CISC 451 - Assignment 3

Will Murray: 20054564, Emma Ritcey: 20007918, Rylen Sampson: 20051918

Preliminaries

For this assignment we used Python which is freely available for download from the official Python website. To enhance its capabilities pertaining to data analytics, numerous libraries were used. The libraries not readily available through the *import* command in Python were installed using Python's built-in library installer Pip. The following list is an overview of the libraries used:

- os
- datetime
- pandas
- numpy
- sklearn
- scipy
- plotly
- matplotlib
- wordcloud

Data Curation

To begin, we computed the Recency, Frequency, and Monetary (RFM) metrics. As the dataset contains data in between 01/12/2010 and 09/12/2011 (a Friday), we used 12/12/2011 as the date for computing recency finding the number of days to this date from the last date a customer made a purchase. We operated under the assumption that at the time this data was being collected, it wouldn't be analyzed until afterwards. Otherwise a customer's recency would have been measured in thousands of days. Frequency was calculated by taking the average number of days in between purchases for a particular customer. Monetary was found by taking the average price of a customer's purchases.
Additionally, we found the number of returns each customer made and a list of the products they bought.

Exploration

We first explored each of the features we obtained in the above step of our analysis. This was done by creating histograms of each feature using the Plotly library. These subplots of *Figure 1* show how each feature is distributed, their scale and gives a shallow idea of possible outliers. Prior to any preprocessing, we ran our data through Principal Component Analysis (PCA) to get an idea of how the clustering might go, standardizing the data in the process. Referencing *Figure 2* in our submission, we see there are no distinct clusters evident yet implying preprocessing is likely necessary.
After preprocessing, which is outlined below, we re-created *Figures 1 & 2* with the refined data. These new plots can be found in *Figures 3 & 4*. Furthermore, we plot the feature histograms one last time with the features on a standardized scale to see the effects of our preprocessing on the algorithm input data. This is available in *Figure 5*.

Further exploration went into varying definitions for frequency as well as uncovering further features we could engineer from the given data. With respect to the latter, in our visual explorations it only appeared that returns would be worthwhile to include as an input to our model. We did consider using the customers country converted to a numeric value but, between

the number of unique values, on top of the fact countries represented as numeric values lack a concept of Euclidean distance, an essential assumption in K-means, we opted against using it. When it came to computing frequency, two different options came to mind such as getting the number of orders a customer makes in a day, and the average number of days between orders. Based on the visual exploration performed in the above paragraph it became apparent the second method gave better clustering results. This is likely due to the first having an over abundance of values close to zero.

Preprocessing

From *Figure 1*, it is clear there are extreme outliers within the features, in particular recency, monetary and returns. To identify these outliers in a programmatic manner we used IQR, under the rule that extreme outliers are given by three times the IQR. This was useful in removing outlandish outliers while keeping those that fall on the brink of being an outlier versus just a large value. As the recency feature did not have an overly large scale and the IQR method would have only eliminated a handful of points, we decided to not remove any records based on the values for this feature.
Before modeling, we standardize each feature using sklearn's *StandardScaler* object. In order to be able to visualize the results of our clustering, we reduced our data to 2D using the *PCA* object from sklearn.

Modeling

To create the clusters for our data we used the K-means data as all our features are numeric and since outliers were removed, the clusters generated by K-means should be more robust. To decide on the number of clusters *k* to generate we used the elbow method. This plots the sum of squared distances of records to their closest cluster center for each k in the tuning loop. From *Figure 6* we see that we should likely use k = 5. Using K = 5, the K-means algorithm gives us the clustering shown in *Figure 7*.

Additional Work

On top of obtaining the metrics for each cluster, we went a step further and attempted to model the top products purchased by each cluster. In the original dataset, every record contained the product bought or returned. Filtering for purchases only, we concatenated all the products a customer bought into a single list. This purchases dataset was then merged with the one containing the RFM results. The wordcloud library provides a functionality to aggregate the text data for every customer within a certain cluster to generate word cloud diagrams of the most frequent re-occurring words in each cluster. These can be seen in *Figure 8*.

Discussion

Our first point of discussion is the distribution of the clusters. Of the five clusters, the first three appear to contain an equal amount of customers while the fourth and fifth are imbalanced. The fourth contains a low percentage of customers and captures the outlying data points evidently

seen in Figure G. The fifth cluster makes up the majority of the customers and its depicting feature metrics are given below.

Due to the number of records and the range of the original records the min and max values for each feature of the clusters are roughly the same. Thus, we find the defining metrics which characterize the clusters from the mean and median. The proceeding table, *Table 1*, displays the results contained in the *../a3/results/cluster_results.csv* file of our submission, which is where the following cluster observations are pulled from as well.

Table 1: Mean, max, min, and median for each feature in all the clusters.

	Recency				Frequency			
	Mean	Max	Min	Median	Mean	Max	Min	Median
Cluster 1	86.02	375	0	65	19.55	142.5	0	5.42
Cluster 2	118.06	375	0	69	37.06	145.5	0	33.3
Cluster 3	131.56	375	0	82	41.52	145	0	37.83
Cluster 4	129	375	0	77	39.61	144	0	37.27
Cluster 5	121.62	375	0	79	39.06	145.5	0	38.2

	Monetary				Returns			
	Mean	Max	Min	Median	Mean	Max	Min	Median
Cluster 1	138.7	708.86	0	91.03	0.39	6	0	0
Cluster 2	188.92	724.95	0	127.01	0.66	6	0	0
Cluster 3	216.83	704.65	0	165.76	0.7	7	0	0
Cluster 4	184.69	714.76	0	133.26	0.72	7	0	0
Cluster 5	205.32	717.22	0	150.2	0.74	7	0	0

Cluster 1: Looking at the mean and median values for Cluster 1 across all four features, it is clear that these values are consistently lower than the other clusters. This would mean that customers falling into Cluster 1 were those who had purchased most recently, bought the most often, but tended to spend the least on average interestingly enough.
Cluster 3: This cluster on the other hand, has the opposite tendencies of Cluster 1. The mean and median for Cluster 3 across all features, are consistently the largest by a noticeable amount. This implies that customers falling in Cluster 3 are older customers, do not buy often and on average spent more. From this, the company could deduce that customers in Cluster 3 are big, one-time spenders, so perhaps they should look into what products these customers bought and around what time. (I.e. Christmas).
Clusters 2, 4 & 5: Based on the mean and median values for these clusters, we see that they act as intermediary clusters between the pattern discovered within Cluster 1 and Cluster 3. That is, Recency and frequency go up while monetary goes down.
As Cluster 2 is the most alike to Cluster 1 in recency, frequency and returns but, instead has strong monetary potential a marketing company may deduce this is the set of customers they should set their sights on for increasing revenue.

From the word cloud diagrams, *Figure 8*, we see that many of the most popular products purchased by customers, or at least the words composing those products, are common across clusters although their prominence differs. Some of the words included in these diagrams are T Light (A light T-Shirt), Jumbo Bag, Lunch Bag, Hot Water, Hanging Heart and other miscellaneous everyday items.These words could either be used by the company to capitalize on their different customer types, or perhaps find ways to broaden the purchasing of each customer group. For example in Cluster 1 some of the most prominent product words include different bag or pack types, products used to store stuff. Likely, these customers won't be routinely buying these products as their longevity should be a couple years at least. The company should thus market products that would pair well with these products to customers falling in Cluster 1. Purchasing various backpack styles could be indicative of someone who likes hiking, or is always on the go. This means they should market products for those always on the go to these customers.

Results

Below we present the common metric for numeric clustering algorithms, sum of squared distance (ssd). This tells us, from the perspective of euclidean distance, how similar data points in a cluster are to their centroids. Given that most clusters contain around 800 records, except for Cluster 4 & 5 which are less than and greater than respectively, we argue that these clusters are well-defined in the sense of distance as no single cluster displays an overly large ssd with respect to its centroid as seen in *Table 2*.

Table 2: The relevant clustering assessment metric for K-means, sum of squared distance.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Total
Sum of Squared Distance	426.85	330.95	434.06	501.66	261.04	1954.56