Name: Arlene Antony D’costa
Seating Pin: 79
SUID: 594303899

CIS 662 HW3 Report

PART 1: Evaluation of number of components for Clustering

Scree Plot technique has been used to evaluate the number of clusters for the Kmeans clustering model.

Normalizing the data would lead to shrinking of the distance between the data points and cluster which might lead to incorrect results, also, all the training data feature set contains the same unit data, hence I have not used the technique of normalization/standardization. However, normalization can be done using Python’s Standard Scaler library.

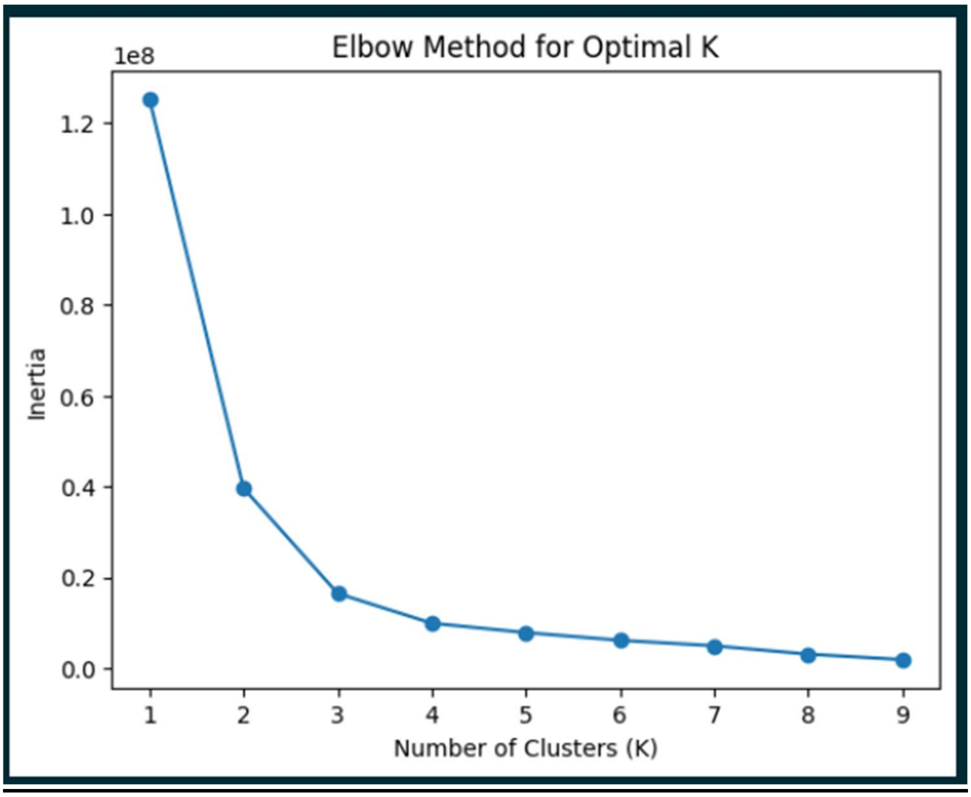
I have used Kmeans Clustering algorithm for predicting the values mentioned in the question.

Dataset used is 70-80.csv. Training and test data is split in 80:20 ratio. The columns selected as features are from cit_2017 to cit_2021 and the target column is cit_2022.

Scree plot is used for selecting the **Kmeans** clustering algorithm with the lowest inertia (The "inertia," which represents the sum of squared distances of data points to their assigned cluster centroids). I have used **Euclidean distance** for grouping as well as computing the nearest neighbour for a particular data point. The outputs for the same are attached below.

OUTPUT:

Using Euclidean distance:



Name: Arlene Antony D’costa
Seating Pin: 79
SUID: 594303899

RESULTS/CONCLUSION:

Data is continuous, numeric and the features belong to the same metric. Data is approximated to be normally distributed in spherical clusters hence **Euclidean distance** is considered for computing the nearest neighbour as well as the centroid.

We can see that at n=4, the steepness of the slope decreases and it stabilizes, hence 4 clusters creation is considered as suitable for the dataset provided.

PART 2: Prediction for the 2022 citation numbers for the test set, using the average difference magnitude to evaluate them.

Summary of Logic/Algorithm used for making predictions mentioned in the question:

For the prediction of point same as of nearest neighbour data point from the training set, first all the data points in the training set lying in the same cluster as the test set are calculated. Then the minimum distance is obtained for the nearest data points and the corresponding 2022 data point value is obtained.

For the prediction of data point same as the point closest to centroid data point, first all the points lying in the same cluster as the test data point are obtained. Then, the cluster centroid value is obtained from the cluster using the **kmeans.cluster_centers_** library and the nearest point to the centroid is obtained.Then the points in the 2022 test set are obtained same or closest to the data point obtained above.[Also distance is also computed between any point(train data point or test data point closest to centroid) and the corresponding value is also calculated as mentioned in the code file.

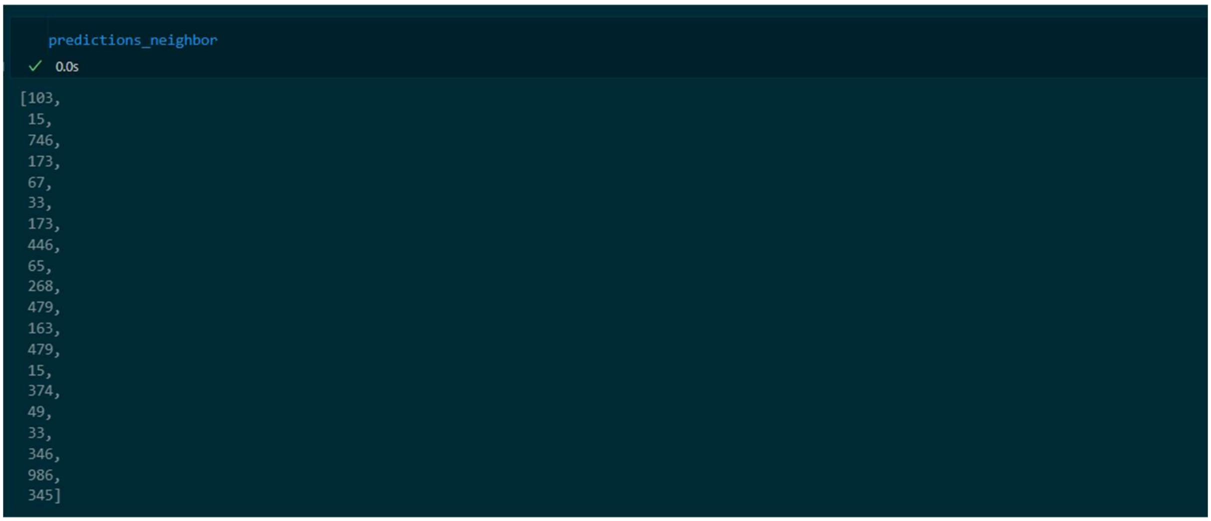
For the prediction of test data point same as the average of all other training data points within the same cluster as the target 2022 test data point, first all the data points in the training set lying in the same cluster as the test set are calculated. Then the average of all the data points in the trained cluster are calculated using np.mean(). Then, the corresponding data point from the 2022 test set is obtained.

At the end, the absolute magnitude difference is calculated between the actual 2022 test data points and the obtained predicted values.

OUTPUT:

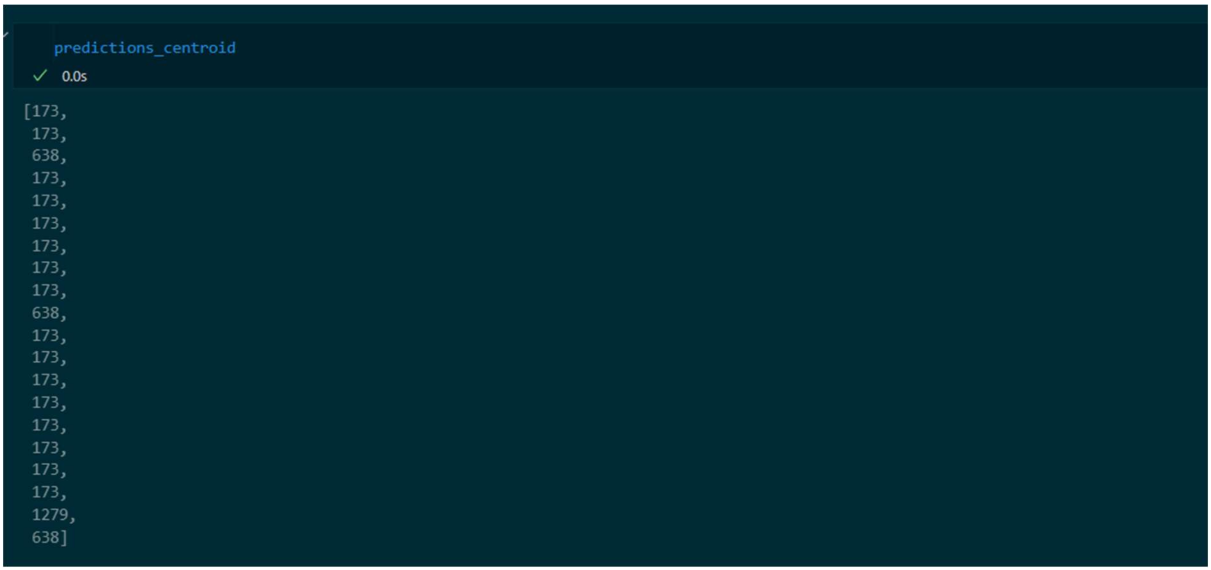
The test set consists of 20 records. The result is stored in the array. Also, the output for the average difference magnitude is attached below:

Prediction of point same as of nearest neighbour from training set within the same cluster as the test set:



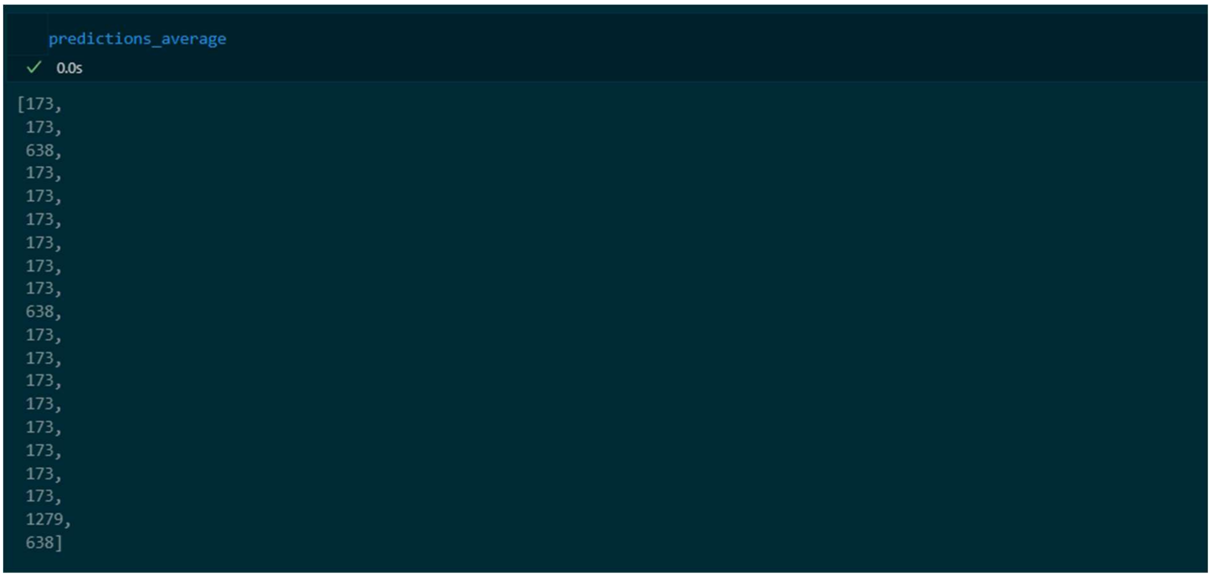
```
predictions_neighbor
✓ 0.0s
[103,
 15,
 746,
 173,
 67,
 33,
 173,
 446,
 65,
 268,
 479,
 163,
 479,
 15,
 374,
 49,
 33,
 346,
 986,
 345]
```

Prediction of point same as of data point closest to centroid for test set:[Test or training data point closest to the cluster is considered]



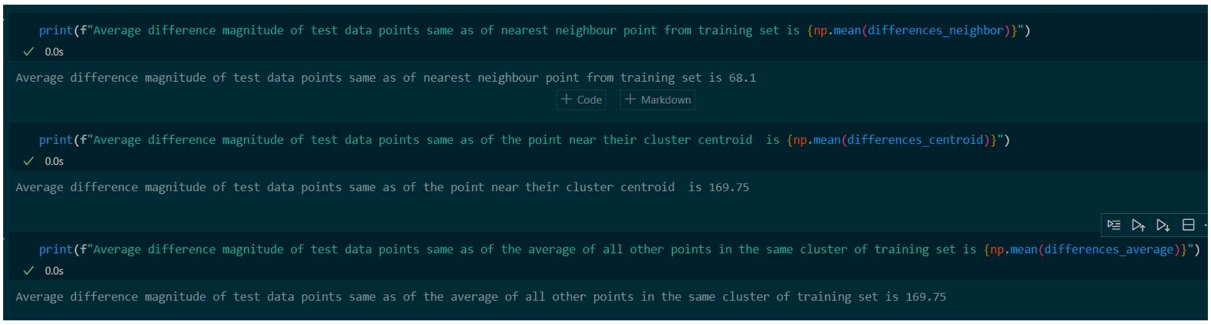
```
predictions_centroid
✓ 0.0s
[173,
 173,
 638,
 173,
 173,
 173,
 173,
 173,
 173,
 638,
 173,
 173,
 173,
 173,
 173,
 173,
 173,
 173,
 1279,
 638]
```

Prediction of point same as of average of all other data points from training set within the same cluster as the test set



```
predictions_average
✓ 0.0s
[173,
 173,
 638,
 173,
 173,
 173,
 173,
 173,
 173,
 638,
 173,
 173,
 173,
 173,
 173,
 173,
 173,
 173,
 1279,
 638]
```

Average absolute difference for all the 3 predictions:



```
print(f"Average difference magnitude of test data points same as of nearest neighbour point from training set is {np.mean(differences_neighbor)}")
✓ 0.0s
Average difference magnitude of test data points same as of nearest neighbour point from training set is 68.1
```

```
print(f"Average difference magnitude of test data points same as of the point near their cluster centroid  is {np.mean(differences_centroid)}")
✓ 0.0s
Average difference magnitude of test data points same as of the point near their cluster centroid  is 169.75
```

```
print(f"Average difference magnitude of test data points same as of the average of all other points in the same cluster of training set is {np.mean(differences_average)}")
✓ 0.0s
Average difference magnitude of test data points same as of the average of all other points in the same cluster of training set is 169.75
```

RESULTS/CONCLUSION:

As attached in the above image, the average difference magnitude is minimum for the test data point to the nearest data point from the training set within the same cluster of **68.1 units** as the distance is calculated with the nearest data point.

If the data point considered, lies at the border of the cluster, then there is a possibility of the distance between the data point and the point nearest the **cluster centroid** being larger. However, in the cluster, the points are comparatively closer to the centroid data points hence the absolute difference value obtained is **169.75 units.**

The average distance magnitude has the value of **169.75** for the average of all points in the training set lying within the same cluster as the test data points as there is a possibility of many data points (outlier points) within the cluster and cluster being spherical and large enough capturing many data points distributed at larger distances.

Average = Distance of all trained points / no of points

In Kmeans, centroid = Average

Since distance is calculated between training data points and the corresponding average and centroid the absolute difference value for **average and centroid values are same**.

Extra observation:

However, if the distance were to be captured between any point (i.e) training data already clustered or test data(predicted) and if the test data were to be closest to the centroid or trained data point closest to the centroid (trained data (fit) centroid) then the value obtained is 131.1 units. [More explanation can be found in the code]