

## **Projektseminar I4 Angewandte Informationswissenschaft, SoSe2016**

**Alesia Varanovich, Matrikel-Nummer: 2199838**

**GitHub Benutzername AlesiaVaranovich**

### **Dokumentation**

Im Rahmen des Projektseminars „Angewandte Informationswissenschaft“ sollte ein individuelles Projekt entwickelt und realisiert werden, dabei sollten eigene Interesse und Stärke einfließen. Ich habe mich entschieden, mich mit dem Thema Datenerhebung, -analyse und Visualisierung mit Python zu konfrontieren. In früheren Semestern hatte ich Gelegenheit, Programmiersprache Python kennenzulernen sowie erste Erfahrungen im Bereich Visualisierung mittels Python zu sammeln.

#### **Idee des Projekts**

Die Projektidee bestand darin, ein Tool zu erstellen, welches auf die aktuellen Börsendaten aus freien Internetquellen zugreift, sie analysiert und anschließend visualisiert. Als ich das Projektplan erstellt habe, habe ich es so vorgestellt: Ein User wählt einen Firmennamen aus dem Dropdown Menü aus oder er kann einen Market Index optional manuell im entsprechenden Textfeld eintragen. Weiterhin wählt er die zeitlichen Rahmen aus, für welche Periode sollen die Informationen ausgegeben werden. Als nächstes besteht die Möglichkeit zwischen zwei Internetquellen (Google Finance und Yahoo Finance) auszuwählen. Nachdem ein User einen "Suche"-Button drückt, wird eine Anfrage an eine ausgewählte Internetquelle gesendet, Daten werden erhalten, mittels Python verarbeitet und analysiert und als Ergebnis werden die Graphiken nach den ausgewählten Kriterien graphisch in GUI dargestellt. Zusätzlich wird für jede Graphik einen interaktiven Navigationstoolbar für weitere Einstellungen/Funktionen mit folgenden Buttons hinzugefügt: "Abspeichern", "Konfigurierung", "Zoom-to-Rectangle", "Zoom", "Vorwärts- und Zurück", "Home button". Alle erhaltenen Informationen zu der ausgewählten Firma werden gespeichert und in entsprechenden Textfeldern in GUI ausgegeben. Vergleich der Firmen nach minimalem und maximalem Preis während einer Börsensitzung und die

Visualisierung solcher Daten in Form von Tabellen habe ich mir als mögliche Erweiterungen des Tools vorgestellt.

Im Folgenden wird auf den Projektaufbau und auf einzelne Arbeitsschritte näher eingegangen. Außerdem werden die verwendeten Bibliotheken und Funktionen erläutert und Abweichungen vom Projektplan erklärt.

### **Aufbau des Projekts/Arbeitsschritte**

1. Erstellung von Graphical User Interface, wo später visualisierte Ergebnisse der Datenanalyse dargestellt werden → in dieser Phase wurden schon alle Funktionen und (mögliche) Optionen, alles was später im Tool implementiert werden sollte, berücksichtigt.
2. Datenerhebung → hier wurden Requests an die ausgewählten freien Internetquelle (Google Finance und Yahoo Finance) implementiert.
3. Datenverarbeitung → Weiterhin wurden die erhaltenen Responses bearbeitet indem aus erhaltenen Daten die Datenmengen erzeugt wurden und die Suche nach den Kriterien (Company Name, first trade, last trade, Date, Currency usw.) durchgeführt wurde.
4. Analyse&Visualisierung → Analyse und graphische Darstellung der Daten nach folgenden Kriterien durchgeführt: min/max Preis für eine Börsensitzung, Preisveränderungen nach oben oder unten während einer Börsensitzung, Preis der Eröffnung/Schließung der Börsensitzung. Vergleich zwischen zwei ausgewählten Firmen hinzugefügt wurde.

### Werkzeuge:

Das Tool wurde mit der Sprache Python 3.5. mithilfe von folgenden Libraries programmiert:

### Verwendete Bibliotheken:

- Tkinter für GUI Erstellung: Toolkit-Interface ist eine Bibliothek, die alles enthält, was zur Erzeugung von Komponenten einer grafischen Benutzeroberfläche mit Python benötigt. Diese Bibliothek gehört zur Standardbibliotheken von Python und kann direkt benutzt werden.

- Urllib für die Anfragen an Google Finance und Yahoo Finance. Das ist die Bibliothek, die Zugriff auf Websites im Internet ermöglicht. Die wichtigste Funktion ist dabei `urllib.request.urlopen`. Diese Funktion greift auf die durch URL adressierte Netzwerkressource zu und gibt ein Dateiojekt auf dieser Ressource zurück. Damit ermöglicht die Funktion es beispielsweise, den Quelltext einer Website herunterzuladen und wie eine lokale Variable zu speichern.
- NumPy für die Analyse der erhaltenen Daten (Datenmengen erzeugen). Die Bibliothek ist bei einer Standardinstallation von Python nicht installiert, deswegen wurde sie von dieser Website hochgeladen: <http://www.numpy.org>. diese Bibliothek beschäftigt sich mit Arrays.
- Matplotlib für die Analyse und Visualisierung von erhobenen Börsendaten in Form von Grafiken. Das ist Programmbibliothek der Programmiersprache Python, die mathematische Darstellungen erstellen kann.

## **Beschreibung der einzelnen Elemente und Funktionen**

**Elemente von Graphical User Interface:** ein gutes Tool braucht eine einfache intuitive Benutzeroberfläche. Dafür wurde die Bibliothek tkinter verwendet. Im Konstruktor der main Klasse wurde zuerst ein Hauptframe sowie ein Tuple aus zusätzlichen Fenstern wie Home Page, Help Page und Graph Builder page erstellt. „Home Page“ Fenster ist ein Hilfsmittel zum Navigieren zwischen zwei weiteren Fenstern des Tools und beinhaltet Steuerelemente – Buttons „Graph Builder“ und „Help“. „Graph Builder“ (M Graph Builder) Fenster ermöglicht es, eine Anfrage an ausgewählte Internetquelle (Google Finance oder Yahoo Finance) zu konfigurieren. Dafür sind folgende Steuerelemente zur Verfügung gestellt:

- optionMenu (range and companyName) – Drop down Menü für die Auswahl einer Firma und einer Zeitperiode, für die Daten erhoben werden sollen.
- Entry(IndexEntry) - ein Feld für die manuelle Eingabe des Market Indexes in dem Fall wenn der gespeicherten Liste mit den Firmenamen fehlt die für einen User relevante Firma.
- Buttons (searchYahooFinance, searchGoogleFinance, exit) - Buttons „search YahooFinance“ und „searchGoogleFinance“ sind für das Absenden von Anfragen an

eine ausgewählte Internetquelle zuständig. Button „Exit“ schickt einen User zum Fenster „HomePage“.

„Help Page“ (Methode der Klasse HomePage) Fenster gibt kurze Informationen über das Tool.

### **Datenerhebung und Datenbearbeitung**

Die Daten werden mittels Absenden von Requests an Yahoo Finance und Google Finance und Erhalten von Responses mit Verwendung von Python Bibliothek `urllib` erhoben. Requests werden unmittelbar vor dem Absenden formuliert und sehen folgenderweise aus (marketIndex und period sind hier die übergebenen Parameter):

<http://chartapi.finance.yahoo.com/instrument/1.0/{marketIndex}/chartdata?type=quote;range={period}/csv>

<https://www.google.com/finance/getprices?q={marketIndex}&i=86401&p={period}/&f=d,o,h,l,c,v>

prepareURL: diese Funktion bekommt Eingangsdaten von der Methode der Klasse `GraphPage`. Solche Eingangsdaten enthalten folgende Parameter:

- marketIndex – mittels dieses Index werden die Daten unmittelbar abgefragt.
- period - das zeitliche Intervall in dem die Daten abgefragt werden sollen.

Das Format dieses Parameters ist für Yahoo Finance und Google Finance nicht gleich. Maßnahmen zur Transformation in das gewünschte Format werden auch innerhalb dieser Funktion durchgeführt je nach Online Service, an den die Requests abgesendet werden.

- source – Name von Online Service.

Das Ergebnis einer erfolgreichen Anfrage sieht folgendermaßen aus:

```
Company-Name:Alphabet Inc.  
Exchange-Name:NMS  
unit:DAY  
timestamp:  
first-trade:20040819  
last-trade:20160920  
currency:USD  
previous_close_price:775.4200  
Date:20160822,20160920  
labels:20160822,20160829,20160906,20160912,20160919  
values:Date,close,high,low,open,volume  
close:759.6600,780.3500  
high:766.2200,782.7300  
low:754.0000,776.2000  
open:755.1300,780.0000  
volume:841000,2046000
```

20160822,772.1500,774.5400,770.0500,773.2700,923400  
20160823,772.0800,776.4400,771.7850,775.4800,925300  
20160824,769.6400,774.5000,767.0700,770.5800,1071100

Für die weitere Arbeit mit den erhobenen Daten werden sie vorbereitet: in Zeilen aufgeteilt (in der Methode `init` in der Klasse `marketGraph`) mit dem Ziel, die Daten zu sortieren und relevante für die Analyse Informationen (wie Datum, der letzte Preis, der minimale und der maximale Preis, Preis im Moment der Eröffnung/Schließung der Börsensitzung) zu extrahieren.

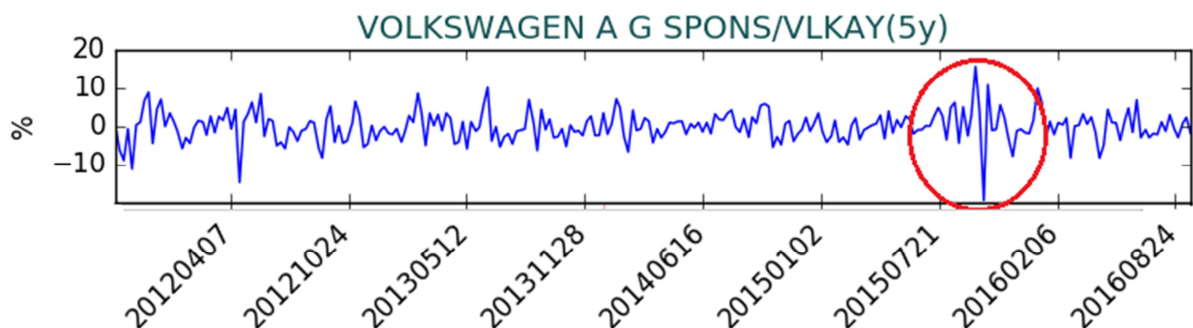
### Analyse und Visualisierung

Analyse und Visualisierung wird mithilfe von der Bibliothek `matplotlib` (graphische Visualisierung) durchgeführt. Die erste Möglichkeit ist die Dynamik der Preisveränderung während einer Börsensitzung zu analysieren.

Funktion `percentOfDayChangePrice`: Diese Grafik wird aufgrund von erhobenen von Google oder Yahoo Finance Daten gebildet mithilfe von der folgenden Formel:

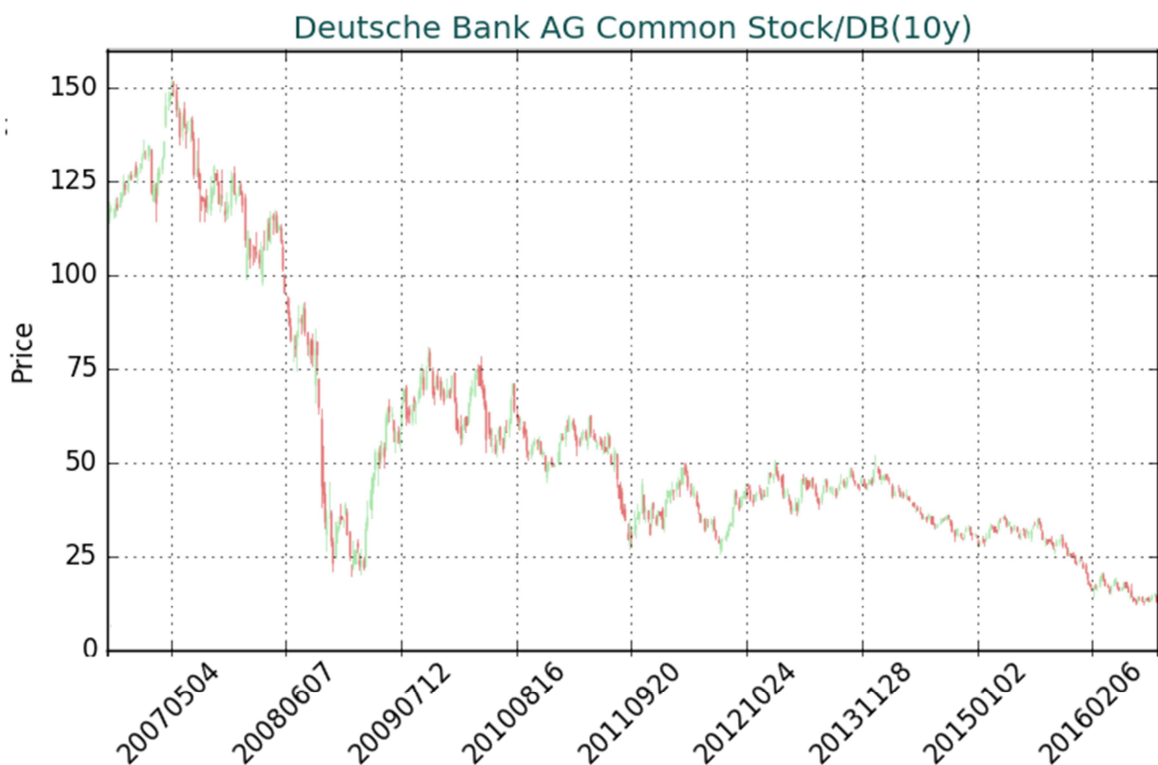
$$(\text{closePrice} * 100 / \text{openPrice}) - 100.$$

Diese Art von Grafik ist besonders interessant bei Analyse der raschen Veränderung im Preis wie z.B. bekannte Ereignisse mit Volkswagen Abgasskandal hatten einen so starken Einfluss, dass die Preise während einer Börsensitzung um mehr als 10 % runtergefallen sind (Abbildung 1.).



(Abbildung 1.).

Funktion ohlcGraph: Die zweite Möglichkeit ist Preisveränderungen nach oben oder unten während einer Börsensitzung sowie einen niedrigsten und einen höchsten Preis analysiert und grafisch dargestellt zu bekommen (open, high, low, close). Für diese Grafik wurde die Bibliothek matplotlib.finance verwendet, die für die Analyse und Visualisierung von Finanzdaten benutzt. (Abbildung 2).



(Abbildung 2).

Die dritte Möglichkeit ist den Börsenumsatz zu analysieren. (Abbildung 3)

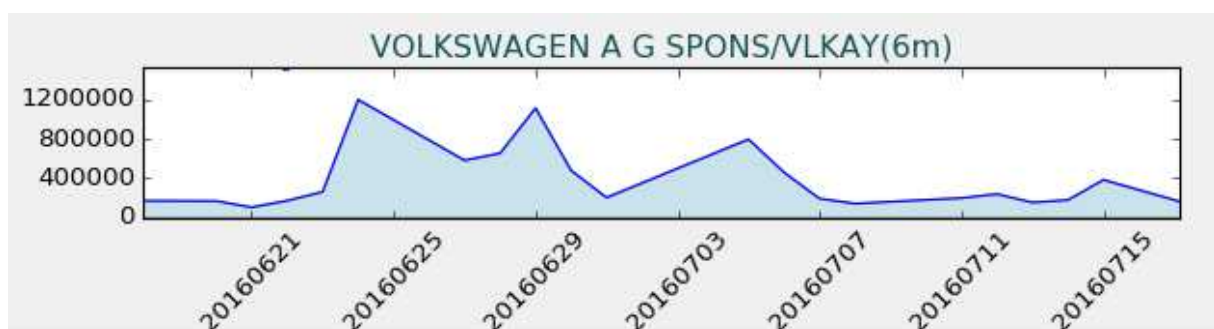
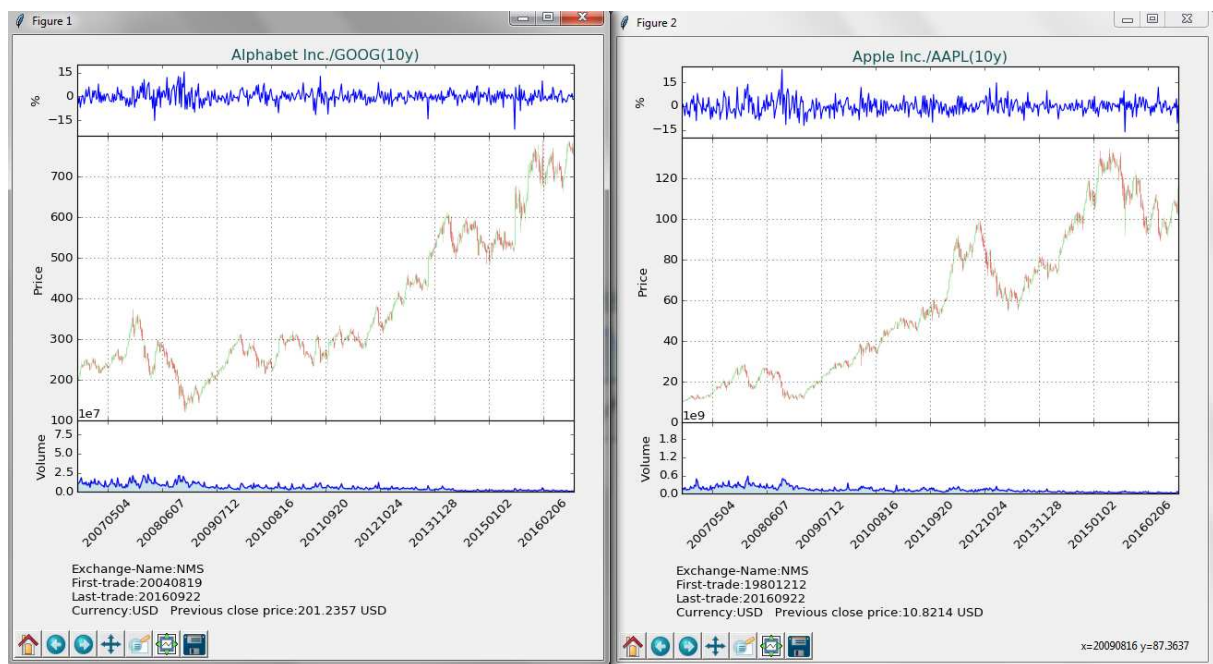


Abbildung 3.

Anschließend ermöglicht das Tool den Vergleich zwischen zwei Firmen, weil das Hauptfenster, das für die Abbildung von einzelnen Grafiken zuständig ist, immer offenbleibt (Abbildung 4). Ich habe die Idee diesen Vergleich tabellarisch darzustellen (wie es im Projektplan stand), abgesagt, weil ich das Konzept des Tools, das Ergebnisse nur mithilfe von Grafiken darstellt, weiter behalten möchte. Das (Möglichkeit gleichzeitig 2 Grafiken anzusehen) ist auch die Ursache dafür, warum das GUI Layout sieht nicht ganz so wie es im Projektplan ausgesehen hat.



(Abbildung 4).

## Anleitung zur Ausführung des Programms

Bevor das Programm zu starten, soll man sicher sein, dass die Internetverbindung besteht, weil sich das Programm mit den Online Services (Google Finance und Yahoo Finance) beschäftigt. Weiterhin soll geprüft werden, ob Python Version 3.5 auf dem PC installiert ist. Dafür kann der Befehl "python -V" in der Kommandozeile getippt werden. Wenn Python nicht installiert ist, soll die entsprechende Version von der Website <https://www.python.org/downloads/> downloaden und installiert werden. Weiterhin soll auch geprüft werden ob alle notwendigen Bibliotheken vorhanden sind und korrekt importiert wurden. Dafür wird es genug sein, in Python shell/prompt den Befehl `help('modules')` auszuführen. Danach werden alle installierten Bibliotheken

angezeigt (Vorsicht: das kann einigerweile dauern!). In dem Fall wenn eine von den Bibliotheken fehlt, sie kann mithilfe von dem Befehl `pip install matplotlib` installiert werden (z.B. `pip install matplotlib`).

Um das Programm zu starten soll man in den Ordner mit dem Skript gelangen. Dafür soll folgender Befehl in der Kommandozeile getippt werden: `cd „Pfad zu Datei“\marketueberblick.py`. Weiterhin soll der Befehl `python marketueberblick.py` ausgeführt werden.

Nach dem Starten des Programms wird das Fenster "HomePage" eröffnet. Von diesem Fenster kann man, wenn es Notwendigkeit besteht, zum "Help" Fenster gelangen, das kurze Informationen zum Tool enthält. Es ist auch möglich für einen Benutzer einen relevanten für ihn Firmennamen aus der vorhandenen Liste auswählen (OptionsMenu) sowie manuell den Aktiennamen eintippen. Dabei erkennt das Tool auch Aktienkurse mit sogenannten Präfixen (BMW.F für Frankfurt oder BMW.MI für Mailand).

Sobald der Firmenname/Aktiename ausgewählt ist, soll man auch die zeitliche Periode für die Analyse der Börsendaten hinzufügen (3 Monate, 6 Monate, 1 Jahr) und auf den Button "Google Finance" oder "Yahoo Finance" drücken, was automatisch zu einem neuen Fenster führt, wo analysierte und visualisierte Börsendaten graphisch dargestellt werden.

In diesem Fenster ---- können folgende Ergebnisse gesehen werden: 1) Veränderung des Aktienpreises während der Börsensitzung. Diese Grafik ist dafür gut geeignet, eine rasche Veränderung im Preis wie man z.B. nach dem Brexit beobachten könnte 2) Nächste Grafik zeigt den Eröffnungs-, Höchst-, Niedrigst- und Schlusspreis (open, high, low, close) während einer ausgewählten Zeitperiode an. 3) Die dritte Grafik zeigt Ergebnisse der Analyse von Börsenumsatz.

Der User kann jede Grafik mithilfe von Navigationstoolbar so betrachten wie es ihm komfortabel ist, es ist auch möglich beliebige Grafik als .png oder .pdf Datei speichern.

Um das Programm zu schließen, soll man den Button „Schließen“ drücken.

## **Reflexion und Fazit**

Das Ziel des Projektes war, ein Tool zu erstellen, welches auf die aktuellen Börsendaten aus freien Internetquellen zugreift, sie analysiert und anschließend



visualisiert. Um das alles zu verwirklichen, sollte ich nicht nur meine Kenntnisse in Python erfrischen, sondern auch im Bereich Börsen und Finanzen recherchieren. Man kann behaupten, dass das Projekt zufriedenstellend abgeschlossen wurde, da das Tool funktioniert und es hat gelungen, alle Ideen aus dem Projektplan zu implementieren. Das Tool kann für die Benutzer nützlich sein, die notwendige Informationen über die aktuelle Börsendaten aus zwei größten Finanz Portalen bekommen wollen und die möchten dabei die Möglichkeit zu haben, Grafiken zu speichern und unterschiedliche Firmen miteinander abgesehen von Börsenumsatz etc. zu vergleichen.

#### Verwendete Quellen:

1. <http://matplotlib.org/index.html>
2. <http://www.tutorialspoint.com/python/>