EXP-3

STUDY OF THE CLASSIFIERS WITH RESPECT TO STATISTICAL PARAMETERS.

Aim:- To study and evaluate the performance of three different classifier using statistical metrics such as accuracy, precision, etc on an open source dataset.

Description:-

1) Accuracy - It measures the proportion of correctly predicted instances among all predictions.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + TN)}$$

2) Precision - It measures the correctness among positive predictions.

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

3) Recall - Measures how well actual positive are identified

$$\text{Recall} = \frac{TP}{(TP + FN)}$$

4) F1 Scores - Harmonic mean of precision & recall

$$F_1 \text{ score} = \frac{2 \times (\text{Precision} + \text{Recall})}{\text{Precision} + \text{Recall}}$$

5) Confusion Matrix - This is a table showing the correct & incorrect predictions across classes. It helps visualize model performance with True Positive (TP), False Positive (FP), True Negative (TN) & False Negative (FN)

Procedure:

1) Load the open source dataset.
2) Split the dataset into training & testing sets
3) Train the classifier Logistic Regression, KNN & Decision Tree.
4) Predict labels on the test data.
5) Evaluate each classifier using accuracy, precision etc
6) Visualise the confusion matrix.

Program:-

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score,
    recall_score, f1_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import kNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier

data = load_breast_cancer(data)
x = data.data
y = data.target
x_train, x_test, y_train, y_test = train_test_split(x, y,
    test_size = 0.25, random_state = 42)
models = {"Logistic Regression": LogisticRegression(max_iter = 5000),
    "KNN": kNeighborsClassifier(n_neighbors = 5), "Decision Tree":
    DecisionTreeClassifier()}
for name, model in models.items():
    model.fit(x_train, y_train)
```

```
y_pred = model.predict(X_test)
print(name)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1 score:", f1_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("-" * 30)
```

Result: The classification of Logistic Regression, KNN, & Decision Tree were successfully evaluated using the Breast cancer dataset.

All the models performed well, with Logistic Regression achieving the highest overall performance across all metrics.

22/8

Output :-

Classifier	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.976	0.981	0.981	0.981
KNN	0.959	0.946	0.990	0.968
Decision Tree	0.918	0.951	0.916	0.933

Confusion Matrix :-

Logistic Regression

61 (TP)	2 (FN)
2 (FP)	106 (TN)

where malignant (cancerous) → +ve

Benign (non-cancerous) → -ve

KNN

57 (TP)	6 (FN)
1 (FP)	107 (TN)

Decision Tree

58 (TP)	5 (FN)
9 (FP)	99 (TN)

In [1]:
```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
data = load_breast_cancer()
x = data.data
y = data.target
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_sta
models = {
    "LogisticRegression": LogisticRegression(max_iter=5000),
    "KNN": KNeighborsClassifier(n_neighbors=5),
    "DecisionTree": DecisionTreeClassifier()
}
for name, model in models.items():
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print(f"Model: {name}")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("Precision:", precision_score(y_test, y_pred))
    print("Recall:", recall_score(y_test, y_pred))
    print("F1 Score:", f1_score(y_test, y_pred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("-" * 30)
```

```
Model: LogisticRegression
Accuracy: 0.9766081871345029
Precision: 0.9814814814814815
Recall: 0.9814814814814815
F1 Score: 0.9814814814814815
Confusion Matrix:
 [[ 61   2]
 [  2 106]]
------------------------------
Model: KNN
Accuracy: 0.9590643274853801
Precision: 0.9469026548672567
Recall: 0.9907407407407407
F1 Score: 0.9683257918552036
Confusion Matrix:
 [[ 57   6]
 [  1 107]]
------------------------------
Model: DecisionTree
Accuracy: 0.9181286549707602
Precision: 0.9519230769230769
Recall: 0.9166666666666666
F1 Score: 0.9339622641509434
Confusion Matrix:
 [[58  5]
 [ 9 99]]
------------------------------
```