TASK - PREDICTION USING UNSUPERVISED ML NAME- RAJ KAMAL SHAKYA Importing all required libraries In [1]: # Importing the libraries import numpy as np import matplotlib.pyplot as plt import pandas as pd from sklearn import datasets **Importing Datasets** In [2]: iris = datasets.load_iris() iris_df = pd.DataFrame(iris.data, columns = iris.feature_names) In [3]: iris_df sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) Out[3]: 5.1 3.5 1.4 0.2 4.9 3.0 1.4 0.2 4.7 3.2 1.3 0.2 4.6 3.1 1.5 0.2 4 5.0 3.6 1.4 0.2 145 6.7 3.0 5.2 2.3 146 6.3 2.5 5.0 1.9 5.2 147 6.5 3.0 2.0 148 5.4 6.2 3.4 2.3 149 5.9 3.0 5.1 1.8 150 rows × 4 columns In [4]: #check the shape of dataset iris_df.shape Out[4]: (150, 4) In [5]: #Reading first 10 observation iris_df.head(10) sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) Out[5]: 5.1 1 4.9 3.0 1.4 0.2 4.6 3.1 1.5 0.2 4 5.0 3.6 1.4 0.2 5 5.4 3.9 1.7 0.4 6 4.6 3.4 1.4 0.3

sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) Out[6]: 140 6.7 3.1 5.6 141 6.9 3.1 5.1 142 2.7 5.8 5.1 143 6.8 3.2 5.9 144 6.7 3.3 5.7 145 6.7 3.0 146 2.5 5.0 6.3 147 6.5 3.0 148 6.2 3.4 5.4 5.1 149 5.9 3.0 In [7]: #cheking numerical data iris_df.describe() sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) Out[7]: count 150.000000 150.000000 150.000000 5.843333 3.758000 3.057333 mean 0.828066 std 0.435866 1.765298 4.300000 2.000000 1.000000 min **25**% 5.100000 2.800000 1.600000 **50**% 5.800000 3.000000 4.350000 6.400000 3.300000 5.100000 7.900000 4.400000 6.900000 max In [8]: #checking null value in dataset iris_df.isnull().sum()

5.0

4.4

4.9

#Reading last 10 observation

iris_df.tail(10)

In [6]:

3.4

2.9

3.1

1.5

1.4

1.5

0.2

0.2

0.1

2.4

2.3

1.9

2.3

2.5

2.3

1.9

2.0

2.3

1.8

150.000000

1.199333

0.762238

0.100000

0.300000

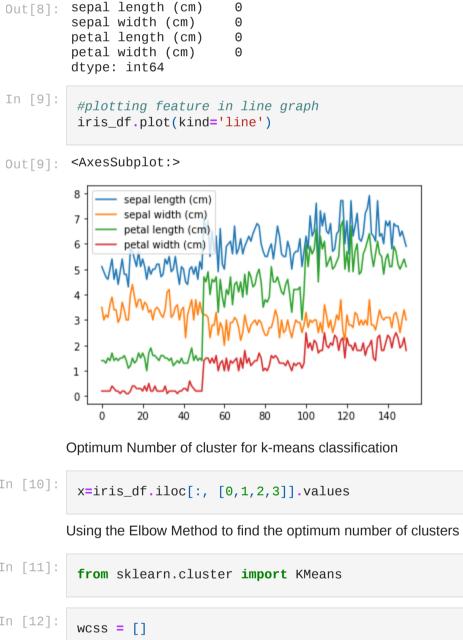
1.300000

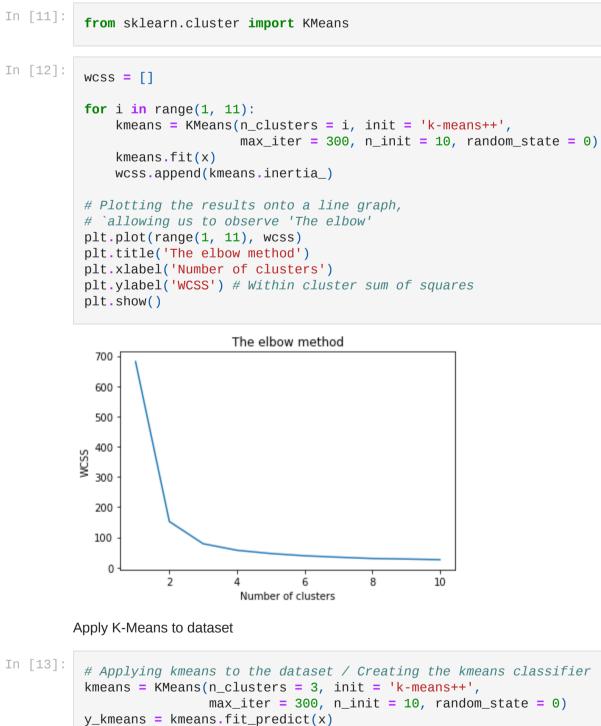
1.800000

2.500000

THE SPARKS FOUNDATION

DATA SCIENCE AND ANALYTICS TASK





Visualising the clusters - On the first two columns plt.scatter($x[y_kmeans == 0, 0], x[y_kmeans == 0, 1],$

plt.scatter($x[y_kmeans == 1, 0], x[y_kmeans == 1, 1],$

plt.scatter($x[y_kmeans == 2, 0], x[y_kmeans == 2, 1],$

Plotting the centroids of the clusters

Out[15]: <matplotlib.legend.Legend at 0x7fc88ca75780>

5.5

6.0

CONCLUSION: The optimum number of cluster is 3 in the Iris dataset

s = 100, c = 'red', label = 'Iris-setosa')

s = 100, c = 'blue', label = 'Iris-versicolour')

s = 100, c = 'green', label = 'Iris-virginica')

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

> Iris-virginica Centroids

7.0

1, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

2, 0, 2, 2, 2, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0], dtype=int32)

In [14]:

In [15]:

#printing y_kmeans

Visualizing the cluster

plt.legend()

4.0

3.5

3.0

2.5

2.0

In []:

Thank you!!

y_kmeans