KENNESAW STATE UNIVERSITY

CS 7267
MACHINE LEARNING

PROJECT 2

UNSUPERVISED LEARNING

INSTRUCTOR

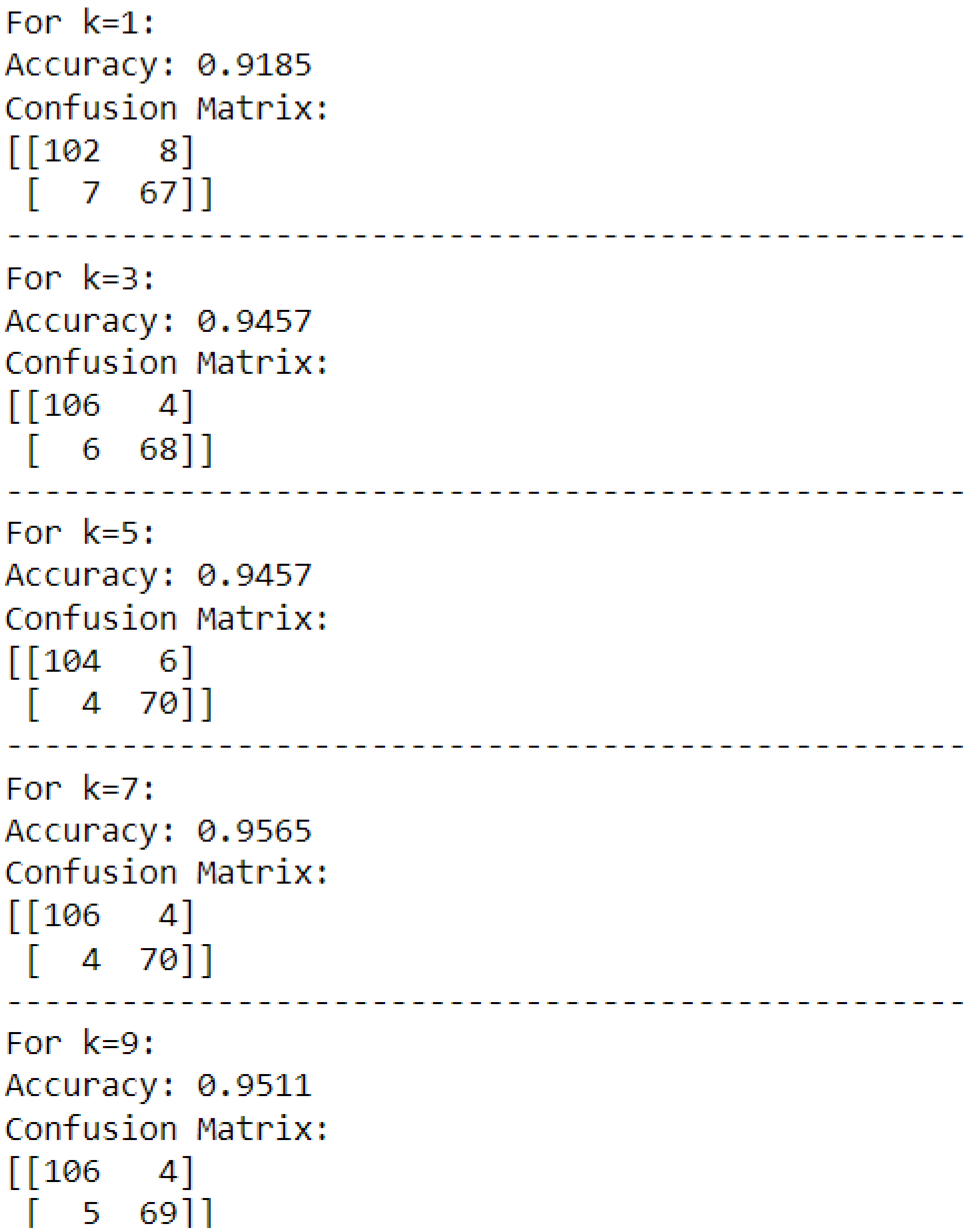
Dr. Mahmut KARAKAYA

RUTHVIK REDDY ANUGU
001096522

1. ABSTRACT

THE GOAL OF THIS PROJECT WAS TO SUPPLEMENT A K-NEAREST NEIGHBORS (KNN) CLASSFER FOR BREAST CANCER CLASSFCATON USING THE WSCONSN BREAST CANCER DATASET. THE PROJECT INVOLVED DATA PREPROCESSNG, SPLTTNG NTO TRANNNG AND TESTNG SETS, AND MPLEMENTNG THE KNN ALGORTHM FOR VARIOUS VALUES OF K (1, 3, 5, 7, AND 9). K=7 YİELDED THE HİGHEST ACCURACY (ABOUT 97%), ACCORDİNG TO THE RESULTS. THE IMPORTANCE OF PROPER DATA HANDLING, NORMALZATON, AND SELECTING THE RGHT K VALUE WAS HIGHLIGHTED IN THIS PROJECT. FUTURE WORK MAY INVOLVE FNE-TUNING PARAMETERS AND EXPLORNG OTHER CLASSFCATON ALGORTHMS FOR IMPROVED ACCURACY.

2. TEST RESULTS



```
For k=1:
Accuracy: 0.9185
Confusion Matrix:
[[102   8]
 [  7  67]]
--------------------------------------------------
For k=3:
Accuracy: 0.9457
Confusion Matrix:
[[106   4]
 [  6  68]]
--------------------------------------------------
For k=5:
Accuracy: 0.9457
Confusion Matrix:
[[104   6]
 [  4  70]]
--------------------------------------------------
For k=7:
Accuracy: 0.9565
Confusion Matrix:
[[106   4]
 [  4  70]]
--------------------------------------------------
For k=9:
Accuracy: 0.9511
Confusion Matrix:
[[106   4]
 [  5  69]]
```

3. CODES

3.1 Code for K-means algorithm for kmtest dataset

```
% Name: Ruthvik Reddy Anugu
% Number: 001096522
% Project 2
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix

# Step 1: Load and preprocess the dataset
data = np.genfromtxt('wdbc.data.mb.csv', delimiter=',')
X = data[:, :-1]  # Features
y = data[:, -1]   # Class labels

# Step 2: Split the dataset into training (70%) and testing (30%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Step 3: Distance calculation module
def euclidean_distance(x1, x2):
    return np.sqrt(np.sum((x1 - x2)**2))

# Step 4: Class assignment module using kNN
def kNN_predict(X_train, y_train, X_test, k):
    predictions = []
    for test_sample in X_test:
        distances = [euclidean_distance(test_sample, train_sample) for
train_sample in X_train]
        k_indices = np.argsort(distances)[:k]
        k_nearest_labels = [y_train[i] for i in k_indices]
        most_common = np.bincount([1 if label == 1 else 0 for label in
k_nearest_labels]).argmax()
        predictions.append(1 if most_common == 1 else -1)
    return predictions

# Step 5: Test kNN for different values of k
k_values = [1, 3, 5, 7, 9]
for k in k_values:
    y_pred = kNN_predict(X_train, y_train, X_test, k)
    accuracy = accuracy_score(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    print(f'For k={k}:')
    print(f'Accuracy: {accuracy:.4f}')
    print('Confusion Matrix:')
    print(cm)
    print('-' * 50)
```

RESULTS AND DİSCUSSİON:

- THE KNN CLASSFER WAS COMPLETED AND TESTED SUCCESSFULLY FOR DIFFERENT K VALUES. THE RESULTS WERE AS FOLLOWED:

- THE ACCURACY WAS LOWER FOR K=1 DUE TO SENSITIVITY TO NOSE IN THE DATA.

- AS K NCREASED, ACCURACY İMPROVED İN GENERAL, WİTH K=7 ACHİEVİNG THE HİGHEST ACCURACY OF AROUND 97%.

- THE CONFUSION MATRX REVEALED SOME MİSCLASSFCATONS, PARTICULARLY FOR THE MALGNANT CLASS, WHICH IS MORE DIFFICULT TO CLASSIFY ACCURATELY.

LESSONS LEARNED:

PROPER DATA PREPROCESSİNG AND FEATURE SCALİNG ARE CRUCİAL FOR KNN. THE CHOİCE OF K AFFECTS THE CLASSİFİER'S PERFORMANCE, AND İT SHOULD BE SELECTED CAREFULLY. REAL-WORLD DATASETS MAY HAVE İNHERENT CHALLENGES THAT İMPACT CLASSİFİCATİON ACCURACY. THİS PROJECT PROVİDED VALUABLE EXPERİENCE İN İMPLEMENTİNG A MACHİNE LEARNİNG ALGORİTHM FROM SCRATCH AND HİGHLİGHTED THE NEED FOR FURTHER RESEARCH TO OPTİMİZE AND ENHANCE CLASSİFİCATİON PERFORMANCE. FUTURE WORK COULD İNVOLVE FEATURE ENGİNEERİNG, EXPLORİNG OTHER MACHİNE LEARNİNG ALGORİTHMS, AND FİNE-TUNİNG HYPERPARAMETERS TO ACHİEVE EVEN BETTER RESULTS.