Name: Durva Kadam Roll No.:23

St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2025-2026
Class: BE-ITA/B, Semester: VII
Subject: Data Science Lab

Experiment – 9

1. **Aim:** To Evaluate classification algorithms.
2. **Objectives:** Students should be define and apply metrics to measure the performance of various learning algorithms
3. **Prerequisite:** Python basics
4. **Pre-Experiment Exercise:**
   Theory:

   Compare
   i. **Popular Classification Algorithms:**
   ii. **Logistic Regression**
   iii. **Naive Bayes**
   iv. **K-Nearest Neighbors**
   v. **Decision Tree**
   vi. **Support Vector Machines**

5. **Laboratory Exercise**
   Procedure:
   i. Compare above listed classifier algorithms and apply metrics to measure the performances of each.

6. **Post-Experiments Exercise:**
   A. Extended Theory:
   i. Explain various performance metrices.

7. **Conclusion:**

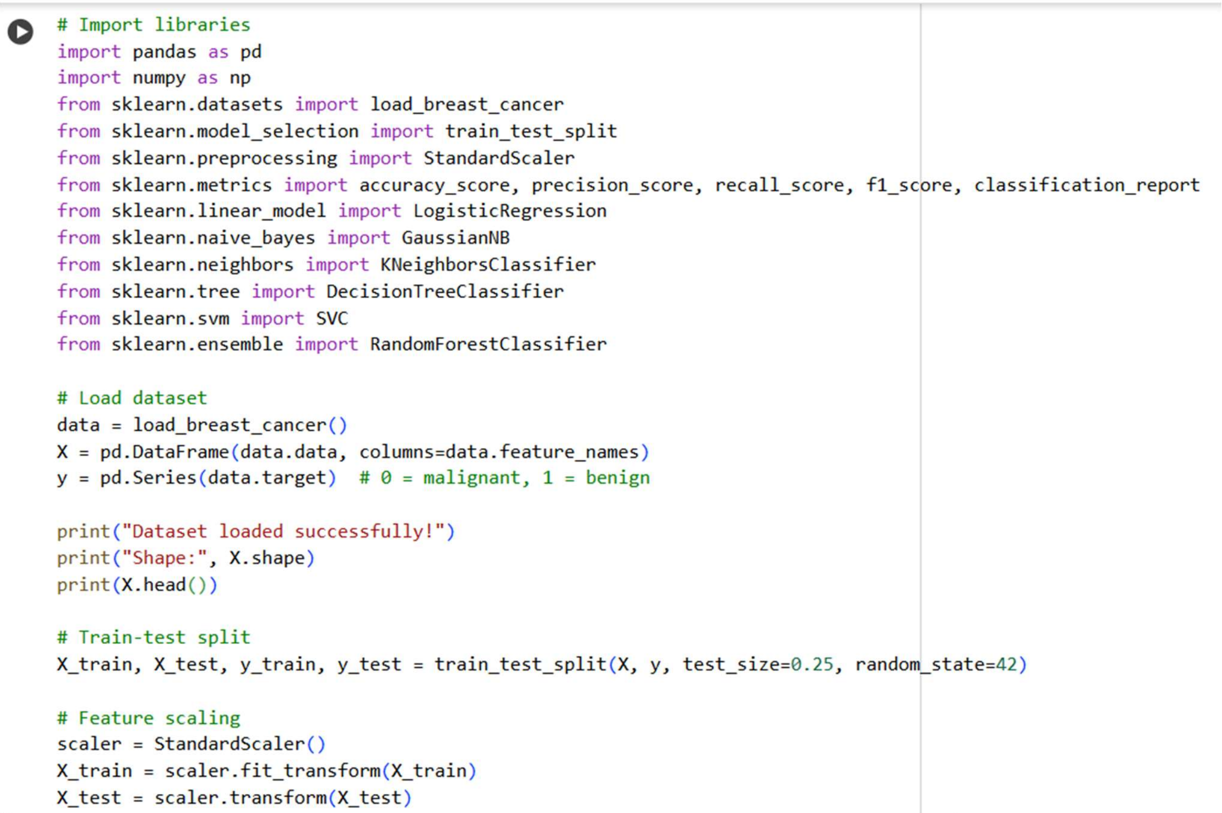
   i. Write what was performed in the program (s) .
   ii. What is the significance of program and what Objective is achieved?

8. **References:**

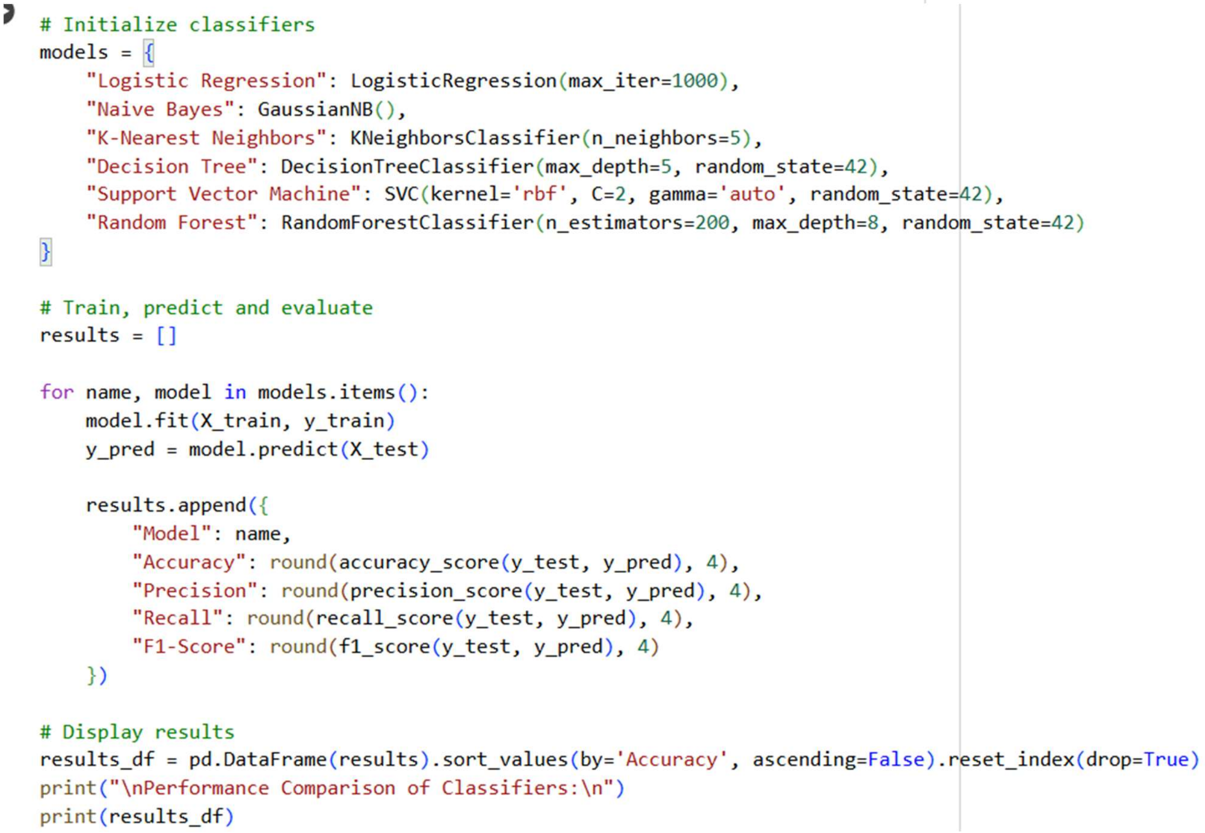
   a. https://scikit-learn.org/stable/modules/cross_validation.html

Name: Durva Kadam Roll No.:23

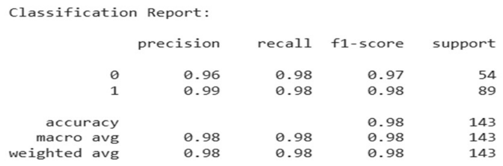
Lab Exercise:



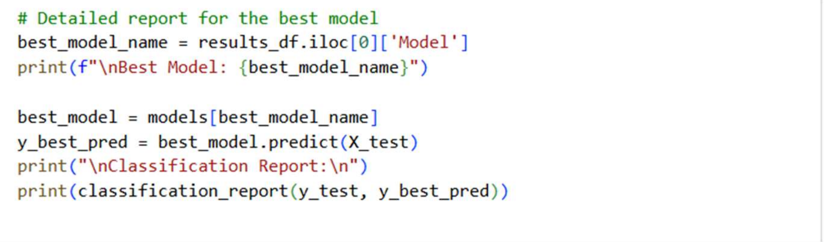
```python
# Import libraries
import pandas as pd
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

# Load dataset
data = load_breast_cancer()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)  # 0 = malignant, 1 = benign

print("Dataset loaded successfully!")
print("Shape:", X.shape)
print(X.head())

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```



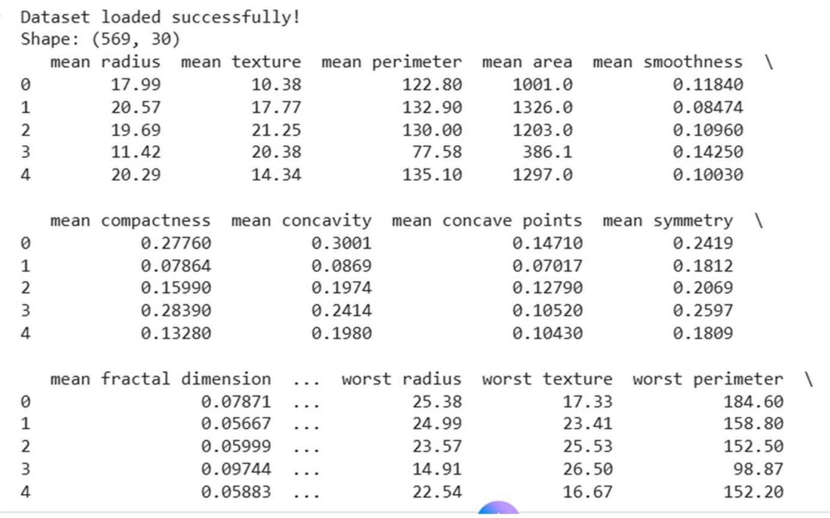
```python
# Initialize classifiers
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Naive Bayes": GaussianNB(),
    "K-Nearest Neighbors": KNeighborsClassifier(n_neighbors=5),
    "Decision Tree": DecisionTreeClassifier(max_depth=5, random_state=42),
    "Support Vector Machine": SVC(kernel='rbf', C=2, gamma='auto', random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=200, max_depth=8, random_state=42)
}

# Train, predict and evaluate
results = []

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    results.append({
        "Model": name,
        "Accuracy": round(accuracy_score(y_test, y_pred), 4),
        "Precision": round(precision_score(y_test, y_pred), 4),
        "Recall": round(recall_score(y_test, y_pred), 4),
        "F1-Score": round(f1_score(y_test, y_pred), 4)
    })

# Display results
results_df = pd.DataFrame(results).sort_values(by='Accuracy', ascending=False).reset_index(drop=True)
print("\nPerformance Comparison of Classifiers:\n")
print(results_df)
```



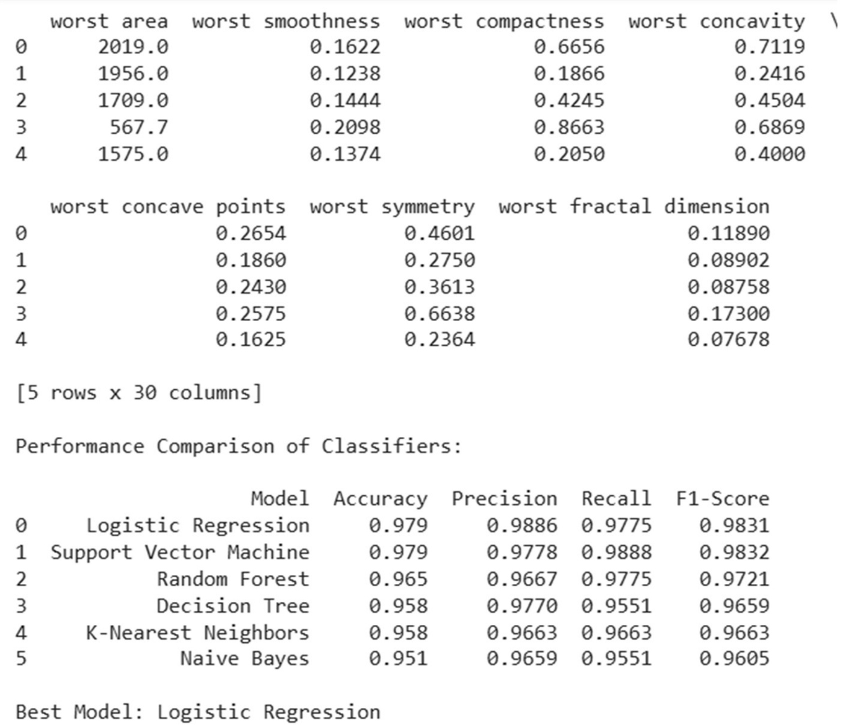
```
Classification Report:

              precision    recall  f1-score   support

           0       0.96      0.98      0.97        54
           1       0.99      0.98      0.98        89

    accuracy                           0.98       143
   macro avg       0.98      0.98      0.98       143
weighted avg       0.98      0.98      0.98       143
```



```
# Detailed report for the best model
best_model_name = results_df.iloc[0]['Model']
print(f"\nBest Model: {best_model_name}")

best_model = models[best_model_name]
y_best_pred = best_model.predict(X_test)
print("\nClassification Report:\n")
print(classification_report(y_test, y_best_pred))
```



```
Dataset loaded successfully!
Shape: (569, 30)
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38          122.80     1001.0          0.11840   
1        20.57         17.77          132.90     1326.0          0.08474   
2        19.69         21.25          130.00     1203.0          0.10960   
3        11.42         20.38           77.58      386.1          0.14250   
4        20.29         14.34          135.10     1297.0          0.10030   

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0           0.27760          0.3001              0.14710         0.2419   
1           0.07864          0.0869              0.07017         0.1812   
2           0.15990          0.1974              0.12790         0.2069   
3           0.28390          0.2414              0.10520         0.2597   
4           0.13280          0.1980              0.10430         0.1809   

   mean fractal dimension  ...  worst radius  worst texture  worst perimeter  \
0                 0.07871  ...         25.38          17.33           184.60   
1                 0.05667  ...         24.99          23.41           158.80   
2                 0.05999  ...         23.57          25.53           152.50   
3                 0.09744  ...         14.91          26.50            98.87   
4                 0.05883  ...         22.54          16.67           152.20   
```



```
   worst area  worst smoothness  worst compactness  worst concavity  \
0      2019.0            0.1622             0.6656           0.7119   
1      1956.0            0.1238             0.1866           0.2416   
2      1709.0            0.1444             0.4245           0.4504   
3       567.7            0.2098             0.8663           0.6869   
4      1575.0            0.1374             0.2050           0.4000   

   worst concave points  worst symmetry  worst fractal dimension  
0                0.2654          0.4601                  0.11890  
1                0.1860          0.2750                  0.08902  
2                0.2430          0.3613                  0.08758  
3                0.2575          0.6638                  0.17300  
4                0.1625          0.2364                  0.07678  

[5 rows x 30 columns]

Performance Comparison of Classifiers:

                    Model  Accuracy  Precision  Recall  F1-Score
0     Logistic Regression     0.979     0.9886  0.9775    0.9831
1  Support Vector Machine     0.979     0.9778  0.9888    0.9832
2           Random Forest     0.965     0.9667  0.9775    0.9721
3           Decision Tree     0.958     0.9770  0.9551    0.9659
4     K-Nearest Neighbors     0.958     0.9663  0.9663    0.9663
5             Naive Bayes     0.951     0.9659  0.9551    0.9605

Best Model: Logistic Regression
```