ABDA2019

Tina Amann
tina.amann@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

Samet Aslan
samet.aslan@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

Daniel Andre Eckardt
daniel.eckardt@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

Jan Bernd Gaida
jan.gaida@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

Patrick David Huget
patrick.huget@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

Hannes Klaus Müller
hannes.mueller@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

Alexander Puchta
alexander.puchta@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

Prof. Dr. Sebastian Leuoth
sebastian.leuoth@hof-university.de
Hochschule Hof
Hof an der Saale, Germany

ZUSAMMENFASSUNG

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Tina Amann, Samet Aslan, Daniel Andre Eckardt, Jan Bernd Gaida, Patrick David Huget, Hannes Klaus Müller, Alexander Puchta, and Prof. Dr. Sebastian Leuoth. 2020. ABDA2019. In *FWPM: FWPM:Big Data Analysis, Wintersemenster, 2019, Hof*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 VORWORT

Im heutigen hochdigitalisierten Zeitalter gewinnt der richtige Umgang mit Daten gerade auf technologischer Ebene immer mehr an Bedeutung. Nicht nur im Hinblick auf die Einhaltung der entsprechenden Datenschutzrichtlinien, sondern gerade im Bezug auf einen nutzenorientierten, zielgerichteten und zukunftsträchtigen Gebrauch dieser Informationen muss man sich neuen Herausforderungen stellen. Es gilt Konzepte zu entwickeln die möglichst sinnvoll auf den verschiedenen Gebieten der Datenverarbeitung einsetzbar sind. Sie sollten sich den stetig neu ergebende Anforderungen anpassen und umso mehr Faktoren gleichzeitig berücksichtigen können.

FWPM:Big Data Analysis, Wintersemenster, 2019, Hof

https://doi.org/10.1145/nnnnnnn.nnnnnnn

Ein solches Konzept, welches verschiedene Bereiche des zweckmäßigen Gebrauchs von Daten auf möglichst effiziente Art und Weise vereint, ist das Data Warehouse.
Dieses Prinzip bildet ein zentrales Datenbanksystem das zu Analysezwecken verwendet wird. Gerade im betriebswirtschaftlichen Bereich, aber auch in der Forschung wird dieses Modell gezielt eingesetzt, um mittels der richtigen Integration und Analyse von Daten entweder im wissenschaftlichen Kontext zu aussagekräftig interpretierbaren Ergebnissen zu gelangen oder im Zusammenhang der Marktforschung Entscheidungen zu entwickeln.
Beim Data Warehouse bildet hierbei eine Art Datenlager. Es bezieht Date aus einer Reihe von verschiedenen, heterogenen Quellen, sammelt diese und legt sie in verdichteter und aufbereiteter Form des Warehouses, also des Lagers ab.
Auf diese Weise kann es die diesem angebundenen Analysesysteme zentral mit dem ermittelten Datenkollektiv versorgen. Diese wiederum werten die Informationen aus. Diese Gesamtheit der Prozesse zur Datenbeschaffung, Verwaltung, Sicherung und Zurverfügungstellen der Daten nennt man dementsprechend Data Warehousing. Dieses beginnt also wiederum mit der Auswahl geeigneter interner oder externer Quellen, die allerdings nicht zum eigentlichen digitalen Lagerhaus gehören. Hierbei muss darauf geachtet werden qualitative Datenbestände ausfindig zu machen, die für das Modell geeignet sind und die allgemein üblichen Grundanforderungen an die Informationen erfüllen. Hierzu gehören unteranderem die Konsistenz, Korrektheit, Vollständigkeit und Zuverlässigkeit der verwendeten Quellen.
Im Architekturschaubild des Data Warehouses folgt die zum eigentlichen Datenbanksystem gehörige Data Staging Area. Diese stellt die zentrale Datenhaltungskomponente des Beschaffungsbereichs dar und dient somit als temprorärer Zwischenspeicher zu Integration, indem sie die Informationen aus den unterschiedlichen Systemen extrahiert, strukturiert, transformiert und ins Warehouse lädt.
Zunächst müssen in diesem Zusammenhang die vorliegenden Daten bereinigt werden, in dem fehlerhafte oder fehlende Werte ausgebessert, Duplikate beseitigt und veraltete Werte geupdatet werden. Hierdurch erfolgt die inhaltliche und durch entsprechende Schemaintegration die strukturelle Anpassung der Daten, um so die

Heterogenität der Quellen zu überwinden.
Nachdem die Daten dann bereinigt und in ein einheitliches Format gebracht wurden, können sie in die eigentliche Basisdatenbank geladen werden. Sie bildet die Presentation Are über die auf verschiedenen Ebenen, die sogenannten Data Marts mittels den entsprechenden Access Tools zugegriffen wird, um die beabsichtigten Analysenverfahren durchführen zu können.
So wird je nach Themengebiet eine Kennzahl gewonnen auf Basis welcher Entscheidungen oder Schlussfolgerungen getroffen werden können.
Um dieses Prinzip des Data Warehousings an einem Beispiel umsetzen zu können, wurde im Rahmen des Wahlfachmoduls „Big Data Analyse“ entschieden ein betriebswirtschaftliches Projekt anhand dieses Konzepts zu

2 RELATED WORK

Im Folgenden werden weitere Projekte aufgezeigt, die den Gedanken eines handelsfähigen Bots erfolgreich umgesetzt haben. Diese weisen sowohl Gemeinsamkeiten als auch Unterschiede zum hier vorgestellten Konzept auf, welche anhand geeigneter Programme vorgestellt werden.
Ein solcher Trading Bot wird mit der Open Source Lösung „Gekko“ umgesetzt. Hierbei handelt es sich um eine kostenlose Open Source Trading Plattform, die mit Kryptowährungen, in diesem Fall Bitcoins arbeitet.
Im Gegensatz zu der von der Studiengruppe angestrebten Lösung wird hier also nur eine Währung realisiert. Außerdem ermöglicht sie es aktuelle, aber auch vergangene Kurse genau zu verfolgen den Zwecken entsprechend zu analysieren.
Gekko erlaubt unteranderem sogar sogenannte „Backtests“. Das sind Strategiesimulationen, die anhand von historischen Daten aufzeigen, ob bestimmte Trades in der Vergangenheit profitabel gewesen wären. Ebenfalls interessant ist, dass man neben auswählbaren, auch eigene Strategien übergeben kann.
Aber auch mit Echtzeitdate können Strategien verfolgt und automatisch Trades bei entsprechenden Signalen gesetzt setzen. Anders als in diesem Projekt wurde die Lösung in Javascript implementiert und läuft auf der Plattform Node.js.
Ein weiteres Bitcoin-Trading Model stellt der Haasbot dar. Der Bot ist zwar für Bitcoin konzipiert, bietet aber dafür hohe sonstige Auswahl- und Einstellungsmöglichkeiten. Dieser Anbieter gibt den Nutzern bereits zu Beginn die Möglichkeit zwischen mehreren Basis-Bots wählen und weitere hinzuzufügen. Auch eigene Script-Bots können erstellt werden. Analysen, Sicherheiten und Absicherungen gehören zu dem Konzept.Haasbot unterstützt zudem eine Vielfalt an unterschiedlichen Börsen, darunter auch bereits vorgestellte Projekt Bitfinex.
Dieses Modell unterscheidet außerdem nicht nur zwischen unterschiedlichen Trading Strategien, sondern auch zwischen unterschiedlichen Bot-Typen. Dazu gehören Arbitrage-, Order-, Script und Haasbot Trading Bots. Im Gegensatz zu Gecko, sind die für Haasbot notwendigen Lizenzen aber auch recht kostspielig.
Eine andere Strategie verfolgt der HodlBot. Dieser basiert auf dem Konzept des "passiven Investierens".
Hierbei wird im Gegensatz zu den meisten andern Bots, nicht auf einzelne Währungen gesetzt, sondern bestimmte Marktindizes, die sogenannten „Hodl Indices“. Kunden erstellen hierfür zuerst ein persönliches Portfolio. In dieses halten sie fest, welche Währungen sie bevorzugen und welches Gewichtungsschema bzw. welche Strategie verfolgt werden soll. Mittels verschiedener APIs werden anschließend automatisch Trades abgeschlossen.
Letztendlich gibt es neben unserer Arbeit also bereits eine Vielzahl an Trading Bots, welche jeweils Vorteile und Nachteile aufweisen. An diesen wird sich anhand dieses Projektes orientiert. Die Recherche zeigt auch, dass Trading Bots von unterschiedlichsten Herstellern weitestgehend, den hier aufgezeigten Grundprinzipien folgen.

3 TRADINGBOT

Im heutigen Zeitalter möchten Menschen so wenig Arbeit wie möglich aufwenden, dennoch aber die bestmöglichen Gewinne erzielen. Hier greift der TradingBot. Viele verbinden mit dieser Software automatisches Geld verdienen ohne viel Zeit zu vergeuden. Im Folgenden Kapitel werden die Vor- und Nachteile des Bots aufgezeigt und wie die Software hier zum Einsatz kommt.

3.1 Recherche

Die Crypto Projektgruppe der Universität Oldenburg hat bereits ein ähnliches Programm entwickelt und eine Dokumentation veröffentlicht. Mithilfe dieser Informationen konnten Punkte eingebracht werden, die Anfangs weniger Beachtung erhalten hätten.

3.2 Ziel

Ziel ist es, dass eine automatisierte Abwicklung von Trades stattfindet, so dass der Nutzer eine Hilfe beim Handeln mit z. B. Kryptowährungen hat. Die Absicht der Verwendung einer solchen Software ist, die menschlichen Eigenschaften wie Angst, Gier und Risiko außen vor zu lassen. Jeder Mensch wird von diesen Punkten gesteuert und kann somit beim Traden zu früh oder auch zu spät Entscheidungen treffen. Ein solches Programm kann hierbei helfen, indem der Verwender keine Trades vorzeitig beenden kann, wenn diese über die Software gesetzt wurden.

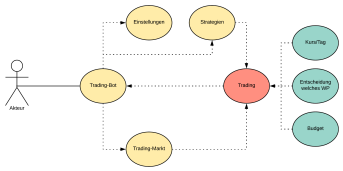

Abbildung 1: Trading Ablauf

3.3 Schnittstelle

Dieser TradingBot soll eine Schnittstelle zu einem Dashboard bieten. Aktuell läuft die Software nur in der Console und kann nicht verändert werden.
Sobald die Funktionalität ausgereift ist und genügend ausgewertete Daten zur Verfügung stehen, können die Ausarbeitung und das Implementieren in eine graphische Oberfläche begonnen werden.

Abbildung 2: Wireframe Dashboard

3.4 Funktionalität

Der TradingBot erhält unterschiedliche Clients, die alle Aufgaben abarbeiten. Nachfolgend werden die Funktionalitäten dieser Clients aufgeführt.

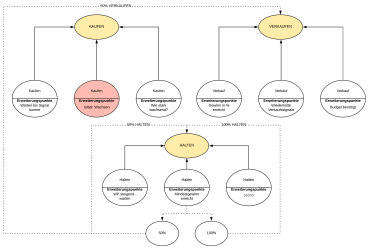

Abbildung 3: Funktionalität Diagramm

3.4.1 Persistenz. Um die Tätigkeiten der Software zu kontrollieren, werden alle durchgeführten Aktionen geloggt. Hierfür wird der Persistenz-Client benötigt, so dass alle Daten gespeichert und aufgerufen werden können, wenn ein Nutzer diese einsehen möchte.

3.4.2 API-Schnittstelle. Aktuell wird hier eine Schnittstelle künstlich erzeugt, die lediglich ausgibt, bestimmte Aktionen durchzuführen. Hier werden Schnittstellen zu Handelsplattformen eingepflegt um im Live-Modus effektiv zu handeln.

3.4.3 Portfolio. Jeder aktive Trader hat ein Portfolio, welches alle Angebote beinhaltet. Hier kann der Bot einsehen, welche dieser Trades bearbeitet werden müssen, falls ein Signal dafür erscheint.

3.4.4 Guthaben. Da ein Trade nur gesetzt werden kann, wenn das nötige Budget auf dem Nutzerkonto vorhanden ist, wird mit einem Budget-Client dieses überwacht. Jede Transaktion passt das Guthaben automatisch an.

3.4.5 Kaufen. Um einen Trade zu setzen wird ein Buy-Helper verwendet. Dieser bekommt die Schnittstelle und das Signal übergeben. So kann der Bot auf der richtigen Handelsplattform arbeiten. Jedes Signal hat eine Stärke, diese bestimmt dann die Kaufkraft. Aktuell werden pro Trade maximal 10% des Guthabens verwendet.
Mit dem Signal-Objekt wird eine Offer erzeugt, diese wird dann an den API-Client übergeben und der Trade wird gesetzt.

```
if (strength == '0'):
 self.signal_amount = 0.15
elif (strength == '1'):
 self.signal_amount = 0.33
elif (strength == '2'):
 self.signal_amount = 0.66
elif (strength == '3'):
 self.signal_amount = 1.0
```

Abbildung 4: Bestimmung der Höhe einer Buy-Order

3.4.6 Verkaufen. Ähnlich wie das Kaufen wird auch der Verkauf durchgeführt. Der Sell-Helper wird mit der API und dem Signal aufgerufen und dadurch wird eine Offer erstellt. Ist dies erledigt, wird die Offer an die API übergeben und der Trade mit einer Restsumme neu gesetzt oder aber geschlossen.

3.5 Vor- und Nachteile

Nachfolgend werden die Vor- und Nachteile eines Bots erklärt.

3.5.1 Vorteile. Durch den Einsatz einer Software für das Handeln mit Währungen können Emotionen minimiert werden, so dass es nicht zu einem voreiligen Verkauf kommt. Des weiteren hält sich der Bot mit hoher Disziplin an die Vorgaben. So werden keine höheren Profite erzwungen, die oft zu einem Verlust führen können. Der Bot wird mit den Regeln für das Handeln getestet, in dem historische Daten verwendet werden.

3.5.2 Nachteile. Jedoch können Programme auch Fehler aufweisen. Somit kann es passieren, dass aufgrund von Internetproblemen ein Trade nicht abgeschlossen werden kann. Weil diese Software TradingBot heißt, bedeutet dies jedoch nicht, dass alles selbständig abläuft. Eine Software muss immer kontrolliert werden um Fehlverhalten feststellen zu können.

3.6 Stand

Der aktuelle Stand ist, dass die Software an einem beliebigen Tag startet und von dort ab 30 Tage handelt. Jeder Tag wird lokal festgehalten und wird somit kontrollierbar. Aufgrund fehlender Daten, konnten nur künstlich erzeugte Daten zum Testen verwendet werden. Hier ist es nun wichtig, dass ein weiterer großer Test mit Echtzeit-Daten stattfindet, um den Effekt der hier genannten Indikatoren und Strategien zu testen.

LITERATUR

[1] Mario Fortier. [n.d.]. taLib. http://ta-lib.org. Accessed: 10.02.2020.

Abbildungsverzeichnis