Predictive Modeling Of Customer Attrition at ABC Corporation

Tracey Zicherman

Executive Summary

This report presents the results of predictive modeling analysis aimed at understanding and predicting customer churn at ABC Corporation. The goal was to leverage customer data to identify patterns and build models capable of predicting whether a customer will close their account. The analysis involved data cleaning, exploratory data analysis, feature engineering, model selection and interpretation of results.

Preliminary EDA of the data showed that it was comprised of various demographic and financial features. Features were categorized and analyzed for distributions, patterns and predictive value. Preliminary cleaning steps ensured data readiness for feature engineering, preprocessing and modeling.

Next steps included data preprocessing and feature engineering. Certain features, in particular “Attrition_Flag”, were encoded using one-hot encoding to convert to binary form to enable utilization for predictive models and enhance interpretability. Initial distributions of numerous features demonstrated that they would require further transformations to prove useful, so algorithms were applied to achieve this. New features were engineered, all of which turned out to prove valuable in regards to feature selection (which ones are most relevant to customer attrition). Additionally, binning was implemented to prepare data for processing and maximize predictive value of features.

Once these steps were completed, the data was split into training and testing sets. Subsequently, 5 different models were fitted and hyperparameters tuned using 5-fold cross validation in order to obtain the highest predictive accuracy possible. After comparing the performance of each of these models on the test data using ROC-AUC and calibration curves, it was ascertained that the randomForest model is best at predicting customer attrition. Feature selection was then implemented to isolate which aspects of the customer data were most adept at predicting potential customer attrition.

Key findings conclude that the feature most predictive with respect to customer attrition was “Total_Ct_Chg_Q4_Q1”. This measures the change in transaction count from Quarter 4 to Quarter

1. The underlying assumption here is that the change is negative in nature, indicating a decrease in customer activity over these quarters. This could be due to a number of factors such as price increases or a change in customer needs which motivate said customer to seek alternative sources, leading to the reduction of their account activity. The next two most important predictors were "Total_Trans_Ct" and "Avg_Trans", These features encapsulate not only the number of transactions taking place, but the average spending pattern of these customers. Once again, the assumption is that the changes taking place that would indicate possible attrition are negative, as a decrease in account activity and spending are likely to precede customer churn.

Recommendations include close account monitoring/early detection system for decreases in account activity and spending, effectively allowing ABC to identify those accounts which appear to be at highest risk of attrition. Targeted efforts can then be made to engage these customers and a) evaluate the basis of the reductions and b) offer personalized incentives in accordance with their specific needs. This proactive approach to reducing customer attrition will not only help ABC retain revenue, but give it a better understanding of its customer base, enabling the development of more effective marketing strategies and improving overall customer satisfaction and loyalty. Ultimately, this will drive long-term business growth for ABC Corporation.

Approach and Data

Overall Approach

Our overall approach to solving the problem of predictive modeling of customer churn can be summarized as follows:

1. Perform basic cleaning and manipulating of raw customer data.
2. Explore raw customer data to understand features of customer behavior and possible relationships to customer churn, with an eye towards predictive modeling.
3. Engineer new customer features from pre-existing features.
4. Perform additional steps to maximize the predictive values of both pre-existing and engineered features
5. Process the resulting data to prepare for predictive modeling with machine learning algorithms.
6. Fit and tune 5 baseline classification machine learning algorithms which can be used to predict whether a given customer will churn based on features.
7. Evaluate the tuned baseline models on a number of different metrics to determine which is the best candidate for a final model.
8. Select a final model and interpret it to gain insights into the relationship between customer features and customer churn.

Analytic and Informational Goals

The overall goal is to learn as much as possible about the relationship between customer behavior, as measured by the features present in the dataset, and customer churn (attrition), and also to predict whether a customer will churn or not, based on a given set of values for these features, as accurately as possible.

Both interpretative and predictive modeling have clear value – both can be used to guide business decisions and policy, and to understand the customer base. It is helpful to think of them as complementary, since they may inform different decisions. Interpreting the relationship between demographic features and customer churn can inform marketing campaigns or retention policy, while predicting whether particular customers will churn can inform other strategic decisions.

Data

Data Dictionary

This is data dictionary for this dataset, which includes each customer feature and a brief description.

We note the following:

1. The client number is a unique identifier for internal purposes, and should have no real predictive value for the customer attrition.
2. Customer attrition is measured by a binary (true/false) feature which says whether or not the given account is closed. This will be our predictive target, the feature we are aiming to predict.
3. There are 6 demographic features, corresponding to customer's age, gender, number of dependents, education level, marital status, and level of income.
4. A single product feature described the customer's credit card category.
5. The remaining 12 features describe various aspects of the customer's credit card related behavior.

We observe features which fall into a number of different categories. For example, there are demographic features and financial features with time information (e.g. `Total_Trans_Amt`) and without time information (e.g. `Credit_Limit`).

Some features with time information measure customer behavior over the last twelve months (e.g. `Months_Inactive_12_mon`) while others measure on a longer time scale (e.g. `Months_on_book`) so we can assume that data are a snapshot of customer behavior over different time scales.

The predictive modeling task, given this snapshot of customer features at a moment in time, is to predict the likelihood their account will be closed at that same moment.

Given such a disparate mix of features, we were optimistic that we would be able to predict customer churn with a reasonably high level of accuracy, and this was borne out in our investigation.

Data Preparation

In preliminary work, our investigation determined that errors and duplicates did not appear to be an issue. The amount of data cleaning needed was minimal, and included:

1. ***Removing unnecessary or problematic features***. Since the client number has no known predictive value, this was removed from the data before modeling.
2. ***Enforcing the correct data types for features***. This step enabled the features to be recognized correctly by modeling algorithms.
3. ***Encoding missing values***. Since roughly 30% of the data contained missing values, these were encoded using the "Unknown" label, which could be processed as an additional possibility for the predictive modeling step.

After preliminary cleaning steps the dataset looks ready for feature engineering. All remaining features are relevant, and are represented by the correct data types, namely `fct` for categorical features, `int` for discrete numerical features, and `dbl` for continuous numerical features.

Descriptive Statistics and Visualizations

Now we present some statistics and plots relevant to the overall predictive modeling task, specifically with an eye towards feature engineering.

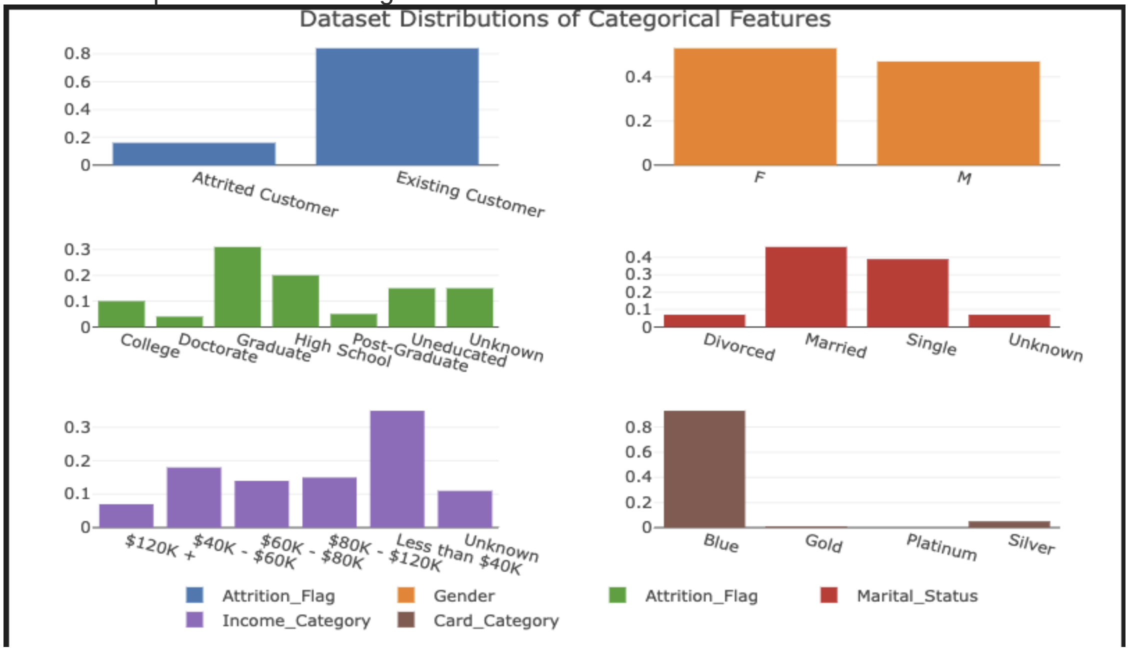
Categorical Features

The categorical features present in the dataset are:

```
## [1] "Attrition_Flag"  "Gender"          "Education_Level" "Marital_Status"
## [5] "Income_Category" "Card_Category"
```

Now, we visualize the distribution in the dataset of all categorical features. Our main goals with this visualization are:

1. To get a quick basic visual sense of the how these features are distributed among the customers represented in the data
2. Inform feature engineering, both the creation of new features, and the modification or transformation of existing ones.
3. Identify any potential issues or corrective action that might be advisable before using these features for predictive modeling.



From these distribution plots, we make the following observations:

- The proportion of churned customers (“attrited” customers, i.e. customers with closed accounts) is relatively low compared to existing customers.
- The overall proportions of female and male customers are more balanced, but with a slightly higher proportion of females.
- The most frequently occurring educational level was graduate, followed by high school. There were relatively fewer customers with doctorate or post-graduate educational level. A reasonably high proportion of customers have unknown educational level
- Almost all customers were either married or single, with a relatively small minority being divorced or of unknown status.

- The most frequently occurring income level by far was the lowest, namely `Income_Category == 'Less than $40k`. Whether this reflects something about the customer base or the underlying population is unknown.

We note that customer attrition, marital status, and card category stand out as somewhat unbalanced features. First we look at their individual distributions in the data.

Since customer attrition is the predictive target, and the imbalance is relatively moderate, we won't take any corrective measures here. Card category is however extremely unbalanced, so we will re-code this into two categories.

Finally for marital status, divorced and unknown values only represent small proportions of customers, but it doesn't make sense to bin these (since information will be erased). Since the proportions aren't too small (single digit percentages), we won't take any corrective measures for this feature, either.

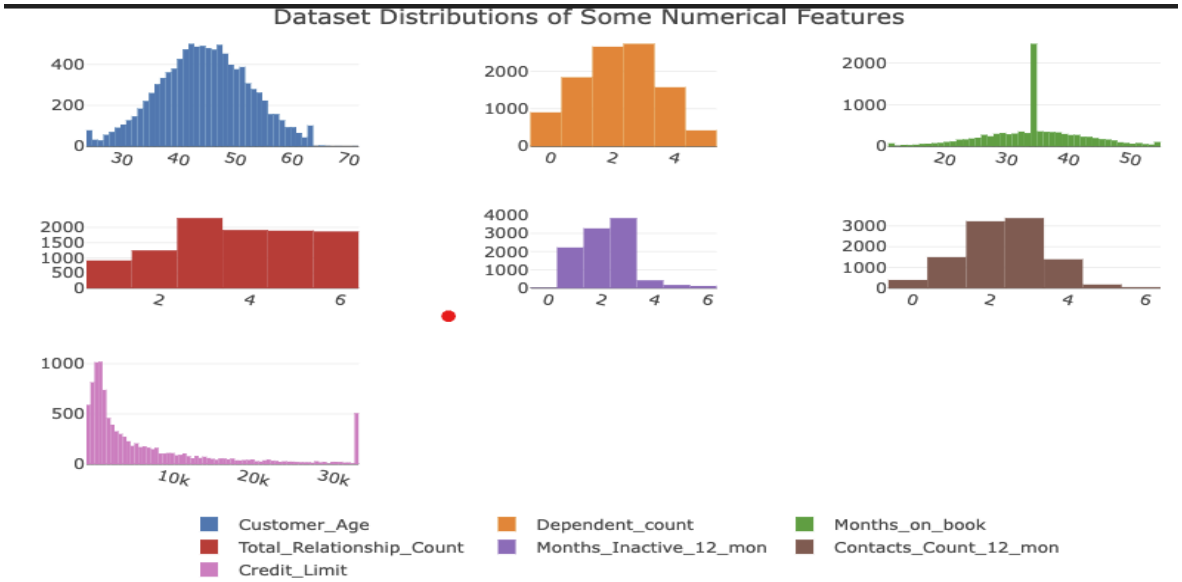
Numerical Features

Now we turn our attention to numerical features. As with the categorical features, we aim to get a sense of how the numerical features are distributed, inform feature engineering, and identify and correct any issues with these features that may affect predictive modeling.

These are numerical features present in the dataset

```
##  [1] "Customer_Age"             "Dependent_count"
##  [3] "Months_on_book"           "Total_Relationship_Count"
##  [5] "Months_Inactive_12_mon"   "Contacts_Count_12_mon"
##  [7] "Credit_Limit"             "Total_Revolving_Bal"
##  [9] "Avg_Open_To_Buy"          "Total_Amt_Chng_Q4_Q1"
## [11] "Total_Trans_Amt"          "Total_Trans_Ct"
## [13] "Total_Ct_Chng_Q4_Q1"      "Avg_Utilization_Ratio"
```

Now let's look at histograms for the first seven numerical features.



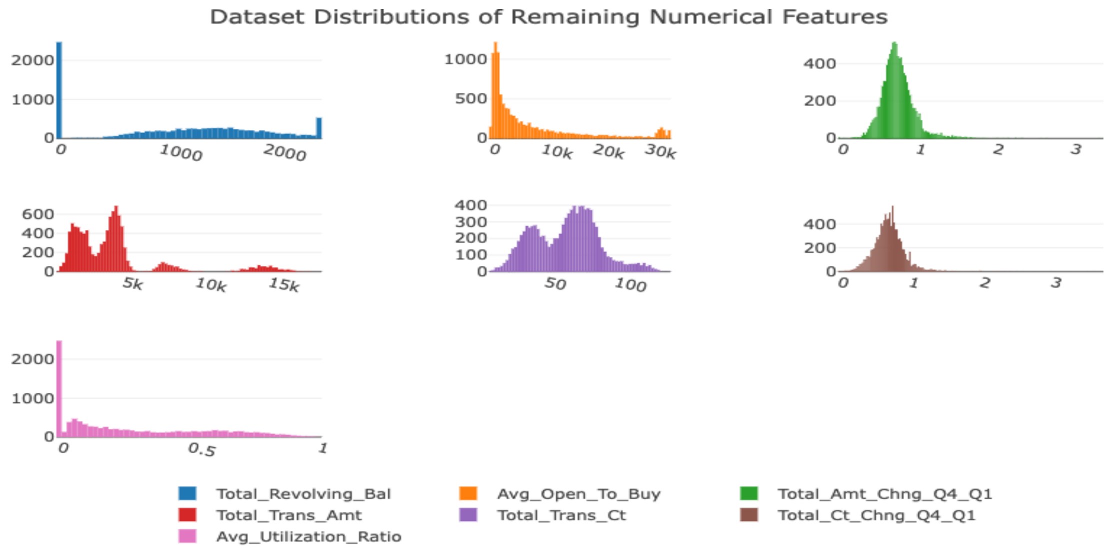
We note that `Customer_Age`, `Dependent_Count`, `Total_Relationship_Count` and `Contacts_Count_12_mon` seem relatively well behaved (e.g, symmetric).

The distribution of `Credit_Limit` has some interesting features - there is clearly very large proportion of customers with small credit limits. There is also an extreme right skew, and a curious spike at the maximum of the range. This is an excellent example of a situation in which binning is called for. For these reasons, we will bin `Credit_Limit`, to create a new ordinal feature with 7 levels.

Furthermore, `Months_on_book` has a single strange spike near the middle of the range.

This represents approximately 25% of the dataset, which is very unusual. Since this is too large a percentage to drop, we will choose to bin this feature. We will also bin `Months_Inactive_12_mon` to lump the values with small proportions together.

Now we plot the dataset distributions of the remaining eight numerical features



We note that `Total_Revolving_Bal` and `Avg_Utilization_Ratio` have a large spike around 0 and an extreme right skew. We will bin these as ordinal features with a low number of levels, with an extra lowest level for the zero value.

`Avg_Open_To_Buy` has a similar distribution compared to `Credit_Limit` with a high right skew and a spike at the lowest part of the range. `Total_Amt_Chng_Q4_Q1` and `Total_Ct_Chng_Q4_Q1` also both show extreme right skew. We will transform these features to try to reduce skew.

Finally, `Total_Trans_Amt` and `Total_Trans_Ct` are clearly multimodal distributions. For this reason, we will bin these features as well.

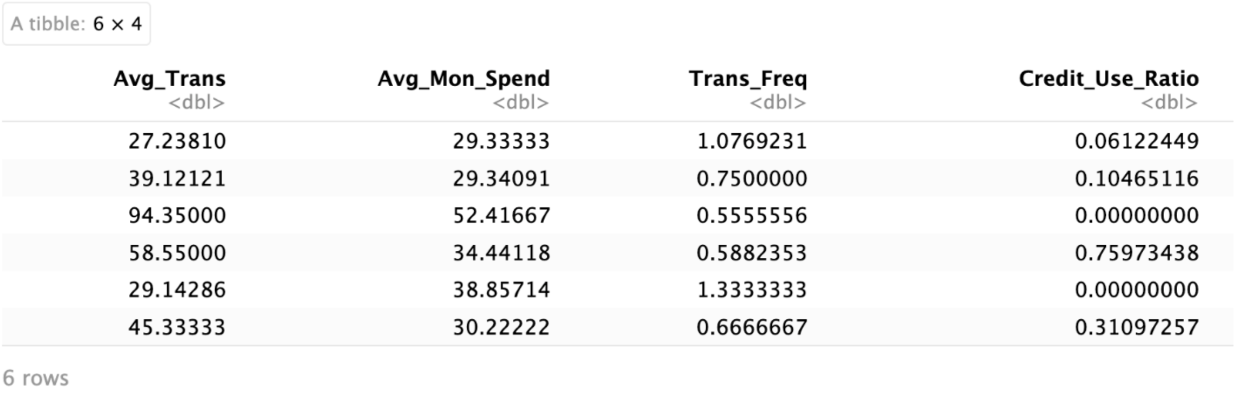
Here is a summary of numerical features which need further engineering:

1. Transformations: `Credit_Limit`, `Avg_Open_To_Buy`, `Total_Amt_Chng_Q4_Q1`, `Total_Ct_Chng_Q4_Q1`
2. Binning: `Card_Category`, `Months_on_book`, `Months_Inactive_12_mon`, `Total_Revolving_Bal`, `Avg_Open_To_Buy`, `Avg_Utilization_Ratio`, `Total_Trans_Amt`, `Total_Trans_Ct`

Feature Engineering

Here the intention is to create features which are both meaningfully interpretable and have additional predictive value. The exploratory analysis, in particular the distribution plots, as well as domain knowledge, informed the decision to create the following new features from pre-existing ones:

- `Avg_Trans = Total_Trans_Amt / Total_Trans_Ct` - Average amount of customer spending per transaction.
- `Avg_Mon_Spend = Total_Trans_Amt / Months_on_book` - Average monthly spending of the customer over the duration of their recorded relationship with the bank.
- `Trans_Freq = Total_Trans_Ct / Months_on_book` - Average number of transactions per month of the recorded customer duration.
- `Credit_Use_Ratio = Total_Revolving_Bal / Credit_Limit` - Proportion of available credit utilized, a well-known indicator of customer credit health.



A tibble: 6 × 4

Avg_Trans <dbl>	Avg_Mon_Spend <dbl>	Trans_Freq <dbl>	Credit_Use_Ratio <dbl>
27.23810	29.33333	1.0769231	0.06122449
39.12121	29.34091	0.7500000	0.10465116
94.35000	52.41667	0.5555556	0.00000000
58.55000	34.44118	0.5882353	0.75973438
29.14286	38.85714	1.3333333	0.00000000
45.33333	30.22222	0.6666667	0.31097257

6 rows

Basic summary statistics on the new customer features

```
##    Avg_Trans       Avg_Mon_Spend       Trans_Freq     Credit_Use_Ratio
##  Min.   : 19.14   Min.   :  10.00   Min.   :0.1887   Min.   :0.00000
##  1st Qu.: 47.51   1st Qu.:  62.36   1st Qu.:1.2727   1st Qu.:0.02271
##  Median : 55.79   Median : 105.80   Median :1.8571   Median :0.17565
##  Mean   : 62.61   Mean   : 131.01   Mean   :1.9231   Mean   :0.27489
##  3rd Qu.: 65.48   3rd Qu.: 141.36   3rd Qu.:2.3611   3rd Qu.:0.50269
##  Max.   :190.19   Max.   :1256.85   Max.   :9.7692   Max.   :0.99877
```

Further exploratory analysis in the preliminary steps showed that these features all had sufficient predictive value to justify including them in the predictive modeling.

Binning

In this section, we perform the previously mentioned binning of the following features: `Card_Category`, `Months_on_book`, `Months_Inactive_12_mon`, `Total_Revolving_Bal`, `Avg_Utilization_Ratio`, `Total_Trans_Amt`, `Total_Trans_Ct`, `Credit_Use_Ratio`. The choice of bins are made by hand in this case, although there are a wide range of options for doing so, including more advanced techniques.

Note that this binning will result in an ordinal (not categorical) feature because of the natural ordering reflecting the ordering of the underlying continuous feature.

Here is a summary of the re-encoding:

1. Card category will be re-grouped into two categories, 1 corresponding to Blue card category, 2 corresponding to all other card types.
2. Months on book will be binned into 6 bins, roughly corresponding to the number of years of book, with an additional category for `Months_on_book == 36`, since this represents roughly 25% of customers.
3. Months Inactive in the last twelve will binned into 4 bins, corresponding to 0 and 1, 2, 3, and more than 4 months.
4. Total Revolving Balances will be binned into six bins, corresponding to 0-500, 500-1000, 1000-1500, 1500-2000, and > 2000
5. Credit utilization ratio and average utilization will be binned into 6 bins of equal width 0-0.2, 0.2-0.4, 0.4-0.6, 0.6-0.8, and 0.8-1.0.
6. Total transaction amount will be binned into 5 bins, 0, >0 - 3000, 3000-5000, 5000-10000, and > 10000
7. Total transaction count will be binned into 3 bins, 0-50, 5-80, and > 80.

*Visuals provided in Appendix.

Transformations

Here we apply the logarithm transformation to the following features: `Credit_Limit`, `Avg_Open_To_Buy`, `Total_Amt_Chng_Q4_Q1`, `Total_Ct_Chng_Q4_Q1`, `Avg_Trans`, `Avg_Mon_Spend`, `Trans_Freq`.

This transformation is intended reduce skew. We cannot straightforwardly apply this transformation to any zero values of these features. To get around this issue, we can drop these rows, or perform a not uncommon step of imputing a negligible, non zero value. Since these customers may be customers which churned and there are so few of these in the dataset, we take the latter approach. This results in distributions which are much more suitable for predictive modeling. In particular, they are closer to normally distributed.

Predictive Modeling

In this section we model the relationship between the customer features (preexisting and engineered) and customer attrition with the goal of maximizing predictive accuracy.

We perform the final preprocessing steps, then split the data into train and test data. Using the training data, we fit and tune five baseline classifier predictive models based on well-known machine learning classification algorithms. We then evaluate these using standard metrics, and comparison with a null (“worst-case”) model, thereby selecting a top choice.

Preprocessing

First we perform the final preprocessing steps necessary to get the data in a final form suitable for fitting machine learning algorithms.

Train/Test Split

A fairly common 75%-25% train-test split is performed. The training data will be used to tune the baseline predictive models a process called cross-validation, which provides a more accurate estimate of the model on unseen data, while the test data will be held out until the last step, to provide a final estimate of future model predictive accuracy.

```
## Train set dimensions: 7595 24

## Test set dimensions: 2532 24
```

Preprocessing Recipe

We next perform the following preprocessing steps

1. `step_other` - this pools infrequently occurring values of categorical features into another category called `"other"`
2. `step_dummy` - this one-hot encodes all categorical features.
3. `step_center` - this centers all features (which are all numeric by this stage in the processing pipeline) by subtracting them from their mean.
4. `step_scale` - this scales all features by dividing them by their standard deviation.

Hyperparameter Tuning with Cross-Validation on Train Data

In this section we tune the hyperparameters of the 5 baseline classification algorithms: logistic regression, k-nearest neighbors, naive Bayes, random forest, and gradient boosted machines (specifically, XGBoost).

We use 5-fold cross-validation to tune the model to the training data, to find the best hyperparameters for each model. Optimization is with respect to the area under the receiver operating characteristic curve (ROC-AUC), a good metric for a balanced classifier with a binary target.

To save training and computational resource time, we only tune a single hyperparameter for each model. We also use 5 CV folds, rather than 10 as in the preliminary steps. In both cases, experimentation revealed little additional benefit to exploring more hyperparameters or using a greater number of folds.

We note some numerical issues with naive Bayes, already indicating that perhaps this model is not a good choice for the dataset.

Detailed Findings and Evaluation

Here we provide an in-depth evaluation of models and description of analytical findings.

Evaluate Tuned Models

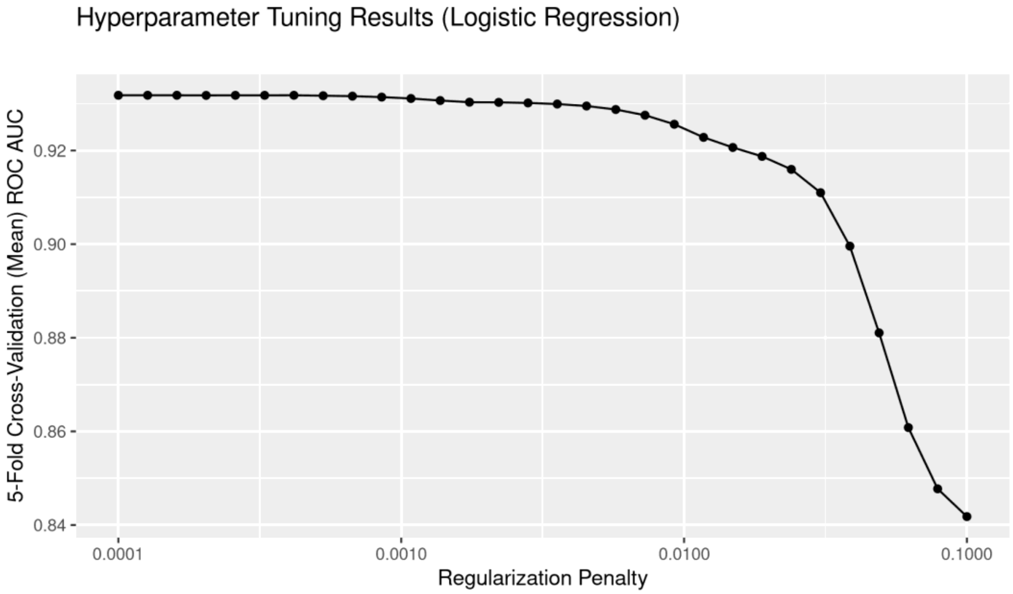
In this section, we look at the 5-fold CV estimates of the ROC-AUC for each classifier, highlighting the top 5 models (corresponding to the hyperparameter values searched in the CV tuning).

We then fit five models using the best hyperparameters obtained in the CV tuning step, obtain their predictions on the test data, and investigate them more deeply, plotting ROC and Calibration curves and calculating accuracy with respect to the model predictions.

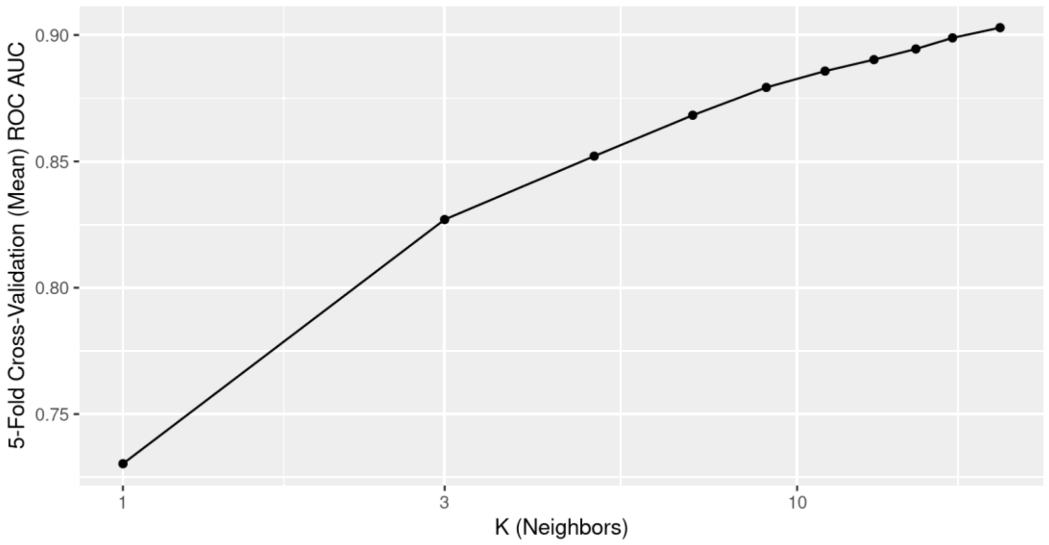
Finally, we use these results to select a top model.

Tuning Results and CV ROC-AUC Estimates

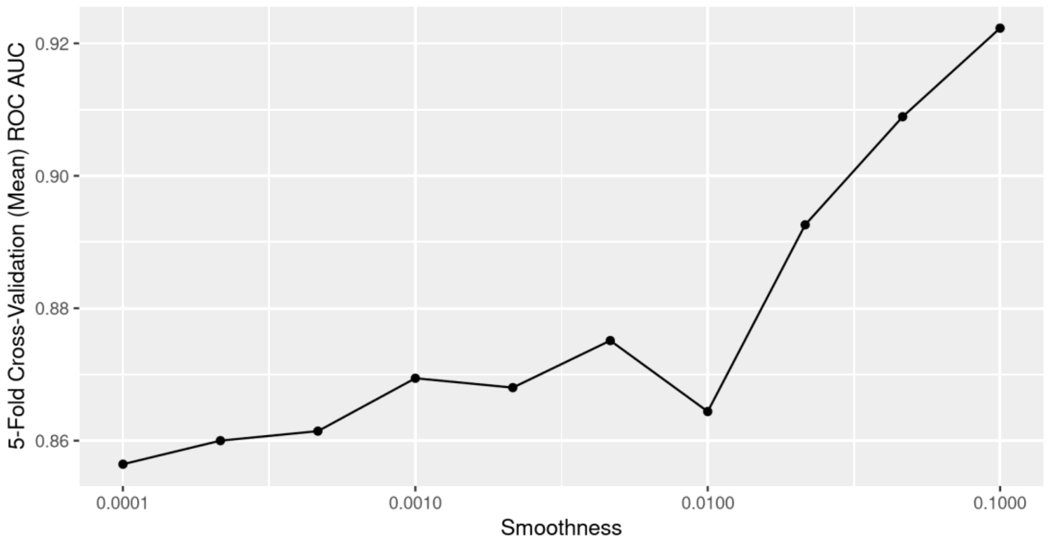
Here are plots of the hyperparameter values for each of the five baseline classification models, versus the mean ROC-AUC score over all 5 cross-validation folds.

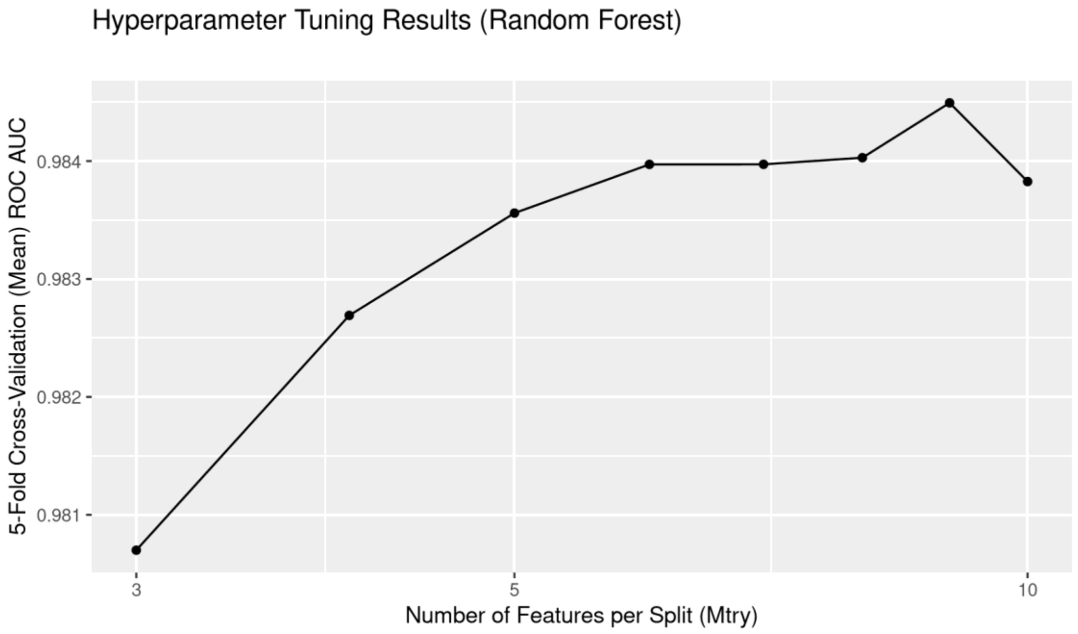


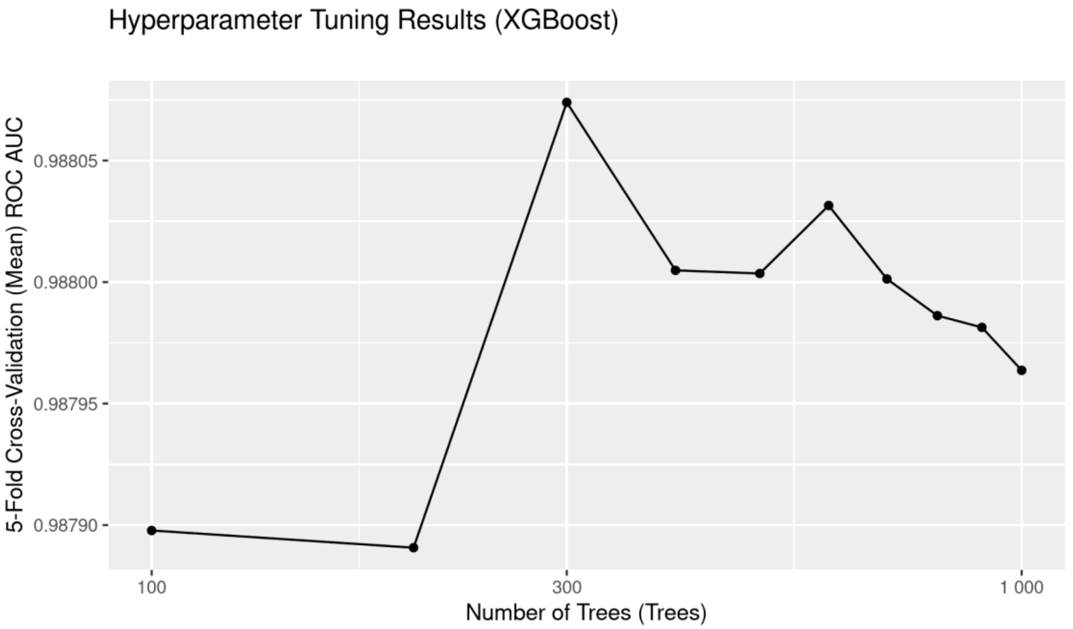
Hyperparameter Tuning Results (K-Nearest Neighbors)



Hyperparameter Tuning Results (Naive Bayes)







Here we summarize the cross-validation tuning results, selecting the top five hyperparameters for each model, ranked by mean CV ROC-AUC, and sorting overall mean CV ROC-AUC in descending order.

```
## # A tibble: 25 × 5
##    Model  Hyperparameter Hyperparameter_Value Mean_CV_ROC_AUC Std_Err_CV_ROC_AUC
##    <chr>  <chr>                         <dbl>           <dbl>              <dbl>
##   1 XGBoo… NumTrees                         200           0.987           0.000863
##   2 XGBoo… NumTrees                         100           0.987           0.000920
##   3 XGBoo… NumTrees                         300           0.987           0.000757
##   4 XGBoo… NumTrees                         400           0.987           0.000744
##   5 XGBoo… NumTrees                         500           0.987           0.000740
##   6 Rando… NumPredictors…                     8           0.983           0.000955
##   7 Rando… NumPredictors…                     6           0.983           0.000963
##   8 Rando… NumPredictors…                    10           0.983           0.00106
##   9 Rando… NumPredictors…                     9           0.983           0.00110
##  10 Rando… NumPredictors…                     7           0.983           0.00113
## # ℹ 15 more rows
```

From these results, there is a strong showing for the XGBoost gradient boosted machine model with all five top hyperparameter values having a better CV ROC-AUC then all other models, all just under 99%.

Random forest is a close second, again with all five top hyperparameter values having a better CV ROC-AUC then the remaining models, all just over 98%.

Naive Bayes and logistic regression models took the next position, with mixed ranking and accuracies between 92-94%.

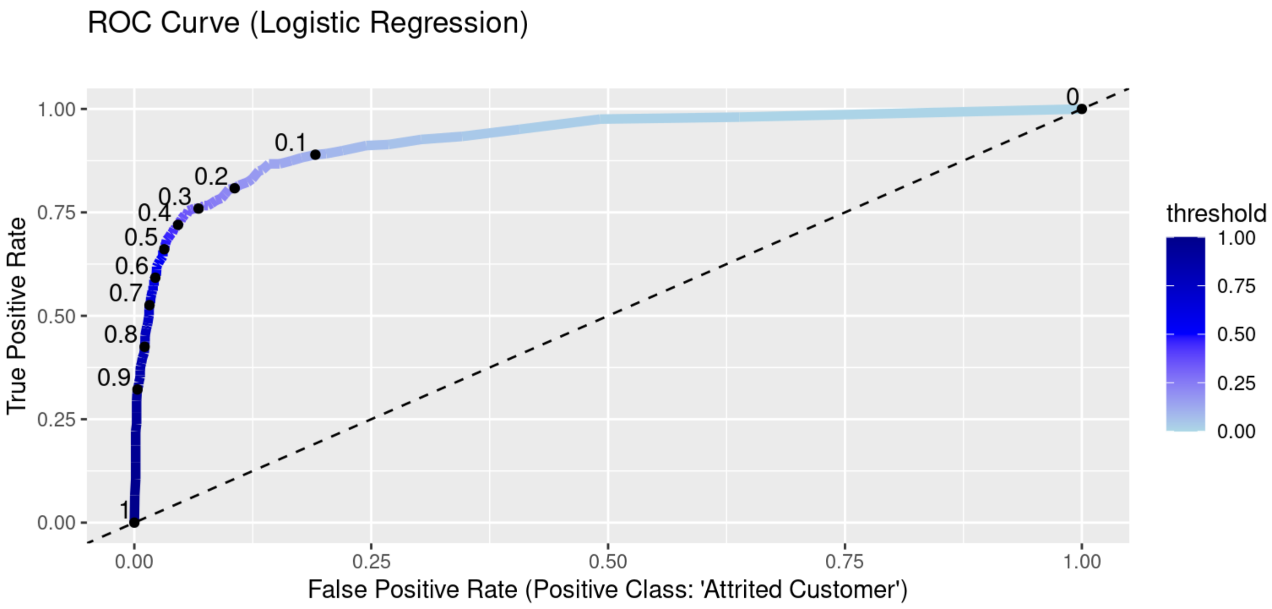
Finally K-nearest neighbors fared worst, with all five hyperparameters ranked lower than all other models, with CV ROC-AUC around 88-89%.

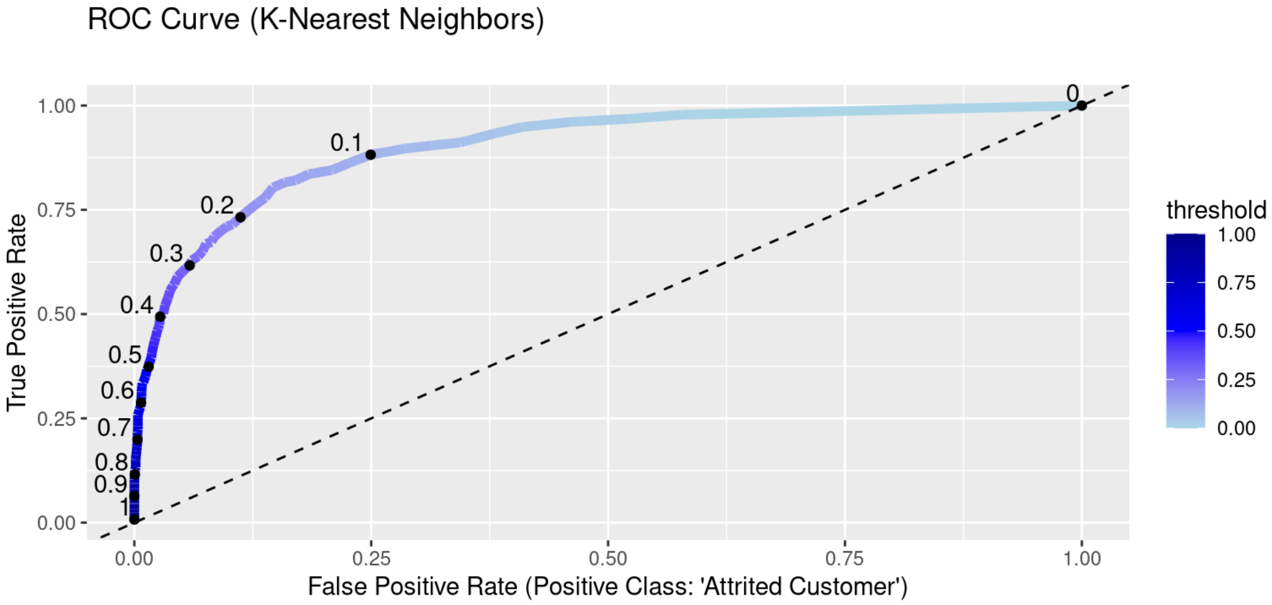
Null Model

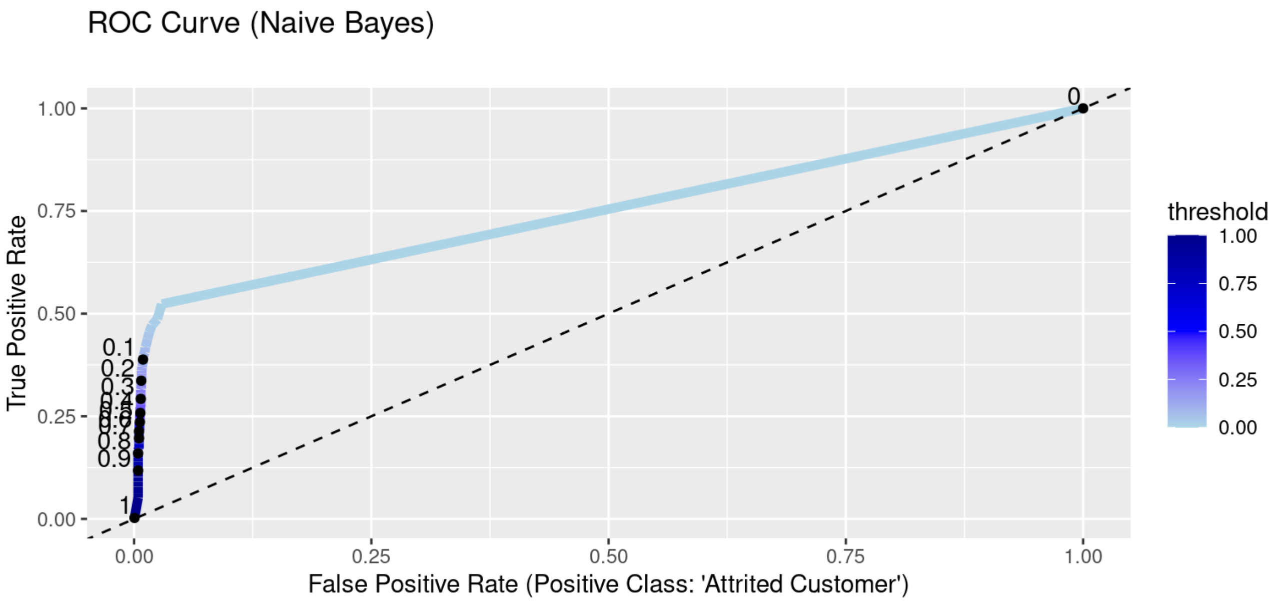
For this classification problem the classes are relatively unbalanced. For this reason, a good (and standard) choice for the null model is the model which predicts the mode (most frequently occurring class), name `"Existing Customer"`, in other words, predicting no customer attrition (this makes sense, as it is a relatively rare event).

ROC Curves on Test Data

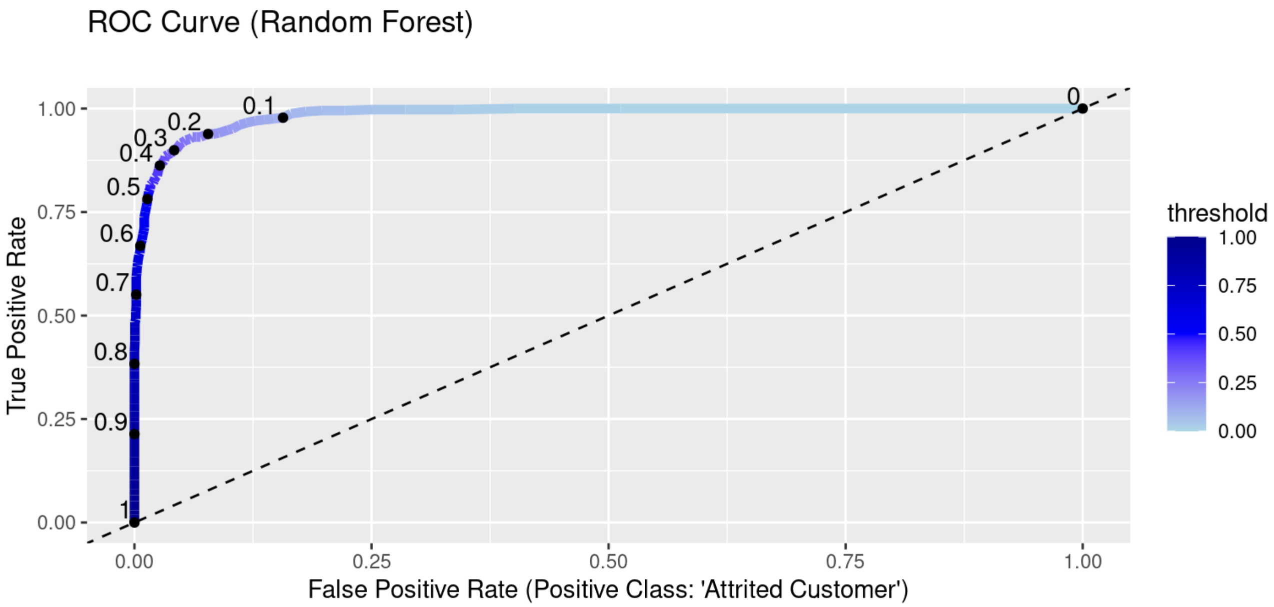
Here are the ROC curves for test data predictions of the five baseline models


ROC Curve (Logistic Regression)
True Positive Rate
False Positive Rate (Positive Class: 'Attrited Customer')
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
threshold
1.00
0.75
0.50
0.25
0.00

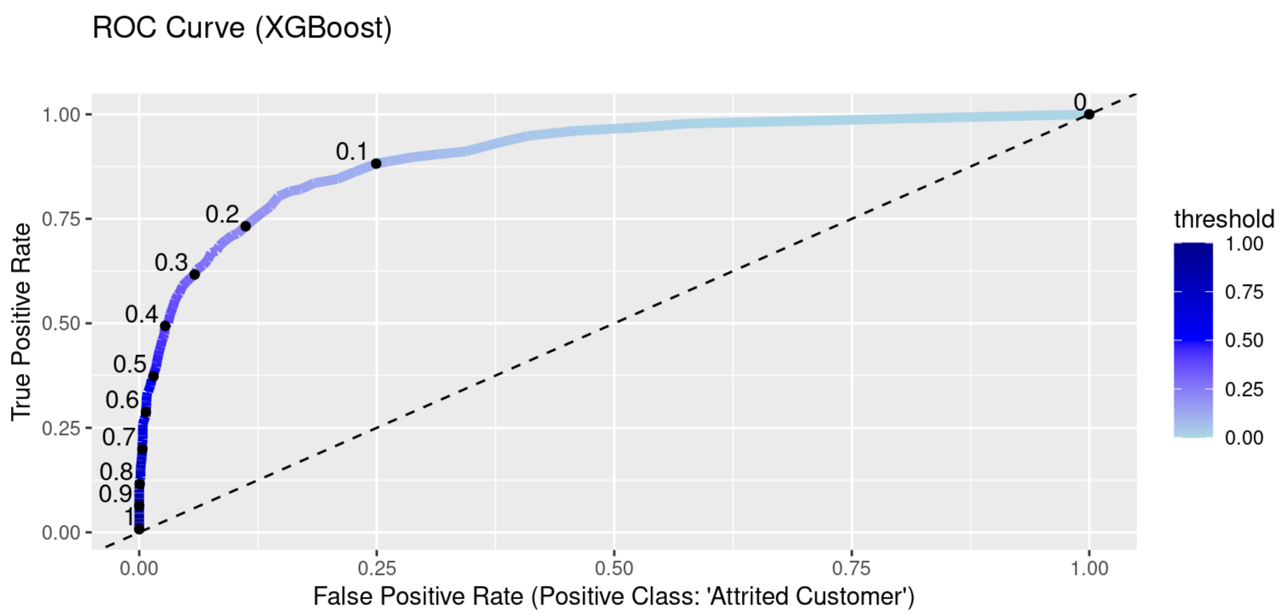

ROC Curve (K-Nearest Neighbors)
True Positive Rate
False Positive Rate (Positive Class: 'Attrited Customer')
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
threshold
1.00
0.75
0.50
0.25
0.00



ROC Curve (Naive Bayes)



ROC Curve (Random Forest)

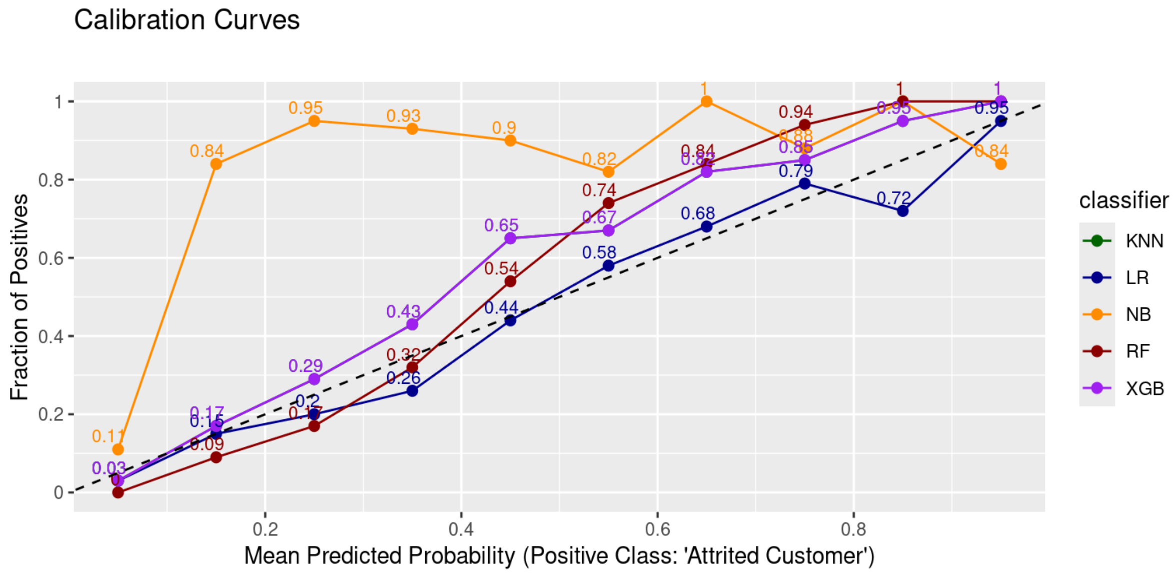


From these plots, we can see that the randomForest classifier has the best ROC-AUC on the test data, not `XGBoost` . This is not completely unexpected, since the `XGBoost` mean ROC AUC found during cross-validation was very close to that of randomForest.

This shows that random forest is much better than all other models at predicting the positive class at all probability thresholds, which indicates it is overall better suited to this data set as a predictive model.

Test Calibration Curves

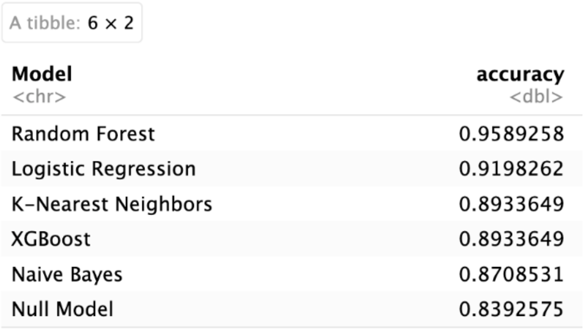
Here are the calibration curves for test data predictions of the five baseline models



The calibration curves provide strong evidence against the suitability of naive Bayes, which has a wild curve. The other models have more well-behaved calibration curves, with logistic regression and random forest clearly the best.

Test Accuracy

Here is the overall accuracy – i.e. the percentage of correctly classified customers – for the test data.



A tibble: 6 × 2

Model	accuracy
<chr>	<dbl>
Random Forest	0.9589258
Logistic Regression	0.9198262
K–Nearest Neighbors	0.8933649
XGBoost	0.8933649
Naive Bayes	0.8708531
Null Model	0.8392575

In terms of overall accuracy, random forest is the clear winner.

Note that the accuracy of the null model is relatively high, reflecting the imbalance of the classes, that is, the dominance of the existing customers in the data set. Note the proportion of the existing (non-attrited) customers in the data set is exactly the null model accuracy.

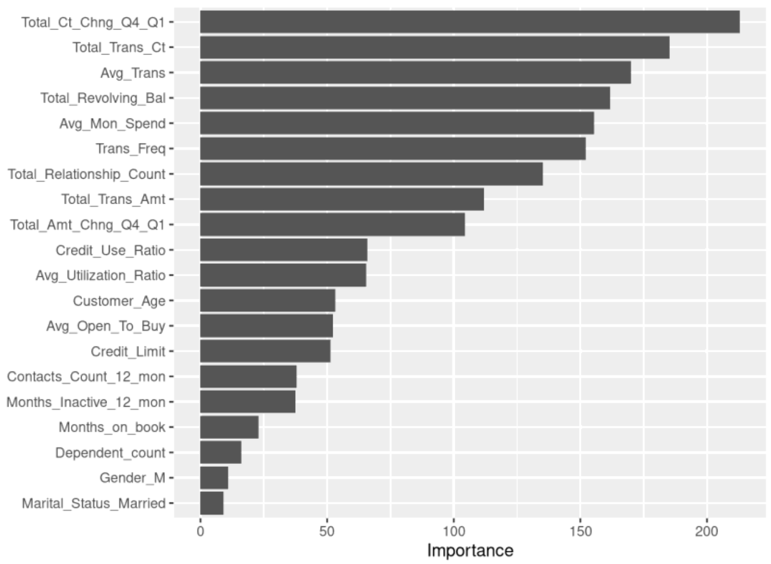
Summary and Interpretation of Findings

The results of the last section indicate that the random forest model is the best choice, with best test ROC-AUC, a top two calibration curve, and best test accuracy at just under 96%, so we select this as the final model for interpretation.

Feature Importances

The first step for interpretation is the feature importances.

Note that, although feature engineering resulted in a total of 24 features, which is a low enough number to practically view all feature importances, the feature encoding, particularly the one-hot encoding, increased this number substantially, so it is not practically feasible to view of all the importances.



A few observations and remarks:

1. Total change in number of transactions from Q4 to Q1 was by far the most important feature, followed by the total transaction count and average transaction.
2. All four engineered features proved important, including transaction frequency, average amount per transaction, average monthly spending and credit use ratio.
3. Demographic variables were clearly less important than financial and transaction variables.

Recommendations

Future Work

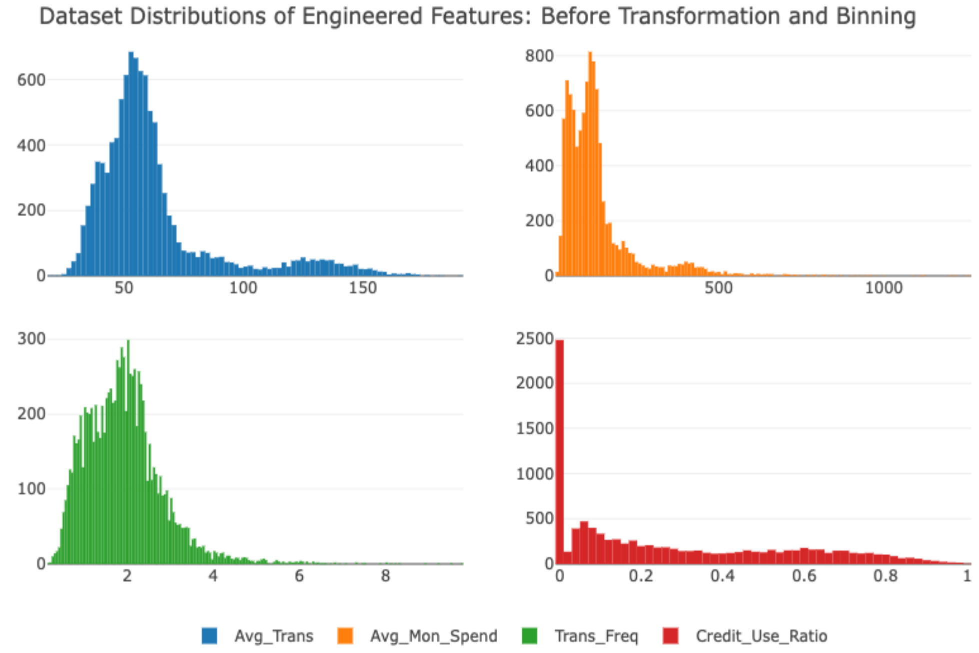
Here are some suggested steps that could be taken to improve the predictive model.

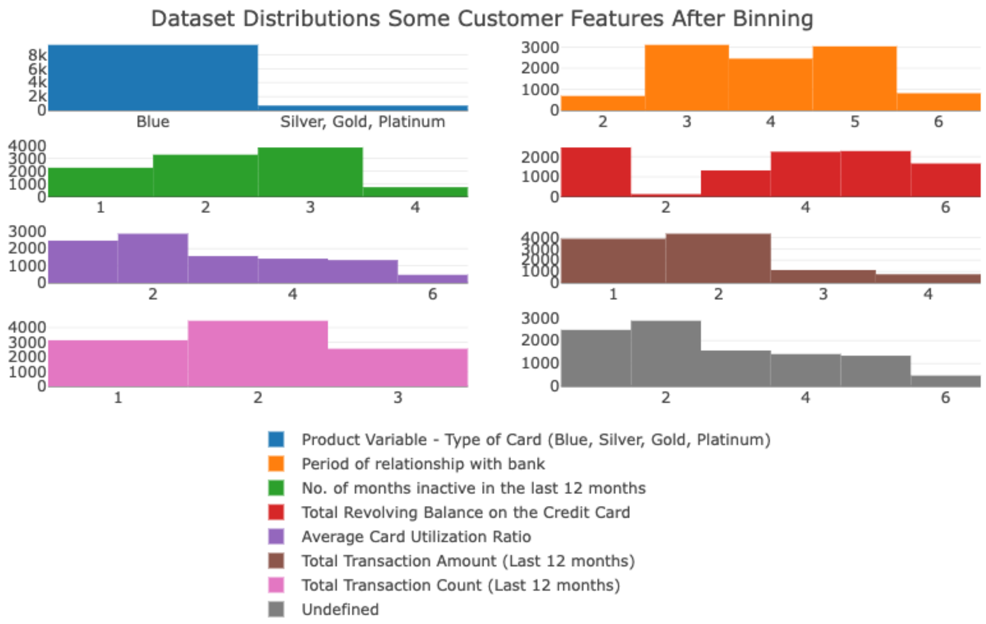
1. Gather data on more customers
2. Add additional features which have predictive value (these could be tested against their relationship to churn before being added to the customer database)
3. Explore more thoroughly the relationship between features and the target, for example with bivariate plots and stastistical hypothesis tests.
4. Recode some of the categorical features with natural orderings as ordinal features, for example `Education_Level` and `Income_Category`.
5. Expand hyperparameter tuning to explore more of the search space or additional hyperparameters.

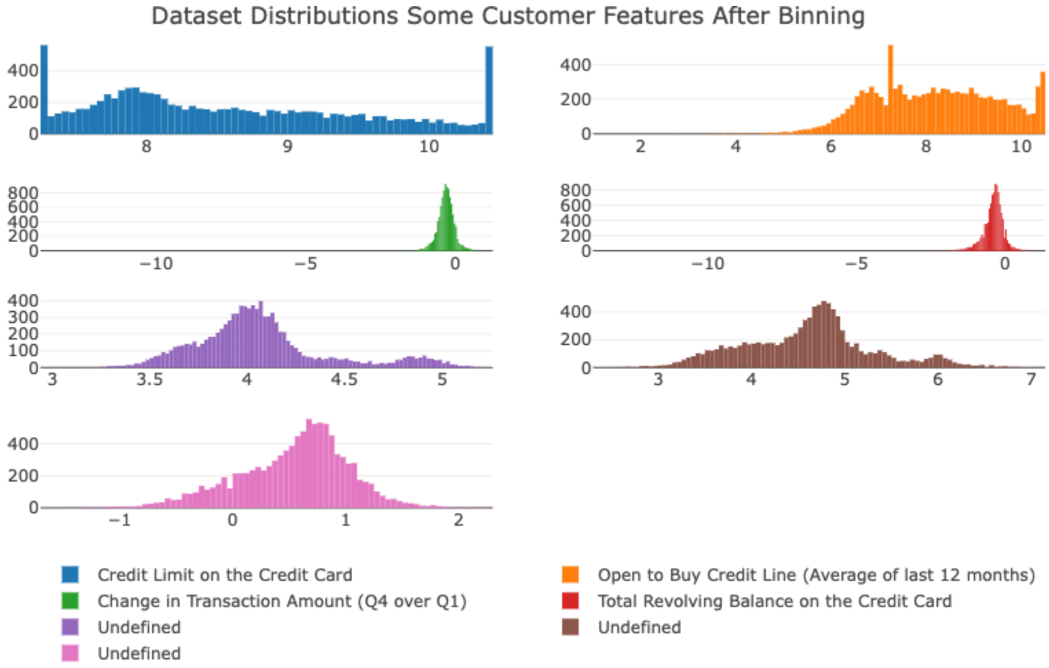
Based on the key findings of the analysis, it is evident that understanding and addressing changes in customer behavior are crucial in reducing customer attrition. With this in mind, it is recommended that close monitoring of transaction and account activity is performed, thereby facilitating early detection of any reductions in these areas. In doing so, ABC can proactively engage these customers and ascertain the basis of these reductions, whether due to price sensitivity, changing needs or other factors. By responding with personalized offers and incentives, ABC can reduce the possibility of attrition and foster long-term customer loyalty and satisfaction, ultimately driving business growth and revenue retention.

Appendix

Below are some of the distribution graphs used to show that binning certain features was necessary.






Dataset Distributions Some Customer Features After Binning
0
200
400
8
9
10
0
200
400
2
4
6
8
10
0
200
400
600
800
−10
−5
0
0
200
400
600
800
−10
−5
0
0
100
200
300
400
3
3.5
4
4.5
5
0
200
400
3
4
5
6
7
0
200
400
−1
0
1
2
Credit Limit on the Credit Card
Change in Transaction Amount (Q4 over Q1)
Undefined
Undefined
Open to Buy Credit Line (Average of last 12 months)
Total Revolving Balance on the Credit Card
Undefined