SUPER MARKET SALES PREDICTION

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

D. Supriya	321910302005
B. Muni kumar	321910302004
K. Sathya Sai Venkat	321910302026
A. Mythreyi	321910302036

Under the esteemed guidance of

Dr.TANVIR H. SARDAR

Assistant Professor

Department of Computer Science & Engineering,

GITAM SCHOOL OF TECHNOLOGY

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Deemed to be University)

Bengaluru Campus.

November 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled **“SUPER MARKET SALES PREDICTION”** is a bonafide record of work carried out by **D.SUPRIYA(321910302005), B.MUNI KUMAR(321910302004),K.SATYA SAI VENKAT(321910302026), A.MYTHREYI(321910302036)** submitted in partial fulfillment of requirement for the award of degree of **Bachelors of Technology in Computer Science and Engineering**.

Project Guide.

SIGNATURE OF THE GUIDE

____

Dr.Tanvir H. Sardar

Assistant Professor

Head of the Department.

SIGNATURE OF THE HOD

____

Prof. Vamsidhar.Y

Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project report entitled **"SUPER MARKET SALES PREDICTION"** is an original work done in the **Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University)** submitted in partial fulfillment of the requirements for the award of the degree of **B.Tech.** in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree.

Date:

Registration No(s).	**Name(s)**	**Signature(s)**
321910302005	D. Supriya	
321910302004	B. Muni Kumar	
321910302026	K. Satya Sai Venkat	
321910302036	A. Mythreyi	

ACKNOWLEDGEMENT

The successful completion of any Project would be incomplete without mentioning the people who made it possible, whose constant guidance and encouragement has concluded our efforts with success. We take this opportunity to express our profound gratitude and appreciations to all those who helped us directly or indirectly towards positive outcome of the Project.

We would like to thank **Dr. Nageshwara Rao T**, Ph.D., Director, GITAM School of Technology for all the facilities provided.

We would like to thank HOD **Dr. VAMSIDHAR YENDAPALLI**, Ph.D., Professor, Department of

Computer Science and Engineering for his support and encouragement that went a long way in successful completion of this Project.

We consider that as a privilege to express our gratitude and respect to **Dr.Tanveer sardar**, assistant professor, department of Computer Science and Engineering for being our internal guide for his integral and incessant support offered to us throughout the course of this project and for constant source of inspiration throughout the Seminar.

We would like to thank our project coordinator **Rati Bhan** for her integral and incessant support offered to us throughout the course of this project and for constant source of inspiration throughout the Project.

Finally, We would like to acknowledge all the teaching and non-teaching staff members of the Computer Science and Engineering Department, for the support in completing of the project in time.

Student Name's	**Registration No.**
D. Supriya	321910302005
B. Muni Kumar	321910302004
K. Satya Sai Venkat	321910302026
A. Mythreyi	321910302036

ABSTRACT

Sales prediction, or sales forecasting, allows businesses to predict how much revenue to expect in a given time frame. This information helps businesses to make informed decisions and estimate performance in the short and long terms. For example, companies may use sales prediction to allocate budgets, manage the workforce, purchase equipment, and control cash flow. It can also be a contributing factor when seeking investment capital as you plan for future business growth. Sales forecasting is an essential task in the retail management field. Intelligent forecasting using machine learning techniques can help discover the selection of feature variables that influence prediction of sales growth.

A Python program implementation is adopted to compute, develop and visualizing forecasting model of historical sales data based on super market sales opted to do the effective sales promotion. Python supports working on predictive algorithms through accessing from Python libraries. For this purpose, it relies on the past observations-based transaction data set file as an input to produce output without worrying about the underlying mechanism. The results indicated that TV media is the feature variable that influences the prediction of sales of linear regression model. Regression results of OLS model type are displayed with coefficient values to substitute in the linear regression equation. Seaborne library of Python is used to generate the visualization of charts and graphs.

TABLE OF CONTENTS

LIST OF FIGURES:

LIST OF TABLES

1.INTRODUCTION

1.1 What is Sales Prediction

Sales Forecasting is the process of using a company’s sales records over the past years to predict the short-term or long-term sales performance of that company in the future. This is one of the pillars of proper financial planning. As with any prediction-related process, risk and uncertainty are unavoidable in Sales Forecasting too.

Hence, it’s considered a good practice for Sales forecasting teams to mention the degree of uncertainties in their forecast. Sales Forecasting is a globally-conducted corporate practice where a number of objectives are identified, action-plans are chalked out as well as budgets and resources are allotted to them.

The first step to proper Sales Forecasting is to know the things that fall within your domain directly as a salesperson. This usually relates to your sales staff, clients and prospects. Other factors to consider during the setup of a forecast are the negative ones like − uncertainty, abrupt changes in consumer shopping patterns, etc.

One of the most common yet basic challenges that the management of companies face in making business sales forecasts is that their usual approach is a “top to down” one. This approach leaves very little scope for interaction with the sales manager and the salespersons during the data collection process.

It pays to have accurate sales forecasts. Research shows that companies with accurate sales forecasts are over 7% more likely to hit their revenue and sales quotas. If sales is a game of inches, precise forecasting can provide that extra inch of leverage that allows you to hit your annual targets and continue year-over-year growth.

Sales forecasting is an indispensable tool that offers several benefits, such as predicting consumer demand, managing inventory, strategic planning, expectation-setting, and devising a marketing strategy. However, nearly 80% of sales organizations miss their forecasts by at least a 10% margin.

the sales prediction is proposed to forecast the sales of super market using machine learning algorithms. Sales forecasting is done by analysing customer purchasing behaviour and it plays an important role in modern business intelligence. Forecasting future sales demand is key to business and business planning activities. Forecasting helps business organizations to make improvements, to make changes to business plans and to provide a stock storage solution. Forecast is determined by the use of data or information from past works and the consideration of recognized feature in future. Sales forecasting plays a vital role in strategic planning and market strategy for every company to assess past and present sales statistics and predict potential results. Overall, accurate sales forecasting helps the company to run more productively and efficiently, to save money on forecasts or predictions. In the proposed study, the regression techniques are used to train and test our dataset. The data is processed to select the features and extract those features. Accurate projections make it easier for the shop to boost demand growth and a higher degree of sales generation. It produces better prediction rate.

Fig 1

1.2 Features of Sales Prediction

Analyze performance with data visualization.

Compare forecasts based on multiple modeling techniques.

Execute sales forecast simulations and outcomes.

1.3 Advantages and Disadvantages of Sales Prediction

Advantages:

Sales Planning

Allocation of Resources

Key Factor in Business Operations

Basis of Salesforce Planning

Major Role in Success

Disadvantages:

Lake of Sales History

Change in Business Environment

Change in Consumer Behaviour

Based on Assumptions

Lake of Facts and Data

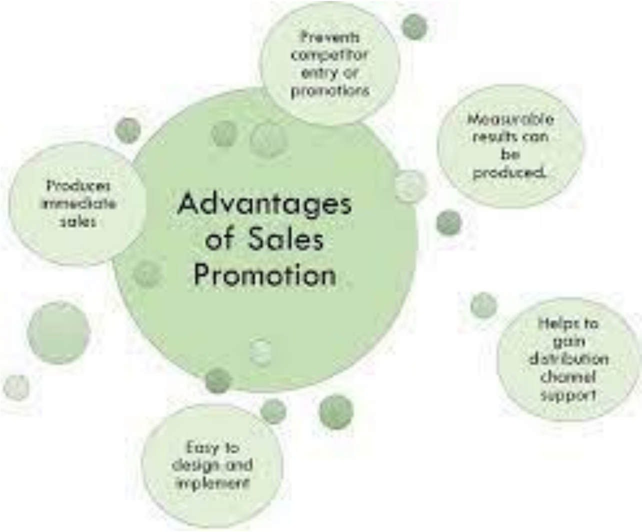

Prevents competitor entry or promotions
Measurable results can be produced.
Produces immediate sales
Advantages of Sales Promotion
Helps to gain distribution channel support
Easy to design and implement

Fig 2

2.OBJECTIVES

Step By Step Process

Data Set Gathering

Analysing the data set

Cleaning the data set

Exploratory Data Analysis

Visualization

Testing and training the dataset

Building the model

Developing predictive system

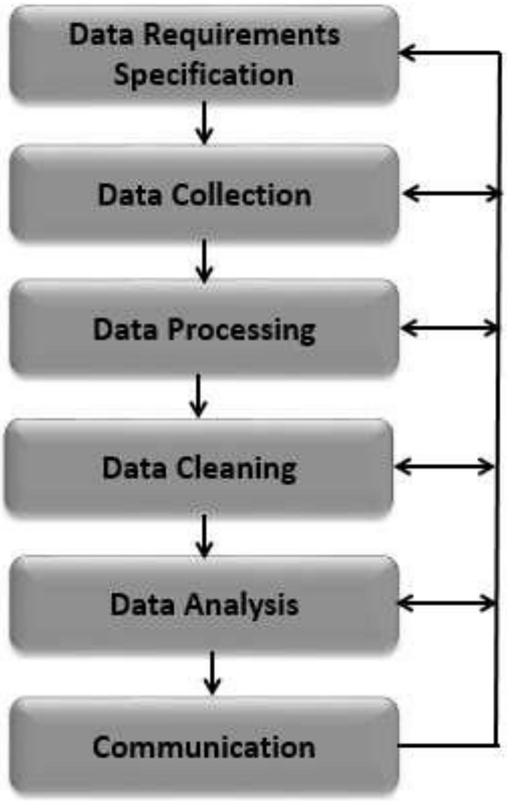


Fig 3

3.LITERATURE SURVEY

The super market dataset is used. Exploratory analysis stages involved in the data mining model include

data understanding, preparation, modelling, evaluation and deployment. The forecast is composed of a smoothed

averaged adjusted for a linear trend. Then the forecast is also adjusted for seasonality. Machine learning

algorithms such as K -nearest neighbours, Decision Tree , Gradient Boost Tree ,Ada boost, Random forest, Bagging, Xgb classifier, Naive bayes, SVM, Extra tree classifiers

used in prediction of future sales.

Based on the performance Random Forest Algorithm and Extra tree provides 100% overall accuracy and the second stands

Gradient boost Algorithms with nearly 88.0% overall accuracy and followed by Ada boost with

67% accuracy followed by KNN with 64.75% and decision tree 63.87%, XGB with 62.625%, SVM with 55.50%, Naive bayes 55.125% and finally Bagging with 40%

best fit for the model is Random Forest and Extra tree which provides the maximum accuracy of prediction across all

the algorithms.

“Machine Learning Models for Sales Forecasting”

“Super market_sales-sheet1.csv” dataset is used to predict the future sales. The calculations were conducted in the

Python environment using the main packages pandas, sklearn, numpy, keras, matplotlib, seaborn. Analysis is done using Jupiter notebook.

Authors Name, Journal Name, Vol., Year, Page	Title of the Paper	Inference	Research Gap	Relevance with the present work
Saraswathi, Renuka Devi Gayatri Devi, Nandhini Devi, Naveen **AIP Conference Proceedings:** 2387(1):140038 **November** 2021	Sales Prediction Using Machine Learning Approaches	In the proposed study, the linear regression and logistic regression model are analysed and Simple Linear Regression (SLR) and Multiple Linear Regression (MLR) are trained and tested for our dataset.	We can also use clustering method for the prediction. For the high accuracy we can use the Random Forest. Low accuracy should be avoided.	EDA is used to the visual techniques. Classifiers are used but most probably XGB Classifier is used newly and also accurately.
Soham Patangia , Rachana Mohite, Kevin Shah, Gaurav Kolhe , Madhura Mokashi, Prajakta Rokade Volume 09, Issue 09 **September** 2020	Sales prediction Of market using machine learning	A data maturity profile for retail businesses and highlight future research directions. Association Rules is one of the data mining techniques which is used for identifying the relation between one item to another. Logistic Regression is used for the analysis.	Random choice of Pearson's correlation was chosen to measure functional connectivity strength between different items.	Big data is point out for the analysis.

4. SOFTWARE AND HARDWARE SPECIFICATIONS

4.1 Hardware Requirements*:*

Memory**:** 128 Gb

Processor**:** Intel or Ryzen

Operating System: Windows 10 or Windows11

Hard Disk**:** 2 Gb or Higher

HARDWARE REQUIREMENTS:

Fig 4

4.2 Software Requirements: -

IDE**:** Jupiter Notebook

Language**:** Python

Internet Browser**:** Internet Explorer 6.0 or above, Google chrome, Mozilla Firefox

SOFTWARE REQUIREMNETS:

Fig 5

5. LIBRARIES

5.1 Pandas

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data. It is used for data analysis in Python and developed by Wes McKinney in 2008.

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python

5.2Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely.

NumPy is a python library mainly used for working with arrays and to perform a wide variety of mathematical operations on arrays. NumPy guarantees efficient calculations with arrays and matrices on high-level mathematical functions that operate on these arrays and matrices

5.3Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

5.4Pyplot

Pyplot is an API (Application Programming Interface) for Python's matplotlib that effectively makes matplotlib a viable open-source alternative to MATLAB. Matplotlib is a library for data visualization, typically in the form of plots, graphs and charts.

Pyplot is a sub-module of the matplotlib library for Python. It is a library consisting of a collection of functions/methods used for plotting simple 2D graphs using Python. Pyplot can be imported using import matplotlib.

5.5Seaborn

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with data frames and the Pandas library. The graphs created can also be customized easily.

5.6 SciPy

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely.

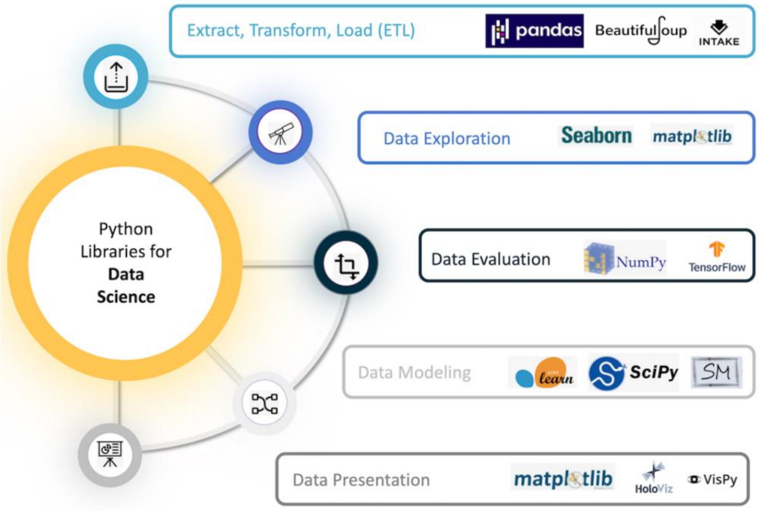
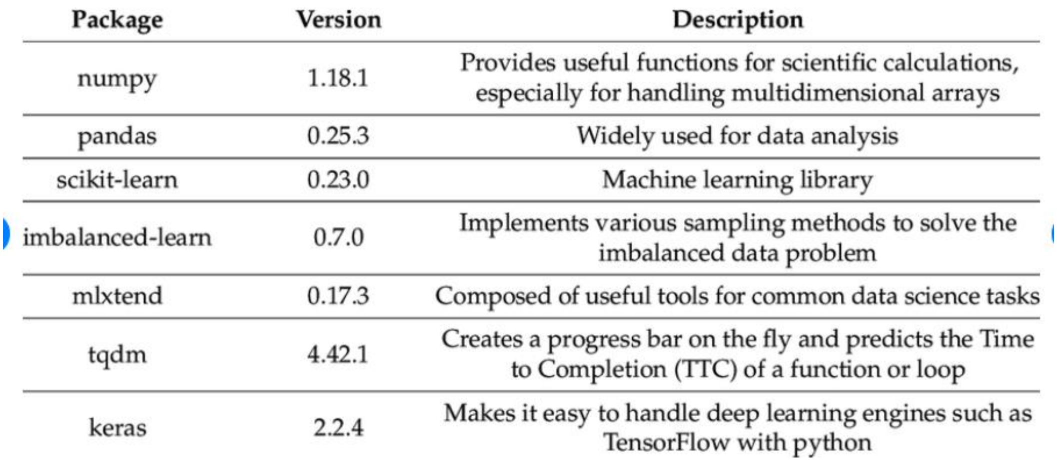


Fig 6

5.7Uses of Libraries



Package	Version	Description
numpy	1.18.1	Provides useful functions for scientific calculations, especially for handling multidimensional arrays
pandas	0.25.3	Widely used for data analysis
scikit-learn	0.23.0	Machine learning library
imbalanced-learn	0.7.0	Implements various sampling methods to solve the imbalanced data problem
mlxtend	0.17.3	Composed of useful tools for common data science tasks
tqdm	4.42.1	Creates a progress bar on the fly and predicts the Time to Completion (TTC) of a function or loop
keras	2.2.4	Makes it easy to handle deep learning engines such as TensorFlow with python

Fig 7

6.PROBLEM STATEMENT

To find out what role certain properties of an item play and how they affect

their sales by understanding Super Market sales. A predictive model can be built to find out for every store, the key factors that can increase their sales and what changes could be made to the product or store's characteristics.

The data scientists have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store. Using this model, super markets will try to understand the properties of products and stores which play a key role in increasing sales.

7.DESIGNING

7.1 What is an Classifiers

A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more of a set of "classes**.**" One of the most common examples is an email classifier that scans emails to filter them by class label: Spam or Not Spam.

In data science, a classifier is a type of machine learning algorithm used to assign a class label to a data input. An example is an image recognition classifier to label an image (e.g., "car," "truck," or "person").

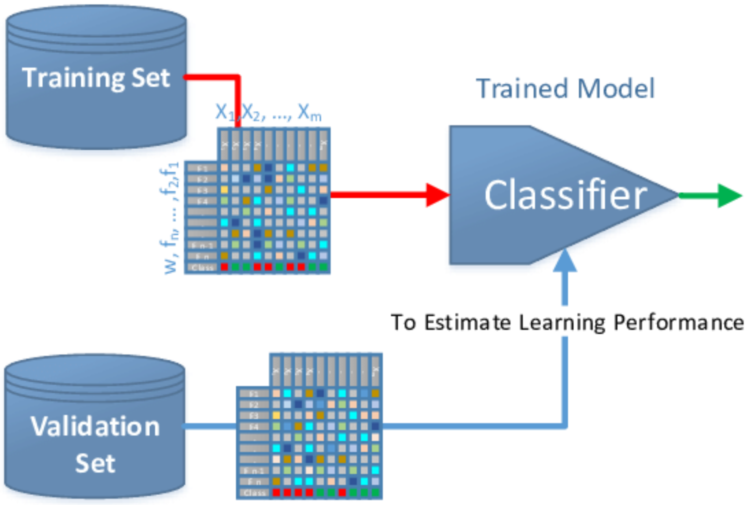


Fig 8

7.2 Classifiers used in our project

Kneighbors Classifier

Support vector system

Naive Bayes

Decision tree classifier

Random Forest Classifier

AdaBoost Classifier

Gradient Boosting Classifier

XGB Classifier

Extra Tree Classifier

Bagging

7.3 K-nearest neighbor Classifier

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. 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 This KNN algorithm is used for regression as well as for the classification but mostly it is used for the classification problems

sklearn.neighbors.KNeighborsClassifier(neighbours int, default=5)

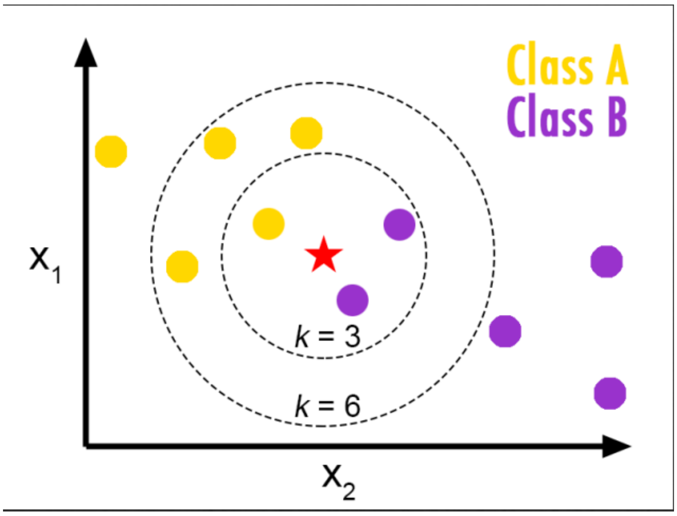


Fig 9

7.4 Support vector machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to

create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.
sklearn.svm import SVC

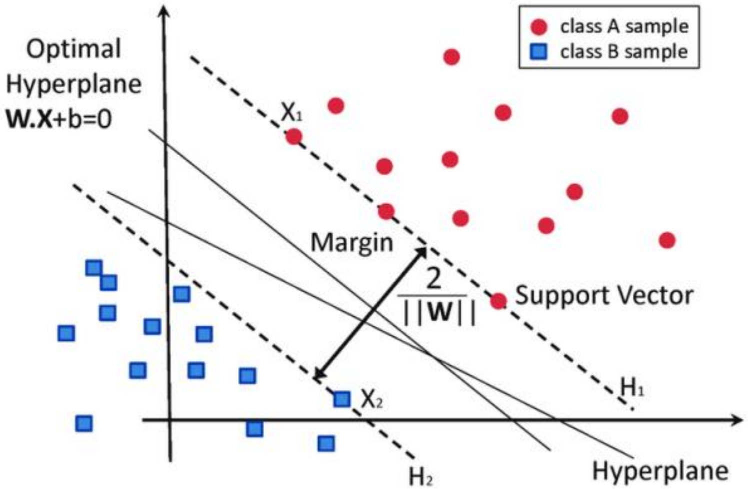


Fig 10

7.5 Naive bayes

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a highdimensional training dataset. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true

7.6 Decision tree Classifier

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a treestructured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome fromsklearn.treeimport DecisionTreeClassifier dtree=DecisionTreeClassifier(max_depth =6, random_state=123,criterion='entropy')

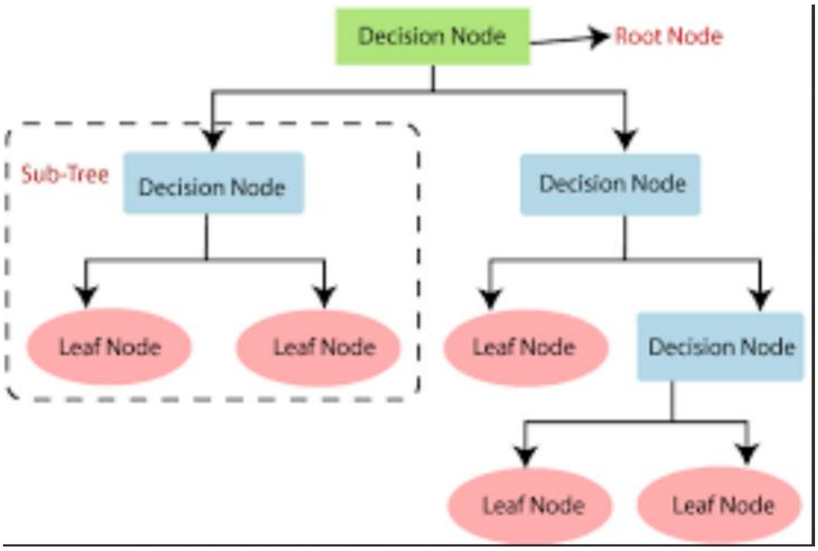


Fig 11

7.7 Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

from sklearn.ensemble import RandomForestClassifier rfc=RandomForestClassifier()

7.8 Ada boost Classifier

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the

weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.
from sklearn.ensemble import AdaBoostClassifier adb = AdaBoostClassifier(base_estimator = None

7.9 Gradient boosting Classifier

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

from sklearn.ensemble import GradientBoostingClassifier gbc=GradientBoostingClassifier()

7.10 XGB classifier

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems . XGBoost provides a wrapper class to allow models to be treated like classifiers or regressors in the scikit-learn framework This means we can use the full scikit-learn library with XGBoost models. The XGBoost model for classification is called XGBClassifier. We can create and and fit it to our training dataset

7.11 Extra tree classifier

It is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it's classification result.
It is also easy to use given that it has few key hyperparameters and sensible heuristics for configuring these hyperparameters

```
from sklearn.ensemble import ExtraTreesClassifier etc =
ExtraTreesClassifier(n_estimators=100, random_state=0
```

7.12 Bagging

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction
This algorithm encompasses several works from the literature. When random subsets of the dataset are drawn as random subsets of the samples, then this algorithm is known as Pasting

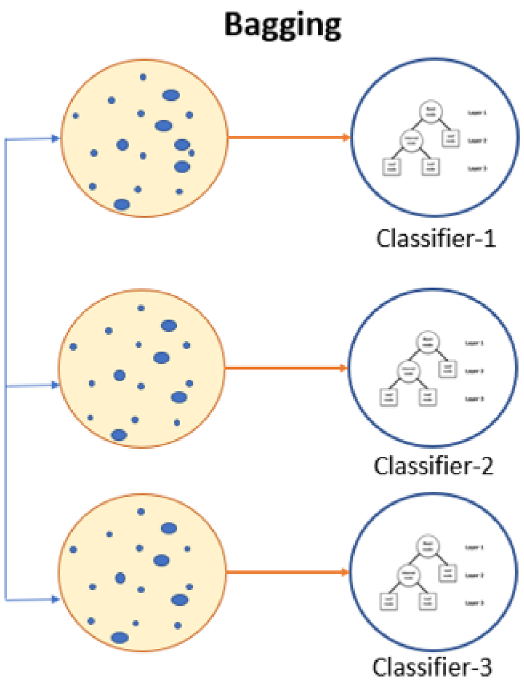
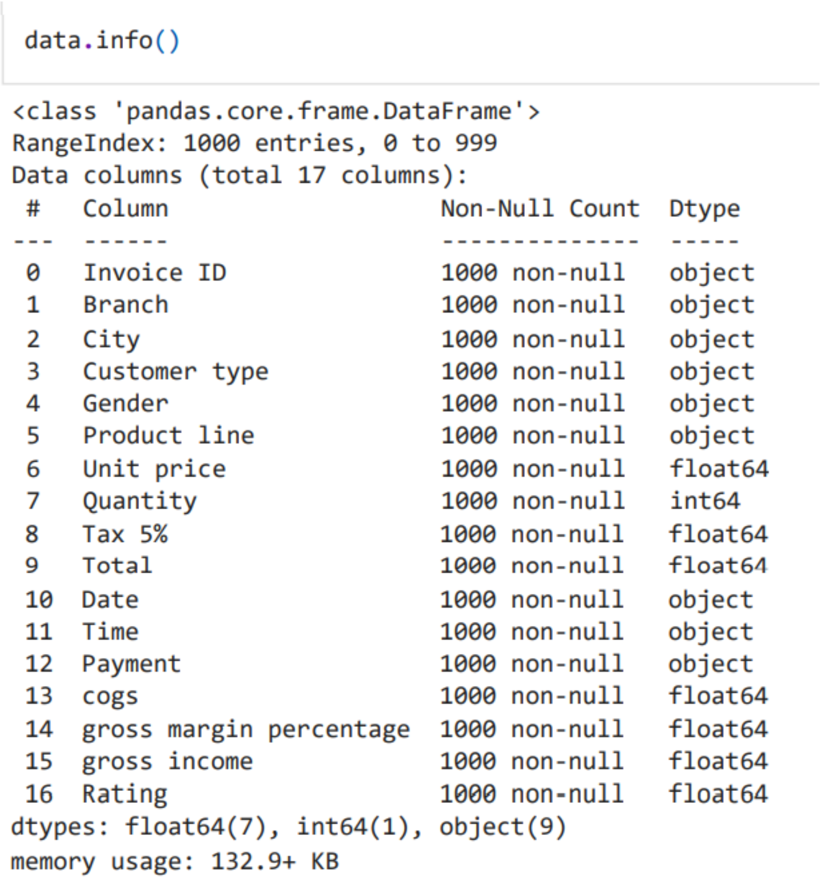


Fig 12

8.DATASET

The challenging situation we faced during the experiments was to find a suitable sales dataset. We used super market_sales-sheet1.csv dataset. This is available publicly available in Kaggle.it comprises approximately 17 parameters. The shape of the data set is (1000, 17)

Information of the data set is follows



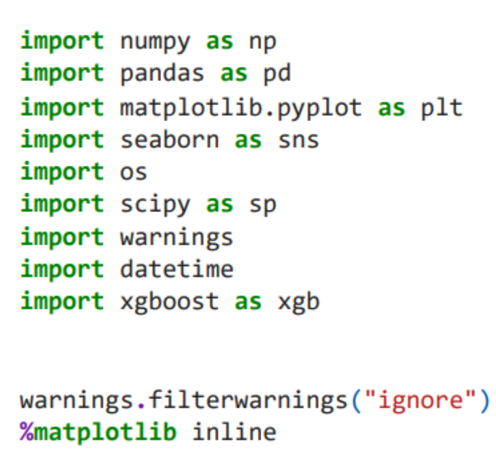
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Invoice ID               1000 non-null   object
 1   Branch                   1000 non-null   object
 2   City                     1000 non-null   object
 3   Customer type            1000 non-null   object
 4   Gender                   1000 non-null   object
 5   Product line             1000 non-null   object
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64
 8   Tax 5%                   1000 non-null   float64
 9   Total                    1000 non-null   float64
 10  Date                     1000 non-null   object
 11  Time                     1000 non-null   object
 12  Payment                  1000 non-null   object
 13  cogs                     1000 non-null   float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   1000 non-null   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

Fig 13

9. Implementation

Python provides various libraries for data processing. Some of them are pandas numpy matplotlib. pyplot seaborn scipy. These are vast libraries which helped us to create applications and models in a variety of fields



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
import scipy as sp
import warnings
import datetime
import xgboost as xgb


warnings.filterwarnings("ignore")
%matplotlib inline
```

Fig 14

9.1 Exploratory data analysis (EDA Analysis)

EDA is an approach to analyse the data using visual techniques.

EDA is applied to investigate the data and summarize the key insights. It will give us the basic understanding of our data, it's distribution, null values and much more. You can either explore data using graphs or through some python functions. Some examples are histogram and box plot.

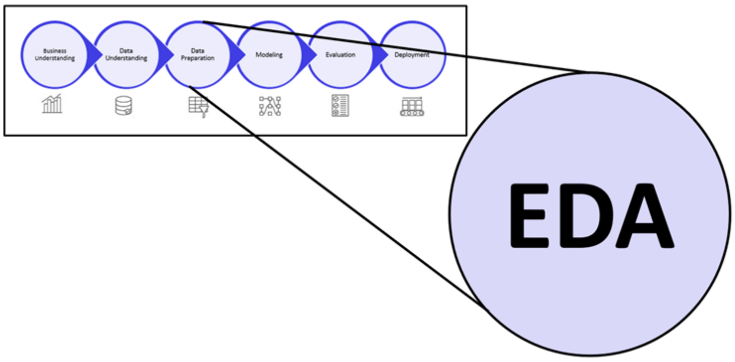


Fig 15

9.1.1 Histogram

A histogram is a graphical display of data using bars of different heights. It is a graphical representation of numerical data

figsize : tuple (width, height) - The size of the output image(graph)

9.1.2 Heat Map

It is a graphical representation of data that uses a system of color-coding to represent different values

9.1.3 Box Plot

It is a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”)

9.1.4 Pair Plot

It is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters.

9.1.5 Scatter Plot

Scatter plots are the graphs that present the relationship between two variables in a data-set. It represents data points on a two-dimensional plane or on a Cartesian system. The independent variable or attribute is plotted on the X-axis, while the dependent variable is plotted on the Y-axis

9.1.6 Joint Plot

it displays a relationship between 2 variables (bivariate) as well as 1D profiles (univariate) in the margins. This plot is a convenience class that wraps JointGrid

9.1.7 Cat Plot

Catplot is a relatively new addition to Seaborn that simplifies plotting that involves categorical variables

9.1.8 LM Plot

It shows a line on a 2 dimensional plane. You can plot it with seaborn or matlotlib depending on your preference. The examples below use seaborn to create the plots, but matplotlib to show

9.1.9 KDE Plot

KDE Plot described as Kernel Density Estimate is used for visualizing the Probability Density of a continuous variable. It depicts the probability density at different values in a continuous variable. We can also plot a single graph for multiple samples which helps in more efficient data visualization

9.1.10 Line Plot

It as a graph that displays data as points or check marks above a number line, showing the frequency of each value

9.1.11 Bar Plot

It shows the relationship between a numeric and a categoric variable. Each entity of the categoric variable is represented as a bar. The size of the bar represents its numeric value

9.2 Training and testing the data

Train/Test is a method to measure the accuracy of your model. It is called Train/Test because you split the data set into two sets: a training set and a testing set. 80% for training, and 20% for testing. You train the model using the training set. This is the phase where we the results that is nothing but the accuracy, precision, recall and f1 score

9.2.1 Precision

$$t_p/(t_p+f_p)$$

Where,
tp is number of true positives
fp is number of false positives

9.2.2 Recall

$$t_p/(t_p+f_n)$$

Where,
tp is number of true positives
fn is number of false negatives

9.2.3 F1 Score

$$2*(precision*recall)/(precision+recall)$$

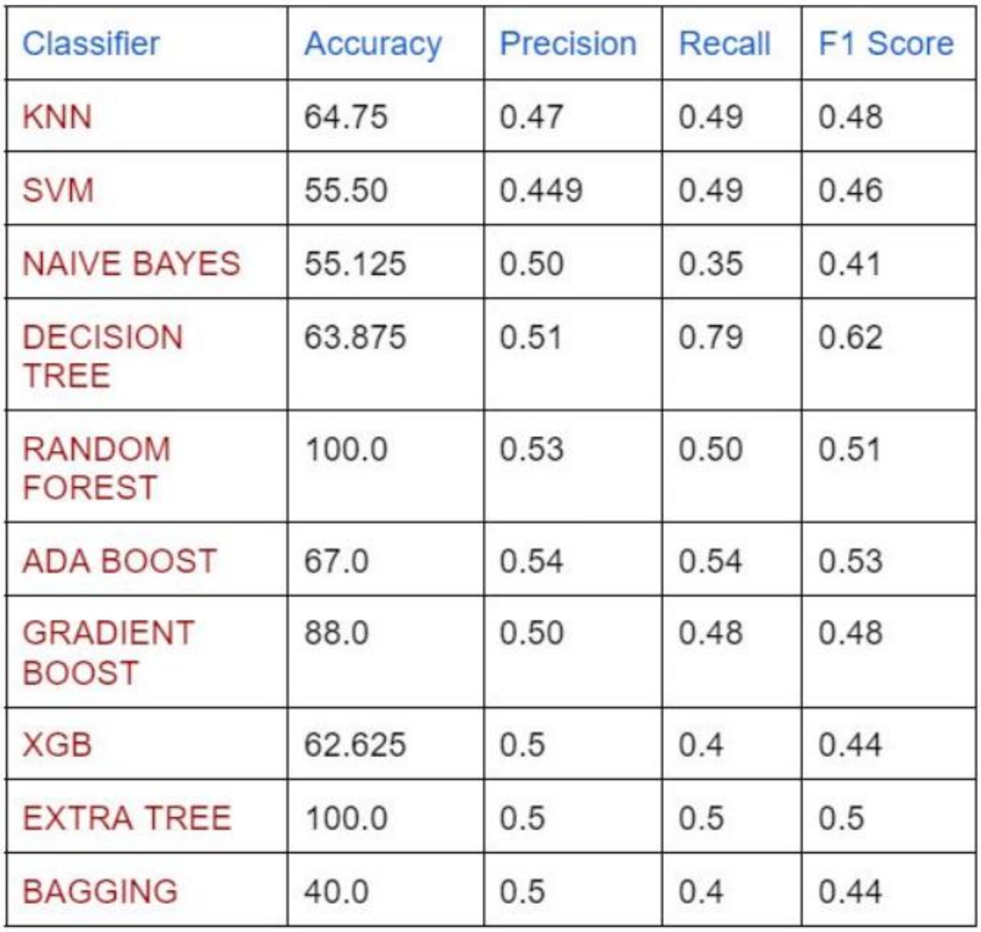
Where,
Precision is the obtained value
Recall is the obtained value

9.2.4 Accuracy

Accuracy is how close a given set of measurements are to their true value, while precision is how close the measurements are to each other.

Every classifier has its own formula to calculate the accuracy value

9.3 Overall output values for our data set



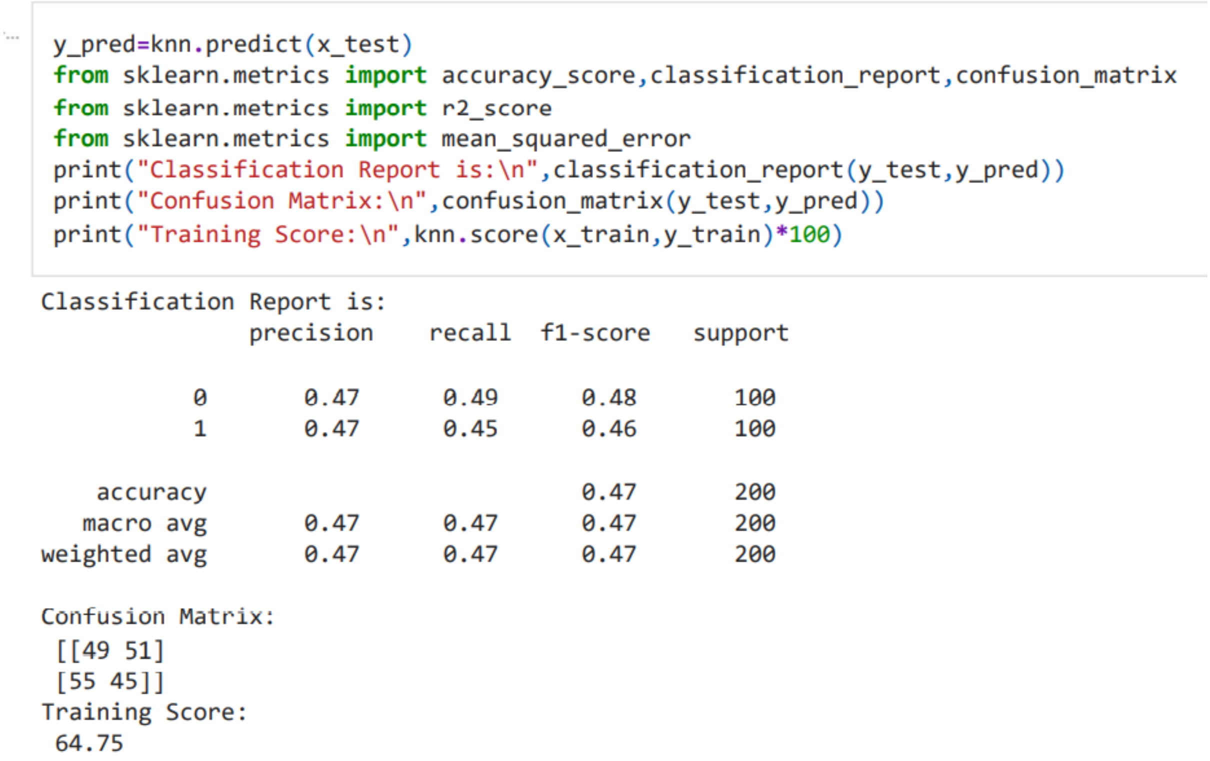
Classifier	Accuracy	Precision	Recall	F1 Score
KNN	64.75	0.47	0.49	0.48
SVM	55.50	0.449	0.49	0.46
NAIVE BAYES	55.125	0.50	0.35	0.41
DECISION TREE	63.875	0.51	0.79	0.62
RANDOM FOREST	100.0	0.53	0.50	0.51
ADA BOOST	67.0	0.54	0.54	0.53
GRADIENT BOOST	88.0	0.50	0.48	0.48
XGB	62.625	0.5	0.4	0.44
EXTRA TREE	100.0	0.5	0.5	0.5
BAGGING	40.0	0.5	0.4	0.44

Fig 16

10. Result

The following are the overall results of our work where we got 100 % accuracy for two classifiers, they are random forest and extra tree classifier.

10.1 K-nearest neighbor Classifier

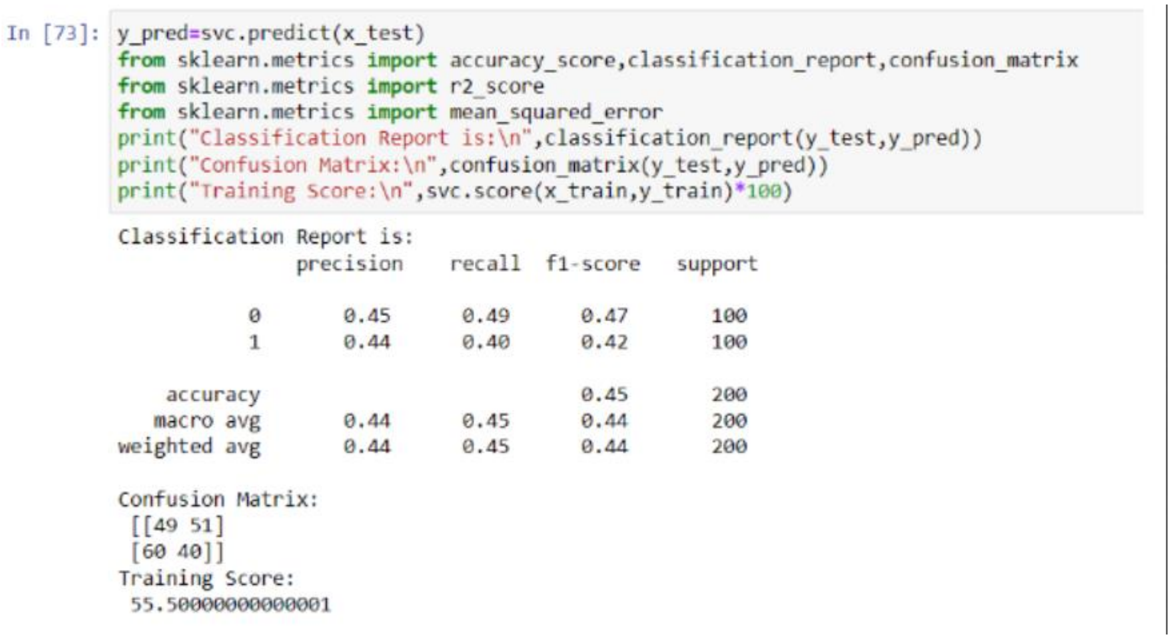


```
y_pred=knn.predict(x_test)
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
print("Classification Report is:\n",classification_report(y_test,y_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
print("Training Score:\n",knn.score(x_train,y_train)*100)
```

```
Classification Report is:
               precision    recall  f1-score   support

           0       0.47      0.49      0.48       100
           1       0.47      0.45      0.46       100

    accuracy                           0.47       200
   macro avg       0.47      0.47      0.47       200
weighted avg       0.47      0.47      0.47       200

Confusion Matrix:
 [[49 51]
 [55 45]]
Training Score:
 64.75
```

Fig 17

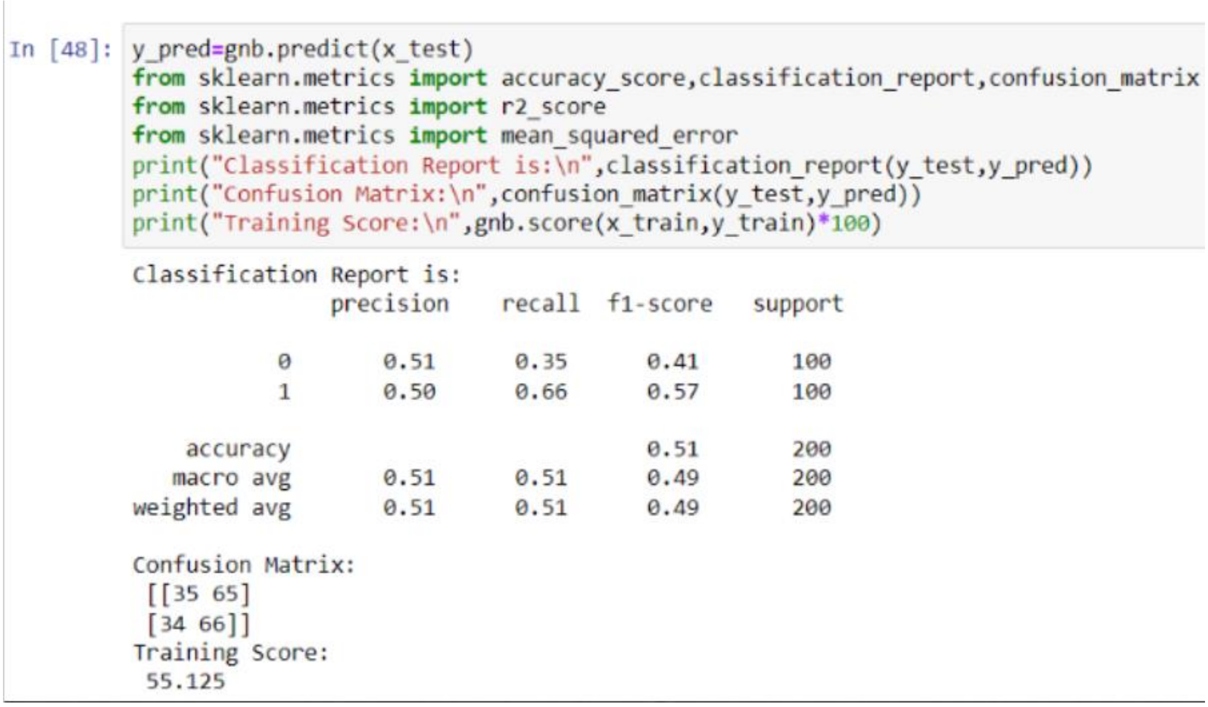
10.2 Support vector machine



```
In [73]: y_pred=svc.predict(x_test)
         from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
         from sklearn.metrics import r2_score
         from sklearn.metrics import mean_squared_error
         print("Classification Report is:\n",classification_report(y_test,y_pred))
         print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
         print("Training Score:\n",svc.score(x_train,y_train)*100)

         Classification Report is:
                       precision    recall  f1-score   support

                    0       0.45      0.49      0.47       100
                    1       0.44      0.40      0.42       100

             accuracy                           0.45       200
            macro avg       0.44      0.45      0.44       200
         weighted avg       0.44      0.45      0.44       200

         Confusion Matrix:
          [[49 51]
          [60 40]]
         Training Score:
          55.50000000000001
```

Fig 18

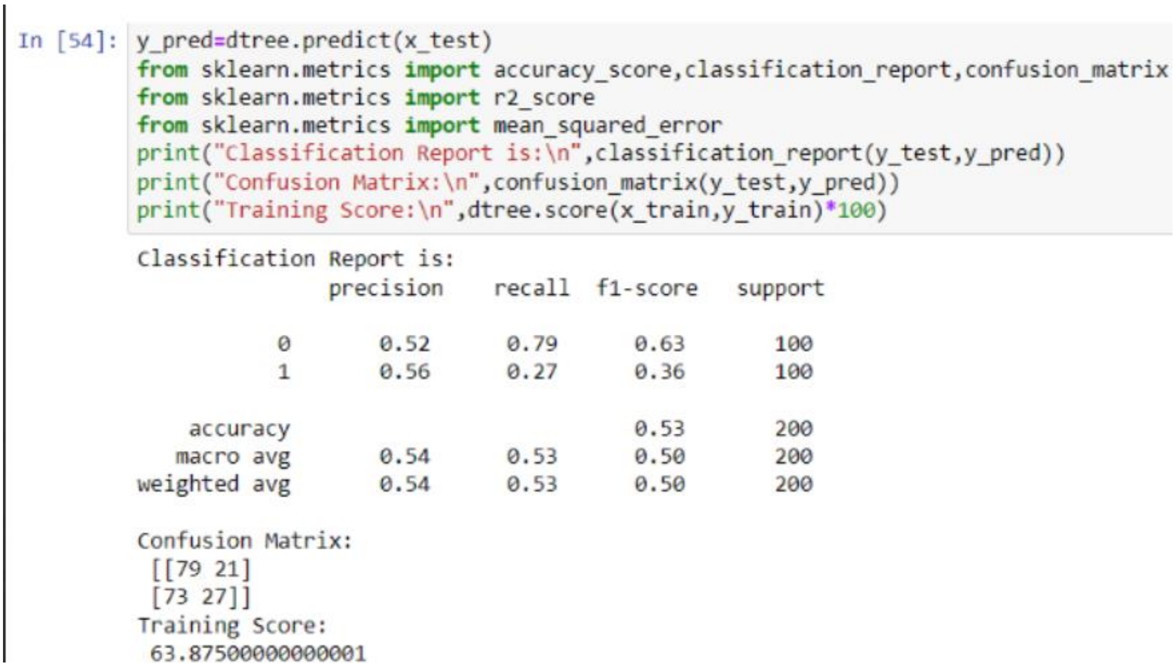
10.3 Naive bayes



```
In [48]: y_pred=gnb.predict(x_test)
         from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
         from sklearn.metrics import r2_score
         from sklearn.metrics import mean_squared_error
         print("Classification Report is:\n",classification_report(y_test,y_pred))
         print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
         print("Training Score:\n",gnb.score(x_train,y_train)*100)

         Classification Report is:
                       precision    recall  f1-score   support

                    0       0.51      0.35      0.41       100
                    1       0.50      0.66      0.57       100

             accuracy                           0.51       200
            macro avg       0.51      0.51      0.49       200
         weighted avg       0.51      0.51      0.49       200

         Confusion Matrix:
          [[35 65]
          [34 66]]
         Training Score:
          55.125
```

Fig 19

10.4 Decision tree Classifier

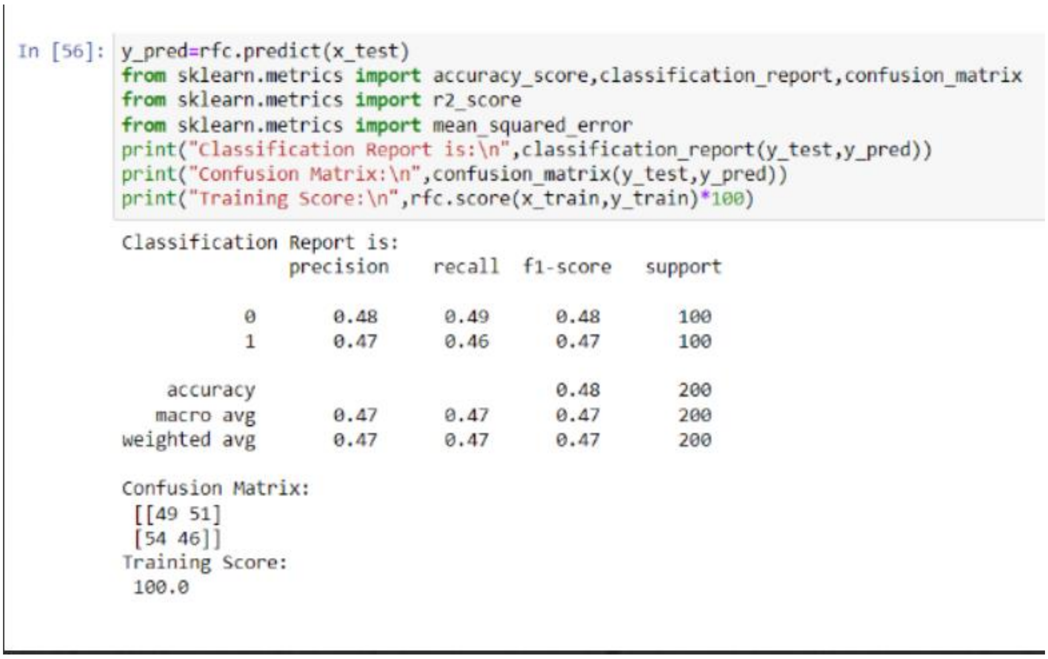


```
In [54]: y_pred=dtree.predict(x_test)
         from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
         from sklearn.metrics import r2_score
         from sklearn.metrics import mean_squared_error
         print("Classification Report is:\n",classification_report(y_test,y_pred))
         print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
         print("Training Score:\n",dtree.score(x_train,y_train)*100)
```

```
Classification Report is:
               precision    recall  f1-score   support

           0       0.52      0.79      0.63       100
           1       0.56      0.27      0.36       100

    accuracy                           0.53       200
   macro avg       0.54      0.53      0.50       200
weighted avg       0.54      0.53      0.50       200

Confusion Matrix:
 [[79 21]
 [73 27]]
Training Score:
 63.87500000000001
```

Fig 20

10.5 Random Forest Classifier

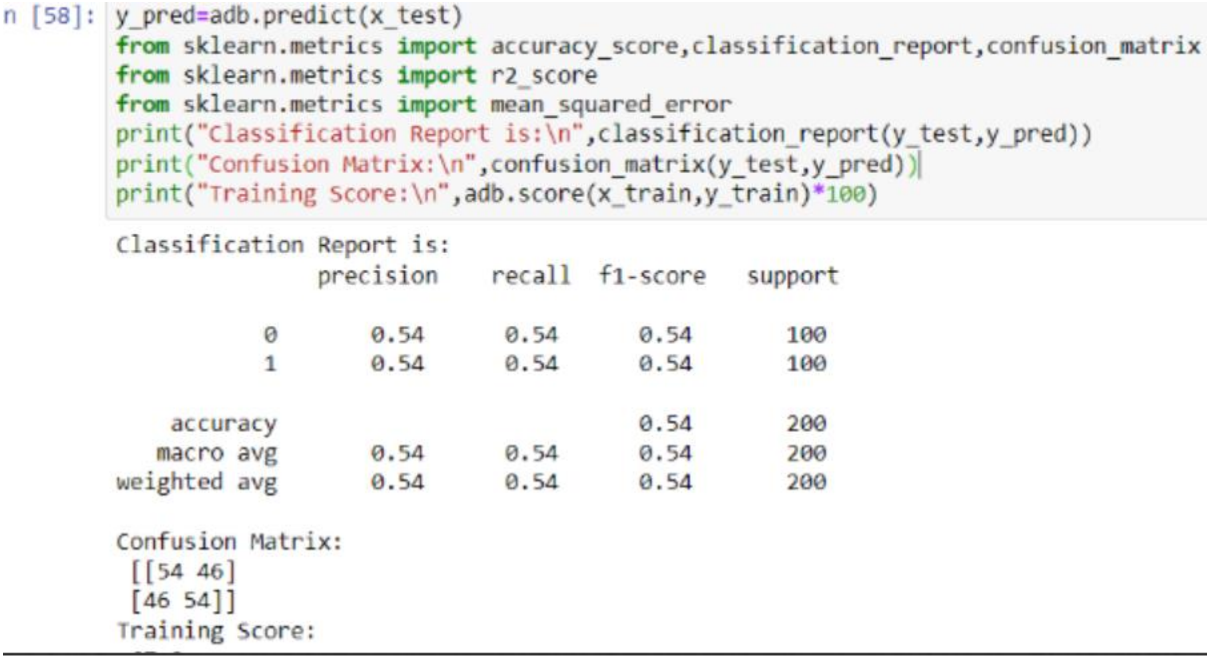


```
In [56]: y_pred=rfc.predict(x_test)
         from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
         from sklearn.metrics import r2_score
         from sklearn.metrics import mean_squared_error
         print("Classification Report is:\n",classification_report(y_test,y_pred))
         print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
         print("Training Score:\n",rfc.score(x_train,y_train)*100)
```

```
Classification Report is:
               precision    recall  f1-score   support

           0       0.48      0.49      0.48       100
           1       0.47      0.46      0.47       100

    accuracy                           0.48       200
   macro avg       0.47      0.47      0.47       200
weighted avg       0.47      0.47      0.47       200

Confusion Matrix:
 [[49 51]
 [54 46]]
Training Score:
 100.0
```

Fig 21

10.6 Ada boost Classifier

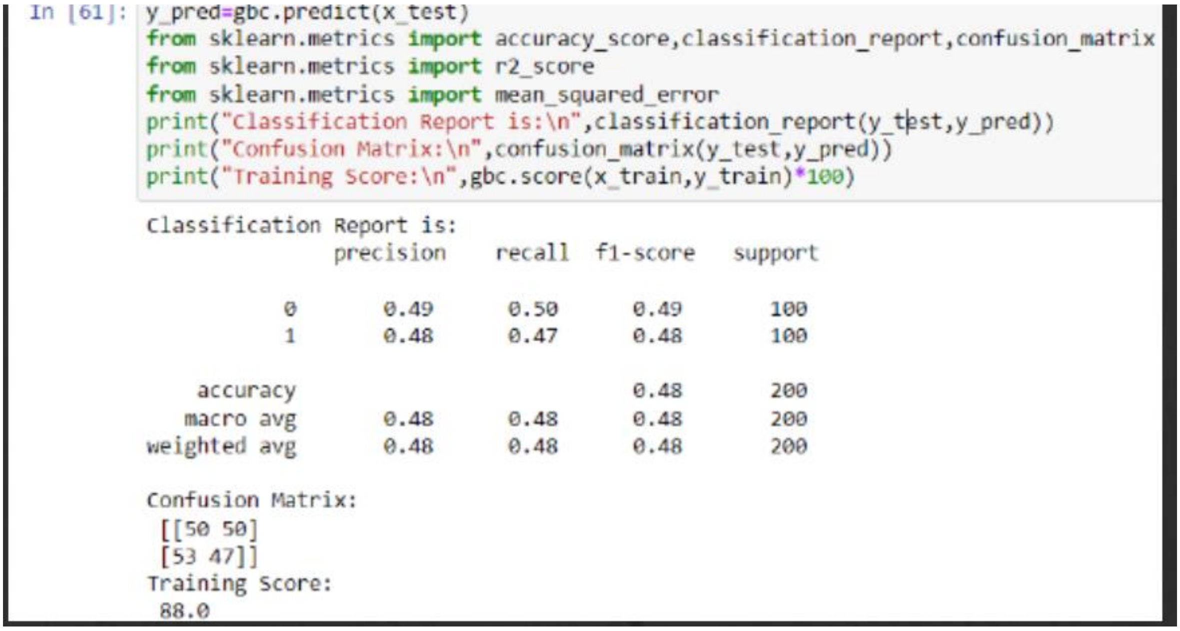


```
n [58]: y_pred=adb.predict(x_test)
        from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
        from sklearn.metrics import r2_score
        from sklearn.metrics import mean_squared_error
        print("Classification Report is:\n",classification_report(y_test,y_pred))
        print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
        print("Training Score:\n",adb.score(x_train,y_train)*100)
```

```
Classification Report is:
              precision    recall  f1-score   support

           0       0.54      0.54      0.54       100
           1       0.54      0.54      0.54       100

    accuracy                           0.54       200
   macro avg       0.54      0.54      0.54       200
weighted avg       0.54      0.54      0.54       200

Confusion Matrix:
 [[54 46]
 [46 54]]
Training Score:
```

Fig 22

10.7 Gradient boosting Classifier

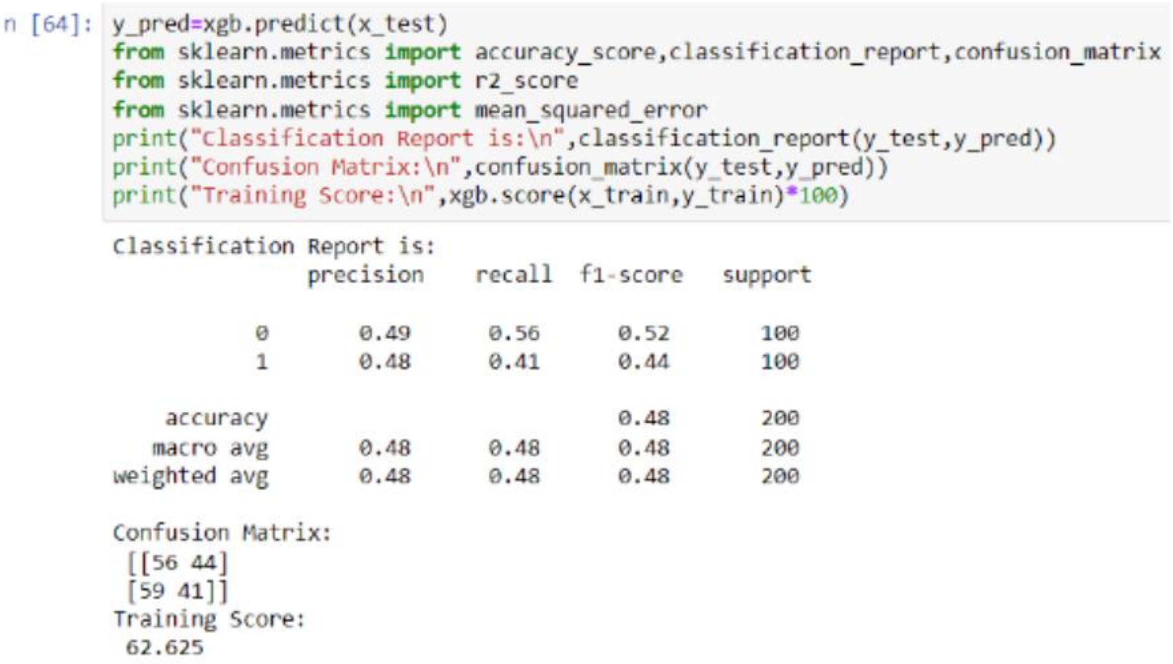


```
In [61]: y_pred=gbc.predict(x_test)
         from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
         from sklearn.metrics import r2_score
         from sklearn.metrics import mean_squared_error
         print("Classification Report is:\n",classification_report(y_test,y_pred))
         print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
         print("Training Score:\n",gbc.score(x_train,y_train)*100)
```

```
Classification Report is:
              precision    recall  f1-score   support

           0       0.49      0.50      0.49       100
           1       0.48      0.47      0.48       100

    accuracy                           0.48       200
   macro avg       0.48      0.48      0.48       200
weighted avg       0.48      0.48      0.48       200

Confusion Matrix:
 [[50 50]
 [53 47]]
Training Score:
 88.0
```

Fig 23

10.8 XGB classifier

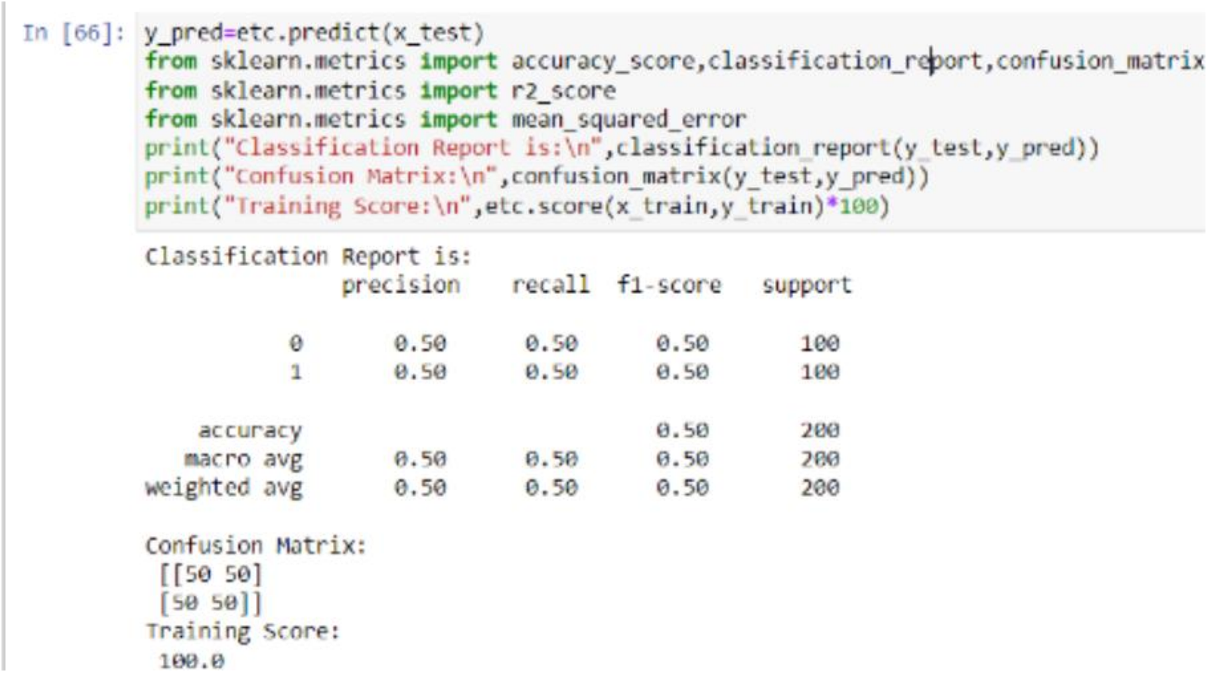


```
n [64]: y_pred=xgb.predict(x_test)
        from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
        from sklearn.metrics import r2_score
        from sklearn.metrics import mean_squared_error
        print("Classification Report is:\n",classification_report(y_test,y_pred))
        print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
        print("Training Score:\n",xgb.score(x_train,y_train)*100)
```

```
Classification Report is:
               precision    recall  f1-score   support

           0       0.49      0.56      0.52       100
           1       0.48      0.41      0.44       100

    accuracy                           0.48       200
   macro avg       0.48      0.48      0.48       200
weighted avg       0.48      0.48      0.48       200

Confusion Matrix:
 [[56 44]
 [59 41]]
Training Score:
 62.625
```

Fig 24

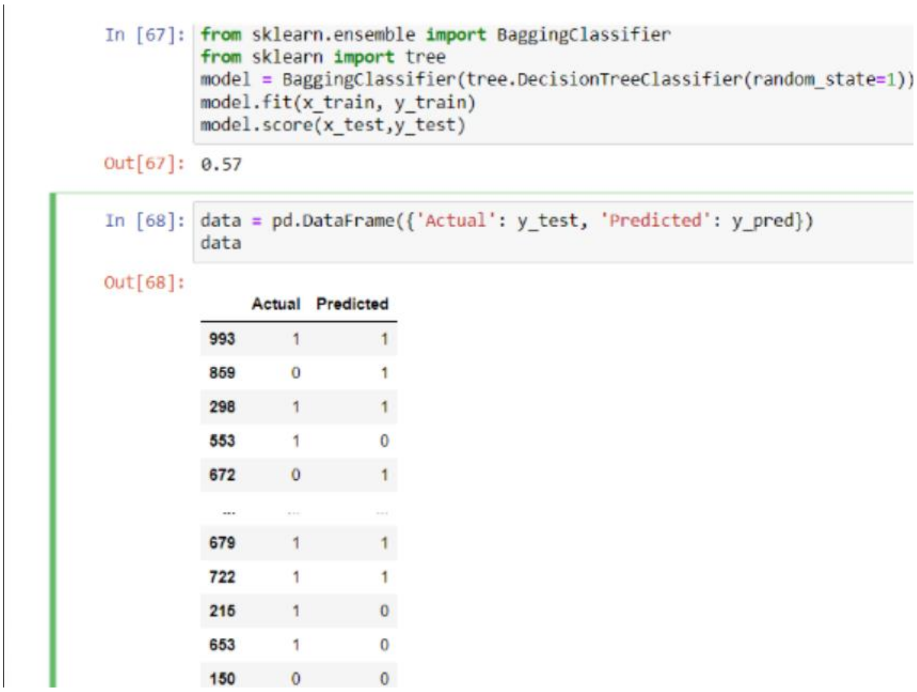
10.9 Extra tree classifier



```
In [66]: y_pred=etc.predict(x_test)
         from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
         from sklearn.metrics import r2_score
         from sklearn.metrics import mean_squared_error
         print("Classification Report is:\n",classification_report(y_test,y_pred))
         print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
         print("Training Score:\n",etc.score(x_train,y_train)*100)

         Classification Report is:
                       precision    recall  f1-score   support

                    0       0.50      0.50      0.50       100
                    1       0.50      0.50      0.50       100

             accuracy                           0.50       200
            macro avg       0.50      0.50      0.50       200
         weighted avg       0.50      0.50      0.50       200

         Confusion Matrix:
          [[50 50]
          [50 50]]
         Training Score:
          100.0
```

Fig 25

10.10 Bagging



```
In [67]: from sklearn.ensemble import BaggingClassifier
         from sklearn import tree
         model = BaggingClassifier(tree.DecisionTreeClassifier(random_state=1))
         model.fit(x_train, y_train)
         model.score(x_test,y_test)

Out[67]: 0.57

In [68]: data = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
         data

Out[68]:
```

	Actual	Predicted
993	1	1
859	0	1
298	1	1
553	1	0
672	0	1
...	...	...
679	1	1
722	1	1
215	1	0
653	1	0
150	0	0

Fig 26

11. Some other data sets we used

1. Summary_of_Neighborhood_Sales_for_Brooklyn
2. sales data-oklahoma-lottary-commission-retailer-ranking-from-oct
3. MVA_vehicle_Sales_Counts_by_Month_for_Calender_year_2001_through
4. Summary_of_Neighborhood_Sales_in_Manhattan_class
5. Summary_of_Neighborhood_Sales_in_Manhattan_home
6. Vg sales(video games)
7. KAG_Conversion_data
8. nyc-rolling-sales
9. walmart-sales-dataset-of-45stores

11.1 Outputs for above used data sets

Summary_of_Neighborhood_Sales_for_Brooklyn

Name of dataset: Summary_of_Neighborhood_Sales_for_Brooklyn

URL: **https://catalog.data.gov/dataset/**

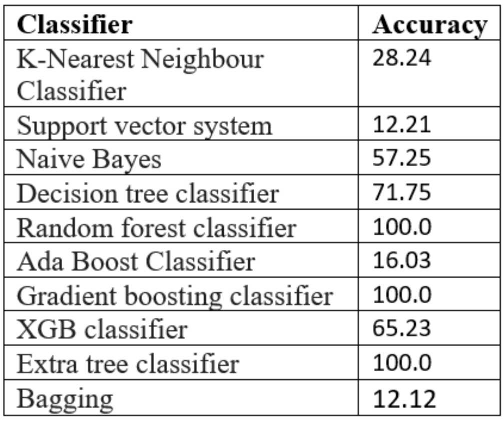
Parameters: NEIGHBORHOOD,TYPE OF HOME,TOTAL NO. OF PROPERTIES,NUMBER OF SALES,LOWEST SALE PRICE,AVERAGE SALE PRICE,MEDIAN SALE PRICE,HIGHEST SALE PRICE

Preprocessing:

1. Gathered data set
2. Uploaded the data set into the jupyter
3. Implemented this data set into the code
4. Getting the result

Codelink: **https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing**

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	28.24
Support vector system	12.21
Naive Bayes	57.25
Decision tree classifier	71.75
Random forest classifier	100.0
Ada Boost Classifier	16.03
Gradient boosting classifier	100.0
XGB classifier	65.23
Extra tree classifier	100.0
Bagging	12.12

Fig 27

sales data-oklahoma-lottary-commission-retailer-ranking-from-oct

Name of dataset: sales data-oklahoma-lottary-commission-retailer-ranking-from-oct

URL: : **https://catalog.data.gov/dataset/**

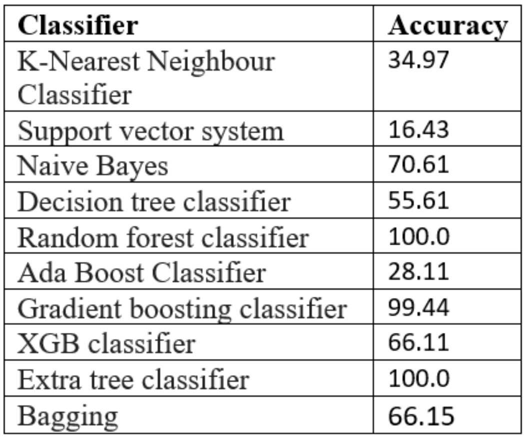
Parameters: Rank, MSR, Retailer, Name, City, Phone, Terminal Type, Weeks Active, Instan t Sales Amt, Online Sales Amt,Total Sales Amt

Preprocessing:

5. Gathered data set
6. Uploaded the data set into the jupyter
7. Implemented this data set into the code
8. Getting the result

Codelink: https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	34.97
Support vector system	16.43
Naive Bayes	70.61
Decision tree classifier	55.61
Random forest classifier	100.0
Ada Boost Classifier	28.11
Gradient boosting classifier	99.44
XGB classifier	66.11
Extra tree classifier	100.0
Bagging	66.15

Fig 28

MVA_vehicle_Sales_Counts_by_Month_for_Calender_year_2001_through

Name of dataset:

MVA_vehicle_Sales_Counts_by_Month_for_Calender_year_2001_through

URL: https://catalog.data.gov/dataset/

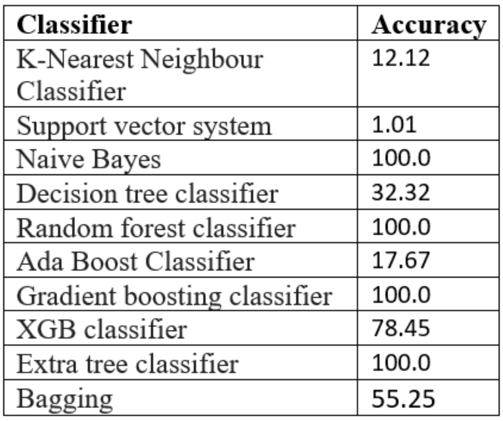
Parameters: Year ,Month ,New,Used,Total Sales New,Total Sales Used

Preprocessing:

1. Gathered data set
2. Uploaded the data set into the jupyter
3. Implemented this data set into the code
4. Getting the result

Codelink: https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	12.12
Support vector system	1.01
Naive Bayes	100.0
Decision tree classifier	32.32
Random forest classifier	100.0
Ada Boost Classifier	17.67
Gradient boosting classifier	100.0
XGB classifier	78.45
Extra tree classifier	100.0
Bagging	55.25

Fig 29

Summary_of_Neighborhood_Sales_in_Manhattan_class

Name of dataset: Summary_of_Neighborhood_Sales_in_Manhattan_class

URL: https://catalog.data.gov/dataset/

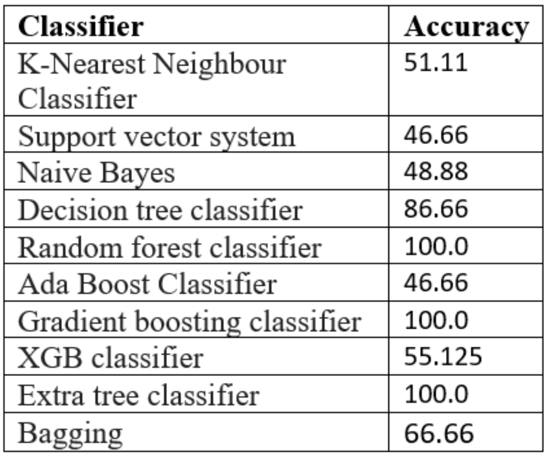
Parameters: NEIGHBORHOOD,TYPE OF HOME,NUMBER OF SALES,LOWEST SALE PRICE AVERAGE, SALE PRICE,MEDIAN SALE PRICE,HIGHEST SALE PRICE

Preprocessing:

1. Gathered data set
2. Uploaded the data set into the jupyter
3. Implemented this data set into the code
4. Getting the result

Codelink: **https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing**

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	51.11
Support vector system	46.66
Naive Bayes	48.88
Decision tree classifier	86.66
Random forest classifier	100.0
Ada Boost Classifier	46.66
Gradient boosting classifier	100.0
XGB classifier	55.125
Extra tree classifier	100.0
Bagging	66.66

Fig 30

Summary_of_Neighborhood_Sales_in_Manhattan_home

Name of dataset: Summary_of_Neighborhood_Sales_in_Manhattan_home

URL: **https://catalog.data.gov/dataset/**

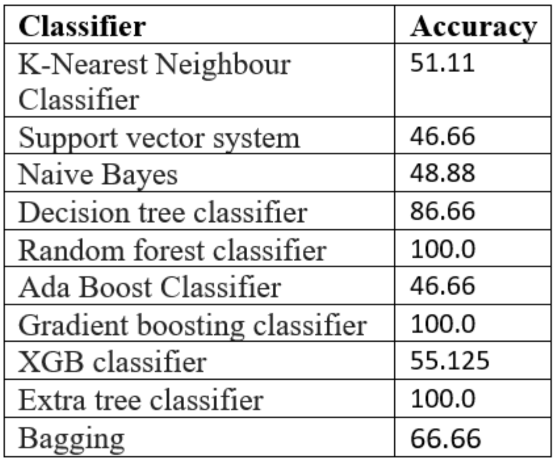
Parameters: NEIGHBORHOOD,TYPE OF HOME,NUMBER OF SALES,LOWEST SALE PRICE AVERAGE, SALE PRICE,MEDIAN SALE PRICE,HIGHEST SALE PRICE

Preprocessing:

1. Gathered data set
2. Uploaded the data set into the jupyter
3. Implemented this data set into the code
4. Getting the result

Codelink: **https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing**

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	51.11
Support vector system	46.66
Naive Bayes	48.88
Decision tree classifier	86.66
Random forest classifier	100.0
Ada Boost Classifier	46.66
Gradient boosting classifier	100.0
XGB classifier	55.125
Extra tree classifier	100.0
Bagging	66.66

Fig 31

Vg sales(video games)

Name of dataset: Vg sales(video games)

URL: **https://www.kaggle.com/datasets/gregorut/videogamesales**

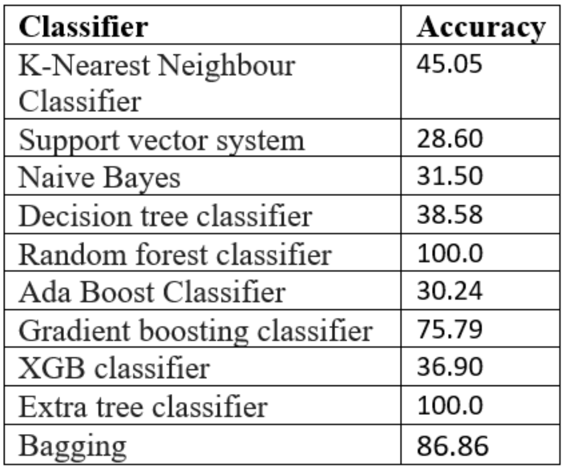
Parameters: Item_Identifier, Item_Weight, Item_Fat_Content, Item_Visibility,Item_Type, I tem_MRP, Outlet_Identifier,Outlet_Establishment_Year, Outlet_Size, Outlet_Location_Type ,Outlet_Type, Item_Outlet_Sales

Preprocessing:

5. Gathered data set
6. Uploaded the data set into the jupyter
7. Implemented this data set into the code
8. Getting the result

Codelink: **https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing**

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	45.05
Support vector system	28.60
Naive Bayes	31.50
Decision tree classifier	38.58
Random forest classifier	100.0
Ada Boost Classifier	30.24
Gradient boosting classifier	75.79
XGB classifier	36.90
Extra tree classifier	100.0
Bagging	86.86

Fig 32

KAG_Conversion_data

Name of dataset: KAG_Conversion_data

URL: https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing

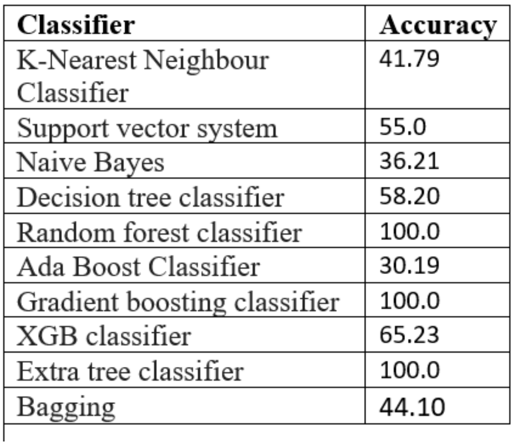
Parameters: ad_id,xyz_campaign_id,fb_campaign_id,age,gender,interest, Impressions,Clicks, Spent, Total_Conversion,Approved_Conversion

Preprocessing:

1. Gathered data set
2. Uploaded the data set into the jupyter
3. Implemented this data set into the code
4. Getting the result

Codelink: https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	41.79
Support vector system	55.0
Naive Bayes	36.21
Decision tree classifier	58.20
Random forest classifier	100.0
Ada Boost Classifier	30.19
Gradient boosting classifier	100.0
XGB classifier	65.23
Extra tree classifier	100.0
Bagging	44.10

Fig 33

nyc-rolling-sales

Name of dataset: nyc-rolling-sales

URL: **https://www.kaggle.com/datasets/new-york-city/nyc-property-sales**

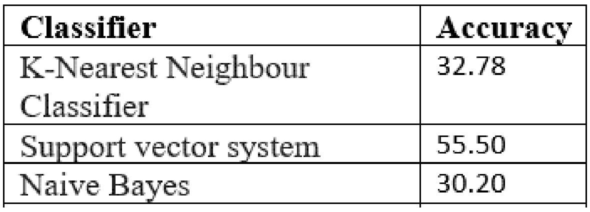
Parameters: BOROUGH, NEIGHBORHOOD, BUILDING CLASS CATEGORY, TAX CLASS AT PRESENT, BLOCK, LOT, EASE-MENT, BUILDING CLASS AT PRESENT, ADDRESS, APARTMENT NUMBER, ZIP CODE,RESIDENTIAL UNITS, COMMERCIAL UNITS, TOTAL UNITS, LAND SQUARE FEET, GROSS SQUARE FEET, YEAR BUILT ,TAX CLASS AT TIME OF SALE, BUILDING CLASS AT TIME OF SALE, SALE PRICE, SALE DATE

Preprocessing:

1. Gathered data set
2. Uploaded the data set into the jupyter
3. Implemented this data set into the code
4. Getting the result

Codelink: **https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing**

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	32.78
Support vector system	55.50
Naive Bayes	30.20

Fig 34

walmart-sales-dataset-of-45stores

Name of dataset: walmart-sales-dataset-of-45stores

URL: https://www.kaggle.com/datasets/varsharam/walmart-sales-dataset-of-45stores

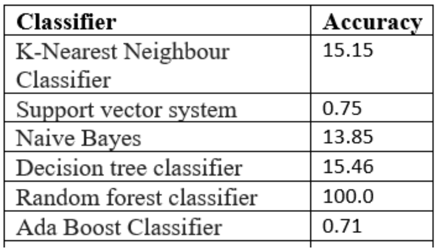
Parameters: Store, Date, Weekly_Sales, Holiday_Flag, Temperature, Fuel_Price, CPI, Unemployment

Preprocessing:

1. Gathered data set
2. Uploaded the data set into the jupyter
3. Implemented this data set into the code
4. Getting the result

Codelink: https://drive.google.com/drive/folders/1Mi3Bghn_admGhAfbMmQ6pMgW-tHSqg9u?usp=sharing

Over all accuracy for all the classifiers we used:



Classifier	Accuracy
K-Nearest Neighbour Classifier	15.15
Support vector system	0.75
Naive Bayes	13.85
Decision tree classifier	15.46
Random forest classifier	100.0
Ada Boost Classifier	0.71

Fig 35

Overall result for all data set we have included in our data sets:

Dataset	Algorithm1	Algorithm2	Algorithm3	Algorithm4	Algorithm5	Algorithm6	Algorithm7	Algorithm8	Algorithm 9	Algorithm10
Summary_of_Neighborhood_Sales_for_Brooklyn	28.24	12.21	57.25	71.75	100.0	16.03	100.0	65.23	100.0	12.12
sales data-oklahoma-lottary-commission-retailer-ranking-from-oct	34.97	16.43	70.61	55.61	100.0	28.11	99.44	66.11	100.0	63.15
MVA_vehicle_Sales_Counts_by_Month_for_Calender_year_2001_through	12.12	1.01	100.0	32.32	100.0	17.67	100.0	78.45	100.0	55.25
Summary_of_Neighborhood_Sales_in_M	51.11	46.66	48.88	86.66	100.0	46.66	100.0	55.125	100.0	66.66

anhattan_class										
Summary_of_Neighborhood_Sales_in_Manhattan_home	51.11	46.66	48.88	86.66	100.0	46.66	100.0	55.125	100.0	66.66
Vg sales(video games)	45.05	28.60	31.50	38.58	100.0	30.24	75.79	36.90	100.0	86.86
KAG_Conversion_data	41.79	55.0	36.21	58.20	100.0	30.19	100.0	65.23	100.0	44.10
nyc-rolling-sales	32.78	55.50	30.20	-	-	-	-	-	-	-
walmart-sales-dataset-of-45stores	15.15	0.75	13.85	15.46	100.0	0.71	-	-	-	-

Fig 36

12.CONCLUSION

In this study we used some techniques for sales prediction such as models and XG Boost algorithms which get better efficiency manipulate the trending sales analysis. Random Forest gives the best accuracy value. Data mining techniques like Linear Regression, Random Forest Regression and XGBoost had been carried out and the outcomes compared. XGBoost which is an expanded gradient boosting algorithm was once found to function the excellent at prediction.

We can conclude it by saying Sales forecasting is very crucial for every company, especially big ones and this process is very complex because there are lots

of factors that should be taken into consideration. In order to implement achievable goals and successfully implement them, supermarkets chains always want to forecast sales. In this study, we used ten machine learning algorithms for sales forecasting, Random Forest and extra tree performed better,we also observed with other data sets the model is working fine.so by this model people can predict the sales and improve the sale accordingly.

13.FUTURE SCOPE

We look forward to use more parameters in our datasets and improve the model so that it predicts good and at the same time get a high accuracy. We would also like to enhance the system by other data sets with less parameters so that it can benefit small merchants as well

It will be interesting to continue what we have started in this work by collecting more information on sales prediction and continue to classify for more

Also, it is reasonable to try this on various other domains so that it will be benefited for other people as well. It will be advantageous for uneducated people if we develop image recognition in it.

we did not get accuracy values for some classifiers we will make sure we will done this for our next project .

14.REFERENCES

https://www.academia.edu/46804506/Supermarket_Sales_Prediction_Using_Regression

https://www.analyticsvidhya.com/blog/2020/08/building-sales-prediction-web-application-using-machine-learning-dataset/

https://www.researchgate.net/publication/344099746_SALES_PREDICTION_MODEL_FOR_BIG_MART