Churn Prediction for Credit Card Customers: Insights and Actions

Advanced Machine Learning

2025-01-30

Melissa Lindskär

Credit: Unsplash

Contents / Agenda

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

Executive Summary

Goal: To predict customer churn and provide actionable insights to improve retention.

Key Result: The Gradient Boosting model with undersampled data achieved the highest recall (96.3%), aligning with the business objective of identifying at-risk customers.

Recommendations:

- Launch targeted campaigns for low-transaction customers.
- Offer financial counseling to high-credit utilization customers.
- Enhance loyalty programs to improve customer retention.

Business Problem Overview

Definition: Customer attrition, or churn in credit card usage, is a critical challenge for businesses, particularly in highly competitive industries. When customers stop or significantly reduce their credit card activity, it negatively impacts revenue streams such as transaction fees, interest income, and other associated benefits. Additionally, acquiring new customers to replace lost ones increases costs.

The goal of this project is to:

1. **Identify customers at risk of reducing or stopping their credit card usage** before they churn.
2. **Understand the key drivers of churn** to take proactive retention actions.

This project specifically analyzes customer transaction and demographic data to:

- Predict which customers are likely to churn in their credit card usage.
- Provide actionable insights to reduce churn and improve customer retention.

Credit: Unsplash

Solution Approach/Methodology

To address the churn problem, we´ve employed a structured and data-driven approach comprising the following steps:

Exploratory Data Analysis (EDA):

- Conducted univariate and bivariate analysis to understand data distributions and relationships.
- Identified key features contributing to churn, such as transaction counts, spending patterns, and credit utilization.

Data Preparation:

- **Target Variable Encoding:** Encoded Attrition_Flag ("Existing Customer" → 0, "Attrited Customer" → 1) to ensure it was numeric and compatible with machine learning algorithms.
- One-hot encoding was applied to categorical variables to make them suitable for machine learning algorithms.
- Avg_Open_To_Buy was removed due to perfect correlation with Credit_Limit, ensuring no redundancy in the features.
- Missing values were imputed (mean for numerical variables and mode for categorical variables).
- Prevented data leakage by splitting the dataset before any preprocessing.

Modeling:

- Evaluated multiple machine learning algorithms, including:
  - **Gradient Boosting**
  - **XGBoost**
  - **AdaBoost**
  - **Baseline Models**
- Emphasized **Recall** as the primary metric to ensure at-risk customers are captured effectively.
- Tuned hyperparameters to enhance model performance further.

Model Comparison:

- Compared models based on Accuracy, Recall, Precision, and F1-score to select the best-performing model.
- Chose Baseline Model Gradient Boosting with undersampled data for its high Recall (0.951 on validation) to minimize missed churn predictions.

Data Overview - Dataset Summary

- **Total Records**: 10,127
- **Total Features**: 21
- **Target Variable**: Attrition_Flag
  - Categories:
    - **Existing Customer**: 8,500
    - **Attrited Customer**: 1,627
- **Customer Activity**:
  - Total_Trans_Amt: Total transaction amount in the last 12 months.
  - Total_Trans_Ct: Total transaction count in the last 12 months.
  - Months_Inactive_12_mon: Number of inactive months in the last year.
- **Demographic Information**:
  - Education_Level, Marital_Status, Income_Category: Key customer characteristics
- **Client Information**:
  - CLIENTNUM: Unique identifier for customers.
  - Customer_Age: Customer's age.
  - Gender: Gender of the customer.
- **Financial Indicators**:
  - Credit_Limit: Credit limit on the card.
  - Avg_Open_To_Buy: Average available credit over the last 12 months.
  - Total_Revolving_Bal: Outstanding balance carried forward.
  - Avg_Utilization_Ratio: Proportion of credit utilized.

Data Quality Insights:

- Missing values in:
  - Education_Level (~15% missing)
  - Marital_Status (~7% missing)
  - No duplicates

```
# let's check the data types of the columns in the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 21 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   CLIENTNUM                 10127 non-null  int64
 1   Attrition_Flag            10127 non-null  object
 2   Customer_Age              10127 non-null  int64
 3   Gender                    10127 non-null  object
 4   Dependent_count           10127 non-null  int64
 5   Education_Level           8608 non-null   object
 6   Marital_Status            9378 non-null   object
 7   Income_Category           10127 non-null  object
 8   Card_Category             10127 non-null  object
 9   Months_on_book            10127 non-null  int64
 10  Total_Relationship_Count  10127 non-null  int64
 11  Months_Inactive_12_mon    10127 non-null  int64
 12  Contacts_Count_12_mon     10127 non-null  int64
 13  Credit_Limit              10127 non-null  float64
 14  Total_Revolving_Bal       10127 non-null  int64
 15  Avg_Open_To_Buy           10127 non-null  float64
 16  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 17  Total_Trans_Amt           10127 non-null  int64
 18  Total_Trans_Ct            10127 non-null  int64
 19  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 20  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(10), object(6)
memory usage: 1.6+ MB
```

Data overview - Summary of Numerical Features:

- **Customer_Age**:
  - Average: 46 years.
  - Minimum: 26 years.
  - Maximum: 73 years.
  - Majority fall between 41–52 years (25th to 75th percentile).
- **Dependent_count** (Number of dependents):
  - Average: 2.3 dependents.
  - Range: 0 to 5 dependents.
  - Most customers have 1–3 dependents (25th to 75th percentile).
- **Months_on_book** (Relationship length with the bank):
  - Average: 36 months (~3 years).
  - Range: 13 to 56 months.
  - Most customers have 31–40 months of relationship with the bank.
- **Total_Relationship_Count** (Products held by the customer):
  - Average: 3.8 products.
  - Range: 1 to 6 products.
  - Most customers hold 3–5 products.
- **Months_Inactive_12_mon** (Months inactive in the last 12 months):
  - Average: 2.3 months.
  - Range: 0 to 6 months.
  - Most customers were inactive for 2–3 months.

	count	mean	std	min	25%	50%	75%	max
CLIENTNUM	10127.000	739177606.334	36903783.450	708082083.000	713036770.500	717926358.000	773143533.000	828343083.000
Customer_Age	10127.000	46.326	8.017	26.000	41.000	46.000	52.000	73.000
Dependent_count	10127.000	2.346	1.299	0.000	1.000	2.000	3.000	5.000
Months_on_book	10127.000	35.928	7.986	13.000	31.000	36.000	40.000	56.000
Total_Relationship_Count	10127.000	3.813	1.554	1.000	3.000	4.000	5.000	6.000
Months_Inactive_12_mon	10127.000	2.341	1.011	0.000	2.000	2.000	3.000	6.000
Contacts_Count_12_mon	10127.000	2.455	1.106	0.000	2.000	2.000	3.000	6.000
Credit_Limit	10127.000	8631.954	9088.777	1438.300	2555.000	4549.000	11067.500	34516.000
Total_Revolving_Bal	10127.000	1162.814	814.987	0.000	359.000	1276.000	1784.000	2517.000
Avg_Open_To_Buy	10127.000	7469.140	9090.685	3.000	1324.500	3474.000	9859.000	34516.000
Total_Amt_Chng_Q4_Q1	10127.000	0.760	0.219	0.000	0.631	0.736	0.859	3.397
Total_Trans_Amt	10127.000	4404.086	3397.129	510.000	2155.500	3899.000	4741.000	18484.000
Total_Trans_Ct	10127.000	64.859	23.473	10.000	45.000	67.000	81.000	139.000
Total_Ct_Chng_Q4_Q1	10127.000	0.712	0.238	0.000	0.582	0.702	0.818	3.714
Avg_Utilization_Ratio	10127.000	0.275	0.276	0.000	0.023	0.176	0.503	0.999

- **Contacts_Count_12_mon** (Customer-bank contacts in the last year):
  - Average: 2.5 contacts.
  - Range: 0 to 6 contacts.
  - Most customers had 2–3 contacts.
- **Credit_Limit**:
  - Average: $8,631.
  - Range: $1,438 to $34,516.
  - Most customers have a credit limit between $2,555–$11,067.

Data overview- Summary of Numerical Features:

- **Total_Revolving_Bal** (Outstanding balance):
  - Average: $1,163.
  - Range: $0 to $2,517.
  - Most customers have balances between $359–$1,784.
- **Avg_Open_To_Buy** (Credit left to use):
  - Average: $7,469.
  - Range: $3 to $34,516.
  - Most customers have $1,324–$9,859 available credit.
- **Total_Amt_Chng_Q4_Q1** (Change in transaction amount from Q4 to Q1):
  - Average: 0.76.
  - Most customers fall within the range of 0.63–0.86.
- **Total_Trans_Amt** (Total transaction amount in the last year):
  - Average: $4,404.
  - Range: $510 to $18,484.
  - Most customers transacted between $2,155–$4,741.
- **Total_Trans_Ct** (Total transaction count in the last year):
  - Average: 65 transactions.
  - Range: 10 to 139 transactions.
  - Most customers had 45–81 transactions.
- **Total_Ct_Chng_Q4_Q1** (Change in transaction count from Q4 to Q1):
  - Average: 0.71.
  - Most customers fall within the range of 0.58–0.82.
- **Avg_Utilization_Ratio** (Credit utilization ratio):
  - Average: 0.28 (~28% utilization).
  - Range: 0 to 0.999.
  - Most customers fall within 2–50% utilization

Insights:

- The data shows wide variability in customer behavior, particularly in transaction amounts, credit limits, and credit utilization.
- Most customers maintain an active relationship with the bank but show varying levels of engagement in terms of product usage and transactions.
- There are potential outliers in features like Total_Trans_Amt and Credit_Limit, which probably need further investigation.

Data Overview - Summary of Objects Features:

Of all the features, 6 of them are objects

Unique values in Attrition_Flag are:
Existing Customer 8500
Attrited Customer 1627
Name: count, dtype: int64
****

Unique values in Gender are:
F 5358
M 4769
Name: count, dtype: int64
****

Unique values in Education_Level are:
Graduate 3128
High School 2013
Uneducated 1487
College 1013
Post-Graduate 516
Doctorate 451
Name: count, dtype: int64
****

Unique values in Marital_Status are:
Married 4687
Single 3943
Divorced 748
Name: count, dtype: int64
****

Unique values in Income_Category are:
Less than $40K 3561
$40K - $60K 1790
$80K - $120K 1535
$60K - $80K 1402
abc 1112
$120K + 727
Name: count, dtype: int64
****

Unique values in Card_Category are:
Blue 9436
Silver 555
Gold 116
Platinum 20
Name: count, dtype: int64

```
data.describe(include=["object"]).T
```

	count	unique	top	freq
Attrition_Flag	10127	2	Existing Customer	8500
Gender	10127	2	F	5358
Education_Level	8608	6	Graduate	3128
Marital_Status	9378	3	Married	4687
Income_Category	10127	6	Less than $40K	3561
Card_Category	10127	4	Blue	9436

Data Preparation and Prevention of Data Leakage

To ensure a robust and reliable analysis, I implemented best practices during the preparation of our dataset, BEFORE the EDA:

Data Splitting for Integrity

The dataset was split into training, validation, and test sets before performing **bivariate** analysis as part of the EDA. However, univariate analysis was conducted on the entire dataset prior to splitting to gain initial insights into individual feature distributions and potential data quality issues. This step provided an overview of the data's structure while preserving the integrity of downstream analyses.

Splitting the data early is critical to prevent data leakage, which can lead to misleading results and overly optimistic models. By ensuring that the model is evaluated on unseen data, we accurately reflect its true performance in real-world scenarios.

Removal of Irrelevant Data

The `CLIENTNUM` column, which contains unique customer IDs, was removed as it does not contribute to the predictive power of the model. By excluding this column, we ensure the model focuses on meaningful insights and features.

Train Set (Training Set)
This is used to train the model. The model learns relationships and patterns in the data.
In the code, this is represented as `X_train` and `y_train`.

Test Set
This is used only at the end to evaluate the final performance of the model.
It remains "unseen" data for the model until this point.
In the code, this is represented as `X_test` and `y_test`.

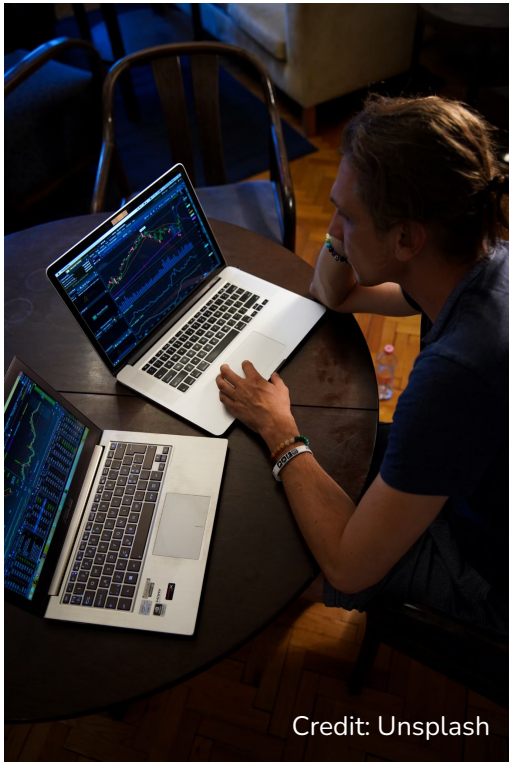
Validation Set
This is used to validate the model during training and to fine-tune hyperparameters.
It ensures the model performs well on data it hasn't seen during training.
In the code, this is represented as `X_val` and `y_val`.

Why Split the Dataset Before EDA of Bivariate?

To maintain model integrity and prevent data leakage, splitting the dataset into training, validation, and test sets **before EDA** is considered best practice. This approach ensures the following:

1. **No Information from the Test Set Leaks into the Model Training**
   EDA often involves identifying patterns, correlations, and outliers in the data. If this is done on the entire dataset, it risks influencing preprocessing steps and model tuning, leading to data leakage. By keeping the test set completely separate, we preserve its role as a truly unseen dataset to evaluate model generalization
2. **Avoiding Overfitting to Data-Specific Characteristics**
   Preprocessing decisions and feature engineering should be based only on the training set. Using the entire dataset during EDA can lead to preprocessing choices that overfit the specific characteristics of the data, rather than focusing on patterns that generalize to new data.
3. **Objective Evaluation of Model Performance**
   A well-isolated test set ensures that model evaluation reflects performance on unseen data, which is crucial for understanding real-world applicability.

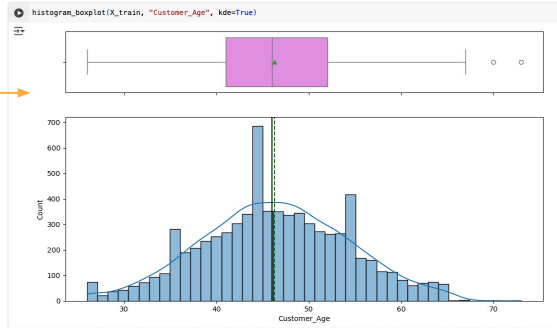
For these reasons, we split the dataset before EDA. EDA and preprocessing are then conducted exclusively on the training set, while the validation and test sets remain untouched until their designated stages in the workflow.


Credit: Unsplash

EDA (Univariate analysis)

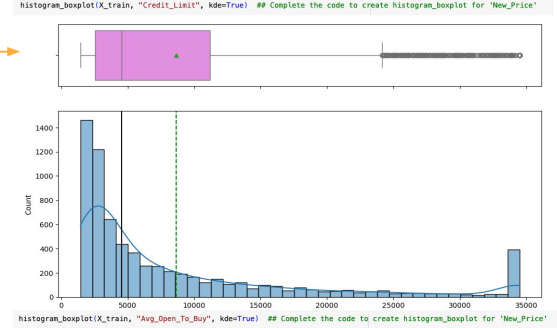
Customer Age

- **Distribution**: The ages of customers appear relatively evenly distributed, with a slight peak around middle age (~46 years).
- **Outliers**: A few outliers are observed in the older age groups, but they seem reasonable and represent older customers.
- **Conclusion**: No significant anomalies in customer age that require special handling.



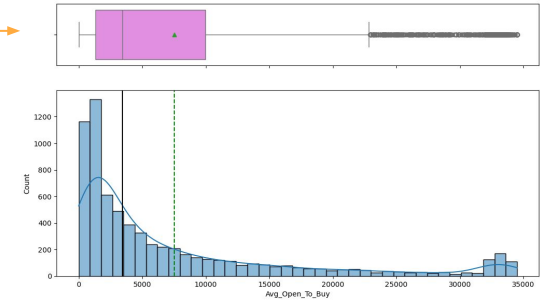
Credit Limit

- **Distribution**: The distribution is right-skewed, with many customers having low credit limits (~$5,000–$10,000).
- **Outliers**: Significant outliers with very high credit limits (up to $35,000), potentially representing premium customers.
- **Conclusion**: These outliers may be worth further analysis, as high credit limits could impact churn behavior.



Avg Open To Buy

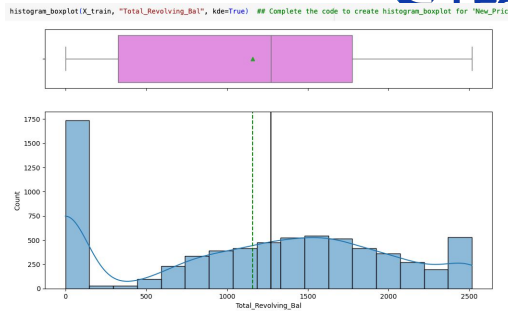
- **Distribution**: Similar to "Credit Limit," with a right-skewed distribution. Most customers have low available credit.
- **Outliers**: High values may correlate with premium customers or low credit utilization rates.
- **Conclusion**: An important metric to analyze for churn prediction, especially in relation to credit behavior.



EDA (Univariate analysis)

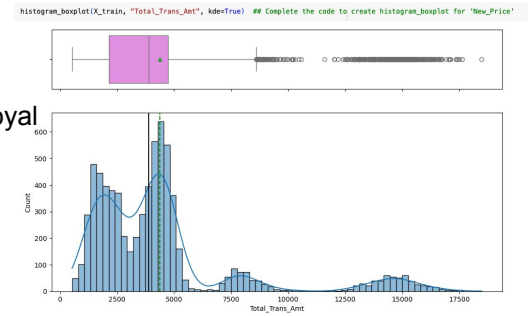
Total Revolving Bal

- **Distribution**: Left-skewed, with most customers having low revolving balances.
- **Outliers**: Present, but not as extreme as in "Credit Limit."
- **Conclusion**: Important to observe correlations between high revolving balances and churn.



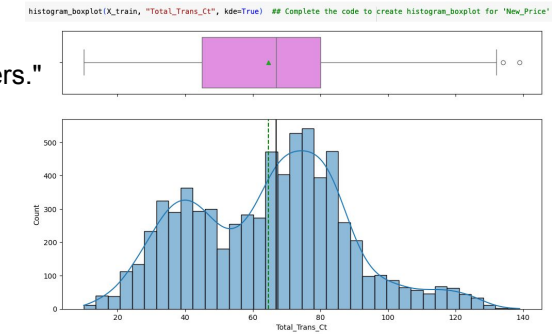
Total Trans Amt

- **Distribution**: Peaks around the mean (~$4,500) with a right-skewed distribution.
- **Outliers**: Customers with very high transaction amounts are outliers and could indicate a loyal customer base.
- **Conclusion**: These outliers could represent "high-value customers" critical for the bank to retain.



Total Trans Ct

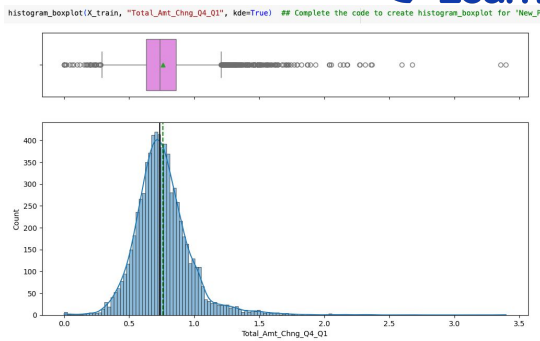
- **Distribution**: Symmetrical, with a peak around the average number of transactions (~65).
- **Outliers**: Customers with very high transaction counts (~100+) may be potential "power users."
- **Conclusion**: These customers are worth analyzing, as frequent users may be less likely to churn.

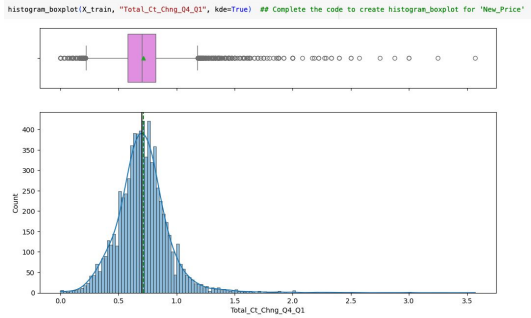


EDA (Univariate analysis)

Total Amt Chng Q4-Q1 and Total Ct Chng Q4-Q1

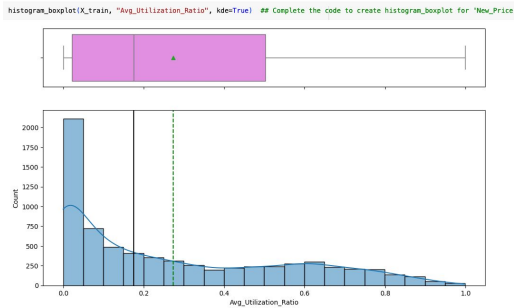
- **Distribution**: Similar patterns, peaking around low changes, with right-skewed distributions.
- **Outliers**: Extremely high changes may indicate unusual customer behavior.
- **Conclusion**: This feature may be significant for identifying customers with notable behavior changes, as high transaction count changes could signal irregular activity. Further analysis of the correlation between these outliers and churn is recommended.





Avg Utilization Ratio

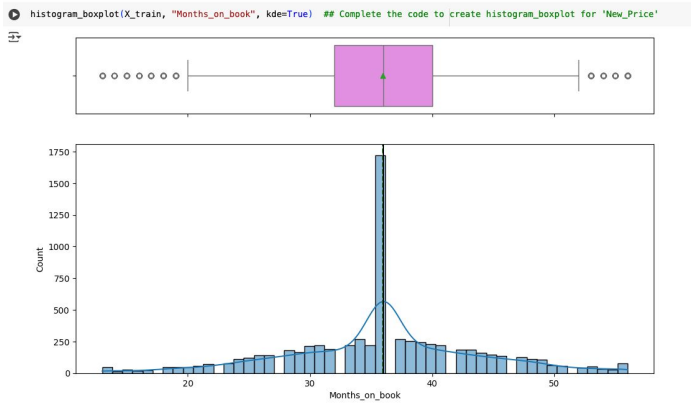
- **Distribution**: Low average credit utilization ratio (~0.27).
- **Outliers**: High values (>0.8) represent customers with higher credit utilization ratios.
- **Conclusion**: High utilization ratios could indicate financially stressed customers who may be more likely to churn.



EDA (Univariate analysis)

Months on Book

- **Distribution**: This feature is symmetrically distributed, with a peak around the average (~36 months). It appears well-balanced across the range of relationship lengths.
- **Outliers**: There are a few outliers at the lower and upper ends (e.g., customers with very short or very long tenures), but they seem plausible and represent rare cases.
- **Conclusion**: This feature reflects the length of the customer relationship, which may be a stable predictor of churn. Outliers might be worth investigating to determine if newer or long-standing customers are more prone to attrition.



EDA (Univariate analysis)

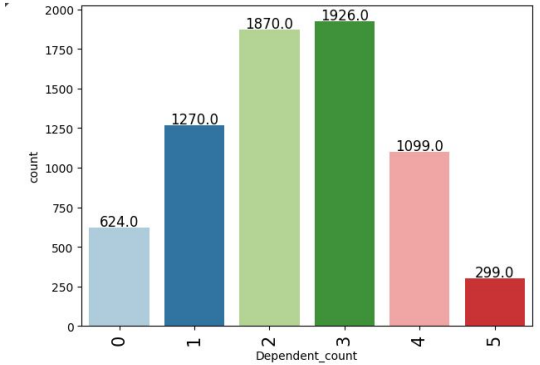
Dependent Count:

- **Distribution**: Most customers have 2 or 3 dependents, with fewer customers having 4 or 5 dependents. A significant portion of customers have no dependents.
- **Conclusion**: This feature may indicate the financial responsibilities of customers and could be relevant for churn analysis.

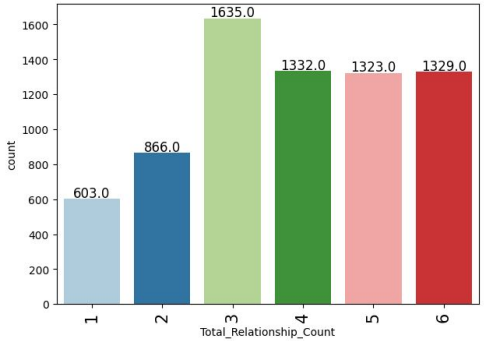
Total Relationship Count:

- **Distribution**: Customers are evenly distributed across relationship counts of 3, 4, 5, and 6 products, with fewer customers having 1 or 2 products.
- **Conclusion**: Customers with higher product relationships are potentially more engaged and could be less likely to churn.

```
] labeled_barplot(X_train, "Dependent_count")
```




```
labeled_barplot(X_train, "Total_Relationship_Count") ## Complete the code to c
```

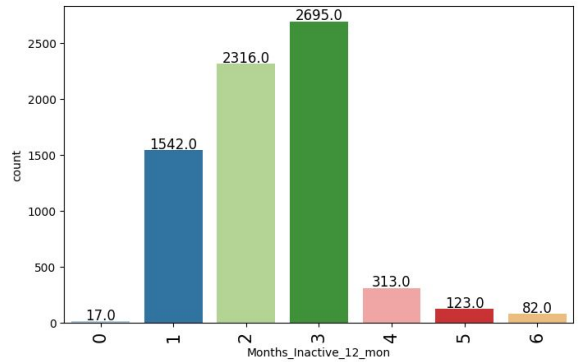


EDA (Univariate analysis)

Months Inactive (12 Months):

- **Distribution**: Most customers were inactive for 2 or 3 months. A small percentage were inactive for 5 or 6 months.
- **Conclusion**: Extended inactivity could be a strong predictor of churn.

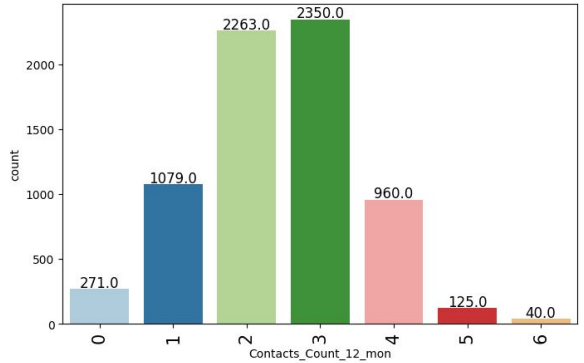
```
labeled_barplot(X_train, "Months_Inactive_12_mon") ## Complete the code to create labeled
```



Contacts Count (12 Months):

- **Distribution**: Most customers had 2 or 3 contacts in the last 12 months. A small number of customers had very few (0 or 1) or many contacts (5 or 6).
- **Conclusion**: Higher contact frequency might indicate higher engagement, while very low or very high contact counts could signal churn risks.

```
labeled_barplot(X_train, "Contacts_Count_12_mon") ## Complete the code to create label
```



EDA (Univariate analysis)

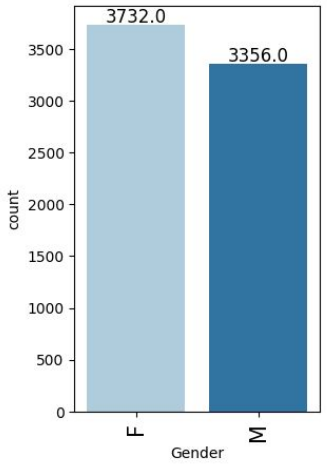
Gender:

- **Distribution**: The dataset has a balanced gender distribution with slightly more female customers.
- **Conclusion**: Gender alone may not have a significant impact on churn, but further bivariate analysis with churn rates is needed.

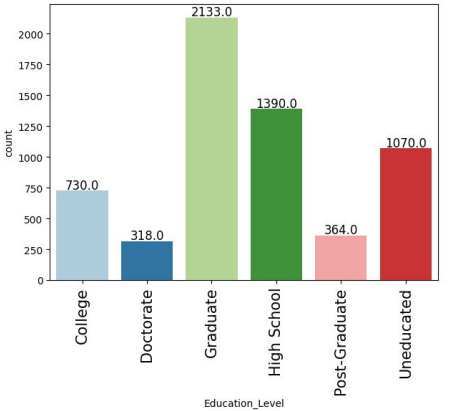
Education Level:

- **Distribution**: Most customers are graduates or have completed high school. Fewer customers have advanced degrees (postgraduate or doctorate).
- **Conclusion**: Education level could reflect customers' financial stability and could influence churn rates.

```
labeled_barplot(X_train, "Gender") ## Compl
```



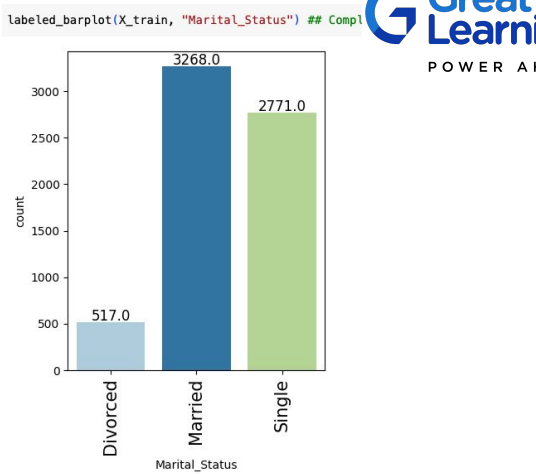

```
labeled_barplot(X_train, "Education_Level") ## Complete the code to create labeled_ba
```



EDA (Univariate analysis)

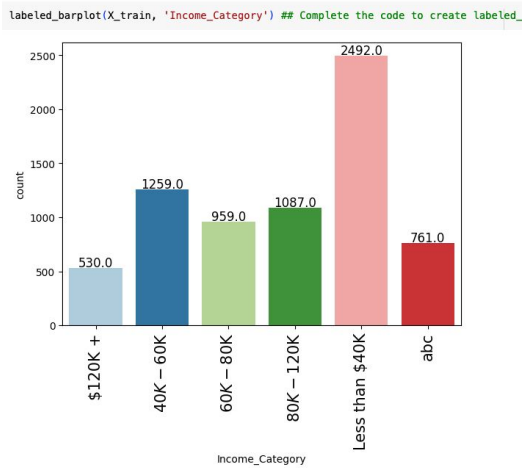
Marital Status:

- **Distribution**: Most customers are either married or single, with a smaller proportion being divorced.
- **Conclusion**: Marital status might influence financial behavior and churn risk.



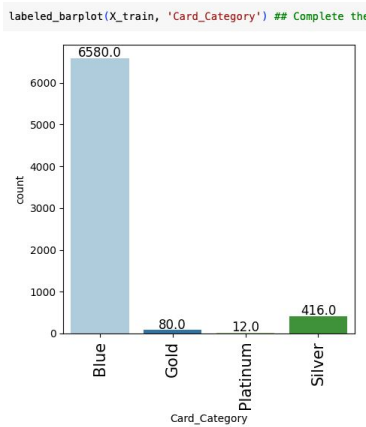
Income Category:

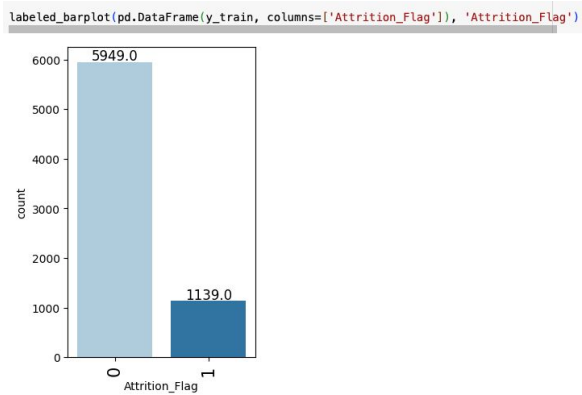
- **Distribution**: Most customers fall into the "Less than $40K" income category, with a steady decrease in frequency as income levels increase. The "abc" category represents invalid or missing data.
- **Conclusion**: Income levels could strongly correlate with churn behavior, especially for low-income or premium customers.



EDA (Univariate analysis)

1. **Card Category**:
   - **Distribution**: The majority of customers have a "Blue" card, with very few holding premium cards like "Gold," "Platinum," or "Silver."
   - **Conclusion**: Premium cardholders might represent a high-value segment critical for retention.
2. **Attrition Flag**:
   - **Distribution**: The majority of customers are existing customers, with a smaller fraction identified as attrited customers.
   - **Conclusion**: This reflects the imbalance in the dataset and highlights the need for techniques to address class imbalance during modeling.





Overall Conclusions Summary of Univariate analysis

Key Features as Predictors of Churn:

- Features such as **Months Inactive (12 Months)**, **Contacts Count (12 Months)**, **Income Category**, and **Card Category** show potential as significant predictors of churn.
- The **imbalance in the Attrition Flag** requires special attention during the modeling phase to avoid biased predictions.
- Invalid categories like `"abc"` in **Income Category** were identified and need proper handling during preprocessing.

Feature Distribution Insights:

- Several features, including **Credit Limit**, **Avg Open To Buy**, **Total Trans Amt**, and **Total Ct Chng Q4-Q1**, exhibit **right-skewed distributions** with notable outliers.
- **Months on Book** has a symmetrical distribution, representing the average customer relationship duration (~36 months), with some plausible outliers.

Outlier Analysis:

- Outliers are not necessarily errors; instead, they may represent important customer segments, such as:
  - Premium customers or "high-value customers."
  - Customers exhibiting significant changes in transaction or account behavior, which could signal churn risk.

Key Predictive Features:

- **Avg Utilization Ratio** and **change metrics** (e.g., **Total Amt Chng Q4-Q1** and **Total Ct Chng Q4-Q1**) are valuable for detecting behavioral changes that might correlate with churn.
- **Months on Book** provides insights into customer tenure, helping to identify whether newer or long-standing customers are more likely to churn

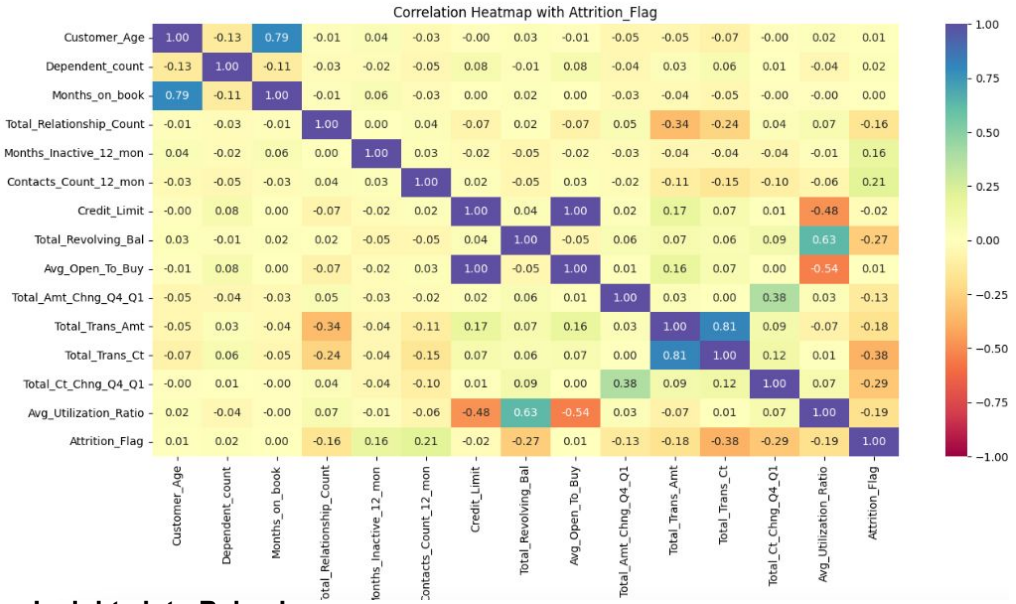
EDA on training set (Bivariate analysis)

Significant Features Correlated with `Attrition_Flag`:

- **Months_Inactive_12_mon (0.16)**: Indicates that higher months of inactivity are moderately correlated with customer attrition.
- **Contacts_Count_12_mon (0.21)**: More customer contact within 12 months positively correlates with attrition.
- **Avg_Utilization_Ratio (-0.19)**: A lower utilization ratio is associated with customer attrition, suggesting that active credit usage may reduce churn likelihood.

Strong Feature Correlations:

- **Total_Trans_Ct vs. Total_Trans_Amt (0.81)**: Total transaction count is strongly correlated with the total transaction amount, as expected.
- **Avg_Open_To_Buy vs. Credit_Limit (1)**: Customers with higher available credit typically have higher credit limits. (drop one)



Insights into Behavior:

- **Negative Correlation of Total_Trans_Ct (-0.38)**: Higher transaction counts reduce the likelihood of attrition, highlighting active users are less likely to churn.
- **Positive Correlation of Months_Inactive_12_mon**: Suggests inactivity is a reliable early indicator of potential churn.

EDA on training set (Bivariate analysis)

Other Observations:

- Many features have near-zero correlation with Attrition_Flag (e.g., Customer_Age, Total_Relationship_Count). These might not be significant predictors of churn on their own.
- Feature Removal**:** Avg_Open_To_Buy was found to have a perfect correlation with Credit_Limit. To eliminate redundancy and prevent multicollinearity, Avg_Open_To_Buy was dropped from the training, validation, and test datasets.

```
# Dropping because has perfect correlation with Credit_limit
# Droppa 'Avg_Open_To_Buy' från varje uppdelat dataset
X_train = X_train.drop(columns=['Avg_Open_To_Buy'])
X_val = X_val.drop(columns=['Avg_Open_To_Buy'])
X_test = X_test.drop(columns=['Avg_Open_To_Buy'])
```

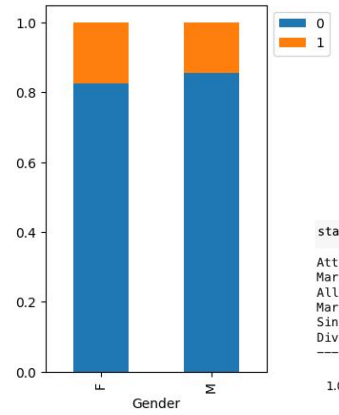
EDA on training set (Bivariate analysis)

Gender vs. Attrition_Flag:

- **Observations**:
  - The dataset is balanced between male (M) and female (F) customers.
  - Attrited customers (label 1) have a slightly higher proportion among female customers compared to male customers.
  - The majority of both male and female customers are existing customers (label 0).
- **Insights**:
  - Gender could potentially influence attrition behavior slightly, but the impact might not be very significant.

```
stacked_barplot(X_train_with_target, "Gender", "Attrition_Flag")
```

```
Attrition_Flag     0     1   All
Gender
All             5949  1139  7088
F               3082   650  3732
M               2867   489  3356
```

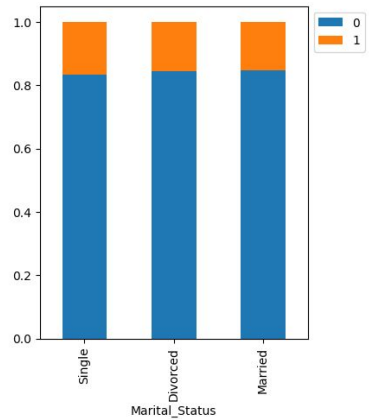


Marital_Status vs. Attrition_Flag:

- The majority of customers in the dataset are married, followed by single customers, and then divorced customers.
- Attrited customers (label 1) show a similar proportion across all marital statuses, with no significant differences.
- Married customers have the largest share of both attrited and existing customers.
- **Insights**:
  - Marital status does not seem to have a strong influence on attrition, as the proportion of attrited customers is similar across categories.

```
stacked_barplot(X_train_with_target,"Marital_Status",
```

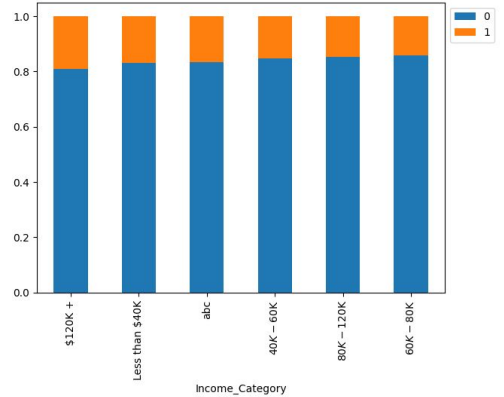
```
Attrition_Flag     0     1   All
Marital_Status
All             5510  1046  6556
Married         2766   502  3268
Single          2307   464  2771
Divorced         437    80   517
```



EDA on training set (Bivariate analysis)

Income_Category vs. Attrition_Flag:

Attrition_Flag	0	1	All
Income_Category			
All	5949	1139	7088
Less than $40K	2070	422	2492
$40K – $60K	1068	191	1259
$80K – $120K	926	161	1087
$60K – $80K	822	137	959
abc	634	127	761
$120K +	429	101	530

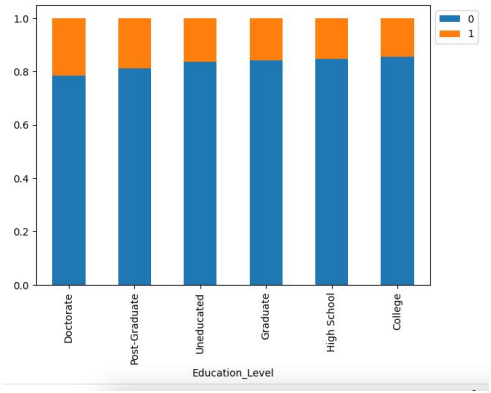


- Most customers fall in the "Less than $40K" income category, followed by "$40K–$60K".
- Attrition rates are relatively consistent across all income categories, with slight variation.
- The highest income categories, "$80K–$120K" and "$120K+", have the smallest number of customers but still show a similar trend of attrition.

Insights: Income category does not appear to be a strong predictor of attrition, though it could provide some nuanced insights when combined with other features.

```
stacked_barplot(X_train_with_target,"Education_Level", "Attrition_Flag") ## Com
```

Attrition_Flag	0	1	All
Education_Level			
All	5042	963	6005
Graduate	1796	337	2133
High School	1179	211	1390
Uneducated	896	174	1070
College	625	105	730
Doctorate	250	68	318
Post-Graduate	296	68	364



Education_Level vs. Attrition_Flag

- The majority of customers have a graduate-level education, followed by high school and uneducated customers.
- Attrited customers (label 1) are proportionately distributed across all education levels.
- There is no clear trend indicating that education level strongly impacts attrition.

Insights: Education level seems to have minimal influence on churn behavior, with similar patterns seen across all levels.

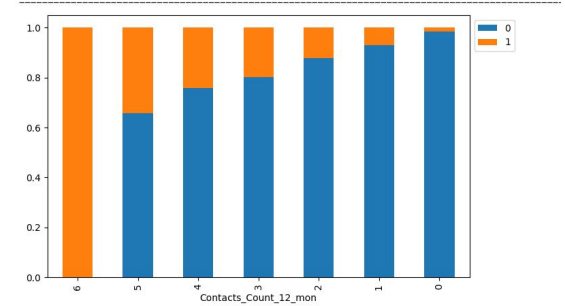
EDA on training set (Bivariate analysis)

Contacts_Count_12_mon vs. Attrition_Flag:

- **Observations**:
  - Customers with 2–3 contacts in the past 12 months have the highest attrition rates.
  - As the number of contacts increases beyond 3, attrition rates decrease significantly.
- **Insights**:
  - Higher customer engagement (more contacts) correlates with lower attrition rates, making this an important feature to analyze further.

```
stacked_barplot(X_train_with_target,"Contacts_Count_12_mon", "Attrition_Flag") ## Complete the
```

Contacts_Count_12_mon	Attrition_Flag 0	1	All
All	5949	1139	7088
3	1886	464	2350
2	1984	279	2263
4	728	232	960
1	1002	77	1079
5	82	43	125
6	0	40	40
0	267	4	271

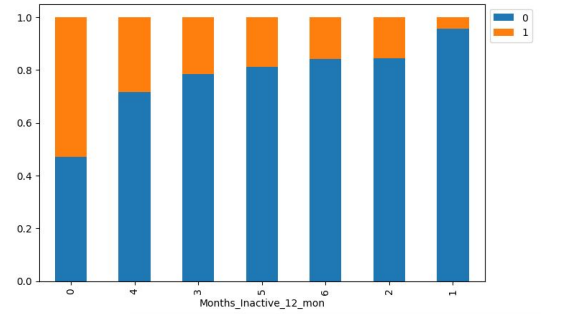


Months_Inactive_12_mon vs vs. Attrition_Flag:

- **Observations**:
  - Customers with 3–4 months of inactivity show the highest attrition rates.
  - Attrition rates decrease for customers with less inactivity or those with 0–1 inactive months.
- **Insights**:
  - Inactivity is a significant indicator of churn behavior, with more inactive months correlating with higher attrition rates.

```
stacked_barplot(X_train_with_target,"Months_Inactive_12_mon", "Attrition_Flag") ## Compl
```

Months_Inactive_12_mon	Attrition_Flag 0	1	All
All	5949	1139	7088
3	2114	581	2695
2	1959	357	2316
4	224	89	313
1	1475	67	1542
5	100	23	123
6	69	13	82
0	8	9	17



EDA on training set (Bivariate analysis)

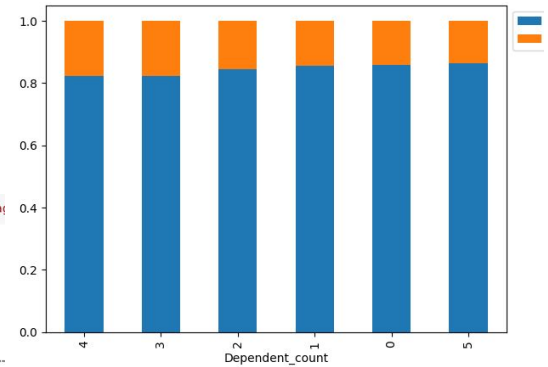
Dependent_Count vs. Attrition_Flag:

- Most customers have 2 or 3 dependents.
- Attrition rates remain consistent regardless of the number of dependents.

Insights: Dependent count does not appear to play a significant role in customer attrition.

```
stacked_barplot(X_train_with_target,"Dependent_count", "Attrition_Flag") #
```

Attrition_Flag	0	1	All
Dependent_count			
All	5949	1139	7088
3	1587	339	1926
2	1581	289	1870
4	903	196	1099
1	1085	185	1270
0	535	89	624
5	258	41	299



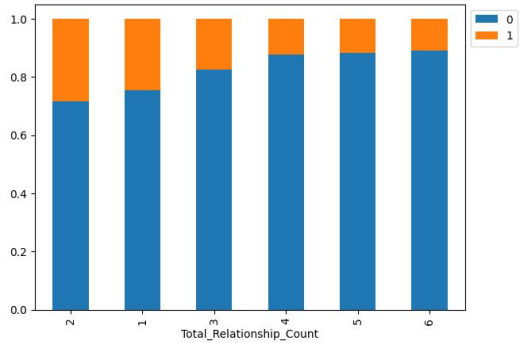
Total_Relationship_Count vs. Attrition_Flag:

- Customers with higher relationship counts (4–6) show lower attrition rates.
- Attrition rates are higher for customers with fewer relationships (1–2).

Insights: A higher number of total relationships correlates with lower attrition, indicating strong customer engagement reduces the likelihood of churn.

```
stacked_barplot(X_train_with_target,"Total_Relationship_Count", "Attrition_Fla
```

Attrition_Flag	0	1	All
Total_Relationship_Count			
All	5949	1139	7088
3	1351	284	1635
2	621	245	866
4	1169	163	1332
5	1168	155	1323
1	455	148	603
6	1185	144	1329



EDA on training set (Bivariate analysis)

Total_Revolving_Bal vs Attrition_Flag:

For Attrited Customers (Attrition_Flag = 1):

- The distribution is heavily skewed towards lower values of "Total_Revolving_Bal."
- Most attrited customers have a revolving balance close to 0.
- There are very few customers with high revolving balances (>1000).

For Non-Attrited Customers (Attrition_Flag = 0):

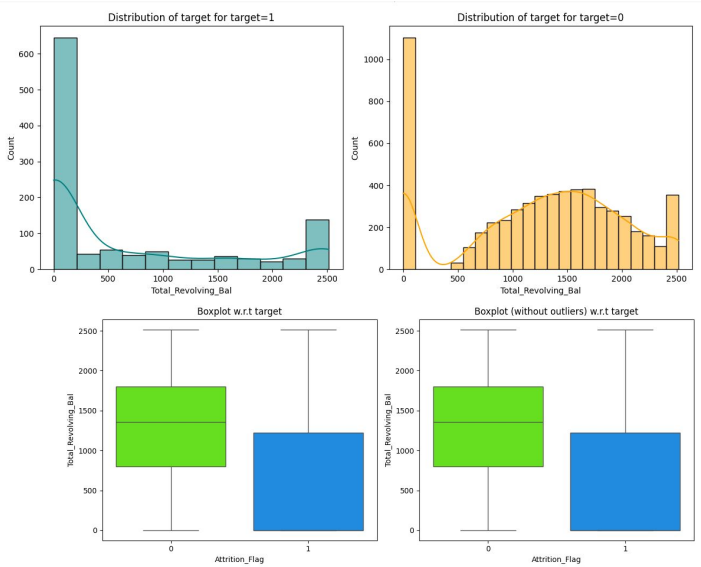
- The distribution is more spread out, with a significant proportion of customers having higher revolving balances.
- Non-attrited customers tend to have a wider range of revolving balances, with noticeable peaks across the spectrum.

Boxplot with Outliers:

- Attrited customers (label 1) generally have lower median "Total_Revolving_Bal" compared to non-attrited customers (label 0).
- There are outliers for both groups, but they are more pronounced in non-attrited customers.

Boxplot without Outliers:

- The difference between the median revolving balances of attrited and non-attrited customers is still apparent, showing that attrited customers tend to have lower balances.



Insights:

- Attrited customers are more likely to have lower revolving balances compared to non-attrited customers.
- Higher revolving balances might indicate more active or financially stable customers, potentially reducing the likelihood of attrition.
- This feature could be a strong predictor of attrition and should be prioritized in further analysis or modeling.

EDA on training set (Bivariate analysis)

Attrition_Flag vs Credit Limit

Target = 1 (Attrited Customers):

- The majority of customers who left (attrited) have lower credit limits.
- The distribution shows a right-skewed pattern, with a few customers having very high credit limits.

Target = 0 (Existing Customers):

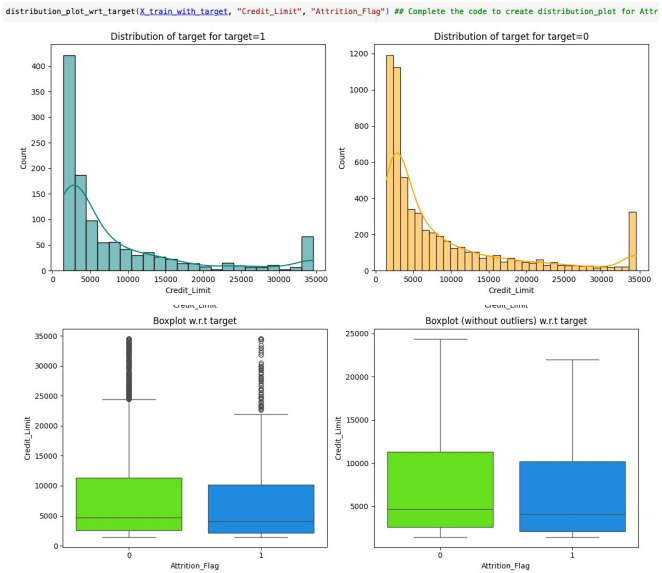
The distribution for existing customers also shows a right-skewed pattern, but a larger portion of these customers have medium-to-high credit limits compared to attrited customers.

Boxplot with Outliers:

- Attrited customers (1) have a narrower interquartile range (IQR), indicating less variability in their credit limits.
- Existing customers (0) show a wider IQR and higher median credit limits, suggesting they might be given more flexibility by the bank.

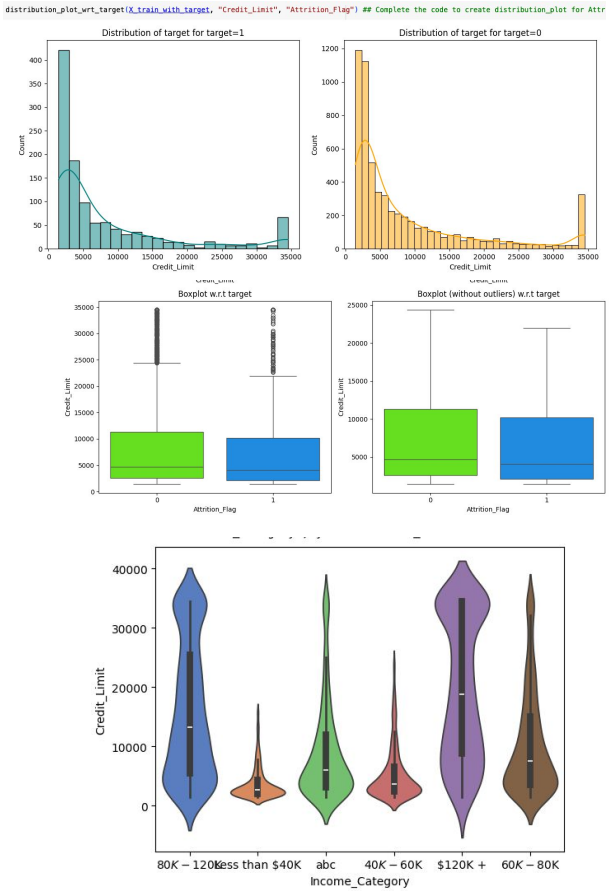
Boxplot without Outliers:

- The patterns remain consistent with the boxplot including outliers, reinforcing that the differences between the two groups are not heavily influenced by extreme values.



EDA on training set (Bivariate analysis)

Attrition_Flag vs Credit Limit



Key Insights:

- **Low Credit Limits Correlate with Attrition:** These customers are more likely to leave, possibly due to dissatisfaction or lower engagement.
- **Existing Customers Have Higher Credit Limits:** Medium-to-high credit limits suggest better satisfaction and engagement.
- **High Credit Limits Are Rarely Attrited:** These customers likely feel valued or benefit significantly from their credit arrangements.
- **Potential Data Errors:** Categories like "abc" indicate possible errors and should be corrected during preprocessing.
- **Income and Credit Limit Correlation:** Higher incomes are strongly associated with larger credit limits.

Recommendations:

- **Monitor Customers with Low Credit Limits:** Use personalized offers to increase engagement and reduce attrition.
- **Analyze High-Credit Attrited Customers:** Investigate rare cases of high-credit attrition for insights into unmet expectations or policy gaps.

EDA on training set (Bivariate analysis)

Attrition_Flag vs Customer_Age

Target = 1 (Attrited Customers):

- The age distribution for attrited customers is approximately normal, with a peak around 45 years.
- Most attrited customers fall within the age range of 30–55 years.

Target = 0 (Existing Customers):

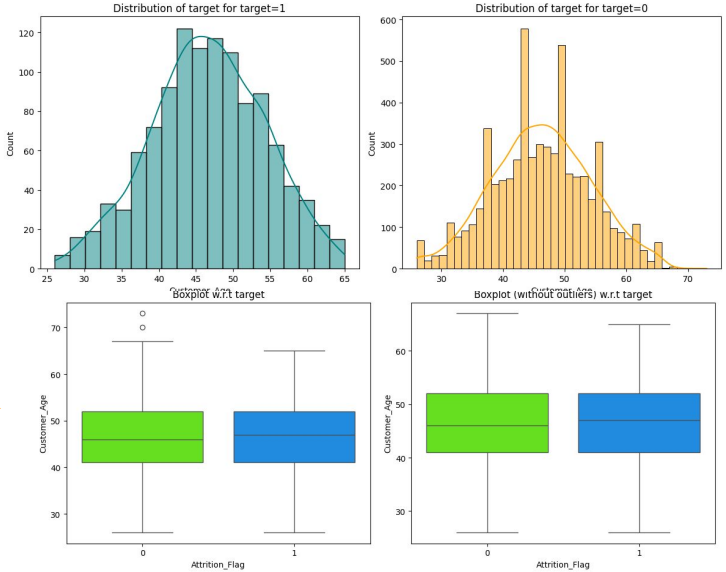
- The distribution for existing customers is also approximately normal, with a similar peak around 45 years.
- The age range for existing customers is broader, spanning from approximately 25–60 years, with a slight concentration in the middle range.

Boxplot with Outliers:

- Both groups have similar median ages (~45 years).
- A few younger and older customers appear as outliers in both groups.

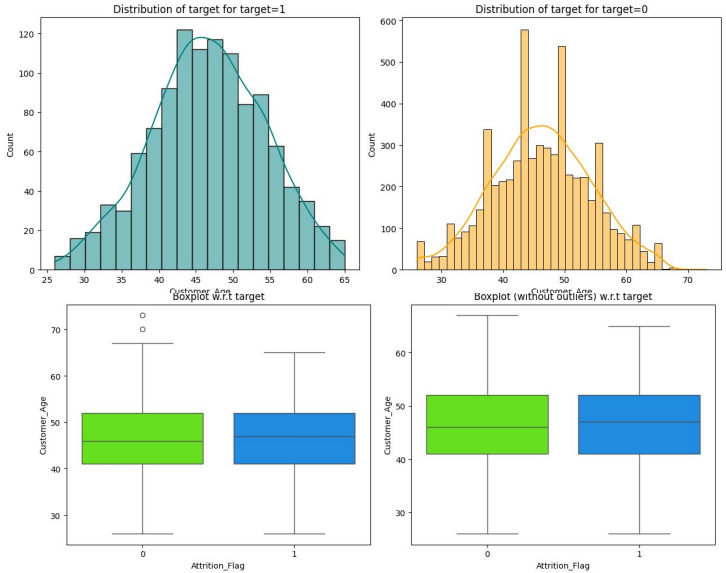
Boxplot without Outliers:

- The interquartile ranges (IQRs) are very similar for both groups, indicating comparable age distributions.



EDA on training set (Bivariate analysis)

Attrition_Flag vs Customer_Age



Key Insights:

- **Similar Age Distribution Across Groups:** Attrited and existing customers both center around ~45 years, with no significant differences in age distribution.
- **Middle-Aged Customers Drive Attrition:** Most attrited customers are aged ~30–55, highlighting a key demographic to address.
- **Minimal Impact from Age Outliers:** Youngest and oldest customers are outliers with negligible effect on attrition.

Recommendations:

- **Target Middle-Aged Customers:** Develop tailored products or loyalty programs to address the demographic most prone to attrition.
- **Engage Younger Customers:** Focus on fostering loyalty among younger customers to build long-term retention.

EDA on training set (Bivariate analysis)

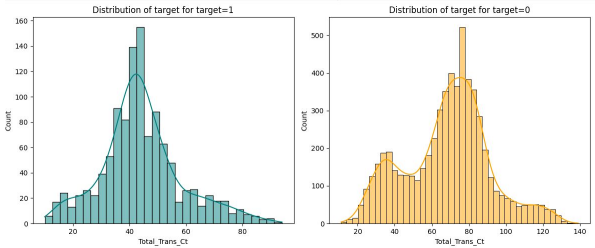
Attrition_Flag vs Total_Trans_Ct

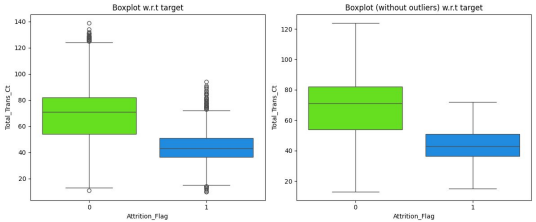
Target = 1 (Attrited Customers):

- The distribution for attrited customers is narrower, with most customers having a low number of transactions.
- The peak transaction count for attrited customers is around **50 transactions**.
- Few customers in this group have a high transaction count (above 80).

Target = 0 (Existing Customers):

- The distribution is broader and right-skewed, with many customers having higher transaction counts.
- The peak transaction count is approximately **90–100 transactions**, indicating frequent use by existing customers.
- There is a noticeable tail extending toward high transaction counts (above 100).





Boxplot with Outliers:

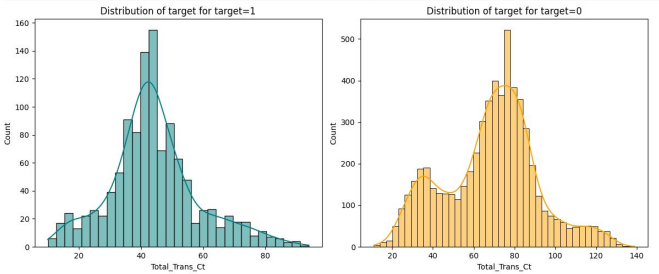
- Attrited customers have a lower median transaction count compared to existing customers.
- Outliers in the attrited group represent customers with a relatively high transaction count.

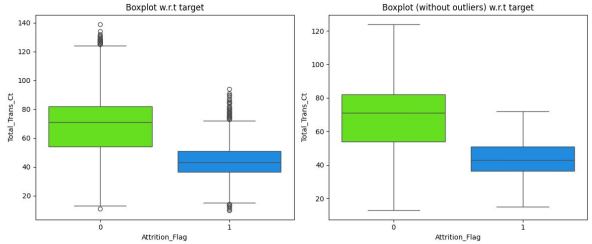
Boxplot without Outliers:

- The median for existing customers is much higher compared to attrited customers, emphasizing the difference in transaction activity between the two groups.
- The interquartile range (IQR) for existing customers is broader, suggesting greater variability in transaction behavior.

EDA on training set (Bivariate analysis)

Attrition_Flag vs Total_Trans_Ct





Key Insights:

High Transaction Counts Are Indicative of Loyalty:

- Customers with a higher number of transactions (above 70–80) are more likely to be retained.
- Attrited customers typically have lower transaction counts, with most falling below 60.

Low Activity Indicates Potential Attrition:

- Customers with a low transaction count are more prone to attrition. This highlights the importance of monitoring and engaging customers with declining activity.

Outliers in Attrited Customers:

- Some attrited customers with high transaction counts might indicate dissatisfaction despite frequent card use. These cases warrant further investigation.

Recommendations:

- **Monitor Low Activity Customers**:
  - Implement engagement campaigns for customers with low transaction counts, offering incentives to increase activity and reduce the risk of attrition.
- **Analyze High-Transaction Attrition Cases**:
  - Investigate the reasons behind attrition for customers with high transaction counts, as these could represent valuable customers who left due to dissatisfaction or unmet needs.

EDA on training set (Bivariate analysis)

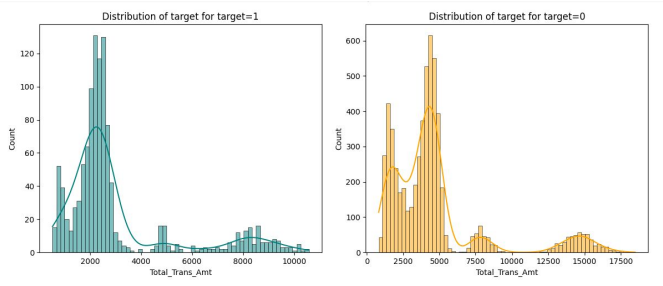
Attrition_Flag vs Total_Trans_Amt

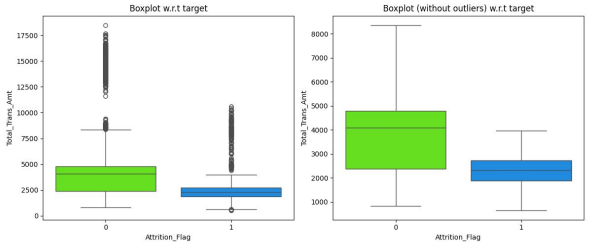
Distribution (Target=1 - Attrited Customers):

- Customers who have churned (Attrition_Flag=1) show a lower transaction amount overall.
- The distribution peaks around $2,500–$4,000, indicating a concentration of attrited customers with relatively low transaction amounts.
- The distribution tails off sharply after $5,000, showing fewer attrited customers with higher transaction amounts.

Distribution (Target=0 - Existing Customers):

- Existing customers (Attrition_Flag=0) display a higher transaction amount compared to attrited customers.
- The distribution is broader and skewed right, with peaks between $4,000–$6,000 and extending to much higher values up to $12,000.
- This suggests that customers with higher transaction amounts are more likely to stay with the company.



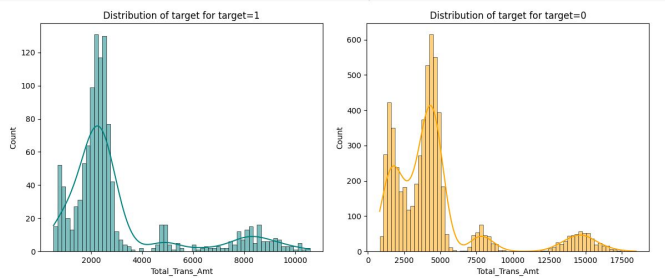


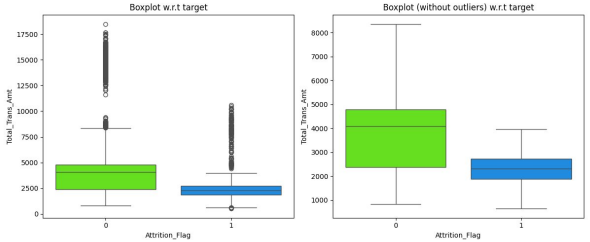
Boxplots:

- The boxplots reveal that the median transaction amount for attrited customers is significantly lower compared to existing customers.
- Outliers are visible for both categories, but they are more pronounced among existing customers, indicating some customers with very high transaction amounts (likely high-value customers).
- The boxplot with outliers removed confirms the gap in median and range between the two categories.

EDA on training set (Bivariate analysis)

Attrition_Flag vs Total_Trans_Amt





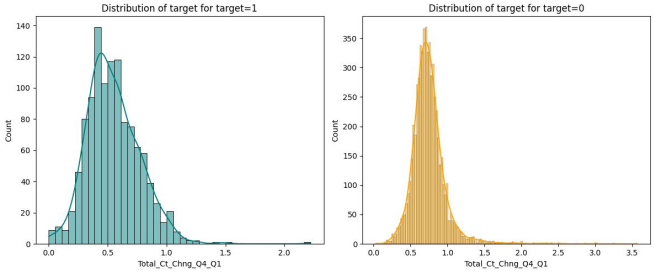
Key Insights:

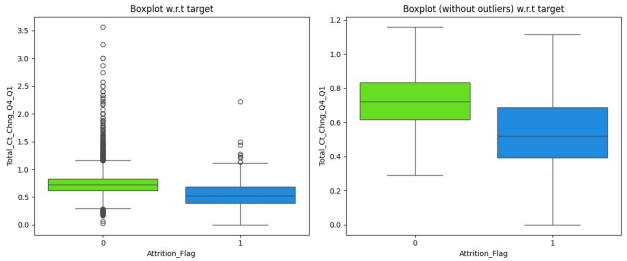
- Transaction amount is strongly correlated with customer retention. Higher transaction amounts are more likely to be associated with existing customers.
- The concentration of attrited customers at lower transaction amounts suggests that low spending behavior might be a predictor of churn.
- Outliers among existing customers represent a valuable segment (high-value customers), emphasizing the importance of retention strategies for these customers.

EDA on training set (Bivariate analysis)

Attrition_Flag vs Total_Ct_Chng_Q4_Q1

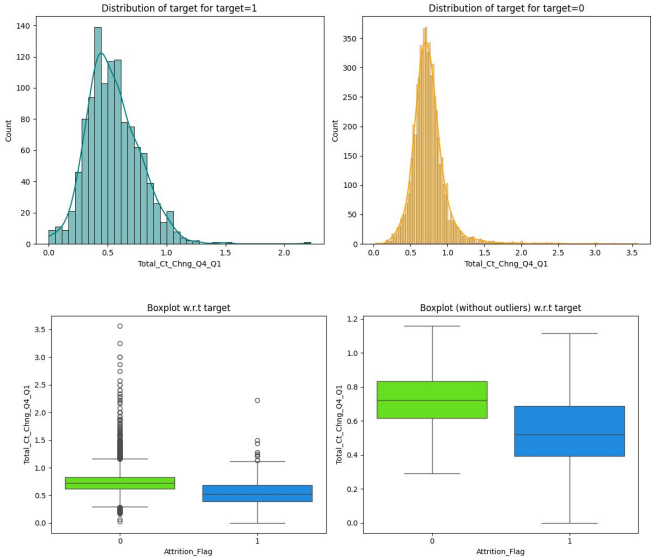
1. **Distributions:**
   - **For Attrition_Flag = 1 (Attrited Customers):**
     - The distribution is slightly left-skewed, with a peak around 0.4 to 0.5. This indicates that customers who have left (attrited) tend to have lower transaction count changes between Q4 and Q1.
   - **For Attrition_Flag = 0 (Existing Customers):**
     - The distribution is more balanced and right-skewed, with a higher concentration around 0.7 to 0.8. Existing customers tend to have higher transaction count changes compared to attrited customers.
2. **Boxplots:**
   - **Boxplot with Outliers:**
     - Existing customers (Attrition_Flag = 0) have a significantly higher median transaction count change compared to attrited customers.
     - The range of values for existing customers is broader, with more extreme outliers on the higher end.
   - **Boxplot without Outliers:**
     - The median difference remains evident, reinforcing that existing customers generally exhibit higher transaction count changes than attrited customers.
     - The interquartile range (IQR) for existing customers is also wider than that for attrited customers.





EDA on training set (Bivariate analysis)

Attrition_Flag vs Total_Ct_Chng_Q4_Q1



Insights:

- Transaction count change between Q4 and Q1 is a strong differentiator between attrited and existing customers.
- Customers with consistently low transaction count changes (e.g., below 0.5) are more likely to churn.
- Customers with higher transaction count changes tend to remain loyal, possibly indicating increased engagement with their credit cards.

Conclusion:

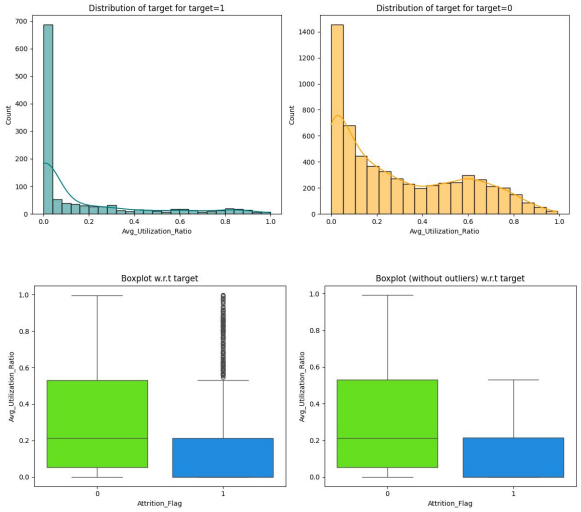
Total_Ct_Chng_Q4_Q1 is a crucial feature for understanding customer attrition. Its strong correlation with Attrition_Flag suggests that changes in transaction behavior over time could be used as a predictive indicator for customer churn.

EDA on training set (Bivariate analysis)

Attrition_Flag vs Avg_Utilization_Ratio

Distribution Analysis:

- **Target=1 (Attrited Customers)**: The distribution of `Avg_Utilization_Ratio` for attrited customers is skewed towards lower utilization ratios, with most customers having a utilization ratio below 0.3. However, there is a noticeable tail extending towards higher utilization ratios, indicating that some attrited customers have utilized their credit more extensively.
- **Target=0 (Existing Customers)**: The distribution for existing customers is also right-skewed, but the peak occurs at a slightly higher utilization ratio compared to attrited customers. The distribution shows that existing customers generally have lower credit utilization ratios than the few outliers with ratios closer to 1.0.



Boxplot Analysis:

- **With Outliers**: The boxplot indicates that existing customers (Target=0) have a higher median utilization ratio compared to attrited customers (Target=1). However, there are several outliers for both groups, particularly in the attrited group, which have very high utilization ratios.
- **Without Outliers**: After removing outliers, the median and interquartile range (IQR) for the utilization ratio remain slightly higher for existing customers, suggesting that credit utilization plays a role in retention.

EDA on training set (Bivariate analysis)

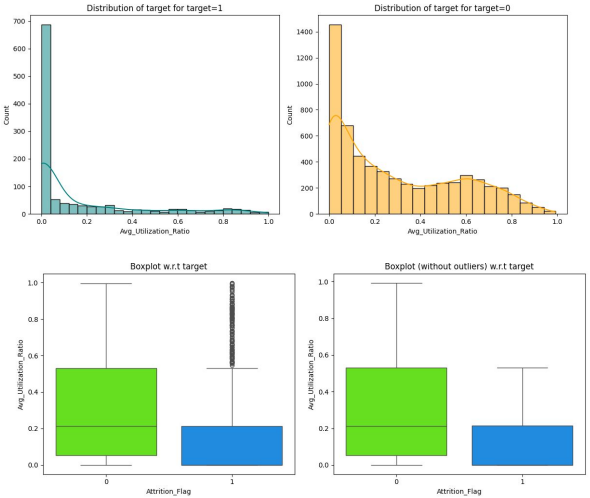
Attrition_Flag vs Avg_Utilization_Ratio

Insights:

- Customers with moderately high utilization ratios tend to stay as existing customers, indicating higher engagement with credit correlates with retention. However, very high utilization ratios (outliers) might indicate financial stress, which could increase the risk of attrition

Business Implication:

- Monitoring customers with very high utilization ratios is essential to identify those at potential risk of attrition due to financial stress or over-reliance on credit. Interventions, such as offering financial counseling or adjusting credit terms, may help retain such customers.
- Further analysis of the relationship between credit utilization and customer satisfaction might provide additional insights for customer retention strategies.



EDA on training set (Bivariate analysis)

Attrition_Flag vs Months_on_book

Target = 1 (Attrited Customers):

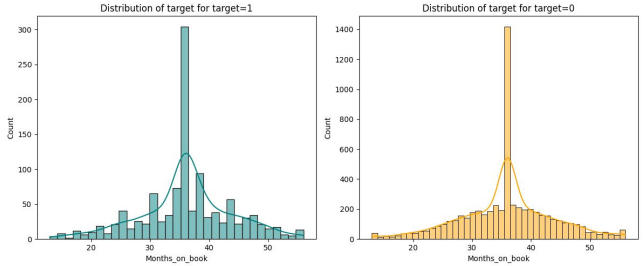
- The distribution shows a slight concentration of attrited customers around **30-40 months** on book.
- A small number of attrited customers have fewer than 20 or more than 50 months on book.
- The distribution is fairly narrow but slightly right-skewed.

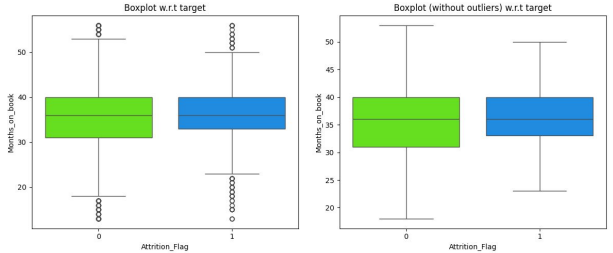
Target = 0 (Existing Customers):

- The distribution is more **centered around 35 months**, indicating most existing customers have similar tenure.
- The curve is more balanced with a broader spread compared to attrited customers.
- There are no extreme outliers or anomalies.

Boxplot Analysis:

- The **mean and median** number of months on book for both groups are similar, but the attrited customers tend to have a slightly wider spread.
- **Without outliers**, the boxplot still shows similar characteristics, with existing customers slightly clustering around the average.





Key Insights:

- Months on book is not a **strong differentiator** for predicting attrition.
- The similarity between the two groups suggests tenure alone is not enough to predict customer behavior, though it might interact with other features.
- This variable could still contribute to modeling as part of multi-feature analysis or feature engineering, especially in combination with customer activity measures like transaction counts.

EDA on training set (Bivariate analysis)

Attrition_Flag vs Total_Revolving_Bal

Target = 1 (Attrited Customers):

- Highly skewed distribution with most customers having low revolving balances (~0–500).
- Few customers have high revolving balances.

Target = 0 (Existing Customers):

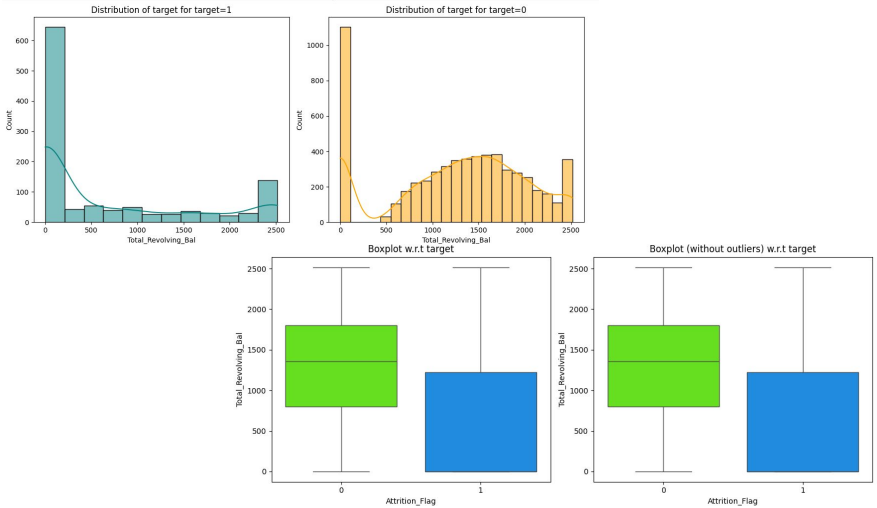
- More uniform distribution, peaking around 500–1000 with a gradual decline at higher balances.
- A larger proportion of existing customers have mid-to-high revolving balances.

Boxplot with Outliers:

- Existing customers have a higher median revolving balance and greater variation compared to attrited customers.
- Attrited customers show a narrower interquartile range (IQR).

Boxplot without Outliers:

- The higher median for existing customers remains, and the spread is more condensed, but the overall trend is consistent.



Insights:

- **Lower Revolving Balances:** Attrited customers tend to have significantly lower revolving balances, which could indicate a lack of engagement or activity with their credit cards.
- **Higher Revolving Balances:** Existing customers with higher revolving balances might indicate more frequent credit card usage and active participation.
- **Key Takeaway:** Total Revolving Bal may serve as an important predictor for customer churn, with lower balances correlating to a higher likelihood of attrition.

EDA on training set (Bivariate analysis) Overall conclusion

1. Key Predictors of Attrition

Total_Trans_Ct (Total Transaction Count):

- Customers with fewer transactions (<60) are more likely to churn, while those with higher transactions (>80) tend to remain loyal.
- This highlights activity level as a strong predictor of customer retention.

Total_Trans_Amt (Total Transaction Amount):

- Lower transaction amounts (<$5,000) correlate with higher attrition rates.
- Customers with higher spending levels (> $6,000) tend to stay, suggesting that spending behavior is linked to loyalty.

Avg_Utilization_Ratio (Credit Utilization):

- Lower utilization ratios (<0.3) are more common among attrited customers, while higher ratios (>0.4) are associated with retention.
- Indicates that active credit usage helps retain customers.

Contacts_Count_12_mon:

- Higher contact frequency (>3 in 12 months) correlates with lower attrition rates, suggesting proactive engagement is important for retention.

Months_Inactive_12_mon:

- Attrited customers often have higher inactivity levels (3+ months of inactivity), making this a significant churn indicator.

Total_Revolving_Bal:

- Attrited customers generally have lower revolving balances (<$500), indicating reduced engagement with their credit accounts.

Credit_Limit:

- Customers with lower credit limits are more likely to attrite, possibly indicating dissatisfaction or lack of engagement.
- Higher credit limits correlate with retention, as these customers may feel more valued by the bank.

EDA on training set (Bivariate analysis) Overall conclusion

2. Features with Minimal Influence

Customer_Age:

- Age distribution is similar across attrited and existing customers, with both groups concentrated around the middle-aged range (30–55 years). This makes age less relevant for predicting attrition.

Income_Category and Education_Level:

- No significant differences in attrition trends across income or education categories.

Months_on_Book:

- Tenure (relationship length) alone does not strongly differentiate attrited and existing customers, though it may interact with other features.

3. Correlation Insights

Strong Correlations Between Features:

- Total_Trans_Ct and Total_Trans_Amt are highly correlated, indicating that transaction count drives transaction amount.
- Avg_Open_To_Buy and Credit_Limit are moderately correlated, as customers with higher credit limits tend to have more available credit.

Attrition_Flag Correlation:

- Moderate positive correlations with Contacts_Count_12_mon and Months_Inactive_12_mon.
- Moderate negative correlations with Total_Trans_Ct, Avg_Utilization_Ratio, and Total_Trans_Amt.

EDA on training set (Bivariate analysis) Overall conclusion

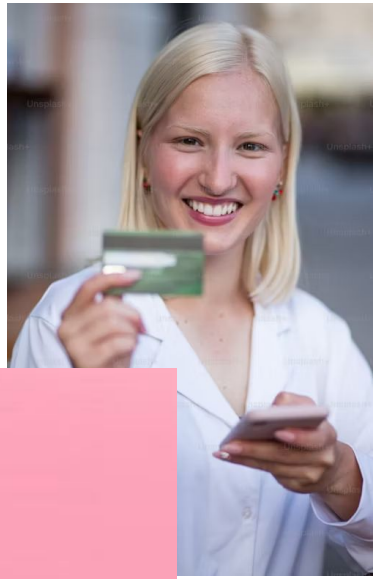
4. Key Recommendations

Focus on Activity Metrics:

- Incentivize increased usage through targeted campaigns.
- Monitor and engage customers with declining transaction counts and amounts.

Encourage Proactive Contact:

- Ensure regular and meaningful communication with customers to improve retention.



Credit: Unsplash

EDA Insights & Improvements

EDA Insights

The exploratory data analysis revealed several key patterns related to customer churn:

Transaction Activity:

- Customers with fewer transactions (<60/year) are more likely to churn, suggesting engagement strategies are crucial.

Credit Utilization:

- Higher credit utilization (>0.4) correlates with retention, emphasizing the need for proactive financial management.

Contact Frequency:

- Higher contact frequency (>3/year) reduces churn, highlighting the importance of customer touchpoints.

Improve Credit Engagement

Analyze customers with low credit utilization and offer tailored solutions to increase engagement.

Target Inactive Customers:

- Implement strategies to re-engage customers with 3+ months of inactivity.

High-Value Customer Retention:

- Focus retention efforts on high-spending and high-transaction customers as they are the most loyal and profitable segment.

Data Preprocessing

Dataset Separation Verification:

- Before beginning the data preprocessing, we confirmed that the training, validation, and test datasets were strictly separated.
- Performed checks to ensure:
  - No overlap existed between the training set (X_train) and the validation or test sets.
  - Index intersections between these sets were verified to have zero overlapping rows.
- This validation step was critical to maintaining the integrity of the datasets and ensuring unbiased model training and evaluation, effectively mitigating any risk of data leakage.

```
print(f"Original dataset length: {len(data)}")
print(f"Training set length: {len(X_train_with_target)}")
print(f"Validation + Test sets length: {len(X_val) + len(X_test)}")
```

```
Original dataset length: 10127
Training set length: 7088
Validation + Test sets length: 3039
```

```
# Check for overlaps between training and validation/test sets
overlap_val = set(X_train_with_target.index).intersection(set(X_val.index))
overlap_test = set(X_train_with_target.index).intersection(set(X_test.index))

print(f"Overlap with validation set: {len(overlap_val)} rows")
print(f"Overlap with test set: {len(overlap_test)} rows")
```

```
Overlap with validation set: 0 rows
Overlap with test set: 0 rows
```

This approach demonstrates adherence to best practices for machine learning workflows and ensures that preprocessing and modeling steps are based on correctly partitioned datasets.

Data Preprocessing

Handling Missing Values

- For categorical columns:
  - We used SimpleImputer with the strategy "most_frequent" to fill missing values with the most common category.
- For numerical columns:
  - We used `SimpleImputer` with the strategy "median" to fill missing values with the median value, minimizing the impact of outliers.
- After imputation, we confirmed that no missing values remained in the training, validation, or test datasets.

Validation of Data Values

- Unique values in each column were examined to identify any invalid or unexpected values.
- Specifically, the invalid value `"abc"` in the Income_Category column was replaced with NaN and subsequently imputed with the most frequent category.

```
[554] from sklearn.impute import SimpleImputer

      # Imputera för kategoriska kolumner
      categorical_cols = X_train.select_dtypes(include=["object"]).columns
      imputer_cat = SimpleImputer(strategy="most_frequent")
      X_train[categorical_cols] = imputer_cat.fit_transform(X_train[categorical_cols])
```

```
[555] # Imputera för numeriska kolumner
      numeric_cols = X_train.select_dtypes(include=["float64", "int64"]).columns
      imputer_num = SimpleImputer(strategy="median")
      X_train[numeric_cols] = imputer_num.fit_transform(X_train[numeric_cols])
```

```
] # Kontrollera unika värden i kategoriska kolumner
  categorical_cols = X_train.select_dtypes(include=["object"]).columns
  for col in categorical_cols:
      print(f"Unique values in {col}: {X_train[col].unique()}")
```

```
Unique values in Gender: ['M' 'F']
Unique values in Education_Level: ['Graduate' 'Uneducated' 'High School' 'College' 'Doctorate'
 'Post-Graduate']
Unique values in Marital_Status: ['Single' 'Married' 'Divorced']
Unique values in Income_Category: ['$80K - $120K' 'Less than $40K' 'abc' '$40K - $60K' '$120K +'
 '$60K - $80K']
Unique values in Card_Category: ['Blue' 'Silver' 'Gold' 'Platinum']
```

```
# Ersätt det ogiltiga värdet "abc" med NaN
X_train["Income_Category"].replace("abc", np.nan, inplace=True)

# Fyll NaN med den vanligaste kategorin (mode)
X_train["Income_Category"].fillna(X_train["Income_Category"].mode()[0], inplace=True)
```

Data Preprocessing

One-Hot Encoding

- Categorical columns were transformed into numerical representations using `pd.get_dummies`, ensuring compatibility with machine learning models.
- The transformation was validated to confirm that all categories were correctly represented in the dataset.
- Encoded categorical variables and split data before preprocessing to prevent leakage.

```
# One-hot encoding för de kategoriska kolumnerna
categorical_cols = X_train.select_dtypes(include=["object"]).columns
X_train = pd.get_dummies(X_train, columns=categorical_cols, drop_first=True)

# Kontrollera resultatet
print(X_train.head())
print(X_train.dtypes)
```

```
# Säkerställ att samma one-hot encoding tillämpas på validerings- och testdataset
X_val = pd.get_dummies(X_val, columns=categorical_cols, drop_first=True)
X_test = pd.get_dummies(X_test, columns=categorical_cols, drop_first=True)

# Justera kolumner för att matcha X_train
X_val = X_val.reindex(columns=X_train.columns, fill_value=0)
X_test = X_test.reindex(columns=X_train.columns, fill_value=0)

# Kontrollera att inga kolumner saknas eller är överflödiga
print(X_val.head())
print(X_test.head())
```

Dataset Structure Validation

- All datasets (training, validation, and test) were checked to ensure:
  - The same number of columns in each dataset.
  - Consistent data types and column names across datasets.
- The datasets now contain only numerical and boolean values, making them ready for modeling.

We now have a clean, well-structured dataset ready for model training. All preprocessing steps were carried out following best practices to ensure data integrity and prevent data leakage. The datasets have been consistently handled and contain no missing or invalid values.

Data Preprocessing

Validation of Preprocessed Data:

- Verified the new minimum and maximum values for each numeric column to confirm that capping was applied correctly.
- Ensured that the dataset is now free of extreme outliers that could distort model training.

Maintaining Data Integrity:

- Confirmed no loss of essential information during preprocessing.
- Retained original dataset characteristics while improving quality for modeling.

Prepared for Next Steps:

- Cleaned data is now ready for encoding categorical variables and imputing missing values.
- Preprocessing aligns with best practices to avoid data leakage and ensures consistent treatment of the train, validation, and test sets.

```python
# Kontrollera unika värden och identifiera konstiga värden
for col in X_train.columns:
    if X_train[col].dtype == "object" or X_train[col].dtype == "bool":
        # För kategoriska och boolska kolumner
        unique_values = X_train[col].unique()
        print(f"Unique values in {col}: {unique_values}")
        print("-" * 30)
    elif X_train[col].dtype in ["int64", "float64"]:
        # För numeriska kolumner, leta efter outliers eller värden som ser avvikande ut
        min_val = X_train[col].min()
        max_val = X_train[col].max()
        print(f"{col}: Min value = {min_val}, Max value = {max_val}")
        # Kontrollera om negativa värden eller extrema värden kan vara problematiska
        if min_val < 0:
            print(f"Warning: Negative value detected in {col}")
        print("-" * 30)

# Kontrollera om det finns null eller NaN-värden i någon kolumn
print("Missing values in X_train:")
print(X_train.isna().sum())
```

This process ensures that the dataset is robust, reliable, and optimized for the subsequent steps in machine learning modeling.

Addressing Class Imbalance for Fair Evaluation

Class imbalance—where one class (e.g., "non-churn") significantly outnumbers the other (e.g., "churn")—can skew model performance by favoring the majority class. To address this and ensure a fair evaluation:

1. **Undersampling**:
   - **What it does**: Reduces the majority class by randomly selecting a subset of its data, balancing it with the minority class.
   - **Benefit**: Encourages the model to focus equally on both classes, improving its ability to identify churners.
   - **Trade-Off**: While effective, it may discard valuable data, potentially impacting the model's ability to generalize.
2. **Oversampling**:
   - **What it does**: Increases the minority class by duplicating its data or generating synthetic samples.
   - **Benefit**: Retains all available data from the majority class while balancing the dataset.
   - **Trade-Off**: Risk of overfitting, as the model might memorize repeated or synthetic samples.
3. **Outcome**:
   - By applying both undersampling and oversampling during different stages of model building, we ensured robust evaluation across models and avoided bias toward the majority class.
   - These techniques were particularly valuable for improving **Recall**, ensuring the model identified as many at-risk customers as possible.

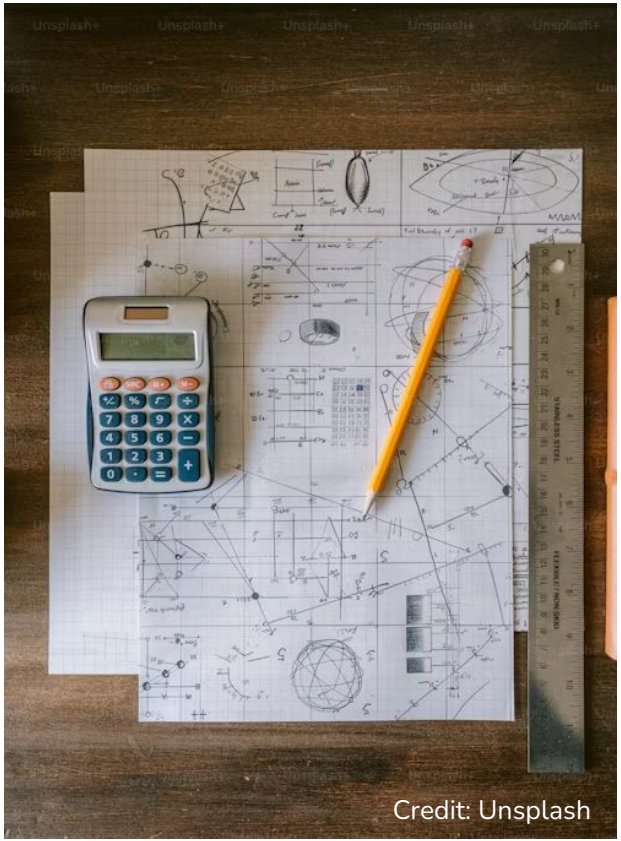
Model building

Objective recap

Objective: To predict customer attrition (whether a customer is likely to leave or stay) using machine learning models.

Business Need: The focus was to maximize **recall**, as it is critical to identify as many attrited customers as possible. Missing potential attrition cases could result in lost revenue opportunities.

Problem Scope: The dataset provided detailed information about customer demographics, transactional history, and engagement with the bank.



Model building

Models Used

- **Why these models?**
  - **Bagging and Random Forest:** Useful for reducing variance and handling complex data relationships.
  - **Boosting Algorithms (AdaBoost, Gradient Boosting, XGBoost):** Known for improving weak learners and handling imbalanced data effectively.
- Briefly explaining each model:
  - **Bagging:** Averages predictions from multiple decision trees to reduce overfitting.
  - **Random Forest:** Extends bagging with feature selection at each split, improving generalization.
  - **AdaBoost:** Sequentially improves weak learners by focusing on misclassified data.
  - **Gradient Boosting:** Optimizes a loss function iteratively, offering strong performance on structured data.
  - **XGBoost:** An efficient and regularized version of Gradient Boosting, designed for scalability and speed.

Baseline model

```
Training Performance:

Bagging: 0.9841966637401229
Random forest: 1.0
AdaBoost: 0.8700614574187884
Gradient Boosting: 0.8928884986830553
XGBoost: 1.0

Validation Performance:

Bagging: 0.7909836065573771
Random forest: 0.7704918032786885
AdaBoost: 0.8237704918032787
Gradient Boosting: 0.8114754098360656
XGBoost: 0.8688524590163934
```

Baseline model (Oversampled)

```
Training Performance:

Bagging: 0.9983190452176837
Random forest: 1.0
AdaBoost: 0.9724323415700118
Gradient Boosting: 0.9860480753067743
XGBoost: 1.0

Validation Performance:

Bagging: 0.8032786885245902
Random forest: 0.8524590163934426
AdaBoost: 0.8647540983606558
Gradient Boosting: 0.9057377049180327
XGBoost: 0.889344262295082
```

Baseline model (Undersampled)

```
Training Performance:

Bagging: 0.9920983318700615
Random forest: 1.0
AdaBoost: 0.9631255487269534
Gradient Boosting: 0.9850746268656716
XGBoost: 1.0

Validation Performance:

Bagging: 0.8975409836065574
Random forest: 0.9385245901639344
AdaBoost: 0.9467213114754098
Gradient Boosting: 0.9631147540983607
XGBoost: 0.9590163934426229
```

Model building

Data Sampling Techniques

- **Undersampled Data:**
  - Reduced the majority class size to balance the dataset.
  - Advantage: Speeds up training and avoids bias toward the majority class.
  - Limitation: Loss of valuable data from the majority class.
- **Original Data:**
  - Used the dataset as-is, without any balancing.
  - Advantage: Preserves the original distribution of data.
  - Limitation: Imbalance can lead to poor recall for the minority class.
- **Oversampled Data:**
  - Used **SMOTE (Synthetic Minority Oversampling Technique)** to create synthetic samples for the minority class.
  - Advantage: Balances the dataset while retaining the full dataset information.
  - Limitation: Synthetic data might introduce noise.

Hyperparameter Tuning

- **Purpose:**
  To optimize the performance of models by finding the best combination of parameters.
- **Method:**
  - Used **RandomizedSearchCV** for an efficient search of hyperparameters.
  - Evaluation metric: **Recall** (aligned with the business goal of minimizing false negatives).
- **Parameters Tuned:**
  - **Gradient Boosting:** Learning rate, number of estimators, maximum depth, and subsample.
  - **XGBoost:** Learning rate, number of estimators, gamma, subsample, and scale_pos_weight.
  - **AdaBoost:** Number of estimators and base learner depth.

Model building

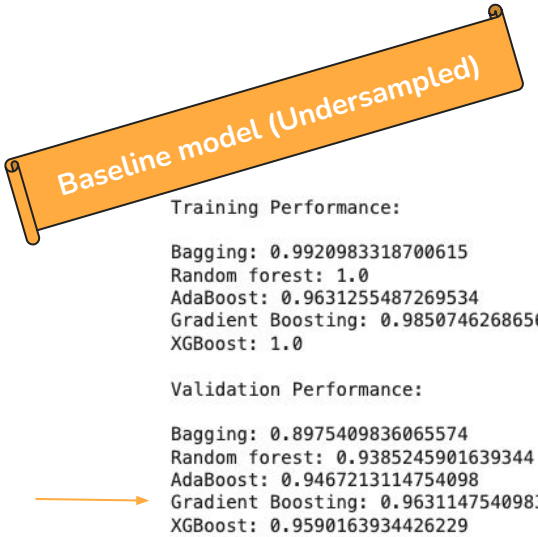
Reflections on Baseline and Tuned Models

Baseline Models Show Strong Performance

- During the modeling phase, the **Baseline Model (Undersampled)** using **Gradient Boosting** achieved remarkable results, particularly a validation recall of **0.963**, demonstrating its ability to meet the objective of identifying attrited customers effectively.
- This raised a critical question: **Do we need hyperparameter tuning if baseline models already perform this well?**

Rationale for Continuing with All Steps

- Despite the strong baseline results, we decided to proceed with hyperparameter tuning and testing advanced techniques for several reasons:
  - **Comprehensive Learning:** The process of tuning models helps identify the true potential of each technique and deepens our understanding of the data and algorithms.
  - **Comparative Analysis:** To ensure that the baseline performance wasn't a coincidence, we wanted to compare it rigorously with tuned models.
  - **Business Confidence:** A systematic approach instills greater confidence in the chosen solution for stakeholders.



Balancing Simplicity and Optimization

While the **Baseline Model (Undersampled)** showed strong performance, the subsequent exploration of oversampling and hyperparameter tuning offered valuable insights into potential trade-offs between complexity and performance.

Model Performance Summary/Comparison

Key Metrics and Insights:

Focus on Recall: Recall was prioritized as the key metric to ensure effective identification of at-risk customers. Models were evaluated on both training and validation datasets to compare their performance.

Validation Results:

- **Gradient Boosting (Baseline, Undersampled):** Achieved the highest Recall (0.963) on the validation set, demonstrating its strength in capturing at-risk customers.
- **Gradient Boosting (Tuned, Undersampled):** Delivered a strong Recall (0.947) while maintaining a good balance with other metrics.
- **AdaBoost (Tuned, Undersampled):** Showed competitive Recall (0.939) but slightly lower Precision, indicating more false positives.
- **Gradient Boosting (Original):** Recall (0.807) was lower compared to undersampled models, but Precision remained high (0.956).

Training performance comparison

Model	Accuracy	Recall	Precision	F1-score
Gradient Boosting (Tuned, Undersampled)	0.977	0.986	0.968	0.977
Gradient Boosting (Original)	0.976	0.888	0.960	0.923
AdaBoost (Tuned, Undersampled)	0.958	0.967	0.950	0.958
AdaBoost (Original)	0.967	0.851	0.936	0.891
Gradient Boosting (Baseline, Undersampled)	0.977	0.985	0.969	0.977

Validation performance comparison

Model	Accuracy	Recall	Precision	F1-score
Gradient Boosting (Tuned, Undersampled)	0.946	0.947	0.770	0.849
Gradient Boosting (Original)	0.963	0.807	0.956	0.876
AdaBoost (Tuned, Undersampled)	0.941	0.939	0.756	0.837
AdaBoost (Original)	0.956	0.795	0.919	0.853
Gradient Boosting (Baseline, Undersampled)	0.955	0.963	0.797	0.872

Model Performance Summary/Comparison

Training Results:

- Most models, including the baseline Gradient Boosting (Undersampled) and tuned versions, achieved high Recall values (>0.98), indicating effective learning from the training data.
- AdaBoost models also performed well on Recall but slightly lagged behind Gradient Boosting models.

Performance Highlights:

- **Gradient Boosting Baseline (Undersampled):** Demonstrated the best overall Recall on validation data (0.963), making it a strong choice for minimizing missed churn cases.
- **Precision vs. Recall Trade-off:** Gradient Boosting (Original) provided the highest Precision (0.956) but sacrificed Recall, which is less favorable for churn prediction.

Training performance comparison

Model	Accuracy	Recall	Precision	F1-score
Gradient Boosting (Tuned, Undersampled)	0.977	0.986	0.968	0.977
Gradient Boosting (Original)	0.976	0.888	0.960	0.923
AdaBoost (Tuned, Undersampled)	0.958	0.967	0.950	0.958
AdaBoost (Original)	0.967	0.851	0.936	0.891
Gradient Boosting (Baseline, Undersampled)	0.977	0.985	0.969	0.977

Validation performance comparison

Model	Accuracy	Recall	Precision	F1-score
Gradient Boosting (Tuned, Undersampled)	0.946	0.947	0.770	0.849
Gradient Boosting (Original)	0.963	0.807	0.956	0.876
AdaBoost (Tuned, Undersampled)	0.941	0.939	0.756	0.837
AdaBoost (Original)	0.956	0.795	0.919	0.853
Gradient Boosting (Baseline, Undersampled)	0.955	0.963	0.797	0.872

Model building - Final Model Selection

Chosen Model:

- Baseline Gradient Boosting trained with undersampled data.

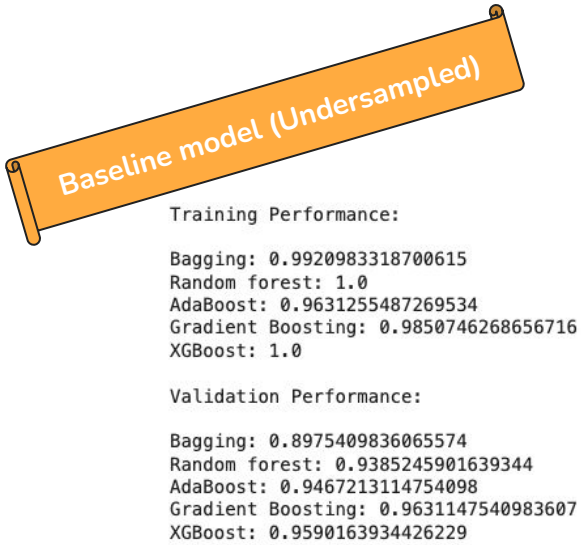
  Reason: Achieved the highest Recall (96.3%), aligning with the business objective of identifying as many attrited customers as possible.

  Recall: Gradient Boosting prioritized identifying at-risk customers with minimal false negatives.

  Simplicity: Baseline Gradient Boosting offered competitive performance with minimal tuning.

Validation Metrics:

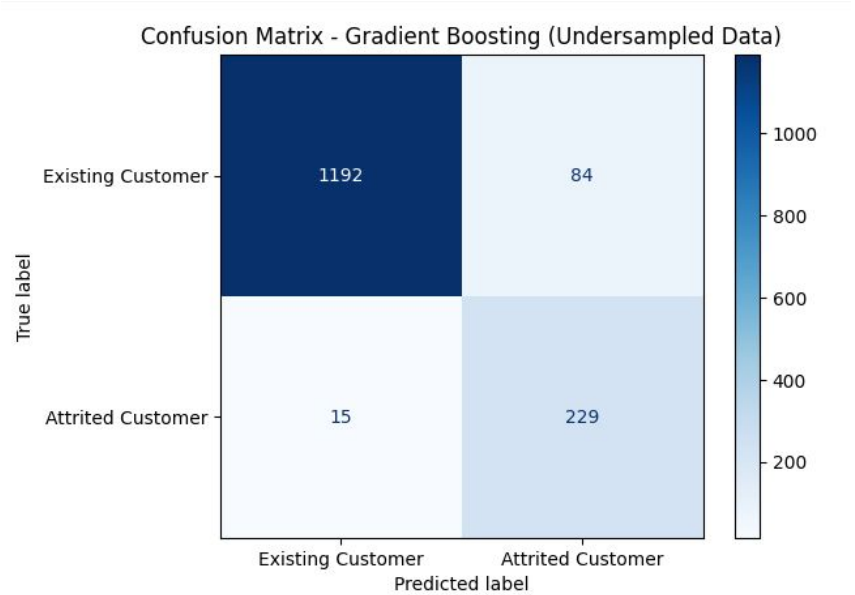
- **Recall**: 96.3%
- **Precision**: 79,7%
- **F1-Score**: 87.2%



Model building

Test Set Performance

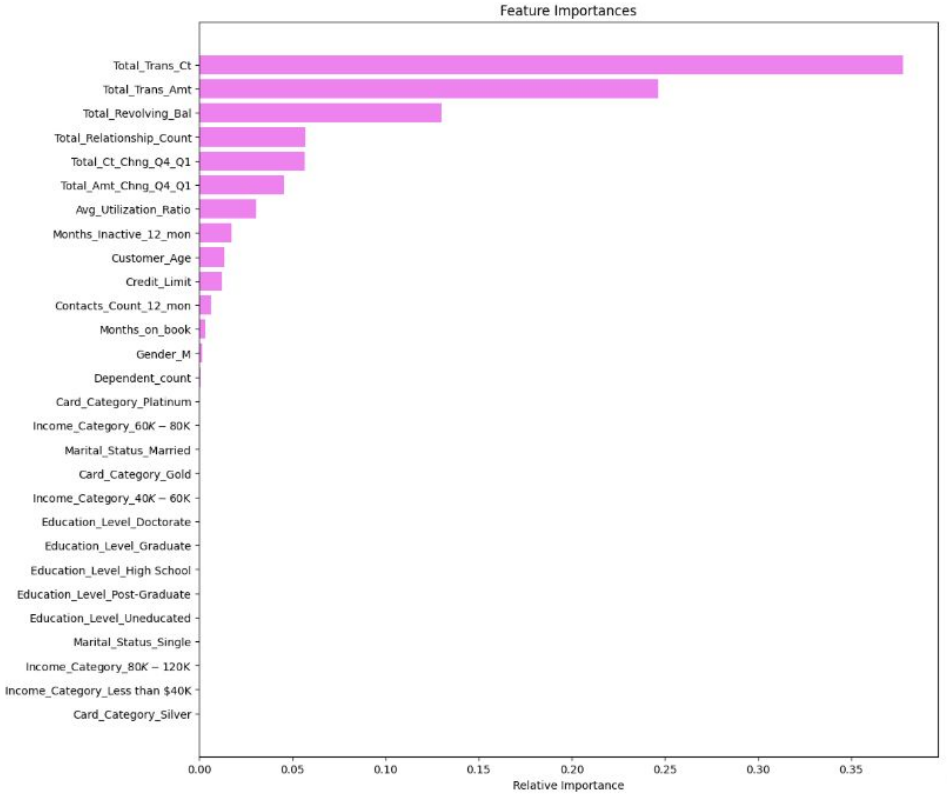
- Evaluated the final model on the unseen test set.
- **Test Metrics:**
  - Accuracy: 93.5%
  - Recall: 93.9%
  - Precision: 73.2%
  - F1-Score: 82.2%
- **Confusion Matrix Analysis:**
  - True Positives: 229 attrited customers correctly identified.
  - False Negatives: 15 attrited customers missed.
  - True Negatives: 1,192 existing customers correctly identified.
  - False Positives: 84 existing customers incorrectly flagged as attrited.
- **Conclusion:** The model effectively prioritized recall while keeping false positives relatively low.



Model building

Feature Importance

- **Top Features:**
  - **Total_Trans_Ct:** The total number of transactions was the most important predictor, likely indicating customer engagement and activity.
  - **Total_Trans_Amt:** The total transaction amount also heavily influenced attrition predictions, reflecting spending behavior.
  - **Total_Revolving_Bal:** A high revolving balance might indicate credit dependency or lack of financial engagement.
  - Other important features: Total_Relationship_Count, Total_Ct_Chng_Q4_Q1, and Avg_Utilization_Ratio.
- **Business Insights:**
  - Customers with low transaction counts or amounts may need proactive engagement to reduce churn risk.
  - Monitoring relationship depth and changes in behavior can serve as early warning signals.



Reflecting on Model Complexity vs. Simplicity

While performing extensive hyperparameter tuning and exploring complex variations of models can yield improvements, it's crucial to evaluate whether the added complexity justifies the results.

In this case:

- The **baseline Gradient Boosting model** (trained on undersampled data) achieved competitive performance, particularly excelling in Recall (96.3%), which aligns with the business objective.
- Tuned versions of models, while slightly improving certain metrics (e.g., Precision or F1-score), required additional computational time and effort.

Key Insight:
Sometimes, simpler models can deliver results that are both effective and efficient. When the baseline model performs well and meets business needs, it may not always be necessary to pursue more complex approaches. This highlights the importance of balancing practicality, interpretability, and performance when selecting a model.

APPENDIX

Lesson Learned

What Worked Well:

1. **Boosting Algorithms:**
   - Gradient Boosting and XGBoost handled the imbalanced dataset effectively, achieving high Recall and stable performance across metrics.
   - Their robustness in identifying at-risk customers highlighted their suitability for churn prediction tasks.
2. **Undersampled Data:**
   - Using undersampling simplified the modeling process while achieving high Recall, reducing the likelihood of missing critical churn cases.

Challenges Encountered:

1. **Imbalanced Data:**
   - Required careful handling, including experimentation with various sampling techniques (undersampling, oversampling) to achieve the right balance.
   - Prolonged the workflow due to iterative tuning and evaluation of data distribution strategies.
2. **Computational Costs:**
   - Hyperparameter tuning, especially for complex models like XGBoost, was computationally expensive and time-consuming.
   - Highlighted the need to evaluate whether simpler models could achieve comparable results with lower resource requirements.

Lesson Learned

Key Takeaways for Future Projects:

Efficient Problem-Solving:

Time and resources are valuable. Starting with simpler approaches and adding complexity only when necessary is a smarter, more sustainable way to work

Simplicity Over Complexity:

Sometimes, simpler models can achieve results that are just as effective as complex ones. The real skill lies in knowing when added complexity is worth the effort—and when it's not.

Commit to Learning:

Every project offers lessons. The goal isn't just to build models but to refine how we approach problems and apply what we learn to the next challenge.

Real-World Relevance:

A model is only as good as the decisions it helps to make. It's important to prioritize models that not only perform well but also align with practical business needs and are easy to explain.

Great Learning

Happy Learning !