STOCK MARKET ANALYSIS USING SUPERVISED MACHINE LEARNING

Mini Project Report

Submitted by

Rahul Roy

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY-683577, ERNAKULAM(DIST)
MARCH 2022

DECLARATION

I, **Rahul Roy**, hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of our original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date : 04-03-2022

Place: Angamaly

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report titled **"Stock Market Analysis Using Supervised Machine Learning"** submitted by **Rahul Roy** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by them during the year 2022.

Project Guide **Head of the Department**

Submitted for the viva-voice held on ..... at .....

Examiner1 :

Examiner2 :

ACKNOWLEDGEMENT

Gratitude is a feeling which is more eloquent than words, more silent than silence. To complete this project work I needed the direction, assistance and cooperation of various individuals, which is received in abundance with the grace of God.

I hereby express our deep sense of gratitude to **Dr. Manoge George**, Principal of FISAT and **Dr. C Sheela**, Vice principal of FISAT, for allowing us to utilize all the facilities of the college.

My sincere thanks to **Dr. Deepa Mary Mathew**, Head of the department of Computer Applications FISAT and scrum master and our Internal guide for this project **Ms.Anju L and Dr. Sujesh P Lal** for giving valuable guidance, constructive suggestions and comment during my project work. I also express my boundless gratitude to all the lab faculty members for their guidance.

Finally I wish to express a whole heart-ed thanks to my parents, friends and well-wishers who extended their help in one way or other in preparation of my project. Besides all, I thank GOD for everything.

ABSTRACT

The stock market broadly refers to the collection of exchanges and other venues where the buying, selling, and issuance of shares of publically held companies take place. It is one of the most complicated and sophisticated way to do business. Small ownerships, brokerage corporations, banking sector, all depend on this very body to make revenue and divide risks; a very complicated model. This paper proposes to use machine learning algorithms to predict the future stock price for exchange by using open source libraries and preexisting algorithms to help make this unpredictable format of business a little more predictable. The outcome is completely based on numbers and assumes a lot of axioms that may or may not follow in the real world so as the time of prediction.

Contents

Chapter 1

INTRODUCTION

A stock is a general term used to describe the ownership certificates of any company. A share, on the other hand, refers to the stock certificate of a particular company.Holding a particular company's share makes you a shareholder. Stocks are of two types—common and preferred. The difference is while the holder of the former has voting rights that can be exercised in corporate decisions, the later doesn't.

However, preferred shareholders are legally entitled to receive a certain level of dividend payments before any dividends can be issued to other shareholders. Stock analysis is the evaluation of a particular trading instrument, an investment sector, or the market as a whole. Stock analysts attempt to determine the future activity of an instrument, sector, or market.

Stock analysis is a method for investors and traders to make buying and selling decisions. By studying and evaluating past and current data, investors and traders attempt to gain an edge in the markets by making informed decisions.

There are two basic types of stock analysis: fundamental analysis and technical analysis. Fundamental analysis concentrates on data from sources, including financial records, economic reports, company assets, and market share. Technical analysis focuses on the study of past and present price action to predict the probability of future price movements.

Chapter 2

PROOF OF CONCEPT

2.1 Existing System

Early work considered using regression models to predict the trends and values of stock price for the next day. But using Linear Regression, the short term trends and price have high error rates.Therefore it might not be the ideal implementation. And also the attributes are not actually contributing enough for training the linear model.

2.2 Proposed System

Rather than taking the general stock prices, here we concentrate on the derivative stock market. In Futures and Options, there can be more relation between the stock data in previous days because of short term trading.
Thus here we focus on Sectorial Stock BANKNIFTY of National Stock Exchange (NSE). The Options stock closes weekly, in Thursdays. Here through proper data analysis, we could find a relation between percentage changes in stock price during the expiry date and previous dates.

2.3 Objectives

The aim of this project is to develop an application which help an investor following a certain strategy to invest in the Options stocks of BANKNIFTY. Here the investor has about 75% profit rate if he follows the mentioned strategy. But if the loss in this method can be huge. If there is way that he could minimize the loss, if there is, then it would increases the total yearly profit.
Therefore, here we input the stock data of previous two days and the system will recommend whether there is chance for profit if he continue for the expiry date or there could be loss and to exit that itself.

Chapter 3

SCRUM MEETINGS

On 24-11-2021

Started searching the miniproject topic based on the new technology such as deep learning, IoT, machine learning, classification, prediction etc.

On 29-11-2021

The topic ”Stock Market Analysis Using Supervised Machine Learning” was selected and did the detail study of the topic, the required data set was selected. The data set was searched from the different site such as kaggle, NSE1 History etc.

On 06-12-2021

This day I submitted the synopsis and research paper to guide for the topic approval.

On 15-12-2021

After getting approval from the guide, the algorithm and model for the project were structured. Then the algorithm were chosen. Then quick study was done about which algorithms would be best for the project.

The selected ones were:
- Logistic Regression
- K-nearest neighbours

On 18-12-2021

On this day guide took a detailed class on how to do the project, what IDEs to use, what paper are referred, what steps are follow to do the project and so on

On 06-01-2022

According to the project the required IDE such as Visual Studio Code, Colab were chosen .Even checked whether the system was efficient to train the model. Here colab to code the project,then started to deploying the model using the algorithm.Python language is used to code the project.

On 10-01-2022

After the project first review according to guide's opinion decided to concentrate on particular strategy rather than just a future stock price. There fore selected the BANKNIFTY OPTIONS sectorial stock.

On 13-01-2022

Used different algorithm/data model then choose the maximum accuracy one. The algorithm used are:-
Logistic Regression
K-nearest neighbours

On 19-01-2022

Started to do project coding. Firstly study the data set and download the data set from NSE History Data. The data set contains stock data for BANKNIFTY sec-

torial stock for about 11 years.

On 25-01-2022

Testing the data application

On 28-01-2022

The training done in two different data model then choose the maximum accuracy with regression for predicting the stock price variation. Logistic Regression model is used for prediction.

On 02-02-2022

Created the git repository.

On 07-02-2022

Used streamlit for connection.

Chapter 4

IMPLEMENTATION

This project aims to develop an application for helping a stock investor for a particular stock by predicting the price variation and suggesting to continue or exit the current strategy. The price variation on the exit day is predicted using the the previous two days stock data. A Machine learning algorithm is used here for predicting this price variation. The algorithm used here is Logistic Regression algorithm. Four derived attributes that depends on the price variation for the expiry date of the Options Stock is derived. And the model is trained using these attributes and the step that investor taken (continue or exit).

ALGORITHM

Logistic Regression

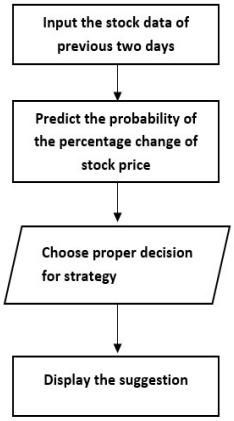
Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

K-nearest neighbours

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

4.1 System Architecture

The use case diagram that describes the operation of the system.



4.2 Data set

The data requirements is very high for the project.We get a data set that contains the component like:-

• Date
• Open (Opening price of Stock)
• High (Highest price possible at an instance of time)
• Low (Lowest price possible at an instance of time)
• Close (Closing price of stock)
• Shares Traded (Total times traded during a day)
• Turnover (Rs. cr)

4.3 Modules

4.3.1 Data Preprocessing

Explore the data set and analyse it. The new derived attributes are required for the proper analysis and prediction of variation of stock price.
The derived attributes are :

```
dif2 = (tue_open - tue_close)/tue_open * 100
dif1 = (tue_open - wed_close)/tue_open * 100
dif0 = (tue_open - thur_close)/tue_open * 100
hl2 = (tue_high - tue_low)/tue_close * 100
hl1 = (wed_high - wed_low)/wed_close * 100
```

4.3.2 Data Splitting

For the proper implementation and testing the processed data is divided into training data (80%) and test data (20%).

4.3.3 Modelling

The training data is used to create the model for the application using Logistic Regression. And the change different parameters and tune to create the best model for the purpose.

4.3.4 Implementation

Model deployment is simply the engineering task of exposing an ML model to real use. The python web framework streamlit is used for the implementation.

Chapter 5

RESULT ANALYSIS

The system predicts the chance for the profit in current strategy that the user invests and suggest the optimal solution to whether to continue or not.
Accuracy is often the most used metric representing the percentage of correctly predicted observations, either true or false. To calculate the accuracy of a model performance, the following equation can be used: In most cases, high accuracy value represents a good model, but considering the fact that we are training a classification model in our case, an article that was predicted as true while it was actually false (false positive) can have negative consequences; similarly, if an article was predicted as false while it contained factual data, this can create trust issues.
Accuarcy = TP+TN/TP+TN+FP+FN

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

To evaluate the conventional algorithm, a data set is built and studied a trend of price variation for the period of limited days. Machine Learning algorithms are applied on the data set to predict the percentage change of stock price. This gives the predicted value for percentage change for expiry date. And then according to that the investor can make decision to minimize the loss. Data was collected from National Stock Exchange History. The model gives about 75% accuracy so that it would valuable for the investor following that particular trading method.

6.2 Future Scope

In the future, if more data could be accessed such as the current availability of seats, the predicted results will be more accurate. And further advanced analysis can find more dependent attributes and this can make the model more accurate.

Chapter 7

SOURCE CODE

7.0.1 model.py

7.0.2 knn.py

7.0.3 app.py

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
import pickle

# Load csv file
df = pd.read_csv('data.csv')

# Data Preprocessing
df['Date'] = pd.to_datetime(df.Date)


df['DoW'] = pd.to_datetime(df['Date']).dt.dayofweek

for i, row in df.iterrows():

  if i >= 2:
    df.loc[i, 'dif2'] = (df.loc[i-2, 'Open'] - df.loc[i-2, 'Close'])/df.loc[i-2, 'Open'] * 100
    df.loc[i, 'dif1'] = (df.loc[i-2, 'Open'] - df.loc[i-1, 'Close'])/df.loc[i-2, 'Open'] * 100
    df.loc[i, 'dif0'] = (df.loc[i-2, 'Open'] - df.loc[i, 'Close'])/df.loc[i-2,'Open'] * 100

    df.loc[i, 'HL_p1'] = (df.loc[i-1, 'High'] - df.loc[i-1, 'Low'])/df.loc[i-1, 'Close'] * 100
    df.loc[i, 'HL_p2'] = (df.loc[i-2, 'High'] - df.loc[i-2, 'Low'])/df.loc[i-2, 'Close'] * 100


df['dif0'] = df['dif0'].abs()
df['dif1'] = df['dif1'].abs()
df['dif2'] = df['dif2'].abs()
```

```
for i, row in df.iterrows():
  if df.loc[i, 'dif0'] > 1.0:
    df.loc[i,'Outcome'] = 0
  else:
    df.loc[i,'Outcome'] = 1

df.Outcome = df.Outcome.astype(int)

# checking for null values
# df.isnull().sum()
df.dropna(inplace=True)

# create dataframe named 'data' where df['DoW'] == 3, Thursday
data = df.loc[df['DoW'] == 3].copy()
data = data[['Date', 'dif2', 'dif1', 'HL_p2', 'HL_p1',  'Outcome']]

# select independent and dependent variables
X = data[['dif2', 'dif1', 'HL_p1', 'HL_p2']]
y = data[['Outcome']]

model_score = 0
for i in range(100):

    # split dataset into training and testing data
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)

    # feature scaling
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.fit_transform(X_test)
```

```
    # Model creation
    model = LogisticRegression()

    model.fit(X_train, y_train.values.ravel())
    if model.score(X_test, y_test) > model_score:

        # Create pickle file for saving model
        pickle.dump(model, open('model.pkl', 'wb'))
        model_score = model.score(X_test, y_test)
        # print(model.score(X_test, y_test))
y_pred = model.predict(X_test)

print(f'Model Score: {model_score}\n')
print('Confusion Matrix: ')
print(confusion_matrix(y_test, y_pred), '\n')
print('Classification Report: ')
print(classification_report(y_test, y_pred))
```

```
import pandas as pd
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
import pickle

# Load csv file
df = pd.read_csv('data.csv')

# Data Preprocessing
df['Date'] = pd.to_datetime(df.Date)


df['DoW'] = pd.to_datetime(df['Date']).dt.dayofweek

for i, row in df.iterrows():

    if i >= 2:
        df.loc[i, 'dif2'] = (df.loc[i-2, 'Open'] - df.loc[i-2, 'Close'])/df.loc[i-2, 'Open'] * 100
        df.loc[i, 'dif1'] = (df.loc[i-2, 'Open'] - df.loc[i-1, 'Close'])/df.loc[i-2, 'Open'] * 100
        df.loc[i, 'dif0'] = (df.loc[i-2, 'Open'] - df.loc[i, 'Close'])/df.loc[i-2,'Open'] * 100

        df.loc[i, 'HL_p1'] = (df.loc[i-1, 'High'] - df.loc[i-1, 'Low'])/df.loc[i-1, 'Close'] * 100
        df.loc[i, 'HL_p2'] = (df.loc[i-2, 'High'] - df.loc[i-2, 'Low'])/df.loc[i-2, 'Close'] * 100


df['dif0'] = df['dif0'].abs()
df['dif1'] = df['dif1'].abs()
df['dif2'] = df['dif2'].abs()
```

```
for i, row in df.iterrows():
  if df.loc[i, 'dif0'] > 1.0:
    df.loc[i,'Outcome'] = 0
  else:
    df.loc[i,'Outcome'] = 1

df.Outcome = df.Outcome.astype(int)

# checking for null values
# df.isnull().sum()
df.dropna(inplace=True)

# create dataframe named 'data' where df['DoW'] == 3, Thursday
data = df.loc[df['DoW'] == 3].copy()
data = data[['Date', 'dif2', 'dif1', 'HL_p2', 'HL_p1',  'Outcome']]

# select independent and dependent variables
X = data[['dif2', 'dif1', 'HL_p1', 'HL_p2']]
y = data[['Outcome']]

model_score = 0
for i in range(100):

    # split dataset into training and testing data
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)

    # feature scaling
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.fit_transform(X_test)
```

```
    # Model creation
    model = KNeighborsClassifier(n_neighbors=5)

    model.fit(X_train, y_train.values.ravel())
    if model.score(X_test, y_test) > model_score:

        # Create pickle file for saving model
        pickle.dump(model, open('model.pkl', 'wb'))
        model_score = model.score(X_test, y_test)
        # print(model.score(X_test, y_test))
y_pred = model.predict(X_test)

print(f'Model Score: {model_score}\n')
print('Confusion Matrix: ')
print(confusion_matrix(y_test, y_pred), '\n')
print('Classification Report: ')
print(classification_report(y_test, y_pred))
```

```
import streamlit as st
import pickle
import numpy as np

model = pickle.load(open('model.pkl', 'rb'))


def predict(input):
    # input = np.array([[dif2, dif1, hl1, hl2]]).astype(np.float)
    prediction = model.predict_proba(input)
    pred = '{0:.{1}f}'.format(prediction[0][0], 2)
    return float(pred)

def result(input):
    outcome = model.predict(input)
    return outcome


def calc(tue_open, tue_close, tue_high, tue_low, wed_close, wed_high, wed_low):
    dif2 = (tue_open - tue_close)/tue_open * 100
    dif1 = (tue_open - wed_close)/tue_open * 100
    hl2 = (tue_high - tue_low)/tue_close * 100
    hl1 = (wed_high - wed_low)/wed_close * 100
    input = np.array([[dif2, dif1, hl1, hl2]]).astype(np.float)
    return input

def main():
    st.title('Stock Market Analysis')
    html_temp = """
    <div style="background-color:#025246 ;padding:10px">
    <h2 style="color:white;text-align:center;">BANK NIFTY OPTIONS </h2>
    </div>
```

```
    """
    st.markdown(html_temp, unsafe_allow_html=True)

    # tue_open = st.number_input()
    tue_open = st.number_input("Tuesday :- Open")
    tue_high = st.number_input("Tuesday :- High")
    tue_low = st.number_input("Tuesday :- Low")
    tue_close = st.number_input("Tuesday :- Close")
    wed_high = st.number_input("Wednesday :- High")
    wed_low = st.number_input("Wednesday :- Low")
    wed_close = st.number_input("Wednesday :- Close")

    # hl2 = st.text_input("HL_2","Type Here")

    profit_html="""
      <div style="background-color:#F4D03F;padding:10px >
       <h2 style="color:white;text-align:center;"> There is a chance for
        profit. Continue with the strategy!</h2>
       </div>
    """
    loss_html="""
      <div style="background-color:#F08080;padding:10px >
       <h2 style="color:black ;text-align:center;"> There is chance of
       loss. It is better to exit now to minimize the loss!</h2>
       </div>
    """

    if st.button("Predict"):
        x_values = calc(tue_open, tue_close, tue_high, tue_low, wed_close, wed_high, wed_low)
        output=predict(x_values)
        outcome = result(x_values)
```

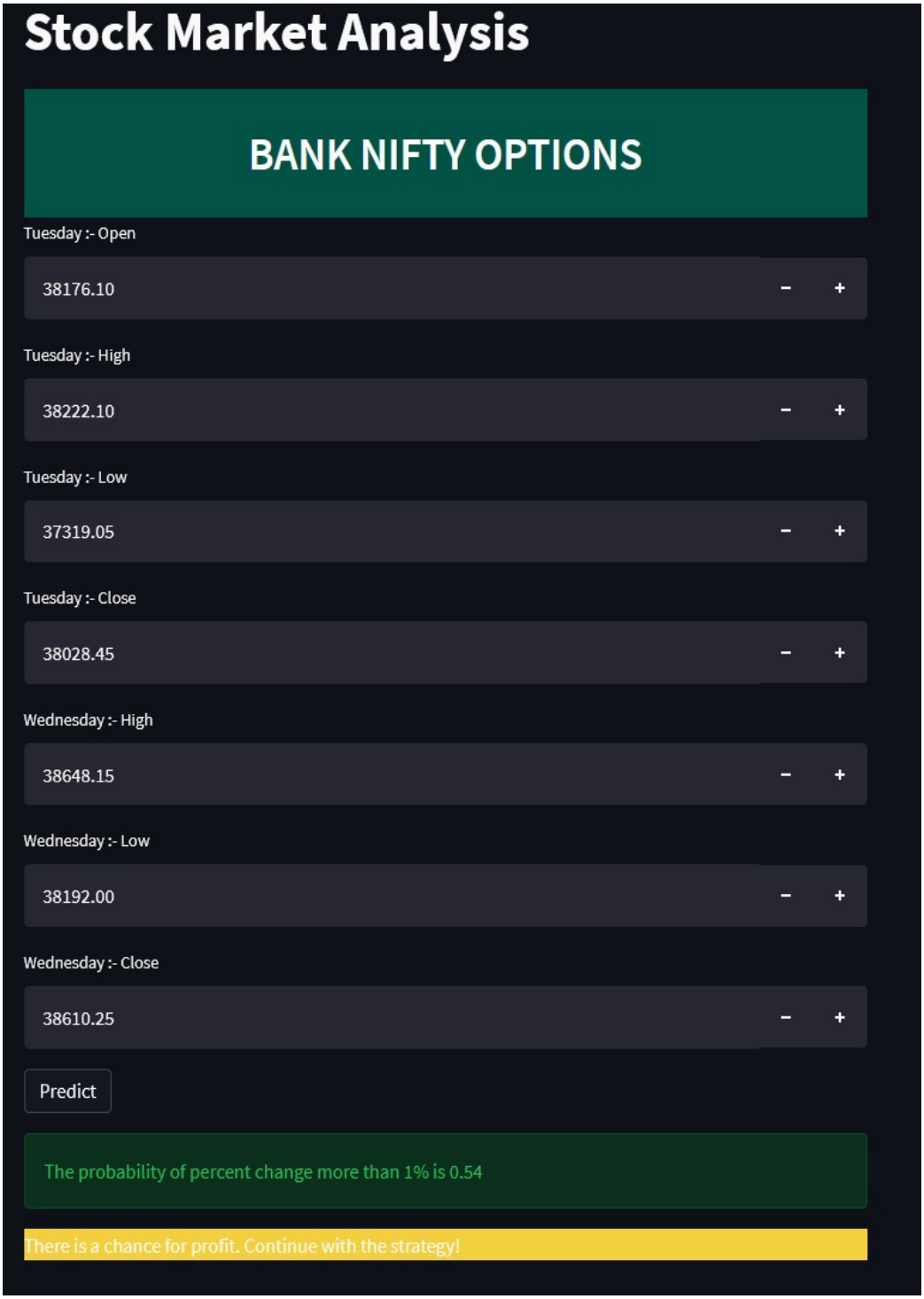
```
    if st.button("Predict"):
        x_values = calc(tue_open, tue_close, tue_high, tue_low, wed_close, wed_high
        output=predict(x_values)
        outcome = result(x_values)
        st.success('The probability of percent change more than 1% is {}'.format(ou

        if outcome == 1:
            st.markdown(loss_html,unsafe_allow_html=True)
        else:
            st.markdown(profit_html,unsafe_allow_html=True)

if __name__=='__main__':
    main()
```

Chapter 8

SCREEN SHOTS


Stock Market Analysis
BANK NIFTY OPTIONS
Tuesday :- Open
38176.10
Tuesday :- High
38222.10
Tuesday :- Low
37319.05
Tuesday :- Close
38028.45
Wednesday :- High
38648.15
Wednesday :- Low
38192.00
Wednesday :- Close
38610.25
Predict
The probability of percent change more than 1% is 0.54
There is a chance for profit. Continue with the strategy!

Chapter 9

REFERENCES

- www.youtube.com
- www.wikipedia.com [1]. Stock Market Analysis using Supervised Machine Learning, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India, 14th -16th Feb 2019

[3] https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh