Credit Card Users Churn Prediction

Advanced Machine Learning Project - PGP AIML

Chandramouli Pasumarthi

Date: 01/25/2024

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model performance summary for hyperparameter tuning.
- Appendix

Executive Summary

- The Thera bank recently saw a steep decline in the number of users of their credit card, credit cards are a good source of income for banks 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 to every user irrespective of usage, while others are charged under specified circumstances.

- Customers' leaving credit cards services would lead bank to loss, so the bank wants to analyze the data of customers and identify the customers who will leave their credit card services and reason for same – so that bank could improve upon those areas

Business Problem Overview and Solution Approach

Business Problem Overview:

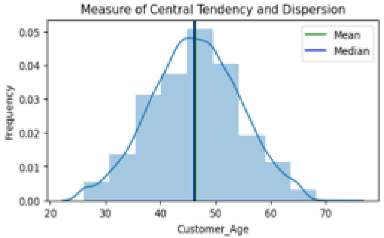
The Thera bank is facing a significant decline in credit card users, impacting a key source of income generated through various fees. These fees include annual fees, balance transfer fees, cash advance fees, late payment fees, foreign transaction fees, and more. The bank is concerned about potential financial losses due to customers leaving their credit card services. To address this, the bank aims to analyze customer data, identify users likely to leave, and understand the reasons for attrition. The goal is to enhance services and prevent customers from discontinuing their credit cards.

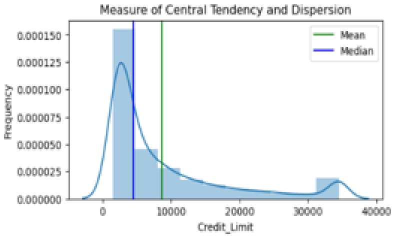
Solution Approach:

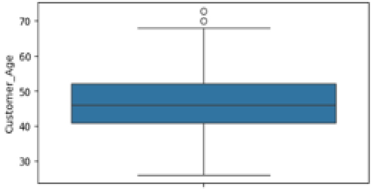
As a Data Scientist at Thera bank, the proposed solution involves developing a classification model to predict customer attrition

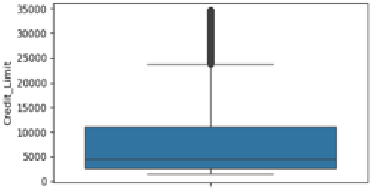
- Data Exploration: Understand the characteristics of the dataset. Explore relationships between variables and identify patterns.
- Data Preprocessing: Handle missing values, outliers, and any data inconsistencies. Encode categorical variables and standardize numerical features.
- Feature Engineering: Create new relevant features if needed. Select important features that contribute to the prediction.
- Model Selection: Choose a suitable classification algorithm for predicting customer attrition. Consider algorithms like Logistic Regression, Decision Trees, or Gradient Boosting.
- Model Training: Split the dataset into training and testing sets. Train the selected model on the training set.
- Model Evaluation: Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1 score. Use cross-validation to ensure robustness.
- The proposed classification model aims to be a proactive tool for the Thera bank, enabling them to predict and prevent customer attrition effectively. By understanding the drivers of attrition, the bank can implement targeted measures to enhance customer satisfaction and retention.

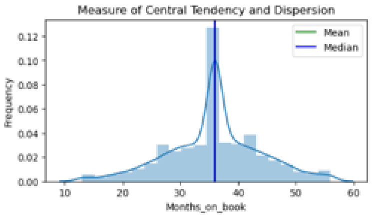
EDA Results-Univariate Analysis-Histogram, Boxplot

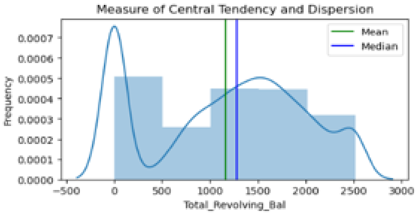


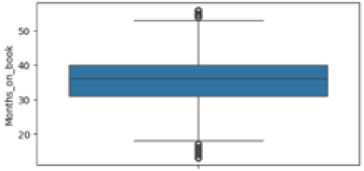








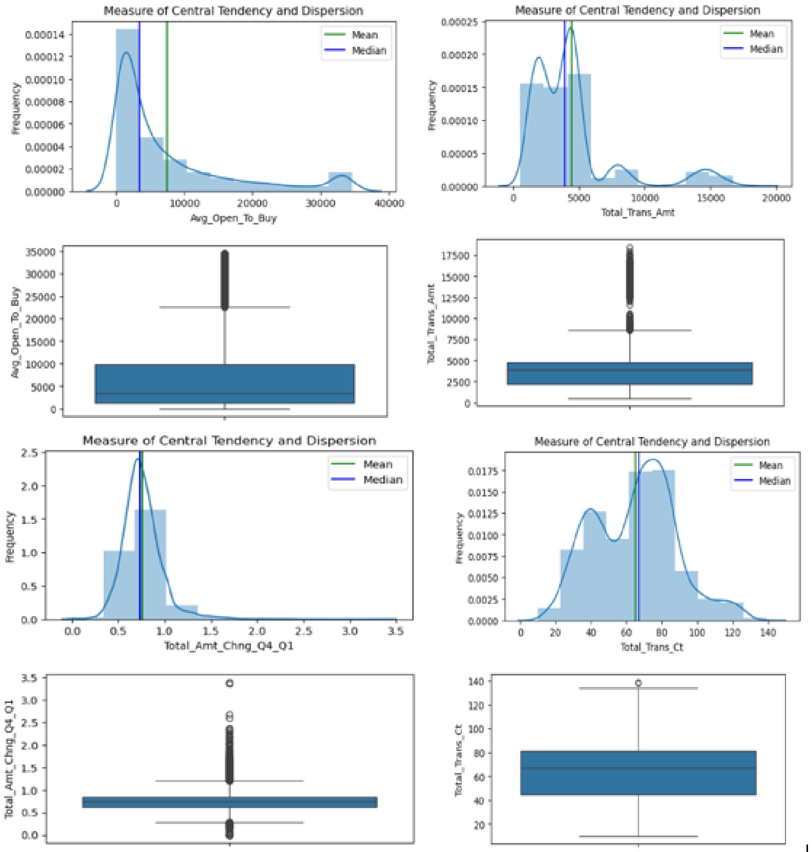




Observations:

- The banks customer age variable has a normal-like distribution.
- The average age of a customer is 46 years
- Most customers are within the ages 41 and 52 years old
- On average, credit cards usually expire 3 years (36 months) after the card was issued.
- Most customers have a relationship with the bank for between 31 and 40 months.
- The average credit limit given to customers is about 4,550 dollars
- As earlier mentioned, there are outliers within this variable with some customers having a credit limit as high as 34,500 dollars
- Most customers however have credit limits between 2,555 and 11,067 dollars
- The average total revolving balance of all customers about is 1,276 dollars
- Most customers however have a total revolving balance between 359 and 1,784 dollars

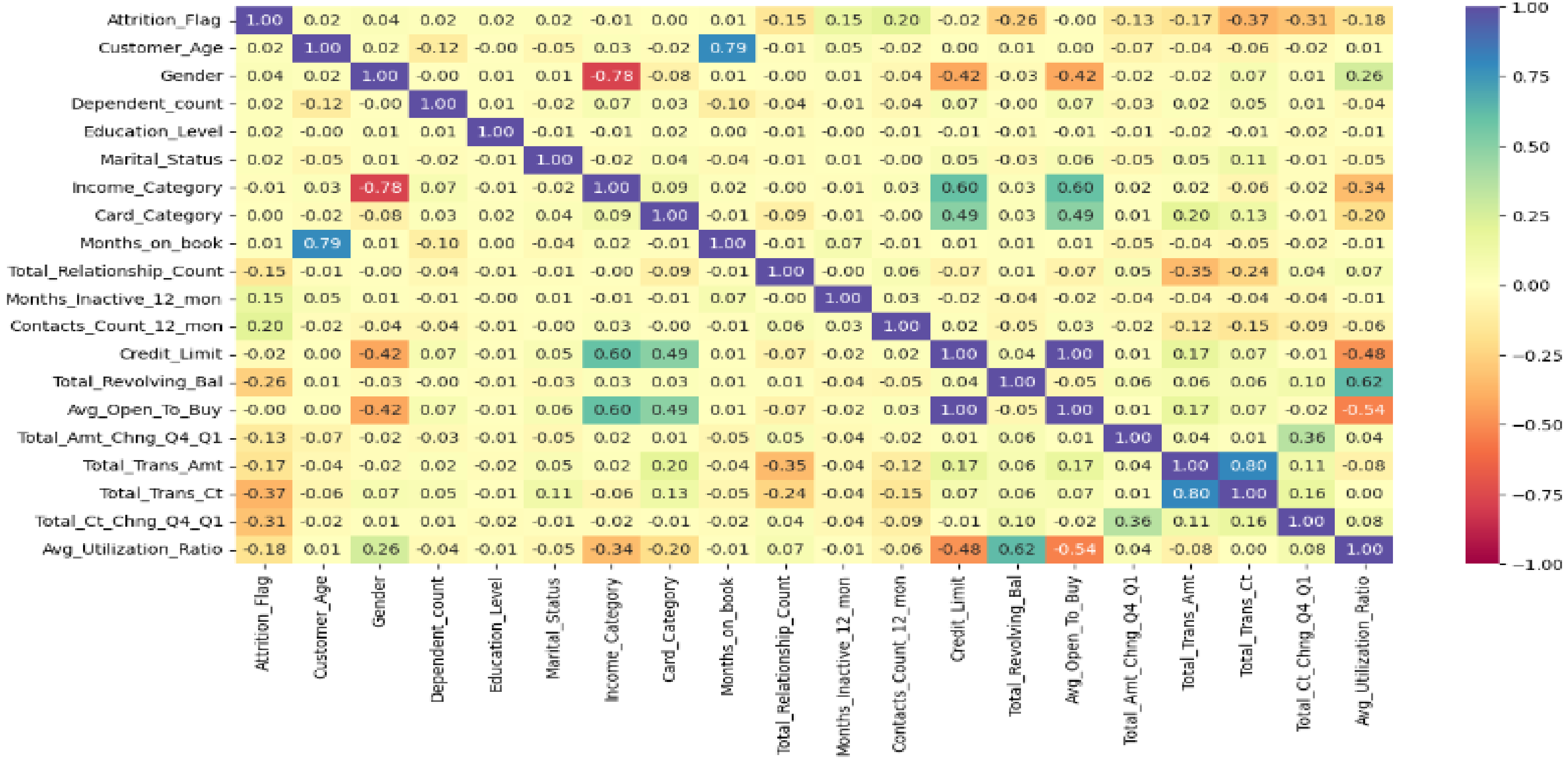
EDA Results-Univariate Analysis-Histogram, Boxplot



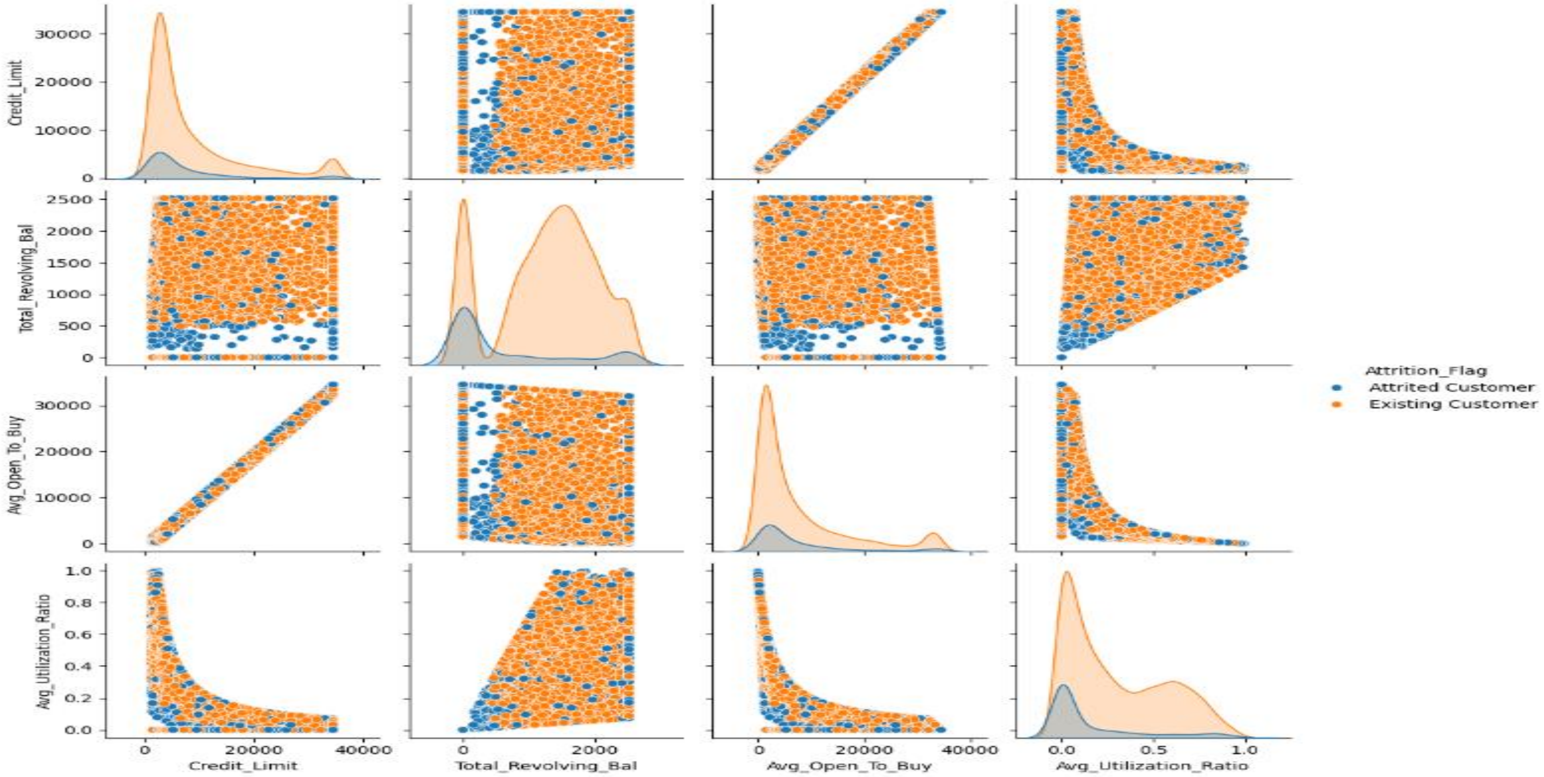
Observations:

- The average open to buy amount of all customers is about 3,474 dollars
- Most customers however have average open to buy between 1,324 and 9,859 dollars
- On average, there was a 73.6% increase in the bank customers transaction spend between Q1 and Q4
- There are some outliers within this variable on the right side with some customers transaction spend rising as much as 340% between Q1 and Q4
- The average amount spent by customers in the last 12 months is 3,899 dollars
- There are some outliers within this variable on the right side with some customers transaction spend rising as much as 18,484 dollars
- Most customers spend between 2,156 and 4,741 dollars
- The average number of transactions made by customers in the last 12 months is 67
- Most customers made between 45 and 81 transactions in the last 12 months
- On average, the number of transactions by customers increased by 70% between Q1 and Q4
- There are some outliers within this variable on the right side with some customers transaction spend rising as much as 370% between Q1 and Q4
- On average, most customers used about 17.6% of their credit limit in the last 12 months
- There are some outliers within this variable on the right side with some customers using as much as 99% of their credit limit
- Most customers used between 2.3% and 50% of their available credit limits

EDA Results-Bivariate Analysis-Heatmap



EDA Results-Bivariate Analysis- Pairplot

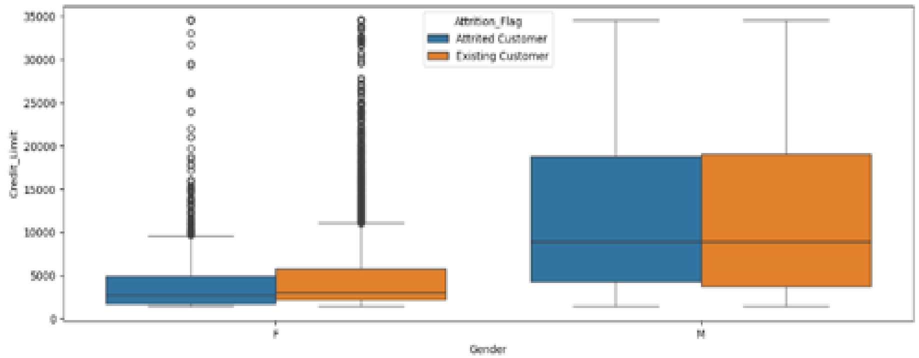


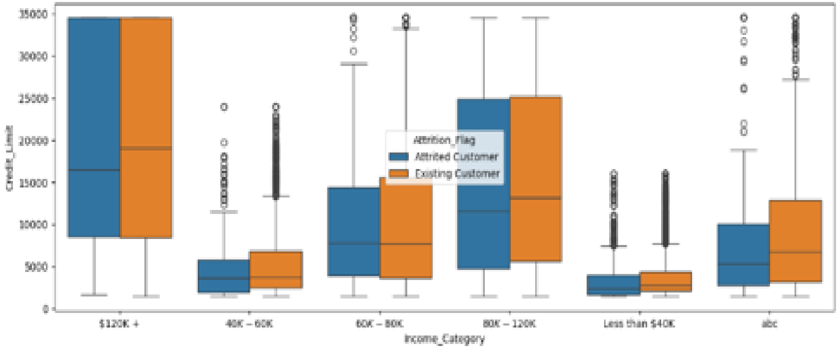
EDA Results-Bivariate Analysis-Summary

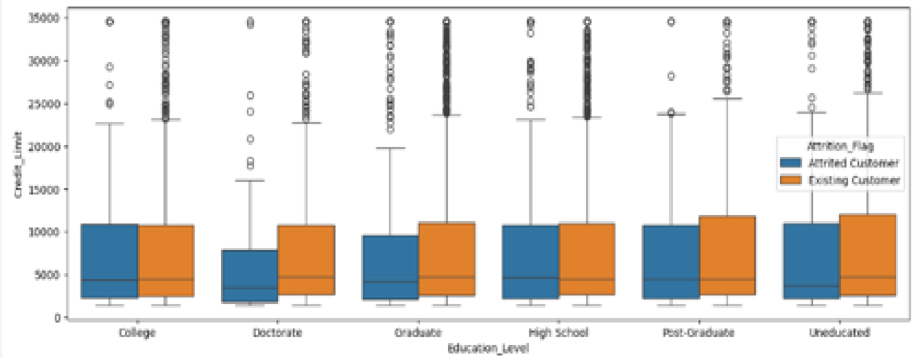
Observation

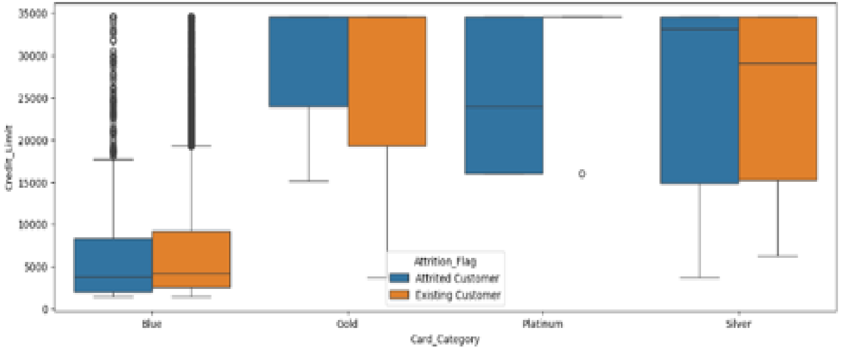
- *There are a few strong positive and negative correlations amongst numerical variables. Customer_Age for example is highly correlated with Months_on_book which makes sense as the older a customer is the more likely they are to have been a customer of the bank for a longer period than a younger customer.*
- *There is a very strong correlation between Credit Limit and Average Open to Buy.*
- *Total Revolving Balance and Average Utilization Ratio also have a strong positive correlation with each other. This suggests that as the amount carried over month over month on a customer's account increases, the amount of given credit the customer actually spend also increases*
- *We can also see that there is a negative correlation between Average Open to Buy and Average Utilization Ratio. This makes sense as having available credit to use on the card at the end of 12 months suggests that card usage may not have been at the highest level over the same period*
- *It is also logical that Total_Trans_Amt and Total_Amt_Chng_Q4_Q1 are correlated as Total_Amt_Chng_Q4_Q values are derived from Total_Trans_Amt. We may have to drop one of these columns.*

EDA Results-MultiVariate Analysis









EDA Results-MultiVariate Analysis-Summary

Observation:

- *From this plot, we can see that Male customers were given higher credit limits than female customers*
- *As earlier stated, there isn't much of a difference between customers that attrited and existing customers for both male and female customers*
- *Customers with Doctorate level education that did not attrite were given a higher credit limit than customers with the same level of education and did attrite.*
- *Customers with college and High School level of education who attrited and did not attrite were given similar credit limits. Interestingly however, customers with Doctorate level education that did attrite were given credit limits lower than customers with college and High School level of education.*
- *Customers who attrited and those who did not attrite that earn 120,000 dollars and above annually were given large credit limits levels. However, those that did attrite received less credit limit on average than those that did attrite*
- *On average, customers with the silver card category that attrite were given more credit on average than those who didn't attrite.*
- *Customers with the Blue card who attrited were given lower credit limit than those who did not attrite*

Link to Appendix slide on data background check

Data Preprocessing

Duplicate value check: - Detected No duplicate rows in the dataset.

Unique Values: - The CLIENTNUM column contains unique values, and we can drop it. As it does not provide meaningful information and can be redundant for analysis.

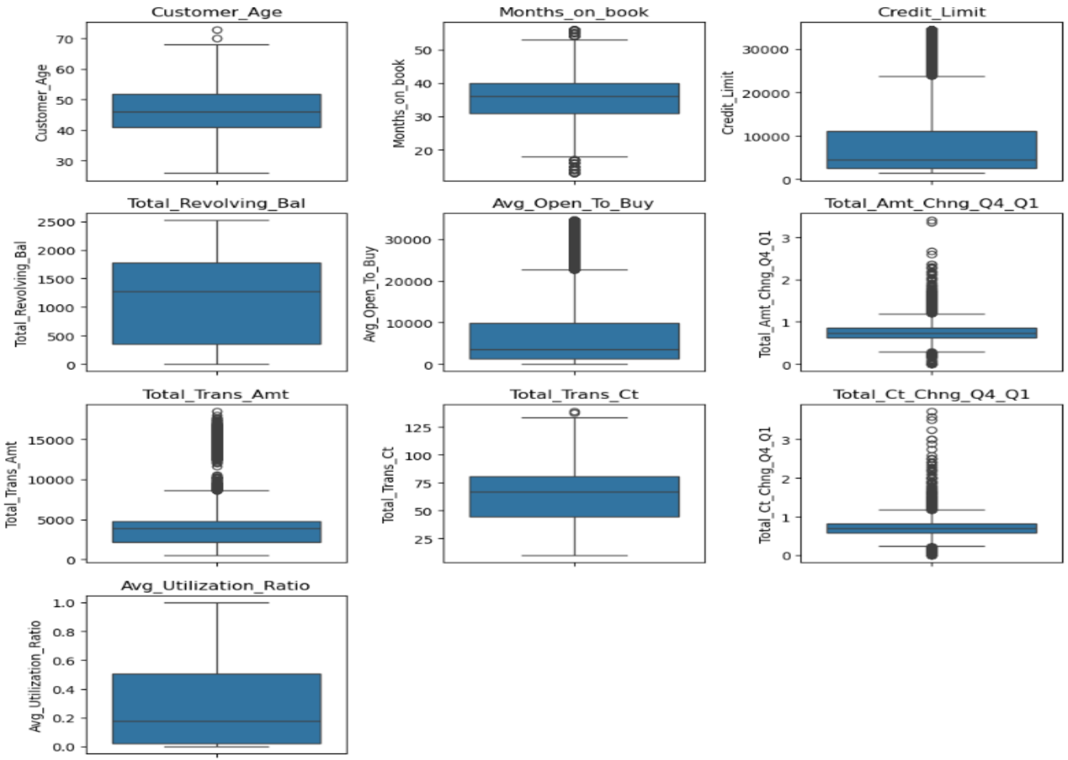
Missing value treatment: - The following columns, 'Education_Level', 'Marital_Status', 'Income_Category’ have 'unknown’ which also signify missing values. This would be treated using KNN algorithm.

Feature Engineering: - In this feature engineering step, the categorical features, including 'Attrition_Flag', 'Gender’, 'Dependent_count','Education_Level', 'Marital_Status’, 'Income_Category’, 'Card_Category', 'Total_Relationship_Count’, 'Months_Inactive_12_mon’, 'Contacts_Count_12_mon' are being optimized by converting their data type to 'category' for more efficient memory usage and enhanced representation in the dataset.

Data Leakage:- To avoid data leakage among training, testing, and validation sets. Followed below steps

1)Split data into train, test, and validation sets before any data preprocessing
2)Set a fixed random seed for data splitting.
3)Use stratified sampling for imbalanced datasets.
4)Ensure each fold maintains the same order or relevant characteristic.

Data Preprocessing – Outlier Check and Treatment



Customer_Age Outlier Treatment: Replaces values in the 'Customer_Age' column that are greater than 65 with np.nan (NaN, indicating missing values).

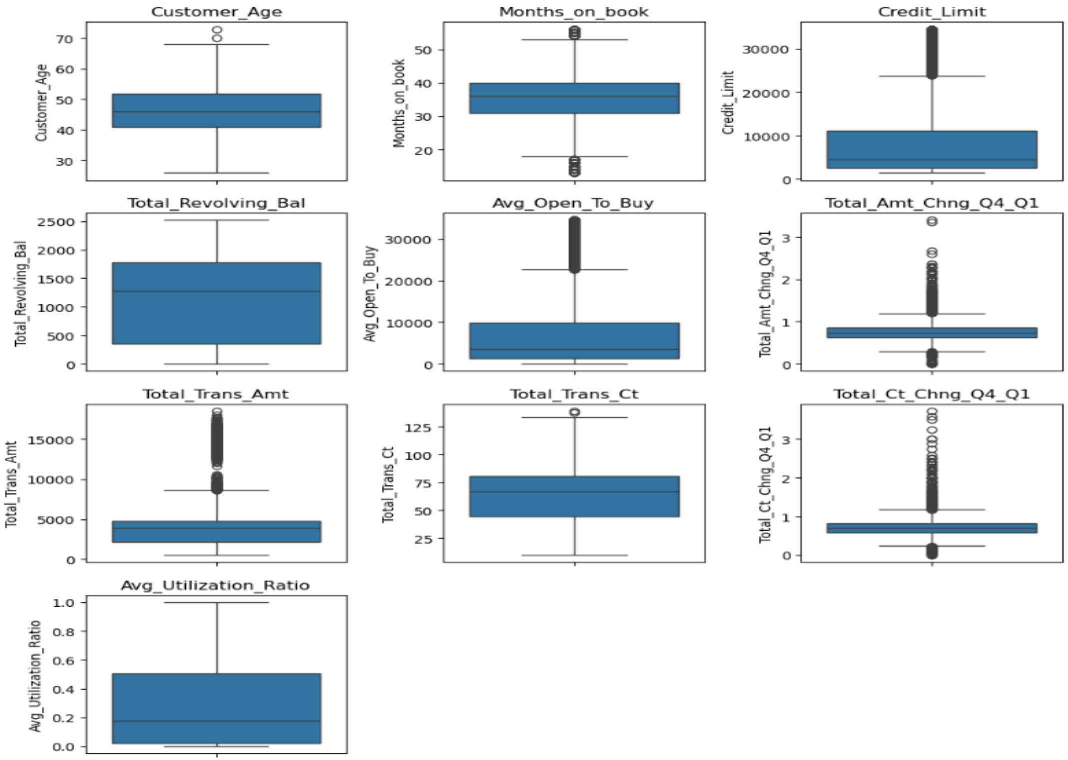
Total_Trans_Amt Outlier Treatment: Replaces values in the 'Total_Trans_Amt' column that are greater than 16990 with np.nan.

Total_Amt_Chng_Q4_Q1 Outlier Treatment: Replaces values in the 'Total_Amt_Chng_Q4_Q1' column that are greater than 2.5 with np.nan.

Total_Trans_Ct Outlier Treatment: Replaces values in the 'Total_Trans_Ct' column that are greater than 129 with np.nan.

Total_Ct_Chng_Q4_Q1 Outlier Treatment: Replaces values in the 'Total_Ct_Chng_Q4_Q1' column that are greater than 2 with np.nan.

Data Preprocessing – Outlier Check and Treatment



Customer_Age Outlier Treatment: Replaces values in the 'Customer_Age' column that are greater than 65 with np.nan (NaN, indicating missing values).

Total_Trans_Amt Outlier Treatment: Replaces values in the 'Total_Trans_Amt' column that are greater than 16990 with np.nan.

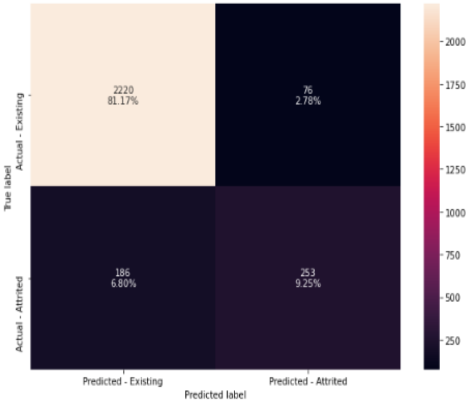
Total_Amt_Chng_Q4_Q1 Outlier Treatment: Replaces values in the 'Total_Amt_Chng_Q4_Q1' column that are greater than 2.5 with np.nan.

Total_Trans_Ct Outlier Treatment: Replaces values in the 'Total_Trans_Ct' column that are greater than 129 with np.nan.

Total_Ct_Chng_Q4_Q1 Outlier Treatment: Replaces values in the 'Total_Ct_Chng_Q4_Q1' column that are greater than 2 with np.nan.

Model Performance Summary-Logistic Regression

Initial

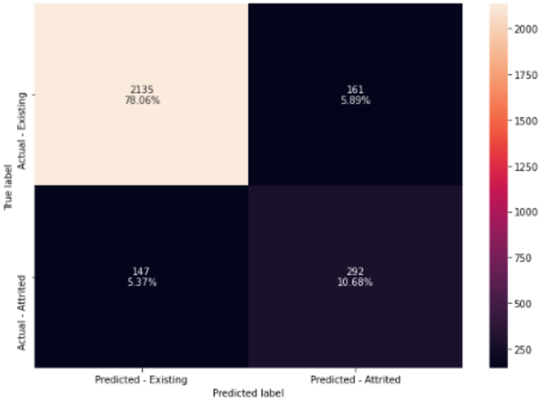


Accuracy on training set : 0.9034331399905942
Accuracy on test set : 0.9042047531992687
Recall on training set : 0.5804878048780487
Recall on test set : 0.5763097949886105
Precision on training set : 0.7618437900128041
Precision on test set : 0.7689969604863222

Observation

•This model has generalized well on training and test set however; it is underfitting on recall. This may be a result of us having an imbalanced dataset. We would need to treat the imbalance using SMOTE for undersampling and oversampling

Oversampled

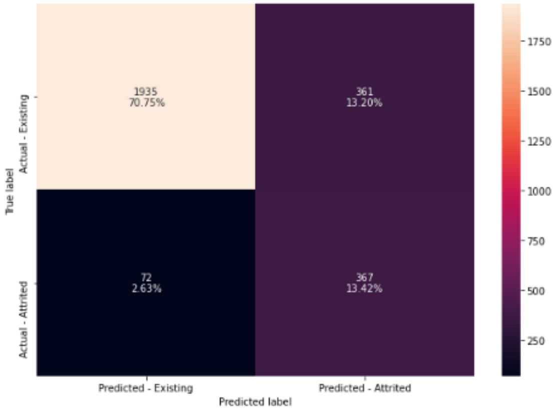


Accuracy on training set : 0.9161374673141577
Accuracy on test set : 0.8873857404021938
Recall on training set : 0.9094135225999253
Recall on test set : 0.6651480637813212
Precision on training set : 0.9218099204846649
Precision on test set : 0.6445916114790287

Observation

This model ad an improved performance on recall however, the model did not generalize well on test with an overfit recall. We would try regularization to see if overfitting can be reduced. We would try undersampling to see if we get a better performance

undersampled



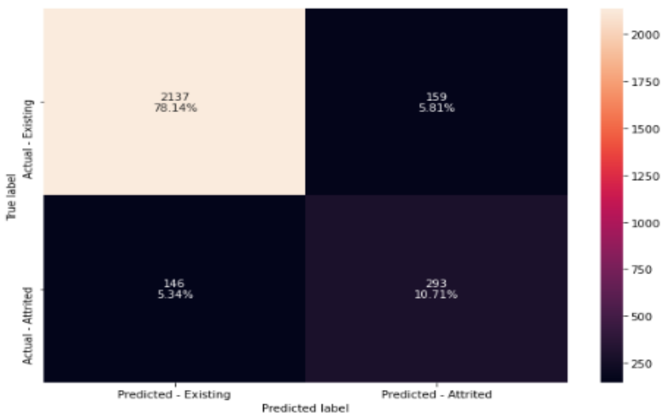
Accuracy on training set : 0.8595121951219512
Accuracy on test set : 0.8416819012797075
Recall on training set : 0.8624390243902439
Recall on test set : 0.835990888382688
Precision on training set : 0.8574199806013579
Precision on test set : 0.5041208791208791

Observation

This model generalized well on training and test set after undersampling. Our recall after undersampling on test was better than our recall after oversampling on test

Model Performance Summary-Logistic-Hyper Parameters

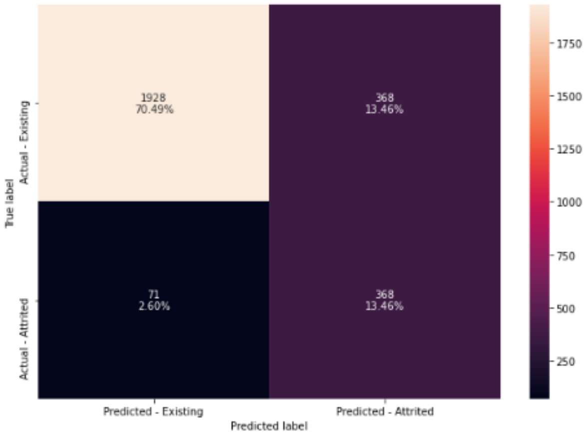
Oversampled



Accuracy on training set : 0.9164176316772507
Accuracy on test set : 0.8884826325411335
Recall on training set : 0.9097870750840493
Recall on test set : 0.6674259681093394
Precision on training set : 0.9220140071928828
Precision on test set : 0.6482300884955752

Observation
After regularization, the models recall improved slightly however, there is still overfitting

Undersampled



Accuracy on training set : 0.8497560975609756
Accuracy on test set : 0.8394881170018281
Recall on training set : 0.8526829268292683
Recall on test set : 0.8382687927107062
Precision on training set : 0.8477206595538312
Precision on test set : 0.5

Observation
After regularization on undersampled data, recall improved, generalized well and also performed better than our regularized oversampled model
This model is our best model so far

Building Models using KFold and CV Score with Pipelines

- **Building a Decision Tree, Bagging Classifier, Random Forest, Adaboost, Gradient Boost, XGBoost to see recall performance.**
- **Building these different models using KFold and cross_val_score with pipelines**

Recall Performance:

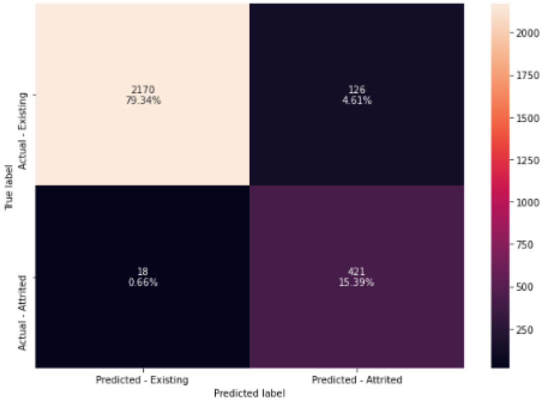
```
oversampled_BaggingClassifier: 96.61928322730961
oversampled_RandomForestClassifier: 97.9642399015681
oversampled_GradientBoostingClassifier: 97.66540136303743
oversampled_AdaBoostClassifier: 96.6195799191951
oversampled_XGBClassifier: 98.24435194638603
oversampled_DecisionTreeClassifier: 95.01308062165675

undersampled_BaggingClassifier: 90.34146341463416
undersampled_RandomForestClassifier: 93.07317073170731
undersampled_GradientBoostingClassifier: 94.53658536585365
undersampled_AdaBoostClassifier: 93.36585365853658
undersampled_XGBClassifier: 94.53658536585365
undersampled_DecisionTreeClassifier: 89.46341463414636
```

- **Hyper Parameter Tuning these best models using GridSearchCV and RandomizedSearchCV.**
- **Also Train the model on both undersampled and oversampled data and pick the one with the better performance**

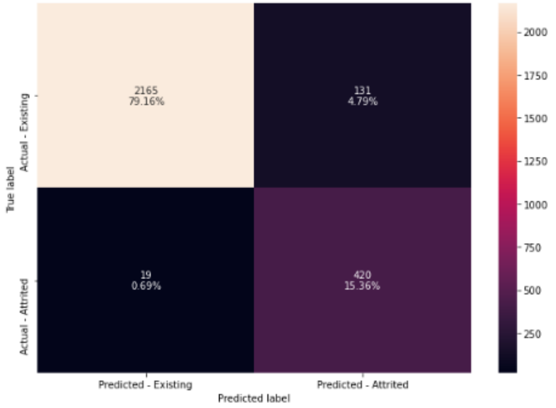
Model Performance Summary –ADA BOOST-HyperTuning

Undersampled –Grid Search

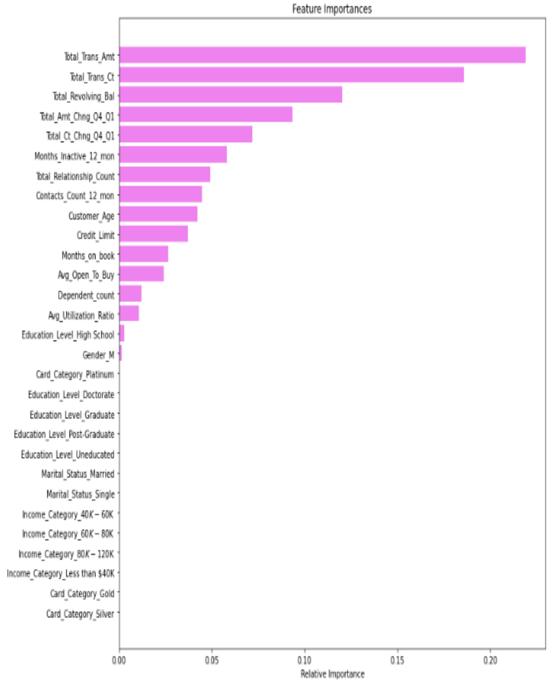


Accuracy on training set : 0.9668292682926829
Accuracy on test set : 0.9473491773308957
Recall on training set : 0.9717073170731707
Recall on test set : 0.958997722095672
Precision on training set : 0.9623188405797102
Precision on test set : 0.7696526508226691

Undersampled –Randomized Search



Accuracy on training set : 0.9921951219512195
Accuracy on test set : 0.9451553930530164
Recall on training set : 0.9921951219512195
Recall on test set : 0.9567198177676538
Precision on training set : 0.9921951219512195
Precision on test set : 0.7622504537205081

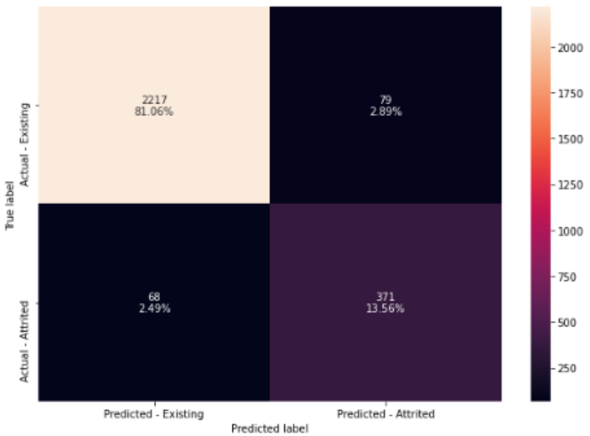


Observation:

- Our AdaBoost model tuned with GridSearch gave us the best result on recall. The model generalized well as there is no overfitting.
- The model predicted that the top 6 most important features in predicting if a credit card customer will churn is Total_Trans_Amt, Total_Trans_Ct, Total_Revolving_Bal, Total_Amt_Chng_Q4_Q1,Total_Ct_Chng_Q4_Q1 and Months_Inactive_12_mon

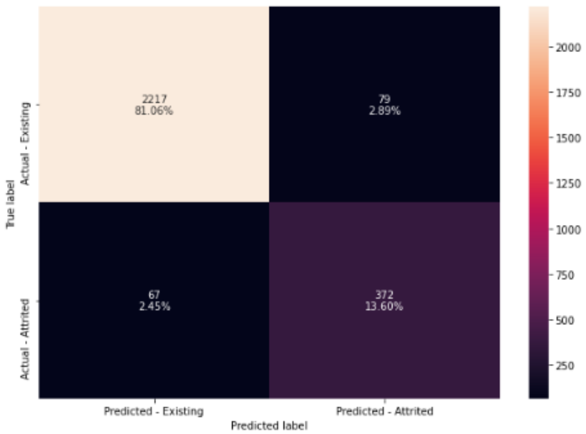
Model Performance Summary-RandomForest-HyperTuning

Oversampled - GridSearch

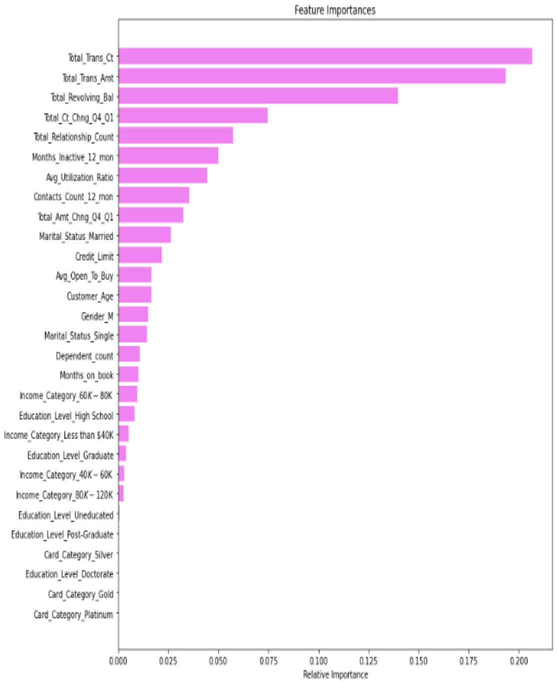


Accuracy on training set : 0.9872058274187523
Accuracy on test set : 0.9462522851919561
Recall on training set : 0.9910347403810236
Recall on test set : 0.8451025056947609
Precision on training set : 0.9835032437442076
Precision on test set : 0.8244444444444444

Oversampled -Randomized Search



Accuracy on training set : 0.9846843481509152
Accuracy on test set : 0.9466179159049360
Recall on training set : 0.9880463205080314
Recall on test set : 0.8473804100227791
Precision on training set : 0.9814471243042672
Precision on test set : 0.8248337028824834

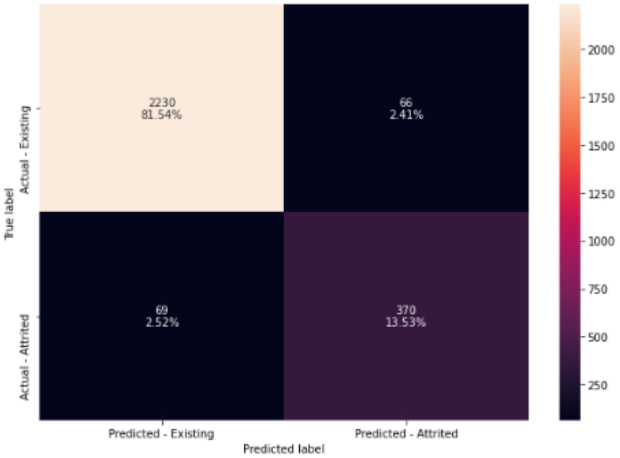


Observation:

- Our Random Forest model tuned with Randomized Search gave us the best result on recall.
- The model however did not generalize well and observed overfitting on recall.
  Part of the reason why this would have occurred is the creation of sythetic data when oversampling our data to deal with the imbalanced classes
  The model predicted that the top 3 most important features in predicting if a credit card customer will churn is Total_Trans_Ct, Total_Trans_Amt, Total_Revolving_Bal

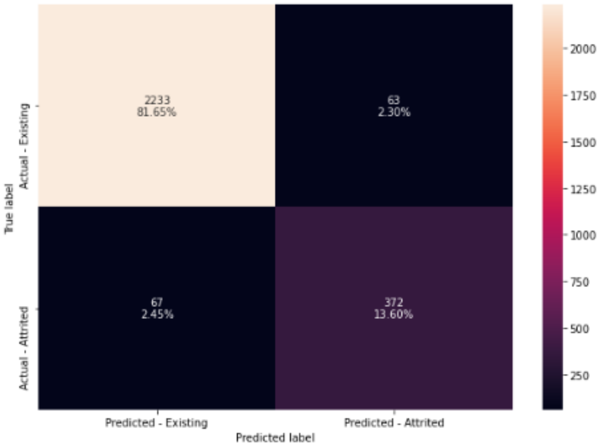
Model Performance Summary-GradientBoost-HyperTuning

Oversampled-Grid Search



Accuracy on training set : 0.9940231602540157
Accuracy on test set : 0.9506398537477148
Recall on training set : 0.9960776989166978
Recall on test set : 0.8428246013667426
Precision on training set : 0.9920014880952381
Precision on test set : 0.8486238532110092

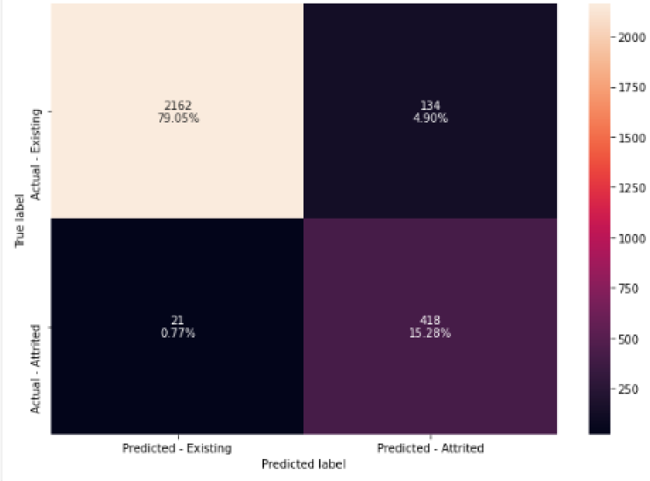
Oversample-Randomized Search



Accuracy on training set : 0.9950504295853567
Accuracy on test set : 0.9524680073126143
Recall on training set : 0.9957041464325738
Recall on test set : 0.8473804100227791
Precision on training set : 0.9944040290990487
Precision on test set : 0.8551724137931035

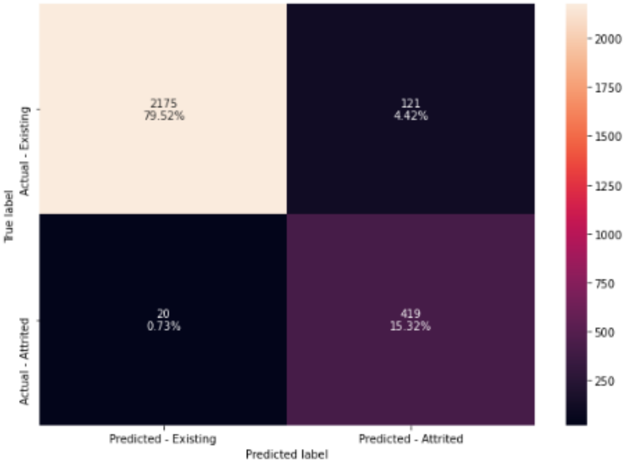
Model Performance Summary-GradientBoost-HyperTuning

Undersampled-Grid Search



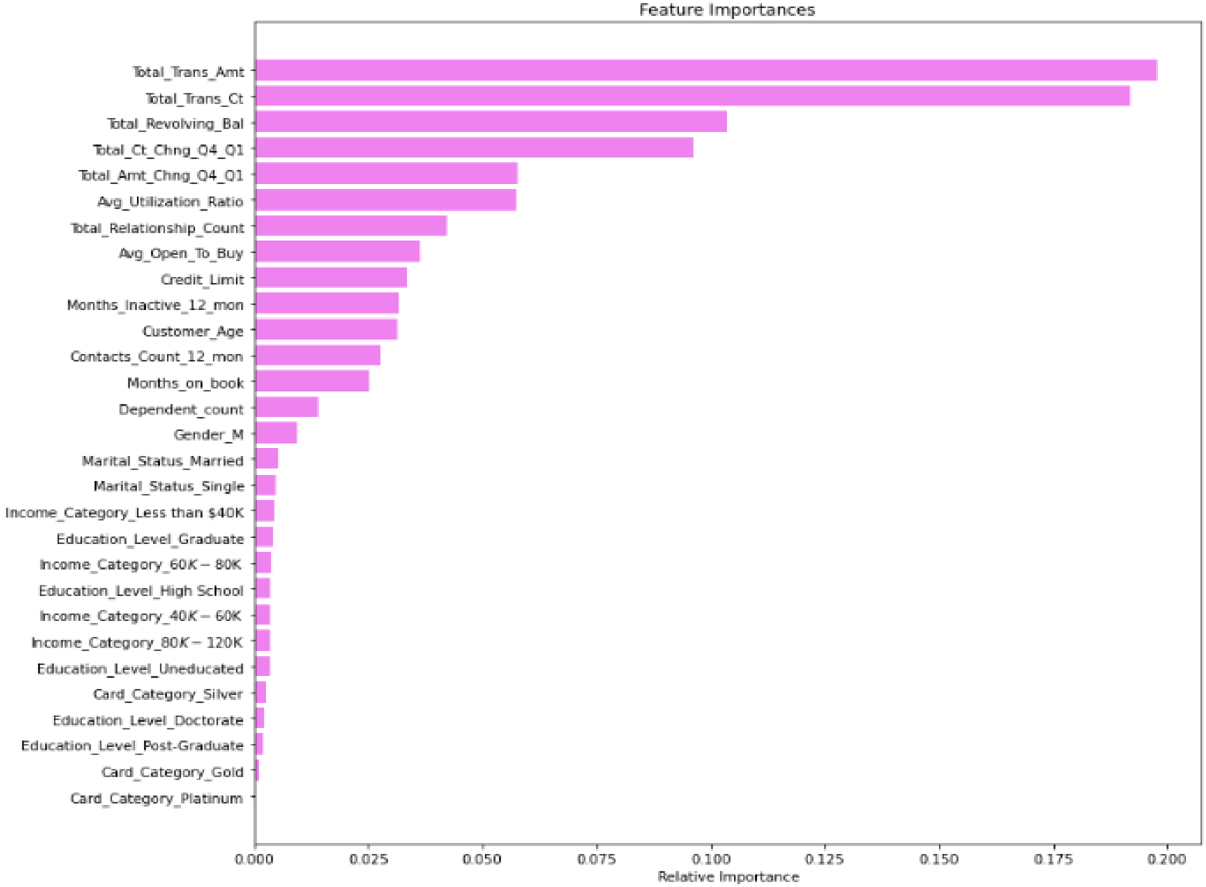
Accuracy on training set : 0.9868292682926829
Accuracy on test set : 0.943327239488117
Recall on training set : 0.9941463414634146
Recall on test set : 0.9521640091116174
Precision on training set : 0.9798076923076923
Precision on test set : 0.7572463768115942

Undersample-Randomized Search



Accuracy on training set : 1.0
Accuracy on test set : 0.9484460694698355
Recall on training set : 1.0
Recall on test set : 0.9544419134396356
Precision on training set : 1.0
Precision on test set : 0.7759259259259259

Model Performance Summary-GradientBoost-FeatureImportance

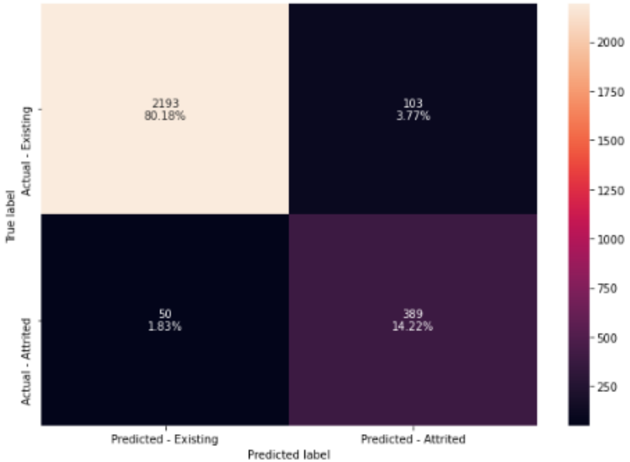


Observation:
Gradient Boost model on undersampled data tuned with Randomized Search gave us the best result on recall. The model generalized well as there is no overfitting on recall.

The model predicted that the top 2 most important features in predicting if a credit card customer will churn is Total_Trans_Amt, Total_Trans_Ct

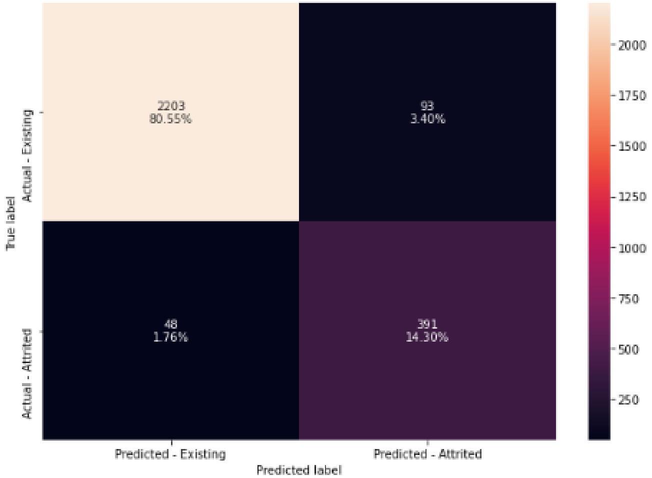
Model Performance Summary-XGBoost-Hyper Tuning

Oversampled-Grid Search



Accuracy on training set : 0.980855435188644
Accuracy on test set : 0.9440585009140768
Recall on training set : 0.9895405304445275
Recall on test set : 0.8861047835990888
Precision on training set : 0.9726454929318891
Precision on test set : 0.790650406504065

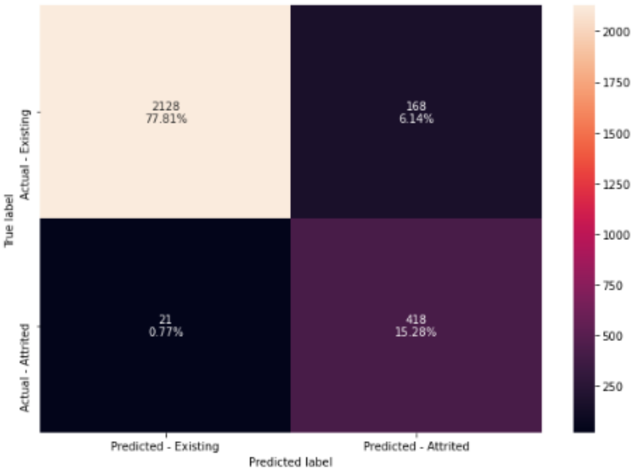
Oversample-Randomized Search



Accuracy on training set : 0.9830033619723572
Accuracy on test set : 0.9484460694698355
Recall on training set : 0.9886066492342174
Recall on test set : 0.8906605922551253
Precision on training set : 0.9776505356483192
Precision on test set : 0.8078512396694215

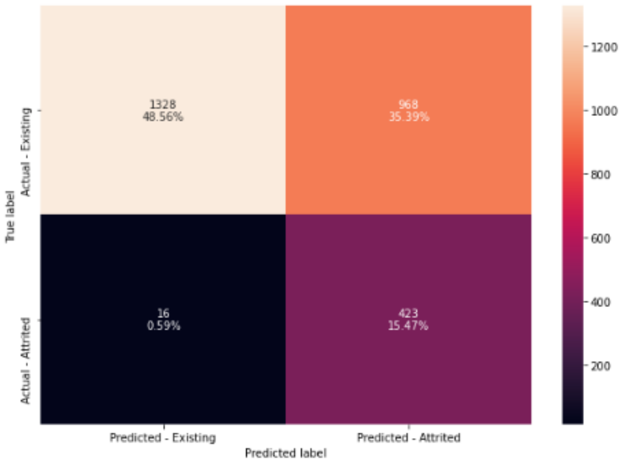
Model Performance Summary- XGBoost –Hyper Tuning

Undersampled-Grid Search



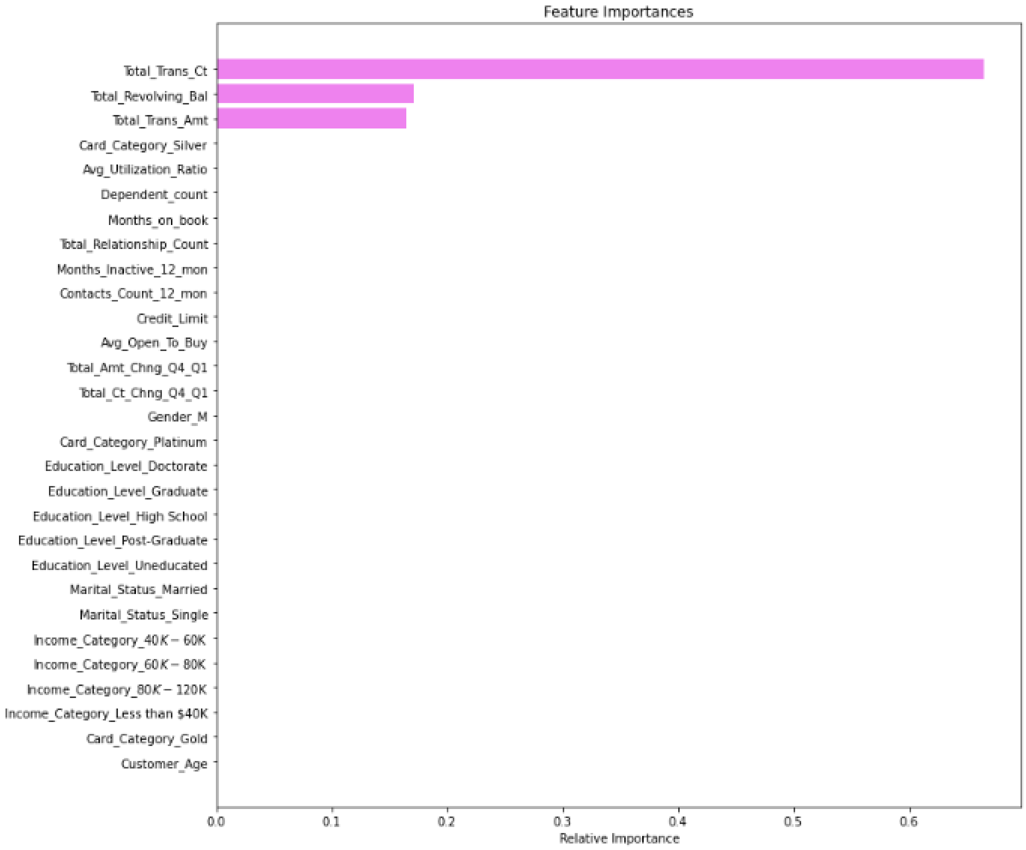
Accuracy on training set : 0.9834146341463414
Accuracy on test set : 0.9308957952468008
Recall on training set : 0.9882926829268293
Recall on test set : 0.9521640091116174
Precision on training set : 0.9787439613526570
Precision on test set : 0.7133105802047781

Undersampled-Randomized Search



Accuracy on training set : 0.7809756097560976
Accuracy on test set : 0.6402193784277788
Recall on training set : 0.9609756097560975
Recall on test set : 0.9635535307517085
Precision on training set : 0.7065997130559540
Precision on test set : 0.3040977713874910

Model Performance Summary- XGBoost –Feature Importance



Observation

- Our XGBoost model on undersampled data tuned with Randomized Search gave us the best result on recall.
  The model generalized well as there is no overfitting on recall.
  This model also had a better recall performance on test than on train which is a rare occurrence.
  This is our best model and is fit to use in production

- The model predicted that Total_Trans_Ct is the most important feature in predicting if a credit card customer will churn.
  is Total_Revolving_Bal and Total_Trans_Amt are also important but not to the extent of Total_Trans_Ct

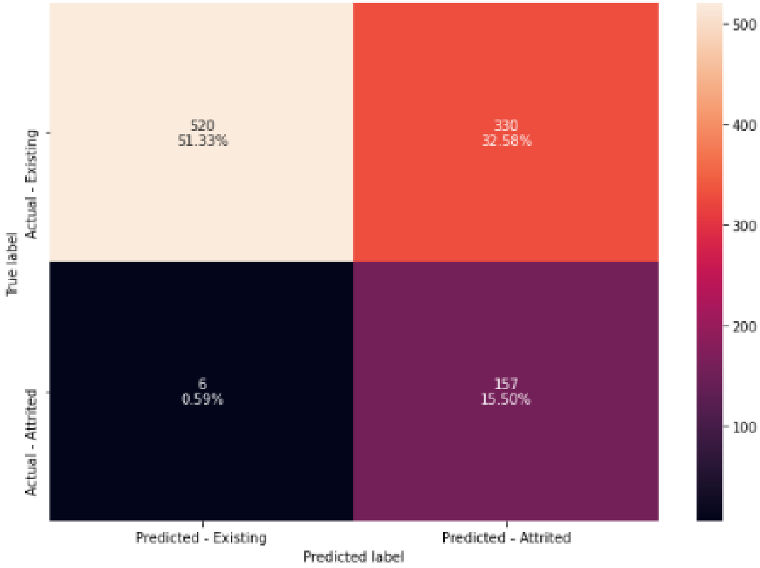
Model Performance Summary

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
0	Logistic Regression	0.903433	0.904205	0.580488	0.57631	0.761844	0.768997
1	Oversampled Logistic Regression	0.916137	0.887386	0.909414	0.665148	0.92181	0.644592
2	Oversampled Logistic Regression Regularized	0.916418	0.888483	0.909787	0.667426	0.922014	0.64823
3	Oversampled GridSearch Random Forest	0.987206	0.946252	0.991035	0.845103	0.983503	0.824444
4	Oversampled RandomizedSearch Random Forest	0.984684	0.946618	0.988046	0.84738	0.981447	0.824834
5	Oversampled GridSearch Gradient Boosting Class	0.994023	0.95064	0.996078	0.842825	0.992001	0.848624
6	Oversampled RandomizedSearch Gradient Boosting	0.99505	0.952468	0.995704	0.84738	0.994404	0.855172
7	Oversampled GridSearch XGBoost Classifier	0.980855	0.944059	0.989541	0.886105	0.972645	0.79065
8	Oversampled RandomizedSearch XGBoost Classifier	0.983003	0.948446	0.988607	0.890661	0.977651	0.807851
9	Undersampled Logistic Regression	0.849756	0.839488	0.852683	0.838269	0.847721	0.5
10	Undersampled GridSearch AdaBoost Classifier	0.966829	0.947349	0.971707	0.958998	0.962319	0.769653
11	Undersampled RandomizedSearch AdaBoost Classifier	0.992195	0.945155	0.992195	0.95672	0.992195	0.76225
12	Undersampled GridSearch Gradient Boosting Clas...	0.986829	0.943327	0.994146	0.952164	0.979808	0.757246
13	Undersampled RandomizedSearch Gradient Boostin...	1	0.948446	1	0.954442	1	0.775926
14	Undersampled GridSearch XGBoost Classifier	0.983415	0.930896	0.988293	0.952164	0.978744	0.713311
15	Undersampled RandomizedSearch XGBoost Classifier	0.780976	0.640219	0.960976	0.963554	0.7066	0.304098

Observations:

Oversampled Logistic Regression models had a poor recall performance even after regularization. These models did not generalize well on test data.
There was however improved performance after undersampling giving a much better performance on recall and generalizing well on test data.
All our models developed with oversampled data did not generalize well on test data. Part of the reason why this would have occurred is the creation of sythetic data when oversampling.
Our undersampled models generalized alot better than our oversampled models on recall.
Tuning our models with RandomizedSearch was computationally less expensive than with GridSearch.
RandomizedSearch Models also performed better than GridSearch models with exception of our AdaBoost Classifiers.
Our best model was the Undersampled RandomizedSearch XGBoost Classifier with a Test Recall score of 0.963554.
There may be yet better parameters which may result in a better performance for all models.

Model Validation for Unseen data



Observations:
Our final model, xgb_tuned_rand_un Performed well with a recall score of 0.9631901840490797 after running it on hold out data. What this means is that this model is able to identify customers that will attrite/churn 96% of the time

Accuracy on training set : 0.7809756097560976
Accuracy on test set : 0.6683119447186574
Recall on training set : 0.9609756097560975
Recall on test set : 0.9631901840490797
Precision on training set : 0.706599713055954
Precision on test set : 0.32238193018480493

Recommendations and Conclusions

- Our Undersampled Randomized Search XGBoost model gave us the best recall score which is the important metric in evaluating this particular classification problem as earlier explained. As such, we will go ahead and recommend this model as a basis for identifying credit card customers that are most likely to churn.

- From this information, we can build a customer profile of potential customers who are likely to churn. The most important feature in predicting if a credit card customer will churn according to our model is Total Revolving Balance. This the total payable amount carried over month over month by the banks credit card customers. From our Exploratory Data Analysis, we discovered that customers who had low to zero Total Revolving Balance were more likely to churn. Total Transaction Amount and Number of Total Transactions were also important features in predicting if a customer will churn as identified by the model. Customers with low total transaction amount and also low number of transactions carried out with their credit card in the last 12 months were more likely to churn as identified in our EDA.

- Retaining existing customers and thereby increasing their lifetime value is something every business considers important. There is little a bank can do about customer churn when they don't see it coming in the first place. This is why predicting customer churn early enough is important especially when clear customer feedback is unavailable. Early and accurate churn prediction empowers the banks CRM and customer experience teams to be creative and proactive in their engagement with the customer. By reaching out to them alone, the bank will have greatly reduced the chances of churn for some customers. A model like ours allows the bank to do this and be accurate 96% of the time. Such a model is the first step in a potential customer 360 view. It can serve as a foundation for other use cases such as cross-sell/upsell recommendation, customer lifetime value calculation, etc.

Great Learning

Happy Learning !