Online Retail

Data Science 4: Introduction to Machine Learning

University of Toronto - School of Continuing Studies

April 12, 2021

Group 5:
Mabigail Mabbayad
Umer Muhammad
Christopher Owen
Jagjeet Rathore
Priyanka Velagala
Guang Xi

Table of Contents

1. Objectives

Our primary objective is to segment customers based on their RFM (Recency, Frequency, Monetary value) metrics and predict the RFM segment using machine learning. RFM is a marketing analysis tool used to identify a firm's best clients, based on the nature of their spending habits. Our secondary objective is to build a machine learning model to predict the customer Lifetime Value (LTV), a metric that indicates the total revenue a business can reasonably expect from a single customer account. LTV is often considered the **"Golden Metric"** for online retailers, as it helps predict future revenue and measures long-term business success. We hypothesize that there is a positive correlation between RFM score and LTV.

In addition, we would like to explore product clustering to better understand customer spending patterns. Through this intelligence we may discover opportunities for a business to better serve their customers and provide an improved personalized retail experience. Further study of the feedback intelligence may provide more venues to improve customer retention and entice customers to make more purchases.

2. Data Preparation

The chosen dataset is a transactional data set from an online e-commerce system obtained from UCI's machine learning repository [1]. It was of excellent quality with only a few challenges to overcome. As this is Point of Sale (POS) data, it required additional thought for manipulation and normalization of the dataset. Some identifiers or behaviors were difficult to interpret as we have zero knowledge of the POS system, eg. stock codes, or behaviours of cancellations, refunds, discounts, etc.

2.1 Data Preprocessing

We wanted to avoid removing any records that would be considered a net positive profit or income. Beyond the scope of standard normalization or data-cleaning process, such as renaming columns or converting to correct types, removal of NaN's or nulls etc. we discovered some discrepancies.

First we examined any duplicate records to remove and we found approximately 5225. Next, we discovered records with negative quantities. After inspection, we concluded many of these to be cancellations. Out of 22,190 records, we could clearly identify 3,654 as cancelled.

Prior to the removal of these records, we had to also identify and flag other cancelled records by incorporating additional logic. These records did not have the original InvoiceNo. approximately 5192 records were identified and dropped by finding records with negative quantities and matching them against their mirror records where the InvoiceNo's were not the same (equivalence of absolute value of quantities).

After removing mismatched cancelled orders, the remaining cancelled orders and their originals we opted to only capture totals or Invoice totals (cart price) greater than 0. This removed any remaining discounts or negative values that did not represent bought products or customer behavior. Finally the cleaned data contains only model-contributing records (**Table 2.1.1**).

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	CancelledQuantity	TotalPrice
column type	object	object	object	int64	datetime64[ns]	float64	object	object	int64	float64
null values (nb)	0	0	0	0	0	0	0	0	0	0
null values (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 2.1.1 Final Cleaned Data

3. Exploratory Data Analysis

The dataset has a total of 1 year of POS data across 38 countries. There are a total of 25900 unique invoice numbers, 4070 unique stock codes and 4223 unique product descriptions. 92% of transactions were from the United Kingdom, followed by Germany, France, Eire, Spain (**see Figure 8.1 in Appendix**). The WHITE HANGING HEART T-LIGHT HOLDER was the most popular product.

4. Model Design

We decided to analyze this dataset from 3 different perspectives:

1. Customer Segmentation and Prediction
2. ML Model to Predict Customer LTV after RFM Scores
3. Product Clustering

4.1 Customer Segmentation and Prediction

In Customer Segmentation and Prediction, we first built 2 data models for clustering and prediction. As the original dataset is transactional, feature creation was required: RFM model and Customer LTV model.

In the RFM model, we used a semi-supervised approach. Our first step was to identify the customer clusters by using the following unsupervised learning algorithms: K-means, Hierarchical cluster and DBSCAN. We wanted a segmentation which clearly identifies the customer's purchase profile.

K-means: We used the Elbow Method and Silhouette Method to determine the parameter : n_clusters (**see Figure 8.2 in Appendix**).
Hierarchical Cluster: We chose to implement agglomerative clustering for this project. We utilized the linkage function from scipy to perform the clustering (**see Figure 8.3 in Appendix**).

DBSCAN: Selecting for epsilon was done by graphing the average of the distances for every point to its nearest neighbour (**see Figure 8.4 in Appendix**). The minimum points parameter was calculated through the method: 2 times the dimension of the features.

After identifying the customer segments, we used several supervised learning models to predict which segment a customer belongs to based on their RFM. We chose the following classifiers (KNN, Decision Tree, Random Forest, Extra Trees, AdaBoost and Gradient Boosting), which work well on scaled datasets with fewer dimensions. Then we used Grid Search to find the best hyperparameters. After running these models, we realized that many of them were overfitted. So we further tuned hyperparameters like max_depth, learning_rate, n_neighbors, and n_estimators to produce more favourable results.

4.1.2 ML Model to Predict LTV

Plan for Model Validation:

1. **RFM Score Calculation** of Customers from a **3 month** time period to rank them on a scale of 1 to 7 (the higher the better,).
2. Calculate LTV Scores of Same Customers from the **6th month**.
3. We plot them together to see if RFM segments are useful in predicting LTV.

Recency: Find the most recent purchase date of each customer and calculate the number of days for which they are inactive for. After getting the number of inactive days for each customer, apply K-means clustering to give each customer a recency score from (0 to 3) we choose k = 4.
Frequency: Total number of orders for each customer. Use K-means (k = 4) to find frequency clusters and give each a score (0 to -3).
Revenue: Feature engineering of quantity and unit price to calculate total revenue (Monetary value) for each customer. Again, apply K-means to find Revenue score.

Now, we have a **rank or score** (cluster number) for recency, frequency and revenue. Create overall score using the formula:
Overall Score = FrequencyCluster + RecencyCluster + RevenueCluster

Table 8.5 (see Appendix**)** shows that score 7 is our best customer and 0 is our worst customer. Below we classify the RFM score range to distinct segments:

Low segment (RFM Score: 0-2): Customers who are less active than other segments, not very frequent buyer/visitor, generate very low, or even negative revenue.

Mid segment (RFM Score: 3-4): Customers that often use the website (but not as much as the high segment), fairly frequent customers and generate moderate revenue.

High segment (RFM Score: 5+) : Customers that generate high revenue, frequency and low inactivity. This group of customers' business don't want to lose since they generate high revenue.

CLTV Prediction Steps:

1. Define a time frame for customers' lifetime value calculation. This depends upon business goals, for this we choose 6 months.
2. Define the predictors that we will use to predict Future and create them.
3. Calculate LTV for ML Model, use RFM scores as predictors for each customerID, utilize 3 months of data for calculating RFM, to predict next 6 months.
4. Lifetime Value calculation: Calculate 6 month LTV for each customer which we will use for training our model. The revenue will become our LTV. Now we need to merge the 2 dataframe to see correlation with LTV and our predictors.

Results: The graph below shows a positive correlation between LTV and RFM:

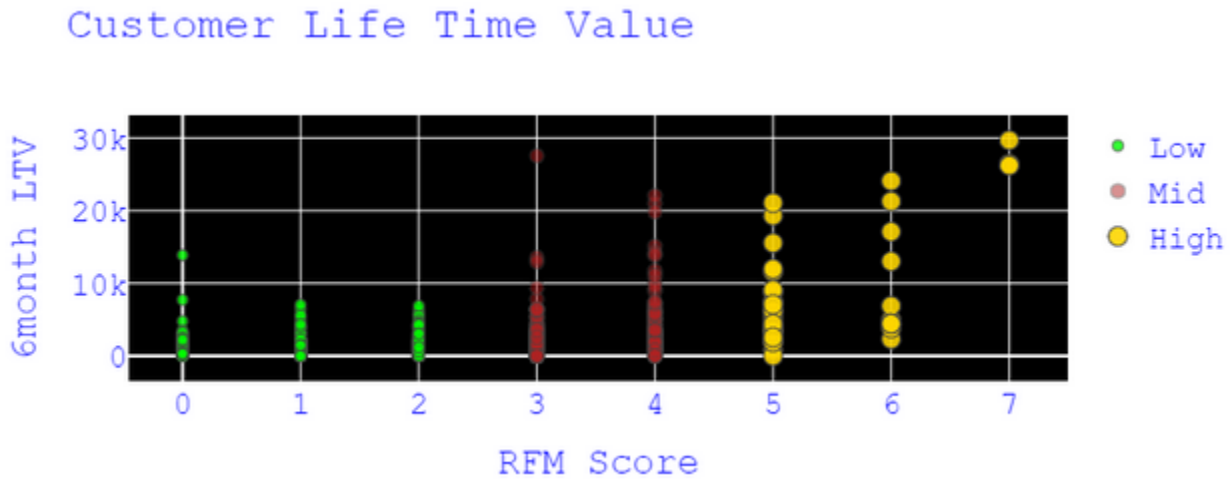

Figure 4.1.1 - Correlation between RFM Score and LTV

LTV is an amount, so we need to convert into LTV segments. We treat customers differently based on their predicted LTV. Again, we apply K-means clustering, but our k-value depends on business. For this project, we assume the business wants a value of 3. After applying K-means, we now have customer segments (Low LTV, Mid LTV and High LTV). 2 is best with average 8.2k and 0 is the worst with 398. (**see Table 8.6 in Appendix**)

Model Design Steps

1) Convert categorical columns into numerical using pandas get dummy function
2) Features recency, frequency and monetary are all rightly skewed, applied transformations to make them normally distributed.
3) Removal of outliers with quantile method from Revenue
4) Check the correlation of features using seaborn heatmap against LTV clusters.
5) Separation of independent and dependent variables (depend will be LTV labels) and independent means [recency, frequency, revenue, overall score, mid-value, high value, low value segments)Split in to training and testing and testing part for the checking the performance of model.Training part for model building.

4.2 Product Clustering

Product clustering was another aspect of analysis that was explored in order to analyze consumer spending patterns more effectively and derive business intelligence. The input features used for this analysis was the product description field, for which the dataset has a total of 3878 unique values. Two approaches were used to analyze the description values, TF-IDF Vectorizer and NLTK. A keywords library was tokenized through NLTK, product category clusters were characterized appropriately using the K-means algorithm. Additionally, the product category clusters created using the NLTK keywords library was also applied towards a customer purchase prediction model. A prediction model was created using a combination of product clustering, customer categorization, and order combinations. Finally a voting classifier was used to discover the prediction accuracy.

4.2.1 Marketing Potentials

Preprocessing of the data for the TF-IDF Vectorizer included using sklearn's feature extraction library to filter out common words from the product description. In addition to the in-built stop words, frequently occurring colors were also filtered out to improve accuracy of clustering. With the given input, K Means algorithm was run for different values of K and using the K-elbow visualizer revealed the optimal value to be 16. A maximum threshold of K = 20 was used to ensure customer patterns across each product cluster can be analyzed and to make meaningful recommendations.

4.2.2 Customer Purchase Prediction

We explored additional applications of product clustering in the attempts to create a prediction model for the next purchase a customer may make based upon their behavior. The approach to manipulating the data for this specific model approach was to create the following features: Product Keywords, Product Keyword Categories utilizing NLTK. These features allowed the creation of a matrix representing customer behaviors and their amounts spent in the respective clustered product categories.

5. Model Evaluation

5.1 RFM & LTV Customer Segmentation and Prediction

5.1.1 Clustering

In the RFM analysis, we applied three different clustering algorithms: K-means, Hierarchical, and DBSCAN. All three algorithms suggested that the optimal cluster was 3. Then we checked the details of each cluster and performed a 3D plot, box plot and scatter plot to visualize the clusterings. From the 3D plots below, we can easily draw the conclusion that K-means is the

best, as K-means clustering is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.

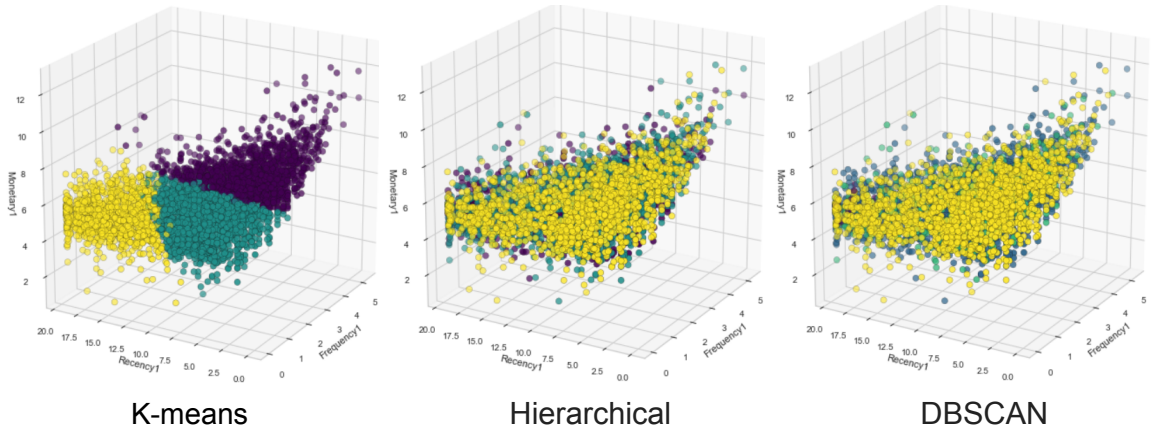

Figure 5.1.1 - 3D Plots of clustering results from 3 models

Using the K-means clusters, the customer purchasing profiles can be described as follows:

Customer Profile	Avg Spending	Avg Recency	Avg Frequency
High Value	£5089	28 days	10
Mid Value	£608	46 days	2
Low Value	£396	234 days	1

Table 5.1.2 - Customer Profile using K-means clusters

5.1.2 Model Evaluation for RFM

After choosing the best clustering algorithms and applied clustering, we performed prediction on the following models and applied GridSearch to tune parameters. GridSearch allowed the hyperparameters to be changed, evaluated, allowing for the best model to be selected.

	Prediction_Time	Accuracy_Score	f1	Precision	Recall
GradientBoostingClassifier	0.253229	99	0.989996	0.990082	0.989992
RandomForestClassifier	0.0170155	95.69	0.956925	0.957663	0.95689
DecisionTreeClassifier	0.00400352	94.69	0.946849	0.9476	0.946882
AdaBoostClassifier	0.0790718	94.53	0.945547	0.950416	0.945343
ExtraTreesClassifier	0.0110104	91.38	0.91336	0.926024	0.91378
KNeighborsClassifier	0.0280252	58.66	0.589759	0.613119	0.586605

Figure 5.1.3 - DataFrame for Models Evaluations on Testing Data

The ExtraTreesClassifier performed the best. Other models are either overfitted or underfitted. We think the reason is that Random Forest and Decision Tree choose the optimum split while ExtraTrees chooses it randomly. However, once the split points are selected, the two algorithms choose the best one between all the subset of features. Therefore, ExtraTrees adds randomization but still has optimization.

5.1.3 Model Evaluation for LTV

Classifications Models: As the data is segmented into 3 groups (low, high and mid segments), this is a multiclass classification problem. Different classification models like RandomForest, ExtraTrees, XGboost, KNN can be applied with and without hyperparameters.
Comparison of Models: Various metrics were used for evaluating the performance of each model**:** Classifiers that use bagging Techniques like random Forest and Extra Tree are used. Classifiers that use boosting techniques like XGboost were used. Precision of Extra tree classifier is high compared to other models and its accuracy increases with the parameter min_sample_leaf**.**f1-score is high too, so this model is good in making predictions.

The Classification Report for the ExtraTreesClasifier is predicting class (high) accurately, as compare to class 1 and class (0) since precision is 1 for 2, 0.47 for class1, 0.85 for low class

	Prediction_Time	Accuracy_Score	f1	Precision	Recall
ExtraTreesClassifier	0.021827	79.920000	0.792593	0.809061	0.822222
RandomForestClassifier	13.963448	76.920000	0.712891	0.692980	0.744444
XgboostClassifier	0.007317	78.150000	0.780219	0.777718	0.788889
KNNClassifier	15.928051	76.000000	0.705888	0.692980	0.788889
Default_ExtraClassifier	0.140897	78.890000	0.701430	0.631673	0.788889
Default_RandomForestClassifier	0.144676	77.780000	0.680556	0.604938	0.777778
default_XgboostClassifier	0.144610	77.780000	0.761590	0.749418	0.777778
default_KNNClassifier	0.006956	75.560000	0.686273	0.655943	0.755556

Figure 5.1.4 - DataFrame for Models Evaluations on Testing Data

ROC-Model Comparison(LTV):
After optimizing models with hyperparameter tuning, the XGboost AUC is very high. This indicates that it is good for distinguishing between classes (low, medium, high).

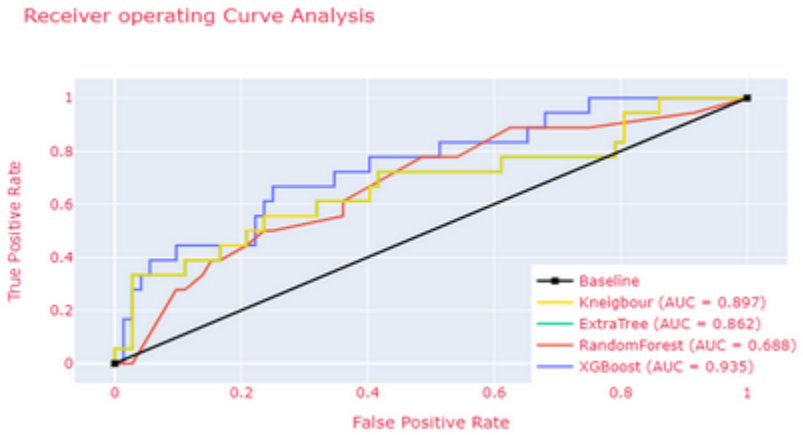

Figure 5.1.5 - Comparison of Various Classifiers using ROC

5.2 Product Clustering

5.2.1 Marketing Potentials

The T-SNE plot with 2 components shows partial formation of distinct clusters, however it is evident that cluster 0 is sparsely distributed. As the clustering is based on similarity in product names, this is to be expected as although products that have similar keywords in the description, it doesn't always make sense to cluster them together. One such example is in cluster 6 where both "SCANDINAVIAN 3 HEARTS NAPKIN RING" and "DROP DIAMANTE EARRINGS CRYSTAL" are in the same cluster however both are vastly different products.

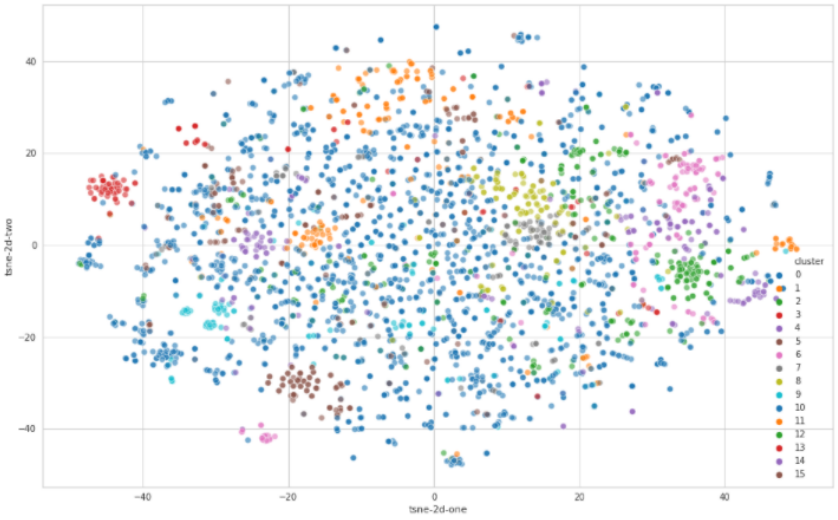

Figure 5.2.1 - Visualizing product clusters in 2-D using t-SNE

With the above mentioned method, each product was associated with a cluster. The sales dataset was then manipulated to find customer spending across each of the 16 product clusters to find their top spending cluster. Based on the top spending cluster, two product recommendations were generated for each cluster. The first, a product that generated the most revenue for the retailer and the second, a product that was bought by the most customers. For example, customer ID 14141's top spending cluster was product cluster 3. In product cluster 3, the product that generated the most revenue was "GIN + TONIC DIET METAL SIGN" and the most popular was "PLEASE ONE PERSON METAL SIGN". For customer ID 14141, recommending the above products would have a favourable outcome as there are multiple instances of metal signs in their purchase history.

5.2.2 Customer Purchase Prediction

Wordcloud library is used to visualize the product clusters (**see Figure 8.8 in Appendix)**. To validate the distinctness of the clusters, PCA is utilized only to inspect explained variance.
The creation of these features enables the bucketing of Invoice Totals across their respective product categories, thus a matrix is developed to represent a customers total invoice counts,

invoice averages, total customer spends, and their respective distributions towards product categories. Average silhouette scores were calculated for a total of fifteen clusters, but for the most appropriate chosen value is eleven. Spider chart or Radar chart visualization depicts the distribution of the customer clusters and their respective (**Fig 5.2.2**)

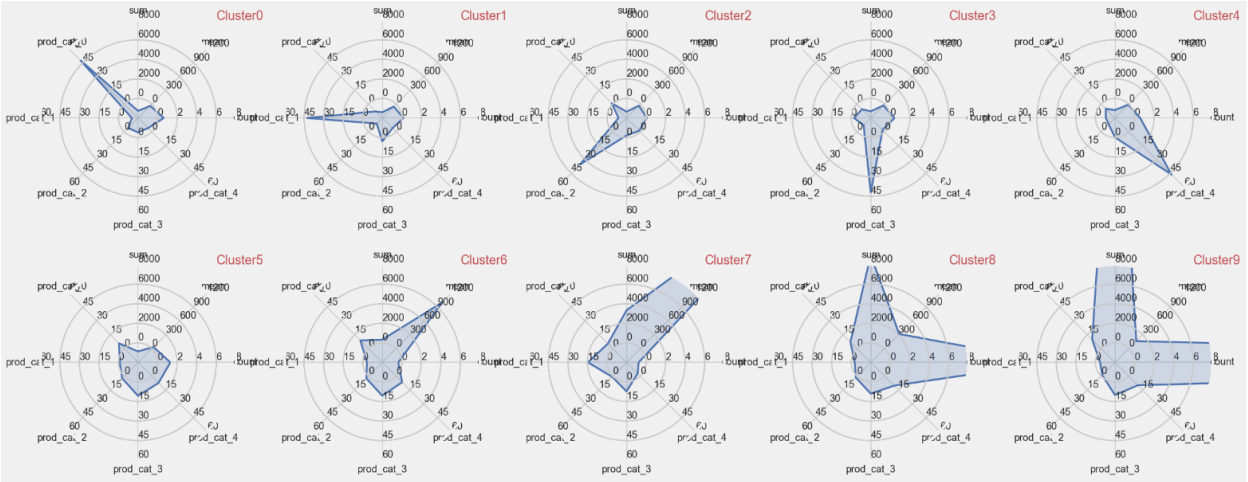

Fig 5.2.2 Spider chart visualization of Customer Product Clusters

Table 5.2.3 presents the accuracy results of classifiers using best estimators of given hyperparameters applied to Training/Test sets.

SVC	Logistic Regression	KNN	DecisionTree	RandomForest	Adaboost	Gradient Boosting
84.626%	90.305%	81.163%	83.102%	91.413%	56.233%	90.166%

Table 5.2.3 Training/Test Classifier Accuracy Results

Most of these look somewhat nice, however, VotingClassifier accuracy results against the actual data were ~76.28%. These are significantly lower, and may need further tweaking or a larger dataset to work with.

6. Conclusion

Exploring and modelling the data using RFM and LTV methodologies, we confirmed our original hypothesis of a positive correlation between RFM scores and LTV. Utilizing K-means clustering we created 3 distinct customer segments. These groupings describe specific customer purchasing patterns and behaviour which is very useful for differentiated target marketing. We also predicted each segment with a 0.91 F1 score using the Extra Trees Classifier. This prediction model may be used to classify new customers to receive the appropriate marketing offers.

Through analysis and model building we discovered that ~40% of the customers within this dataset had only one transaction. These may be consumers performing a guest checkout. These customers are important to identify and target for customer retention strategies, requiring further investigation and business intelligence.

The dataset was limited to only 13 months of transactional data, however, we prefer it to be a larger time frame as this may have shed light on undiscovered areas for analysis such as seasonality, annually or bi-annual analysis, or customer loyalty. Insights into how the e-commerce system handles the data manipulation of transactions such as bonuses, coupons, discounts, cancellations and refunds would be helpful in creating a cleaner data file.
There were also significant challenges encountered while cleaning the data as we often found transactions with non-zero quantity and a unit price of zero, or product descriptions with value ‘Manual’ etc. We were also limited to the processing power of our local machines with a dataset size of approximately half a million records. This required long run-times and may have limited the scope of our hyper-tuning of parameters.

Finally, a part of the analysis that would be redundant in the real-world is product clustering. As the dataset is of an e-commerce system, the underlying product catalogue would have already been built from product information provided by suppliers. Hence, had this been readily available more efforts could’ve been focused on the recommendation and prediction models.

7. References

1. Daqing Chen, Sai Liang Sain, and Kun Guo, Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining, Journal of Database Marketing and Customer Strategy Management, Vol. 19, No. 3, pp. 197-208, 2012 (Published online before print: 27 August 2012. doi: 10.1057/dbm.2012.17).
Accessed Mar. 6, 2021 [Online]
Available: https://archive.ics.uci.edu/ml/datasets/online+retail.

8. Appendix

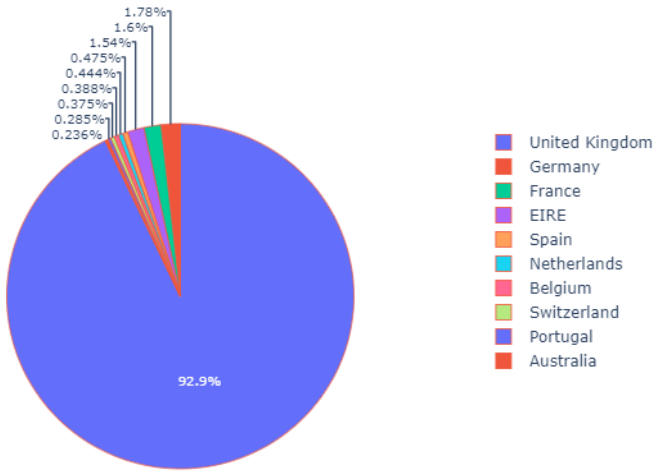

Figure 8.1 - Percentage of Transactions in All Countries

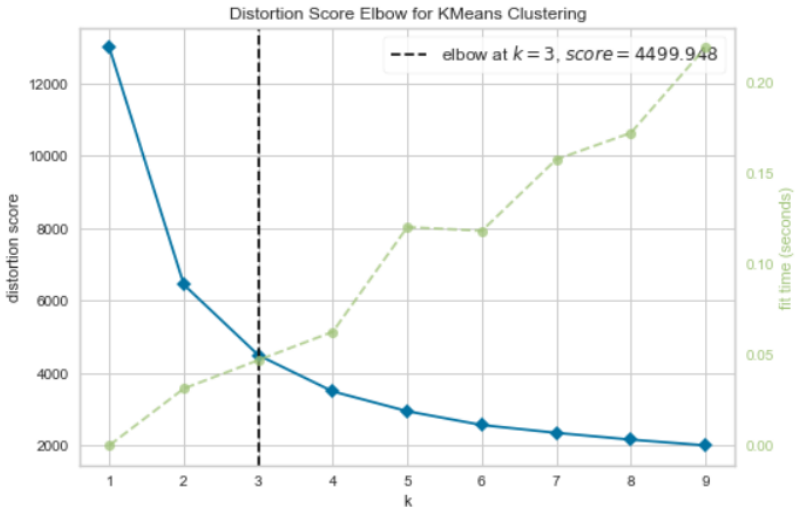

Figure 8.2 -Elbow Graph for K-means Clustering

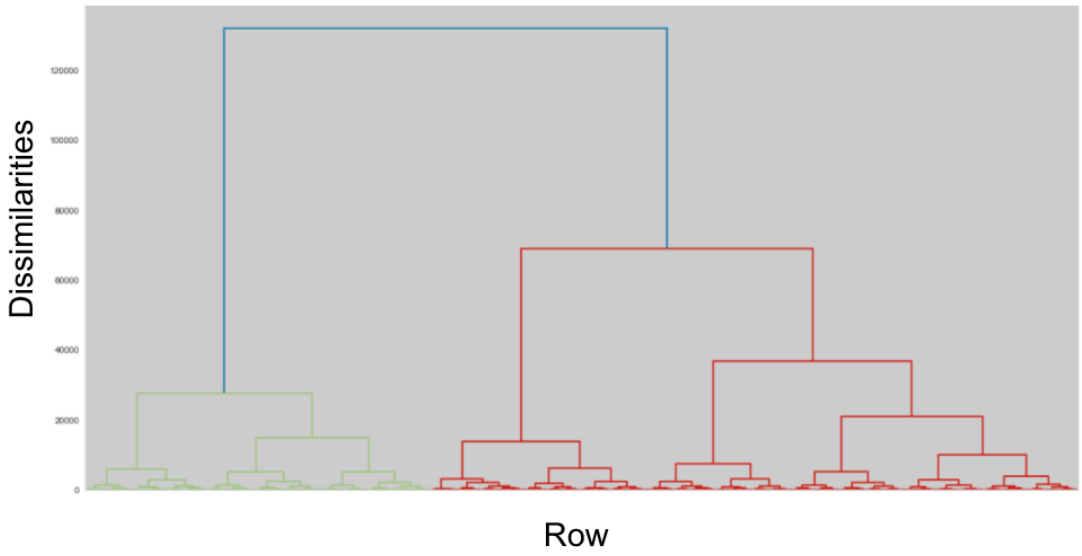

Figure 8.3 - Hierarchical Cluster dendrogram

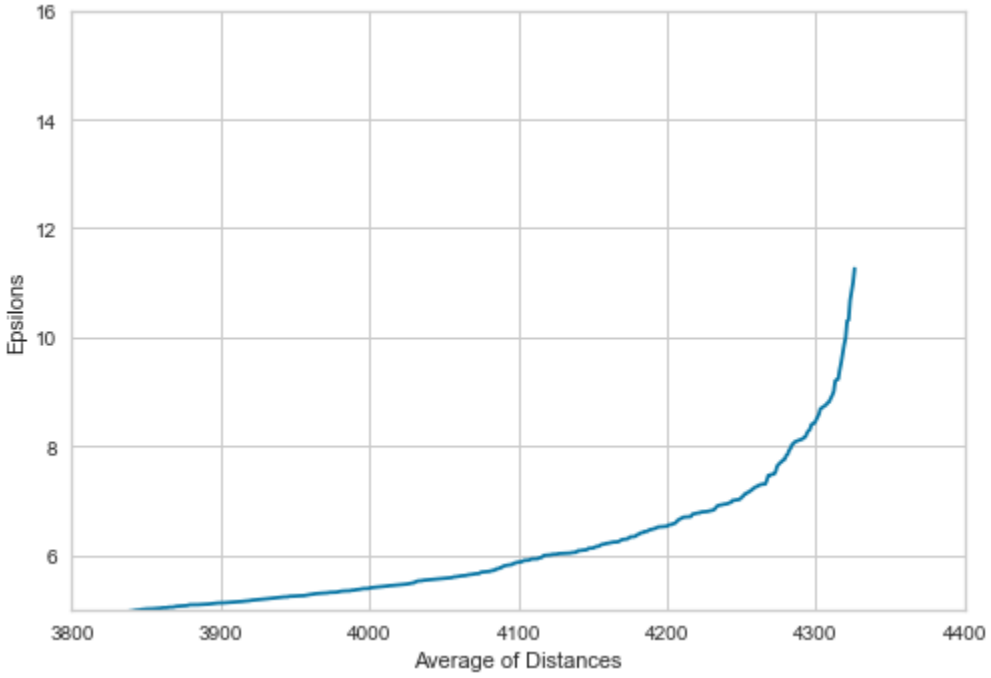

Figure 8.4 - Knee Method for DBSCAN

OverallScore	Frequency	Recency	Revenue
0	18.432836	77.014925	295.824239
1	24.522673	50.887828	367.334177
2	31.028571	26.590476	500.973095
3	36.652439	10.243902	633.871402
4	103.835938	8.109375	1983.985086
5	138.965517	6.482759	4326.155172
6	322.071429	7.357143	12177.627143
7	779.333333	3.000000	11350.896667

Table 8.5 - Overall Score Description

LTVlabels	count	mean	std	min	25%	50%	75%	max
0	1397.0	398.393172	422.252144	-609.40	0.0000	298.000	687.7200	1448.78
1	368.0	2501.323098	936.361335	1464.05	1739.9875	2165.315	3054.7925	5287.39
2	56.0	8222.565893	2983.572030	5396.44	6151.4350	6986.545	9607.3225	16756.31

Table 8.6 - Characteristic of Each Cluster

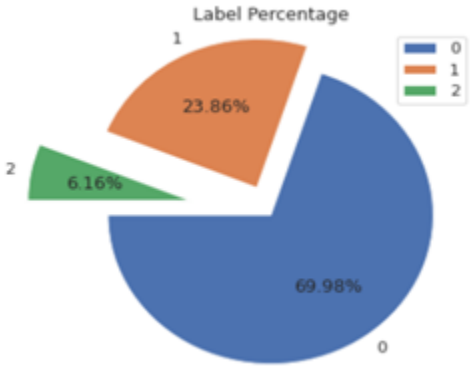

Figure 8.7 - Percentage of LTV segments

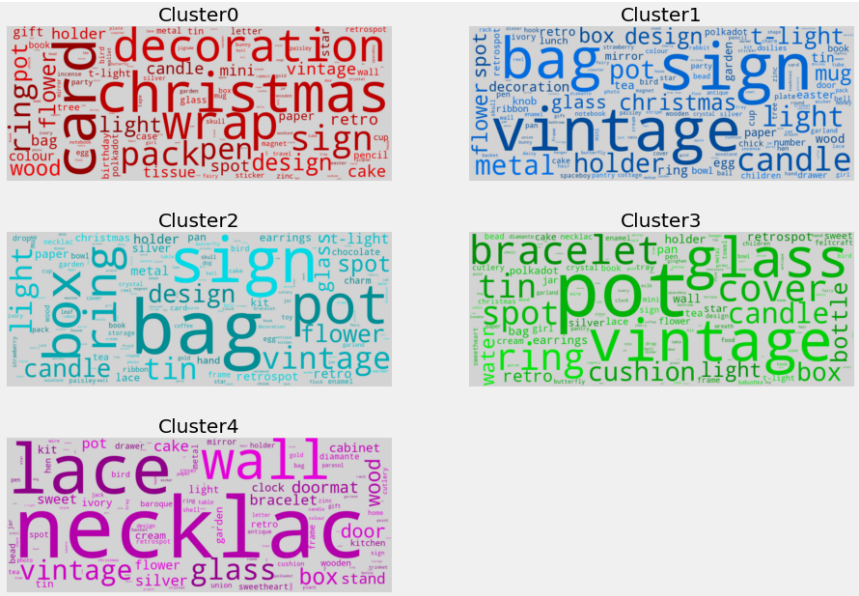

Figure 8.8 - This looks great but requires further distinction