Predicting Credit Card Customer Churn

Advanced Machine Learning

David Lundvall
01/17/25

Contents / Agenda

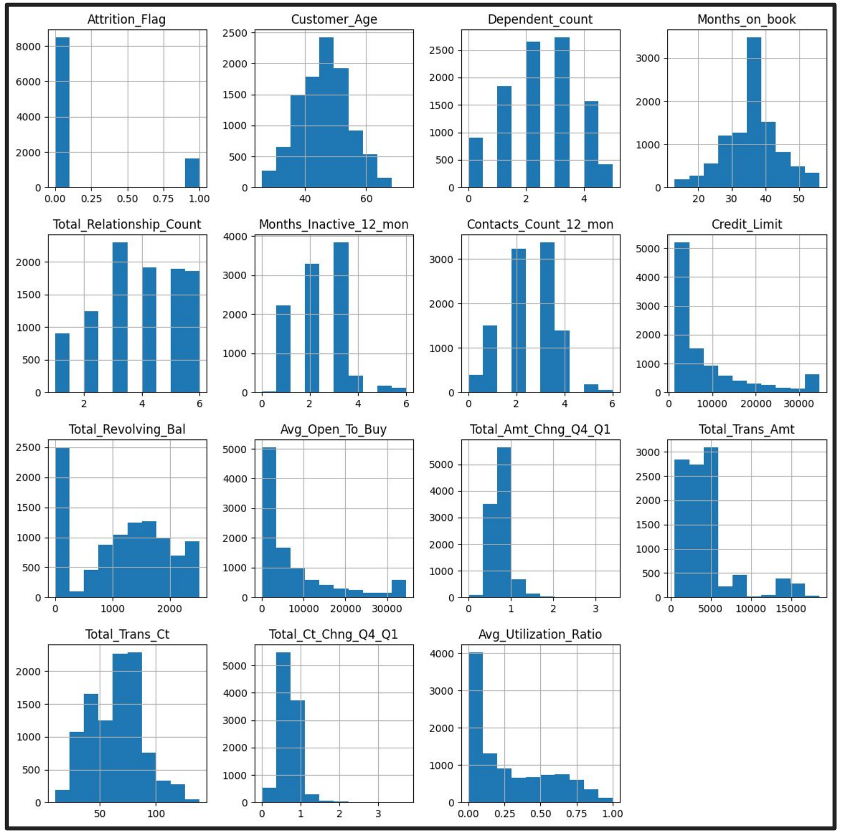
Executive Summary

- **Insights**
  - The business shared that it's of the utmost importance to identify and not miss any customers at risk of attrition so that proactive measure can be put in place to retain these customers
  - The main factors that lead to attrition were identified as a customer's low total transaction count, low total transaction amount, and low total revolving balance
- **Recommendations**
  - Identify the attrition-at-risk set of customers that are using their cards infrequently (or not at all) and that carry a low balance from month to month
  - Near-term mitigation: Proactively reach out to these set of customers and offer them 5% cash back on all purchases over the next few months to get them to start using their cards
  - Long-term solution: Create a card loyalty campaign that enables customers to earn cash back and/or loyalty points that can be used toward purchases such as travel or products from retail partners; this will encourage and strengthen a longer term relationship

Business Problem Overview and Solution Approach

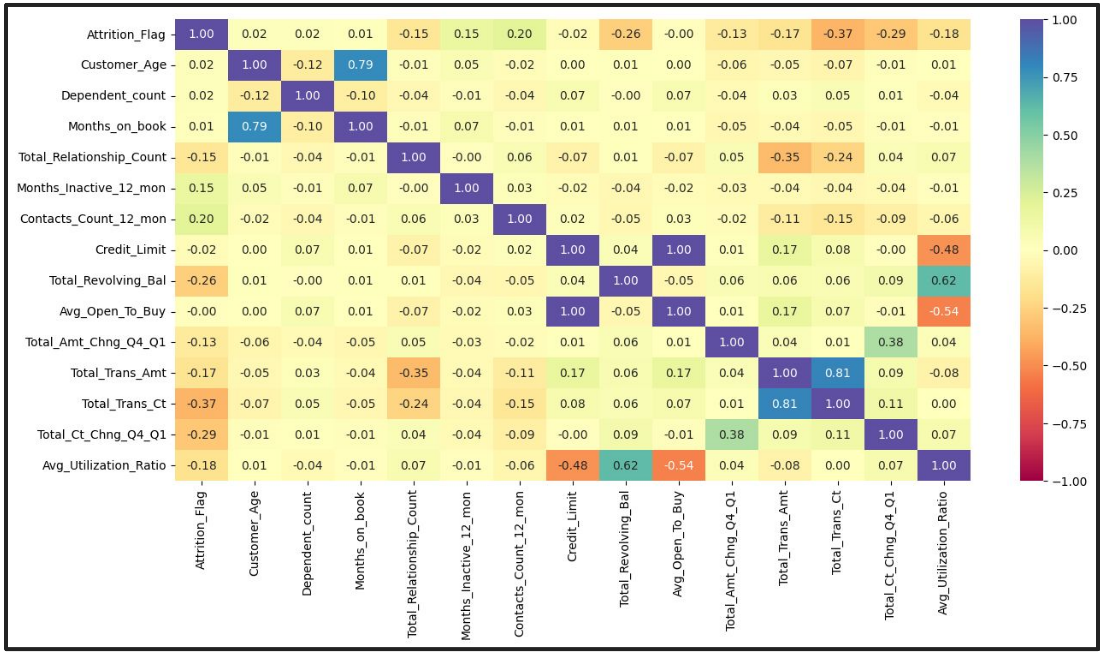
- **Business Problem**
  - Thera Bank recently saw a dramatic increase in customers attriting from their credit cards, and since credit cards are a good source of revenue, any decrease negatively affects the bottomline
  - The bank wants to identify existing customers that have similar traits as those that attrited so that an retention plan can be put in place and for those at-risk customers
- **Solution Approach**
  - The bank's customer data, which includes current and attrited customers, was explored via univariate & bivariate analysis to understand the data and fix any issues before modeling
  - The data was divided into train, validation, and test datasets and prepared with no data leakage
  - Focus was put on creating a model with the best recall so that no at-risk customers are missed
  - Several classifiers were used to train and validate the models and the results were compared; these models were then validated with undersampled and oversampled data sets
  - Techniques were used for hyperparameter tuning and the best models were then compared against the validation data to determine the best model to use; the best model was then compared using the test data

EDA Results: Univariate Analysis



- **Summary:** Most of the features have normalized distributions and/or the data looks accurate, however here are some additional observations worth noting
- **Data Anomalies:** Months_on_book count is heavily concentrated near the 36-month mark. Heavy customer acquisition, data-entry errors, or imputed? Additional input is needed from the business.
- **Right-Skewed:** Credit_Limit, Total_Revolving_Bal, Ave_Open_To_Buy, Total_Trans_Amt, and Avg_Utilization_Ratio
- **High % of Outliers:** 9-10% for Credit_Limit, Avg_Open_To_Buy, Total_Trans_Amount
- **Missing Values:** Education_Level and Marital_Status
- **Inaccurate Data Values:** Income_Category
- **Much Higher Density Count for a Category Feature:** Card_Category: Blue

EDA Results: Bivariate Results



	Attrition_Flag	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Revolving_Bal	Avg_Open_To_Buy	Total_Amt_Chng_Q4_Q1	Total_Trans_Amt	Total_Trans_Ct	Total_Ct_Chng_Q4_Q1	Avg_Utilization_Ratio
Attrition_Flag	1.00	0.02	0.02	0.01	-0.15	0.15	0.20	-0.02	-0.26	-0.00	-0.13	-0.17	-0.37	-0.29	-0.18
Customer_Age	0.02	1.00	-0.12	0.79	-0.01	0.05	-0.02	0.00	0.01	0.00	-0.06	-0.05	-0.07	-0.01	0.01
Dependent_count	0.02	-0.12	1.00	-0.10	-0.04	-0.01	-0.04	0.07	-0.00	0.07	-0.04	0.03	0.05	0.01	-0.04
Months_on_book	0.01	0.79	-0.10	1.00	-0.01	0.07	-0.01	0.01	0.01	0.01	-0.05	-0.04	-0.05	-0.01	-0.01
Total_Relationship_Count	-0.15	-0.01	-0.04	-0.01	1.00	-0.00	0.06	-0.07	0.01	-0.07	0.05	-0.35	-0.24	0.04	0.07
Months_Inactive_12_mon	0.15	0.05	-0.01	0.07	-0.00	1.00	0.03	-0.02	-0.04	-0.02	-0.03	-0.04	-0.04	-0.04	-0.01
Contacts_Count_12_mon	0.20	-0.02	-0.04	-0.01	0.06	0.03	1.00	0.02	-0.05	0.03	-0.02	-0.11	-0.15	-0.09	-0.06
Credit_Limit	-0.02	0.00	0.07	0.01	-0.07	-0.02	0.02	1.00	0.04	1.00	0.01	0.17	0.08	-0.00	-0.48
Total_Revolving_Bal	-0.26	0.01	-0.00	0.01	0.01	-0.04	-0.05	0.04	1.00	-0.05	0.06	0.06	0.06	0.09	0.62
Avg_Open_To_Buy	-0.00	0.00	0.07	0.01	-0.07	-0.02	0.03	1.00	-0.05	1.00	0.01	0.17	0.07	-0.01	-0.54
Total_Amt_Chng_Q4_Q1	-0.13	-0.06	-0.04	-0.05	0.05	-0.03	-0.02	0.01	0.06	0.01	1.00	0.04	0.01	0.38	0.04
Total_Trans_Amt	-0.17	-0.05	0.03	-0.04	-0.35	-0.04	-0.11	0.17	0.06	0.17	0.04	1.00	0.81	0.09	-0.08
Total_Trans_Ct	-0.37	-0.07	0.05	-0.05	-0.24	-0.04	-0.15	0.08	0.06	0.07	0.01	0.81	1.00	0.11	0.00
Total_Ct_Chng_Q4_Q1	-0.29	-0.01	0.01	-0.01	0.04	-0.04	-0.09	-0.00	0.09	-0.01	0.38	0.09	0.11	1.00	0.07
Avg_Utilization_Ratio	-0.18	0.01	-0.04	-0.01	0.07	-0.01	-0.06	-0.48	0.62	-0.54	0.04	-0.08	0.00	0.07	1.00

- **Negative correlation with Attrition_Flag:** Total_Trans_Ct, Total_Ct_Chng_Q4_Q1, Total_Revolving_Bal → As these data values decrease, chances of attrition increases
- **Positive correlation:** Avg_Utilization_Ratio and Total_Revolving_Bal
- **Initial analysis:** Hypothesized that these features have an impact on customers attriting

Data Preprocessing

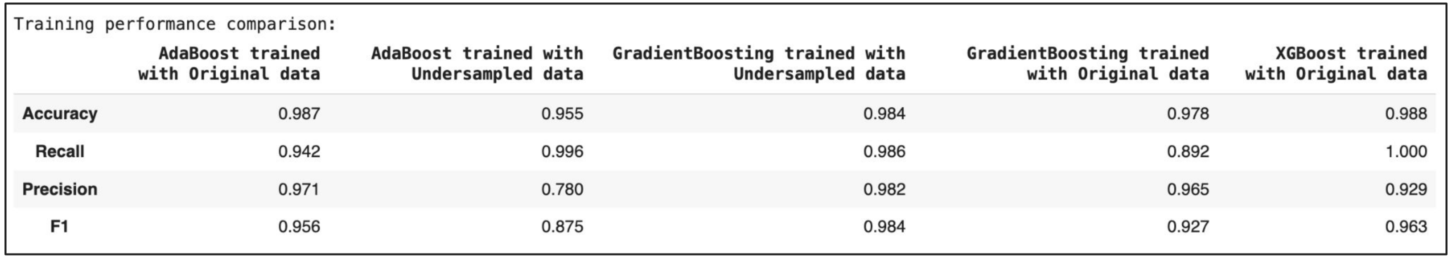
- **Duplicate values:** Checked and none were found
- **Outliers:** Checked & identified; the data was deemed to be normal, so no treatment was necessary
- **Missing values:** Found for Education_Level, Marital_Status, and Income_Category; these were imputed with the most frequent value for that particular feature
- **Feature engineering**
  - Clientnum was dropped because it's unique for a customer and doesn't provide value for modeling
  - One-Hot Encoding was used for categorical features: Gender, Education_Level, Marital_Status, Income_Category, and Card_Category
  - Attrition_Flag data was updated from "Existing Customer" to 0 and "Attrited Customer" to 1
  - Income_Category data had 1,112 instances of "abc" which was changed to null (this was done before imputing)

Data Preprocessing (cont.)

- **Data preparation for modeling**
    - The original dataset was split into 3 datasets: 80% train, 20% validation, and 20% test
    - The imputing on the missing values (mentioned on the previous slide) was done on each of these dataset independently so as not to introduce data leakage
    - One-hot encoding on the categorical features (mentioned on the previous slide) was done on each of these dataset independently so as not to introduce data leakage

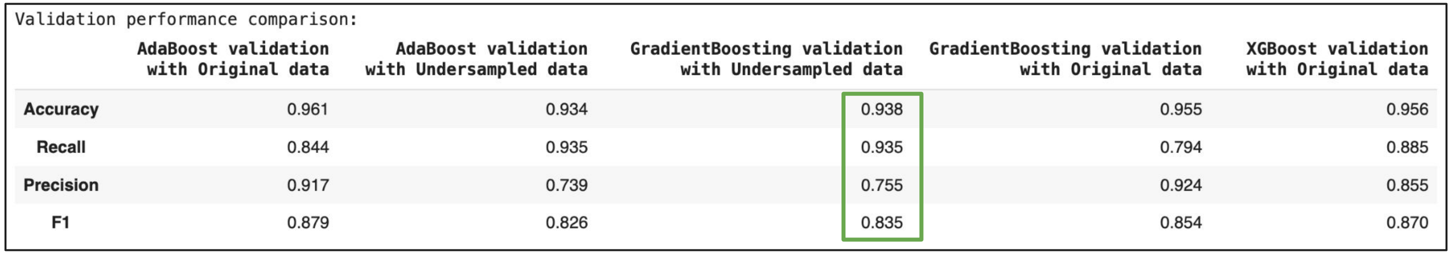
Model Performance Summary

- 5 types of classification models (Bagging, RandomForest, AdaBoost, GradientBoosting, & XGBoost) were trained and validated on the original data, oversampled data, and undersampled data.
- 5 models were chosen from the 15 created in the previous step. The best models based on recall and the lowest differential between test and validation were chosen for hyperparameter optimization using RandomizedSearchCV for cross-validation. These results are below.



Training performance comparison:

	AdaBoost trained with Original data	AdaBoost trained with Undersampled data	GradientBoosting trained with Undersampled data	GradientBoosting trained with Original data	XGBoost trained with Original data
Accuracy	0.987	0.955	0.984	0.978	0.988
Recall	0.942	0.996	0.986	0.892	1.000
Precision	0.971	0.780	0.982	0.965	0.929
F1	0.956	0.875	0.984	0.927	0.963

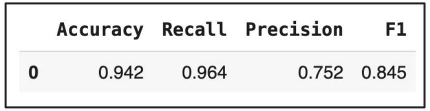


Validation performance comparison:

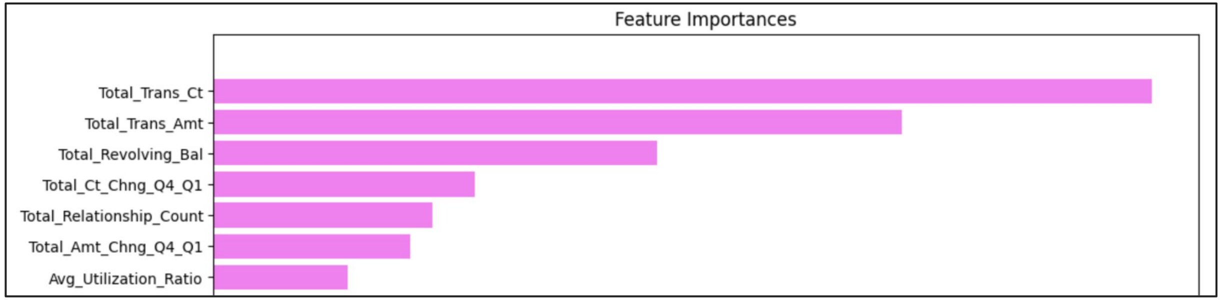
	AdaBoost validation with Original data	AdaBoost validation with Undersampled data	GradientBoosting validation with Undersampled data	GradientBoosting validation with Original data	XGBoost validation with Original data
Accuracy	0.961	0.934	0.938	0.955	0.956
Recall	0.844	0.935	0.935	0.794	0.885
Precision	0.917	0.739	0.755	0.924	0.855
F1	0.879	0.826	0.835	0.854	0.870

Model Performance Summary (cont.)

- The final model chosen was the "GradientBoosting with Undersampled Data" model. The rational is that, while tied for the highest Recall of all the models (with the "AdaBoost with Undersampled Data" model), it outperforms on Accuracy, Precision, and F1.
- The "GradientBoosting with Undersampled Data" model's results on the test data is shown below. It performs extremely well with Recall of 0.964.
- The top Feature Importances for the model are shown as well.



	Accuracy	Recall	Precision	F1
0	0.942	0.964	0.752	0.845



Gold
Credit Card
Credit Card
Credit Card
Credit Card
Credit

APPENDIX

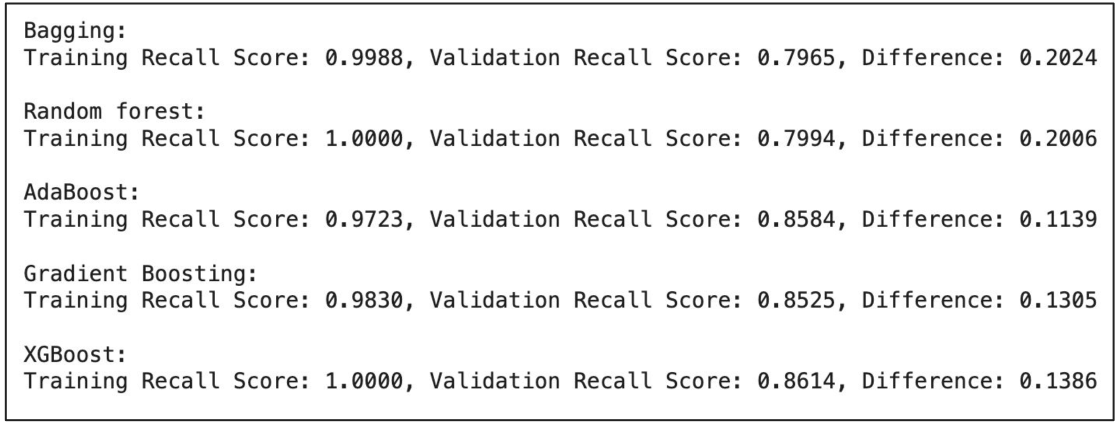
Model Performance Summary (Original Data)

- Below is a summary of the recall scores using the original training data and the validation data on the 5 different models used on this project
- XGBoost, AdaBoost, and Gradient Boosting have the best validation recall scores. XGBoost has the best score but may be overfit on training data. AdaBoost's score is not as good but might be the best model based on how close the train and validation recall scores are to each other.

```
Bagging:
Training Recall Score: 0.9864, Validation Recall Score: 0.7729, Difference: 0.2135

Random Forest:
Training Recall Score: 1.0000, Validation Recall Score: 0.7404, Difference: 0.2596

AdaBoost:
Training Recall Score: 0.8586, Validation Recall Score: 0.8171, Difference: 0.0415

Gradient Boosting:
Training Recall Score: 0.8942, Validation Recall Score: 0.7906, Difference: 0.1037

XGBoost:
Training Recall Score: 1.0000, Validation Recall Score: 0.8407, Difference: 0.1593
```

Model Performance Summary (Oversampled Data)

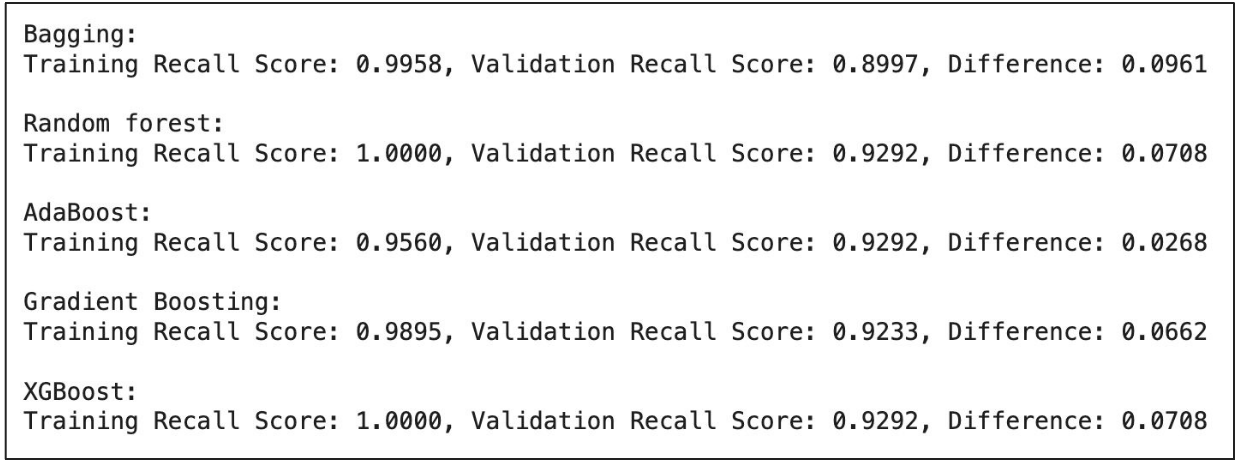
- SMOTE was the oversampling method that was used
- Below is a summary of the recall scores using the new oversampled training data and the validation data on the 5 different models used on this project
- XGBoost has the best validation recall score but might be overfit based on the training recall score being 1.0. AdaBoost is a close second with less overfit.



```
Bagging:
Training Recall Score: 0.9988, Validation Recall Score: 0.7965, Difference: 0.2024

Random forest:
Training Recall Score: 1.0000, Validation Recall Score: 0.7994, Difference: 0.2006

AdaBoost:
Training Recall Score: 0.9723, Validation Recall Score: 0.8584, Difference: 0.1139

Gradient Boosting:
Training Recall Score: 0.9830, Validation Recall Score: 0.8525, Difference: 0.1305

XGBoost:
Training Recall Score: 1.0000, Validation Recall Score: 0.8614, Difference: 0.1386
```

Model Performance Summary (Undersampled Data)

- RandomUndersampler was the undersample method used
- Below is a summary of the recall scores using the new undersampled training data and the validation data on the 5 different models used on this project
- AdaBoost, Random Forest, and XGBoost all have the same and highest validation recall scores. AdaBoost has the lowest difference and may not be as overfit on the training data.



```
Bagging:
Training Recall Score: 0.9958, Validation Recall Score: 0.8997, Difference: 0.0961

Random forest:
Training Recall Score: 1.0000, Validation Recall Score: 0.9292, Difference: 0.0708

AdaBoost:
Training Recall Score: 0.9560, Validation Recall Score: 0.9292, Difference: 0.0268

Gradient Boosting:
Training Recall Score: 0.9895, Validation Recall Score: 0.9233, Difference: 0.0662

XGBoost:
Training Recall Score: 1.0000, Validation Recall Score: 0.9292, Difference: 0.0708
```