Exp-2

AIM

To implement a classifier using an open-source dataset.

OBJECTIVE.

↳ To understand and implement a supervised machine learning classifier.

↳ To train and test the classifier on an open source data (Iris.).

↳ To evaluate the classifier's performance using accuracy metrics.

PSEUDOCODE

```
IMPORT PANDAS AS PD
FROM sklearn.model-selection IMPORT train_test_split.
FROM sklearn.linear_model IMPORT LogisticRegression.
FROM sklearn.metrics IMPORT accuracy_score.
DATA = load_iris_dataset()
X = data.features
Y = data.lables.
X_train, X_test, y_train, y_test = train_test split(x, y,
                                   test_size=0.2)
model = Logistic Regression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

PRINT("ACCURACY", accuracy)

OBSERVATION

DATASET : IRIS Dataset (150 samples, 4 features, 3 classes)

CLASSIFIER : Logistic Regression

Test/Train Split : 80% - Training, 20% Testing

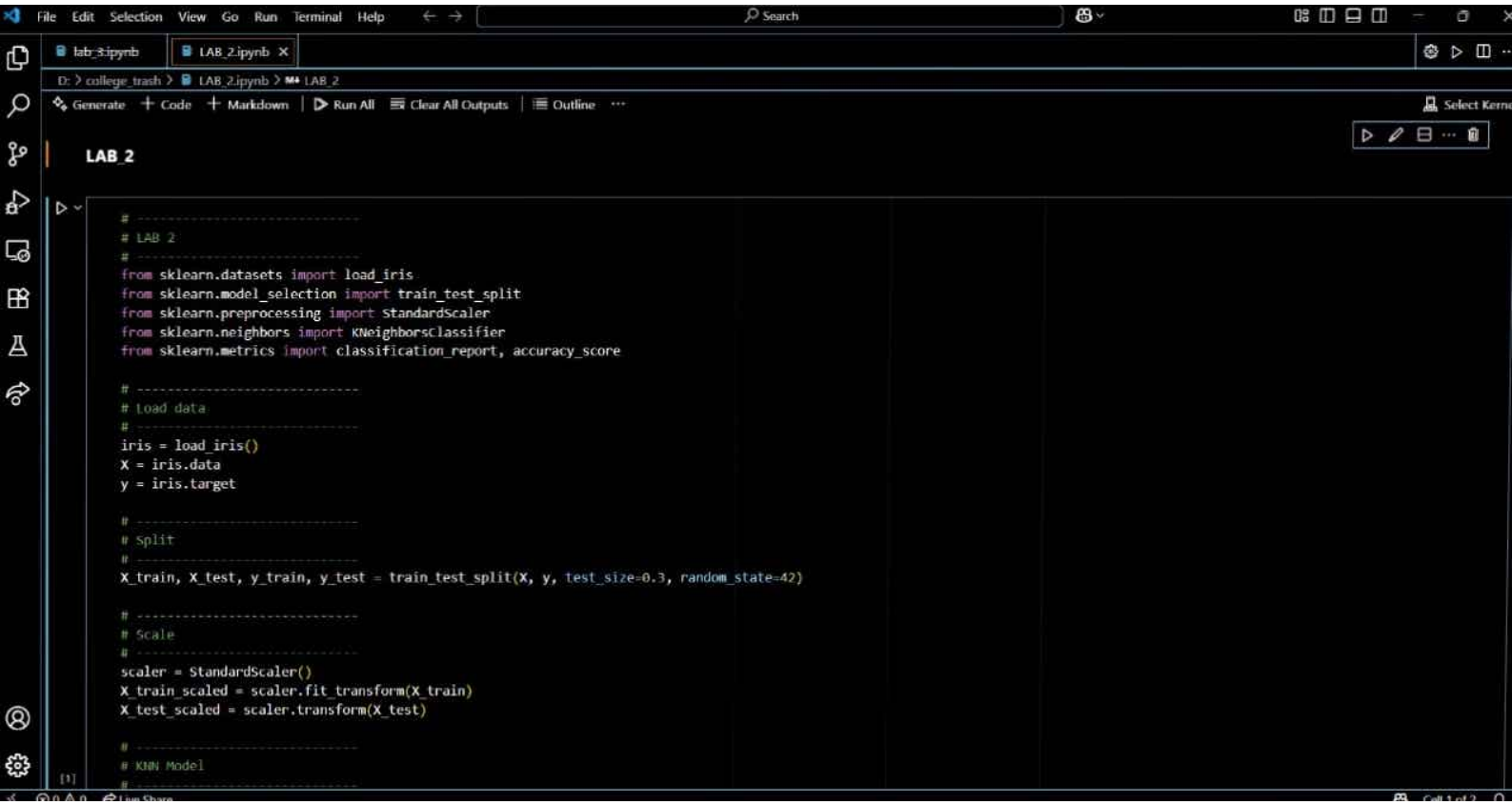
ACCURACY :

Classification Report:

	Precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	19
Versicolor	1.00	0.62	0.76	13
virginica	0.72	1.00	0.84	13
accuracy			0.89	45
macro avg	0.91	0.87	0.87	45
weighted avg	0.92	0.89	0.88	45

RESULT

A classifier was successfully implemented using the open-source Iris dataset. The model achieved high accuracy indicating effective classification of flower species.


File Edit Selection View Go Run Terminal Help
Search
lab_3.ipynb
LAB_2.ipynb
D: > college_trash > LAB_2.ipynb > LAB_2
Generate
Code
Markdown
Run All
Clear All Outputs
Outline
Select Kernel
LAB_2
# ----------------------------
# LAB 2
# ----------------------------
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score
# ----------------------------
# Load data
# ----------------------------
iris = load_iris()
X = iris.data
y = iris.target
# ----------------------------
# Split
# ----------------------------
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# ----------------------------
# Scale
# ----------------------------
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# ----------------------------
# KNN Model
# ----------------------------
[1]

File Edit Selection View Go Run Terminal Help

lab_3.ipynb | LAB_2.ipynb

D: > college_trash > LAB_2.ipynb > LAB_2

Generate | Code | Markdown | Run All | Clear All Outputs | Outline | ... | Select Kernel

```
# Split
# ----------------------------
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# ----------------------------
# Scale
# ----------------------------
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# ----------------------------
# KNN Model
# ----------------------------
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train_scaled, y_train)

# ----------------------------
# Predict
# ----------------------------
y_pred = knn.predict(X_test_scaled)

# ----------------------------
# Results
# ----------------------------
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

[1] Python

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

           0       1.00      1.00      1.00        19
           1       1.00      0.92      0.96        13
```

File Edit Selection View Go Run Terminal Help

lab_3.ipynb | LAB_2.ipynb

D: > college_trash > LAB_2.ipynb > LAB_2

Generate | Code | Markdown | Run All | Clear All Outputs | Outline | ... | Select Kernel

```
# Results
# ----------------------------
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

[1] Python

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

           0       1.00      1.00      1.00        19
           1       1.00      0.92      0.96        13
           2       0.93      1.00      0.96        13

    accuracy                           0.98        45
   macro avg       0.98      0.97      0.97        45
weighted avg       0.98      0.98      0.98        45

Accuracy Score: 0.9777777777777777
```

0 0 Live Share

Cell 1 of 2