CREDIT CARD CUSTOMERS CHURN PREDITION MODEL

Thera Bank – July 2021

TABLE OF CONTENTS

01 **INTRODUCTION**

Introduces the scope of the study, major assumptions and prominent limitations

02 **KEY FINDINGS**

Outlines the major findings of the study and highlights key recommendations

03 **CUSTOMER DATA**

Describes the key variables examined from a sample of the bank's customers who hold credit cards with the bank

04 **METHODOLOGY**

Outlines how the logistic regression model was constructed and how it can be applied as a business tool

05 **RECOMMENDATIONS**

Demonstrates how to apply the key findings of the study to improve business strategy

06 **CONCLUSIONS**

Model evaluation overview and summary of key findings

01

INTRODUCTION

Brief overview and scope of the study

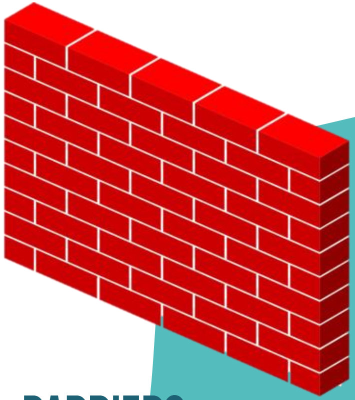
OUR COMPANY

Thera Bank is a financial institution which enjoyed a relatively stable, traditional customer base. Credit cards make up the cornerstone of the bank's profitability. Bank-issued credit cards are a good source of income because of different kinds of fees charged by the banks like annual fees, balance transfer fees, and cash advance fees, late payment fees, foreign transaction fees, and others. Some fees are charged on every user irrespective of usage, while others are charged under specified circumstances.

Thera Bank
A FINANCIAL
COMPANY

THE PROBLEM:

The Thera bank recently saw a steep decline in the number of users of their credit card and the Bank is unable to identify customers with accounts that are at risk of attrition. Therefore, they are unable to stop customer erosion.



CREDIT CARD CUSTOMERS

Thera Bank's customers of interest are credit card users

BARRIERS

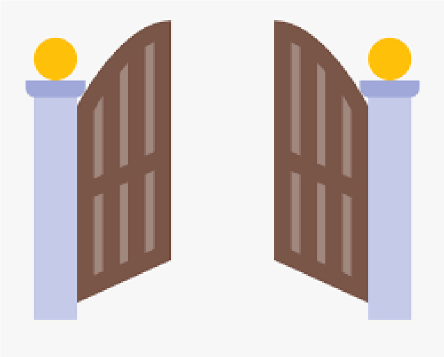
Barriers include inability to directly target the customer segments at risk of account attrition

ACCOUNT CLOSURES

These barriers are resulting in a loss of customers, leading to a loss in profitability due to account closures

THE SOLUTION:

A classification model that will help the bank improve its services so that customers do not renounce their credit cards



CREDIT CARD CUSTOMERS

Loyal customers continue using their credit cards because they are happy with the bank's services

SOLUTIONS

Identifying at-risk customers and developing bespoke strategies to keep them happy

HAPPY CUSTOMERS

Happy customers will continue to use Thera's credit cards, incurring fees in the process

MAJOR ASSUMPTIONS

Assumption 1

The cost of predicting that a customer's account is at risk of attrition when it is actually not is not known. This phenomenon is considered to be a False Positive. It is assumed that the bank will take steps to encourage customers at risk of attrition to continuing using their cards, but will not go so far as to spend more on enticements than it would receive in return if the customer continues to use the card. Therefore, the assumption is that False Negatives – failing to identify a customer at risk of attrition – is more costly than False Positives. The cost of a False Negative is the loss of all the associated credit card fees when the account becomes attrited.

Assumption 2

All customers in the sample have cleared a pre-screening process, in which they have already met an acceptable risk tolerance threshold and risk of default is not an issue in this case. Customers are at risk of closing their account, but not defaulting on balances owed.

LIMITATIONS OF THE MODEL

Limitation 1

The user of the model will need to know most, if not all, of the relevant customer information used to build the model. XG Boosting can impute some missing values, but too many missing values may cause errors. Also, the precision of this model when imputing errors is unknown and can increase the likelihood of False Negatives and/or False Positives. As such, every effort should be made to find out all relevant information about a customer before attempting to run the prediction.

Limitation 2

The equation is only as good as its user and the underlying dataset. While the data has been thoroughly cleaned, small human error input mistakes would not be flagged during the cleaning process and could cause unintended branching of the decision tree, lowering the reliability of the model. Similarly, users of the model who input an incorrect value for one of the variables risks receiving an incorrect decision, due to the input error.

Limitation 3

The model will not work well if the original data is altered or changed. For example, if additional information is added to the dataset, the model would have to be reconstructed in order to compensate for these changes.

02

KEY FINDINGS

Overview of final model and highlights of key takeaways from the research

FINAL MODEL

The primary objective of our model was to minimize False Negatives.
Our final model reduced False Negatives to just 16 misclassifications.

95.49%

This is our test data accuracy rate

96.72%

This is our test data recall rate

OUR BEST MODEL

SIMPLE LOGISTIC REGRESSION? BAGGING OR BOOSTING? GRIDSEARCH OR RANDOMSEARCH?

We followed 5 approaches and constructed 32 distinct models. The first approach was a Simple Logistic Regression model that was modified by under sampling and over sampling followed by Simple, Ridge and Lasso regularization techniques. The second approach tried Decision Trees, Random Forests, Bagging and Boosting techniques with hyperparameter tuning. The third approach utilized pipelines to test GridSearch and RandomSearch cross validation techniques on XGBoost, AdaBoost and Gradient Boosting models. Our fourth approach repeated approach 3, except we used knn imputer to impute the unknown variables which we had previously cleaned manually. Approach 5 was a recreation of the previous 2 approaches, except we relied on OneHotEncoder instead of manually imputing the unknown variables or using knn imputer. We found that XGBoost tuned with GridSearch was the best performing model out of the 32 models tried. In the appendix, we then used ROC tuning to further improve our best model. The best model was the one that generalized well and had the highest Test Recall scores.

	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
XGBoost with ROC Tuning	0.969667	0.954919	0.987705	0.967213	0.848416	0.795953

BEST MODELS FROM EACH APPROACH

The best models from each approach were selected and placed in the chart below for comparison. The model labeled 'Best' is the best performing model from Approach 3, then tuned with ROC and fitted using Optimal Threshold. As we can see from the pink circles, the model created by Approach three had the highest test recall score and the lowest False Negatives. It was improved upon (green circles) using ROC to raise the recall score and lower False Negatives further.

	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision	False Negatives	Approach #
XGBoost with ROC Tuning	96.97%	95.49%	98.77%	96.72%	84.84%	79.60%	16	Best
Logistic Regression - Regularized (Oversampled)	85.71%	85.26%	86.37%	83.81%	85.25%	52.57%	79	1
Tuned XGBoost Classifier	97.00%	95.00%	97.00%	90.00%	87.00%	82.00%	47	2
XGBoost tuned with GridSearch (manually imputed)	91.61%	89.14%	98.42%	93.85%	66.02%	60.42%	30	3
XGBoost tuned with GridSearch (knn imputed)	92.48%	90.10%	98.33%	93.24%	68.54%	62.93%	33	4
XGBoost tuned with GridSearch (OHE imputed)	92.51%	90.33%	98.42%	93.65%	68.60%	63.47%	31	5

THE SIX KEY VARIABLES

1 Transaction Counts represents almost 35% of explaining the dependent variable. By itself, it explains more than any other variable. Not surprisingly, it has the strongest correlation with the dependent variable (-0.37).

2 Balance is the second most important feature to our best model. It explains almost 15% of the variance in the model. The two variables are moderately correlated (-0.26)

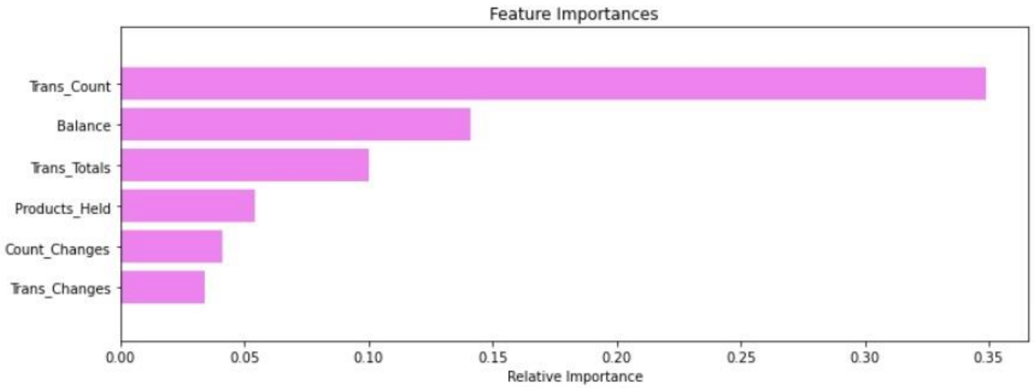
3 Transaction Totals ranks as the third most important variable to the model, explaining roughly 10% of all variance. It has a weak correlation with the dependent variable (-0.17).

4 Products Held explains just over 5% of the dependent variable. It also has a weak, negative correlation with the dependent variable (-0.15).

5 Count Changes explains about 4% of the dependent variable. It had the second highest correlation score with the dependent variable (-0.29)

6 Transaction Changes explains less than 4% of the dependent variable and the two are weakly correlated (-0.13).

THE SIX KEY VARIABLES



- The top six important features were consistent in Approaches 2 and 3, although Products Held surpassed Trans Totals in the best model found using Approach 2 techniques.
- Should the bank wish to simplify the model further, a good place to start would be to prune the variables listed near the bottom of the Feature Importance table: Dependents, Education, Income, Months, Contacts and Cards are the 6 least important features (ranked from least importance to more important).

ACTIONABLE INSIGHT

Be Wary of Low Values	The top six most important features ALL have negative correlations with the dependent variable. This means that, all other things held equal, a drop in any single of these columns will increase the likelihood of account attrition. .
Check Transaction Counts regularly	Transaction Counts explains over 35% of the dependent variable on its own. Customers with who are not using their cards are the most likely to allow their accounts to go into attrition. Target accounts with low transactions and encourage these customers to spend!
Flag Zero Balances	Balance is the second most important variable and relatively easy to monitor. Accounts that approach zero are more likely to go into attrition, all else being equal. Encourage customers to carry balances, even if it means slashing interest rates on outstanding accounts.
Error on the Side of Possible Attrition	The cost of a False Positive is negligible. Therefore, if ever in doubt, offer the customer an incentive. Even if the customer's account is not actually in danger of going into attrition, it may encourage the customer to spend more often and with higher transactions. This additional interest should more than off-set any 'sweeteners' issued to the customer.

03

CUSTOMER DATA

Examines the most important variables affecting whether a customer’s credit card account remains open or falls into attrition

CUSTOMER DATASET INFORMATION

NUMBER OF CUSTOMERS:

10,127

DISTINCT VARIABLES:

21

KEY NUMERICAL VARIABLES

- CLIENTNUM: Client number. Unique identifier for the customer holding the account
- Attrition_Flag: Internal event (customer activity) variable - if the account is closed then 1 else 0
- Customer_Age: Age in Years
- Gender: Gender of the account holder
- Dependent_count: Number of dependents
- Education_Level: Educational Qualification of the account holder
- Marital_Status: Marital Status of the account holder
- Income_Category: Annual Income Category of the account holder
- Card_Category: Type of Card
- Months_on_book: Period of relationship with the bank
- Total_Relationship_Count: Total no. of products held by the customer
- Months_Inactive_12_mon: No. of months inactive in the last 12 months
- Contacts_Count_12_mon: No. of Contacts in the last 12 months
- Credit_Limit: Credit Limit on the Credit Card
- Total_Revolving_Bal: Total Revolving Balance on the Credit Card
- Avg_Open_To_Buy: Open to Buy Credit Line (Average of last 12 months)
- Total_Amt_Chng_Q4_Q1: Change in Transaction Amount (Q4 over Q1)
- Total_Trans_Amt: Total Transaction Amount (Last 12 months)
- Total_Trans_Ct: Total Transaction Count (Last 12 months)
- Total_Ct_Chng_Q4_Q1: Change in Transaction Count (Q4 over Q1)
- Avg_Utilization_Ratio: Average Card Utilization Ratio

CORRELATIONS

RELATIONSHIPS BETWEEN VARIABLES

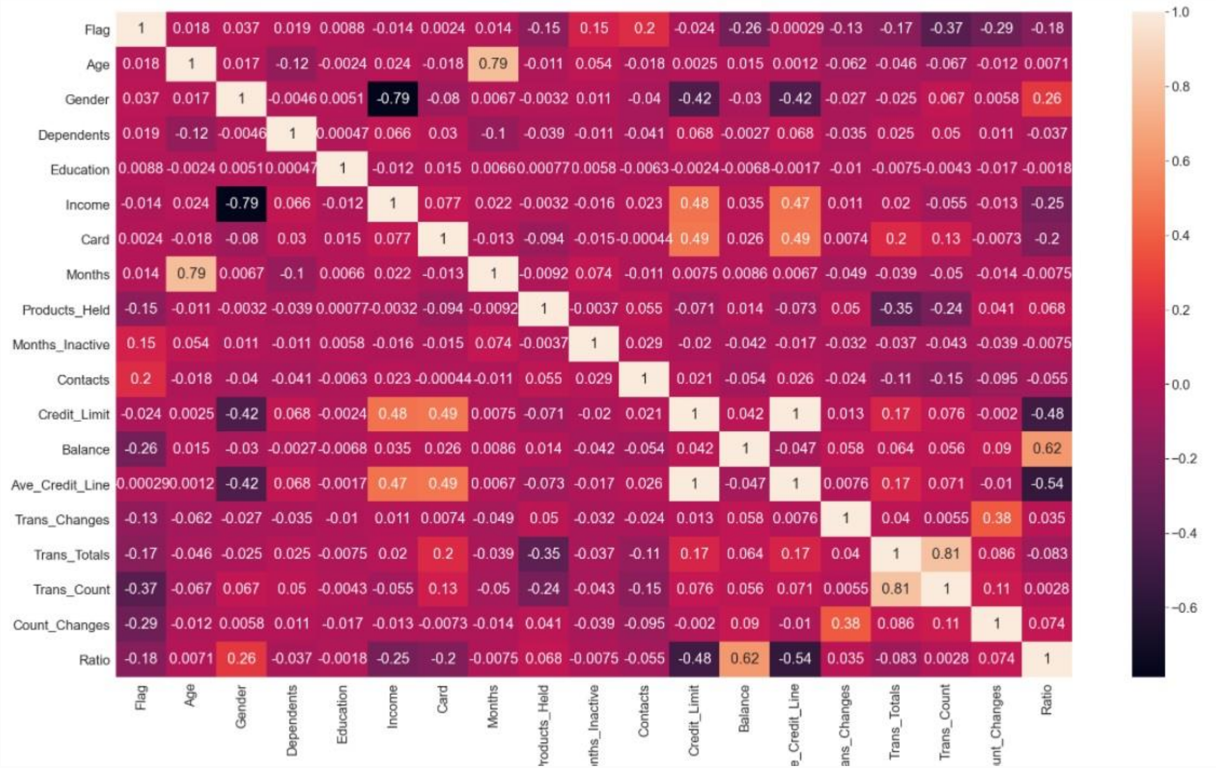
The independent variables in this study have varying degrees of correlation with one another. -1.0 denotes perfect negative correlation, 1.0 is perfect positive correlation. For this study, we define -0.09 to 0.09 as no correlation, 0.10 to 0.24 as a weak, positive correlation, 0.25 to 0.59 as a moderate, positive correlation, and 0.60 to 0.99 as a strong, positive correlation. Similarly, on the negative side, -0.09 to .-0.24, -0.25 to -0.59 and -0.60 to -0.99 represent weak, moderate and strong negative relationships, respectively.

KEY RELATIONSHIPS

The most important relationships are between the dependent variable and the independent variable. The dependent variable, Flag Attrition, was most positively correlated with Contacts and Months Inactive (albeit weakly), while negatively correlated with Transaction Counts, Count Changes and Balance. We also found that there was virtually zero (as defined above) correlation between Flag Attrition and the following variables: Credit Limit, Average Open To Buy, Months On Book, Customer Age, and Dependent Count. As such, these variables were dropped from the final model. We also saw that Credit Limit and Average Open to Buy were perfectly correlated. This results in multicollinearity and so one of these variables should be dropped, however, since neither had any correlation with Attrition Flag, this problem was avoided since both variables were dropped. See following slides for addition info...

CORRELATIONS

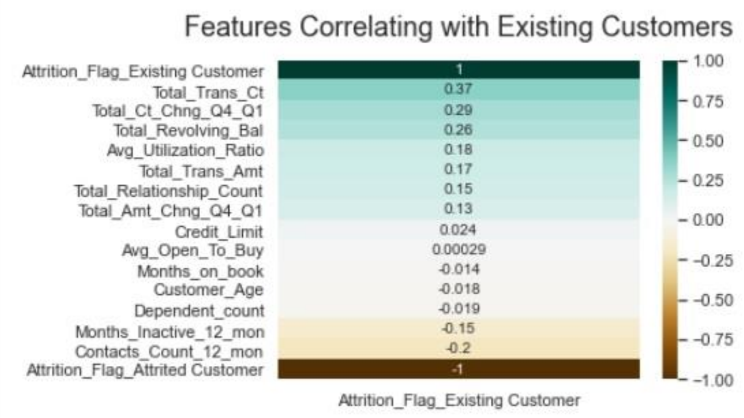
RELATIONSHIPS BETWEEN VARIABLES – HEAT MAP



CORRELATIONS

DEPENDENT VARIABLE

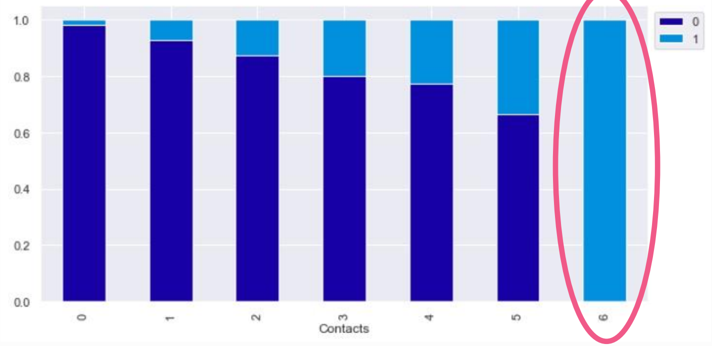
In our model, Attrition Flag is the dependent model. Thera Bank is interested in predicting customers whose accounts are at risk of attrition. We can see that when Attrition Flag = 0, there is moderate, positive correlations with Total Transaction Counts, Total Change in Transaction Count and Total Revolving Balance. When Attrition Flag = 1, we see weak, positive correlations with Contacts and Months Inactive. These last two features are of particular interest to us, because they can be used as goals to ensure customers keep their accounts open. Features with no correlations were dropped.



KEY POSITVE CORRELATIONS

Attrition Flag and Contacts had the highest positive correlation at 0.20. This marks a weak correlation. The graph to the right shows that once attrition goes up as more contacts are made. Once 6 contacts are made, there is 100% change of attrition.

1



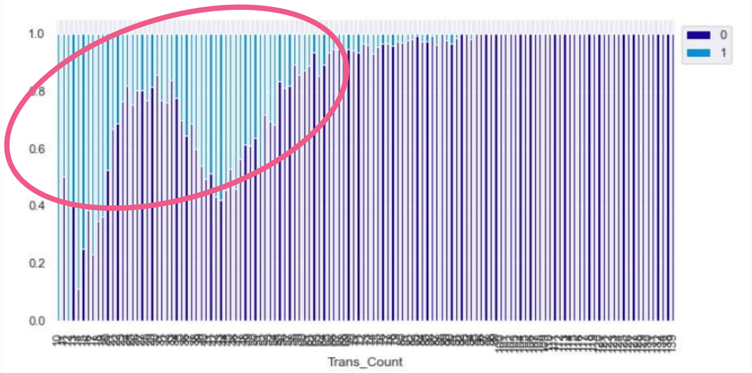
Attrition Flag and Months inactive had the second highest positive correlation at 0.15. This marks a weak correlation, but we can see from the graph on the right that there are peaks in months 2 and 3 of inactivity. Thera Bank needs to be especially vigilant during these sensitive months.

2

KEY NEGATIVE CORRELATIONS

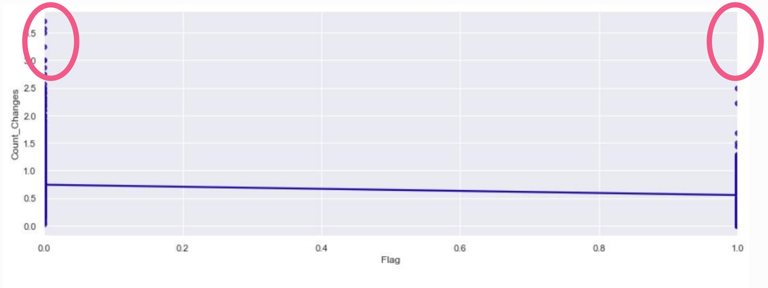
Attrition Flag and Transaction Counts had the highest negative correlation at 0.37. This marks a moderate correlation. A bit hard to read, but the graph to the right shows that most attrition happens on the left side of the graph, where transaction counts are lower.

1



Attrition Flag and Count Changes had the second highest negative correlation at 0.29. This marks a moderate correlation. With so many data points, its hard to get a good visual, but the graph on the right shows the negative correlation slope and shows that there are no data points above 2.5 for accounts in attrition, whereas existing accounts range as high as ~3.7.

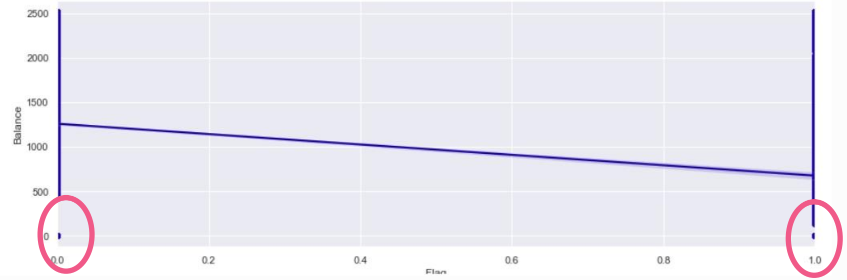
2



KEY NEGATIVE CORRELATIONS

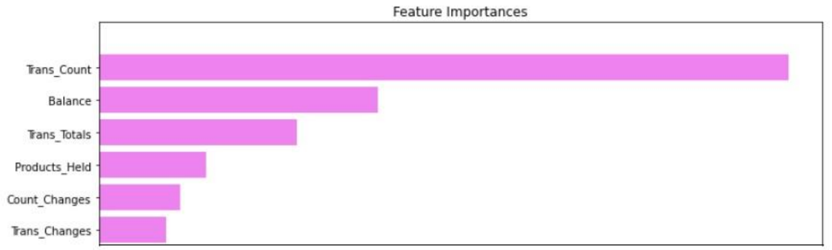
Attrition Flag and Balance had the third highest negative correlation at 0.26. This marks a moderate correlation. The graph to the right shows right shows the negative correlation slope and shows that there are very few data points below 500 for existing accounts, whereas the range from 0 to 500 is much more populated for accounts in attrition. .

3



KEY OBSERVATION

The three variables with the strongest, negative correlations with Flag Attrition composed the top five most important features of the final model. The remaining variables making up the most important features also had negative correlations with Flag Attrition, albeit slightly weaker.



04

METHODOLOGY

Building the best possible model for predicting whether customers accounts will remain open or fall into attrition

WHAT ARE WE ATTEMPTING TO DO?

Model can make wrong predictions as:

- Predicting a credit card account is closed when it is actually open - False Positive - no significant loss
- Predicting a credit card account is open when it is actually closed - False Negative - business loss

Which case is more important?

If the model predicts an account is closed when it really is not (FP), then there is no significant loss - the customer will continue to use the account and be charged fees and interest
If the model predicts an account is open when it really is closed (FN), the Bank is losing money in terms of fees and interest charges and is unable to take corrective action by reaching out to the customer to lure them back.

Which metric to optimize?

We would want Recall Score to be maximized. The greater the Recall Score, fewer accounts in attrition are being flagged as open, which is what we hope to minimize. Since F1 scores seek to balance False Negatives and False Positives, it would not necessarily minimize False Negatives. We do not recognize False Positives as being particularly costly, so we can ignore them, while minimizing False Negatives.

Approach 1: Logistic Regression

STEP 1:

Split data and build a simple logistic regression model

STEP 2:

Over sample using SMOTE

STEP 3:

Under sample using Random Under Sample

STEP 4:

Apply Regularization techniques – Saga, Ridge, Lasso

STEP 5:

Choose model with best recall and lowest False Negatives

Approach 2: Bagging and Boosting

STEP 1:

Partition the data into train and test set

STEP 2:

Build a Decision Tree Model

STEP 3:

Build Bagging models on the data sets

STEP 4:

Build Boosting models on the data sets

STEP 5:

Choose model with best recall and lowest False Negatives

Approach 3: Hyperparameter Tuning using GridSearchCV and RandomSearchCV Pipelines

*STEP 1:

Split the data into train and test set

STEP 2:

Use K-Fold cross-validation, StandardScaler and Pipelines to build different classifiers

STEP 3:

Find the best cross-validation score by tuning the model using hyperparameters

STEP 4:

Build the model and test its performance on the test set.

STEP 5:

Choose model with best recall and lowest False Negatives

*NOTE: Step 1 begins with data that has been cleaned and the unknown values for Income, Education and Marital Status have been imputed manually using EDA techniques

Approach 4: Reclassifying Unknown Values using KNN Imputer

*STEP 1:
Split the data into train and test set

STEP 2:
Use K-Fold cross-validation, StandardScaler and Pipelines to build different classifiers

STEP 3:
Find the best cross-validation score by tuning the model using hyperparameters

STEP 4:
Build the model and test its performance on the test set.

STEP 5:
Choose model with best recall and lowest False Negatives

*NOTE: Step 1 begins with data that has been cleaned and the unknown values for Income, Education and Marital Status have been imputed using knn imputer (nearest neighbour).

Approach 5: Reclassifying Unknown Values using OneHotEncoder

*STEP 1:

Split the data into train and test set

STEP 2:

Use K-Fold cross-validation, StandardScaler and Pipelines to build different classifiers

STEP 3:

Find the best cross-validation score by tuning the model using hyperparameters

STEP 4:

Build the model and test its performance on the test set.

STEP 5:

Choose model with best recall and lowest False Negatives

*NOTE: Step 1 begins with data that has been cleaned and the unknown values for Income, Education and Marital Status have been imputed using OneHotEncoder.

Further Tuning Best Model

STEP 1:

Best model from above approaches chosen

STEP 2:

ROC curve fitted to the test data

STEP 3:

Optimal Threshold is found

STEP 4:

Tune the model using the Optimal Threshold

STEP 5:

Evaluate recall scores and False Negatives to determine performance

Techniques to Evaluate Bagging Models

- n_estimators: The number of trees in the forest, default = 100.
- max_features: The number of features to consider when looking for the best split.
- class_weight: Weights associated with classes in the form {class_label: weight}.If not given, all classes are supposed to have weight one. The frequency of class 0 is 21.2% and the frequency of class 1 is 18.8% in the data to prevent 0 from becoming dominant, we can pass a dictionary {0:0.19,1:0.81} to the model to specify the weight of each class and the random forest will give more weightage to class 1.
- bootstrap: Determines if bootstrap samples are used when building trees. If False, the entire dataset is used to build each tree, default=True.
- max_samples: If bootstrap is True, then the number of samples to draw from X to train each base estimator. If None (default), then draw N samples, where N is the number of observations in the train data.
- oob_score: The out-of-bag (OOB) error is the average error for each observation calculated using predictions from the trees that do not contain that observation in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained, default=False.

Techniques to Evaluate Boosting Models

- base_estimator: The base estimator from which the boosted ensemble is built. By default the base estimator is a decision tree with max_depth=1 (AdaBoost)
- n_estimators: The maximum number of estimators at which boosting is terminated. (AdaBoost Default value is 50; Gradient Default value is 100)
- learning_rate: Learning rate shrinks the contribution of each classifier by learning_rate (Ada, Gradient, XGBoost)
- Init: An estimator object that is used to compute the initial predictions. If ‘zero’, the initial raw predictions are set to zero. By default, a DummyEstimator predicting the classes priors is used (Gradient)
- subsample: The fraction of samples to be used for fitting the individual base learner,default=1.0 (Gradient)
- max_features: The number of features to consider when looking for the best split, default=None (Gradient)
- gamma: A node is split only when the resulting split gives a positive reduction in the loss function. Gamma specifies the minimum loss reduction required to make a split. Higher the gamma value, lesser the chances of overfitting (XGBoost)
- scale_pos_weight: Control the balance of positive and negative weights. It can have any positive value as input. It helps in imbalanced classification problems (XGBoost)

Techniques to Evaluate Pipeline Models

The techniques used to evaluate the Boosting Models in Approach three are the same as the ones used on the previous slide. The reason for this, is that both Approach 2 and Approach 3 used Boosting techniques (Gradient Boosting, AdaBoost and XGBoost). In fact, both Approaches used GridSearch to tune the boosting models. The difference between Approach 2 and 3, however, was the use of a pipeline to implement the boosting and the hyperparameter tuning. Also, RandomSearchCV was introduced in Approach 3. RandomSearch uses the same estimators as GridSearch, they just apply them different. GridSearch uses an approach that tests every single estimator in the grid, sequentially, for best fit. RandomSearch uses a random approach, selecting different estimators indiscriminately and evaluating which is the best. In Approach 3, the following evaluation techniques were used (see previous slide for description):

- base_estimator
- n_estimators
- learning_ratesubsample
- max_features
- gamma
- scale_pos_weight

MISCLASSIFICATION ANALYSIS

False Negatives - Test Data

```
FN1 = TB5[(TB5['Flag'] == 1)&(TB5['Predicted'] == 0)] #Creates a dataframe of just the false negatives
```

```
FN1.shape
```

```
(16, 17)
```

We can build a dataset composed just of the misclassified values in the Test data of our best model. There were 16 entries in the Test data misclassification dataset, which is consistent with our findings when using the Confusion Matrix. Accuracy was found to be 95.5% and recall was 96.7% on the test data. We could then analyze the 16 misclassified customers to see if any patterns emerge.

05

RECOMMENDATIONS

Applying key findings to improve Thera Bank's business strategy

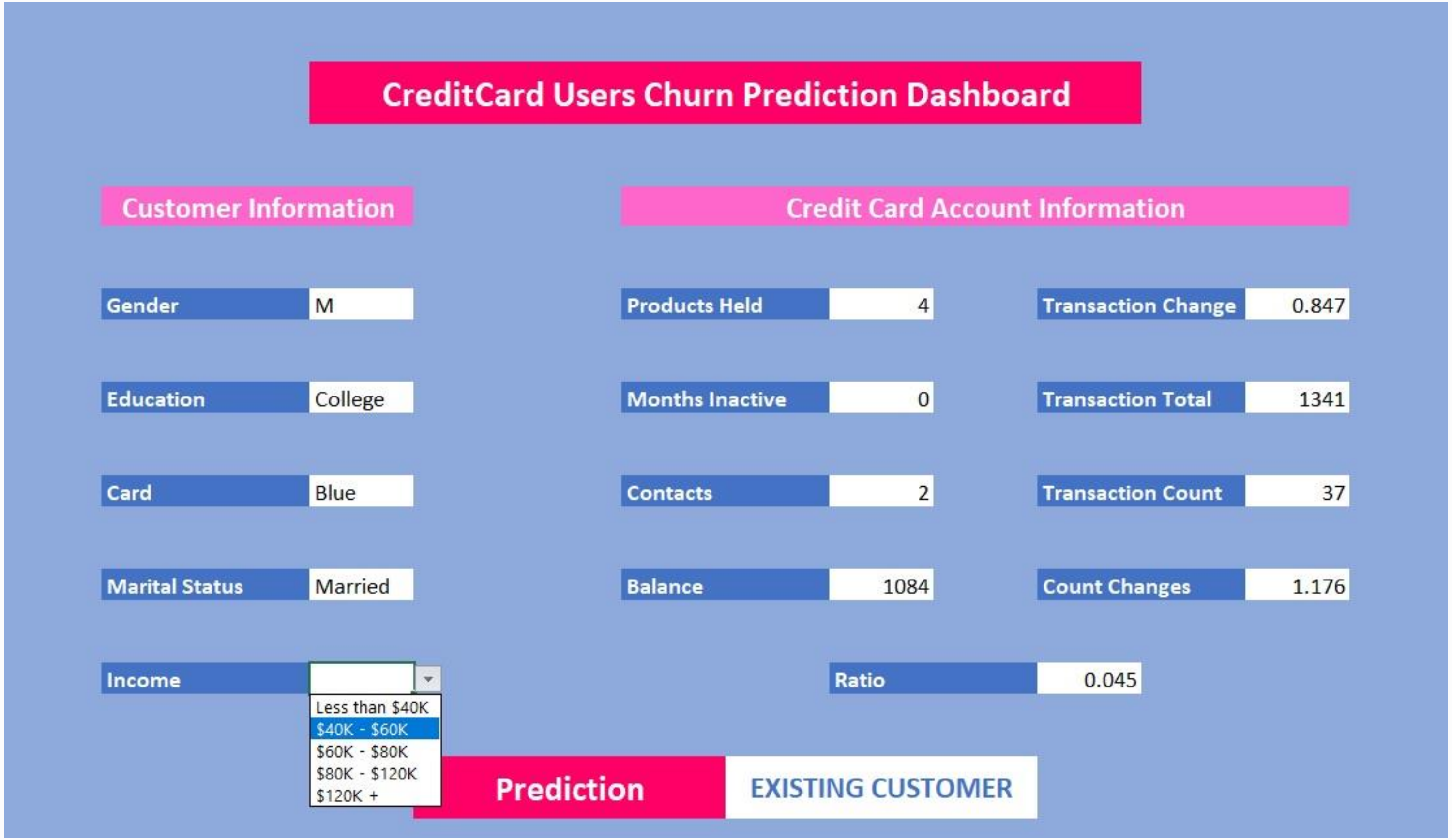
KEY ACTION ITEMS

	Design Dashboard	Improve Dataset	Evolve Model
Key Actions	Design a user-friendly dashboard where categorical variables can be selected from drop down list and numerical variables can be inputted.	The dataset has limitations, especially for underrepresented customers such as Silver, Gold and Platinum. By continuing to add to the dataset, distributions and samples should be come more and more normal and easier to model.	As more data is added, the structure of the model will change. The model will have to be re-run and altered, but this is an evolution which will only help to improve accuracy in the future.
Rationale	The ensemble technique method can be difficult for a user comprehend and follow decision-making steps to the logical conclusion. Bank employees would be trained to use the dashboard to ameliorate model application.	As more data is entered, the dataset becomes more robust. With more information, the model will generalize better without overfitting the data. False Negatives will continue to go down as more of the data is easily explained.	With more data, variables that are currently not helpful may turn out to provide key information as model evolves. We dropped several variables in the model, but they could ultimately prove to provide insight and increase accuracy.

KEY ACTION ITEMS

	Engage the Customer	Reward the Customer	Share the Wealth
Key Actions	Demonstrate to existing customers that it is worth their while to continue to use Thera Bank products, especially credit cards. Incentives or 'sweetners' may incentivize customers to use their cards more frequently and spend more.	Upgrade some existing cardholders from Blue to Silver. Lower interest rates. Give incentives for automatic billing on credit cards. Lower annual fees or other charges when spending targets are hit or when cards are used frequently.	Reinforce customer engagement and rewarding the customer by sharing some of the profits. Create a rewards program or points system that returns some of the profits generated by fees and interest charges back to the customer.
Rationale	16% of customers' credit card accounts are already in attrition. Customers with low balances and low transaction totals are most likely to close. Incentives like lower monthly fees or grace periods on interest charges may make customers feel more appreciated.	The data shows that Silver, Gold and Platinum card holders are less likely to allow their accounts to go into attrition. We have also seen that high transaction counts, transaction totals and regular monthly activity encourages accounts to remain open.	Sharing the wealth makes customers feel appreciated. By offering discounts, card upgrades or simple cashback rewards, less customers will close their accounts, leading to a higher volume of fees and interest, ultimately driving profits upward.

SAMPLE DASHBOARD



BUSINESS INSIGHTS

- Having a well-functioning model such as the XGBoost tuned with GridSearchCV is a powerful tool
- The bank can now easily identify customers who are at risk of their credit card account going into attrition and can take appropriate steps to prevent this
- By taking preventative steps early, the bank can avoid losing out on all the fees associated with the use of its credit cards
- The Bank can also avoid expensive marketing programs and other 'sweeteners' designed to lure customers back to the bank after their accounts have gone into attrition
- False Positives are negligible because any expense the Bank incurs on customers who are identified as at risk of attrition but are actually not, will probably be made back in the long run
- For example, if a customer is falsely identified as at risk, the Bank could lower their interest rates, or provide longer grace periods, which would result in a short-term loss in profit, but the customer is not really at risk of attrition so (s)he will continue to use the card, spending and frequency may even increase due to the 'sweetners' and the fees associated with continued use will increase profitability in the long-term
- The model is also able to identify keep attributes that help predict attrition: Transaction Counts, Balance and Transaction Totals being the top three
- These three attributes are negatively correlated to attrition, meaning, as one of these three variables decreases (all else remaining equal), the likelihood of attrition increases
- The bank now has key criteria to monitor, and can set up an internal system that automatically flags any attribute it feels is entering a pre-determined 'danger zone'
- For example, we saw that when Monthly Inactivity, another important attribute that is positively correlated with attrition, reaches 2 or 3 months, the account is at risk. The bank should set up a monitoring system that reaches out to customers when their account is inactive for more than a month
- Caution is needed, however, because we also saw that the number of Contacts a bank makes with customers is positively correlated to attrition. It seems customers do not like to be bothered excessively. Once a 6th contact is made, 100% of customer accounts are attrited. This could be due to internal bank policy, ie a maximum is set at 5 and only after the account enters attrition dues the bank make a 6th call to inform the customer of the change in status. Regardless, a better system needs to be contrived to encourage customers to continue to use the card without actually making contact with the customer. Further studies using customer surveys or focus groups may recommend notifications within online banking, email promotions or text messages instead of direct contacts. Further research is necessary.
- The best model has 15 attributes, a simple dashboard could be constructed - and is recommended - to facilitate the application of the model. Bank employees could simply take the customer information - ignoring any additional information outside of these 15 attributes, and enter it into an interface which would quickly return the prediction of whether the customer is at risk of attrition or not.

06

CONCLUSIONS

Model evaluation overview and summary of key findings

FINAL MODEL

After training the computer extensively, the model with the best recall score is:

	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
XGBoost with ROC Tuning	0.969667	0.954919	0.987709	0.967213	0.848416	0.795953

Our goal was to maximize the test recall scores, which would force False Negatives to a minimum. After experimenting uses different techniques and combinations, 32 distinct models were build, trained, tested and evaluated. In the end, the best model turned out to be the XGBoost tuned with GridSearchCV on the manually imputed unknown values dataset (in Approach 3). The recall score on the test data was 93.85%, the accuracy on the test data was 89.14% and it minimized False Negatives to 30.

We took this model and tuned it further with ROC tuning We were able to raise the recall score by almost 3%, accuracy by almost 6%, and further reduce False Negatives by 14… an 88% improvement!

DID THE MODEL IMPROVE?

Model	Accuracy Train	Accuracy Test	Recall Train	Recall Test
Simple Logistic Regression	90.79%	90.16%	60.23%	56.56%
Approach 1 – Best Model	85.71%	85.26%	86.37%	83.81%
Approach 2 – Best Model	97.00%	95.00%	97.00%	90.00%
Approach 3 – Best Model	91.61%	89.14%	98.42%	93.85%
Approach 4 – Best Model	92.48%	90.10%	98.33%	93.24%
Approach 5 – Best Model	92.51%	90.33%	98.42%	93.65%
Best Model tuned with ROC (FINAL MODEL)	96.97%	95.49%	98.77%	96.72%

The above table shows the substantial improvements as the model progressed to its final form. The Simple Logistic Model, a model that could be done quickly in MS-Excel, had relatively good accuracy but was not maximizing recall scores. As we were focused on recall, we were willing to sacrifice some accuracy for better recall scores. As a result, the best model of Approach 1 is a significant improvement over the base model. We can see that further gains are made using Approach 2 and Approach 3. Approach 4 and 5 use the same techniques as Approach 3, but the dataset used different imputation methods on unknown data. While they improved the accuracy scores, the recall dropped slightly. Approach 3 was deemed the best model and it was further tuned with ROC to create the final model. As illustrated above, the final model has the highest accuracy and recall scores on the test data .

DID THE MODEL IMPROVE?

Model	False Negatives (# of Customers)	False Negatives (Percentage)
Simple Logistic Regression	212	6.98%
Approach 1 – Best Model	79	2.06%
Approach 2 – Best Model	47	1.55%
Approach 3 – Best Model	30	0.99%
Approach 4 – Best Model	33	1.09%
Approach 5 – Best Model	31	1.02%
Best Model tuned with ROC (FINAL MODEL)	16	0.80%

Our stated goal was to reduce the number of False Negatives, since they represent the biggest cost to Thera Bank. We can see that the base model was not attempting to minimize False Negatives, and it had the highest number. We can see that approach 1 and 2 substantially lowered the False Negative counts. Approaches 3, 4 and 5 reduced False Negatives further, but a slight edge goes to Approach 3. This was the reason Approach 3 was selected for ROC tuning which allowed us to drop the False Negatives by almost 100% to just 16.

SUMMARY OF FINDINGS

- We were able to use the customer information dataset provided by the bank to construct a model to help predict credit card user churn
- The bank’s primary concern is loss of income associated with interest rates, user fees, etc when customer accounts fall into attrition
- With this concern in mind, a model was sought that focused on maximizing recall score, which would force False Negatives to a minimum
- Accuracy scores were a secondary concern, since it is still important that the model generalizes well and is not overfitting the data
- By looking at the spread of the test and train recall scores in combination with the spread of the test and train accuracy scores, we can determine with a certain degree of confidence whether the model is generalizing or overfitting the data

SUMMARY OF FINDINGS

Approach 1

- In the first approach, we used simple logistic regression. We then tried oversampling the minority (Flag: 1) data, followed by under sampling the majority (Flag:0) data.
- We found that oversampling the data gave us better results than the other two methods
- Next, we tried regularizing all three models using three different techniques: Simple Regularization using saga, Ridge and Lasso
- We found that the oversampled data regularized with saga gave us slightly better results than any other model
- Recall scores on the test data of this model were tied with the scores on the oversampling model, but accuracy had improved slightly, giving it a slight edge
- Recall score improved from a base score (no tuning) of 56.56% to 83.81% and False Negatives fell from a base score of 212 to 79

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
2	Logistic Regression on Oversampled data	0.857119	0.852583	0.863675	0.838115	0.852497	0.525707
3	Logistic Regression-Regularized (Oversampled d...	0.857371	0.853241	0.864011	0.838115	0.852687	0.527062
5	Logistic Regression-Regularized (Undersampled ...	0.839772	0.857519	0.841967	0.834016	0.838287	0.536232
4	Logistic Regression on Undersampled data	0.839772	0.854557	0.841967	0.829918	0.838287	0.530105
0	Logistic Regression	0.907872	0.901612	0.602283	0.565574	0.774266	0.760331
1	Logistic Regression-Regularized	0.908155	0.901283	0.601405	0.563525	0.776644	0.759669

SUMMARY OF FINDINGS

Approach 2

- For the second approach we used Bagging and Boosting techniques
- We ran a total of 12 combinations using various models and tuning techniques
- For Bagging, we tried: Bagging Classifier with default parameters, Tuned Bagging Classifier, Bagging Classifier with base estimator = LR, Random Forest (default parameters), Tuned Random Forest Classifier, and Random Forest with class weights
- Of these six models, Random Forest with class weights was the best model
- Test recall was 90% and test accuracy was 96%, False Negatives were lowered to 48
- However, due to the recall on the train set of 96%, we did fear the model was slightly overfitting the data
- For Boosting we tried: Gradient Boosting, AdaBoost and XGBoost and then all three again with tuned hyperparameters for 6 more models
- Of these six, the Tuned XGBoost Classifier performed best with test recall scores of 90% and 95% accuracy. False Negatives were lowered further to 47
- However, due to the recall on the train set of 97%, we did fear the model was slightly overfitting the data

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision	Train_F1-Score	Test_F1-Score
5	Tuned XGBoost Classifier	0.97	0.95	0.97	0.90	0.87	0.82	0.91	0.86
4	XGBoost Classifier	1.00	0.96	1.00	0.87	1.00	0.90	1.00	0.89
3	Tuned Gradient Boosting Classifier	0.99	0.97	0.95	0.86	0.97	0.92	0.96	0.89
1	Tuned AdaBoost Classifier	0.98	0.96	0.92	0.83	0.95	0.90	0.94	0.86
2	Gradient Boosting Classifier	0.98	0.96	0.89	0.82	0.95	0.92	0.92	0.86
0	AdaBoost Classifier	0.96	0.95	0.86	0.78	0.90	0.88	0.88	0.83

SUMMARY OF FINDINGS

Approach 3

- For the third approach we created pipelines to take our three Boosting models and hyper tune the parameters first using GridSearchCV and second using RandomSearchCV
- Of the six models, XGBoost tuned with GridSearchCV performed the best
- This model had test recall scores of almost 94%
- The accuracy on the test data dropped to 89%, however
- The number of False Negatives improved to just 30 and the gap between test and train recall scores was reduced to about 4.5%, meaning we were less worried that this model was overfitting the data
- Since our primary focus is on recall scores and reducing False Negatives, the slightly lower accuracy score was not a major concern
- The model, we felt, generalized well
- We considered this to be our best model of all 5 approaches

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
0	XGBoost tuned with GridSearchCV	0.916055	0.891412	0.984197	0.938525	0.660188	0.604222
1	XGBoost tuned with RandomizedSearchCV	0.909565	0.883185	0.984197	0.928279	0.642775	0.586028
4	Gradient Boosting tuned with GridSearchCV	0.980672	0.970385	0.922739	0.893443	0.955455	0.919831
5	Gradient Boosting tuned with RandomizedSearchCV	0.980672	0.970385	0.922739	0.893443	0.955455	0.919831
2	AdaBoost tuned with GridSearchCV	0.987726	0.964791	0.963126	0.891393	0.960595	0.889571
3	AdaBoost tuned with RandomizedSearchCV	0.987726	0.964791	0.963126	0.891393	0.960595	0.889571

SUMMARY OF FINDINGS

Approach 4

- We returned to the dataset before the unknown values in Marital Status, Income and Education were manually imputed using data preprocessing techniques and allowed the computer to impute the unknown values using a ‘nearest neighbour’ strategy called knn imputer
- After which, we applied the same approach on this altered data as approach 3
- Again, of the six models, XGBoost tuned with GridSearchCV performed the best
- This model had test recall scores of over 93% - a slight drop from Approach 3
- The accuracy on the test data improved by 1%, however
- The number of False Negatives worsened with 33
- The model, we felt, generalized well
- However, since there was a drop in recall score and False Negative performance, we consider this model to be inferior to the best model of Approach 3

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
0	XGBoost tuned with GridSearchCV	0.924802	0.900954	0.983319	0.932377	0.685435	0.629322
2	AdaBoost tuned with GridSearchCV	0.995485	0.974005	0.985075	0.926230	0.986807	0.913131
3	AdaBoost tuned with RandomizedSearchCV	0.995485	0.974005	0.985075	0.926230	0.986807	0.913131
1	XGBoost tuned with RandomizedSearchCV	0.913798	0.889437	0.985075	0.922131	0.653846	0.601604
4	Gradient Boosting tuned with GridSearchCV	0.989419	0.974663	0.952590	0.903689	0.981013	0.936306
5	Gradient Boosting tuned with RandomizedSearchCV	0.989419	0.974663	0.952590	0.903689	0.981013	0.936306

SUMMARY OF FINDINGS

Approach 5

- We returned to the dataset before the unknown values in Marital Status, Income and Education were manually imputed using data preprocessing techniques once again but this time we allowed OneHotEncoder to convert all the categorical variables to separate columns which were either on or off. We could then drop the 3 columns labelled ‘unknown’
- This strategy could give us more flexibility, since bank employees could run the model on customers who have missing information in these three columns
- After which, we applied the same approach on this altered data as approach 3
- Again, of the six models, XGBoost tuned with GridSearchCV performed the best
- This model had test recall scores of almost 94% - just 0.2% behind Approach 3
- The accuracy on the test data improved by 1%, however
- The number of False Negatives worsened with 31
- The model, we felt, generalized well
- However, since there was a drop in recall score and False Negative performance, we consider this model to be inferior to the best model of Approach 3

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
0	XGBoost tuned with GridSearchCV	0.925085	0.903258	0.984197	0.936475	0.686047	0.634722
1	XGBoost tuned with RandomizedSearchCV	0.912246	0.889766	0.985075	0.930328	0.649682	0.601325

SUMMARY OF FINDINGS

ROC Tuning of Best Model

- We selected the best model (Approach 3) and tuned it more finely with ROC tuning.
- Recall scores improved to 96.72%
- False Negatives dropped to 16 from a previous best of 30!
- Accuracy scores increased
- This is an improvement of almost 88%

Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
XGBoost tuned with GridSearchCV	0.916055	0.891412	0.984197	0.938525	0.660188	0.604222

	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
XGBoost with ROC Tuning	0.969667	0.954919	0.987709	0.967213	0.848416	0.795953

CHECKING LOGISTIC REGRESSION ASSUMPTIONS

Logistic regression does not make many of the key assumptions of linear regression. First, logistic regression does not require a linear relationship between the dependent and independent variables. Second, the error terms (residuals) do not need to be normally distributed. Third, homoscedasticity is not required. Finally, the dependent variable in logistic regression is not measured on an interval or ratio scale. However, there are still some assumptions applicable to logistic regression:

1) Binary logistic regression requires the dependent variable to be binary.

Attrition Flag

```
TB["Flag"].value_counts()
```

```
0     8500
1     1627
Name: Flag, dtype: int64
```

- After converting the dependent variable (Attrition Flag), Existing Customer: 0, Attrited Customer: 1, the dependent variable (Flag) is binary

2) Logistic regression requires the observations to be independent of each other. Observations should not come from repeated measurements or matched data

```
dupes = dataframe.duplicated()
sum(dupes)
```

```
0
```

- We know from EDA that none of the customer entries were duplicates

CHECKING LOGISTIC REGRESSION ASSUMPTIONS

3) Logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other.

- Credit Limit and Average Credit Line were perfectly correlated, but both columns had zero correlation with the dependent variable so were dropped before the model was constructed

4) Logistic regression typically requires a large sample size. A general guideline is that you need at minimum of 10 cases with the least frequent outcome for each independent variable in your model. The initial model started with 20 independent variables. Assuming the expected probability of our least frequent outcome is 10%, then we need a minimum sample size of 2000 (10*20 / .10).

```
dataframe.shape
(10127, 21)
```

- We know from EDA that are 10,127 rows of data, and this sufficiently meets this criteria of 2000.

* Note: the ID variable was not included in this calculation and was dropped prior to any EDA analysis.

MISCLASSIFICATION ANALYSIS

	Flag	Predicted	Gender	Education	Income	Card	Products_Held	Months_Inactive	Contacts	Balance	Trans_Changes	Trans_Totals	Trans_Count	Count_Changes	Ratio	Married	Single
1257	1	0	1	3.0	0	0	5	1	2	951	0.737	1096	27	0.588	0.174	0	1
1950	1	0	0	0.0	2	0	6	2	3	0	0.733	967	24	1.182	0.000	0	0
2109	1	0	0	3.0	4	0	3	3	0	0	0.995	1564	31	0.476	0.000	0	1
2411	1	0	0	0.0	4	0	5	2	4	787	0.763	1312	32	0.600	0.023	1	0
3810	1	0	0	1.0	2	0	6	3	3	0	0.432	1329	35	0.591	0.000	1	0
5660	1	0	0	3.0	3	0	3	3	3	321	0.760	2078	62	0.550	0.025	1	0
6125	1	0	0	0.0	3	2	6	3	2	1104	0.591	1989	47	0.741	0.032	0	1
6246	1	0	1	3.0	0	0	6	1	3	1647	0.513	1903	44	0.630	0.898	0	1
6284	1	0	1	4.0	0	0	5	2	3	0	0.865	2716	63	0.703	0.000	1	0
6434	1	0	0	0.0	3	0	4	2	3	0	0.500	2625	63	0.500	0.000	0	1
6667	1	0	0	3.0	1	0	5	3	2	2517	0.690	2516	64	0.641	0.435	1	0
7594	1	0	1	1.0	0	0	3	2	4	0	0.838	3121	56	0.867	0.000	0	1
7782	1	0	1	0.0	1	0	2	3	2	965	0.928	3300	58	0.568	0.306	0	0
8967	1	0	0	2.0	3	0	4	3	3	0	0.890	4520	63	0.370	0.000	0	0
9014	1	0	1	2.0	1	0	2	3	1	0	1.005	5242	74	0.574	0.000	0	1
9142	1	0	1	1.0	1	0	4	3	3	0	0.494	4399	54	0.862	0.000	0	1

There were 16 instances where the model predicted the customer would continue using the bank's credit cards but the account went into attrition (pink rectange). This is considered a False Negative and we sought to minimize their occurrences. We can use this information to look for patterns….

MISCLASSIFICATION ANALYSIS

	Flag	Gender	Education	Income	Card	Products_Held	Months_Inactive	Contacts	Balance	Trans_Changes	Trans_Totals	Trans_Count	Count_Changes	Ratio	Married	Single	Predicted
1257	1	1	3.0	0	0	5	1	2	951	0.737	1096	27	0.588	0.174	0	1	0
1950	1	0	0.0	2	0	6	2	3	0	0.733	967	24	1.182	0.000	0	0	0
2109	1	0	3.0	4	0	3	3	0	0	0.995	1564	31	0.476	0.000	0	1	0
2411	1	0	0.0	4	0	5	2	4	787	0.763	1312	32	0.600	0.023	1	0	0
3810	1	0	1.0	2	0	6	3	3	0	0.432	1329	35	0.591	0.000	1	0	0
5660	1	0	3.0	3	0	3	3	3	321	0.760	2078	62	0.550	0.025	1	0	0
6125	1	0	0.0	3	2	6	3	2	1104	0.591	1989	47	0.741	0.032	0	1	0
6246	1	1	3.0	0	0	6	1	3	1647	0.513	1903	44	0.630	0.898	0	1	0
6284	1	1	4.0	0	0	5	2	3	0	0.865	2716	63	0.703	0.000	1	0	0
6434	1	0	0.0	3	0	4	2	3	0	0.500	2625	63	0.500	0.000	0	1	0
6667	1	0	3.0	1	0	5	3	2	2517	0.690	2516	64	0.641	0.435	1	0	0
7594	1	1	1.0	0	0	3	2	4	0	0.838	3121	56	0.867	0.000	0	1	0
7782	1	1	0.0	1	0	2	3	2	965	0.928	3300	58	0.568	0.306	0	0	0
8967	1	0	2.0	3	0	4	3	3	0	0.890	4520	63	0.370	0.000	0	0	0
9014	1	1	2.0	1	0	2	3	1	0	1.005	5242	74	0.574	0.000	0	1	0
9142	1	1	1.0	1	0	4	3	3	0	0.494	4399	54	0.862	0.000	0	1	0

It is difficult to see clear patterns. At first glace, customers identified as False Negatives overwhelmingly hold Blue cards (0), yet this is consistent with the original data set (93.75% vs 93.2%). Products Held range between 3 and 6, but the original data set median is 4. Months Inactive are mostly 2s and 3s… we know this is the 'danger zone' for attrition amongst this category. Transaction Counts are all lower than the median of 67, save one, and *most* of the Balance values are below the mean of 1162 (14/16). These two variables are negatively correlated with Flag, so they are more likely to be accounts in attrition the lower these numbers go.

Key Recommendation: Whenever a customer is predicted to remain an existing customer, check these three variables. Low Transaction Counts, Balances and accounts that have been inactive for 2 months should be monitored closely for other signs of attrition.

THANK YOU!

Do you have any questions?
mullin.scott@gmail.com
+91 620 421 838