STOCK MARKET ANALYSIS USING PYTHON

A Report for the Evaluation of Project

Submitted by

DIVYA YADAV

(1613112016)

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Prof. Dr. N. Thillaiarasu

April/May-2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report " **STOCK MARKET ANALYSIS USING PYTHON**" is the bonafide work of "**DIVYA YADAV (1613112016)**" who carried out the project work under my supervision.

SIGNATURE OF HEAD	**SIGNATURE OF SUPERVISOR**
	Dr. N.Thillaiarasu
	Professor

School of Computing Science & Engineering

School of Computing Science & Engineering

TABLE OF CONTENTS

S.no	DESCRIPTION	PAGE NUMBER
1.	Abstract	5
2.	Introduction (i) Overall description (ii) Purpose (iii) Motivations and scope	8
3.	Stock market Insight	10
4.	Literature Review	13
5.	Problem statement	13
6.	Output	15
7.	Proposed model	18
8.	Conclusion	19
9.	References	20

Abstract

In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.

Modeling and Forecasting of the financial market have been an attractive topic to scholars and researchers from various academic fields. The financial market is an abstract concept where financial commodities such as stocks, bonds, and precious metals transactions happen between buyers and sellers. In the present scenario of the financial market world, especially in the stock market, forecasting the trend or the price of stocks using machine learning techniques and artificial neural networks are the most attractive issue to be investigated. As Giles explained, financial forecasting is an instance of signal processing problem which is difficult because of high noise, small sample size, non-stationary, and non-linearity. The noisy characteristics mean the incomplete information gap between past stock trading price and volume with a future price. The stock market is sensitive with the political and macroeconomic environment. However, these two kinds of information are too complex and unstable to gather. The above information that cannot be included in features are considered as noise. The sample size of financial data is determined by real-world transaction records. On one hand, a larger sample size refers a longer period of transaction records; on the other hand, large sample size increases the uncertainty of financial environment during the 2 sample period.

In Burton's hypothesis, he indicates that predicting or forecasting the financial market is unrealistic, because price changes in the real world are unpredictable. All the changes in prices of the financial market are based on immediate economic events or news. Investors are profit-

oriented, their buying or selling decisions are made according to most recent events regardless past analysis or plans. The argument about this Efficient Market Hypothesis has never been ended. So far, there is no strong proof that can verify if the efficient market hypothesis is proper or not. However, as Yaser claims, financial markets are predictable to a certain extent. The past experience of many price changes over a certain period of time in the financial market and the undiscounted serial correlations among vital economic events affecting the future financial market are two main pieces of evidence opposing the Efficient Market Hypothesis. In recent years, machine learning methods have been extensively researched for their potentials in forecasting and prediction of the financial market. Multi-layer feed forward neural networks, SVM, reinforcement learning, relevance vector machines, and recurrent neural networks are the hottest topics of many approaches in financial market prediction field. Among all the machine learning methods, neural networks are well studied and have been successfully used for forecasting and modeling financial market. "Unlike traditional machine learning models, the network learns from the examples by constructing an input-output mapping for the problem at hand. Such an approach brings to mind the study of nonparametric statistical inference; the term "nonparametric" is used here to signify the fact that no prior assumptions are made on a statistical model for the input data", according to Simon. As Francis E.H. Tay and Lijuan Cao explained in their studies, Neural networks are more noise tolerant and more flexible compared with traditional statistical models. By noise tolerance, one means neural networks have the ability to be trained by incomplete and overlapped data. Flexibility refers to that neural networks have the capability to learn dynamic systems through a retraining process using new data patterns. Long short-term memory is a recurrent neural network introduced by Sepp Hochreite and Jurgen Schmidhuber in 1997.

INTRODUCTION

OVERALL DESCRIPTION

Basically, quantitative traders with a lot of money from stock markets buy stocks derivatives and equities at a cheap price and later on selling them at high price. The trend in a stock market prediction is not a new thing and yet this issue is kept being discussed by various organizations. There are two types to analyze stocks which investors perform before investing in a stock, first is the fundamental analysis, in this analysis investors look at the intrinsic value of stocks, and performance of the industry, economy, political climate etc. to decide that whether to invest or not. On the other hand, the technical analysis it is an evolution of stocks by the means of studying the statistics generated by market activity, such as past prices and volumes. In the recent years, increasing prominence of machine learning in various industries have enlightened many traders to apply machine learning techniques to the field, and some of them have produced quite promising results.

PURPOSE

Basically, quantitative traders with a lot of money from stock markets buy stocks derivatives and equities at a cheap price and later on selling them at high price. The trend in a stock market prediction is not a new thing and yet this issue is kept being discussed by various organizations. There are two types to analyze stocks which investors perform before investing in a stock, first is the fundamental analysis, in this analysis investors look at the intrinsic value of stocks, and performance of the industry, economy, political climate etc. to decide that whether to invest or not. On the other hand, the technical analysis it is an evolution of stocks by the means of studying the statistics generated by market activity, such as past prices and volumes. In the recent years, increasing prominence of machine learning in various industries have enlightened many traders to apply machine learning techniques to the field, and some of them have produced quite promising results. With the proposed models, we achieve a potent improvement in the current state-of-the-art for time series classification using deep neural networks. Our baseline models, with and without fine-tuning, are trainable end-to-end with nominal preprocessing and are able to achieve significantly improved performance. This report will develop a financial data predictor program in which there will be a dataset storing all historical stock prices and data will be treated as training sets for the program. The main purpose of the prediction is to reduce uncertainty associated to investment decision making.

MOTIVATION AND SCOPE

Stock Market follows the random walk, which implies that the best prediction you can have about tomorrow's value is today's value. Indisputably, the forecasting stock indices is very difficult because of the market volatility that needs accurate forecast model. The stock market indices are highly fluctuating and it effects the investor's belief. Stock prices are considered to be a very dynamic and susceptible to quick changes because of underlying nature of the financial domain and in part because of the mix of a known parameters (Previous day's closing price, P/E ratio etc.) and the unknown factors (like Election Results, Rumors etc.). There has been numerous attempts to predict stock price with Machine Learning. The focus of each research projects varies a lot in three ways. (1) The targeting price change can be near-term (less than a minute), short-term (tomorrow to a few days later), and a long-term (months later), (2) The set of stocks can be in limited to less than 10 particular stock, to stocks in particular industry, to generally all stocks. (3) The predictors used can range from a global news and economy trend, to particular characteristics of the company, to purely time series data of the stock price. The probable stock market prediction target can be the future stock price or the volatility of the prices or market trend. In the prediction there are two types like dummy and a real time prediction which is used in stock market prediction system. In Dummy prediction they have define some set of rules and predict the future price of shares by calculating the average price. In the real time prediction compulsory used internet and saw current price of shares of the company.

Stock Market Insight

The stock market alludes to public markets that exist for issuing, buying and selling stocks that exchange on a stock trade or over-the-counter It is where you can purchase and sell portions of organizations listed there for exchanging. Stock trades list shares of common equity just as other security types, for example corporate e.g. corporate bonds and convertible bonds.

The financial exchange fills two significant needs. The first is to give cash-flow to organizations that they can use to support and extend their organizations. The auxiliary reason the stock market serves is to give investors – the individuals who buy stocks – the chance to partake in the the profits of publicly-traded companies.

Investing in the stock market is among the most common ways investors attempt to grow their money, but it's also among the riskier investment options available. The securities once issued in the primary market become the part of the secondary market. It provides a place or mechanism for active trading of securities among investors themselves. The stock market is a secondary market, which aids to the liquidity of securities traded there on. When investors have to buy securities in the secondary market, they have to contact the securities brokers for opening the account for the purchase of securities. After the account has been opened, the securities broker conveys the order of investor to the securities dealers who handle the inventory of securities. There are two basic types of stock markets- organized stock exchange and over-the-counter market.

A composed stock trade is the physical areas where securities are exchanged under some settled principles and guideline through the authorized individuals from the trade. It is one of the significant optional markets where the financial specialists purchase and sell the securities between themselves. Sorted out stock trades encourage the exchanging of protections, which are recorded in it. This implies the protections, which are not recorded, are not exchanged sorted out stock trade. It gives organizations access to capital in return for giving financial giving investors a slice of ownership.

This trade is either through formal exchanges or OTC[i] marketplaces. OTC market was traditionally concerned with the trading of securities which were not listed in an organized stock exchange. However, today the securities listed in organized stock exchange are also traded in OTC market. OTC market is an informal type of market for securities where no compulsory listing of securities is required. Any security can be traded on OTC market as long as a registered dealer is willing to make a market in the security (willing to buy and sell the security). It is not a central physical place like an organized stock exchange, rather it is the network of brokers and dealers scattered across the country. Buy and sell in an OTC market are conducted through negotiated bidding through a network of the communication line and computer system, which links brokers and dealers in an OTC market to their clients. The brokers and dealers in the OTC market can compete with both investment bankers and the organized exchanges because they can operate in both primary as well as secondary market.

The stock exchanges were once physical market places where the brokers of sellers and buyers operated through the auction process. Now, these have been replaced with electronic exchanges where sellers and buyers are connected over a telecommunications network by computers. Auction trading is providing a way to "screen-based" trading, where offer (or ask prices) and bid prices are displayed on the computer screen.

A stock exchange may be described in different ways. In simple words, a stock exchange is "A centralized market for selling buying stocks where the quotes is determined through demand-supply mechanisms".

Functions of Stock Exchanges

The stock exchange performs four important functions.

1 First, it renders a market place for sale and purchase of securities such as bonds, debentures, shares etc. Thus stock exchanges provide the feasibility for continuous trading in securities.
2 Second, they provide liquidity to the investments in securities. It allows the capitalist a place to liquidate their holdings.
3 Third, they help in the valuation of securities by providing the market quotations of the securities prices.
4 Fourth, stock exchanges act as a barometer, that indicates the state of health of the nation's economy.

Organization, Membership and Management of Stock Exchanges

Basically, for trading securities , a stock exchange is an organized market. It is also known as bourse. It is an association of individuals which is governed by certain specific rules and regulations. The rules of membership and mode of organization are important features of stock exchanges.

Over the decades, exchanges in the country have been organized in various ways. For example public limited company, voluntary non-profit making association and company limited by guarantee. In India, earlier stock exchanges were formed as voluntary non-profit making associations of persons. Later on they began to be organized as companies.

Listing of Securities

For trading of their securities in a stock exchange, a company has to be listed in that stock exchange. For the purpose of trading, Listing is the procedure of including the securities of a company in the official list of the stock exchange.

Permitted Securities

The securities of companies which have signed listing agreement with an exchange are traded at the exchange as listed securities. A stock exchange sometimes allows certain securities which actively trading in other stock exchanges but are not listed at the exchange. Such securities are called as permitted securities.

REGULATION OF STOCK EXCHANGES

The stock exchanges play a very sensitive and critical role in the functioning of the economy, especially the private sector of the economy. Therefore, the functioning of the exchanges needs to be efficient, fair and transparent. This is assured through proper regulation of the functioning of stock exchanges. There are rules, regulations,acts, guidelines and by-laws governing the working of secondary markets or stock exchanges in the country.

Literature Survey

Stock Market prediction has always had a certain appeal for researchers. While numerous scientific attempts have been made, no method has been discovered to accurately predict stock price movement. The difficulty of prediction lies in the complexities of modeling market dynamics. Even with a lack of consistent prediction methods, there havebeen some mild successes. Stock Market research encapsulates two elemental trading philosophies; Fundamental and Technical approaches. In Fundamental analysis, Stock Market price movements are believed to derive from a security's relative data. Fundamentalists use numeric information such as earnings, ratios, and management effectiveness to determine future forecasts. In Technical analysis, it is believed that market timing is key. Technicians utilize charts and modeling techniques to identify trends in price and volume. These later individuals rely on historical data in order to predict future outcomes. One area of limited success in Stock Market prediction comes from textual data. Information from quarterly reports or breaking news stories can dramatically affect the share price of a security. Most existing literature on financial text mining relies on identifying a predefined set of key words and machine learning techniques. These methods typically assign weights to keywords in proportion to the movement of a share price. These types of analyses have shown a definite, but weak ability to forecast the direction of share prices.

Problem Statement

The stock market appears in the news every day. We hear about it every time it reaches a new high or a new low. The rate of investment and business opportunities in the Stock market can increase if an efficient algorithm could be devised to predict the short term price of an individual stock.

Previous methods of stock predictions involve the use of Artificial Neural Networks and Convolution Neural Networks which has an error loss at an average of 20%.

We'll dive into the implementation part of this article soon, but first it's important to establish what we're aiming to solve. Broadly, stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis.

1. Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance.
2. Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market.

We will first load the dataset and define the target variable for the problem:

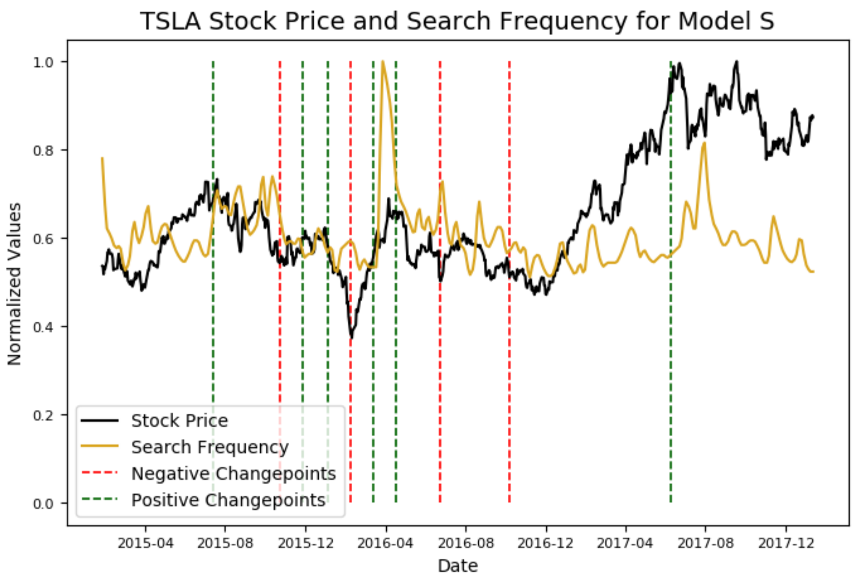
```
#import packages

import pandas as pd

import numpy as np



#to plot within notebook

import matplotlib.pyplot as plt

%matplotlib inline



#setting figure size

from matplotlib.pylab import rcParams

rcParams['figure.figsize'] = 20,10
```

```
#for normalizing data

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0, 1))



#read the file

df = pd.read_csv('NSE-TATAGLOBAL(1).csv')



#print the head

df.head()
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

Stocker, a Python class-based tool for stock analysis and prediction (the name was originally arbitrary, but I decided after the fact it nicely stands for “stock explorer”). I had tried several times to conquer classes, the foundation of object-oriented programming in

Python, but as with most programming topics, they never quite made sense to me when I read the books. It was only when I was deep in a project faced with a problem I had not solved before that the concept finally clicked, showing once again that experience beats theoretical explanations! In addition to an exploration of Stocker, we will touch on some important topics including the basics of a Python class and additive models. For anyone wanting to use Stocker, the complete code can be found on GitHub along with documentation for usage. Stocker was designed to be easy to use (even for those new to Python), and I encourage anyone reading to try it out. Now, let's take a look at the analysis capabilities of Stocker!



After installing the required libraries, the first thing we do is import the Stocker class into our Python session. We can do this from an interactive Python session or a Jupyter Notebook started in the directory with the script.

```
from stocker import Stocker
```

We now have the Stocker class in our Python session, and we can use it to create an instance of the class. In Python, an instance of a class is called an object, and the act of creating an object is sometimes called instantiation or construction. In order to make a Stocker object we need to pass in the name of a valid stock ticker (**bold** indicates output).

microsoft = Stocker('MSFT')**MSFT Stocker Initialized. Data covers 1986-03-13 to 2018-01-16.**

Now, we have a microsoft object with all the properties of the Stocker class. Stocker is built on the quandl WIKI database which gives us access to over 3000 US stocks with years of daily price data (full list). For this example, we will stick to Microsoft data. Although Microsoft might be seen as the opposite of open-source, they have recently made some changes that make me optimist they are embracing the open-source community (including Python).

A class in Python is comprised of two main parts: attributes and methods. Without going into too much detail, attributes are values or data associated either with the class as a whole or with specific instances (objects) of the class. Methods are functions contained in the class which can act on that data. One attribute of a Stocker object is stock data for a specific company that is attribute is associated with the object when we construct it. We can access the attribute and assign it to another variable for inspection:

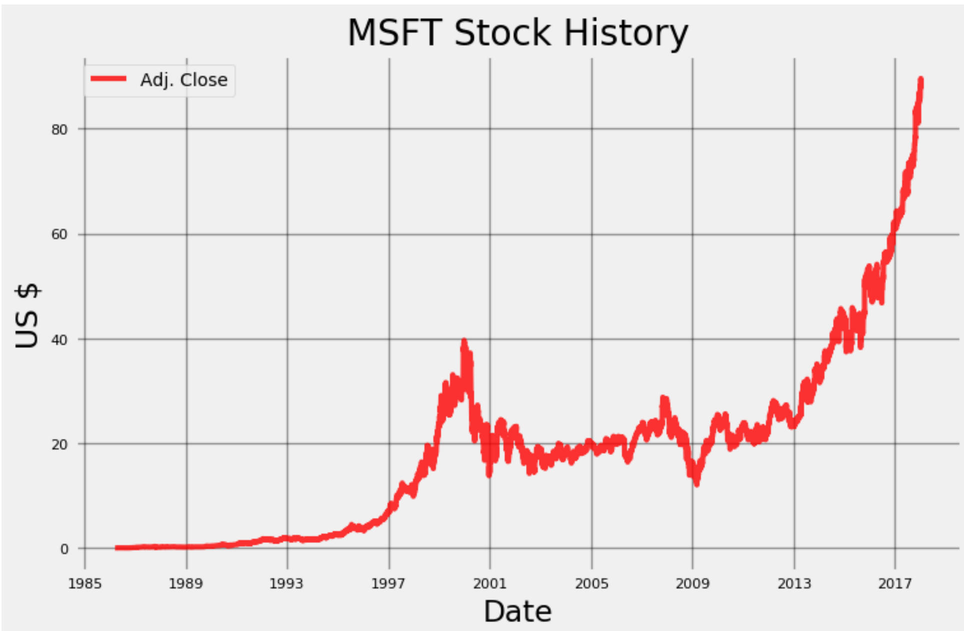
```
# Stock is an attribute of the microsoft object
stock_history = microsoft.stock
stock_history.head()
```

	Date	Open	High	Low	Close	Volume	Ex-Dividend	Split Ratio	Adj. Open	Adj. High	Adj. Low	Adj. Close	Adj. Volume	ds	y	Daily Change
0	1986-03-13	25.50	29.25	25.5	28.00	3582600.0	0.0	1.0	0.058941	0.067609	0.058941	0.064720	1.031789e+09	1986-03-13	0.064720	0.005779
1	1986-03-14	28.00	29.50	28.0	29.00	1070000.0	0.0	1.0	0.064720	0.068187	0.064720	0.067031	3.081600e+08	1986-03-14	0.067031	0.002311
2	1986-03-17	29.00	29.75	29.0	29.50	462400.0	0.0	1.0	0.067031	0.068765	0.067031	0.068187	1.331712e+08	1986-03-17	0.068187	0.001156
3	1986-03-18	29.50	29.75	28.5	28.75	235300.0	0.0	1.0	0.068187	0.068765	0.065876	0.066454	6.776640e+07	1986-03-18	0.066454	-0.001734
4	1986-03-19	28.75	29.00	28.0	28.25	166300.0	0.0	1.0	0.066454	0.067031	0.064720	0.065298	4.789440e+07	1986-03-19	0.065298	-0.001156

The benefit of a Python class is that the methods (functions) and the data they act on are associated with the same object. We can use a method of the Stocker object to plot the entire history of the stock.

```
# A method (function) requires parentheses
microsoft.plot_stock()
```

Maximum Adj. Close = 89.58 on 2018-01-12.
Minimum Adj. Close = 0.06 on 1986-03-24.
Current Adj. Close = 88.35.



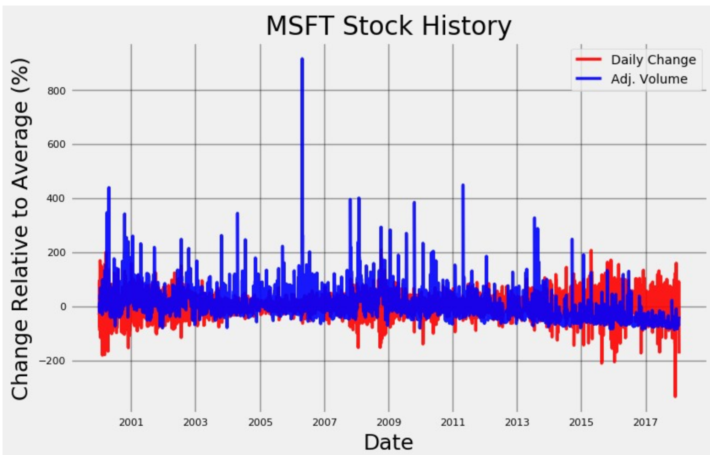
The default value plotted is the Adjusted Closing price, which accounts for splits in the stock (when one stock is split into multiple stocks, say 2, with each new stock worth 1/2 of the original price).

This is a pretty basic plot that we could have found from a Google Search, but there is something satisfying about doing it ourselves in a few lines of Python!

The `plot_stock` function has a number of optional arguments. By default, this method plots the Adjusted Closing price for the entire date range, but we can choose the range, the stats to plot, and the type of plot. For example, if we want to compare the Daily Change in price with the Adjusted Volume (number of shares) traded, we can specify those in the function call.

microsoft.plot_stock(start_date = '2000-01-03', end_date = '2018-01-16', stats = ['Daily Change', 'Adj. Volume'], plot_type='pct')**Maximum Daily Change = 2.08 on 2008-10-13.**
Minimum Daily Change = -3.34 on 2017-12-04.
Current Daily Change = -1.75.

Maximum Adj. Volume = 591052200.00 on 2006-04-28.
Minimum Adj. Volume = 7425503.00 on 2017-11-24.
Current Adj. Volume = 35945428.00.



Notice the y-axis is in percentage change relative to the average value for the statistic. This scale is necessary because the daily volume is originally in shares, with a range in the hundreds of millions, while daily price change typically is a few dollars! By converting to percentage change we can look at both datasets on a similar scale. The plot shows there is no correlation between the number of shares traded and the daily change in price. This is surprising as we might have expected more shares to be traded on days with larger price changes as people rush to take advantage of the swings. However, the only real trend seems to be that the volume traded decreases over time. There is also a significant decrease in price on December 4, 2017 that we could try to correlate with news stories about Microsoft. A quick news search for December 3 yields the following:

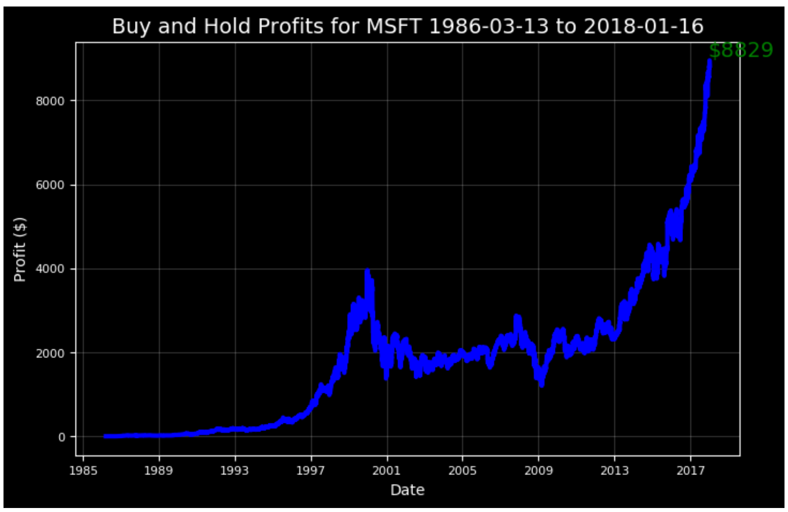
There certainly does not seem to be any indication that Microsoft stock is due for its largest price decrease in 10 years the next day! In fact, if we were playing the stock market based on news, we might have been tempted to buy stock because a deal with the NFL (second result) sounds like a positive!

Using plot_stock,we can investigate any of the quantities in the data across any date range and look for correlations with real-world events (if there are any). For now, we will move on to one of the more enjoyable parts of Stocker: making fake money!

Let's pretend for a moment we had the presence of mind to invest in 100 shares of Microsoft at the company's Initial Public Offering (IPO). How much richer would we be now?

```
microsoft.buy_and_hold(start_date='1986-03-13',
                       end_date='2018-01-16', nshares=100)
```

MSFT Total buy and hold profit from 1986-03-13 to 2018-01-16 for 100 shares = $8829.11



In addition to making us feel better, using these results will allow us to plan our trips back in time to maximize profits.

If we are feeling too confident, we can try to tweak the results to lose money:

```
microsoft.buy_and_hold(start_date='1999-01-05',
                       end_date='2002-01-03', nshares=100)
```

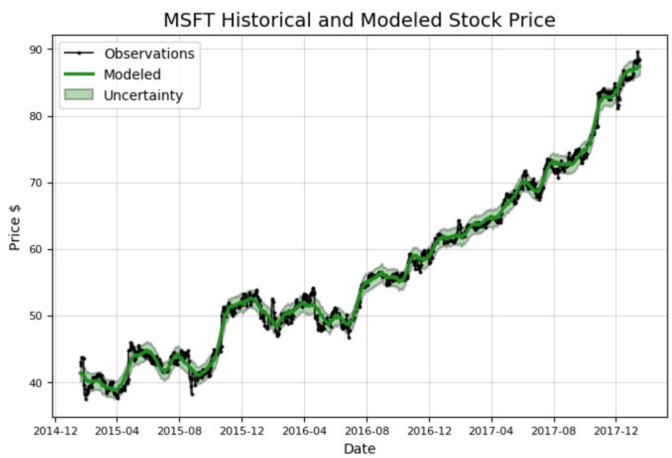
MSFT Total buy and hold profit from 1999-01-05 to 2002-01-03 for 100 shares = $-56.92

Surprisingly, it is possible to lose money in the stock market!

Additive Models

Additive models are a powerful tool for analyzing and predicting time series, one of the most common types of real world data. The concept is straightforward: represent a time series as a combination of patterns on different time scales and an overall trend. We know the long-term trend of Microsoft stock is a steady increase, but there could also be patterns on a yearly or daily basis, such as an increase every Tuesday, that would be economically beneficial to know. A great library for analyzing time series with daily observations (such as stocks) is Prophet, developed by Facebook. Stocker does all the modeling work with Prophet behind the scenes for us, so we can use a simple method call to create and inspect a model.

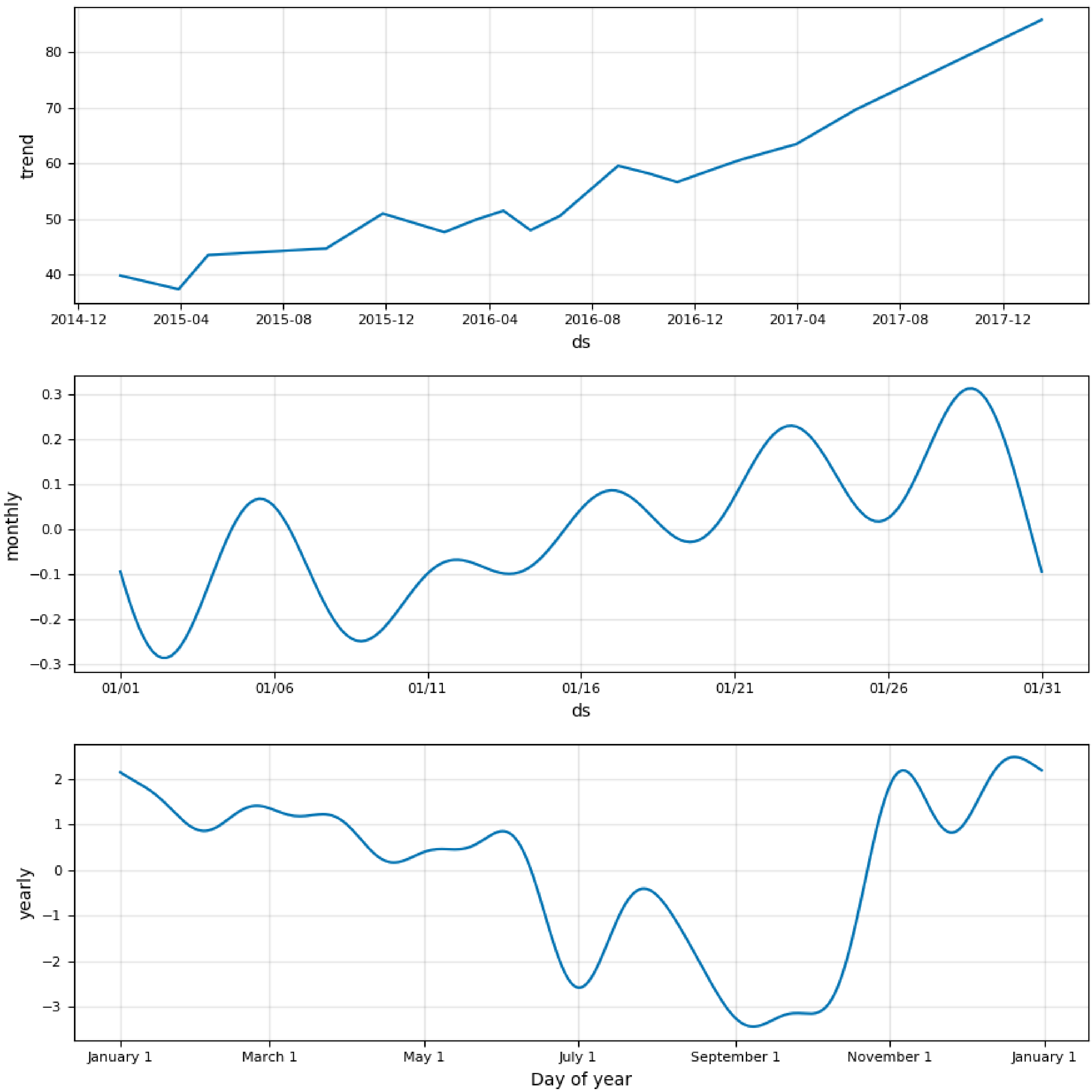
```
model, model_data = microsoft.create_prophet_model()
```



The additive model smooths out the noise in the data, which is why the modeled line does not exactly line up with the observations. Prophet models also calculate uncertainty, an essential part of modeling as we can never be sure of our predictions when dealing with fluctuating real life processes. We can also use a prophet model to make predictions for the future, but for now we are more concerned with past data. Notice that this method call returned two objects, a model and some data, which we assigned to variables. We now use these variables to plot the time series components.

```
# model and model_data are from previous method call
model.plot_components(model_data)
plt.show()
```

The overall trend is a definitive increase over the past three years. There also appears to be a noticeable yearly pattern (bottom graph), with prices bottoming out in September and October and reaching a peak in November and January. As the time-scale decreases, the data gets noisier. Over the course of a typical month, there is more signal than noise! If we

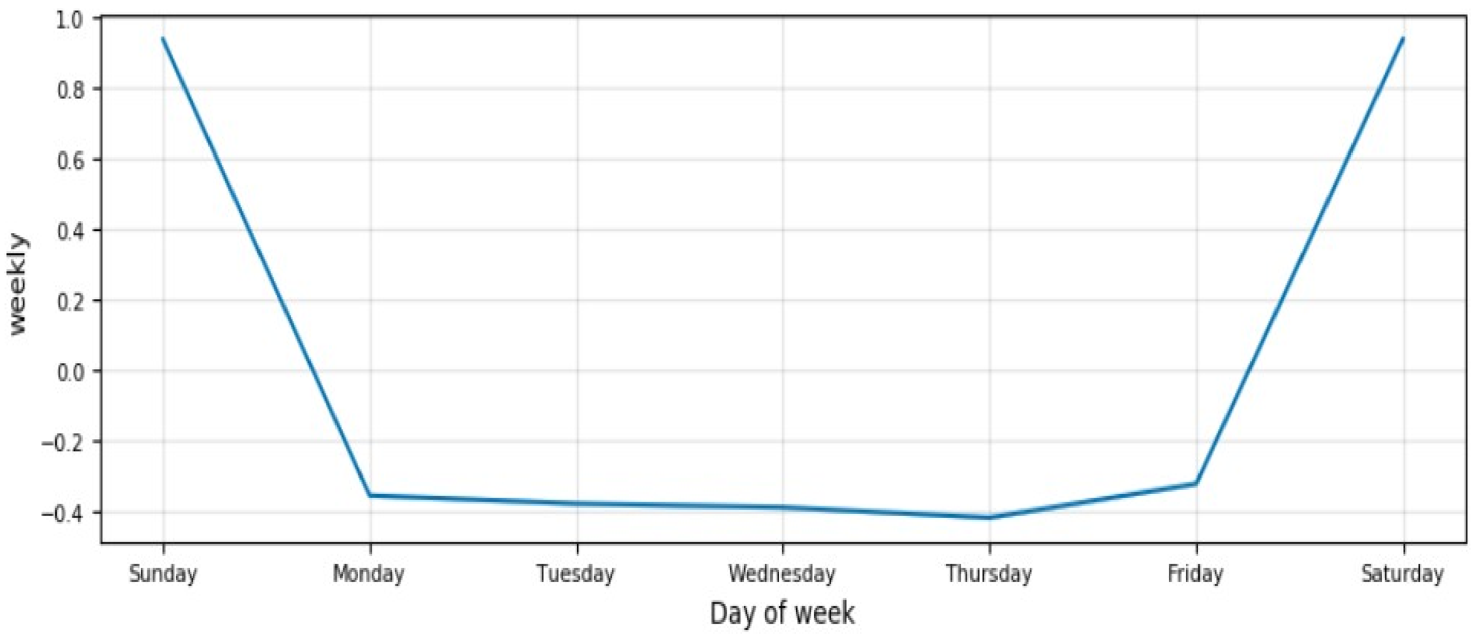


believe there might be a weekly pattern, we can add that in to the prophet model by changing the weekly_seasonalityattribute of the Stocker object:

```
print(microsoft.weekly_seasonality)
microsoft.weekly_seasonality = True
print(microsoft.weekly_seasonality)
```

False
True

The default value for weekly_seasonalityis False, but we changed the value to include a weekly pattern in our model. We then make another call to create_prophet_modeland graph the resulting components. Below is the weekly seasonality from the new model.



There was no way I could make this graph look good

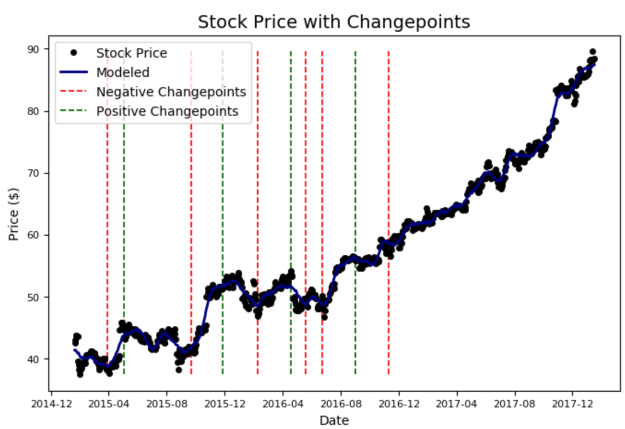
We can ignore the weekends because the price only changes over the week (in reality the price changes by a small amount during after-hours training but it does not affect our analysis). Unfortunately, there is not a trend over the week that we can use and before we continue modeling, we will turn off the weekly seasonality. This behavior is expected: with stock data, as the time scale decreases, the noise starts to wash out the signal. On a day-to-day basis, the movements of stocks are essentially random, and it is only by zooming out to the yearly scale that we can see trends. Hopefully this serves as a good reminder of why not to play the daily stock game!

Changepoints

Changepoints occur when a time-series goes from increasing to decreasing or vice versa (in a more rigorous sense, they are located where the change in the rate of the time series is greatest). These times are extremely important because knowing when a stock will reach a peak or is about to take off could have significant economic benefits. Identifying the causes of changepoints might let us predict future swings in the value of a stock. The Stocker object can automatically find the 10 largest changepoints for us.

microsoft.changepoint_date_analysis()**Changepoints sorted by slope rate of change (2nd derivative):**

	Date	Adj. Close	delta
48	2015-03-30	38.238066	2.580296
337	2016-05-20	48.886934	2.231580
409	2016-09-01	55.966886	-2.053965
72	2015-05-04	45.034285	-2.040387
313	2016-04-18	54.141111	-1.936257



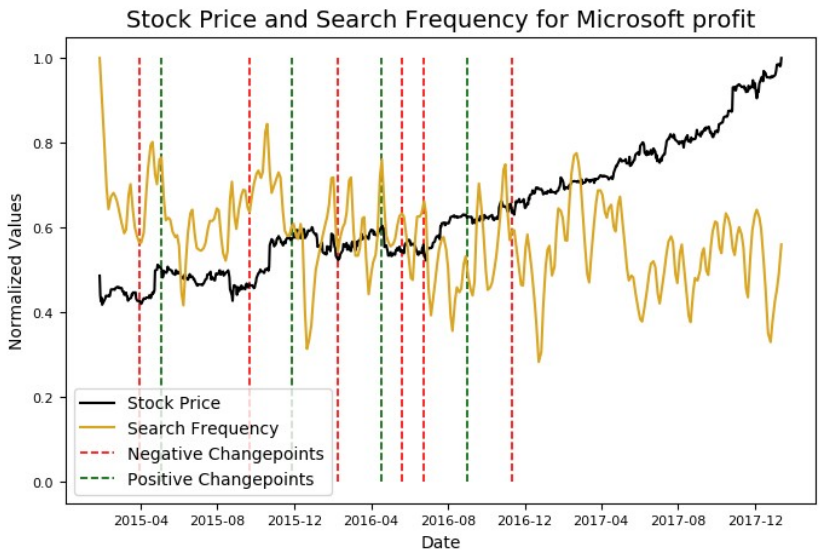
The changepoints tend to line up with peaks and valleys in the stock price. Prophet only finds changepoints in the first 80% of the data, but nonetheless, these results are useful because we can attempt to correlate them with real-world events. We could repeat what we did earlier and manually search for Google News around these dates, but I thought it would be preferable if Stocker did that for us. You might have seen the Google Search Trends tool which allows you to see the popularity of any search term over time in Google searches. Stocker can automatically retrieve this data for any search term we specify and plot the result on the original data. To find and graph the frequency of a search term, we modify the previous method call.

```
# same method but with a search term
microsoft.changepoint_date_analysis(search = 'Microsoft profit')
```
Top Related Queries:

	query	value
0	microsoft non profit	100
1	microsoft office	55
2	apple	30
3	microsoft 365	30
4	microsoft office 365	20

Rising Related Queries:

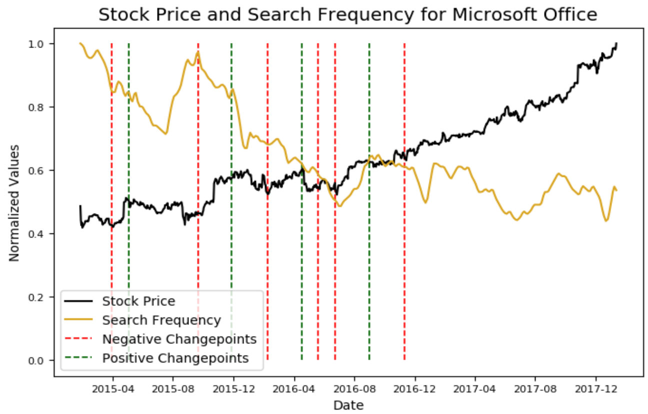
	query	value
0	microsoft 365	120
1	microsoft office 365	90
2	microsoft profit 2014	70



In addition to graphing the relative search frequency, Stocker displays the top related queries and the top rising queries for the date range of the graph. On the graph, the y-axis is normalized between 0 and 1 by dividing the values by their maximums, allowing us to compare two variables with different scales. From the figure, there does not appear to be a correlation between searches for “Microsoft profit” and the stock price of Microsoft.

Had we found a correlation, there would still be the question of causation. We would not know if searches or news caused the price to change, or if the change in price caused the searches. There might be some useful information to be found, but there are also many chance correlations. (For a humorous take on such random relationships, check out spurious correlations). Feel free to try out some different terms to see if you can find any interesting trends!

microsoft.changepoint_date_analysis(search = 'Microsoft Office')



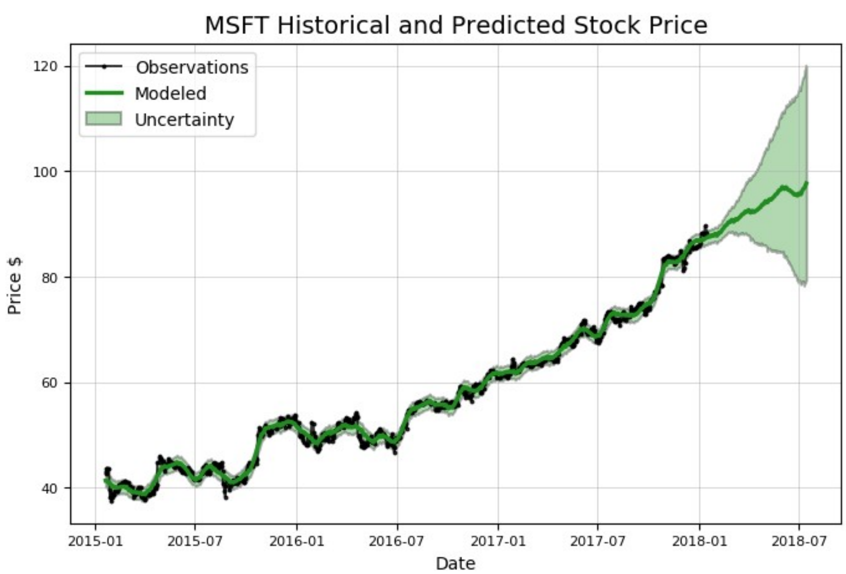
Looks like declining searches for Office leads to increasing stock prices. Maybe someone should let Microsoft know.

Predictions

We have only explored the first half of Stocker capabilities. The second half is designed for forecasting, or predicting future stock price. Although this might be a futile exercise (or at least will not pay off), there is still plenty to learn in the process! Stay tuned for a future article on prediction, or get started predicting with Stocker on your own (check out the documentation for details). For now, I'll leave you with one more image.

```
# specify number of days in future to make a prediction
model, future = microsoft.create_prophet_model(days=180)
```

Predicted Price on 2018-07-15 = $97.6



OUTPUT

There are multiple variables in the dataset – date, open, high, low, last, close, total_trade_quantity, and turnover.

1. The columns Open and Close represent the starting and final price at which the stock is traded on a particular day.

2. High, Low and Last represent the maximum, minimum, and last price of the share for the day.

3. Total Trade Quantity is the number of shares bought or sold in the day and Turnover (Lacs) is the turnover of the particular company on a given date.

Another important thing to note is that the market is closed on weekends and public holidays.Notice the above table again, some date values are missing – 2/10/2018, 6/10/2018, 7/10/2018. Of these dates, 2nd is a national holiday while 6th and 7th fall on a weekend.

The profit or loss calculation is usually determined by the closing price of a stock for the day, hence we will consider the closing price as the target variable. Let’s plot the target variable to understand how it’s shaping up in our data:

```
#setting index as date

df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')

df.index = df['Date']



#plot
```

```
plt.figure(figsize=(16,8))

plt.plot(df['Close'], label='Close Price history')
```

In the upcoming sections, we will explore these variables and use different techniques to predict the daily closing price of the stock.

Proposed Model

In this proposed system, we focus on predicting the stock values using machine learning algorithms like Random Forest and Support Vector Machines. We proposed the system "Stock market price prediction" we have predicted the stock market price using the random forest algorithm. In this proposed system, we were able to train the machine from the various data points from the past to make a future prediction. We took data from the previous year stocks to train the model. We majorly used two machine-learning libraries to solve the problem. The first one was numpy, which was used to clean and manipulate the data, and getting it into a form ready for analysis. The other which was used for real analysis and prediction. The data set we used was from the previous years stock markets collected from the public database available online, 80 % of data was used to train the machine and the rest 20 % to test the data. The basic approach of the supervised learning model is to learn the patterns and relationships in the data from the training set and then reproduce them for the test data. We used the python pandas library for data processing which combined different datasets into a data frame. The tuned up data frame allowed us to prepare the data for feature extraction. The data frame features were date and the closing price for a particular day. We used all these features to train the machine on random forest model and predicted the object variable, which is the price for a given day. We also quantified the accuracy by using the predictions for the test set and the actual values. The proposed system touches different areas of research including data pre-processing, random forest, and so on.

Conclusion

1. Python as a language has an enormous community behind it. Any problems that might be encountered can be easily solved with a trip to Stack Overflow. Python is among the most popular languages on the site which makes it very likely there will be a direct answer to any query.

2. Python has an abundance of powerful tools ready for scientific computing. Packages such as Numpy, Pandas, and SciPy are freely available and well documented. Packages such as these can dramatically reduce, and simplify the code needed to write a given program. This makes iteration quick.

3. Python as a language is forgiving and allows for programs that look like pseudo code. This is useful when pseudo code given in academic papers needs to be implemented and tested. Using Python, this step is usually reasonably trivial.

However, Python is not without its flaws. The language is dynamically typed and packages are notorious for Duck Typing. This can be frustrating when a package method returns something that, for example, looks like an array rather than being an actual array. Coupled with the fact that standard Python documentation does not explicitly state the return type of a method, this can lead to a lot of trials and error testing that would not otherwise happen in a strongly typed language. This is an issue that makes learning to use a new Python package or library more difficult than it otherwise could be.

REFERENCES

1. Ashish Sharma, Dinesh Bhuriya, Upendra Singh. "Survey of Stock Market Prediction Using Machine Learning Approach", ICECA 2017.
2. Loke.K.S. “Impact Of Financial Ratios And Technical Analysis On Stock Price Prediction Using Random Forests”, IEEE, 2017.
3. Xi Zhang1, Siyu Qu1, Jieyun Huang1, Binxing Fang1, Philip Yu2, “Stock Market Prediction via Multi-Source Multiple Instance Learning.” IEEE 2018.
4. VivekKanade, BhausahebDevikar, SayaliPhadatare, PranaliMunde, ShubhangiSonone. “Stock Market Prediction: Using Historical Data Analysis”, IJARCSSE 2017.

[i] Over-the-counter