Daffodil International University

PROJECT REPORT

COUSE CODE: CSE322

COURSE TITLE: DATA MINING AND MACHINE LEARNING LAB

SUBMITTED TO:

FARIHA JAHAN

Lecturer, Department OF Computer Science & Engineering, DAFFODIL INTERNATIONAL UNIVERSITY.

SUBMITTED BY:

SHAZID NAWAS SHOVON (191-15-12929)

SAKIB ROKONI (191-15-12961)

MD. TARIQUL ISLAM (191-15-12963)

SECTION: N

DEPARTMENT OF CSE,

DAFFODIL INTERNATIONAL UNIVERSITY.

Breast-Cancer-Wisconsin

Abstract: Classification is a type of supervised learning. Classifier prediction can assist us in resolving real-world problems by employing various classification techniques. In this experiment, we ran five algorithms side by side to determine which one was the best. Machine learning techniques were applied using the Colaboratory.

Keyword: Breast-cancer, classifier, SVM, K-Nearest Neighbors, Logistic Regression, Naïve Bayes and Random Forest

I. INTRODUCTION

In this paper, we have taken a dataset regarding Breast-Cancer-Wisconsin. This is a small dataset of 570 instances and 31 attributes. Each student represents a human and attributes are representing patients ID, Diagnosis, radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension. Predicting the Diagnosis of Breast-Cancer-Wisconsin from physical measurement. We applied 5 types of classifiers. Those classifiers are,

1. SVM,
2. K-Nearest Neighbors,
3. Logistic Regression,
4. Naïve Bayes and
5. Random Forest

II. LITERATURE REVIEW

Our dataset is downloaded from UCI Machine Learning Repository. This dataset is a Breast-Cancer-Wisconsin database similar to a database already present in the repository (Breast-Cancer-Wisconsin) but in a slightly different form.

Number of Instances: 570

Number of attributes: 30

Missing values: No

Some Attribute Information:

1. ID
2. Diagnosis
3. Radius
4. Texture
5. Perimeter
6. Area
7. Smoothness
8. Compactness
9. Concavity
10. Concave Points
11. Symmetry
12. Fractal dimension

III. Methodology:

In this section we will discuss our working procedure for this project , we divide it into 4 major parts.

a) Preparing the tools:

As this project has done using the python programming language, working with the python requires some libraries to be imported. That's why before starting the actual working procedure we have imported necessary libraries or tools.

They are given below :

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from sklearn import metrics

from sklearn.metrics import confusion_matrix, make_scorer
```

from sklearn.model_selection import cross_validate

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import roc_auc_score, accuracy_score

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive_bayes import GaussianNB

import warnings

warnings.filterwarnings("ignore")

b) Preprocessing:

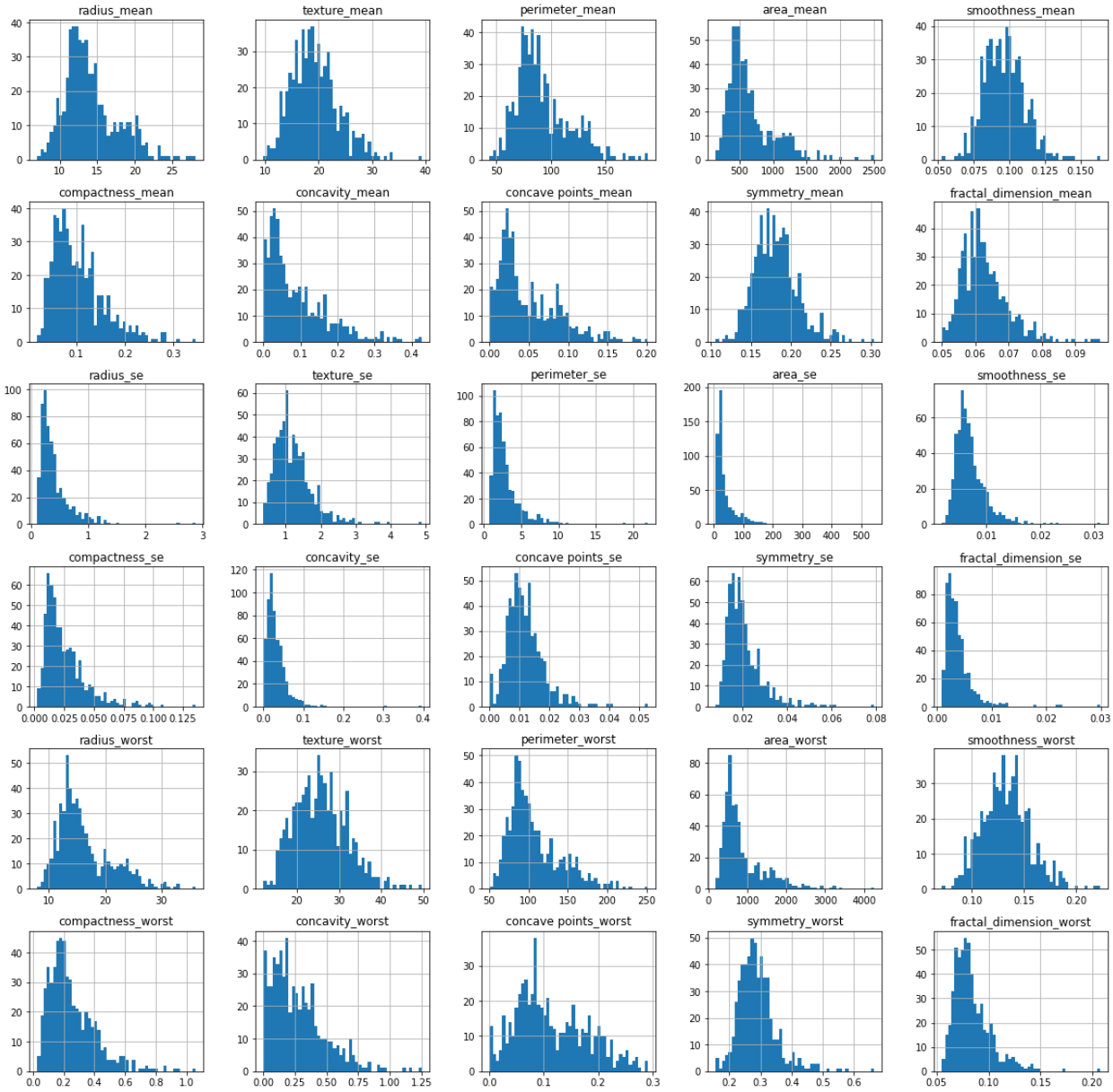
When we had downloaded the data , it was in .dat format. Different lines represent attributes of different instances and attributes of a single instance are separated by spaces. At first we convert this .dat file to a csv file.Then we have imported the csv on our python project file using the

df = pd.read_csv("/content/drive/MyDrive/DMML PROJECT/data.csv")

c) Visualizing the dataset attributes:

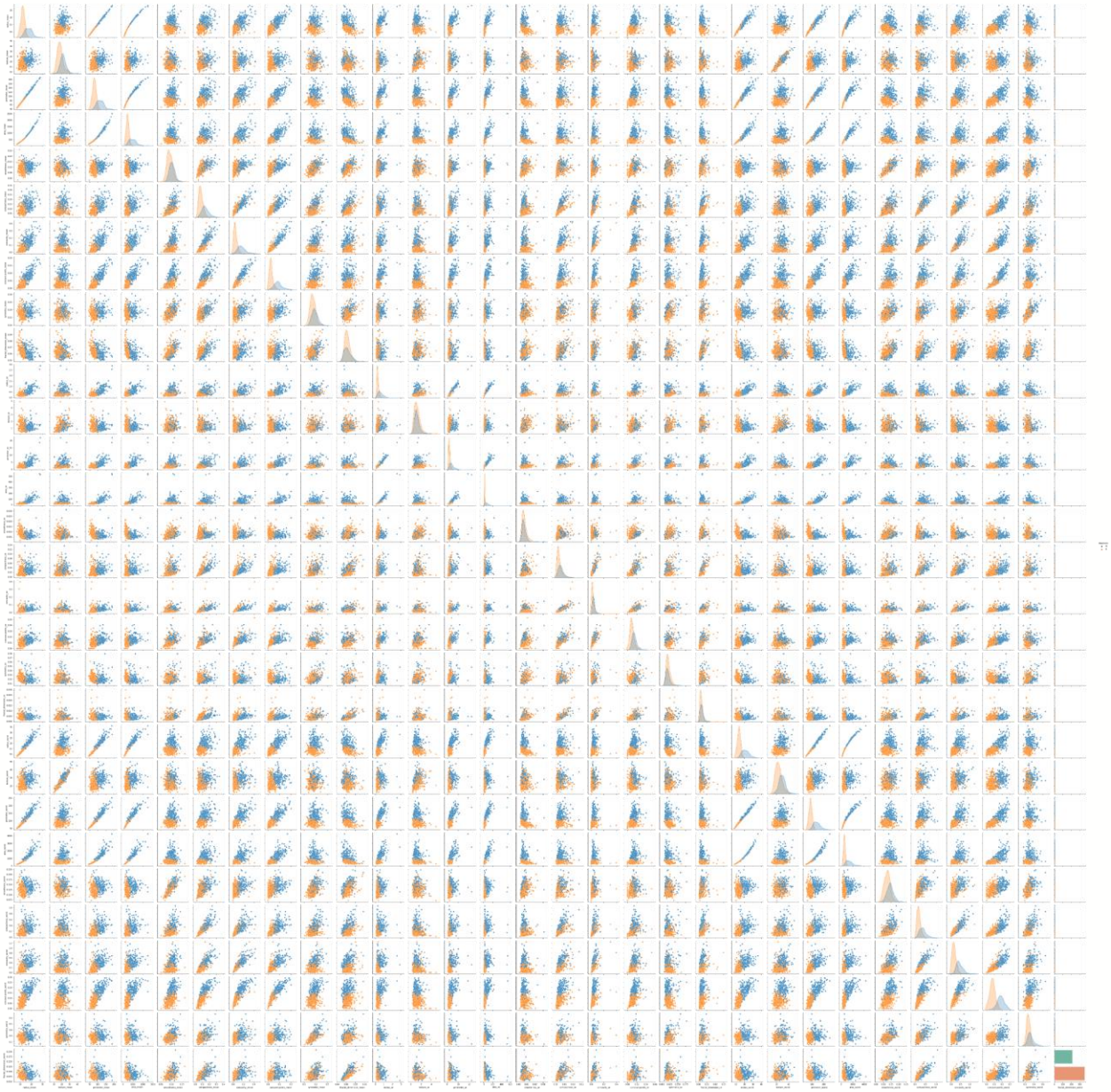
Visualizing the attributes is very important to gather knowledge from the given dataset. There are 31 columns or attributes or variables.

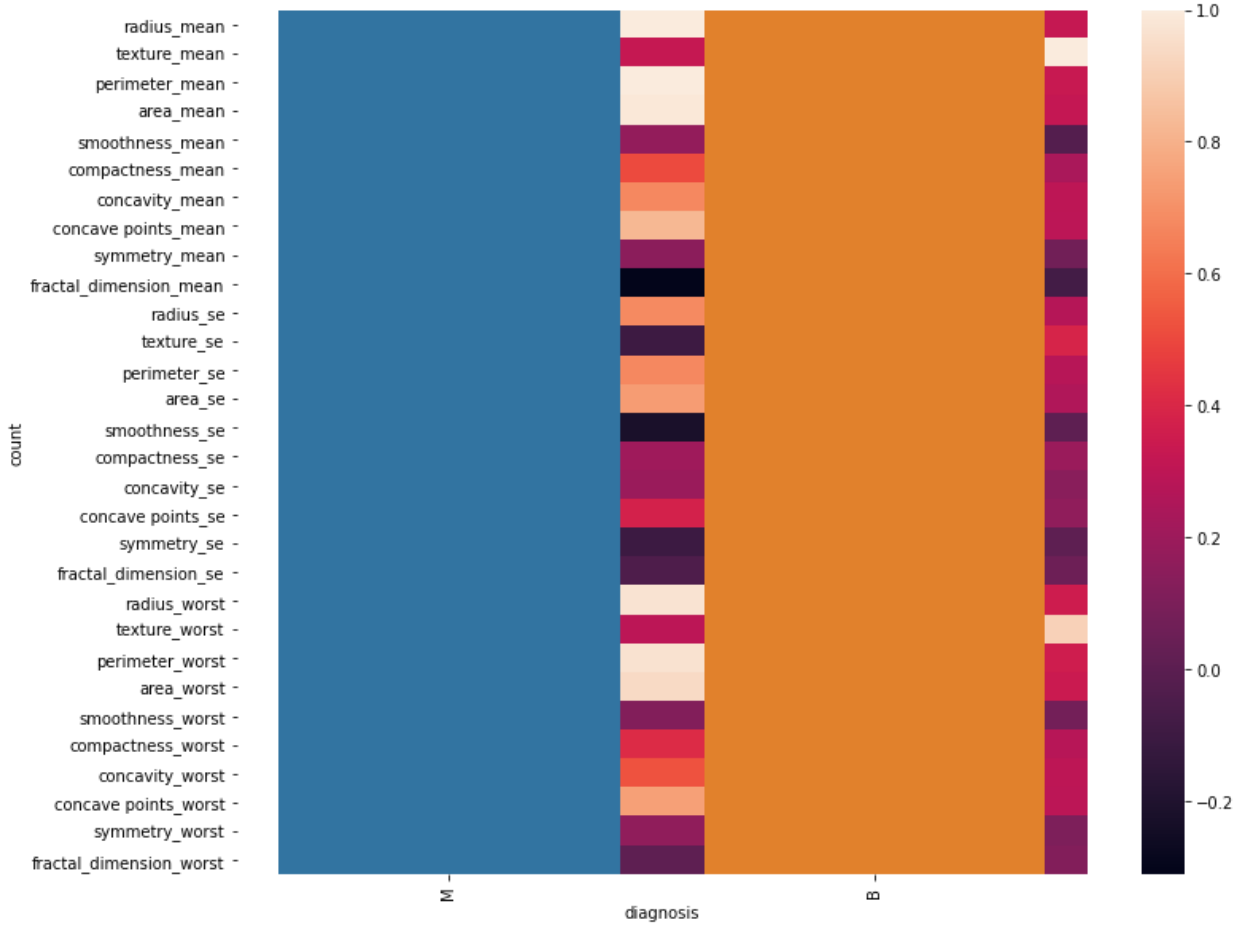
```
# Data visualization to create histogram

df.hist(bins=50, figsize=(20, 20))

plt.show()
```



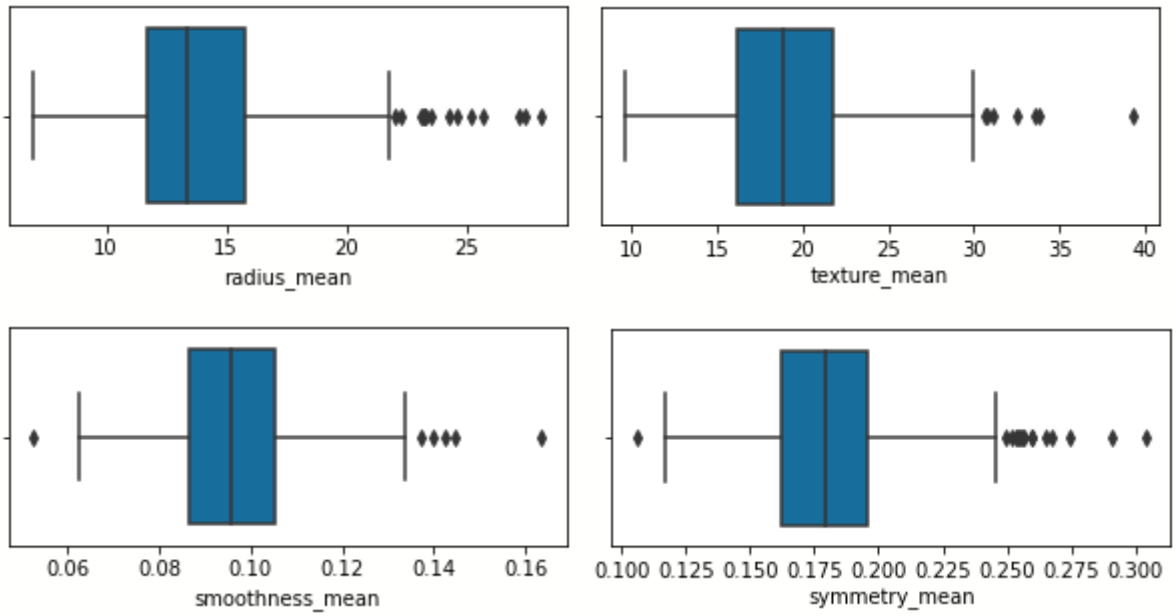

```
# Scatter matrix to check correlation between two attributes

sns.pairplot(df, hue='diagnosis')

# Count each label

ax = sns.countplot(y='diagnosis', data=df, palette='Set2')

# Finding correlation
```

```
fig, ax = plt.subplots(figsize=(12, 10))

sns.heatmap(df.corr(), ax=ax)

print(df.groupby('diagnosis').size())

sns.countplot(df['diagnosis'],label="Count")

plt.show()
```



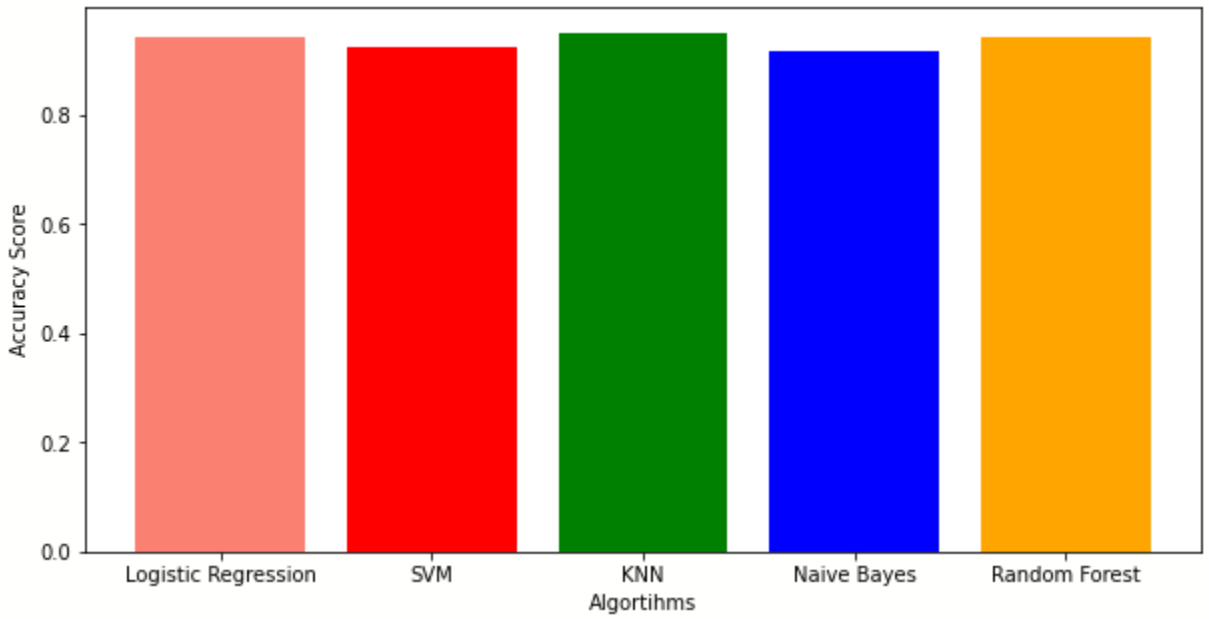



```
# box plot to check outlier in each category

def boxPlot(dff):

    d = dff.drop(columns=['diagnosis'])

    for column in d:

        plt.figure(figsize=(5, 2))

        sns.boxplot(x=column, data=d, palette="colorblind")

boxPlot(df)
```

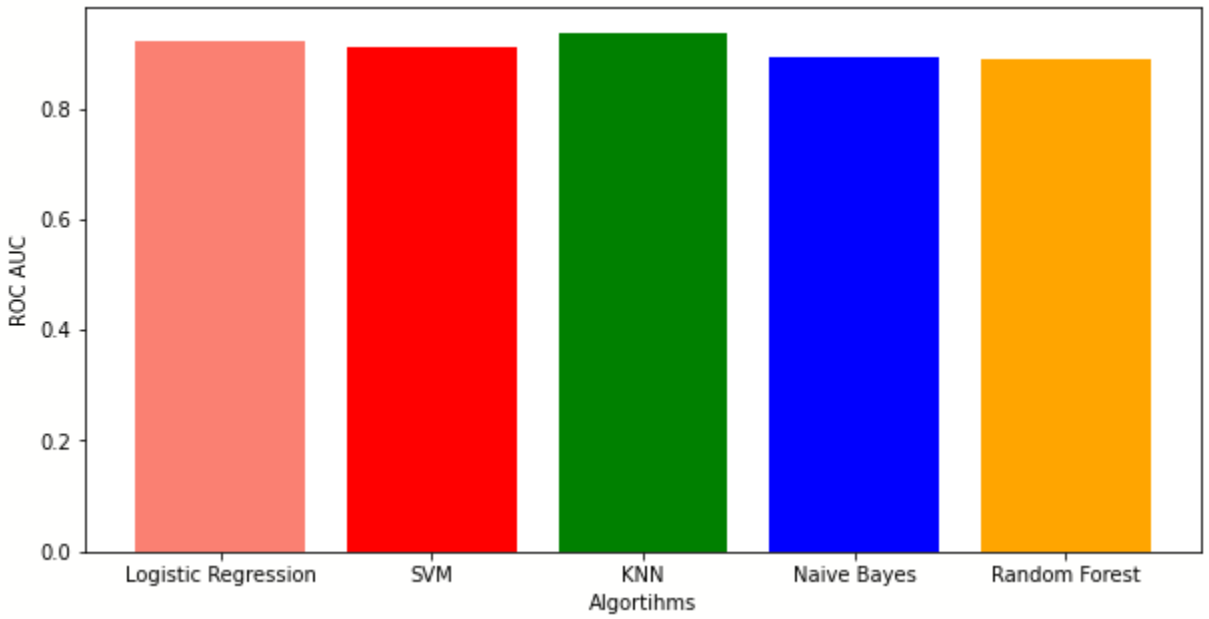


IV. COMPARATIVE ANALYSIS

```
# Plot the bar graph for accuracy and  roc-auc

# accuracy score
plt.figure(figsize=(10, 5))
plt.bar(['Logistic Regression', 'SVM', 'KNN', 'Naive Bayes', 'Random
Forest'], acc,
        color=['salmon', 'r', 'g', 'b', 'orange'], label='Accuracy')
plt.ylabel('Accuracy Score')
plt.xlabel('Algortihms')
```




```
# roc-auc
plt.figure(figsize=(10, 5))
plt.bar(['Logistic Regression', 'SVM', 'KNN', 'Naive Bayes', 'Random
Forest'], roc,
        color=['salmon', 'r', 'g', 'b', 'orange'], label='ROC AUC')
plt.ylabel('ROC AUC')
plt.xlabel('Algortihms')
plt.show()
```



Based on the above model evaluation process we have gathered the results of the 5 classifiers.

Naive Bayes :

Training Set Accuracy : 0.9532374100719424

Test Set Accuracy 0.9416666666666667 ROC 0.9204545454545454

Accuracy: [0.96 0.96 0.89 0.93 1. 0.93 0.93 0.93 0.96 1.]

Logistic Regression :

Training Set Accuracy : 0.9532374100719424

Test Set Accuracy 0.925 ROC 0.9090909090909092

Accuracy: [0.86 0.89 0.89 0.96 0.96 0.93 0.96 1. 0.96 0.96]

Support Vector Machine :

Training Set Accuracy : 0.9820143884892086

Test Set Accuracy 0.95 ROC 0.9360795454545455

Accuracy: [0.93 1. 1. 0.96 1. 0.96 1. 1. 0.93 1.]

K-Nearest Neighbors :

Training Set Accuracy : 0.9424460431654677

Accuracy 0.9166666666666666 ROC 0.8934659090909091

Accuracy: [0.93 0.89 0.93 0.96 0.93 0.89 1. 0.96 0.96 0.96]

Random Forest :

Training Set Accuracy : 0.9964028776978417

Accuracy 0.9416666666666667 ROC 0.890625

Accuracy: [1. 0.93 0.93 0.93 0.93 0.89 0.89 0.93 0.96 0.96]

V. RESULTS AND DISCUSSION

The result we got from the 5 classifiers . According to the above results, the accuracy obtained by the K-Nearest Neighbors classifier is the highest of any classifier. From the obtained result we can conclude that Naive Bayes classifier is best for our dataset

K-Nearest Neighbors :

Training Set Accuracy : 0.9424460431654677

Accuracy 0.9166666666666666 ROC 0.8934659090909091

Accuracy: [0.93 0.89 0.93 0.96 0.93 0.89 1. 0.96 0.96 0.96]

Project Link:

https://colab.research.google.com/drive/1l6N70gLOgmGIY5FcexZXHkjTd1Ju0Ifv?usp=sharing