Appendix

Codebook:

Object Name	Type	Description
Section 1: Big Data Apache File Format		
Opening the parquet file and create a reference in Spark		
transaction_parquet	Dataframe	Used to store and open saved parquet file
transaction_df	Dataframe	Collects data into a dataframe
transaction_ref	Dataframe	Used to copy the transactions in the R dataframe to Spark memory
Section 2.1: Data Checking		
2.1.2 Generating summary statistics for every column - applicable to numerical variables		
na_values	Dataframe	Check the number of na values in each column
2.1.3 Checking for unique values in different columns		
unique_values	List	Checks for unique values, and different values can be checked by changing the parameter in select()
Section 2.2: Data Cleaning		
2.2.1 removing rows with NA CustomerNo, and rows with negative quantity		
transaction_clean	Dataframe	Stores the new dataframe after performing data cleaning
Section 3: Customer EDA and Visualisation		
3.1: Geographical Analysis		
country_plot	Spark Dataframe	Dataframe used for geographical analysis, used to create a bar plot for the number of customers in different countries
3.2 Visualizing transaction volume by month		
temp_df1	Spark Dataframe	Dataframe used for visualizing transaction volume by month.
3.3: Average spent per transaction		
temp_df2	Spark Dataframe	DataFrame used for analyzing average spend per transaction. Includes filtering for high variability and subsequent visualization.

3.4: Number of Unique Products bought per customer		
temp_df3	Spark Dataframe	Dataframe used for analyzing the number of unique products bought per customer.
3.5: Average Basket size		
temp_df4	Spark Dataframe	Dataframe used for analyzing average basket size per customer.
Section 4: Feature Engineering		
4.1: Feature engineering for ref_customer		
ref_customer	Dataframe	Creates a customer summary dataset and aggregates transaction data
temp_df5	Spark DataFrame	Creates a column to identify the month where each customer spent the most
median_duration	Tibble	Finds the median of duration values based from the calculations in ref_customer
Section 5.1: Modelling in Spark - Logistic Regression		
5.1.1 Correlation Analysis		
corr_matrix	Dataframe	Contains a correlation matrix for all predictors, excluding “CustomerNo”, “duration”, “festive_spender”, and “loyal_customer”
5.1.2 Splitting into training/testing set		
ref_customer_split	Dataframe	Stores the randomly split data, where the probability assigned to training set is 0.8, and 0.2 for the testing set
ref_customer_split_train	Dataframe	Stores the split training data from ref_customer_split
ref_customer_split_test	Dataframe	Stores the split test data from ref_customer_split
5.1.3: Standardization of variables		
ref_customer_stats	Dataframe	Contains the mean and standard deviation values based on the training set
ref_customer_split_train	Dataframe	Stores the split training data after standardization
ref_customer_split_test	Dataframe	Stores the split test data after standardization
5.1.4: Evaluating the models		
Model_1	Logistic Regression Model	Fits a logistic regression to RFM variables in the training data for loyal_customer target data

validation_summary1	Dataframe	Validation metrics for Model_1
Model_2	Logistic Regression Model	Fits a logistic regression to several variables in the training data
validation_summary2	Dataframe	Validation metrics for Model_2
5.1.5: CI plot - with all the variables		
tidy_glm_fit	Dataframe	Fits a logistic regression with the RFM training data and used to create a confidence interval plot for Model_2 predictors
5.2: ML Pipeline - Logistic Regression		
5.2.1: Logistic Regression ML pipeline		
logistic_pipeline	ML pipeline	ML pipeline used for logistic regression
5.2.2: Cross-validation		
cv	ML pipeline	Used to create a cross-validation pipeline to finetune hyperparameters
cv_model	ML model	Fits cross-validator to the training data from ref_customer_split_train
5.3 Modelling in Spark - Kmeans Clustering		
rfm_stats	Dataframe	Used to get the mean and sd values based on the whole dataset
ref_customer_for_kmeans	Dataframe	Contains the dataframe after standardsing the “Recency”, “Frequency”, and “Monetary” columns from the ref_customer dataframe
k_values	Vector	Stores a vector of k values ranging from 2-6
silhouette_scores	Vector	Stores the length of k_values
k_means_model	ML model	Used to train K-Means model, using “R_standardized, “F-standardized, and “M_standardized” features for clustering
Silhouette_data	Dataframe	Creates a dataframe for k-cluster values & respective silhouette scores
5.4: ML Pipeline - Kmeans Clustering		
5.4.1: Creating the pipeline, fitting the model and collecting the predictions		

kmeans_pipeline	ML pipeline	Stores the K-Means pipeline using “Recency”, “Frequency”, and “Monetary” as input columns
kmeans_pipeline model	ML model	Used to train the K-means pipeline using ref_customer
5.4.3: Understanding the clusters		
predictions	Dataframe	The result of applying data to the K-means pipeline

Citations:

Murphy, C., & Kvilhaug, S. (2022, November 19). *What Is Recency, Frequency, Monetary Value (RFM) in Marketing?* Investopedia. Retrieved November 18, 2023, from https://www.investopedia.com/terms/r/rfm-recency-frequency-monetary-value.asp

Ramos, G. (2021). *E-commerce Business Transaction*. Kaggle. Retrieved November 18, 2023, from https://www.kaggle.com/datasets/gabrielramos87/an-online-shop-business

Appendix

Figure 1: Geographical Distribution of Sales

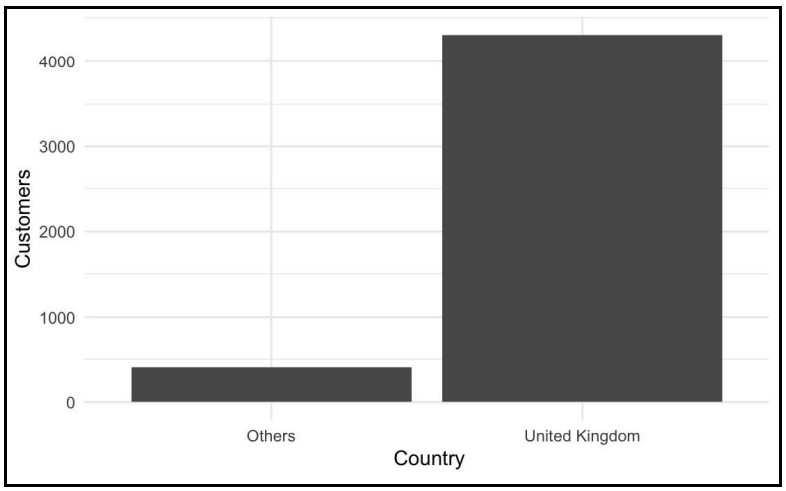

Figure 2: Monthly Transaction Volume

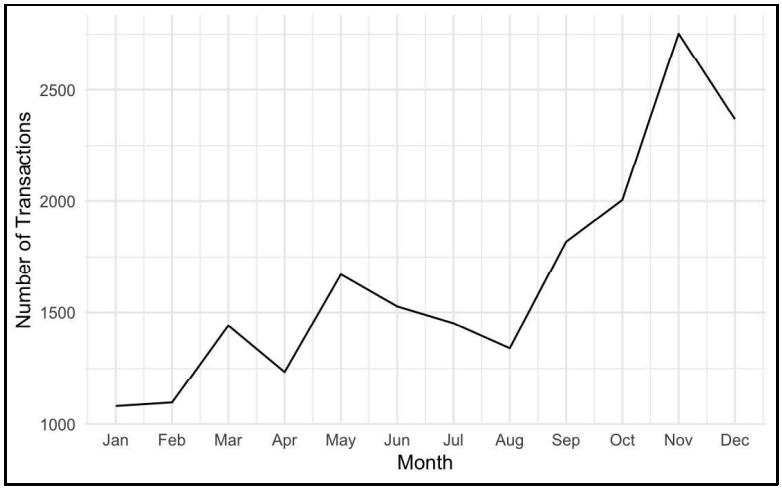

Figure 3: Average Spend per Transaction

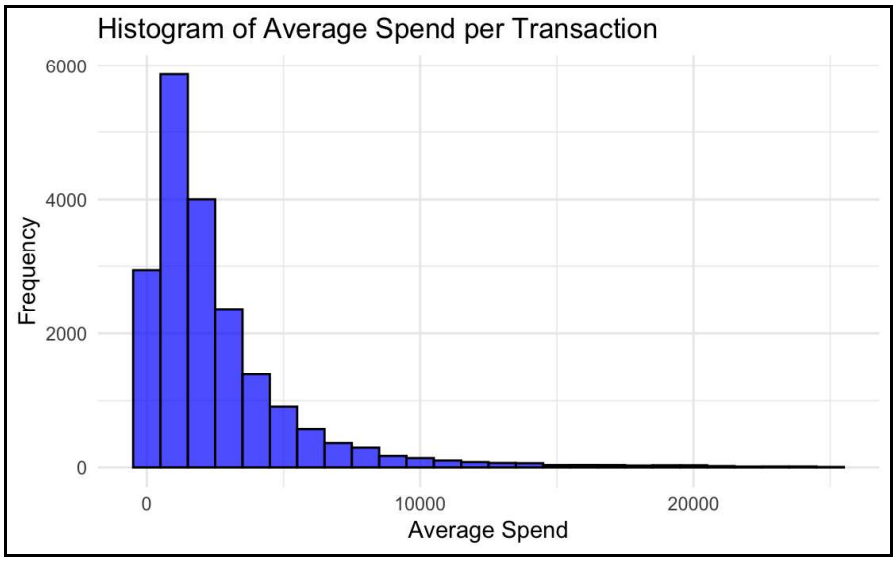

Figure 4: Number of unique products bought per customer

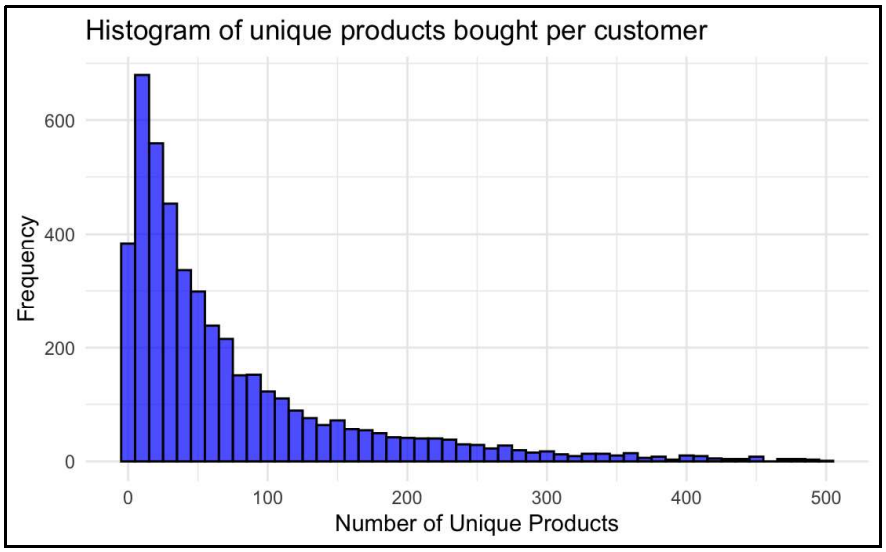

Figure 5: Average Basket Size per Customer

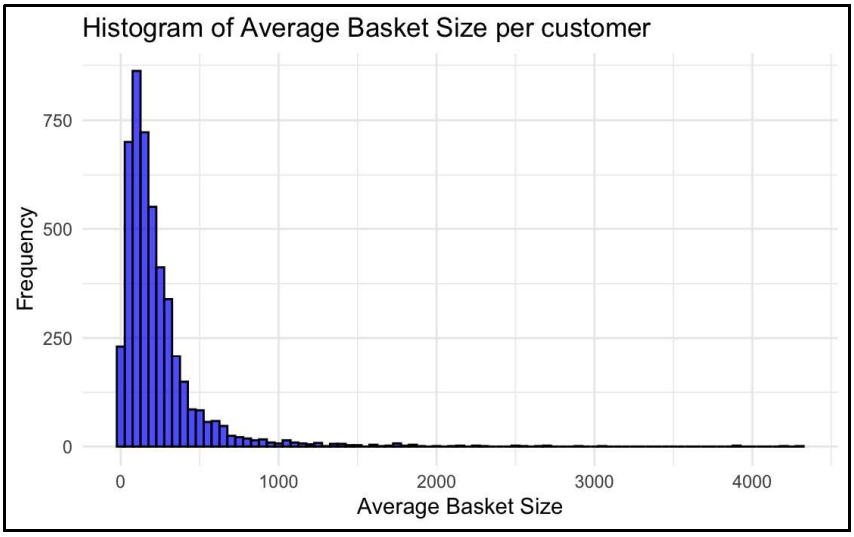

Figure 6: Correlation matrix of feature engineered variables

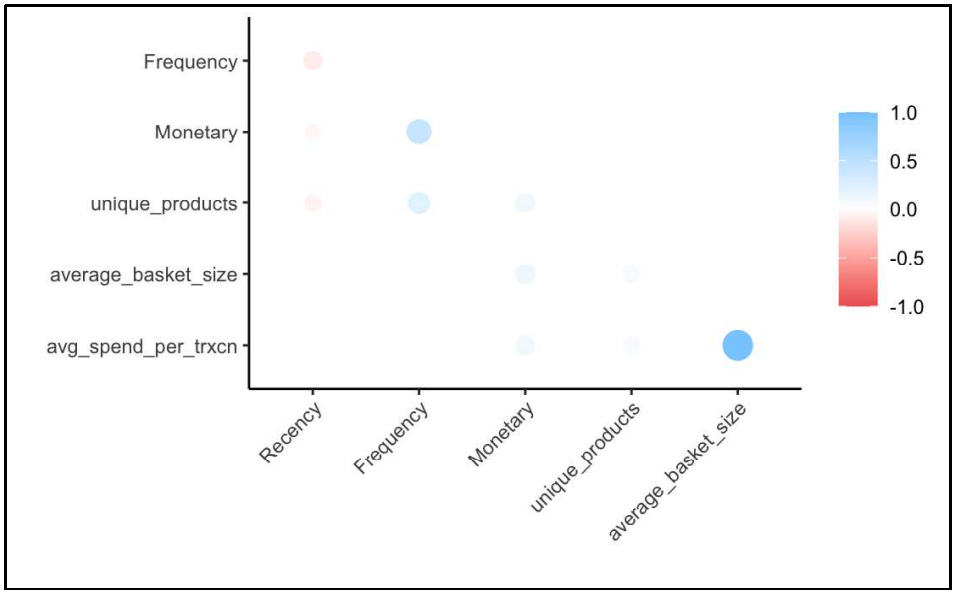

Figure 7: Confidence Intervals for Model_2's variables

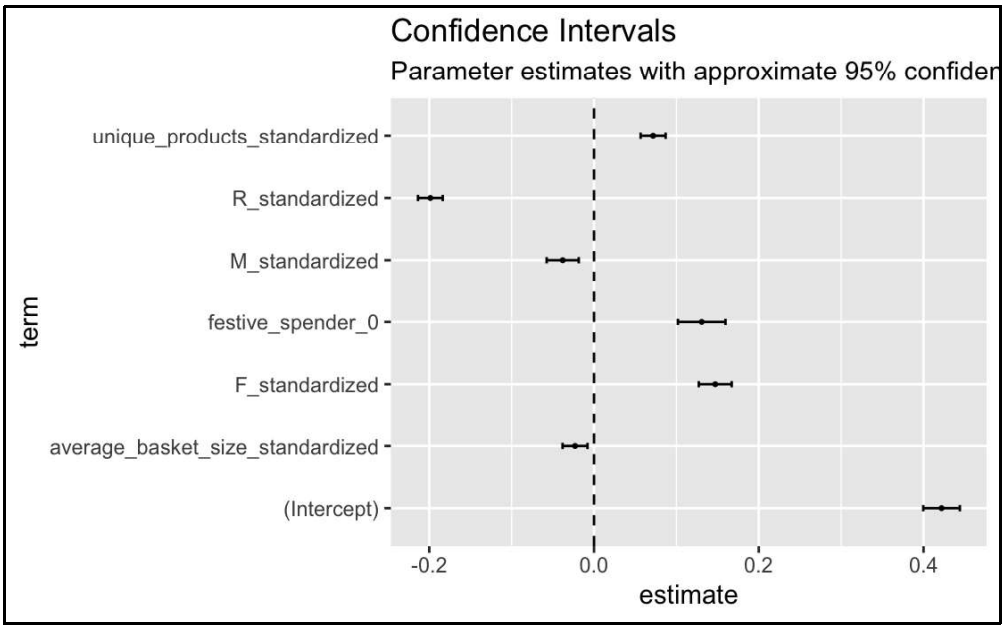

Figure 8: AROC values from Cross-Validation of Model_2

areaUnderROC <dbl>	elastic_net_param_1 <dbl>	reg_param_1 <dbl>
0.9352331	0.00	0.000
0.9352331	0.25	0.000
0.9352331	0.50	0.000
0.9352331	0.75	0.000
0.9352331	1.00	0.000
0.9347354	1.00	0.001
0.9343542	0.75	0.001
0.9338775	0.50	0.001
0.9333375	0.25	0.001
0.9330107	0.00	0.001

Figure 9: Silhouette Scores of K-means clusters

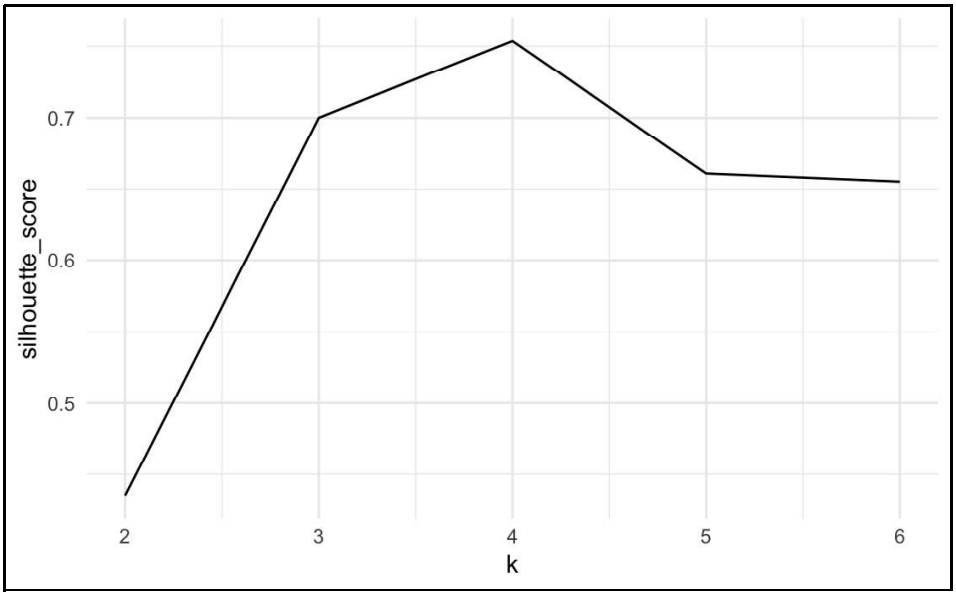

Figure 10: Predicted clusters from K-means modelling

cluster <int>	mean_recency <dbl>	mean_frequency <dbl>	mean_monetary <dbl>
0	6.142857	64.190476	407690.546
1	16.371336	16.775244	49537.912
2	244.989655	1.557759	3441.405
3	44.550266	3.409947	8114.547

Figure 11: **ref_customer**

Source: SQL [?? x 10] Database: spark_connection

CustomerNo <int>	Recency <dbl>	Frequency <dbl>	Monetary <dbl>	duration <dbl>	unique_products <dbl>	average_basket_size <dbl>	avg_spend_per_trxcn <dbl>	festive_spender <chr>	loyal_customer <dbl>
16705	0	20	65877.78	358	133	273.80000	3293.8890	0	1
17581	0	25	56242.15	372	229	237.08000	2249.6860	0	1
13777	0	34	150785.02	373	712	428.38235	4434.8535	1	1
17389	0	34	98926.42	331	45	223.88235	2909.6006	1	1
14520	1	2	930.04	289	3	73.00000	465.0200	0	1
12256	1	2	6606.82	55	56	537.50000	3303.4100	1	0
14135	1	16	43102.87	371	96	241.37500	2693.9294	0	1