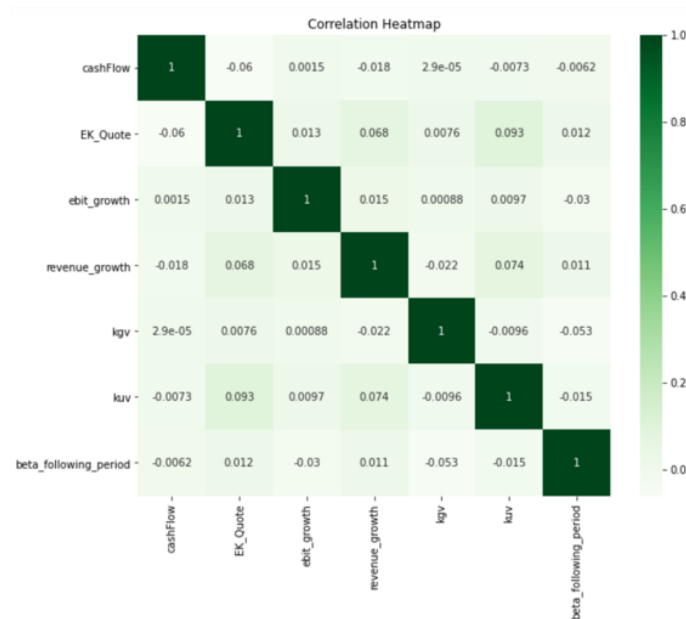


Abbildung 4.5: Korrelationsheatmap
Quelle: Eigene Darstellung

Es ist deutlich zu erkennen, dass die Input-Features und die Zielvariable „beta_following_period“ keine große Korrelation haben. Daher ist eine Vorhersage der zukünftigen Betas anhand dieser Input-Features nur mit einer geringen Genauigkeit möglich.

4.3.2 Zweiter Ansatz

Da der erste Ansatz aufgrund der starken Differenzen der einzelnen Unternehmen und der daraus resultierenden geringen Korrelation der Daten mit dem zukünftigen Beta nicht die erforderliche Güte erreicht hat, ist der Grundgedanke des zweiten Ansatzes nur anhand des Betas der vergangenen Jahre über eine Zeitreihenanalyse das zukünftige vorherzusagen. Hierfür werden wiederum zuerst passende Daten benötigt. Anschließend wird das Modell trainiert und der verwendete Algorithmus beschrieben und das Kapitel mit der Evaluation des zweiten Ansatzes abgeschlossen.

Datenauswahl

Der zweite Ansatz ist in vielerlei Hinsicht ähnlich zu dem ersten Ansatz. Er unterscheidet sich lediglich in der Herangehensweise bei dem Trainieren eines Modells zur Vorhersage von Betas der Unternehmen.

Während beim ersten Ansatz verschiedene Kennzahlen (wie z.B. das Kurs-Gewinn Verhältnis) eines Unternehmens als Input für die Vorhersage der Betas für die Folgeperiode herangezogen werden, wird bei dem zweiten Ansatz eine Zeitreihenanalyse durchgeführt. Dabei werden die verschiedenen Betas eines Unternehmens über einen bestimmten Zeitraum analysiert. Hierbei wurde, um die best möglichste Genauigkeit zu erreichen, der gesamte Zeitraum verwendet, den das Unternehmen an der Börse gelistet ist bzw. unsere Datenquelle zur Verfügung stellt. Das Model soll dann mit Hilfe dieser Zeitreihe, bestehend aus vergangene Betas, die Betas für die Zukunft vorhersagen.

In der folgenden Abbildung sieht man einen Ausschnitt des Datensatzes.

	AAPL	ABBV	ABT	ACN	ADBE	AMAT	AMD	AMGN	AMT	AMZN	...	TTE	TXN	UNH	UNP
2016-09-26	1.118880	0.970062	0.646694	1.084655	3.653503	1.188356	-0.330802	0.419234	0.841920	0.844249	...	2.516470	1.041781	1.298252	1.450814
2016-09-27	0.551523	1.355273	0.909713	0.646566	2.915732	0.906525	1.561437	0.693301	0.874884	1.130297	...	1.845411	0.831651	1.141961	1.035344
2016-09-28	0.813592	1.481104	1.359952	1.283286	1.062998	1.066967	2.424090	0.423326	0.614316	1.758504	...	1.229876	1.311292	0.915358	0.566097
2016-09-29	0.751767	0.814345	1.171871	0.898278	1.069238	1.598159	2.447464	-0.358809	0.079046	1.721416	...	1.851637	1.448942	0.584510	1.061423
2016-09-30	0.802314	1.329890	1.629379	-0.796106	1.370262	0.985357	2.189574	0.373876	-0.050628	1.493296	...	0.673657	0.882732	0.897621	0.809417
...
2022-01-14	1.473418	0.944804	1.155879	2.191781	1.592340	1.940273	2.959859	0.420415	0.141886	1.921947	...	1.689419	0.836150	1.002481	-0.583209
2022-01-18	1.392810	1.594731	0.842288	2.324045	2.138407	4.178470	2.940756	0.633236	0.310685	1.596707	...	2.153442	1.285564	1.252091	-0.205003
2022-01-19	1.344372	1.038171	0.311338	0.055526	1.448977	7.098729	3.734036	1.409753	-0.724376	1.507557	...	1.366148	2.011734	1.013026	0.923523
2022-01-20	1.292155	0.643031	0.021859	-0.421417	1.063763	7.951373	3.578510	1.408108	-0.915983	1.359253	...	0.972780	2.174388	0.959475	1.368408
2022-01-21	0.227361	-2.430128	0.714949	0.619193	1.677691	4.789746	0.023653	-0.506843	-0.746530	-0.187303	...	-1.292065	0.177060	2.305821	5.035354

1342 rows × 89 columns

Abbildung 4.6: Trainingsdaten
Quelle: Eigene Darstellung

In den Spalten sind die verschiedenen Ticker-Symbole der Unternehmen zu sehen. Ein Ticker-Symbol ist eine Abkürzung der Bezeichnung eines börsennotierten Unternehmens. Für die Firma Apple ist beispielsweise das Symbol AAPL reserviert.

Jede Zeile ist das aggregierte Beta dieser Unternehmen für das betrachtete Zeitintervall. Für diesen Anwendungsfall wurden die Betas der Unternehmen über n Jahre in die Vergangenheit gleitend über 30 Tage zu einem Wert zusammengefasst. Diese aggregierten Werte sind in den jeweiligen Zeilen zu sehen.

Die Darstellung der ersten Spalte, welche die Betas von Apple über die Vergangenheit repräsentiert, sieht wie folgt aus.

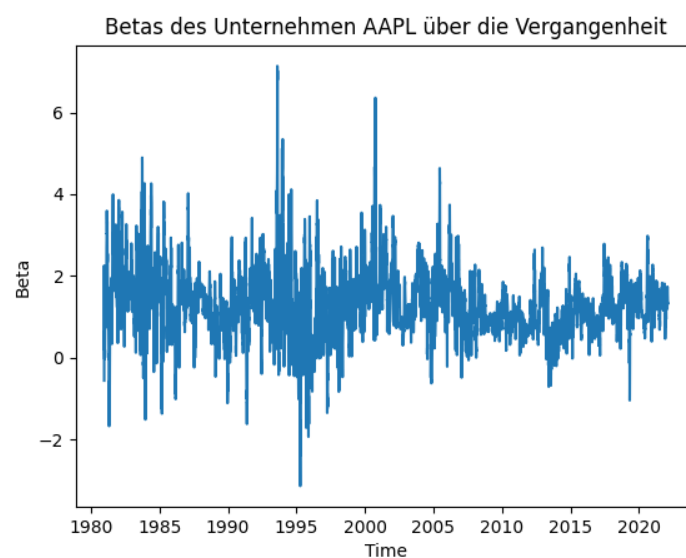


Abbildung 4.7: Betas der vergangenen Jahre von Apple
Quelle: Eigene Darstellung

In der obigen Abbildung wird die Zeitreihe auf der X-Achse und die Entwicklung der Betas auf der Y-Achse dargestellt. Hier wird die Schwankungen der Betas von der Firma Apple in den vergangenen fünf Jahren deutlich.

Modelltraining

Um die Genauigkeit und Aussagekraft zu erhöhen, wird - wie eingangs erwähnt - für jedes Unternehmen ein spezifisches Modell erstellt. Der gesamte Erstellungs- und Vorhersage-

prozess für eine Aktie findet in der Klasse „ModellBetaPredictor“ statt, wodurch die Wiederverwendbarkeit und Übersichtlichkeit des Quellcodes erhöht wird. Die Funktionalitäten und Funktionsweisen dieser Klasse sind Inhalt dieses Kapitels.

Bevor das Modelltraining innerhalb der Objekte der Klasse startet, werden die Daten im Skript skaliert. Anschließend erfolgt die Erstellung eines neuronalen Netzes, wobei die vergangenen Betas als Input dienen und das durchschnittliche Beta der Folgeperiode (z.B. 3 Monate) als Label dient. Das Input-Layer wird von drei Hidden-Layers, bestehend aus zwei LSTM-Layers und einem Dense-Layer, ergänzt. Die Ergebnisse werden wieder in einem „fully connected“ Dense-Layer zusammengeführt. Hierbei sind alle Neuronen in dieser Ebene mit jeweils allen Neuronen der vorangegangenen Ebene verknüpft, wodurch alle vorangegangenen Zwischenergebnisse zu einem Output aggregiert werden.

Der gesamte Trainingsprozess dauert pro Modell ca. 10 Minuten, weswegen jedes Modell - sofern es noch nicht vorhanden ist - abgespeichert wird und für zukünftige Vorhersagen dienen kann. Die darauffolgende Prediction beansprucht unter einer Sekunde und das Ergebnis wird als csv-Datei gespeichert, damit es im Frontend performant abgerufen werden kann.

Die Hauptfunktionalitäten der Klasse sind neben der Modellerstellung und Prediction das Durchführen eines Test-Runs. Hierbei werden nicht alle vorhandenen Daten zum Trainieren verwendet, sondern die letzten 60 Datenpunkte dienen dem Testen. Beim Durchführen eines Test-Runs werden darüber hinaus mehrere Diagramme erstellt und gespeichert, die die folgende Evaluation erleichtern. Neben dem berechneten Root-Mean-Square-Error Root-Mean-Square-Error (RSME), der die Abweichung der Vorhersage unseres Modells vom tatsächlichen Ergebnis misst, dient ein erstellter Naiver-Klassifikator als Vergleichsmöglichkeit. Der Naive-Ansatz sagt immer den Beta-Wert vorher, der zuletzt bekannt war. Dieser weitere Indikator ist sinnvoll, da ansonsten eine Genauigkeit von 90% oder eine Abweichung von 1 wenig aussagekräftig ist und stark von dem vorhandenen Anwendungsfall abhängt.

Es wird also ein spezialisiertes Modell für jede Aktie erstellt, wodurch sowohl die Genauigkeit erhöht wird, als auch der initiale Berechnungsaufwand. Des Weiteren bietet die Klasse umfangreiche Testmöglichkeiten, die im folgenden Kapitel „Evaluation“ verwendet werden.

Beschreibung des LSTM-Algorithmus

Für den Algorithmus nutzen wir das LSTM, ausgesprochen "Long Short-Term Memory", was auf deutsch "langes Kurzzeitgedächtnis" bedeutet (Rosenkranz, 20.02.2020). Es ist ein Verfahren, um die Leistungsfähigkeit der neuronalen Netze zu verbessern. Das neuronale Netz muss nicht mehr für jede Aufgabe von vorne beginnen, sondern greift auf bekannte Strukturen und erworbenen Fähigkeiten zurück.

Das LSTM-Verfahren verwendet zur besseren Erinnerung drei Torsorten. Ein Eingangstor (Input Gate), ein Merk und Vergesstor (Forget Gate) und ein Ausgangstor (Output Gate). Dadurch ermöglicht es eine Art Erinnerung an frühere Erfahrungen. Außerdem kann es aus den Trainingsdaten lernen, welche Informationen aus der Vergangenheit für den Output wichtig sind und welche Vergessen werden können.

Bisherige Einsatzgebiete sind beispielsweise die Spracherkennung auf Smartphones, smarte Assistenten und das intelligente Übersetzungsprogramm Google Translate (Luber, 12.11.2018).

Vorteile vom LSTM sind, dass nur die Hälfte an Deep Neuronal Network (DNN) benutzt wird. Auch wird die Netzwerkgröße nicht erhöht, trotz langem Betrachten der Eingabesequenzen („Ein Vergleich von DNN, CNN und LSTM mit TF / Keras“, 25.09.2020).

Allerdings werden mehr Parameter als bei anderen Methoden, wie z.B. beim Convolutional Neuronal Network (CNN) benötigt. Auch das Trainieren dauert länger („Ein Vergleich von DNN, CNN und LSTM mit TF / Keras“, 25.09.2020).

In unserem Ansatz nutzen wir LSTM für die Vorhersage von Aktienkursen, weil es spezialisiert auf das Erlernen von Langzeitabhängigkeiten ist. Wir haben ein LSTM Netzwerk entworfen bestehend aus einem Input-Layer, drei Hidden-Layer, mit zwei LSTM-Layer und einem Dense-Layer und einem Output-Layer, der eine Vorhersage macht.

Evaluation

Während im ersten Ansatz vor allem die Genauigkeit Maßstab der Evaluierung war, steht beim zweiten Ansatz aufgrund der kontinuierlichen Zielvariable vor allem der RSME und die graphische Interpretation im Fokus. Insgesamt werden bei jedem Modell fünf Prozent der Daten zum Testen verwendet, also je nach Unternehmen ca. 500 Datenpunkte.

Einen guten Überblick über die Daten und die Genauigkeit liefert die Grafik „Train-Test-Prediction quarterly forecast“. Es wird deutlich, dass die Predictions des Modells (grün) die Validierungsdaten (orange) meist sehr gut nachbilden. Dennoch werden vor allem starke Schwankungen durch das Modell nicht vorhergesagt.

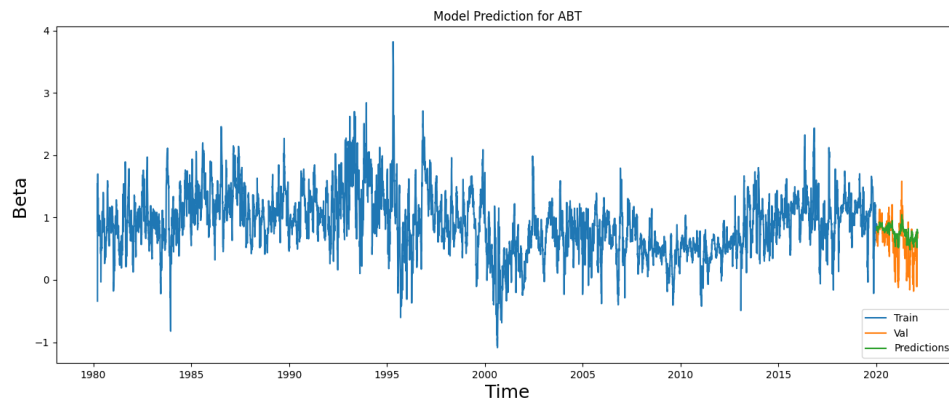


Abbildung 4.8: Train-Test-Prediction quarterly forecast
Quelle: Eigene Darstellung

Obwohl die Daten gut gefittet werden, bleibt offen, wie gut das Modell im Vergleich ist. Dafür wird der in der Modellerstellung erwähnte Naive-Ansatz als Vergleichsmodell verwendet. Beide Vorhersagen werden im Vergleich zu den Testdaten in einem Diagramm dargestellt. Hier wird erkenntlich, dass das LSTM-Modell die Testdaten deutlich genauer nachbildet als der Naive-Regressor.

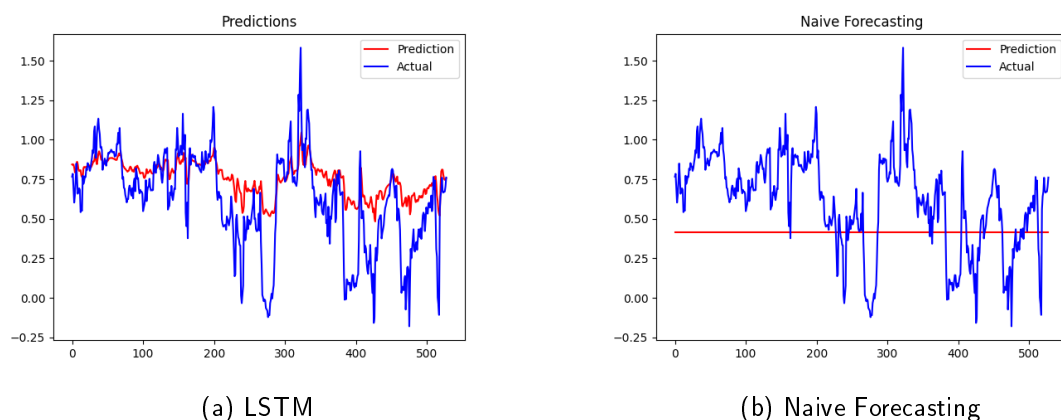


Abbildung 4.9: Test vs. Regression
Quelle: Eigene Darstellung

Das wird auch bei Betrachtung des RSME deutlich, der bei dem Naiven-Modell 0.78 beträgt, während dieser beim LSTM-Modell durchschnittlich bei 0.413 liegt. Für beide wurde die durchschnittliche Wurzel des quadratischen Fehlers für 10 Modelle gewählt, was mit kumuliert 5000 Testdaten aussagekräftig erscheint. Bei 9 dieser 10 Modelle schneidet das trainierte LSTM-Modell hinsichtlich des RMSE besser ab als das Naive.

Eine weitere Variation die erheblichen Einfluss auf die Genauigkeit hat, ist die Wahl des Vorhersagezeitraums. Es können deutlich bessere Ergebnisse erzielt werden mit dem LSTM-Modell, wenn der Vorhersagezeitraum anstelle von ca. drei Monaten nur noch einen Tag beträgt. Andererseits führt ein längerer Zeitraum wie ein Jahr zu einem ungenaueren Modell. Hierbei werden die Schwankungen nicht mehr nachgebildet, sondern das Modell nähert sich einer linearen Regression an (s. Abbildung 4.11).

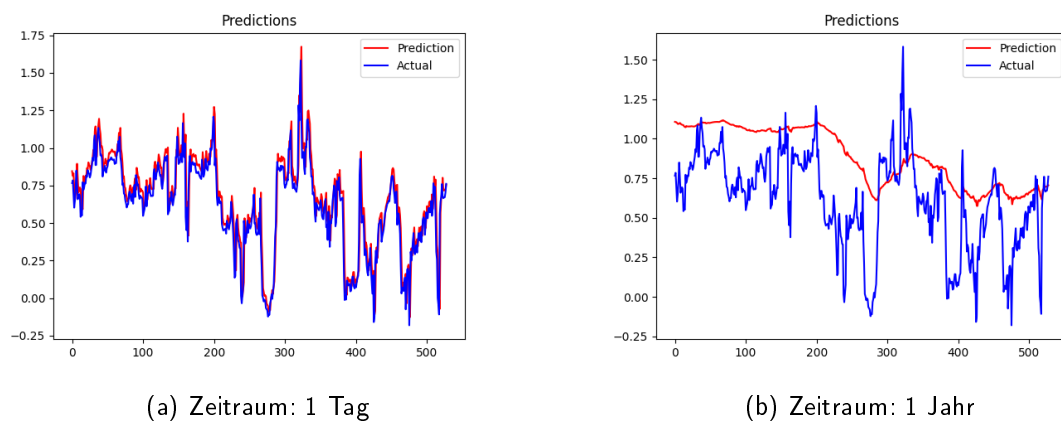


Abbildung 4.10: Validierung LSTM mit unterschiedlichen Vorhersagezeiträumen
Quelle: Eigene Darstellung

Diese Eigenschaft wird auch im RMSE erkenntlich, der im Fall von unserem Testbeispiel Abbott Laboratories (ABT) für die Vorhersage von einem Tag 0.12, bei drei Monaten 0.2, beziehungsweise 0.37 bei einem Jahr beträgt.

Da für unseren Anwendungsfall, die Vorhersage des Risikos einer Investition für Privatanleger, sowohl eine hohe Genauigkeit, als auch ein großer Betrachtungszeitraum wünschenswert ist, deckt die quartalsweise Vorhersage beide Punkte am besten ab als „Kompromiss“.

Mit dem zweiten Ansatz gelingt es also mit ausreichender Güte das Beta für einen bestimmten zukünftigen Zeitraum vorherzusagen. Je kürzer der gewählte Zeitraum ist, desto

höher ist die Genauigkeit des Modells. Weitere Modellverbesserungen wie das Erhöhen der Epochenanzahl des Trainings haben keine nennenswerte Verbesserung gebracht.

Der Nachteil an diesem Ansatz ist jedoch die lange Berechnungsdauer für das initiale Training, ausgelöst durch die fehlende Verallgemeinerbarkeit eines spezifischen Modells auf alle Aktien.

4.3.3 Resümee des zweiten Ansatzes

Abschließend lässt sich über das Projekt sagen: Die Vorhersage von Aktienkurse mit Hilfe der LSTM Methode hat eine gute Funktionalität, allerdings wurde sie in unseren Projekt nur auf die Apple Aktie angewendet, sodass es im Moment noch kein allgemeingültiges Modell ist. Ein großer Nachteil ist die hohe Berechnungszeit. Sie ist darauf zurück zu führen, dass die Berechnungsdauer von LSTM an sich schon lange ist und im Moment für jede Aktie ein eigenes Modell aufgrund ihrer historischen Daten berechnet wird. Vermehrtes Testen und Modellverbesserung ist möglich, da bisher nur wenig Bereinigung der Daten (beispielsweise durch Entfernen von Ausreißern) und Hyperparametertuning erfolgt ist. Hyperparameter sind die Stellschrauben, die über den Trainingsprozess hinweg gleich bleiben und großen Einfluss auf die Genauigkeit des Modells haben (TWT Interactive GmbH, n. d.). Das Finden der optimalen Einstellung, dem Hyperparameterturning, kann die Vorhersagegenauigkeit erheblich verbessern.

Ausblick:

Um das Produkt weiter zu perfektionieren und dauerhaft in "Produktion" umzusetzen, könnten folgende weitere Maßnahmen durchgeführt werden:

- die Daten- und Modellaktualisierungen können automatisiert werden. (Im Moment werden die Skripte noch manuell ausgeführt).
- Denkbar wäre ein tägliches Update z.B. mit Jenkins.
- das Modell kann auch noch durch Bereinigungen von Ausreißern verbessert werden.
- Die Performance kann durch inkrementelles Lernen auf neuen Daten verbessert werden.

5 Fazit

Abschließend lässt sich das Projekt so zusammenfassen, dass wir mit Hilfe der StockTimes App Kunden, je nach individueller Risikoneigung, passende Aktien empfehlen. Dabei besteht unsere App aus drei Hauptbestandteilen. Zum ersten, dem Frontend, dass der Nutzer auf seinem Bildschirm angezeigt bekommt, zum zweiten, der Sentiment Analyse, durch die die aktuelle Marktstimmung ermittelt wird, und zum dritten, das Machine Learning, durch das die Aktien den jeweiligen Risikoklassen zugewiesen werden. Insgesamt funktioniert das Zusammenspiel der drei Bestandteile sehr gut. Doch wie bei jeder Fallstudie gibt es in den drei Bereichen Möglichkeiten, um StockTimes in der Zukunft weiter zu verbessern. Für das Frontend könnte man beispielsweise statt der Desktop App eine Website schreiben, über die man StockTimes nutzen kann. Zur Verbesserung der Sentiment Analyse könnte man in Zukunft verfeinerte Suchanfragen an die Twitter API geben. Dadurch würde sich die Datenbasis qualitativ verbessern und die Marktstimmung würde deutlicher herausgefiltert werden können. Um das Machine Learning zu verbessern wäre eine Möglichkeit, die Datenabfrage zu automatisieren und da durch immer mit den aktuellsten Daten und einer stetig wachsenden Datenmenge zu arbeiten. Wenn man noch die Ausreise aus dem Datensatz entfernt würde die Performance des Modells sich verbessern. Das Aufteilen der Fallstudie in drei Bereiche (Frontend, Backend, Sentiment Analyse) hat sich als richtige Methode zum bearbeiten der Fallstudie herausgestellt. Durch das Aufteilen konnte jeder an dem Bereich, der ihn interessiert arbeiten und die Teile konnten parallel erarbeitet werden, wodurch einiges an Wartezeit gespart werden konnte. Außerdem funktionieren alle drei Teile unabhängig voneinander, so dass man sie auch für andere Projekte wiederverwenden kann. Auch die gewählte Scrum Methode war ideal für das Bearbeiten der Fallstudie und hat zu ihrem Erfolg beigetragen. Um nochmal die ganze Fallstudie zusammenzufassen, lässt sich sagen, dass alles gut funktioniert hat und am Ende eine App entwickelt wurde, welche dem Kunden Aktien auf Basis der aktuellen Marktstimmung und seiner individuellen Risikoneigung vorschlägt, und damit die Anforderungen der Aufgabenstellung ideal erfüllt.

6 Anhang

6.1 Entscheidungstabelle

		1	2	3	4
Bedingungen:	Risikovorliebe konfiguriert	J	J	J	N
	Risikovorliebe im erlaubten Bereich	J	J	N	-
	Passende Aktie für die Vorliebe existiert	J	N	-	-
Aktionen:	Ausgaben des Aktienvorschlags	X	-	-	-
	Fehlermeldung: ungültige Eingabe	-	-	X	-
	Fehlermeldung: keine passende Aktie für Vorliebe	-	X	-	-

Abbildung 6.1: Entscheidungstabelle
Quelle: Eigene Abbildung

6.2 Use-Case-Schablone

- 1) Use Case: Aktienvorschlag erhalten (UC #001)
- 2) Ziel: Kunde erhält nach Risikoeinstellung einen passenden Aktienvorschlag
- 3) Vorbedingung: Kunde hat ein Depot bei der Bank und ist autorisiert, Kunde ist auf der Website/ hat das Programm gestartet
- 4) Nachbedingung Erfolg: Kunde erhält personalisierten Aktienvorschlag
- 5) Nachbedingung Fehlschlag: Mitteilung an Kunden, welcher Fehler vorliegt
- 6) Akteure: Kunde, (ML-Algorithmus, Datenquellen)
- 7) Auslösendes Ereignis: Kunde möchte mit Risikoeinstellung Aktienvorschlag erhalten
- 8) Beschreibung:
 1. Kunde stellt Risiko ein und bestätigt dies
 2. Risikovorliebe im erlaubten Bereich
 3. Ermittlung eines passenden Aktienvorschlags
 4. Ausgabe des Aktienvorschlags
 Erweiterungen:
 - 4a. Kunde will vorgeschlagene Aktie kaufen (UC #002)
 Alternative:
 - 2a. Fehlermeldung ungültige Eingabe
 - 3a. Fehlermeldung: keine passende Aktie für den Kunden konnte ermittelt werden, Ausgabe der Aktien, die am ehesten den Kundenwünschen entsprechen

Abbildung 6.2: Use-Case-Schablone
Quelle: Eigene Abbildung

Literatur

- Aditya Mishra. (2020). Metrics to Evaluate your Machine Learning Algorithm. <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>
- Antonia Mannweiler. (n. d.). Wer ist die Generation Neobroker? Verfügbar 13. Februar 2022 unter <https://www.faz.net/aktuell/finanzen/diw-studie-zu-neobroker-wer-nutzt-trade-republic-17792143.html>
- Asana. (16.02.2022). SWOT-Analyse: Definition, Beispiele, Vorteile und Anwendungstipps • Asana. Verfügbar 16. Februar 2022 unter <https://asana.com/de/resources/swot-analysis>
- Baijayanta Roy. (n. d.). All about Feature Scaling. 2020. <https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35#:~:text=Feature%20scaling%20is%20essential%20for,that%20calculate%20distances%20between%20data.&text=Therefore%20the%20range%20of%20all,proportionately%20to%20the%20final%20distance.>
- Betafaktor_Radinger_Heinze_2011. (n. d.). Verfügbar 20. Februar 2022 unter https://www.controllerakademie.de/wp-content/uploads/2016/08/Betafaktor_Radinger_Heinze_2011.pdf
- Borchers, D. (n. d.). *Elektronische Unterschrift: Bundesregierung reformiert Schriftformerfordernis*. Verfügbar 1. Januar 2016 unter <http://m.heise.de/newsticker/meldung/Elektronische-Unterschrift-Bundesregierung-reformiert-Schriftformerfordernis-3259140.html>
- DeltaValue.de. (08/02/2022). *Betafaktor von Aktien - Erklärung & Berechnung*. Verfügbar 20. Februar 2022 unter <https://www.deltavalue.de/betafaktor/>
- DIN 69901-2: Prozesse, Prozessmodell. (n. d.). *Deutsches Institut für Normung*.
- Ein Vergleich von DNN, CNN und LSTM mit TF / Keras. (25.09.2020). Verfügbar 16. Februar 2022 unter <https://ichi.pro/de/ein-vergleich-von-dnn-cnn-und-lstm-mit-tf-keras-36910827797438>
- Geierhos, M. (n. d.). *Sentimentanalyse — Enzyklopaedie der Wirtschaftsinformatik*.
- Jason Brownlee. (2020). What is a Confusion Matrix in Machine Learning. <https://machinelearningmastery.com/confusion-matrix-machine-learning/>
- Luber, S. (12.11.2018). Was ist ein Long Short-Term Memory? *BigData-Insider*. Verfügbar 16. Februar 2022 unter <https://www.bigdata-insider.de/was-ist-ein-long-short-term-memory-a-774848/>
- Musterfrau, E., & Exempelmann, M. (2012). *Zitationen am Beispiel* (5. Aufl.). Wissenschaftstestverlag.
- N. Andruskiewitsch & J. Devoto. (1996). Extensions of Hopf algebras. *St. Petersburg Math. J.*, 7(1), 17–52.
- NN. (n. d.). *Mögliche Stile: Numerisch, Alphabetisch, Harvard, Autor-Jahr*.

- Rosenkranz, J. (20.02.2020). LSTM. *Onlinesprache*. Verfügbar 16. Februar 2022 unter <https://www.onlinesprache.de/begriffe/lstm/>
- Testmann, H., Demofrau, M., & Checker, E. (2015). Das Testen von Artikeln. *Int. Journal of Testing*, 5(42), 111–222.
- TWT Interactive GmbH. (n.d.). Hyperparameter Tuning: Der Feinschliff eines Machine-Learning-Modells - News - TWT Interactive. <https://www.twt.de/news/detail/hyperparameter-tuning-der-feinschliff-eines-machine-learning-modells.html>
- Was ist Sentiment-Analyse?* (2022).