

EXPERIMENT – 17

Implement Perceptron based IRIS classification

CODE :

```
# Iris Flower Classification using Perceptron

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Perceptron
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
)

# Perceptron model
model = Perceptron(max_iter=1000, eta0=0.01, random_state=42)
model.fit(X_train, y_train)

# Prediction
y_pred = model.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 a new flower
new_flower = [[5.1, 3.5, 1.4, 0.2]]
prediction = model.predict(new_flower)
print("Predicted Iris Class:", iris.target_names[prediction[0]])
```

OUTPUT :

```
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-16.py ==
Accuracy: 0.7894736842105263
Classification Report:
      precision    recall   f1-score   support
      setosa       0.75     1.00     0.86      15
versicolor       0.73     0.73     0.73      11
virginica       1.00     0.58     0.74      12
      accuracy          0.79      38
macro avg       0.83     0.77     0.77      38
weighted avg     0.82     0.79     0.78      38

Predicted Iris Class: setosa
>>>
```