

HOCHSCHULE MANNHEIM

ALGORITHMEN FÜR MODERNE
RECHNERARCHITEKTUREN

Methoden zur Börsenkursvorhersage

Autoren:

Marc MISOCH

David MARQUANT

June 21, 2015

Contents

1	Abstract	2
2	Einleitung	2
2.1	Einführung in die Börsenkursvorhersage	2
2.2	Motivation	3
3	Grundlagen	4
4	Algorithmen	5
4.1	Latest Trend Algorithmus	5
4.1.1	Umsetzung	5
4.1.2	Ergebnisse	6
4.2	Moving Average Algorithmus	7
4.2.1	Umsetzung	7
4.2.2	Ergebnisse	8
4.3	Überwachtes Lernen	9
4.3.1	Umsetzung	10
4.3.2	Ergebnisse	10
4.4	Output Only Neural Network	11
4.4.1	Umsetzung	12
4.4.2	Ergebnisse	12
5	Fazit	12
6	Ausblick	12

1 Abstract

In diesem Paper geht es darum, wie man Börsenkurse vorhersagen kann. Dies ist kein leichtes Unterfangen, denn die Wertpapierbörse ist ein hoch komplexes Gebilde mit unglaublich vielen Abhängigkeiten. All diese Abhängigkeiten ausfindig zu machen ist unmöglich. Deshalb muss man sich mit anderen Methoden behelfen. Die von uns getestete und implementierten Algorithmen benutzen ausschließlich alten Börsenkurse um vorhersagen zu treffen. Wir haben uns für diese Methode entschieden, da wir einen gut funktionierenden Algorithmus finden wollten ohne andere Variablen wie Nachrichten mit einzubeziehen, da man funktionierende Algorithmen in dieser Hinsicht relativ leicht erweitern könnte.

Im Verlauf des Papers werden wir genauer auf die verschiedenen Algorithmen eingehen und diese im Detail erläutern. Außerdem werden wir unsere Testergebnisse vorstellen und versuchen möglichst gute Schlüsse aus diesen zu ziehen. Um eine Sache schon einmal vorweg zu nehmen, es gibt weltweit keinen Algorithmus der Börsenkurse zu 100% vorhersagen kann. Dennoch ist es möglich durch hochkomplexe und geheime Algorithmen eine Genauigkeit von 60-65% zu erreichen.

2 Einleitung

2.1 Einführung in die Börsenkursvorhersage

Es gibt eine große Vielfalt an Methoden, die dafür genutzt werden Börsenkurse möglichst genau vorherzusagen. Viele dieser Algorithmen beruhen ausschließlich auf historischen Preisen der Aktie die vorhergesagt werden soll. Selbst wenn man nur diese Daten berücksichtigt kann man gute Ergebnisse erzielen. Jedoch sind diese Ergebnisse sehr anfällig gegen Kurzfristige Veränderungen durch unvorhersehbare Umwelteinflüsse.

Dennoch verfolgen all diese Algorithmen ein Hauptziel: **Den Profit zu maximieren während das Risiko so niedrig wie möglich gehalten wird.**

Da der Aktienmarkt global ist, gibt es viele Beziehungen zwischen den einzelnen Börsen weltweit und somit auch zwischen den einzelnen Aktien. Manche Aktienkurse hängen zusammen und werden stark von einander beeinflusst. Dies kann man an folgendem Beispiel leicht erläutern:

Der Aktienkurs eines Motorenherstellers verliert an Wert wenn der Aktienkurs der Autoherstellers, den er beliefert, an Wert verliert. Die meisten dieser Abhängigkeiten sind Wechselseitig und sehr stark. Dennoch sind nicht alle dieser Abhängigkeiten auf den ersten Blick so klar erkennbar. Selbst zum heutigen Zeitpunkt ist nicht immer klar wie sich der globale Markt verhalten wird, wenn es zu großen Veränderungen an einer einzigen Aktie kommt. Das lässt den Schluss zu, dass es für eine möglichst genaue und robuste vorhersage mehr braucht als nur historische Aktienkurse. Dennoch ist es möglichst selbst mit diesen Daten gute Vorhersagen zu machen.

Eine weitere wichtige Sache die man im Auge behalten muss sind die Nachrichten. Wenn die globale Wirtschaft einen negativen Trend zeigt, ist davon auszugehen, dass der Aktienmarkt diesem Trend folgt und auch einen Abstieg zu verzeichnen hat. Jedoch sind diese Trends eher langsam und man kann sie auch mit Algorithmen, deren vorhersagen auf alten Aktienkursen basieren, relativ gut vorhersehen. Eine Sache die man nicht anhand von alten Börsenkursen vorhersagen kann, sind Katastrophen. Dabei macht es wenig bis keinen Unterschied ob es sich um Naturkatastrophen oder vom Menschen verursachte Katastrophen handelt. Falls so ein Fall eintritt kann der globale Aktienmarkt innerhalb von wenigen Minuten einen enormen Absturz erfahren. Das führt dazu, dass eine wirklich sichere Vorhersage nur mit Algorithmen möglich ist, die auch die Nachrichten nach Neuigkeiten durchsuchen, die den Aktienmarkt betreffen werden.

All unsere Tests wurden mit echten historischen Aktienkursen durchgeführt. Dennoch haben wir nicht an der Wertpapierbörse gehandelt. Die Aktien mit denen wir gearbeitet haben wurden von uns nie reell an der Börse gekauft und hatten somit hatten unsere Tests auch keinen Einfluss auf den Aktienmarkt.

2.2 Motivation

Dieses Paper dient dazu, verschieden Algorithmen und ihre Arbeitsweise zu vergleichen. Dabei werden wir sowohl auf die Implementierung selbst als auch auf die Überlegungen die dahinter stehen eingehen. Das Hauptziel all dieser Algorithmen ist das selbe. Es soll ein Anstieg oder Abfall eines Aktienkurses vorhergesagt werden, noch bevor dieser passiert. Dafür ist es sehr wichtig ein geeignetes Muster zu finden, mit dessen Hilfe man von alten Aktienkursen auf zukünftige schließen kann. Im Anschluss werden wir auf die Tests eingehen die wir mit den Implemen-

tierten Algorithmen durchgeführt haben. Die Algorithmen die wir in diesem Paper ausführen möchten, benutzen nur Aktienpreise der Vergangenheit und versuchen mit deren Hilfe eine möglichst genaue Vorhersage zu treffen. Dafür haben wir uns entschieden, da wir Algorithmen finden wollten die in einem isolierten system gute vorhersagen treffen können. Diese Algorithmen dann noch um einen Teil zu erweitern der die Nachrichten durchsucht, wäre ein weiterer Schritt um die Algorithmen noch besser zu machen, und im Nachhinein nicht sonderlich kompliziert.

All unsere Testergebnisse werden wir später in dem Paper vorstellen. Außerdem werden wir aufzeigen mit welchen Parametern wir jeweils die besten Ergebnisse erzielen konnten.

3 Grundlagen

Wir untersuchten nur Algorithmen, die Börsenkurse mit Hilfe von Zeitreihenanalyse vorhersagen können. Alle Algorithmen die wir implementierten waren darauf spezialisiert, unter Berücksichtigung alter Börsenkurse eine möglichst genaue Vorhersage zu treffen. Diese Algorithmen wurden dann anhand von realen Preisen an der Börse getestet. Jedoch wurden die Aktien nicht real an der Börse gehandelt sondern nur virtuell.

Die erste Aufgabe die sich uns stellte, war diese Börsenkurse aus einer verlässlichen Quelle beziehen zu können. Außerdem musste gewährleistet sein, dass die Kurse immer auf dem aktuellen Stand und in einem möglichst geeigneten Format vorlagen. Nach einer Recherche im Internet bekamen wir immer wieder die gleichen Hinweise auf eine Verlässliche Quelle und zwar Yahoo Finance. Bei Yahoo Finance gibt es alle Aktienkurse die wir für unsere Zwecke benötigten. Außerdem ist es durch eine von Yahoo bereitgestellte API sehr einfach sich diese Kurse auch herunterzuladen. Das einzige was man tun muss, ist einen Link zu bauen, der die richtigen Daten herunterlädt. Dieser Link kann dann für verschiedene Zwecke angepasst werden.

`http://real-chart.finance.yahoo.com/table.csv?s=IBM&a=00&b=2&c=1962&d=06&e=18&f=2015&g=d&ignore=.csv`

Hier kann man sehen wie der Link aufgebaut ist. Hinter dem Parameter "s" muss man die Abkürzung der gewünschten Aktie eintragen. Die Parameter a, b und c spiegeln das Startdatum wieder. Lässt man hier den 2 Januar 1962 konstant

stehen, bekommt man jeden Aktienkurs seit dem ersten Tag der Aufzeichnung. Die nächsten 3 Parameter stehen für das Enddatum. Danach kann man noch weitere Parameter angeben, wie beispielsweise die Intervalle in der der Kurs in die resultierende .csv-Datei geschrieben wird. Mit dieser Methode kamen wir immer schnell an die Börsenkurse die wir benötigten.

4 Algorithmen

Im Allgemeinen gibt es zu sagen, dass all unsere Algorithmen die Daten benutzen, die wir mit Hilfe der Yahoo Finance API heruntergeladen haben. Zuerst implementierten wir einen Algorithmus in Java, mit dem man sich den gewünschten Link zur Laufzeit zusammenbauen kann und der diesen dann nutzt um mit Hilfe der eingebauten Pakete *java.net.URL* und *java.net.URLConnection* die Daten herunterzuladen. Diese heruntergeladenen Daten werden dann in getrennte Listen geschrieben. Eine dieser Listen enthält dann beispielsweise alle Close-Prices einer Aktie im durch den Yahoo Finance Link definierten Zeitraum. Mit diesen Listen kann man dann alle gewünschten Operationen ausführen. Da wir uns aber auch mit Python beschäftigten, weil sich damit Grafiken besser zeichnen lassen, fanden wir später heraus, dass es in Python schon ein eingebautes Paket gibt mit dem man die Yahoo Finance API einfacher nutzen kann als in Java.

4.1 Latest Trend Algorithmus

Mit diesem Algorithmus wollten wir einen möglichst genauen Preis für eine bestimmte Aktie am nächsten Tag errechnen. Außerdem wollte wir herausfinden, für welchen Zeitraum in der Zukunft noch gute Prognosen möglich sind. Es ist natürlich klar, dass je weiter man in die Zukunft geht desto ungenauer wird die Vorhersage. Außerdem muss darauf aufmerksam gemacht werden, dass dieser Algorithmus sehr gute Ergebnisse erzielt wenn der Markt sehr ruhig verläuft und es keine plötzlichen Einbrüche gibt.

4.1.1 Umsetzung

Diesen Algorithmus implementierten wir in Java. Da wir uns diesen Algorithmus selbst überlegt hatten, mussten wir erst einmal die genauen Formeln aufstellen um

mit dem Implementieren zu beginnen. Dabei kamen wir auf folgendes:

$$M = \sum_{n=0}^D C_{n+1} - C_n$$

$$prediction = \frac{M}{D}$$

$$stock_{predicted} = stock_{today} + prediction$$

Hierbei ist D die Anzahl der Tage die in die Vorhersage mit einbezogen werden und C der Closing-Price am Tag n.

Nachdem wir die Formel erst einmal aufgestellt hatten war die Implementierung nicht mehr allzu kompliziert. Wie man anhand der Formel sehen kann, kann man diesen Algorithmus eigentlich unbegrenzt in die Zukunft weiterführen. Man muss dann lediglich die anderen vorhergesagten Werte in die neue Vorhersage mit einbeziehen. Im Ergebniskapitel werden wir genauer darauf eingehen für wie viele Tage in die Zukunft man mit diesem Algorithmus gute Ergebnisse erzielen kann.

4.1.2 Ergebnisse

Mit diesem Algorithmus versuchten wir den exakten Wert einer Aktie für n Tage in der Zukunft vorherzusagen. Um aussagekräftige Resultate zu bekommen, testeten wir unseren Algorithmus nicht an einer einzigen Aktie sondern bildeten Sets von 150 verschiedenen zufälligen Aktien. In der nachfolgenden Tabelle ist der Mittelwert der Abweichungen zwischen Vorhersage und reellem Aktienpreis aufgelistet.

Days in the future	Set 1	Set 2	Set 3
Day 1.	0.043 \$	0.012 \$	0.028 \$
Day 2.	0.049 \$	0.010 \$	0.033 \$
Day 3.	0.059 \$	0.013 \$	0.041 \$
Day 4.	0.089 \$	0.017 \$	0.038 \$
Day 5.	0.092 \$	0.024 \$	0.049 \$
Day 6	0.126 \$	0.035 \$	0.062 \$
Day 15.	1.740 \$	0.092 \$	0.107 \$

Tabelle 1. Unterschied zwischen Vorhersage und den realen Daten

Da der Börsenhandel global ist, werden die Preise hier in USDollar angegeben. Auch wir haben all unsere Berechnungen in USDollar angestellt um eine gute Vergleichbarkeit zu erreichen.

Man kann deutlich erkennen das die Vorhersagen für den Zeitraum der ersten drei bis vier Tage durchaus gut ist. Eine durchschnittliche Abweichung von höchstens 0.1 \$ halten wir für eine gute Vorhersage. Wenn man sich jedoch die letzte Zeile der Tabelle anschaut wird klar, dass eine Vorhersage für mehr als diesen kurzen Zeitraum von drei bis vier Tagen eher ein Glücksspiel ist. Man kann sehen, dass bei Set zwei und drei die vorhersagen auch nach 15 Tagen noch im Rahmen von ca 0.1\$ liegen. Bei Set eins jedoch kam es zu einer mittleren Abweichung von 1.74\$ was wir als nicht akzeptabel einstufen. Während unserer Test probierten wir viele verschiedenen Anzahl an Input-Tagen aus. Wir hatten die besten Ergebnisse mit diesem Algorithmus, wenn wir die letzten 15 Tage als Anhaltspunkte für den Algorithmus genommen haben. Dieser Zeitraum scheint die perfekte Mischung aus: Erkennen von längerfristigen Trends und herausfiltern von kurzfristigen Verschiebungen zu sein. Abschließend ist zu sagen, dass dieser Algorithmus seinen Zweck einer kurzfristigen Vorhersage durchaus erfüllt. Eine längerfristige Vorhersage jedoch lässt sich mit diesem Algorithmus nur sehr schwer bis gar nicht treffen.

4.2 Moving Average Algorithmus

Dieser Algorithmus ist anders als der Latest Trend Algorithmus nicht dafür geeignet den exakten Aktienkurs vorherzusagen. Viel mehr ist er dafür gemacht, Entscheidungen zu treffen wann eine Aktie gekauft und verkauft werden soll. Er ist außerdem eher auf eine langfristige Anlage ausgelegt und sehr anfällig gegenüber kurzfristigen Kurseinbrüchen.

4.2.1 Umsetzung

Diesen Algorithmus implementierten wir zunächst auch in Java. Dann wurde uns aber bewusst, dass es viel anschaulicher wäre wenn man auch anhand eines plots sehen könnte was genau passiert. Da wir uns mit plotten in Python besser auskannten als in Java, schrieben wir den Algorithmus noch einmal in Python. Um diesen Algorithmus umzusetzen benötigt man zunächst einmal zwei verschiedene Berechnungen. Zum einen muss man den Moving Average der letzten 50 und

200 Tage berechnen. Falls der Moving Average der letzten 50 Tage über dem der letzten 200 Tage liegt soll man die Aktie zum aktuellen Preis kaufen. Fällt jedoch der Moving Average der letzten 200 Tage unter den der letzten 50 soll man die Aktie wieder verkaufen. Zum besseren Verständnis dient das folgende in Python gezeichnete Schaubild.



Figure 1: Moving Average MSFT

Die blaue Linie auf diesem Schaubild ist der Preis der Aktie. Die rote Kurve der Moving Average der letzten 50 Tage und die grüne Kurve der Moving Average der letzten 200 Tage. Immer wenn sich die Moving Average Kurven schneiden sieht man entweder einen grünen oder einen roten Punkt. Bei einem grünen Punkt soll die Aktie gekauft und bei einem roten verkauft werden.

4.2.2 Ergebnisse

Auch diesen Algorithmus wollten wir nicht nur an einzelnen Aktien testen da uns robuste Ergebnisse wichtig waren. In der ersten Spalte unserer Ergebnistabelle stehen die Aktienkurse mit denen der Algorithmus getestet wurde. Die Random Combinations bestehen aus 100 zufällig gewählten Aktienkursen.

Stock(s) used	Outcome
MSFT ¹	13.39 \$
MCD ²	2.71 \$
YHOO ³	20.05 \$
NVDA ⁴	-13.70 \$
INTC ⁵	5.70 \$
AMZN ⁶	128.87 \$
Random combination 1.	34.53 \$
Random combination 2.	-17.32 \$
Random combination 3.	56.94 \$
Random combination 4.	15.74 \$

Tabelle 2. Ergebnisse der Moving Average Algorithmus

In dieser Tabelle kann man sehen, dass wir durchaus gute Ergebnisse mit diesem Algorithmus erzielt haben. Diese Tabelle beruht auf den Aktienkursen der letzten 1000 Tage. Wenn wir uns nur die Random Combinations anschauen, stellen wir fest das wir insgesamt 89.89\$ Gewinn gemacht haben. Jedoch ist dieses Positive Ergebnis keine Garantie dafür, das der selbe Test noch einmal so positiv ausfallen würde. Dennoch sind wir zufrieden mit unseren Testergebnissen.

4.3 Überwachtes Lernen

Das Ziel war es verschiedene Techniken des Überwachten Lernens zu testen und zu vergleichen. Wichtig dafür war es geeignete Features als Eingabe für die Algorithmen zu finden. Die verwendeten Merkmale waren Return Werte und Moving Averages der Return Werte.

$$Return_i = \frac{Close_i - Close_{i-n}}{Close_{i-n}}$$

$$MA_t = \frac{\sum_{n=0}^t Return_{i-n}}{t}$$

¹Microsoft CorporationTM

²McDonald's CorpTM

³Yahoo! Inc.TM

⁴NVIDIA CorporationTM

⁵Intel CorporationTM

⁶Amazon.com Inc.TM

4.3.1 Umsetzung

Zur Umsetzung haben wir Python verwendet, da die Python Bibliothek scikit-learn das Arbeiten mit Lernalgorithmen besonders einfach gestaltet. Außerdem ist das arbeiten mit Daten mithilfe von weiteren Bibliotheken wie pandas, scipy und numpy besonders komfortabel. Des weiteren gibt es in pandas Funktionen zum Herunterladen von Daten mit Yahoo Finance.

Die getesteten Algorithmen sind:

- k-nearest neighbour
- Linear Discriminant Analysis
- Logistic Regression
- Quadratic Discriminant Analysis
- Support Vector Classification

Als Features wurden die Returns zu gestern und vorgestern verwendet und der Moving Average der Returns der letzten 5 Tage.

4.3.2 Ergebnisse

	MSFT	^AXJO	^FCHI	^FTSE	^HSI	^IXIC	^N225
KNN	0.43	0.47	0.48	0.45	0.51	0.51	0.49
LDA	0.53	0.50	0.54	0.53	0.52	0.57	0.55
LR	0.53	0.50	0.54	0.53	0.52	0.57	0.54
QDA	0.55	0.54	0.51	0.52	0.50	0.58	0.53
SVC	0.50	0.48	0.53	0.49	0.46	0.58	0.48

Tabelle 3. Ergebnisse mit überwachtem Lernen

Wie in Tabelle 3. zu sehen ist konnten wir mit keinem der Algorithmen genaue Ergebnisse hervor sagen. Um akkurate Vorhersagen treffen zu können, wäre es nötig genauer zu untersuchen welche Features sinnvoll sind.

4.4 Output Only Neural Network

Von Herrn Fischer haben wir ein Java Programm erhalten, welches mithilfe eines neuronalen Netzwerkes versucht zwei Spiralen voneinander trennen. Dieses haben wir versucht so umzubauen, dass damit Börsentrends für den nächsten Tag geschätzt werden können.

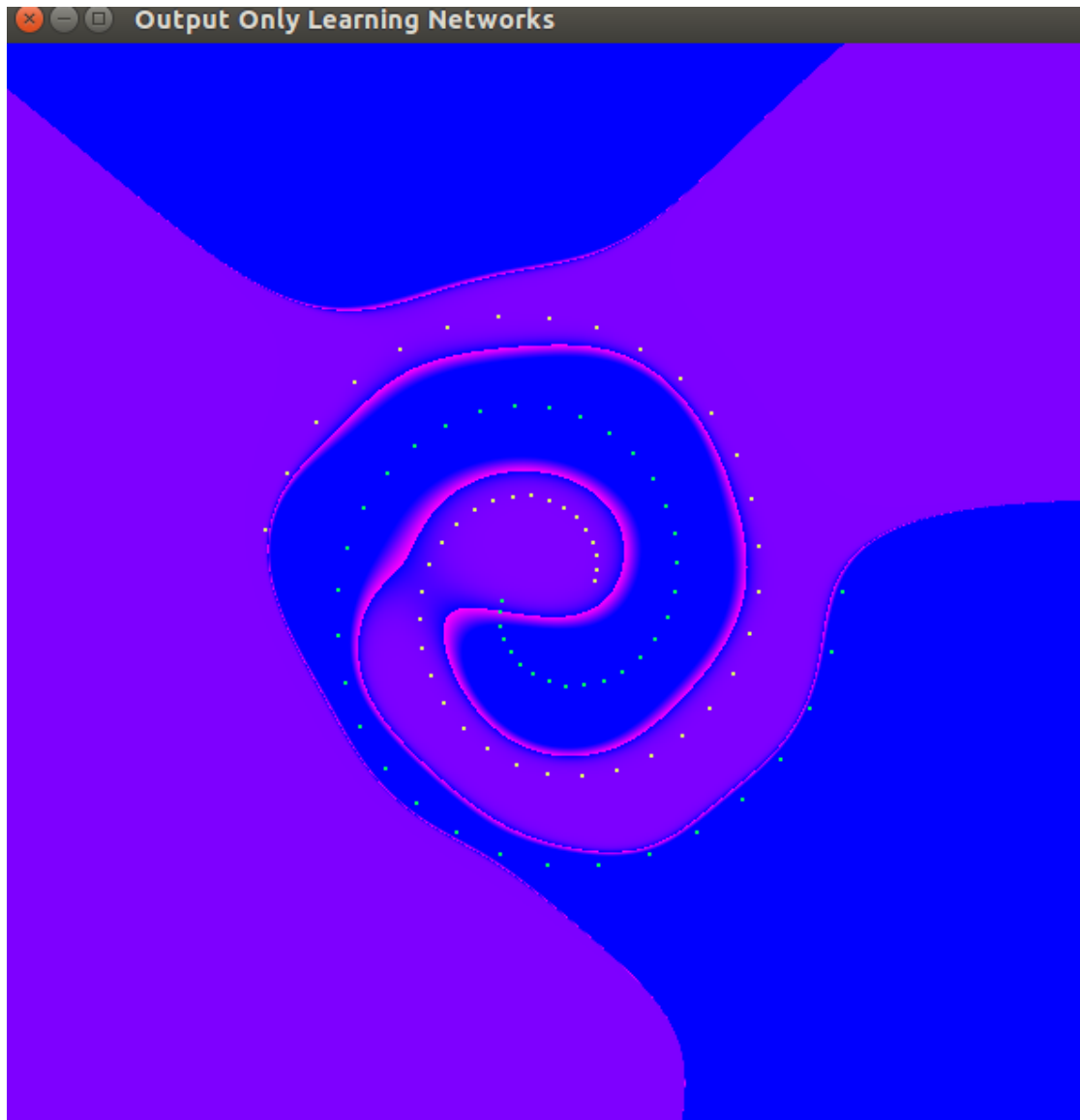


Figure 2: Output Only Neural Network: Spiralen

4.4.1 Umsetzung

Zuerst mussten die Daten heruntergeladen und auf das vorgegebene Eingabe Format gebracht werden. Als Eingabe für das Netz benötigten wir Werte zwischen 0.0 und 1.0. Deswegen wurden die Close Preise erst normalisiert bevor wir sie in einer CSV Datei dem Programm übergaben. Den Download und die Daten Transformationen implementierten wir in Python.

Das Programm wurde so abgeändert, dass keine graphische Ausgabe mehr vorhanden war. Zuerst haben wir den Trainingsdatensatz eingelesen und damit das Netzwerk die Trends der Aktienkurse lernen lassen. Danach wurden die Testdaten geladen und im Netzwerk aktiviert, um die Performance der Vorhersage zu testen.

4.4.2 Ergebnisse

Ähnlich wie mit den Algorithmen des Überwachten Lernens konnten wir hier eine Genauigkeit von 50%-60% erzielen. Die Genauigkeit variierte zwischen Wiederholung mit gleichen Eingaben stark. Um die Genauigkeit der Vorhersagen zu erhöhen wäre es sinnvoll auch mit Return Werte anstatt von konkreten Close Preisen zu arbeiten.

5 Fazit

Das Fazit unserer Arbeit ist, dass es durchaus möglich ist mittels unserer Algorithmen gute Ergebnisse zu erzielen. Dennoch ist uns das, trotz dem anpassen sämtlicher Parameter nicht mit allen Algorithmen gelungen. Das liegt vor allem daran, dass der Aktienmarkt sehr schnelllebig und teilweise unberechenbar ist. Die besten Ergebnisse hatten wir mit unserem Moving Average Algorithmus da dieser eher auf langfristige Vorhersagen ausgelegt ist und man somit auch leichte Fehlvorhersagen gut ausgleichen kann.

6 Ausblick

In der Zukunft könnte man unsere Algorithmen noch mit einem News-Crawler verbinden der verschiedene Nachrichtenseiten nach Neuigkeiten durchsucht. Diese

Neuigkeiten könnten dann verwendet werden um eventuelle Veränderungen am Aktienmarkt besser und schneller vorherzusagen.