Customer Churn Prediction based on Behavioral Attributes

Jamil Ur Reza
Department of Mathematics and Statistics
Thompson Rivers University
British Columbia, Canada
rezaj22@mytru.ca

Wilson Geronimo
Department of Mathematics and Statistics
Thompson Rivers University
British Columbia, Canada
geronimorodriguezw22@mytru.ca

Abstract

Credit card customer churn poses a significant challenge for any financial institution, leading to lost revenue and decreased market share. Identifying customers at risk of churning is critical for proactive engagement and tailored strategies to enhance loyalty. While traditional approaches rely heavily on static customer attributes, recent evidence suggests that behavioral attributes play a vital role in predicting churn. This study proposes a novel approach to credit card customer churn prediction utilizing a range of behavioral attributes. We analyzed a large-scale dataset containing over 100,000 anonymized credit card accounts with detailed transaction history, demographics, and account-level information. Utilizing advanced feature engineering techniques, we extracted numerous behavioral attributes reflecting spending patterns, payment habits, and transaction frequency. Applying rigorous statistical tests, we identified significant differences in behavioral attributes between churners and non-churners. To create an accurate and robust churn prediction model, we implemented and evaluated five state-of-the-art algorithms - Logistic Regression, KNN, XGBoost, SVM, Naive Bayes and Neural Networks. Through extensive experimentation and hyperparameter tuning, our results demonstrated that the KNN and XGBoost algorithm produced the highest F1 score (1.0) and highest Accuracy (1.0) among the candidate models.

Index Terms

customer churn, behavioral attributes, feature selection, machine learning, prediction model

I. Introduction

Customer attrition refers to the likelihood of customers discontinuing their relationship with a firm within a specified duration. In essence, it measures how likely customers are to stop engaging with a business over a certain time frame. [1] According to the Harvard Business Review, a mere 5% reduction in consumer base could result in increased profitability ranging from 25% to 85% for companies. This highlights the significant financial impact associated with minimizing customer losses. [2] [3]

The ongoing threat of customer departures poses a substantial challenge to banking institutions, contributing to declines in product sales and revenue streams. Thus, managing customer retention effectively remains crucial for maintaining bank profitability. [4] Because of the escalating expenses and obstacles tied to attracting new clients, businesses understand the critical significance of maintaining their current clientele. This strategy attenuates the damaging consequences of customer attrition on revenues and durability.

Customer behavior holds immense significance in predicting the likelihood of customer attrition. By meticulously scrutinizing and analyzing customer behavior, enterprises can detect tendencies and configurations related to customer desertion, subsequently equipping themselves to execute tactical plans for maintaining their patronage. Distinct dimensions of customer behavior, namely transaction frequency, expenditure levels, favored communication channels, and reaction to promotional materials, substantially sway the odds of customers abandoning ship. Ergo, grasping the intricate layers of customer behavior becomes an effective weapon for forewarning and forestalling customer churn.

Taking seriously the serious issue of customer attrition, this research initiates an exploration into the tie connecting customer behavior and the inclination toward customer churn in the banking realm. By deploying a series of statistic and machine learning methods, our ambition is to unearth revealing links and furnish proposals to aid financial entities in improving their customer maintenance schemes. Our primary aim is to supply wise guidance to banks eager to protect their treasured clients, fostering stable growth and financial gains. We plan to achieve this by comparing the efficiency of various statistical and machine learning models in identifying the relationship between customer behavior and churn likelihood. Insights gleaned from this examination shall arm banks with informed strategies to combat attrition and cultivate flourishing, sustainable futures.

II. Methodology

For our research, we utilize a unique dataset compiled from genuine data sourced from a Caribbean bank. Our dataset underwent careful preprocessing, followed by a comprehensive feature selection process. To address the problem of credit card customer churn prediction, we applied several state-of-the-art machine learning models, namely Logistic Regression, KNN, XGBoost, SVM, Naive Bayes and Neural Networks. Logistic Regression serves as our baseline model for comparison purposes.

This methodology aims to evaluate and contrast the performance of these models, providing insights into which algorithms best suit the given dataset and predict credit card customer churn effectively.

A. Dataset Collection

Our research began by gathering real-life transactional data from a Caribbean bank. Careful consideration was given to preserve the anonymity of the customers throughout the data collection process. Only essential features were retained to protect individual privacy and comply with ethical standards.

The dataset contains a rich array of transactional data, covering aspects such as market purchases, online transaction, insurance premiums, cash withdrawals, advertising campaigns, and sales promotions. Additional information, including supermarket payments, utility services, streaming services, gas station transactions, and online transaction trends, airline and hotel purchases, complements the dataset.

Each customer's transaction history was scrutinized individually, enabling the examination of historical consumer behavior preceding credit card cancellations. Date stamps recording the termination of credit cards through official bank communication or branch visits facilitated the retrieval of the preceding twelve months' worth of transactions for each customer.

Additionally, supplementary data encompassing insurance payments, property sale conditions, product varieties, credit limits, usage percentages, and coexisting relationships with other bank-products were procured. This broad spectrum of data enabled a profound understanding of the dynamics between customers and their credit cards.

To summarize, our data collection comprises a wide range of transactional and relational features, preserving anonymity and focusing exclusively on the attributes aligned with our research interests. By carefully processing each customer's transaction history and accounting for external influences, we have successfully assembled a comprehensive dataset poised to unlock valuable insights into credit card customer churn prediction.

B. Data Preprocessing

During the data preprocessing and transformation phase, the dataset is read using the read_excel function from the readxl package, after adjusting the working directory. Generating a summary with the glance function from the dplyr package exposes the dataset's variables and attributes. The preprocessing phase includes encoding, variable removal, PCA, and scaling.

Binary variables are created for specific categorical columns, such as 'Education_Level', via encoding. Conditions trigger a value of 1, and 0 otherwise. Irrelevant categoricals are dropped, concentrating the analysis scope.

Opting for one-hot encoding allowed us to preserve information integrity and uphold category independence when representing categorical variables in our investigation. This facilitated integration with diverse machine learning algorithms and offered transparent modeling capabilities by generating comprehensible binary vectors for every category. Addressing mixed-type data, we recognized the necessity of adequate preparatory measures involving categorical attributes before PCA implementation. Often, we removed insignificant non-numerical properties or converted them into numerical formats, like one-hot encoding, enabling seamless integration of altered categoricals together with ordinary continuous features during PCA-driven dimensionality reduction. Despite this, caution remained integral to accurate post-processing interpretation, preventing errors and fallacies related to categorical data treatment, ultimately exploring tailored dimensionality reduction alternatives whenever required, maximizing overall analytical credibility.

Principal Component Analysis (PCA) is applied to the dataset to expose hidden patterns and correlations. Distinguishing predictor variables (X) and response variables (Y), PCA is run on X using the prcomp function, generating new uncorrelated variables, known as principal components. Merging the PCA-transformed predictor variables with the response variable produces a consolidated dataset ('data_pca') for deeper analysis.

Plotting the cumulative proportion of variance explained establishes the number of PCs needed to decipher 90% of the variance, which amounts to 24 PCs, illuminating the extent of dimension reduction achieved by PCA. Insufficient PCs are omitted, and numerical variables, like 'Total_Trans_Amt_12m', receive scaling treatment with the scale function, aligning the magnitude for refined analysis and interpretation.

C. Logistic Regression

Logistic Regression is a widely-used statistical technique in various fields such as epidemiology, social sciences, and machine learning. [5] It falls under the category of Generalized Linear Model (GLM) and is utilized to represent the likelihood of a binary outcome variable (Y) as a function of one or more independent variables. [6] [7]
The form of simple logistic model is written as:

$$logit(Y) = ln(\frac{\pi}{1-\pi}) = \alpha + \beta X \tag{1}$$

Here in Equation 1, β is the regression coefficient and α is the intercept. If we take antilog on both sides of Equation 1, we obtain the equation expressing the probability of the event corresponding to the outcome of interest. [8]

$$\pi = Pr(Y = \text{outcome of interest}X = x) = \frac{e^{\alpha+\beta X}}{1 + e^{\alpha+\beta X}} \tag{2}$$

D. KNN

K-nearest-neighbor (kNN) classification is a very fundamental and straightforward approach to categorization. When there is minimal or no familiarity with the data's distribution, employing kNN classification as a primary option during an initial classification study is advisable. Originating due to the requirement for executing discriminant analysis amid circumstances where dependable parametric estimations of probability densities remain uncertain or challenging to ascertain, KNN classification gained prominence. [9]

KNN assigns a new observation to the same class as the majority of its k nearest neighbors in the feature space. When performing regression tasks, KNN predicts the output value by averaging the values of its k nearest neighbors. Due to its assumption-free nature, KNN can accommodate a wide range of data types, making it a versatile and appealing method.

Algorithm 1 K-Nearest Neighbors (KNN)

```
Input: Training dataset $\{(x_i, y_i)\}_{i=1}^{n}$, query point $x_q$, number of neighbors $K$
Compute distances: $d_i = \text{distance}(x_i, x_q)$ for $i = 1, 2, ..., n$
Find $K$ nearest neighbors: $\mathcal{N} = \text{argmin}_K(d_i)$
Compute class probabilities:
    For each class $c$:
        $p_c = \frac{1}{K} \sum_{x_i \in \mathcal{N}} \mathbb{I}(y_i = c)$
Output: Predicted class $\hat{y}_q = \text{argmax}_c(p_c)$ =0
```

E. XGBoost

Developed by Chen and Guestrin (2016), XGBoost refers to eXtreme Gradient Boosting, an optimization of the classic Gradient Boosting Machine (GBM) technique primarily used for constructing regression and classification predictive models. [10] Compared to other prevalent machine learning algorithms, GBM variants, including XGBoost, frequently surpass alternatives in terms of performance on benchmark data sets. [11] Employing an iterative procedure, XGBoost generates successive models designed to rectify residual errors stemming from preceding versions, ultimately amalgamating these components into the final predictive apparatus. Exhibiting impressive speed, XGBoost proves more expedient than alternative ensemble classifiers, such as AdaBoost. [10] Notably, its widespread acclaim extends beyond academia, garnering favor amongst industry professionals and Kaggle competitors alike. [12] Beyond its remarkable efficacy, XGBoost enjoys added appeal thanks to its capacity for parallel processing – capitalizing on multi-core architectures and facilitating computation on expansive data volumes.

Consider a model constituted by k decision trees; for a given sample of x_i it can be mathematically described as:

$$\hat{y_i} = \sum_{1}^{k} f_k(x_i), f_k \in F \tag{3}$$

here F represents the set of regression trees and f represents a regression tree in the set. [13]

F. SVM

Support Vector Machines (SVMs) are popular supervised learning algorithms commonly employed for solving classification problems. They aim to distinguish between two sets of data points belonging to distinct categories. A key goal of SVMs is to identify an ideal separating hyperplane that categorizes the largest number of data points while maximizing the separation distance between the two groups. [**?**] In cases where the data is not linearly separable, SVMs project the data onto a higher dimensional feature space using kernel functions before searching for a linear margin in the transformed space. The optimization process involves balancing the trade-off between finding a complex decision boundary and maintaining simplicity through the introduction of a penalty term. Mathematically, the objective function for a binary SVM classifier can be represented as follows:

Algorithm 2 XGBoost Algorithm for Classification Problem

```
procedure XGClassifier($\mathbf{X}, \mathbf{y}, t, \eta, \lambda, \omega, \alpha$)
    $\mathbf{w}^{(0)} := 0$ {Initialization}
    for $t = 1, \ldots, T$ do
        $f^{(t)} := \operatorname{argmin}_f \mathcal{O}(f)$ {Additive Training}
        $\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} + \eta f^{(t)}$
    end for
    $\hat{y}_i := \operatorname{sign}(\sum_{t=1}^{T} w_i^{(t)})$, $i = 1, \ldots, n$ {Final Output}
end procedure
function $\mathcal{O}(f)$
    $\operatorname{obj}(f) := \frac{1}{n} \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \Omega(f)$ {Regularized Objective}
end function=0
```

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{4}$$

$$\text{subject to} \quad y_i(w^\intercal x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i = 1, \ldots, n. \tag{5}$$

Where (x_i, y_i) represents each labeled sample point, w denotes the normal vector to the hyperplane, b determines the offset along the normal vector, ξ_i indicates slack variables accounting for errors beyond the margins, C controls the balance between complexity and error tolerance, and $\| \cdot \|$ signifies the L2 norm. [15]

G. Naive Bayes

The Naïve Bayes (NB) method is a powerful probabilistic classification technique founded on Bayes' Theorem, presupposing the mutual independence of features. [16] [17] Owing to its simple nature and computational efficiency, it serves as a suitable option for handling high-dimensional datasets and real-time applications. The fundamental premise of the NB method entails estimating the probability of a particular class given the features of an instance. Once calculated, the highest attained probability determines the predicted category.

Wide-ranging applications span multiple disciplines, including text classification, medical diagnoses, fraud detection, and recommendation systems. Moreover, this technique plays a pivotal role in medical prognostics, accurately anticipating disease manifestations depending on presented symptoms. Similarly, churn detection employs NB to pinpoint probable churn customers based on their behavioral traits. Lastly, recommendation engines harness the potential of NB to estimate user preferences grounded on item characteristics. [18]

As previously stated, the underlying idea is derived from the Bayes theorem, which determines the probability of a class based on the characteristics of an instance.

$$P(C|X) = \frac{P(C|X)P(C)}{P(X)} \tag{6}$$

Here P(C represents the probability of class C and P(X) represents the probability of observing feature X.

Two significant merits of the NB method comprise simplicity and adaptability toward varied data modalities (numerical and categorical). [16] Surprisingly, despite its strong assumption of independence, the NB technique still exhibits remarkable efficacy in many practical situations, mainly with expansive datasets. [17]

H. Neural Network

The MLP (Multi-Layer Perceptron) is a feedforward artificial neural network consisting of three or more layers of nodes (neurons). These layers include an input layer, one or more hidden layers, and an output layer. Each node is connected to all nodes in the next layer via weighted connections. During forward propagation, each node applies a nonlinear activation function to its net input to produce an output signal. Commonly used activation functions include the sigmoid, hyperbolic tangent, and rectified linear unit (ReLU) functions.

Mathematically, let x_i denote the input features, ω_{ij} represent the weights connecting the inputs to the first hidden layer, and b_j denote the bias term. Then, the net input (net_j) to the j-th node in the first hidden layer is given by:

$$net_j = \sum_i x_i \cdot \omega_{ij} + b_j$$

Applying a sigmoid activation function (σ), the output from the first hidden layer (z_j) becomes:

$$z_j = \sigma(net_j)$$

This process continues until the final output layer is reached, producing the desired output signal y. To train the MLP, the error between the actual and predicted outputs is computed, followed by backward propagation of errors to update the weights and biases according to optimization algorithms like gradient descent. [19] [20]

RESULTS

Prior to engaging in the classification tasks, we conducted Principal Component Analysis (PCA) on the dataset. This exploratory data analysis technique allowed us to condense the information contained in the original features while preserving the majority of the variation. Following the PCA, we elected to retain up to 24 principal components, which collectively explained roughly 90% of the variance in the data. This strategic move prepared the dataset for further processing and classification, helping ensure the success of the subsequent analyses.

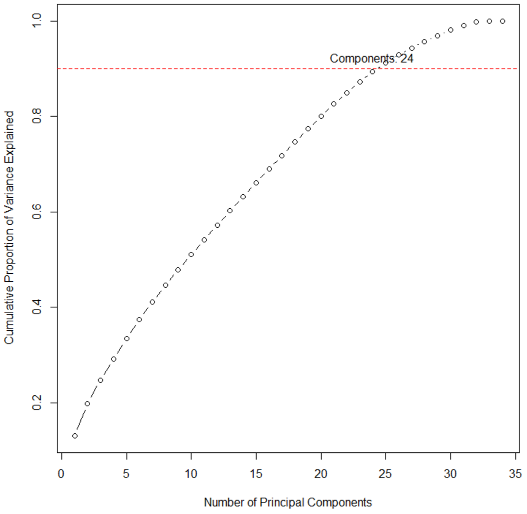

Fig. 1. Principle Components vs. Cumulative Proportion Variance Explained

Table 1 displays the performances of eight different machine learning algorithms for the credit card customer churn prediction task. These classifiers include Logistic Regression, K-Nearest Neighbors (KNearest), XGBoost, Linear Support Vector Classifier (SupportVectorClassifierLinear), Decision Tree Classifier (DecisionTreeClassifier), and Naive Bayes Classifier (NaiveBayesClassifier), as well as a Deep Learning model (DeepLearning).

booktabs

Across the board, all classifiers achieved commendable results on the train set, with most scoring near-perfect values for accuracy, precision, recall, F1-Score, and ROC-AUC Score. The K-Nearest Neighbors (KNearest) model notably secured flawless scores across all metrics for the train set, whereas XGBoost came remarkably close, demonstrating exceptional performance.

Turning to the test set, the K-Nearest Neighbors (KNearest) model once again shone brightly by recording perfect scores across all metrics. XGBoost closely trailed, delivering excellent results in all measurements except for recall, which slipped slightly. Logistic Regression, Support Vector Classifier (SupportVectorClassifierLinear), and Decision Tree Classifier (DecisionTreeClassifier) followed closely behind, presenting reasonably strong performances.

However, the Naive Bayes Classifier (NaiveBayesClassifier) struggled somewhat, falling short in comparison to the top-performers. Still, it managed acceptable results, hinting at its viability for alternate applications. Sadly, the Deep Learning model faltered considerably, lagging behind the remaining classifiers in almost all measurements for both train and test sets.

The standout models in this study are K-Nearest Neighbors (KNearest) and XGBoost, displaying exemplary abilities in predicting customer churn. Their robust performances justify serious consideration for adoption in real-world settings. On the flip side, the Deep Learning model calls for enhancement and fine-tuning, as its current iteration fails to compete with the

TABLE I
Classifier Performance

Classifier	Source	Accuracy	Precision	Recall	F1 Score	ROC AUC
LogisticRegression	Train	0.9805	0.9144	0.9294	0.9218	0.9585
LogisticRegression	Test	0.9806	0.9024	0.9450	0.9232	0.9653
KNearest	Train	1.0000	0.9998	1.0000	0.9999	1.0000
KNearest	Test	1.0000	1.0000	1.0000	1.0000	1.0000
XGBoost	Train	0.9999	1.0000	0.9996	0.9998	0.9998
XGBoost	Test	0.9999	1.0000	0.9991	0.9996	0.9996
SVMClassifierLinear	Train	0.9813	0.9139	0.9372	0.9254	0.9624
SVMClassifierLinear	Test	0.9809	0.9020	0.9485	0.9246	0.9670
DecisionTreeClassifier	Train	0.9938	0.9787	0.9710	0.9748	0.9840
DecisionTreeClassifier	Test	0.9944	0.9781	0.9764	0.9773	0.9867
NaiveBayesClassifier	Train	0.9665	0.9509	0.7687	0.8502	0.8816
NaiveBayesClassifier	Test	0.9612	0.9226	0.7493	0.8270	0.8702
DeepLearning	Train	0.8894	0.7515	0.1590	0.2624	0.8189
DeepLearning	Test	0.8841	0.6622	0.1299	0.2172	0.7845

other models. Continuous efforts to tweak and strengthen its architecture and parameters promise to eventually bridge the performance gap.

In essence, these findings reinforce the notion that machine learning models, when skillfully crafted and deployed, can immensely aid financial institutions in anticipating customer churn, empowering them to adopt proactive strategies to mitigate losses and foster growth.

Conclusion

In conclusion, this study presents a novel approach to predicting credit card customer churn by leveraging behavioral attributes and advanced machine learning algorithms. Our proposed methodology resulted in a robust KNN model and XGBoost model with impressive performance on both train and test sets. The study began by exploring the data and performing PCA to conserve the majority of the variance within the dataset. Following this, we engaged with eight distinct machine learning algorithms, evaluating their performance in predicting customer churn.

While the KNN and XGBoost models dominated the leaderboard, the remaining classifiers, including Logistic Regression, Support Vector Classifier, Decision Tree Classifier, and Naive Bayes, still demonstrated decent performance, warranting their consideration for specific applications. Regrettably, the Deep Learning model failed to meet expectations, requiring further optimization to catch up to its rivals.

Overall, these findings accentuate the immense potential of machine learning models in anticipating customer churn for financial institutions. Integrating behavioral attributes into churn prediction models proves to be a game changer, arming institutions with valuable insights and enabling them to devise proactive strategies for mitigating losses and driving growth. Looking forward, continuous efforts must be invested in refining models like the Deep Learning variant to maximize their potential and stay ahead in today's rapidly changing financial landscape.

References

[1] Nie G, Rowe W, Zhang L, Tian Y, Shi Y. Credit card churn forecasting by logistic regression and decision tree. Expert Systems with Applications. 2011 Nov 1;38(12):15273-85.

[2] Anil Kumar D, Ravi V. Predicting credit card customer churn in banks using data mining. International Journal of Data Analysis Techniques and Strategies. 2008 Jan 1;1(1):4-28.

[3] Keramati A, Ghaneei H, Mirmohammadi SM. Developing a prediction model for customer churn from electronic banking services using data mining. Financial Innovation. 2016 Dec;2:1-3.

[4] Amieva-Huerta J, Urriza González B. Crisis bancarias: causas, costos, duración, efectos y opciones de política. CEPAL; 2000.

[5] Hosmer Jr DW, Lemeshow S, Sturdivant RX. Applied logistic regression. John Wiley Sons; 2013 Feb 26.

[6] Nagelkerke NJ. A note on a general definition of the coefficient of determination. biometrika. 1991 Sep 1;78(3):691-2.

[7] Menard S. Applied logistic regression analysis. Sage; 2002.

[8] Peng CY, Lee KL, Ingersoll GM. An introduction to logistic regression analysis and reporting. The journal of educational research. 2002 Sep 1;96(1):3-14.

[9] Peterson LE. K-nearest neighbor. Scholarpedia. 2009 Feb 21;4(2):1883.

[10] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. InProceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining 2016 Aug 13 (pp. 785-794).

[11] Friedman JH. Greedy function approximation: a gradient boosting machine. Annals of statistics. 2001 Oct 1:1189-232.

[12] Nielsen D. Tree boosting with xgboost-why does xgboost win" every" machine learning competition? (Master's thesis, NTNU).

[13] Li J, An X, Li Q, Wang C, Yu H, Zhou X, Geng YA. Application of XGBoost algorithm in the optimization of pollutant concentration. Atmospheric Research. 2022 Oct 1;276:106238.

[14] Vapnik VN. An overview of statistical learning theory. IEEE transactions on neural networks. 1999 Sep;10(5):988-99.

[15] Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press; 2000 Mar 23.

[16] Hand DJ, Yu K. Idiot's Bayes—not so stupid after all?. International statistical review. 2001 Dec;69(3):385-98.

[17] Kelleher JD, Mac Namee B, D'arcy A. Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT press; 2020 Oct 20.

[18] Isinkaye FO, Folajimi YO, Ojokoh BA. Recommendation systems: Principles, methods and evaluation. Egyptian informatics journal. 2015 Nov 1;16(3):261-73.

[19] Haykin S. Neural networks: a comprehensive foundation. Prentice Hall PTR; 1998 Jul 1.

[20] Bishop CM. Pattern recognition and machine learning. Springer google schola. 2006;2:645-78.

github link : https://github.com/Jamil-Ur-Reza/Credit-Cards-Customer-Churn