Stock Market Prediction using Machine Learning Algorithms

CONSTANTIN – FLORIN IRIMIA

Primary Advisor: DR. XIWEI WANG

Secondary Advisor: DR. CRISTINA HAIDAU

April 2020

STOCK MARKET PREDICTION USING MACHINE LEARNING ALGORITHMS

By

CONSTANTIN – FLORIN IRIMIA

Primary Advisor: Dr. XIWEI WANG
Secondary Advisor: Dr. CRISTINA HAIDAU

REPORT

Presented to the Faculty of the Graduate School of Northeastern Illinois University
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Department of Computer Science

NORTHEASTER ILLINOIS UNIVERSITY

April 2020

Abstract

Stock markets are considered one of the best ways to keep our money since over the years the values have always increased, and it could generate very large profits with a low risk rate of return.

Prediction of the stock market is a very difficult task to realize as it involves a huge array of variables and as there is a lot of uncertainty among the markets even when the times are considered to be safe. In this paper I had used various machine learning algorithms such as Support vector machine, K-nearest neighbor and Artificial neural networks to create and to test models in order to predict stock market prices.

In this project I have tested the KNN model with different values for k, implemented the Support Vector Machine with three types of kernels: linear, polynomial and radial basis function and performed the comparison between those models with an artificial neural network model.

The predictions were made using the values of the whole months average from a year period of time. Accuracies were compared and it was predicted if the following month it will go up or down.

To create the prediction models, I have used the apple stock data for a period of one year downloaded from the yahoo finance website and calculated each month averages. Then I predicted using the above mentioned machine learning algorithms if the next month price will go up or down compared to the average of the whole year, which will be useful in a long term investment strategy; and if the next month price will go up or down compared to the last month, which can be used in a short term idea of investing into stock market.

Acknowledgements

I would like to thank and to express my sincere gratitude and thankfulness to Dr. Xiwei Wang and to Dr. Cristina Haidau which through their careful guidance helped me realizing this project which essentially is a bridge between theoretical ideas that I have learned in school and physical world.

Without their passion for teaching and their always availability, the road to completion of this project would have been much harder and not that fulfilling and rewarding.

I am also very thankful to the computer science and software engineering community in Chicago which proved itself a real help in researching and documenting process for this project.

I am highly obliged to thank to my family who supported me while implementing my ideas inside this document and who offered me support on all the other paths of life.

The process of realizing this project was a challenging one but very rewarding since I have expanded my area of knowledge and gained new skills which will help me in the career journey I am about to embark now.

Table of contents

Table of Figures

Introduction

The aim of this project is to predict the stock market price getting as input for the building of the prediction models the average price of the apple stock for the whole month. This is a very challenging task as the number of variables that have an impact on the direction of the stock price is exponential.

I could go and develop very complicated and intelligent models to predict the price; that would include the sentiment analysis of the particular stock brand taken from the financial newspapers or from the political speeches of the global leaders, which would have a huge impact on the prediction. Yet I have to keep in mind that even I can mold and put a big impact on the stock price by buying or selling the stocks. So, when I am thinking that I might have a perfect model and I would decide to buy a very large amount of stocks to make profits, I would influence the price of the stock.

So therefore, not even a perfect model would give us exactly a hundred percent accuracy in what we will be calling profitability. With these in mind I moved further on and developed three models that would predict the stock price and direction of the thirteenth month, getting in as input a whole year of data.

The models created implemented three different machine learning algorithms and produced similar results in terms of accuracy. All of them used classification algorithms where the whole data set is divided into parts; training set which usually makes up to 70 to 80 percent and the remaining which is names testing set. All the functions in algorithmically computations are applied on the training data set where the model is built. Then we compare the values predicted with the results in the test set, fact that will determine the accuracy and reliability of our model

I could analyze the markets in two ways when it comes about creating models to predict the trend. The first one is fundamental analysis and this includes factors such as performance or political views towards our targeted company or even random events that could have a huge impact on our stock; (e.g. public opinion switching from using carbon emission cars in favor of electrical vehicles, which could lead in a decreased value of a generic vehicle manufacturer stock).

All the variables that are building this first set of analysis are very hard to be taken as input when it comes to build prediction models for stock trends.

The second type of analysis is a technical analysis. This would take as input factors such as previous day price, the new open price in the new day, the volume of the stocks traded in that day, the highest value that the stock reached in a particular day, the lowest point in price and the adjusted close price. This is the analysis I have performed in this project since it gives more insights into the real data and tends to produce more accurate predictions.

Also, this is a much more manageable analysis to create in terms of algorithms and computational values since we already have the values while on the other side, we can't predict any of the out of financial sphere factors.

These are a few terms that we are going to reuse along this project:

- stock: equity stake of all the shareholders
- share: a part in the ownership of that particular company. It makes the owner of a share to be also the owner of a very small part of the profits and assets of a company
- assets: everything that the company owns
- dividends: a payment to shareholders that is made in shares rather than cash

These are the most important stock exchanges in the United States:

- NASDAQ: National Association of Security Dealers, with a market capital of $10.857 billions
- NYSE: New York Stock Exchange, with a market capital of $ 22.923 billions
- AMEX: American Stock Exchange

Statement of the problem

Stock market trading can be intimidating, especially for someone who has no knowledge in the financial world. Buying and selling stocks could be defined by some people as a way of betting on that particular company. Yet the more experience someone will get the more realizes that this process could be based on science and if it's done in the right way will only maximize the profits while the losses are close to zero. As the technology and access to it increases rapidly around the world, stock market also changes.

Anyone can buy stock and sell from the tip of their fingers on their smartphone as long as they are connected to the internet. Billions of transactions are made every hour around the world which makes iterative the building of some models that would use artificial intelligence and machine learning in helping us decide which stock to buy and which one to sell.

Scope of the project

Stock market prediction is a very challenging task and comes with a lot of risks. But at the same time is a very good way to keep out liquid assets as it could generate large amounts of profits and over a long period of time it could lead to independency in terms of work. Also, if we choose to keep our money in the stock markets, we would find that this is one of the easiest ways to liquidate almost instantly our assets which makes it one of the best ways.

Creating a model that would give us a perspective in what direction a stock price would take over a period of time is perfect since we would choose wisely in which particular company to invest in order to make profits. At the same time, it would help us in deciding what is the right moment to sell the stocks for keeping our profits and minimizing our losses.

Implementation

The Stock Market Prediction project was implemented by using the Agile methodology because this is one of the methods that provides great results in terms of speed and accuracy.
I have spent a good amount of time in defining the requirements and what exactly our project should do for the user as well as what would be the primary needs of the latter when would make use of our prediction models.

Fig 1: Gantt chart of the project development

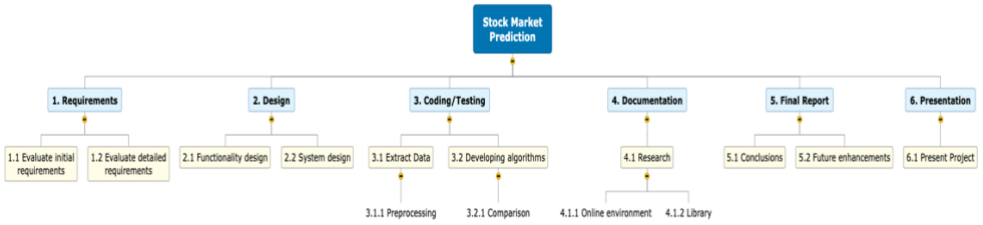


Fig 2: Work Breakdown Structure of the project development - WBS

PyCharm IDE

The following project was built in PyCharm IDE using a virtual environment where all the dependencies were created, and the libraries were imported. The data sets were already present on our local machine and were downloaded from the *yahoofinannce.com/historicaldata.*

Configuring the Python Interpreter

This was an essential step in creating the virtual environment as it basically hosts all our libraries and dependencies necessary for running and compiling our code.

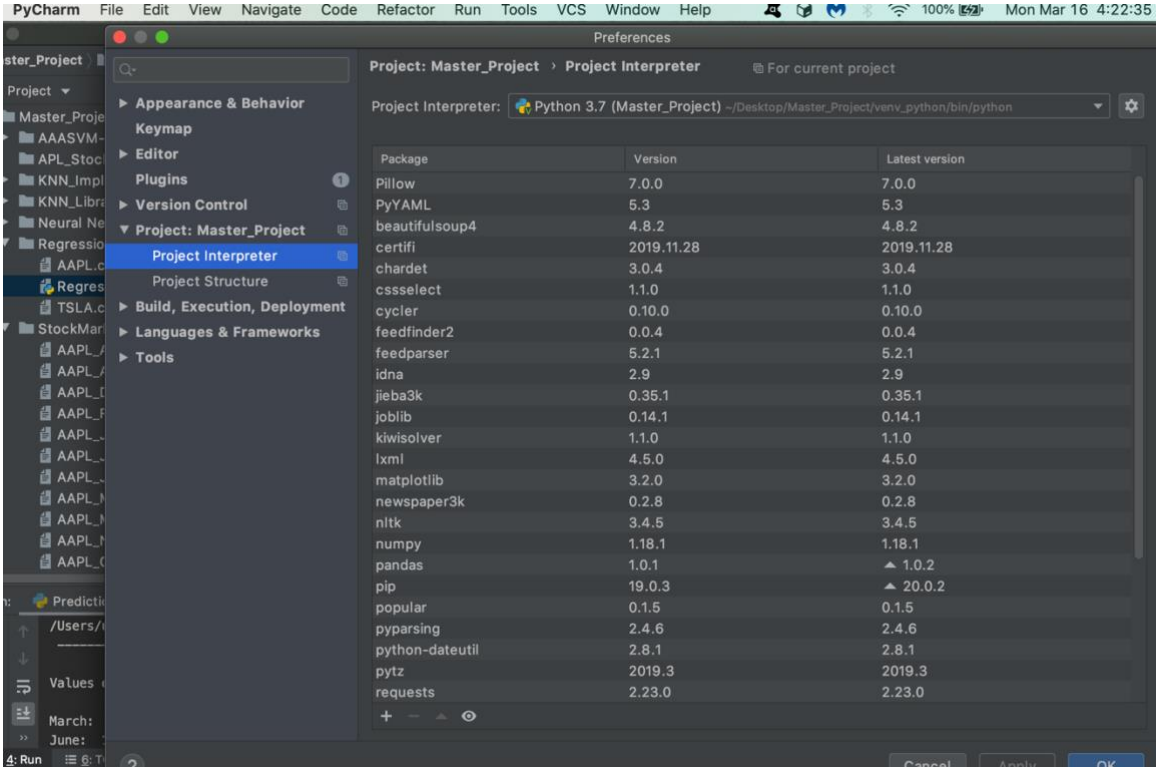


Fig 3: Configuring the virtual environment in PyCharm

After the new environment is selected:

type: `--system-site-packages`

in the *virualenv* terminal tool and select *make available to all projects.*

Libraries installed

I have imported the following libraries as needed to build our project:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import Ridge
from sklearn.neighbors import KNeighborsRegressor
from sklearn.pipeline import make_pipeline
from sklearn.svm import SVR
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import preprocessing
from scipy import stats
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM
import math
import matplotlib.pyplot as plt
```

Flow of the project

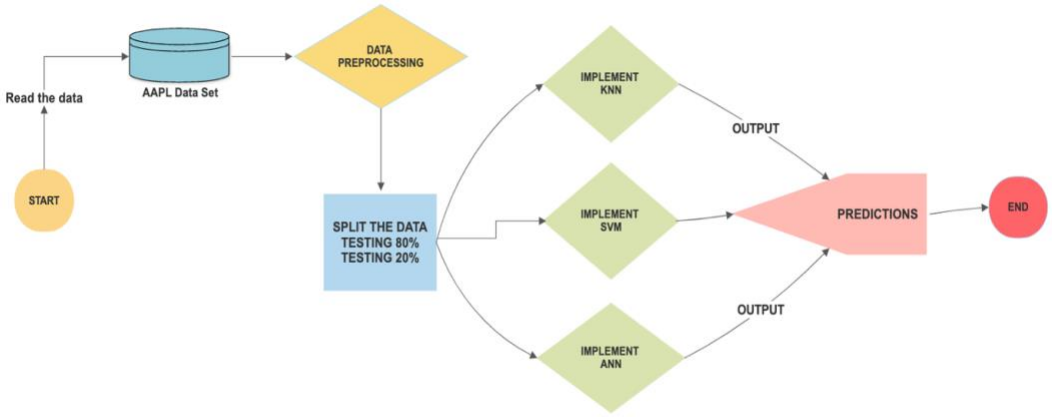


Fig 4: Flow of the project

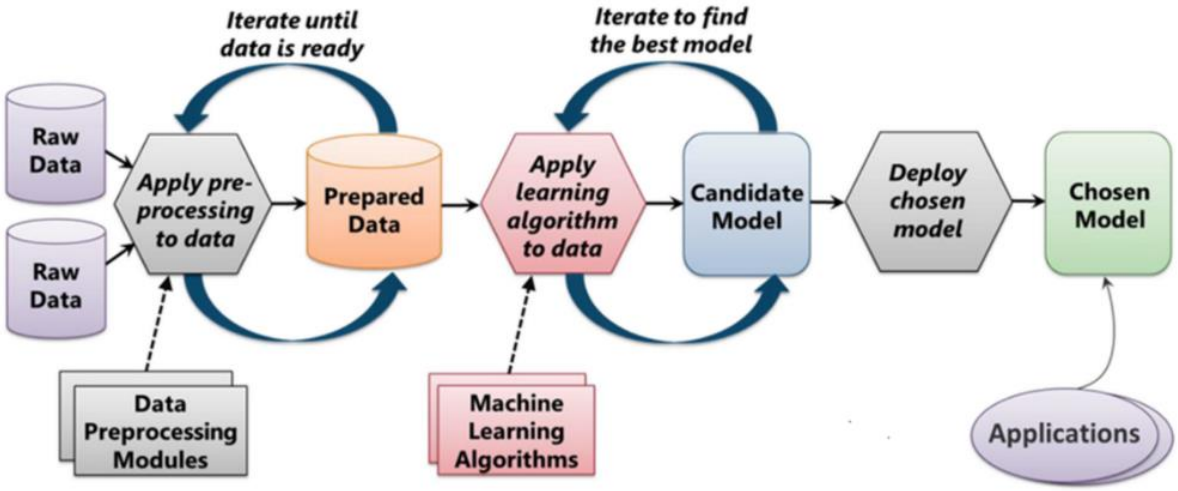


Figure 5: Machine Learning algorithms implementation process

Reading the data and calculating the averages.

I used the library pandas to read the csv file that was on the local machine on the same directory as the *file.py* .

This is done by invoking the method where we are passing as a parameter the name of the file and we are assingning the results to a variable

```
may = pd.read_csv('AAPL_May.csv')
```

Data attributes

Our data set has a number of seven attributes.

Date: representing the date of that particular transaction

Open: representing the price that the particular stock had it at 9:00 AM

High: the highest value that the stock reached in that particular day

Low: the lowest value that the stock reached in that particular day

Close: the cash value of that specific stock at day's end

Adj Close: reflects the closing price of the stock in relation to other stock attributes. E.g.: dividends

Volume: the total number of shares that were traded in that day

Below we can see a whole month of data, having 23 days since the stock market are open only on business days.

```
Date,Open,High,Low,Close,Adj Close,Volume
2019-08-01,213.899994,218.029999,206.740005,208.429993,206.529373,54017900
2019-08-02,205.529999,206.429993,201.630005,204.020004,202.159607,40862100
2019-08-05,197.990005,198.649994,192.580002,193.339996,191.576981,52393000
2019-08-06,196.309998,198.070007,194.039993,197.000000,195.203613,35824800
2019-08-07,195.410004,199.559998,193.820007,199.039993,197.225006,33364400
2019-08-08,200.199997,203.529999,199.389999,203.429993,201.574982,27009500
2019-08-09,201.300003,202.759995,199.289993,200.990005,199.913925,24619700
2019-08-12,199.619995,202.050003,199.149994,200.479996,199.406647,22474900
2019-08-13,201.020004,212.139999,200.479996,208.970001,207.851212,47218500
2019-08-14,203.160004,206.440002,202.589996,202.750000,201.664505,36547400
2019-08-15,203.460007,205.139999,199.669998,201.740005,200.659912,27227400
2019-08-16,204.279999,207.160004,203.839996,206.500000,205.394424,27620400
2019-08-19,210.619995,212.729996,210.029999,210.350006,209.223816,24413600
2019-08-20,210.880005,213.350006,210.320007,210.360001,209.233765,26884300
2019-08-21,212.990005,213.649994,211.600006,212.639999,211.501556,21535400
2019-08-22,213.190002,214.440002,210.750000,212.460007,211.322525,22253700
2019-08-23,209.429993,212.050003,201.000000,202.639999,201.555099,46818000
2019-08-26,205.860001,207.190002,205.059998,206.490005,205.384476,26043600
2019-08-27,207.860001,208.550003,203.529999,204.160004,203.066956,25873300
2019-08-28,204.100006,205.720001,203.320007,205.529999,204.429626,15938800
2019-08-29,208.500000,209.320007,206.660004,209.009995,207.890976,20990500
2019-08-30,210.160004,210.449997,207.199997,208.740005,207.622437,21143400
```

Fig 6: Example of data over a month

To calculate the average of the whole month I used the following formula:

$$Average = \frac{\sum_{i=0}^{n} |Adj\ Close|}{n}, where\ n = total\ number\ of\ days$$

After the completion of getting the average of each month I used the following function to get the total average of the year:

Input: total average prices of each month

Output: total yearly price average

foreach *price* in Prices **do**

```
        total = total + price
        count++
    return (total / count)
```

Normalizing the data using Z-Score Normalization

If a value would be equal to the mean of all the respective values of that feature the normalized value is exact 0. Otherwise, if the mean is smaller the result will be a negative number and if it is positive the output would be a positive number.

I made use of the scipy library when we computed the z-score normalization:

```
from scipy import stats
```

and it involves the following formula:

$$z = \frac{X - \mu}{\sigma}, \text{where } \mu = \text{mean, and } \sigma = \text{standard deviation}$$

Algorithms

K-nearest neighbor Implementation

KNN is a machine learning algorithm that used in creating similarity-based classifications. It is also named as a memory-based algorithm since the entire training dataset is viewed as the model.

When a prediction is required for a new instance, the algorithm will search through the whole training set and will return the k most similar instances.

In our case I built a model to predict the direction of the stock market using KNN and used the data to be mapped into vectors.

I have computed a similarity metric – *Euclidean Distance,* in order to make a decision and have applied a majority vote to select the k records to determine the class label. Since KNN is a

lazy learning algorithm which does not keep the model built before, it gives all the k-s of the training set that have the highest similarity to the testing set.

Here is the process that our KNN algorithm followed when applied to our data set:

- *Reading the data*
- *Determine the k: nearest neighbors*
- *Calculate the Euclidean distance*
- *Perform a sorting of all the training data based on distance results*
- *Implement it using the majority vote for the class label and make a prediction*

The accuracy of the model was calculated by implementing a function *getAccuracy()*, where a test set and a predictions test would be passed in as parameters. The formula used to determine the accuracy was:

number of how many correct predictions / *number of records in test set*

Support vector machine Implementation

Support vector machine is a supervised algorithm that is used to create prediction models. It could be used for both classification and regression. The way how the SVM works is separating the data points that are being taken as inputs as far as possible from the hyperplane; where the hyperplane is the plane that f=divides the data by their labels.

Here SVM can come in handy since it uses the kernel tricks to deal even with non-linear data, which will map the data into multi-dimensional space.

Some of the SVM kernel functions are:

-linear

-radial basis function

-polynomial

-sigmoid

-gaussian

I have used for our project to implement the SVM using three kernel functions: RBF, Linear and Polynomial.

The final classifier which is given by the SVM algorithm is resulted from the following function:

$$f(x) = \theta\left(\sum_{i=1}^{n} y_i \alpha_i K\left(x, x_i + b\right)\right)$$

Radial basis function kernel measures the distance between the input vector and the hyperplane vector such as:

$$exp\left(-\frac{|x - \mu|^2}{2\sigma^2}\right)$$

The Support vector machine in our project is trained by using the *fit()* method and then in order to get the result we are using the predict method from the SVM class where we are predicting the forecasted month.

Artificial neural network (ANN) -

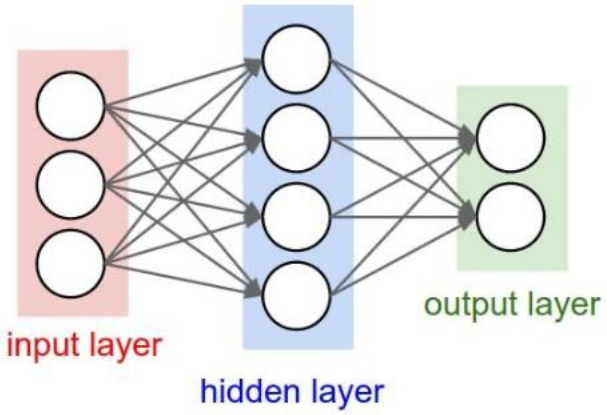
LSTM (Long Short-Term Memory) Implementation

Artificial neural networks are sum of computational operations inspired from the human brain. Biologists analyzed how the human brain access different process in order to learn something and tried to reproduce the same process to be applied on machine learning and create artificial intelligence.

These processes are composed of connected cells – neurons which are similar with the human brain neurons in a way that they take simple input data and perform tasks and passing the results to the next neurons. Each value of a specific task carries a weight and each of them are associated with a vector also known as biases.

Each of the weight is established in a training phase of the network where each of the neurons are learning how to identify specific classes of the data inputted.

Once the neurons have learned that skill, the whole network could be used to perform predictions and to classify new data sets



.

Fig 7: The artificial neural network process

In our program the numpy library is invoked to reshape the inputs that will go into training the neural network since they were initially in a 2-dimendsional form and we wanted them to be in a 3-dimensional form.

The above task was executed by calling the *reshape()* method.

The model uses for compiling a "*adam"* optimizer and "*mean squared error*" as loss.

The calculation is done in PyCharm by using the numpy library as follows:

$$np.sqrt(((a-b) ** 2).mean())$$

Results and evaluation

In our project we have implemented three different algorithms in order to build prediction models to be able to foresee the direction of the stock market. We have made use of the historical data from the financial portal *yahoofinannce* and we used the Apple stock - AAPL.

We got as input a whole year of data, where we have computed the averages of each month and tried to predict the thirteenth month.

We, then compared the results of each model, plot the graphs to see which one is more accurate and then used a function to get the closest price and predict if the following month will go up or down, compared to the average of the whole year.

The average accuracy is a major factor that would bring up issues when try to predict the stock market as the price fluctuates every minute and each stock is very volatile. All the fluctuations could happen without a valid reason or even actions or statements from a very big longitudinal or temporal distance could have influences on it.

For getting the accuracy of the of the models implemented we have used the method score found the libraries used:

$$score\left(X_{test,}\, Y_{test}\right)$$

, where *X_test* and *Y_test* are the test sets of the variables after we have trained the respective model and are array like objects. The above method calculates and returns the coefficient of determination R^2 of the prediction, where

$$R^2 = \frac{1-u}{v},$$

$$where\ u\ is\ regression\ sum\ of\ squares\ and\ v\ is\ residual\ sum\ of\ squares$$

We have trained all the models on the X-train values and Y-train, but the results could lead to overfitting. So therefore, we calculated the difference between Y-train and the model prediction for X-train.

So, in our methods to find the accuracy we have calculated the prediction score for X and Y which calculates Y' = prediction of X and then compared all the values in Y' with the values in Y. The score method is outputting a classification report of the values in Y test and our prediction/ A score of 0 would mean that our model would probably not be able to predict anything correctly. If a model would give a high accuracy on the training data but if tested on the test data would output a low accuracy (lower than 40%), that would probably mean that the model is overfitting.

Comparison of accuracy between prediction models:

No.	Prediction Model	Accuracy %	Average Accuracy %	
1	SVM *kernel:* rbf *kernel:* linear *kernel:* poly	 92. 035 81. 859 83. 179	85. 691	
2	ANN	65. 234	65. 234	
3	KNN	85. 451	85. 451	

Fig 8: Comparison between model accuracies

The accuracy was calculated on a scale from 0 to 1 and the result produced was multiplied by 100 to get the actual values. As we can see from the above table the best result was produced by the Support vector Machine Algorithm with an average of accuracy of 85.691 %, and the best model used a kernel trick as a radial basis function which had an accuracy of 92.035 %.

Therefore, all the indications given to the users will be given made on the support vector machine predictions. The user will have outputted all the results from the other models as well but he will make the decision on the type of the investment he will chose to make.

Here are the plotted results of each model and how it performs on our data set:

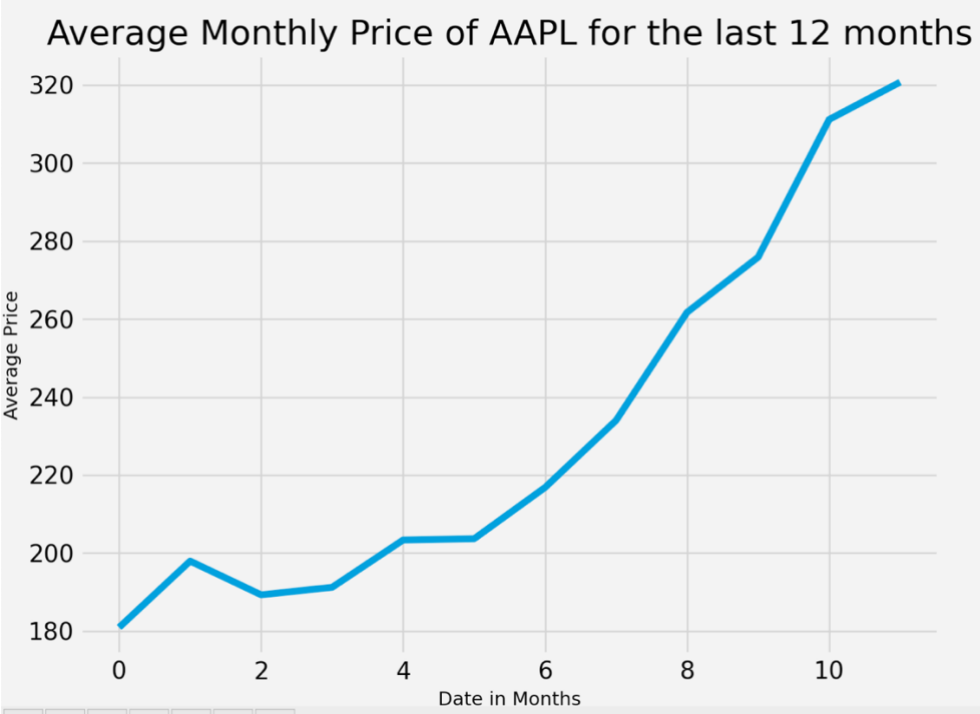


Fig 9: AAPL price over the past 12 months

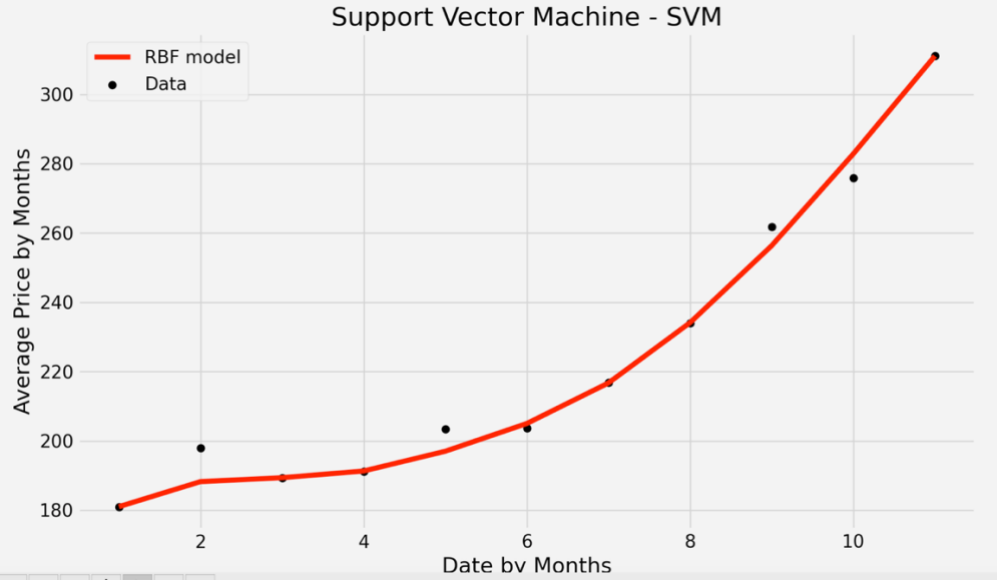


Figure 10: Support vector machine with radial basis function kernel

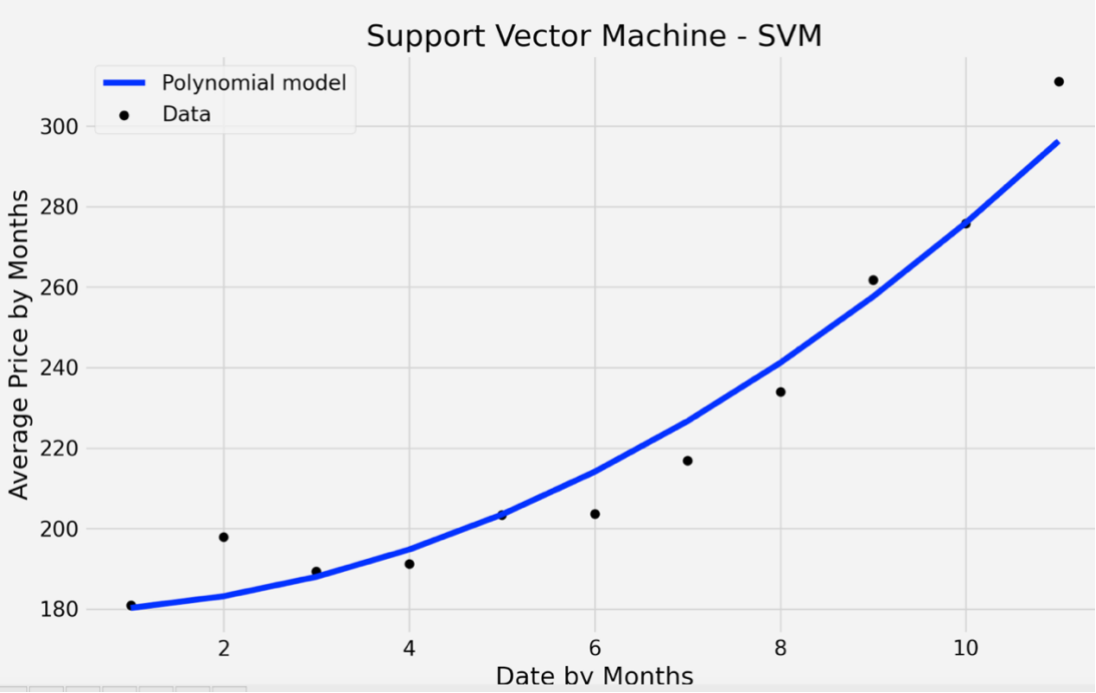


Figure 11: Support vector machine with polynomial kernel

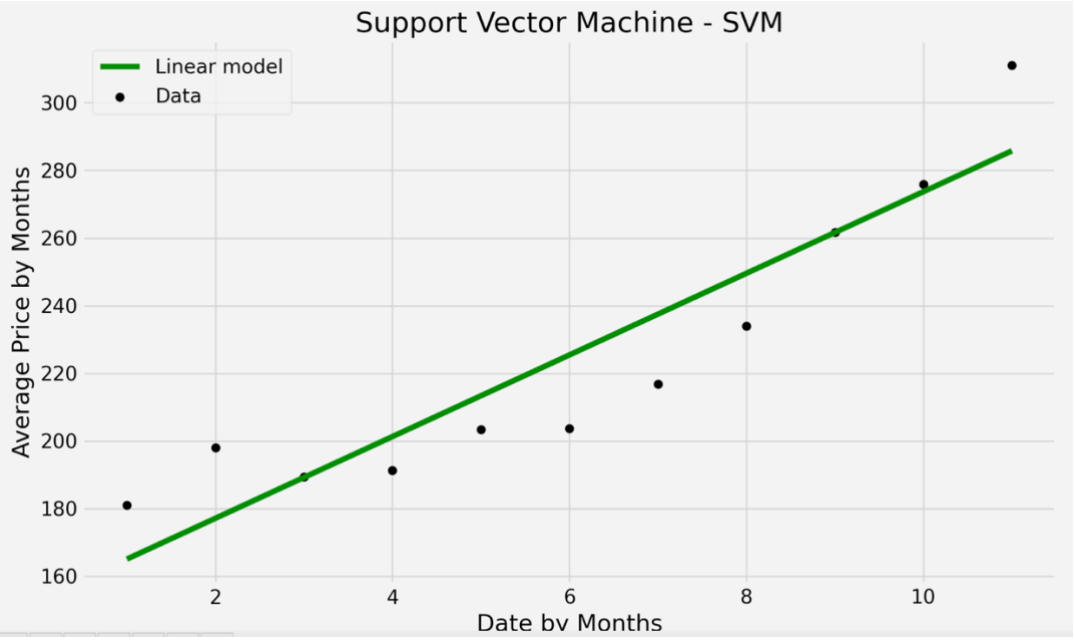


Figure 12: Support vector machine with linear kernel

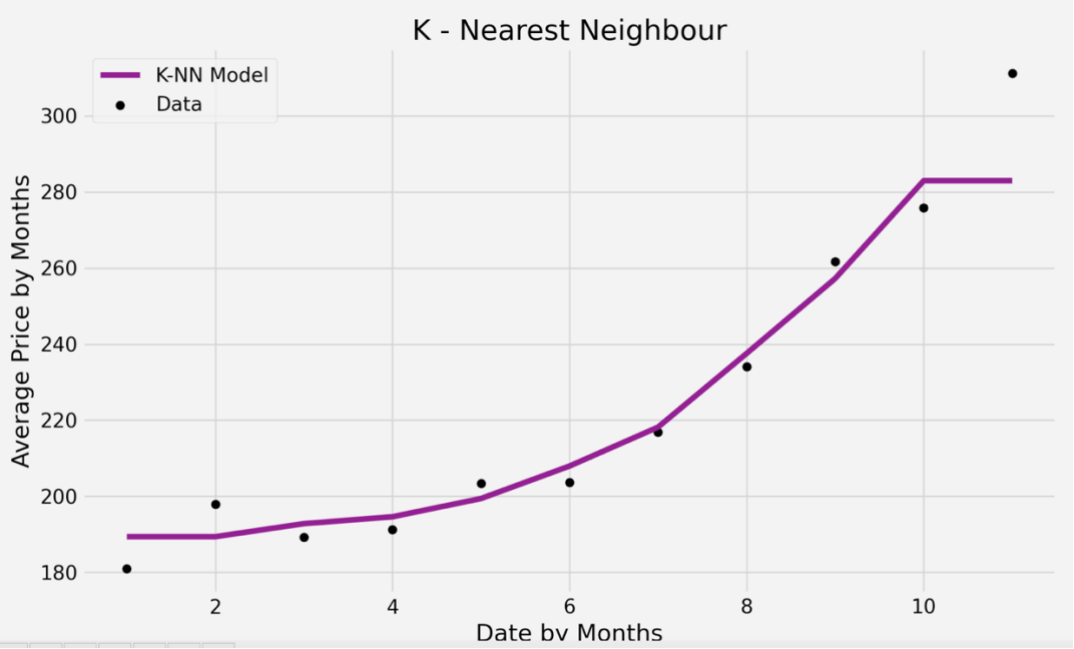


Figure 13: K nearest neighbor model

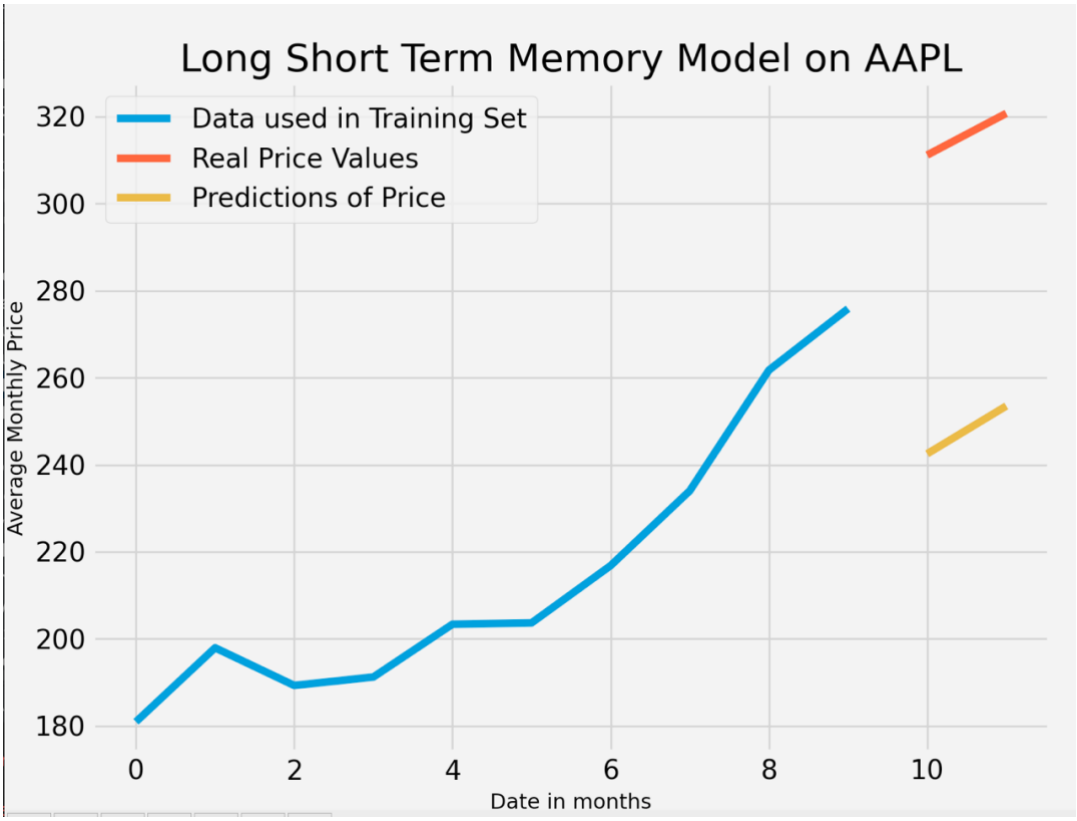


Fig 14: Long short-term memory model

Average monthly price of AAPL stock

Month	April	May	June	July	Aug	Sep	Oct	Nov	Dec	Jan	Feb	March
Average Price $	180	197	189	191	203	201	216	234	261	275	311	320

Price and direction prediction based on the annual average

Prediction model	Average price year ($)	Predicted Price April ($)	Direction (UP/DOWN)
SVM	232. 22	328. 77	UP
ANN	232. 22	256. 52	UP
KNN	232. 22	282. 93	UP

Price and direction prediction based on the last month price

Prediction model	Last month price ($)	Predicted Price April ($)	Direction (UP/DOWN)
SVM	320. 76	328. 77	UP
ANN	320. 76	256. 52	DOWN
KNN	320. 76	282. 93	DOWN

Since the model with the best accuracy is represented by the implementation of the support vector machine algorithm, the predicted price for the 13th month will be $ 328.77, and the direction is UP since the predicted price is greater than the average of total year.

If we decide to look at the short-term investment, we would see that in this case two of the prediction models predicted that the price is going to be smaller than the last month, so the direction is down.

The user would have to use the particular instance when using the models based on the intentions that he has when investing such as short-term investments or long-term investments.

Future enhancements

In the past years, along with developing of the internet and its accessibility to a larger amount of people the stock market itself is transitioning from the physical world to a new environment- virtually.

Anyone can trade stocks on their phone now and even the largest companies such as Charles Schwab are going on a 0% commission for its users since they have reduced at a high level their costs. Even if you have a simple account with Chase Bank, they will allow you to buy stocks and trade for free.

So therefore, since trading becomes more accessible to all of us it is imperative to make use of a specialized trained algorithmically models to help us deciding when to buy and when to sell. In fact, chase has an algorithmically model which is called *Robo-Trade* which, can help the new users to get the right decisions when trading at a commission fee for each transaction.

Since this project represents such a high interest for people, I would like to go further on in developing the respective prediction models to make them more stable and more reliable.

Sentiment analysis

A sentiment is a view or a thought in regard to something. In our case a sentiment would be a view on a particular company whose stock we are looking to trade (buy or sell). If the sentiment is negative, we would want to sell our stock before the price would go down or we would want to postpone buying it until the price would have gone lower.

Data mining offers a range of algorithms and techniques which we can use in order to extract and analyze a particular sentiment for a stock.

We could use the Twitter Sentiment Analysis library which is available in python and which invokes a simple classification algorithm such as Naïve Bayes Classification and indicates if a particular source of information about a stock outputs a positive, neutral or negative sentiment.

It uses the following formula:

$$sentiment = \frac{\sum(positive\ matches) - \sum(negative\ matches)}{\sum(positive\ matches) + \sum(negative\ matches)}$$

I would want as a further ad due to this project to implement the sentiment analysis using R language and NLP(Natural Processing Language), since it uses a determined list of positive and negative words and the comparison between the text in our articles(source of information) and NLP lists, outputs the sentiment.

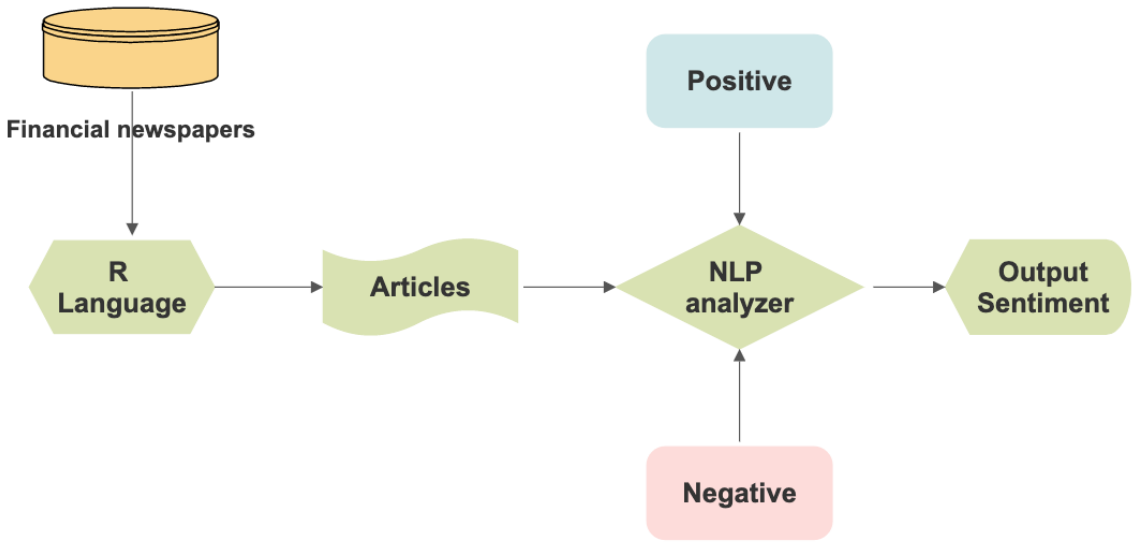


Fig 15: Sentiment analysis working diagram

Along with the historical data and the news sentiment analysis we can create a more stable and reliable prediction model.

User interface

In order to make the functionality of the model increase I would add a user interface which will be designed in a simple manner. I would to use a library such as *Tkinter* which is already available in Python 3.7 as incorporated or I would design a page in the same python directory where I would write java script code and customize it as preferred using Cascade Style Sheets - CSS.

If the second option is chosen, the file.py in the python directory is marked only as readable since the security measures will not allow modifications on it. In order to be able to make the front-end page to communicate with the back-end file located in the same directory I would first have to make the python file executable.

This is done by writing the following command in terminal:

```
sudo chmod -R 777 /path/to/your/file/or/directory
```

Then in the Java script file we would create a method that would prompt the execution of a back-end file(python), on a trigger event. This would be done by using the ..*ajax()* command found in the jQuery library.

Since the latest version of browsers have extra security measures implemented in order to restrain the access, setting operations have to be done to disable those features. The *./* command has to be put in front of the python name file to assure the accessibility and execution.

The function would be in the following format:

```
<script>
    function goPython(){
        $.ajax({
          url: "./myPythonFileName.py",
          context: document.body
        }).done(function());
    }
</script>
```

Fig 16: Function to make possible communication between front-end and back-end

Conclusions

Stock markets and trading are becoming more accessible to everyone and this process would increase the need of machine learning and artificial intelligence algorithms in building prediction models that we can use to maximize profits and minimize losses. In this project we have used data mining and machine learning algorithms to create models that would help us predict the output on a long run when it comes about the price of a stock.

I have used Python -*version 3.7* in a virtual environment where we imported some of the most useful libraries in order to implement the algorithms. For visualizing the results, I have made use of the *Mathplot* library existent in python.

As at this moment after the completion of the project and after the implementation of the prediction models we are able to see that accuracy is one of the biggest problems that engineers would face when try to predict a trend in the stock market. Having an enormous number of variables doesn't make things easy when it comes to predict the direction on where the price will go.

We were able to read very large amounts of historical data for the specific prices of the analyzed stocks, yet our models take in consideration only the past prices and create a prediction based on that; where the factors that are influencing the stock trends are various and situated in different temporal or positional dimensions.

In order to have our models produce reliable results that we could use in real life situations we would have to constantly update the training and testing data and monitors the accuracies outputted.

This project was a very good learning experience and it made me work with software and tools that I was not familiar with. Also, it made me strengthen my skills in communication and made me able to see the big image of a project and understand each phase of Software development lifecycle.

Reflections on learning

Working on this project offered me a huge opportunity in practicing self-learning and how to find alternative routes when we are hitting the wall in the learning process. I had no knowledge of python at the beginning of the semester yet through hours of self-study and under the careful guidance of the advisors I managed to build this application.

I have become proficient in using python and its very useful libraries such as pandas, numpy, sklearn or scikit. I learned how to integrate all these powerful algorithms for making my project a reliable model that I could improve in the future and use afterwards. Also, I became familiar with financial terms used in trading world which I could use further on as I am developing in my career

My research in a field where I had no knowledge in and writing the code in a language that I was not familiar with gave me a high boost in self confidence that with the proper mindset and with the necessary resources allocation I can develop anything; mindset which is in a very high demand for a future software engineer.

I was using chase bank for trading small amounts of money myself and I was always curios how their AI would advise you which stocks to buy and which to sell. After the completion of this project I understand the mechanisms and the algorithms that they use, and I feel more confident in communicating and sharing my knowledge.

The above project made use of all what I have learned over the past two years while I was in the graduate program and much more it gave me a perspective that implies self-study along with self-discipline, things that are extremely useful to have a successful career.

References

- BREIMAN, L., FRIEDMAN, J., STONE, C., J., & OLSHEN, R., A. (1984). Classification and regression trees, CRC Press.

- ZACHARSKI, RON (2015), A Programmer's Guide to Data Mining: The Ancient Art of the Numerati, www.guidetofatamining.com

- HUANG, C.-L., & TSAI, C.-Y. (2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. Expert Systems with Applications.

- HAN, J., KAMBER, M., & PEI, J. (2006). Data Mining: Concepts and techniques. Morgan Kaufmann.

- KHAN, W., GHAZANFAR, M., A., ASAM, M., IQBAL, A., AHMAD, S., JAVED. (2016). Predicting Trend in Stock Market Exchange Using Machine Learning Classifiers. http://www.sci-int.com/pdf/636327843900768486.pdf

- SRIVASTAVA, DURGESH K., LEKHA, BHAMBHU. (2010). Data classification using support vector machine. ACM Press.

- HUANG, WEI, Y. NAKAMORI, S., WANG. (2005). Forecasting stock market movement direction with support vector machine. Computers & Operations Research.

- SHEN, SHUNRONG, H., JIANG, T., ZHANG. (2014). Stock market forecasting using machine learning algorithms. Journal of Computational Science

- W3SCHOOLS, PYTHON. (2018). Python Tutorials. https://www.w3schools.com/python/numpy_array_indexing.asp

- DOCS PYTHON, PYTHON. (2019). Python 3 Tutorial. https://docs.python.org/3/tutorial/

- TUTORIALS POINTS, PYTHON. (2020). Python 3.7. https://www.tutorialspoint.com/python/index.htm

- TOWARDS DATA SCIENCE, PYTHON 3.7. (2020). Stock Predictions in Python. https://towardsdatascience.com/stock-prediction-in-python-b66555171a2