Leveraging RFM Analysis for Customer Segmentation, Lifetime Value Prediction, and Churn Forecasting in the Banking and Financial Sector

Ashika S S
Department of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
ashika.ss2022@vitstudent.ac.in

Sushmithaasri K N
Department of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
sushmithaasri.kn2022@vitstudent.ac.in

***Abstract*— Customer behavior understanding and forecasting are crucial in the competitive banking and finance industry for long-term growth and customer satisfaction. The three basic goals that this project will concentrate on are RFM-based analysis, sophisticated customer segmentation, CLTV prediction, and churn forecasting. In this research, Python was used as the first choice for data extraction, analysis, visualization, and application of machine learning and deep learning models due to its rich set of libraries that accommodate end-to-end data science pipelines.**

RFM customer segmentation can be improved using the assistance of K-Means clustering, Gaussian mixture model, HDBSCAN, Deep Embedded Clustering, ClusterGAN, Spectral Net, Self Organizing maps, thus assisting to categorize the customers with dense and diverse clusters and heterogeneous data types. It assists in the delivery of personal services and pinpoint marketing.

For CLTV forecasting, sophisticated models like XGBoost, LightGBM, Extra Trees Regressor, HistGradientBoostingRegressor, Ridge and Lasso Regression, NGBoost Regressor have been employed, taking into account the transaction history and dynamic behavioral features while forecasting the long-term value of the relations for the customer, and therefore the output will be robust and interpretable.

Sophisticated supervised learning methods such as feature embedding CatBoost and Graph Neural Network (GNN) are employed to identify risky customers for churn prediction. These methods analyze enriched RFM data along with interaction and engagement metrics. The performance of models is tested based on important evaluation metrics like Accuracy, Precision, Recall, AUC-ROC and F1-score.

The conclusions of the study are in the direction of driving data-driven decision-making as well as customer retention and profitability through novel machine learning methods in the banking and financial sector.

Keywords—RFM analysis, CLTV, Churn forecasting, Customer Segmentation, Banking

I. Introduction

In today's rapidly evolving digital economy, the banking and financial sector faces mounting pressure to not only retain existing customers but also to deliver personalized and value-driven experiences. As customers demand more personalized services and better financial products, institutions must rely on data-driven strategies to understand, segment, and predict customer behavior effectively. In this context, Recency, Frequency, and Monetary (RFM) analysis emerges as a foundational approach to gain actionable insights into customer engagement patterns. [1]

RFM analysis evaluates how recently a customer has made a transaction(Recency) as customers who transacted more recently are typically more engaged and more likely to respond to future communications, how often they make transactions in a given period (Frequency) as higher frequency indicates a loyal and active customer , and how much they spend (Monetary) as customers with higher monetary value contribute more significantly to revenue. These dimensions provide a structured framework for understanding customer value and behavior. However, traditional RFM techniques, while useful, fall short in capturing the complexities of modern customer interactions in the financial sector. To bridge this gap, the integration of advanced machine learning algorithms, and deep learning architectures can significantly enhance the precision of customer segmentation, the accuracy of Customer Lifetime Value (CLTV) prediction, and the reliability of churn forecasting.[1]

RFM analysis can be utilized for customer segmentation in the banking sector. By analyzing transaction data, banks can categorize customers based on their transaction recency, frequency, and monetary value. This segmentation allows banks to target specific customer groups with tailored marketing strategies, enhancing customer engagement and loyalty.

Estimating customer lifetime value (CLV) is crucial for banks to prioritize high-value customers and allocate resources effectively. RFM analysis serves as a foundation for CLV prediction by identifying key customer segments and their potential future value. By adapting the RFM model to include weighted factors, banks can better estimate the future value of different customer segments, aiding in strategic decision-making for marketing and customer relationship management.[2]

Churn prediction is a critical application of RFM analysis in the banking sector. By integrating RFM metrics with advanced machine learning techniques banks can accurately classify churn and non-churn customers. This approach enables banks to identify the most influential factors in churn prediction, such as transaction patterns and customer demographics, and develop strategies to mitigate churn.

The study is poised to create a complete analytics pipeline, using RFM analysis for the baseline of any complex machine learning tasks involving banking and financial implementations. In the customer segmentation space, the study uses the latest curation methods (including K-Means, Gaussian Mixture Model (GMM), HDBSCAN, Deep Embedded Clustering (DEC), ClusterGAN, Spectral Net, and self-organizing maps (SOM)). To predict Customer Lifetime Value (CLTV) we employ a variety of different regression models (XGBoost, LightGBM, Extra Trees Regressor, HistGradientBoosting Regressor, Ridge, and Lasso Regression and NGBoost Regressor), each of which is purported to handle high-dimensional and non-linear financial data as well as have the appropriate understanding of computational challenges. For the RFM, churn prediction will entail using trusted and interpretable models (CatBoost) and deep neural architectures (such as Multi-Layer Perceptron (MLP)). By using these innovative methodologies in an RFM framework, we provide a scalable, practical approach to financial institutions to facilitate customer retention, institutionalize relevant marketing practices, and increase long-term profitability.

II. Literature Review

Understanding customer behavior through Recency, Frequency, and Monetary value (RFM) analysis has become a foundational strategy in customer segmentation and lifetime value prediction across industries, particularly in retail banking and digital marketing. Recent literature has expanded the classical RFM framework to incorporate modern data processing techniques and machine learning algorithms for more precise segmentation and actionable insights.

Khajvanda M and Tarokh M.J. (2012) in their paper "Estimating customer future value of different customer segments based on adapted RFM model in retail banking context" proposed a framework for customer segmentation and future value prediction using a weighted RFM model tailored for the retail banking sector. They emphasized the importance of transitioning from traditional need-based segmentation to value-based segmentation using Customer Lifetime Value (CLV) as a quantifiable metric. Their study demonstrated how customer future value, as a component of CLV, can be estimated for each segment, thereby aiding in strategic decision-making for marketing and customer relationship management (CRM) programs. The adapted RFM model they presented serves as a foundational approach to more personalized and effective customer targeting in financial services.[3]

Ahmed G et al. (2023) in their paper "Indian Banking Precision Marketing: A Comparative Analysis of Machine Learning Customer Segmentation Algorithms" investigated the role of precision marketing in enhancing customer engagement within the Indian banking industry through the application of advanced machine learning techniques. The study conducted a comparative analysis of RFM-based K-means clustering and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), using real-world transactional data from a leading private bank in India. While K-means effectively segmented customers based on transactional behavior, the study highlighted challenges in using DBSCAN—particularly the difficulty in estimating the epsilon ('e') parameter and handling higher-dimensional data with varying densities. Despite these challenges, the research demonstrated the transformative potential of sophisticated segmentation methods in improving targeted marketing efforts and offered actionable insights for financial institutions, researchers, and policymakers operating in dynamic banking environments.[4]

Brito et al.(2024) in their paper "A framework to improve churn prediction performance in retail banking" proposed a robust methodology combining advanced data preprocessing techniques and state-of-the-art classification models to enhance churn prediction in the banking industry. Emphasizing the importance of feature engineering (FE) based on RFM (Recency, Frequency, Monetary) principles, the study introduced a comprehensive framework involving adaptive synthetic oversampling (ADASYN) and NEARMISS undersampling to effectively handle class imbalance. The authors employed XGBoost and elastic net algorithms on a large-scale dataset of over 3 million customers and 170 million transactions, demonstrating improved performance in key metrics such as PR-AUC, recall, and specificity. Their work highlighted that detailed data preparation significantly improves model predictive power across different algorithmic paradigms. The study contributes both methodologically and practically to the field of decision support systems in financial services by reinforcing the value of preprocessing in churn modeling and offering actionable insights for enhancing customer retention strategies in retail banking. [5]

Liao et al. (2022) introduced a Multi-Behavior RFM (MB-RFM) model based on an improved Self-Organizing Map (SOM) neural network algorithm. Unlike traditional RFM approaches that rely solely on purchase behavior, their model incorporated various user actions—such as clicks, favorites, and cart additions—into the segmentation process. Through entropy-based weighting and enhanced clustering via SOM, the model demonstrated superior accuracy in sparse datasets, enabling application vendors to design more effective and targeted promotional strategies. [6]

Patel et al. (2016) developed a revolutionary RFM-based banking model, which combines k-means, k-means++, hierarchical clustering, and fuzzy C-means techniques to cluster customers into 8 segments based on the transaction pattern. Dunn Index was utilized to validate cluster goodness and Analytic Hierarchy Process (AHP) weighting to project Customer Lifetime Value ($CLV = 0.623M + 0.304F + 0.073R$), demonstrating the ability of hybrid clustering techniques in a proficient segmentation of 32,422 customers into High (14%), Medium (83%), and Low-value (3%) segments.The innovation in the integration of clustering models and security protocols in the research provides an automated approach to resource allocation. While the multi-model clustering framework provides robust segmentation, the AHP constant weights and lack of churn predictive modeling are areas that dynamic machine learning approaches can improve upon in future studies, particularly for real-time customer valuation for banking use. [7]

Feng et al. (2019)'s modeling of RFM for a bank made a few simple advancements by offering metrics for frequency (Fq) and monetary (Mq) metrics that were changed quarterly and understanding the results through Spark based k-means clustering, using a dataset of 144,109 transactions. The model produced interesting results, such as being able to identify high value customers ("Important Customers": R=7.16, F=1.71, M=7.39) accurately with 26.5% precision (and low value customers "Loss Customers": R=2.05, F=1.29, M=2.75). However, the reliance on quarterly averages limited the model's ability to react to changing financial behavior. Also, the proposed model did not use data in real time. While it provides a compelling advancement in the model of RFM, we wondered if it may have value to add an understanding of temporal value migration patterns, opening up the possibility of including multi-channel digital banks. Future modifications to improve the model could include evolving customer relationships, dynamic parameter weighting, and using machine learning models. [8]

Aliyev et al. (2020) introduced a new approach for bank customer segmentation using an RFM approach for bank transactional data in combination with three machine learning methods (K-means, DBSCAN, and Agglomerative clustering). Their combination was a hybrid approach where DBSCAN combined with K-means uncovered which customers were high-value customers through outlier detection while also identifying cluster boundaries dynamically. Their research also confirmed that K-means was a good technique to use for larger datasets, while DBSCAN was an advantage at successfully identifying any patterns of value in the transactional data. They were also limited since they only analyzed transactional data, without incorporating any multi-channel value behaviors or value changes over time. Future research may improve on this research by incorporating live real-time data stream for each banking customer, and will use more cutting-edge clustering techniques with more complex financial data. They showed that it is possible to build on this research of machine-learning-based RFM approaches for banking customer management. [1]

Lalitha et al. (2024) also designed a new hybrid CLTV predictive model that merges the BG/NBD and Pareto/NBD probability models and incorporates machine learning. The presented approach was proved to have 93.4% accuracy while performing a forecast on a new customer with linear regression, and it actually increased the estimated error in CLTV by 10%, with their novel Modified BG/NBD model, in contrast to traditional practices. The RFM analysis by the authors gives useful customer segmentation with practical utility to the real-world scenario of retail marketing, in real-time. There were some limitations mentioned for this study, to which I would like to add their discussions were noted transactional data, in addition to any behavioral, or demographic data. This work as a whole gives a useful contribution to this line of literature for real-time CLTV predictions, with a pragmatic approach for businesses. [9]

Dadfarnia et al. (2020) presented a reliable churn prediction methodology for payment terminals, employing RFM analysis and integrating a five-layer deep neural network (DNN). They recorded an improved F-measure of 73% in classifying terminals into six value-based categories using actual transaction data from Parsian Bank. Their three-hidden-ReLU-layer DNN architecture, featuring 512 neurons within each layer, outperformed traditional decision trees and naïve Bayes methods, demonstrating the feasibility of RFM metrics paired with deep learning algorithms to forecast churn in financial service terminals. Overall, the work presented banks with an insight into the ability to identify payment terminals that are less likely to remain active, and the extended observation window of five weeks, and Iranian only data, might lend itself to additional research opportunities and discussions surrounding the findings through a larger time frame or a variety of other markets. [10]

Fader, Hardie, and Lee (2005) recommended a model synthesizing the RFM (Recency, Frequency, Monetary) method with Customer Lifetime Value (CLV). Their approach utilizes iso-value curves to segment customers by similar future valuations in the face of different buying histories, providing a straightforward graphical modeling of RFM and CLV interactions. Their model is based on the Pareto/NBD model of buying flow and gamma-gamma submodel of expenditure per purchase. They demonstrated the technique to work on a variety of holdout tests, implementing it on real CDNOW data, in order to enhance analysis of customer bases.[11]

III. Dataset Description

A. Customer Segmentation and CLTV Prediction

One of the datasets utilized for this research is known as "Credit Card Customer Data", which was located on Kaggle. Each customer's activity with an institution has been anonymously gathered. The file consists of seven applicable measures regarding their credit card usage and customer activity. These are: Sl_No (a serial number), Customer Key (a customer identifier), Avg_Credit_Limit (the average credit limit per customer), Total_ Credit_Cards (total credit cards owned by the customer), Total_visits_bank (total visits made to the bank in person), Total_visits_online (total visits made to online banking), and Total_calls_made (total calls made to the servicing center of the bank). The data is missing an aspect of transaction of each customer, or customer churn indicators; yet, the dataset can still fulfill the planned purposes of customer segmentation and Customer Lifetime Value (CLTV) estimation with significant behavioral factors. [12]

B. Churn Prediction

The data used in this research for RFM based Churn prediction is the Default of Credit Card Clients Dataset from the UCI Machine Learning Repository, containing 30,000 instances and 24 features related to credit behavior and payment history. The features are listed below:

ID (unique identifier for every client), LIMIT_BAL (total credit limit given to the client in New Taiwan dollars), SEX (gender: 1 = male, 2 = female), EDUCATION (education level from graduate school to unknown), MARRIAGE (marital status: 1 = married, 2 = single, 3 = others), and AGE (client's age in years). The following series of variables, PAY_0 through PAY_6, is the repayment condition for the previous six months with values from -1 (pay in due course) to 8 (eight-month payment delay), measuring delinquency behavior. BILL_AMT1 through BILL_AMT6 indicate the value of bill statements (in NT dollars) in each of the previous six months, and PAY_AMT1 through PAY_AMT6 indicate the amount actually paid by the client during those months. The last feature, default payment next month, is a binary target variable (1 = default, 0 = no default), and it shows whether the client defaulted on their payment next month. This data is suited for the RFM-based churn prediction model in this research. The Recency can be calculated from recent payment history (e.g., last delay status), Frequency from counts of on-time payments or payment events, and Monetary from bill and repayment amounts over time. The temporal pattern and heterogeneity of behavioral variables allow for building rich features to support predictive modeling. Furthermore, the existence of a labeled target variable makes it easier to use supervised learning methods to classify customers as potential churners and loyal users. [13]

IV. Methodology

Fig 1. Methodology of our study

All data preprocessing, feature engineering, and model development activities were performed in the Python 3.10 programming language on Google Colab, a cloud-based Jupyter notebook environment. The Google Colab environment is scalable, collaborative, and provides a workflow for model training, visualization, and experimentation. Google Colab has support for GPU/TPU accelerators for deep learning models, which can be easily activated. Moreover, Google Colab fits into the commonly used python libraries proposed for all layers of our model.[17] [18] [19]

A. Data Preprocessing

1) Customer Segmentation and CLTV Prediction

Preprocessing is an essential step to prepare raw data for analysis and modeling. For Customer Segmentation and CLTV Prediction,"Credit Card Customer Data" was used as mentioned before. For this, we will apply preprocessing, using three major Python libraries namely Pandas, NumPy, and MinMaxScaler from scikit-learn. All the libraries served a particular purpose in making the dataset clean, consistent, and well scaled for clustering.

a) Data Cleaning

The data was read into the dataset with the read_csv() function from Pandas. Initial data cleaning entailed the detection of missing values with isnull().sum() and duplicate records through the duplicated() method. This was necessary to avoid compromising the analysis due to incomplete or duplicate entries. All columns' data types were validated to ensure they were appropriate for calculations, with numeric variables specifically. The column 'Total_visits_online', which also contained non-numeric and floating-point values, was given special attention. The column values were rounded to the nearest integer to ensure consistency, and the type was changed to integer for further calculations.

b) Handling Outliers

To remove extreme outliers from key numerical variables: 'Avg_Credit_Limit' and 'Total_visits_online', the Interquartile Range (IQR) method was used to first get the first (Q1) and third (Q3) quartile values. Then, any values that were greater than or less than the acceptable interval of [Q1−1.5×IQR, Q3 1.5×IQR] were classified as outliers. Instead of deleting the extreme values, the IQR approach was implemented to cap the extreme outliers using Pandas' clip() function, to allow for outlier values to still be within logical values and ultimately keep the data frame size.

c) Feature Engineering

In order to enable effective customer segmentation, RFM (Recency, Frequency, Monetary) features were built:

Recency was calculated as the reciprocal of the number of times a call was made by the customer, which indicated how recently the customer made contact with the bank.

Frequency was calculated as the frequency of visits, (online and at the bank blended together) indicating how engaged the customer is.

Monetary, defined as the average credit limit, is a proxy for the financial value of the customer.

These designed attributes provided scope for a multi-dimensional perspective on customer behavior, appropriate for clustering that would follow. [14]

d) Normalization

The RFM attributes were on varying scales, which could skew the clustering algorithms that depend on distance-based values. To balance this, the MinMaxScaler of scikit-learn was used to scale every RFM attribute to a consistent range of 1 to 5. The normalization process ensured that one feature did not overwhelm the clustering operation, enabling each feature to equally contribute to building groups of customers.[15]

2) Churn Prediction

a) Data Cleaning

The data used in this research was already clean, with no missing values or outliers to correct. Nevertheless, to further maintain consistency and compatibility with modeling pipelines, categorical variables like SEX, EDUCATION, MARRIAGE, and the payment status indicators (PAY_0 to PAY_6) were checked to be in integer format. Even though these features were already integers in the original dataset, an explicit type check and cast were made to ensure their availability for subsequent processing.

b) Feature engineering

To reformat the original credit card data into RFM-based churn prediction format, we derived three important features—Recency, Frequency, and Monetary—out of six months of payment history information (PAY_AMT1 through PAY_AMT6), which illustrate the client's previous payment activities.

- Recency (R): This attribute captures the time since the customer last made a payment. We reversed the columns of monthly payments and selected the most current month in which a non-zero payment was paid. A lower numeric value (e.g., 1) reflects a more recent payment, and a higher value (e.g., 6) reflects that the last payment was several months ago. This conversion was done through a mapping to the reversed index of non-zero payments.
- Frequency (F): Frequency indicates the number of times the customer paid during the six-month duration. It is determined as the number of non-zero entries in the PAY_AMT1 through PAY_AMT6 columns, which is the number of successful monthly payments.
- Monetary (M): This variable measures the amount the customer has paid in the past six months. It is calculated by adding the values in the PAY_AMT1 through PAY_AMT6 columns.

These characteristics present an overview of the customer's current financial behavior and are the foundation for subsequent modeling within the context of churn prediction. [16]

B. Exploratory Data Analysis

To assess the structure, distribution, and relationships among the engineered Recency, Frequency, and Monetary (RFM) features in the customer data clustered by segmentation, Exploratory Data Analysis (EDA) was performed. Popular Python libraries such as Seaborn, Matplotlib, and Pandas were utilized for creating visualizations that would effectively inform users about understanding customer segmentation, CLTV and churn prediction patterns. The exploratory data analysis had the objective of visualizing patterns in the data across the clusters, and it is essential to extract insights which justifies the quality and interpretability of the clustering results.

1) Customer Segmentation

a) Pair Plot of RFM Features

In order to analyze the relationships between Recency, Frequency, and Monetary traits, Seaborn's pairplot() was implemented, resulting in the production of kernel density estimation (KDE) plots on the diagonal and scatterplots off the diagonal colored by cluster labels, which showed clear evidence of patterns. For example, customers in high frequency (cluster 0) had correspondingly low recency, indicating that they had purchased recently and frequently as well. Visual separation of clusters also provided additional evidence that this clustering approach was appropriate.

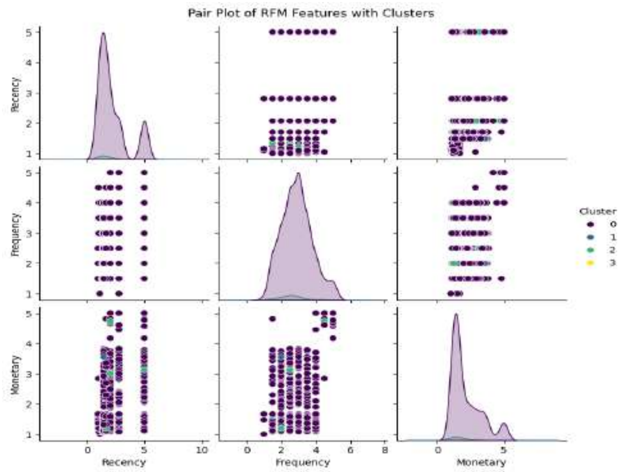


Fig 2. Pair plot of RFM Features

b) Box Plots of RFM Features by Cluster

Boxplots were applied to examine the distribution of RFM values across each of the clusters and show the medians, ranges, and any outliers. Cluster 3 displayed the most recent purchases (about 3 weeks), as well as having the highest frequency and monetary value, identifying high-value, engaged customers; while Cluster 2 indicated the greatest elapsed time since the last activity (approx. 5 weeks). These understandings strengthened the use of segmentation for identifying behavioral differences across customer segments.

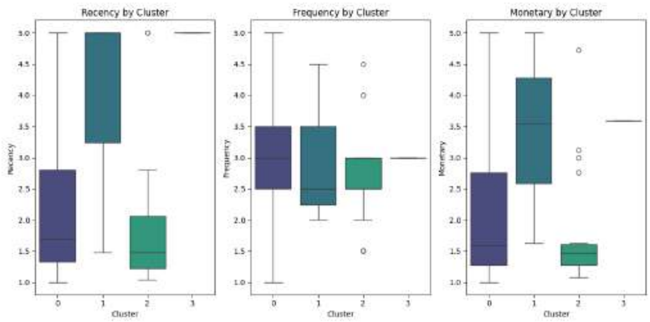


Fig 3. Box Plots of RFM Features by Cluster

c) Heatmap of Cluster Centers

A heatmap illustrated the average Recency, Frequency, and Monetary values per cluster, and it provided an overview of the general activity by customer. The color distinction allowed for a quick and easy comparison of clusters at a glance. It is clearly visible that cluster 3 had the highest values for all the features.The individuals in group 1 were most engaged and gave the most value of all groups. Although we did show some evidence of higher, relative engagement and spend patterns for group 1, no other spending or engagement besides job and live sports resulted with group 1 being above the average of 1. Overall, the heatmap gave a clear and intuitive overview of the customer profile segments by group along with strategic decisions.

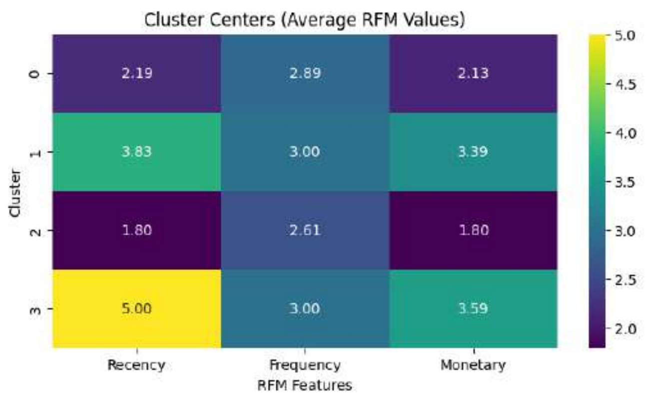


Fig 4. Heatmap of Cluster Centers

d) Count Plot of RFM Segments

In order to evaluate the total distribution of customers according to their cumulative RFM score, the count plot was utilized to plot the number of customers in four different segments, namely Low, Medium, High, and Very High.A count plot is a bar chart presenting the number of observations in every category of a categorical variable. The segmentation was derived through quartile-based binning on the merged RFM Score. The count plot revealed a fairly balanced spread of customers among all four segments, indicating a diversified customer base. This extent of segmentation helps in resource allocation by allowing businesses to target high-value customers first while crafting engagement strategies for low-value segments.

Fig 4. Count Plot of RFM Segments

2) CLTV Prediction

a) Distributional Analysis

Distributions and skewness of numeric features were evaluated using histograms with KDE plots. Avg_Credit_Limit and Monetary showed a substantial right skew, with a limited number of customers owning high values. Total_Credit_Cards showed a nearly symmetric distribution with the most common counts being around the 4-6 card range. Numeric behavioral features such as Total_visits_bank, Total_visits_online, and Total_calls_made were primarily negatively skewed, supporting a conclusion of low levels of engagement/interaction on behalf of the customers. To assess recency, again customers' transactions showed a negative skew. Frequency and RFM_Score, however, were much more normally distributed suggesting consistent and steady engagement/interaction patterns to support its continued use for customer segmentation purposes.

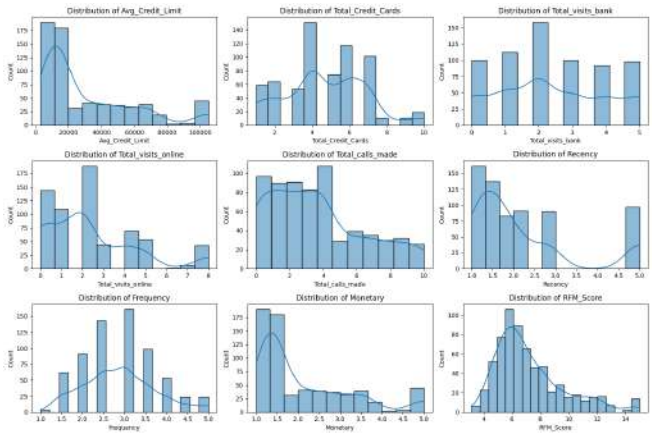


Fig 5. Histograms for numerical features

b) Outlier Detection using Boxplots

Boxplots were employed to detect outliers and understand the spread of data on key attributes. Avg_Credit_Limit had high percentages of outliers towards the top that could be interpreted as high-spending customers or corporate accounts. RFM_Score also had strong high-end outliers, denoting a pool of highly transacting and revenue-generating customers.Other features, such as Total_calls_made, Total_visits_online, and Monetary, also had outliers that could impact the sensitivity of the model. Recency and Frequency, however, had a relatively tight spread with low outliers. This analysis played a critical role in determining if preprocessing techniques such as capping, scaling, or employing robust models less sensitive to extreme values are needed.

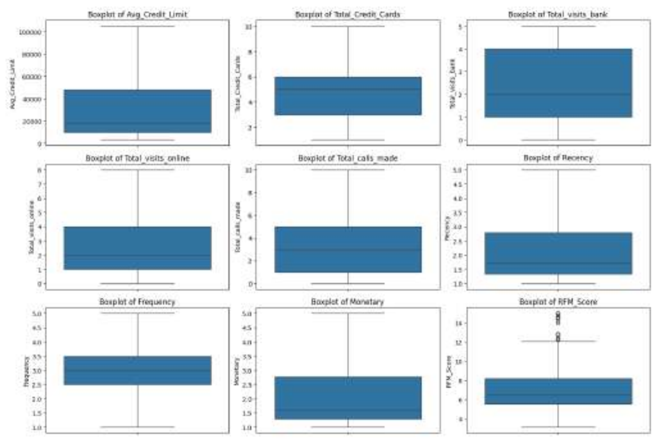


Fig 6. Box Plots for numerical features

c) Correlation Analysis

The correlation heatmap demonstrated several strong relationships among your variables. There was a strong correlation between Avg_Credit_Limit and Monetary (r = 0.80). Furthermore, RFM_Score correlated strongly with Monetary, Recency, and Frequency, reinforcing its usefulness as a summary measure. The Frequency variable correlated to Total_visits_online, which indicated more online engagement. However, Total_calls_made correlated negatively with Recency and RFM_Score, which suggested dissatisfaction/value from those that called frequently. This was useful for feature selection and multicollinearity.

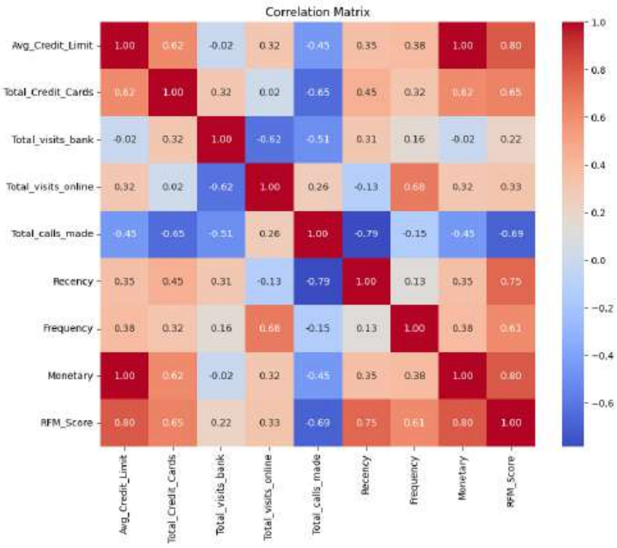


Fig 7. Correlation Matrix

3) Churn Prediction

a) Churn Distribution

A pie chart was utilized to graphically represent the distribution of the target variable, default payment next month. It revealed that most customers did not default, with around 77.9% non-defaulters and 22.1% defaulters, revealing a class imbalance that needs to be addressed in modeling.

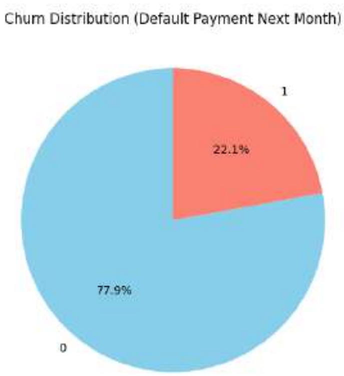


Fig 8. Churn Distribution

b) Numerical Features by Default Status

Box plots of important numerical features like LIMIT_BAL (credit limit) and AGE were graphed against the target label. Non-defaulters tend to have higher median and maximum credit limits than defaulters, implying that creditworthy customers are most likely to be assigned greater limits and also exhibit stronger payment discipline. Age, however, failed to demonstrate any distinguished difference between the two groups, implying that customer age per se may not necessarily be a decisive predictor of default behavior. These results stress the significance of financial variables such as credit limit in credit risk modeling, although demographic variables such as age would need additional multivariate analysis to draw clearer conclusions.

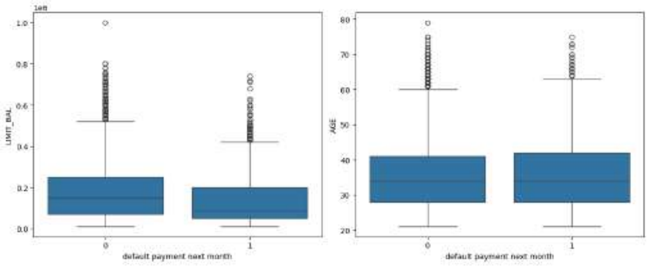


Fig 9.Numerical Features by Default Status

c) Categorical Feature Analysis

Categorical variables (SEX, EDUCATION, and MARRIAGE) were explored via count plots stratified by default status. The count plots depict the distribution of default payment status across different categorical variables. Females are more in number than males, but default rates seem slightly higher for males. Education level depicts a pattern where customers with lower education (e.g., high school) have a comparatively higher default rate. Single customers also depict a slightly higher default incidence than married customers. These demographic variables can be used in credit risk profiling when combined with financial indicators.

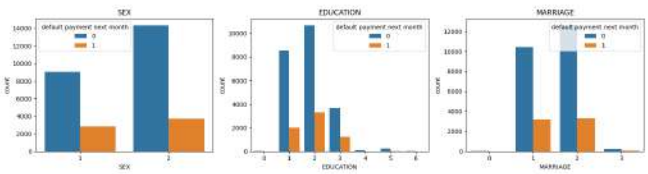


Fig 10.Categorical Feature Analysis

d) Delinquency Trend Analysis

To capture customer payment behavior in the long run, we calculated the average delinquency status over six months (PAY_0 through PAY_6). These variables capture each customer's monthly payment status, with values taking on -2 (no consumption) through positive integers representing months of payment delay.

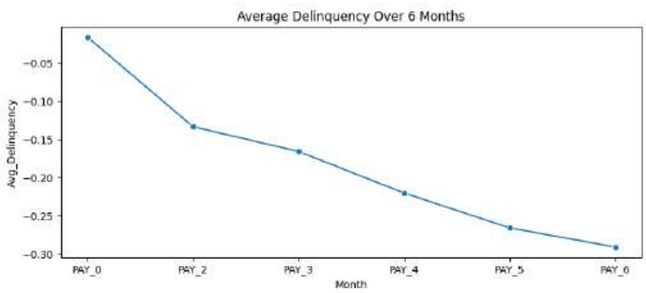


Fig 11.Average Delinquency Over 6 Months

This pattern suggests that customers begin with relatively better payment behavior (PAY_0), and there is a gradual weakening of repayment discipline as months progress, possibly due to accumulating debt or financial strain.

e) RFM analysis

To represent customer payment behavior, we extracted Recency, Frequency, and Monetary (RFM) features from the previous six months of bill payments (PAY_AMT1–PAY_AMT6) and plotted their pairwise associations by color for the next month's default payment flag. Based on the scatter plot following observations were made : a)Customers with higher frequency and more recent payments (lower recency) are less likely to default, b)Non-defaulters have higher total monetary payments implying stronger financial stability can be represented by bigger payments, c) a positive correlation between frequency and monetary value, where customers who pay more frequently also tend to pay larger amounts. Defaulters are concentrated in the lower-left quadrant, where both frequency and monetary contributions are low.

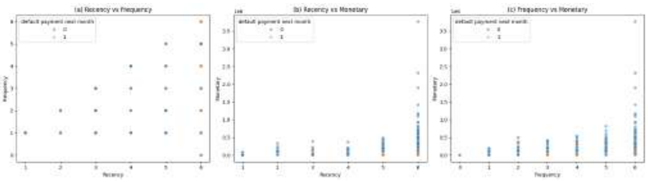


Fig 12.RFM Analysis

C. Data Modeling and Evaluation

1) Customer Segmentation

Customer segmentation helps organizations categorize customers based on behavior, enabling more targeted marketing and improved customer management. This project uses the RFM model—Recency, Frequency, and Monetary value—to analyze purchasing patterns and apply clustering algorithms such as K-Means, GMM, HDBSCAN, Deep Embedded Clustering, ClusterGAN, SpectralNet, and SOM to identify distinct customer groups.

To assess clustering quality, the following metrics are used:

- **Silhouette Score:** Measures how well each data point fits within its cluster (higher is better).
- **Calinski-Harabasz Index:** Evaluates cluster separation and compactness (higher is better).
- **Davies-Bouldin Index:** Assesses cluster similarity (lower is better).

These metrics help determine the most effective clustering model for RFM-based customer segmentation.

a) K-Means Clustering

The K-Means algorithm is a partition method that divides data into k non-overlapping and non-hierarchical clusters. It partitions observations or all data points based on their similarity and minimizes the distance between all observations in a cluster to the cluster center. In this case, K-Means was applied to RFM (Recency, Frequency, Monetary) data with k = 4 cluster. K-Means algorithm was used to partition customers into 4 clusters based on the scores of RFM. A cluster label was assigned to each customer that indicated a segment of customers.

EVALUATION

We evaluated clustering performance using the steps that follow:

- Silhouette Score: 0.7306
- Calinski-Harabasz Index: 5395.36
- Davies-Bouldin Index: 0.384

A score of these three indicators constitutes values for well-separated, distinct, and non-overlapping clusters.

VISUALIZATION

The scatter plot illustrates customer clusters based on their Frequency and Monetary features. Each point represents a respective customer, where the different colors signify the cluster each customer belongs to (0-3).

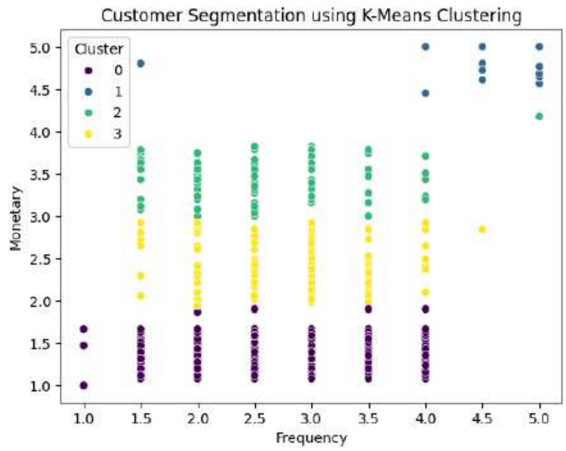


Fig 13.Customer segmentation using K-means Clustering

The physical presence of clusters and physical separation on the scatter plot confirms visually that the K-Means model was able to separate heterogeneous groups of customers from each other.

b) Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) is a probabilistic clustering algorithm using the assumption that points are sampled from a mixture of Gaussian distributions. GMM assigns each point to a probability distribution to multiple clusters instead of K-means, which assigns every point to a single cluster. This provides GMM with the ability to capture more sophisticated stages in the data. GMM can be used when clusters can overlap or are not spherical.

The GMM model was run on the scaled RFM dataset with the cluster number being 4, as in K-Means, to ensure comparison.

Evaluation

- Silhouette Score: 0.3176
- Calinski-Harabasz Index: 366.55
- Davies-Bouldin Index: 1.1097

These results indicate that although GMM is able to identify some overlapping cluster behavior, the clustering separation is poorer than that of K-Means.

Visualization

Visualizing the scatter plot shows the customer segmentation using Gaussian Mixture Model across Frequency and Monetary metrics. The color representing each plotted point refers to different customer clusters (0-3), identified by GMM.

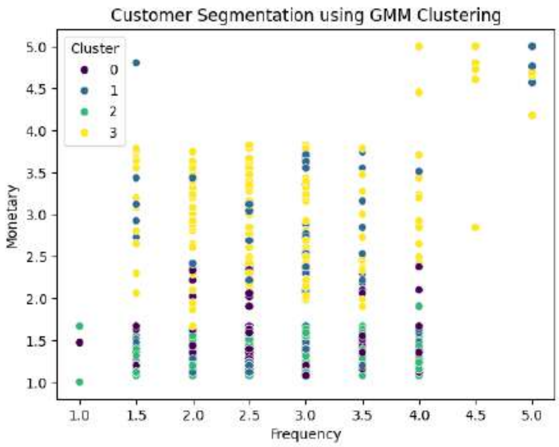


Fig 14. Customer Segmentation using Gaussian Mixture Model

The plot indicates overlapping clusters, reflecting GMM's ability to allocate probabilities to points instead of rigid boundaries.

c) HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that detects clusters of different shapes, sizes, and densities. It does not need the number of clusters to be predefined like K-Means or GMM, nor can it detect noise points (outliers). It is especially useful for datasets where clusters are not well-separated or not globular.

HDBSCAN was run on RFM data with a min_cluster_size of 5, the minimum set of points to cluster. The model itself identifies the best number of clusters by density distribution.

Evaluation

- Silhouette Score: 0.1926
- Calinski-Harabasz Index: 105.82
- Davies-Bouldin Index: 1.4529

These indicate that although HDBSCAN does pick up fine structure in the data, cluster separation is comparatively weak because it is density-based.

Visualization

The scatter plot shows customer segments generated through the HDBSCAN analytical methodology, along the dimensions of Frequency and Monetary. Each color denotes a different cluster produced by HDBSCAN.

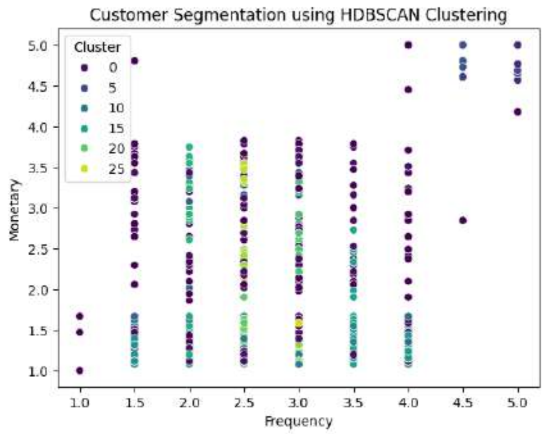


Fig 15.Customer Segmentation using HDBSCAN Clustering

The plot demonstrates HDBSCAN's ability to find fine-tuned clusters and work with non-spherical clusters, the clusters themselves may be irregularly distributed, showing added usefulness and insights based on segments of customer behavior.

d) Deep Embedded Clustering (DEC)

Deep Embedded Clustering integrates representation learning and clustering within a single framework. This technique uses an autoencoder to encode high dimensional data into low dimensional latent space and then applies K-Means in the low dimensional space for clustering. Deep Embedded Clustering (DEC) integrates feature learning and clustering into one framework. An autoencoder was trained on the RFM data to encode it into a 2-dimensional latent space. K-Means clustering was then performed on the encoded representations with 4 clusters after training.

Evaluation

- Silhouette Score: 0.8952
- Calinski-Harabasz Index: 2349.83
- Davies-Bouldin Index: 0.3513

These measures show high cluster cohesion and separation. The large Silhouette and Calinski-Harabasz values, coupled with a small Davies-Bouldin value, imply that DEC created well-separated and tight clusters in the low-dimensional feature space.

Visualization

This scatter plot represents customer segmentation with Deep Embedded Clustering, tools applied to cluster customers based on 2 encoded features from the autoencoder model.

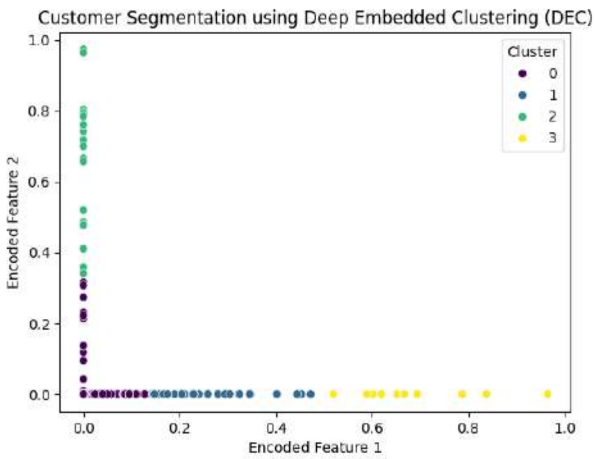


Fig 16.Customer Segmentation with Deep Embedded Clustering

The clear separation of clusters given the encoded feature space is indicative of the DEC's ability to learn representations that are both dense in the sense of those very compact plotted customer in plot to help discriminatively represented customer segmentation.

e) ClusterGAN (Cluster Generative Adversarial Network)

ClusterGAN is a Generative Adversarial Network (GAN) specifically tailored for clustering applications. In contrast to general GANs, ClusterGAN injects discrete latent variables into the generator to encourage the generation of insightful clusters in the latent space. ClusterGAN learns a low-dimensional latent representation of high-dimensional data while maintaining cluster structure.

Architecture

- Generator: The generator is learned to transform latent vectors (drawn from a multivariate Gaussian) into synthetic customer data with the same marginal distribution as the actual RFM data.
- Discriminator: The discriminator is learned to determine if customer data is real or synthetic.
- Clustering: After training is finished K-Means will be used to cluster the predicted cluster labels in the generator's latent space.

Evaluation

- Silhouette Score: 0.8791
- Calinski-Harabasz Index: 3369.38
- Davies-Bouldin Index: 0.3116

These outcomes suggest highly effective clustering with very strong compactness and inter-cluster separation. ClusterGAN demonstrates excellent promise in identifying hidden customer segments via deep generative modeling.

Visualization

The scatter plot represents customer segments based upon ClusterGAN built from the latent features of the generator network. Each color corresponds to a cluster (0-3) based upon K-Means applied to the latent space from the GAN.

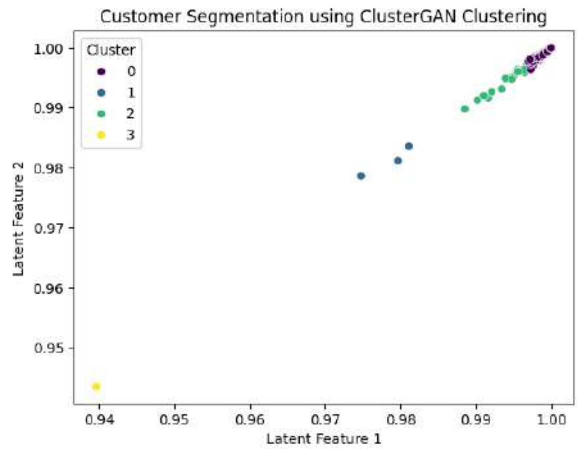


Fig 17.Customer Segmentation using ClusterGAN

In general this scatter plot demonstrates that the ClusterGAN is able to produce clusters which are distinct and tight together. This implies the ClusterGAN classifier can learn latent representations that are useful for customer segments.

f) Spectral Net Clustering

Spectral Net is a neural network-based method that combines spectral clustering and deep learning to cluster complex, high-dimensional data. Conventional spectral clustering suffers from scalability and generalization issue. Spectral Net uses a neural network to learn a nonlinear mapping of the data to a low-dimensional embedding space in which clusters are separable. This approach supports end-to-end training via reconstruction loss (autoencoder-fashion) with local structure and similarity relationship preservation of the original input.

Architecture

- Encoder: The model processes customer RFM information and transforms it into a 2-dimensional latent form via two dense hidden layers. The encoding stores important features distinguishing customer behaviors.
- Decoder: There is a reconstruction layer to guarantee the encoding keeps important features by trying to restore the original input.

- Clustering: The encoded feature vectors are then clustered using K-Means to label customer segments after training.

EVALUATION

- Silhouette Score: 0.5997
- Calinski-Harabasz Index: 2635.934
- Davies-Bouldin Index: 0.52947

These metrics of evaluation indicate good-quality clusters.A Silhouette Score near 0.6 depicts a good separation between the clusters.A high Calinski-Harabasz Index shows dense and separated clusters.A low Davies-Bouldin Index further depicts compact clusters with large separation, proving that Spectral Net learns useful embeddings for customer segmentation.

VISUALIZATION

The scatter plot depicts customer segmentation from the 2-dimensional encoded space generated by Spectral Net. K-Means was run on this space to ascertain cluster membership, and each color depicts a distinct cluster (0–3).

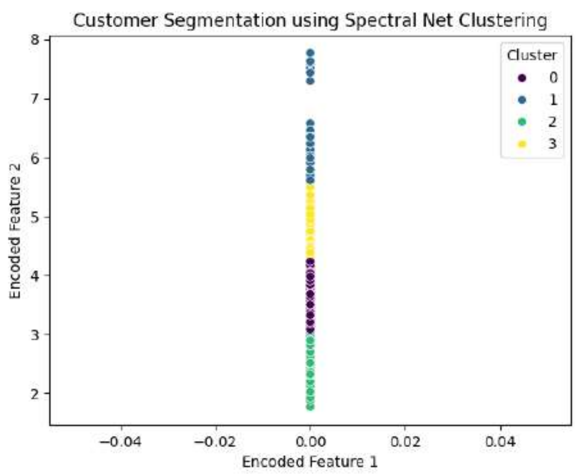


Fig 18.Customer Segmentation using Spectral Net Clustering

The visualization is such that the clusters are well-separated, and the vertical spread implies that Encoded Feature 2 has high discriminatory power. In general, Spectral Net demonstrates strong ability in modeling customer behavior for segmentation tasks.

g)Self-Organizing Maps (SOM) Clustering

Self-Organizing Maps (SOMs) represent a type of unsupervised learning that falls under the umbrella of artificial neural networks, and are generally used for both dimensional reduction and clustering. SOMs essentially take high-dimensional data and map it to a 2D grid while maintaining the topological structure of the input space, which entails that similar data points will be mapped to close locations on the 2D output map. In customer segmentation, SOMs can show behavioral patterns by mapping characteristics of customers (e.g., Recency, Frequency, Monetary) onto the 2D mapping. Each node (or neuron) in the SOM grid operates as a prototype vector and represents similar groups of customers.

Architecture

- Input: Customer RFM features. SOM grid: 10x10 grid of neurons (100 units), initialized to random weights.
- Training: Each customer RFM feature data is input to the SOM, and the neuron most similar to the input (Best Matching Unit- BMU) is updated and becomes more like the input, as well as its neighbors.
- Clustering: After training, each customer is ultimately mapped to a neuron (the grid coordinates taken the mapped neuron), which is then mapped to an integer that is assigned to the clusters.

EVALUATION

- Silhouette Score: 0.0312
- Calinski-Harabasz Index: 59.1557
- Davies-Bouldin Index: 1.3916

These statistics indicate poor clustering quality.The low Silhouette Score (~0.03) reflects overlapping clusters with weak separation.The relatively low Calinski-Harabasz Index implies that the clusters may not be well-separated or very compact.The higher Davies-Bouldin Index (~1.39) suggests that the level of similarity among clusters is great and, therefore, separation is weak.

Yet this does not make SOM worthless—it can still reveal interesting topological groupings or outliers and can be an exploratory tool prior to the application of more sophisticated models.

VISUALIZATION

The scatter plot shows customer clusters superimposed on two RFM dimensions: Frequency and Monetary. Each dot is a customer, and each color represents a distinct cluster label by the SOM grid neuron.

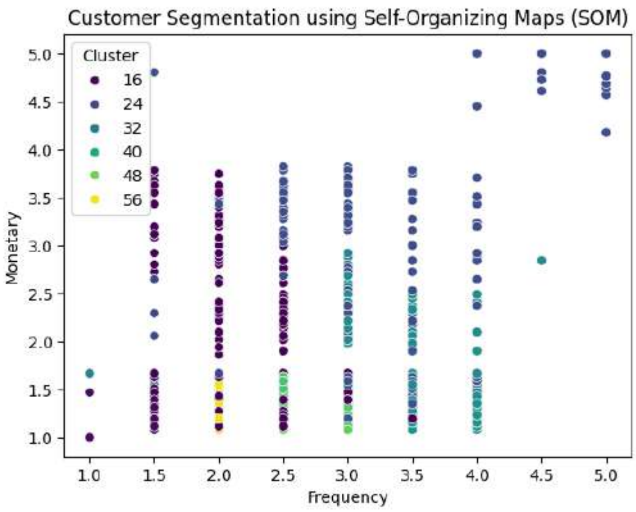


Fig 19.Customer Segmentation using Self Organizing maps

Though SOM preserves the topological relations of customer data, its resulting clusters here seem blurred and not well-defined. With so many clusters (based on a 10x10 grid), over-segmentation could result and make interpretation very difficult. However, SOM can still be useful for visualization and discovering smooth transitions in customer behavior that are not discernible with hard clustering techniques.

2) CLTV Prediction

Customer Lifetime Value (CLTV) prediction is initiated by the choice of crucial demographic and behavior features like average credit limit, number of credit cards, visit frequency (bank and online), call frequency, and RFM metrics like a composite RFM score. The synthetic CLTV target variable is derived by taking the product of average credit limit and total visits. The dataset is divided into an 80-20 split for training and testing.

a)XGBoost

To calculate CLTV, we employed XGBoost, a gradient-boosted decision tree algorithm with in-built L1 and L2 regularization to avoid overfitting. We started with a basic model with conservative parameters (learning rate = 0.05, max depth = 3, n_estimators = 300) and optimized it through manual optimization. The optimum model, xgb_best, employed a learning rate of 0.1, max depth of 5, and 500 estimators. With reg:squarederror objective, it drastically improved test set predictive accuracy.

Model Evaluation

Performance of the tuned XGBoost model was measured using standard regression metrics. The findings included:

- Mean Absolute Error (MAE): 3733.89: The predictions on average differ from the actual CLTV by about 3.7K units.
- Mean Squared Error (MSE): 67,885,800.0:Scores the average squared error; responsive to large errors.
- Root Mean Squared Error (RMSE): 8239.28:Shows that predictions are largely within an 8.2K unit error band.
- R-squared (R^2): 0.9987:Accounts for 99.87% of the variance in the target variable (CLTV), reflecting a very good model fit.

These values decisively suggest that the XGBoost model is extremely good at the task of predicting CLTV, both in terms of high accuracy and consistency of predictions.

SHAP Value Interpretation

To interpret the model and discover key drivers of CLTV, SHAP analysis was utilized. A SHAP waterfall plot for one customer revealed that Total_Visits_Online had the largest positive contribution, followed by Avg_Credit_Limit and RFM_Score. Frequency and Total_Visits_Bank had small negative contributions, while Recency and Monetary exhibited no contribution in this instance. This confirms XGBoost's feature to concentrate on the most fitting features, increasing model transparency and business usability.

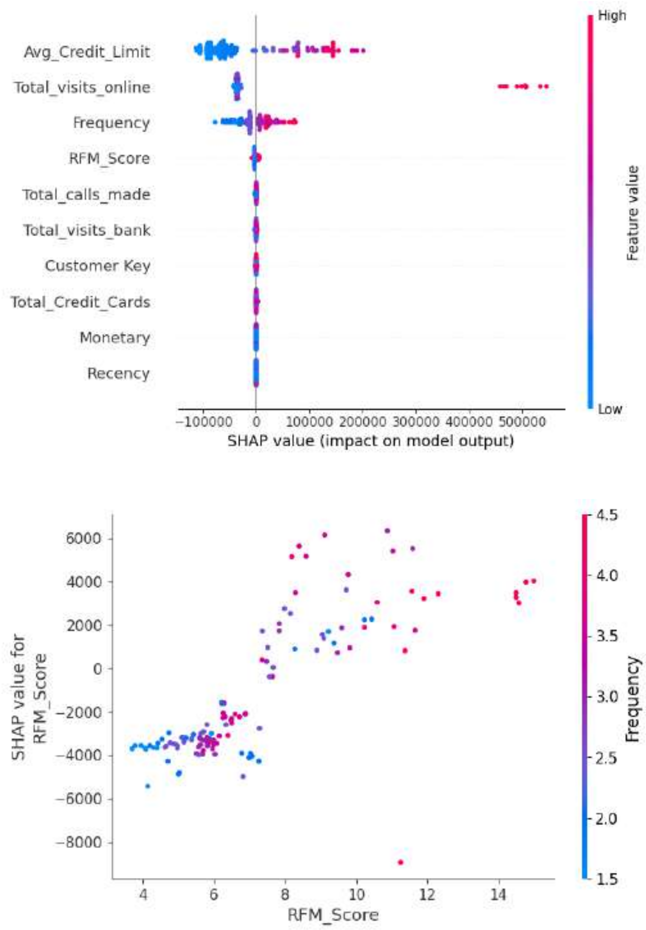


Fig 20.SHAP Plots for XGBoost

b) LightGBM

LightGBM is an efficient, fast gradient boosting framework suitable for structured data applications such as CLTV prediction. It differs from the standard approach in that it employs a leaf-wise tree construction strategy, histogram-based learning, and natively supports categorical feature handling. We began with a baseline model (learning rate = 0.05, 100 estimators), and subsequently fine-tuned parameters to enhance performance. The last model utilized a learning rate of 0.1, 500 estimators, max depth of 6, and 31 leaves with regularization (reg_alpha and reg_lambda = 0.1) to avoid overfitting. These changes improved the model's accuracy and generalizability.

MODEL EVALUATION

The LightGBM model's performance was measured with standard regression evaluation metrics. The results are:

- Mean Absolute Error (MAE): 3774.82.On average, the model's predictions are away from the true CLTV by around 3.7K units.
- Mean Squared Error (MSE): 70,111,212.86. Measured the mean of the squared differences of predicted and actual values, but places higher weight upon larger errors.
- Root Mean Squared Error (RMSE): 8369.67. Most predictions are within an 8.3K unit error boundary of the actual CLTV values.
- R-squared (R^2): 0.9986.The model accounts for 99.86% of the variance in CLTV, reflective of its great fit and prediction accuracy.

These values validate that the LightGBM model has a very good ability to predict CLTV, providing precision as well as reliability for business decision-making.

SHAP VALUE INTERPRETATION

To gain a clearer insight into feature impact in the LightGBM model, SHAP (SHapley Additive exPlanations) analysis was used. A SHAP waterfall plot for a representative customer indicated that Total_Visits_Online contributed most positively to the estimated CLTV (465,101.16), followed by Avg_Credit_Limit and RFM_Score. Frequency and Total_Visits_Bank had negligible negative impacts. Notably, aspects such as Recency and Monetary did not play any significant role here, illustrating LightGBM's capacity for selectively assigning importance to influential features. Such interpretability enables more data-driven, informed business decisions.

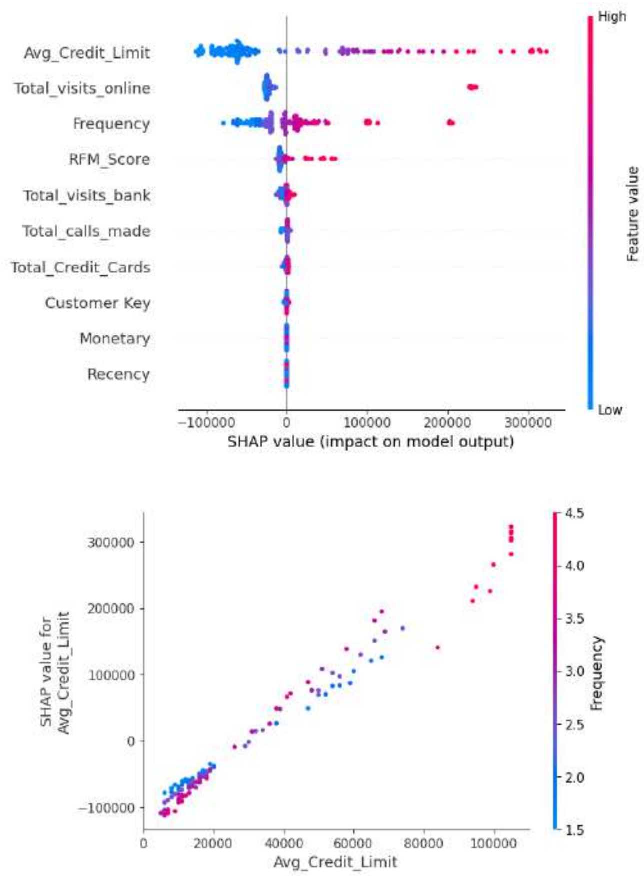


Fig 21. SHAP Plots for LightGBM

c) Extra Trees Regressor (ETR)

The Extra Trees Regressor (ETR) is an ensemble learner that constructs trees with random feature splits, providing quicker training and improved generalization compared to regular decision trees or Random Forests. Its randomness reduces variance and resists overfitting, making it appropriate for CLTV prediction. A baseline model was first created, then tuned. The last model utilized 500 trees, max_depth of 10, min_samples_split of 5, min_samples_leaf of 2, and max_features as 'sqrt'. These parameters enhanced accuracy and enabled the model to learn intricate patterns while being good at unseen data.

MODEL EVALUATION

The Extra Trees Regressor was evaluated using traditional regression evaluation weightings on a previously hold-out test set and the results were the following:

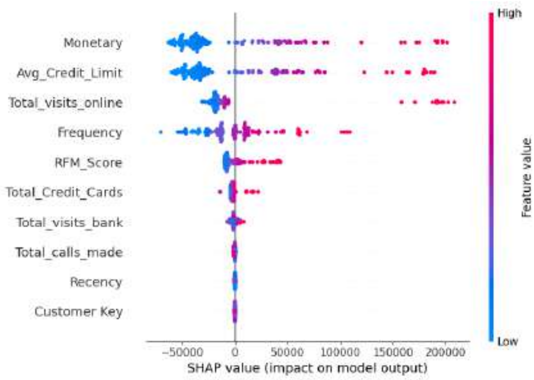
- Mean Absolute Error (MAE): 3,363.13 The average prediction obtained from the model was off from the actual CLTV value by about 3.3K, which is rated as very good accuracy.
- Mean Squared Error (MSE): 55,802,246.13 The MSE was rated as moderate as the model punished increasing errors, but still performed well overall.
- Root Mean Squared Error (RMSE): 7,464.70 The model's predictions were off by about 7.4 K for the

vast majority of predictions which is not a large number considering the range of CLTV values.
- R-squared (R^2): 0.9989.The model accounts for 99.89% of the variation in the CLTV values, reflecting outstanding predictive capability.

These measures validate that ETR is a robust predictor for CLTV with tight prediction intervals and reliable results.

SHAP Value Interpretation

In a representative case, Total_Visits_Online once more was the most significant feature, adding more than 459,000 units to the end CLTV estimate. Other substantial contributors were Avg_Credit_Limit and RFM_Score, adding large positive weight to the estimate. It is noteworthy that features such as Total_Visits_Bank and Frequency contributed marginally in a negative direction to the estimated value. Other features, such as Recency and Monetary, contributed nothing to that instance. Nonetheless, the overall SHAP interpretation confirmed that the ETR model was capable of zoning in on informative features while ignoring the noise, thus providing more model transparency to business leaders and decision makers.



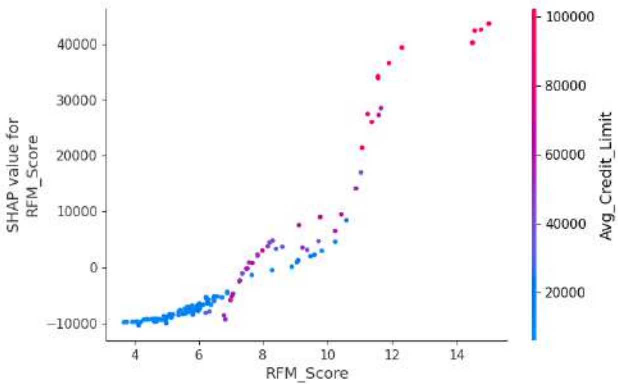


Fig 22.SHAP plots for Extra Trees Regressor

c) HistGradientBoosting Regressor (HGBR)

The HistGradientBoosting Regressor (HGBR) is a lightweight gradient boosting method that works efficiently with big tabular data and supports quick training, efficient memory, and built-in handling of missing values. For the task of predicting CLTV, it picks up sophisticated patterns and relationships well. The model was created using a learning rate of 0.1 and 300 estimators to ensure balanced performance. To prevent overfitting, max_leaf_nodes was 31 and min_samples_leaf was 20, with L2 regularization (0.1) and early stopping to improve generalization and model stability.

Model Evaluation

The HistGradientBoosting Regressor was evaluated against an independent holdout test set. The metrics below were calculated:

- Mean Absolute Error (MAE): 3,144.99.The average prediction missed by about 3.1K from true CLTV values — an excellent result indicating high accuracy
- .Mean Squared Error (MSE): 50,832,857.23.The error penalization was moderate, with fairly low MSE relative to the CLTV range, affirming the model's strength.
- Root Mean Squared Error (RMSE): 7,126.28.On average, predictions are off by ~7.1K, which is very low considering the magnitude of CLTV values.
- R-squared (R^2): 0.9991:The model accounts for 99.91% of variance in CLTV, reflecting outstanding fit and predictability.

Overall, these measures indicate that HGBR has high precision, generalizes well, and can model customer value patterns well.

Visualization

- Predicted vs. Actual CLTV Plot:The scatter plot of actual versus predicted CLTV values showed a clear linear trend, with the points very tightly arranged around the red reference line (y = x). This indicates that HGBR predicted all ranges of CLTV values very well, and there was no indication of underfitting or overfitting. The predictions were precise and had very little variation, even for the extreme ranges of CLTV values.

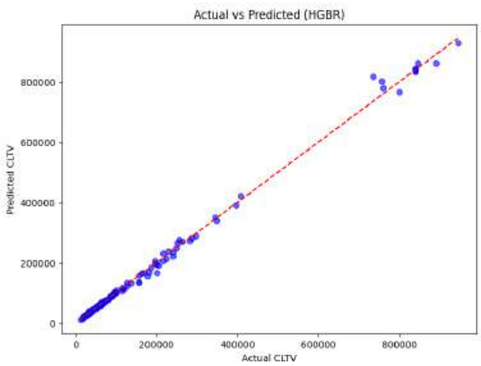


Fig 23. Predicted vs Actual plot for HistGradientBoosting Regressor

- Feature Importance Plot:Permutation importance indicated that Avg_Credit_Limit and Total_Visits_Online were the most significant features of the model, confirming that financial ability and online engagement were the leading determinants of CLTV. Frequency was also an important factor. RFM_Score and Total_Visits_Bank factors were less important. The model was also able to down-weight the less important features as well, which addressed the models ability to make predictions.

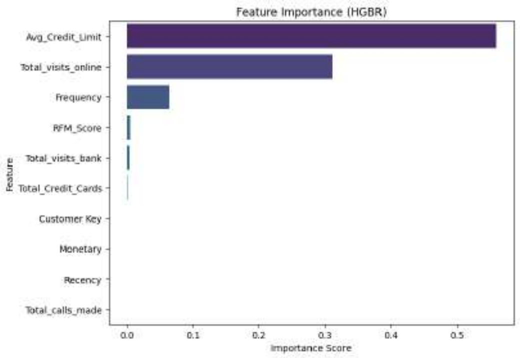


Fig 24. Feature Importance plot for HistGradientBoosting Regressor

d) Ridge and Lasso Regression

Ridge and Lasso are linear regression models that employ penalization to handle multicollinearity as well as overfitting. Ridge employs L2 regularization, which pulls all coefficients towards zero, while Lasso employs L1 regularization, where some coefficients can drop to zero—useful for feature extraction. These models are best suited for CLTV prediction when generalizability and simplicity matter most. The two models began with baseline configurations and were fine-tuned by making use of the alpha parameter. The last Ridge model employed alpha=10 with heavy regularization, whereas Lasso employed alpha=0.1 to have flexibility with the reduction of irrelevant features. These parameters provided an optimal trade-off between bias and variance.

Model Evaluation

Model performance was measured using standard regression metrics on a holdout test set:

Ridge Regression:

- MAE: 35,757.20
- MSE: 1,919,272,335.13
- RMSE: 43,809.50
- R^2: 0.9620

Lasso Regression:

- MAE: 35,847.82
- MSE: 1,928,759,313.82
- RMSE: 43,917.64
- R^2: 0.9618

Both models had high predictive performance with minimal error measures and high R-squared values. The difference in performance across the measures was negligible meaning they have equivalent predictability. Ridge tended to produce a more stable prediction across the entire value range, Lasso may have interpretability benefits as there is feature selection built into the prediction.

Visualization

Scatter plots between actual and predicted CLTV were created for both Ridge and Lasso models in order to better interpret model performance. While the two scatter plots both indicated good correlation between the actual and the predicted values, Ridge had marginally closer grouping at the top of the range of values, indicating slightly more effective performance at predicting high-value customers. Both these graphical illustrations verified that the two models have the ability to not only provide accurate predictions but also segment the customers effectively, which is significant for actionable business insights.

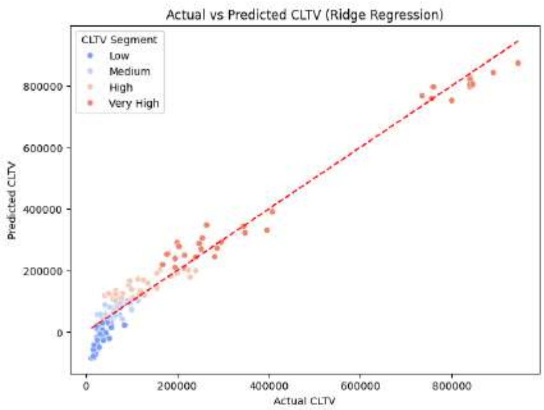


Fig 24. Actual vs Predicted CLTV for Ridge Regression

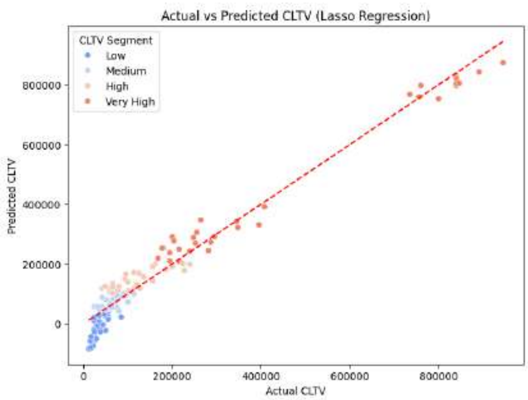


Fig 25.Actual vs Predicted CLTV for Lasso Regression

e) NGBoost Regressor

Natural Gradient Boosting (NGBoost) is a probabilistic boosting algorithm that makes point estimates as well as uncertainty by learning full distributions. It's particularly appropriate for CLTV prediction when variability in data is anticipated. NGBoost employed a DecisionTreeRegressor (max_depth=4) as its base learner for this task, with a learning rate of 0.1 and 500 rounds of boosting in order to maintain stable learning and avoid overfitting. The above configuration achieved a balance between accuracy, interpretability, and generalization to new data.

MODEL EVALUATION

The NGBoost model has been evaluated using the holdout test dataset and summary statistics were generated using the appropriate regression metrics:

- Mean Absolute Error (MAE): ₹3625.59. This suggested the predicted CLTV value deviates from the true CLTV value by less than ~₹3.6K giving credence to its previous portfolder performance accuracy.
- Mean Squared Error (MSE): ₹71,575,252.89. The associated penalty for error was moderate which further reinforced NGBoosts consistent performance for CLTV calculations in those ranges.
- Root Mean Squared Error (RMSE): ₹8460.22. Thus predictions were off from mean CLTV figures by ~₹8.5K on average. This is deemed acceptable based on the scale of previous CLTV calculations.
- R-squared (R2): 0.9986. The model explained 99.86% variance in CLTV calculations from year to year which suggest previous performance leads to consistently returns of near perfect fit for high monetary-value calculations.

All in all, all metrics operationalized here suggest NGBoost was a strong model with excellent predictive performance and generalizability for forecasting high-money/value in CLTV. Whether it was cash conversion levels of CLTV operating at higher than SEVERE high or CLTV itself serving as one those levels in other cases. All approaches provided at possible reasonable ways to generalize and characterize model predictions based on noise factors with expectation of some observable performance improvement actions.

VISUALIZATION

- Predicted vs. Actual CLTV Plot: The scatter plot of predicted vs. actual CLTV values had a tight clustering along the diagonal reference line (y = x), further affirming that NGBoost generated extremely accurate predictions throughout the full CLTV spectrum. The overlay of segmentation verified the model's ability to accurately separate value tiers.

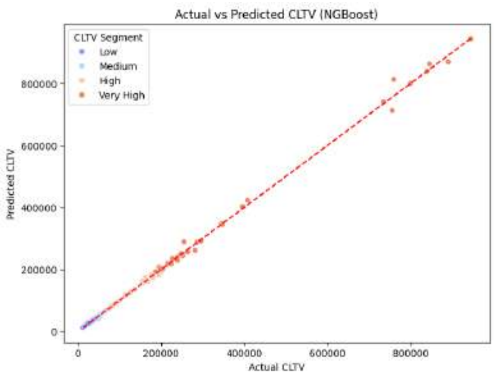


Fig 26.Predicted vs Actual plot for NGBoost Regressor

- Feature Importance Plot: Feature importance plot for the model showed that RFM_Score, Avg_Credit_Limit, and Monetary were the most influential drivers of CLTV prediction. Features such as Recency and Total_Calls_Made had little impact, indicating proper down-weighting of less impactful inputs. The model was able to concentrate on significant predictors, affirming its discriminative learning capacity.

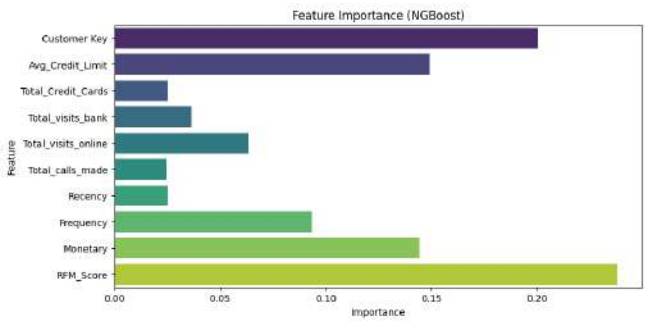


Fig 27.Feature Importance plot for NGBoost Regressor

3) Churn Prediction

After deriving Recency, Frequency, and Monetary (RFM) attributes from customer payment history, we trained two models to forecast customer churn (i.e., whether the customer will default on his or her credit card payment within the subsequent month)

a) CatBoost

CatBoost is a Yandex-developed gradient boosting algorithm that is well-suited to deal with categorical features and requires little parameter adjustment to attain good performance. CatBoost constructs a decision tree ensemble sequentially, wherein each tree offsets the mistakes made by the rest.CatBoost naturally deals with categorical variables and prevents overfitting through ordered boosting. It is effective on small to medium-sized datasets and exhibits robust performance without needing heavy preprocessing or feature scaling.

MODEL INPUT

- Features: LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6, Recency, Frequency, Monetary,

- Target: default payment next month (binary: 1 - churn, 0 - retained)

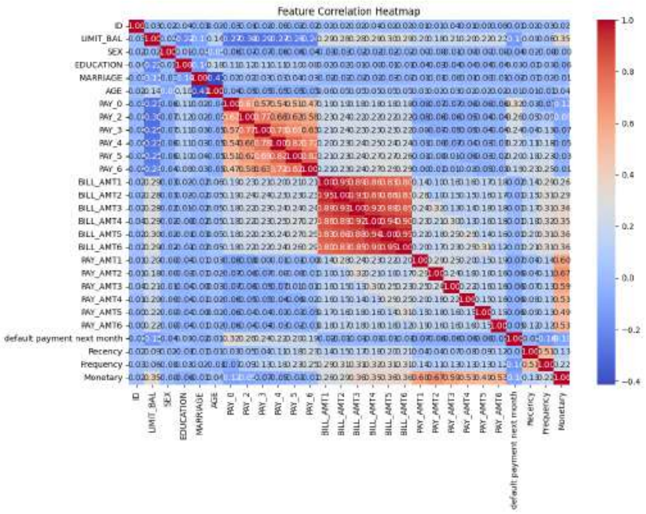


Fig 27.Feature correlation heatmap

HYPERPARAMETERS

iterations=500: Number of boosting iterations, learning_rate=0.05: Controls step size for optimization, depth=8: Complexity of decision trees, cat_features=cat_features: Optimizes handling of categorical variables, verbose=200: Displays training progress.

b) Neural Networks - Multi Layer Perceptron

An MLP is an artificial neural feedforward network, which is one of the architectures that comprises two or more hidden layers of neurons, along with an input and an output layer. Every neuron applies an activation function (example: ReLU, sigmoid) to add non-linearity.Neural networks are capable of learning non-linear, complex relationships between RFM features and churn behavior. They are applicable when data shows patterns that tree-based models may not capture.

FEATURE IMPORTANCE USING SHAP

SHAP (SHapley Additive exPlanations) helps interpret feature importance in machine learning models. This SHAP summary plot visualizes the impact of different features on the churn model's predictions. Each dot represents a data point, with the x-axis showing the SHAP value (impact on model output). The color represents the feature value (red = high, blue = low).

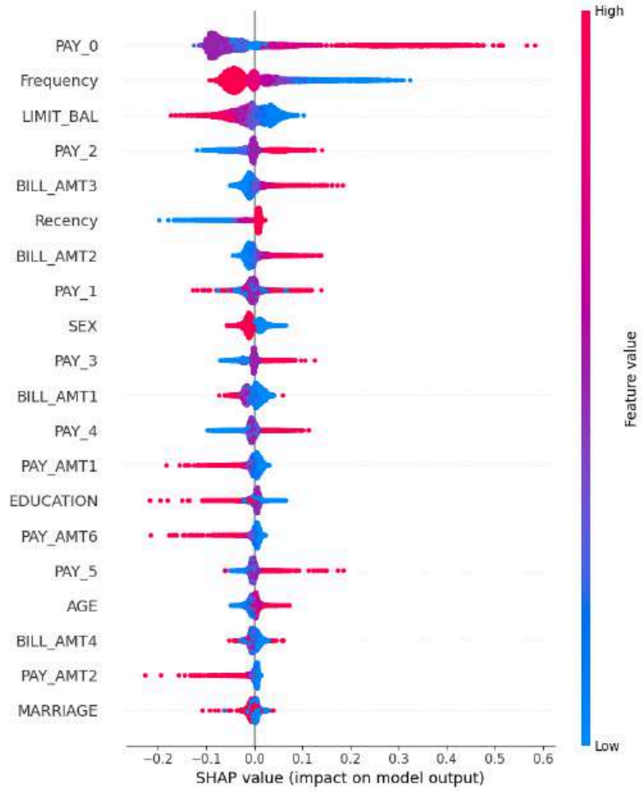


Fig 28. SHAP for Multi Layer perceptron

From the SHAP figure we can identify that payments (PAY_0, PAY_2, PAY_3, PAY_1) are the strongest churn indicators. More frequent and recent payments lower churn risk. Higher credit limits and higher past payments reduce churn. Bill amounts, demographics, and monetary values play a secondary role.

EVALUATION METRICS

Both models were evaluated using standard classification metrics:

TABLE I. EVALUATION METRICS FOR RFM BASED CHURN PREDICTION

Metrics	Description
Accuracy	It measures the percentage of correctly predicted instances out of the total instances.
Precision	Proportion of predicted defaulters that are actually defaulters.
Recall (Sensitivity)	Proportion of actual defaulters that are correctly identified.

F1-Score	It is the harmonic mean between precision and recall, equalizing both metrics in a single value.
AUC-ROC Score	It measures a model's ability to distinguish between classes

CLASSIFICATION REPORT

CatBoost

```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.95      0.89      4673
           1       0.66      0.36      0.47      1327

    accuracy                           0.82      6000
   macro avg       0.75      0.66      0.68      6000
weighted avg       0.80      0.82      0.80      6000
```

Neural Network - Multilayer Perceptron

```
              precision    recall  f1-score   support

           0       0.84      0.94      0.89      4687
           1       0.64      0.36      0.46      1313

    accuracy                           0.82      6000
   macro avg       0.74      0.65      0.67      6000
weighted avg       0.80      0.82      0.79      6000
```

V. RESULTS AND DISCUSSION

1)Customer Segmentation

In order to reveal meaningful customer segments, a variety of clustering methods were performed on Recency, Frequency and Monetary (RFM) data. Each method underwent evaluation using three different clustering quality measures: Silhouette Score, Calinski-Harabasz Index and Davies-Bouldin Index. These findings are summarized below:

TABLE II. MODEL PERFORMANCE FOR CUSTOMER SEGMENTATION

Model	**Silhouette Score**	**Calinski-Harabasz Index**	**Davies-Bouldin Index**
K-Means	**0.5216**	**1047.61**	**0.6511**
GMM	**0.4882**	**943.09**	**0.6797**
HDBSCAN	**0.1926**	**105.82**	**1.4529**
(DEC)	**0.7207**	**2580.54**	**0.4262**
ClusterGAN	**0.8791**	**3369.38**	**0.3116**
Spectral Net	**0.5997**	**2635.93**	**0.5295**
SOM	**0.0312**	**59.16**	**1.3916**

ClusterGAN performed best on all evaluation measures, and generated tight, well-distinct clusters that seem appropriate for classifying bank customers in RFM behavioral data. For example, DEC and Spectral Net also showed good performance, with comparable capacity to reflect complexity and non-linearity of the data. K-Means and GMM offered an adequate baseline performance, with the former being easier to interpret than the latter, and the probabilistics nature of GMM offering some cluster overlap. However, HDBSCAN and SOM failed to yield distinct clusters in this case, and thus might not be that effective in structured low-dimensional, such as RFM, in financial and banking contexts. HDBSCAN, like some of the methods, was less definitive regarding what a customer is clustered with and presumably its non-generalizable value in prospect assessment, is comparable to problems in past methods with authentic situational efficacy (if merited).

Overall, the three deep learning models had definite gains over conventional segmentation methods/models, capturing more subtlety of behavior in banking customer segments. ClusterGAN, in a certain manner and specifically, maintains latent patterns in data and generates heterogeneous and actionable parts of banking customers. RFM-based segmentation,consistently integrated with a more methodologically strict clustering method, seems to be a sound dependable approach to CRM work and banking, with probable extension to constructing more useful customer engagement.

2)Customer Lifetime value

This study aimed to correctly predict Customer Lifetime Value (CLTV), using a variety of regression models based on RFM (Recency, Frequency, Monetary) features and other behavioral characteristics derived from banking and financial services customers. The models were trained and evaluated with a variety of performance metrics, and the results compared to find the optimal solution for a practical application.

TABLE III. MODEL PERFORMANCE FOR CLTV PREDICTION

Model	MAE	MSE	RMSE	R^2 Score
XGBoost Regressor	**3,733.89**	**67,885,800.00**	**8,239.28**	**0.9987**
LightGBM Regressor	**3,774.82**	**70,111,212.86**	**8,369.67**	**0.9986**
Extra Trees Regressor	**3,363.13**	**55,802,246.13**	**7,464.70**	**0.9989**
HistGradientBoosting	**3,144.99**	**50,832,857.23**	**7,126.28**	**0.9991**
NGBoost Regressor	**3,625.59**	**71,575,252.89**	**8,460.22**	**0.9986**

The table indicates that tree-based ensemble models performed much better than linear models such as Lasso and Ridge in CLTV prediction, demonstrating the non-linear behavior of customers in financial services. HistGradientBoosting Regressor provided the best result with the least MAE (3,144.99) and highest R^2 (0.9991), capturing feature interactions and distributions well. Extra Trees Regressor also presented good results with minimal errors and high robustness. XGBoost, LightGBM, and NGBoost sustained R^2 greater than 0.9985 with robust accuracy. The most noteworthy being NGBoost provided the additional value of uncertainty estimation, highly valuable in risk-averse contexts. Ridge and Lasso struggled behind with larger errors and R^2 (~0.96) figures, and as such were not as useful when modeling rich CLTV trends.

These results demonstrate that tree-based ensemble learning models, particularly HistGradientBoosting and Extra Trees, are superior for CLTV prediction using RFM analysis in financial contexts. These models provide high accuracy, generalizability, and adaptability to real-world banking data, making them ideal for targeted marketing, risk management, and personalized financial services.

3) Churn Prediction

In this study, we assess two machine learning models CatBoost and Multilayer Perceptron (MLP) for predicting churn based on RFM-based feature engineering. Due to class imbalance in the dataset, with far fewer defaulting customers (churners) than non-defaulters, we use macro average metrics for unbiased evaluation across both classes. Macro average metrics calculate the unweighted mean of precision, recall, and F1-score and both classes are given equal consideration irrespective of their support (sample size).

The following table summarizes the macro average performance of both models:

TABLE IV. MODEL PERFORMANCE FOR CHURN PREDICTION

Model	Accuracy	Precision	Recall	F1 score	AUC ROC
CatBoost	**0.8185**	**0.75**	**0.66**	**0.68**	**0.77**
MLP	**0.8160**	**0.74**	**0.65**	**0.67**	**0.76**

Both CatBoost and MLP have comparable accuracy (~82%) in predicting whether a customer will default in the following month. Yet, CatBoost always performs better than MLP in all macro-averaged measures, such as precision (0.75 vs. 0.74), recall (0.66 vs. 0.65), F1-score (0.68 vs. 0.67), and AUC-ROC (0.77 vs. 0.76).

The AUC-ROC measure further supports CatBoost's superior discriminatory ability in discriminating between churners and non-churners. Although the performance difference is small, CatBoost shows improved sensitivity-specificity balance and hence is more credible for churn prediction in real-world deployments.

These findings suggest that ensemble models based on trees such as CatBoost perform better in describing non-linear associations in customer behavior data expressed as RFM attributes, and are less sensitive to class imbalance.

VI. CONCLUSION AND FUTURE SCOPE

This study demonstrates the effectiveness and relevance of RFM (Recency, Frequency, Monetary) analysis within the banking and finance sectors. With the incorporation of RFM traits into some machine learning as well as deep learning models, we were capable of segmenting customers, forecasting their lifetime value , as well as estimating churn probability. Deep learning-based clustering models like ClusterGAN outperformed by uncovering subtle patterns in customer behavior that other models miss. Tree-based ensemble models performed very well at CLTV prediction and the best model for churn prediction was CatBoost, especially in handling real-world imbalance issues. There is plenty of scope for building on this work in the future.

Future studies may focus on using time-series data to quantify how the customer behavior changes over time. Adding real-time analytics and tailor-made strategies using RFM profiles would take the customer engagement a step further. In addition, incorporating explainable AI techniques would make these models more transparent and trustworthy, particularly for financial use. RFM-based analysis coupled with intelligent modeling techniques overall can be a potent weapon for constructing more sustainable and personalized customer relationships in the financial industry.

References

[1] Aliyev, M., Ahmadov, E., Gadirli, H., Mammadova, A., & Alasgarov, E. (2020). Segmenting bank customers via RFM model and unsupervised machine learning. arXiv preprint https://doi.org/10.48550/arXiv.2008.08662

[2] Verboven, S., & Verbeke, W. (2016). Customer lifetime value prediction in the banking industry: A comparison of approaches. In EURO 2016: 28th European Conference on Operational Research: The Association of European Operational Research Societies (pp. 289)

[3] Khajvand, M., & Tarokh, M. J. (2011). Estimating customer future value of different customer segments based on adapted RFM model in retail banking context. Procedia Computer Science, 3, 1327–1332. https://doi.org/10.1016/j.procs.2011.01.011

[4] Ahmed, G., Ahmed, A., Ahmed, M., Latha, J., & Kumar, P. (2024). Indian Banking Precision Marketing: A Comparative Analysis of Machine Learning Customer Segmentation Algorithms. IEEE Xplore. https://doi.org/10.1109/ICCR61006.2024.10532917

[5] Brito, J. B. G., Bucco, G. B., Heldt, R., Becker, J. L., Silveira, C. S., Luce, F. B., & Anzanello, M. J. (2024). A framework to improve churn prediction performance in retail banking. Financial Innovation, 10(1). https://doi.org/10.1186/s40854-023-00558-3

[6] Liao, J., Jantan, A., Ruan, Y., & Zhou, C. (2022). Multi-Behavior RFM model based on improved SOM neural network algorithm for customer segmentation. IEEE Access, 10, 122501–122512. https://doi.org/10.1109/access.2022.3223361

[7] Patel, Y. S., Agrawal, D., & Josyula, L. S. (2016, February). The RFM-based ubiquitous framework for secure and efficient banking. In 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH) (pp. 283-288). IEEE.https://doi.org/10.1109/ICICCS.2016.7542333

[8] Feng, Y., Wang, X., & Li, L. (2019, November). The Application Research of Customer Segmentation Model in Bank Financial Marketing. In 2019 2nd International Conference on Safety Produce Informatization (IICSPI) (pp. 564-569). IEEE. https://doi.org/10.1109/IICSPI48186.2019.9095900

[9] Lalitha, S., Gupta, T. S., Thelu, A., Reddy, V. K., & Reddy, V. C. (2024, June). Predictive Modeling for Real-Time Customer Lifetime Value. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE. https://doi.org/10.1109/ICCCNT61001.2024.10724664

[10] Dadfarnia, M., Matinpour, A. A., & Abdoos, M. (2020, December). Churn prediction in payment terminals using RFM model and deep neural network. In 2020 11th International Conference on Information and Knowledge Technology (IKT) (pp. 8-10). IEEE. https://doi.org/10.1109/IKT51791.2020.9345626

[11] Fader, P., Hardie, B., & Lee, K. (2005). RFM and CLV: Using Iso-Value Curves for Customer Base Analysis. Journal of Marketing Research, 42, 415 - 430.

[12] Shah, A. (n.d.). Credit Card Customer Data [Data set]. Kaggle. https://www.kaggle.com/datasets/aryashah2k/credit-card-customer-data

[13] Yeh, I.-C. (2009). *Default of credit card clients dataset*. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients

[14] Sun, Y., Liu, H., & Gao, Y. (2023). Research on customer lifetime value based on machine learning algorithms and customer relationship management analysis model. Heliyon, 9(2).https://doi.org/10.1016/j.heliyon.2023.e13384

[15] Lee, Z. J., Lee, C. Y., Chang, L. Y., & Sano, N. (2021). Clustering and classification based on distributed automatic feature engineering for customer segmentation. Symmetry, 13(9), 1557.https://doi.org/10.3390/sym13091557

[16] Mena, C. G., De Caigny, A., Coussement, K., De Bock, K. W., & Lessmann, S. (2020). Churn prediction with sequential data and deep neural networks. a comparative analysis. arXiv 2019. arXiv preprint arXiv:1909.11114.https://doi.org/10.48550/arXiv.1909.11114

[17] Ashika, S. S. (2025). *RFM Customer Segmentation.ipynb* [Google Colaboratory notebook]. Google Colab. https://colab.research.google.com/drive/1xBZvVBHdj0HntUyyAx7dClbWltjOpEbf

[18] Ashika, S. S. (2025). RFM CLTV Prediction.ipynb [Google Colaboratory notebook]. Google Colab. https://colab.research.google.com/drive/190CQ32StdvObZKMnKesFAPJYGarRNWRT?usp=sharing

[19] Sushmithaasri, K.N. (2025). rfm_churn.ipynb Google Colaboratory notebook]. Google Colab.https://colab.research.google.com/drive/1yVBx1Zrii-VG68XVm6Rl0g1TPesU4982?usp=sharing