Customer Profiling & Lifetime Value

Final Project Report

Group 63

Debanjan Saha

Ritika Rao

781-600-6019

201-892-5836

saha.deb@northeastern.edu

rao.rit@northeastern.edu

Percentage of Effort Contributed by Debanjan: 50%

Percentage of Effort Contributed by Ritika: 50%

Signature of Debanjan: Debanjan Saha

Signature of Ritika: Ritika Rao

Submission Date: 04-21-2023

Code: https://github.com/debanjansaha-git/CustomerProfiling

Table of Contents

Code: https://github.com/debanjansaha-git/CustomerProfiling

1. Problem Setting

Customer Lifetime Value (CLV) is a prediction of the net profit attributed to the entire future relationship with a customer. It is calculated by considering the customer's potential revenue, the cost of acquiring and servicing the customer, and the probability that the customer will continue doing business with the company over time.

Churn prediction, also known as customer attrition prediction, is the process of identifying customers who are likely to cancel their subscription or stop doing business with a company. Customer Churn classifies customers according to their Purchase Intent.

Both CLV and churn prediction are key elements of customer relationship management (CRM) and are used to inform strategic decision making in areas such as marketing, sales, and customer service. The CLV combined with Customer Churn will assist in identifying customer segments that add value but are losing engagement. Ideally, this would be reflected primarily in the customer's information and transaction history.

2. Problem Definition

The goal of this project is to create a generalized holistic customer profile by predicting customer lifetime value (CLV) and churn using historical customer data. By accurately predicting CLV, the company will be able to identify high-value customer segments and tailor marketing campaigns to target these segments. Additionally, by predicting churn, the company can proactively address issues with at-risk customers.

We conducted Market Basket Analysis for the various products items available and look at various combinations of products which could be recommended to the customers., Exploratory Data Analysis to look at various trends and patterns in the data, and RFM segmentation to identify the best clustering algorithms and attributes for correctly categorizing customers as High, Mid, or Low value customers. We then performed churn prediction using Logistic Regression, Random Forest, and XGboost to layer on top of the customer segmentation. Additionally, we predicted customer lifetime value using Linear regression and LARS regression. Finally, we combined all the analyses, including customer demographics data, to create a comprehensive customer profile. The project is completely generalized (plug and play) for any historic transactional data with an objective to help the organizations increase revenue and improve customer retention.

3. Data Sources

The data has been taken from Kaggle *(https://www.kaggle.com/code/darpan25bajaj/retail-casestudy-analysis)*, which is a crowd sourced platform containing various datasets and challenges for data scientists & machine learning engineers to be trained.

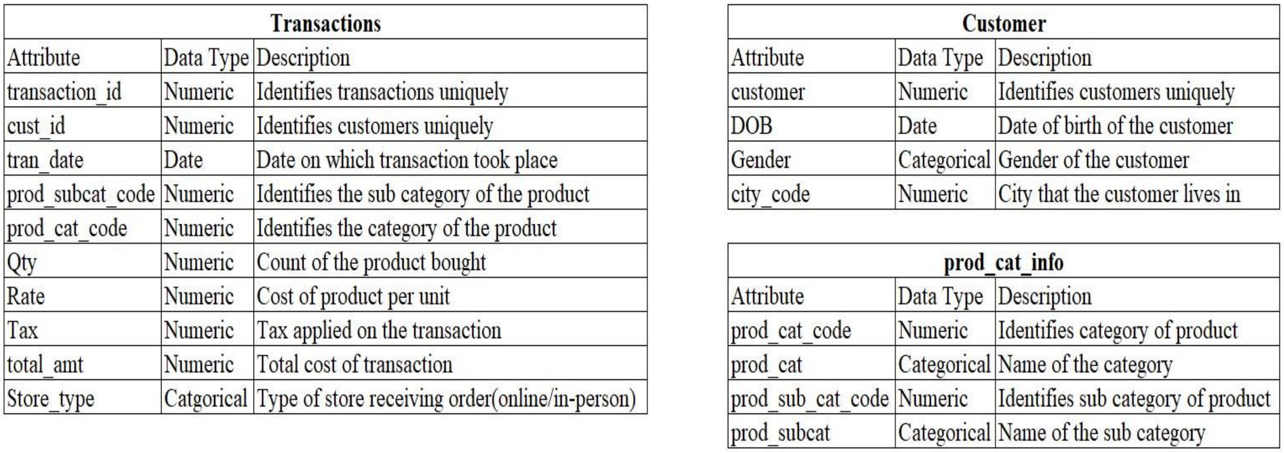
4. Data Description

The data under consideration is that of a retail store's day-to-day transactions and customers across various locations. The dataset contains 3 tables:

'Customer': This table contains demographic data for unique customers. It contains records for 5,647 customers, identified through their customer id.

'Transactions': This table contains the transaction history for this retail store. It has a total of 10 attributes and 23,053 records. The 10 attributes describe the transactions, like the date of transaction, product purchased, amount, etc.

'prod_cat_info': This table consists of the mapping for product category & sub-category codes to the category names. It consists of 24 records and 4 attributes.



Transactions		
Attribute	Data Type	Description
transaction_id	Numeric	Identifies transactions uniquely
cust_id	Numeric	Identifies customers uniquely
tran_date	Date	Date on which transaction took place
prod_subcat_code	Numeric	Identifies the sub category of the product
prod_cat_code	Numeric	Identifies the category of the product
Qty	Numeric	Count of the product bought
Rate	Numeric	Cost of product per unit
Tax	Numeric	Tax applied on the transaction
total_amt	Numeric	Total cost of transaction
Store_type	Catgorical	Type of store receiving order(online/in-person)

Customer		
Attribute	Data Type	Description
customer	Numeric	Identifies customers uniquely
DOB	Date	Date of birth of the customer
Gender	Categorical	Gender of the customer
city_code	Numeric	City that the customer lives in

prod_cat_info		
Attribute	Data Type	Description
prod_cat_code	Numeric	Identifies category of product
prod_cat	Categorical	Name of the category
prod_sub_cat_code	Numeric	Identifies sub category of product
prod_subcat	Categorical	Name of the sub category

Fig 1: Data Description

5. Data Mining Tasks

5.1. Data Understanding

Transactions data: The dataset for retail transaction and customer analysis consists of 23,054 transactions, each identified by a unique 'transaction_id'. Since this is an unsupervised learning problem, there are no target variables to predict. However, there are several categorical variables that can be used, including product categories and store type. While IDs are not typically used as variables in most algorithms, transaction_id and customer_id can be considered categorical variables. The dataset also includes several numerical variables, such as quantity, rate, tax, and amount. Additionally, there is a date variable included in the dataset.

Customer data: The customer dataset contains 5,648 unique rows, each identified by the 'customer_id'. It provides us with detailed information about our customers, including their gender and date of birth, as well as the city they reside in. There are a total of 10 cities, each identified by a unique city code. The city code and gender variables are categorical, while the date of birth is a date variable. This information can be useful for segmenting customers and tailoring marketing strategies to specific demographics.

Product cat info: The product category info just gives us the names of various product categories which are identified by numbers. This can be helpful in identifying associations and patterns between different products. While the numerical categories are easier to process for an algorithm, the names help us make more sense of the transactional data.

5.2. Data Pre-processing:

Upon thorough examination of the data, it was noted that the Customer dataset was missing only 2 values for the city codes. These 2 customers have made a total of 6 transactions in the Transactions dataset. As customer data holds paramount importance in assessing their value and potential to foster long-term business relationships, we deemed it fit to remove the said 2 customers and their 6 transactions from the datasets. The resulting Customer dataset comprises 5646 rows, while the Transactions dataset comprises 23048 rows. To facilitate the process of data analysis and mining, we introduced several new columns to the data. Leveraging the date column, we created additional columns containing information about the month, week, day of the week, and month-year.

For effective customer churn and lifetime value prediction, it is essential to aggregate transaction data at the level of individual customers. Thus, we computed the sum of total amount, quantity, and number of transactions for all unique customers. In addition, we determined the most recent transaction date and the date of the first transaction for each customer, allowing us to calculate the length of their association with our organization and the revenue generated from each customer per unit time. We also recorded the count of returned products and the amount refunded.

5.3. Data Exploration & Analysis

As part of the feature extraction process, we enriched the transaction data by adding month, week, and day of the week columns. We also calculated the ages of the customers based on their date of birth and grouped them to convert them into a categorical variable. To create a comprehensive master dataset, we joined the customers, transactions, and product category datasets. This allowed us to incorporate information such as customer demographics, transaction details, and product categories into a single cohesive dataset. This master dataset has been used for subsequent analysis.

5.3.1. Univariate and Bivariate Analysis

Pairplot and Correlation Heatmap

The pairplot and correlation heatmap provide valuable insights into the relationships and patterns among the numerical variables in the dataset. Negative values in certain columns indicate returns or refunds, which are important factors in the analysis.

Upon examining the pairplot, we can observe that Quantity and Rate exhibit no discernible relationship, and the same holds for Quantity and Tax. However, we can see that Rate, Quantity and Tax, are highly positively correlated with Total Amount.

These observations are reinforced by the correlation heatmap, which provides a visual representation of the correlation coefficients between pairs of variables. It confirms the strong positive correlation between Quantity and Total Amount, as well as Rate and Tax, with correlation coefficients being 0.83 and 1, which were dropped.

Distribution of Quantity & Total Amount

Mean of quantity is 3.003 which is very close to the median 3. The distribution of quantity is relatively symmetric. For the total amount, the fact that the mean and median in this range of values have a difference of approximately 1000, with the mean being lower than the median, suggests that the distribution of values is negatively skewed. In this case, the range of values is quite wide, going from 8287 to 77, which suggests that there may be some extreme values that are contributing to the skewness of the distribution. The fact that the mean is significantly lower than the median indicates that there may be a few very low values that are pulling the mean towards the left, while most of the values are clustered towards the higher end of the range.

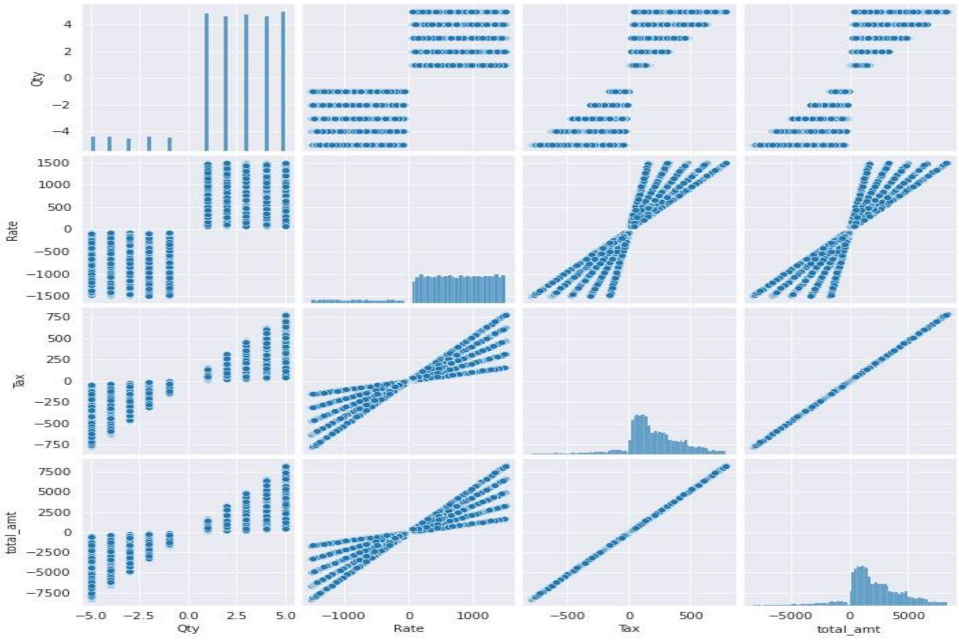


Fig 2: Pair plot of Quantity, Rate, Tax and Total Amount

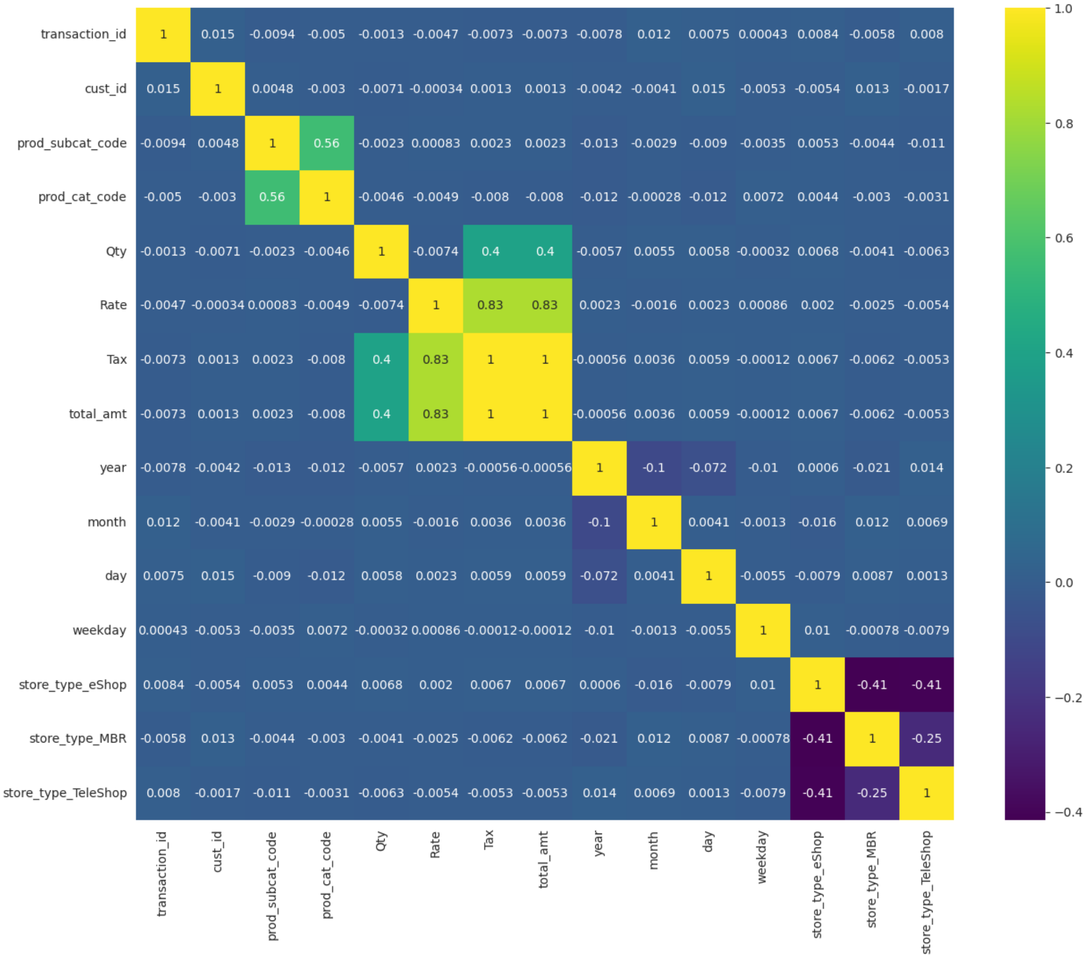


Fig 3: Correlation heatmap for transactions dataset

Code: https://github.com/debanjansaha-git/CustomerProfiling
License: Apache License 2.0

5.3.2. Categorical Variables Analysis

<u>Gender & Age Group</u>*:* The following graphs provide us with a comprehensive understanding of how transactions are distributed across gender, age groups, and product categories. From Figure 1, we can infer that males tend to spend slightly more than females. Additionally, the most active age groups in terms of transaction volume are between 25-35, 35-45, and 45-55, with little difference in spending habits between them.

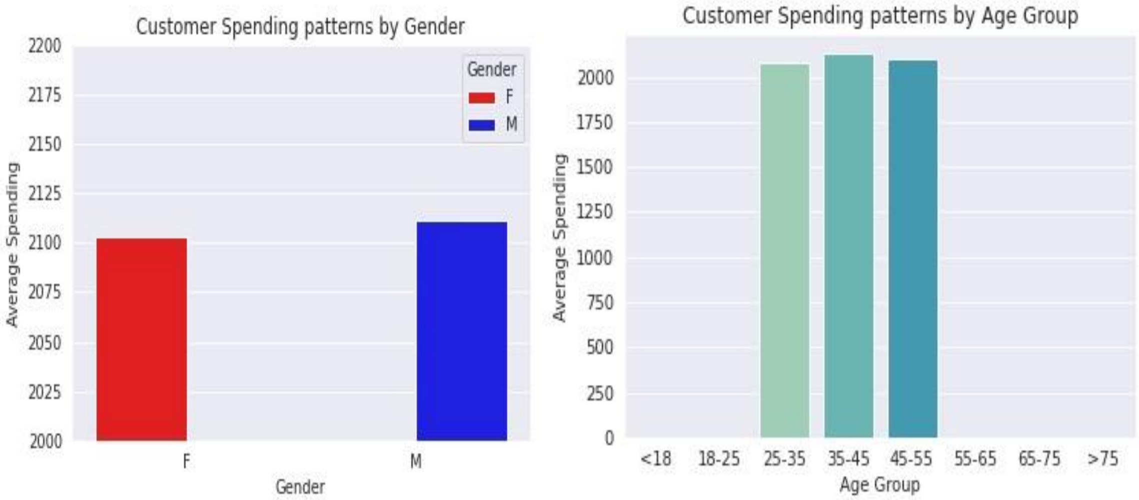


Fig 4: Bar plots for Customer Spending by Gender and Age groups

<u>Product Categories</u>:

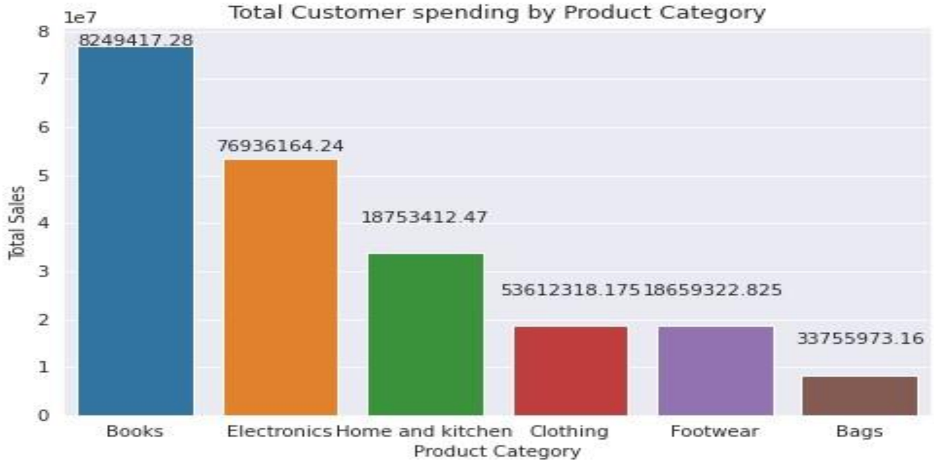


Fig 5: Bar plot for Customer Spending across Product Categories

Fig 5. offers valuable insights into customers' expectations when visiting the website or store. We can see that the most popular product categories are Books, Electronics, and Home/Kitchen Appliances. This information also helps us understand the popular age groups, as Home and Kitchen Appliances are likely to be more popular among settled families, while expensive electronics tend to be more popular among younger people who have just started earning. Overall, this analysis offers a valuable foundation for further exploration and modeling.

City Codes:

The violin plot displaying total revenue from various city codes provides an overview of the variance among the different city codes. However, the plot reveals that each city code has customers with similar buying patterns and does not offer rich information about whether a particular city code has customers who are more likely to buy frequently. Further examination of the customer count across different city codes indicates that the number of customers is approximately the same for each code. Thus, no significant differentiating factor can be found within this categorical variable.

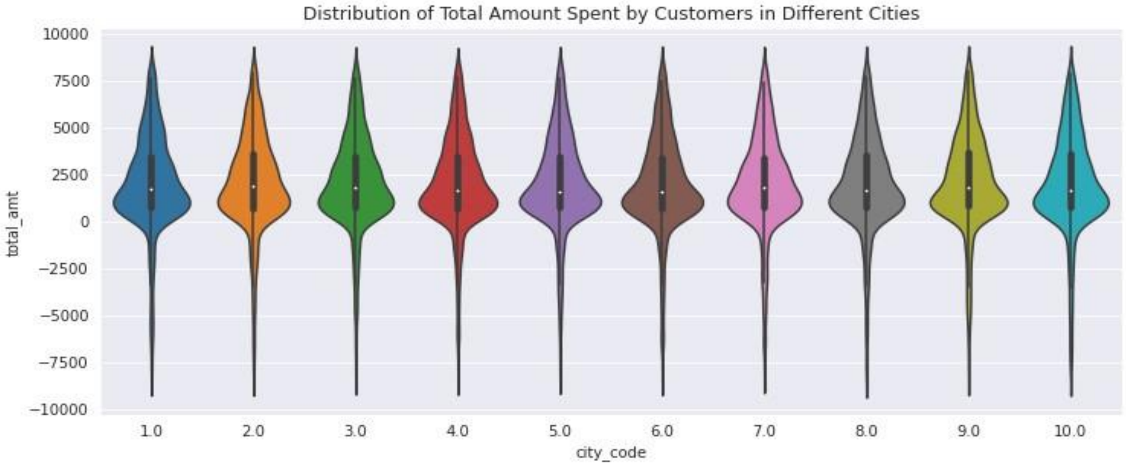


Fig 6: Violin plot for Amount spent by customer in different cities

5.3.3. Sales Trend in Product Category by Store type:

It can be observed from Figure 8 that the sales trend for various product categories is depicted across different types of stores and genders. The graph provides insights into the popularity of each category and the preferred mode of purchase. It is evident that the e-shop is the most favored means of buying for customers. This information can be utilized to enhance the customer experience and improve the e-shop's functionalities. The teleshop, on the other hand, is the second-most popular store for book sales but seems to be less preferred for electronic items. Additionally, the distribution of sales across genders for almost all categories appears to be relatively uniform.

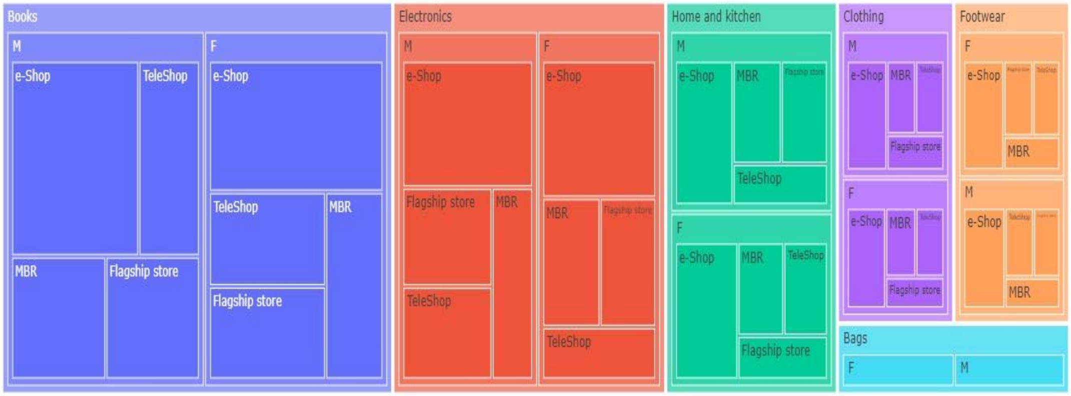
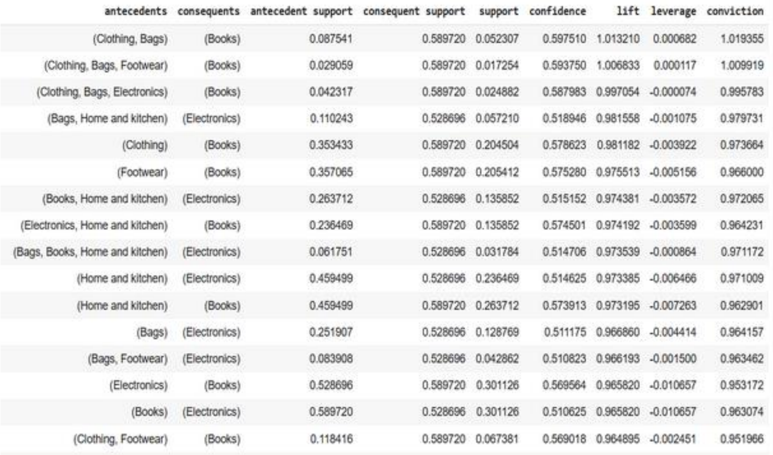


Fig 8: Product Category Sales Trend by Gender and Store Type

5.4. Market Basket Analysis:

It is evident from the graph that customers who are interested in a particular set of categories are also more likely to be interested in books. This valuable insight can be leveraged from a marketing and recommendation perspective by studying the preferred genres of books among different customer demographics, and making personalized book recommendations to customers before they check out. Similarly, placing a dedicated shelf of books in physical stores can also drive sales. Electronics is the other popular category that customers tend to purchase with other categories. Although the specific type of gadget purchased is not available in the dataset, we can deduce that it is likely to be a smaller gadget, as customers are less likely to make impulsive purchases on high-ticket electronics. Identifying the popular gadgets can also assist in pitching similar products to customers right before checkout.



antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
(Clothing, Bags)	(Books)	0.087541	0.589720	0.052307	0.597510	1.013210	0.000682	1.019355
(Clothing, Bags, Footwear)	(Books)	0.029059	0.589720	0.017254	0.593750	1.006833	0.000117	1.009919
(Clothing, Bags, Electronics)	(Books)	0.042317	0.589720	0.024882	0.587983	0.997054	-0.000074	0.995783
(Bags, Home and kitchen)	(Electronics)	0.110243	0.528696	0.057210	0.518946	0.981558	-0.001075	0.979731
(Clothing)	(Books)	0.353433	0.589720	0.204504	0.578623	0.981182	-0.003922	0.973664
(Footwear)	(Books)	0.357065	0.589720	0.205412	0.575280	0.975513	-0.005156	0.966000
(Books, Home and kitchen)	(Electronics)	0.263712	0.528696	0.135852	0.515152	0.974381	-0.003572	0.972065
(Electronics, Home and kitchen)	(Books)	0.236469	0.589720	0.135852	0.574501	0.974192	-0.003599	0.964231
(Bags, Books, Home and kitchen)	(Electronics)	0.061751	0.528696	0.031784	0.514706	0.973539	-0.000864	0.971172
(Home and kitchen)	(Electronics)	0.459499	0.528696	0.236469	0.514625	0.973385	-0.006466	0.971009
(Home and kitchen)	(Books)	0.459499	0.589720	0.263712	0.573913	0.973195	-0.007263	0.962901
(Bags)	(Electronics)	0.251907	0.528696	0.128769	0.511175	0.966860	-0.004414	0.964157
(Bags, Footwear)	(Electronics)	0.083908	0.528696	0.042862	0.510823	0.966193	-0.001500	0.963462
(Electronics)	(Books)	0.528696	0.589720	0.301126	0.569564	0.965820	-0.010657	0.953172
(Books)	(Electronics)	0.589720	0.528696	0.301126	0.510625	0.965820	-0.010657	0.963074
(Clothing, Footwear)	(Books)	0.118416	0.589720	0.067381	0.569018	0.964895	-0.002451	0.951966

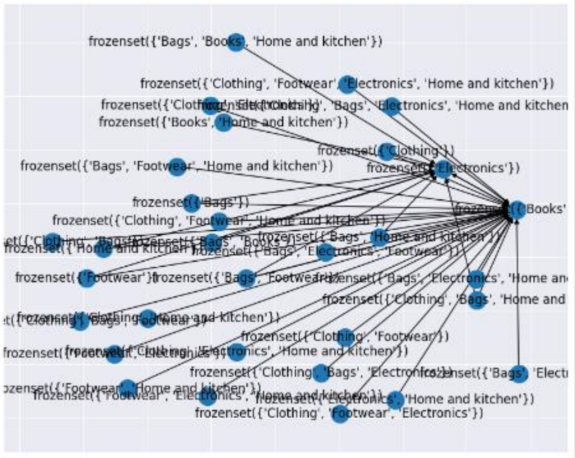


Fig 7: Network graph showing product categories bought together frequently.

Code: https://github.com/debanjansaha-git/CustomerProfiling
License: Apache License 2.0

5.5. Customer Segmentation based on total amount:

Fig 9 represents customer segmentation although, it should be noted that the customer segmentation depicted in the graph is based solely on the total amount of purchases made by each customer, and therefore may not be a fully accurate reflection of the potential value that each customer may provide to the business. Further analysis using more sophisticated data mining techniques, incorporating additional contextual factors may be necessary to make more precise predictions regarding customer value and potential.

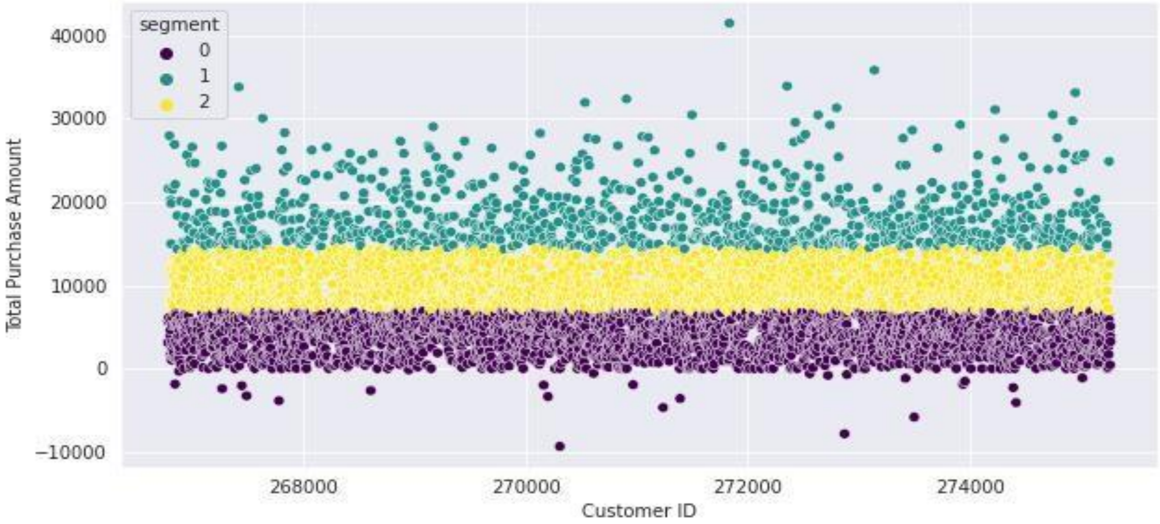


Fig 9: Customer Segmentation based on total amount

5.6. Time Series Analysis:

The plot in Fig 10. helps identify the regular buying pattern of the top customers over time. Any deviation from their regular pattern can be an indication of their dissatisfaction with the product or service provided by the business. This information can be used to take corrective actions and retain the loyalty of these top customers. Additionally, this plot can also be used to identify the best time to run promotions and offer discounts to these loyal customers, which can further improve their buying behavior and increase their lifetime value for the business.

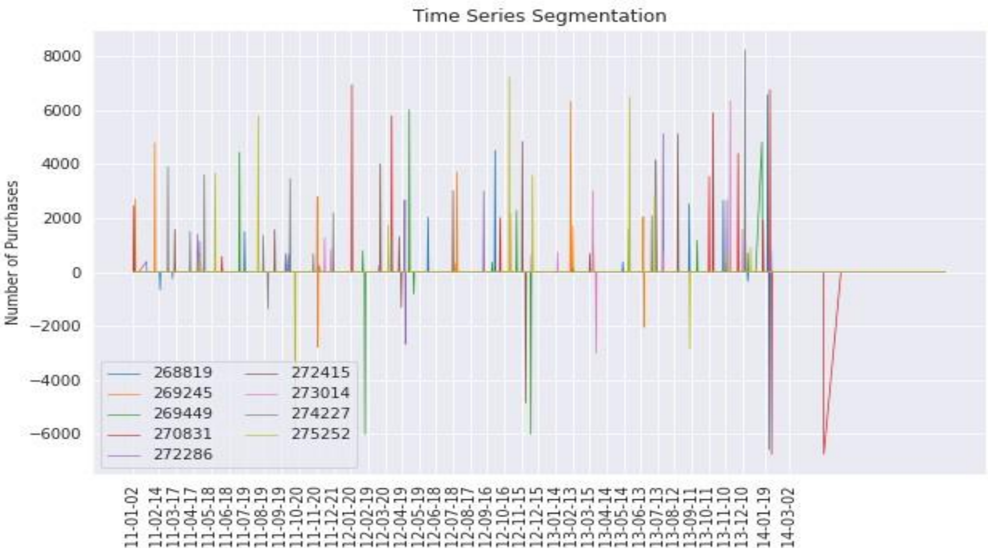


Fig 10. Time series of purchases and returns for 9 most loyal customers

6. Feature Selection:

In our dataset, we aimed to predict Customer Lifetime Value using Total Amount, which represents the revenue generated from each transaction. Recursive feature engineering was performed using linear regression and random forest regressor. However, the outputs for both methods differed significantly, indicating that the importance of certain features varied between the two algorithms. While features such as Quantity, Rate, and Tax were deemed important by both methods since we know Total Amount = Quantity * Rate + Tax, store_type was considered important by linear regression but not by random forest regressor. Given these discrepancies, we opted to keep all features in our dataset to avoid potential loss of information.

7. Model Selection:

7.1. KMeans++ Clustering

KMeans clustering is a widely used unsupervised machine learning algorithm that groups data points into K clusters based on their similarities. This is done by minimizing the variance between data points and the centroids of a cluster.

This report presents the application of the RFM method to segment customers based on their value and identify churned-out customers. To achieve this, we first run K Means clustering for each of the criteria in RFM individually and combine the scores to obtain an overall score for each customer.

Recency: The recency is given by the difference in the number of days passed since the last order of a customer to our most recent order, we use this data to cluster our observations into groups of customer. We determine the optimal number of clusters for recency using an elbow graph, which yields 3 clusters with means of 356, 641, and 1057. However, following box plot indicates that a cut-off value between 1100-1200 is necessary to identify low-value customers that require attention.

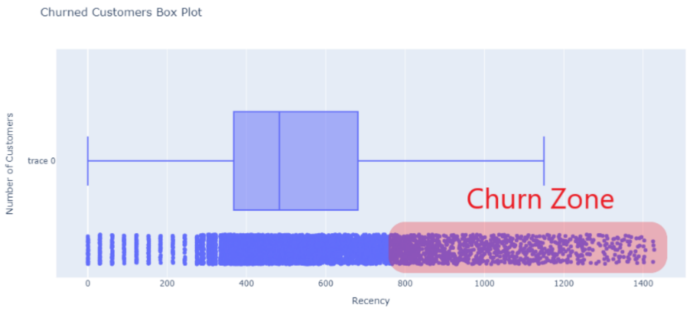
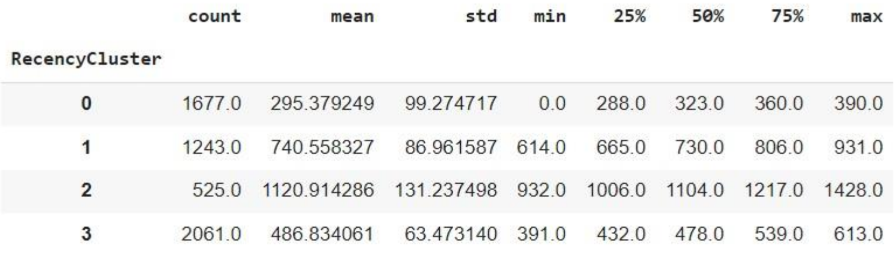


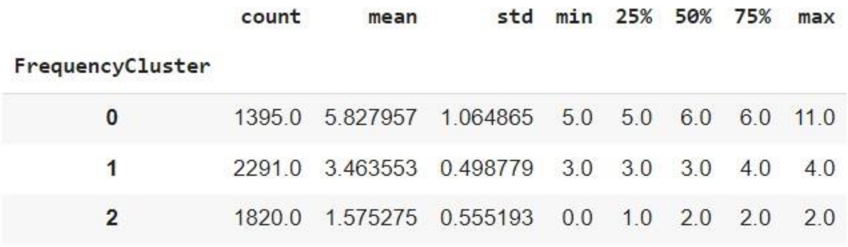
Fig 12. Distribution of customers according to recency

Thus, we perform clustering with K=4 and successfully identify churned-out customers with a mean score of 1100 or more. Clustering with K=5 yields no additional insights. Below are the means we arrive at for clusters given K=4:



	count	mean	std	min	25%	50%	75%	max
RecencyCluster								
0	1677.0	295.379249	99.274717	0.0	288.0	323.0	360.0	390.0
1	1243.0	740.558327	86.961587	614.0	665.0	730.0	806.0	931.0
2	525.0	1120.914286	131.237498	932.0	1006.0	1104.0	1217.0	1428.0
3	2061.0	486.834061	63.473140	391.0	432.0	478.0	539.0	613.0

Frequency: We derive the frequency by taking a sum total of the number of transactions performed by each customer. We try K=3 and K=4, and settle at K=3 as the change is information is not as high among the 2 choices. Below is the description of clusters based on Frequency where K = 3:



	count	mean	std	min	25%	50%	75%	max
FrequencyCluster								
0	1395.0	5.827957	1.064865	5.0	5.0	6.0	6.0	11.0
1	2291.0	3.463553	0.498779	3.0	3.0	3.0	4.0	4.0
2	1820.0	1.575275	0.555193	0.0	1.0	2.0	2.0	2.0

The discrepancy in range of cluster ‘0’ [5,11], can be explained since the distribution of the number of customers across the Frequency is right skewed as shown below:

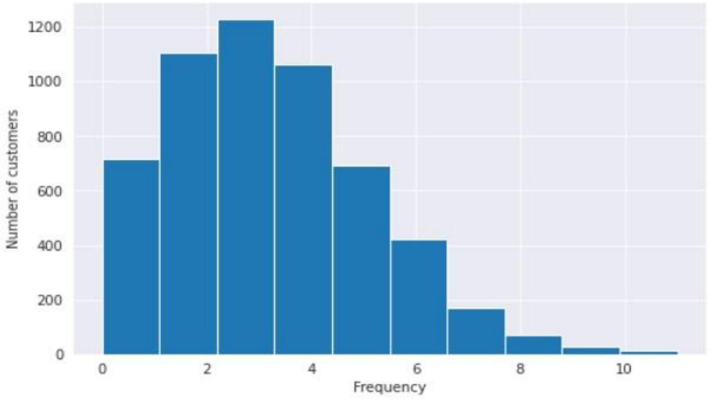
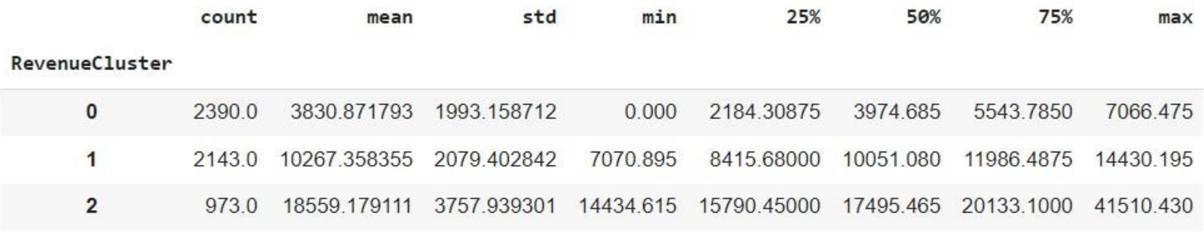


Fig 15：Histogram of frequency vs number of customers

Revenue: Revenue is given as the sum of total purchase amount per customer. We again perform an elbow test for getting optimal number of clusters, we get K=3 from the test.

We repeat the same steps as above for revenue and get the following results:



	count	mean	std	min	25%	50%	75%	max
RevenueCluster								
0	2390.0	3830.871793	1993.158712	0.000	2184.30875	3974.685	5543.7850	7066.475
1	2143.0	10267.358355	2079.402842	7070.895	8415.68000	10051.080	11986.4875	14430.195
2	973.0	18559.179111	3757.939301	14434.615	15790.45000	17495.465	20133.1000	41510.430

7.2. Agglomerative Clustering

Agglomerative clustering is a hierarchical clustering algorithm that merges data points iteratively until they belong to a single cluster. To identify subgroups in a dataset, the dendrogram can be cut at a certain height. In this dataset, we apply agglomerative clustering by combining Recency, Frequency, and Monetary features and settling for 3 clusters, which is determined from the dendrogram. The clustering results indicated that the Revenue criteria have been accurately clustered. However, the Recency clustering is suboptimal, while the clustering based on Frequency is moderately accurate. We experimented with various linkage methods, including complete, ward, average, and single, and found that the ward linkage method produced the best results. Nevertheless, the Recency clusters remain unsatisfactory, and the clustering based on Frequency is only moderately accurate.

7.3. DBScan Clustering

We utilized DBSCAN to cluster the RFM criteria of the dataset, and determined optimal values for minPts and eps through an elbow curve. However, the clustering results were sensitive to parameter choice, and inaccurate for both frequency and revenue clusters. The Recency clusters produced six clusters, but five of them had a uniform count of observations and one had a significantly higher count, indicating imbalance in data distribution. The frequency clusters generated too many clusters for smaller eps values and only one cluster for the optimal eps value, indicating inability to accurately cluster based on frequency. The Revenue clusters also showed a similar issue with one cluster having a significantly higher count of observations and a wider range than the rest. In summary, DBSCAN did not accurately cluster all three criteria.

7.4. SMOTE:

Using data from RFM clustering, we identified churned customers as those who had not made a purchase in the past two years, resulting in an imbalanced dataset with 1146 churners and 4360 non-churners. To address this issue, we used the SMOTE algorithm to oversample the minority class and create synthetic samples based on existing samples. SMOTE selects a random sample from the minority class, computes the k-nearest neighbors, and interpolates between the original sample and its neighbors to generate new samples. Through this process, we increased the number of churners in the dataset and balanced the distribution of the two classes, resulting in a final oversampled dataset containing 6104 observations from both classes. By using SMOTE, we improved the performance of our machine learning models in predicting churn..

7.5. XGBoost:

We used XGBoost, an ensemble machine learning algorithm, to predict churn from the dataset. The first step was to preprocess the data, encode categorical variables, and split into training and testing sets. SMOTE was then used to address dataset imbalance. Next, we trained an XGBoost classifier on the training data using 10 fold cross-validation and performed hyperparameter tuning using GridSearchCV. The best hyperparameters for the XGBoost model were found, and the model was evaluated on the testing data using classification metrics and a confusion matrix. The results showed that XGBoost is a powerful algorithm for churn prediction in this dataset, achieving high accuracy and F1 score.

7.6. Logistic Regression:

We employed logistic regression to classify customers who have churned out due to no purchase data in the past 2 years. SMOTE was used to address the imbalanced data issue and enhance model performance. Both XGBoost and logistic regression models were evaluated using cross-validation. Despite both models achieving high accuracy and F1 scores, XGBoost outperformed logistic regression. By applying SMOTE and cross-validation techniques, we improved model performance and achieved better results.

7.7. Linear Regression Models:

In order to predict customer lifetime value (CLV), we first performed data preprocessing on the transaction data to extract relevant features such as recency, frequency, and monetary value.

Next, we trained and tested several linear regression models including Linear Regression, Stochastic Gradient Descent Regressor, Lasso Regression, Ridge Regression, Least-Angle Regression (LARS), LARS Lasso, RidgeCV, Huber Regressor, Poisson Regressor, KNN Regressor, Random Forest Regressor, Gradient Boosting Regressor, Light GBM, Adaboost, XGBoost and Transformed Regressor. We evaluated the performance of each model using a variety of metics such as R-squared, adjusted R-squared, Root Mean Squared Error (RMSE), median absolute error, Spearman's Rank Coefficient and many more. We found that the Linear Regression model had the highest adjusted R-squared value and the lowest RMSE and median absolute error indicating that it was the best model for predicting CLV. We also used feature importance scores to identify the most important features for predicting CLV, which were recency and frequency.

8. Performance Evaluation:

8.1. Clustering Customer

We used three clustering algorithms: Agglomerative, KMeans++, and DBScan. To determine the optimal number of clusters, we employed the elbow method, which identifies the point at which the error rate drops. Additionally, we relied on our domain knowledge based on the cluster metrics observed later. Below are the elbow plots for KMeans++ and DBScan (we have only considered Recency for the example), as well as the dendrogram for Hierarchical Clustering:

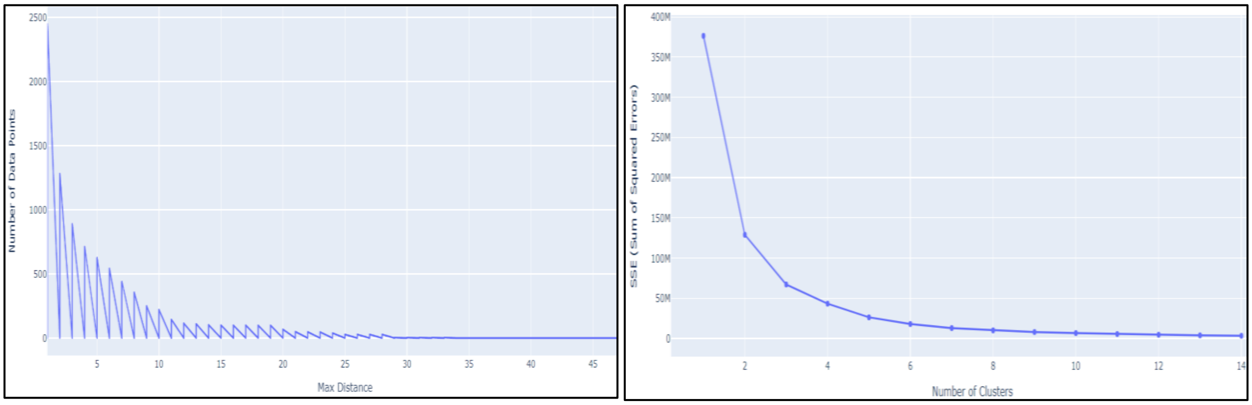


Fig16: Elbow Curve for DBScan - Recency Cluster (left) KMeans++ – Recency Cluster (right)

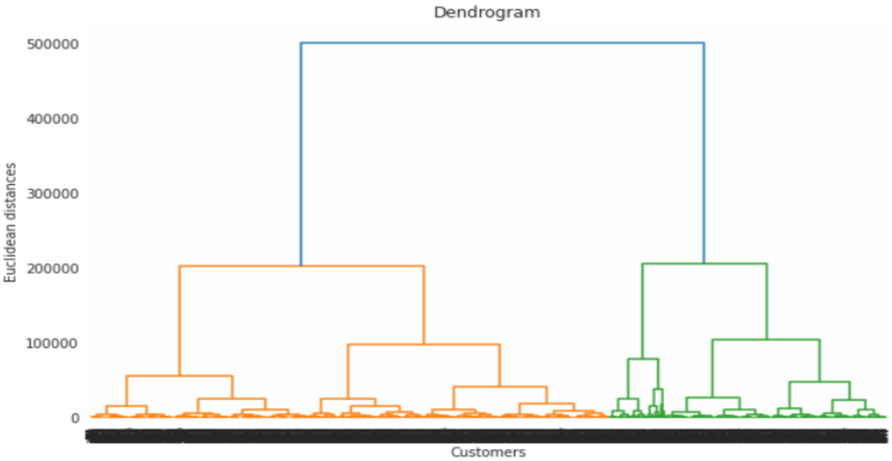


Fig 17: Dendrogram for hierarchical clustering

We evaluated the performance using the Dunn Index and Silhouette Score for both Hierarchical and KMeans, as they appeared to have the best performance out of the three.

Below are the results we obtained:

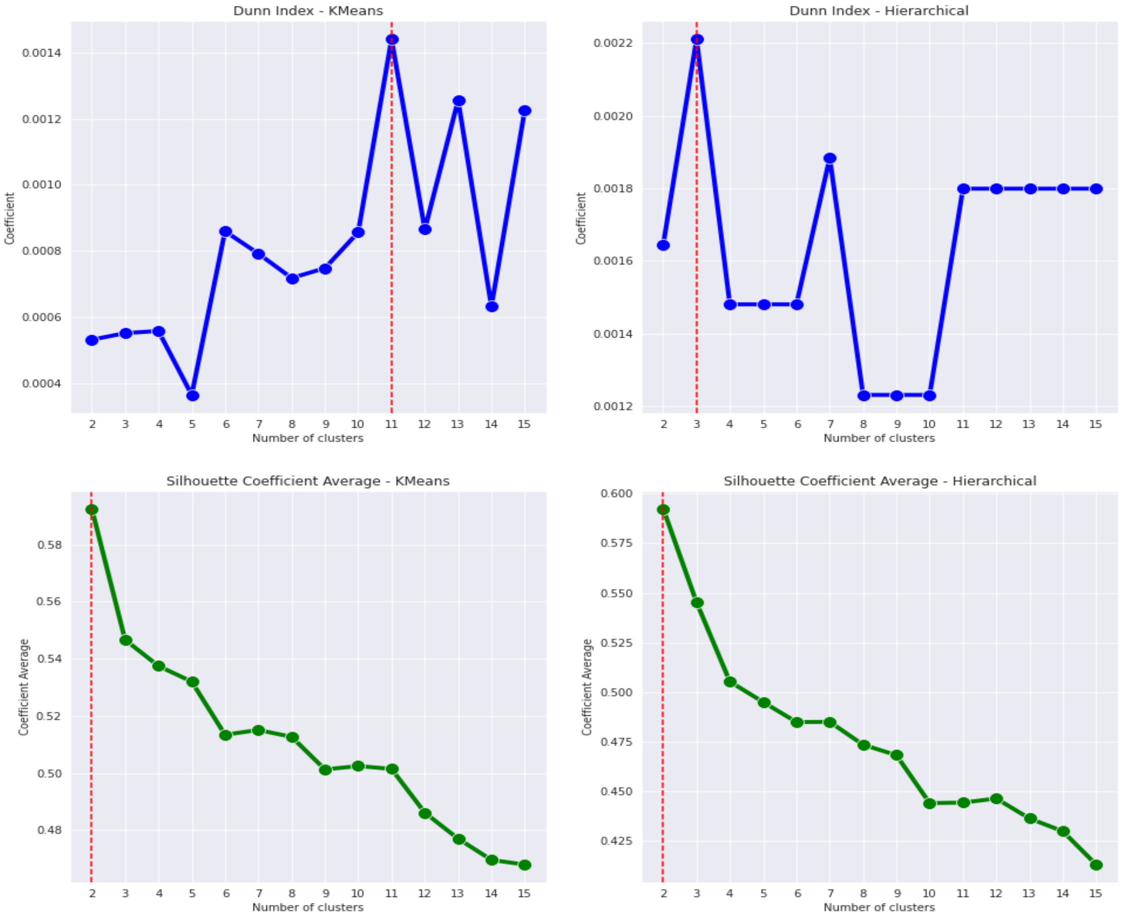


Fig 18: Silhouette score and Dunn Index for KMean(left) and Hierarchical(right)

As we can observe, the curves indicate slightly better performance in hierarchical clustering. However, before we use this information to make a choice, we need to understand the intuition behind the two measures and determine if it applies to our specific case. Intuitively, the silhouette score and Dunn index indicate how well separated and compact the classes are by measuring inter-cluster and intra-cluster distances.

These metrics can be misleading in some cases:

- The Silhouette Index can be misleading if the clusters have uneven sizes, in which case the scores will be high even if the sparse clusters are not well separated, and the majority of the larger clusters are well separated. This is evident in the case of hierarchical clusters, as shown in the following graphs:

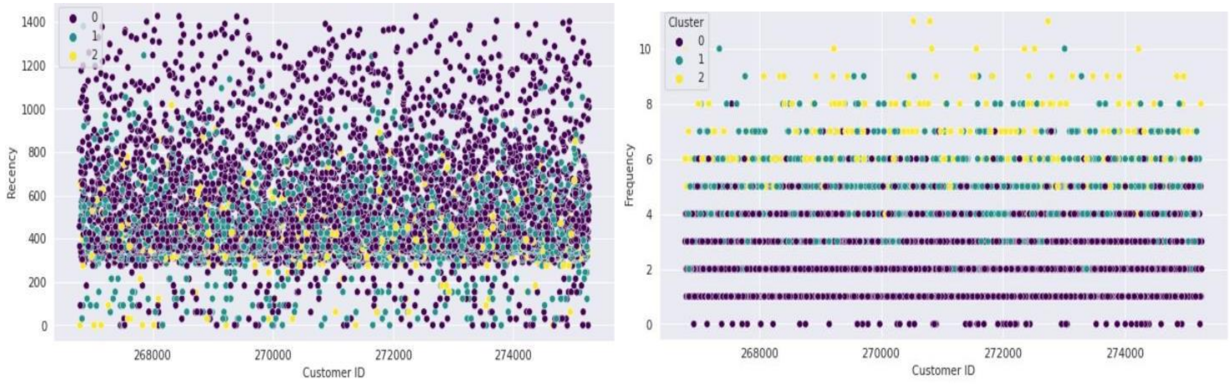


Fig 19: Clusters based on Recency(left) and Frequency(right) for hierarchical clustering

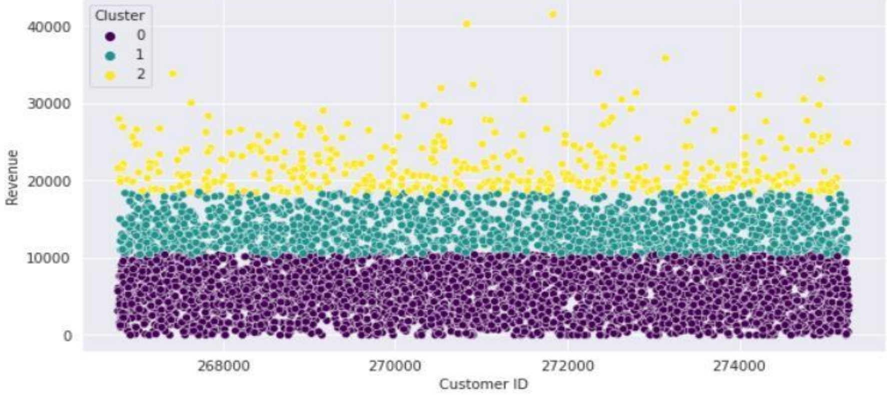


Fig 20: Clusters based on Revenue for hierarchical clustering

- The Dunn Index can be misleading when the inter-cluster distance is very small, even if the clusters are well-separated, resulting in a higher Dunn Index value. Additionally, if the cluster densities are different or the cluster sizes are sparse, it can also be misleading. This can be observed in the following graphs for KMeans clustering:

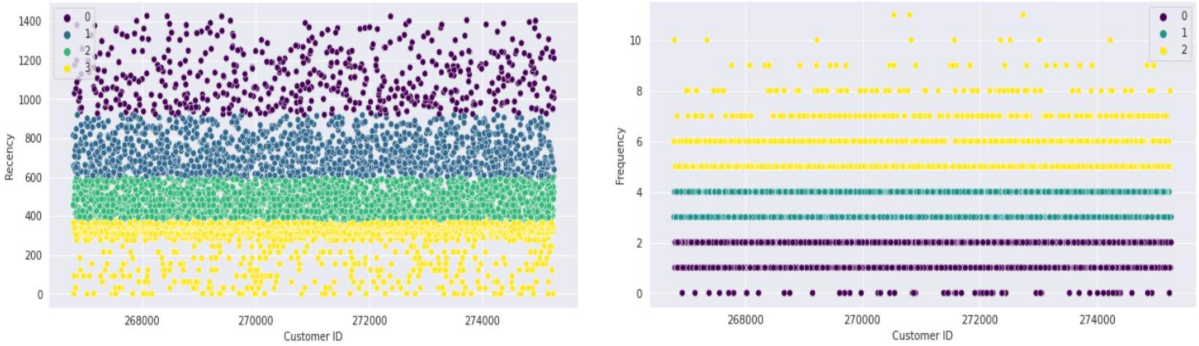


Fig 21: Clusters based on Recency(left) and Frequency(right) for KMeans++ Clustering

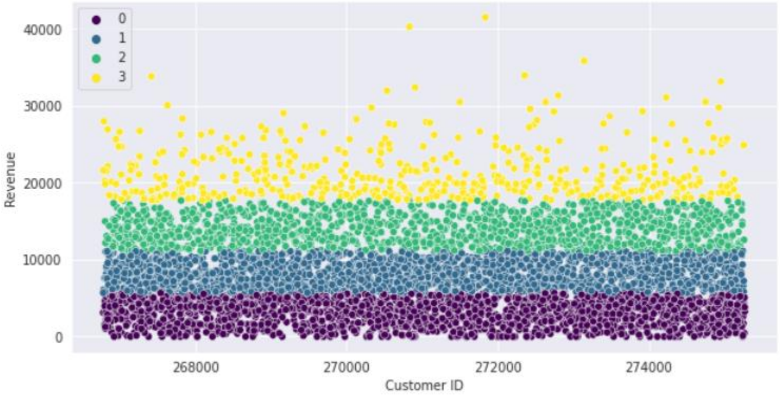


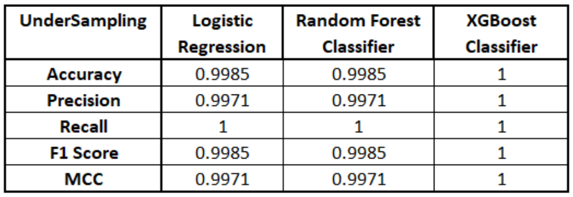
Fig 22: Clusters based on Revenue for KMeans++ Clustering

Now, with the context of the metrics and the visualization of the clusters, it is evident that the clusters are more distinct with very little separation in KMeans++.

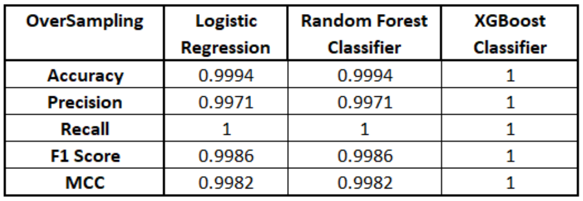
8.2. Customer Churn Prediction (Classification)

The Customer Churn metric is a simple indicator based on the recency of purchase from customers. Considering the straightforward nature of the problem, we began by using Logistic Regression, Random Forest Classifier (after undersampling and oversampling was performed, since the classes were unbalanced) to solve it. There was only one misclassification, which was resolved when we switched to the XGBoost Classifier.

Performance Evaluation Metrics for classification are as given below:



UnderSampling	Logistic Regression	Random Forest Classifier	XGBoost Classifier
Accuracy	0.9985	0.9985	1
Precision	0.9971	0.9971	1
Recall	1	1	1
F1 Score	0.9985	0.9985	1
MCC	0.9971	0.9971	1



OverSampling	Logistic Regression	Random Forest Classifier	XGBoost Classifier
Accuracy	0.9994	0.9994	1
Precision	0.9971	0.9971	1
Recall	1	1	1
F1 Score	0.9986	0.9986	1
MCC	0.9982	0.9982	1

i. ROC

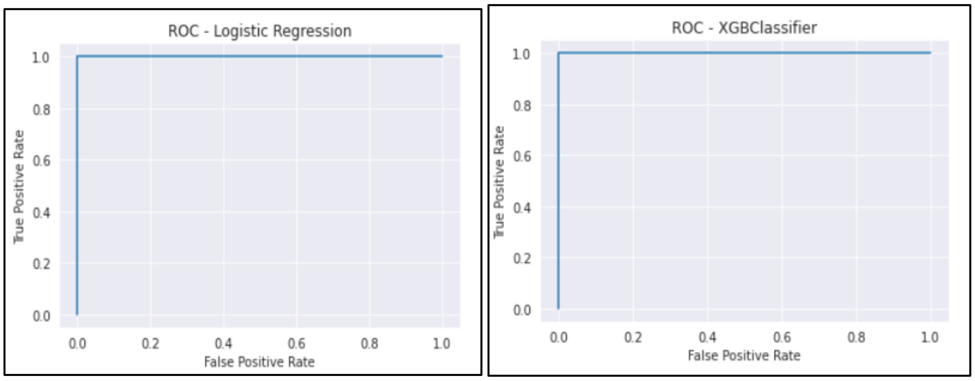


Fig 23: ROC Curve for Logistic Regression and XGBClassifier

Code: https://github.com/debanjansaha-git/CustomerProfiling
License: Apache License 2.0

From the above metrics and ROC we can derive the fact that the model is able to decipher the exact relationship between input variables and target variables. This, is due to the fact that the relationship is simple.

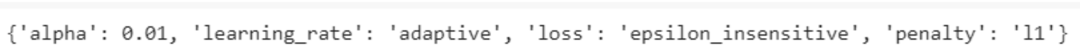
8.3. Customer Lifetime Value Prediction (Regression)

In order to predict the Customer Lifetime Value, we employed several regression models. As a part of our model selection process, we conducted hyperparameter tuning for certain algorithms to determine if any of them offered improved performance. The following section presents the results of our analysis:

SGD Regressor:

To enhance the performance of the SGDRegressor algorithm, we utilized GridSearchCV for hyperparameter tuning. We present the information on the optimal parameters and the observed reduction in error below:

Best Parameters



```
{'alpha': 0.01, 'learning_rate': 'adaptive', 'loss': 'epsilon_insensitive', 'penalty': 'l1'}
```

Prior to hyperparameter tuning, the Root Mean Squared Error (RMSE) was recorded as 219672554947157.9062. Following hyperparameter tuning, the RMSE decreased significantly to 7323.9049. Though this error value does not surpass the best RMSE recorded during the project, it is a noteworthy improvement that indicates the success of the model optimization process.

KNN Regressor: In order to determine the optimal value of k for our model, we conducted an iterative process to evaluate the error on the validation set while varying k from 1 to 1000. We observed that the number of nearest neighbors between 6 and 10 resulted in the lowest validation error. The following section presents the observed RMSE values for various distance functions:

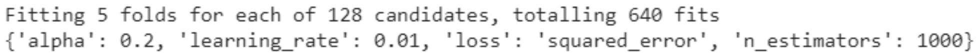
Chebyshev = 4214.4135

Euclidean = 4128.3531

Manhattan = 4162.6955

LGBM Regressor: Similar to the SGD Regressor, we utilized GridSearchCV to identify optimal parameters for the LGBM Regressor model. The results of our analysis are presented below:

Best Parameters



```
Fitting 5 folds for each of 128 candidates, totalling 640 fits
{'alpha': 0.2, 'learning_rate': 0.01, 'loss': 'squared_error', 'n_estimators': 1000}
```

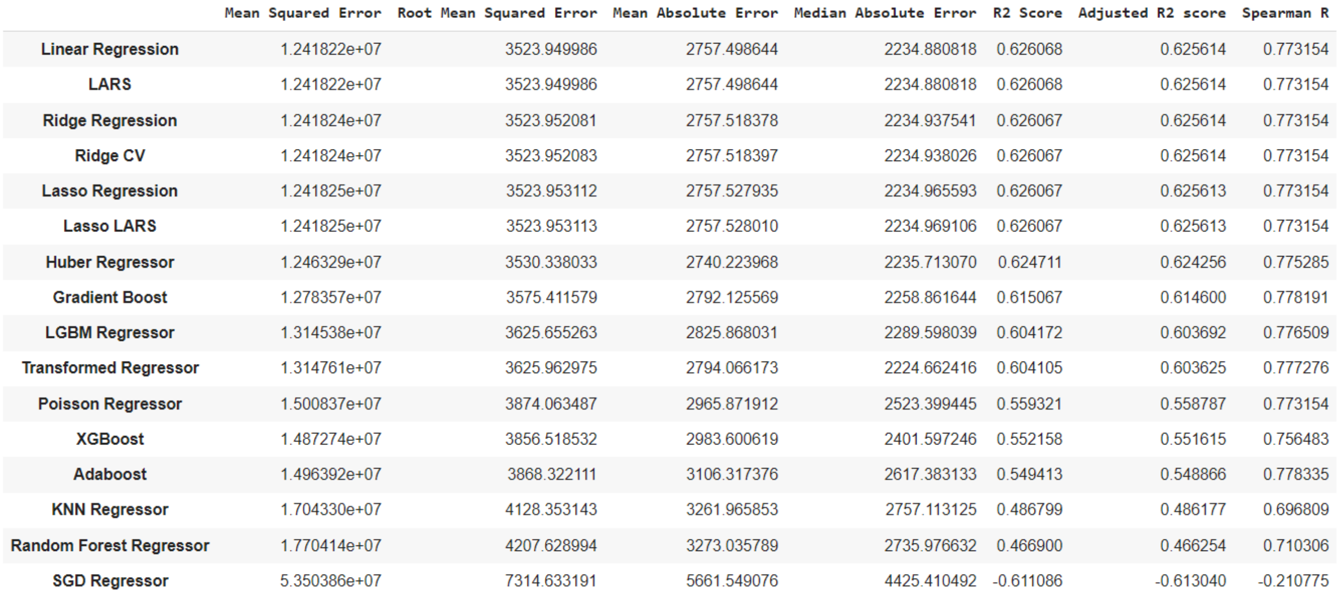
Prior to hyperparameter tuning, the RMSE was recorded as 3628.4738. c. Following hyperparameter tuning, the RMSE improved slightly to 3625.6553. It should be noted that, in comparison to the SGD Regressor, the LGBM Regressor did not exhibit a significant reduction in error even after hyperparameter tuning.

Lasso & Ridge Regression

a. For Lasso Regression, with alpha ranging from [0.1 , 2], we observed very little change in the R2 score to get the goodness-of-fit of the model, which stayed approximately 0.638 for all values of alpha. Thus, indicating that applying a regularization term to the cost function doesn't improve the model fit.

b. With Ridge Regression, a very similar trend was observed with the R2 score, which stayed approximately 0.638 for all alphas.

After exploring the best model for every algorithm, we evaluated on the basis of Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, Median Absolute Error, R2 Score, Adjusted R2 Score, Spearman Correlation Coefficient. Below is a table compiling the performance of all the algorithms:



	Mean Squared Error	Root Mean Squared Error	Mean Absolute Error	Median Absolute Error	R2 Score	Adjusted R2 score	Spearman R
Linear Regression	1.241822e+07	3523.949986	2757.498644	2234.880818	0.626068	0.625614	0.773154
LARS	1.241822e+07	3523.949986	2757.498644	2234.880818	0.626068	0.625614	0.773154
Ridge Regression	1.241824e+07	3523.952081	2757.518378	2234.937541	0.626067	0.625614	0.773154
Ridge CV	1.241824e+07	3523.952083	2757.518397	2234.938026	0.626067	0.625614	0.773154
Lasso Regression	1.241825e+07	3523.953112	2757.527935	2234.965593	0.626067	0.625613	0.773154
Lasso LARS	1.241825e+07	3523.953113	2757.528010	2234.969106	0.626067	0.625613	0.773154
Huber Regressor	1.246329e+07	3530.338033	2740.223968	2235.713070	0.624711	0.624256	0.775285
Gradient Boost	1.278357e+07	3575.411579	2792.125569	2258.861644	0.615067	0.614600	0.778191
LGBM Regressor	1.314538e+07	3625.655263	2825.868031	2289.598039	0.604172	0.603692	0.776509
Transformed Regressor	1.314761e+07	3625.962975	2794.066173	2224.662416	0.604105	0.603625	0.777276
Poisson Regressor	1.500837e+07	3874.063487	2965.871912	2523.399445	0.559321	0.558787	0.773154
XGBoost	1.487274e+07	3856.518532	2983.600619	2401.597246	0.552158	0.551615	0.756483
Adaboost	1.496392e+07	3868.322111	3106.317376	2617.383133	0.549413	0.548866	0.778335
KNN Regressor	1.704330e+07	4128.353143	3261.965853	2757.113125	0.486799	0.486177	0.696809
Random Forest Regressor	1.770414e+07	4207.628994	3273.035789	2735.976632	0.466900	0.466254	0.710306
SGD Regressor	5.350386e+07	7314.633191	5661.549076	4425.410492	-0.611086	-0.613040	-0.210775

Fig 24: Table showing various performance evaluation metric for all algorithms implemented

Based on the evaluation metrics, it is evident that Linear Regression is the optimal choice among all the algorithms tested. To gain further insight into the performance of the model, we used additional evaluation techniques. The following section elaborates on our findings:

i. **R2 score**: An R2 score of 0.638 indicates that the model explains about 63.8% of the variance in the dependent variable based on the independent variables included in the model. It means that the model has a moderate to good fit, but there is still about 36.2% of the variance in the dependent variable that is not explained by the model. Therefore, further improvements to the model may be necessary to increase its predictive power.

ii. Residual plot

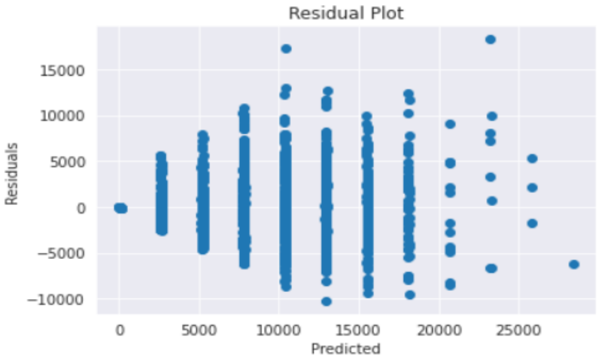


Fig 25: Residual Plot for Linear Regression (vanilla)

This plot gives us the errors plotted against predicted values, as we can see the higher predictions have higher errors while lower ranged predictions are more accurate.

iii. **Actual vs predicted**

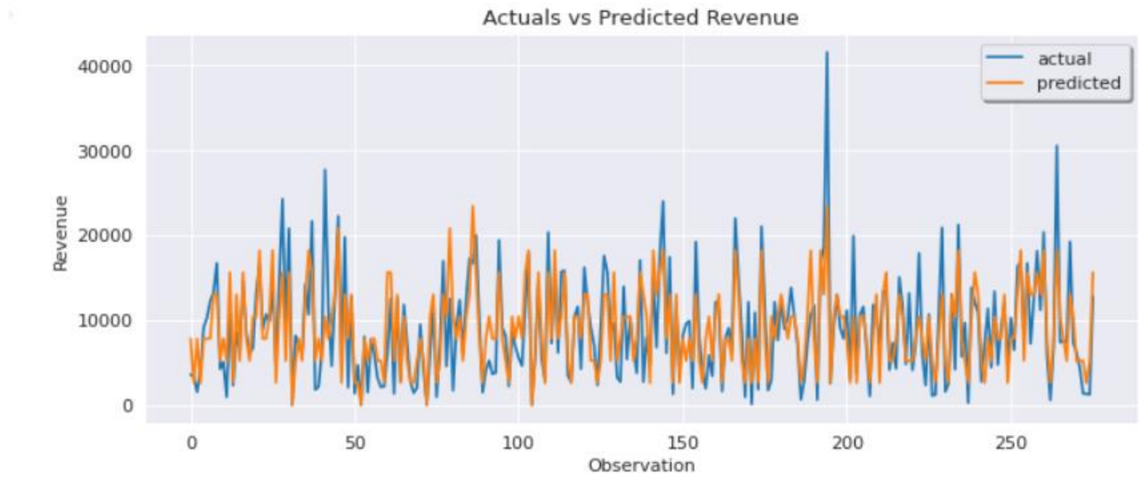
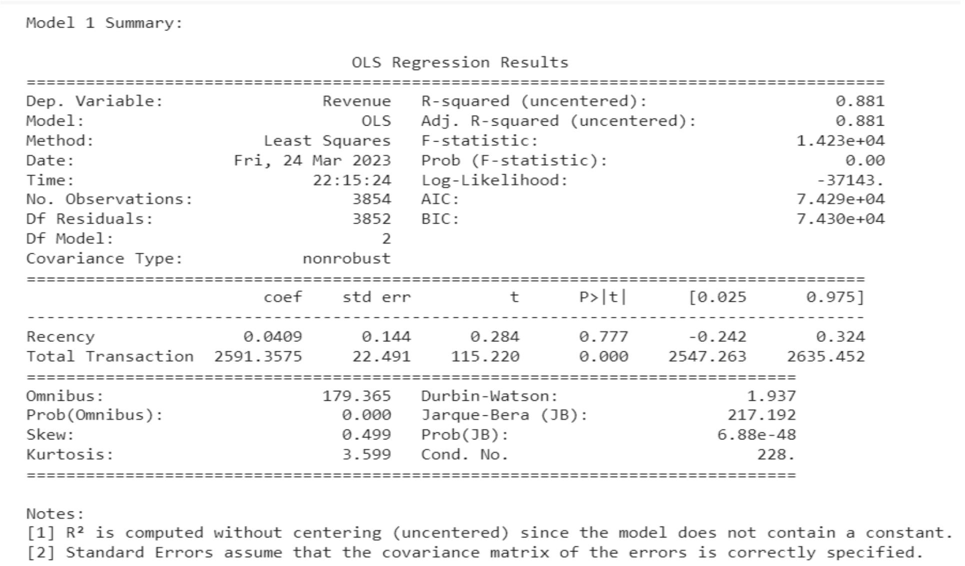


Fig 26: Predicted vs Actuals for Linear Regression

The above graph for actuals vs predicted values of revenue tells us that the model is relatively accurate in predicting the actual trend a customer is prone to follow, even if the actual values might be a little off. This is of great help as it tells us which customer is way more likely to spend.

i. **OLS Model (Linear Regression) Summary**:



```
Model 1 Summary:

                                 OLS Regression Results
=======================================================================================
Dep. Variable:                Revenue   R-squared (uncentered):                   0.881
Model:                            OLS   Adj. R-squared (uncentered):              0.881
Method:                 Least Squares   F-statistic:                          1.423e+04
Date:                Fri, 24 Mar 2023   Prob (F-statistic):                        0.00
Time:                        22:15:24   Log-Likelihood:                         -37143.
No. Observations:                3854   AIC:                                  7.429e+04
Df Residuals:                    3852   BIC:                                  7.430e+04
Df Model:                           2
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
Recency               0.0409      0.144      0.284      0.777      -0.242       0.324
Total Transaction  2591.3575     22.491    115.220      0.000    2547.263    2635.452
==============================================================================
Omnibus:                      179.365   Durbin-Watson:                   1.937
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              217.192
Skew:                           0.499   Prob(JB):                     6.88e-48
Kurtosis:                       3.599   Cond. No.                         228.
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

The above table gives a model summary for linear regression through least squares method. The adjusted R2 score of 0.881 indicates a high goodness-of-fit.

We used the linear regression model trained above to find the revenue that can be generated from every customer in the next 365 days. We then combined out results from Churn Prediction and Future CLTV with the Recency, Frequency and Revenue data, and the demographics data of the customer to create proper customer profiles and get actionable metrics from the data.

8.4. Future Customer Lifetime Value (FCLV)

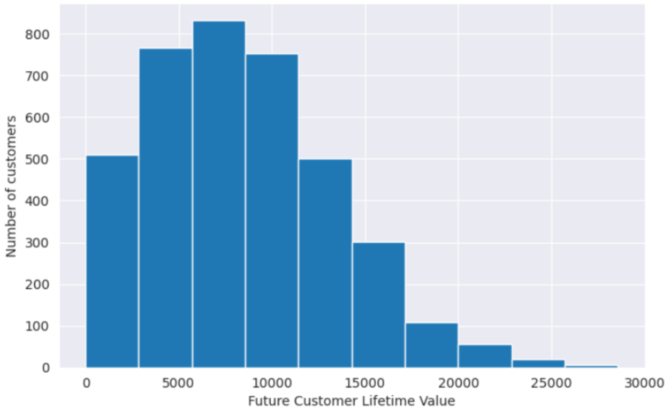
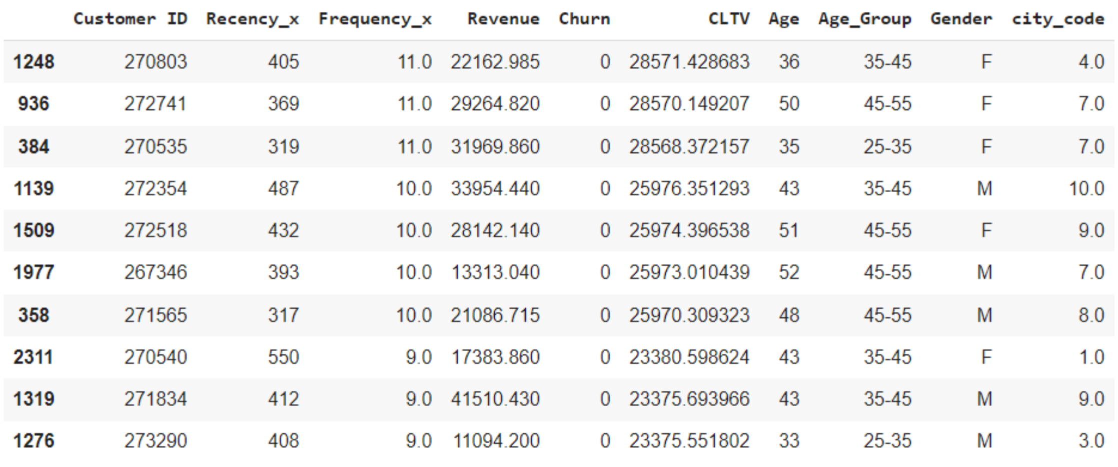


Fig 27: Histogram of Future Customer Lifetime Value vs number of customers

The above plot provides us with a wealth of valuable information. It is evident that a majority of customers fall within the 0-10,000 spending range. With this distribution, it is relatively straightforward to comprehend and forecast customer behavior. By identifying customers with a potential spend of over 15,000, we can classify them as high-value customers, while those with a potential spend of over 20,000 can be classified as extremely high-value customers. Such customer groupings can be used effectively to design targeted marketing campaigns, pricing strategies, and customer retention programs.



	Customer ID	Recency_x	Frequency_x	Revenue	Churn	CLTV	Age	Age_Group	Gender	city_code
1248	270803	405	11.0	22162.985	0	28571.428683	36	35-45	F	4.0
936	272741	369	11.0	29264.820	0	28570.149207	50	45-55	F	7.0
384	270535	319	11.0	31969.860	0	28568.372157	35	25-35	F	7.0
1139	272354	487	10.0	33954.440	0	25976.351293	43	35-45	M	10.0
1509	272518	432	10.0	28142.140	0	25974.396538	51	45-55	F	9.0
1977	267346	393	10.0	13313.040	0	25973.010439	52	45-55	M	7.0
358	271565	317	10.0	21086.715	0	25970.309323	48	45-55	M	8.0
2311	270540	550	9.0	17383.860	0	23380.598624	43	35-45	F	1.0
1319	271834	412	9.0	41510.430	0	23375.693966	43	35-45	M	9.0
1276	273290	408	9.0	11094.200	0	23375.551802	33	25-35	M	3.0

Fig 28: Table showing information on 10 customers with highest Future CLTV

Code: https://github.com/debanjansaha-git/CustomerProfiling

Furthermore, we analyzed the top 10 customers with the highest future CLTV (customer lifetime value) to identify patterns and gain a better understanding of their behavior. The table below demonstrates that the top three customers with the highest future CLTV are female, belonging to different age groups, and have a buying frequency of 11. This buying frequency serves as a benchmark that indicates loyal customers. All ten customers are active buyers and have not churned.

9. Conclusion

In this project, we applied various data mining techniques to analyze and predict customer behavior for retail business. We first used RFM analysis to segment customers into different groups based on their transaction behavior. Then we applied KMeans++ clustering algorithm to cluster the customers into different categories based on their RFM scores. We also found that KMeans++ clustering algorithm was better than DBSCAN in clustering the customers based on RFM scores. We also used logistic regression and XGBoost models to predict customer churn and linear regression model to predict customer lifetime value. To handle the problem of imbalanced data, we used the SMOTE technique to oversample the minority class. Then we used cross-validation to evaluate the performance of our models on the oversampled data. We also used undersampling of the majority class and found same results.

Overall, our models were effective in predicting customer behavior and churn with good accuracy. In conclusion, the various methods used in this project were effective in providing insights into customer behavior and predicting customer churn and lifetime value.

10. Scope for Future Work

While we tried to address most of the various analysis and modeling related to some customer data, the whole code can be modified very easily to suit any particular needs, this study was conducted in limited time during the Spring semester of 2023. We wanted to try more models like self-organizing maps for unsupervised RFM modeling along with the application of linear models using neural networks for customer lifetime value prediction, and many more. For now, we shall keep the scope of this analysis until here, but in future, we would like to try out all these options. The code has been made public at https://github.com/debanjansaha-git/CustomerProfiling

Code: https://github.com/debanjansaha-git/CustomerProfiling