Name: Rahima Siddiqui(2303.KHI.DEG.030)
Peer Name: Osama Abdul Razzak(2303.KHI.DEG.029)

Assignment 3.2

> Implement a single classification model of your choice and try to achieve at least an 80% F1 score on the wine dataset provided by Sklearn.

Using KNN Classification Model

First, we import the following required libraries

```
[29]: from sklearn.neighbors import KNeighborsClassifier
      from sklearn.preprocessing import StandardScaler

[30]: from sklearn import datasets

[31]: import numpy as np

[32]: from sklearn.metrics import f1_score, mean_squared_error
      from sklearn.model_selection import train_test_split
```

then we import ‘wine.csv’ from sklearn dataset

```
[41]: wine = datasets.load_wine()
```

For Applying KNN classification model, we extract last five Features from our dataset as wine_x

```
[42]: wine_x = wine.data[:,8:]

[43]: wine_x.shape

[43]: (178, 5)
```

Name: Rahima Siddiqui(2303.KHI.DEG.030)
Peer Name: Osama Abdul Razzak(2303.KHI.DEG.029)

and target or labeled column as wine_y

```
wine_y = wine.target
wine_y.shape
```

```
(178,)
```

then we applied the train_test_split function which splits the ‘wine_x’ and ‘wine_y’ into to training and testing set
and test size is set to 0.2, which mean 20% data will used for testing and 80% data will used for training and standardized the features datasets

```
x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.2)
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

then we call KNeighbors Classifier and by default it takes k = 5 for predicting Five closest value

```
model = KNeighborsClassifier()
```

```
Init signature:
KNeighborsClassifier(
    n_neighbors=5,
    *,
    weights='uniform',
    algorithm='auto',
    leaf_size=30,
    p=2,
    metric='minkowski',
    metric_params=None,
    n_jobs=None,
)
Docstring:
Classifier implementing the k-nearest neighbors vote.

Read more in the :ref:`User Guide <classification>`.

Parameters
----------
n_neighbors : int, default=5
```

and fit our training dataset init

```
model.fit(x_train, y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

then we predict target value based on x_test datasets(input) and store it in ‘y_pred’ variable

```
y_pred = model.predict(x_test)
y_pred
```

```
array([2, 1, 1, 2, 0, 0, 2, 0, 0, 0, 1, 1, 0, 1, 0, 1, 2, 0, 0, 2, 0, 1,
       0, 0, 1, 2, 1, 2, 2, 0, 0, 0, 0, 1, 1, 2])
```

In the end, we apply F1 score for model prediction ‘y_pred’ and ‘y_test’

```
f1_score(y_test, y_pred, average="micro")
```

```
0.9722222222222222
```

we get 97% result