Lab 1 (due: Jan 24)
Machine Learning - COSC 4360
Department of Computer Science and Electrical Engineering
Spring 2025
Exercise 1
Download the wine dataset from the following Machine Learning repository: ML Repository. Rename wine.data to wine.data.csv. In wine.names, you can read the descriptions of all the attributes. The dataset contains 13 features (columns 2-14) that contribute to the quality of wine. The dataset contains data for three types, i.e., labels, of wines, identified by the category values: 1, 2, and 3 (column 1). The dataset contains 178 records. Perform classification using KNN and, compute the accuracy of your model, print the confusion matrix, and predict to which one of the three classes the following four wines belong to given the following feature values:
[14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065],
[12.64,1.36,2.02,16.8,100,2.02,1.41,.53,.62,5.75,.98,1.59,450],
[12.53,5.51,2.64,25,96,1.79,.6,.63,1.1,5,.82,1.69,515],
[13.49,3.59,2.19,19.5,88,1.62,.48,.58,.88,5.7,.81,1.82,580]
Note 1: Use a test size of 30% and K=5
Note 2: The names of all columns are:
names = [’class’, ’Alcohol’,’Malic Acid’,’Ash’,’Acadlinity’,’Magnisium’,’Total Phenols’,’Flavanoids’, ’NonFlavanoid Phenols’, ’Proanthocyanins’, ’Color Intensity’, ’Hue’, ’OD280/OD315’, ’Proline’]

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier

names = ['class', 'Alcohol','Malic Acid','Ash','Acadlinity','Magnisium','Total Phenols',
         'Flavanoids','NonFlavanoid Phenols', 'Proanthocyanins', 'Color Intensity', 'Hue', 'OD280/OD315', 'Proline' ]
df = pd.read_csv('wine.data.csv', names=names)
X = np.array(df.iloc[:, 1:14])
y = np.array(df['class'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 )

knn = KNeighborsClassifier(n_neighbors=5).fit(X_train, y_train)
pred = knn.predict(X_test)

print('Model accuracy score: ', accuracy_score(y_test, pred))
print('Index\tPredicted\tActual')
for i in range(len(pred)):
    if pred[i]!=y_test[i]:
        print(i, '\t', pred[i], '\t', y_test[i], '***')

print(f'\nConfusion Matrix: \n{confusion_matrix(y_test, pred)}')

np.set_printoptions(precision = 2, suppress = True)
DataToPredict = np.array([[14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065],
                          [12.64,1.36,2.02,16.8,100,2.02,1.41,.53,.62,5.75,.98,1.59,450],
                          [12.53,5.51,2.64,25,96,1.79,.6,.63,1.1,5,.82,1.69,515],
                          [13.49,3.59,2.19,19.5,88,1.62,.48,.58,.88,5.7,.81,1.82,580]])

pred = knn.predict(DataToPredict)

print('Predicted Results\n')
for i in range(len(pred)):
    print('\t', DataToPredict[i], '\t', pred[i])
```

```
C:\Users\Aden\anaconda3\py
Model accuracy score:  0.7222222222222222
Index   Predicted       Actual
5       3       1 ***
7       3       1 ***
8       3       2 ***
16      1       3 ***
20      3       2 ***
24      1       3 ***
25      3       2 ***
28      3       2 ***
31      2       3 ***
33      1       2 ***
34      3       2 ***
37      3       2 ***
38      3       1 ***
41      1       2 ***
44      3       2 ***

Confusion Matrix:
[[19  0  3]
 [ 2 11  7]
 [ 2  1  9]]
Predicted Results

        [  14.23    1.71    2.43   15.6   127.      2.8     3.06    0.28    2.29
    5.64    1.04    3.92 1065.  ]        1
        [ 12.64   1.36   2.02  16.8  100.     2.02   1.41   0.53   0.62   5.75
   0.98   1.59 450.  ]   2
        [ 12.53   5.51   2.64  25.    96.     1.79   0.6    0.63   1.1    5.
   0.82   1.69 515.  ]   3
        [ 13.49   3.59   2.19  19.5   88.     1.62   0.48   0.58   0.88   5.7
   0.81   1.82 580.  ]   3
Press any key to continue . . .
```

Exercise 2

Given the iris.data.csv, produce the following two plots as shown in Figs. 1 and 2 below

iris.data.csv | e2.py | wine.data.csv | MachineLearn...ssmanAden.py

```python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt

names = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class']
df = pd.read_csv('iris.data.csv', names=names)
X = np.array(df.iloc[:, 0:4])
y = np.array(df['class'])
for i in range(len(y)):
    if y[i] == 'Iris-setosa':
        y[i] = 1
    elif y[i] == 'Iris-versicolor':
        y[i] = 2
    else:
        y[i] = 3

figure, axis = plt.subplots(1, 2)

axis[0].scatter(df['sepal_length'], df["sepal_width"], c=y)
axis[0].set_xlabel("Sepal Length")
axis[0].set_ylabel("Sepal Width")
axis[0].set_title("Figure 1: Sepal features")

axis[1].scatter(df['petal_length'], df["petal_width"], c=y)
axis[1].set_xlabel("Petal Length")
axis[1].set_ylabel("Petal Width")
axis[1].set_title("Figure 2: Petal features")

plt.show()
```