A Data-Driven Approach to Bank Customer Segmentation and Churn Forecasting

Palak Mangla
IT department (AI & ML)
IGDTUW
Delhi, India
e-mail: palak043btaiml22@igdtuw.ac.in

Radhika Gupta
IT department (AI & ML)
IGDTUW
Delhi, India
e-mail: radhika050btaiml22@igdtuw.ac.in

Shreya Singh
IT department (AI &ML)
IGDTUW
Delhi, India
e-mail: shreya062btaiml22@igdtuw.ac.in

Abstract

Attracting new customers is no longer a good strategy for mature businesses since the cost of retaining existing customers is much lower. For this reason, customer churn management becomes instrumental for any service industry.

This analysis is combining churn prediction and customer segmentation and aims to come up with an integrated customer analytics outline for churn management. There are five components in this analysis, starting with data pre-processing, exploratory data analysis, customer segmentation, customer characteristics analytics and churn prediction

Customer data of a bank is used for this analysis. After preprocessing and exploratory data analysis, customer segmentation is carried out using K-means clustering. A Random Forest model is used focusing on optimizing f-1 score to validate the clustering and get feature importance. By using this model, customers are segmented into different groups, which sanctions marketers and decision makers to implement existing customer retention strategies more precisely. Then different machine learning models are used with the preprocessed data along with the segmentation prediction from the K-means clustering model. For this type of modeling, models were optimized for precision. To address class imbalance Synthetic Minority Oversampling Technique (SMOTE) is applied to the training set. For factor analysis feature importance of models are used. Based on cluster characteristics, clients are labeled as Low value frequent users of services, High risk clients, Regular clients, Most loyal clients, and High value clients. Final model accuracy is high with good precision of predicting churn.

Keywords— k-mean clustering, random forest model, data pre-processing, exploratory data analysis, customer segmentation, customer characteristics analytics and churn prediction

I. Introduction

Customer churn is a big issue that occurs when consumers abandon your products and go to another provider. Because of the direct impact on profit margins, firms are now focusing on identifying consumers who are at danger of churning and keeping them through tailored promotional offers. Customer churn analysis and customer turnover rates are frequently used as essential business indicators by banks, insurance firms, streaming service providers, and telecommunications service providers since the cost of maintaining existing customers is significantly less than the cost of obtaining a new one.

When it comes to customers, the financial crisis of 2008 changed the banking sector's strategy. Prior to the financial crisis, banks were mostly focused on acquiring more and more clients. However, once the market crashed after the market imploded, banks realized rapidly that the expense of attracting new clients is multiple times higher than holding existing ones, which means losing clients can be monetarily unfavorable. Fast forward to today, and the global banking sector has a market capitalization of $7.6 trillion, with technology and laws making things easier than ever to transfer assets and money between institutions. Furthermore, it has given rise to new forms of competition for banks, such as open banking, neo-banks, and fin-tech businesses (Banking as a Service (BaaS))[1]. Overall, today's consumers have more options than ever before, making it easier than ever to transfer or quit banks altogether. According to studies, repeat customers seem to be more likely to spend 67 percent more on a bank's products and services, emphasizing the necessity of knowing why clients churn and how it varies across different characteristics. Banking is one of those conventional sectors that has undergone continuous development throughout the years. Nonetheless, many banks today with a sizable client base expecting to gain a competitive advantage have not tapped into the huge amounts of data they have, particularly in tackling one of the most well-known challenges, customer turnover.

Churn can be expressed as a level of customer inactivity or disengagement seen over a specific period. This expresses itself in the data in a variety of ways e.g., frequent balance transfers to another account or unusual drop in average balance over time. But how can anyone look for churn indicators? Collecting detailed feedback on the customer's experience might be difficult. For one thing, surveys are both rare and costly. Furthermore, not all clients receive it, or bother to reply to it. So, where else can you look for indicators of future client dissatisfaction? The solution consists in identifying early warning indicators from existing data. Advanced machine learning and data science techniques can learn from previous customer behavior and external events that lead to churn and use this knowledge to anticipate the possibility of a churn-like event in the future.

II. LITERATURE REVIEW

The Indian banking sector has experienced significant growth and transformation in recent years. As one of the largest and fastest-growing banking markets globally, it faces unique challenges related to customer retention and service quality. This literature review explores the

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

importance of customer churn prediction and segmentation in Indian banking services, drawing upon Indian data and statistics to underscore its significance.According to data from the Reserve Bank of India (RBI), the number of scheduled commercial banks in India increased from 89 in 1969 to over 1,500 by 2021, demonstrating the sector's dynamism. This growth has intensified competition among banks for market share and customer loyalty.

Customer churn, or the phenomenon of customers discontinuing their relationships with a business, is a critical concern in the banking sector. Accurate prediction of customer churn can enable banks to implement proactive strategies to retain valuable customers. We Have used various machine learning models and customer segmentation techniques used to address this challenge.

CHURN PREDICTION USING MACHINE LEARNING

Various machine learning models, including logistic regression (LR), decision tree (DT), k-nearest neighbor (KNN), random forest (RF), were used in this study (Kaur & Kaur, 2020) to estimate the likelihood that a client will leave. Performance measures including memory, accuracy, and others are compared. Support vector machines (SVM), which might increase system performance, were not taken into account in this study. Inspired by this, we added the SVM model to our work.

Guliyev and Tatoğlu (2021) emphasized the importance of explainable AI and employed SHapley Additive exPlanations (SHAP) values to evaluate machine learning models.

Utilizing actual banking data, the research aimed to estimate the explainable machine learning model and assess a variety of machine learning models using test data. The XG-boost model fared better than other machine learning techniques in categorizing clients who churn.Yaseen (2021) utilized feature selection techniques to determine important variables for churn prediction.They used the wrapper-based feature selection approach, where particle swarm optimization (PSO) was applied for searches, and different classifiers, such as decision tree, naive bayes, k-nearest neighbor, and logistic regression, were applied for evaluation to judge the enactment on optimally sampled and condensed datasets. Last, but not least, simulations showed that their recommended strategy did well for forecasting churners and might therefore be helpful for the telecommunications sector’s constantly expanding rivalry.

In contrast, our approach combines oversampling with different models, including SVM, to achieve superior results.

Elyusufi and Ait Kbir (2022) explored ensemble learning techniques to predict churn, achieving improved accuracy. In our study, we also utilize ensemble learning but introduce SVM as a model choice.

Rahman and Kumar (2020) employed KNN, SVM, DT, and RF classifiers but did not consider ensemble learning methods. In contrast, our research explores the use of ensemble learning techniques, such as random forests and AdaBoost, which can enhance predictive accuracy.

2.2. Customer Segmentation and Churn Prediction

Customer segmentation is a critical precursor to effective churn prediction and customer retention strategies. Sivasankar and Vijaya (2017) employed fuzzy c-means, probabilistic fuzzy c-means, and k-means clustering for customer segmentation, enhancing classification accuracy using decision trees. In our research, we combine k-means clustering with SVM to leverage the benefits of both techniques.

Using the clustering technique and the k-means clustering algorithm, Olaniyi et al. (2020) analyzed consumer competency and sector continuity to anticipate customer behavior. The data were grouped into three labels according to the inflow and outflow of transactions.They used support vector machines to classify customers based on transaction behavior, achieving a high accuracy of 97%. This approach is similar to our integration of SVM for churn prediction but differs in the specific application. Zhang et al. (2022) explored regression-based models for telecom customer churn prediction after segmentation, achieving a prediction accuracy of 93.94%. While we adopt segmentation and prediction techniques, our research focuses on the banking industry, thereby extending the applicability of the approach.

The empirical findings of the above studies demonstrated that consumer segmentation would greatly enhance each predictor, so we experimented with the data from the banking industry to support this.

II. METHODOLOGY

The Kaggle website (https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers) provides a freely accessible dataset for the prediction job. The final two columns of the 23 variables should be eliminated because they are useless for categorizing. After removing the final two columns, the dataset now has 21 variables, with a total of 10,127 entries.

1) CUSTOMER SEGMENTATION USING KMEANS

Customer Segmentation is the process of dividing a customer base into distinct groups or segments based on specific characteristics or behaviors. It helps businesses better understand their customers, tailor marketing strategies, and provide more personalized services.

K-Means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into distinct, non-overlapping groups or clusters based on the similarity of data points. It's a type of centroid-based clustering algorithm that seeks to divide data points into K clusters, where K is a user-defined parameter.

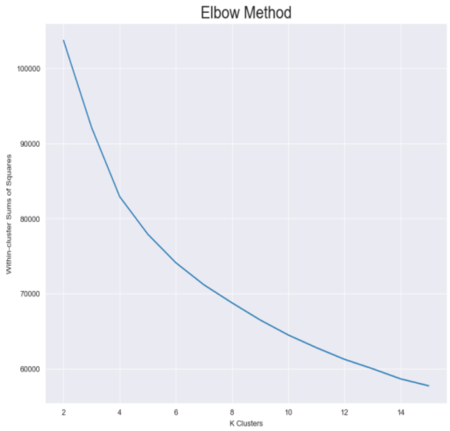
1.1 Initialization

The K-Means algorithm commences with the selection of the number of clusters, denoted as K. This parameter is essential and requires careful consideration, as it influences the granularity of data grouping. Various techniques exist for determining an optimal K, including the Elbow Method and the Silhouette Score, which were used in this study.

1.1.1 Elbow Method

The Elbow Method is a graphical technique used to determine the optimal number of clusters (K) in a dataset for clustering algorithms like K-Means. It helps you find the point at which adding more clusters does not significantly improve the model's performance, leading to an "elbow" point in the plot. It works by plotting the number of clusters against a performance metric (typically the Within-Cluster Sum of Squares or WCSS) and observing the graph's shape. As K increases, the WCSS tends to decrease because more clusters allow data points to get closer to their cluster centroids. However, the Elbow Method identifies the point at which the reduction in WCSS slows down, forming an "elbow" in the graph. This elbow point is considered the optimal K because it represents a trade-off between capturing meaningful patterns in the data (with more clusters) and simplicity (with fewer clusters). It helps practitioners make an informed decision about the number of clusters to use in their analysis.

Our result:



1.1.1.1 WCSS

The Within-Cluster Sum of Squares (WCSS) is a crucial metric in clustering algorithms, such as K-Means, used to assess the quality of cluster assignments. WCSS measures how compactly data points are grouped within clusters. It is computed by summing the squared distances between each data point and the centroid (the center) of its assigned cluster across all clusters. In essence, WCSS quantifies the extent to which data points are close to the center of their respective clusters. A lower WCSS indicates that data points are tightly clustered around their centroids, suggesting well-defined and cohesive clusters. This metric serves as a key component in techniques like the Elbow Method for determining the optimal number of clusters, helping practitioners strike a balance between cluster quality and simplicity in their analyses.

Mathematically, WCSS can be expressed as:

$$WCSS = \sum_{i=1}^{K} \sum_{j=1}^{n_i} \left\| x_j - \mu_i \right\|^2$$

Where:

- *K* is the total number of clusters.
- *ni* is the number of data points in cluster *i*.
- *xj* represents a data point in cluster *i*.
- *μi* is the centroid of cluster *i*.
- ∥*xj*−*μi*∥ represents the Euclidean distance between data point *xj* and the centroid *μi* of cluster *i*.

1.1.2 Silhouette Score

The Silhouette Score is a metric used to evaluate the quality of clusters in unsupervised machine learning, particularly in clustering algorithms like K-Means. It quantifies how well-separated and internally coherent the data points within clusters are. The Silhouette Score calculates, for each data point, two values: "a," representing the average distance from the data point to other points within the same cluster (cohesion), and "b," representing the smallest average distance from the data point to points in a different cluster (separation). It then computes the Silhouette Coefficient, which is the difference between "b" and "a" divided by the greater of the two. This coefficient ranges from -1 (indicating poor clustering, with points assigned to the wrong clusters) to +1 (indicating excellent clustering, with well-separated and distinct clusters), with 0 suggesting overlapping clusters. A higher Silhouette Score indicates better-defined and more appropriate clusters, aiding in the selection of the optimal number of clusters for a given dataset.

1.2 Assignment Phase

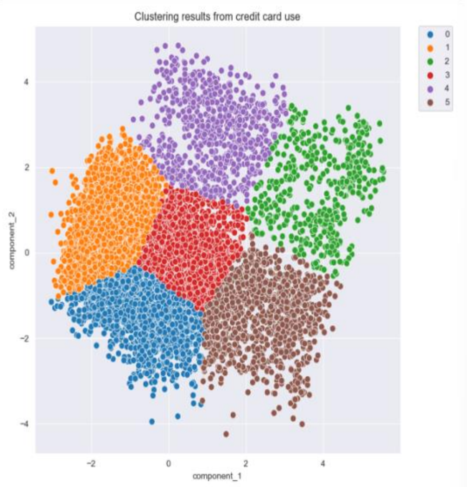
In the assignment phase, K-Means iteratively assigns data points to the nearest cluster centroid based on a chosen distance metric. Euclidean distance, a commonly utilized metric, measures the dissimilarity between data points and cluster centroids. Each data point is assigned to the cluster with the closest centroid, signifying its membership within that cluster.

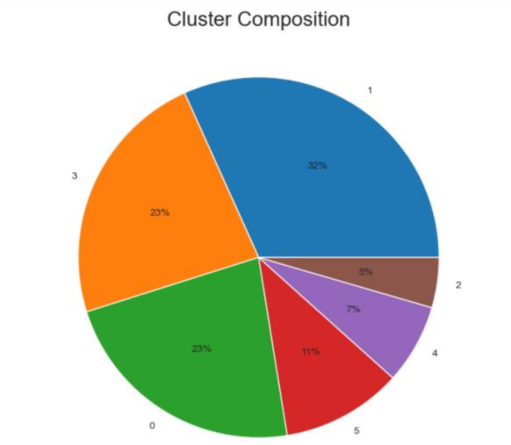
1.3 Update Phase

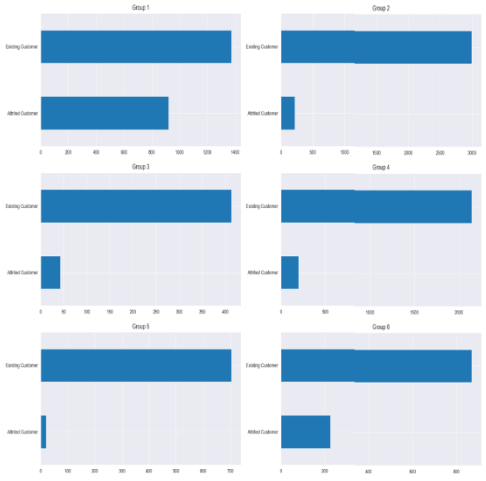
Following the assignment of data points to clusters, the centroids of each cluster are updated. The updated centroid positions are calculated as the mean of all data points belonging to the respective cluster. This process refines the cluster centers and represents the central tendency of the data points within each cluster.

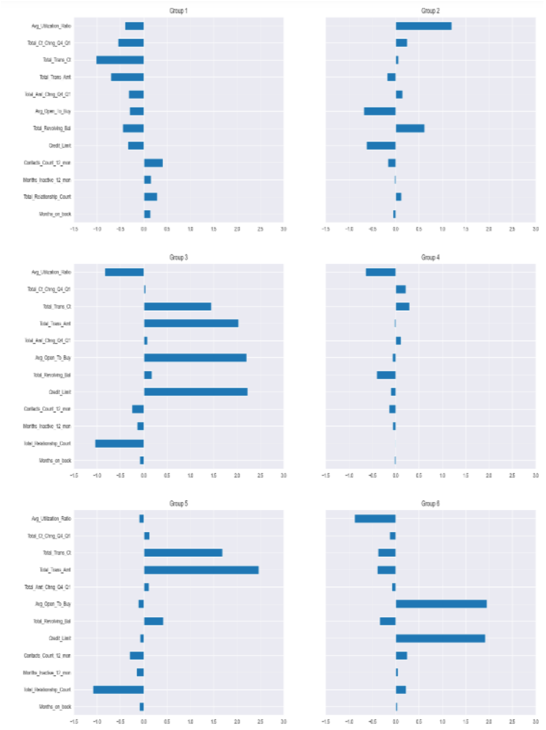
1.4 Iterative Convergence

The assignment and update phases are performed iteratively until a predefined stopping criterion is met. Common stopping criteria include a maximum number of iterations or reaching convergence, where the centroids no longer exhibit substantial changes between iterations. Convergence ensures that the algorithm has effectively partitioned the data into coherent clusters.









Looking at the charts above, we can see the groups have the following characteristics:

Group 1: High transaction count and amount, higher balance, low relationship count, lower contact count
Group 2: Low transaction count and amount, lower count and amount change from Q4 to Q1, high contact count, higher relationship count, higher months on book, higher months inactive, lower balance
Group 3: High utilization ratio, higher count and amount change from Q4 to Q1, low avg open to buy, higher balance, low credit limit
Group 4: Low transaction count and amount, low utilization ratio, lower count and amount change from Q4 to Q1, high contact count, higher relationship count, lower balance
Group 5: Lower utilization ratio, higher transaction count, higher transaction count change from Q1 to Q4, low balance
Group 6: High transaction count and amount, high avg open to buy, high credit limit, low relationship count, low utilization ratio
Based on the results above, we decided to give the following names to each of the groups:
Group 1: Higher balance heavy users
Group 2: Long-time very light users
Group 3: Low limit, high balance light users
Group 4: High limit, low balance light users
Group 5: Low limit, low balance light users
Group 6: High limit, low balance heavy users

2) EXPLORATORY DATA ANALYSIS (EDA)

Data description:

Variable	**Type**	**Description**
Clientnum	Num	Client number. Unique identifier for the customer holding the account
Attrition_Flag	obj	Internal event (customer activity) variable - if the account is closed then 1 else 0
Customer_Age	Num	Demographic variable - Customer's Age in Years
Gender	obj	Demographic variable - M=Male, F=Female
Dependent_count	Num	Demographic variable - Number of dependents
Education_Level	obj	Demographic variable - Educational Qualification of the account holder (example: high school, college graduate, etc.)

Marital_Status	obj	Demographic variable - Married, Single, Divorced, Unknown
Income_Category	obj	Demographic variable - Annual Income Category of the account holder (< $40K, $40K - 60K, $60K - $80K, $80K-$120K, > $120K, Unknown)
Card_Category	obj	Product Variable - Type of Card (Blue, Silver, Gold, Platinum)
Months_on_book	Num	Months on book (Time of Relationship)
Total_Relationship_Count	Num	Total no. of products held by the customer
Months_Inactive_12_mon	Num	No. of months inactive in the last 12 months
Contacts_Count_12_mon	Num	No. of Contacts in the last 12 months
Credit_Limit	Num	Credit Limit on the Credit Card
Total_Revolving_Bal	Num	Total Revolving Balance on the Credit Card
Avg_Open_To_Buy	Num	Open to Buy Credit Line (Average of last 12 months)
Total_Amt_Chng_Q4_Q1	Num	Change in Transaction Amount (Q4 over Q1)
Total_Trans_Amt	Num	Total Transaction Amount (Last 12 months)
Total_Trans_Ct	Num	Total Transaction Count (Last 12 months)
Total_Ct_Chng_Q4_Q1	Num	Change in Transaction Count (Q4 over Q1)
Avg_Utilization_Ratio	Num	Average Card Utilization Ratio

Data preprocessing is a critical stage in machine learning that improves the quality of the data to en-courage the extraction of valuable insights from the data (Alexandropoulos et al., 2019). Preparing (cleaning and arranging) raw data to make it acceptable for creating and training Machine Learning models is known as data preprocessing in machine learning. Data preprocessing in machine learning is, to put it simply, a data mining approach that converts raw data into a format that is legible and intelligible. First, the Unknown values discovered in the data preparation phase will be eliminated from the dataset. In the same stage, Customer_Age and Months_on_Book were figured out to have a dis-tribution similar to the normal distribution. Thus, the standardization method will be applied to Cus-tomer_Age and Months_on_Book. With the rest of the quantitative data that needs to be trans-formed, normalization will be used. For the categories with two values, such as Gender and Attrition_Flag, label encoding will then be used. For category data with numerous values, such as Educa-tion_Level, Marital_Status, Income_Category, and Card_Category, one hot encoding will be utilized. The final steps include replacing the Divorced value with the Single value and the College value with the Graduate value, as was specified during the data preparation process.

Feature	Transformation method
CLIENTNUM	Not used in prediction
Attrition_Flag	Label encoding
Customer_Age	Standardization
Gender	Label encoding
Dependent_count	Unchanged
Education_Level	One-hot encoding
Marital_Status	One-hot encoding
Income_Category	One-hot encoding
Card_Category	One-hot encoding
Months_on_book	Standardization
Total_Relationship_Count	Unchanged
Months_Inactive_12_mon	Unchanged
Contacts_Count_12_mon	Unchanged
Credit_Limit	Normalization
Total_Revolving_Bal	Normalization
Avg_Open_To_Buy	Normalization
Total_Amt_Chng_Q4_Q1	Unchanged
Total_Trans_Amt	Normalization
Total_Trans_Ct	Normalization
Total_Ct_Chng_Q4_Q1	Unchanged
Avg_Utilization_Ratio	Unchanged

3) DATA PREPROCESSING:

Pearson Product-Moment Correlation and Spearman Rank-Order Correlation are two commonly used statistical methods for measuring the relationship or association between two sets of data. However, they have different characteristics and are suited for different types of data.

3.1 Pearson Product-Moment Correlation:

Assumption: Pearson correlation assumes that the data follows a linear relationship, which means that as one variable increases, the other tends to increase (positive correlation) or decrease (negative correlation) in a consistent and linear manner.

Data Type: It is appropriate for continuous, numeric data with roughly equal intervals. It's also sensitive to outliers, meaning that extreme data points can have a significant impact on the correlation coefficient.

Calculation: The Pearson correlation coefficient, denoted as "r," ranges from -1 to +1.

A positive value of "r" indicates a positive linear relationship (as one variable goes up, the other tends to go up).

A negative value of "r" indicates a negative linear relationship (as one variable goes up, the other tends to go down).

An "r" value close to 0 suggests a weak or no linear relationship.

Use Cases: Pearson correlation is commonly used in fields like economics, psychology, and social sciences to analyze relationships between variables like income and education level, or temperature and ice cream sales.

3.2 Spearman Rank-Order Correlation:

Assumption: Spearman correlation does not assume a linear relationship between variables. Instead, it assesses the strength and direction of the monotonic relationship, which means that as one variable increases, the other either tends to increase or tends to decrease consistently.

Data Type: It is appropriate for both continuous and ordinal data. It is less sensitive to outliers compared to Pearson correlation.

Calculation: Spearman correlation is calculated based on the ranks (ordinal positions) of the data points rather than their actual values. The Spearman correlation coefficient, denoted as "ρ" (rho), ranges from -1 to +1, similar to Pearson correlation.

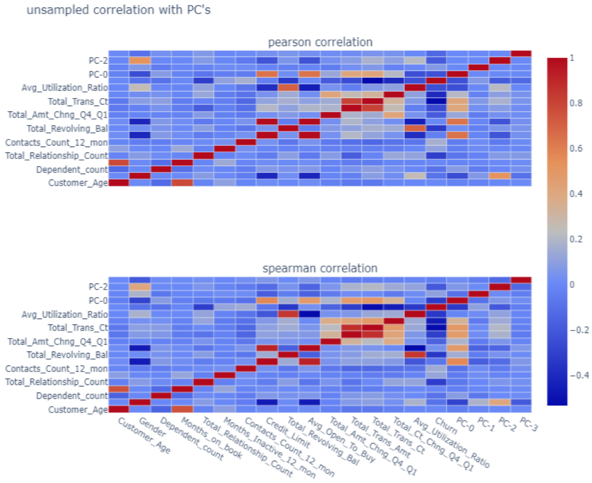
A positive ρ indicates a monotonic positive relationship.

A negative ρ indicates a monotonic negative relationship.

A ρ value close to 0 suggests a weak or no monotonic relationship.

Use Cases: Spearman correlation is often used when the assumption of linearity is not met or when working with ordinal data. It's also robust against outliers, making it suitable for data with extreme values.

In summary, Pearson correlation is used to measure linear relationships between continuous or numeric data, while Spearman correlation assesses monotonic relationships and can be used with both continuous and ordinal data. The choice between the two methods depends on the nature of the data and the research question being addressed.



4) SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is a resampling technique used in machine learning to address the problem of class imbalance, especially in classification tasks. Class imbalance occurs when one class (usually the minority class) is underrepresented in the dataset compared to the other class(es).

SMOTE helps balance the class distribution by generating synthetic examples of the minority class. It does this by creating synthetic data points that are similar to existing minority class samples but introduces slight variations.

Here's how SMOTE works:

1) Select a Minority Data Point: SMOTE starts by randomly selecting a data point from the minority class.

2) Find Nearest Neighbors: For the selected data point, SMOTE identifies its k-nearest neighbors from the same class (minority class). The value of "k" is specified by the user.

3) Generate Synthetic Samples: SMOTE generates synthetic data points by interpolating between the selected data point and its nearest neighbors. For each feature, it computes the difference between the feature values of the selected point and its neighbors, multiplies this difference by a random number between 0 and 1, and adds the result to the selected point. This process is repeated to create a specified number of synthetic samples.

4) Repeat: Steps 1 to 3 are repeated until the desired level of oversampling is achieved.

SMOTE effectively increases the number of minority class samples in the dataset, which can lead to a more balanced dataset and better model performance, especially for classifiers that are sensitive to class imbalance.

5) PRINCIPAL COMPONENT ANALYSIS OF ONE HOT ENCODED DATA

Principal Component Analysis (PCA) can be applied to one-hot encoded data to reduce its dimensionality while preserving most of the information. One-hot encoding is often used for categorical variables, but it can lead to high-dimensional datasets with many binary (0/1) columns. PCA helps in reducing the number of columns (features) while retaining the most significant information.

Here are the steps to perform PCA on one-hot encoded data:

1) One-Hot Encoding: First, ensure that your categorical variables are properly one-hot encoded, resulting in binary columns where each column represents a category.

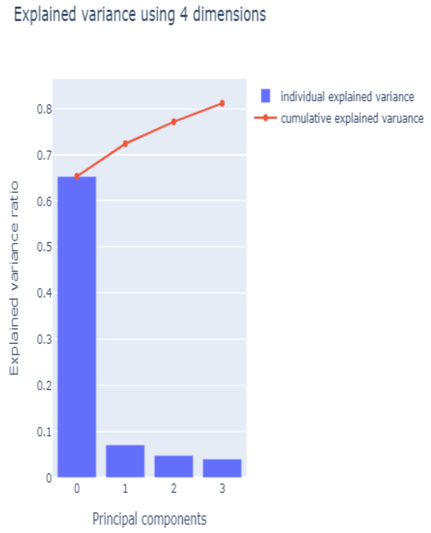
2) Standardize the Data: Standardization is crucial when applying PCA because it scales the features to have a mean of 0 and a standard deviation of 1. This ensures that features with larger scales don't dominate the PCA process.

3) Apply PCA: Use PCA to transform the standardized one-hot encoded data into its principal components. You can specify the number of components you want to keep, or you can let PCA choose the number based on explained variance.

4) Select the Number of Components: Decide how many principal components to retain based on the explained variance. A common approach is to select enough components to retain a certain percentage of the total variance, such as 95% or 99%. This is often done using the cumulative explained variance plot.

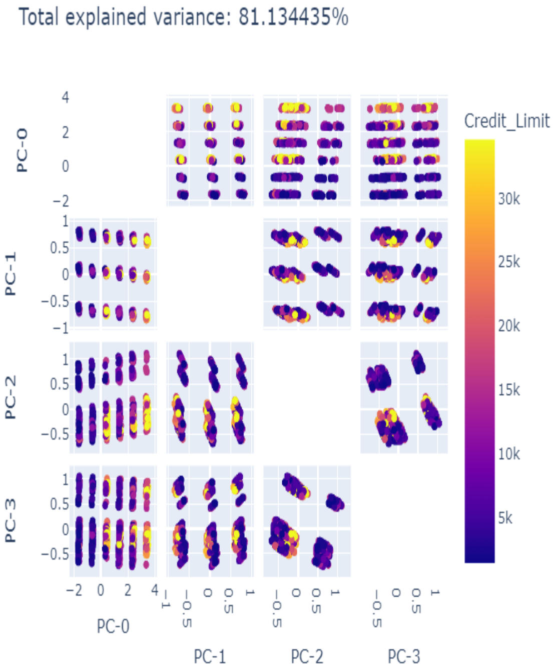
5) Transform Data: Apply the selected number of principal components to transform the data. The transformed data will have fewer columns (equal to the number of components selected) but should capture most of the variance in the original data.

6) Optional: Inverse Transform: If needed, you can inverse transform the reduced data to get an approximation of the original one-hot encoded data, but with reduced dimensionality.



6) PC

The "total explained variance" is the cumulative percentage of variance explained by all the selected principal components. It's a crucial metric because it tells you how much information from your original dataset is retained when you reduce the dimensionality by selecting a subset of PCs. In your case, with a total explained variance of 81.134435%, it means that the combination of the first N_COMPONENTS principal components you've chosen explains 81.134435% of the variability in your data.



7) MODELING AND CROSS VALIDATION:

Random Forest, AdaBoost, and Support Vector Classifier (SVC) are all popular machine learning algorithms used for various types of supervised learning tasks, including classification and regression. Each algorithm has its own characteristics and strengths, making it suitable for different types of datasets and problem scenarios:

1) Random Forest:

Algorithm Type: Ensemble Learning (Bagging)

Strengths:

1) Robust to overfitting: Random Forest reduces overfitting by combining multiple decision trees.

2) Handles both classification and regression tasks.

3) Can handle a mix of categorical and numerical features.

4) Provides feature importance, allowing feature selection.

How it Works: Random Forest builds multiple decision trees (a forest) by resampling the training data and using bootstrapped samples. Each tree is constructed based on a random subset of features. The final prediction is an ensemble average or majority vote of individual tree predictions.

Use Cases: Widely used in various domains, including finance, healthcare, and natural language processing, for both classification and regression tasks.

2) AdaBoost (Adaptive Boosting):

Algorithm Type: Ensemble Learning (Boosting)

Strengths:

1) Combines weak learners into a strong learner.

2) Focuses on examples that previous models found difficult, improving overall performance.

3) Suitable for a wide range of classification problems.

How it Works: AdaBoost trains a sequence of weak classifiers (e.g., decision stumps) and assigns weights to misclassified data points. It iteratively trains new classifiers, giving more weight to previously misclassified samples. The final prediction is a weighted sum of the individual classifier predictions.

Use Cases: Face detection, text classification, and many binary classification problems where the dataset is imbalanced or contains noisy data.

3) Support Vector Classifier (SVC):

Algorithm Type: Discriminative

Strengths:

1) Effective in high-dimensional spaces.

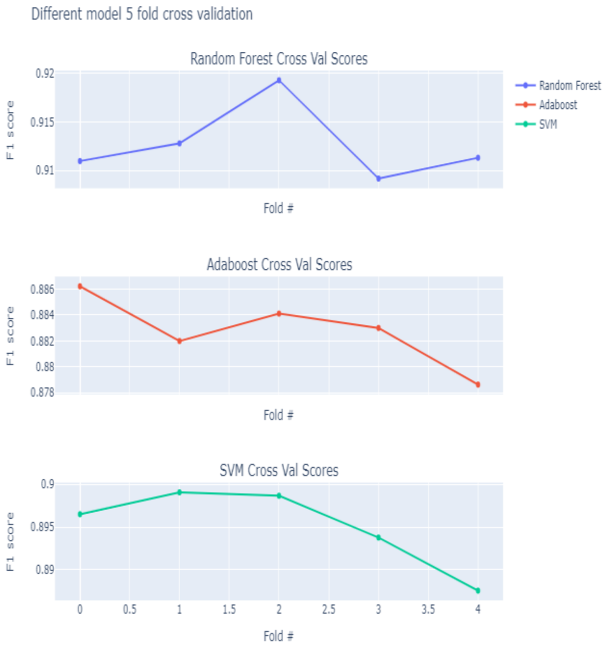
2) Good for binary classification and can be extended to multi-class problems.

3) Uses a subset of training points (support vectors) for decision boundary calculation, making it memory-efficient.

How it Works: SVC finds the optimal hyperplane that maximizes the margin between different classes. It works by mapping data points into a higher-dimensional space (kernel trick) and finding the hyperplane that best separates the classes. The margin is the distance between the hyperplane and the nearest data points.

Use Cases: Image classification, text classification, and various binary and multi-class classification problems where clear class separation is important.

The choice between these algorithms depends on several factors, including the nature of your data, the problem type (classification or regression), and the trade-offs you're willing to make in terms of computational complexity, interpretability, and predictive performance. Experimenting with different algorithms and assessing their performance on your specific dataset is often the best way to determine which one works best for your task.



8) F1 SCORE:

The F1 Score is a widely used performance metric in machine learning classification tasks that provides a balance between precision and recall. It is calculated as the harmonic mean of precision and recall and is particularly valuable when dealing with imbalanced datasets or when there are differing costs associated with false positives and false negatives. Precision measures the accuracy of positive predictions made by the model, while recall gauges its ability to identify all actual positive instances. The F1 Score combines these two metrics into a single value, ranging from 0 to 1, where higher values indicate better model performance. It serves as a comprehensive measure of a classifier's ability to make accurate positive predictions while capturing all relevant positive instances, making it a crucial tool for model evaluation and selection in real-world applications.

$$F1\ \mathrm{Score} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

The F1 Score is calculated as the harmonic mean of precision and recall and is defined as:

Here's a breakdown of the components used to calculate the F1 Score:

1) Precision (also known as Positive Predictive Value):

1) Precision measures the accuracy of positive predictions made by a classification model.

2) It is calculated as the ratio of true positive (TP) predictions to the total number of positive predictions (TP + false positives, FP).

3) Precision is a measure of how many of the positive predictions made by the model were actually correct.

$$Precision = \frac{TP}{TP+FP}$$

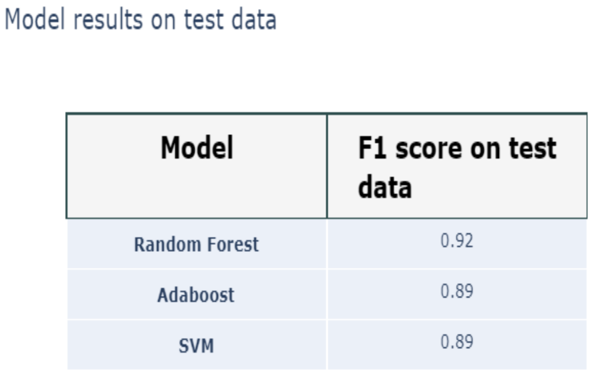
2) Recall (also known as Sensitivity or True Positive Rate):

1) Recall measures the ability of a classification model to correctly identify all relevant instances (true positives) out of all actual positive instances.

2) It is calculated as the ratio of true positive (TP) predictions to the total number of actual positive instances (TP + false negatives, FN).

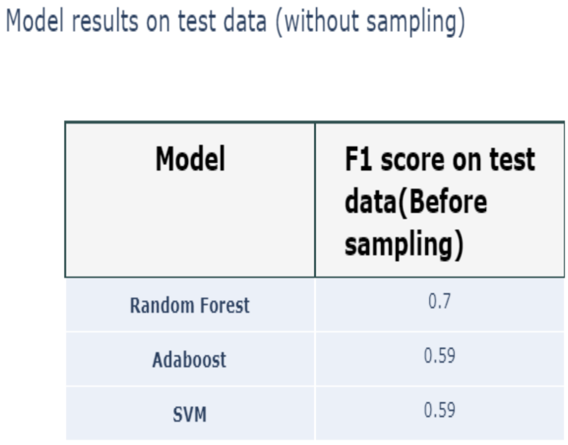
3) Recall is a measure of how well the model captures all positive instances.

$$Recall = \frac{TP}{TP+FN}$$



Model results on test data

Model	F1 score on test data
Random Forest	0.92
Adaboost	0.89
SVM	0.89



Model results on test data (without sampling)

Model	F1 score on test data(Before sampling)
Random Forest	0.7
Adaboost	0.59
SVM	0.59

9) CONFUSION MATRIX

A confusion matrix is a table or matrix used in machine learning and statistics to evaluate the performance of a classification algorithm, particularly in binary classification problems (problems with two classes or categories). It provides a detailed breakdown of the model's predictions and actual outcomes, allowing you to assess the model's accuracy and understand its strengths and weaknesses.

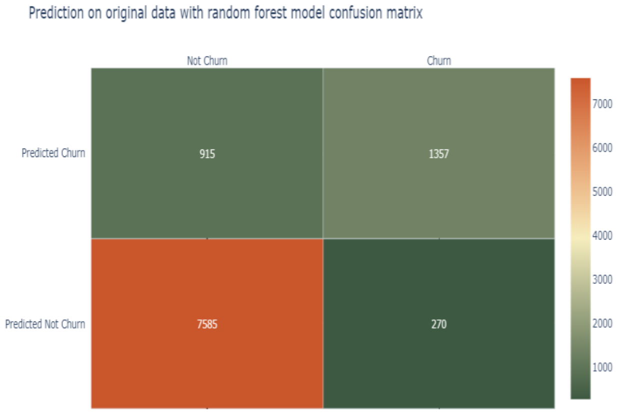
A confusion matrix consists of four components:

1) True Positives (TP): This represents the number of instances that were correctly predicted as belonging to the positive class (the class of interest).

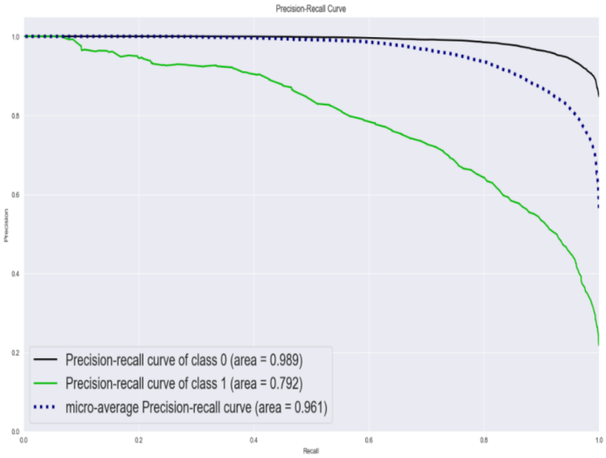
2) True Negatives (TN): This represents the number of instances that were correctly predicted as belonging to the negative class (the class other than the one of interest).

3) False Positives (FP): Also known as a Type I error, this represents the number of instances that were incorrectly predicted as belonging to the positive class when they actually belong to the negative class.

4) False Negatives (FN): Also known as a Type II error, this represents the number of instances that were incorrectly predicted as belonging to the negative class when they actually belong to the positive class.



Using this confusion matrix, we can calculate the accuracy of our model which comes out to be 89.02% using the true positive, false positive, true negative and false positive.



III. CONCLUSION

In conclusion, this research paper has provided a thorough examination of customer churn, customer segmentation, and the application of supervised machine-learning techniques for customer attrition prediction. The study's findings have significant implications for industries such as banking, e-commerce, telecommunications, and insurance, offering a benchmark for the implementation of customer churn prediction strategies.

Key takeaways from this research include the effectiveness of random forest and support vector machines as the top-performing training methods for customer attrition prediction. Additionally, the study underscores the importance of data preprocessing, correlation analysis, and principal component analysis (PCA) in preparing and analysing customer data.

Through K-Means clustering, the research has demonstrated the ability to identify distinct customer groups based on specific characteristics and behaviours, paving the way for personalized services and targeted marketing efforts. Furthermore, the evaluation of machine learning models using the F1 Score and confusion matrix highlights the importance of rigorous model assessment, especially in situations with imbalanced datasets.

While this study has made significant contributions to the field of customer segmentation and churn prediction, it also acknowledges its limitations and suggests avenues for future research. These include expanding data sources beyond Kaggle, investigating the most influential features for customer attrition, and developing practical applications to help businesses retain their customers.

Overall, this research paper provides a comprehensive framework and a rich set of methodologies for businesses to better understand customer behaviour and enhance their customer-centric strategies across diverse industries. The insights and techniques discussed here are valuable assets in the pursuit of data-driven decision-making and customer relationship management.

REFERENCES

[1] https://github.com/tamjid-ahsan/capstone_customer_churn

[2] https://www.researchgate.net/publication/368911804_Customer_Churn_Prediction_in_the_Banking_Sector_Using_Machine_Learning-Based_Classification_Models

[3] https://www.scirp.org/journal/paperinformation.aspx?paperid=115949#ref5

[4] Guliyev, H., & Tatoğlu, F. Y. (2021). Customer churn analysis in banking sector: Evidence from explainable ma-chine learning models. Journal of Applied Microeconometrics, 1(2), 85–99. https://doi.org/10.53753/jame.1.2.03

[6] Elyusufi, Y., & Ait Kbir, M. (2022). Churn prediction analysis by combining machine learning algorithms and best features exploration. International Journal of Advanced Computer Science and Applications, 13(7), 615–622. https://doi.org/10.14569/IJACSA.2022.0130773

[7] Sivasankar, E., & Vijaya, J. (2017). Customer segmentation by various clustering approaches and building an effective hybrid learning system on churn prediction dataset. In H. Behera, & D. Mohapatra (Eds.), Compu-tational intelligence in data mining (pp. 181–191). Springer. https://doi.org/10.1007/978-981-10-3874-7_18

[8] Rahman, M., & Kumar, V. (2020, November). Machine learning based customer churn prediction in banking. Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 1196–1201. https://doi.org/10.1109/ICECA49313.2020.9297529

[9] Olaniyi, A. S., Olaolu, A. M., Jimada-Ojuolape, B., & Kayode, S. Y. (2020). Customer churn prediction in bank-ing industry using k-means and support vector machine algorithms. International Journal of Multidisciplinary Sciences and Advanced Technology, 1(1), 48–54.

[10] Zhang, T., Moro, S., & Ramos, R. F. (2022). A data-driven approach to improve customer churn prediction based on telecom customer segmentation. Future Internet, 14(3), 94. https://doi.org/10.3390/fi14030094