Preliminary Results

Tracey Zicherman

2024-04-01

# Setup

In this section we load the necessary libraries, and configure other settings for the remainder of the document.

# read .csv in as a tibble  
data <- as.tibble(read.csv("customer\_data.csv")) # modify local path to file

## Warning: `as.tibble()` was deprecated in tibble 2.0.0.  
## ℹ Please use `as\_tibble()` instead.  
## ℹ The signature and semantics have changed, see `?as\_tibble`.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

data

## # A tibble: 10,127 × 21  
## CLIENTNUM Attrition\_Flag Customer\_Age Gender Dependent\_count Education\_Level  
## <int> <chr> <int> <chr> <int> <chr>   
## 1 768805383 Existing Custo… 45 M 3 High School   
## 2 818770008 Existing Custo… 49 F 5 Graduate   
## 3 713982108 Existing Custo… 51 M 3 Graduate   
## 4 769911858 Existing Custo… 40 F 4 High School   
## 5 709106358 Existing Custo… 40 M 3 Uneducated   
## 6 713061558 Existing Custo… 44 M 2 Graduate   
## 7 810347208 Existing Custo… 51 M 4 Unknown   
## 8 818906208 Existing Custo… 32 M 0 High School   
## 9 710930508 Existing Custo… 37 M 3 Uneducated   
## 10 719661558 Existing Custo… 48 M 2 Graduate   
## # ℹ 10,117 more rows  
## # ℹ 15 more variables: Marital\_Status <chr>, Income\_Category <chr>,  
## # Card\_Category <chr>, Months\_on\_book <int>, Total\_Relationship\_Count <int>,  
## # Months\_Inactive\_12\_mon <int>, Contacts\_Count\_12\_mon <int>,  
## # Credit\_Limit <dbl>, Total\_Revolving\_Bal <int>, Avg\_Open\_To\_Buy <dbl>,  
## # Total\_Amt\_Chng\_Q4\_Q1 <dbl>, Total\_Trans\_Amt <int>, Total\_Trans\_Ct <int>,  
## # Total\_Ct\_Chng\_Q4\_Q1 <dbl>, Avg\_Utilization\_Ratio <dbl>

# Overview

ABC Corporation is in need of a supervised classification model which will predict the probability of customer attrition based on demographic, behavioral and service-related features. Target (dependent variable) is risk of attrition, represented by a value between 0 and 1 which indicates probability that a customer will terminate services rendered by ABC Corporation.

The expected outcomes of these preliminary results are the following:

1. ***Analytic***: Trained classification model which predicts customer attrition based on selected features with a high degree of accuracy. Determination as to which features are most influential in predicting customer attrition.
2. ***Informational***: Development of insights as to customer behaviors and demographics associated with customer attrition.
3. ***Model Usage***: Prediction of likelihood of customer attrition for both new and existing customers. Those identified as likely of attrition can then be targeted via direct advertising or incentives, etc. so as to prevent said attrition. This will lead to increased customer retention, and subsequently increased revenue for ABC Corporation.

## Data Dictionary

Here we display the data dictionary for this dataset, which includes each variable and a description.

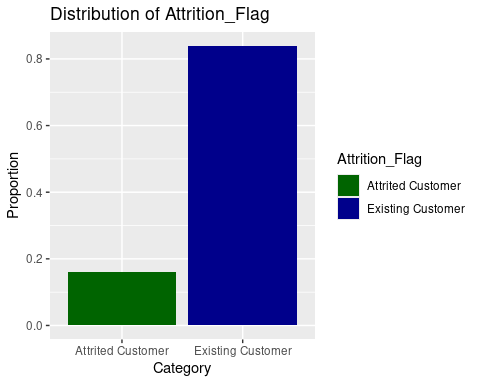
## Rows: 21 Columns: 2  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: "\t"  
## chr (2): Variable, Definition  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## # A tibble: 21 × 2  
## Variable Definition   
## <chr> <chr>   
## 1 CLIENTNUM Client number. Unique identifier for the customer holding th…  
## 2 Attrition\_Flag Internal event (customer activity) variable - if the account…  
## 3 Customer\_Age Demographic variable - Customer's Age in Years   
## 4 Gender Demographic variable - M=Male, F=Female   
## 5 Dependent\_count Demographic variable - Number of dependents   
## 6 Education\_Level Demographic variable - Educational Qualification of the acco…  
## 7 Marital\_Status Demographic variable - Married, Single, Divorced, Unknown   
## 8 Income\_Category Demographic variable - Annual Income Category of the account…  
## 9 Card\_Category Product Variable - Type of Card (Blue, Silver, Gold, Platinu…  
## 10 Months\_on\_book Period of relationship with bank   
## # ℹ 11 more rows

Client Number is a unique identifier, and thus isn’t useful as predictor of probability of attrition, we will drop this column before modeling.

## Classification Task

First, let us consider the overall classification task - namely, to predict customer attrition. To that end, let us inspect the distribution of attrited vs non-attrited (“existing”) customers.



# Data Processing

In this section we perform any steps necessary to prepare the data

## Remove Problematic Columns or Rows

Early investigation determined that errors and duplicates did not appear to be an issue, so we only need to remove the CLIENTNUM column.

## # A tibble: 10,127 × 20  
## Attrition\_Flag Customer\_Age Gender Dependent\_count Education\_Level  
## <chr> <int> <chr> <int> <chr>   
## 1 Existing Customer 45 M 3 High School   
## 2 Existing Customer 49 F 5 Graduate   
## 3 Existing Customer 51 M 3 Graduate   
## 4 Existing Customer 40 F 4 High School   
## 5 Existing Customer 40 M 3 Uneducated   
## 6 Existing Customer 44 M 2 Graduate   
## 7 Existing Customer 51 M 4 Unknown   
## 8 Existing Customer 32 M 0 High School   
## 9 Existing Customer 37 M 3 Uneducated   
## 10 Existing Customer 48 M 2 Graduate   
## # ℹ 10,117 more rows  
## # ℹ 15 more variables: Marital\_Status <chr>, Income\_Category <chr>,  
## # Card\_Category <chr>, Months\_on\_book <int>, Total\_Relationship\_Count <int>,  
## # Months\_Inactive\_12\_mon <int>, Contacts\_Count\_12\_mon <int>,  
## # Credit\_Limit <dbl>, Total\_Revolving\_Bal <int>, Avg\_Open\_To\_Buy <dbl>,  
## # Total\_Amt\_Chng\_Q4\_Q1 <dbl>, Total\_Trans\_Amt <int>, Total\_Trans\_Ct <int>,  
## # Total\_Ct\_Chng\_Q4\_Q1 <dbl>, Avg\_Utilization\_Ratio <dbl>

## Check Data Types

Inspecting the data types of the columns in the tibble above, that were inferred when the .csv was read, everything looks correct. The categorical variables were read in as string which makes sense, however we will convert the character (string) type columns to factor type (R’s native categorical data type), for better compatibility with other functions, features, etc.

## Rows: 10,127  
## Columns: 21  
## $ CLIENTNUM <int> 768805383, 818770008, 713982108, 769911858, 7…  
## $ Attrition\_Flag <fct> Existing Customer, Existing Customer, Existin…  
## $ Customer\_Age <int> 45, 49, 51, 40, 40, 44, 51, 32, 37, 48, 42, 6…  
## $ Gender <fct> M, F, M, F, M, M, M, M, M, M, M, M, M, M, F, …  
## $ Dependent\_count <int> 3, 5, 3, 4, 3, 2, 4, 0, 3, 2, 5, 1, 1, 3, 2, …  
## $ Education\_Level <fct> High School, Graduate, Graduate, High School,…  
## $ Marital\_Status <fct> Married, Single, Married, Unknown, Married, M…  
## $ Income\_Category <fct> $60K - $80K, Less than $40K, $80K - $120K, Le…  
## $ Card\_Category <fct> Blue, Blue, Blue, Blue, Blue, Blue, Gold, Sil…  
## $ Months\_on\_book <int> 39, 44, 36, 34, 21, 36, 46, 27, 36, 36, 31, 5…  
## $ Total\_Relationship\_Count <int> 5, 6, 4, 3, 5, 3, 6, 2, 5, 6, 5, 6, 3, 5, 5, …  
## $ Months\_Inactive\_12\_mon <int> 1, 1, 1, 4, 1, 1, 1, 2, 2, 3, 3, 2, 6, 1, 2, …  
## $ Contacts\_Count\_12\_mon <int> 3, 2, 0, 1, 0, 2, 3, 2, 0, 3, 2, 3, 0, 3, 2, …  
## $ Credit\_Limit <dbl> 12691.0, 8256.0, 3418.0, 3313.0, 4716.0, 4010…  
## $ Total\_Revolving\_Bal <int> 777, 864, 0, 2517, 0, 1247, 2264, 1396, 2517,…  
## $ Avg\_Open\_To\_Buy <dbl> 11914.0, 7392.0, 3418.0, 796.0, 4716.0, 2763.…  
## $ Total\_Amt\_Chng\_Q4\_Q1 <dbl> 1.335, 1.541, 2.594, 1.405, 2.175, 1.376, 1.9…  
## $ Total\_Trans\_Amt <int> 1144, 1291, 1887, 1171, 816, 1088, 1330, 1538…  
## $ Total\_Trans\_Ct <int> 42, 33, 20, 20, 28, 24, 31, 36, 24, 32, 42, 2…  
## $ Total\_Ct\_Chng\_Q4\_Q1 <dbl> 1.625, 3.714, 2.333, 2.333, 2.500, 0.846, 0.7…  
## $ Avg\_Utilization\_Ratio <dbl> 0.061, 0.105, 0.000, 0.760, 0.000, 0.311, 0.0…

## Missing Values

Preliminary EDA showed that missing values are in form of “Unknown,” which occurred in the columns Education\_Level, Marital\_Status\_Income, Income\_Category, all of which are categorical variables.

The total number of rows which contained the value “Unknown” in at least one of these columns was around 3,000, or approximately 30% of the data, the decision was made to leave “Unknown” present as an additional category in these columns, leading to the creation of an additional binary variable during one-hot encoding.

## Standardizing

As we will be fitting a k-nearest neighbors model, standardizing (namely, centering and scaling )is very important, since the learning algorithm is based on distances between points in a high dimensional space, it is important that each dimension (i.e. axis) is measured on the same scale.

To make comparison between the k-nearest neighbors model and any other models meaningful, we will standardize the entire data set. Note that this will be accomplished in a recipe in the [Preprocessing](#preproccessing) section below.

# Feature Engineering

Here we create or discuss the later creation of new variables, known as feature engineering.

## One Hot Encoding

All categorical variables will be converted to numerical binary variable using one-hot encoding. We have stored all such variables as fct (factor) type, here are the names

## [1] "Attrition\_Flag" "Gender" "Education\_Level" "Marital\_Status"   
## [5] "Income\_Category" "Card\_Category"

## New Features:

Here we create some new columns which are ratios of previous columns. The particular choices are guided by intuition for relevant new features describing customer behavior which is relevant to attrition.

# Preprocessing

Here we prepare cleaned data for modeling

## Train Test Split

## Train set dimensions: 7595 24

## Test set dimensions: 2532 24

## Preprocessing Recipe

We perform the following preprocessing steps

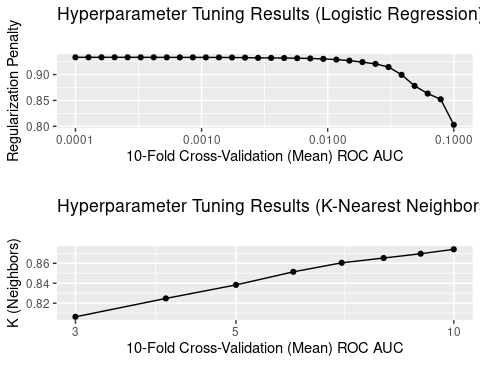
1. step\_other - this pools infrequently occurring values of categorical variables into another category called "other"
2. step\_dummy - this one-hot encodes all categorical variables.
3. step\_center - this centers all variables (which are all numeric by this stage in the processing pipeline) by subtracting them from their mean.
4. step\_scale - this scales all variables by dividing them by their standard deviation.

# Hyperparameter Tuning with Cross-Validation on Train Data

In this section we tune the hyperparameters classifier models, logistic regression (a parametric model) and k-nearest neighbors (a non-parametric model).

We use 10-fold cross-validation step when fitting the model to the training data, to find improved hyperparameters for each, namely the regularization penalty for logistic regression, and the number of neighbors for k-nearest neighbors. Optimization is with respect to the area under the receiver operating characteristic curve (ROC-AUC), a good metric for a balanced classifier with an

Here is a plot of the hyperparameter values versus the mean ROC-AUC score over all 10 cross-validation folds.



## # A tibble: 10 × 5  
## Model Hyperparameter Hyperparameter\_Value Mean\_CV\_ROC\_AUC Std\_Err\_CV\_ROC\_AUC  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Logis… Penalty 0.000161 0.933 0.00459  
## 2 Logis… Penalty 0.0001 0.933 0.00457  
## 3 Logis… Penalty 0.000204 0.933 0.00461  
## 4 Logis… Penalty 0.000127 0.933 0.00457  
## 5 Logis… Penalty 0.000259 0.933 0.00462  
## 6 K-Nea… Neighbors 10 0.874 0.00477  
## 7 K-Nea… Neighbors 9 0.870 0.00480  
## 8 K-Nea… Neighbors 8 0.865 0.00490  
## 9 K-Nea… Neighbors 7 0.860 0.00503  
## 10 K-Nea… Neighbors 6 0.851 0.00539

# Evaluate Tuned Models

In this section we fit final tuned models using the best hyperparameters obtained from cross-validation in the last section.

## Collect Predictions

## # A tibble: 2,532 × 3  
## Attrition\_Flag .pred\_Attrited.Customer .pred\_Existing.Customer  
## <chr> <dbl> <dbl>  
## 1 Existing Customer 0.00000227 1.00   
## 2 Existing Customer 0.0439 0.956   
## 3 Existing Customer 0.000784 0.999   
## 4 Existing Customer 0.0427 0.957   
## 5 Existing Customer 0.0585 0.942   
## 6 Existing Customer 0.000415 1.00   
## 7 Existing Customer 0.00600 0.994   
## 8 Existing Customer 0.00827 0.992   
## 9 Attrited Customer 0.903 0.0971  
## 10 Existing Customer 0.0397 0.960   
## # ℹ 2,522 more rows

## # A tibble: 2,532 × 3  
## Attrition\_Flag .pred\_Attrited.Customer .pred\_Existing.Customer  
## <chr> <dbl> <dbl>  
## 1 Existing Customer 0 1   
## 2 Existing Customer 0 1   
## 3 Existing Customer 0 1   
## 4 Existing Customer 0 1   
## 5 Existing Customer 0.192 0.808  
## 6 Existing Customer 0.0302 0.970  
## 7 Existing Customer 0 1   
## 8 Existing Customer 0 1   
## 9 Attrited Customer 0.862 0.138  
## 10 Existing Customer 0 1   
## # ℹ 2,522 more rows

## Classifier Metrics

For this classification problem the classes are relatively unbalanced. For this reason, a good choice (and indeed, the 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)

## # A tibble: 2,532 × 3  
## Attrition\_Flag .pred\_Attrited.Customer .pred\_Existing.Customer  
## <chr> <dbl> <dbl>  
## 1 Existing Customer 0 1  
## 2 Existing Customer 0 1  
## 3 Existing Customer 0 1  
## 4 Existing Customer 0 1  
## 5 Existing Customer 0 1  
## 6 Existing Customer 0 1  
## 7 Existing Customer 0 1  
## 8 Existing Customer 0 1  
## 9 Attrited Customer 0 1  
## 10 Existing Customer 0 1  
## # ℹ 2,522 more rows

### Accuracy

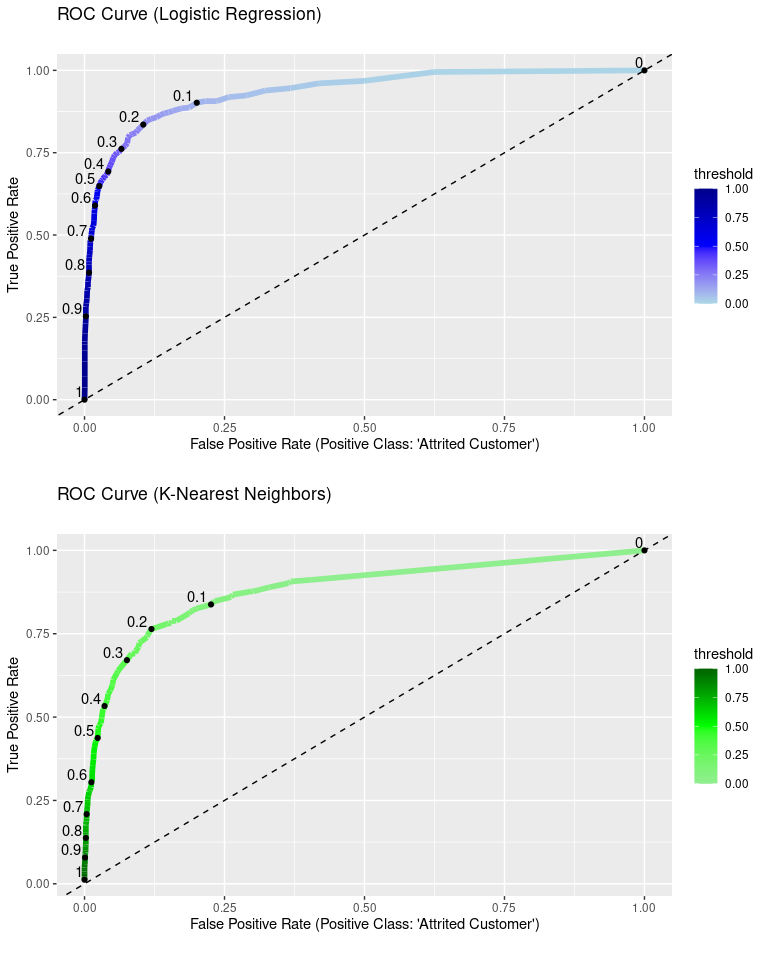
In this section we report the overall accuracy for the test data.

## # A tibble: 3 × 2  
## Model accuracy  
## <chr> <dbl>  
## 1 Null Model 0.839  
## 2 Logistic Regression 0.921  
## 3 K-Nearest Neighbors 0.890

In terms of overall accuracy, logistic regression 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.

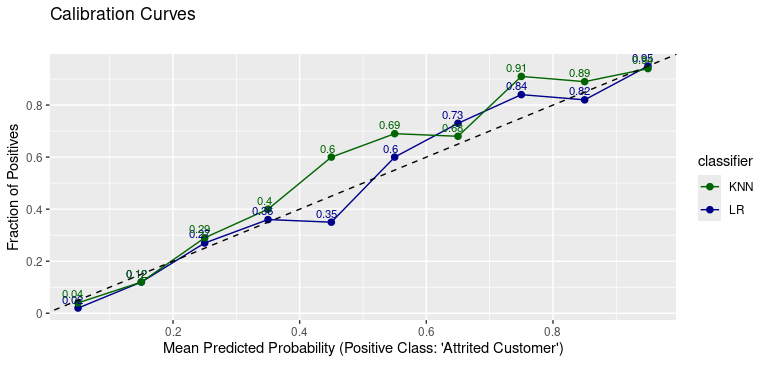
### ROC Curves



From these plots, we can see the tuned logistic regression classifier has a better area under the curve than the k-nearest neighbors classifier, as expected, since the logistic regression mean ROC AUC found during cross-validation was higher.

This shows that logistic regression is better at predicting the positive class at all probability thresholds, which indicates it is overall better suited to this data set as a predictive model.

### Calibration Curves

 Again, we see better results from logistic regression. Recall that the dashed line represents a perfect calibration, where the mean predicted probability of the positive class matches the true proportion of the positive class for each probability bin.

In this case, the k-nearest neighbors classifier is clearly regularly overestimating the positive class - for most bins, the fraction of predicted positive class is higher than expected for a perfect fit (where the expected probability is the midpoint of the probability bin) and higher than logistic regression.

# Results

It is pretty clear from these results a logistic regression model will provide a better model overall, both from the perspective of interpretation (since it is a parametric model) and from the perspective of prediction (since it performed better than K-Nearest neighbors on all metrics).

With respect to further pre-processing steps and feature engineering, binning some of the categorical features might be helpful, particularly since there are overall a large number of features. We will explore this in the final report.

We will also explore feature selection techniques in the final report, which will enable us to find further evidence of which features have a strong association with customer attrition, potentially leading to further feature engineering.