Predictive Modeling for Customer Churn in Savings Accounts

Shivam Sekhri

Humber Institute of Technology and Advance Learning, M8W 1M8, Canada

Shivam Sekhri (e-mail: N01574357@humbermail.ca).

ABSTRACT   
The study helps the Canadian National Bank to identify customers efficiently who could likely churn in the future from their savings account services. This is achieved by the development of the robust machine learning model. The motivation behind this study is to train the best machine learning model for the bank that could assist them in predicting near-precise target customers who are likely to leave their services. The contribution of my study will make it easier for the Canadian National Bank to identify and target at-risk customers through measures like personalized emails and offers, aiming to increase retention and promote continued use of savings accounts. Once integrated in the bank’s system, my predictive model further aids in cost savings by targeting only necessary customers, avoiding unnecessary outreach or false positives. Overall, the study offers the strong strategic solution to the Canadian National Bank by giving them ability to make informed business decision and help in aligning them with the imperative of customer relationship building in the banking industry.

INDEX TERMS customer relationship, efficiently, false positives, informed business decision, machine learning model, predictive model, retention, robust.

1.INTRODUCTION

Machine learning models are incredibly useful, particularly in the world of prediction. Their ability to analyze patterns and learn from data enables them to make accurate predictions or forecasts. Whether it's predicting customer behavior, stock market trends, these models excel at recognizing patterns and making informed predictions based on the patterns they've learned. This predictive capability is invaluable across various industries, ultimately assisting in better decision-making, risk assessment, and resource optimization. In essence, machine learning models act as powerful tools for foreseeing outcomes and trends, contributing significantly to data-driven decision processes.

In the context of my study, the dataset provided by the bank became the cornerstone of my exploration. I explored and cleaned the dataset, laying the groundwork for my analysis. Distinct models were constructed, and through rigorous validation, I identified the random forest model as the most effective in yielding optimal results on my validation data.

2.MACHINE LEARNING PIPELINE

The machine learning pipeline is the backbone of transforming raw data into actionable insights that guide us from data extraction to model deployment. This framework begins with data extraction, where information is sourced from various databases.

The data undergoes a thorough exploration phase of patterns, outliers, and trends that inform subsequent decision-making. In the training phase, machine learning algorithms study the data, learning patterns and relationships. Validation follows to assess the model's performance on unseen data, fine-tuning it for optimal predictive capability.

Once validated, the model is prepared for deployment, making it ready to process new data and provide real-time predictions. This pipeline, encompassing data exploration, validation, and model training, ensures a methodical and effective approach in harnessing the power of machine learning for informed decision-making.

3. DATA EXPLORATION AND PREPERATION

3.1 DATA COLLECTION

The dataset for this project has been pulled from GitHub. The dataset has 28382 records and 21 variables. In this project, my target is to calculate the propensity of customers who could churn down the road. The data can be found [here](https://github.com/tusharmishra288/Bank-Customer-Churn-Prediction#data-dictionary).

3.2 DATA PREPARTION  
In machine learning, data preparation involves cleaning and transforming raw data to ensure it's suitable for training models. Tasks include handling missing values, addressing outliers, and encoding categorical variables, ensuring the dataset is optimized for effective model learning and predictive accuracy.

3.2.1 MISSING VALUES

Missing values were found in the data set while exploring it. It is important to deal with these values appropriately to conduct further analysis and develop good machine learning models. There was total 7094 missing values observed in the data set including 525 for Gender, 2463 for dependents, 80 for occupation, 803 for cities and 3223 for days\_since\_last\_transaction. There are two approaches while dealing with missing values wither by dropping them or imputing values.

In this project, I chose the second approach because I would lose part of the data set by dropping values and my machine learning model could be biased, which further would lead to wrong insights and incorrect business decisions.

I built a missing values heat map to better visualize the level of missingness in the data. This way I was able to easily analyze the overall impact of missing values because this heat map shows missing values with respect to the rows in which they were present.

A red and black text on a white background

Description automatically generated

The heat map displayed above has red lines indicating the individual missing data values.

Now I will use the histograms to analyze the distribution of values within that variable to better understand how to impute the missing values in these five variables.

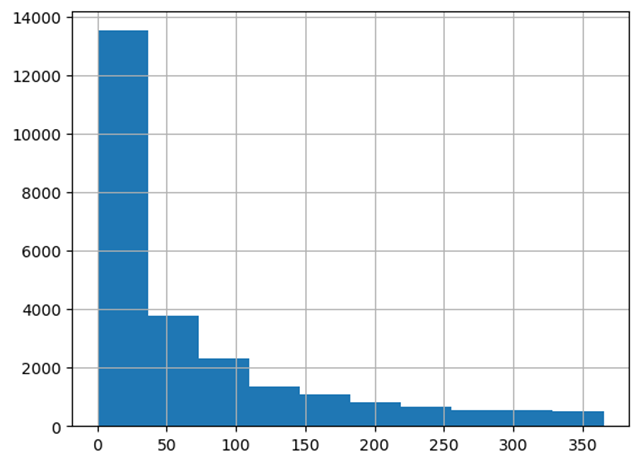
a. **Dependents**

A graph with a bar

Description automatically generated

In the dependents histogram displayed above, most customers have zero dependents, so the missing value in this column is imputed with zero, which is also mode for this variable. This will maintain the proportion of the values in the “Dependents” column.

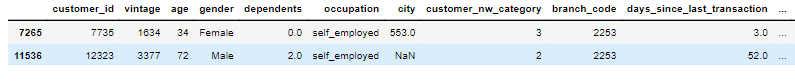
b. **days\_since\_last\_transaction**

****

The histogram for days\_since\_last\_transaction is highly positively skewed, which suggests that the mean of the variable will be distorted because of the outliers and the mode of the column would be the better value to impute the missing values in this column.

c. **City**

It was observed that I can use branch code to impute city values. In the above snapshot, I can see that both rows have the same branch code. While one row is missing city, the other one has the city to which the branch code belongs. I created a unique city and branch combination list from the data set and imputed city values accordingly.



c. **Gender**

A graph of a number of blue rectangular bars

Description automatically generated with medium confidence

The Gender bar chart displayed above depicts the frequency of male and female in the data set. It was observed that the male to female ratio in the data set is 3:2. So I created a list with 3 males and 2 females and randomly picked values from the list to impute the missing values in the gender column. This approach helped us impute values without distorting the distribution of males and females in the data set.

d. **Occupation**

A graph with blue bars

Description automatically generated

The occupation bar chart above depicts the frequency of different types of occupation in the data set. It was observed that the ratio among these variables 8:3:1:1:0. So I created a list with 8 self-employed, 3 salaried, 1 retired and 1 student values. I used the same approach that I used in imputing null values for the gender variable without compromising the distribution of values in the data set.

3.2.2. DUPLICATE AND LOW VARIATION DATA

To analyze the presence of duplicate information I created a correlation heatmap for the data set, which is shown below.

A blue and white squares with black text

Description automatically generated

In the above heatmap, I can see that many columns like current\_balance and current\_month\_balance have a positive correlation of 1, which indicates that they inherently carry the same information. So, columns like these that have correlation 1 or -1 and are on the same scale can be dropped. After dropping one of the pairs showing correlation +1, I got the heatmap shown below.

A graph with numbers and letters

Description automatically generated with medium confidence

2.2.3 INCORRECT AND IRRELEVANT DATA

I created a unique list for the dependent’s variable, wherein I observed some values, which were incorrect. In the above snapshot, I can see those values of dependents 25, 32, 36, 50, 52 seems unreal in the realistic scenario. I will not drop these values because I don’t want to lose the other information in the rows, which could further contribute to my analysis. I would rather impute these values with the mode for the column i.e., 0.

The “customer ID” column is irrelevant to my purpose as it is an arbitrary value, which bank uses to identify each individual customer. But the customer ID column does not provide any insights and will not add any value to my machine learning models. Thus, the customer ID column was dropped.

The city and branch\_code variables could be dropped because they have 1604 and 3154 unique values respectively and it is not feasible to create dummy variables for these variables. Furthermore, they might not be important for my purpose as customer churn depends more upon the customer profile than the branch or city and bank policies and products are usually same across.

2.2.4 CATEGORICAL DATA

Categorical data is any data that is in the form of text, class or category. In my data set, occupation, customer\_nw\_category and gender are categorical variables. I will convert these variables into dummy variables so that they can be fit into different machine learning models that I will train going forward. I will conduct this step before training specific models as some models do not take categorical variables and require numerical continuous data.

***2.2.5 OUTLIERS***

A graph of a graph

Description automatically generated with medium confidence

A graph of a graph

Description automatically generated with medium confidence

I analyzed all variables for outliers by creating side-by-side boxplots as shown above. For all variables, the number of outliers seemed to be realistic values and not incorrect entries. Some variables had extreme outliers (for ex. current\_balance value of 5905904.03), which did not contribute to distorting my data much.

***2.2.5. VALIDATION SPLIT***

I will perform validation split on my data set having 28,382 rows and I split them into the training and validation data. The training data will be used to train machine learning models and the validation data will be used to check the accuracy of the trained models. I use 70% data as the training data as this provides enough data to model to accurately assess the relationships between variables in the data set and having 30% validation data allows us to see how concrete the performance of the model is.

(1)

3. MACHINE LEARNING MODEL DEVELOPMENT

3.1. CHOICE OF MODEL

I will use the following models to predict the customer churn rate and deploy the best model with the minimal error rate:

1. Classification trees
2. K-nearest neighbors
3. Linear Regression
4. Random Forest

***3.1.1 CLASSIFICATION TREES***

Classification trees are a type of decision trees in machine learning that recursively partitions data based on feature attributes, creating a hierarchical structure to predict categorical outcomes for given inputs. They are widely used for classification tasks, providing interpretable and effective models.

***Steps followed:***

To build a classification tree I imported the built-in model Decision Tree Classifier from tree module in scikit-learn library. For this model, dummy variables were created out of ‘occupation’, ‘gender’ and customer\_nw\_category columns as this this function does not take string data for input. It requires several parameters including random\_state which was set to 1 and controls the randomness of the bootstrapping of samples, max\_depth which was set to 3 and regulates how deep the trees are built, min\_impurity\_decrease which was set to 0 and controls if a split should made in regard to impurity that is reduced by it and lastly min\_samples\_split that was set to 10 and controls the minimum number of sample to be present in the resulting subsets of data for a split to be made.

***Accuracy:***

The model provided an accuracy of .8513 on the training data and .8453 on the validation data. A snapshot which shows the confusion matrix and accuracies for both training and validation data is shown below.

A screenshot of a computer

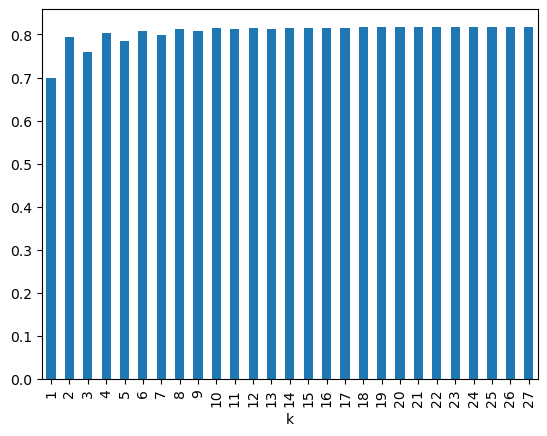
Description automatically generated

***3.1.2 K-NEAREST NEIGHBORS***

K Nearest Neighbors (KNN) is a non-parametric and instance-based learning algorithm. In KNN, the classification or regression of a new data point is determined by the majority class or average value of its k nearest neighbors in the feature space. The choice of k is a critical parameter, influencing the model's sensitivity to local patterns. KNN is robust to noisy data, and easy to understand.

***Steps followed:*** To build a KNN model module neighbor was imported from scikit-learn library and the function K Neighbors Classifier was used. The processed training data with all numeric variables was passed to train this model. This function requires an input parameter named n\_neighbors which refers to the number of similar records that should be considered before classifying a record. Values from 1 to 27 were passed for this parameter.

***Accuracy:***

The model accuracy for different values of n\_neighbors are shown in the bar chart below. The model gave and accuracy of .8147 for the value of 10 for the n\_neighbors parameter, this can be seen as the optimal values for this parameter as larger values would not allow the model to capture the local properties of the record.

***3.1.3 LOGISTIC REGRESSION***

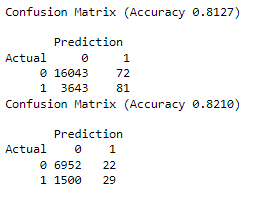
Linear regression is a statistical method used for modeling the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The goal is to find the best-fit line that minimizes the sum of squared differences between predicted and actual values. Linear regression quantifies the linear relationship between variables, making it a fundamental tool in predictive modeling and understanding the correlation between different factors.

***Steps followed:***

To train this model the data needed to be preprocessed again as this model requires that I drop the first column that is obtained while I build dummy variables, or I would encounter multicollinearity problems. Logistic regression model was trained using Logistic Regression function of module linear model from scikit-learn library. This model takes three parameters to train. One is penalty, which was set to l2, this specifies the type of regularization to be used for the objective function, this helps in preventing a model from overfitting. The second parameter is C which helps in controlling the strength of regularization, this was set to 1e+42. Third parameter is ‘solver’ which specifies the algorithm to be used in the optimization problem, this was set to liblinear.

***Accuracy:***

The model provided an accuracy score of 0.8127 on the training data and .8210 on the validation data. The two confusion matrices below show the model performance on the train and validation data.

***3.1.4 RANDOM FOREST MODEL***

Random Forest is an ensemble learning technique that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or the average prediction (regression) of the individual trees. Known for high accuracy and resilient to overfitting, random forest is widely used for various machine learning tasks, offering great performance.

***Steps followed:***

Random Forest model was trained using the Random Forest Classifier function of the ensemble module in scikit-learn library. This function requires a parameter called ‘n-estimators’ which states how may individual trees will be built by the model; this was set to 2000 and a parameter called ‘random\_state’ which controls both the randomness of the bootstrapping of the samples used when building trees and the sampling of the features to consider when looking for the best split at each node, this was set to 1. Choosing an integer value for the random\_state parameter produces the same splits every time the model is re-trained.

***Accuracy score:***

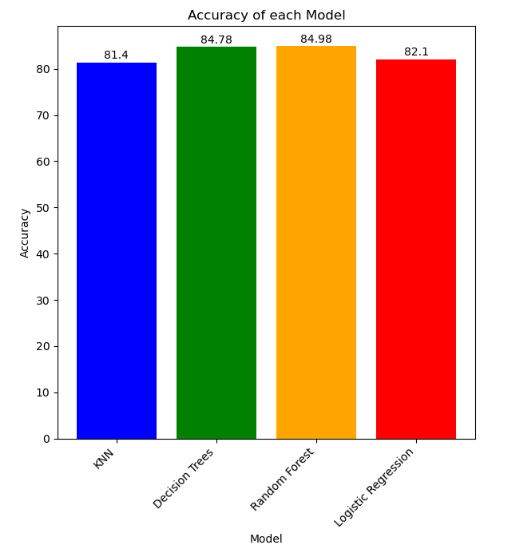
The confusion matrices for train and validation data for the model are shown below. Random Forest produced an accuracy of 1 for the train data and an accuracy of .8488 for the validation data.

A screenshot of a computer

Description automatically generated

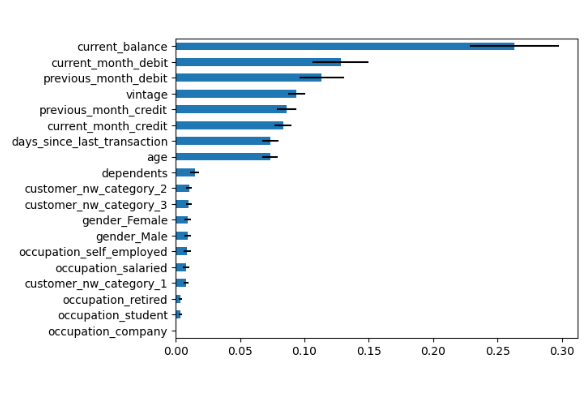
4. MACHINE LEARNING MODEL EVALUATION

After training the model, I ran my validation data through each model and checked the prediction accuracy for each model. The Random Forest model slightly outperformed the decision trees model by providing an accuracy of 84.98% slightly more than that of decision trees, which was 84.78%. Thus, I chose Random Forest model for deployment.



This could be because it uses the ensemble approach that helps to reduce overfitting and increases the model's performance. It is highly effective in my situation because the data I have received has more records of customers who did not churn than those who left the services.

An added advantage of random forest model is that it provides feature importance scores. These scores tell us which variables are more important in separating records of customers who will churn and who will not.



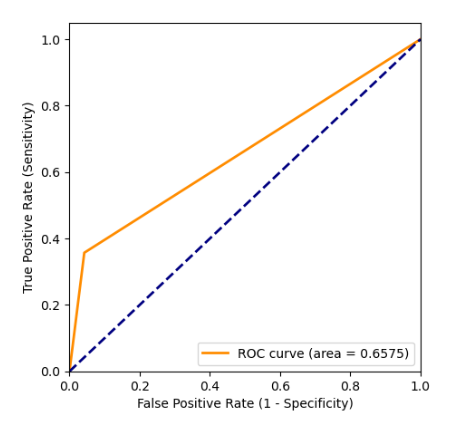
Through a thorough analysis of feature importance, I identified that the current balance holds a feature importance score of 27%. This underscores the critical role of monitoring and addressing issues related to the current balance in order to engage and retain customers at risk of churning. Additionally, the variable "vintage" emerged with a feature importance score of 9%, indicating importance of considering the duration of a customer's association with the bank. Understanding the vintage variable provides valuable insights into the long-term relationship dynamics, further guiding targeted retention strategies. These findings emphasize the strategic importance of leveraging specific features in my predictive model for proactive customer retention efforts.

**5. CONCLUSION**

After my complete analysis, I concluded that it is highly expensive to acquire new customers, thus the focus of the bank should be on retaining the existing ones. The bank should reach out to those customers who were predicted by my model and are at risk of leaving the services. The bank can take measures like sending personalized emails and offers, aiming to increase retention and promoting continued use of savings accounts.

In conclusion, the usage of a Receiver Operating Characteristic (ROC) curve in my study served as a pivotal component for evaluating and presenting the performance of my classification model i.e., Random Forest. The ROC curve provided a comprehensive view of the model's ability to distinguish between classes at varying classification cut-offs. This visual representation not only facilitated the selection of an optimal threshold based on specific requirements but also allowed us to assess the trade-off between sensitivity and specificity.

The ROC curve proved to be an effective diagnostic tool, aiding in decision-making processes and enhancing the interpretability of results, especially in critical decision contexts.



In my analysis, I can see two lines plotted in relation with false positive and true positive rates. One is the dotted blue line, which is a depiction of base model with zero prediction power and the other is the orange line that depicts the ROC curve. This means that it has no ability to predict the customer who could churn in the future. The ROC curve has the cut-off value for this model, which starts from zero at the top right. This is where a model identifies a lot of customers who are likely to churn, which is indeed good for the bank. But at the same time, it produces many records as false positives.

As the cut-off value increases, I can still identify many customers who could leave the service, but this also reduces the false positives.

**6 REFERENCES:**

G. Shmueli, P. C. Bruce, P. Gedeck, and N. R. Patel, "Data Mining for Business Analytics: Concepts, Techniques, and Applications in Python," 1st ed. Boston, MA, USA: Wiley, 2020.

Zach. "How to Plot a ROC Curve in Python (Step-by-Step)," Statology, posted on April 6, 2021. Available: <https://www.statology.org/plot-roc-curve-python/>. [Accessed: December 2, 2023].

A. Wyatt, "How to Evaluate Classification Model Performance with Cumulative Gains, Lift Curves and Python," Towards Data Science, Jul. 4, 2020. [Online]. Available: [[How to Evaluate Classification Model Performance with Cumulative Gains, Lift Curves and Python](https://towardsdatascience.com/evaluate-model-performance-with-cumulative-gains-and-lift-curves-1f3f8f79da01)]. [Accessed: December 3, 2023].