K-Nearest Neighbors (K-NN)

Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

```
dataset = pd.read_csv('Data_class.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
print(X_train)
```

```
[[-1.38219432  0.91903747  0.9407658  ...  2.22576767  2.27129602
   0.24623928]
 [ 0.03390689  1.27578287 -0.04290763 ...  1.82407819  1.94996317
   3.74830911]
 [ 0.22797663  1.27578287  2.25233038 ...  2.62745714  2.27129602
  -0.33743902]
 ...
 [ 0.16939025 -1.22143494 -0.69868992 ... -0.98774815 -0.62069958
  -0.33743902]
 [ 0.29888258 -0.50794414 -0.69868992 ... -0.58605867 -0.62069958
  -0.33743902]
 [-1.04129794  1.98927367  1.92443923 ...  1.42238871  1.30729749
  -0.33743902]]
```

```
print(X_test)
```

```
[[ 0.11037076 -1.22143494 -0.69868992 ... -0.98774815 -0.62069958
  -0.33743902]
 [ 0.08526811 -0.50794414 -0.69868992 ... -0.58605867 -0.62069958
  -0.33743902]
 [-0.56596836  0.20554667  0.61287466 ...  ✦ 732028  0.02196611
  -0.33743902]
```

```
...
[-0.48116108  0.20554667 -0.69868992 ... -0.18436919 -0.62069958
  0.24623928]
[ 0.05794779 -0.86468954 -0.37079877 ...  1.42238871 -0.62069958
 -0.33743902]
[ 0.09172701 -0.86468954 -0.69868992 ... -0.18436919 -0.62069958
 -0.33743902]]
```

Training the K-NN model on the Training set

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier()
```

Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[103   4]
 [  5  59]]
0.9473684210526315
```