```
Unsupervised Machine Learning - Iris Dataset
        Task 2: From the given 'Iris' dataset, predict the
        optimum number of clusters and represent it
        visually.
        Task completed during Data Science & Business
        Analytics Internship @THE SPARKS
        FOUNDATION (GRIP)
        Author: Harish Nakireddy
 In [1]: # Importing the required libraries
         import pandas as pd
         import matplotlib.pyplot as plt
         df = pd.read_csv("C:\\Users\\Asus\\Desktop\\Iris.csv")
Out[1]:
              Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
                                                                Species
              1
           0
                                                               Iris-setosa
                                                1.4
                                                          0.2 Iris-setosa
                         4.7
                                    3.2
                                                1.3
                                                          0.2 Iris-setosa
                                    3.1
                                                1.5
                                                          0.2 Iris-setosa
                                    3.6
                                                1.4
                                                          0.2 Iris-setosa
         145 146
                         6.7
                                    3.0
                                                5.2
                                                          2.3 Iris-virginica
         146 147
                                    2.5
                                                          1.9 Iris-virginica
         147 148
                                    3.0
                                                5.2
                                                          2.0 Iris-virginica
         148 149
                                                5.4
                                                          2.3 Iris-virginica
                                    3.0
                                                5.1
         149 150
                                                          1.8 Iris-virginica
        150 rows × 6 columns
        Visualising Data
In [2]: df.info
 Out[2]: <bound method DataFrame.info of</pre>
                                             Id SepalLengthCm SepalWidthCm P
        etalLengthCm PetalWidthCm \
           1
                          5.1
                                        3.5
                                                       1.4
                                                                    0.2
                                      3.0
3.2
        1
             2
                          4.9
                                                       1.4
                                                                    0.2
                     4.7
4.6
5.0
                                                   1.3
1.5
                                                                    0.2
                                      3.1
                                                                    0.2
             5
                                      3.6
                                                       1.4
                                                                    0.2
                                                       5.2
        145 146
                           6.7
                                         3.0
                                                                    2.3
                           6.3
                                         2.5
                                                       5.0
        146 147
                                                                    1.9
                           6.5
                                         3.0
                                                       5.2
                                                                     2.0
        147 148
         148
        149
            150
                                         3.0
                                                       5.1
                                                                     1.8
                    Species
        0
                Iris-setosa
        1
                Iris-setosa
        2
                Iris-setosa
        3
                Iris-setosa
                Iris-setosa
            Iris-virginica
        145
        146
            Iris-virginica
             Iris-virginica
             Iris-virginica
            Iris-virginica
         [150 rows x 6 columns]>
In [3]:
        df.describe()
Out[3]:
                    Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
         count 150.000000
                          150.000000
                                     150.000000
                                                 150.000000
                                                           150.000000
               75.500000
                            5.843333
                                       3.054000
                                                  3.758667
                                                             1.198667
         mean
                            0.828066
                                       0.433594
                                                             0.763161
               43.445368
                                                  1.764420
           std
                1.000000
                            4.300000
                                       2.000000
                                                  1.000000
                                                             0.100000
          min
          25%
               38.250000
                            5.100000
                                       2.800000
                                                  1.600000
                                                             0.300000
               75.500000
                            5.800000
                                       3.000000
                                                  4.350000
                                                             1.300000
              112.750000
                            6.400000
                                       3.300000
                                                  5.100000
                                                             1.800000
          max 150.000000
                            7.900000
                                       4.400000
                                                  6.900000
                                                             2.500000
        df.isnull().sum()
In [4]:
 Out[4]: Id
         SepalLengthCm
                         0
        SepalWidthCm
                         0
        PetalLengthCm
                         0
        PetalWidthCm
                         0
                         0
        Species
         dtype: int64
 In [5]: | df.head()
Out[5]:
           Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
                                                            Species
         0 1
                      5.1
                                 3.5
                                             1.4
                                                        0.2 Iris-setosa
         1 2
                      4.9
                                 3.0
                                             1.4
                                                        0.2 Iris-setosa
         2 3
                      4.7
                                 3.2
                                             1.3
                                                        0.2 Iris-setosa
         3 4
                                 3.1
                                             1.5
                      4.6
                                                        0.2 Iris-setosa
                      5.0
                                 3.6
                                             1.4
                                                        0.2 Iris-setosa
 In [7]: #making a new dataframe by droping unneccesery columns
         iris = pd.DataFrame(df,columns=df.columns.drop(["Id", "Species"]))
        iris.head()
Out[7]:
           SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
         0
                    5.1
                               3.5
                                          1.4
                                                     0.2
         1
                    4.9
                               3.0
                                          1.4
                                                     0.2
                    4.7
                               3.2
                                          1.3
                                                     0.2
         3
                    4.6
                               3.1
                                          1.5
                                                     0.2
                                                     0.2
        Finding the optimal number of clusters for K-
        Means and determining the value of K
In [9]: # Finding the optimum number of clusters for k-means classification
         x = iris.iloc[:,[0,1,2,3]].values
        # Importing KMeans from scikit learn
        from sklearn.cluster import KMeans
         sse=[]
        for k in range(1,11):
            km=KMeans(n_clusters=k, random_state=0)
            km.fit(x)
            sse.append(km.inertia_)
In [15]: # Plotting the results onto a line graph, allowing us to observe 'The el
         bow'
         plt.xlabel("K value")
        plt.ylabel("SSE")
        plt.title("The Elbow Method")
         plt.plot(range(1,11), sse, marker ="o")
         plt.show()
                          The Elbow Method
           700
           600
           500
           400
           300
           200
           100
                               K value
        We can clearly see why it is called 'The elbow
        method' from the above graph, the optimum
        clusters is where the elbow occurs. This is when
        the within cluster sum of squares (WCSS)
        doesn't decrease significantly with every
        iteration.
In [16]: # Applying KMeans to the dataset / Creating the KMeans Classifier
         # (number of clusters = 3)
         km = KMeans(n_clusters=3, random_state=0)
        y_predict = km.fit_predict(x)
        y_predict
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
               2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0,
               0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0,
               0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2])
In [17]: | #we can see there are 3 type of values 0s,1s,2s
         # Visualising the clusters - On the first two columns
        plt.scatter(x[y\_predict == 0, 0], x[y\_predict == 0, 1], s = 100, c = 're
        d', label = 'Iris-setosa')
         plt.scatter(x[y\_predict == 1, 0], x[y\_predict == 1, 1], s = 100, c = 'bl
        ue', label = 'Iris-versicolour')
         plt.scatter(x[y\_predict== 2, 0], x[y\_predict== 2, 1], s = 100, c = 'gre
         en', label = 'Iris-virginica')
         plt.legend()
Out[17]: <matplotlib.legend.Legend at 0x20202f59e50>
                                        Iris-setosa
                                        Iris-versicolour
                                         lris-virginica
         4.0
         3.5
         2.5
         2.0
In [20]: # Plotting the centroids of the clusters
         plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],label='cen
         troid',color="yellow")
        plt.legend()
Out[20]: <matplotlib.legend.Legend at 0x20202f93490>
                                            centroid
         3.3
         3.2
         3.1
         3.0
         2.9
         2.8
                     5.50
                          5.75
                               6.00
                                    6.25
                                        6.50
        Now combining both the above graphs and
        visualizing the clusters and their centroids
        together.
In [21]: | #we can see there are 3 type of values 0s,1s,2s
         # Visualising the clusters - On the first two columns
        plt.scatter(x[y\_predict == 0, 0], x[y\_predict == 0, 1], s = 100, c = 're
         d', label = 'Iris-setosa')
         plt.scatter(x[y\_predict == 1, 0], x[y\_predict == 1, 1], s = 100, c = 'bl
        ue', label = 'Iris-versicolour')
         plt.scatter(x[y\_predict== 2, 0], x[y\_predict== 2, 1], s = 100, c = 'gre
         en', label = 'Iris-virginica')
         # Plotting the centroids of the clusters
        plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],label='cen
         troid',color="yellow")
        plt.legend()
Out[21]: <matplotlib.legend.Legend at 0x202034a7a90>
         4.0
         3.5
         3.0
```

Therefore the optimum number of clusters is 3.

centroid

2.5

2.0