Churn Prediction in Retail Banking

Adi Srikanth • Andre Chen • David Roth

Jin Ishizuka • Lev Paciorkowski

NYU Center for Data Science

December 22, 2021

I. Introduction

In this analysis, we look at a dataset of $10,000$ retail banking customers and try to predict whether an individual customer is likely to "churn", that is, if they are likely to close their account and cease their financial relationship with their bank. We evaluate the significance of several segmentations of customers as predictors of churn, and experiment with a number of model classes, including Logistic Regression, kNN (k-Nearest Neighbors) and Random Forests to predict the likelihood of customer churn given financial and demographic data.

II. Data Summary

Our dataset contains information on customer churn at a retail bank, specifically features/predictor variables and a target variable (1 if the customer churned, 0 if the customer is still at the bank). The features include general information about the customers including demographics, socioeconomic information, and the level of engagement each customer has with the bank. Our goal is to study the relationships between the customer predictor variables and their probability of churning to understand what leads to customer churn at this bank.

We determined that investigating this dataset would be broadly useful as practice for future roles as data scientists in industry. Specifically, a very common use case in both consumer and enterprise-facing companies is understanding leading indicators of churn, as churn contributes to significant revenue erosion and needs to be proactively addressed. This dataset offers a great opportunity for us to practice not only the technical skills involved in determining predictors of customer churn, but also the more business-oriented aspects of asking suitable questions and presenting a cohesive conclusion that could benefit our hypothetical company. The data is broken down as follows:

Rows: Each row represents an individual customer of the bank, providing information about that customer and whether or not they churned, which we assume in this context means they closed their account with the bank and stopped doing business with them altogether.

Columns: The 14th and last column of the dataset (“Exited”) is a binary target variable that indicates whether a customer churned (1) or not (0). The first 13 columns of the dataset include information about each individual customer, of which a large subset were identified as potentially useful predictor variables and are described in detail below. We omitted columns like “CustomerId” and “Surname” from the columns listed below, as these columns likely have no relationship with a customer’s likelihood of churning:

- *CreditScore*: The customer’s credit score on a scale from 350 (worst) to 850 (best)
- *Geography*: The customer’s country of residence (dataset includes France, Germany, and Spain)
- *Gender*: Male or Female
- *Age*: Age of customer $(18 - 92)$
- *Tenure*: How long the customer has been with the bank, in years.
- *Balance*: Customer balance to date. (Note: customers who have closed their accounts with the bnk may still have uncollected balances)
- *NumOfProducts*: The number of bank products

used by the customer.

- *HasCrCard*: Whether the customer has a credit card (1) or not (0)
- *IsActiveMember*: Binary variable indicating if customer's activity is above the bank's minimim threshold for "Active".
- *Estimated Salary*: Customer's estimated yearly salary.

III. Hypothesis Testing

Do customers with varying levels of engagement with the bank have significantly different probabilities of churn? What does this tell us about the measures of engagement which matter the most?

OUR ultimate goal is To determine indicators of churn that the bank can act on, so that we could then recommend specific areas the bank could target for improvement in their interactions with customers. We accomplish this by defining customer profiles based on three measures of engagement (*NumOfProducts*, *IsActiveMember* and *Tenure*), and then implement hypothesis testing to see if churn proportion is significantly different from one profile to another.

i. "Product-Driven" Customers

We find that only about 3% of customers have more than two products with the bank, and of them, 85.9% churned compared to just 18.2% of non Product-Driven customers.

$$H_0 : P(churn)_{\text{product-driven}} = P(churn)_{\text{non-product-driven}}$$
$$H_1 : P(churn)_{\text{product-driven}} > P(churn)_{\text{non-product-driven}}$$

ii. "Active" Customers

Customers are evenly split between Active and not Active (as determined by the bank), and only 14.3% of Active customers churned, compared to 26.9% of non-Active customers.

$$H_0 : P(churn)_{\text{active}} = P(churn)_{\text{non-active}}$$
$$H_1 : P(churn)_{\text{active}} > P(churn)_{\text{non-active}}$$

iii. "Established" Customers

We find that Customers with tenures longer than 1 year churned at a rate of 20%, compared to new customers, who churned at a rate of 22.6%.

$$H_0 : P(churn)_{\text{established}} = P(churn)_{\text{new}}$$
$$H_1 : P(churn)_{\text{established}} > P(churn)_{\text{new}}$$

Table 1: *Significance Test Results*

	Segment	p-value
0	*Product-Drivenness*	$2.9e-196$
1	*"Active"-ness*	$3.0e-55$
2	*Tenure*	$1.2e-02$

Table 2: *Boruta Example*

A	*A_copy*	*B*	*B_copy*
1	5	22	30
5	6	25	21
6	7	21	25
7	1	30	22

Results

In general, Product-Drivenness and "Active"-ness are highly significant predictors of churn at $\alpha = 0.005$, while Tenure is not (See Table 1). Although Product-Driven customers are quite rare, they have a much greater tendency to churn compared to mainstream customers who use only one or two products. Perhaps this could be an indication of so-called credit card "hopping", but deeper investigation would be required to evaluate this. Based on these results, the bank can be confident in how it assesses "Active" customers, but might consider interventions to disincentivize customers from becoming Product-Driven. In contrast, a customer's Tenure should not be a central focus.

IV. Feature Selection

IN order to remove excess "noise" from our model, we embark on feature selection. Specifically, we implement the feature selection process known as the Boruta algorithm. Boruta functions by creating a copy of each variable in the dataset, randomizing the copy, and attaching it back to the dataset. For example, if a dataset has original columns A and B, Boruta takes that dataset and produces a new dataset with columns *A*, *A_Copy*, *B*, and *B_Copy*, where *A_Copy* and *B_Copy* are shuffled versions of A and B, respectively (consider the copy columns essentially a random number generator sampling from the range of their original columns). Table 2 illustrates this example.

Next, Boruta takes all of the "copy" columns and selects the copy column that was the best predictor of the dependent variable. If any of the original columns is a worse indicator of the dependent variable than the best copy column, that original column is removed. This entire process is repeated to ensure statistical significance. The foundational idea is that no variable in the data set is a worse predictor than a random column of data – each variable must have a non-random contribution towards predicting the dependent variable.

Ultimately, we whittle our original set of variables down to seven variables: age, estimated salary, credit score, country of origin, number of products owned, member status, and remaining balance. We use these variables in the next step of our analysis, model implementation, tuning, and evaluation.

V. Feature Correlation and PCA

Before moving on to model construction, tuning and evaluation, we pause to inspect our selected features and determine whether any are substantial correlated with one another, and if so, if they might be effectively represented in terms of a smaller number of principal factors. This will be important for downstream predictors like Logistic Regression which rely on the assumption that their inputs are uncorrelated.

We find no significant correlative relationships between our predictor variables (Figure 1). This is confirmed by our PCA analysis: We find that explained variance scales linearly with the number of principal components used in our reconstruction, only reaching 90% when 8 of 9 principal components are included (Figure 2). While the scree plot does display an “elbow” after the 3^{rd} principal component, only $\sim$ 50% of the total variance is explained at that point (Figure 3).

Given these results, we elect to continue on to model evaluation using our full, unreduced dataset.

VI. Model Implementation, Tuning, & Evaluation

To build an effective model using our dataset, we take a rapid prototyping approach by optimizing and comparing various supervised classification models in their ability to predict churn. Below we detail steps in our generalized approach to assessing each model, including train/test split, hyperparameter tuning, and model evaluation:

i. Evaluation Metric

Considering the costly impacts of both false positives (wasting resources on customers not at risk of churn) and false negatives (losing customers), a reasonable metric

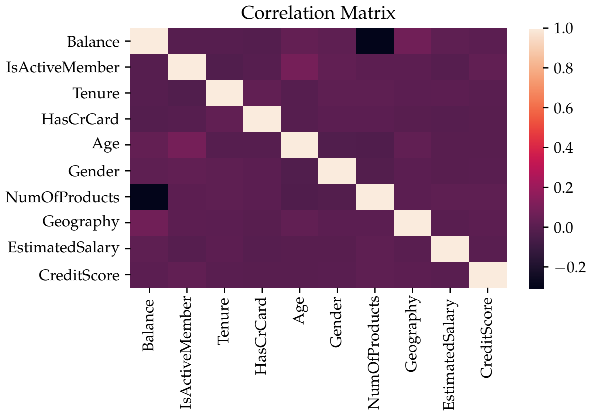

Figure 1: *Correlations among predictor variables.*

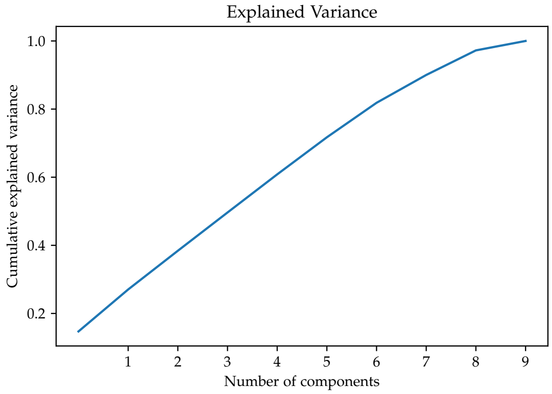

Figure 2: *Cumulative explained variance.*

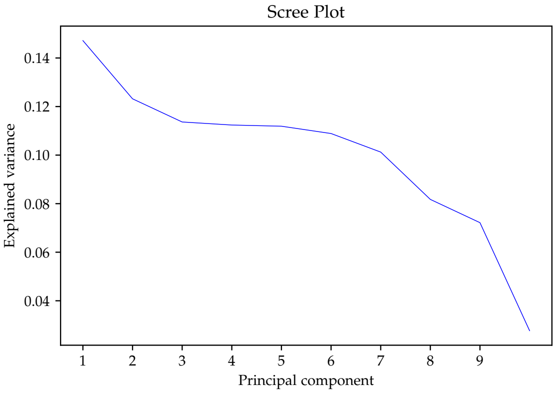

Figure 3: *Explained variance by principal component.*

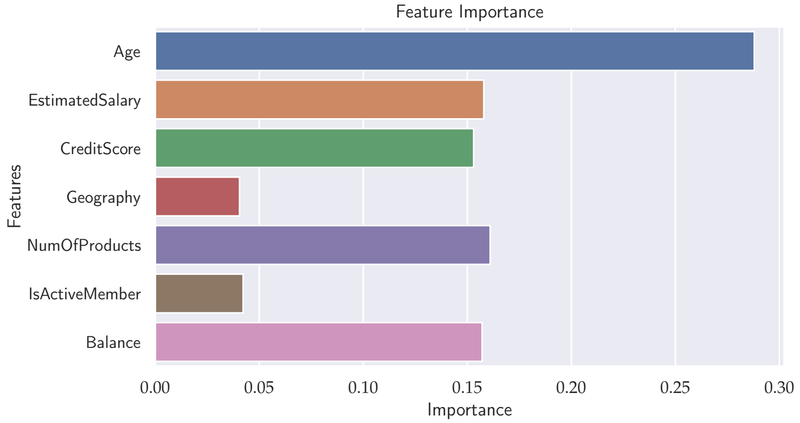

Figure 4: *Feature importance, ranked.*

that helps us balance between the two in model optimization is the F1 score. Specifically, we look at the F1 score for our "positive" predictions (overall ability to predict churn) as well as the macro-averaged F1 (unweighted average of F1 scores on predicting churn and retention, respectively) as a secondary metric.

ii. Train/Test Split

We used an 80/20 split for training/testing (holdout). The 20% holdout set was used only to compute a final evaluation metric for a given model after optimizing on the training set.

iii. Standardization

For models that require standardization (e.g. Logistic Regression, SVM, KNN), we center the data and scale to unit variance.

iv. Class Imbalance

We observe an imbalance in our data between positive and negative classes. To rectify this, we employ SMOTE when training our models, which over-samples the minority class and under-samples the majority class. We found this to improve performance across all model classes.

v. Hyperparameter Tuning

For each model, we identify its most important hyperparameters and define a reasonable search space for each parameter. We use grid search or randomized search to approximate an optimal combination of hyperparameters based on F1 score.

vi. Final Evaluation

For each optimized model, we compute its F1 score by comparing its predictions on the test set features against the true values of the test set target variable. F1 scores for five different models are reported in Table 3

Table 3: *Model Performance*

Model	F1	Averaged F1
Logistic Regresion	0.50	0.65
K-Nearest Neighbors	0.55	0.73
Support Vector Machine	0.54	0.73
Random Forest	0.6	0.75

VII. Results

The best-performing model is the Random Forest Classifier, with an F1 score of 0.60 and a Macro-Averaged F1 Score of 0.75. We generate this model using the approach outlined above, including train/test splitting, standardization, and hyperparameter tuning. Prior to model training, we employ the SMOTE algorithm to offset a class imbalance between Churn = 1 and Churn = 0 by oversampling data points where Churn = 1. Before the inclusion of SMOTE, only 20% of data points cover the case of Churn = 1. After employing SMOTE, we see a roughly even split.

We implement our model using the *scikit-learn* library, specifically the *RandomForestClassifier* package for the model and *RandomizedSearchCV* for hyperparameter tuning. After hyperparameter tuning, the optimal version of the Random Forest model has 1500 estimators and a maximum depth of 1500 along with additional tuned parameters (see code for additional detail). Based on this version of the Random Forest, we extract feature importances displayed in Figure 4. (Note: feature importance is pragmatically unit-less, as the values for feature importance are primarily useful as relative measures against other features in this model). Age appears, by far, to be the most impactful feature, followed by Number of Products and Remaining Balance. The remaining features are significant, as established by our feature selection algorithm, but do not exhibit an outsized impact on the model.

Remarks

We note that with additional computation power, a more exhaustive search for optimal parameters could yield even better results. As such, we do not claim this iteration of the model as a global optimal. Adding even more estimators or increasing the maximum depth could produce a more thorough and consequently a more accurate model.

Additionally, we acknowledge that our F1 score localized on churn = 1 is worse than our aggregated F1 score. In short, this indicates that our model is worse at predicting churn than at predicting non-churn. Given that the business case is specifically predicting churn, this represents a shortcoming of our model. A potential step to rectify this could be to place a disproportionate weight on Churn = 1 during model development. Currently, after using SMOTE, the importance of data points where Churn = 1 matches that of data points where Churn = 0. Modifying these weights would communicate a priority on correctly predicting cases where Churn = 1 to our model.

VIII. Conclusion

In our analysis, we began by identifying customer groups that are at high risk of leaving the bank and looked at whether customers with varying levels of engagement have statistically significant differences in their likelihood to churn. Through our hypothesis testing, we identified Product-Drivenness and “Active”-ness as two engagement metrics likely to predict at-risk customers. Given these results, we feel it would likely be in the bank’s best interest to identify customers with multiple products and high “Active” scores early and target these customers with loyalty programs and retention campaigns.

Following our hypothesis testing, we explored feature correlation and PCA in an effort to minimize multicollinearity in our models. Looking at the correlation matrix for our predictive features, we find no significant correlations. This finding was further confirmed by our PCA analysis. With these findings, we proceeded to our model implementations with our full dataset.

In the model implementation phase of our analysis, we compared a number model types to identify which best predicts customer churn given the set of available features. Prior to implementing our models, we utilized the Boruta algorithm to select a subset of features that have a non-spurious contribution toward predicting customer churn. We used this set of seven features to implement a series of supervised learning algorithms. After implementing Logistic Regression, K-Nearest Neighbors, Support Vector Machine, and Random Forest algorithms, we find that our Random Forest classifier saw the strongest performance with an F1 score of 0.60 and a Macro-Averaged F1 Score of 0.75. Furthermore, in our Random Forest model, we were able to identify age, number of products, and remaining balance as the most impactful features in terms of predicting churn.

The results of our model indicate that there is room for improvement in terms of creating a more accurate classifier, though these improvements may require a more comprehensive dataset. One limitation in our data was a class imbalance between customers who left the bank and customers who remained with the bank, with exiting customers accounting for only 20% of our data. A more balanced dataset may help in improving the accuracy of our models. Furthermore, with only 10 features in this dataset, it could also be the case that the current dataset simply does not contain enough information to create an accurate predictive model. Further efforts to model customer churn could utilize datasets with a more diverse set of features to more accurately predict turnover.