

EXPERIMENT – 12

Implement Iris Flower Classification using KNN

CODE :

```
# Iris Flower Classification using K-Nearest Neighbors (KNN)

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

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

# KNN model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Prediction
y_pred = knn.predict(X_test)

# Output
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n",
```

```
classification_report(y_test, y_pred, target_names=iris.target_names))
```

```
# Predict for new flower
```

```
new_flower = [[5.1, 3.5, 1.4, 0.2]]
```

```
prediction = knn.predict(new_flower)
```

```
print("Predicted Iris Class:", iris.target_names[prediction[0]])
```

OUTPUT :

The screenshot shows the Python 3.13.2 shell interface. The code runs a KNN classifier on the Iris dataset, resulting in a 100% accuracy score and a detailed classification report. The report includes precision, recall, f1-score, and support for each class (setosa, versicolor, virginica) and overall metrics (accuracy, macro avg, weighted avg). The final output shows the predicted class for a new flower as 'setosa'.

```
IDLE Shell 3.13.2
File Edit Shell Debug Options Window Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/lalit/AppData/Local/Programs/Python/Python313/exp-12.py ==
Accuracy: 1.0
Classification Report:
precision    recall    f1-score   support
setosa       1.00     1.00     1.00      15
versicolor   1.00     1.00     1.00      11
virginica    1.00     1.00     1.00      12
accuracy          1.00      -        -       38
macro avg       1.00     1.00     1.00      38
weighted avg    1.00     1.00     1.00      38
Predicted Iris Class: setosa
>>> |
```