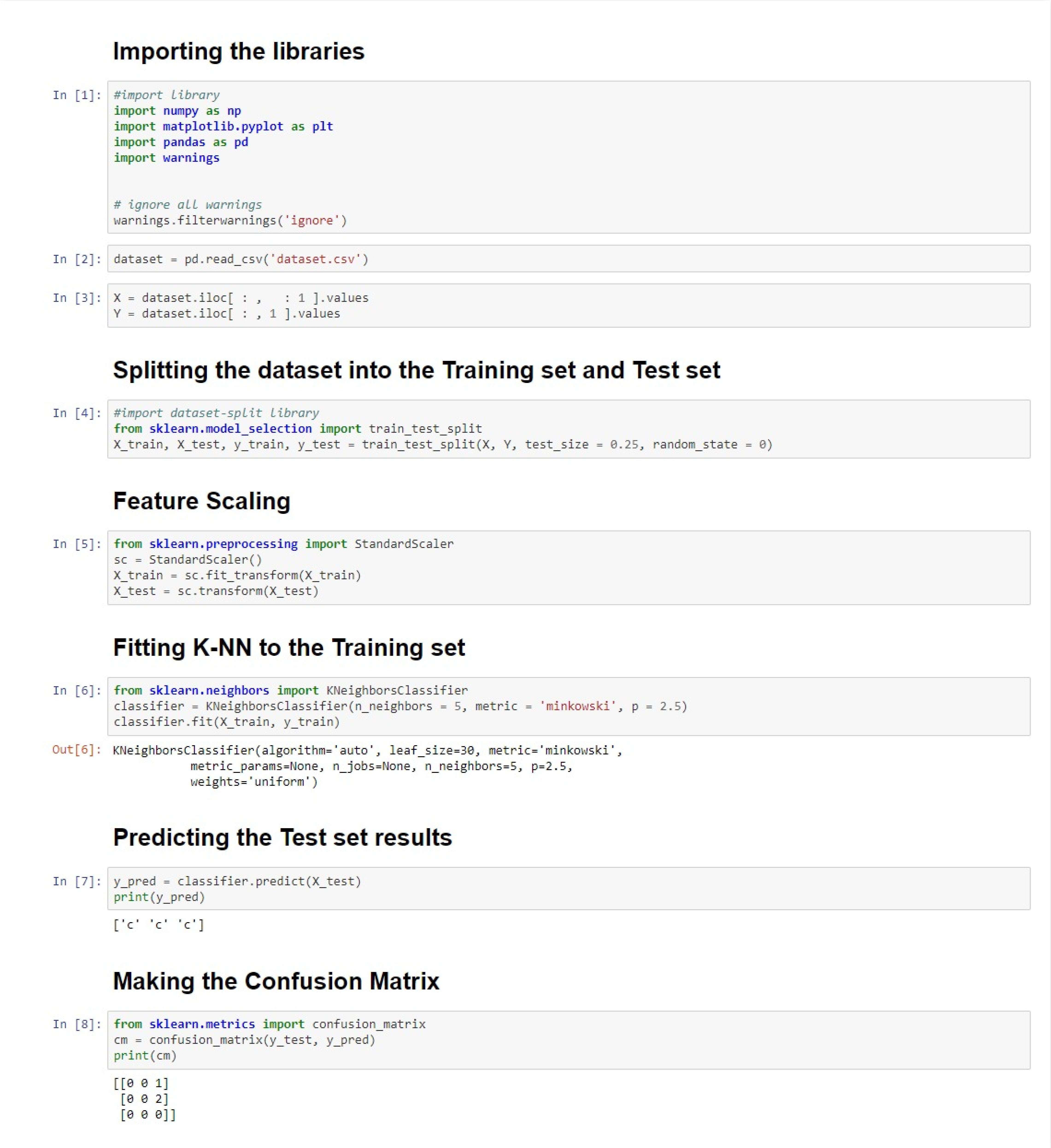

Importing the libraries

```
In [1]:  #import library
         import numpy as np
         import matplotlib.pyplot as plt
         import pandas as pd
         import warnings


         # ignore all warnings
         warnings.filterwarnings('ignore')
```

```
In [2]:  dataset = pd.read_csv('dataset.csv')
```

```
In [3]:  X = dataset.iloc[ : ,   : 1 ].values
         Y = dataset.iloc[ : , 1 ].values
```

Splitting the dataset into the Training set and Test set

```
In [4]:  #import dataset-split library
         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

```
In [5]:  from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         X_train = sc.fit_transform(X_train)
         X_test = sc.transform(X_test)
```

Fitting K-NN to the Training set

```
In [6]:  from sklearn.neighbors import KNeighborsClassifier
         classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2.5)
         classifier.fit(X_train, y_train)

Out[6]:  KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=5, p=2.5,
                    weights='uniform')
```

Predicting the Test set results

```
In [7]:  y_pred = classifier.predict(X_test)
         print(y_pred)

         ['c' 'c' 'c']
```

Making the Confusion Matrix

```
In [8]:  from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(y_test, y_pred)
         print(cm)

         [[0 0 1]
          [0 0 2]
          [0 0 0]]
```