Machine Learning Task

Eng:-Ahmed Osman

احمد محمد عبدالمعطى محمد	سكشن:-1
بيتر رزق الفونس رزق	سكشن:-1
محمد سعيد عبدالله السيد	سكشن:- 3
محمد احمد فكري ابراهيم	سكشن:- 3
محمود خالد عمر عبدالعزيز	سكشن:-4

Abstract

This is a report about a machine learning model for classifying breast cancer cases and predicting whether the cancer is benign or malignant.

Introduction & Dataset Overview

The Task is to predict whether the breast cancer case is benign or malignant, the dataset used is **Breast Cancer Wisconsin (Diagnostic)** dataset.

The dataset contains 569 cases where 357 of them are benign and the remaining 212 are malignant.

The dataset has the following columns:

1) ID number
2) Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

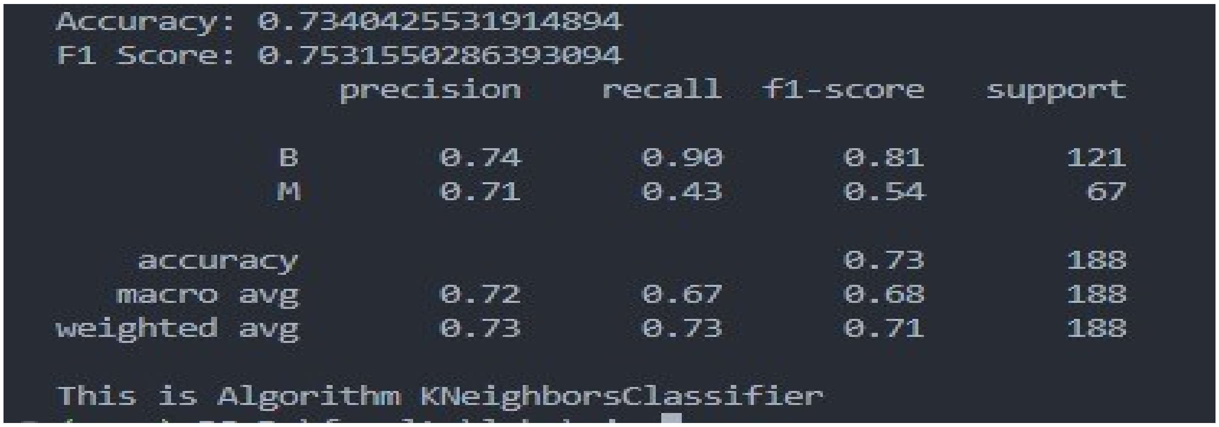
All feature values are recorded with four significant digits.

- ## Total code of 3 algorithm:-

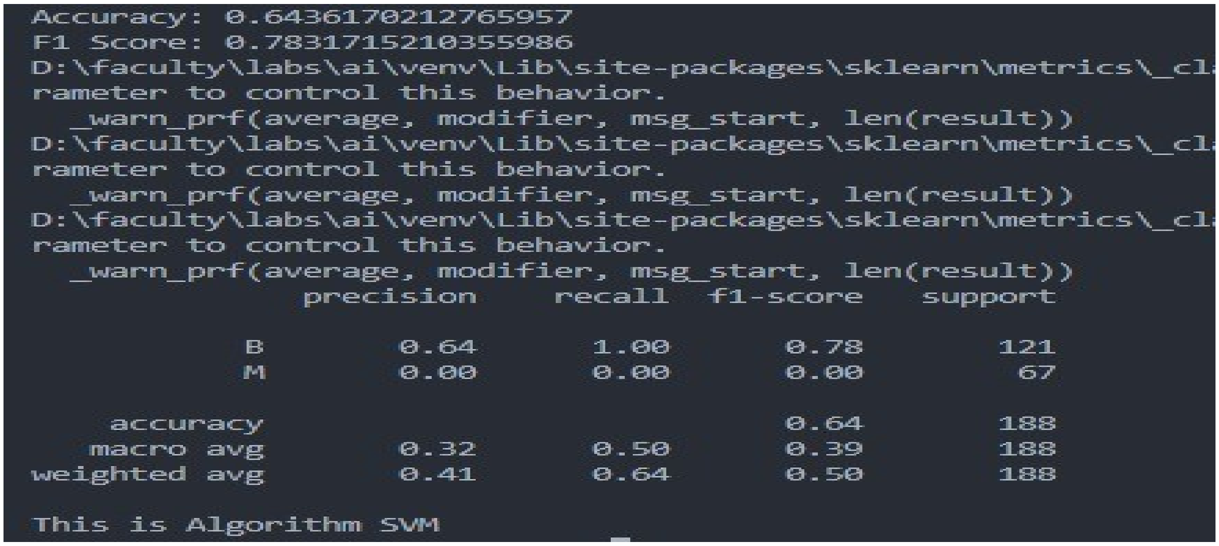
```
# Naive Base Classifier SVC KNN
import pandas as pd   # to load data
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import (
    accuracy_score,
    confusion_matrix,
    ConfusionMatrixDisplay,
    f1_score,
    classification_report
)
from sklearn.impute import SimpleImputer
from sklearn.pipeline import make_pipeline
df = pd.read_csv('./data.csv')
df.head()
df.info()
sns.countplot(data=df,x='diagnosis',hue='radius_mean')
plt.xticks(rotation=45, ha='right')
# Dropping the 'Unnamed: 32' column
# pre_df = df.drop('Unnamed: 32', axis=1)
# Separating features and target variable
# X = df.drop('diagnosis', axis=1)  # Assuming you want to drop the 'diagnosis'
column
X = df[['radius_mean', 'texture_mean', 'perimeter_mean', 'smoothness_mean',
'compactness_mean','compactness_mean','concavity_mean','concave
points_mean','symmetry_mean','fractal_dimension_mean','radius_se','texture_
se','perimeter_se','area_se','smoothness_se','compactness_se','concavity_se','
concave
points_se','symmetry_se','fractal_dimension_se','radius_worst','texture_wors
t','perimeter_worst','area_worst','smoothness_worst','compactness_worst','co
ncavity_worst','concave
points_worst','symmetry_worst','fractal_dimension_worst']]
y = df['diagnosis']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42
)
# Fit the model on the transformed datam
```

```
model = GaussianNB()
model = SVC()
model = KNeighborsClassifier()
imputer = SimpleImputer(strategy='mean')
# Create a pipeline with the imputer and Gaussian Naive Bayes classifier
pipeline = make_pipeline(imputer, GaussianNB())
pipeline.fit(X_train, y_train)
model.fit(X_train, y_train)
# Predicting the target variable using the test data
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_pred, y_test)
f1 = f1_score(y_pred, y_test, average='weighted')
print('Accuracy', accuracy)
print("F1 Score", f1)
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
cmd = ConfusionMatrixDisplay(cm, display_labels=model.classes_)
cmd.plot()
Classification Report
print(classification_report(y_test, y_pred))
plt.show()
# Assuming you have a new_data variable containing the new data point
new_data = [[14.2, 20.5, 94.7, 0.102, 0.145, 0.186, 0.104, 0.172, 0.175, 0.055, 0.543, 0.476, 3.07, 35.2, 0.008, 0.036, 0.031, 0.011, 0.024, 0.004, 15.3, 24.2, 102.5, 0.163, 0.369, 0.508, 0.242, 0.382, 0.091, 0.03]]
# Use the predict method on your model
# print(X.columns)
output = model.predict(new_data)
# Print the output
print("Predicted Output:", output)
```

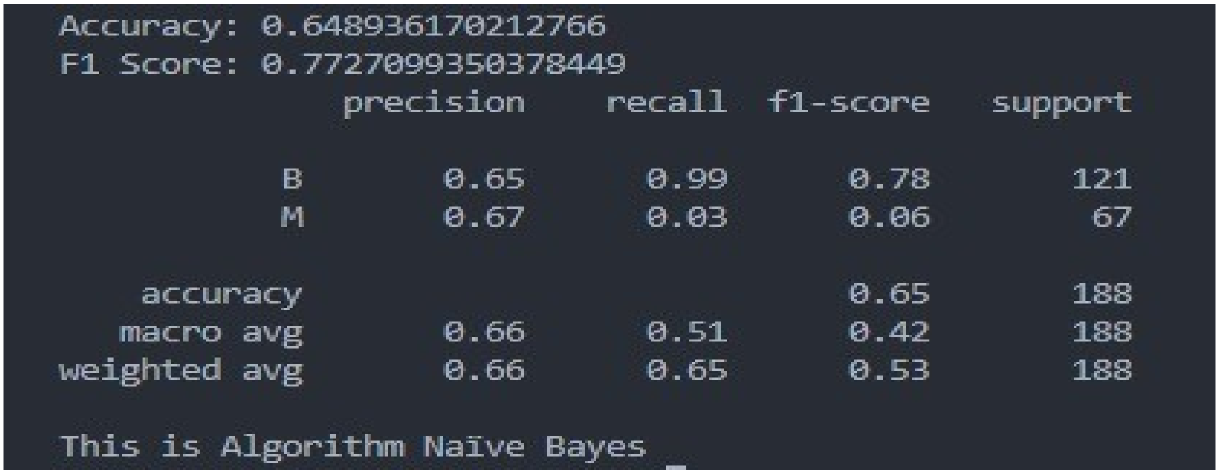
- **Total output of 3 algorithm:-**



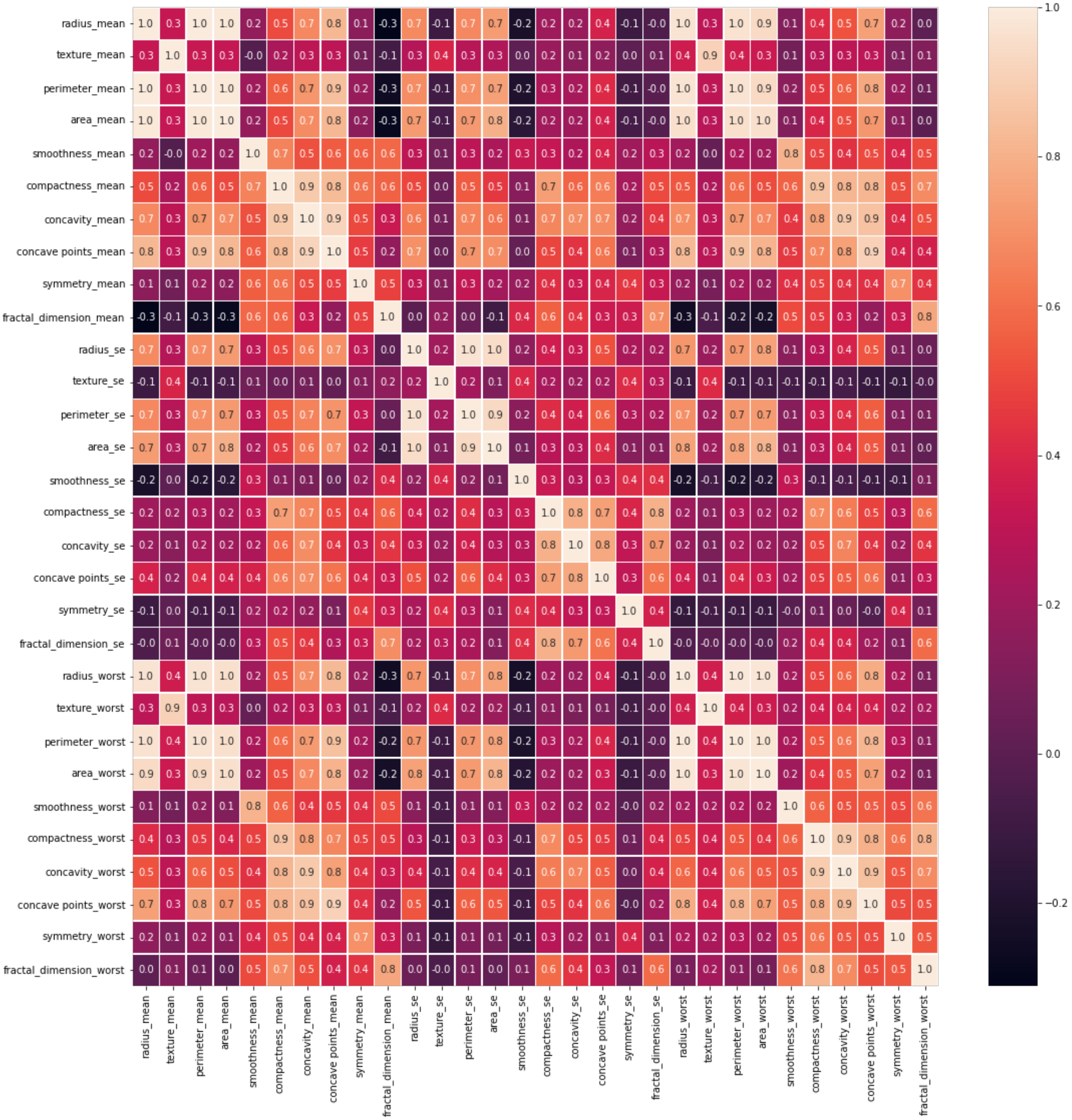
```
Accuracy: 0.7340425531914894
F1 Score: 0.7531550286393094
              precision    recall  f1-score   support

           B       0.74      0.90      0.81       121
           M       0.71      0.43      0.54        67

    accuracy                           0.73       188
   macro avg       0.72      0.67      0.68       188
weighted avg       0.73      0.73      0.71       188

This is Algorithm KNeighborsClassifier
```



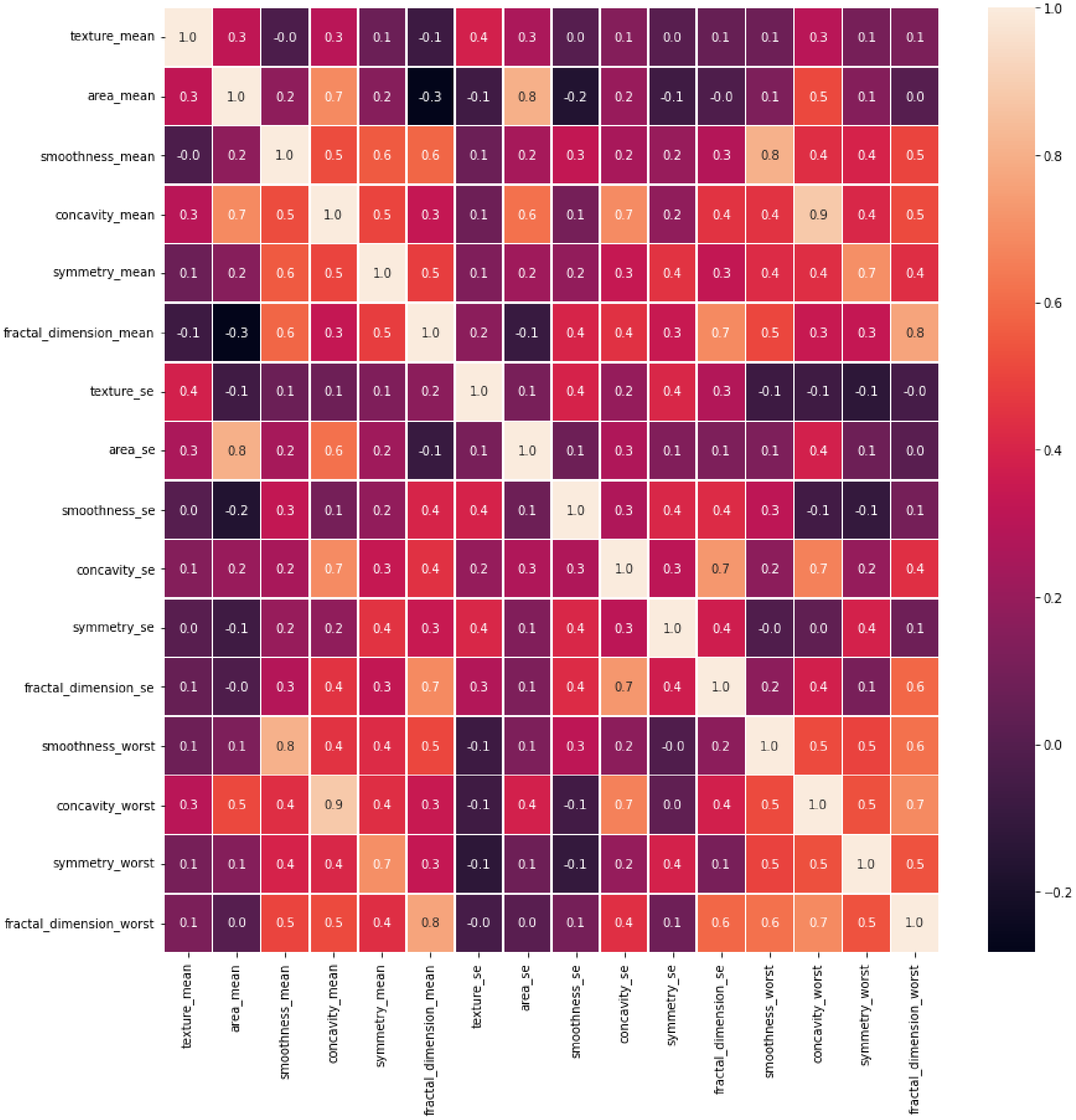
```
Accuracy: 0.6436170212765957
F1 Score: 0.7831715210355986
D:\faculty\labs\ai\venv\Lib\site-packages\sklearn\metrics\_cla
rameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
D:\faculty\labs\ai\venv\Lib\site-packages\sklearn\metrics\_cla
rameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
D:\faculty\labs\ai\venv\Lib\site-packages\sklearn\metrics\_cla
rameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           B       0.64      1.00      0.78       121
           M       0.00      0.00      0.00        67

    accuracy                           0.64       188
   macro avg       0.32      0.50      0.39       188
weighted avg       0.41      0.64      0.50       188

This is Algorithm SVM
```



```
Accuracy: 0.648936170212766
F1 Score: 0.7727099350378449
              precision    recall  f1-score   support

           B       0.65      0.99      0.78       121
           M       0.67      0.03      0.06        67

    accuracy                           0.65       188
   macro avg       0.66      0.51      0.42       188
weighted avg       0.66      0.65      0.53       188

This is Algorithm Naïve Bayes
```


radius_mean 1.0 0.3 1.0 1.0 0.2 0.5 0.7 0.8 0.1 -0.3 0.7 -0.1 0.7 0.7 -0.2 0.2 0.2 0.4 -0.1 -0.0 1.0 0.3 1.0 0.9 0.1 0.4 0.5 0.7 0.2 0.0
texture_mean 0.3 1.0 0.3 0.3 -0.0 0.2 0.3 0.3 0.1 -0.1 0.3 0.4 0.3 0.3 0.0 0.2 0.1 0.2 0.0 0.1 0.4 0.9 0.4 0.3 0.1 0.3 0.3 0.3 0.1 0.1
perimeter_mean 1.0 0.3 1.0 1.0 0.2 0.6 0.7 0.9 0.2 -0.3 0.7 -0.1 0.7 0.7 -0.2 0.3 0.2 0.4 -0.1 -0.0 1.0 0.3 1.0 0.9 0.2 0.5 0.6 0.8 0.2 0.1
area_mean 1.0 0.3 1.0 1.0 0.2 0.5 0.7 0.8 0.2 -0.3 0.7 -0.1 0.7 0.8 -0.2 0.2 0.2 0.4 -0.1 -0.0 1.0 0.3 1.0 1.0 0.1 0.4 0.5 0.7 0.1 0.0
smoothness_mean 0.2 -0.0 0.2 0.2 1.0 0.7 0.5 0.6 0.6 0.6 0.3 0.1 0.3 0.2 0.3 0.3 0.2 0.4 0.2 0.3 0.2 0.0 0.2 0.2 0.8 0.5 0.4 0.5 0.4 0.5
compactness_mean 0.5 0.2 0.6 0.5 0.7 1.0 0.9 0.8 0.6 0.6 0.5 0.0 0.5 0.5 0.1 0.7 0.6 0.6 0.2 0.5 0.5 0.2 0.6 0.5 0.6 0.9 0.8 0.8 0.5 0.7
concavity_mean 0.7 0.3 0.7 0.7 0.5 0.9 1.0 0.9 0.5 0.3 0.6 0.1 0.7 0.6 0.1 0.7 0.7 0.7 0.2 0.4 0.7 0.3 0.7 0.7 0.4 0.8 0.9 0.9 0.4 0.5
concave points_mean 0.8 0.3 0.9 0.8 0.6 0.8 0.9 1.0 0.5 0.2 0.7 0.0 0.7 0.7 0.0 0.5 0.4 0.6 0.1 0.3 0.8 0.3 0.9 0.8 0.5 0.7 0.8 0.9 0.4 0.4
symmetry_mean 0.1 0.1 0.2 0.2 0.6 0.6 0.5 0.5 1.0 0.5 0.3 0.1 0.3 0.2 0.2 0.4 0.3 0.4 0.4 0.3 0.2 0.1 0.2 0.2 0.4 0.5 0.4 0.4 0.7 0.4
fractal_dimension_mean -0.3 -0.1 -0.3 -0.3 0.6 0.6 0.3 0.2 0.5 1.0 0.0 0.2 0.0 -0.1 0.4 0.6 0.4 0.3 0.3 0.7 -0.3 -0.1 -0.2 -0.2 0.5 0.5 0.3 0.2 0.3 0.8
radius_se 0.7 0.3 0.7 0.7 0.3 0.5 0.6 0.7 0.3 0.0 1.0 0.2 1.0 1.0 0.2 0.4 0.3 0.5 0.2 0.2 0.7 0.2 0.7 0.8 0.1 0.3 0.4 0.5 0.1 0.0
texture_se -0.1 0.4 -0.1 -0.1 0.1 0.0 0.1 0.0 0.1 0.2 0.2 1.0 0.2 0.1 0.4 0.2 0.2 0.2 0.4 0.3 -0.1 0.4 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.0
perimeter_se 0.7 0.3 0.7 0.7 0.3 0.5 0.7 0.7 0.3 0.0 1.0 0.2 1.0 0.9 0.2 0.4 0.4 0.6 0.3 0.2 0.7 0.2 0.7 0.7 0.1 0.3 0.4 0.6 0.1 0.1
area_se 0.7 0.3 0.7 0.8 0.2 0.5 0.6 0.7 0.2 -0.1 1.0 0.1 0.9 1.0 0.1 0.3 0.3 0.4 0.1 0.1 0.8 0.2 0.8 0.8 0.1 0.3 0.4 0.5 0.1 0.0
smoothness_se -0.2 0.0 -0.2 -0.2 0.3 0.1 0.1 0.0 0.2 0.4 0.2 0.4 0.2 0.1 1.0 0.3 0.3 0.3 0.4 0.4 -0.2 -0.1 -0.2 -0.2 0.3 -0.1 -0.1 -0.1 -0.1 0.1
compactness_se 0.2 0.2 0.3 0.2 0.3 0.7 0.7 0.5 0.4 0.6 0.4 0.2 0.4 0.3 0.3 1.0 0.8 0.7 0.4 0.8 0.2 0.1 0.3 0.2 0.2 0.7 0.6 0.5 0.3 0.6
concavity_se 0.2 0.1 0.2 0.2 0.2 0.6 0.7 0.4 0.3 0.4 0.3 0.2 0.4 0.3 0.3 0.8 1.0 0.8 0.3 0.7 0.2 0.1 0.2 0.2 0.2 0.5 0.7 0.4 0.2 0.4
concave points_se 0.4 0.2 0.4 0.4 0.4 0.6 0.7 0.6 0.4 0.3 0.5 0.2 0.6 0.4 0.3 0.7 0.8 1.0 0.3 0.6 0.4 0.1 0.4 0.3 0.2 0.5 0.5 0.6 0.1 0.3
symmetry_se -0.1 0.0 -0.1 -0.1 0.2 0.2 0.2 0.1 0.4 0.3 0.2 0.4 0.3 0.1 0.4 0.4 0.3 0.3 1.0 0.4 -0.1 -0.1 -0.1 -0.1 -0.0 0.1 0.0 -0.0 0.4 0.1
fractal_dimension_se -0.0 0.1 -0.0 -0.0 0.3 0.5 0.4 0.3 0.3 0.7 0.2 0.3 0.2 0.1 0.4 0.8 0.7 0.6 0.4 1.0 -0.0 -0.0 -0.0 -0.0 0.2 0.4 0.4 0.2 0.1 0.6
radius_worst 1.0 0.4 1.0 1.0 0.2 0.5 0.7 0.8 0.2 -0.3 0.7 -0.1 0.7 0.8 -0.2 0.2 0.2 0.4 -0.1 -0.0 1.0 0.4 1.0 1.0 0.2 0.5 0.6 0.8 0.2 0.1
texture_worst 0.3 0.9 0.3 0.3 0.0 0.2 0.3 0.3 0.1 -0.1 0.2 0.4 0.2 0.2 -0.1 0.1 0.1 0.1 -0.1 -0.0 0.4 1.0 0.4 0.3 0.2 0.4 0.4 0.4 0.2 0.2
perimeter_worst 1.0 0.4 1.0 1.0 0.2 0.6 0.7 0.9 0.2 -0.2 0.7 -0.1 0.7 0.8 -0.2 0.3 0.2 0.4 -0.1 -0.0 1.0 0.4 1.0 1.0 0.2 0.5 0.6 0.8 0.3 0.1
area_worst 0.9 0.3 0.9 1.0 0.2 0.5 0.7 0.8 0.2 -0.2 0.8 -0.1 0.7 0.8 -0.2 0.2 0.2 0.3 -0.1 -0.0 1.0 0.3 1.0 1.0 0.2 0.4 0.5 0.7 0.2 0.1
smoothness_worst 0.1 0.1 0.2 0.1 0.8 0.6 0.4 0.5 0.4 0.5 0.1 -0.1 0.1 0.1 0.3 0.2 0.2 0.2 -0.0 0.2 0.2 0.2 0.2 0.2 1.0 0.6 0.5 0.5 0.5 0.6
compactness_worst 0.4 0.3 0.5 0.4 0.5 0.9 0.8 0.7 0.5 0.5 0.3 -0.1 0.3 0.3 -0.1 0.7 0.5 0.5 0.1 0.4 0.5 0.4 0.5 0.4 0.6 1.0 0.9 0.8 0.6 0.8
concavity_worst 0.5 0.3 0.6 0.5 0.4 0.8 0.9 0.8 0.4 0.3 0.4 -0.1 0.4 0.4 -0.1 0.6 0.7 0.5 0.0 0.4 0.6 0.4 0.6 0.5 0.5 0.9 1.0 0.9 0.5 0.7
concave points_worst 0.7 0.3 0.8 0.7 0.5 0.8 0.9 0.9 0.4 0.2 0.5 -0.1 0.6 0.5 -0.1 0.5 0.4 0.6 -0.0 0.2 0.8 0.4 0.8 0.7 0.5 0.8 0.9 1.0 0.5 0.5
symmetry_worst 0.2 0.1 0.2 0.1 0.4 0.5 0.4 0.4 0.7 0.3 0.1 -0.1 0.1 0.1 -0.1 0.3 0.2 0.1 0.4 0.1 0.2 0.2 0.3 0.2 0.5 0.6 0.5 0.5 1.0 0.5
fractal_dimension_worst 0.0 0.1 0.1 0.0 0.5 0.7 0.5 0.4 0.4 0.8 0.0 -0.0 0.1 0.0 0.1 0.6 0.4 0.3 0.1 0.6 0.1 0.2 0.1 0.1 0.6 0.8 0.7 0.5 0.5 1.0
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

figure(1)


texture_mean 1.0 0.3 -0.0 0.3 0.1 -0.1 0.4 0.3 0.0 0.1 0.0 0.1 0.1 0.3 0.1 0.1
area_mean 0.3 1.0 0.2 0.7 0.2 -0.3 -0.1 0.8 -0.2 0.2 -0.1 -0.0 0.1 0.5 0.1 0.0
smoothness_mean -0.0 0.2 1.0 0.5 0.6 0.6 0.1 0.2 0.3 0.2 0.2 0.3 0.8 0.4 0.4 0.5
concavity_mean 0.3 0.7 0.5 1.0 0.5 0.3 0.1 0.6 0.1 0.7 0.2 0.4 0.4 0.9 0.4 0.5
symmetry_mean 0.1 0.2 0.6 0.5 1.0 0.5 0.1 0.2 0.2 0.3 0.4 0.3 0.4 0.4 0.7 0.4
fractal_dimension_mean -0.1 -0.3 0.6 0.3 0.5 1.0 0.2 -0.1 0.4 0.4 0.3 0.7 0.5 0.3 0.3 0.8
texture_se 0.4 -0.1 0.1 0.1 0.1 0.2 1.0 0.1 0.4 0.2 0.4 0.3 -0.1 -0.1 -0.1 -0.0
area_se 0.3 0.8 0.2 0.6 0.2 -0.1 0.1 1.0 0.1 0.3 0.1 0.1 0.1 0.4 0.1 0.0
smoothness_se 0.0 -0.2 0.3 0.1 0.2 0.4 0.4 0.1 1.0 0.3 0.4 0.4 0.3 -0.1 -0.1 0.1
concavity_se 0.1 0.2 0.2 0.7 0.3 0.4 0.2 0.3 0.3 1.0 0.3 0.7 0.2 0.7 0.2 0.4
symmetry_se 0.0 -0.1 0.2 0.2 0.4 0.3 0.4 0.1 0.4 0.3 1.0 0.4 -0.0 0.0 0.4 0.1
fractal_dimension_se 0.1 -0.0 0.3 0.4 0.3 0.7 0.3 0.1 0.4 0.7 0.4 1.0 0.2 0.4 0.1 0.6
smoothness_worst 0.1 0.1 0.8 0.4 0.4 0.5 -0.1 0.1 0.3 0.2 -0.0 0.2 1.0 0.5 0.5 0.6
concavity_worst 0.3 0.5 0.4 0.9 0.4 0.3 -0.1 0.4 -0.1 0.7 0.0 0.4 0.5 1.0 0.5 0.7
symmetry_worst 0.1 0.1 0.4 0.4 0.7 0.3 -0.1 0.1 -0.1 0.2 0.4 0.1 0.5 0.5 1.0 0.5
fractal_dimension_worst 0.1 0.0 0.5 0.5 0.4 0.8 -0.0 0.0 0.1 0.4 0.1 0.6 0.6 0.7 0.5 1.0
texture_mean
area_mean
smoothness_mean
concavity_mean
symmetry_mean
fractal_dimension_mean
texture_se
area_se
smoothness_se
concavity_se
symmetry_se
fractal_dimension_se
smoothness_worst
concavity_worst
symmetry_worst
fractal_dimension_worst
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Figure(2)

- Results

We monitored 4 metrics to evaluate our model : **Accuracy, Precision, Recall, and F1 score.**

The results obtained are as follows:

Metric	value
Accuracy	**96.83 %**
Precision	**96.81 %**
Recall	**95.75 %**
F1 score	**96.25 %**

Attachments

Kaggle notebook that contains the code :

https://www.kaggle.com/code/muhammadabdalsattar/notebook26af019eb6