Indian Institute of Information Techonlogy Pune

Department of Computer Science Engineering

2022-2023

Machine Learning Project
on
"Stock Market Prediction using Machine Learning and Python"
By

G. Manoj Reddy (112015054)
R. Suraj Rao (112015108)
U. Vinod Kumar (112015158)
V. Bharadwaj (112015160)

Mentor :- Amruta Lipare

Stock Market Prediction using Machine Learning and Python

- **Problem Statement:** This Project is based on Machine learning algorithms like Linear Regression , Moving Average , Decision Tree to predict the stock market prices.

Outline

- Introduction
- Algorithms used
- Methodology
- Flow chart
- Code
- Experiment result
- Conclusion
- References

Introduction

- Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction – physical factors vs. physiological, rational and irrational behavior, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy.

- It is to determine the calculation of market closing index for a given trading day efficiently such that the customer can decide whether to buy or sell or hold.

- Here comes Machine Learning as a game changer in this domain. Using features like the latest announcements about an organization, their quarterly revenue results, etc., machine learning techniques have the potential to unearth patterns and insights we didn’t see before, and these can be used to make accurate predictions.

Algorithms used

- Linear Regression
- Decision Tree
- Moving average

Linear Regression

- Linear regression is a statistical approach that models the relationship between input features and output.
- The purpose of the linear regression function is to find a line that is closest from all data points so that whenever we want to calculate the prediction for a new dependent variable we can pick the subsequent point on the line corresponding to the independent variable on X axis.
- We predicted the output based on only one input feature.
- Simple linear regression is given by

$$y = m*x + c$$

Where ‘c’ = constant or y intercept of line
‘m’ = coefficient of input feature
‘x’ = input feature in which output is based
‘y’ = output

Decision Tree

- Decision trees in Machine Learning are used for building classification and regression models to be used in data mining and trading.
- A decision tree algorithm performs a set of recursive actions before it arrives at the end result and when you plot these actions on a screen, the visual looks like a big tree, hence the name ‘Decision Tree’.
- Basically, a decision tree is a flowchart to help you make decisions. Machine Learning uses the same technique to make better decisions
- In stock market prediction we use decision tree by classifying closing price considering remaining attributes like open price ,high price ,low price ,volume etc.,

Moving Average

- The predicted closing price for each day will be the average of a set of previously observed values.
- Instead of using the simple average, we will be using the moving average technique which uses the latest set of values for each prediction.
- In other words, for each subsequent step, the predicted values are taken into consideration.
- It is an equally weighted mean of previous n data.
- Here we use the formula:

  closing prices are Pm,Pm-1,……Pm-(n-1),

  ‘Psm’ is the average of previous values

$$\begin{aligned}\bar{p}_{\mathrm{SM}} &= \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}\\ &= \frac{1}{n}\sum_{i=0}^{n-1} p_{M-i}.\end{aligned}$$

- When calculating successive values, a new value comes into the sum, and the oldest value drops out, meaning that a full summation each time is unnecessary for this simple case:

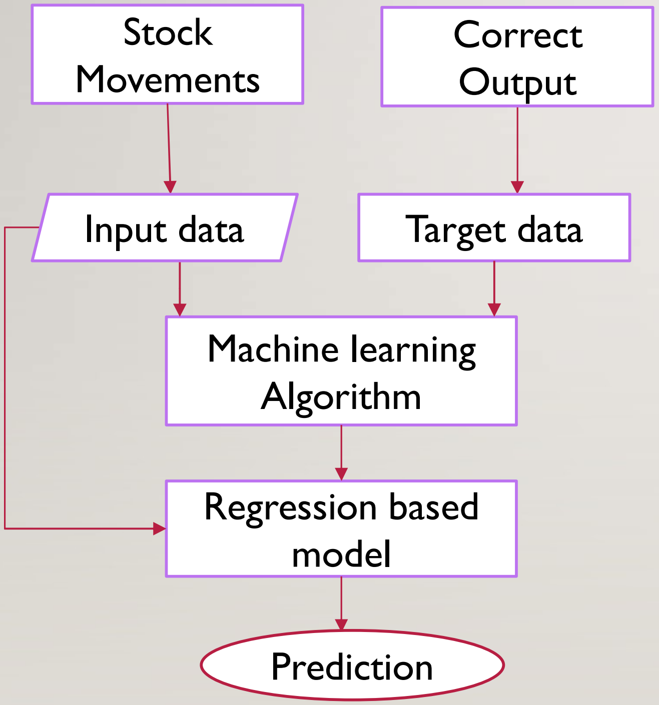
$$\bar{p}_{\mathrm{SM}} = \bar{p}_{\mathrm{SM,prev}} + \frac{1}{n}(p_M - p_{M-n}).$$

Methodology

- First we imported all libraries that are required such as pandas , numpy ,sklearn ,matplotlib.
- Next we read the data using panadas library from .csv file
- Then Visualized the data and get the closing price from data and created a variable to predict ‘x’ days out into the future.
- Now we created the target data and split the data into training and testing datasets.
- There after we implemented the models like decision tree , linear regression and printed the predicted values respectively and plotted the predicted values against original ones.
- For Moving average technique we again loaded the data and defined the moving average function and printed the predicted values for each day and plotted them in graph against original data.

Flow Charts

Linear Regression model



Stock Movements

Correct Output

Input data

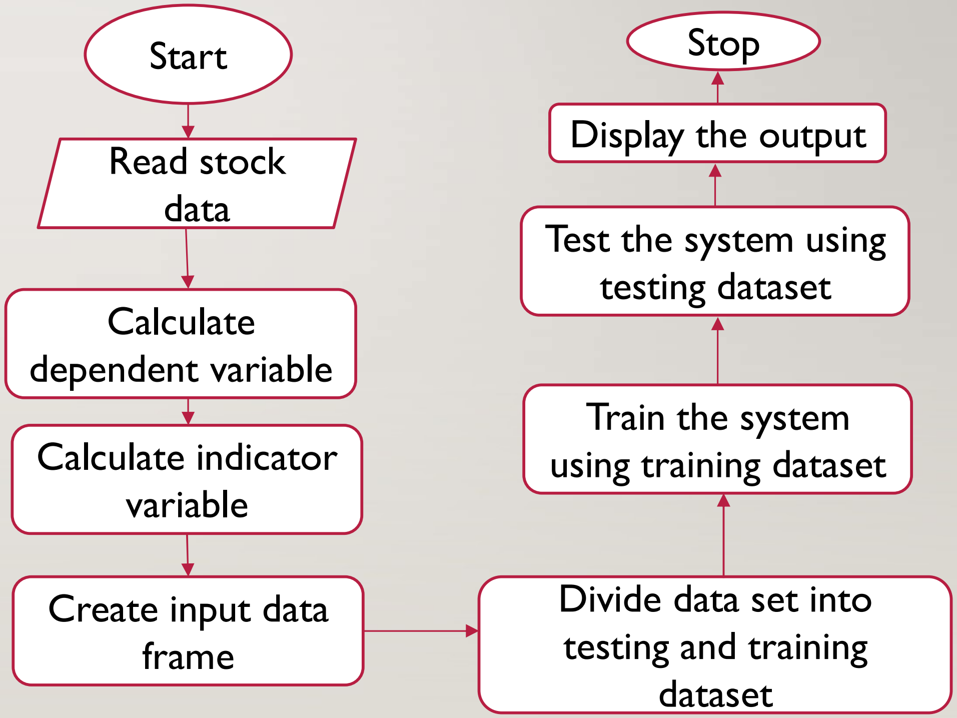
Target data

Machine learning Algorithm

Regression based model

Prediction

Decision Tree Model



Start

Read stock data

Calculate dependent variable

Calculate indicator variable

Create input data frame

Divide data set into testing and training dataset

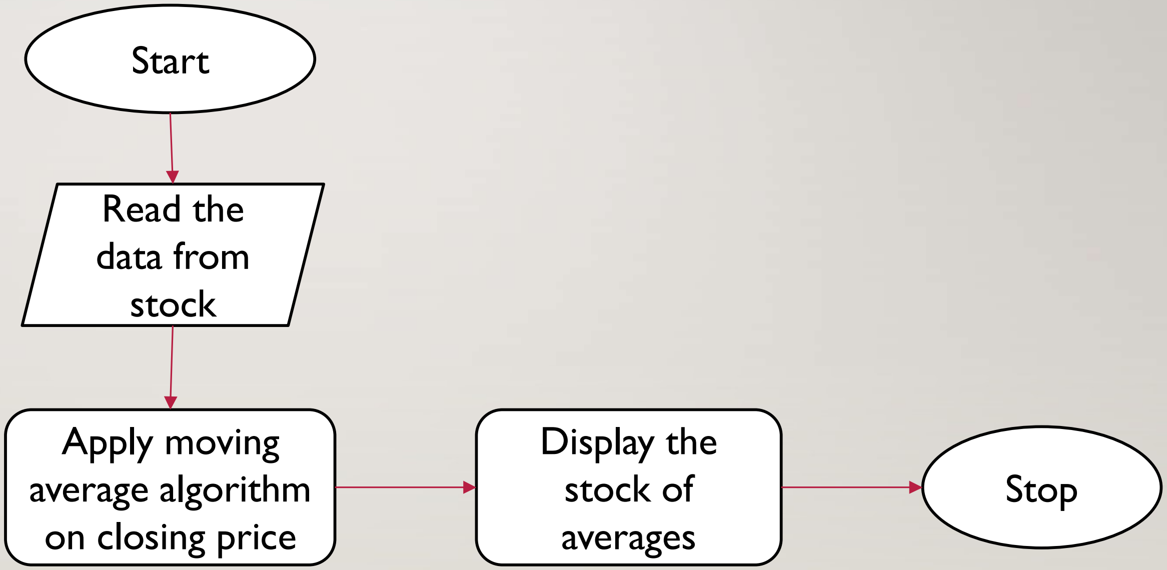
Train the system using training dataset

Test the system using testing dataset

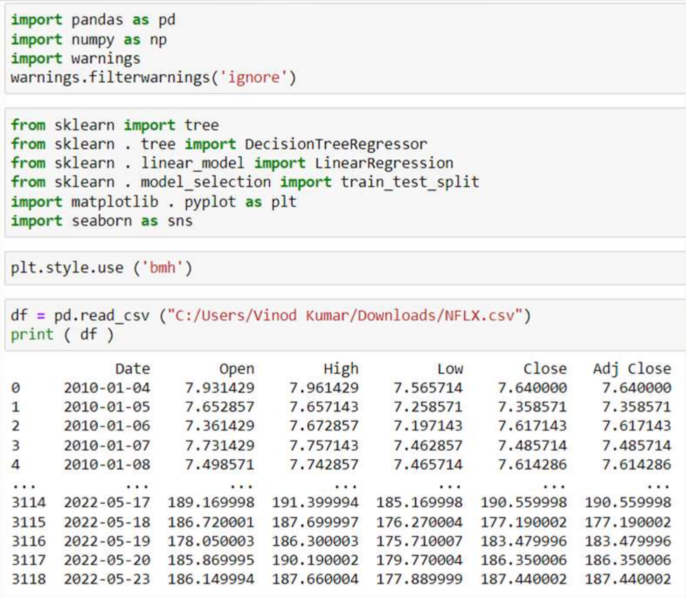
Display the output

Stop

Moving Average Model


Start
Read the data from stock
Apply moving average algorithm on closing price
Display the stock of averages
Stop

CODE



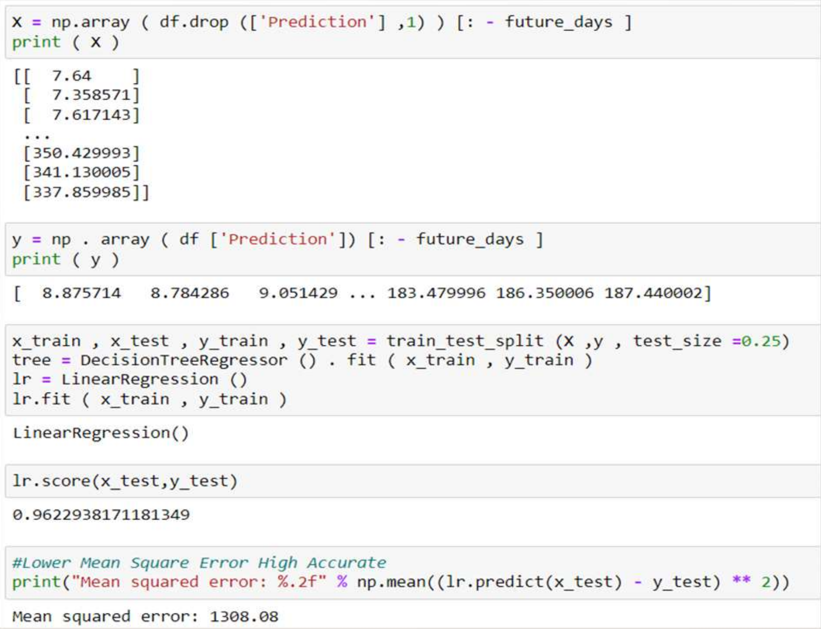
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
from sklearn import tree
from sklearn . tree import DecisionTreeRegressor
from sklearn . linear_model import LinearRegression
from sklearn . model_selection import train_test_split
import matplotlib . pyplot as plt
import seaborn as sns
```

```
plt.style.use ('bmh')
```

```
df = pd.read_csv ("C:/Users/Vinod Kumar/Downloads/NFLX.csv")
print ( df )
```

```
            Date        Open        High         Low       Close   Adj Close
0     2010-01-04    7.931429    7.961429    7.565714    7.640000    7.640000
1     2010-01-05    7.652857    7.657143    7.258571    7.358571    7.358571
2     2010-01-06    7.361429    7.672857    7.197143    7.617143    7.617143
3     2010-01-07    7.731429    7.757143    7.462857    7.485714    7.485714
4     2010-01-08    7.498571    7.742857    7.465714    7.614286    7.614286
...          ...         ...         ...         ...         ...         ...
3114  2022-05-17  189.169998  191.399994  185.169998  190.559998  190.559998
3115  2022-05-18  186.720001  187.699997  176.270004  177.190002  177.190002
3116  2022-05-19  178.050003  186.300003  175.710007  183.479996  183.479996
3117  2022-05-20  185.869995  190.190002  179.770004  186.350006  186.350006
3118  2022-05-23  186.149994  187.660004  177.889999  187.440002  187.440002
```



```
X = np.array ( df.drop (['Prediction'] ,1) ) [: - future_days ]
print ( X )
```

```
[[  7.64     ]
 [  7.358571]
 [  7.617143]
 ...
 [350.429993]
 [341.130005]
 [337.859985]]
```

```
y = np . array ( df ['Prediction']) [: - future_days ]
print ( y )
```

```
[  8.875714   8.784286   9.051429 ... 183.479996 186.350006 187.440002]
```

```
x_train , x_test , y_train , y_test = train_test_split (X ,y , test_size =0.25)
tree = DecisionTreeRegressor () . fit ( x_train , y_train )
lr = LinearRegression ()
lr.fit ( x_train , y_train )
```

```
LinearRegression()
```

```
lr.score(x_test,y_test)
```

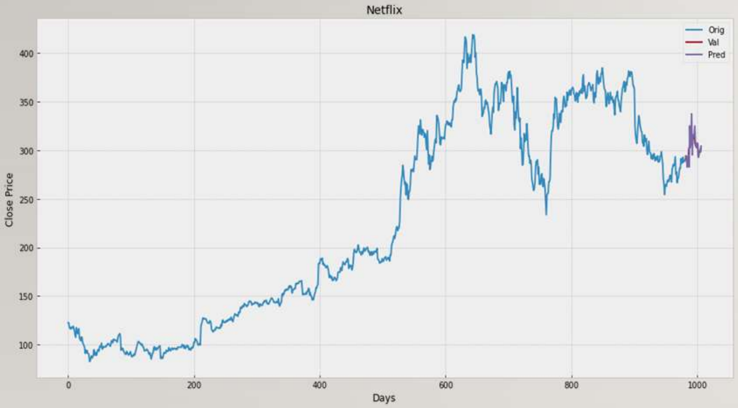
```
0.9622938171181349
```

```
#Lower Mean Square Error High Accurate
print("Mean squared error: %.2f" % np.mean((lr.predict(x_test) - y_test) ** 2))
```

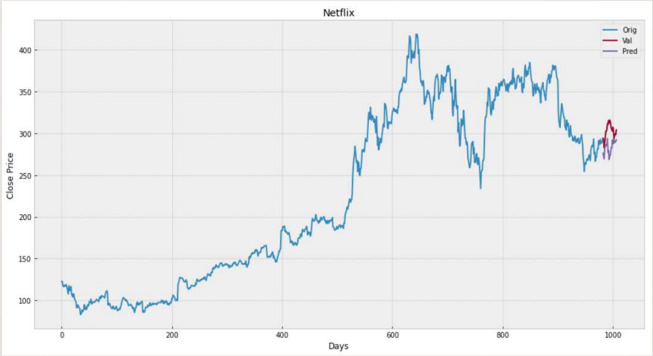
```
Mean squared error: 1308.08
```

Experiment Results

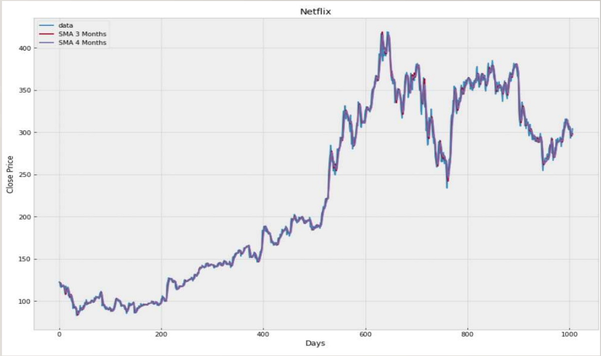
The proposed system is trained and tested over the dataset taken from Netflix company. It is split into training and testing sets respectively and yields the following results upon passing through the different models



Result for Decision Tree



Result for Linear regression



Result for moving average

- ➢ The plot in the above figures is the result of application of Decision tree ,linear regression ,simple moving average algorithms on the dataset to predict varying prices with respect to the time.

Conclusion

- Three techniques have been utilized in this project Decision Tree ,Linear Regression , Moving Average, on the Netflix dataset. These all techniques have shown an improvement in the accuracy of predictions, thereby yielding positive results.
- Use of recently introduced machine learning techniques in the prediction of stocks have yielded promising results and thereby marked the use of them in profitable exchange schemes.
- In the future, the stock market prediction system can be further improved by utilizing a much bigger dataset than the one being utilized currently. This would help to increase the accuracy of our prediction models.
- It has led to the conclusion that it is possible to predict stock market with more accuracy and efficiency using machine learning techniques.