National University of Computer and Emerging Sciences

Artificial Intelligence

Wajeeha Malik

18I-1587

DEGREE PROGRAM:	**BSSE/CS**
SECTION:	**D**
DATE OF SUBMISSION:	**May 24, 2021**
SUBMITTED TO:	**Sir Saad Salman**

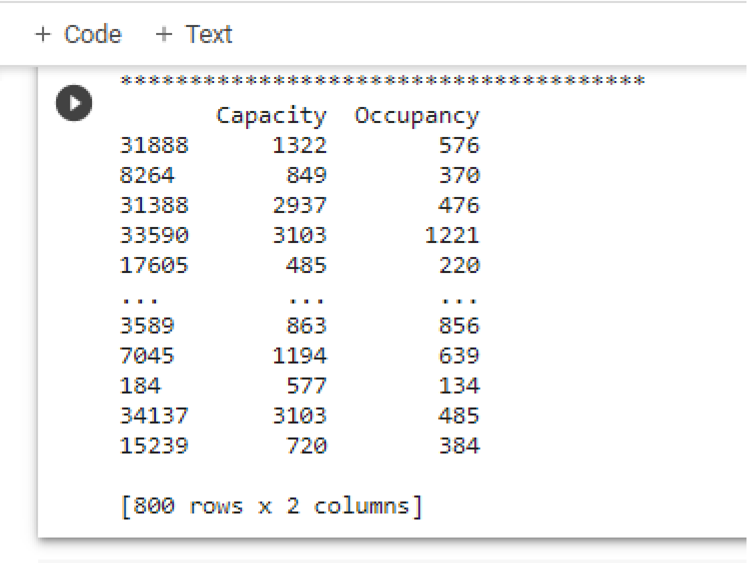
K NEAREST NEIGHBOUR

The k nearest neighbor classification (KNN) a distance-based algorithms based on measuring the distances between the test sample and the training samples to determine the final classification output.

STEPS TO IMPLEMENT KNN:

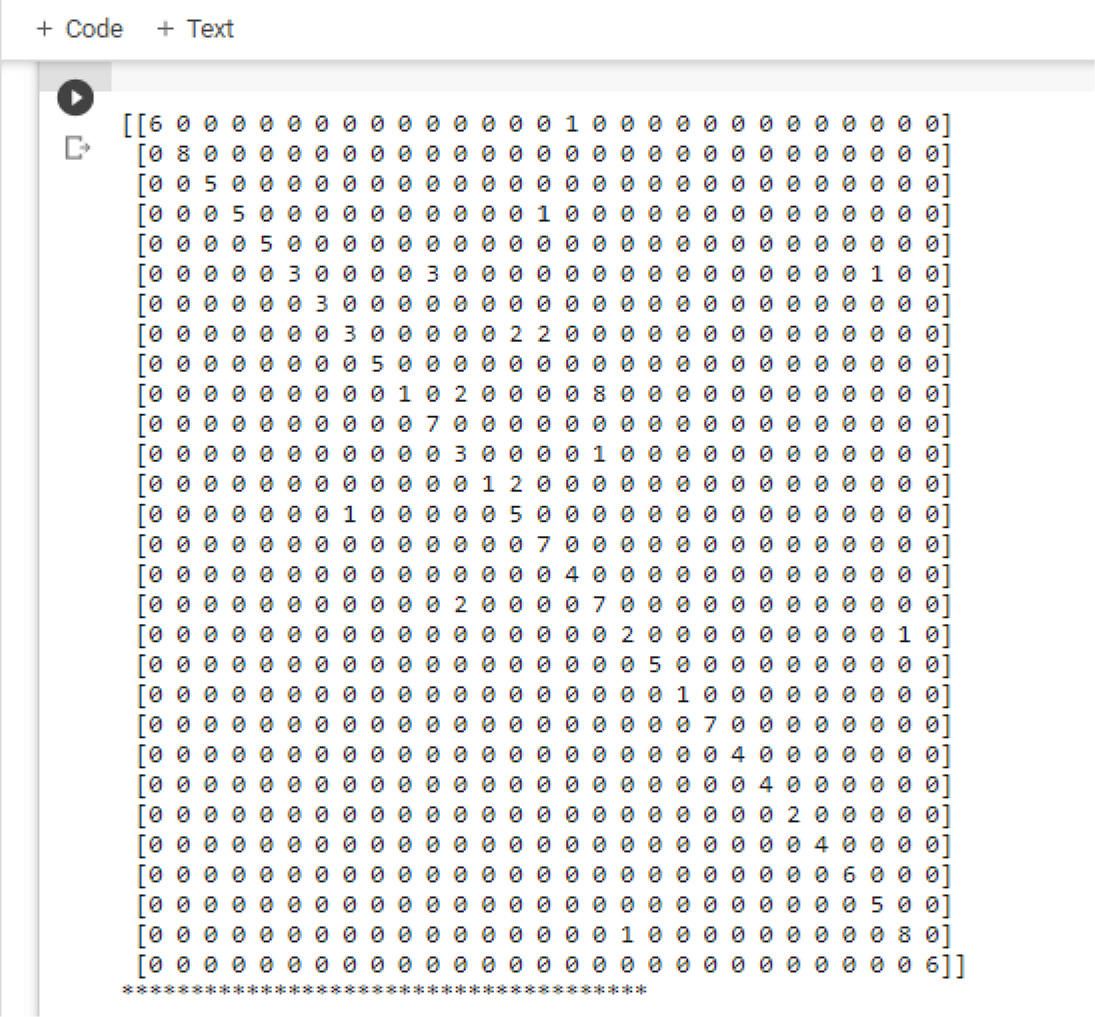
1. Imported dataset
2. Splitting the dataset in test and train sub data
3. Select the k numbers of neighbors
4. Find Euclidean distance of k neighbors
5. Sorting the distances list
6. Finding first k distances as nearest neighbors
7. From the k neighbors, count the number of the data points in each class.
8. Assign the new data points to that class for which the number of the neighbor is maximum.

DATASET:



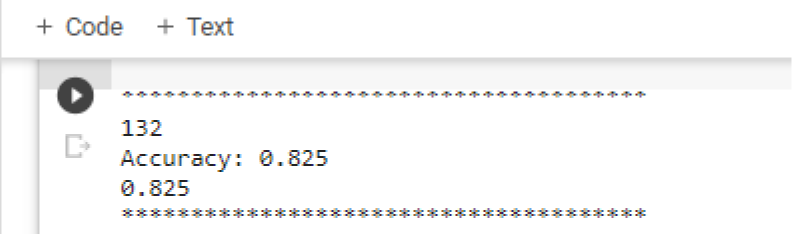
```
**************************************
       Capacity  Occupancy
31888      1322        576
8264        849        370
31388      2937        476
33590      3103       1221
17605       485        220
...         ...        ...
3589        863        856
7045       1194        639
184         577        134
34137      3103        485
15239       720        384

[800 rows x 2 columns]
```

CONFUSION METRICS:



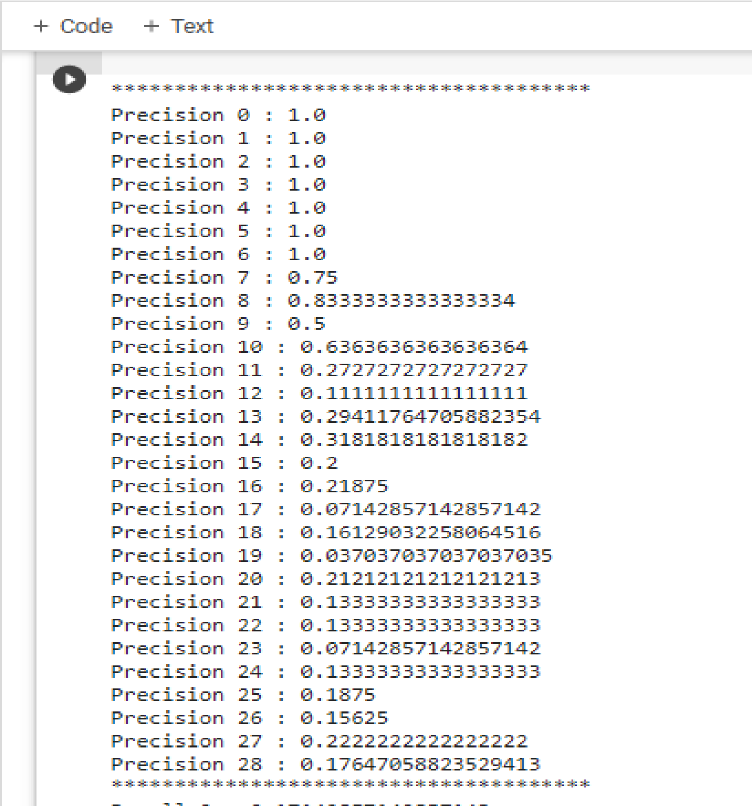
```
[[6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 3 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 3 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 2 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 8 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6]]
**************************************
```

ACCURACY:



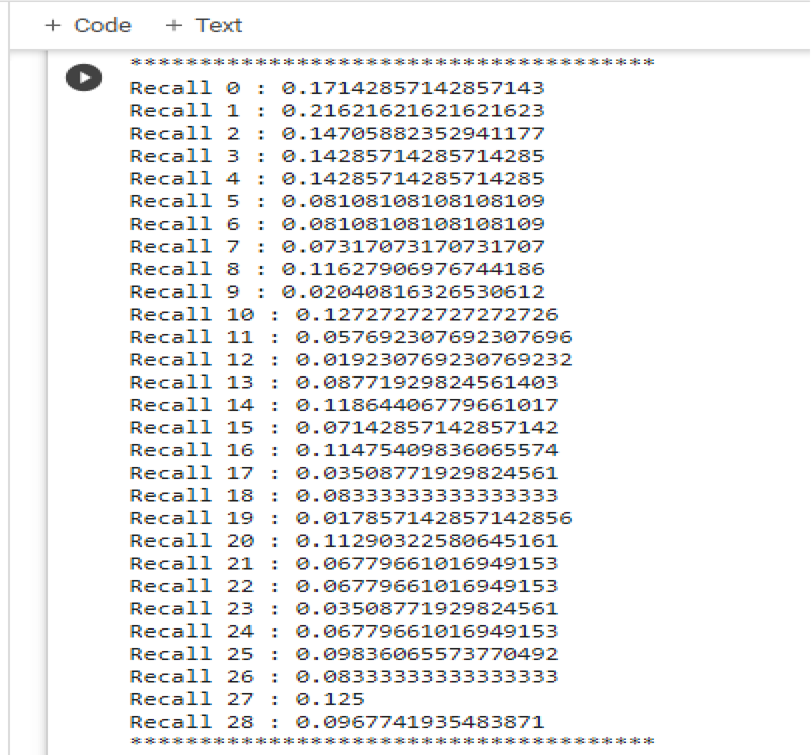
```
**************************************
132
Accuracy: 0.825
0.825
**************************************
```

PRECISION:



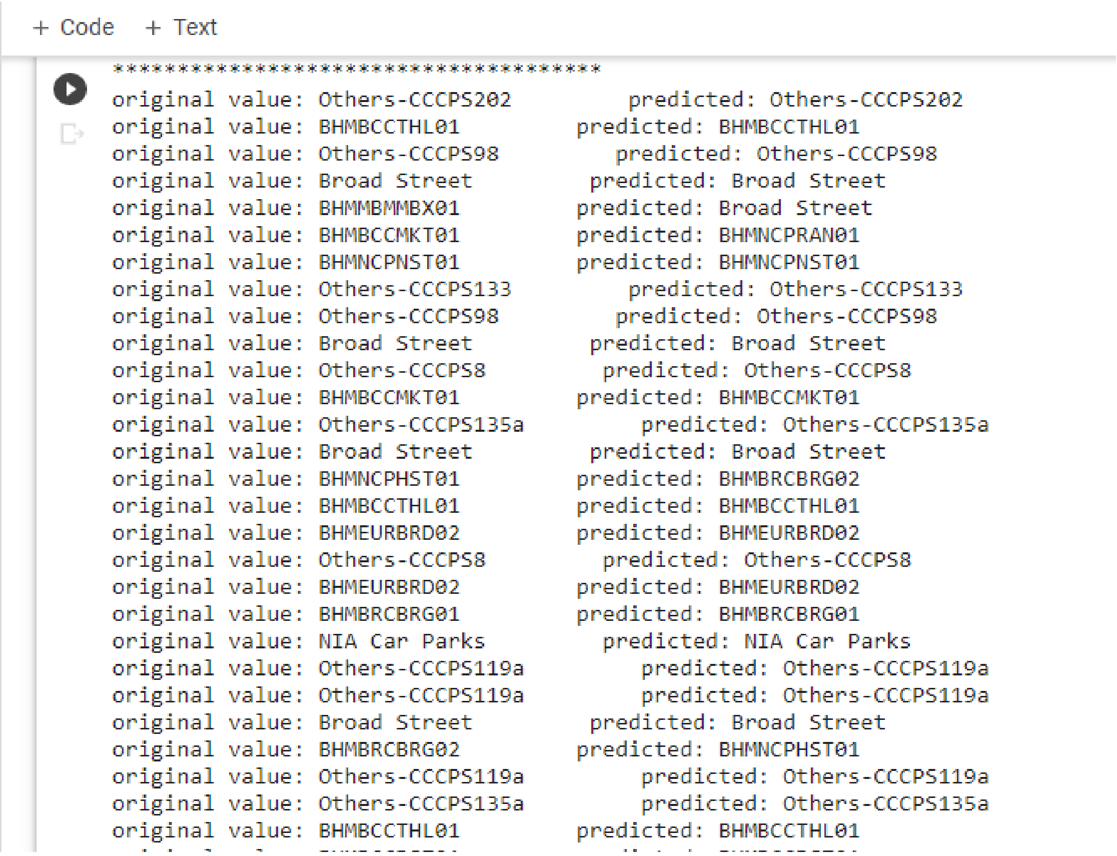
```
**************************************
Precision 0 : 1.0
Precision 1 : 1.0
Precision 2 : 1.0
Precision 3 : 1.0
Precision 4 : 1.0
Precision 5 : 1.0
Precision 6 : 1.0
Precision 7 : 0.75
Precision 8 : 0.8333333333333334
Precision 9 : 0.5
Precision 10 : 0.6363636363636364
Precision 11 : 0.2727272727272727
Precision 12 : 0.1111111111111111
Precision 13 : 0.29411764705882354
Precision 14 : 0.3181818181818182
Precision 15 : 0.2
Precision 16 : 0.21875
Precision 17 : 0.07142857142857142
Precision 18 : 0.16129032258064516
Precision 19 : 0.037037037037037035
Precision 20 : 0.21212121212121213
Precision 21 : 0.13333333333333333
Precision 22 : 0.13333333333333333
Precision 23 : 0.07142857142857142
Precision 24 : 0.13333333333333333
Precision 25 : 0.1875
Precision 26 : 0.15625
Precision 27 : 0.2222222222222222
Precision 28 : 0.17647058823529413
**************************************
```

RECALL:



```
*****************************************
Recall 0 : 0.17142857142857143
Recall 1 : 0.21621621621621623
Recall 2 : 0.14705882352941177
Recall 3 : 0.14285714285714285
Recall 4 : 0.14285714285714285
Recall 5 : 0.08108108108108109
Recall 6 : 0.08108108108108109
Recall 7 : 0.07317073170731707
Recall 8 : 0.11627906976744186
Recall 9 : 0.02040816326530612
Recall 10 : 0.12727272727272726
Recall 11 : 0.057692307692307696
Recall 12 : 0.019230769230769232
Recall 13 : 0.08771929824561403
Recall 14 : 0.11864406779661017
Recall 15 : 0.07142857142857142
Recall 16 : 0.11475409836065574
Recall 17 : 0.03508771929824561
Recall 18 : 0.08333333333333333
Recall 19 : 0.017857142857142856
Recall 20 : 0.11290322580645161
Recall 21 : 0.06779661016949153
Recall 22 : 0.06779661016949153
Recall 23 : 0.03508771929824561
Recall 24 : 0.06779661016949153
Recall 25 : 0.09836065573770492
Recall 26 : 0.08333333333333333
Recall 27 : 0.125
Recall 28 : 0.0967741935483871
*****************************************
```

PREDICTED AND ORIGINAL CLASSIFICATION:



```
**************************************
original value: Others-CCCPS202        predicted: Others-CCCPS202
original value: BHMBCCTHL01         predicted: BHMBCCTHL01
original value: Others-CCCPS98        predicted: Others-CCCPS98
original value: Broad Street         predicted: Broad Street
original value: BHMMBMMBX01         predicted: Broad Street
original value: BHMBCCMKT01         predicted: BHMNCPRAN01
original value: BHMNCPNST01         predicted: BHMNCPNST01
original value: Others-CCCPS133        predicted: Others-CCCPS133
original value: Others-CCCPS98        predicted: Others-CCCPS98
original value: Broad Street         predicted: Broad Street
original value: Others-CCCPS8        predicted: Others-CCCPS8
original value: BHMBCCMKT01         predicted: BHMBCCMKT01
original value: Others-CCCPS135a        predicted: Others-CCCPS135a
original value: Broad Street         predicted: Broad Street
original value: BHMNCPHST01         predicted: BHMBRCBRG02
original value: BHMBCCTHL01         predicted: BHMBCCTHL01
original value: BHMEURBRD02         predicted: BHMEURBRD02
original value: Others-CCCPS8        predicted: Others-CCCPS8
original value: BHMEURBRD02         predicted: BHMEURBRD02
original value: BHMBRCBRG01         predicted: BHMBRCBRG01
original value: NIA Car Parks         predicted: NIA Car Parks
original value: Others-CCCPS119a        predicted: Others-CCCPS119a
original value: Others-CCCPS119a        predicted: Others-CCCPS119a
original value: Broad Street         predicted: Broad Street
original value: BHMBRCBRG02         predicted: BHMNCPHST01
original value: Others-CCCPS119a        predicted: Others-CCCPS119a
original value: Others-CCCPS135a        predicted: Others-CCCPS135a
original value: BHMBCCTHL01         predicted: BHMBCCTHL01
```

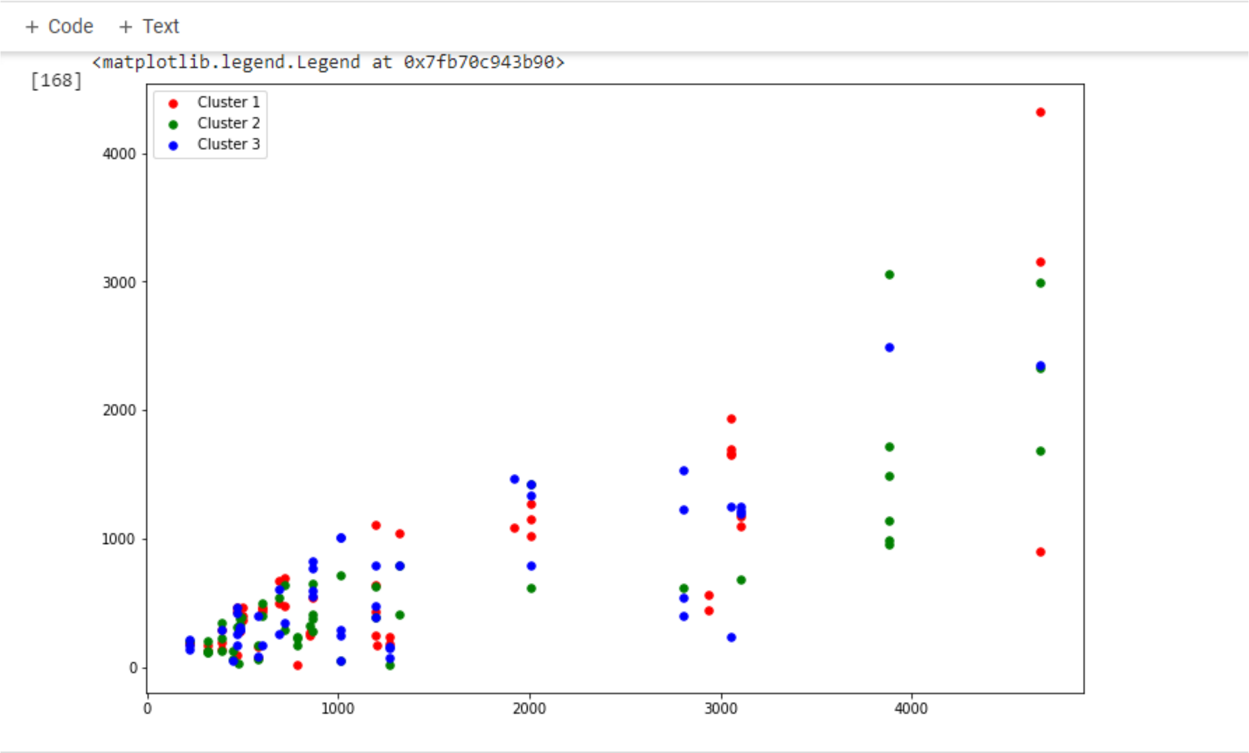
K MEAN CLUSTERING

K mean’s algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.

STEPS TO IMPLEMENT KNN:

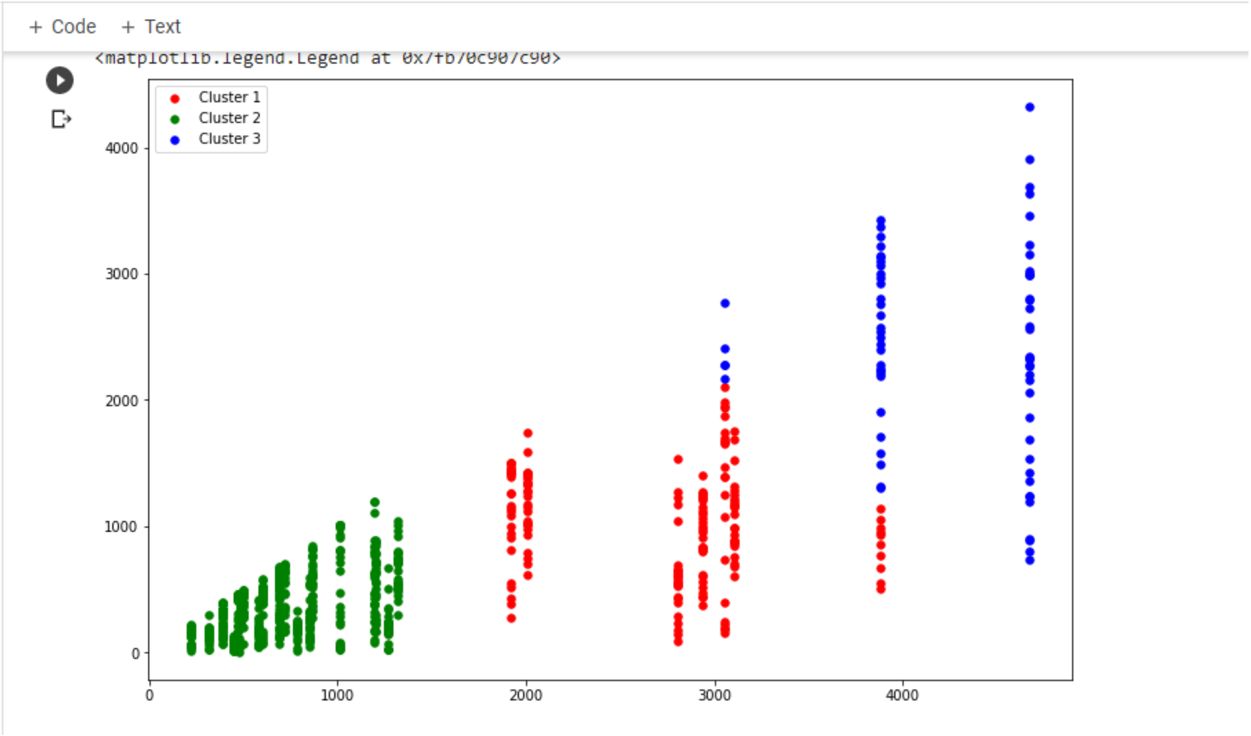
1. Select the number of clusters as k
2. Initialize centroids by shuffling the dataset
3. randomly select *K* data points for the centroids without replacing.
4. Keep iterating until there is no change to the centroids.
5. Compute the sum of the squared distance between data points and all centroids.
6. Assign each data point to the closest cluster (centroid).
7. Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

Initial input with random clusters



```
[168] <matplotlib.legend.Legend at 0x7fb70c943b90>
```

Clusters form by our k-mean



```
<matplotlib.legend.Legend at 0x7fb70c907c90>
```