Arinjay Jain (A20447307)

# CS 584-04: Machine Learning

Spring 2020 Assignment 2

## Question 1 (35 points)

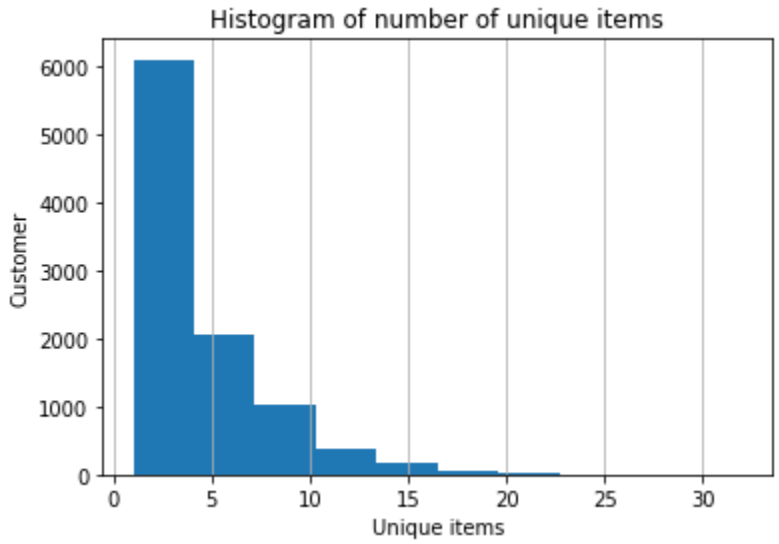
The file Groceries.csv contains market basket data. The variables are:

1. Customer: Customer Identifier
2. Item: Name of Product Purchased

After you have imported the CSV file, please discover association rules using this dataset. For your information, the observations have been sorted in ascending order by Customer and then by Item. Also, duplicated items for each customer have been removed.

a) (5 points) Create a data frame that contains the number of unique items in each customer's market basket. Draw a histogram of the number of unique items. What are the 25$^{th}$, 50$^{th}$, and the 75$^{th}$ percentiles of the histogram?

**Histogram**:




**25$^{th}$, 50$^{th}$, and the 75$^{th}$ percentiles of the histogram:**

**25th percentile: 2.0**
**50th percentile: 3.0**
**70th percentile: 6.0**

b) (10 points) We are only interested in the *k*-itemsets that can be found in the market baskets of at least seventy five (75) customers. How many itemsets can we find? Also, what is the largest *k* value among our itemsets?

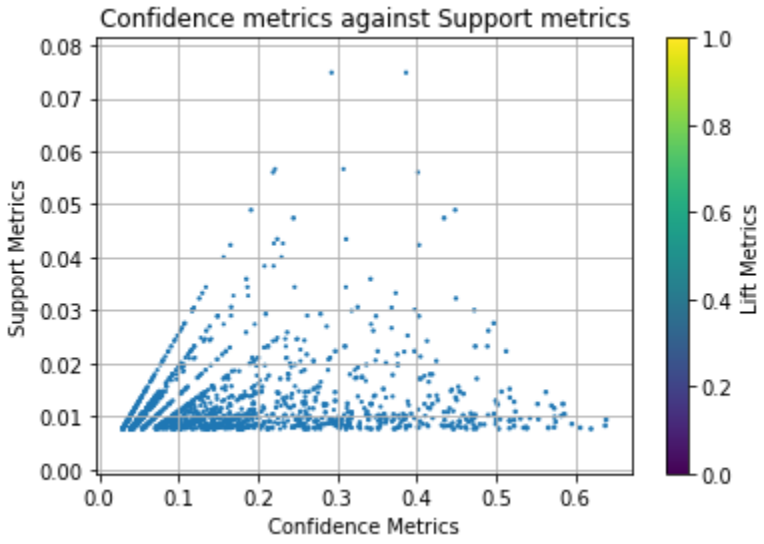
Ans. Number of item sets: **524;** Largest k value among our item sets = 4

c) (10 points) Find out the association rules whose Confidence metrics are greater than or equal to 1%. How many association rules can we find? Please be reminded that a rule must have a non-empty antecedent and a non-empty consequent. Please **do not** display those rules in your answer.

Ans. No. of association rules: **1228** with Minimum Threshold **= 0.01 (1%)**

d) (5 points) Plot the Support metrics on the vertical axis against the Confidence metrics on the horizontal axis for the rules you have found in (c). Please use the Lift metrics to indicate the size of the marker.

Ans. **Scatter Plot**




e) (5 points) List the rules whose Confidence metrics are greater than or equal to 60%. Please include their Support and Lift metrics.

>0.60

| antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|---|
| (butter, root vegetables) | (whole milk) | 0.012913 | 0.255516 | 0.008236 | 0.637795 | 2.496107 | 0.004936 | 2.055423 |
| (butter, yogurt) | (whole milk) | 0.014642 | 0.255516 | 0.009354 | 0.638889 | 2.500387 | 0.005613 | 2.061648 |
| (other vegetables, root vegetables, yogurt) | (whole milk) | 0.012913 | 0.255516 | 0.007829 | 0.606299 | 2.372842 | 0.004530 | 1.890989 |
| (tropical fruit, other vegetables, yogurt) | (whole milk) | 0.012303 | 0.255516 | 0.007626 | 0.619835 | 2.425816 | 0.004482 | 1.958317 |

## Question 2 (30 points)

The K-means algorithm works only with interval features. One way to apply the k-means algorithm to categorical features is to transform them into a new interval feature space. However, this approach can be very inefficient, and it does not produce good results.

For clustering categorical features, we should consider the K-modes clustering algorithm which extends the K-means algorithm by using different dissimilarity measures and a different method for computing cluster centers. See this article for more details. Huang, Z. (1997). "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining." In Proceedings of the *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, 1–8. New York: ACM Press.

Please implement the K-modes clustering method in Python and then apply the method to the cars.csv. Your input fields are these four categorical features: Type, Origin, DriveTrain, and Cylinders. **Please do not remove the missing or blank values in these four features**. Instead, consider these values as a separate category.

The cluster centroids are the modes of the input fields. In the case of tied modes, choose the lexically or numerically lowest one.

Suppose a categorical feature has observed values $v_1, \dots, v_p$. Their global frequencies (i.e., number of observations) are $f_1, \dots, f_p$. Please be noted that these global frequencies do not change with the cluster assignment. The distance metric between two values is $d(v_i, v_j) = 0$ if $v_i = v_j$. Otherwise, $d(v_i, v_j) = \frac{1}{f_i} + \frac{1}{f_j}$. The distance between any two observations is the sum of the distance metric of the four categorical features.

a) (5 points) What are the frequencies of the categorical feature Type?

**Sedan 262**
**SUV 60**
**Sports 49**
**Wagon 30**
**Truck 24**
**Hybrid 3**

b) (5 points) What are the frequencies of the categorical feature DriveTrain?
**FWD 226**
**RWD 110**
**AWD 92**

c) (5 points) What is the distance metric between 'Asia' and 'Europe' for Origin?
Frequency of Asia Origin: **158**
Frequency of Europe Origin: **123**
Distance metric between Asia and Europe is: **0.014459195224863643**

d) (5 points) What is the distance metric between Cylinders = 5 and Cylinders = Missing?

Distance metric between Cylinders '5' and cylinders 'Missing' is: **0.6428571428571428**

e) (5 points) Apply the K-modes method with **three clusters**. How many observations in each of these three clusters? What are the centroids of these three clusters?
Observations in 1st cluster: **254**
Observations in 2nd cluster: **112**
Observations in 3rd cluster: **62**

**Centroids**:
*[['Sedan' 'USA' 'FWD' '4.0']*
*['Sedan' 'Europe' 'RWD' '6.0']*
*['SUV' 'Asia' 'AWD' '6.0']]*

f) (5 points) Display the frequency distribution table of the Origin feature in each cluster.

Frequency Distribution Table

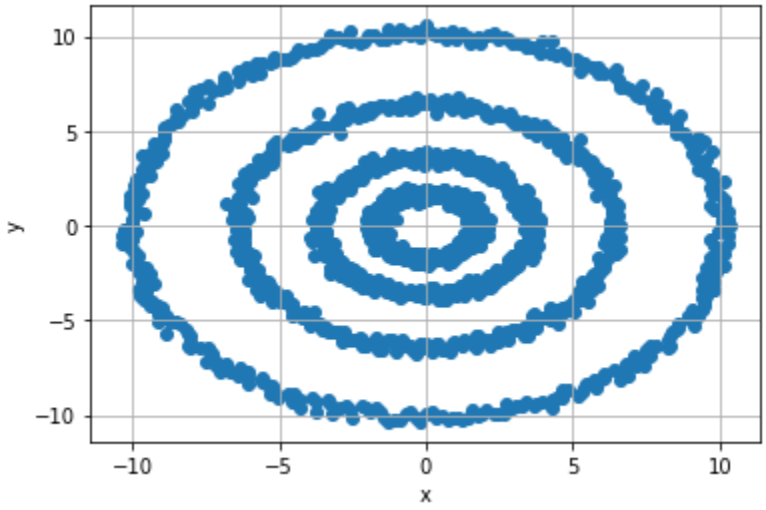
| clusters | Origin | |
|---|---|---|
| 0 | Asia | 98 |
| | Europe | 31 |
| | USA | 125 |
| 1 | Asia | 19 |
| | Europe | 82 |
| | USA | 11 |
| 2 | Asia | 41 |
| | Europe | 10 |
| | USA | 11 |

## Question 3 (35 points)

Apply the Spectral Clustering method to the FourCircle.csv. Your input fields are x and y. Wherever needed, specify random_state = 60616 in calling the KMeans function.

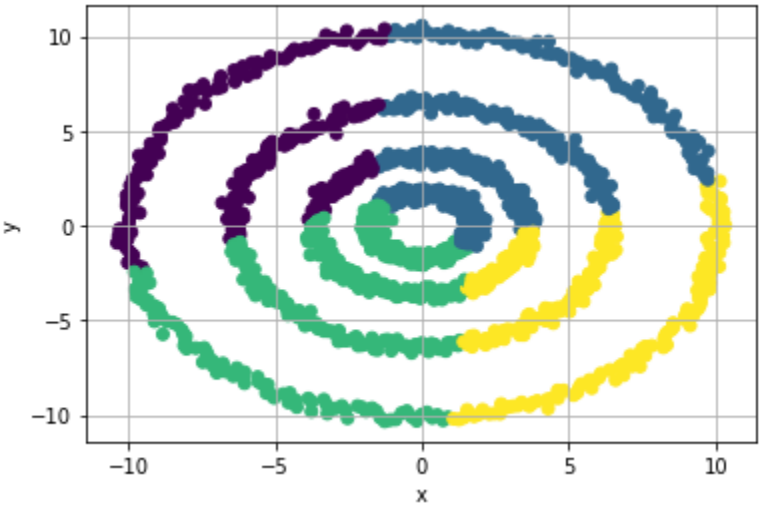
a) (5 points) Plot y on the vertical axis versus x on the horizontal axis. How many clusters are there based on your visual inspection?

Ans:

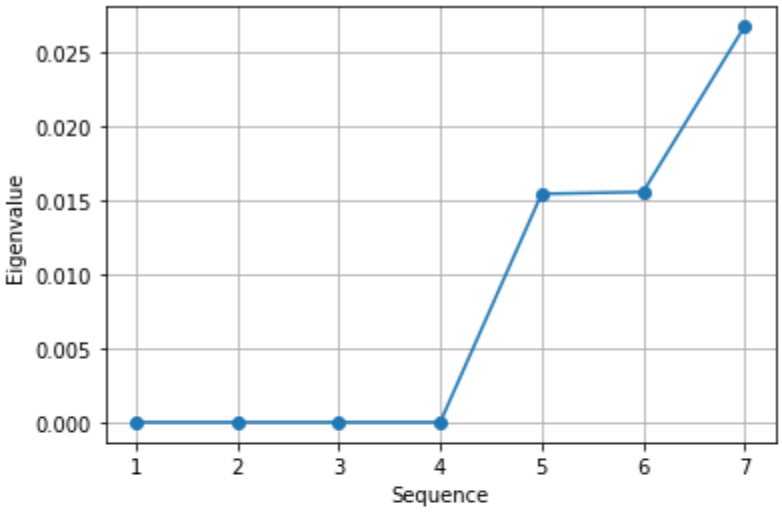


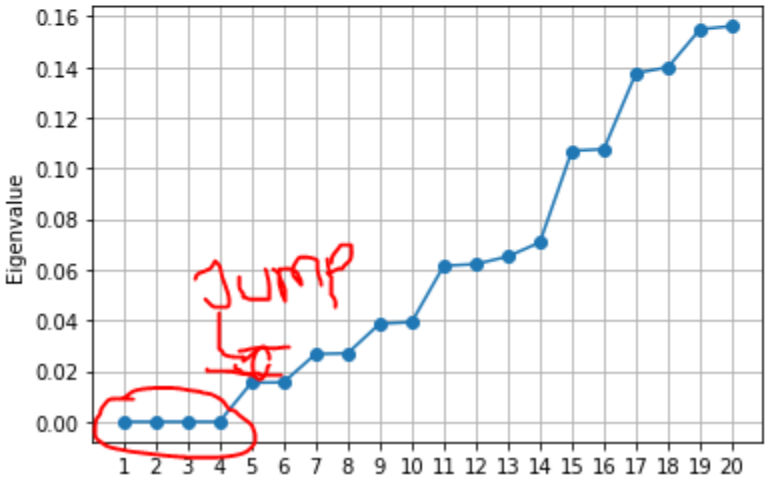

To see this plot, we can say that there are **4 clusters** because we have 4 concentric circles and we can div ide them in 4 quarters

b) (5 points) Apply the K-mean algorithm directly using your number of clusters that you think in (a). Regenerate the scatterplot using the K-mean cluster identifiers to control the color scheme. Please comment on this K-mean result.



c) (10 points) Apply the nearest neighbor algorithm using the Euclidean distance. We will consider the number of neighbors from 1 to 15. What is the smallest number of neighbors that we should use to discover the clusters correctly? Remember that we may need to try a couple of values first and use the eigenvalue plot to validate our choice.




Since we have **4** eigenvalues zeros and significant jump **at 6th,** jump of more than (0.01). So our zero **eigenvalues are 4 and number of neighbors are 6**

d) (5 points) Using your choice of the number of neighbors in (c), calculate the Adjacency matrix, the Degree matrix, and finally the Laplacian matrix. How many eigenvalues do you determine are practically zero? Please display values of the "zero" eigenvalues in scientific notation.

**Adjacency Matrix**:

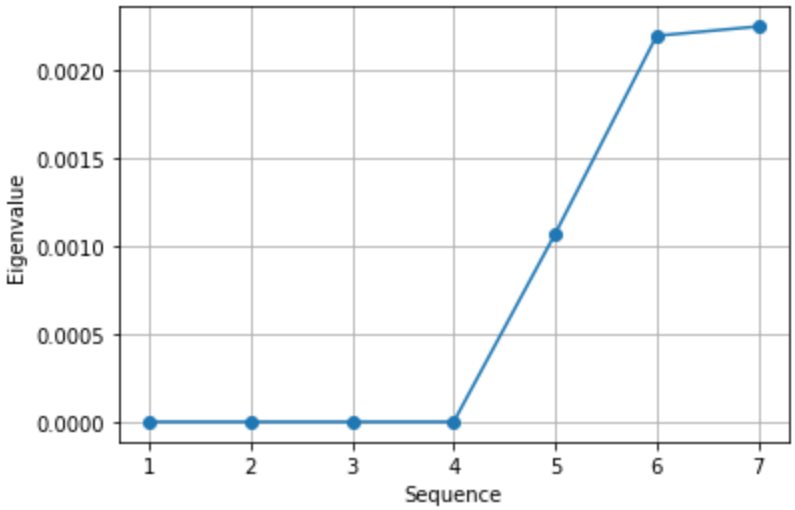
```
[[1.         0.         0.         ... 0.         0.         0.        ]
 [0.         1.         0.         ... 0.         0.         0.        ]
 [0.         0.         1.         ... 0.         0.96602229 0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.96602229 ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

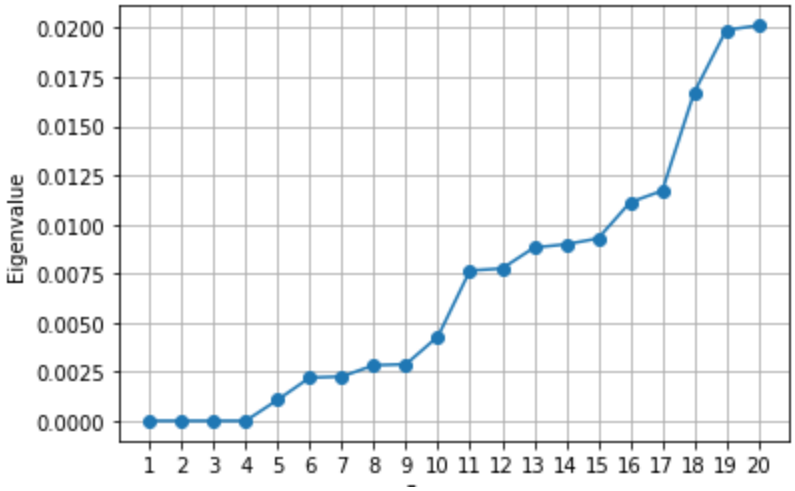
**Degree Matrix:**

```
[[4.80117773 0.         0.         ... 0.         0.         0.        ]
 [0.         4.29598338 0.         ... 0.         0.         0.        ]
 [0.         0.         5.55116784 ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 5.29371731 0.         0.        ]
 [0.         0.         0.         ... 0.         4.88916173 0.        ]
 [0.         0.         0.         ... 0.         0.         4.94116662]]
```

**Laplacian Matrix:**

```
[[ 3.80117773  0.         0.         ...  0.         0.
   0.        ]
 [ 0.         3.29598338  0.         ...  0.         0.
   0.        ]
 [ 0.         0.         4.55116784 ...  0.        -0.96602229
   0.        ]
 ...
 [ 0.         0.         0.         ...  4.29371731  0.
   0.        ]
 [ 0.         0.        -0.96602229 ...  0.         3.88916173
   0.        ]
 [ 0.         0.         0.         ...  0.         0.
   3.94116662]]
```




Eigenvalues are practically zero: [-2.1853679383486055e-15, -2.1537079867095397e-15, -3.834987
3779272163e-16, 4.802750469104981e-16]

e) (10 points) Apply the K-mean algorithm on the eigenvectors that correspond to your “practically” zero eigenvalues. The number of clusters is the number of your “practically” zero eigenvalues. Regenerate the scatterplot using the K-mean cluster identifier to control the color scheme.