RIPHAH
INTERNATIONAL
UNIVERSITY

Artificial Intelligence

Lab 10 Tasks

Name: Samreen Bibi

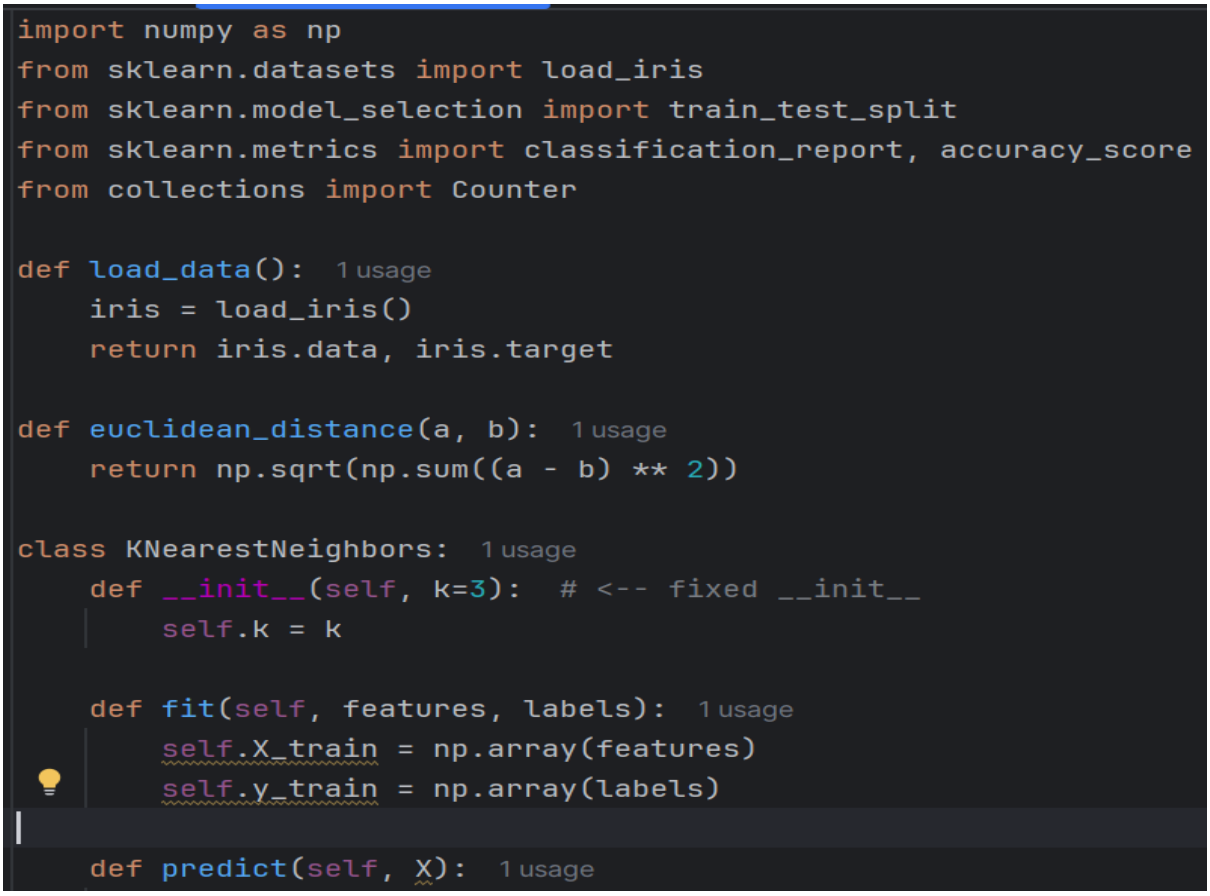
Sap ID: 46484

Batch: BSCS-6th semester

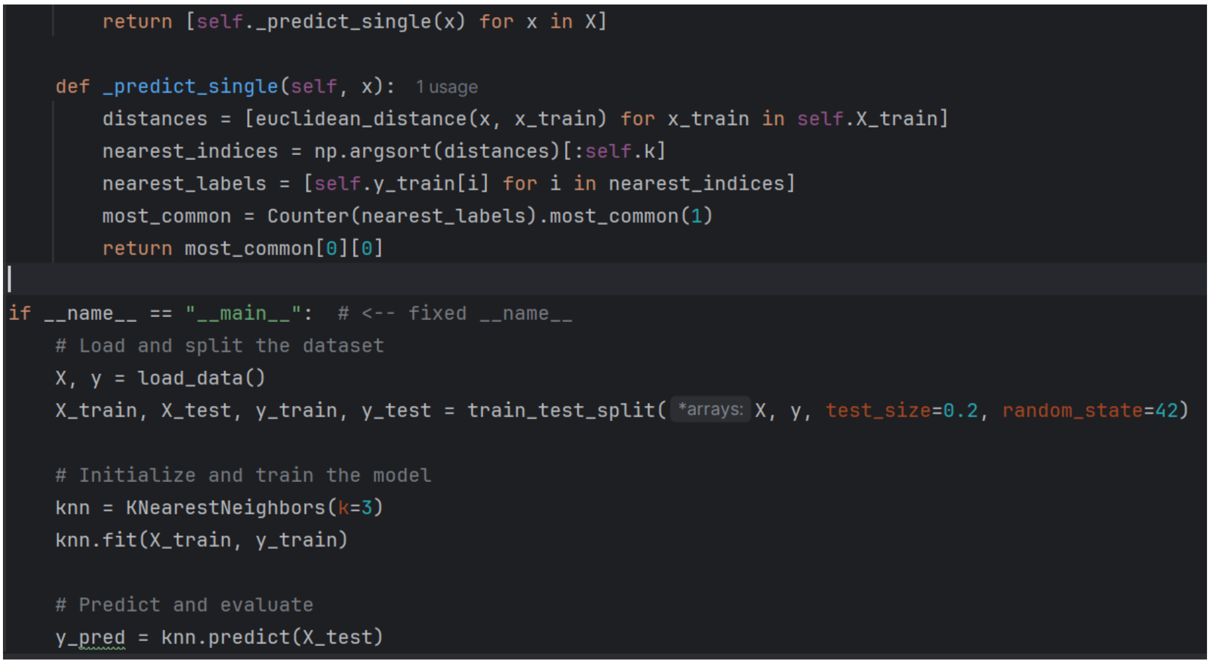
Lab Instructor:

Mam Ayesha Akram

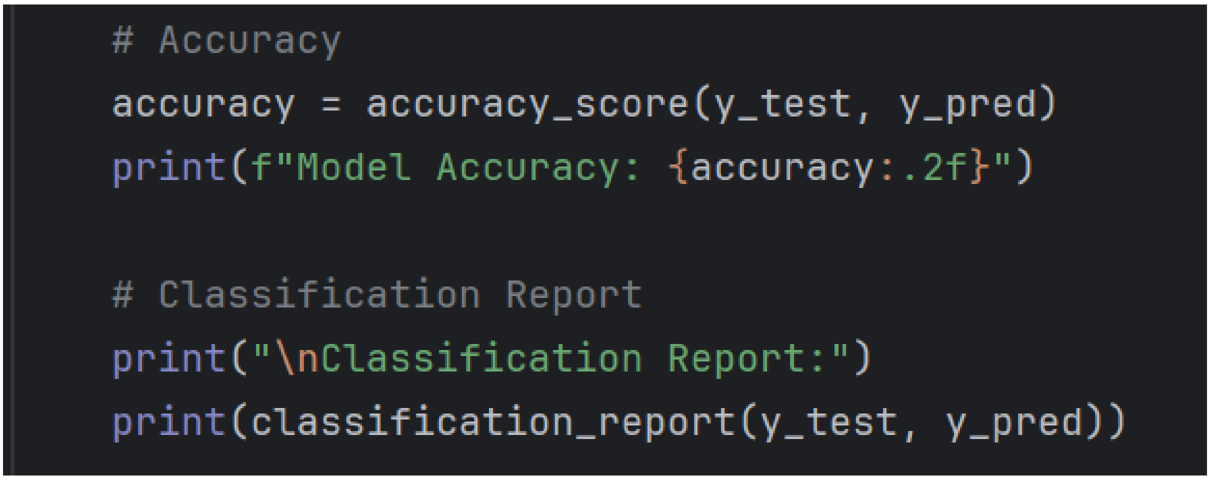
Task#1



```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from collections import Counter

def load_data():  1 usage
    iris = load_iris()
    return iris.data, iris.target

def euclidean_distance(a, b):  1 usage
    return np.sqrt(np.sum((a - b) ** 2))

class KNearestNeighbors:  1 usage
    def __init__(self, k=3):  # <-- fixed __init__
        self.k = k

    def fit(self, features, labels):  1 usage
        self.X_train = np.array(features)
        self.y_train = np.array(labels)

    def predict(self, X):  1 usage
```

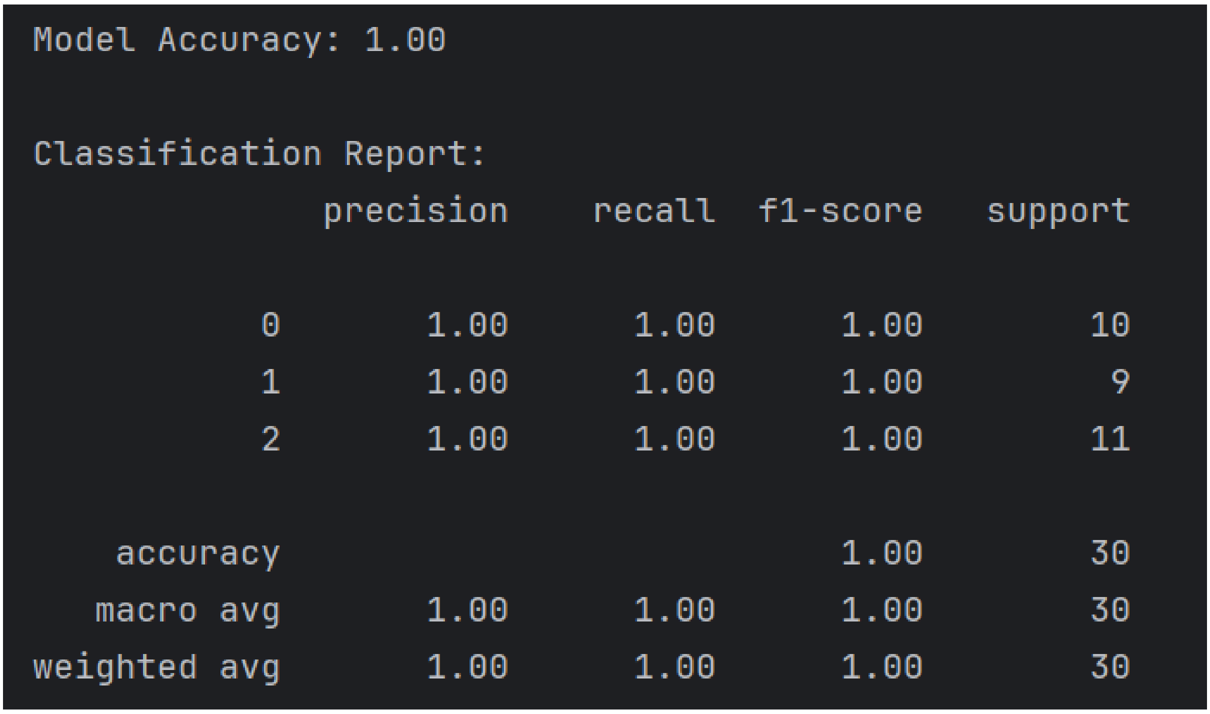


```
        return [self._predict_single(x) for x in X]

    def _predict_single(self, x):  1 usage
        distances = [euclidean_distance(x, x_train) for x_train in self.X_train]
        nearest_indices = np.argsort(distances)[:self.k]
        nearest_labels = [self.y_train[i] for i in nearest_indices]
        most_common = Counter(nearest_labels).most_common(1)
        return most_common[0][0]

if __name__ == "__main__":  # <-- fixed __name__
    # Load and split the dataset
    X, y = load_data()
    X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)

    # Initialize and train the model
    knn = KNearestNeighbors(k=3)
    knn.fit(X_train, y_train)

    # Predict and evaluate
    y_pred = knn.predict(X_test)
```



```
    # Accuracy
    accuracy = accuracy_score(y_test, y_pred)
    print(f"Model Accuracy: {accuracy:.2f}")

    # Classification Report
    print("\nClassification Report:")
    print(classification_report(y_test, y_pred))
```

Output:



```
Model Accuracy: 1.00

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```