

Die Bestimmung optimaler Kauf- und Verkaufspreise von Basiswerten zur Wahrung der täglichen Delta-Neutralität beim Handel von Aktioptionen mit Hilfe verschiedener Machine Learning Methoden

Fabian Gehring, Zürcher Hochschule für Angewandte Wissenschaften

Inhaltsverzeichnis

Management Summary	2
1 Einleitung	2
1.1 Motivation	2
1.2 Forschungsfrage	4
1.3 Daten	6
1.4 Methoden	7
1.5 Disclaimer	7
2 Daten	7
2.1 Bezug und Umfang	7
2.2 Aufbereitung	11
3 Infrastruktur und Tools	16
3.1 Cloud Setup	16
3.2 Verwendete Software	18
4 Analyse	20
4.1 Einfache Optimierungen	21
4.2 Nearest Neighbor Ansätze	24
4.3 Neuronale Netzwerke	30
5 Fazit und Ausblick	37

Management Summary

In vorliegender Arbeit wird untersucht, ob mit Hilfe maschineller Lernalgorithmen Kaufs- und Verkaufsschranken von Aktien so prognostiziert werden können, dass der Ausgleich von aufgelaufenem Delta bei Optionspositionen mit höheren Payoff erfolgen kann als mit einfacheren Strategien. Es gilt dabei die Nebenbedingung, dass die Position zum Tagesende bezüglich Preisschwankungen abgesichert sein muss. Als Referenz dient dabei die Strategie, bei welcher untertags kein Kauf- und Verkauf von Aktien erfolgt, sondern der Ausgleich des Deltas komplett zum Tagesschlusskurs erfolgt. Als Datenbasis dienen die historischen Eröffnungs-, Tiefst-, Höchst- und Schlusskurse von rund 1'600 grossen kotierten Unternehmen weltweit. Für die Preisprognosen werden drei verschiedene Ansätze verfolgt und verglichen.

In einem ersten Ansatz werden die Kaufs- und Verkaufsschranken als symmetrische Auslenkungen vom jeweiligen Eröffnungskurs gesetzt. Der Payoff der Referenzstrategie lässt sich so um rund 6% übertreffen. Es zeigt sich dabei, dass die Volatilität der vergangenen 10 Tage einen erklärenden Einfluss auf die Höhe der optimalen Auslenkung hat. Die Überperformance lässt durch Aufteilung der Daten in Tief- und Hochvolatilitätsumfeld auf rund 8% steigern.

Der zweite Ansatz basierend auf der Suche vergangener ähnlicher Kursverläufe mittels K-Nearest-Neighbor Verfahren. Die Überperformance dieses Ansatzes liegt bei rund 8%. Die Analyse zeigt ferner, dass die Performance mit steigender Anzahl Vergleichskurse über die Zeit zunimmt. Stehend genügen Vergleichskurse zur Verfügung kann eine Sättigung des Lerneffekts beobachtet werden.

Im dritten Ansatz werden die Kaufs- und Verkaufsschranken mittels neuronaler Netze vorhergesagt. Als einzige der untersuchten Methoden wird mit diesem Verfahren auch die Verteilungen der prognostizierten Preise mit einbezogen. Es resultieren Überschussrenditen bis zu 15% im Vergleich zur Referenzstrategie. Dabei zeigt sich, dass mit der Modellierung ordinaler Preisklassen als auch mit der Modellierung voneinander abhängiger Innertagspreise bessere Ergebnisse erzielt werden können als bei der Modellierung voneinander unabhängiger Preise.

1 Einleitung

1.1 Motivation

Aktienoptionen sind derivative Finanzinstrumente, welche dem Inhaber das Recht (nicht aber die Pflicht) geben, die zugrunde liegende Aktie (auch Basiswert oder Underlying) zu einem im Voraus festgelegten Ausübungspreis (auch Strike) zu kaufen oder zu verkaufen. Wie sich der Preis einer Option entwickelt, ist dabei von verschiedenen Faktoren abhängig. Dazu gehören unter anderem der Preis des Basiswertes, dessen Volatilität oder die Restlaufzeit der Option.

Der Zusammenhang zwischen Preis des Underlyings und Preis der Option ist nicht linear, sondern nimmt bei steigendem Preis zu (Kaufoption), resp. ab (Verkaufsoption). Diese nicht-Linearität wird mit Hilfe der Kennzahl Gamma (Γ) beschrieben. Im professionellen Optionenhandel ist es üblich, den linearen Teil der Veränderung (Delta, auch Δ) durch Kauf oder Verkauf des Underlyings abzusichern. Damit lassen sich Strategien verfolgen, welche von der Richtung der Preisbewegung unabhängig sind (Delta-Neutral Trading). Der Nicht-lineare Zusammenhang führt allerdings dazu, dass die Delta-Neutralität mit sich ändernden Preisen wieder verloren geht. Dies sei nachfolgend an einem Beispiel demonstriert:

Sei dazu angenommen, dass eine Aktie zum Preis von 10 gehandelt wird. Die Aktie dient als Basiswert sowohl für eine Kaufoption (Call) wie auch eine Verkaufsoption (Put). Die Laufzeit beträgt 6 Monate, der Strike ebenfalls 10. Mit dem Kauf der Option erwirbt der Besitzer damit das Recht, die Aktie in 6 Monaten zu 10 zu kaufen (im Falle des Calls) oder zu verkaufen (im Falle des Puts). In diesem Beispiel entspricht der Aktienkurs dem Strike. Solche Optionen bezeichnet man als “at-the-money (atm)”. Bei Veränderung des Basispreises werden diese zu “in-the-money (itm)” oder “out-of-the-money (otm)” Optionen.

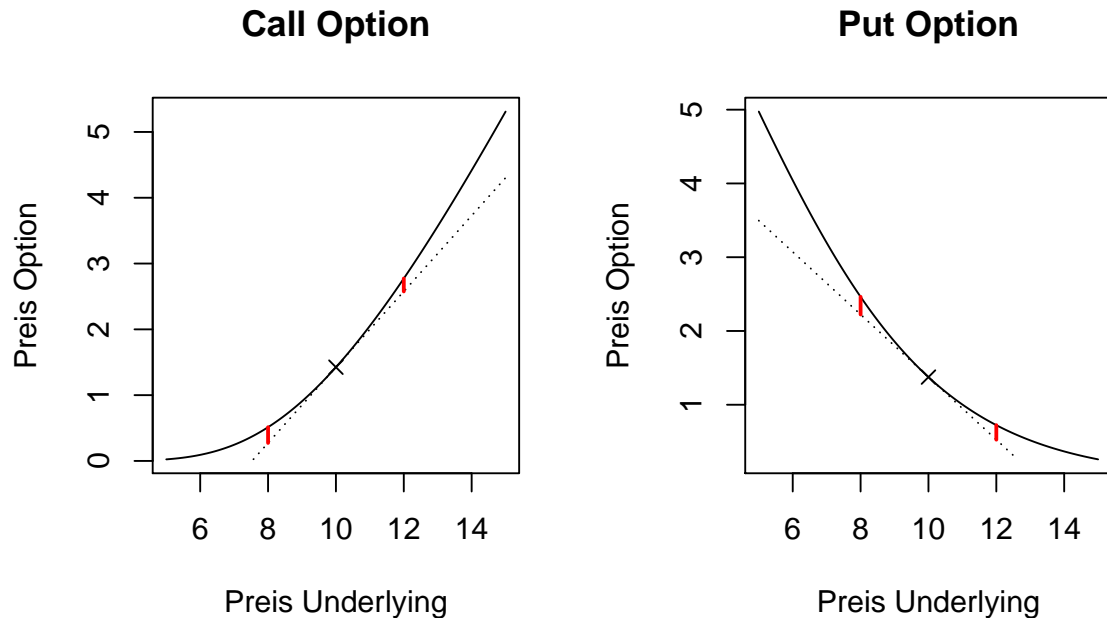


Abbildung 1: Zusammenhang zwischen Preis der Underlyings und Preis der Option.

Den Zusammenhang zwischen Preis der Option und Preis des Underlyings ist in Abbildung 1 abgebildet. Die gestrichelten Linien bilden die Tangenten beim Strikepreis der Optionen. Deren Steigungen (Δ) betragen 1.42 im Falle des Calls, und -1.37 im Falle des Puts. Dies bedeutet, dass sich der Besitzer der Option sich gegen Preisbewegungen absichern (hedgen) kann, indem er zusätzlich zum Besitz der Option diese Anzahl an Aktien

verkauft (Call), resp. kauft (Put). Wertänderungen der Option werden dann durch die entgegenschüssige Wertentwicklung in der Aktienposition ausgeglichen. Eine solche gegenüber Preisänderungen nicht sensitive Position wird “deltaneutral” genannt. Bei grösseren Preisbewegungen führt die Konvexität des Optionspreises allerdings dazu, dass es trotz Absicherung zu einer Abweichung kommt. Diese “Fehler” sind in Abbildung 1 mit roten Linien dargestellt. Es mag auf den ersten Blick erstaunen, dass diese Abweichung immer zugunsten des Optionsbesitzers ausfällt (sowohl für Call- als auch Put-Option verlaufen die Tangenten unterhalb des Optionspreises). Dass es sich dabei aber nicht eine Geldmaschine handelt liegt daran, dass die beschriebenen Preisbewegungen erst im Laufe der Zeit erfolgen und die Restlaufzeit der Option kleiner wird (Theta, auch Θ). Mit sinkender Restlaufzeit nimmt auch der Wert der Option ab. In effizienten Märkten gleichen sich diese Effekte im Durchschnitt aus.

Wenn die zukünftigen Bewegungen des Preises grösser sind, als dies durch den Markt mit dem Wertverlust aufgrund der Zeit einpreist, kann die Verfolgung einer solchen Handelsstrategie allerdings lukrativ sein. Die Schwierigkeit liegt hier natürlich bei der Auswahl der richtigen Titel. Diese Strategie wird als Gamma-Scalping bezeichnet.

Der Einfluss des Gammas ist aber nicht nur zu Spekulationszwecken interessant. Dessen Bewirtschaftung ist auch für die Steuerung des eingegangenen Risikos von Relevanz. Soll nämlich die Delta-Neutralität nach Preisbewegungen wieder hergestellt werden, müssen laufend neue Titel des Underlyings gekauft oder verkauft werden. Grafisch entspricht dies der Einnahme einer Aktienposition, welche wieder der Steigung der Tangente beim nun vorherrschenden Preis entspricht. Typischerweise ist es so, dass diese Anpassungen abends (zur Reduktion des Risikos über Nacht) oder je nach Entscheidung des Händlers auch untertags erfolgen. Der Händler wird dabei versuchen, den Ausgleich nach möglichst grosser Preisbewegung zu machen, d.h. negatives Delta am Tiefstpunkt zu kaufen resp. positives Delta am Höchstpunkt zu verkaufen. Verpasst er diesen Zeitpunkt und die Preise entwickeln sich wieder Richtung Ausgangspunkt, so entgehen ihm mögliche Gewinne.

1.2 Forschungsfrage

Für die vorliegende Arbeit ergibt sich damit folgende Forschungsfrage:

Kann mit Hilfe datengestützter Methoden eine Strategie gefunden werden, welche einen Optionshändler unterstützt, die idealen (intraday) Preise zum Kauf / Verkauf von aufgelaufenem Delta zu finden?

Als Vergleich bietet sich dabei eine Strategie an, welche keine intraday Anpassungen vornimmt, sondern die aufgelaufene Delta-Position erst am Abend ausgleicht. Die Voraussage jeweils eines Verkaufs- und eines Kaufpreises soll dabei bei Markteröffnung des jeweiligen Tages erfolgen. Bei Tagesende noch vorhandenes Delta (aufgrund nicht erreichter Kaufs- oder Verkaufspreise oder neu aufgelaufenes Delta bis Tagesende) wird immer zum Tagesschlusskurs ausgeglichen.

Mathematisch lassen sich beide Strategien mit Hilfe einer Payoff-Funktion ausdrücken. Für die einfachere Referenzstrategie, welche die Delta-Neutralität erst bei Tagesende wieder herstellt, lautet diese wie folgt:

$$Payoff = \frac{1}{2} \times 100 \times \$\Gamma_{close_{t-1}} \times \ln \left(\frac{close_t}{close_{t-1}} \right)^2 \quad (1)$$

Für die Notation werden log>Returns verwendet. Diskrete Returns sind aber genauso möglich. Als $close_i$ wird der Tagesendkurs am Tag i bezeichnet. Unter $\$ \Gamma_i$ verstehen wir dabei das Gamma Cash (auch Dollar Cash). Die Herleitung dieses Payoffs ergibt sich aufgrund folgender Überlegungen (vgl. SP-Finance, 2020):

Bezeichne S_0 den Preis des Underlyings und P_0 den Preis der Option zu Beginn und S_1 sowie P_1 die jeweiligen Preise nach einer Preisbewegung des Underlyings um dS . Δ_0 und Γ_0 bezeichnen die erste und zweite Ableitung des Optionspreises nach dem Underlyingpreis zum Ausgangszeitpunkt. Ferner sei als Delta Cash ($\$ \Delta = \Delta \times S$) der Wert des Hedge Portfolio bezeichnet und Gamma Cash ($\$ \Gamma = \Gamma \times S^2/100$) bezeichne die Veränderung des $\$ \Delta$ bei einer 1-prozentigen Veränderung des Underlying-Preises. Dies lässt sich wie folgt herleiten: Verändert sich der Underlying-Preis um 1% $S \Rightarrow S + S/100$, verändert sich auch das Delta $\Delta \Rightarrow \Delta + \Gamma \times S/100$. Das neue Delta Cash verändert sich damit in der Höhe von Gamma Cash $\$ \Delta \Rightarrow \$ \Delta + \Gamma \times S^2/100$.

Mit Hilfe dieser Notation lässt sich der Gewinn einer zu Beginn delta-gehedgten Position herleiten:

1. Veränderung des Underlying Preises $S_1 = S_0 + dS$
2. Veränderung des Deltas $\Delta_1 = \Delta_0 + \Gamma_0 \times dS$
3. Durchschnittliches Deltas $\Delta_{Avg} = (\Delta_0 + \Delta_1)/2 = \Delta_0 + \Gamma_0 \times dS/2$
4. Neuer Optionspreis $P_1 = P_0 + \Delta_{Avg} \times dS = P_0 + \Delta_0 \times dS + \Gamma_0 \times dS^2/2$
5. Gewinn als Summe der Veränderungen des Optionspreis und des Hedge-Portfolios:
 $\Delta_0 \times dS + \Gamma_0 \times dS^2/2 - \Delta_0 \times dS = \Gamma_0 \times dS^2/2$
6. Etwas umgeformt lässt sich der Gewinn schreiben als: $\Gamma_0 \times dS^2/2 = \Gamma_0 \times S^2/100 \times dS^2/S^2 \times 100 = \$ \Gamma \times (dS/S)^2 \times 100$

Auch im Falle der etwas komplexeren Strategie mit der Möglichkeit von Käufen und Verkäufen innerhalb des Tages lässt sich die gleiche Payoffstruktur wiederfinden:

$$Payoff = \frac{1}{2} \times 100 \times \$\Gamma_{close_{t-1}} \times \ln \left(\frac{first}{close_{t-1}} \right)^2 \times I_{first} \times (1 - I_{second}) + \quad (2)$$

$$\frac{1}{2} \times 100 \times \$\Gamma_{first} \times \ln \left(\frac{close_t}{first} \right)^2 \times I_{first} \times (1 - I_{second}) + \quad (3)$$

$$\frac{1}{2} \times 100 \times \$\Gamma_{close_{t-1}} \times \ln \left(\frac{second}{close_{t-1}} \right)^2 \times I_{second} \times (1 - I_{first}) + \quad (4)$$

$$\frac{1}{2} \times 100 \times \$\Gamma_{second} \times \ln \left(\frac{close_t}{second} \right)^2 \times I_{second} \times (1 - I_{first}) + \quad (5)$$

$$\frac{1}{2} \times 100 \times \$\Gamma_{close_{t-1}} \times \ln \left(\frac{first}{close_{t-1}} \right)^2 \times I_{second} \times I_{first} + \quad (6)$$

$$\frac{1}{2} \times 100 \times \$\Gamma_{first} \times \ln \left(\frac{second}{first} \right)^2 \times I_{first} \times I_{second} + \quad (7)$$

$$\frac{1}{2} \times 100 \times \$\Gamma_{second} \times \ln \left(\frac{close_t}{second} \right)^2 \times I_{second} \times I_{first} + \quad (8)$$

$$\frac{1}{2} \times 100 \times \$\Gamma_{close_{t-1}} \times \ln \left(\frac{close_t}{close_{t-1}} \right)^2 \times (1 - I_{first}) \times (1 - I_{second}) \quad (9)$$

wobei,

$$\$ \Gamma_i = \Gamma \times S_i^2 / 100 \quad (10)$$

Zu beachten gilt es insbesondere, dass der Payoff quadratisch auf Preisbewegungen reagiert. Dies führt dazu, dass es von Relevanz ist, ob zuerst der Kaufs- (untere Grenze) oder der Verkaufsschwelle erreicht wird. Die Ausführung zum jeweiligen Schwellenwert wird durch die Indikatorvariablen I_{first} und I_{second} modelliert. Damit es zu einer Ausführung kommt müssen dabei folgende beiden Kriterien kumulativ erfüllt sein:

- a) Der Schwellenwert wird erreicht.
- b) Die vorangegangene Transaktion hat bei einem höheren Preis für Aktion an unterer Schwelle, resp. tieferen Preis für eine Aktion an der oberen Schwelle stattgefunden.

1.3 Daten

Als Datengrundlage stehen historische Open-, Low-, High- und Close- (adjustiert wie unadjustiert) Aktienpreise zur Verfügung. Diese sind öffentlich und können von Yahoo Finance bezogen werden. So kommen 9.5 Mio. historische Datenpunkte für die grössten rund 1600 Unternehmen weltweit zusammen. Die Datenqualität scheint grundsätzlich ok. Gewisse Datenbereinigungsschritte sind aber nötig.

Die Verwendung öffentlicher Daten hat den Vorteil, dass diese problemlos auf Cloud-Infrastruktur (bsp. unabdingbar im Falle des Einsatzes von Neuronalen Netzen / Deep Learning Techniken) prozessiert werden können, ohne dabei Lizenzvereinbarungen berücksichtigen zu müssen.

1.4 Methoden

Es werden verschiedene Methoden versucht und gegeneinander verglichen. Diese sind:

- Einfach Optimierungen
- Nearest Neighbor Ansätze
- Neuronale Netzwerke

1.5 Disclaimer

Etliche Untersuchungen haben sich bereits damit beschäftigt, kursrelevante Informationen vorauszusagen. Viele der dabei gefundenen Erkenntnisse sind nicht oder nur unter speziellen Voraussetzungen anwendbar. Es sei deshalb an dieser Stelle erwähnt, dass sich dem sowohl der Autor als auch dessen Auftraggeber bewusst ist. Auch wenn diese ausbleiben bietet diese Arbeit sowohl für den Autoren wie auch dessen Arbeitgeber die Chance, verschiedene moderne Analysetechniken/ -methoden an Finanzdaten anzuwenden, welche auch für andere Projekte wertvoll sein können.

2 Daten

2.1 Bezug und Umfang

2.1.1 Aktienuniversum

Für die vorliegende Analyse werden die Aktienpreise grosser Unternehmen weltweit herangezogen. Konkret sind es alle Aktienkomponenten des “iShares MSCI World UCITS ETF” (vgl. Blackrock, 2020) per 28. Februar 2020.¹ Dieser Exchange Traded Fund (ETF) besteht zu diesem Zeitpunkt aus 1639 Aktien, welche sich automatisiert auslesen lassen. Ferner stellt die Webseite des ETF Emittenten weitere Attribute zur Verfügung. Diese lauten für das Beispiel Nesté wie folgt:

¹Das Startdatum dieser Arbeit.

Tabelle 1: Attribute der Nestlé Aktie per 28. Februar 2020

Attribut	Wert
Ticker	NESN
Name	NESTLE SA
Asset Class	Equity
Weight (%)	0.74265
Price	104.76
Shares	361433
Market Value	37862437
Notional Value	37862437
Sector	Consumer Staples
ISIN	CH0038863350
Exchange	Six Swiss Exchange Ag
Location	Switzerland
Market Currency	CHF

Neben eindeutigen Kennzeichner wie “Ticker” und “ISIN” enthält der Datensatz auch Informationen zur “Asset Class”. Für die vorliegende Analyse ist hierbei lediglich die Ausprägung “Equity” zulässig. Die Kennzahlen “Weight”, “Market Value” und “Notional Value” geben Auskunft über die Grösse der betrachteten Unternehmung und eignen sich auch zum Vergleich ebendieser. Zu beachten gilt, dass im Falle von Aktien der “Notional Value” dem “Market Value” entspricht und sich dieser bis auf rundungsbedingte Differenzen auch aus der Anzahl ausgegebener Titel mal Tagesendkurs (“Shares” x “Price”) ermitteln lässt.

Als weitere Unterscheidungsmerkmale sind der Hauptbörsenplatz “Exchange” (27 unterschiedliche Ausprägungen), der Sitz “Location” (23 Ausprägungen) sowie die Währung “Market Currency” (14 Ausprägungen) aufgeführt. Alle drei Attribute bilden stark verwandte Informationen ab. Geschlüsselt auf Kontinente ergeben sich für den Börsensitz folgende Anteile:

- Nordamerika: 44%
- Europa: 28%
- Asien: 23%
- Australien: 5%

Als letztes Attribut enthält der Datensatz Angaben zum “Sector” in welchem das jeweilige Unternehmung tätig ist. Bis auf die Kategorien “Energy” und “Other” ist jeder der 12 aufgeführten Sektoren mit mindestens 5% Anteil vorhanden. Die beiden Sektoren mit dem höchsten Anteil sind “Industrials” und “Financials”.

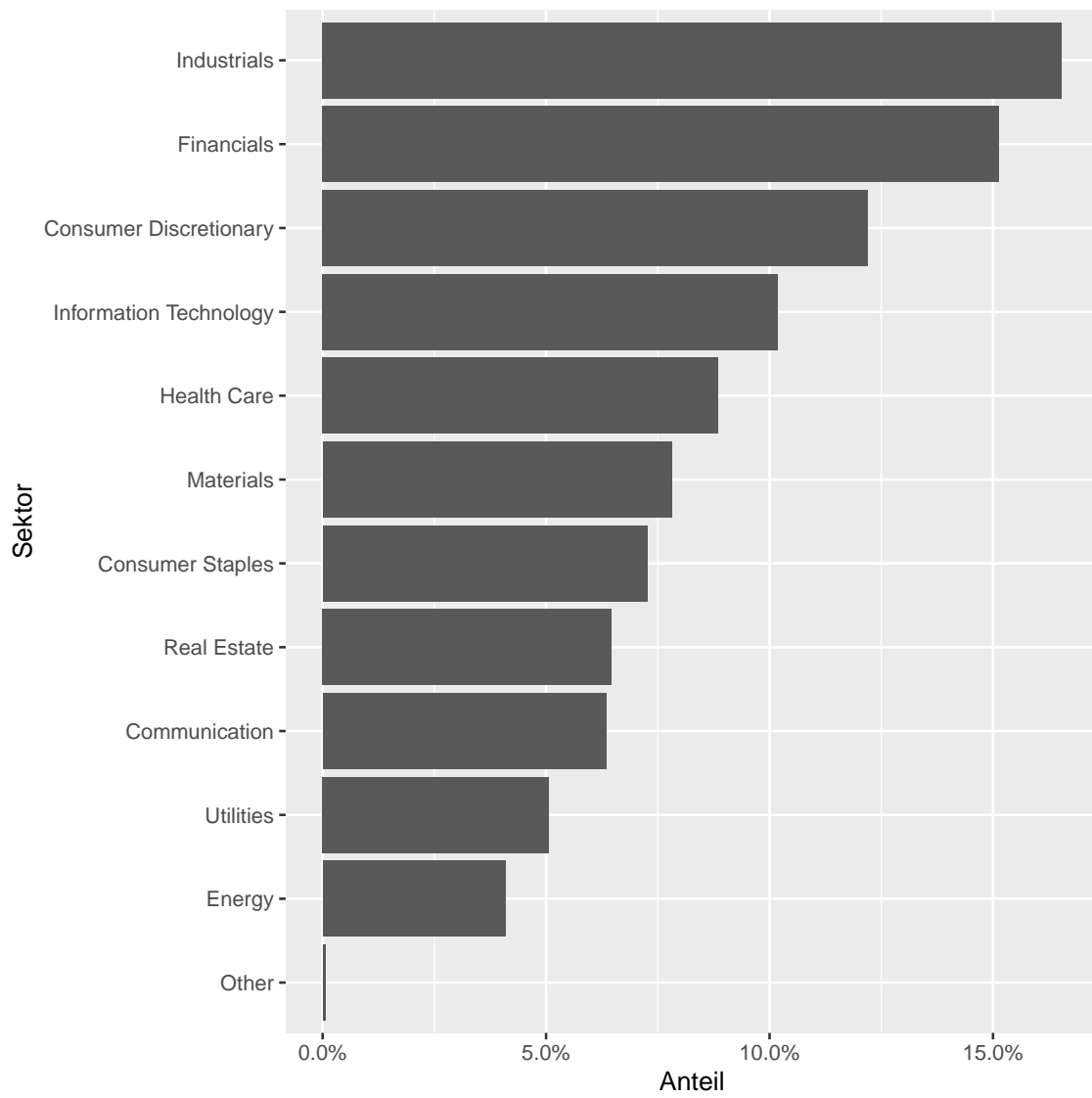


Abbildung 2: Anzahl im Datensatz vorhandene Titel je Sektor

2.1.2 Preisinformationen

Für alle Titel werden in einem zweiten Schritt die historischen Aktienkurse zum Eröffnungs-, Höchst-, Tiefst- und Endkurs des jeweiligen Tages bezogen. Diese Daten stehen via Yahoo Finance auf täglicher Basis zur freien Nutzung zur Verfügung (vgl. Yahoo, 2020). Zu beachten gilt es hierbei, dass die Yahoo Ticker für ausserhalb der USA gehandelte Titel einen Suffix je Börsenplatz verwenden. Für die Analyse kommen so 9'212'052 tägliche Datenwerte für 1'636 Titel zusammen. Für 3 Aktien können keine Werte gefunden werden. Tabelle 2 zeigt einen Beispieleintrag für die Aktie von Nestlé per 14. Februar 2019.

Tabelle 2: Kursinformationen der Nestlé Aktie per 12. April 2019

Attribut	Wert
Ticker	NESN.SW
Date	2019-04-12
Open	96.22
Low	95.11
High	96.6
Close	95.53
Adjusted	93.08
Volume	6991647

Die Werte “Open”, “Low”, “High” und “Close” enthalten die erwähnten Eröffnungs-, Tiefst-, Höchst- und Schlusskurse des Titels. Mit “Volume” werden die Anzahl gehandelter Titel am jeweiligen Tag angegeben. Unter “Adjusted” ist der um Dividendenausschüttungen korrigierte Schlusskurs aufgeführt.²

Die Werte in Tabelle 2 sind insofern speziell, als dass sie den letzten Tag vor dem Ex-Dividend Datum von Nestlé für 2019 betreffen. Das heisst, es sind die Kurse des letzten Tages, bevor die Aktie ohne die für das Jahr 2019 ausgeschüttete Dividende gehandelt wurde. Die Dividende betrug in jenem Jahr CHF 2.45 (vgl. Nestle, 2020). Diese Differenz widerspiegelt sich als Differenz des Close- und Adjusted-Preises. Da alle Werte (auch Open, Low, High und Close) am Folgetag ohne den Anspruch auf diese Dividende gehandelt werden, fallen diese typischerweise tiefer aus. Um eine Vergleichbarkeit der Renditen über die Zeit zu gewährleisten ist daher eine Anpassung der Werte mit Hilfe des Adjustment-Faktors nötig. Dieser ergibt sich als Quotient von Adjusted und Close Preis und wird auf allen Einträgen angewendet.

Weiter erwähnenswert ist, dass Yahoo den Adjusted Kurs des jeweils aktuellsten Tages - ausser eben am Tag vor Ex-Dividend - mit auf aktuellen Kurs festlegt. Im Lauf der Zeit und mit neuen Ausschüttungen verändert sich damit auch die Historie der Adjusted

²Allfällige Aktiensplits sind in allen Werten bereits berücksichtigt.

Werte. Dies lässt sich zeigen, wenn der Beispieleintrag von Nestlé per 12. April 2019 nach der nächsten Dividendenausschüttung (27. April 2020) noch einmal abgerufen wird (vgl. Nestle, 2020). Während alle Preise ausser “Adjusted” identisch ausgewiesen sind, hat sich dieser neu auf 90.72 verändert (vgl. Yahoo, 2020). Mit der nächsten Dividenden-ausschüttung (voraussichtlich im April 2021) wird sich dieser Wert dann wieder ändern. Mit Hilfe der Adjustierung ist aber gewährleistet, dass die Werte vergleichbar bleiben.

2.2 Aufbereitung

2.2.1 Bereinigung

Mit Yahoo Finance wird ein bekannter Datenanbieter gewählt. Der Blick auf einige Quantilskennzahlen der Rohdaten in Tabelle ?? zeigt aber, dass dennoch einige Datenprobleme ausgemacht werden können. So enthält der Datensatz beispielsweise offensichtlich falsche Werte (z.b. negative Preise) aber auch einige Lücken.

##	Open	Low	High	Close
## Min. :	0	Min. : 0	Min. : 0	Min. : 0
## 1st Qu.:	12	1st Qu.: 12	1st Qu.: 12	1st Qu.: 12
## Median :	36	Median : 35	Median : 36	Median : 36
## Mean :	1564	Mean : 1548	Mean : 1589	Mean : 1564
## 3rd Qu.:	157	3rd Qu.: 155	3rd Qu.: 158	3rd Qu.: 157
## Max. :	3750000	Max. : 3600000	Max. : 9046660	Max. : 3656250
## NA's :	135267	NA's : 135267	NA's : 135267	NA's : 135267
## Adjusted				
## Min. :	-1.181e+20			
## 1st Qu.:	7.000e+00			
## Median :	2.600e+01			
## Mean :	7.285e+17			
## 3rd Qu.:	1.220e+02			
## Max. :	5.487e+22			
## NA's :	135267			

Die Behebung dieser Mängel erfolgt in verschiedenen Schritten. Allen gemeinsam ist, dass die Bereinigung keinen Ausschluss der Daten zur Folge hat, sondern betroffene Werte als “Nicht verfügbar, (NA)” klassifiziert werden. Diese Unterscheidung ist insbesondere bei rollierender Betrachtung eines Zeitfensters der Vergangenheit von Bedeutung.

1. Entfernen von fehlerhaften Adjustierungsdaten

Eine Eigenschaft von Aktienpreisen ist es, dass sie nicht negativ sein können. Der Datensatz weist aber vereinzelt negative Adjusted Werte auf. Dies lässt sich auch durch die Dividendenbereinigung nicht erklären. Bei diesen Einträgen scheinen daher Datenfehler vorzuliegen. Erschwerend kommt hinzu, dass sich Fehler bei der

Adjustierung nicht auf den jeweiligen Eintrag beschränken. Aufgrund der Funktionsweise der Adjustierung (vgl. 2.1.2) ist ein Vererben des Fehlers auf andere Einträge des jeweiligen Titels wahrscheinlich. Bei genauerer Betrachtung der Adjusted Werte fällt ferner auf, dass komischerweise auch einige sehr sehr kleine (im Bereich 10^{-6}) - wenn auch knapp positive Werte - gefunden werden können. Aus diesen Grund werden alle Ticker, welche in ihrer Historie einen Adjusted Preis von weniger als 0.001 aufweisen, ausgeschlossen.

2. Entfernen von Einträgen mit unerwarteter Reihenfolge

Die Werte für Open, Low, High und Close implizieren eine klare Reihenfolge. Kein anderer der Werte darf höher als das High oder kleiner als das Low sein. Ist dies der Fall, werden die entsprechenden Werte von der Analyse ausgeschlossen.

3. Erkennen und Ausschluss von Tippfehlern

Einzelne Einträge lassen sich als Tippfehler identifizieren. Der Titel "AV.L" weist per 09. August 2019 beispielsweise einen Low-Wert von 3.87 aus, währenddem alle anderen Werte des gleichen Tages wie auch der benachbarten Tage bei ca. 380 liegen. Es liegt auf der Hand, dass dieser Wert um einen Faktor 100 falsch erfasst wurde. Solchen Fehlern wird begegnet, indem alle paarweisen Verhältnisse von Open, Low, High und Close Preis kleiner als 8 sein müssen. Andernfalls erfolgt ein Ausschluss der als Tipfehler identifizierten Kennzahl.

4. Fehlende Preisbewegungen innerhalb des Tages

Der Aktienkurs eines grösseren Unternehmens bewegt sich typischerweise auch an ruhigen Börsentagen immer ein wenig. Die vorhandenen Preise liegen mit der Genauigkeit mehrerer Nachkommastellen vor. Unterscheiden sich hierbei Tagestiefst- und Tageshöchstpreis nicht, muss von einem Datenfehler ausgegangen werden. Einträge ohne Preisbewegung innerhalb des Tages werden daher von der Analyse ausgeschlossen.

5. Fehlende Kursbewegungen über nacheinanderfolgende Börsentage

Ähnlich wie bei den Preisbewegungen innerhalb des Tages verhält es sich auch bei Bewegungen über sich folgende Börsentage hinweg. Es ist zu erwarten, dass sich mindestens einer der fünf betrachteten Preise vom Vortag unterscheidet. Ist dies nicht der Fall, wird der Eintrag ausgeschlossen.

6. Aussergewöhnlich hohe Preisbewegungen

Es liegt in der Natur der Sache, dass sich Aktienkurse verändern. Grosse Kurs-sprünge sind bei Aktien sehr grosser Unternehmen wie sie in dieser Arbeit betrachtet werden aber selten. Die Chance, dass das Verhältnis zwischen adjustiertem Preis des Vortages und adjustiertem Preis des aktuellen Tages (und vice versa) um mehr als einen Faktor 2 unterscheidet, erachten wir als kleiner, als dass es sich dabei um einen Datenfehler handelt. Entsprechende Einträge werden deshalb entfernt.

7. Ausschluss von Extremwerten

Die der vorliegenden Analyse zugrundeliegende Payoff-Funktion ist abhängig von den Preisbewegungen einer Aktie innerhalb des Tages (vgl. 1.2). Speziell dabei ist,

dass die Höhe der Bewegung nicht linear, sondern quadratisch Niederschlag findet. Dies führt dazu, dass die Analyse sehr sensitiv auf Ausreisser reagiert. Hinzu kommt, dass es Ziel der Arbeit ist, Aussagen über Kursbewegungen eines “typischen” Börsentages zu machen. Eine Prognose von Werten an aussergewöhnlichen Tagen liegt ausserhalb des Geltungsbereichs der Analyse. Aus diesem Grund werden nach obigen Bereinigungen für jede der fünf Preiskennzahlen die jeweils 1% extremsten Werte nach oben wie auch unten ausgeschlossen. Der Ausschluss erfolgt aufgrund der unterschiedlichen Preislevels der Aktien auf einem täglich auf den Schlusskurs des Vortages indexierten Wert.

8. Verfügbarkeit durchgängiger Historie

Zur Prognose des zukünftigen Preisverlaufs könnte der direkt vorangegangene Verlauf von Bedeutung sein. Es werden daher alle Datensätze ausgeschlossen, für welche keine durchgängige Historie von 10 Börsentagen verfügbar ist. Zu beachten gilt es hierbei, dass nicht alle nachfolgenden Analyseansätze den ganzen Umfang dieser Historie tatsächlich auch verwenden. Um die Vergleichbarkeit der Analyseansätze zu gewährleisten, gilt dieser Ausschluss allerdings generell.

Zusammenfassend lässt sich festhalten, dass der rohe Datensatz aus 9'212'052 Einträgen besteht. Davon weisen 135'267 Zeilen vor Bereinigung einen fehlenden Wert auf. Nach Bereinigung erhöht sich dieser Wert auf 3'639'965. Für die Analyse bleiben somit 5'572'087 verwendbare Einträge.

2.2.2 Normalisierung

Eine weitere Herausforderung, welche sich beim Vergleich von Preisen verschiedener Aktien ergibt, ist deren unterschiedliche Preisniveaus. Während eine Aktie bei USD 30 handelt, bewegt sich eine andere auf einem Niveau von USD 1000. Eine Vergleichbarkeit lässt sich dadurch herstellen, wenn nicht absolute Preise, sondern relative Returns in der Analyse verwendet werden. Tatsächlich ist dies in der Payoff-Funktion grösstenteils sichergestellt. Das absolute Preisniveau fliesst allerdings auch in die Berechnung des Gamma Cash mit ein. Um auch hier eine Vergleichbarkeit der Werte sicherzustellen, wird bei allen nachfolgenden Analysen der Preis des adjustierten Vortagesendkurs auf ein Niveau von 100 (dies entspricht dem letzten Neutralisierungszeitpunkt des Deltas) standardisiert.

Sind all diese Schritte durchgeführt, lässt sich der Payoff berechnen. Da der Payoff - wie bereits ausgeführt - quadratisch auf Preisveränderungen reagiert, lohnt sich an dieser Stelle eine Untersuchung der Daten auf einflussreiche Beobachtungen. Ist der gesamte Payoff über alle Datensätze hinweg durch wenige Einträge dominiert könnten nachfolgende Analysen verfälscht werden. In diesem Fall besteht die Gefahr, dass lernende Algorithmen zu stark an diesen Einträgen orientieren. Weniger einflussreiche aber häufiger realisierte Beobachtungen gingen dabei unter. Dies ist insbesondere unter dem Gesichtspunkt von Relevanz, als dass vorliegende Analysen auf eine Prognose “normaler” Marktsituationen abzielt.

Zur Veranschaulichung einflussreicher Beobachtungen zeigt Abbildung 3 die Lorenzkurve des Payoffs (bei Tagesend-Ausgleich) über alle in der Analyse berücksichtigten Werte. Dazu werden die Payoffs der einzelnen Einträge in aufsteigender Weise sortiert. Deren kumulativer Anteil (Y-Achse) wird dem kumulativen Anteil der Datenpunkte (X-Achse) gegenübergestellt. Bei genau gleichverteiltem Beitrag jedes einzelnen Eintrages zum Payoff würde eine Gerade mit Steigung 45° resultieren. Je stärker der Einfluss einzelner Beobachtungen, desto stärker konvex die Kurve.

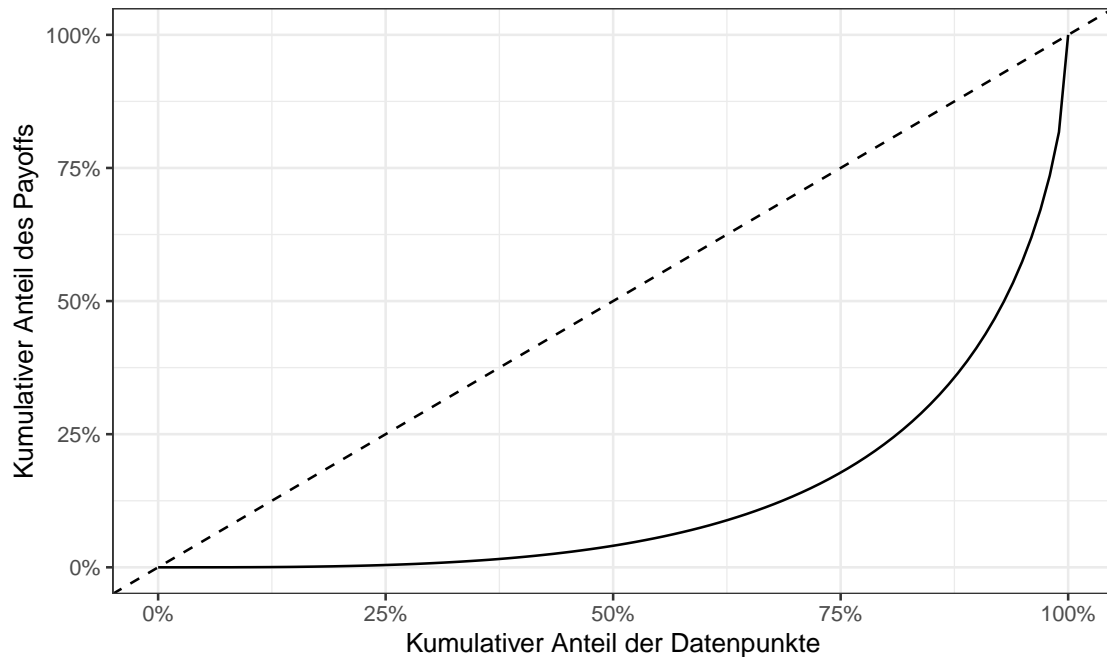


Abbildung 3: Lorenzkurve des Payoffs

Im vorliegenden Fall weist die Lorenzkurve eine deutliche Konvexität auf. Zumal im Datensatz sowohl unterschiedliche Titel als auch ein langer Zeithorizont mit volatileren als auch ruhigeren Marktsituationen repräsentiert ist, überrascht dieses Ergebnis nicht. Eine gewisse Vielfalt der Daten ist im Hinblick auf die Vorhersage von unterschiedlichen Kaufs- und Verkaufspreisen gar erwünscht. Sollen nachfolgende Algorithmen ja gerade versuchen, die unterschiedlichen Preisschwankungen zu prognostizieren. Andererseits zeigt die Darstellung auch, dass kein allzu extremer Einfluss einzelner Einträge auszumachen ist. In diesen Fällen wäre die Kurve fast auf dem unteren und rechten Rand der Grafik zu liegen gekommen. Es lässt sich damit der Schluss ziehen, dass die Bereinigung der Daten für den vorliegenden Zweck erfolgreich war. Die Daten zeigen sowohl die gewünschte Variabilität ohne dass dabei wenige Extremwerte das Ergebnis verfälschen würden.

2.2.3 Adjustierung unterschiedlicher Volatilitäten

Vorliegende Analyse basiert auf einer Vielzahl ganz unterschiedlicher Aktien. Es ist bekannt, dass die Preise unterschiedlicher Aktien in gleichen Marktsituationen mit unterschiedlich starken Ausschlägen reagieren. Beispielsweise reagieren Unternehmen, welche im Luxusgüterbereich (Bsp. Richemont, Swatch) tätig sind stärker als Anbieter im Bereich der Grundversorgung (Bsp. Nestle). Man spricht in diesem Zusammenhang auch von zyklischen und defensiven Werten.

Die Adjustierungen der Preisbewegungen auf eine vergleichbare Ausschlagshöhe / Volatilität könnte den nachfolgenden Algorithmen helfen, grundsätzlich ähnliche Verhaltensmuster zu erkennen. Diese Adjustierung kann auf verschiedene Arten erfolgen. Einige der Freiheitsgrade sein:

- Welcher Volatilitätsschätzer soll verwendet werden?
- Über welche Zeitdauer soll die Volatilität geschätzt werden?
- Soll ein gleichgewichteter Schätzer verwendet werden, oder ein gewichteter?
- etc.

Wir entscheiden uns für einen Volatilitätsindikator von Yang and Zhang (2000). Dieser ist in der Lage, nicht nur Close-to-Close Preise, sondern alle im Datensatz vorhandenen Preise - namentlich Open, High, Low und Close - zu berücksichtigen. Es handelt sich dabei um eine modifizierte Version des Garman und Klass Schätzers, der auch mit Opening Sprüngen umzugehen weiss.

Als Zeitdauer wählen wir 10 Tage. Dies ist einerseits ein gebräuchliches Standardfenster, zudem erfolgte auch die Datenaufbereitung (vgl. 2.2.1) auf der Bedingung, dass für jeden im Datensatz verbliebenen Eintrag dieses Fenster ohne Unterbruch vorhanden sein muss. Auf eine Gewichtung des Schätzers wird aus Gründen der Einfachheit verzichtet.

Bezeichne $vol_{i,10}$ die beschriebene 10-Tages Volatilität jeden Eintrages, so erfolgt die Adjustierung aller Preisdaten mittels folgender Formel:

$$P_{i,vol10} = (P_i - 100) / \sqrt{vol10_i} + 100$$

Für ein zweites adjustiertes Datenset wählen wir ein längerfristiges Volatilitätsmass. Die offensichtlichste Variante, dazu das Berechnungsfenster von 10 auf 250 oder mehr Tage zu erhöhen greift allerdings zu kurz, zumal im Datensatz sowohl bereinigte wie auch bereits von Beginn weg fehlende Werte vorkommen. Eine Möglichkeit bestünde darin, diese mittels Interpolationsverfahren zu füllen. Wir entscheiden uns aber dafür, als langfristiges Volatilitätsmass den Medianwert aller annualisierten 10-Tages Volatilitäten des jeweiligen Ticker j zu wählen.

$$P_{i,vol} = (P_i - 100) / \text{median}(\sqrt{vol10_i, j}) + 100$$

Nachfolgende Analysen werden teilweise mit und teilweise ohne diese Volatilitätsadjustierung durchgeführt und miteinander verglichen. Wo nicht anders vermerkt erfolgt die Analyse ohne Adjustierung.

3 Infrastruktur und Tools

3.1 Cloud Setup

Gewisse der in dieser Arbeit gemachte Analysen sind sehr rechen- und Speicherintensiv. Sie stellen damit erhöhte Anforderungen an die zur Berechnung eingesetzte Infrastruktur. Für die vorliegenden Analysen stellte sich folgender Setup mit 3 unterschiedlichen virtuellen Maschinentypen (VMs) als geeignet heraus:

- Datenaufbreitung: Eine Maschine mit 4 Cores und mindestens 32 GB Memory zum Prototyping und die Datenaufbereitung
- Multi-CPU: Eine Maschine mit mindestens 16 CPUs zur Berechnung parallelisierbarer Aufgaben (z.B. KNN) auf grösseren Datensätzen
- GPU: Eine Maschine mit mindestens 16 GB RAM, 4 CPUs und eine für kleinere Machine Learning Probleme geeigneten GPU (bsp. NVIDIA Tesla K80 oder NVIDIA Tesla M60).

Im Rahmen dieser Arbeit wurden deshalb 3 der bekanntesten Cloud-Anbieter ausprobiert. Aus Sicht des Autors unterscheiden sich diese in ihrer Handhabung stärker als dies ursprünglich vermutet hätte werden können. Der Aufbau eines geeigneten Infrastruktursetups stellte sich trotz auf den ersten Blick vorgefertigter Varianten als zeitintensiv heraus. Aus Sicht des Autors lohnt sich der Einsatz dieser Zeit allerdings bereits zu Beginn des Projektes. Die eingesetzte Zeit lässt sich später durch Zeiteinsparnisse aufgrund geeigneter Infrastruktur später wieder aufholen. Hinzu kommt, dass eine einmal gefundene und funktionierende Einstellung auch für spätere Projekte wieder eingesetzt werden kann.

Aus diesem Grund seien die gemachten Erkenntnisse an dieser Stelle festgehalten. Allen beschriebenen Lösungen gemein ist, dass sie ein für Studenten freies Start-Kontingent anbieten. Die Ausführungen beziehen sich auf diese. Ebenfalls allen Anbietern gemein ist, dass sich virtuelle Computer mit wenigen Klicks und dem gewünschten Betriebssystem erstellen lassen. Alle 3 getesteten Dienste bieten ferner neben normalen Instanzen auch sogenannte “Spot” Instanzen an. Diese unterscheiden sich von normalen VMs insofern, als dass es sich dabei um Einmalinstanzen handeln, welche nicht beendet und wieder hochgefahren werden können. Einmal beendet erlöschen Spot-Instanzen. Bei grosser Nachfrage nach Rechenkapazität können Spot Instanzen vom Anbieter zudem ohne Vorwarnung heruntergefahren werden. Sie eignen sich daher nur für nicht Unterbrechungsanfällige Prozesse. Im Gegensatz sind sie deutlich günstiger als reguläre Instanzen.

3.1.1 Microsoft Azure

Die Cloud-Computing Dienste von Microsoft nennen sich Azure. Die Plattform bietet verschiedene vordefinierter Maschinentypen, welche sich im Wesentlichen in der Anzahl CPU, RAM und persistentem Speicher unterscheiden. Einige Maschinen bieten zudem Zugriff auf eine oder mehrere GPU.

Beim Test zeigte sich hingegen, dass Account für Bildungseinrichtungen oft keine Maschinen verfügbar waren. Die Verfügbarkeit unterscheidet sich zudem je nach Tageszeit. Während am frühen Morgen Mitteleuropäischer Zeit manchmal Maschinen verfügbar waren, war dies weder am Nachmittag noch am Abend der Fall. Die grösste Maschine, welche beim Test über mehrere Tage hinweg erstellt werden konnte war eine NC6 Instanz mit 6 CPUs 56 GB Memory und einer NVIDIA Tesla K80 GPU. Insbesondere für CPU-lastige Analysen stellte sich dieser Setup als zu wenig gut heraus.

Wie bei den anderen Anbietern auch unterscheidet sich die Verfügbarkeit der VMs je nach Region. Bei Azure kommt allerdings erschwerend hinzu, dass alle Regionen einzeln durchprobiert werden müssen. Spot Instanzen stehen zudem nur für gewisse Maschinentypen zur Verfügung. Auch für die oben erwähnte Instanz stand beim Test die Spot-Option nicht zur Verfügung.

Spot Instanzen lassen sich vor allem dann gut nutzen, wenn Systemabbilder einfach erstellt und davon später wieder neue Instanzen erzeugt werden können. Azure unterscheidet sich hierbei von seinen Konkurrenten, als dass dies nicht einfach auf Knopfdruck erfolgt. Viel mehr muss manuell eine “Generalisierung” der Instanz vorgenommen werden (Vgl. <https://docs.microsoft.com/en-us/azure/virtual-machines/windows/capture-image-resource>).

Die Beurteilung der Qualität der Dokumentation ist subjektiv. Aus Sicht des Autors ist diejenige von Azure weniger ausführlich und selbsterklärend als diejenige der anderen getesteten Kandidaten.

3.1.2 Google Cloud

Die Cloud-Computing Dienste von Google nennen sich “Google Cloud”. Neu registrierende Kunden profitieren im Vergleich mit den anderen Anbietern vom höchsten kostenlosen Startguthaben (Azure mit Studenten Account: \$100, AWS (\$30) mit Github Starter Package (\$70): \$100, Google Cloud: \$300).

Die Management Oberfläche wirkt im Vergleich zu den Konkurrenten besser aufgeräumt. Die Dokumentation ist gut. Als einzige der getesteten Anbieter konnten die VMs zudem völlig individuell gestaltet werden (Anzahl CPU, Memory, Anzahl und Art GPU).

Als Nachteil entpuppte sich im Test allerdings das komplexe Limitensystem. Fast alle Komponenten unterliegen verschiedenen Limiten. Um eine GPU hinzufügen zu können müssen die beispielsweise gleichzeitig die Limiten für “GPU global”, “GPU der gewählten Region” und “GPU des jeweiligen Typs” (Bsp. NVIDIA Tesla K80) erfüllt sein.

Während die Standardeinstellungen Instanzen von bis zu 24 CPUs erlauben, sind noch mehr Einheiten nicht möglich. Die Standardeinstellung für GPUs beträgt gar 0. Eine Erhöhung der Quote kann im Management Portal beantragt werden. Im Test zeigte sich jedoch, dass sowohl Anträge zu Erhöhung der CPU wie auch GPU Limiten innerhalb weniger Minuten mit dem Verweis auf fehlende Zahlungshistorie automatisch abgelehnt wurden. Die Erstellung einer Instanz mit GPU Support gelang so auch über mehrere Tage hinweg nicht. Schriftliche Kontaktaufnahmen mit Bitte und Begründung der GPU Limite auf 1 wurden wiederholt mit Standard-Antworten abgelehnt. Auch nach telefonischer Kontaktaufnahme mit dem Support gelang es nicht, die Quote zu erhöhen. Das Problem stiess beim Mitarbeitenden von Google zwar auf Verständnis, er selbst konnte die Quote allerdings ebenfalls nicht erhöhen.

3.1.3 Amazon Web Services (AWS)

Das Cloud Computing Angebot von Amazon nennt sich “Elastic Compute Cloud” (Amazon EC2). Als einziger der getesteten Anbieter fällt dieser Anbieter nicht durch nicht verfügbare Maschinentypen oder unzureichenden Limiten auf. Ferner können auch die meisten Maschinen (im Test bis 40 CPU) als Spot Instanzen ausgeführt werden. Da die Imageerstellung zudem sehr einfach auf Knopfdruck erfolgt, können diese sehr kostengünstig verwendet und wiederhergestellt werden. AWS fällt zudem durch eine sehr gute ausführliche Dokumentation auf. So ist beispielsweise die Erstellung eines von der Instanz unabhängigen persistenten Speichers (in AWS-Lingo: EBS) sowie der Prozess zum Mounten ebendieses Schritt-für-Schritt erklärt (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html>). Damit kann auch bei der Verwendung von Spot Instanzen dauerhaft gespeichert werden. Ferner sind auch die Preise im Vergleich zur Konkurrenz sehr transparent und wettbewerbsfähig.

Da auch nach längeren Versuchen nur mit AWS der oben beschriebene Ziel-Infrastruktur einer Basis, Multi-CPU und einer einfachen GPU-Instanz erstellt werden konnte, wurde für die vorliegende Arbeit dieser Anbieter als Cloud Computing Lösung verwendet.

3.2 Verwendete Software

Für die Analysen der vorliegenden Arbeit hat sich der Einsatz von R als auch Python bewährt. Es zeigte sich, dass beide Tools / Sprachen ihre Stärken in verschiedenen Bereichen haben. Grundsätzlich lässt sich dies so zusammenfassen, dass Datenaufbereitende Schritte in R durchgeführt wurden. Für das Training der neuronalen Netze stellten sich die entsprechenden Bibliotheken von Python als geeigneter heraus.

3.2.1 R / RStudio Server

Die Datenaufbereitung und -bereinigung wie die Analysen ohne neuronale Netze wurden in der Sprache R und der Entwicklungsumgebung RStudio Server durchgeführt.

Ersteres kann auf eine sehr breite Community mit entsprechend gut dokumentierten Beispielen in diversen Foren zurückgreifen. Ferner hat sich in den letzten Jahren unter dem Namen “tidyverse” eine Sammlung gut unterhaltener und dokumentierter Packages zum defacto-Standard etabliert. Dieses wird von Mitarbeitern der Firma RStudio unterhalten und steht wie für R üblich Open Source zur freien Benützung zur Verfügung. Gleiche Firma ist es auch, welche unter dem Namen RStudio Server eine hostbare Entwicklungsumgebung (IDE) anbietet. Diese wird in einer frei verfügbaren und einer kostenpflichtigen Variante angeboten. Für die Arbeit wurde die freie Version verwendet und es kam nie der Bedarf zum Upgrade auf die kostenpflichtige Version auf. Verschiedene Builds der gängigsten Linux Distributionen stehen auf der Unternehmenswebseite zur Verfügung (Vgl. <https://rstudio.com/products/rstudio/download-server/>). Der Zugriff auf die IDE erfolgt über den Browser. Das Look and Feel unterscheidet sich nicht von einer ebenfalls frei verfügbaren lokalen Installation. Auf diese Weise lässt sich Code direkt auf den erstellten Cloud Instanzen entwickeln, ohne auf den Komfort von Entwicklungsumgebungen unterstützen zu müssen.

Schwächen bei der Verwendung von R zeigt sich in der inherenten Single-Threadigkeit des Tools. Die Multithreadigkeit kann mit Hilfe zusätzlicher Packages (z.B. parallel, pbapply) erreicht werden und ist insbesondere beim Einsatz auf der oben beschriebenen Multi-CPU Instanz von Relevanz. Zwar stehen mit Keras und Tensorflow auch entsprechende Packages für Deep Learning Ansätze zur Verfügung. Deren Verwendung im Zusammenspiel mit RStudio Server stellte sich im vorliegenden Anwendungsfall allerdings als sehr instabil heraus, was sich in mehrerer Abstürzen der Entwicklungsumgebung manifestierte. Da diese Packages selber lediglich Wrapper auf die gleichnamigen Python Libraries darstellen, stellte sich deren Verwendung direkt in Python als die bevorzugte Variante heraus.

Zum Austausch zwischen den beiden Sprachen stellte sich dabei der Weg über zwischengespeicherte “Feather” Files als am geeignetsten heraus. Es handelt sich dabei um ein Format, das beide Sprachen auch für grössere Datenmengen sehr performant laden und speichern können. Tatsächlich kann in R / RStudio mit Hilfe des Packages “reticulate” auch Python Code direkt ausgeführt werden, resp. Objekte beider Sprachen automatisch ausgetauscht werden. Auf diesen Austausch wurde aus Einfachheitsüberlegungen allerdings verzichtet.

Gewöhnungsbedürftig ist hingegen die matrix- und vektorbasierte programmierweise in R. Zwar sind gleiche Manipulation auch mit klassischer “loop-Ansätzen” möglich. Diese gehen allerdings mit erheblicher Performanceeinbussen speziell bei Datengrößen wie sie diese Arbeit verwenden einher. Performancekritische Funktionen können aber mit Hilfe des Packages “Rcpp” in C++ geschrieben und einfach mit R verknüpft werden. Dies wurde im Laufe der Arbeit für wenige kritische Funktionen verwendet. Insbesondere bei stark parallelisierbaren Aufgaben, nicht zuletzt auch, um grössere Datenmengen Memory-effizient ausführen zu können.

3.2.2 Python / (Ana)conda

Für Python stand keine auf dem Server ausführbare Entwicklungsumgebung zur Verfügung. Da sich der Einsatz von Python im Rahmen des Projektes auf das Training neuronaler Netze beschränkte, reichte das Ausführen einer Jupyter Notebook Servern gut aus. Auf das Notebook lässt sich bei diesem Setup wiederum einfach via Webbrowser zugreifen.

Ebenfalls als wertvoll zeigte sich die Verwendung von kapselbaren conda Environments. Diese erlauben projektspezifische Library Installationen sowohl für Python wie auch R-Bibliotheken. Es zeigt sich, dass das erwähnte Problem der Instabilität von Keras und Tensorflow bei direkter Verwendung der Python Libraries nicht auftauchte. Erwähnenswert ist hier aber, dass für die Verwendung der GPU Version auf dem System sowohl passende Grafiktreiber sowie die zur Tensorflow Version passende Version von CUDA installiert sein muss. Erwähnenswert ist dies insbesondere daher, da es auch bei aktuellster Tensorflow Bibliothek nicht die aktuellste CUDA Version sein durfte. Über die zu verwendenden Versionen gibt die Tensorflow Website Auskunft (<https://www.tensorflow.org/install/gpu>).

4 Analyse

Für die Analyse der Daten mit dem Ziel einen Kaufs- sowie einen Verkaufskurs zu prognostizieren, bei dem der Delta-Hedge nachgezogen werden soll, werden nachfolgend verschiedene Techniken eingesetzt. Diese sind:

- Einfache Optimierungen
- Nearest Neighbors (KNN)
- Neuronale Netzwerke

Allen Analysen gemein ist, dass jeweils gefundene Strategien mit der Referenzstrategie verglichen wird, welche keine innertägliche Anpassung des Deltas vorsieht. Eine weitere Gemeinsamkeit liegt darin, dass die verwendeten Daten keine Aussage über den Verlauf des Preises innerhalb des Tages zulassen. Insbesondere kann nicht ermittelt werden, ob zuerst eine obere oder eine untere Preisgrenze überschritten wurde. Da diese Reihenfolge aber wie in Kapitel 1.2 ausgeführt von Relevanz ist, wird für alle Analysen ein Ansatz verwendet, bei welchem zufällig bestimmt wird, ob am jeweiligen Tag zuerst eine Abwärts- oder eine Aufwärtsbewegung stattgefunden hat. [^Alternative denkbare Vorgehensweisen sind: Immer zuerst Aufwärtsbewegung, immer zuerst Abwärtsbewegung, immer die bezügl. Payoff schlechtere Reihenfolge oder immer die bezügl. Payoff bessere Variante.] Auch ein mehrmaliges Erreichen der Kaufs- und Verkaufsschwelle ist innerhalb des Tages bei sehr fluktuierenden Preisen in Realität denkbar. Es wären bezüglich Optimierung des Payoffs sogar sehr wünschenswerte Ereignisse. Auf die Berücksichtigung solcher Fälle wird in der Analyse allerdings verzichtet. Das Bewusstsein über deren Existenz ist aber bei der Interpretation der Ergebnisse interessant, da die Payoffs der

gefundenen Strategien diesbezüglich als untere Grenzen des Payoffs betrachtet werden können.

Eine weitere Gemeinsamkeit aller Analysen ist, dass der bereinigte Datensatz in ein Trainings- (80%) und ein Testdatensatz (20%) aufgeteilt wird. Diese Aufteilung erfolgt zufällig und wird für alle Analysen zwecks Vergleichbarkeit der Ergebnisse beibehalten.

4.1 Einfache Optimierungen

4.1.1 Ohne Berücksichtigung der Marktvolatilität

Eine erste Möglichkeit, optimale Kaufs- und Verkaufspreise zu finden, besteht darin, diese im Trainingsdatensatz mittels einfacher Optimierung zu ermitteln. In einer ersten sehr einfachen Evaluation sollen Kaufs- und Verkaufsmarken als prozentuale Abweichungen vom aktuellen Eröffnungspreis festgelegt werden. Die resultierenden Payoffs bei einer solchen Festlegung lassen sich dann ins Verhältnis zum Referenzpayoff mit Ausgleich per Tagesende stellen. Ein Payoff-Verhältnis über 1 kennzeichnet damit eine Strategie, welche der Referenzstrategie überlegen ist. Verhältnisse unter 1 kennzeichnen unterlegene Strategien.

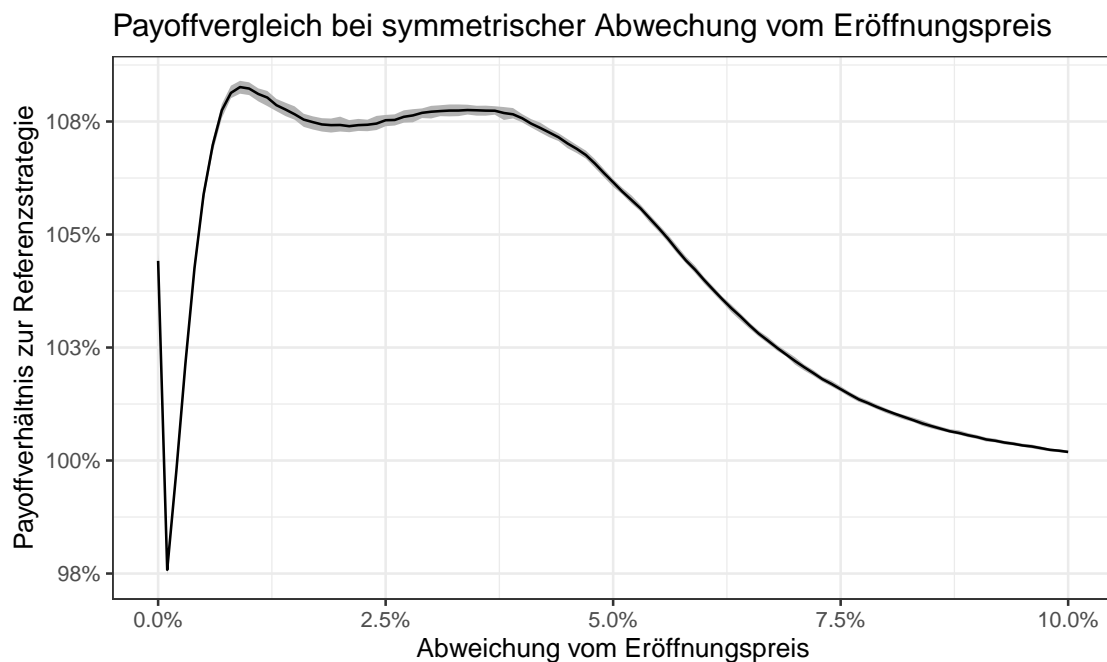


Abbildung 4: Payoffvergleich bei symmetrischer Abweichung vom Eröffnungspreis (ohne Berücksichtigung Marktvolatilität)

Abbildung 4 veranschaulicht dieses Verhältnis bei variierender symmetrischer Abweichung vom Startpreis. Lesebeispiel: Werden die Kaufs- und Verkaufspreise zum Handel

untertags 2.5% unter resp. über dem Eröffnungskurs des jeweiligen Tages gesetzt, so resultiert ein Gewinn, welcher rund 8% über demjenigen der Referenzstrategie liegt.

Bei genauerer Betrachtung weist die Kurve einige interessante Eigenschaften auf: Der höchste Payoff wird bei einer Auslenkung der Preise um 0.9% erreicht. Der Payoffüberschuss beträgt an diesem Punkt rund 8.3%. Gleichzeitig zeigt sich, dass ein mehr oder weniger konstanter Payoff-Überschuss von rund 8% im ganzen Auslenkungsbereich von 0.7 bis rund 4% erreicht werden kann.

Während im Bereich tieferer Auslenkungen viele kleinere Gewinne realisiert werden, sind es beim setzen breiterer Schranken nur noch wenige, dafür grössere. Die Kurve zeigt, dass sich diese beiden Effekte im genannten Bereich in etwa die Waage halten. Dieses Ergebnis ist insofern interessant, als dass die genaue Preisbestimmung gar nicht von so grosser Relevanz sein könnte. Wichtig dabei zu erwähnen ist auch, dass während der Datenbereinigung tendenziell grosse Auslenkungen aus dem Datensatz entfernt wurden (2.2.1). Werden vermehrt auch extreme Marktbewegungen zugelassen, verschiebt sich die optimale Auslenkung der Preisschranken nach oben. In Kombination mit der Erkenntnis, dass auch bei stärkerer Bereinigung gute Payoffs bis 4% Auslenkung erreicht werden, könnte dies eine Motivation sein, die Preise eher breiter zu setzen.

Eine weitere Besonderheit der Kurve zeigt sich mit dem Abwärtsknick bei sehr kleinen Auslenkungen. Erklären lässt sich dieser Knick dadurch, dass bei allen Kursverläufen, bei denen der Eröffnungskurs gleichzeit Höchst- oder Tiefstkurs ist, mindestens eine Schranke nicht mehr erreicht werden kann. Bereits beim Setzen etwas grösserer Schranken wird dieser Effekt wieder mehr als ausgeglichen.

Auffällig ist auch die Tatsache, dass eine Auslenkung von 0 (und damit einem Wiederherstellen der Delta-Neutralität gleich zum Eröffnungskurs) eine deutlich bessere Performance als die Referenzstrategie aufweist. Dies lässt sich damit erklären, dass die Werte im Datensatz offenbar die Tendenz eines “Overshootings” der Eröffnungspreise zeigen. Das beobachtete Bild lässt vermuten, dass sich die Preise im Laufe des Tages in der Tendenz wieder eher Richtung Schlusskurs des Vortages entwickeln. Ein Ausgleich der aufgebauten Delta-Position “über Nacht” gleich zu Beginn des Handelstages auszugleichen, scheint daher ebenfalls besser, als bis am Abend zu warten.

Schliesslich stellt sich auch die Frage, inwiefern die gefundenen Ergebnisse als statistisch signifikant bezeichnet werden können. Zur Beurteilung dieser Frage wurde mittels Bootstrapverfahren ein 95%-Konfidenzband der Kurve ermittelt. Dieses ist als graue schraffierte Fläche am Rand der Kurve ersichtlich. Es zeigt sich, dass dieses Band relativ schmal ausfällt. Dies kann als Konsequenz der ausführlichen Datenbereinigung gesehen werden. Diese führt dazu, dass auch über verschiedene Bootstrap-Samples hinweg die Payoffs stabil und wenig beeinflusst durch einzelne Beobachtungen ausfallen.

Als zweites Mass zur Beurteilung der Aussagekraft der gefundenen Ergebnisse lassen sich zudem auch die Werte des Testdatensatzes heranziehen. In diesem beträgt der Payoffüberschuss im Vergleich zur Referenzstrategie bei 0.9% Auslenkung ebenfalls rund 8.3% und auch bei einer Auslenkung von 4% kommt der Überschuss bei 7.7 zu liegen. Beide

Werte zeigen hohe Ähnlichkeit mit dem Trainingsdatensatz und unterstreichen damit auf Signifikanz der Ergebnisse.

4.1.2 Mit Berücksichtigung der Marktvolatilität

Die bisherige Analyse untersucht die Auslenkung der Kaufs- und Verkaufspreise um den gleichen prozentualen Wert für alle Einträge im Datensatz. Die Bimodalität der Kurve in Abbildung 4 deutet darauf hin, dass es sich dabei um eine Überlagerung mehrerer Kurven handeln könnte. Gelingt es, diese zu separieren und einzelnen Gruppen von Kursverkäufen im Datensatz zuzuweisen, könnten individuellere Preisschranken gewählt werden. Mit Hilfe dieser könnte der Payoff im Idealfall weiter gesteigert werden.

Als Klassifizierung sei dazu die Volatilität der vergangenen 10 Handelstage herangezogen. Die Vermutung liegt nahe, dass eine volatile Marktsituation in der kurzfristigen Vergangenheit auch am nächsten Tag fortgesetzt werden könnte (bsp. Zeiten mit vielen marktrelevanten Informationen wie Finanzkrise, Corona-Krise, Dividend-Season etc.). Umgekehrt könnten eher ruhig verlaufende Börsentage in den vergangenen Tagen auf eine ruhige Situation auch am aktuellen Tag hinweisen (bsp. Ruhigere Zeiten während Sommerferien, etc).

Um dies zu untersuchen werden alle Einträge des Datensatzes in 2 Gruppen aufgeteilt. Einträge, welche eine aktuelle 10-Tages-Volatilität über demjenigen des Median aufweisen werden in eine Gruppe hoher Volatilität, die andern Einträge einer Gruppe tiefer Volatilität zugeordnet. Zu beachten gilt es hierbei, dass der Medianwert dabei einerseits nur innerhalb des jeweiligen Tickers betrachtet wird und für dessen Berechnung auch nur vergangene Werte mit einbezogen werden.

Für beide Gruppen lassen sich danach im Trainingsset die bereits bekannten Payoffvergleiche zum Referenzszenario durchführen und graphisch darstellen (vgl. Abbildung 5).

Die Grafiken zeigen das erwartete Bild. Für die Gruppe tieferer vergangener Volatilitäten wird der maximale Payoff bei einer Auslenkung von 0.6% erreicht. Bei der Gruppe höherer Volatilität bei 3.7%. Angewendet auf das Testset resultiert ein Payoffüberschuss von rund 9.6%. Dies ist noch einmal deutlich höher als im ungruppierten Fall.

Insbesondere bei der Gruppe der höheren Volatilitäten weist die Kurve aber weiterhin eine bimodale Form auf. Es scheint, als ob mit der gemachten Gruppierung zwar ein Teil der Varianz des aktuellen Tages erklärt werden konnte, weitere Teile davon aber unerklärt bleiben. Eine Möglichkeit bestünde nun darin, die Anzahl Gruppierungen weiter zu erhöhen, indem beispielsweise nicht nur der Median, sondern die Quartile, Dezile, Percentile etc. als Klassifikatoren gewählt werden. Darauf sei an dieser Stelle verzichtet und zu andern Ansätzen des maschinellen Lernens übergegangen.

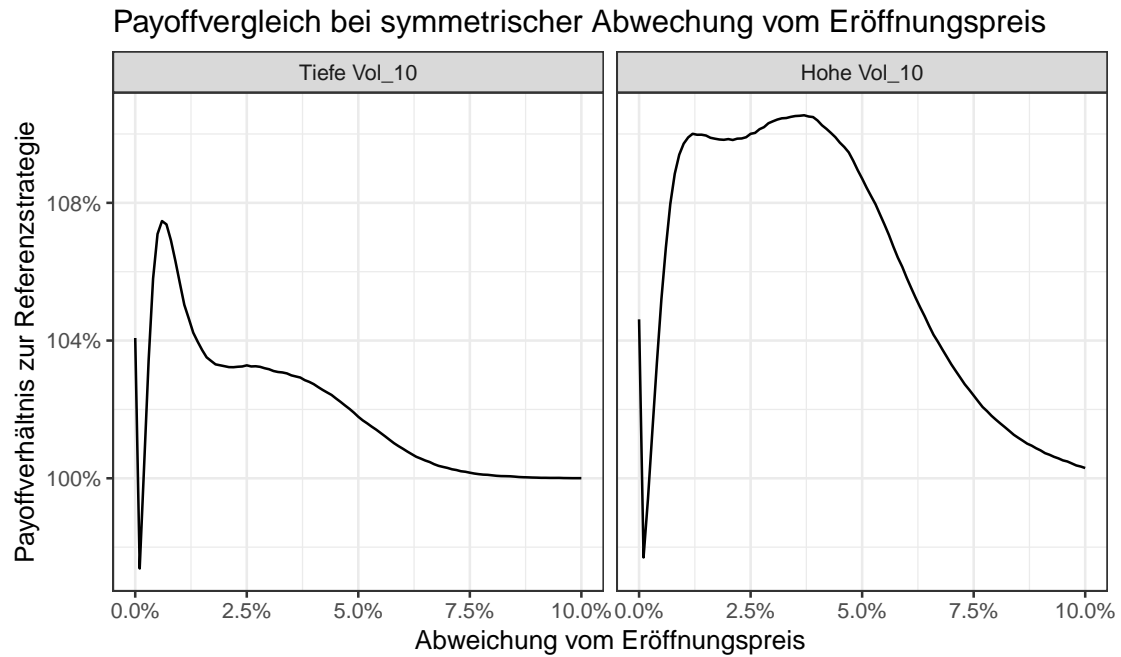


Abbildung 5: Payoffvergleich bei symmetrischer Abweichung vom Eröffnungspreis (mit Berücksichtigung Marktvolatilität)

4.2 Nearest Neighbor Ansätze

Die ökonomische Theorie scheint keinen offensichtlichen Grund zu liefern, weshalb die Volatilität des aktuellen Tages in genau 2 (oder x) Gruppen eingeteilt werden sollte. Auf der anderen Seite haben bisherige Analysen gezeigt, dass die aktuelle Volatilität ein erklärender Faktor für die aktuelle Volatilität sein kann. Im vorliegenden Kapitel soll dieser Gedanke weiter verfolgt werden.

4.2.1 Distanzmasse

Die Idee des Ansatzes dieses Kapitels besteht darin, nicht lediglich x vordefinierte Gruppen für symmetrische Abweichungen zu finden, viel mehr soll versucht werden, ähnliche Kursverläufe wie den aktuellen in der Vergangenheit zu finden und individuell darauf zu reagieren. Die zugrund liegende Hypothese ist dabei, dass bei ausreichender Historie in der Vergangenheit ähnliche Muster erkannt und daraus Rückschlüsse auf die Zukunft (genauer: die Kursentwicklung des aktuellen Tages) gemacht werden können.

Da es sehr unwahrscheinlich ist, die genau gleichen Kursverläufe in der Historie wiederzufinden, muss ein Distanzmass definiert werden, welches die Nähe der Kursverläufe quantifiziert. Um diese zu berechnen formulieren wir für jeden Eintrag i den bisherigen Kursverlauf als Vektor $hist$.

$$hist_{i,t} = (Open_{i,t}, High_{i,t}, Low_{j,t+1}, Close_{i,t+1}, Open_{i,t+1}, ..., Close_{i,1}, Open_{i,0})$$

Der zweite Index gibt dabei an, wieviele Tage der Vergangenheit mit einbezogen werden. Ein Index von 0 bezieht sich auf den zu prognostizierenden, aktuellen Tag. Um die Ähnlichkeit zweier Einträge i und j zu berechnen, bieten sich 2 Distanzmasse an:

- 1) Die Manhattan Distanz (auch L1-Norm)

$$\|hist_{i,t} - hist_{j,t}\|_1 = |Open_{i,t} - Open_{j,t}| + |High_{i,t} - High_{j,t}| + \dots + |Open_{j,0} - Open_{i,0}|$$

- 2) Die Euklidische Distanz (auch L2-Norm)

$$\|hist_{i,t} - hist_{j,t}\|_2 = \sqrt{(Open_{i,t} - Open_{j,t})^2 + (High_{i,t} - High_{j,t})^2 + \dots + (Open_{j,0} - Open_{i,0})^2}$$

Die Ergebnisse beider Masse sollen anschliessend miteinander verglichen werden.

4.2.2 Setup und Berechnungsdauer

Mithilfe obiger Distanzmasse lassen sich für jeden Kursverlauf, die k ähnlichsten Kursverläufe der Vergangenheit ermitteln. Zuvor seien an dieser Stelle aber einige Überlegungen zur Berechnungskomplexität des Problems gemacht: Der vorliegende bereinigte Datensatz weist 5'572'087 tägliche Kursverläufe auf. Soll jeder Kursverlauf mit jedem andern verglichen werden, so ergeben sich bei einem Brute-Force Ansatz $5'572'087^2$ Distanzberechnungen. Unter Berücksichtigung der Tatsache, dass das Problem symmetrisch ist und jeder Kursverlauf nicht mit sich selbst verglichen werden muss, halbiert sich die Komplexität zwar um mehr als die Hälfte, bleibt aber so gross, dass es zum Zeitpunkt des Schreibens dieser Arbeit nicht innerhalb weniger Sekunden oder Minuten auf einem handelsüblichen Heim-Computer berechnet werden kann.

Die für die Nearest Neighbors Suche eingesetzten Bibliotheken greifen daher typischerweise auf sophistiziertere Vorgehensweisen zurück. Eine davon sieht die Verwendung von Multidimensionalen Suchbäumen (Search Trees, auch Kd-Trees) vor. Diese gehen auf eine Idee von Bentley (1975) zurück. Sie basiert darauf, dass jede Dimension in 2 Bereiche aufgeteilt wird (z.B. beim Median). Dadurch wird der Raum in viele kleinere Sub-Räume aufgeteilt. Der Algorithmus macht sich danach zu Nutze, dass er nicht den ganzen Raum absuchen muss. Beginnend im aktuellen Sub-Raum werden schrittweise alle benachbarten Räume abgesucht, bis die geforderte Anzahl nächster Nachbarn gefunden ist. Allerdings benötigt dabei sowohl der Aufbau des Baumes wie auch die Suche im Baum Berechnungszeit.

Die Anzahl der Dimensionen beeinflusst dabei die benötigte Zeit erheblich. Im vorliegenden Fall liegt diese bei Historie von 10 Tagen à 4 Werten bei 41, wenn zusätzlich auch der (bekannte) Eröffnungspreis des aktuellen Tages mit einbezogen wird. Tests ergaben, dass die dafür benötigte Rechenzeit zu hoch war. Die zu verwendende Zeitperiode

wurde daher auf 3 Tage beschränkt. Dies resultiert in einer deutlichen Reduktion der Suchdimensionen auf 13.

Der vorliegende Fall unterscheidet sich von andern Nearest Neighbor Problemen ferner dadurch, als dass für jeden Eintrag lediglich Kursverläufe der Vergangenheit betrachtet werden sollten. Im Hinblick auf die Suchstruktur bedeutet dies, dass der KD-Tree nicht nur einmalig aufgebaut und danach für alle Verläufe auf Nachbarn durchsucht werden kann. Vielmehr muss der KD-Tree für jedes Datum mit allen Kursverläufen vor diesem Datum neu aufgebaut werden. [Es sind auch Mischlösungen denkbar, bei denen nur alle x Daten der Tree neu aufgebaut wird und dafür mehr Nachbarn ermittelt werden, die im Nachgang um nicht vorher realisierte Verläufe gefiltert würden. Dies hat aber das Problem, dass a) mehr Nachbarn ermittelt werden müssen und b) nicht 100% sichergestellt ist, dass die Anzahl "Reservenaachbarn" ausreichen.] Dieser spezielle Setup kommt einer Brute-Force Methode ihrerseits wieder entgegen, da aufgrund des Datums-Filter nicht stets alle Einträge durchsucht werden müssten.

Beiden Ansätzen gemein ist hingegen, dass sie sich sehr gut parallelisieren lassen und Bibliotheken zur Verfügung stehen, welche diese Methoden effizient implementieren. Wir entscheiden uns im vorliegenden Fall für eine KD-Tree Implementation ohne Approximation. Auf einem Rechner mit 40 Cores (20 physisch, 20 Hyperthreading Cores) dauert die Berechnung von 50 Nachbarn ca. 2 Stunden im Falle des euklidischen Distanzmaßes und ca. 3 Stunden im Falle der Manhattan Distanz. Dieser Hohe Anspruch eines KNN-Vorgehens an die Rechenleistung sollte bei der späteren Abwägung verschiedener Algorithmen mit berücksichtigt werden. Erwähnt sei an dieser Stelle aber auch, dass eine allfälligen Anwendung der Methode später nur wenige Titel (resp. nur diejenigen des aktuellen Tages) umfasst. Dies ist auch unter der Verwendung einer KNN-Methode in wenigen Sekunden möglich.

4.2.3 Bestimmung Kaufs- und Verkaufspreise

```
## Warning in ans * length(l) + if1: Länge des längeren Objektes
##           ist kein Vielfaches der Länge des kürzeren Objektes
```

Sind die Nachbarn eines Eintrages ermittelt lässt sich eine Prognose für den weiteren Kursverlauf des aktuellen Eintrages ableiten. Dazu wird aus den bekannten Kursverläufen der Nachbarn mittels geeigneter Metrik ein Prognosewert für den Folgetag abgeleitet. Abbildung @ref(fig:example_nearest_neighbor_outlook) illustriert dieses Vorgehen anhand eines Beispiels. Die Grafiken zeigen die Kursverläufe des aktuellen Eintrages (rot) sowie diejenigen der nächsten 5 Nachbarn für den aktuellen zu prognostizierenden Tag (t) sowie die 3 jeweils vorangegangenen Handelstage ($t-1$, ..., $t-3$). Die gestrichelte rote Linie zeigt den auf Basis der Nachbarn prognostizierten Wert.

Werden die prognostizierten Werte für Höchst- und Tiefstpreise als Kaufs- resp. Verkaufsschranken gewählt, lässt sich der Payoff berechnen und mit demjenigen der Referenzstrategie vergleichen. Wie bereits ausgeführt werden bei der Suche nach Nachbarn

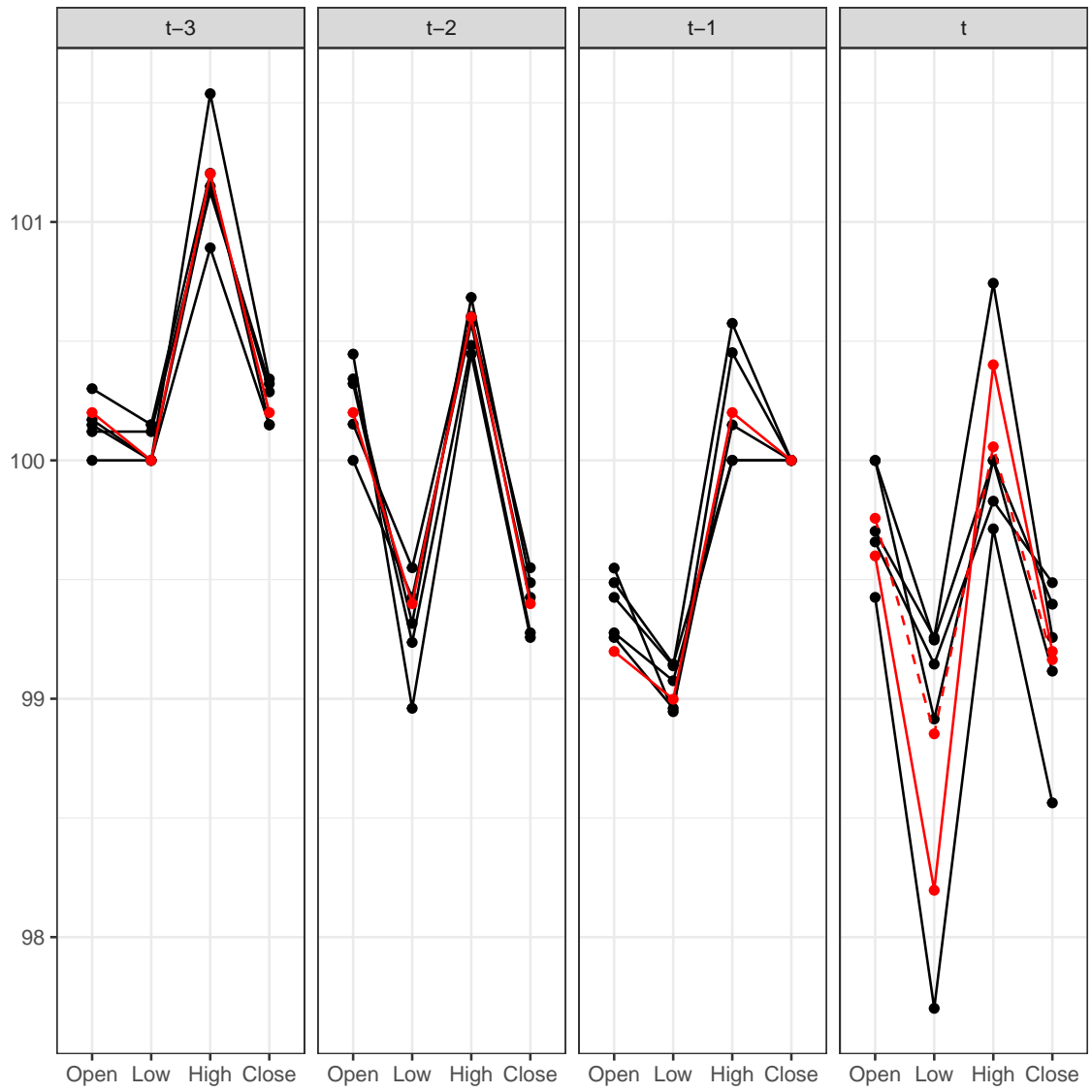


Abbildung 6: (#fig:example_nearest_neighbor_outlook)Exemplarische Kursprognose auf Basis nächster Nachbarn

nur Einträge der Vergangenheit berücksichtigt. Die Menge an verfügbaren Vergleichsverläufen nimmt damit mit fortlaufender Zeit zu. Stellt man den Payoff-Vergleich für das Training Set über die Zeit dar (vgl. Abbildung 7), wird der Lerneffekt der Methode sichtbar. Insbesondere 3 Eigenschaften der Kurve lassen sich ausmachen:

1. Hohe Volatilität des Vergleichsfaktor zu Beginn
2. Steigende Faktorhöhe mit fortlaufender Zeitdauer
3. Abflachung im Laufe der Zeit auf ein stabiles Level

Beide Eigenschaften decken sich mit der Intuition. Da zu Beginn der Datenreihe nur sehr wenige Vergleichsverläufe zur Verfügung stehen, reagiert die Kurve sehr sensitiv und schlägt entsprechend aus. Dies glättet sich im Laufe der Zeit mit dem Vorhandensein von mehr Vergleichsmöglichkeiten. Die zweite Eigenschaft des steigenden Faktors zeigt, dass der erhoffte Lerneffekt einzutreten scheint. Tatsächlich scheinen ähnliche Kursverläufe in der Vergangenheit zukünftige Entwicklungen teilweise erklären zu können. Dies deckt sich mit der Erkenntnis des vergangenen Kapitels. Anders als zuvor kann dieser Lernmechanismus hier aber sehr individuell und nicht beschränkt auf 2 Gruppen erfolgen. Die dritte Eigenschaft zeigt, dass dieser Lerneffekt nach gewisser Zeit gesättigt scheint.

Während sich bisherige Ausführungen auf die Analyse mit 5 Nachbarn ermittelt auf Basis der euklidischen Distanz beziehen, sind bei der Ermittlung der Prognosewerte auch andere Parametrisierungen denkbar. Neben der Unterscheidung des Distanzmasses sind dies insbesondere die Anzahl der zu berücksichtigenden Nachbarn und die Metrik zur Prognoseermittlung.

- **Anzahl Nachbarn:**

In der vorliegenden Arbeit werden die 50 nächsten Nachbarn jedes Eintrages ermittelt. Einmal ermittelt lassen sich davon auch weniger verwenden. Damit kann sehr einfach der Einfluss der Anzahl Nachbarn auf die Prognosequalität ermittelt werden. Denkbar sind dabei grundsätzlich unterschiedliche Ergebnisse. So lässt sich argumentieren, dass bei der Verwendung weniger Nachbarn auch diejenigen mit grösster Ähnlichkeit verwendet werden. Insbesondere bei Marktbewegungen die relativ selten sind, könnten fernere, weniger gut passende Nachbarn das Ergebnis hier nicht verzerren. Umgekehrt lässt sich argumentieren, dass bei häufigeren Marktsituationen die Berücksichtigung und Mittelung von mehr Nachbarn zu einem unverzerrteren Ergebnis führen könnte. Schliesslich könnte sich als drittes Ergebnis auch eine Konfiguration mit "mittlerer" Anzahl Nachbarn beähren, wenn beide vorherhigen Argumentationen verschmolzen werden. Aus diesem Grund analysiert und vergleicht die vorliegende Arbeit die Ergebnisse bei der Verwendung von 5, 20 und 50 Nachbarn.

- **Metrik der Prognoseermittlung:**

Eine erste Möglichkeit zur Prognose von Tiefst-, Höchst- und Schlusskurs besteht darin, den Mittelwert der jeweiligen Kursfortsetzungen der Nachbarn zu wählen. Alternativ zur einfachen Mittelwertbildung sind auch andere Verfahren denkbar. Beispielsweise wäre auch die Berücksichtigung der Distanz als Gewichtungsfaktor

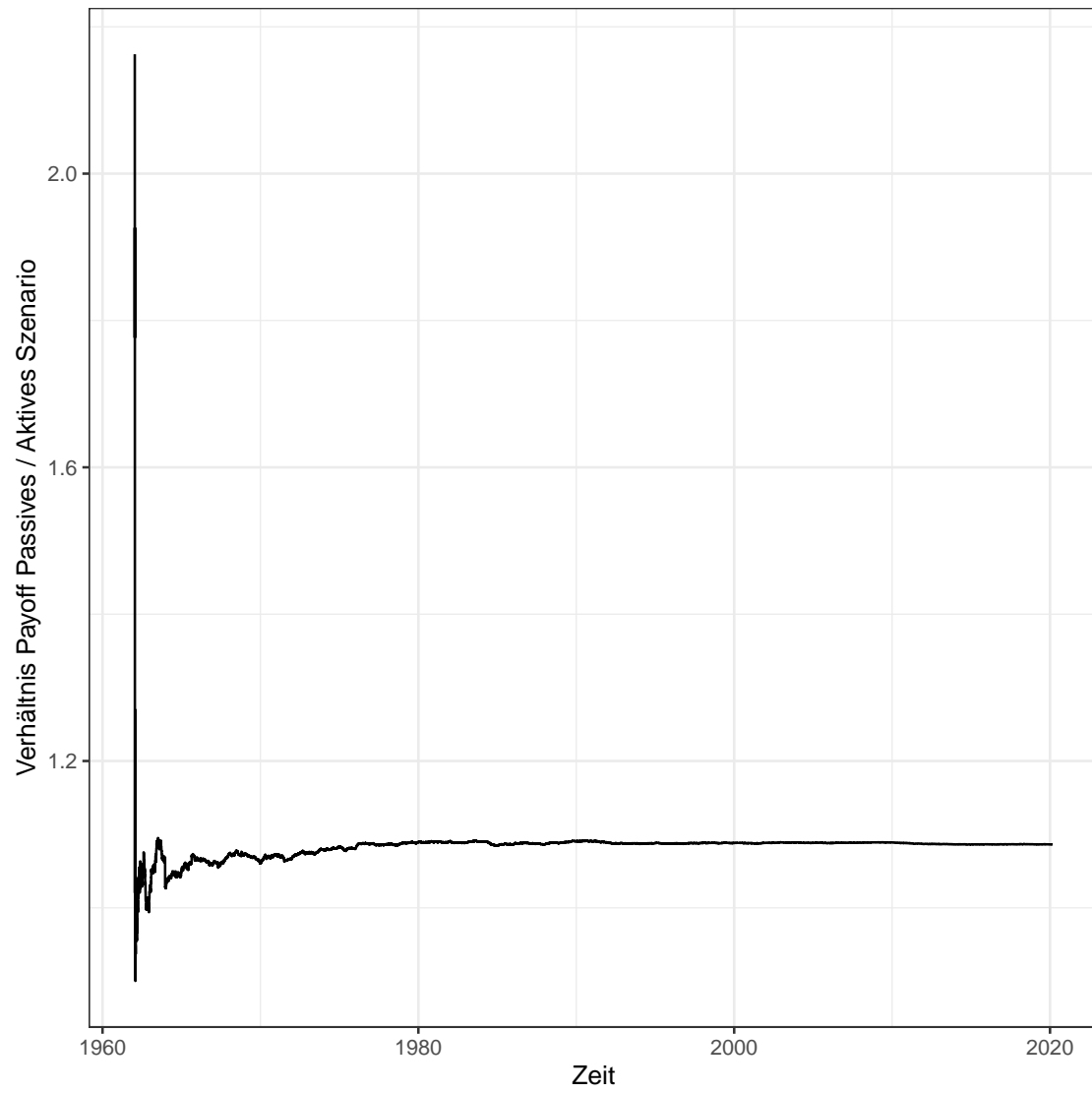


Abbildung 7: Entwicklung der Modellperformance über die Zeit

denkbar. Aus Gründen der Einfachheit verzichten wir an dieser Stelle allerdings darauf und verwenden neben dem gleichgewichteten Mittelwert den Median als zweites Prognosemass. Dieses reagiert weniger sensitiv auf Ausreisser innerhalb der Nachbarn.

Die zweite hier betrachtete Möglichkeit besteht darin, die Kaufs- und Verkaufspreise nicht direkt, sondern als Abweichung vom Eröffnungskurs zu modellieren. Hierzu wird für jeden Nachbarn die Differenz von Eröffnungs- und Tiefstpreis resp. Eröffnungs- und Höchstpreis berechnet. Die entsprechende Metrik (Mittelwert oder Median) wird dann auf diese Werte angewandt und auf den tatsächlichen Eröffnungspreis des zu analysierenden Titels appliziert. Dies hat den Vorteil, dass der bekannte Eröffnungspreis keiner Unsicherheit mehr unterliegt.

Tabelle ?? stellt die Ergebnisse aller Parametrisierungen für des Testingset einander gegenüber. Es zeigt sich, dass eine direkte Prognose der Kaufs- und Verkaufspreise derjenigen einer Prognose der Veränderung im Vergleich zum Eröffnungspreis klar unterlegen ist. Bezüglich Distanzmass lässt sich kein eindeutiger Gewinner feststellen, beide Masse weisen ähnliche Performance aus. Ähnliches gilt für die Anzahl der berücksichtigten Nachbarn.

Allen Ergebnissen gemein ist hingegen, dass sie der einfachen Strategie des Vorkapitels nicht überlegen sind. Im Gegenteil fallen die Payoffs im Vergleich trotz deutlich höherem Berechnungsaufwand in der Tendenz schlechter aus, wenn auch der Payoff der Referenzstrategie weiter deutlich geschlagen wird. Die Hoffnung, dass mit der Individualisierung der einzelnen Einträge mehr kursrelevante Information extrahiert werden kann, bestätigt sich nicht. Ein Grund könnte in der durch die Berechnungskomplexität beschränkte Begrenzung auf ein Zeitfenster von 3 vorangegangenen Tagen sein.

	Euklidisch	
	Mittelwert	Median
5	Error in curr_config\$aggr : \$ operator is invalid for atomic vectors	Error in curr_config\$aggr : \$
20	Error in curr_config\$aggr : \$ operator is invalid for atomic vectors	Error in curr_config\$aggr : \$
50	Error in curr_config\$aggr : \$ operator is invalid for atomic vectors	Error in curr_config\$aggr : \$

4.3 Neuronale Netzwerke

Als dritte Analysemethode sollen im vorliegenden Kapitel neuronale Netzwerke verwendet werden. Diese Methoden erlauben den Umgang mit sehr grossen Datenmengen, weshalb die Beschränkung auf ein sehr kurzes Zeitfenster hier nicht nötig ist. Für die vorliegenden Analysen verwenden wir eine einfache Architektur mehrerer aufeinanderfolgender Dense Layer. Das Ziel der Analyse bleibt hingegen das gleiche wie in den Kapiteln zuvor: Es soll basierend auf vergangenen Kursverläufen möglich optimale Kaufs- und Verkaufskurse für den laufenden Tag prognostiziert werden. Wie bereits früher ausgeführt ist der resultierende Payoff dabei abhängig von der Höhe der Preisbewegung und der der Wahrscheinlichkeit der Ausführung. Da der Payoff weiter in quadratischer Form von der

Höhe der Kursbewegung abhängt haben frühere Überlegungen bereits gezeigt, dass es allenfalls lukrativ sein könnte eher breite Preisschranken zu setzen, welche zwar weniger oft erreicht werden, in diesem Fall aber einen umso höheren Payoff abwerfen. Als erste der betrachteten Methoden sollen nachfolgend daher nicht nur der kommende Höchst- und Tiefstwerte prognostiziert werden, sondern auch deren Verteilung. Das Vorgehen zum Finden optimaler Kaufs- und Verkaufspreise wird dadurch zweistufig. In einem ersten Schritt wird die Verteilung der prognostizierten Tiefst-, Höchst- und Schlusspreise geschätzt.

4.3.1 Diskretisierung

Zur Modellierung der Verteilung werden die zu prognostizierenden zukünftigen Preis (Tiefst-, Höchst- und Schlusspreis) in einem ersten Schritt diskretisiert. Für jeden Preisbereich [Bucket] lässt sich mit dem Modell später eine Eintretenswahrscheinlichkeit berechnen. Wichtig für die ein solches Klassifikationsproblem ist dabei, dass die einzelnen Buckets möglichst gleich viele Werte enthalten. Ist dies nicht der Fall bestehen für den Algorithmus Anreize, ein Element dem grössten Bucket zuzuordnen. Im vorliegenden Fall werden dazu zwei Strategien verfolgt:

1. Unabhängige Modellierung von Low, High und Close Preisen

Eine erste Möglichkeit besteht darin, die Diskretisierung von Tiefts-, Höchst- und Schlusspreisen unabhängig voneinander zu gestalten. Dies hat den Nachteil, dass allenfalls wertvolle Information verloren geht. Dies zeigt sich insbesondere auch dadurch, dass Bucketkombinationen entstehen, welche in Realität nicht möglich sind. Beispielsweise sind dies Kombinationen, bei denen der Tiefstpreis höher vorausgesagt wird als der Tiefstpreis. Gelöst wird dieses Problem in der vorliegenden Analyse so, dass für solche Fälle keine Kaufs- und Verkaufspreise gestellt werden. All dies Fälle werden damit gleich wie im Referenzfall behandelt - namentlich wird damit aufgelaufenes Delta erst zum Tagesendkurs ausgeglichen. Der Vorteil dieses Vorgehens liegt darin, dass die Ermittlung gleich grosser Buckets sehr einfach gelingt. Zudem lässt sich das Problem unter der Annahme der Unabhängigkeit in 3 kleinere Klassifikationsprobleme aufteilen. Werden beispielsweise sowohl Tiefts-, Höchst- und Schlusspreise mit jeweils 30 Buckets modelliert, resultiert dies in 3 Klassifikationsproblemen mit 30 Klassen. Kombiniert man diese, resultieren $27'000 (= 30^3)$ Preisszenarien, wobei sich die Wahrscheinlichkeit für jedes diese Szenarien als Multiplikation der einzelnen Preiswahrscheinlichkeiten ermitteln lässt.

2. Abhängige Modellierung von Low, High und Close Preisen Eine zweite Möglichkeit besteht darin, die Diskretisierung ohne Annahme der Unabhängigkeit der einzelnen Tagespreise zu gestalten. Die Diskretisierung unter dem Ziel möglich gleich grosser Buckets gestaltet sich dabei etwas schwieriger. Wir verwenden dazu den Ansatz, welcher zuerst die Tiefstpreise in gleich grosse Buckets aufteilt. Für jedes dieser Bucket werden danach die Höchstkurse in gleich grosse Buckets aufgeteilt. Die resultierenden Buckets werden danach wiederum in möglichst gleich grosse Buckets bezüglich Schlusspreise auf-

geteilt. Anders als im Unabhängigen Fall sind die Bucket-Grenzen damit nicht immer gleich. Auch das Klassifikationsproblem lässt sich damit nicht mehr auf kleinere Modelle aufteilen. Im obigen Beispiel müssen damit alle 27'000 Szenarien auf einmal bearbeitet werden. Dies erhöht die Komplexität der Bearbeitung. Der Vorteil dieser Methode liegt darin, dass eine Abhängigkeit der Preise ökonomisch plausibler ist und auch unmögliche Szenarien nicht mehr vorkommen.

4.3.2 Architektur

Eine zweite Fragestellung die sich bei der Verwebung von neuronalen Netzen ergibt, ist diejenige nach der geeigneten Architektur. Neuronale Netze haben sich insbesondere im Bereich der Computer Vision - beispielsweise zur Klassifikation des Bildinhaltes - als sehr leistungsfähig herausgestellt. Diese Probleme zeichnen sich dadurch aus, dass mit der Verwendung von Pixel-Daten sehr grosse Datenmengen zur Verfügung stehen und verarbeitet werden müssen. Im Vorliegenden Fall sind die Datenmengen in Relation zu Bilddaten bedeutend kleiner. Die positiven Erfahrungen, welche bei der Bilderkennung mit Convolutional Layers gemacht wurden lassen sich beim vorliegenden Sachverhalt auch nicht direkt übertragen. Aus diesem Grund und aus Überlegungen der Einfachheit entscheiden wir uns daher für eine einfache Architektur mit 2 hidden Dense Layers mit Grösse 512. Diverse Tests zur Erhöhung der Anzahl Knoten oder Gestaltung des Output-Layer dem Beifügen weiterer Layer haben zu keinen wesentlichen Verbesserungen geführt. Neben der Anzahl Layers stellt sich zudem die Frage nach der geeigneten Abbildung des Output Layers. Hierbei wurden zwei Vorgehensweisen untersucht.

1. Klassische Klassifikation

Eine erste Möglichkeit besteht darin, die Ordinalität der Klassen zu vernachlässigen. Eine solche Vernachlässigung macht insbesondere dann Sinn, wenn die einzelnen Klassen in keiner Abhängigkeit zueinander stehen. Wiederum bietet sich dabei der Vergleich mit der Bildklassifikation an - beispielsweise der Klassifikation des Bildes einer Katze. Es lässt sich je nach Verwendungszweck argumentieren, dass es bei einer Fehlklassifikation keine Rolle spielt, ob das Bild als Auto oder als Apfel erkannt wurde. Keines der beiden ist weniger oder mehr falsch, beide sind einfach falsch. Für diese Art der Klassifikation bietet sich eine Verlustfunktion der Art "Categorical Crossentropy" mit Aktivierung "softmax" des Outputlayers an.

2. Ordinale Klassifikation

Im vorliegenden Fall - insbesondere bei unabhängiger Klassifikation von Tiefst-, Höchst- und Schlusspreisen - lassen sich die Buckets in eine logische Reihenfolge bringen. Wird ein Wert von 100 anstatt Klasse 10 fälschlicherweise Klasse 1 zugeordnet, so scheint dieser Fehler grösser als wenn die Fehlklassifikation in Klasse 9 erfolgt wäre. Eine Möglichkeit, diese Art von Klassifikationen in neuronalen Netzen zu modellieren zeigen Frank and Hall (2001). Grob besteht die Idee darin, nicht die Wahrscheinlichkeit der aktuellen Klasse zu modellieren, sondern die kumulierte Wahrscheinlichkeit der aktuellen und der darunterliegenden. Die Wahrscheinlichkeit der Zugehörigkeit zu einer spezifischen Klasse

lässt sich dann einfach als Differenz der kumulierten Wahrscheinlichkeiten benachbarter Klassen ableiten. Zu berücksichtigen gilt es hierbei, dass die kumulative Wahrscheinlichkeit nicht sinken darf, dies durch das geschätzte Modell aber nicht unbedingt garantiert ist. In der praktischen Umsetzung unterscheidet sich die Architektur zur Schätzung dieser Art von Modellen nicht gross von derjenigen der klassischen Klassifikation. Die Unterscheidungen beziehen sich auf eine andere Verlustfunktion (Binary Crossentropy), andere Aktivierung (Sigmoid) des Output Layers sowie eine etwas anderen Kodierung der Labels.

4.3.3 Ergebnisse

Alle der nachfolgenden Modelle wurden mit 30 Buckets im Falle unabhängig modellierter Preise, respektive 27'000 Preisszenarien im Falle abhängiger Preise gerechnet. Ebenfalls allen Modellen gemein ist, dass eine Batch-Grösse von 512 und ein Training über 10 Epochen gewählt wurde. Ferner werden als Features alle 4 Preiskennzahlen der letzten 10 Tage plus der Eröffnungskurs des aktuellen Tages verwendet. Jedes dieser Features wird vor dem Training skaliert.³ Das eigentliche Training erfolgt auf 80% des Trainingssets, 20% der Trainingsdaten dienen der Validierung.

4.3.3.1 Unabhängige Modelle

```
## [1] "col_id: 1"
## [1] "curr_group: 0"
## [1] "col_id: 1"
## [1] "curr_group: 0"
## [1] "col_id: 1"
## [1] "curr_group: 0"
```

Während des Trainings der Netze für Tiefst-, Höchst und Schlusskurs zeigt sich der in Abbildung 8 visualisierte Lernfortschritt. Als Genauigkeit wird hierbei der Anteil welcher dem richtigen Bucket zugeordnet wurde angegeben. Bei der Betrachtung des Lernprozesses lassen sich folgende Erkenntnisse gewinnen:

- Die Genauigkeit richtig zugeordner Klassen liegt deutlich über demjenigen, welcher bei zufälliger Zuteilung erreicht werden könnte. Im vorliegenden Fall von 30 Buckets bei ungefähr gleichverteilter Anzahl Datenbunkte wäre dies lediglich 3.3%.
- Die Genauigkeit der Prognose für Tiefst- und Höchstkurs ist deutlich höher als diejenige des Schlusskurses.
- Über den Trainingsverlauf nimmt die Genauigkeit zu. Dies gilt sowohl für die Trainings- wie auch die Validierungsdaten. Ein Overfitting ist damit im vorliegenden Fall nicht auszumachen.

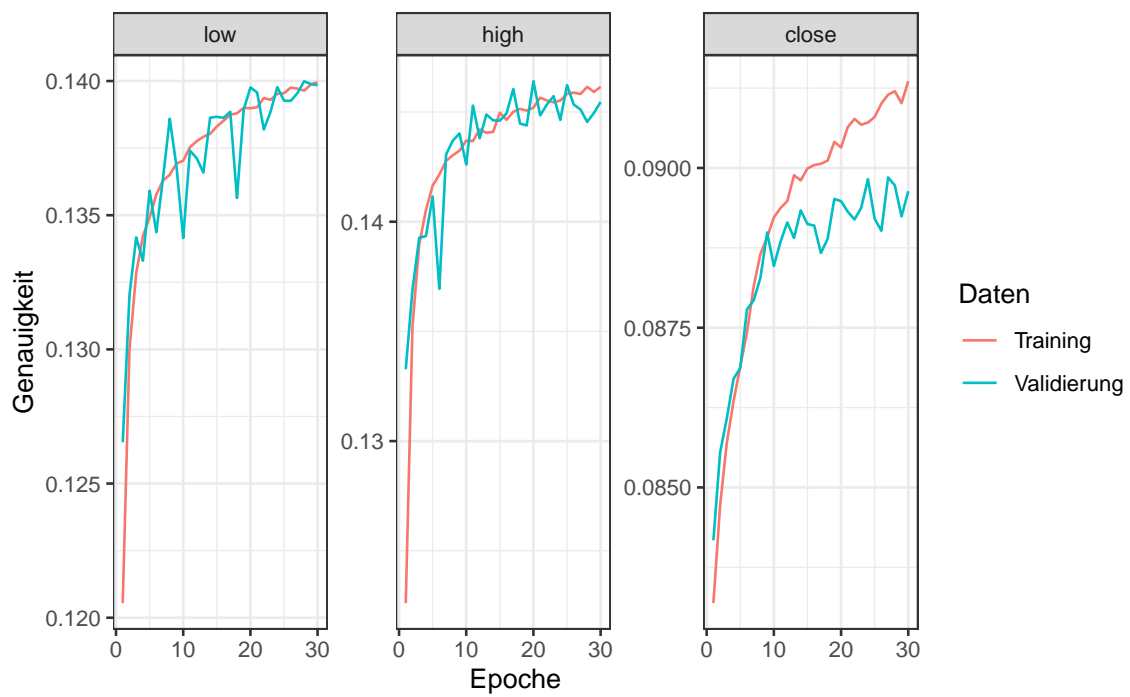


Abbildung 8: Trainingsfortschritte im Falle unabhängiger Modelle

Mit Hilfe der trainierten Modelle lassen sich nun Voraussagen über die Wahrscheinlichkeitsverteilungen der Daten im Testset machen. Für einzelne Beobachtungen lassen sich diese gar visualisieren. Abbildung 10 zeigt die prognostizierten Verteilungen zweier exemplarischen Einträge im Testset. Diese unterscheiden sich deutlich. Während für ersteren Eintrag ein wenig volatiler Kursverlauf prognostiziert ist, erwartet das Modell im zweiten Fall eine deutlich volatilere Entwicklung.

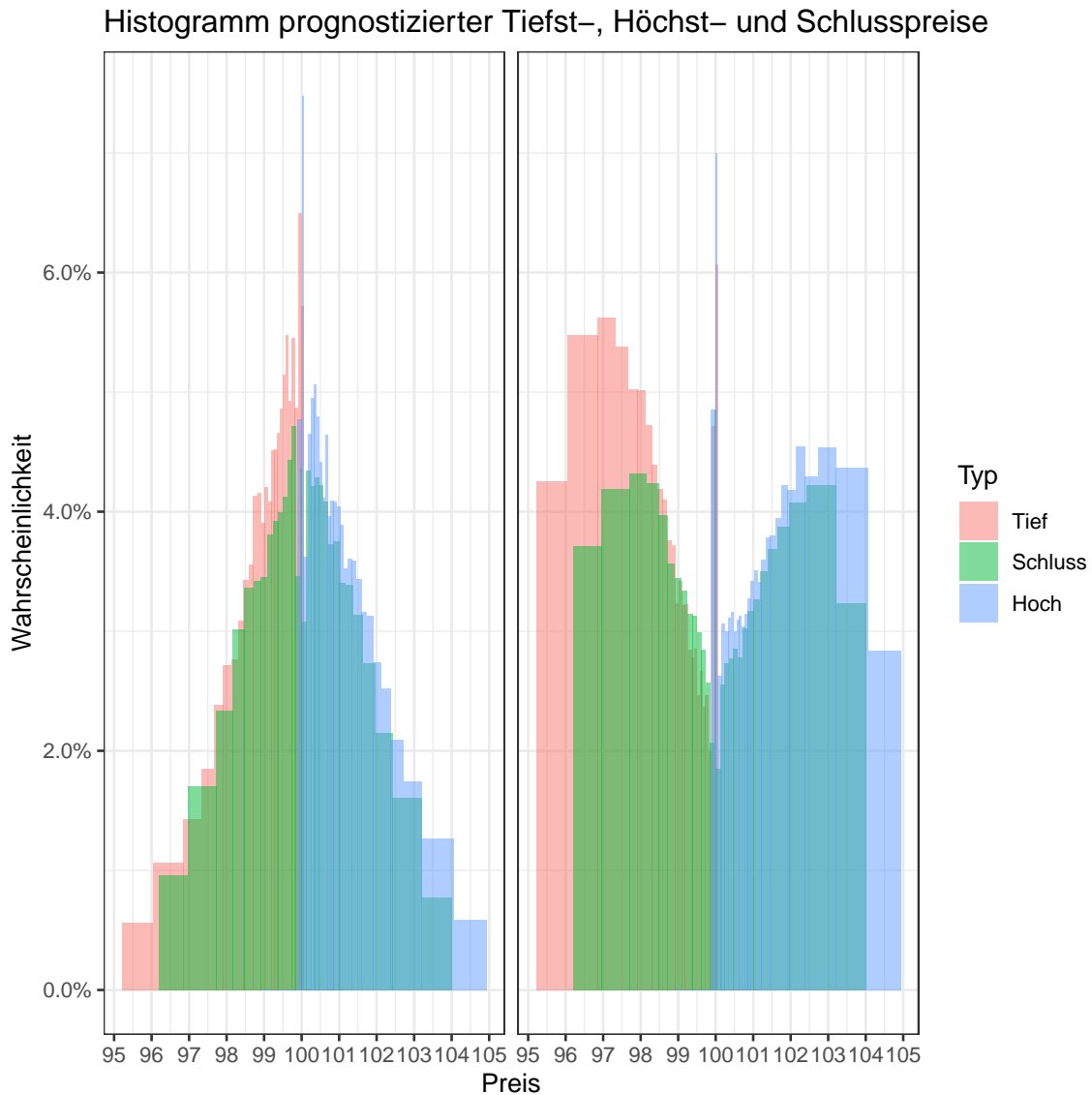


Abbildung 9: Exemplarische Verteilung der vorausgesagten Preise

³Im Detail erfolgt die Skalierung auf Basis des Mittelwertes und Standardabweichung jedes Features des Trainingsets.

Tatsächlich unterscheiden sich die realisierten Tiefts- und Schlusskurse, wenn auch nicht so deutlich wie dies vom Modell prognostiziert wird. Die entsprechenden Realisierungen für Low und High sind 99.37 und 100.83 für das erste und 98.75 und 101.5 für das zweite Beispiel.

Zur Überprüfung der Modelleignung soll erneut der Payoff des Testsets ermittelt und mit demjenigen der Referenzstrategie verglichen werden. Eine erste Möglichkeit besteht darin, die Kaufs- resp. Verkaufsschranken als Tiefst- respektive den Höchstpreis desjenigen Buckets vorauszusagen, welchem das Modell die höchste Wahrscheinlichkeit vorhersagt. [Da ein Bucket durch untere und obere Grenze bestimmt ist, verwenden wir den Mittelwert von oberer und unterer Grenze als Vorhersagewert. Ist eine der Grenzen nicht finit, wird die andere Grenze als Vorhersagewert verwendet.]

Es resultiert ein Überschusspayoff von 9.8%. Bereits in dieser einfachen Form ist das gelernte Modell damit gleich gut gut, wie das Model mit Kaufs- und Verkaufsschranken als symmetrischer Abweichungen vom aktuellen Eröffnungskurs bei Unterscheidung von Hoch- und Tiefvolatilitätsphasen. Bei der Konzeption des Modelles war es gerade das Ziel, nicht nur eine Klassifikation in das wahrscheinlichste Bucket zu machen, sondern auch Voraussagen über die Verteilung der prognostizierten Preise machen zu können. Lediglich das Bucket mit höchster Wahrscheinlichkeit als Prognosewert zu verwenden ging damit zuwenig weit. Viel können nun verschiedene Kaufs- und Verkaufskombinationen für alle Preisszenarien berechnet und mit der jeweiligen Wahrscheinlichkeit gewichtet werden. Die optimale Strategie lässt sich dann diejenige mit dem höchsten erwarteten Payoff auswählen. Hierbei lohnt wiederum ein Blick auf die Berchnungskomplexität des Problems. In der vorliegenden Analyse wurden je Preistyp 30 Buckets verwendet. Dies resultiert in 27'000 möglichen Preisszenarien. Diese können für jeden Eintag des Testsets (rund 1'000'000 Einträge) ermittelt werden. Für jedes dieser Preisszenarien sollen wiederum verschiedene Paare von Kaufs- und Verkaufspreisen getestet werden. Orientiert man sich dabei an einer ähnlichen Granularität wie für die Preise resultieren 900 Szenarien für die Schranken (je 30 Werte für die Kaufs- und Verkaufsschranke). Damit müssen zur kompletten Evaluation $27'000 \times 1'000'000 \times 900$ Payoffs berechnet werden. Dies ist mit Hilfe der in Kapitel @ref(infrastruktur_und_tools) vorgestellten Cloud-Infrastruktur und Tools nicht innerhalb kurzer Zeit möglich. Zur Reduktion der Komplexität lässt sich dabei ausnützen, dass nicht alle Szenarien von gleicher Bedeutung sind. Während die Verwendung des häufigsten Szenarios und die Verwendung aller Szenarios die beiden Extrempositionen einnehmen, ist auch die Verwendung einiger wichtiger Preis- und Schrankenszenarien denkbar. Konkret werden diejenigen Szenarien ausgewählt, welche die höchsten Eintretenswahrscheinlichkeiten aufweisen. Dabei hat es sich bewährt, diese Grenze als Quantil der jeweiligen Wahrscheinlichkeiten zu wählen. Damit werden nur sehr wenige, dafür wichtige Szenarien berücksichtigt. Die Berechnungszeits lässt sich damit sehr deutlich senken. Diese Vorgehensweise hat zudem den Vorteil, dass mit einer Reduzierung des Quantil sehr einfach mehr Werte einbezogen werden können. Ferner wird auch die Schrankenszenarien sehr stark reduziert, indem als Kaufs- und Verkaufspreise nur die Tiefs- und Höchstwerte der betrachteten Preisszenarien berechnet werden.

Das beschriebene Vorgehen sowohl für unabhängige wie auch abhängige Modelle möglich. Bei den unabhängigen Modellen gibt es ferner zusätzlich zu beachten, dass nicht mögliche Szenarien (bsp. prognostizierter Höchstpreis < prognostizierter Tieftspreis) entfernt, resp. deren prognostizierte Wahrscheinlichkeiten auf 0 gesetzt werden. Bei den Modellen mit voneinander abhängigen Preisen treten solche Szenarien nicht auf.

Bezieht man für die Pronose der Preisschranken alle Buckets mit 0.1% höchster Wahrscheinlichkeit mit ein (99.9% Quantil) resultiert für den Testdatensatz ein Überschusspayoff von 12% gegenüber der Referenzstrategie. Durch Berücksichtigung (eines Teils) der geschätzten Verteilung kann dieser damit noch einmal um 2.2 Prozentpunkte gesteigert werden und übertrifft damit die Performance einfacherer früherer Modelle.

Wie in Abbildung @ (fig:plot-binary-histogram) gezeigt schien das Modell auch ohne explizite Modellierung der Ordinalität der Klassen diese implizit gelernt zu haben. Dies zeigt sich dadurch, dass benachbarte Klassen typischerweise auch ähnliche Wahrscheinlichkeiten aufweisen. Eine explizite Modellierung der Ordinalität scheint damit im vorliegenden Fall nicht unbedingt nötig. Dies zeigt sich auch beim Vergleich der Ergebnisse mit ebendiesem Modell.

4.3.3.2 Abhängige Modelle

5 Fazit und Ausblick

Literatur

- Bentley, J. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.
- Blackrock (2020). *iShares MSCI World UCITS ETF: IWRD*. <https://www.ishares.com/ch/individual/en/products/251881/ishares-msci-world-ucits-etf-inc-fund> [Abgerufen am: 19. März 2020].
- Frank, E. and Hall, M. (2001). A simple approach to ordinal classification. volume 2167, pages 145–156.
- Nestle (2020). *Dividendenzahlungen*. <https://www.nestle.com/investors/individual-shareholders/dividends/dividenden> [Abgerufen am: 21. März 2020].
- SP-Finance (2020). *Delta Hedging, Gamma and Dollar Gamma*. <http://sp-finance.e-monsite.com/pages/the-greeks/gamma/delta-hedging-gamma-and-dollar-gamma.html> [Abgerufen am: 29. April 2020].
- Yahoo (2020). *Yahoo Finance*. <https://finance.yahoo.com/> [Abgerufen am: 21. März 2020].
- Yang, D. and Zhang, Q. (2000). Drift-independent volatility estimation based on high, low, open, and close prices. *The Journal of Business*, 73(3):477–492.

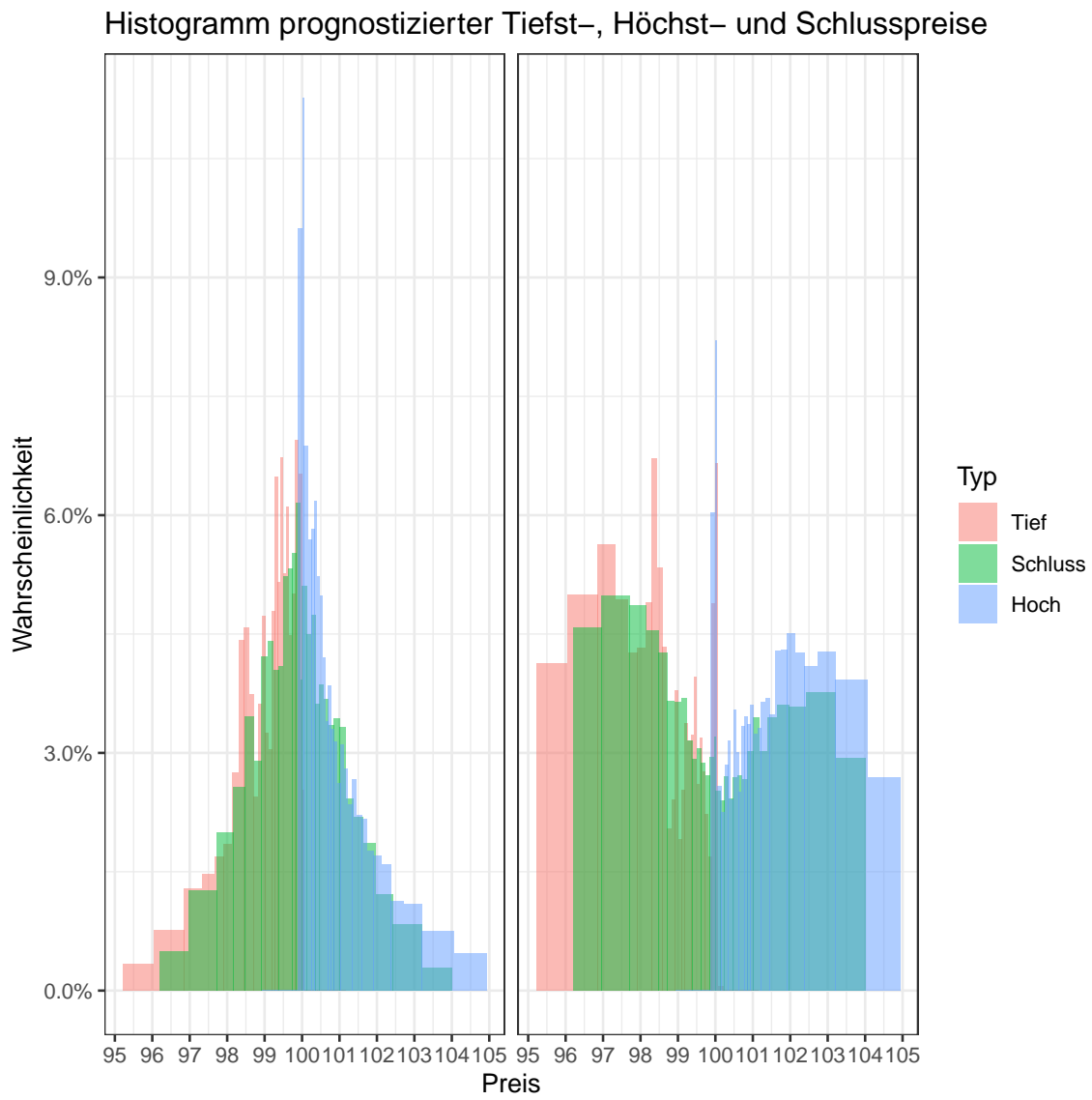


Abbildung 10: Exemplarische Verteilung der vorausgesagten Preise