Data Warehouse & Data Mining Assignment

K-Means Clustering Algorithm (Implementation)

**Group Members:**

Abhay Kumar (1711001)

Adarsh Kumar (1711002)

Amit Oraon (1711003)

Anibrato Adhikari (1711004)

Ranjan Kumar Choubey (1711025)

Tanveer Azam Ansari (1711038)

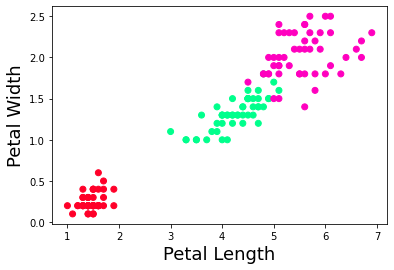
**What is K-Means Algorithm?**

K-Means Clustering Algorithm is an unsupervised Machine Learning algorithm which takes in data and clusters them in ‘k’ clusters as specified by user. This algorithm works on the principle of ‘k’ centroids who cluster data points on the basis of distance metrics calculated from each centroid to the data points.

For example, there is a data of traffic vehicles that have crossed a toll and the algorithm is to cluster the data into 3 clusters (cars, bikes, and trucks) but we do not tell the algorithm which is a car, or a bike or a truck. This is why the algorithm is called Unsupervised Learning Algorithm.

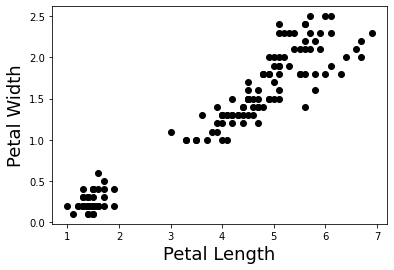
Now, let us see the dataset on which we will apply and test our algorithm code in python programming language.

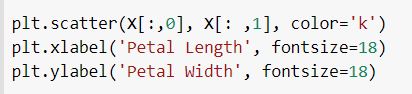
Iris dataset: In this dataset we have parameters Petal length and Petal width on the basis of which we can segregate the species of the flowers in to three categories. Now, if we see the three categories that we SHOULD get from the K-Means algorithm it is as follows:



**Algorithm Design and Workflow**

**Step 1:** First of all the unprocessed or the unclustered data could be visualised in the following way:





**Step 2:** We create a class for our K-Means algorithm and equip it with a constructor, a fit() function and a predict() method.

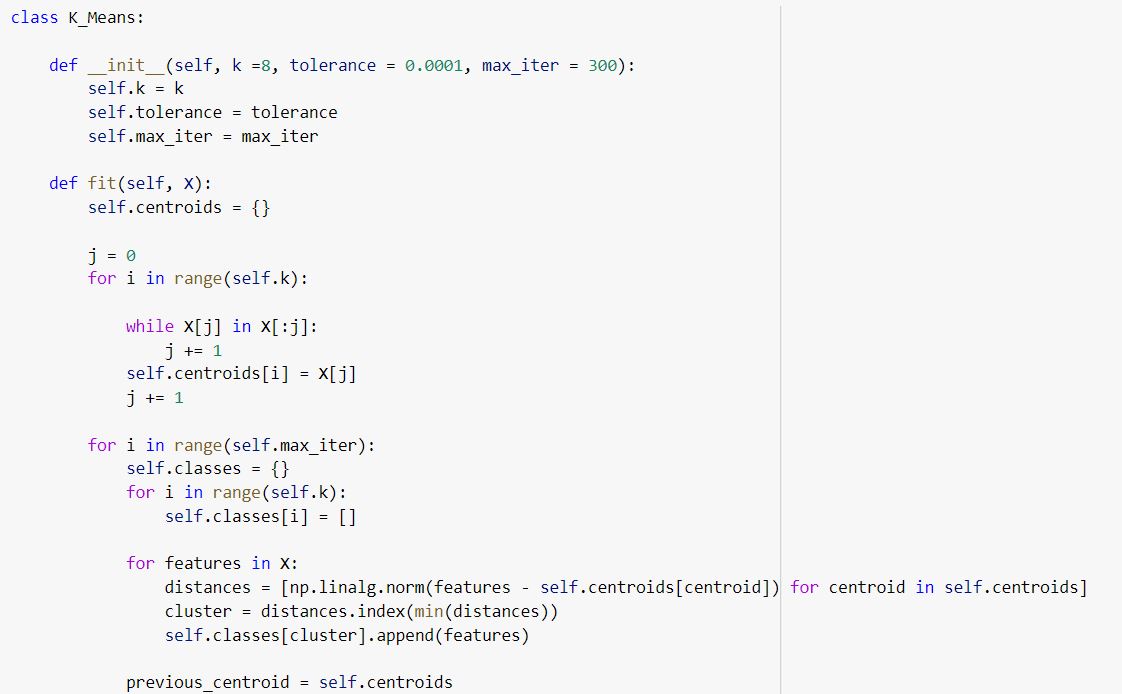
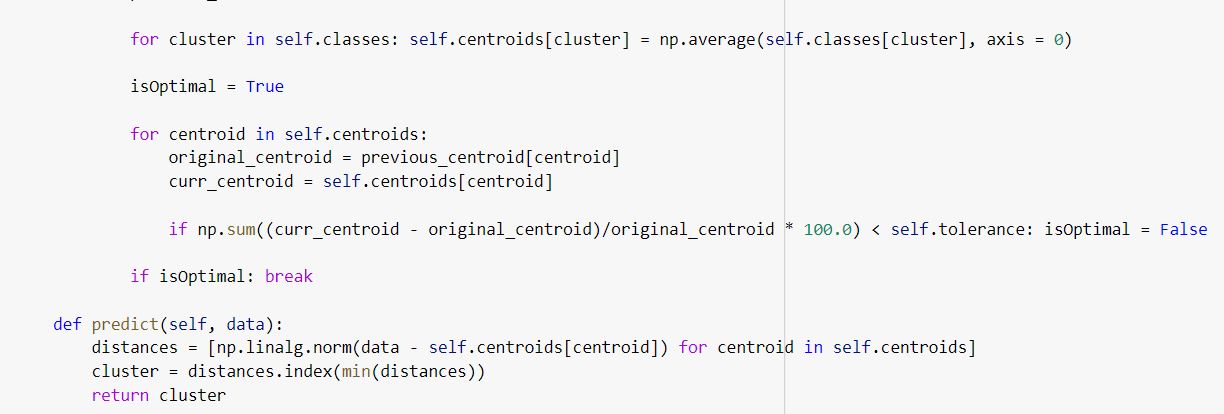
**Constructor:** This is used to initialise all the default and required parameters that we use during our algorithm.

* K value is given to the algorithm by the user. (By default, k is 8)
* Tolerance is the minimum shift in centroid position which can be considered for update loop.
* Max\_iter is the maximum times we shift the centroid while forming clusters.

**Fit()** function: It is used to do the actual work aka Clustering of the data by defining centroid positions and updation and distance metric calculation.

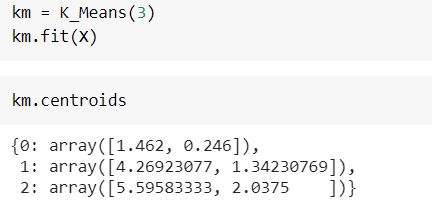
* Self.centroids[] store the position of the centroids in the algorithm.
* Self.classes{} store the data points that are concerned with each cluster.
* We use a nested loop structure to keep updating the data points in each cluster till the tolerance value is met or the max\_iterations are exhausted.
* We take each data point and compare the distance it has with all k centroids, the minimum distant centroid is assigned to the data point.

**Predict()** function: Once we have clustered a set of given data into k clusters, we can then enter any one data point and predict which cluster the stated data point belongs to.

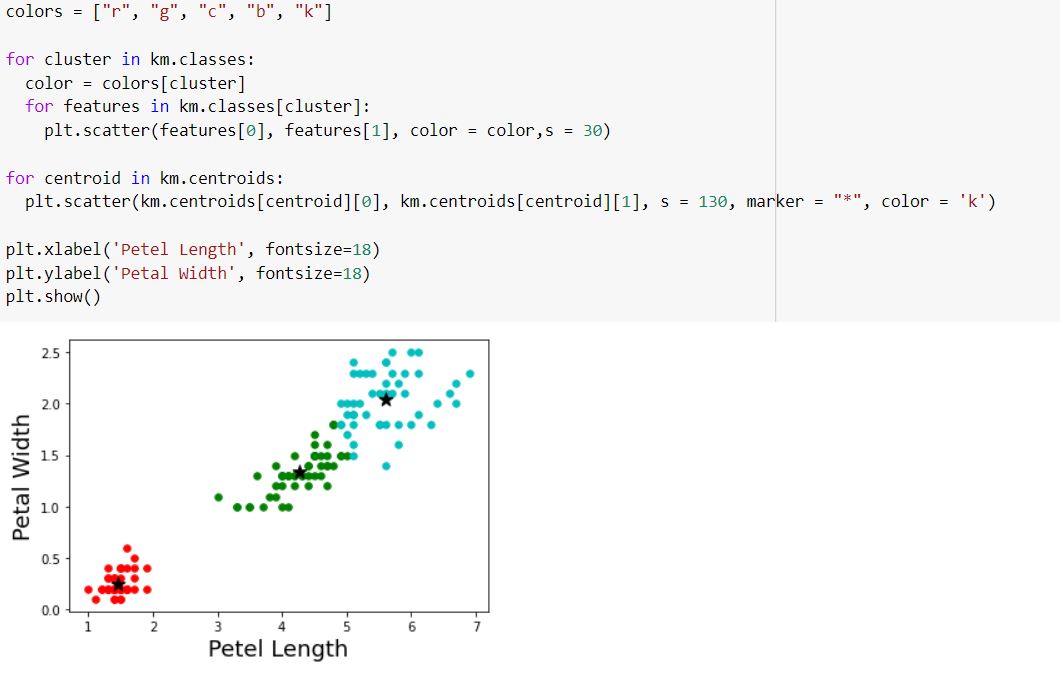


**Step 3:** We now call our algorithm using the fit() function and also display the final values of the centroids after clustering is done.

Here are the snapshots:



Now that we have clustered our data let us see what the final output for our algorithm is on a plot:

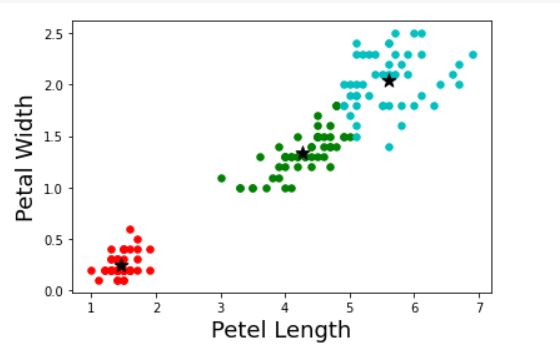


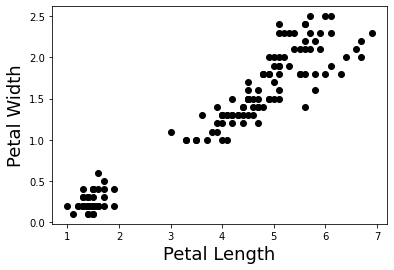
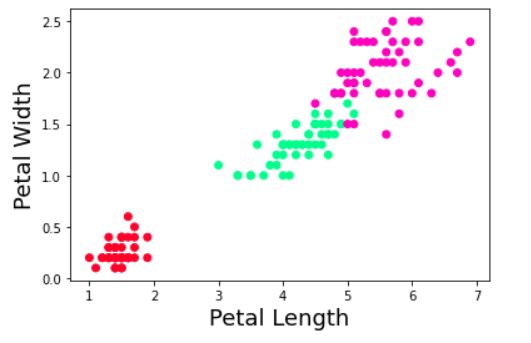
**Here, we have plotted our data points of each cluster and their respective centroids as star marks.**

**Comparison**

Let us see if our designed algorithm compares and competes with the expected outcome or not.

**Original Dataset Expected Output K-Means Output**





**CONCLUSION**

From the above results, we can conclude that our algorithm works as per requirements and provides an almost similar results as compared to original data.

**Note:** We have attached the code file and output snapshot along with this documentation.

**ACKNOWLEDGEMENT**

We would like to thank Pooja ma’am for giving us this assignment which made us examine and understand the interior functioning of a commonly used Machine Learning algorithm which helped us reverse engineer some of its functions and code them in our own way.

The assignment also taught us how to work and cooperate in a team digitally over various platform.

Thank you