Ensemble Techniques and Model Tuning

Nick Lutostanski
AI/ML for Business Applications

2.27.24

Contents / Agenda

Executive Summary

- We built a predictive model that will help the bank predict whether or not a customer is likely to churn (cancel their card). We chose the Adaboost method with undersampled data as the best-performing model.

- We found that the most important features to predict whether or not a customer will churn are the total annual spend, total annual transactions, and total revolving balance.

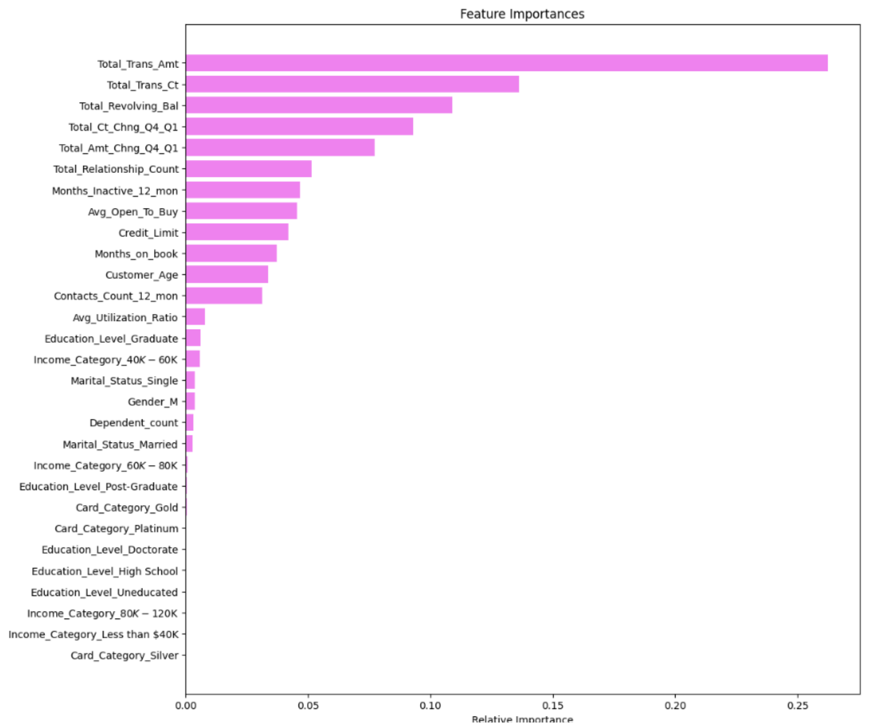
  - Looking a the bivariate data, most of the customers who cancelled their credit cards with the bank had a total annual spend of about $4,000 per year.

  - Customers with less than about 75 transactions a year are predicted to churn.

  - 75% of the customers that churned have a total revolving account balance of less than $1500.

- Using these insights, we can target promotions towards these customers to incentivize them to use their cards more, which we would predict would keep them as credit card customers. A 0% interest window targeted to these customers might be considered to increase usage in both annual transactions, annual spend, and total revolving balance, which would increase the bank's ability to retain customers.

- Customers with lower % credit utilization are more likely to churn. They may be less likely to use credit and use cash as an alternative. Promotion consideration might be targeted to these customers as well to reduce likelihood of churn.



Business Problem Overview and Solution Approach

- The Thera bank recently saw a steep decline in the number of users of their credit cards. Credit cards are a good source of income for banks because of the different kinds of fees charged by the banks; annual fees, balance transfer fees, and cash advance fees, late payment fees, foreign transaction fees, and others. Some fees are charged to every user irrespective of usage, while others are charged under specified circumstances.
- Customers' leaving credit card services would lead the bank to lose revenue, so the bank wants to analyze the data of customers and identify the customers who will leave their credit card services and why– so that the bank could improve upon those areas
- You as a Data Scientist at Thera Bank need to come up with a classification model that will help the bank improve its services so that customers do not cancel their credit cards.

- We are going to build 5 models (decision tree, bagging, Adaboost, Gradient Boosting, and random forest) using the original data, oversampled data, and under-sampled data.
- We are then going to pick 3 of these models and use hyperparameters to tune the performance using GridSearchCV to find optimal values for the hyperparameters.
- From there we will pick the best-performing model and present business insights and recommendations.
- The model criterion of importance will be the Recall score. The greater the recall score, the higher the chances of minimizing false negatives. This is so that the bank has the highest chance of identifying the true positives(i.e. Class 1) so that the bank can retain their valuable customers by identifying the customers who are at risk of churn.

EDA Results

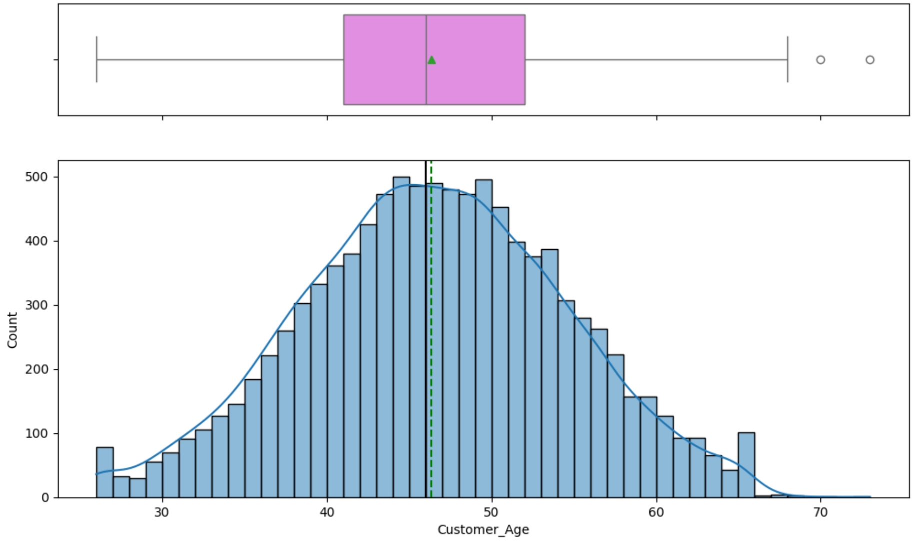
Features

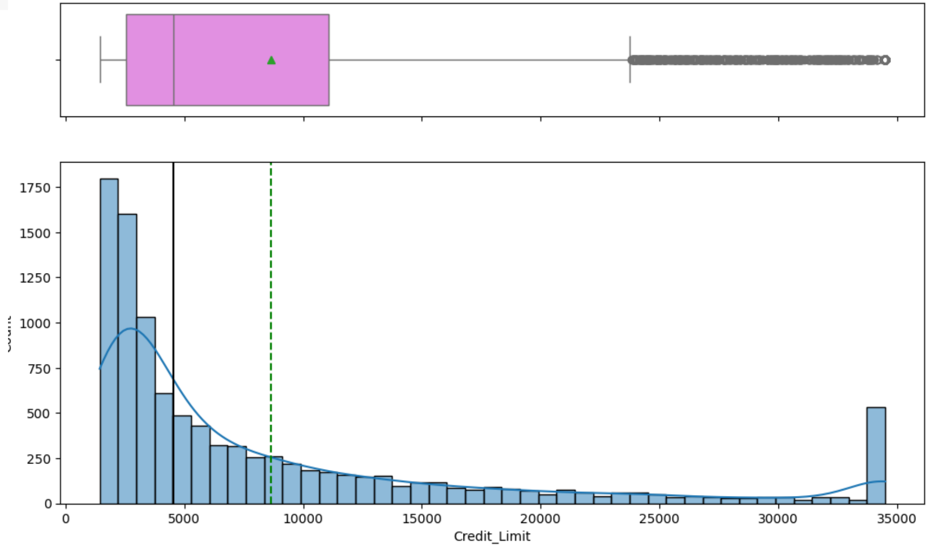
- CLIENTNUM: Client number. Unique identifier for the customer holding the account
- Attrition_Flag: Internal event (customer activity) variable - if the account is closed then "Attrited Customer" else "Existing Customer"
- Customer_Age: Age in Years
- Gender: Gender of the account holder
- Dependent_count: Number of dependents
- Education_Level: Educational Qualification of the account holder - Graduate, High School, Unknown, Uneducated, College(refers to college student), Post-Graduate, Doctorate
- 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 (in months)
- 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

- **Note: Attrited customers will be coded to 1 and Existing Customers will be coded 0 for the purpose of our analysis.**

EDA Results – Univariate Analysis

```
ıistogram_boxplot(data, "Customer_Age", kde=True)
```



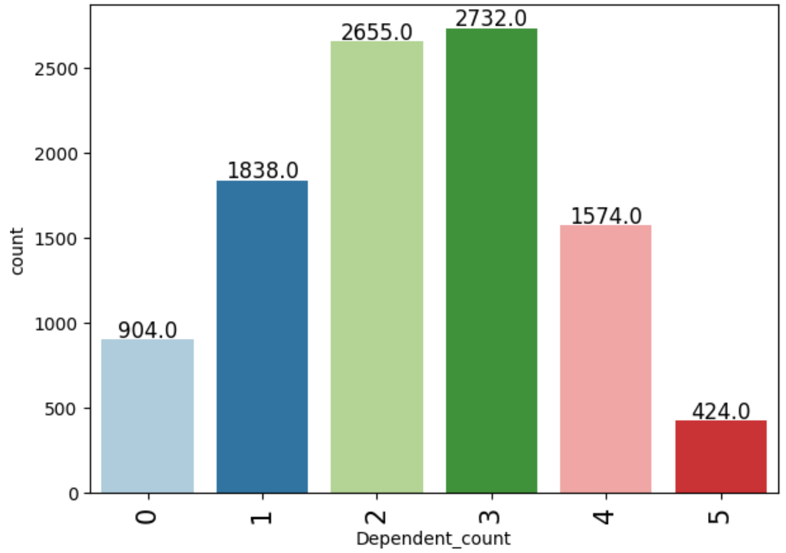


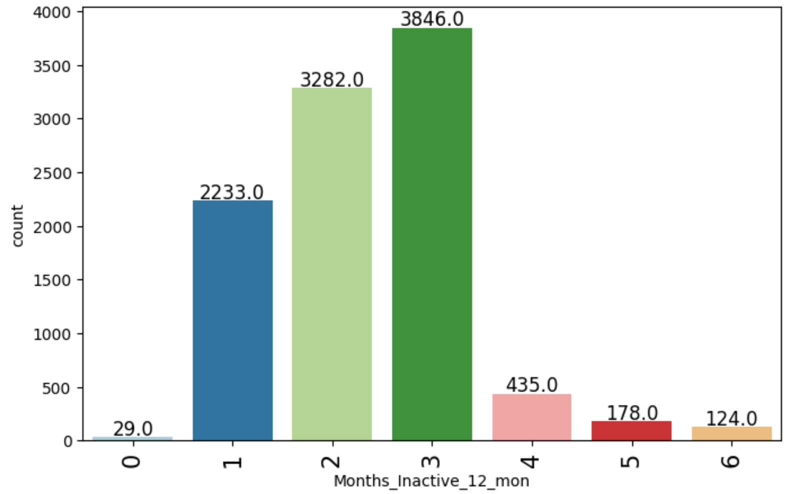
Customer_age is observed to be a normal distribution with a mean/median around 46 years of age.

Credit_Limit is observed to be skewed right distribution with a median of $8,600 and significant outliers

Link to Appendix slide on data background check

EDA Results - Univariate Analysis



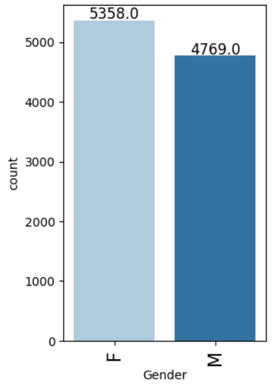


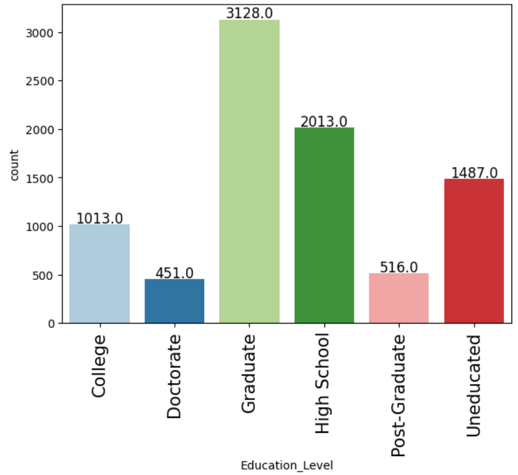
Dependent_count might be an interesting variable to look at since that could potentially be correlated with higher levels of spending. It's somewhat normally distributed with a mean/median ~ 2

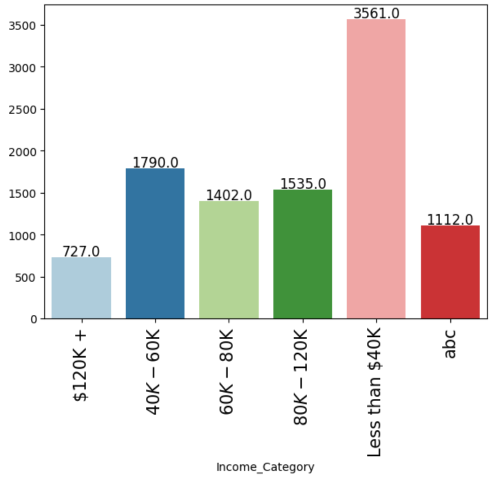
Months_Inactive_12_mon is an interesting variable because only 29 out of the more than 10 accounts are 0 months. Then the number of entries increases greatly for months 1-3 and then drops down between 4-5 months. No records are inactive >6 months.

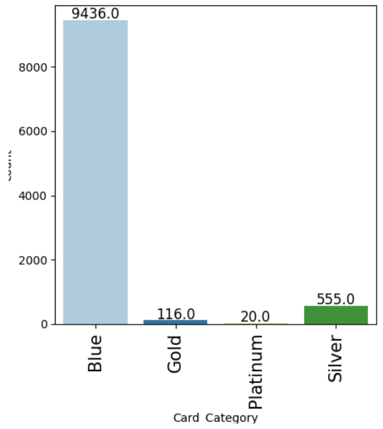
Link to Appendix slide on data background check

EDA Results - Univariate Analysis









In our dataset, the gender split is almost 50/50 with slightly more females than males.

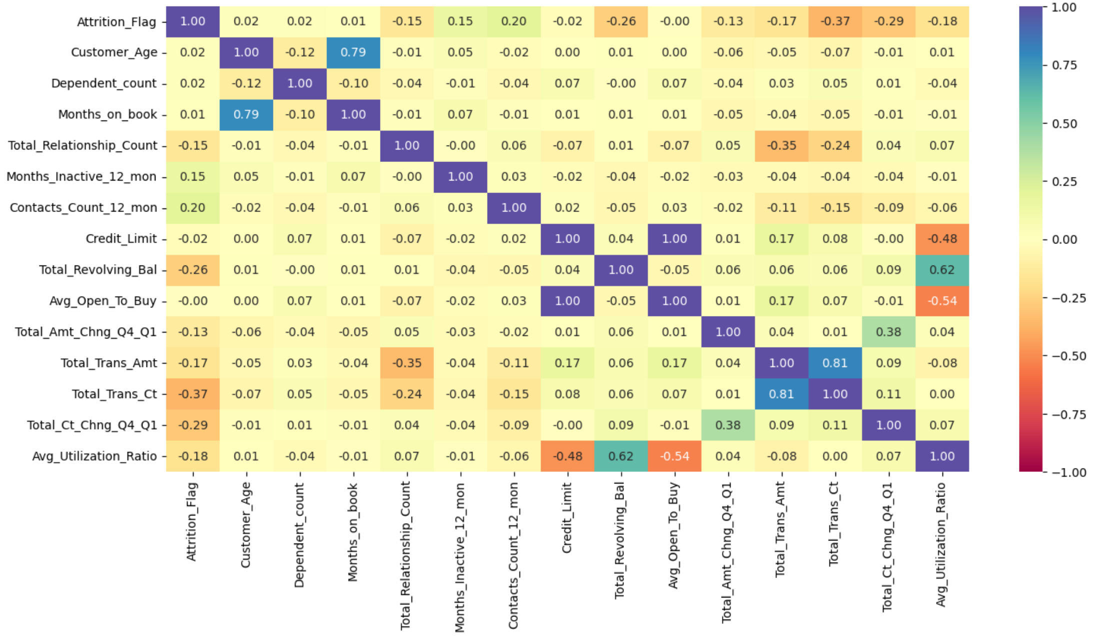
Masters/Graduate degrees make up almost a 1/3 of the data set with high school and uneducated following. Note that there are still 1,519 observations with null data for education so we need to impute them before starting our model building.

The highest number of observations in income make <$40k annually. There are still records from Income_Category that need to be imputed before we can continue with our model-building.

A vast majority of card-holders have the Blue Card (lowest level). Only 20 records hold platinum status (highest level).

Link to Appendix slide on data background check

EDA Results – Bivariate Analysis – Correlation Heatmap



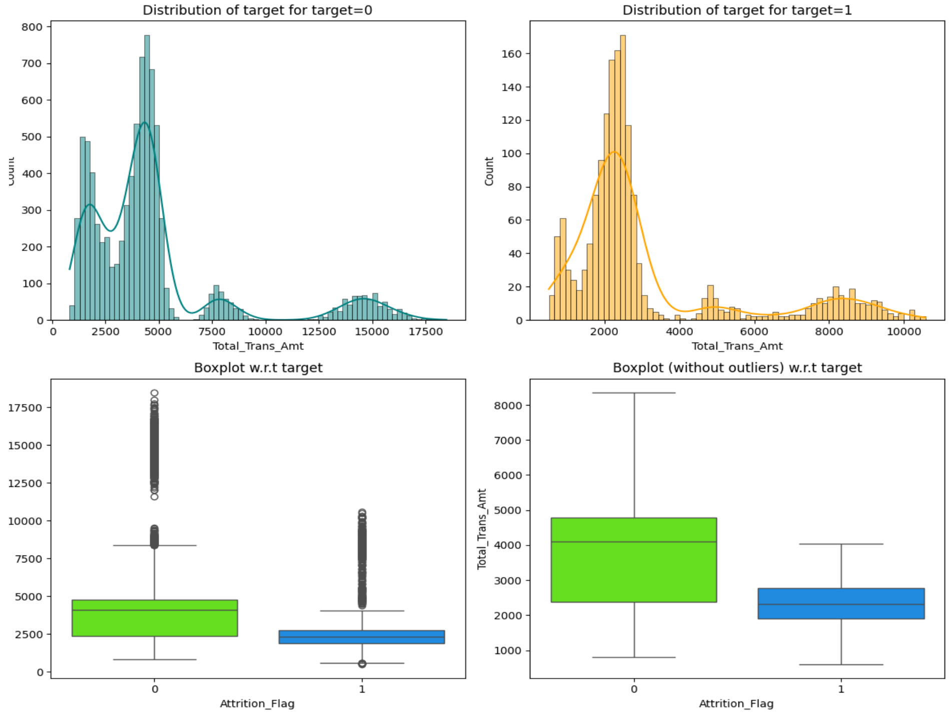
	Attrition_Flag	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Revolving_Bal	Avg_Open_To_Buy	Total_Amt_Chng_Q4_Q1	Total_Trans_Amt	Total_Trans_Ct	Total_Ct_Chng_Q4_Q1	Avg_Utilization_Ratio
Attrition_Flag	1.00	0.02	0.02	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

Among the highest correlated values, we see that Credit_Limit and Avg_Open_To_Buy have a correlation coefficient of 1 which makes sense because when you make a payment, your credit limit and amount open to buy increase by the same amount.

Months_on_book is highly correlated with age for obvious reasons. Every increase in 1 year of age is an increase of months_on_book in a positive relationship.

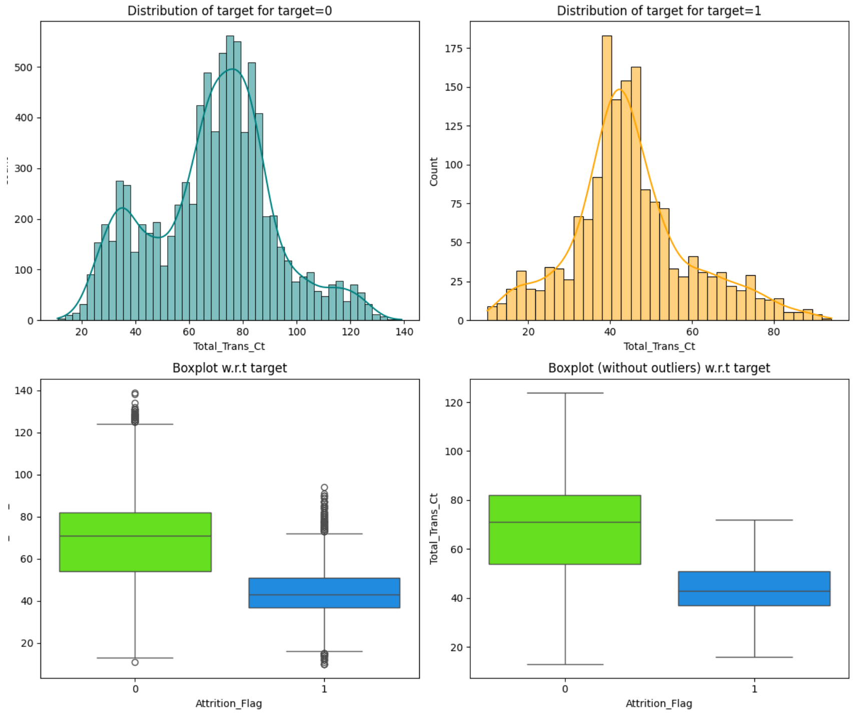
Avg_Utilization_Ratio is highly correlated with Total_Revolving_Balance – as money is spent, revolving balance and average utilization both increase positively.

EDA Results – Bivariate Analysis



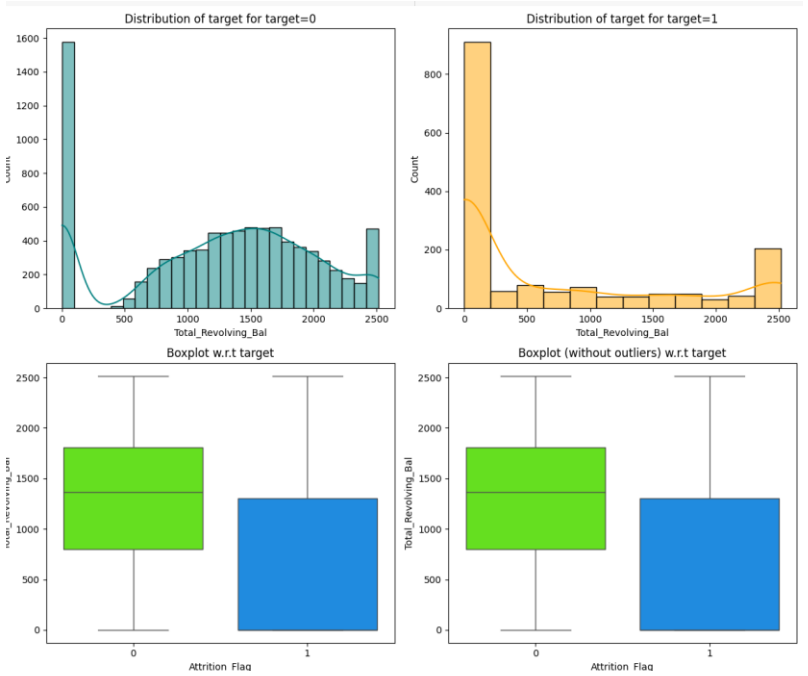
Customers who spend more annually are more likely to stay as existing customers versus those who don't spend as much. This reinforces in the data that the more the card is utilized, the less likely the customer is to churn.

EDA Results – Bivariate Analysis



Customers with lower numbers of annual transactions are more likely to close their accounts compared to those with higher numbers of transactions. This aligns with our previous slide on higher balances – the more incentives we can provide for people to use the card and increase their balance, the less likely they to close their accounts.

EDA Results – Bivariate Analysis

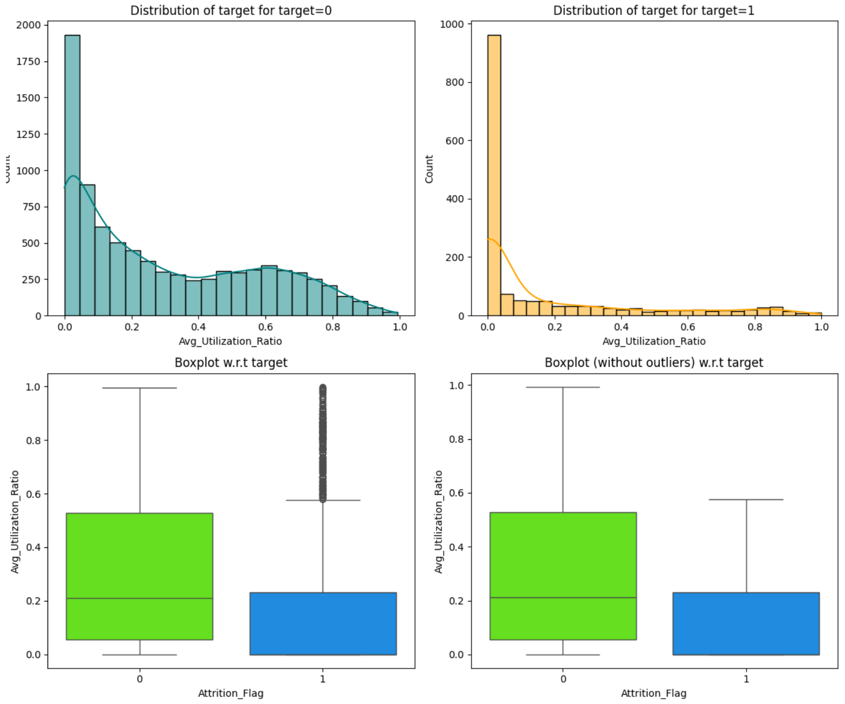


When examining the total revolving balance against the target variable of Attrition, those with lower balances were more likely to close their credit card accounts.

In both the most customers among the existing customers and those that close their accounts, most of them have zero balances. An interesting consideration is how we can lower attrition rates by offering promotions that increase balances.

Customers with higher revolving balances, meaning those who do not pay off their credit cards in full every month are more likely to remain customers.

EDA Results – Bivariate Analysis

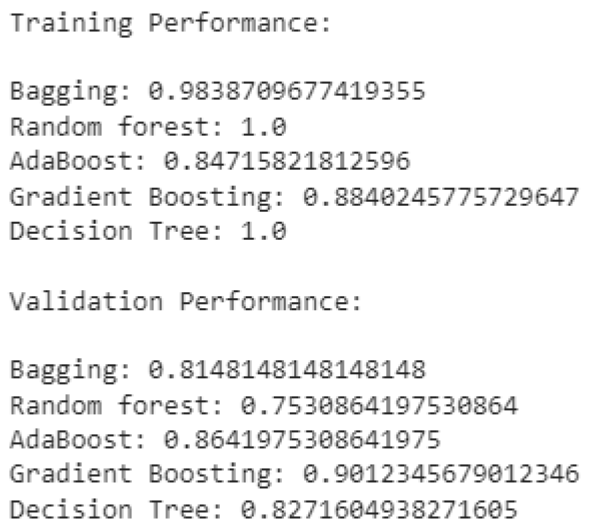


Customers with a lower average utilization ratio are more likely to churn than those with higher average utilization. This may be because they have more credit accounts open over multiple banks/accounts. This supports our observation that churn is related to no longer using the card or having other options.

Data Preprocessing

- There were no duplicate values in the dataset.
- There were a few missing values in Income_category and Marital_status. We will impute these before continuing with the model-building. There were also some strange values in the Income_Category listed as ‘abc’ that we will convert to np.na variables and then impute them using “most frequent strategy”.
- There we a few outliers, most commonly in Total_Trans_Amt, Avg_Open_To_Buy and Credit_Limit. We will not treat any of these to avoid tampering with the data.
- We impute data using most_frequent strategy for the columns Education_Status, Marital_Status, and Income_Category.
- After being imputed, we encode Gender, Education_Level, Marital_Status, Income_Category, and Card_Category for analysis.

Model Performance Summary – Original Data

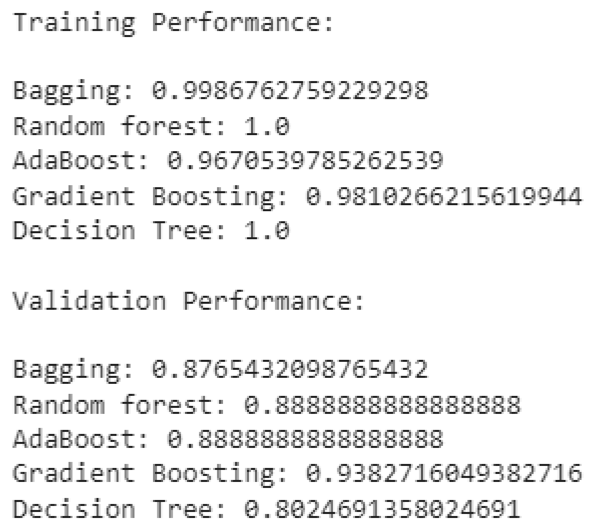


```
Training Performance:

Bagging: 0.9838709677419355
Random forest: 1.0
AdaBoost: 0.84715821812596
Gradient Boosting: 0.8840245775729647
Decision Tree: 1.0

Validation Performance:

Bagging: 0.8148148148148148
Random forest: 0.7530864197530864
AdaBoost: 0.8641975308641975
Gradient Boosting: 0.9012345679012346
Decision Tree: 0.8271604938271605
```

- Bagging results in good results on the training set, but does poorly on the validation set.
- Random forest is overfitted on the training set and performs poorly on the validation set.
- Adaboost performance is decent on the training set and is better on the validation set
- Gradient boosting shows a similar performance to Adaboost
- Decision Tree is overfitting on training set and performs poorly on the validation set.
- So far the boosting methods seem to do very well at recall for both training and validation set.

Link to Appendix slide on model assumptions

Model Performance Summary – Oversampled data



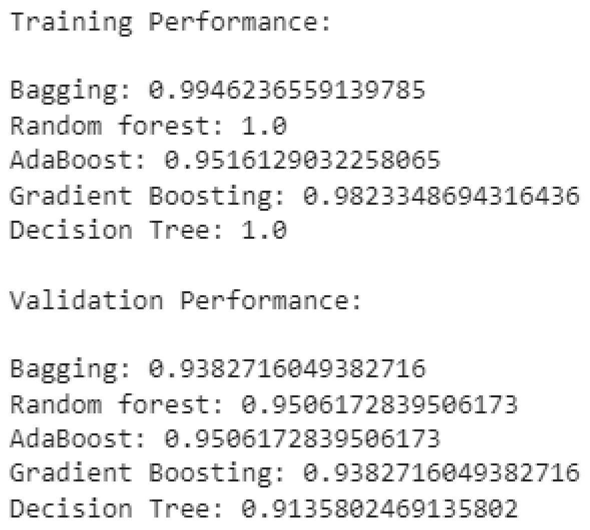
```
Training Performance:

Bagging: 0.9986762759229298
Random forest: 1.0
AdaBoost: 0.9670539785262539
Gradient Boosting: 0.9810266215619944
Decision Tree: 1.0

Validation Performance:

Bagging: 0.8765432098765432
Random forest: 0.8888888888888888
AdaBoost: 0.8888888888888888
Gradient Boosting: 0.9382716049382716
Decision Tree: 0.8024691358024691
```

Oversampled data provides synthetic data points for the minority class to hopefully improve performance and recall performance. It does this by creating the same number in the minority class as there is in the majority class.

- Random Forest, Decision Tree, and Bagging appear to be overfit on the training data and performs poorly on the validation data.
- The boosting methods seemed to give great recall scores and test better on the validation set, with an advantage going to Gradient boosting

Link to Appendix slide on model assumptions

Model Performance Summary – Undersampled Data



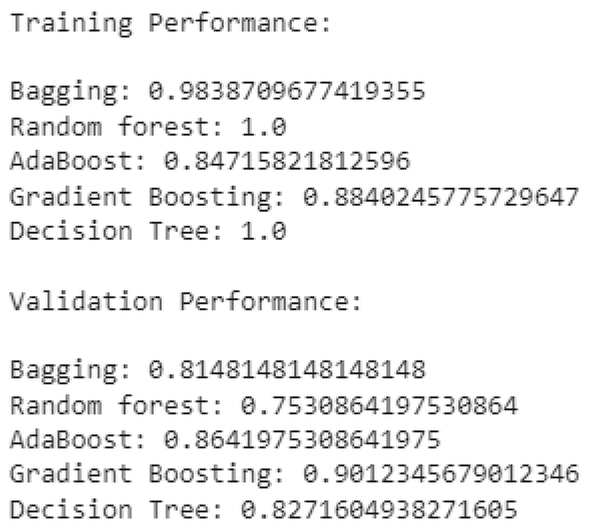
```
Training Performance:

Bagging: 0.9946236559139785
Random forest: 1.0
AdaBoost: 0.9516129032258065
Gradient Boosting: 0.9823348694316436
Decision Tree: 1.0

Validation Performance:

Bagging: 0.9382716049382716
Random forest: 0.9506172839506173
AdaBoost: 0.9506172839506173
Gradient Boosting: 0.9382716049382716
Decision Tree: 0.9135802469135802
```

Under sampled data balances the data set by removing majority class data from the data set until is the same sample size as the previously minority class.

- Using under sampled data, all models seem to provide good performance against the validation set, with Adaboost performing the best.

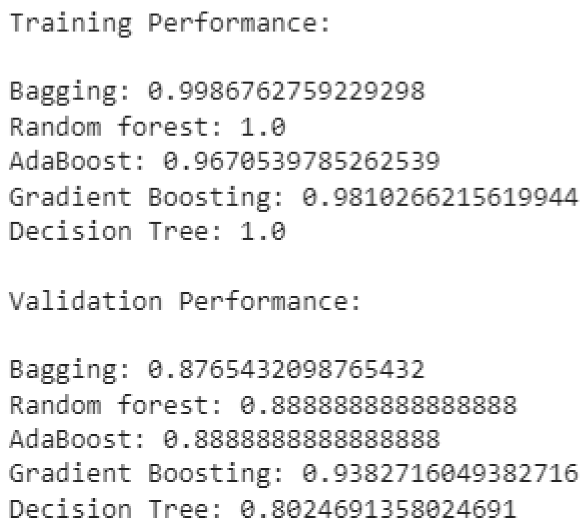
Link to Appendix slide on model assumptions

Model Performance Summary Comparison



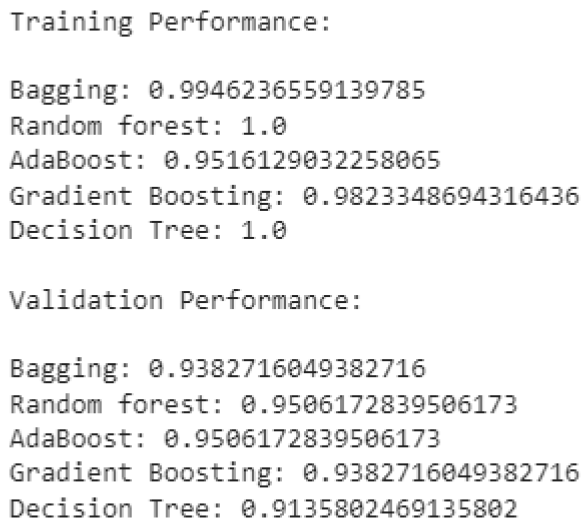
```
Training Performance:

Bagging: 0.9838709677419355
Random forest: 1.0
AdaBoost: 0.84715821812596
Gradient Boosting: 0.8840245775729647
Decision Tree: 1.0

Validation Performance:

Bagging: 0.8148148148148148
Random forest: 0.7530864197530864
AdaBoost: 0.8641975308641975
Gradient Boosting: 0.9012345679012346
Decision Tree: 0.8271604938271605
```

Original Data



```
Training Performance:

Bagging: 0.9986762759229298
Random forest: 1.0
AdaBoost: 0.9670539785262539
Gradient Boosting: 0.9810266215619944
Decision Tree: 1.0

Validation Performance:

Bagging: 0.8765432098765432
Random forest: 0.8888888888888888
AdaBoost: 0.8888888888888888
Gradient Boosting: 0.9382716049382716
Decision Tree: 0.8024691358024691
```

Oversampled Data



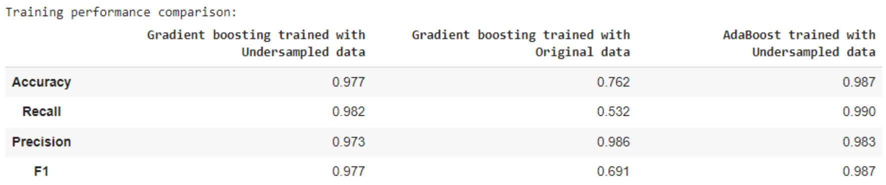
```
Training Performance:

Bagging: 0.9946236559139785
Random forest: 1.0
AdaBoost: 0.9516129032258065
Gradient Boosting: 0.9823348694316436
Decision Tree: 1.0

Validation Performance:

Bagging: 0.9382716049382716
Random forest: 0.9506172839506173
AdaBoost: 0.9506172839506173
Gradient Boosting: 0.9382716049382716
Decision Tree: 0.9135802469135802
```

Undersampled Data

Based on the comparisons with all the 3 different models, Adaboost and Gradient boosting seem to yield the highest recall abilities. We will use these to tune more models to make a final decision.

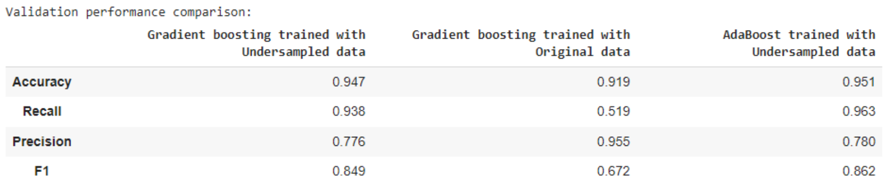
Link to Appendix slide on model assumptions

Model Performance Summary HyperParameter Tuning of Adaboost and Gradient Boost



Training performance comparison:

	Gradient boosting trained with Undersampled data	Gradient boosting trained with Original data	AdaBoost trained with Undersampled data
Accuracy	0.977	0.762	0.987
Recall	0.982	0.532	0.990
Precision	0.973	0.986	0.983
F1	0.977	0.691	0.987



Validation performance comparison:

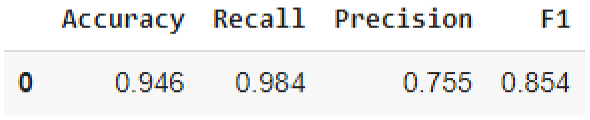
	Gradient boosting trained with Undersampled data	Gradient boosting trained with Original data	AdaBoost trained with Undersampled data
Accuracy	0.947	0.919	0.951
Recall	0.938	0.519	0.963
Precision	0.776	0.955	0.780
F1	0.849	0.672	0.862

We chose to tune the Adaboost with undersampled data, Gradient boosting with Original Data, and Gradient Boosting with Undersampled data.

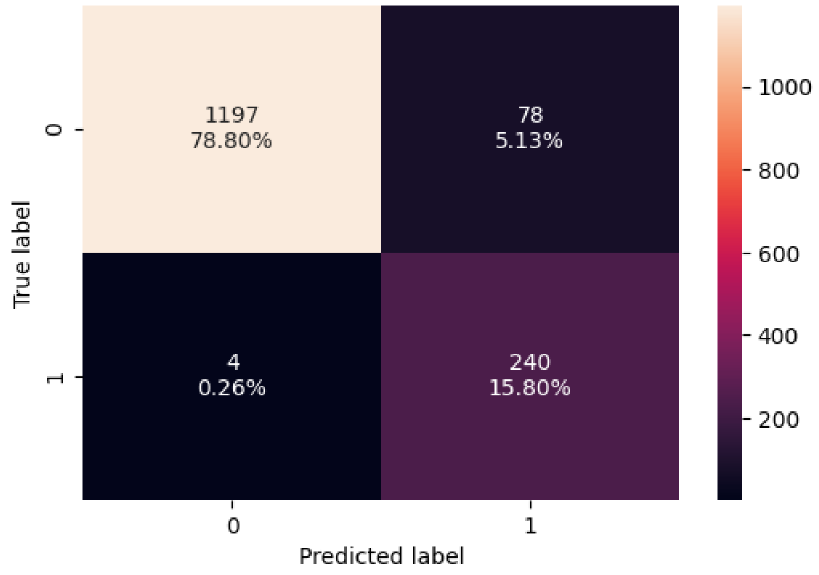
According to our results, the Adaboost trained with undersampled data performs the best. It produces a .99 recall score on the training set and a .963 recall score on the validation set.

Link to Appendix slide on model assumptions

Model Performance Summary (Adaboost w/ undersample model – test set)



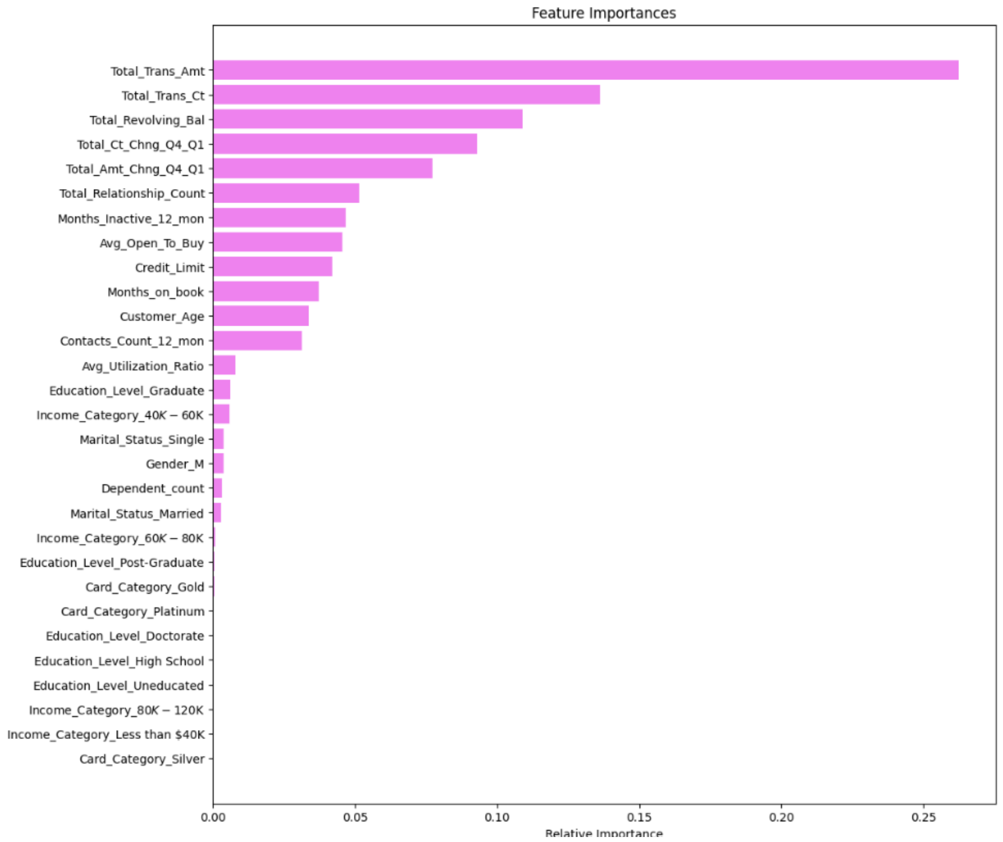
	Accuracy	Recall	Precision	F1
0	0.946	0.984	0.755	0.854



Here are the performance results for the Adaboost with undersampling. It performs even better on the unseen test set than it did on the validation set, producing a recall score of .984. There were only 4 observations that were incorrectly classified as false negatives.

Link to Appendix slide on model assumptions

Feature Importance (AdaBoost w/ undersampled



APPENDIX

Great Learning

Happy Learning !