Assignment No.7

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn import metrics

import matplotlib.pyplot as plt

import seaborn as sns

# Load dataset

iris = load\_iris()

X = iris.data

y = iris.target

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Create KNN classifier with k=5

knn = KNeighborsClassifier(n\_neighbors=5)

knn.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = knn.predict(X\_test)

# Evaluate model performance

print("Accuracy:", metrics.accuracy\_score(y\_test, y\_pred))

# Hyperparameter tuning: Evaluate different k values

k\_values = range(1, 21)

training\_scores = []

test\_scores = []

for k in k\_values:

knn\_k = KNeighborsClassifier(n\_neighbors=k)

knn\_k.fit(X\_train, y\_train)

training\_scores.append(knn\_k.score(X\_train, y\_train))

test\_scores.append(knn\_k.score(X\_test, y\_test))

# Plotting training and test scores

plt.figure(figsize=(12, 6))

sns.lineplot(x=k\_values, y=training\_scores, marker='o', label='Training Score', color='blue')

sns.lineplot(x=k\_values, y=test\_scores, marker='o', label='Test Score', color='red')

plt.xlabel('Number of Neighbors (k)')

plt.ylabel('Score')

plt.title('KNN Classifier Performance')

plt.legend()

plt.grid()

plt.show()

#OUTPUT

Accuracy: 1.0

