**Name: Sathish Kumar Chinnala Student ID: 1002050768**

# MACHINE LEARNING - CSE 6363-005 - PROJECT 3

## K-means Clustering

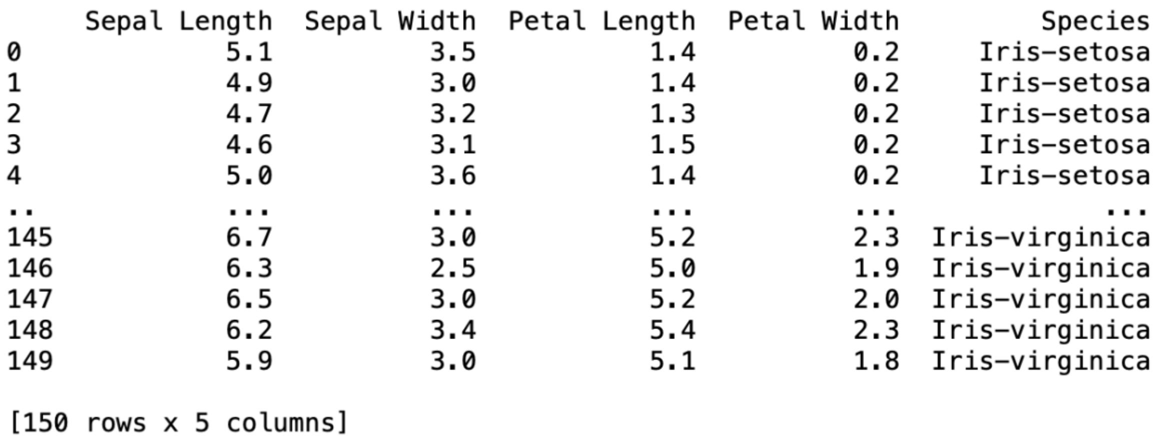
K-means is a centroid-based or distance-based technique in which the distances between points are calculated to allocate a point to a cluster. Each cluster in K-Means is paired with a centroid. The algorithm's main goal is to reduce the sum of distances between points and their corresponding cluster centroid.

### Problem:

Given the Iris dataset, the goal of the project is to implement K-means Clustering on the given dataset and analyze the results.

### Data:

The Iris dataset may be obtained on the UCI Machine Learning Repository and was used in R.A. Fisher's classic 1936 study, The Use of Multiple Measurements in Taxonomic Problems.



```
     Sepal Length  Sepal Width  Petal Length  Petal Width         Species
0             5.1          3.5           1.4          0.2     Iris-setosa
1             4.9          3.0           1.4          0.2     Iris-setosa
2             4.7          3.2           1.3          0.2     Iris-setosa
3             4.6          3.1           1.5          0.2     Iris-setosa
4             5.0          3.6           1.4          0.2     Iris-setosa
..            ...          ...           ...          ...             ...
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica

[150 rows x 5 columns]
```

It contains three iris species, each with 50 samples, as well as information about each flower.One flower species can be distinguished from the other two in a linear manner, but the other two cannot. The following are the columns in this dataset: Id, SepalLength(cm), SepalWidth(cm), PetalLength(cm), PetalWidth(cm) and Species.

## Method:

### K-means clustering:

One of the most basic and often used unsupervised machine learning algorithms is K-means clustering. Unsupervised algorithms make inferences from datasets based solely on input vectors,without referring to known, or labelled, outcomes.

The goal of K-means is straightforward: group comparable data points together to discover hidden patterns. K-means searches a dataset for a fixed number (k) of clusters to achieve this goal.
A cluster is a collection of data points that have been grouped together due to particular similarities.

You'll set a target number, k, for the number of centroids required in the dataset. A centroid is a fictional or real location that represents the cluster's center. By lowering the in-cluster sum of squares, each data point is assigned to one of the clusters, i.e., the K-means algorithm finds ‘k’ centroids and then assigns each data point to the cluster closest to it, keeping the centroids as tiny as possible. The average of the data, or determining the centroid, is what the 'means' in K-means refers to.

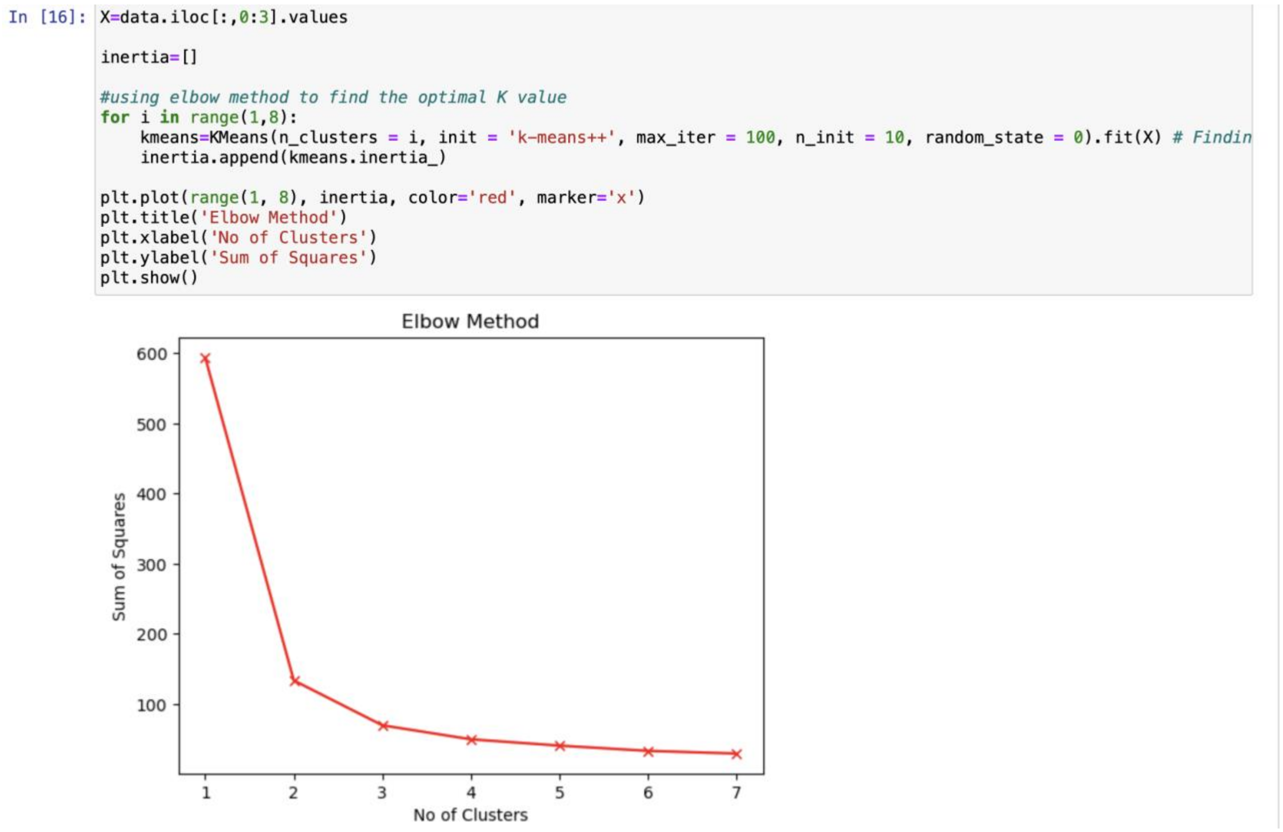
Given the inputs x1, x2, x3,…, xn and K value, the algorithm works as follows: Step 1: Select the K point as the center of the cluster called the centroid.
Step 2: Calculate the distance (Euclidean or Manhattan distance) to each centroid and match each $x_i$ to the nearest cluster.
Step 3: Find a new cluster center by averaging the allocated points.
Step 4: Repeat steps 2 and 3 until the cluster assignments do not change.

Finding the optimal number of clusters is an important part of this algorithm. The commonlyused method for finding the optimal K value is the elbow method.

## Elbow method:

The Elbow method actually changes the number of clusters (K) from 1 to 10, calculating WCSS (Within Cluster Sum of Square) for each value of K. WCSS is the sum of the squares of the distance between each point in the cluster and the center of gravity. When plotting the WCSS using the K value, the graph looks like an elbow. As the number of clusters increases, the WCSS value begins to decrease. The WCSS value is maximal when K = 1. Analyzing the graph reveals that at some point the graph changes rapidly and is in the shape of an elbow.

From this point on, the graph begins to move almost parallel to the X-axis. The corresponding K value in this regard is the optimal K value or the optimal number of clusters.



```
In [16]: X=data.iloc[:,0:3].values

         inertia=[]

         #using elbow method to find the optimal K value
         for i in range(1,8):
             kmeans=KMeans(n_clusters = i, init = 'k-means++', max_iter = 100, n_init = 10, random_state = 0).fit(X) # Findin
             inertia.append(kmeans.inertia_)

         plt.plot(range(1, 8), inertia, color='red', marker='x')
         plt.title('Elbow Method')
         plt.xlabel('No of Clusters')
         plt.ylabel('Sum of Squares')
         plt.show()
```

Using elbow method, it was found that the optimal K value is 3. So the algorithm has been implemented with k value 3. And since the dataset has 3 target classes, we can say that K value is3.

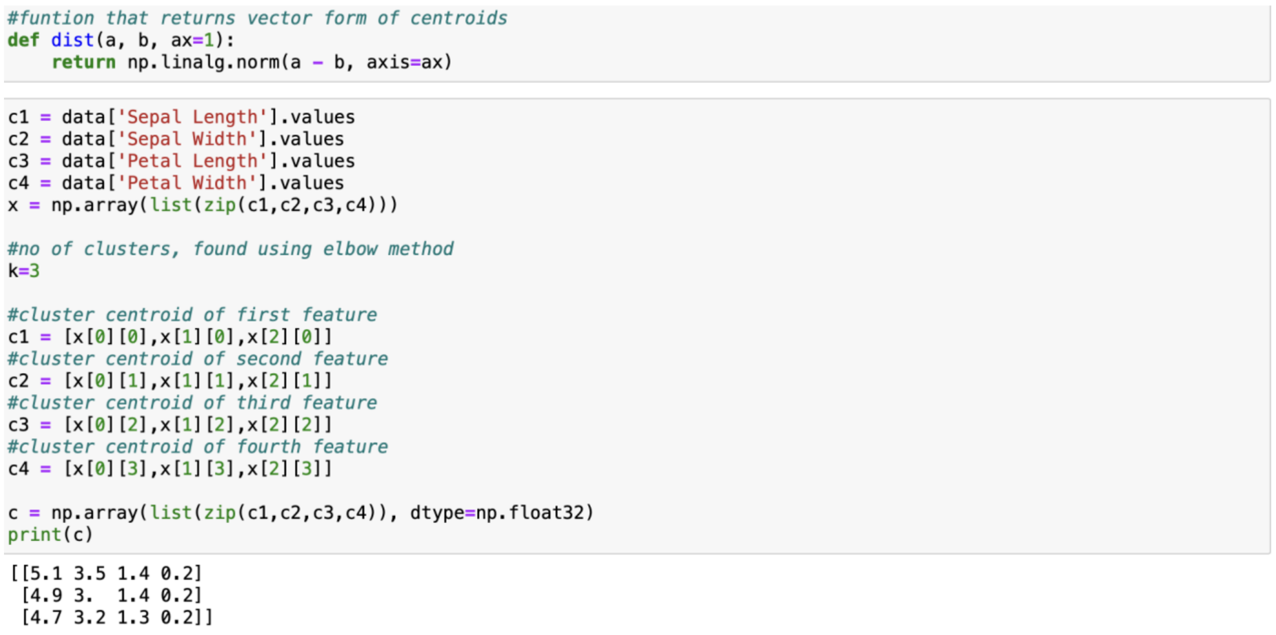
**Cluster Centroids:**

- c1 = [x[0][0],x[1][0],x[2][0]]: This line of code initializes the cluster centroids for the first feature (i.e., 'Sepal Length') by selecting the first three data points from x and extracting the corresponding values for this feature. The resulting values are assigned to the variable c1.
- c2 = [x[0][1],x[1][1],x[2][1]]: This line of code initializes the cluster centroids for the second feature (i.e., 'Sepal Width') by selecting the first three data points from x and extracting the corresponding values for this feature. The resulting values are assigned to the variable c2.
- c3 = [x[0][2],x[1][2],x[2][2]]: This line of code initializes the cluster centroids for the third feature (i.e., 'Petal Length') by selecting the first three data points from x

and extracting the corresponding values for this feature. The resulting values are assigned to the variable c3.

- c4 = [x[0][3],x[1][3],x[2][3]]: This line of code initializes the cluster centroids for the fourth feature (i.e., 'Petal Width') by selecting the first three data points from x and extracting the corresponding values for this feature. The resulting values are assigned to the variable c4.

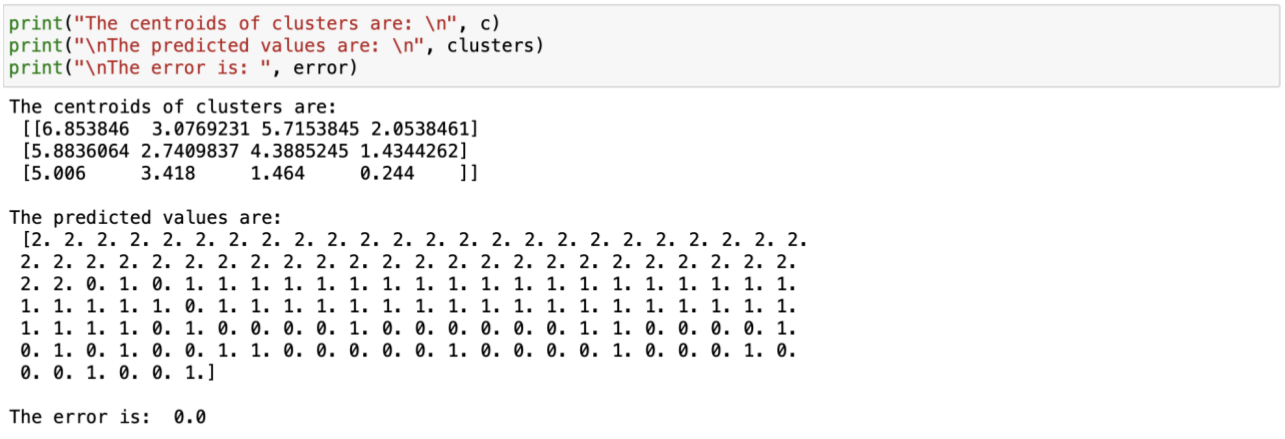
- c = np.array(list(zip(c1,c2,c3,c4)), dtype=np.float32): This line of code combines the four arrays c1, c2, c3, and c4 into a single 2D array c using the zip and list functions. Each row of c represents a single cluster centroid, with the values for each feature in a separate column. The dtype parameter is used to specify the data type of the array as np.float32. The resulting array c is then printed to the console.



```python
#funtion that returns vector form of centroids
def dist(a, b, ax=1):
    return np.linalg.norm(a - b, axis=ax)
```

```python
c1 = data['Sepal Length'].values
c2 = data['Sepal Width'].values
c3 = data['Petal Length'].values
c4 = data['Petal Width'].values
x = np.array(list(zip(c1,c2,c3,c4)))

#no of clusters, found using elbow method
k=3

#cluster centroid of first feature
c1 = [x[0][0],x[1][0],x[2][0]]
#cluster centroid of second feature
c2 = [x[0][1],x[1][1],x[2][1]]
#cluster centroid of third feature
c3 = [x[0][2],x[1][2],x[2][2]]
#cluster centroid of fourth feature
c4 = [x[0][3],x[1][3],x[2][3]]

c = np.array(list(zip(c1,c2,c3,c4)), dtype=np.float32)
print(c)
```

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]]
```



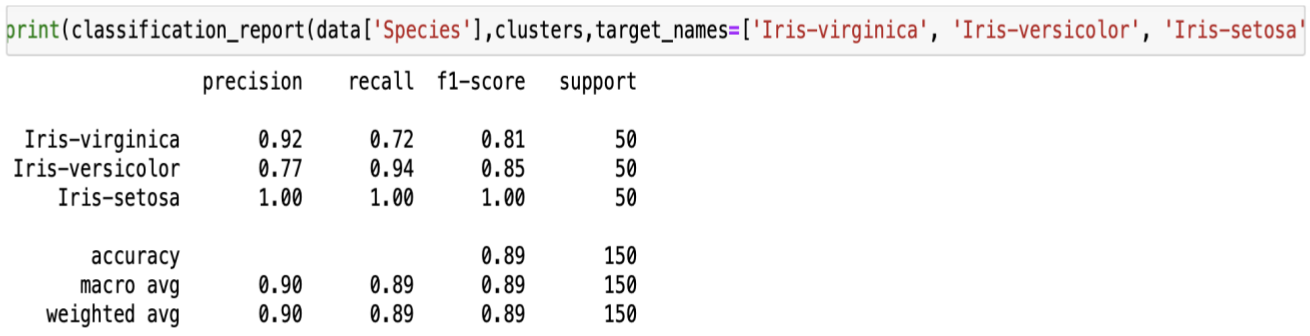
```
print("The centroids of clusters are: \n", c)
print("\nThe predicted values are: \n", clusters)
print("\nThe error is: ", error)
```

```
The centroids of clusters are:
 [[6.853846  3.0769231 5.7153845 2.0538461]
 [5.8836064 2.7409837 4.3885245 1.4344262]
 [5.006     3.418     1.464     0.244    ]]

The predicted values are:
 [2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1.
 0. 1. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0.
 0. 0. 1. 0. 0. 1.]

The error is:  0.0
```

## Results:

we are evaluating the performance of the k-means clustering algorithm on the Iris dataset, which has three classes: Iris-virginica, Iris-versicolor, and Iris-setosa. The precision, recall, and F1-score are computed for each class, as well as the support (i.e., the number of data points in each class). The accuracy represents the proportion of correctly assigned data points.

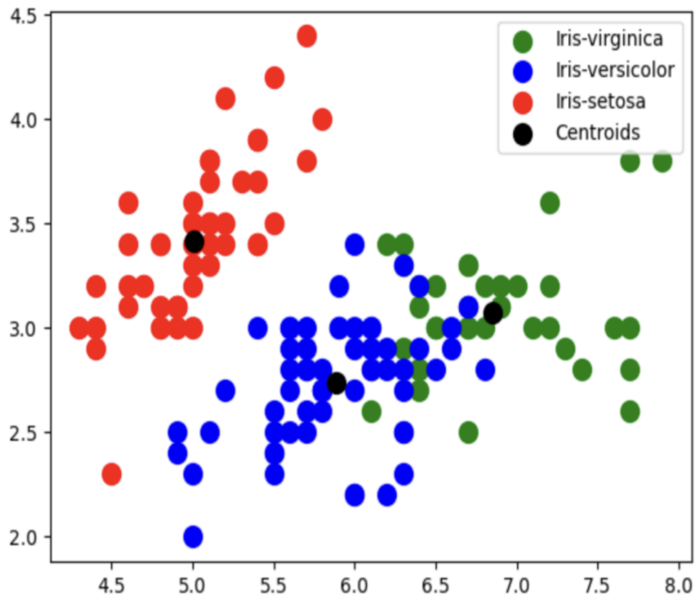
The implemented algorithm could predict the label of the given data points. The model has been well-trained with the dataset and can produce results with  an accuracy of 89%. From the graph, we can say that the 3 clusters have been performed with their respective centroids and classified accurately.



```
print(classification_report(data['Species'],clusters,target_names=['Iris-virginica', 'Iris-versicolor', 'Iris-setosa'
```

```
                 precision    recall  f1-score   support

 Iris-virginica       0.92      0.72      0.81        50
Iris-versicolor       0.77      0.94      0.85        50
    Iris-setosa       1.00      1.00      1.00        50

       accuracy                           0.89       150
      macro avg       0.90      0.89      0.89       150
   weighted avg       0.90      0.89      0.89       150
```

```
In [11]: #plotting the clusters
         plt.scatter(X[clusters == 0, 0], X[clusters == 0, 1], s = 100, c = 'green', label = 'Iris-virginica')
         plt.scatter(X[clusters == 1, 0], X[clusters == 1, 1], s = 100, c = 'blue', label = 'Iris-versicolor')
         plt.scatter(X[clusters == 2, 0], X[clusters == 2, 1], s = 100, c = 'red', label = 'Iris-setosa')

         #plotting the centroids of the clusters
         plt.scatter(c[:, 0], c[:,1], s = 100, c = 'black', label = 'Centroids')

         plt.legend()

Out[11]: <matplotlib.legend.Legend at 0x7fdfe89f2280>
```



## References:

1. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
2. https://towardsdatascience.com/k-means-explained-10349949bd10