STOCK PRICE PREDICTION USING DATA SCIENCE TECHNIQUES

A MINOR PROJECT REPORT

Submitted by

KOTI MUNI YOGESWAR REDDY
[RA1911003010876]
MEDIDA SHIVANANDA REDDY
[RA1911003010869]

Under the guidance of

Dr.D.Vinod
(Assistant Professor, CTECH)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M.Nagar, Kattankulathur, Chengalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section3 of UGC Act,1956)

BONAFIDE CERTIFICATE

Certified that 18CSP107L minor project report titled "**STOCK PRICE PREDICTION USING DATA SCIENCE TECHNIQUES**" is the bonafide work of "**KOTI MUNI YOGESWAR REDDY [RA1911003010876], MEDIDA SHIVANANDA REDDY [RA1911003010869]**" who carried out the minor project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

GUIDE
SIGNATURE

Dr. D. Vinod
Assistant Professor

SIGNATURE

Dr. M. Pushpalatha
HEAD OF THE DEPARTMENT
Professor
Department of Computing Technologies

Signature of the Panel Head
Dr. K. Sreekumar

Associate Professor

ABSTRACT

Generally, predicting how the stock market will perform is one of the most difficult things to do. It can be described as one of the most critical process to predict that. This is a very complex task and has uncertainties. To prevent this problem in One of the most interesting (or perhaps most profitable) time series data using machine learning techniques. Hence, stock price prediction has become an important research area. The aim is to predict machine learning based techniques for stock price prediction results in best accuracy. The analysis of dataset by supervised machine learning technique(SMLT) to capture several information's like, variable identification, uni-variate analysis, bi-variate and multi-variate analysis, missing value treatments and analyze the data validation, data cleaning/preparing and data visualization will be done on the entire given dataset. To propose a machine learning-based method to accurately predict the stock price Index value by prediction results in the form of stock price increase or stable state best accuracy from comparing supervise classification machine learning algorithms. Additionally, to compare and discuss the performance of various machine learning algorithms from the given transport traffic department dataset with evaluation. Dataset with evaluation classification report, identify the confusion matrix and to categorizing data from priority and the result shows that the effectiveness of the proposed machine learning algorithm technique can be compared with best accuracy with precision, Recall and F1 Score.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

HTML	Hypertext markup language
CSS	Cascading Style Sheet
CV	Computer Vision
DB	Data Base
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface

CHAPTER1

INTRODUCTION

1.1 PURPOSE

Data science is Associate in Nursing knowledge domain field that uses scientific strategies, processes, algorithms and systems to extract data and insights from structured and unstructured information, and apply data and unjust insights from information across a broad vary of application domains. The term "data science" has been copied back to 1974, when Peter Naur proposed it as another name for applied science. In 1996, the International Federation of Classification Societies became the primary conference to specifically feature information science as a subject. However, the definition was still in flux. The term “data science” was initial coined in 2008 by D.J. Patil, and Jeff Hammer bacher, the pioneer leads of knowledge and analytics efforts at LinkedIn and Facebook. In but a decade, it's become one amongst the most popular and most trending professions within the market. Data science is that the field of study that combines domain experience, programming skills, and data of arithmetic and statistics to extract meaningful insights from information. Data science may be outlined as a blend of arithmetic, business acumen, tools, algorithms and machine learning techniques, all of that facilitate USA in finding out the hidden insights or patterns from data which may be of major use within the formation of massive business choices. Data Scientist: Data scientists examine that queries would like responsive and wherever to find the connected information.

1.2 OBJECTIVE

The goal is to develop a machine learning model for Stock price Prediction. Supervised machine learning classification models used for predicting results inthe form of best accuracy by comparing supervised algorithm.

1.3 SOFTWARE REQUIREMENTS SPECIFICATION

1. Software Requirements:

Operating System : Windows

Tool : Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor : Pentium IV/III

Hard disk : minimum 80 GB

RAM : minimum 2 GB

CHAPTER 2

2 LITERATURE SURVEY

Due to the massive volume of customers in market.A literature review may be a body of text that aims to review the crucial points of current information on and/or method approaches to a specific topic. It is secondary sources and discuss revealed data during a specific subject area and generally data during a specific space discipline subject field of study bailiwick branch of knowledge domain knowledge base} among an explicit time period. Its final goal is to bring the reader up thus far with current literature on a subject and forms theidea for an additional goal, like future analysis that will be required within the space and precedes a groundwork proposal and should be simply a straightforward summary of sources. Usually, it's associate structure pattern and combines both outline and synthesis. A outline may be a recap of necessary data concerning the supply, but a synthesis may be a re- organization, shuffling of knowledge.it'd provides a new interpretation of recent material or mix new with recent interpretations or it might trace the intellectual progression of the sector, together with major debates. Depending on the case, the literature review could measure the sources and advise the reader on the foremost pertinent or relevant of them

[1]DEEP LEARNING BASED APPROACH FOR FRESH PRODUCEMARKET PRICE PREDICTION

Author: Lobna Nassar, Ifeanyi Emmanuel Ok wuchi, Muhammad Saad

DESCRIPTION

The ARIMA model has the very best mean absolute proportion error (MAPE) thence very cheap performance compared to the traditional millilitre models. additionally, among the traditional the Gradient Boosting (GB) is that the best thanks to having the smallest amount MAPE error. Finally, the performance of LSTM simple deciliter model is on top of all of the tested typical millilitre models for two FP (Watermelon and Bok Choy). this is often thanks to having less markets for these 2 FP that leaves North American country with knowledge that's nearer in nature to statistic. It is also found that the simplest activity model in keeping with the aggregative measure is that the compound deciliter model, ATTCNN-LSTM, that outperforms the ML and easy deciliter models in accuracy of worth prediction issue faced by enterprises. Products fraud detection will enable companies to better understand the market and contribute to increasing the company's revenue. Therefore, this paper's research topic mainly focuses on the problem of DataCo supply chain fraud, which is provided by Kaggle competition. Therefore, the objective of this article is to utilize the data of the company DataCo smart supply chain for the analysis of product fraud. And we propose a hybrid model, which output is the Arithmetic mean of the output of algorithm XGBoost [1][2] and algorithm Random Forests. Compared with other models, this model has better advantages in performance. In the whole process, Python is the main tool for data processing, modeling, and analysis.

[2]A DEEP HYBRID FUZZY NEURAL HAMMERSTEIN-WIENERNETWORK FOR STOCK PRICE PREDICTION

Author: Xie Chen, Deepu Rajan, Chai Quek

DESCRIPTION

A deep hybrid fuzzy neural Hammerstein-Wiener model (FNHW), is planned during this paper. The implication and reasoning of a neuro-fuzzy is based onthe fuzzy rule base that has been shaped throughout coaching. It needs the training information to be ready to adequately represent entire system behaviors. However, the take a look at information could vary with distribution shift in statistic domain. Furthers, the coaching information could also be derived from steady state whereas the take a look at information which is within the sort of dynamically dynamic described by forceful information shift under bound state of affairs like money crisis. the soundness of rule base inference from neuro-fuzzy system on the steady-state information is achieved in addition as inheritable the nice approximation accuracy and wonderful straight line tracking benefits of Hamerstein-Wiener model on the dynamically dynamic data. The effectiveness of planned model is evaluated on 2 money stock price prediction datasets. A deep hybrid fuzzy neural Hammerstein-Wiener network for stock value prediction by combining the advantages of neural fuzzy system and Hammerstein-Wiener model to handle steady-state and constantly changing information correspondingly.we tend to performed the experiments on 2 completely different money stock value prediction datasets and showed that the prediction performance has been considerably improved victimisation our model when put next to alternative state-of-art neuro-fuzzy systems.

[3] SENTIMENT ANALYSIS FOR STOCK PRICE PREDICTION

Author: Rubi Gupta, Min Chen

DESCRIPTION

Analysis on Stock Twits information and to know the impact of sentiments on stock value movements. They commit to additional improve the add the following areas. First, during this work, we have a tendency to use 2 forms of sentiments: optimistic (positive) and pessimistic (negative). Adding neutral sentiment would possibly cut back noise and doubtless enhance accuracy of the work. Second, our analysis is proscribed to five corporations. associate degree growth to broader set of corporations or all StockTwits data would possibly yield a lot of insights into {the information|the info|the information}, resulting in simpler application available value prediction. they're use the ex gratia sentiment labels provided by StockTwits users because the ground truth information for model coaching. sentiment data is employed additionally to the past stock statistic information to improve the accuracy of stock value movement prediction. The effectiveness of the projected work on stock value prediction is incontestable through experiments on 5 corporations. StockTwits may be a comparatively new microblogging website, that is changing into progressively common for users to share their discussions and sentiments concerning stocks and monetary markets. provided a reasonable proof that sentiments information contains a positive impact on the accuracy of stock value amendment prediction.

[3] PRINCIPAL COMPONENT ANALYSIS TO DETERMINE MAINFACTORS STOCK PRICE OF CONSUMER GOODS INDUSTRY Author: Rahma Firsty Fitriyana, Brady Rikumahu, Andry Alamsyah

DESCRIPTION

Stock value is thar the necessary think about achieving the profit available investment, and also the prediction is typically done by relating the value of a stock to factors that influence it the matter is, there are an outsized variety of variables that can be wont to predict the stock costs its tough for a possible investor to decide on that variables ought to be employed in predicting the stock costs. This analysis used the Principal part Analysis because the dimension reduction technique to create major parts that influence the stock costs without losing the knowledge} and uses data from 5 firms. Analysis method are often wont to realize the most determinants of stock costs by adding new variables to induce a lot of correct results like political economy factors and adding alternative money ratios, as a result of there are several variables moving stock prices, together with political economy factors that didn't enclosed during this analysis. Next analysis might embody those factors to visualize their impact on stock.

CHAPTER 3

3.1 SYSTEM ARCHITECTURE

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

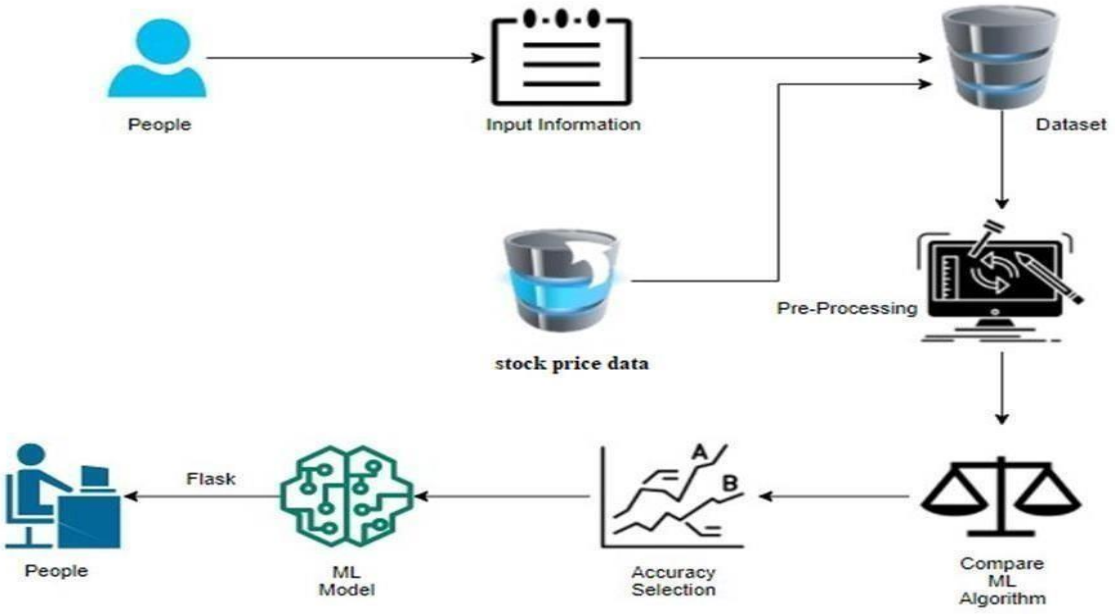


Fig 2

3.2 DATABASE DESIGN

ER DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

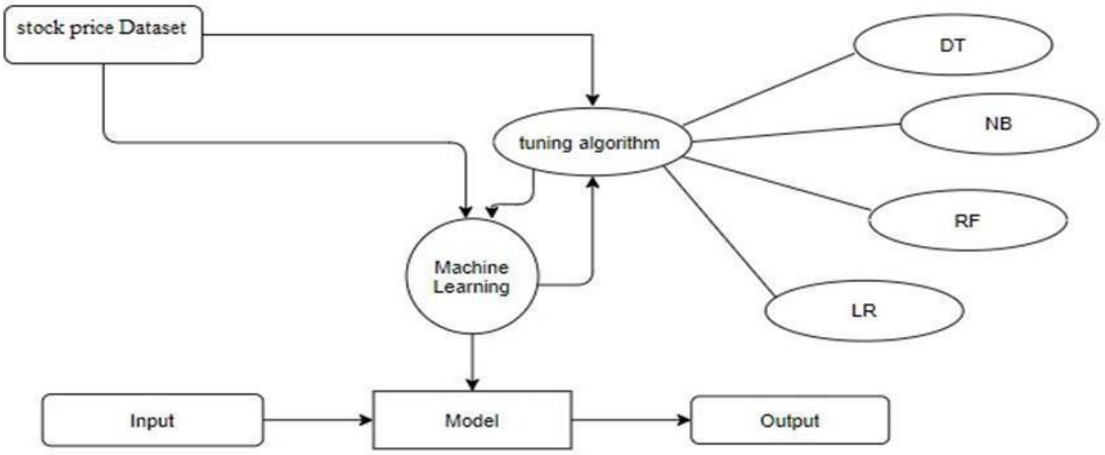


Fig 3

3.3 CLASS DIAGRAM

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

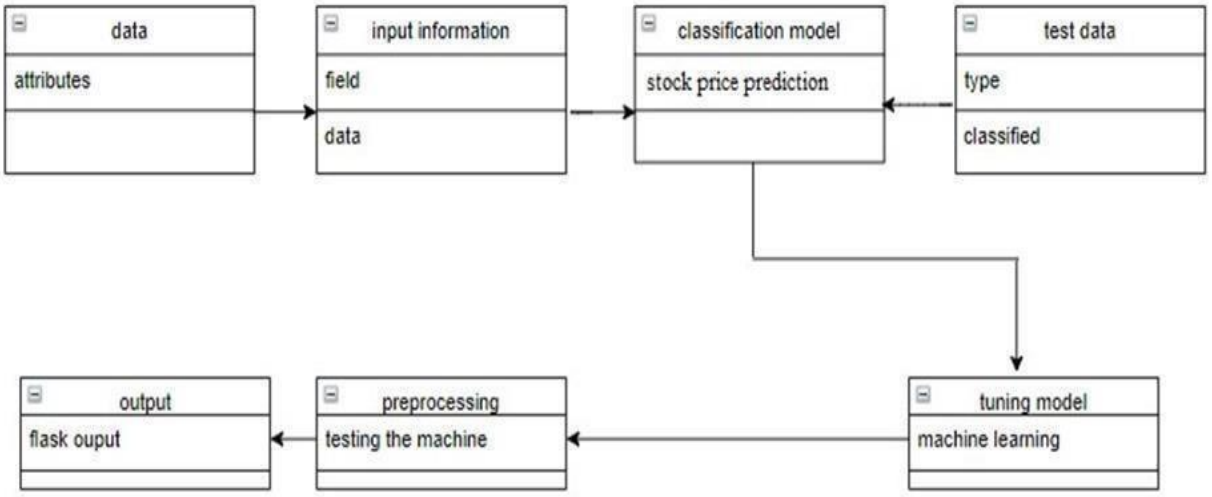


Fig 4

3.4 ACTIVITY DIAGRAM

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel,branched, concurrent and single.

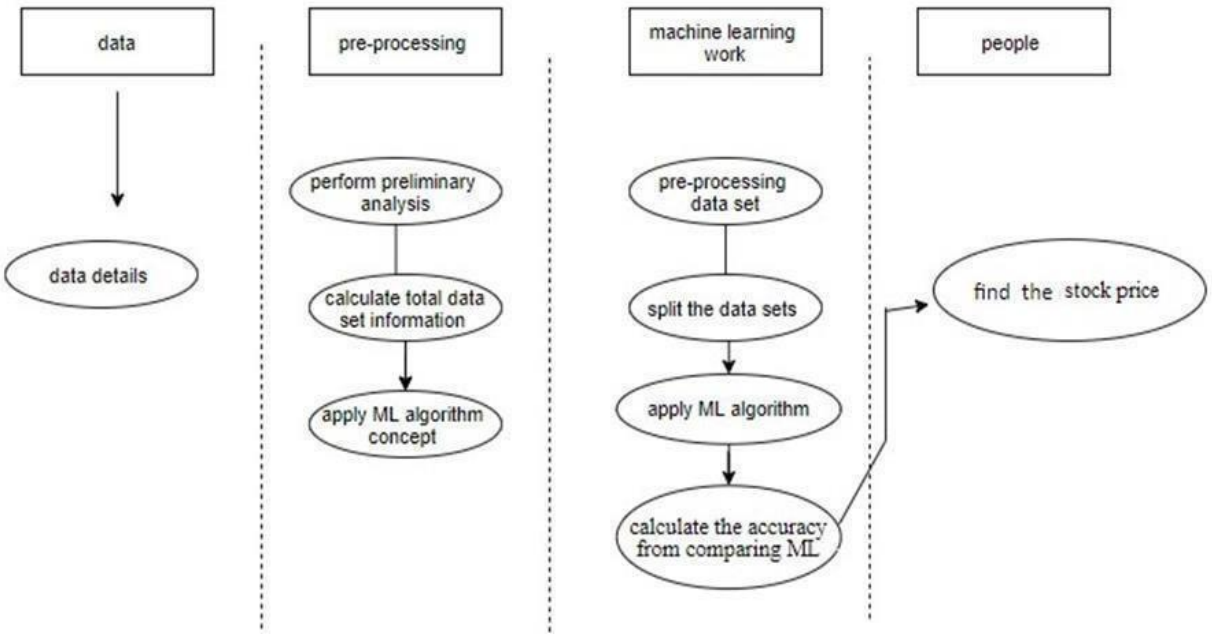


Fig 5

3.5 SEQUENCE DIAGRAM

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Other dynamic modelling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design-level models for modern business application development.

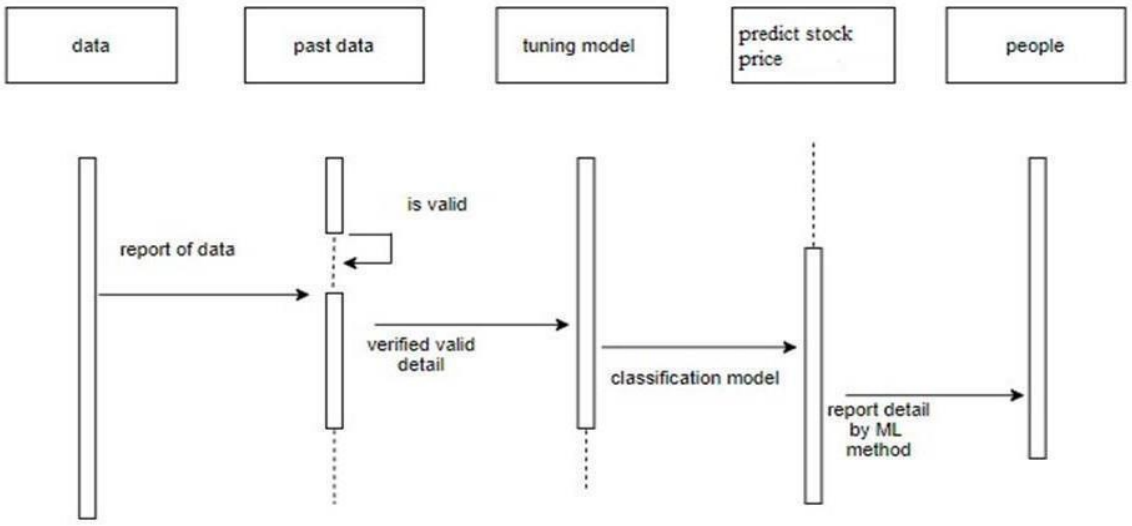


Fig 6

3.6 List of modules

List of Modules:

- Data Pre-processing
- Data Analysis of Visualization
- Comparing Algorithm with prediction in the form of best accuracy result
- Deployment Using Flask

CHAPTER 4

METHODOLOGY

Algorithm Explanation

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train_test_split, DecisionTreeClassifier or Logistic Regression and accuracy_score.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

Logistic Regression

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts P(Y=1) as a function of X. Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.

- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

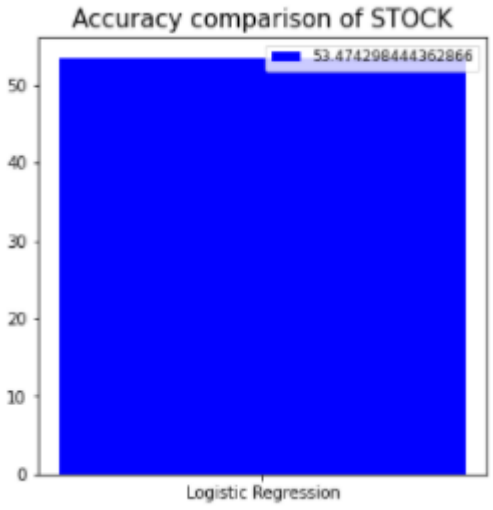
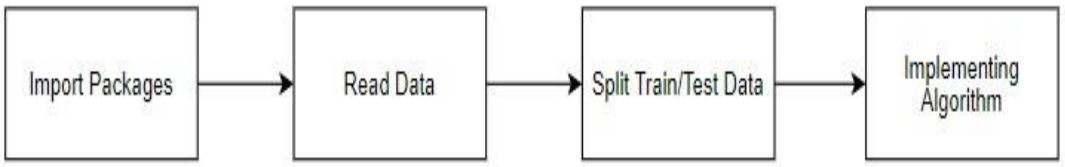


Fig 7

```
Accuracy result of Logistic Regression is: 54

Classification report of Logistic Regression Results:

              precision    recall  f1-score   support

           0       0.00      0.00      0.00       277
           1       0.54      1.00      0.70       319

    accuracy                           0.54       596
   macro avg       0.27      0.50      0.35       596
weighted avg       0.29      0.54      0.37       596


Confusion Matrix result of Logistic Regression is:
 [[  0 277]
 [  0 319]]

Sensitivity :  0.0

Specificity :  1.0

Cross validation test results of accuracy:
[0.53517588 0.53652393 0.53400504 0.53400504 0.53400504]

Accuracy result of Logistic Regression is: 53.474298444362866
Cross validation of Logistic Regression is: 0.535234899328859
```

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of

decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees,* resulting in a *forest of trees,* hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.
The following are the basic steps involved in performi ng the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

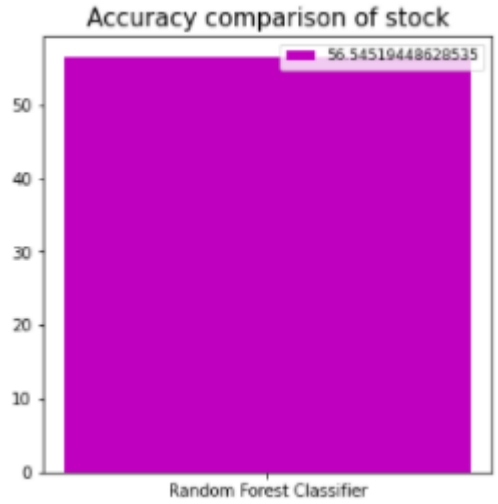
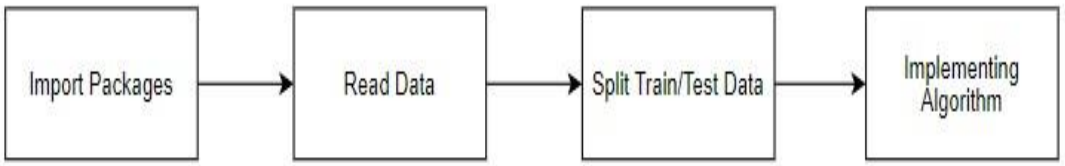


Fig 8

```
Accuracy result of Random Forest Classifier is: 67

Classification report of Random Forest Results:

              precision    recall  f1-score   support

           0       0.66      0.61      0.63       277
           1       0.68      0.73      0.70       319

    accuracy                           0.67       596
   macro avg       0.67      0.67      0.67       596
weighted avg       0.67      0.67      0.67       596


Confusion Matrix result of Random Forest Classifier is:
 [[168 109]
 [ 87 232]]

Sensitivity :  0.6064981949458483

Specificity :  0.7272727272727273

Cross validation test results of accuracy:
[0.57788945 0.52896725 0.5768262  0.59949622 0.5440806 ]

Cross validation of Random Forest Classifier is: 56.54519448628535
```

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

Decision Tree Classifier

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

- At the beginning, we consider the whole training set as the root.
- Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

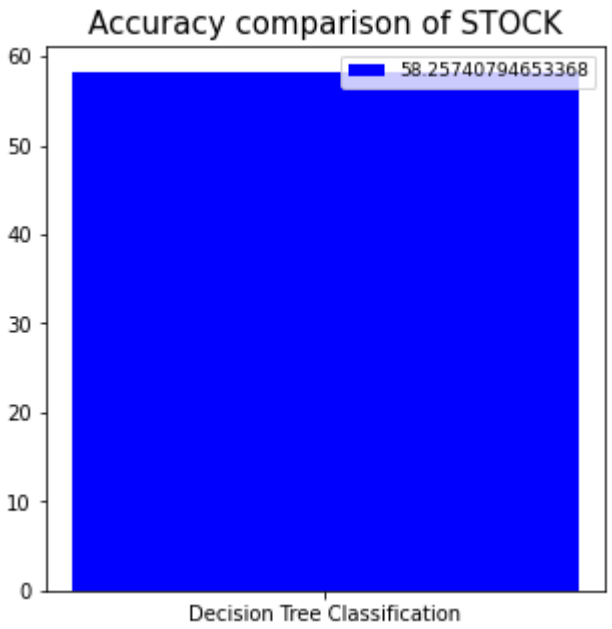
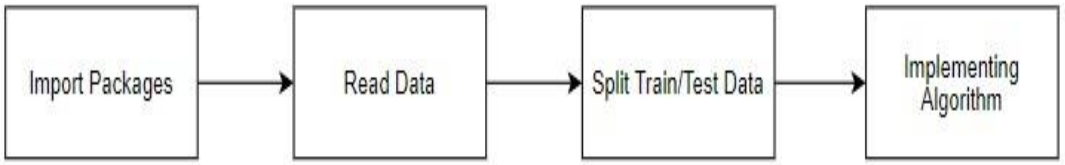


Fig 9

```
Accuracy of  Decision Tree Classifier 63

Classification report of Decision Tree Results:

              precision    recall  f1-score   support

           0       0.60      0.63      0.61       277
           1       0.66      0.64      0.65       319

    accuracy                           0.63       596
   macro avg       0.63      0.63      0.63       596
weighted avg       0.63      0.63      0.63       596


Confusion Matrix result of Decision Tree Classifier is:
 [[174 103]
 [116 203]]

Sensitivity :  0.628158844765343

Specificity :  0.6363636363636364

Cross validation test results of accuracy:
[0.59045226 0.59697733 0.55667506 0.63224181 0.53652393]

Cross validation of Decision Tree Classifier is: 58.25740794653368
```

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

Naive Bayes algorithm:

- The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically.
- Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.
- The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.
- Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms.Naive Bayes classifier is the

fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

- Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location.
- Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence.

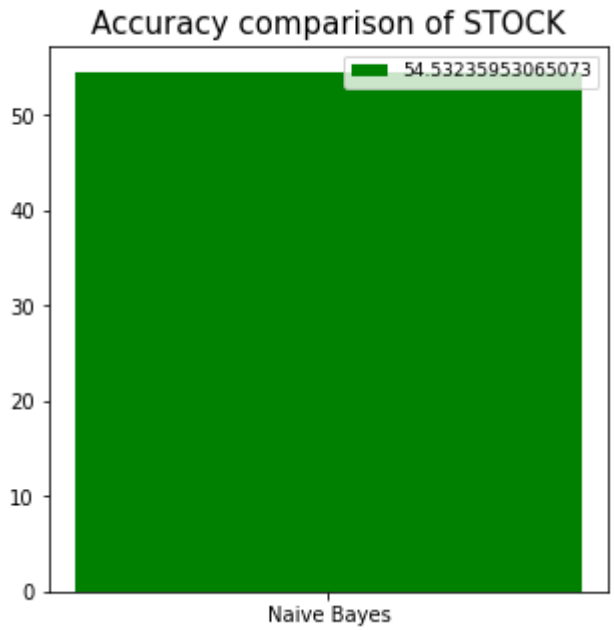
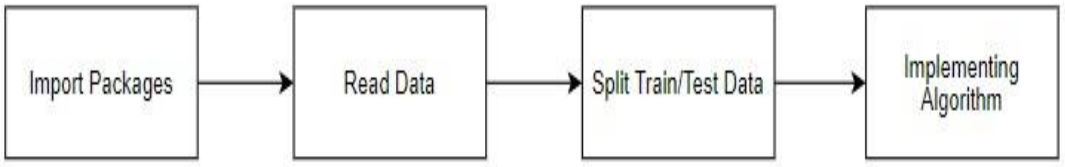


Fig 10

```
Accuracy result of Naive Bayes Algorithm is 56

Classification report of Naive Bayes Results:

              precision    recall  f1-score   support

           0       0.58      0.15      0.24       277
           1       0.55      0.91      0.69       319

    accuracy                           0.56       596
   macro avg       0.57      0.53      0.46       596
weighted avg       0.57      0.56      0.48       596


Confusion Matrix result of Naive Bayes is:
 [[ 42 235]
 [ 30 289]]

Sensitivity :  0.15162454873646208

Specificity :  0.9059561128526645

Cross validation test results of accuracy:
[0.53266332 0.53904282 0.5465995  0.55415617 0.55415617]

Cross validation of Naive Bayes Algorithm is: 54.53235953065073
```

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

CHAPTER 5

CODE AND TESTING

Module – 1

5.1 Pre-Processing

```
#import library packages
```

```
import pandas as pd
```

```
import numpy as np
```

```
import warnings
```

```
warnings.filterwarnings("ignore")
```

```
#Load given dataset
```

```
data = pd.read_csv("heart.csv")
```

Before drop the given dataset:

```
data.head()
```

```
#shape
```

```
data.shape
```

After drop the given dataset:

```
df = data.dropna()
```

```
df.head()
```

```
#shape
```

```
df.shape
```

```
#columns
```

```
df.columns
```

```
#To describe the dataframe
```

```
df.describe()

#Checking datatype and information about dataset

df.info()
```

Checking duplicate values of dataframe

```
#Checking for duplicate data

df.duplicated()

sum(df.duplicated())

#Checking sum of missing values

df.isnull().sum()

df.age.unique()

df.sex.unique()

df.cp.unique()

df.trestbps.unique()

df.chol.unique()

df.restecg.unique()

df.target.unique()

print("Age of patient ranges :", sorted(df['age'].unique()))

df["target"].value_counts()

df.corr()
```

Module – 2

5.2 Visualization

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings
warnings.filterwarnings('ignore')
data = pd.read_csv("heart.csv")
df = data.dropna()
df.columns
pd.crosstab(df.age, df.target)
pd.crosstab(df.cp, df.target)
#Histogram Plot of Age distribution
df["age"].hist(figsize=(10,8), color="red")
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("No of Patients")
plt.show()
#barplot for ecg and target
fig, ax = plt.subplots(figsize=(15,8))
sns.barplot(x = "restecg", y = "target", ax=ax, data=df)
plt.title("ECG vs TARGET")
#Propagation by variable
```

```
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' \n', fontsize = 15)
    return np.round(dataframe_pie/df.shape[0]*100,2)
PropByVar(df, 'thal')
fig, ax = plt.subplots(figsize=(15,6))
sns.boxplot(df.age, ax =ax)
plt.title("Age distribution")
plt.show()
sns.pairplot(df)
plt.show()
fig, ax = plt.subplots(figsize=(15,6))
sns.violinplot(y = df['age'], x = df['target'], ax=ax)
plt.title("Patients age and target")
plt.show()
# Heatmap plot diagram
fig, ax = plt.subplots(figsize=(15,10))
sns.heatmap(df.corr(), ax=ax, annot=True)
```

Spliting Train/Test:

```
#preprocessing, split test and dataset, split response variable
X = df.drop(labels='target', axis=1)
#Response variable
```

```
y = df.loc[:,'target']
#We'll use a test size of 30%. We also stratify the split on the response
variable, which is very important to do because there are so few fraudulent
transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)
print("Number of training dataset: ", len(X_train))
print("Number of test dataset: ", len(X_test))
print("Total number of dataset: ", len(X_train)+len(X_test))
def qul_No_qul_bar_plot(df, bygroup):
    dataframe_by_Group = pd.crosstab(df[bygroup], columns=df["target"], normalize
= 'index')
    dataframe_by_Group = np.round((dataframe_by_Group * 100), decimals=2)
    ax = dataframe_by_Group.plot.bar(figsize=(15,7));
    vals = ax.get_yticks()
    ax.set_yticklabels(['{:3.0f}%'.format(x) for x in vals]);
    ax.set_xticklabels(dataframe_by_Group.index,rotation = 0, fontsize = 15);
    ax.set_title('Stroke having or not by given attributes (%) (by ' +
dataframe_by_Group.index.name + ')\n', fontsize = 15)
    ax.set_xlabel(dataframe_by_Group.index.name, fontsize = 12)
    ax.set_ylabel('(%)', fontsize = 12)
    ax.legend(loc = 'upper left',bbox_to_anchor=(1.0,1.0), fontsize= 12)
```

```
    rects = ax.patches


    # Add Data Labels


    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x() + rect.get_width()/2,
                height + 2,
                str(height)+'%',
                ha='center',
                va='bottom',
                fontsize = 12)
    return dataframe_by_Group
qul_No_qul_bar_plot(df, 'ca')
```

Module – 3

5.3 Logistic Regression Algorithm

```
#import library packages

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
import warnings
warnings.filterwarnings('ignore')
#Load given dataset
data = pd.read_csv("heart.csv")
df=data.dropna()
df.columns
#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
matthews_corrcoef, cohen_kappa_score, accuracy_score, average_precision_score,
roc_auc_score
X = df.drop(labels='target', axis=1)
#Response variable
y = df.loc[:,'target']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)
```

Logistic Regression :

```
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
```

```
logR= LogisticRegression()

logR.fit(X_train, y_train)

predictR = logR.predict(X_test)

print("")
print('Classification report of Logistic Regression Results:')
print("")
print(classification_report(y_test, predictR))

print("")
cm=confusion_matrix(y_test, predictR)
print('Confusion Matrix result of Logistic Regression is:\n', cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")
```

```
accuracy = cross_val_score(logR, X, y, scoring='accuracy')
print('Cross validation test results of accuracy:')
print(accuracy)
#get the mean of each fold
print("")
print("Accuracy result of Logistic Regression is:", accuracy.mean() * 100)
LR=accuracy.mean() * 100
def graph():
    import matplotlib.pyplot as plt
    data=[LR]
    alg="Logistic Regression"
    plt.figure(figsize=(5,5))
    b=plt.bar(alg, data, color=("b"))
    plt.title("Accuracy comparison of Stock price", fontsize=15)
    plt.legend(b, data, fontsize=9)
graph()
TP = cm[0][0]
FP = cm[1][0]
FN = cm[1][1]
TN = cm[0][1]
print("True Positive :", TP)
```

```
print("True Negative :", TN)
print("False Positive :", FP)
print("False Negative :", FN)
print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN)
FNR = FN/(TP+FN)
print("True Positive Rate :", TPR)
print("True Negative Rate :", TNR)
print("False Positive Rate :", FPR)
print("False Negative Rate :", FNR)
print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :", PPV)
print("Negative predictive value :", NPV)
def plot_confusion_matrix(cm2, title='Confusion matrix-LogisticRegression',
cmap=plt.cm.Blues):
    target_names=['Predict','Actual']
    plt.imshow(cm2, interpolation='nearest', cmap=cmap)
    plt.title(title)
```

```
    plt.colorbar()
    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')


cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-LogisticRegression:')
print(cm2)
plot_confusion_matrix(cm2)
import joblib
joblib.dump(logR,"model.pkl")
 ['model.pkl']
```

Module – 4

5.4 Random Forest Algorithm

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import numpy as np
import warnings
warnings.filterwarnings('ignore')
#Load given dataset
data = pd.read_csv("heart.csv")
df=data.dropna()
df.columns
#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
matthews_corrcoef, cohen_kappa_score, accuracy_score, average_precision_score,
roc_auc_score
X = df.drop(labels='target', axis=1)
#Response variable
y = df.loc[:,'target']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)
RandomForestClassifier:
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.model_selection import cross_val_score

rfc = RandomForestClassifier()

rfc.fit(X_train, y_train)

predictR = rfc.predict(X_test)

print("")
print('Classification report of Random Forest Classifier Results:')
print("")
print(classification_report(y_test, predictR))

print("")
cm=confusion_matrix(y_test, predictR)
print('Confusion Matrix result of Random Forest Classifier is:\n', cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
```

```
print("")


accuracy = cross_val_score(rfc, X, y, scoring='accuracy')
print('Cross validation test results of accuracy:')
print(accuracy)
#get the mean of each fold
print("")
print("Accuracy result of Random Forest Classifier is:",accuracy.mean() * 100)
LR=accuracy.mean() * 100
def graph():
    import matplotlib.pyplot as plt
    data=[LR]
    alg="Random Fores tClassifier"
    plt.figure(figsize=(5,5))
    b=plt.bar(alg,data,color=("b"))
    plt.title("Accuracy comparison of Stock price",fontsize=15)
    plt.legend(b,data,fontsize=9)
graph()
TP = cm[0][0]
FP = cm[1][0]
FN = cm[1][1]
TN = cm[0][1]
```

```
print("True Positive :", TP)
print("True Negative :", TN)
print("False Positive :", FP)
print("False Negative :", FN)
print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN)
FNR = FN/(TP+FN)
print("True Positive Rate :", TPR)
print("True Negative Rate :", TNR)
print("False Positive Rate :", FPR)
print("False Negative Rate :", FNR)
print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :", PPV)
print("Negative predictive value :", NPV)
def plot_confusion_matrix(cm2, title='Confusion matrix-RandomForestClassifier',
cmap=plt.cm.Blues):
    target_names=['Predict','Actual']
    plt.imshow(cm2, interpolation='nearest', cmap=cmap)
```

```
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')


cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-RandomForestClassifier:')
print(cm2)
plot_confusion_matrix(cm2)
```

Module – 5

5.5 Decision Tree Algorithm

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
import warnings
warnings.filterwarnings('ignore')
#Load given dataset
data = pd.read_csv("heart.csv")
df=data.dropna()
df.columns
#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
matthews_corrcoef, cohen_kappa_score, accuracy_score, average_precision_score,
roc_auc_score
X = df.drop(labels='target', axis=1)
#Response variable
y = df.loc[:,'target']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)
```

Decision Tree Classifier:

```
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
```

```
dtc = DecisionTreeClassifier()

dtc.fit(X_train, y_train)

predictR = dtc.predict(X_test)

print("")
print('Classification report of Decision Tree Classifier Results:')
print("")
print(classification_report(y_test, predictR))

print("")
cm=confusion_matrix(y_test, predictR)
print('Confusion Matrix result of Decision Tree Classifier is:\n', cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")
```

```
accuracy = cross_val_score(dtc, X, y, scoring='accuracy')
print('Cross validation test results of accuracy:')
print(accuracy)
#get the mean of each fold
print("")
print("Accuracy result of Decision Tree Classifier is:", accuracy.mean() * 100)
LR=accuracy.mean() * 100
def graph():
    import matplotlib.pyplot as plt
    data=[LR]
    alg="Decision Tree Classifier "
    plt.figure(figsize=(5,5))
    b=plt.bar(alg, data, color=("b"))
    plt.title("Accuracy comparison of Stock price", fontsize=15)
    plt.legend(b, data, fontsize=9)
graph()
TP = cm[0][0]
FP = cm[1][0]
FN = cm[1][1]
TN = cm[0][1]
print("True Positive :", TP)
```

```
print("True Negative :",TN)
print("False Positive :",FP)
print("False Negative :",FN)
print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN)
FNR = FN/(TP+FN)
print("True Positive Rate :",TPR)
print("True Negative Rate :",TNR)
print("False Positive Rate :",FPR)
print("False Negative Rate :",FNR)
print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)
def plot_confusion_matrix(cm2, title='Confusion matrix-DecisionTreeClassifier',
cmap=plt.cm.Blues):
    target_names=['Predict','Actual']
    plt.imshow(cm2, interpolation='nearest', cmap=cmap)
    plt.title(title)
```

```
    plt.colorbar()
    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')


cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-DecisionTreeClassifier:')
print(cm2)
plot_confusion_matrix(cm2)
```

Module – 6

5.6 Naïve Bayes Algorithm

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
#Load given dataset
data = pd.read_csv("heart.csv")
df = data.dropna()
df.columns
#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
matthews_corrcoef, cohen_kappa_score, accuracy_score, average_precision_score,
roc_auc_score
X = df.drop(labels='target', axis=1)
#Response variable
y = df.loc[:,'target']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)
```

Naive Bayes:

```
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import cross_val_score

nb = GaussianNB()
```

```
nb.fit(X_train, y_train)


predictR = nb.predict(X_test)


print("")
print('Classification report of Naive Bayes Results:')
print("")
print(classification_report(y_test, predictR))


print("")
cm=confusion_matrix(y_test, predictR)
print('Confusion Matrix result of Naive Bayes is:\n', cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")


accuracy = cross_val_score(nb, X, y, scoring='accuracy')
```

```
print('Cross validation test results of accuracy:')
print(accuracy)
#get the mean of each fold
print("")
print("Accuracy result of Naive Bayes is:", accuracy.mean() * 100)
LR=accuracy.mean() * 100
def graph():
    import matplotlib.pyplot as plt
    data=[LR]
    alg="GaussianNB"
    plt.figure(figsize=(5,5))
    b=plt.bar(alg, data, color=("b"))
    plt.title("Accuracy comparison of Stock price", fontsize=15)
    plt.legend(b, data, fontsize=9)
graph()
TP = cm[0][0]
FP = cm[1][0]
FN = cm[1][1]
TN = cm[0][1]
print("True Positive :", TP)
print("True Negative :", TN)
print("False Positive :", FP)
```

```
print("False Negative :",FN)
print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN)
FNR = FN/(TP+FN)
print("True Positive Rate :",TPR)
print("True Negative Rate :",TNR)
print("False Positive Rate :",FPR)
print("False Negative Rate :",FNR)
print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)
def plot_confusion_matrix(cm2, title='Confusion matrix-Naive Bayes',
cmap=plt.cm.Blues):
    target_names=['Predict','Actual']
    plt.imshow(cm2, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(target_names))
```

```
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')


cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-Naive Bayes:')
print(cm2)
plot_confusion_matrix(cm2)
```

5.7 HTML Code:

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>TITLE</title>
<link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>
```

```
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>
<style>
.back{
 background-image: url("{{ url_for('static', filename='image/img.jpg') }}");
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-size: 100% 100%;
}
.white{
color:white;
}
.space{
margin:10px 30px;
padding:10px 10px;
background: palegreen;
width:500px
}
.gap{
padding:10px 20px;
}
</style>

</head>

<body class="back">
 <div>
          <div class="jumbotron">
```

```
        <h1 style="text-align:center">STOCK PRICE CLASSIFICATION USING MACHINE LEARNING TECHNIQUE </h1>

        </div>
    <!-- Main Input For Receiving Query to our ML -->
   <form class="form-group" action="{{ url_for('predict')}}"method="post">

                <div class="row">
                <div class="gap col-md-6 ">
                <label class="white" for="">Open</label>
                <input type="number" class="space form-control" step="0.01" name="Open" placeholder="Open" required="required" /><br>

                <label class="white"  for="">High</label>
                <input type="number" class="space form-control" step="0.01" name="High" placeholder="High" required="required" /><br>

                <label class="white"  for="">Low</label>
                <input type="number" class="space form-control" step="0.01" name="Low" placeholder="Low" required="required" /><br>

                <label class="white" for="">Close</label>
                <input type="number" class="space form-control" step="0.01" name="Close" placeholder="Close" required="required" /><br>

                <label class="white"  for="">Volume</label>
                <input type="number" class="space form-control" step="0.01" name="Volume" placeholder="Volume" required="required" /><br>
```

```
                <label class="white"  for="">Adj_Close</label>
                <input type="number" class="space form-control" step="0.01"
name="Adj_close" placeholder="Adj_close" required="required" /><br>

                </div>



                <div class="row">
                <div class="gap col-md-6 ">
                <label class="white" for="">Subjectivity</label>
                <input type="number" class="space form-control" step="0.01"
name="Subjectivity" placeholder="Subjectivity" required="required" /><br>

                <label class="white"  for="">Objectivity</label>
                <input type="number" class="space form-control" step="0.01"
name="Objectivity" placeholder="Objectivity" required="required" /><br>

                <label class="white"  for="">Positive</label>
                <input type="number" class="space form-control" step="0.01"
name="Positive" placeholder="Positive" required="required" /><br>

                <label class="white" for="">Neutral</label>
                <input type="number" class="space form-control" step="0.01"
name="Neutral" placeholder="Neutral" required="required" /><br>

                <label class="white"  for="">Negative</label>
                <input type="number" class="space form-control" step="0.01"
name="Negative" placeholder="Negative" required="required" /><br>
```

```
                    </div>
<div style="padding:2% 35%">
  <button type="submit" class="btn btn-success btn-block"
style="width:350px;padding:20px">Predict</button>
</div>

    </form>

   <br>
   <br>
<div  style="background:skyblue;padding:2% 40%">
   {{ prediction_text }}
</div>
 </div>
</body>
</html>
```

5.8 Flask Deploy:

```
import numpy as np

from flask import Flask, request, jsonify, render_template

import pickle

import joblib

app = Flask(__name__)

model = joblib.load('model.pkl')
```

```
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    print(final_features)
    prediction = model.predict(final_features)
    print(prediction)
    output = prediction[0]
    if output == 1:
        return render_template('index.html', prediction_text='STOCK PRICE NOT
OCCURED')
    else:
        return render_template('index.html', prediction_text='STOCK PRICE
OCCURED')
    print(output)
```

```
if __name__ == "__main__":
    app.run(host="localhost", port=8000)
```

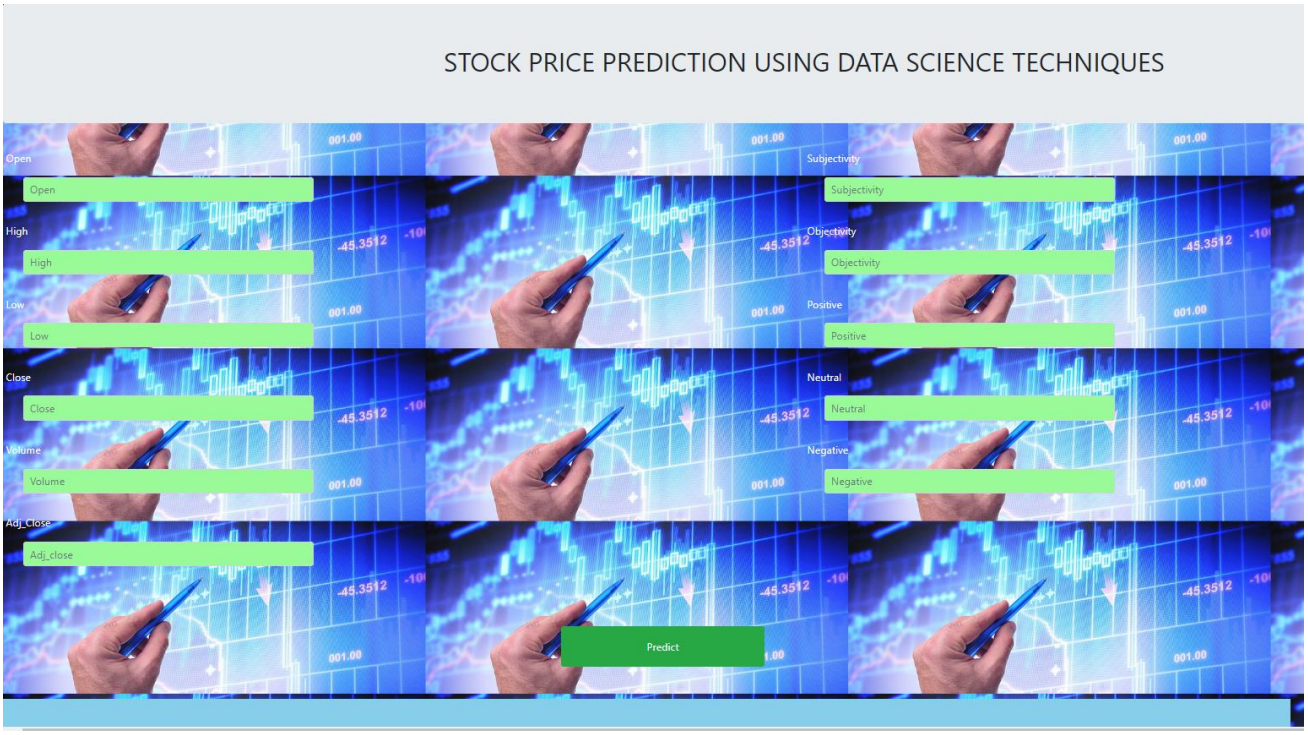


Fig 11

Testing

Test case description	Expected output	Actual output	Test status (P/F)
Price difference	Stock price increase/ decrease	Stock price increase/ decrease	p
Highest value	Stock price increase/ decrease	Stock price increase/ decrease	p
Lowest value	Stock price increase/ decrease	Stock price increase/ decrease	p

Share volume	Stock price increase/ decrease	Stock price increase/ decrease	p
Closing Price	Stock price increase/ decrease	Stock price increase/ decrease	P
General Index	Stock price increase/ decrease	Stock price increase/ decrease	P
Daily Share Volume	Stock price increase/ decrease	Stock price increase/ decrease	P
Monthly Share Volume	Stock price increase/ decrease	Stock price increase/ decrease	P
Opening Price	Stock price increase/ decrease	Stock price increase/ decrease	P

CHAPTER 6

RESULTS AND DISCUSSION

DATA PRE-PROCESSING/CLEANING:

loading the specified dataset while importing the library packages. To evaluate the missing values, duplicate values, and variable identification by data type, shape, and size. A validation dataset is a sample of data withheld from model training and used to measure model competence when fine-tuning models and processes that you may employ to make the greatest use of validation and test datasets when assessing your models. To analyse the uni-variate, bi-variate, and multi-variate processes, data cleaning and preparation steps include renaming the provided dataset, deleting columns, etc. Depending on the dataset, different procedures and methods will be used to clean the data. Data cleaning's main objective is to find and eliminate mistakes and abnormalities in order to maximise the value of data in analytics and decision making.

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	...	Top22	Top23	Top24	Top25	Para	Subjectivity	Objectivity	Positive	Neutral	Negative
0	1508	1	84	624	1277	327	50	242	230	460	...	1232	1447	71	984	84	95	25	0	25	99
1	1468	1	755	1203	270	1195	1227	994	1061	1031	...	1438	1466	1393	787	755	120	0	96	0	0
2	1455	1	501	1503	1358	431	264	83	865	1401	...	591	447	205	1336	501	111	9	0	9	106
3	1449	1	60	1303	1115	634	274	244	636	258	...	17	245	59	850	60	64	56	0	56	84
4	1444	0	256	1023	1010	1344	1160	1210	940	1199	...	388	403	901	1000	256	42	78	36	78	34

Table-1

DATA ANALYSIS OF VISUALIZATION:

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.
- How to summarize data distributions with histograms and box plots.

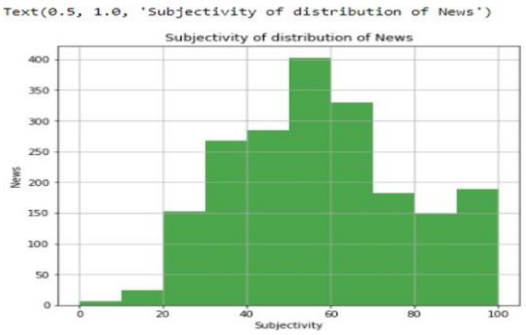


Fig 12

6.1 Test cases:

Test case Description	Expected output	Actual Output	Test Status(P/F)
Price Difference	Stock price increase/decrease	Stock price increase/decrease	P
Highest Value	Stock price increase/decrease	Stock price increase/decrease	P
Lowest Value	Stock price increase/decrease	Stock price increase/decrease	P
Share Volume	Stock price increase/decrease	Stock price increase/decrease	P
Closing price	Stock price increase/decrease	Stock price increase/decrease	P
General Index	Stock price increase/decrease	Stock price increase/decrease	P
Daily Share Volume	Stock price increase/decrease	Stock price increase/decrease	P
Monthly Share Volume	Stock price increase/decrease	Stock price increase/decrease	P
Opening Price	Stock price increase/decrease	Stock price increase/decrease	P

6.2 Output Screenshots:


Positive
Neutral
Negative
Predict

Fig 13

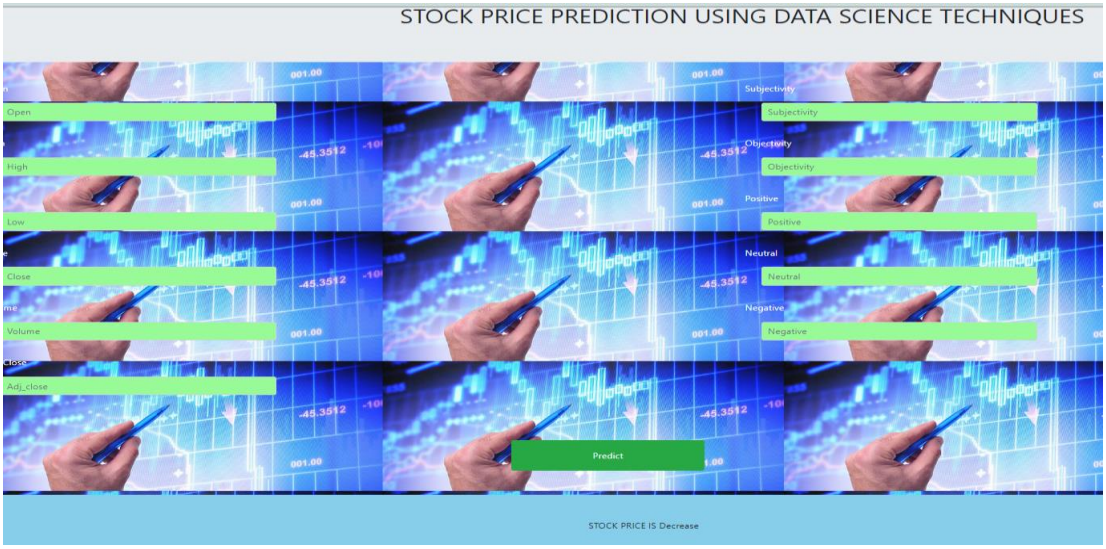

STOCK PRICE PREDICTION USING DATA SCIENCE TECHNIQUES
Open
High
Low
Close
Volume
Adj_close
Subjectivity
Objectivity
Positive
Neutral
Negative
Predict
STOCK PRICE IS Decrease

Fig 14

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The proposed system applied on this data and tried to analytical method started from knowledge cleaning and process, missing value, beta analysis and eventually model building and analysis. The best accuracy on public take a look at set is higher accuracy score are going to be determine. This application will facilitate to seek out the Prediction of Stock worth.

FUTURE ENHANCEMENT

- Stock price prediction to connect with cloud.
- To optimize the work to implement in Artificial Intelligence environment.

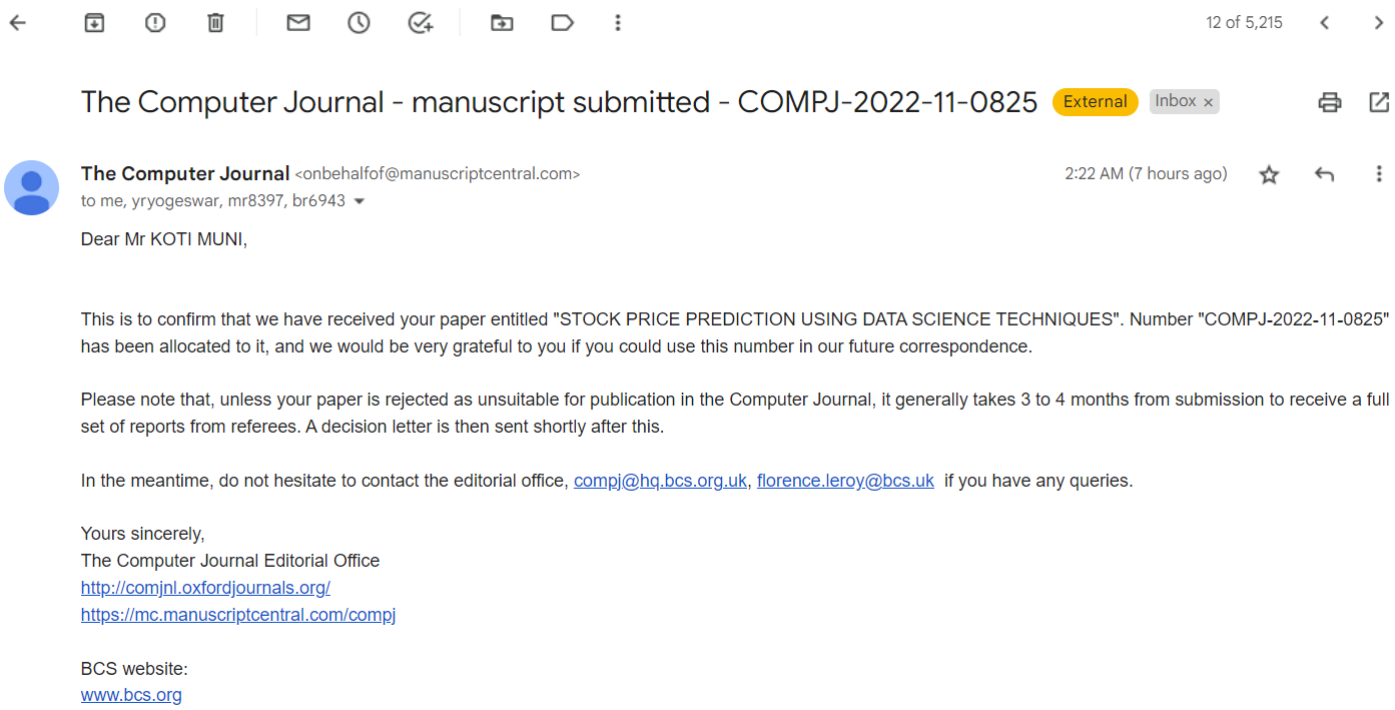
REFERENCES

[1]A Deep Learning Based Approach for Predicting the Price of Fresh Produce. Ifeanyi and Lobna Nassar, authors Muhammad Saad and Emmanuel Okwuchi 19-24 July 2020
[2] "Unsupervised real-time anomaly detection for streaming data," Neurocomputing, vol. 262, pp. 134–147, Nov. 2017, S. Ahmad, A. Lavin, S. Purdy, and Z. Agha.
[3] A Hammerstein-Wiener deep hybrid fuzzy neural network for predicting stock prices. Authors: Xie Chen, Chai Quek, and DeepuRajan International Conference on Artificial Intelligence in Information and Communication, 16 April 2020 (ICAIIC).
[4] A Deep Learning Based Approach for Predicting the Price of Fresh Produce. Ifeanyi and Lobna Nassar, authors Muhammad Saad and Emmanuel Okwuchi 19-24 July 2020 Unsupervised real-time anomaly detection for streaming by S. Ahmad, A. Lavin, S. Purdy, and Z. Agha
[5] Deep Learning Based Approach for Predicting the Price of Fresh Produce. Ifeanyi Emmanuel LobnaNassar is the author. Saad Muhammad Okwuchi 19-24 July 2020

APPENDIX A

PUBLICATION DETAILS

We submitted our research paper for publication at Oxford University PRESS.Proof of publication is attached in figure B.1 The research paper cover page has been attached below.



12 of 5,215

The Computer Journal - manuscript submitted - COMPJ-2022-11-0825 External Inbox ×

The Computer Journal <onbehalfof@manuscriptcentral.com> 2:22 AM (7 hours ago)
to me, yryogeswar, mr8397, br6943

Dear Mr KOTI MUNI,

This is to confirm that we have received your paper entitled "STOCK PRICE PREDICTION USING DATA SCIENCE TECHNIQUES". Number "COMPJ-2022-11-0825" has been allocated to it, and we would be very grateful to you if you could use this number in our future correspondence.

Please note that, unless your paper is rejected as unsuitable for publication in the Computer Journal, it generally takes 3 to 4 months from submission to receive a full set of reports from referees. A decision letter is then sent shortly after this.

In the meantime, do not hesitate to contact the editorial office, compj@hq.bcs.org.uk, florence.leroy@bcs.uk if you have any queries.

Yours sincerely,
The Computer Journal Editorial Office
http://comjnl.oxfordjournals.org/
https://mc.manuscriptcentral.com/compj

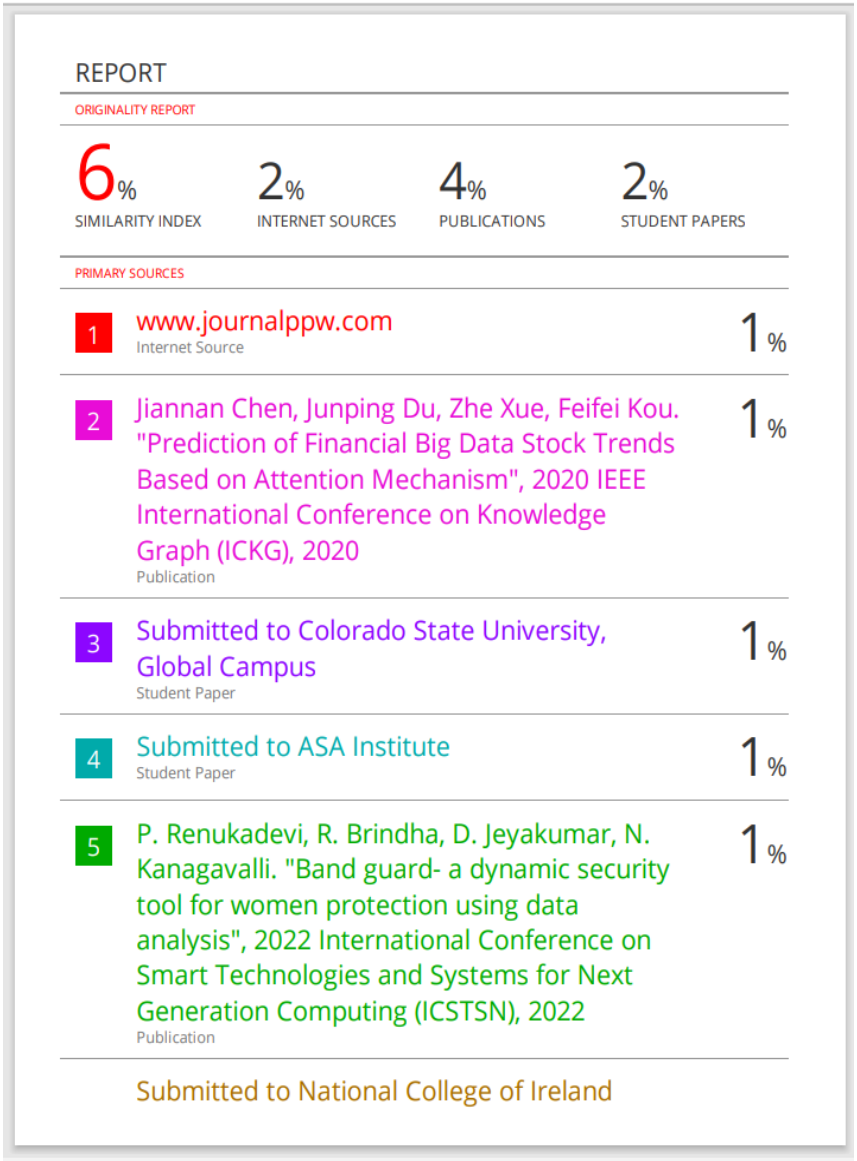
BCS website:
www.bcs.org

FigureB.1:PublicationNotification

APPENDIX B

PLAGIARISM REPORT

STOCK PRICE PREDICTION USING DATA SCIENCE TECHNIQUES



REPORT

ORIGINALITY REPORT

6%	2%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.journalppw.com Internet Source	1%
2	Jiannan Chen, Junping Du, Zhe Xue, Feifei Kou. "Prediction of Financial Big Data Stock Trends Based on Attention Mechanism", 2020 IEEE International Conference on Knowledge Graph (ICKG), 2020 Publication	1%
3	Submitted to Colorado State University, Global Campus Student Paper	1%
4	Submitted to ASA Institute Student Paper	1%
5	P. Renukadevi, R. Brindha, D. Jeyakumar, N. Kanagavalli. "Band guard- a dynamic security tool for women protection using data analysis", 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), 2022 Publication	1%
	Submitted to National College of Ireland	

6	Student Paper	<1 %
7	Malhar Bangdiwala, Rutvik Choudhari, Adwait Hegde, Abhijeet Salunke. "Using ML Models to Predict Points in Fantasy Premier League", 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), 2022 Publication	<1 %
8	Xu Zhang, Mingxuan Du, Yixian Wang, Huiting Zhang, Yun Guo. "Research on power grid fault diagnosis based on a quantitative representation of alarm information", IEEE Transactions on Industrial Electronics, 2022 Publication	<1 %
9	Xiaoyan Zhu, Ting Wang, Jiayin Wang, Ying Xu, Yuqian Liu. "A new multiple instance algorithm using structural information", 2021 IEEE International Conference on Data Mining (ICDM), 2021 Publication	<1 %
10	Xie Chen, Deepu Rajan, Chai Quek. "A Deep Hybrid Fuzzy Neural Hammerstein-Wiener Network for Stock Price Prediction", 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2020 Publication	<1 %