Stock Price Prediction

Group ID: G02

Nazat Al Humaidy (2016-2-60-010)

Md.Farhaduzzaman (2017-1-60-128)

Nazifa Bhuiyan (2017-2-60-138)

Sharaf Anjum Khan (2016-2-60-021)

Januray 6, 2021

Contents

1 ABSTRACT:

Stock price forecasting is a popular and important topic in financial and academic studies. Share Market is an untidy place for predicting since there are no significant rules to estimate or predict the price of share in the share market. Many methods like technical analysis, fundamental analysis, time series analysis and statistical analysis, etc are all used to attempt to predict the price in the share market but none of these methods are proved as a consistently acceptable prediction tool.

This project predicts the price of stocks using two different machine learning algorithms so far, one is called Support Vector Regression (SVR) and the other is Linear Regression. In the past decades, there is an increasing interest in predicting markets among economists, policymakers, academics, and market makers. The objective of the project is to study and improve the supervised learning algorithm to predict the stock price.

Keywords: SVR, Regression, Machine Learning, Supervised learning

2 Introduction:

The prediction of a stock market direction may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. This project predicts the price of stocks using two different machine learning algorithms so far, one is called a Support Vector Regression (SVR) and the other is Linear Regression.

A Support Vector Regression (SVR) is a type of Support Vector Machine and is a type of supervised learning algorithm that analyzes data for regression analysis. We will be using C-classification Support Vector Machine with RBF Kernel. The model produced by SVR depends only on a subset of the training data because the cost function for building the model ignores any training data close to the model prediction.

Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).

2.1   OBJECTIVE:

The prediction of a stock market direction may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. This project predicts the price of stocks using two different machine learning algorithms so far, one is called a Support
Vector Regression (SVR) and the other is Linear Regression

A Support Vector Regression (SVR) is a type of Support Vector Machine and is a type of supervised learning algorithm that analyzes data for regression analysis. We will be using C-classification Support Vector Machine with RBF Kernel. The model produced by SVR depends only on a subset of the training data because the cost function for building the model ignores any training data close to the model prediction.

Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).

2.2   MOTIVATION:

Predicting anything is always very interesting, Machine learning helps us very much in this sector, it can easily generate values of the future with the help of many approaches. There is an increasing interest in predicting markets stock price, that's why we choose to do this project.

2.3   EXISTING WORKS:

Recently, a lot of interesting work has been done in the area of applying Machine Learning Algorithms for analyzing price patterns and predicting stock price. Most stock traders nowadays depend on Intelligent Trading Systems which help them in predicting prices based on various situations and conditions.

Recent researches used input data from various sources. Some systems use historical stock data, some use financial news articles, some use expert reviews while some use a hybrid system that takes multiple

inputs to predict the market. Also, a wide range of machine learning algorithms is available that can be used to design the system. These systems have different approaches to solve the problem. Some systems perform mathematical analysis on historic data for prediction while some perform sentiment analysis on financial news articles and expert reviews for prediction. However, because of the volatility of the stock market, no system has a perfect or accurate prediction.

2.4   NECESSITY:

Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in irrational behavior etc. To implement a mix of machine learning algorithms to predict the future stock price of any company and that's the reason someone should invest his/her to develop this project.

2.5   SYSTEM OVERVIEW:

This system named "Stock Market Prediction using Machine Learning" is an application that aims to predict stock market value using Machine Learning Approaches. This project is intended to solve the economic dilemma created by individuals that want to invest in the Stock Market.

2.6   FLOW CHART:

```
                    ┌─────────────────┐
                   (      Start        )
                    └─────────────────┘
                            │
                            ▼
                    ╱─────────────────╱
                   ╱    Read Data      ╱
                  ╱─────────────────╱
                            │
                            ▼
                    ┌─────────────────┐
                    │ Data processing │
                    └─────────────────┘
                            │
                            ▼
                    ┌───────────────────┐
                    │ Model implementation │
                    └───────────────────┘
                            │
                            ▼
                    ┌─────────────────┐
                    │ Divide dataset into │
                    │ training and testing │
                    └─────────────────┘
                            │
                            ▼
                    ┌─────────────────┐
                    │ Train the dataset using │
                    │ training dataset │
                    └─────────────────┘
                            │
                            ▼
                    ┌─────────────────┐
                    │ Test the dataset using │
                    │ testing dataset │
                    └─────────────────┘
                            │
                            ▼
                    ╱─────────────────╱
                   ╱ Display the output ╱
                  ╱─────────────────╱
                            │
                            ▼
                    ┌─────────────────┐
                   (       Stop        )
                    └─────────────────┘
```
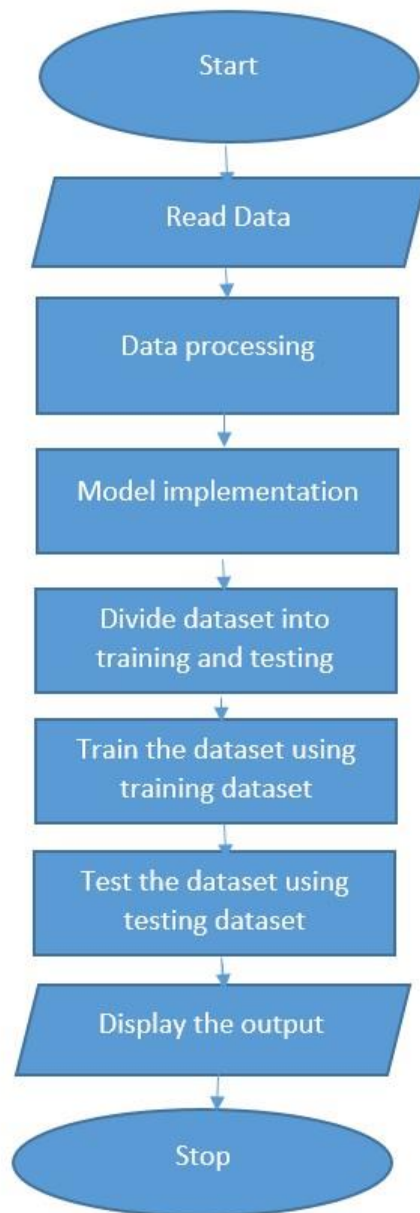
3   Methodology:

3.1   HOW TO COLLECT INPUT DATA?

Input data is taken from Yahoo Finance using the following steps:

1.   For our project, we are getting stock data AAL (American AirlinesGroup Inc.) from yahoo finance.

2.   Use stock's ticker symbol from step a to get data from Yahoo Finance.

3.   We can choose the stock duration, in here we consider 40days of stock data.

4.   Further, we divide the data into two parts, training data and testing data, where 80 percent of the data will be used for training and 20 percent of the data will be used for testing.

3.2   HOW TO SOLVE THE PROBLEM?

We will solve the problem using below supervised learning techniques to build our model –

• Linear Regression.

• Support Vector Machine.

To solve the problem, we will follow the below steps –

1.   Using dataset or We can import data from the web of any duration time.

2.   Provide the data to the system.

3.   Train the system.

4.   The system will predict the output.
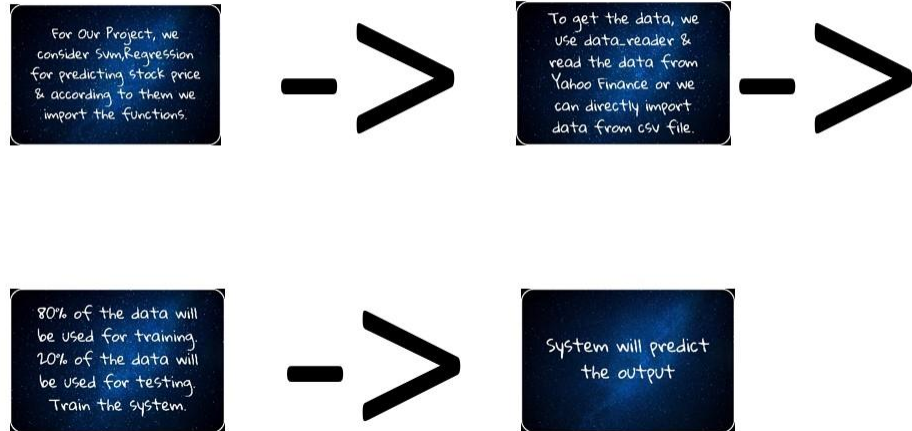
4 Hypothesis:

4.1 POSITIVE/NEGATIVE HYPOTHESIS:

The performance of the stock depends on various factors. If the tasting model works properly then we can obtain satisfactory prediction results.

5 Implementation:

This program predicts the prices of a stock 10 days in the future based on the current Adjusted Close price. We can easily change the number of days to predict the prices of a stock.

First, importing the functions, that will make this program easier to write. We are importing the machine learning library sklearn, pandas, matplotlib and numpy.

5.1 Import the functions:

import math
import numpy as np

```
import pandas as pd
import pandas datareader as web
from sklearn.linear model import LinearRegression
from sklearn.svm import SVR
from sklearn.model selection import train test split
import matplotlib.pyplot as plt
plt.style.use('_vethirtyeight')
```

Now, creating a variable called N, to store the number of days (10 days) into the future that we want to predict. We also need a column (the dependent variable) that will hold the predicted price values 10 days into the future. So, we will create a new column called 'Prediction' and populate it with data from the Adj Close column and then print the new data set. Since we shifted the data up 10 rows, so the last 10 rows of data for the new column 'Prediction' will be empty 'NaN' (Not A Number).

5.2   DATA COLLECTION:

we can get the stock data from the web or import it from a CSV file, and take a look at the data set. Here we are getting stock data AAL (American Airlines Group Inc.) from yahoo finance, we can get data from any market such as AMD, BAC, MSFT, DAL, etc. We can also choose any duration of time, in here we choose 2month recent data. Now, storing it into a variable called 'df' which is short for the data frame, and printing the data.

5.2.1   Get the data:

df = web.DataReader('AAL', data-source='yahoo', start='2020-02-01', end='2020-03-30') or, df = pd.read-csv('D:/data/AAL.csv')

5.3   DATA PROCESSING:

We only need the Adjusted Close (Adj Close) price. First, we can check the data shape. Then we deduct all columns excepting Adjusted Close (Adj Close) price.

5.3.1   Calculate Size:

df.shape

      (40, 6)

5.3.2   Get the Adjusted Close Price :

df = df[['Adj Close']]

5.3.3   Visualization :

plt.figure(figsize=(16,8)

     plt.title('Adj Close Price History')
     plt.plot(df['Adj Close'])
     plt.xlabel('Date', fontsize=18)
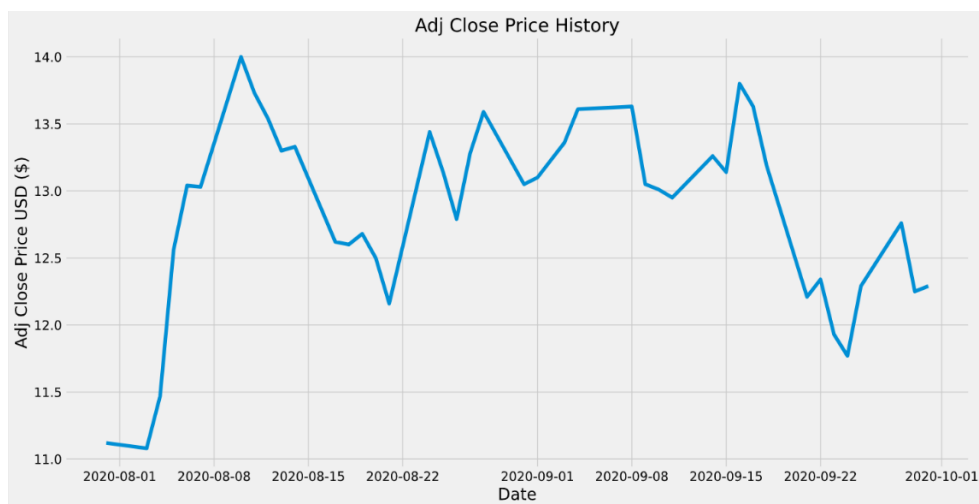   plt.ylabel('Adj Close Price USD ()', fontsize=18)
     plt.show()

Figure 1: Adj Close Price History Vs. Adj Price USD ($)

Now, creating a variable called N, to store the number of days (10 days) into the future that we want to predict.

We also need a column (the dependent variable) that will hold the predicted price values 10 days into the future. So, we will create a new column called 'Prediction' and populate it with data from the Adj Close column and then print the new data set. Since we shifted the data up 10 rows, the last 10 rows of data for the new column 'Prediction' will be empty 'NaN' (Not A Number).

5.3.4 A variable for predicting 'n' days out into the future:

N = 10 ,N=10 days

5.3.5 Create another column shifted 'n' units up:

df['Prediction'] = df[['Adj Close']].shift(-N)
5.3.6 print the new data set:

print(df.tail()) Next, we create the independent data set (X). This is the data set that we will use to train the machine learning models. Convert the data into a numpy (np) array after dropping the 'Prediction' column, then store this new data into 'X'. Then we will remove the last 10 rows of data from 'X', and store the new data back into 'X'.

5.3.7 Create an independent data set (X) :

5.3.8 Convert the data frame to a numpy array:

X = np.array(df.drop(['Prediction'],1))

5.3.9 Remove the last '10' rows:

X = X[:-N]
print(X)
We created the independent data set in the previous step, now we will create the dependent data set called 'y'. This is the target data, that holds the future price predictions.

We will convert the data frame into a numpy array and from the 'Prediction' column, store it into a new variable called 'y' and then remove the last 10 rows of data from 'y'.

5.3.10   Create the dependent data set (y):

5.3.11   Convert the dataframe to a numpy array :

y = np.array(df['Prediction'])

5.3.12   Get all of the y values except the last '10' rows:

y = y[:-N]
        print(y)
        Now we have the new cleaned and processed data sets 'X' 'y'. we can split them up into 80 percent training and 20 percent testing data for the

        models.

5.3.13 Split the data into 80 percent training and 20 percent testing:

x-train, x-test, y-train, y-test = train-test-split(X, y, test-size=0.2)
        Here, we created and trained the models. We will create and train the Support Vector Machine. We used the'rbf' kernel.
5.4   MODEL IMPLEMENTATION:

5.4.1   Create and train the Support Vector Machine (Regressor) :

svr-rbf = SVR(kernel='rbf')

5.4.2   Train the model:

svr-rbf.fit(x-train, y-train)

        The model is tested by getting accuracy. The best possible accuracy is 100 percent, and the model returns a score of 63.09 percent.

5.4.3   Testing Model:

svm-accuracy   =   svr-rbf.score(x-test,   y-test)
        print("svm accuracy: ", svm-accuracy)

        svm accuracy: 63.09284382167379
        Next, we will create train the Linear Regression model.

5.4.4   Create and train the Linear Regression Model

lr = LinearRegression()

5.4.5   Train the model

lr.fit(x-train, y-train)
          The model is tested by getting accuracy. The best possible accuracy is 100 percent, and the model returns a score of 26.29 percent.

5.4.6   Testing Model

lr-accuracy = lr.score(x-test, y-test)
print("lr accuracy: ", lr-accuracy)
lr accuracy: 26.29263060791923
          In this case, the Linear Regression model will be better to use to predict the future price.

5.4.7   Set x-N equal to the last 10 rows of the original data set from Adj. Close column

x-N = np.array(df.drop(['Prediction'],1))[-N:]
       print(x-N
)
          Finally, we will print out the future price (next 10 days) predictions of stock using the linear regression model, and then print out the stock price predictions for the next 10 days of the support vector machine using the x-N data

5.5   RESULTS:

In this paper, we get the result of SVM and Regression, First, we generate the accuracy of both algorithms to identify the most preferable result. In our result, we printed the future 10days predicted price. We can also predict old data and can match them with actual old data, where we can differentiate.

Date	Actual price	predicted price using regression	Predicted price using SVM
3/31/2020	12.25	11.8516714	13.76974332
4/1/2020	12.19	8.98574071	14.66524868
4/2/2020	10.69	7.99396871	15.09135557
4/3/2020	10.06	8.05960081	15.06427704
4/4/2020	9.39	7.96479897	15.10330908
4/5/2020	9.5	10.64112561	14.02076705
4/6/2020	10.22	11.71311493	13.78555362
4/7/2020	11.33	11.91001053	13.76413708
4/8/2020	12.51	10.72863484	13.99463086
4/9/2020	11.56	9.42328757	14.4763295

5.6   TOOL USED

Languages used: Python

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch

applications and manage conda packages, environments. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, MacOs, and Linux.

> The following applications are available by default in Navigator

- JupyterLab
- Jupyter Notebook
- Spyder
- Visual Studio Code

Among them, we use to run our code on Jupyter Notebook. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

6 SUMMARY AND CONCLUSION

In this paper, we study the use of regression and support vector machines to predict financial movement direction. Of both these algorithms, we saw that Linear Regression gave us better results because of better accuracy. In this report, this is a clear message for financial traders, which can lead to a capital gain. However, each method has its own strengths and weaknesses. Also, a particular Machine Learning Algorithm might be better suited to a particular type of stock, say Technology Stocks, whereas the same algorithm might give lower accuracies while predicting some other types of Stocks, says Energy Stocks.

6.1 CHALLENGES:

Each method has its own strengths and weaknesses. A particular Machine Learning Algorithm might be better suited to a particular type of stock. Stock price predicting using machine learning is always a challenging task because predicting something using any tool or function

is always not accurate with the future. Though machine learning approaches makes the problem much easier.

6.2   LIMITATIONS:

We implement the SVM and Regression to the task of stock market prediction. Our initial analysis showed a significant correlation between different input parameters. The result obtained in both cases was fairly accurate. The main aim of this system is to provide a predicted price of where the stock market is headed. It is only limited to a very basic prediction model. Thus, it cannot be used as a critical decision making. We only implemented the SVM and Regression accuracy. Other approaches like Decision Tree, Neural Network may work differently. In prediction, which approach is more accurate, we did not know. So, there is always a limitation in predicting price.