**prediction-using-unsupervised-ML**

**Importing the libraries**

import numpy as np import matplotlib.pyplot as plt import pandas as pd from sklearn import datasets

**Load the iris dataset**

iris = datasets.load\_iris() iris\_df = pd.DataFrame(iris.data, columns = iris.feature\_names) iris\_df.head() # See the first 5 rows

**Finding the optimum number of clusters for k-means classification**

x = iris\_df.iloc[:, [0, 1, 2, 3]].values

from sklearn.cluster import KMeans wcss = []

for i in range(1, 11): kmeans = KMeans(n\_clusters = i, init = 'k-means++', max\_iter = 300, n\_init = 10, random\_state = 0) kmeans.fit(x) wcss.append(kmeans.inertia\_)

**Plotting the results onto a line graph,**

**`allowing us to observe 'The elbow'**

plt.plot(range(1, 11), wcss) plt.title('The elbow method') plt.xlabel('Number of clusters') plt.ylabel('WCSS') # Within cluster sum of squares plt.show()

**Applying kmeans to the dataset / Creating the kmeans classifier**

kmeans = KMeans(n\_clusters = 3, init = 'k-means++', max\_iter = 300, n\_init = 10, random\_state = 0) y\_kmeans = kmeans.fit\_predict(x)

**Visualising the clusters - On the first two columns**

plt.scatter(x[y\_kmeans == 0, 0], x[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Iris-setosa') plt.scatter(x[y\_kmeans == 1, 0], x[y\_kmeans == 1, 1], s = 100, c = 'blue', label = 'Iris-versicolour') plt.scatter(x[y\_kmeans == 2, 0], x[y\_kmeans == 2, 1], s = 100, c = 'green', label = 'Iris-virginica')

**Plotting the centroids of the clusters**

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:,1], s = 100, c = 'yellow', label = 'Centroids')

plt.legend()