Autonomus University of Queretaro

Faculty of Engineering

Master Science in Artificial Intelligence

k-Nearest Neighbors.

Student:
Iván Alejandro García Amaya

Instructor:
Phhd. Marco Antonio Aceves Fernández.

Course:
Machine Learning.

Contents

1 Introduction. 3

2 Methodology 5

2.1 K-means. 5

2.2 Estimating the number of clusters in a data set via the gap statistic. 5

2.2.1 Initialization Method. 10

2.2.2 Euclidean distance. 10

2.2.3 Update Centroide. 10

2.2.4 Evaluating K-means Clusters. 11

2.3 K-Nearest Neighbors (K-NN). 12

2.3.1 k-folds. 13

2.3.2 Euclidean distance. 13

2.3.3 Deciding k – The hyper parameter in KNN 14

2.3.4 Accuracy. 14

2.3.5 Recall. 14

2.3.6 Precision. 14

3 Results and discussion. 16

4 Conclusion and Future Work 22

5 References. 23

List of Figures

1 Subset. 3

2 Distribution Quantity-Price. 4

3 Gap - number of clusters. 8

4 Labeled data. 12

5 distribution of labeled data. 13

6 k-folds. 13

7 Accuracy for differents k's. 17

8 k-nn. 18

9 Confution matrix. 19

10 Clusters k-means. 20
11 Sum of Squared Error(SSE). 21

List of Tables

1 Accuracy - test. 17

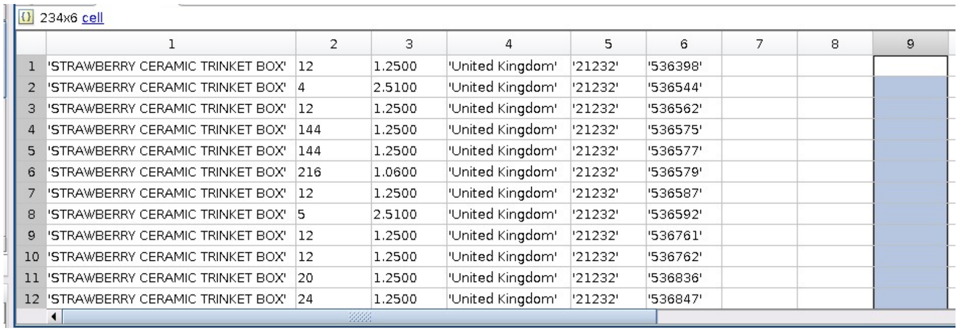
1 Introduction.

In this work an introduction to the Clousterig k-means algorithm is presented, taking into account the Principal Component Analysis work, starting from the selection of subsets, seeking to have a better distributed dataset, as shown below.

Example of missing value: 14341
Quantity : 3
Price : NA
Description : ‘GREEN REGENCY TEACUP AND SAUCER’
Country : ‘United Kingdom’
Stock Code : ‘15658’

- We withdraw products with different description: ’SANDALWOOD FAN’, ’POPCORN HOLDER’.

After implementing this series of steps, the size of the set decreases to a subset of 1014 as shown in figure 1.



234x6 cell

	1	2	3	4	5	6	7	8	9
1	'STRAWBERRY CERAMIC TRINKET BOX'	12	1.2500	'United Kingdom'	'21232'	'536398'			
2	'STRAWBERRY CERAMIC TRINKET BOX'	4	2.5100	'United Kingdom'	'21232'	'536544'			
3	'STRAWBERRY CERAMIC TRINKET BOX'	12	1.2500	'United Kingdom'	'21232'	'536562'			
4	'STRAWBERRY CERAMIC TRINKET BOX'	144	1.2500	'United Kingdom'	'21232'	'536575'			
5	'STRAWBERRY CERAMIC TRINKET BOX'	144	1.2500	'United Kingdom'	'21232'	'536577'			
6	'STRAWBERRY CERAMIC TRINKET BOX'	216	1.0600	'United Kingdom'	'21232'	'536579'			
7	'STRAWBERRY CERAMIC TRINKET BOX'	12	1.2500	'United Kingdom'	'21232'	'536587'			
8	'STRAWBERRY CERAMIC TRINKET BOX'	5	2.5100	'United Kingdom'	'21232'	'536592'			
9	'STRAWBERRY CERAMIC TRINKET BOX'	12	1.2500	'United Kingdom'	'21232'	'536761'			
10	'STRAWBERRY CERAMIC TRINKET BOX'	12	1.2500	'United Kingdom'	'21232'	'536762'			
11	'STRAWBERRY CERAMIC TRINKET BOX'	20	1.2500	'United Kingdom'	'21232'	'536836'			
12	'STRAWBERRY CERAMIC TRINKET BOX'	24	1.2500	'United Kingdom'	'21232'	'536847'			

Figure 1: Subset.

In the figure 2, the distribution of the products is shown depending on their quantity and their price.

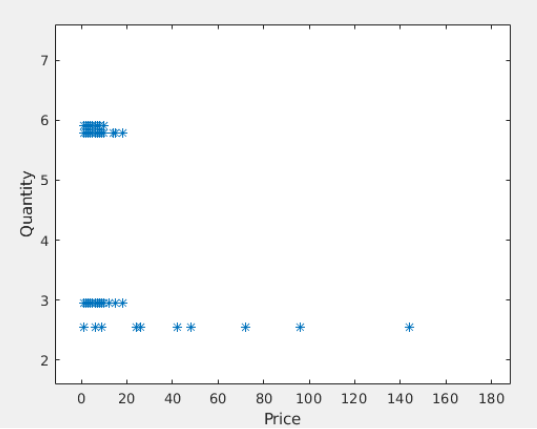


Figure 2: Distribution Quantity-Price.

2 Methodology

2.1 K-means.

2.2 Estimating the number of clusters in a data set via the gap statistic.

In the paper Estimating the number of clusters in a data set using the gap statistic, shows us the following explanation for the Gap Statistic technique

The data given by:

$\{x_{ij}\}$, i = 1,2,....,n, j = 1,2,...,p, consist fo p features measured on n independent observation.

$d_{ii'}$ denote the distane between observation i and i'. The most common choice for $d_{ii'}$ is the squared Euclidean distance. $\sum_j (x_{ij} - x_{i'j})^2$

Suppose that we have clustered the data into k clusters $C_1, C_2, ..., C_k$, with C_r denoting the indices of observations in cluster r, and $n_r = |C_r|$. Let Robert T (2001)

$$D_r = \sum_{i,i' \in C_r} d_{ii'} \tag{1}$$

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \tag{2}$$

So if the distance d is the squared Euclidean distance, then W_k is the pooled within-cluster sum of squares around the cluster means (the factor 2 makes this work exactly). the sample size n is suppressed in this notation.

In the paper say that the idea of the approach is to standardize the graph of log(W_k) by comparing it with its expecation under an appropriate null reference distribution of the data(the importance of the choice of an appropriate null model is demonstrated in Gordon (1996).). The estimate of the optimal number of cluster is then the value if k for which log(W_k) fall the farthest below this reference curve. Hence they define:

$$Gap_n(k) = E_n^*\{log(W_k)\} - log(W_k) \tag{3}$$

For compute the gap statics proceeds as follows

Step1: cluster the observed data, varying the total number of cluster frim k = 1,2,...,k, givin within-dispersion measure W_k, k = 1,2,..,k

Step2: generate B reference data sets, using the uniform prescription (a) or (b) above, and cluster each one giving within-depresion measures W_{kb}^* b = 1,2,...,B, k = 1,2,...,k Compute the (Estimated) gap Static

$$Gap(k) = \frac{1}{B}\sum_b log(W_{kb}^*) - log(W_k) \tag{4}$$

Step3: let

$$\bar{l} = \frac{1}{B}\sum_b log(W_{kb}^*) \tag{5}$$

compute the satandard deviation

$$sd_k = [\frac{1}{B}\sum_b \{log(W_{kb}^*) - \bar{l}\}^2]^{\frac{1}{2}} \tag{6}$$

and define

$$s_k = sd_k\sqrt{(1 + \frac{1}{B})} \tag{7}$$

Finally choose the number of cluster via

$$\hat{k} = \textit{ smallest k such that } Gap(k) \geq Gap(k+1) - s_{k+1} \tag{8}$$

If we take a closer look at the equation that the paper describes as the standard deviation.

$$\bar{l} = \frac{1}{B}\sum_b log(W_{kb}^*) \tag{9}$$

$$sd_k = [\frac{1}{B}\sum_b \{log(W_{kb}^*) - \bar{l}\}^2]^{\frac{1}{2}} \tag{10}$$

For the purpose of a better understanding we can observe the equation 11, it is the mathematical representation that is usually given for the standard deviation of a data vector.

$$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{11}$$

Then we can arrive at the following equality:

$$\overline{x} = \bar{l} = \frac{1}{B}\sum_{b} log(W^*_{kb}) \tag{12}$$

$$x_i = log(W^*_{kb}) \tag{13}$$

$$(x_i - \overline{x})^2 = (log(W^*_{kb}) - \bar{l})^2 \tag{14}$$

$$N = B \tag{15}$$

So recapping W_{kb} is a vector and $\bar{l}$ is a scalar.

S_k is given by:

$$s_k = sd_k\sqrt{(1 + 1/B)} \tag{16}$$

We can obtain Gap (k) and satisfy the condition given by

$$\hat{k} = \textit{ smallest } k \textit{ such that } Gap(k) \geq Gap(k+1) - s_{k+1} \tag{17}$$

For 10 clusters we have the following values for Gap and s_k

$gap = [-1.935, -6.306, -3.0176, -3.1732, -5.1542, -7.1977, -6.0642, -15.4204, -17.0975]$
$s_k = [0.0494, 0.1171, 0.0727, 0.0860, 0.2832, 0.1086, 0.1659, 0.1694, 0.2958]$

-3.0176

$Gap(k+1) - s_k(k+1) = -3.1732 - 0.0860$

$$Gap(k+1) - s_k(k+1) = -3.1036$$

$$-3.0176 \geq -3.1036$$

We can see in the figure 3, the best number of clusters is 3.

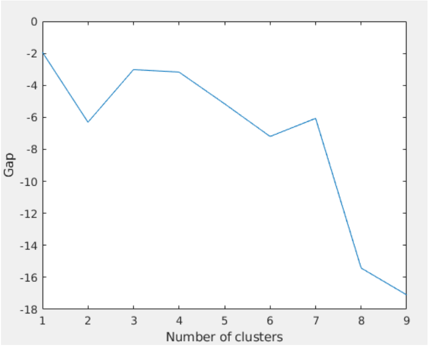


Figure 3: Gap - number of clusters.

Algorithm 1: Gap Statics.

```
input  : dataset
output: $Gap(k)$
$NClusters \leftarrow 10$;
$B \leftarrow 15$;
$clusters \leftarrow random\_initialization(dataset)$;
$values \leftarrow eval\_eucli\_dis(dataset, clusters)$;
$DrVector, n_r \leftarrow getDr\_nr(values)$;
$W_k \leftarrow withinCluster(DrVector, nr)$;
$SumlogW_{kb}, \leftarrow GetSumlogW_{kb}(Clusters, B)$;
$Gap \leftarrow GetGapK(B, SumlogW_{kb}, W_k)$;
```

Algorithm 2: $GetSumlogW_{kb}()$.

```
input  : Clusters, B
output: $SumlogW_{kb}$
for $i \rightarrow B$ do
    $NewData \leftarrow ReferenceDistribution()$;
    $Values \leftarrow eval\_eucli\_dis(NewData, Clusters)$;
    $DrVector, n_r \leftarrow getDr\_nr(values)$;
    $W_k^* \leftarrow withinCluster(DrVector, n_r)$;
    $logW_k \leftarrow log(W_k)$;
    $VectorW_{kb}(i, :) \leftarrow W_k$;
    $VectorlogW_{kb}(i, :) \leftarrow logW_k$;
end
for $j \rightarrow size(VectorW_{kb}, 2)$ do
    $a \leftarrow VectorlogW_{kb}(:, j)$;
    $aSum \leftarrow sum(a)$;
    $VectoraSum(1, j) \leftarrow aSum$;
end
```

2.2.1 Initialization Method.

input: K, set of points $x_1....x_n$.
Place centroides $c_1...c_k$ at random locations.
Repeat until convergence:

- for each point x_i :
  find nearest centroide c_i
  assign the point x_i for cluster j
  (Distance Euclidian)

- for each cluster j = 1...K:
  new centroide c_j = mean of all points x_i
  assigneed to cluster j in previous step

(Christopher M, 2008)

2.2.2 Euclidean distance.

$$d_1 = \frac{1}{N(N-1)} \sum_{i=1}^{N} -1 \sum_{j=i+1}^{N} ||x_i - x_j|| \tag{18}$$

To obtain the distance between two points p with coordinates (p_1, p_2) and the point q with coordinates (q_1, q_2), it is given by the equation (23)

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \tag{19}$$

(Charu A, 2014)

2.2.3 Update Centroide.

We use the equation 20, to calculate the n dimensional centroid point k n-dimensional points.

$$CP(x_1, x_2......x_k) = \frac{\sum_{i=1}^{k} x1st_i}{k}, \frac{\sum_{i=1}^{k} x2nd_i}{k}, \frac{\sum_{i=1}^{k} xnth_i}{k} \tag{20}$$

For find the centroide of 3 2D points, (2,4), (5,2) and (8,9)

$CP = \frac{2+5+8}{3}.\frac{4+2+9}{3} = (5,5)$

2.2.4 Evaluating K-means Clusters.

The Sum of Squared Error (SSE) is for each point the distance to the nearest cluster and it's defined by equation 21

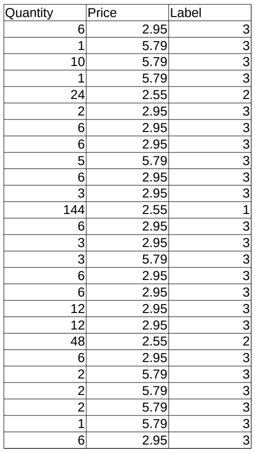
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x) \tag{21}$$

Algorithm 3: k-means.

```
input  : dataset
output: $Gap(k)$
$C = 4$;
$iteration = 100$;
$Ci \leftarrow Random\_initialization(data\_, C)$;
for  $i \rightarrow iteration$ do
    $values \leftarrow eval\_eucli\_dis(data\_, Ci)$;
    $G \leftarrow find\_specific\_cluster(values)$;
    $SSE \leftarrow Eval\_Sum\_of\_Squared\_Error(G, Ci)$;
    $Cluster \leftarrow UpdateCentroides(G, Ci)$;
end
```

2.3 K-Nearest Neighbors (K-NN).

K-NN is a supervised machine learning and the data must be labeled.
As can be seen in the figure 4, we have our data with their respective labels, it is necessary to mention that this data is not originally labeled, so it can be defined as a synthetic dataset, since the labels are placed depending on the clustering work carried out. by k-means, that is, the labels represent the cluster that was assigned to them, in addition to adding a bit of noise to the original dataset shown in fig 2.



Quantity	Price	Label
6	2.95	3
1	5.79	3
10	5.79	3
1	5.79	3
24	2.55	2
2	2.95	3
6	2.95	3
6	2.95	3
5	5.79	3
6	2.95	3
3	2.95	3
144	2.55	1
6	2.95	3
3	2.95	3
3	5.79	3
6	2.95	3
6	2.95	3
12	2.95	3
12	2.95	3
48	2.55	2
6	2.95	3
2	5.79	3
2	5.79	3
2	5.79	3
1	5.79	3
6	2.95	3

Figure 4: Labeled data.

In the figure 5, the new distribution of the data is observed, in its different classes in addition to the added noise.

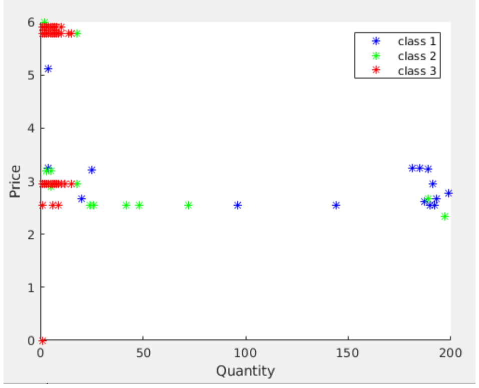
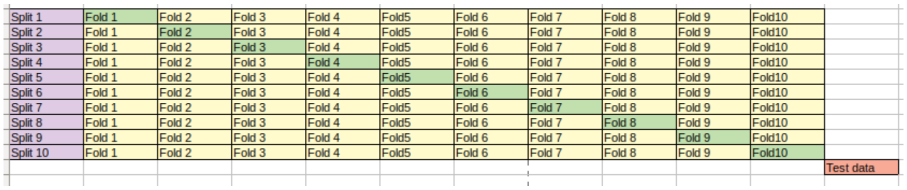


Figure 5: distribution of labeled data.

2.3.1 k-folds.

To find the best value of k, k-folds were used, as shown in the figure (ref), the original dataset was separated into two main folds, one for training and one for test, then the training one was separated into 10 different folds, this procedure was repeated for the different values of k, and obtaining the sum of the accuracy, the best value of k was determined.



Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 6	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 7	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 8	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 9	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
Split 10	Fold 1	Fold 2	Fold 3	Fold 4	Fold5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	
											Test data

Figure 6: k-folds.

2.3.2 Euclidean distance.

$$d_1 = \frac{1}{N(N-1)} \sum_{i=1}^{N} -1 \sum_{j=i+1}^{N} ||x_i - x_j|| \tag{22}$$

To obtain the distance between two points p with coordinates (p_1, p_2) and the point q with coordinates (q_1, q_2), it is given by the equation (23)

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \tag{23}$$

(Charu A, 2014)

2.3.3 Deciding k – The hyper parameter in KNN

k is nothing but the number of nearest neighbors to be selected to finally predict the outcome of new data point. Decision of choosing the k is very important, although there is no mathematical formula to decide the k.

We start with some random value of k and then start increases accuracy. once it start increasing the accuracy we stop there. Also overfitting case need to be taken care here. Hence we divide the data in three parts train, validation and test.

2.3.4 Accuracy.

Accuracy can be a misleading metric for imbalanced data sets. Consider a sample with 95 negative and 5 positive values. Classifying all values as negative in this case gives 0.95 accuracy score.

$$Accuracy = \frac{CorrectPrediction}{Records} \tag{24}$$

2.3.5 Recall.

Recall is the fraction of the total amount of relevant instances that were actually retrieved

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

2.3.6 Precision.

Precision is the fraction of relevant instances among the retrieved instances.

$$Precision = TPTP + FP \tag{26}$$

Algorithm 4: best k, k-NN.

```
input  : data
output: $best\_k$
$data_1 ..., data_K \leftarrow Kfolds$;
for $i \rightarrow k$ do
    $k \rightarrow i$;
    for $j \rightarrow K$ do
        $n \leftarrow j$;
        $data_a, data_b \leftarrow K\_folds(n, data_1)$;
        $prediction\_values \leftarrow get\_pred\_val(data_a, data_b, k)$;
        $acc \leftarrow get\_accuracy(prediction\_values, data_b)$;
        $Array\_acc \leftarrow acc$;
    end
    $ArraySumAcc \leftarrow sum(Array\_acc)$;
    $index \leftarrow max(ArraySumAcc)$;
    $best\_k \leftarrow index$;
end
```

Algorithm 5: Evaluate model best k, k-NN.

```
input  : data,k
output: $Accuracy$
$Numtest \leftarrow 5$;
for $i \rightarrow Numtest$ do
    $k \rightarrow i$;
    $data_a, data_b \leftarrow K\_folds(data)$;
    $prediction\_values \leftarrow get\_pred\_val(data_a, data_b, k)$;
    $acc \leftarrow get\_accuracy(prediction\_values, data_b)$;
end
```

3 Results and discussion.

As mentioned in the methodology section, k-folds were used to find the best value of K for the kNN algorithm, for example, the data was separated into 10 different subsets, the number 1 is used as a test and the 9 for the train, this procedure is done successively, for the 10 folds, this procedure is done with k = 1, .. n, where n = 50, as seen in figure 7, the accuracy for each value of k is the sum of the accuracy for the 10 k-folds, giving as the best result for k = 3 with the Accuracy = 8.822.

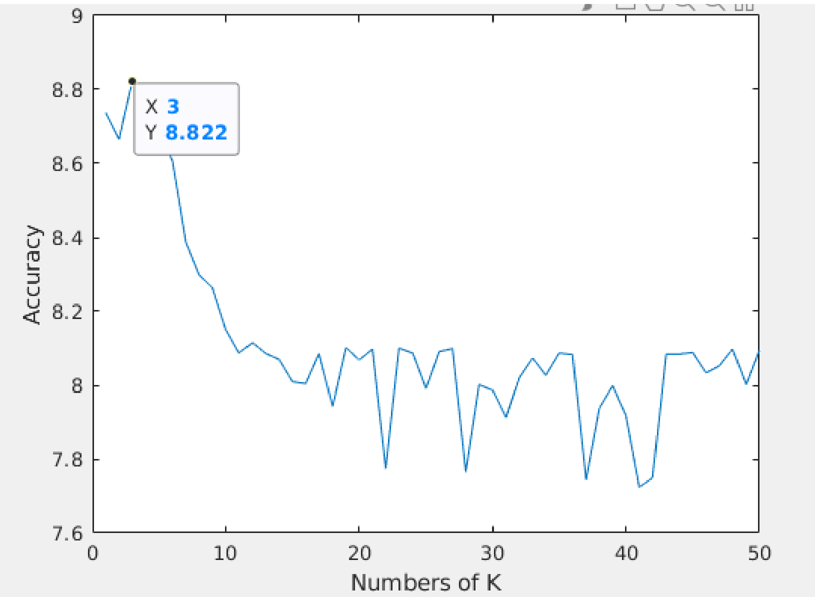


Figure 7: Accuracy for differents k's.

Once k was obtained, the model was evaluated some times, as shown in the table 1, for the 10 different tests, a high accuracy is shown.

Number of test	Accuracy	k
1	0.9859	3
2	0.9765	3
3	0.9577	3
4	0.9718	3
5	0.9765	3
6	0.9718	3
7	0.9671	3
8	0.9718	3
9	0.9906	3
10	0.9624	3

Table 1: Accuracy - test.

In the figure 8, it is observed the different classes, in addition to separating in train data '*' and test data 'o', the color represents the class to which, in the case of test data, the expected class belongs for each point.

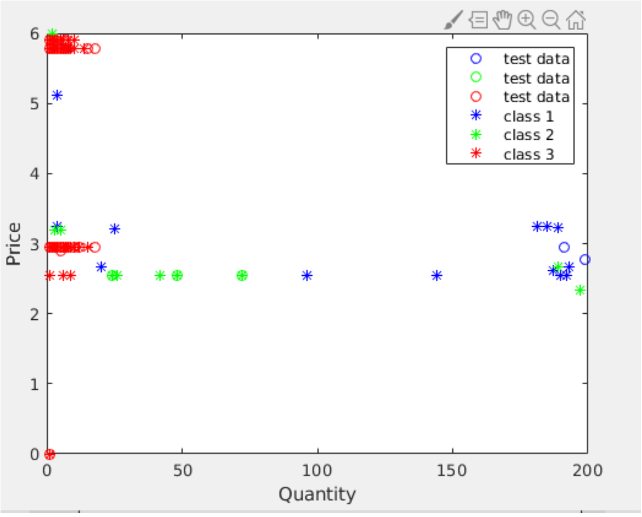
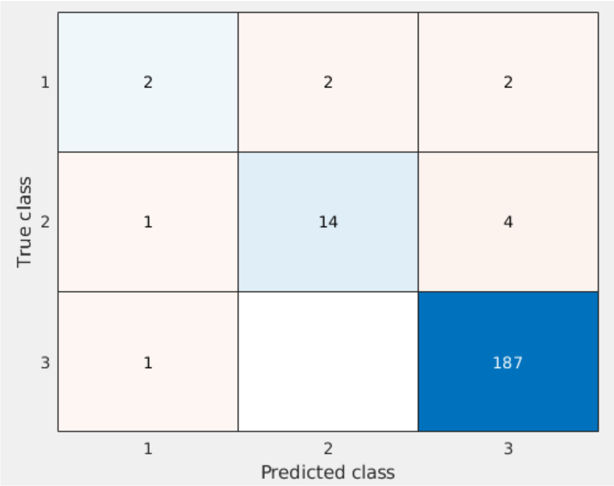


Figure 8: k-nn.

The figure 9 shows the confusion matrix for the expected and real data, where the number of components per class, in addition to a defined distribution, becomes more evident.



True class \ Predicted class	1	2	3
1	2	2	2
2	1	14	4
3	1		187

Figure 9: Confution matrix.

In the equations 27 28 29 , the values obtained from calculating recall of the different classes are observed, if we observe the confusion matrix figure 9, it is observed that this is the value of the class between the sum of the values of the columns.

$$RecallClass1 = \frac{2}{2+1+1} = 0.5 \tag{27}$$

$$RecallClass2 = \frac{14}{2+14} = 0.875 \tag{28}$$

$$RecallClass3 = \frac{187}{2+4+187} = 0.968911917 \tag{29}$$

In the equations 30 31 32, the result of calculating the precision of the different classes is observed, it is observed that it is the value of the class between the sum by rows of the other classes.

$$PrecisionClass1 = \frac{2}{2+2+2} = 0.333333 \tag{30}$$

$$PrecisionClass2 = \frac{14}{1+14+4} = 0.736842105 \tag{31}$$

$$PrecisionClass3 = \frac{187}{1 + 187} = 0.994680851 \tag{32}$$

In the figure 10, the clusters are shown after 100 iterations, it is observed that there are 3 different clusters, and the different points that compose it, these with the k-means algorithm.

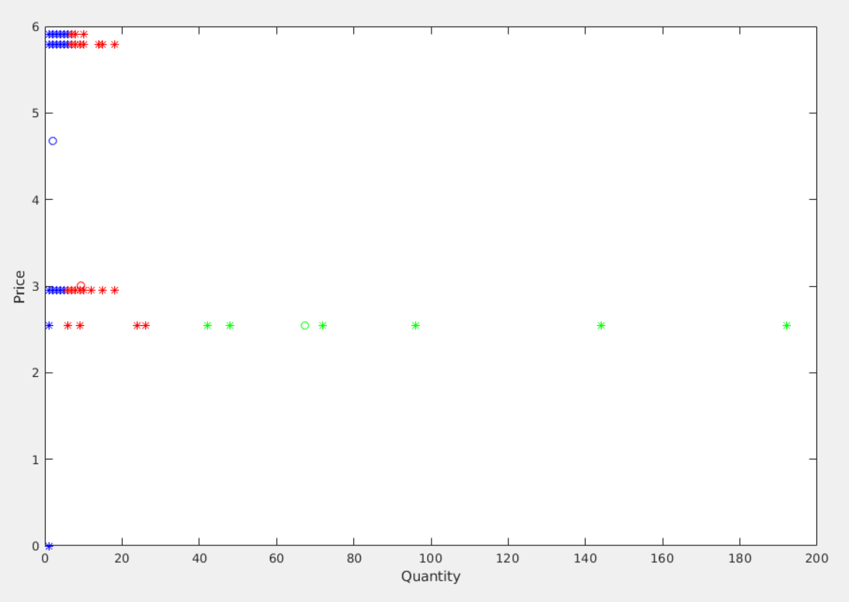


Figure 10: Clusters k-means.

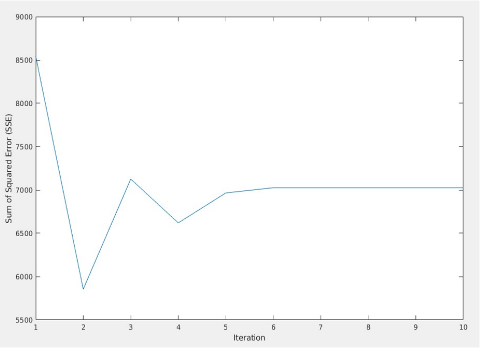


Figure 11: Sum of Squared Error(SSE).

4 Conclusion and Future Work

The first point that needs to be addressed is the question of the most obvious difference between the k-means and k-NN algorithm, the first in this work is approached as an unsupervised clustering algorithm, the second is approached as an algorithm of Supervised classification, delving into this the first algorithm does not have labels, in fact, if we look at the data, it largely belongs to a class that is a store product, but within this, we observe a different distribution, which is defined by price-quantity, then when k-means is applied, the result shows three possible clusters, with better results, starting from these results, the second algorithm can be approached, as was done in this work, these data were classified into three different classes starting from the three clusters given as a result by the previous algorithm.

A second point to mention, if we observe the algorithms internally, they have certain similarities such as the calculation of the distance, in both the Euclidean distance technique is used, then, if the first defined the classes based on this technique, then when we implement test in the second algorithm, we can relatively easily obtain accuracy .99, to avoid this situation, it was decided to add a little noise to the second dataset, as seen in the images already mentioned we observe small differences in the distribution of both datasets.

5 References.

Charu C. Aggarwal, Chandan K. Reddy . (2014). Data Clustering . New York: CRC Press.

Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. (2008). introduction to information Retrieval. United States of America: Cambridge University Press.

David Arthur, Segei Vassilvitskii. (2007). k-means++: The advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms, 8, 1027-1035.

Gordon, A. (1996) Null models in cluster validation. In From Data to Knowledge (eds W. Gaul and D. Pfeifer), pp. 32-44. New York: Springer.

Robert Tibshirani, Guenther Walther and Trevor Hastie. (2001). Estimating the number of clusters in a data set via the gap statistic. J. R. Statist. Soc. B, 63, 411-423.