Thera Bank Credit Card Churn Classification Model

Project #3- Advanced Machine Learning

Saturday, September 7, 2024

Presented by: Sarah Lasater

Contents / Agenda

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

Executive Summary

Executive Summary

Based on the observations and results from the predictive model evaluations, here are some business insights and conclusions:

BUSINESS INSIGHTS

Model Performance and Recall Scores:

- **XGBoost with Original Data:** This model achieved the highest Recall score of 1.00 on the training set, indicating perfect performance in correctly identifying positive cases within the training data. However, its performance on the validation set dropped to 0.921, showing some potential overfitting. Despite this, it performed best on the test set with a Recall score of 0.99, highlighting its strong generalization capabilities on unseen data.
- **AdaBoost and Gradient Boosting with Undersampled Data:** Both models performed well on the training set and achieved identical Recall scores of 0.941 on the validation set. However, their performance was not as high as XGBoost on the test set. This suggests that while these models are robust with undersampled data, they do not generalize as effectively as XGBoost.
- **Tuned Models:** The tuned XGBoost model demonstrated the best performance on unseen data with a test Recall score of 0.99, indicating it is highly effective at identifying positive cases in the test dataset. The next best models were tuned AdaBoost (0.990) and tuned Gradient Boosting (0.984), suggesting these models are also strong contenders for practical deployment and if time constraints are a major concern to Thera Bank.

Executive Summary

Feature Importance:

The XGBoost model highlighted several critical features:

- ***Total_Trans_Ct and Total_Trans_Amt:*** Reflect customer engagement and spending behavior.
- ***Total_Revolving_Bal:*** Indicates credit utilization, which is crucial for assessing credit risk.
- ***Total_Relationship_Count and Months_Inactive_12_mon:*** Provide insight into the customer's engagement and activity level.
- ***Card_Category:*** The type of card (Platinum) is a significant predictor.

These features should be closely monitored and leveraged in decision-making processes, as they provide valuable insights into customer behavior and credit risk.

Executive Summary

CONCLUSIONS

Final Model Selection:

- **XGBoost: Given its strong performance across training, validation, and test datasets, XGBoost appears to be the most robust model for identifying positive cases. Its ability to generalize well on unseen data makes it a preferred choice for practical applications.**
- **AdaBoost and Gradient Boosting:** These models also show promising results, particularly with undersampled data. While they perform well on the training and validation sets, they slightly lag behind XGBoost on the test set. They could be used as secondary options or in conjunction with XGBoost to further improve upon model performance.
- **Feature Utilization:** Focus on features like transaction count, credit card balance, and card type, which are highly predictive of the target outcome. Incorporating these features into business strategies can enhance customer targeting and risk assessment processes.
- **Data Handling:** The use of undersampled data has shown to produce competitive models, but the original data used with XGBoost produced the best overall performance. It's important to balance data sampling techniques with model selection to achieve optimal results.

Executive Summary

CONCLUSIONS

Next Steps:

- Further fine-tuning of the XGBoost model could potentially improve its performance even more.
- It may also be beneficial to explore ensemble methods that combine multiple models to leverage their strengths and mitigate their weaknesses.
- Continuous monitoring of feature importance and model performance is essential for maintaining the effectiveness of predictive models in dynamic business environments.

By leveraging these insights, businesses can make informed decisions on model deployment and feature utilization, ultimately enhancing their predictive capabilities and operational efficiency.

Business Problem Overview and Solution Approach

Business Problem Overview and Solution Approach

Problem Overview:

Thera Bank has recently experienced a significant decline in credit card usage among its customers, leading to a loss of revenue. To address this issue, the bank aims to analyze customer data to identify those at risk of discontinuing their credit card services and understand the underlying reasons for their departure. By developing a classification model, Thera Bank seeks to enhance its services and retention strategies, ensuring that customers are less likely to abandon their credit cards. This analysis will enable the bank to focus on key areas for improvement, ultimately fostering customer loyalty and increasing revenue.

Business Problem Overview and Solution Approach

Solution Approach / Methodology:

Thera Bank should aim to maximize Recall in its classification model. A higher Recall score indicates a reduced likelihood of false negatives, which is crucial for identifying customers at risk of attrition. By focusing on increasing Recall, the bank can better retain valuable customers by accurately identifying those who may leave.

Failing to predict a customer's attrition poses a significant risk, as it may lead to the loss of valuable customers. Therefore, enhancing the model's ability to correctly identify at-risk customers is essential for preserving customer loyalty and protecting the bank's assets.

In this analysis, data from both current and attrited customers of Thera Bank will be evaluated and preprocessed. This data will be split into training and testing sets, allowing us to train several machine learning models. These models will then be validated and tested on unseen data to identify the best model for predicting customer attrition.

Exploratory Data Analysis (EDA)

Univariate EDA

This advanced machine learning project is centered on an in-depth analysis of customer data from Thera Bank, aiming to uncover key patterns and insights that will inform strategic decisions in regard to their credit card offerings and operations. The exploratory data analysis (EDA) phase provided a comprehensive understanding of the dataset's univariate features, revealing significant trends and distributions across various customer demographics and account-related metrics.

Based on the EDA of the univariate data features of the dataset provided by Thera Bank, here are some observations and conclusions:

Univariate EDA- Observations

1. **Customer Age:**
   - The average and median customer age is 46 years, with the majority (50%) between 41 to 52 years. There are a few outliers above 70 years, indicating a potential for targeted services towards an aging customer base. *(See U1)*
2. **Age of Account:**
   - The average account age is 36 months, with a notable 50% of accounts aged between 31 to 40 months. Outliers exist with accounts older than 53 months, suggesting long-term customer loyalty or, on the other hand, potential stagnation. *(See U2)*
3. **Credit Limit:**
   - The distribution of credit limits is highly right-skewed, with an average of $8,632 and a median of $4,550. A significant number of outliers above $23,000 indicate a small group of customers with substantially higher credit access, warranting a closer look at risk and reward dynamics in this segment. *(See U3)*
4. **Revolving Balance:**
   - The average revolving balance is $1,163, with a median of $1,276. The majority of balances are concentrated below $1,276, but some customers carry much higher balances, which could represent higher risk or higher profitability depending on their payment behaviors. *(See U4)*
5. **Open to Buy:**
   - Similar to credit limits, the Open to Buy feature is right-skewed, with an average of $7,470 and a median of $3,475. The data suggests that while most customers have modest credit availability, a subset has significantly higher credit access. *(See U5)*

Univariate EDA- Observations

6. Transaction Behavior:

- The total transaction count and transaction amounts show a wide range, with a substantial number of customers engaging in high-frequency transactions or large transaction amounts, which may correlate with customer engagement and profitability. *(See U6 & U8)*

7. Utilization Ratio:

- The average utilization ratio is 0.275, with half of the customers falling between 0.025 and 0.5. The distribution indicates that while most customers are not maxing out their credit, a significant minority are using a large portion of their available credit, which may highlight potential areas of financial stress or opportunity. *(See U10)*

8. Demographics:

- The dataset includes a balanced mix of male (5,358) and female (4,769) customers, with diverse educational backgrounds ranging from uneducated to doctorate holders. The majority of customers are married (4,687), followed by single (3,943), and divorced (748). *(See U15, U16, &U17)*

9. Income and Card Categories:

- Income distribution skews towards lower income categories, with 35% earning less than $40K annually. The majority of customers hold Blue cards (93%), with very few in higher-tier Silver, Gold, or Platinum categories, which may indicate opportunities for upselling. *(See U18 & U19)*

10. Customer Attrition:

- The data reveals that 1,627 customers have left the institution, representing a churn rate of approximately 16%. Understanding the factors driving this attrition will be critical for developing effective retention strategies. *(See U20)*

Univariate EDA- Conclusions

1. **Customer Demographics:**
   - The customer base is generally middle-aged, with most customers falling between 41 to 52 years old. The presence of older customers (above 70 years) suggests that there may be opportunities to cater to an aging demographic, possibly through targeted products or services.
   - The age of account data shows that many customers have been with the institution for a moderate length of time (31 to 40 months), indicating a stable customer base. However, outliers with longer tenure suggest a small group of loyal, long-term customers who could be valuable for retention-focused strategies.

1. **Credit Limits and Usage:**
   - Credit limits are highly skewed, with a small percentage of customers having access to significantly higher credit limits. This suggests a diverse customer base, with different segments likely requiring different risk management strategies.
   - The revolving balance and utilization ratio data indicate that while most customers maintain relatively low balances and utilization rates, there are customers who carry high balances or utilize a significant portion of their credit. This group may represent higher risk but also an opportunity for increased interest revenue if managed properly.

1. **Customer Behavior:**
   - Transaction data (both the counts and the amounts) shows significant variability, with some customers engaging in frequent and/or high-value transactions. This indicates differing levels of engagement, suggesting that personalized marketing or loyalty programs could prove effective in encouraging increased activity among less engaged customers.
   - The open-to-buy metric, which is also right-skewed, implies that while many customers have limited remaining credit, a subset has a considerable amount of available credit, which could be leveraged for upselling or promotional offers.

Univariate EDA- Conclusions

4. **Customer Segmentation:**
   - The data on dependents, total relationship count, and months inactive provides a strong basis for customer segmentation. For instance, customers with a higher number of dependents or those with fewer months inactive may be more engaged and could be targeted for cross-selling or retention campaigns.
   - The analysis of educational levels and income categories reveals a broad spectrum of customer backgrounds. With a large portion of customers in the lower income brackets and with a graduate-level education, tailored financial products could be developed to meet their specific needs.

5. **Attrition and Retention:**
   - The churn rate of 16% is significant and suggests that understanding the factors leading to customer attrition is crucial. The univariate analysis provides insights that could help identify at-risk customers and develop strategies to improve retention.

6. **Product and Service Opportunities:**
   - The vast majority of customers hold basic Blue cards, with very few in higher-tier categories (Silver, Gold, Platinum). This indicates potential for product diversification and upselling to customers who might benefit from more premium services.
   - There is a need for targeted efforts to move customers up the value chain by offering them products that align with their credit behavior, income, and overall engagement with the institution.

Univariate EDA- Conclusions

Overall Univariate EDA Conclusions

The exploratory data analysis underscores the importance of segmenting Thera Bank's customers based on key features such as credit limit, transaction behavior, and demographic attributes to tailor services and marketing efforts. The diverse customer base, with varying levels of credit usage and engagement, suggests a need for targeted segmentation, personalized marketing, and a focus on both retention and upselling strategies. The insights from this analysis will drive the development of predictive models to identify high-risk customers, optimize credit offerings, and reduce attrition, ultimately enhancing Thera Bank's customer lifetime value and profitability. Addressing outliers in credit usage and customer attrition will be crucial for improving their overall profitability and reducing risk.

Bivariate EDA

This project also required an in-depth analysis of Thera Bank's bivariate customer data from Thera Bank. The EDA phase again provided a comprehensive understanding of the dataset's bivariate features, revealing additional details of trends and distributions across known demographics and metrics.

Based on the EDA of the bivariate data features of the dataset, here are some observations and conclusions:

Bivariate EDA- Observations

Correlation Insights: *(See B1)*

- **Total Transaction Count and Total Transaction Amount** show a strong positive correlation (0.81), indicating that as transaction counts increase, the transaction amount also tends to rise significantly.
- **Months on Book and Customer Age** also have a high positive correlation (0.79), suggesting that older customers tend to have been with the bank longer.
- **Total Revolving Balance and Average Utilization Ratio** are positively correlated (0.62), indicating that higher balances are associated with higher utilization ratios.
- Conversely, **Average Open to Buy and Average Utilization Ratio** have a moderate negative correlation (-0.54), suggesting that as available credit increases, utilization decreases.
- **Credit Limit and Average Utilization Ratio** similarly show a negative correlation (-0.48), reinforcing the relationship between credit limits and utilization.

Bivariate EDA- Observations

1. **Attrition Status by Demographics:**
   - **Gender**: More female customers exist both in total and attrited categories (4,428 females vs. 4,072 males). This indicates a higher likelihood of attrition among females. *(See B2)*
   - **Marital Status**: Married customers show a high retention rate, with 3,978 not attrited versus 709 attrited. Single and divorced customers have fewer total numbers and higher attrition rates. *(See B3)*
2. **Education Level:**
   - Graduate customers have the highest retention (2,641 not attrited) compared to other education levels, while the uneducated group shows significant attrition. *(See B4)*
3. **Income Category Insights:**
   - Customers earning less than $40K have the highest attrition rate (612 attrited), while those earning $120K+ have the lowest (126 attrited), indicating that higher income correlates with lower attrition. *(See B5)*
4. **Engagement Metrics**:
   - **Contacts Count**: Customers who have had more contacts in the last 12 months tend to not attrite. For example, those with 5 or more contacts show higher retention. *(See B6)*
   - **Months Inactive**: The more months a customer is inactive, the higher their likelihood of attrition, emphasizing the importance of engagement. *(See B7)*
5. **Transaction Behavior:**
   - Higher transaction counts and amounts correlate with lower attrition. For instance, customers with the highest transaction count (over 75) are predominantly retained, while those with significantly lower transaction counts tend to attrite. *(See B8)*
6. **Credit Utilization:**
   - **Average Utilization Ratio**: Customers who are not attrited generally have higher average utilization ratios, while attrited customers show a right-skewed distribution indicating a potential issue with credit management. *(See B16)*

Bivariate EDA- Conclusions

1. **Identifying High-Risk Segments:**
    - The analysis reveals that specific demographic groups, such as lower-income customers and those with less education, exhibit higher attrition rates. Thera Bank should focus on these segments to develop targeted retention strategies, possibly through tailored communication and personalized offers.
2. **Engagement and Activity Monitoring:**
    - Customers with higher engagement metrics, such as a greater number of contacts and lower inactivity rates, tend to stay with the bank. Monitoring these factors can help Thera Bank identify at-risk customers early and intervene with retention efforts, such as outreach programs or personalized financial advice.
3. **Utilization and Credit Management:**
    - The relationship between average utilization ratios and attrition suggests that customers with lower utilization might be disengaged. Thera Bank could implement educational initiatives to encourage better credit usage, ensuring customers understand how to manage their accounts effectively.
4. **Transaction Behavior as a Predictor:**
    - Higher transaction counts correlate with lower attrition, indicating that active customers are less likely to leave. Encouraging more transactions through rewards programs or promotional offers can help keep customers engaged and reduce the likelihood of attrition.
5. **Leveraging Customer Profiles:**
    - The data on customer age, months on book, and transaction history can be integrated into predictive modeling. By analyzing these profiles, Thera Bank can develop models to forecast which customers are likely to attrite based on historical data and behavior patterns.

Bivariate EDA- Conclusions

6. **Strategic Marketing Initiatives**:
   - Insights regarding marital status and gender suggest that marketing strategies should be tailored. For instance, creating campaigns that resonate with married customers or addressing the needs of female customers could enhance engagement and reduce attrition.
7. **Retention Programs for Specific Income Brackets**:
   - Since customers earning less than 40K are at a higher risk of attrition, Thera Bank should consider specialized programs for this demographic, such as financial literacy workshops, budgeting assistance, or tailored loan products that align with their financial capabilities.
8. **Continuous Monitoring and Adaptation**:
   - Regularly updating and analyzing the attrition data will allow Thera Bank to adapt its strategies proactively. Employing machine learning algorithms to predict churn based on real-time data can enhance the bank's ability to mitigate potential losses.
9. **Feedback Mechanisms**:
   - Establishing channels for customer feedback can provide insights into why customers might consider leaving. Understanding their concerns and addressing them promptly can significantly impact retention.

Bivariate EDA- Conclusions

Overall Bivariate EDA Conclusions

For Thera Bank, the observed correlations and demographic trends offer valuable insights for targeted retention strategies. By focusing on enhancing engagement with lower-income, inactive, and less-educated customer segments, the bank can effectively reduce attrition rates. Personalized communication and tailored offers, informed by transaction behavior and customer demographics, are likely to improve retention, especially among high-risk groups identified through this analysis. Ongoing monitoring of these features will facilitate adjustments to marketing and service strategies, helping to maintain customer loyalty and minimize attrition.

To further strengthen customer engagement, Thera Bank should develop a comprehensive strategy that leverages the identified predictors of attrition. Implementing machine learning models will allow for continuous analysis of customer data, enabling the prediction of attrition risks and timely interventions. Additionally, creating targeted marketing campaigns based on demographic and behavioral insights can foster loyalty among high-risk groups. It's essential to monitor the effectiveness of these retention strategies and make necessary adjustments based on customer feedback and attrition rates. By adopting these conclusions and recommendations, Thera Bank can enhance its customer retention efforts, ultimately improving overall customer satisfaction and reducing attrition rates.

Data Preprocessing

Data Preprocessing

- **Duplicate value check**
  - No duplicates entries were present in the original dataset.

- **Outlier check**
  - After finding the 25th and 75th percentiles, the Interquartile Range, and the lower and upper bounds for all values, the following features in the dataset were found to contain outliers:
    - Attrition_Flag, Customer_Age, Months_on_Book, Months_Inactive_12_mon, Contacts_Count_12_mon, Credit_Limit, Avg_Open_To_buy, Total_Amt_Chng_Q4_Q1, Total_Trans_Amt, Total_Trans_Ct, Total_Ct_Chg_Q4_Q1 *(See DP1)*

- **Missing value treatment**
  - There were 1,519 missing values for “Education Level” and 749 “Marital Status” categories. During Train-Test split, it was found that there were 1,112 anomalous values of “abc” in the “Income Category” category. These values were replaced with NaN and a simple imputer of “most_frequent” was used to impute the missing values in X_train, X_val, and X_test sets. *(See DP2)*

Data Preprocessing

- **Feature engineering**
  - “Attrition Flag” was dropped from the X train data set and was used as the sole feature in the Y train data set.
  - After encoding of categorical variables in X_train, X_val, and X_test by dummy imputation, there were now 29 columns in the data set instead of the original 19 after data split.

- **Data preparation for modeling**
  - The data was split twice (80:20 ratio / 75:25 ratio) into three sets for train/ validation/ test:
    - X_train: 8,101 rows, 19 columns *(See DP3)*
    - X_val: 2,532 rows, 19 columns *(See DP4)*
    - X_test: 7,595 rows, 19 columns *(See DP5)*

Model Performance Summaries

Model Performance Summary- Tuned AdaBoost using Original Data (tuned_adb)

Classifier

AdaBoostClassifier
base_estimator: DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, random_state=1)
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, random_state=1)

Training Set Performance

	Accuracy	Recall	Precision	F1
0	0.983	0.925	0.965	0.945

Validation Set Performance

	Accuracy	Recall	Precision	F1
0	0.973	0.870	0.957	0.911

Best Parameters for Hyperparameter Tuning

```
Best parameters are {'n_estimators': 100, 'learning_rate': 0.1, 'base_estimator': DecisionTreeClassifier(max_depth=3, random_state=1)} with CV score=0.8632802829354553:
CPU times: user 6.08 s, sys: 438 ms, total: 6.51 s
Wall time: 2min 43s
```

- The recall on the training set is strong (92.5%), suggesting it can identify a high proportion of actual positive cases.
- However, there is a noticeable drop in recall during validation (87.0%), which may indicate some overfitting, as the model performs slightly worse on unseen data.
- Overall, the AdaBoost model is effective, but further tuning may be necessary to improve recall on validation data, ensuring better generalization to unseen cases.

Link to Appendix slide on model assumptions

Model Performance Summary- Tuned AdaBoost using Undersampled Data (tuned_adb2)

Classifier

```
AdaBoostClassifier
AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=3,
                                                         random_state=1),
                   learning_rate=0.1, n_estimators=100, random_state=1)
base_estimator: DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, random_state=1)
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, random_state=1)
```

Training Set Performance

	Accuracy	Recall	Precision	F1
0	0.962	0.991	0.815	0.895

Validation Set Performance

	Accuracy	Recall	Precision	F1
0	0.949	0.941	0.786	0.857

Best Parameters for Hypertuning

```
Best parameters are {'n_estimators': 100, 'learning_rate': 0.1, 'base_estimator': DecisionTreeClassifier(max_depth=3, random_state=1)} with CV score=0.8632802829354553:
CPU times: user 6.08 s, sys: 438 ms, total: 6.51 s
Wall time: 2min 43s
```

- The model achieved a recall of 0.991 on the training set, indicating that it is very effective at identifying true positives, or customers at risk of attrition.
- The recall of around 99% in training and 94.1% in validation indicates that the model performs well overall.
- While the model excels at recall, efforts may be needed to enhance the relatively low precision score and balance the performance further.

Link to Appendix slide on model assumptions

Model Performance Summary- Tuned Gradient Boost using Oversampled Data (tuned_gbm2)

Classifier

```
GradientBoostingClassifier
GradientBoostingClassifier(init=AdaBoostClassifier(random_state=1),
                           max_features=0.5, random_state=1, subsample=0.9)
    init: AdaBoostClassifier
    AdaBoostClassifier(random_state=1)
        AdaBoostClassifier
        AdaBoostClassifier(random_state=1)
```

Training Set Performance

	Accuracy	Recall	Precision	F1
0	0.921	0.849	0.991	0.915

Validation Set Performance

	Accuracy	Recall	Precision	F1
0	0.962	0.799	0.956	0.870

Best Parameters for Hypertuning

```
Best parameters are {'subsample': 0.9, 'n_estimators': 100, 'max_features': 0.5, 'learning_rate': 0.1, 'init': AdaBoostClassifier(random_state=1)} with CV score=0.8287297376952549:
CPU times: user 6.1 s, sys: 601 ms, total: 6.7 s
Wall time: 3min 56s
```

- While the training scores mostly indicate a strong fit to the training data, the recall score of 84.9% suggests some missed positive cases.
- Despite the high precision score, the recall drop in validation at 79.9% indicates the model may struggle with unseen data.
- Overall, while the model performs well on training data, the validation results highlight potential overfitting, suggesting a need for further tuning or evaluation of its generalization capabilities.

Link to Appendix slide on model assumptions

Model Performance Summary- Tuned Gradient Boost using Undersampled Data (tuned_gbm1)

Classifier

GradientBoostingClassifier
init: AdaBoostClassifier
AdaBoostClassifier(random_state=1)
AdaBoostClassifier
AdaBoostClassifier(random_state=1)

Training Set Performance

	Accuracy	Recall	Precision	F1
0	0.977	0.984	0.970	0.977

Validation Set Performance

	Accuracy	Recall	Precision	F1
0	0.947	0.941	0.777	0.851

Best Parameters for Hypertuning

```
Best parameters are {'subsample': 0.9, 'n_estimators': 100, 'max_features': 0.5, 'learning_rate': 0.1, 'init': AdaBoostClassifier(random_state=1)} with CV score=0.9592956086059534:
CPU times: user 2.88 s, sys: 279 ms, total: 3.16 s
Wall time: 1min 42s
```

- With a recall of 0.984 in training, the model effectively identifies most true positives, suggesting it's reliable in detecting customers at risk of attrition.
- The model shows strong recall on both training (98.4%) and validation (94.1%) datasets, indicating good generalization.
- Overall, the Gradient Boost model performs well, especially in training, but the drop in precision during validation suggests potential challenges in generalizing to unseen data. Improvements in model tuning or data handling may enhance performance further.

Link to Appendix slide on model assumptions

Model Performance Summary- Tuned XGBoost using Original Data (tuned_xgb)

Classifier

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, gamma=1, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=0.1, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=100,
              n_jobs=None, num_parallel_tree=None, random_state=1, ...)
```

Training Set Performance

	Accuracy	Recall	Precision	F1
0	0.992	1.000	0.951	0.975

Validation Set Performance

	Accuracy	Recall	Precision	F1
0	0.968	0.921	0.884	0.903

Best Parameters for Hypertuning

```
Best parameters are {'subsample': 0.7, 'scale_pos_weight': 5, 'n_estimators': 100, 'learning_rate': 0.1, 'gamma': 3} with CV score=0.9400913645741232:
CPU times: user 2.68 s, sys: 294 ms, total: 2.97 s
Wall time: 1min 20s
```

- The XGBoost model trained on the original data demonstrated impressive performance metrics. During training, it achieved a perfect recall score of 1.00, indicating that, despite possible overfitting, the model successfully identified all actual positive cases.
- In terms of validation, the model recall score dropped slightly to 0.921 but still showed that it still effectively identified a significant majority of positive cases.
- Overall, the XGBoost model exhibits excellent training and validation capabilities, making it a strong candidate for predicting customer attrition.

Link to Appendix slide on model assumptions

Model Training & Validation Performance Comparison

Training performance comparison:

	AdaBoost trained with Original data	AdaBoost trained with Undersampled data	Gradient boosting trained with Undersampled data	Gradient boosting trained with Original data	XGBoost trained with Original data
Accuracy	0.983	0.962	0.977	0.921	0.992
Recall	0.925	0.991	0.984	0.849	1.000
Precision	0.965	0.815	0.970	0.991	0.951
F1	0.945	0.895	0.977	0.915	0.975

Validation performance comparison:

	AdaBoost trained with Original data	AdaBoost trained with Undersampled data	Gradient boosting trained with Undersampled data	Gradient boosting trained with Original data	XGBoost trained with Original data
Accuracy	0.973	0.949	0.947	0.962	0.970
Recall	0.870	0.941	0.941	0.799	0.921
Precision	0.957	0.786	0.777	0.956	0.895
F1	0.911	0.857	0.851	0.870	0.908

Link to Appendix slide on model assumptions

Model Testing Performance Comparison

```
adb2_test = model_performance_classification_sklearn(tuned_adb2, X_test, y_test)
adb2_test
```

	Accuracy	Recall	Precision	F1
0	0.963	0.990	0.817	0.895

```
gbm1_test = model_performance_classification_sklearn(tuned_gbm1, X_test, y_test)
gbm1_test
```

	Accuracy	Recall	Precision	F1
0	0.959	0.984	0.803	0.884

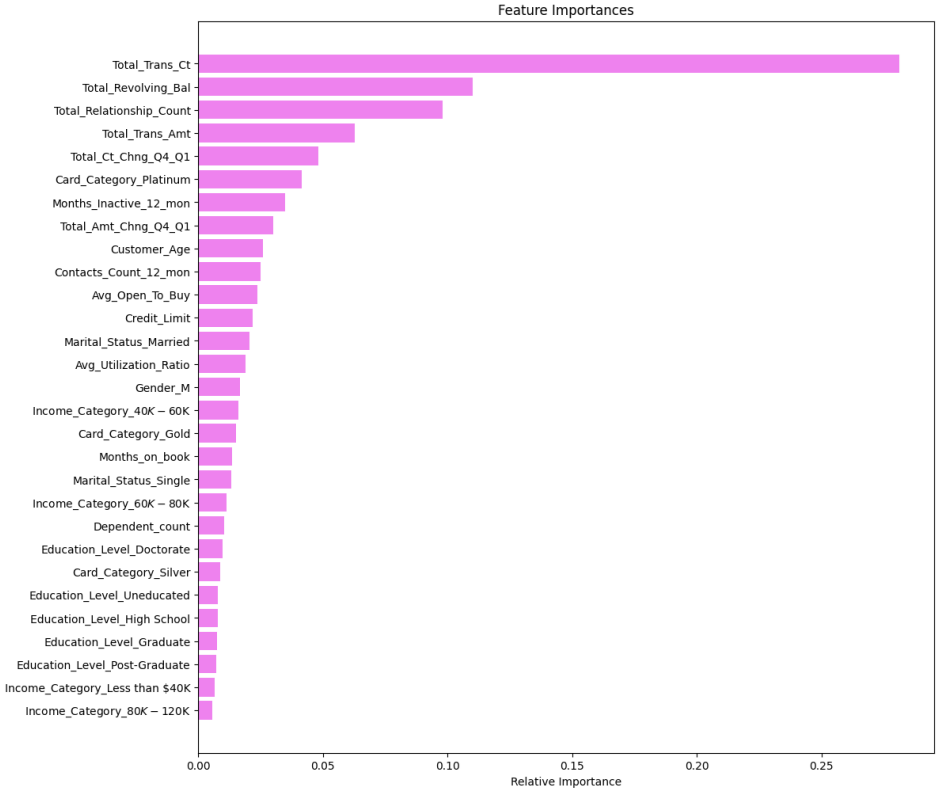
```
xgb_test = model_performance_classification_sklearn(tuned_xgb, X_test, y_test)
xgb_test
```

	Accuracy	Recall	Precision	F1
0	0.992	0.999	0.956	0.977

Link to Appendix slide on model assumptions

Feature Importances

The Tuned XGB model weighted the following features in order of importance as follows:



TOP 10

1. **Total_Trans_Ct:** Total Transaction Count (Last 12 months)
2. **Total_Revolving_Bal:** Total Revolving Balance on the Credit Card
3. **Total_Relationship_Count:** Total no. of products held by the customer
4. **Total_Trans_Amt:** Total Transaction Amount (Last 12 months)
5. **Total_Ct_Chng_Q4_Q1:** Change in Transaction Count (Q4 over Q1)
6. **Card_Category:** Type of Card (Platinum)
7. **Months_Inactive_12_mon:** No. of months inactive in the last 12 months
8. **Total_Amt_Chng_Q4_Q1:** Change in Transaction Amount (Q4 over Q1)
9. **Customer_Age:** in years
10. **Contacts_Count_12_mon:** No. of Contacts in the last 12 months

Final Model Determination

Based on the performance data the XGBoost model trained on original data is the best suited for prediction of attrition rate of Thera Bank's customer base.

1. Training Performance:
    - Accuracy: 0.992 (highest accuracy among all models)
    - Recall: 1.00 (perfect ability to identify true positives)
    - Precision: 0.955 (strong balance of correct positive predictions)
    - F1 Score: 0.976 (excellent balance between precision and recall)
2. Validation Performance:
    - Accuracy: 0.970 (second highest after AdaBoost)
    - Recall: 0.921 (still robust)
    - Precision: 0.895 (good precision)
    - F1 Score: 0.908 (solid overall performance)
3. Testing Performance:
    - The XGBoost model maintained high performance with an accuracy of 0.992, recall of 0.999, precision of 0.956, and F1 score of 0.977, indicating it generalizes extremely well on unseen data.

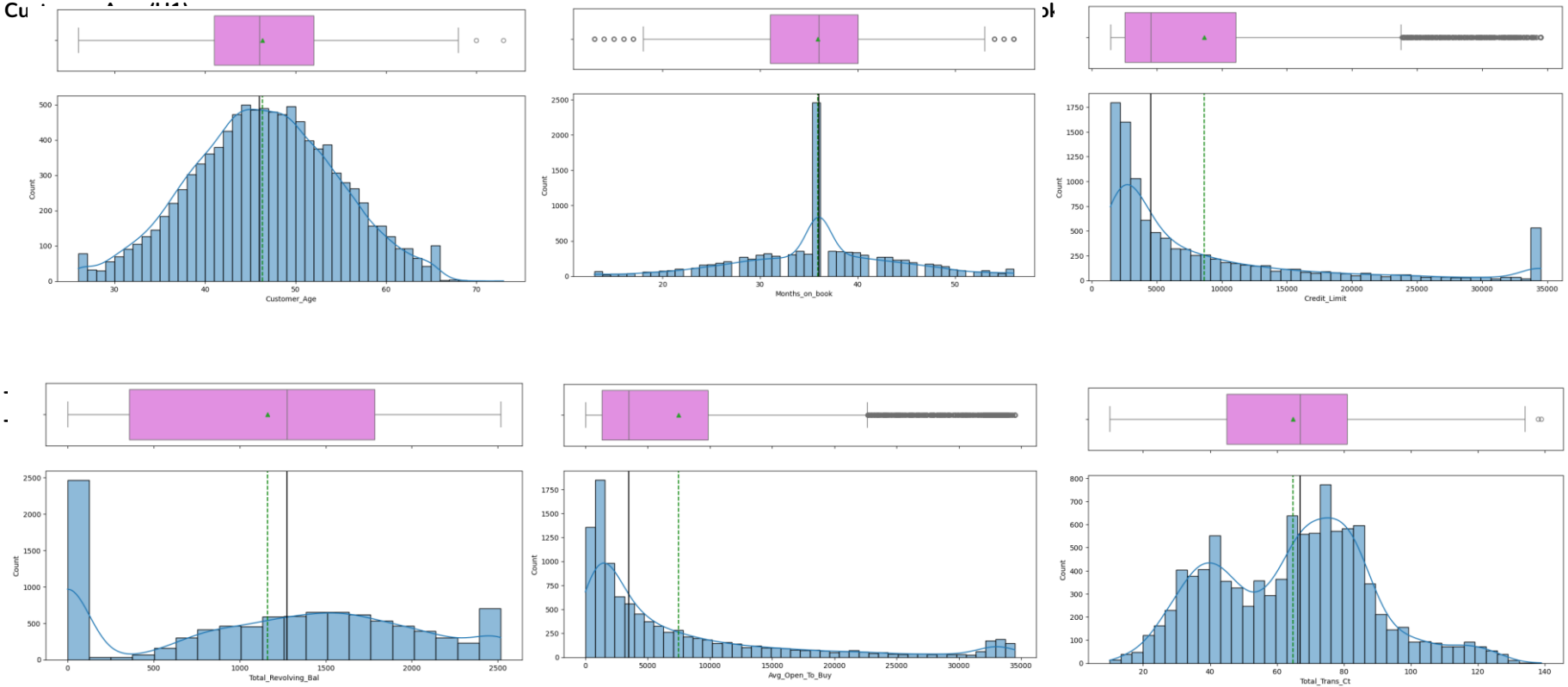
Comparison with other top models:

- AdaBoost trained with Undersampled Data shows high recall (0.990) but lower precision (0.817), indicating it may identify many positives but with more false positives.
- Gradient Boosting trained with Undersampled Data also performed well, but its metrics are consistently lower than those of XGBoost.

Overall, the XGBoost model's superior performance across all metrics in both training and testing phases makes it the most reliable choice for predicting Thera Bank's customer attrition effectively.

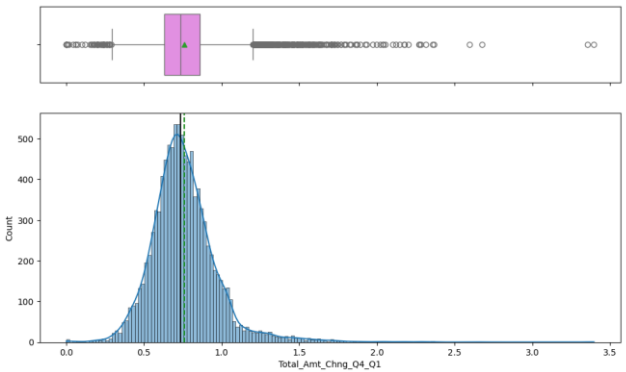
APPENDIX

Exploratory Data Analysis- Univariate Data

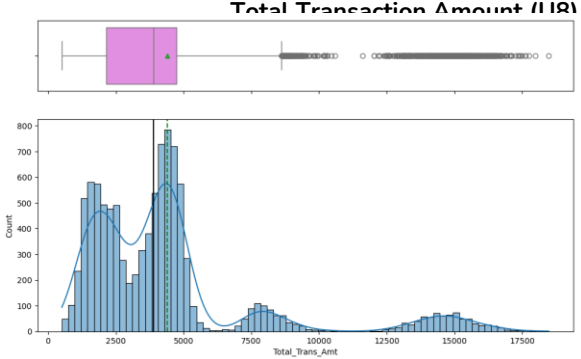


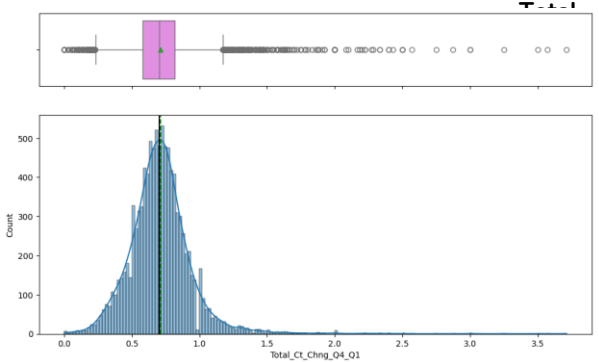
Exploratory Data Analysis- Univariate Data

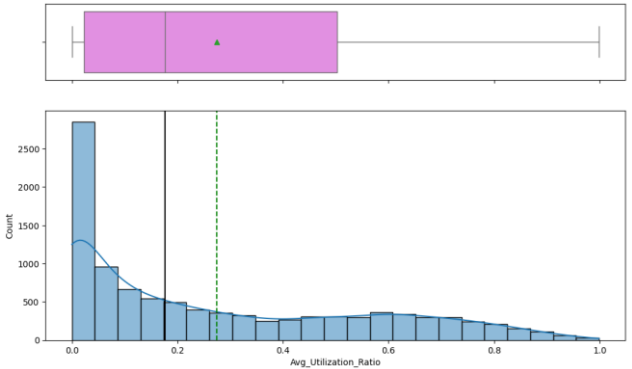
Total Amount Change Q4 to Q1 (U7)

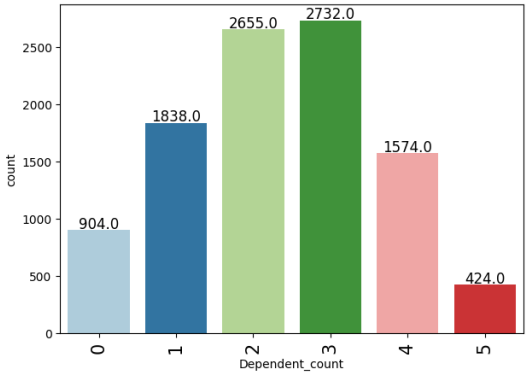


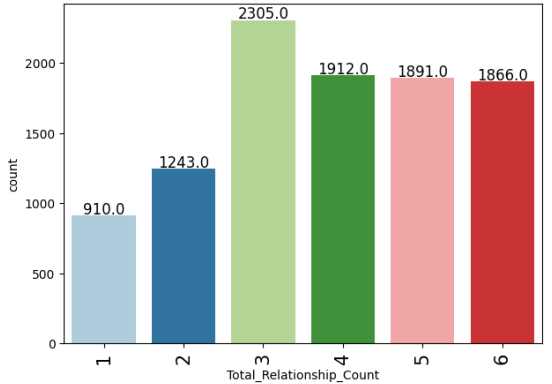
Total Transaction Amount (U8)



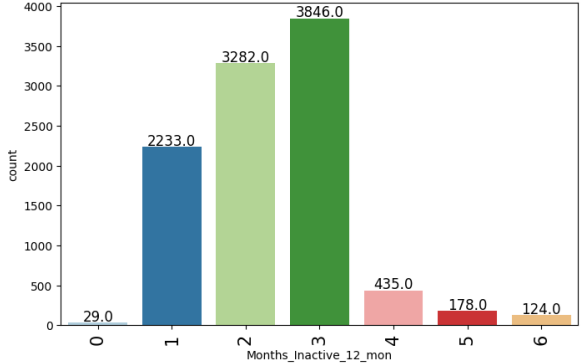




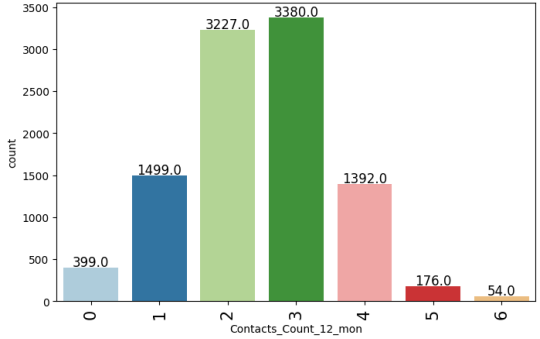




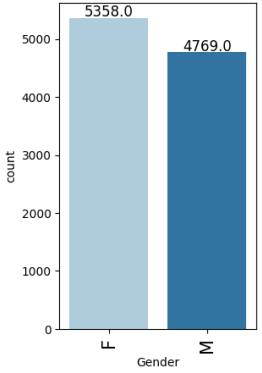
Exploratory Data Analysis- Univariate Data

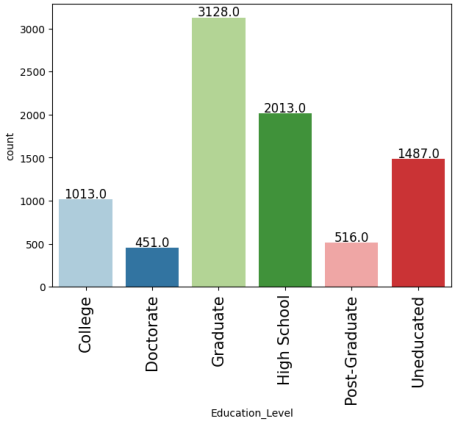


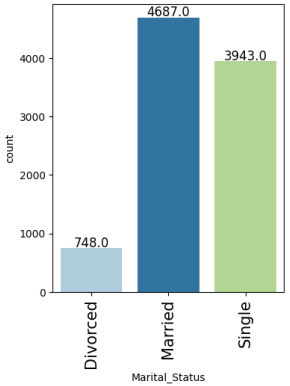
Contacts Count in last 12 months (U14)



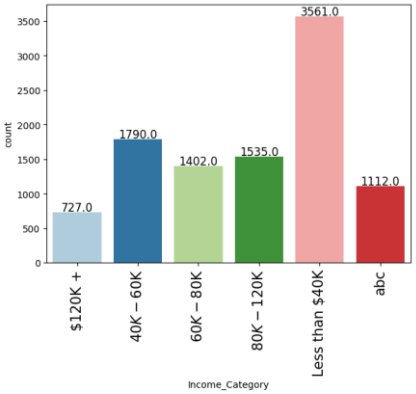
Gender



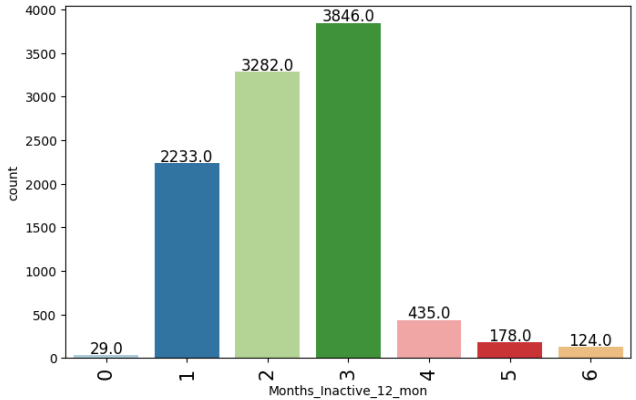


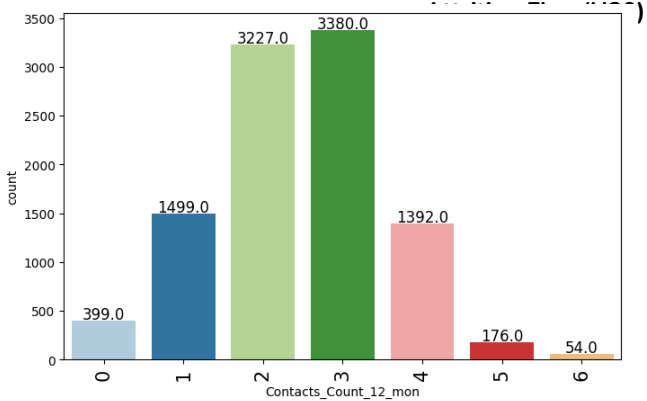


Marital Status

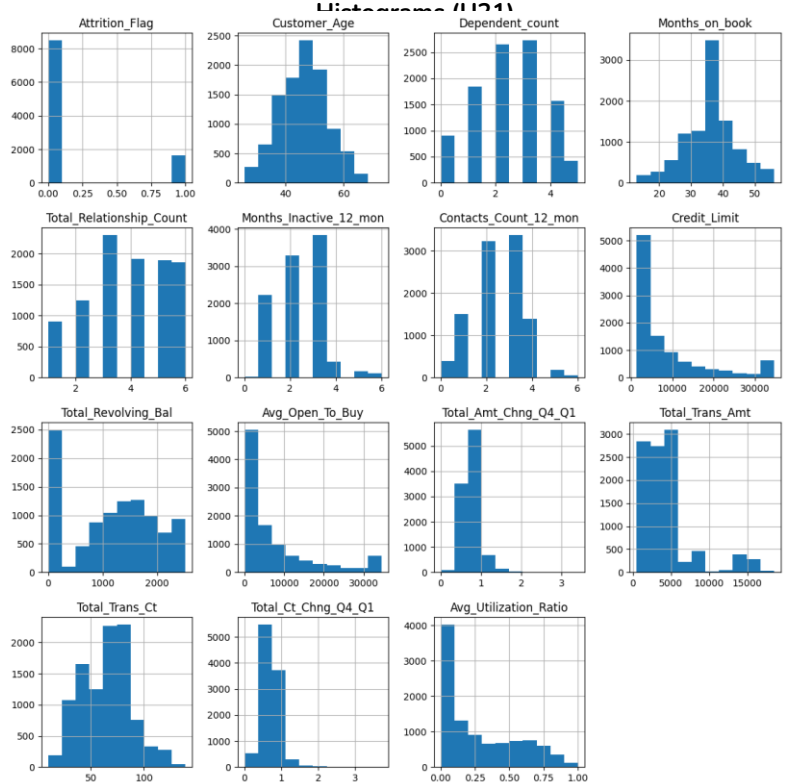


Exploratory Data Analysis- Univariate Data

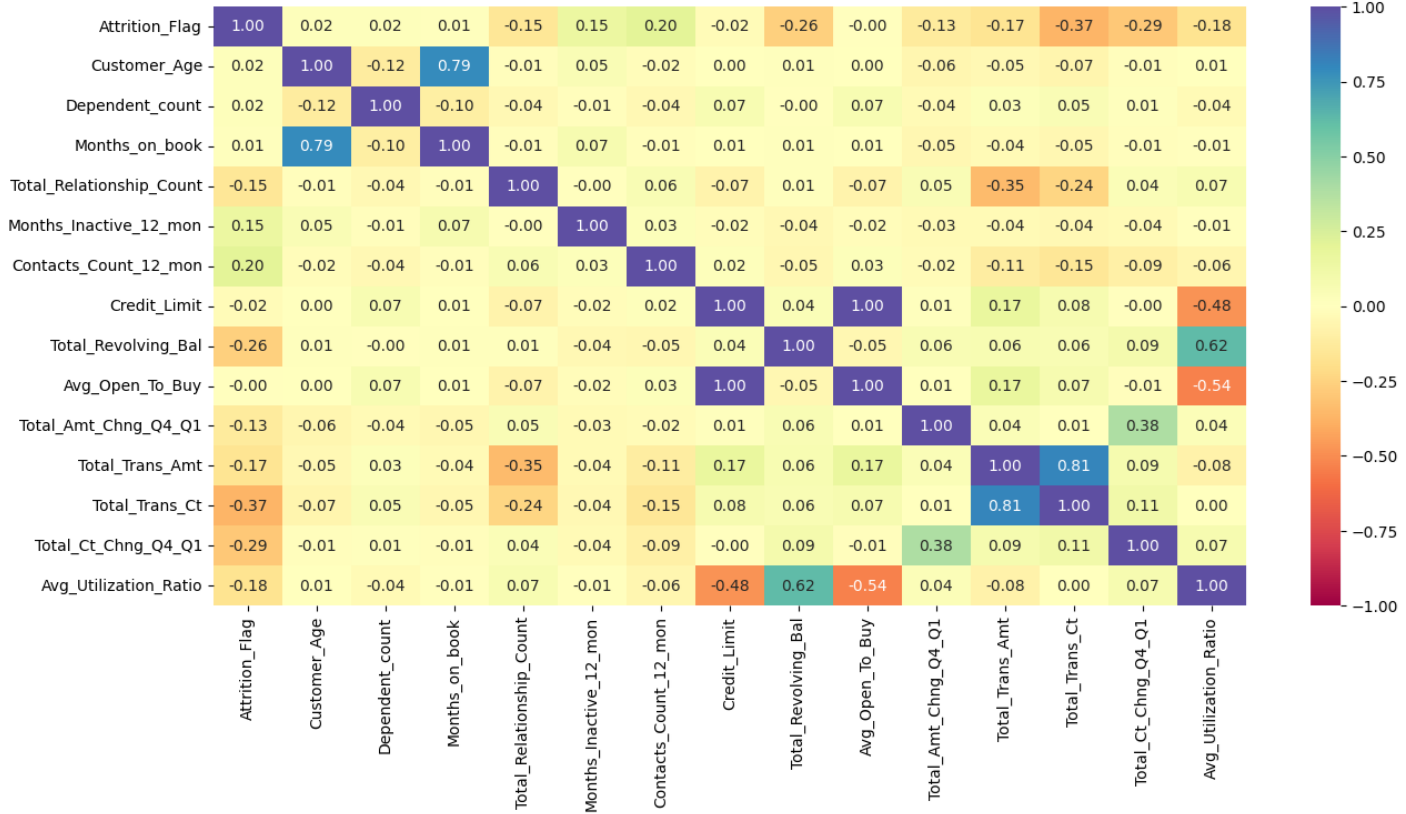




Exploratory Data Analysis- Univariate Data

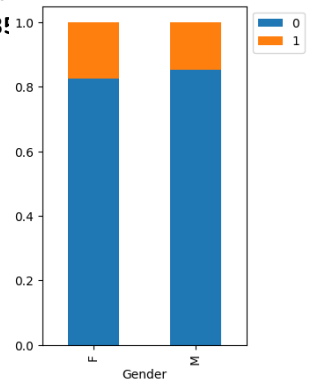


Exploratory Data Analysis- Bivariate Data



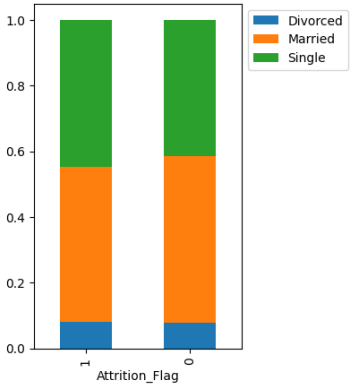
Exploratory Data Analysis- Bivariate Data

Attrition Flag vs Gender (B2)

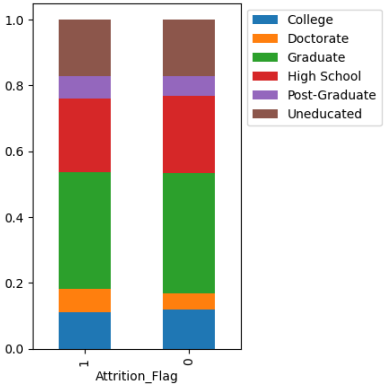


(B5

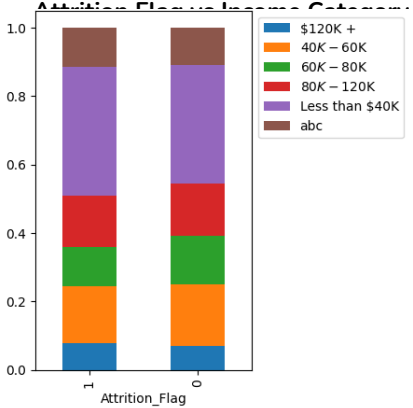
Attrition Flag vs Marital Status (B3)



Attrition Flag vs Education Level (B4)

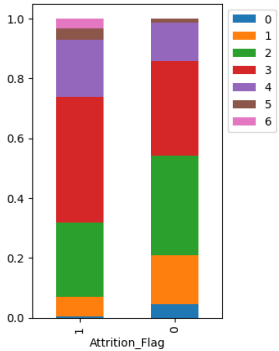


Attrition Flag vs Income Category



A

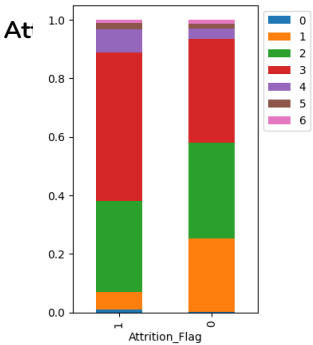
ount (B6)



(B

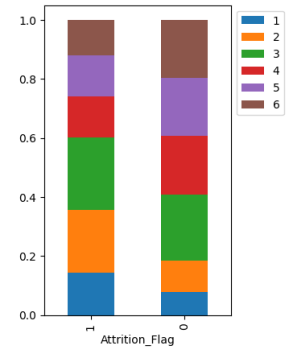
At

ctive (B7)

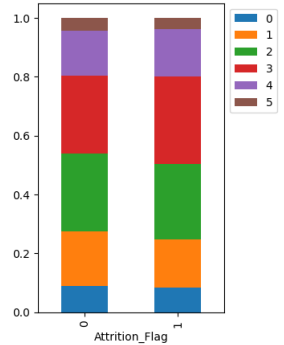


Att

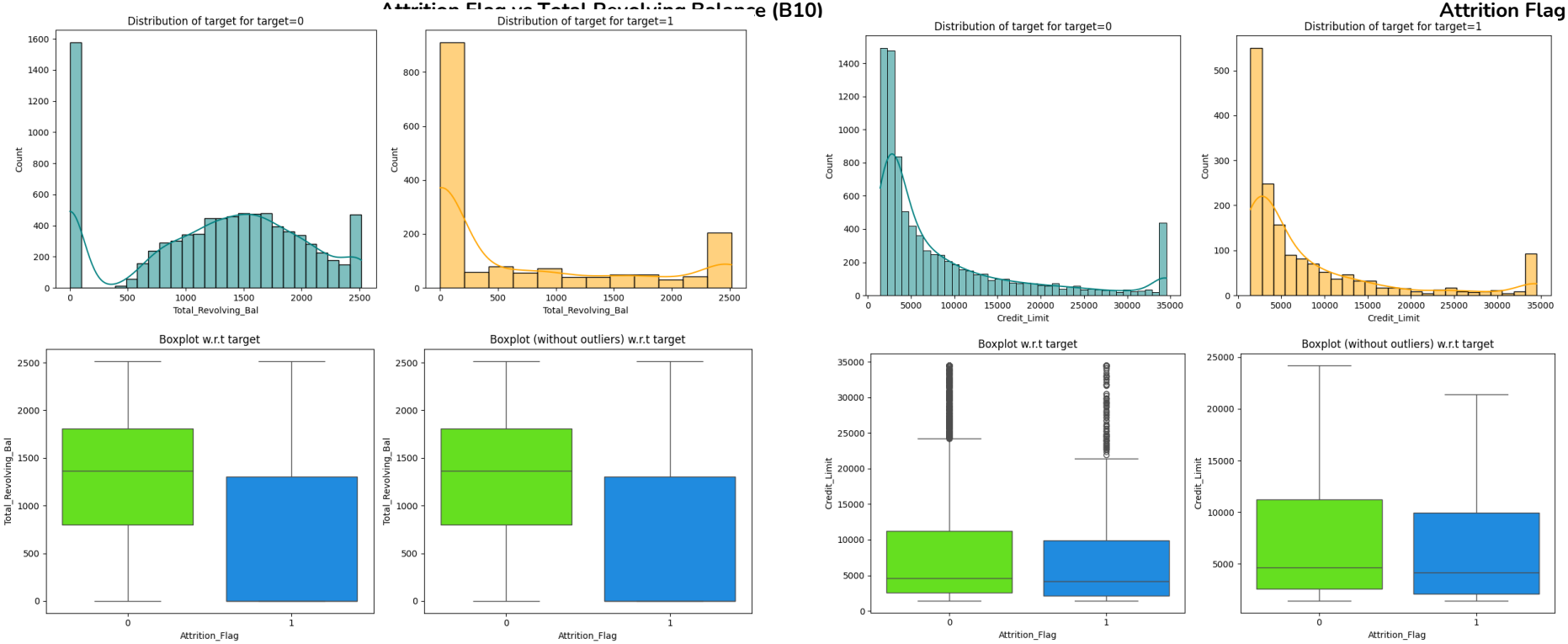
nship Count (B



dent Count

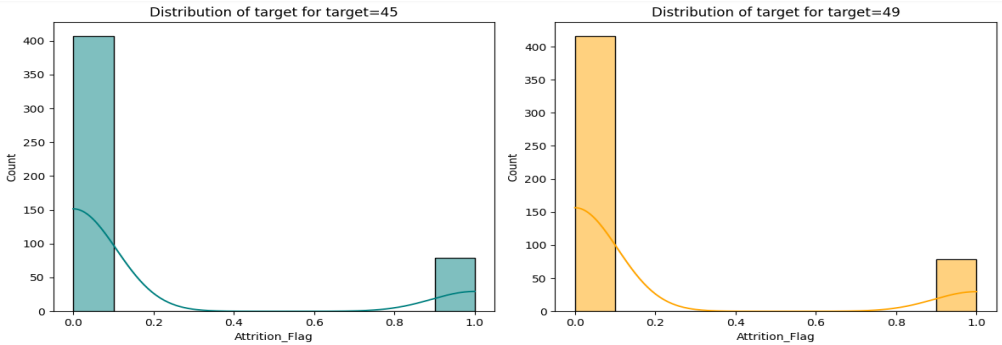


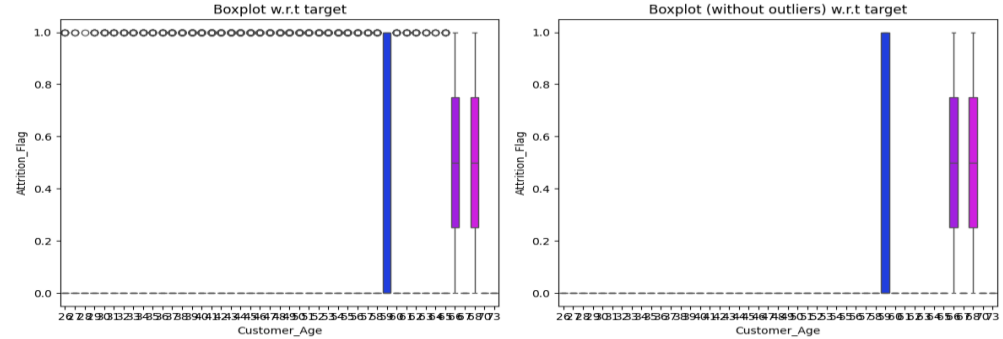
Exploratory Data Analysis- Bivariate Data



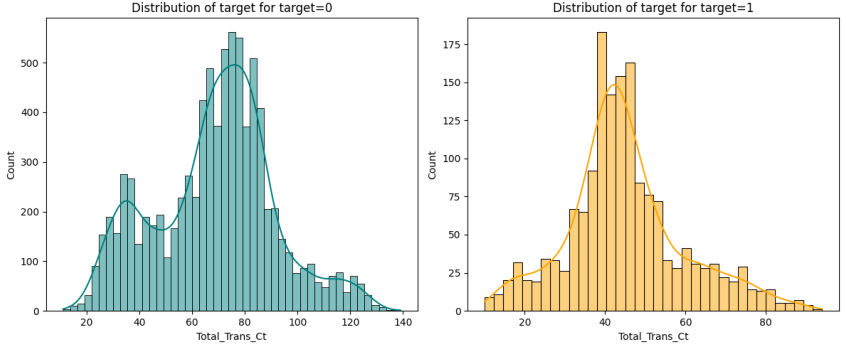
Exploratory Data Analysis- Bivariate Data

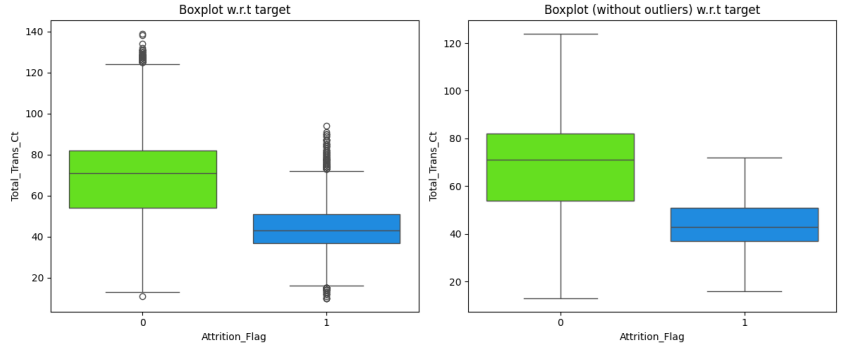
Attrition Flag vs Customer Age (B12)



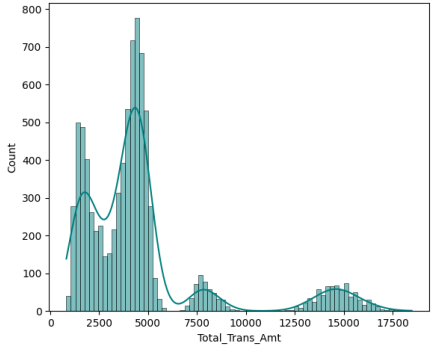


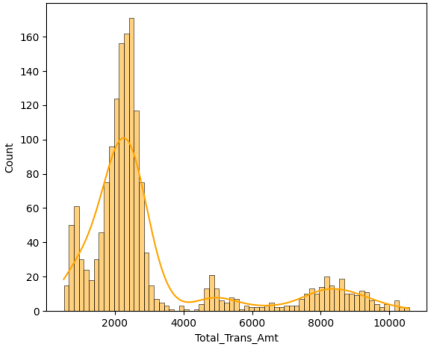
Attrition Flag vs Total

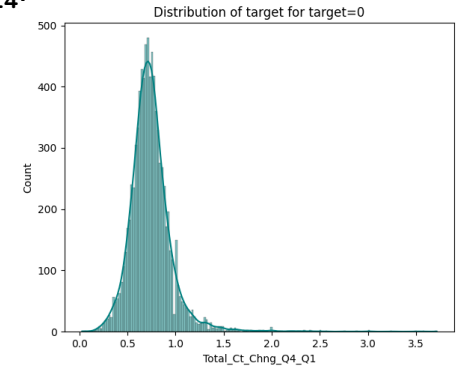


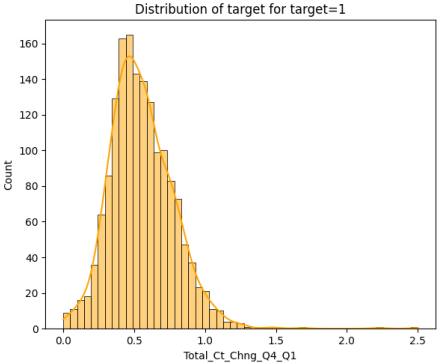


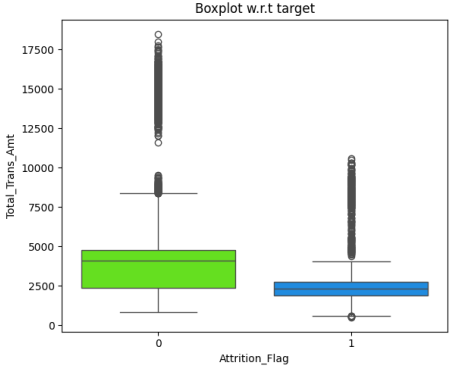
Exploratory Data Analysis- Bivariate Data

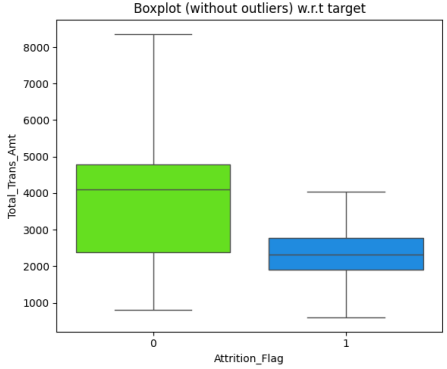


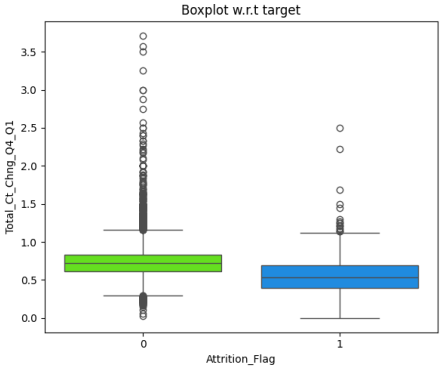


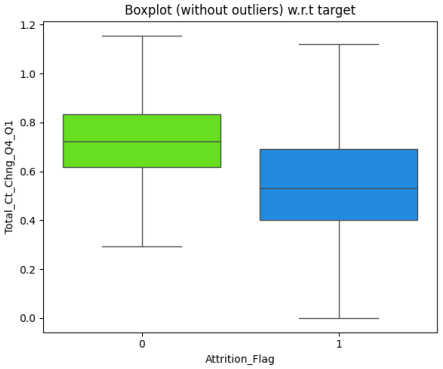






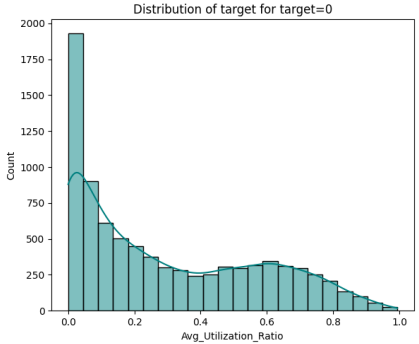


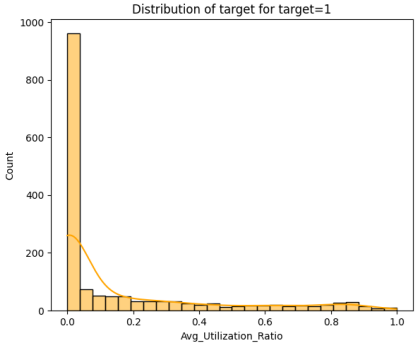


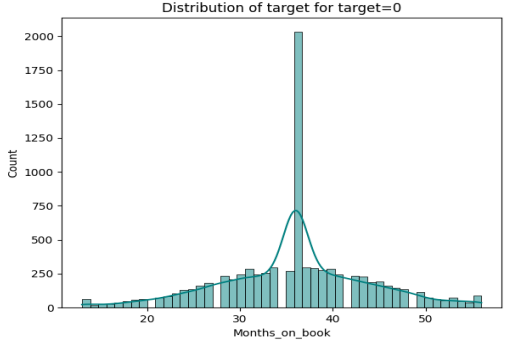


Exploratory Data Analysis- Bivariate Data

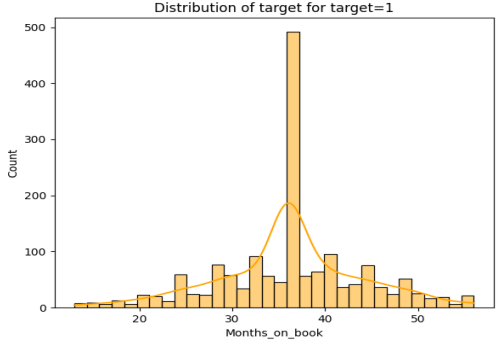
Attrition Flag vs Average Utilization Ratio (B16)

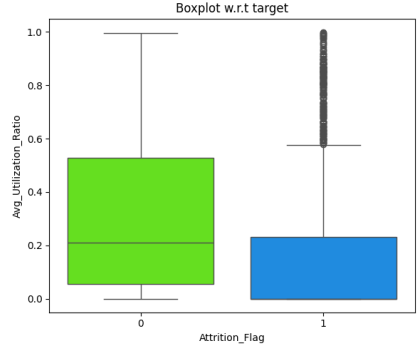


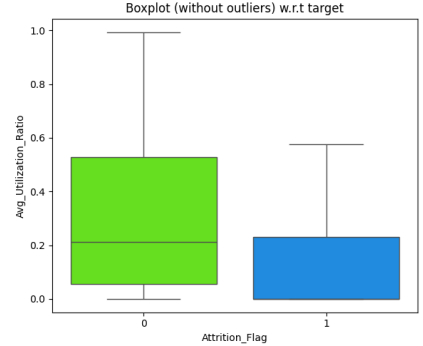


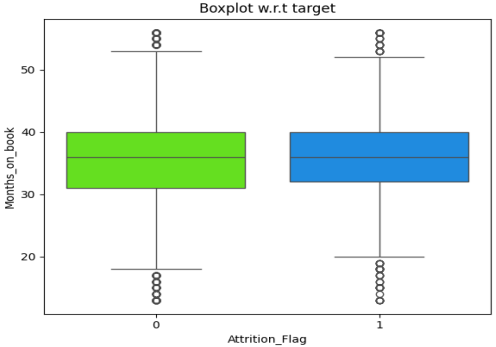


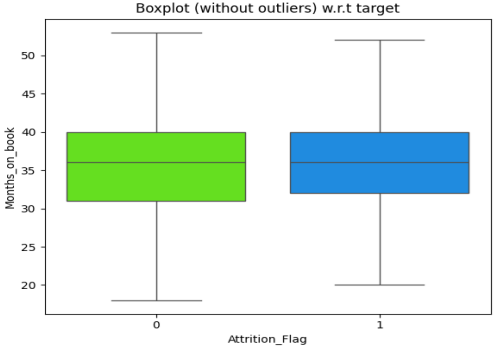
Attrition Flag vs



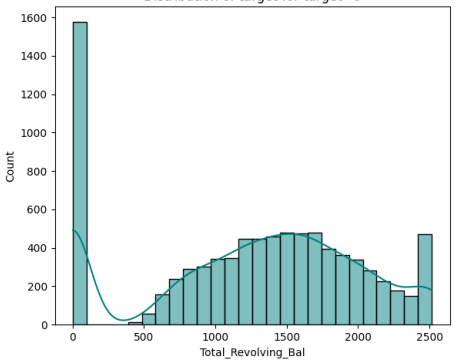


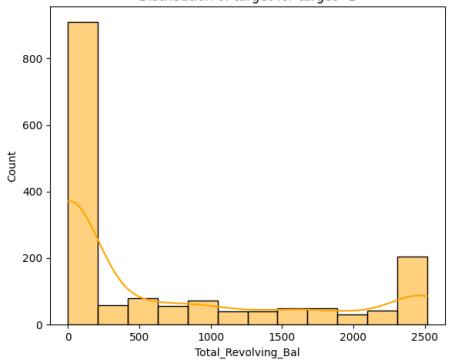


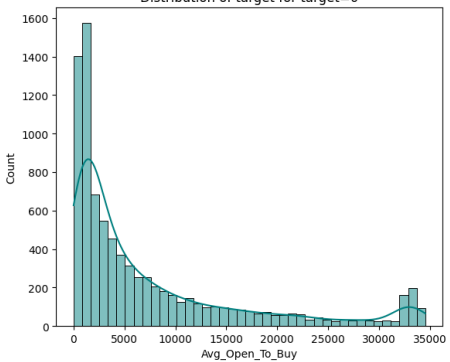


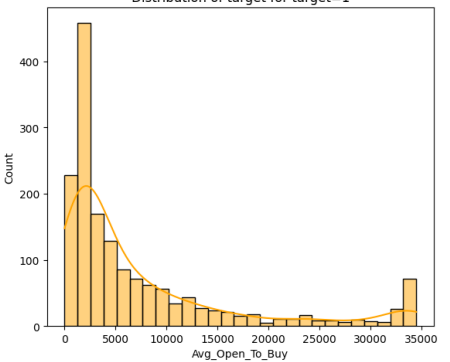


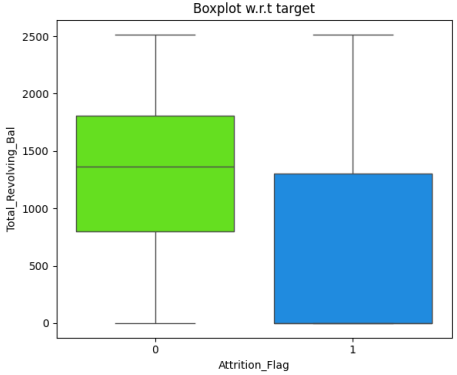
Exploratory Data Analysis- Bivariate Data

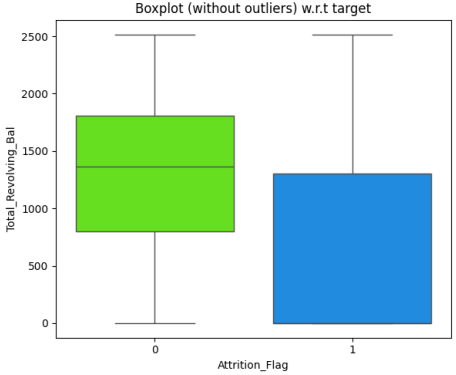


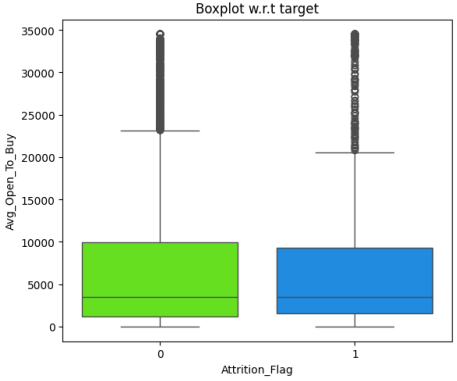


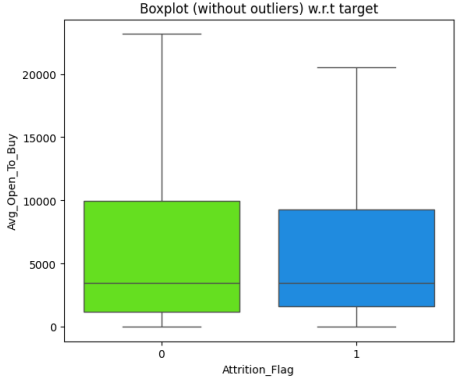












Data Preprocessing

Outlier Detection (DP1)

	0
Attrition_Flag	16.066
Customer_Age	0.020
Dependent_count	0.000
Months_on_book	3.812
Total_Relationship_Count	0.000
Months_Inactive_12_mon	3.268
Contacts_Count_12_mon	6.211
Credit_Limit	9.717
Total_Revolving_Bal	0.000
Avg_Open_To_Buy	9.509
Total_Amt_Chng_Q4_Q1	3.910
Total_Trans_Amt	8.848
Total_Trans_Ct	0.020
Total_Ct_Chng_Q4_Q1	3.891
Avg_Utilization_Ratio	0.000

dtype: float64

data1.isna().sum() (DP2)

	0
Attrition_Flag	0
Customer_Age	0
Gender	0
Dependent_count	0
Education_Level	1519
Marital_Status	749
Income_Category	1112
Card_Category	0
Months_on_book	0
Total_Relationship_Count	0
Months_Inactive_12_mon	0
Contacts_Count_12_mon	0
Credit_Limit	0
Total_Revolving_Bal	0
Avg_Open_To_Buy	0
Total_Amt_Chng_Q4_Q1	0
Total_Trans_Amt	0
Total_Trans_Ct	0
Total_Ct_Chng_Q4_Q1	0
Avg_Utilization_Ratio	0

dtype: int64

Data Preprocessing- Missing Value Imputation

X_train (DP3)

```
Customer_Age                0
Gender                      0
Dependent_count             0
Education_Level             0
Marital_Status              0
Income_Category             0
Card_Category               0
Months_on_book              0
Total_Relationship_Count    0
Months_Inactive_12_mon      0
Contacts_Count_12_mon       0
Credit_Limit                0
Total_Revolving_Bal         0
Avg_Open_To_Buy             0
Total_Amt_Chng_Q4_Q1        0
Total_Trans_Amt             0
Total_Trans_Ct              0
Total_Ct_Chng_Q4_Q1         0
Avg_Utilization_Ratio       0
dtype: int64
```

```
Gender
F    4272
M    3829
Name: count, dtype: int64
******************************
Education_Level
Graduate         3689
High School      1650
Uneducated       1183
College           796
Post-Graduate     427
Doctorate         356
Name: count, dtype: int64
******************************
Marital_Status
Married     4360
Single      3142
Divorced     599
Name: count, dtype: int64
******************************
Income_Category
Less than $40K    3718
$40K - $60K       1450
$80K - $120K      1209
$60K - $80K       1136
$120K +            588
Name: count, dtype: int64
******************************
Card_Category
Blue        7559
Silver       431
Gold          94
Platinum      17
Name: count, dtype: int64
******************************
```

X_val (DP4)

```
Customer_Age                0
Gender                      0
Dependent_count             0
Education_Level             0
Marital_Status              0
Income_Category             0
Card_Category               0
Months_on_book              0
Total_Relationship_Count    0
Months_Inactive_12_mon      0
Contacts_Count_12_mon       0
Credit_Limit                0
Total_Revolving_Bal         0
Avg_Open_To_Buy             0
Total_Amt_Chng_Q4_Q1        0
Total_Trans_Amt             0
Total_Trans_Ct              0
Total_Ct_Chng_Q4_Q1         0
Avg_Utilization_Ratio       0
dtype: int64
```

```
Gender
F    1334
M    1198
Name: count, dtype: int64
******************************
Education_Level
Graduate         1171
High School       470
Uneducated        385
College           268
Doctorate         120
Post-Graduate     118
Name: count, dtype: int64
******************************
Marital_Status
Married     1350
Single       991
Divorced     191
Name: count, dtype: int64
******************************
Income_Category
Less than $40K    1159
$40K - $60K        438
$80K - $120K       398
$60K - $80K        362
$120K +            175
Name: count, dtype: int64
******************************
Card_Category
Blue        2361
Silver       141
Gold          27
Platinum       3
Name: count, dtype: int64
******************************
```

```
Gender
F    4024
M    3571
Name: count, dtype: int64
******************************
Education_Level
Graduate         3476
High School      1543
Uneducated       1102
College           745
Post-Graduate     398
Doctorate         331
Name: count, dtype: int64
******************************
Marital_Status
Married     4086
Single      2952
Divorced     557
Name: count, dtype: int64
******************************
Income_Category
Less than $40K    3514
$40K - $60K       1352
$80K - $120K      1137
$60K - $80K       1040
$120K +            552
Name: count, dtype: int64
******************************
Card_Category
Blue        7075
Silver       414
Gold          89
Platinum      17
Name: count, dtype: int64
******************************
```

Model Building- Evaluation Criteria

Thera Bank would want Recall to be maximized, greater the Recall higher the chances of minimizing false negatives. Therefor the focus should be on increasing Recall or minimizing the false negatives or in other words identifying the true positives so that the bank can retain their valuable customers by identifying the customers who are at risk of attrition.

1. Use sklearn to define a function to compute different metrics to check performance of classification models

1. Use sklearn to define a function to show confusion matrix

Link to Appendix slide on model assumptions

Model Building (Original data)

```
models = []  # Empty list to store all the models

# Appending models into the list
models.append(("Bagging", BaggingClassifier(random_state=1)))
models.append(("Random forest", RandomForestClassifier(random_state=1)))
models.append(("AdaBoost", AdaBoostClassifier(random_state=1))) ## Complete the code to append remaining 3 models in the list models
models.append(("Gradient Boost", GradientBoostingClassifier(random_state=1)))
models.append(("XGBoost", XGBClassifier(random_state=1)))

print("\n" "Training Performance:" "\n")
for name, model in models:
    model.fit(X_train, y_train)
    scores = recall_score(y_train, model.predict(X_train))
    print("{}: {}".format(name, scores))

print("\n" "Validation Performance:" "\n")

for name, model in models:
    model.fit(X_train, y_train)
    scores_val = recall_score(y_val, model.predict(X_val))
    print("{}: {}".format(name, scores_val))
```

Link to Appendix slide on model assumptions

Model Performance Summary (Original data)

Summary of performance metrics for training and validation data

```
Training Performance:

Bagging: 0.9731182795698925
Random forest: 1.0
AdaBoost: 0.8602150537634409
Gradient Boost: 0.8940092165898618
XGBoost: 1.0

Validation Performance:

Bagging: 0.8230958230958231
Random forest: 0.8181818181818182
AdaBoost: 0.8058968058968059
Gradient Boost: 0.8181818181818182
XGBoost: 0.8918918918918919
```

Comments on the model performance

1. Bagging: Overfit
2. Random forest: Overfit (Perfect training score)
3. AdaBoost: Generalized
4. Gradient Boost: Generalized
5. XGBoost: Generalized (Perfect training score)

Link to Appendix slide on model assumptions

Model Building (Oversampled data)

The undersampling method chosen is ***Synthetic Minority Oversampling Technique***.

- sm = SMOTE (sampling_strategy=1, k_neighbors=5, random_state=1)

```
print("Before Oversampling, counts of label 'Yes': {}".format(sum(y_train == 1)))
print("Before Oversampling, counts of label 'No': {} \n".format(sum(y_train == 0)))

sm = SMOTE(
    sampling_strategy=1, k_neighbors=5, random_state=1
)  # Synthetic Minority Over Sampling Technique
X_train_over, y_train_over = sm.fit_resample(X_train, y_train)


print("After Oversampling, counts of label 'Yes': {}".format(sum(y_train_over == 1)))
print("After Oversampling, counts of label 'No': {} \n".format(sum(y_train_over == 0)))


print("After Oversampling, the shape of train_X: {}".format(X_train_over.shape))
print("After Oversampling, the shape of train_y: {} \n".format(y_train_over.shape))
```

```
Before Oversampling, counts of label 'Yes': 1302
Before Oversampling, counts of label 'No': 6799

After Oversampling, counts of label 'Yes': 6799
After Oversampling, counts of label 'No': 6799

After Oversampling, the shape of train_X: (13598, 29)
After Oversampling, the shape of train_y: (13598,)
```

Link to Appendix slide on model assumptions

Model Performance Summary (Oversampled data)

Summary of performance metrics for training and validation data

```
Training Performance:

Bagging: 0.998529195469922
Random forest: 1.0
AdaBoost: 0.9667598176202382
Gradient Boost: 0.9764671275187528
XGBoost: 1.0

Validation Performance:

Bagging: 0.8845208845208845
Random forest: 0.8771498771498771
AdaBoost: 0.8402948402948403
Gradient Boost: 0.8771498771498771
XGBoost: 0.9017199017199017
```

Comments on the model performance

1. Bagging: Overfit
2. Random forest: Overfit (Perfect training score)
3. AdaBoost: Generalized
4. Gradient Boost: Generalized
5. XGBoost: Slightly Overfit (Perfect training score)

Link to Appendix slide on model assumptions

Model Performance Summary (Undersampled data)

The undersampling method chosen is ***Random Undersampling***.

- rus = RandomUnderSampler (random_state=1)

```
rus = RandomUnderSampler(random_state=1)
X_train_un, y_train_un = rus.fit_resample(X_train, y_train)
```

```
print("Before Under Sampling, counts of label 'Yes': {}".format(sum(y_train == 1)))
print("Before Under Sampling, counts of label 'No': {} \n".format(sum(y_train == 0)))

print("After Under Sampling, counts of label 'Yes': {}".format(sum(y_train_un == 1)))
print("After Under Sampling, counts of label 'No': {} \n".format(sum(y_train_un == 0)))

print("After Under Sampling, the shape of train_X: {}".format(X_train_un.shape))
print("After Under Sampling, the shape of train_y: {} \n".format(y_train_un.shape))
```

```
Before Under Sampling, counts of label 'Yes': 1302
Before Under Sampling, counts of label 'No': 6799

After Under Sampling, counts of label 'Yes': 1302
After Under Sampling, counts of label 'No': 1302

After Under Sampling, the shape of train_X: (2604, 29)
After Under Sampling, the shape of train_y: (2604,)
```


```
Before Under Sampling, counts of label 'Yes': 1302
Before Under Sampling, counts of label 'No': 6799

After Under Sampling, counts of label 'Yes': 1302
After Under Sampling, counts of label 'No': 1302

After Under Sampling, the shape of train_X: (2604, 29)
After Under Sampling, the shape of train_y: (2604,)
```


Link to Appendix slide on model assumptions

Model Performance Summary (Undersampled data)

Summary of performance metrics for training and validation data

```
Training Performance:

Bagging: 0.9900153609831029
Random forest: 1.0
AdaBoost: 0.9631336405529954
Gradient Boost: 0.978494623655914
XGBoost: 1.0

Validation Performance:

Bagging: 0.9385749385749386
Random forest: 0.9385749385749386
AdaBoost: 0.9361179361179361
Gradient Boost: 0.9434889434889435
XGBoost: 0.9508599508599509
```

Comments on the model performance

1. Bagging: Generalized
2. Random forest: Overfit (Perfect training score)
3. AdaBoost: Generalized
4. Gradient Boost: Generalized
5. XGBoost: Overfit (Perfect training score)

Link to Appendix slide on model assumptions