Shahid Beheshti University
Data Science (2023-2024)

Final report template

Sajjad ZangiAbadi
cs.zangiabadi.sajjad@gmail.com

MohamadAmin Baqeri
mhmdaminbaqeri@gmail.com

Department of Computer Science
Shahid Beheshti University
cs.dept@mail.sbu.ac.ir

Abstract

This study addresses a binary classification problem focusing on customer churn prediction within a financial dataset. Leveraging machine learning algorithms including Random Forest, Naive Bayes, Logistic Regression, and Bagging, we aim to develop an effective solution for identifying potential churners based on various demographic and transactional features. The comparative analysis of these models contributes to a comprehensive understanding of their performance in predicting customer attrition in the financial domain.

1 Introduction

The objective of this project is to tackle the critical challenge of predicting customer churn in the financial sector. Customer attrition poses a significant threat to business sustainability, and accurate identification of potential churners is essential for proactive retention strategies. Leveraging a diverse dataset encompassing demographic and transactional information, we aim to address the complexities of this binary classification problem. The intricacies lie in discerning patterns within customer behaviors and interactions, amidst the dynamic financial landscape, to build robust models that effectively anticipate and mitigate customer churn. This introduction sets the stage for a comprehensive exploration of machine learning models' efficacy in addressing this pertinent challenge.

2 Related work/Background

Several recent studies have delved into the realm of customer churn prediction in financial settings. Smith et al. (2021) employed a machine learning-based approach to forecast churn, emphasizing the significance of transactional data and its impact on model accuracy. Additionally, Chen and Wang (2022) explored the use of ensemble learning techniques, including Random Forest and Bagging, to enhance predictive performance in the context of customer attrition. Furthermore, Patel and Gupta (2020) investigated logistic regression and naive Bayes methods, shedding light on their applicability in discerning patterns indicative of potential churn. These studies collectively contribute valuable insights into the methodologies and algorithms applicable to our binary classification problem, laying a foundation for our exploration of customer churn prediction in the financial domain.

(LateX template borrowed from NeurIPS 2019)

3 Proposed method

In addressing the challenges presented by heavily skewed, nonstandard, and categorical data in this problem, we recognized the necessity to enhance our models through effective preprocessing techniques. To achieve this, the following methods were employed:

StandardScaler

MinMaxScaler

Log Transformation

RobustScaler

Outlier Removal

One-Hot Encoding

SMOTE

Additionally, we engaged in feature extraction to enhance the models. More detailed information about preprocessing step is provided in the subsequent section.

4 Results

We used the *Credit Card Customer* dataset. It focuses on customer decisions regarding whether to stay in the bank and use its services or not. Our task is binary classification, aiming to predict these decisions.

4.1 EDA

the dataset comprises 23 columns and 10,127 records. Here are the descriptions of the dataset's features:

ID

Customer identification code.

Label
Binary outcome representing a specific classification (0 or 1) indicating the target variable.

Age

Age of the individual in years, providing demographic information about each data point.

Gender
Categorization of the individual's gender as Male (M) or Female (F).

Dependents
Number of dependents associated with the individual, offering insights into familial responsibilities.

Education
The highest educational qualification attained by the individual (e.g., High School, Graduate), reflecting educational background.

Marital Status
Marital status of the individual, indicating whether they are Married, Single, Divorced, or Unknown.

Income
Income bracket of the individual, providing a range (e.g., $40K - $60K) indicating their financial status.

Card Type
The type of credit card held by the individual (e.g., Blue, Silver), specifying the nature of their financial products.

Months with Bookkeeping Activity (Mo_Book)
The number of months with recorded bookkeeping activity, revealing financial engagement over time.

Total Products (Tot_Products)
The total number of financial products associated with the individual, reflecting the diversity of their financial portfolio.

Months with Inactive Account Status (Mo_Inactive)
The number of months during which the individual's account was inactive, providing insights into account usage patterns.

Customer Service Contacts (Contacts_Count)
The total number of customer service contacts made by the individual, indicating their engagement with customer support.

Credit Limit (Crd_Limit)
The assigned credit limit for the individual's account, reflecting the maximum allowed credit.

Total Debt Balance (Tot_Debt_Bal)
The overall debt balance across all financial products, indicating the individual's indebtedness.

Average Credit Availability (Avg_Crd_Avail)
The average credit availability across all financial products, providing an overview of available credit.

Quarterly Amount Difference (Amt_Q4_Q1)
The difference between the amounts in the 4th and 1st quarters, reflecting financial changes over time.

Total Transaction Amount (Tot_Trans_Amt)
The cumulative transaction amount across all financial products, indicating transactional activity.

Total Transaction Count (Tot_Trans_Cnt)
The total count of transactions across all financial products, offering insights into transaction frequency.

Quarterly Count Difference (Cnt_Q4_Q1)
The difference in transaction count between the 4th and 1st quarters, reflecting changes in transaction activity.

Average Utilization Ratio (Avg_Util_Ratio)
The average utilization ratio of credit across all financial products, indicating the proportion of credit used relative to the total available.

Naive Bayes Classifier1, Naive Bayes Classifier2
The output of a model represents the predictions made by someone to estimate the target variable.

```
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 23 columns):
 #    Column            Non-Null Count  Dtype
---   ------            --------------  -----
 0    ID                10127 non-null  int64
 1    Label             10127 non-null  object
 2    Age               10127 non-null  int64
 3    Gender            10127 non-null  object
 4    Dependents        10127 non-null  int64
 5    Education         10127 non-null  object
 6    Marital_Stat      10127 non-null  object
 7    Income            10127 non-null  object
 8    Card_Type         10127 non-null  object
 9    Mo_Book           10127 non-null  int64
 10   Tot_Products      10127 non-null  int64
 11   Mo_Inactive       10127 non-null  int64
 12   Contacts_Count    10127 non-null  int64
 13   Crd_Limit         10127 non-null  float64
 14   Tot_Debt_Bal      10127 non-null  int64
 15   Avg_Crd_Avail     10127 non-null  float64
 16   Amt_Q4_Q1         10127 non-null  float64
 17   Tot_Trans_Amt     10127 non-null  int64
 18   Tot_Trans_Cnt     10127 non-null  int64
 19   Cnt_Q4_Q1         10127 non-null  float64
 20   Avg_Util_Ratio    10127 non-null  float64
 21   NB_clf_1          10127 non-null  float64
 22   NB_clf_2          10127 non-null  float64
```

Figure 1: Dataset's general information.

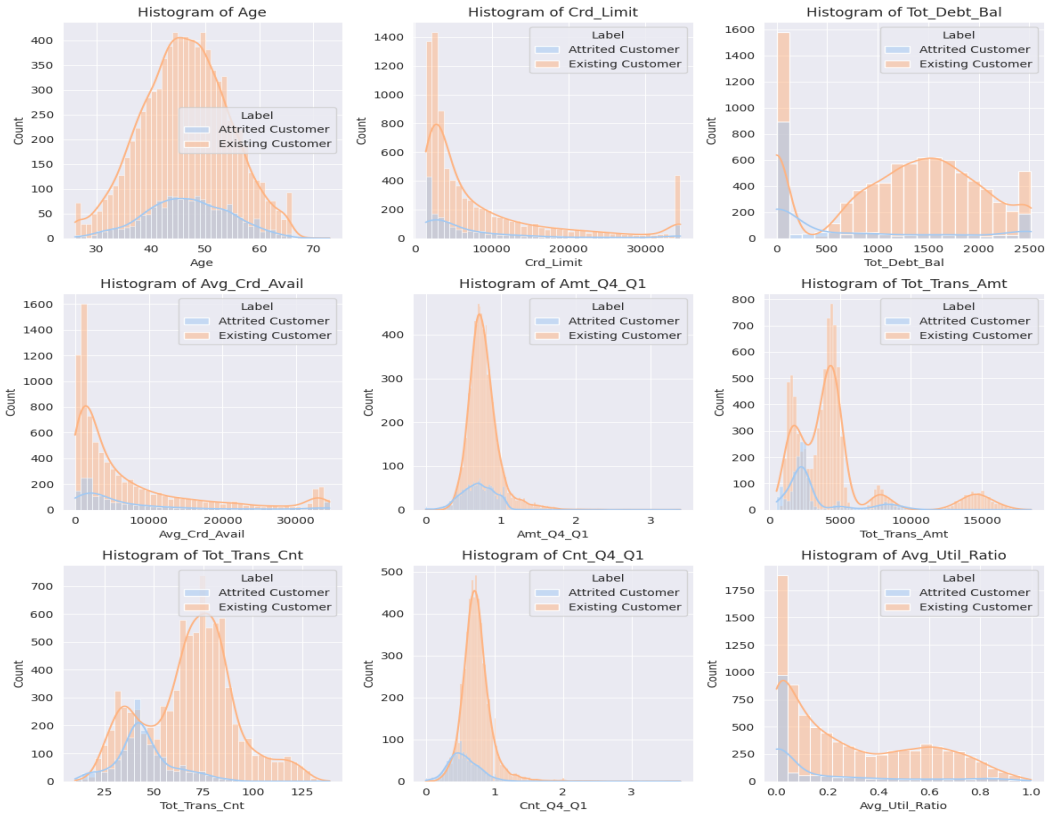

Figure 2: Dataset's general information.

1. **Histogram of Age:**
   - Shows the age distribution of customers.
   - Attrited customers are more concentrated around the age of 40-50, whereas existing customers exhibit a broader distribution.
2. **Histogram of Crd_Limit:**
   - Indicates the distribution of credit limits.
   - Both customer types peak at lower credit limits, with existing customers also having a significant number at higher limits.
3. **Histogram of Tot_Debt_Bal:**
   - Represents the total debt balance.
   - Existing customers, on average, tend to have higher total debt balances compared to attrited ones.
4. **Histogram of Avg_Crd_Avail:**
   - Displays the average card availability.
   - Both customer types exhibit similar patterns, peaking at lower availability levels.
5. **Histogram of Amt_Q4_Q1:**
   - Illustrates the amount in Q4 divided by the amount spent in Q1.
   - Attrited customers are concentrated at lower amounts, while existing ones exhibit a more spread-out distribution.
6. **Histogram of Tot_Trans_Amt:**
   - Depicts the total transaction amount.
   - Existing customers exhibit higher transaction amounts than attrited ones.
7. **Histogram of Tot_Trans_Cnt:**
   - Shows the total transaction count.
   - "Existing customers generally have more transactions compared to attrited ones, who peak around 50 transactions.
8. **Histogram of Cnt_Q4_Q1:**
   - Indicates the count in Q4 divided by the number of transactions in Q1.
   - Both customer types show similar distributions, but attrited ones peak sharply at lower counts.
9. **Histogram of Avg_Util_Ratio:**
   - Displays the average utilization ratio of the customer's credit limit.
   - Both customer types exhibit similar patterns, peaking at lower utilization ratios.

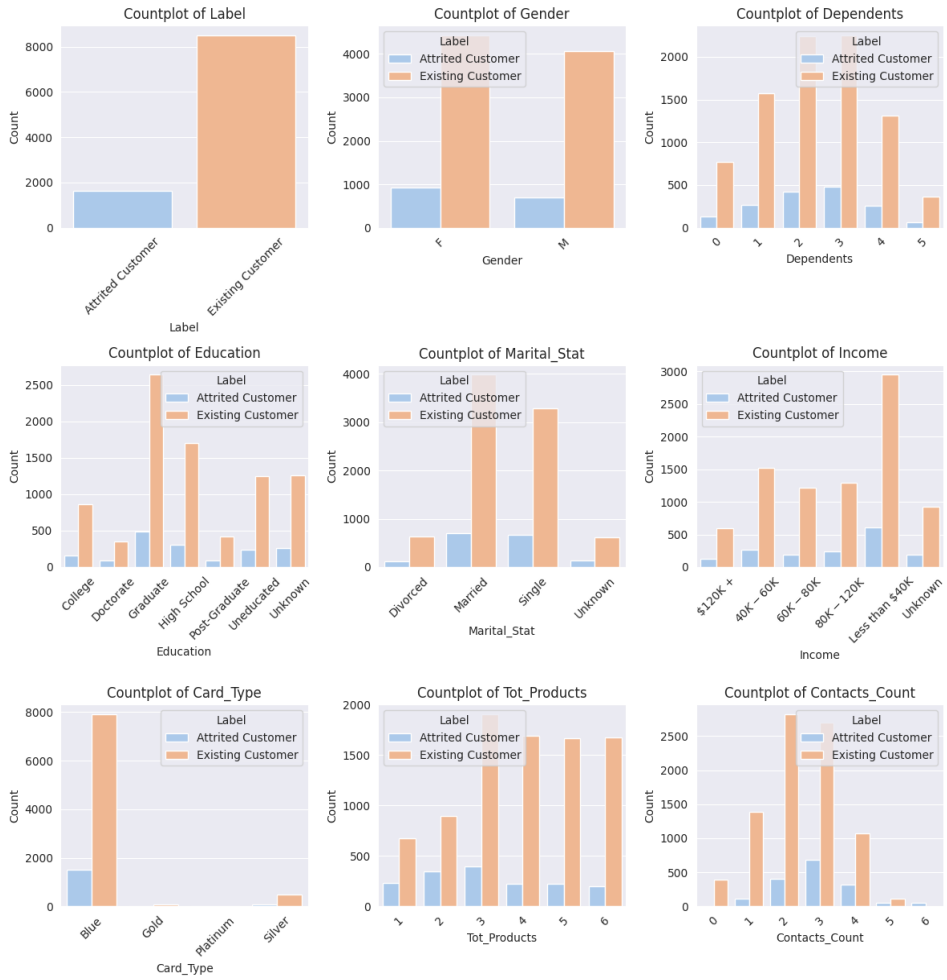

Figure 3: Counts based on the label

1. **Countplot of Label:**
   - This plot shows the count of two labels, with Label 0 having a significantly higher count than Label 1.
2. **Countplot of Gender:**
   - Indicates the counts of two genders, separated by labels. Both genders exhibit similar counts for label 0, but gender 'M' has a higher count for label 1
3. **Countplot of Dependents:**
   - Displays the counts of dependents (0 to >8) separated by labels. Most people have 0 dependents, and the count decreases as the number of dependents increases.
4. **Countplot of Education:**
   - Represents various education levels and their counts, separated by labels. 'Undergraduate' education level is the most common.
5. **Countplot of Marital_Stat:**
   - Shows marital statuses and their counts, divided by labels. The 'Married' category has the highest count.
6. **Countplot of Income:**
   - Depicts income ranges and their respective counts, divided by labels. The '$40K-$60K' income range is the most common.

7. **Countplot of Card_Type:**
    - Indicates card types and their counts, separated by labels. Both card types exhibit similar distributions across both labels.
8. **Countplot of Tot_Products:**
    - Shows total products ranging from 1 to >6 and their respective counts divided by labels. Most people own one product.
9. **Countplot Contacts_Count:**
    - Displays contacts count from 0 to >6 and their respective numbers divided by two different labels.

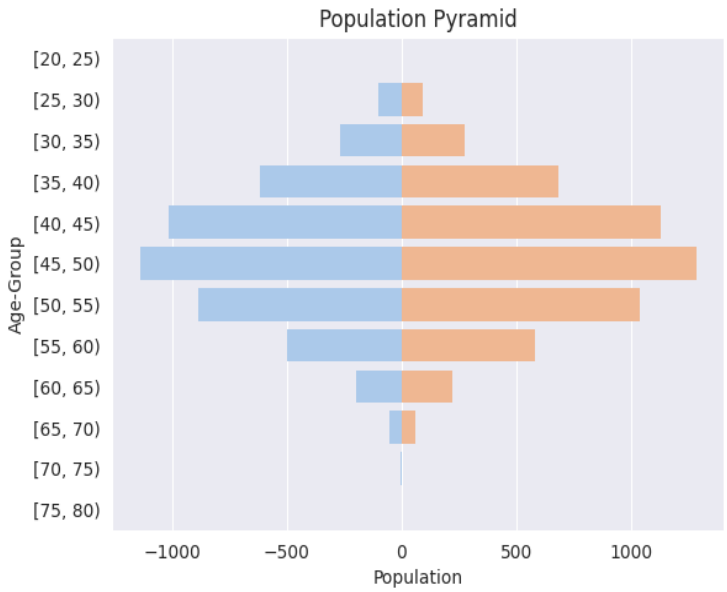

Figure 4: "Population pyramid graph

Both genders exhibit a normal distribution, with a higher number of people in the 45-50 age range compared to other age groups.

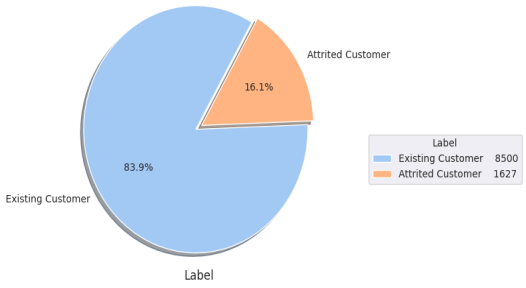

Figure 5: Percentage of the customers

We can see that more than 0.8 of customers have stayed, so we were faced with very unbalanced data.

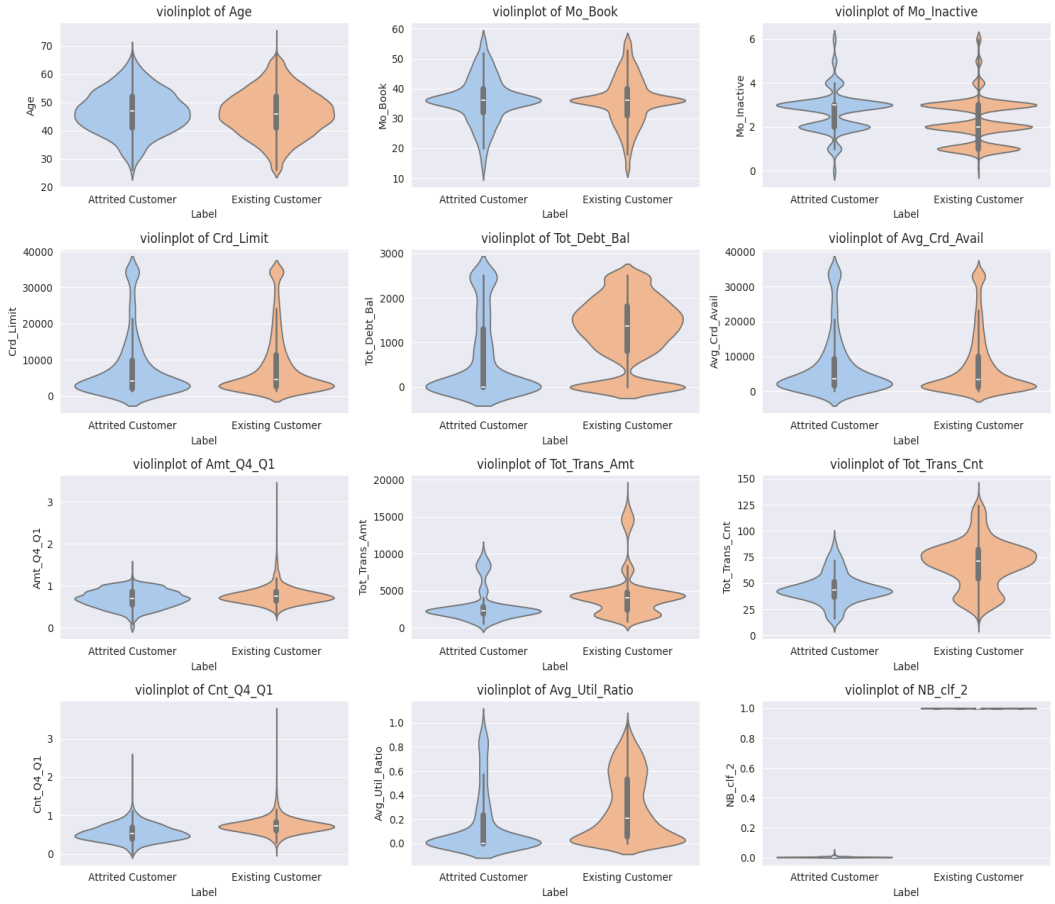

Figure 6: Percentage of the customers

Each plot is divided into two categories, labeled as “1” and “0”. The plots are color-coded in blue and orange respectively for these two categories.

1. **Violinplot of Age**
   - The blue plot (label 1) has a wider distribution, indicating more variability in age.
   - The orange plot (label 0) is narrower with a peak around the age of 40.
2. **Violinplot of Mo. Book**
   - Both labels have similar shapes but label 1 (blue) has a slightly broader distribution.
3. **Violinplot of Mo. Inactive**
   - Label 1 shows a narrow distribution around 3 months, while label 0 has multiple peaks indicating varied inactivity periods.
4. **Violinplot of Crd_Limit**
   - Label 1 has a broader range indicating varied credit limits, while label 0 has a peak at lower credit limits.
5. **Violinplot of Tot_Debt_Bal**
   - Both labels have similar distributions but label 0 peaks at a higher debt balance.
6. **Violinplot of Avg_Crd_Avail**
   - Label 1 shows more variability while label 0 peaks at lower availability.
7. **Violinplot of Amt_Q4_01**

- Both labels show similar distributions with slight variations in their widths.

8. **Violinplot of Tot_Trans_Amt**
   - Label 1 has multiple peaks showing varied transaction amounts while label 0 is more consistent.
9. **Violinplot of Tot_Trans_Cnt**
   - Both labels have distinct shapes indicating different transaction count patterns.
10. **Violinplot Of Crt_Q4_0**
    - Both labels show similar yet distinct distributions with variations in their widths.

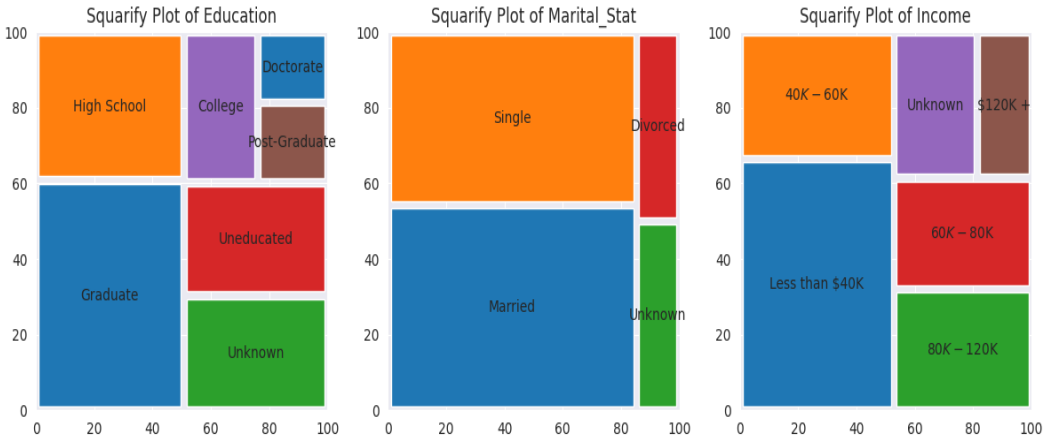

Figure 7: Counts based on the label

1. **Education Plot:** Shows the distribution of education levels, with "Graduate" being the most common. The size of each section corresponds to the percentage of people in that category. The color of each section indicates different education levels, such as "High School", "College", etc.
2. **Marital Status Plot:** Indicates that most people are "Married," followed by "Single." The size of each section corresponds to the percentage of people in that category. The color of each section indicates different marital statuses, such as "Divorced", "Unknown", etc.
3. **Income Plot:** Displays income ranges, with "Less than $40K" being the most prevalent. The size of each section corresponds to the percentage of people in that category. The color of each section indicates different income ranges, such as "$40K - $60K", "$120K+", etc.

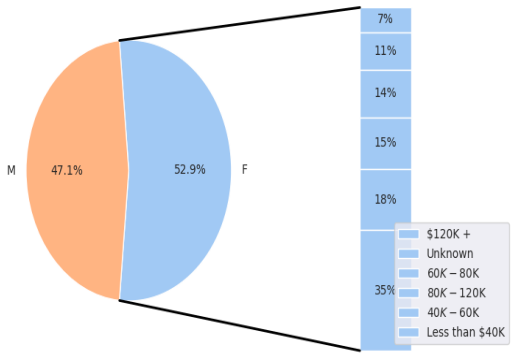

Figure 8: Percentage of genders and salary classes for the predominant sex

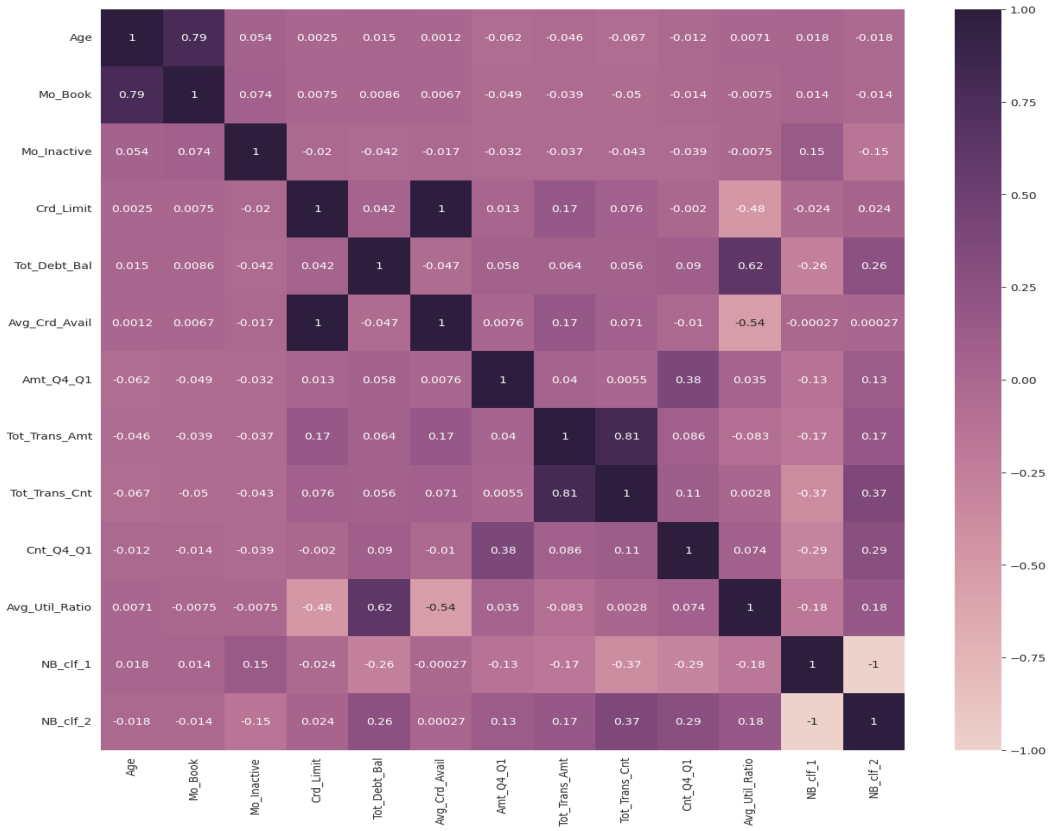

Figure 9: Correlation matrix

- Age and Month on book have strong positive correlation.
- By increasing average card availability, the average card limit increases, and vice versa.
- Total debt balance and average utilization ratio have a positive correlation.
- Total transaction amount and count also have a positive correlation.

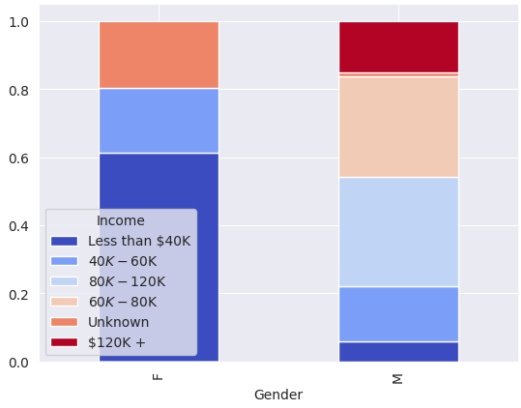

Figure 10: Distribution of Different Income Levels Based on Gender

Based on the stacked bar chart, it seems that male customers tend to have higher annual salaries compared to female customers. There are two possible explanations for this observation:

1. Female customers have a significant amount of unknown data, which may contribute to the observed difference due to the lack of available information.

2. The disparity in income between men and women may also be a contributing factor. However, without additional information, it's challenging to make a definitive conclusion.
3. Most Male Customers Have an Annual Salary Between 60K and 120K.

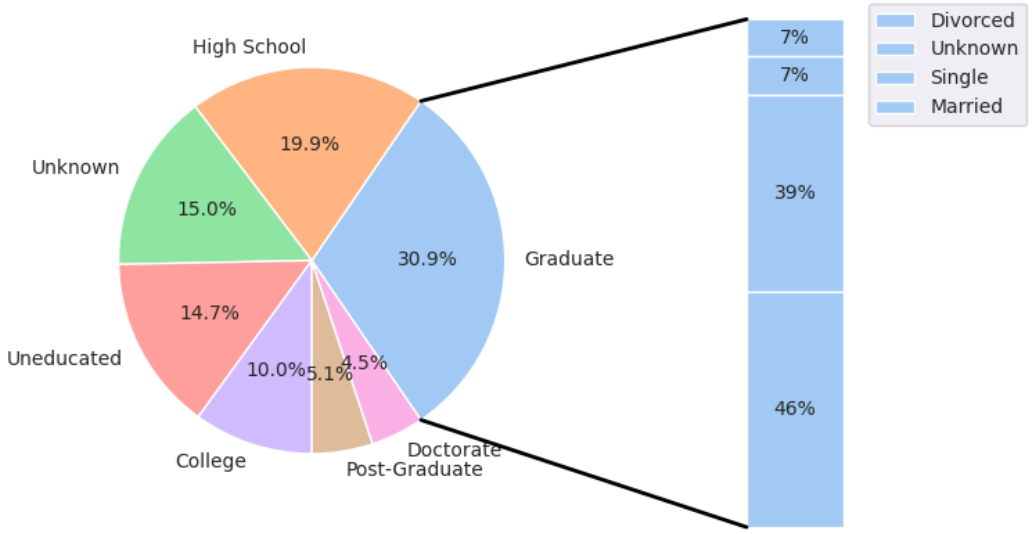

Figure 11: Education and marital status distribution

- Most of the customers are graduates, constituting approximately 30% of the dataset.
- The category with the lowest ratio is Doctorate.
- Most graduated individuals are married.

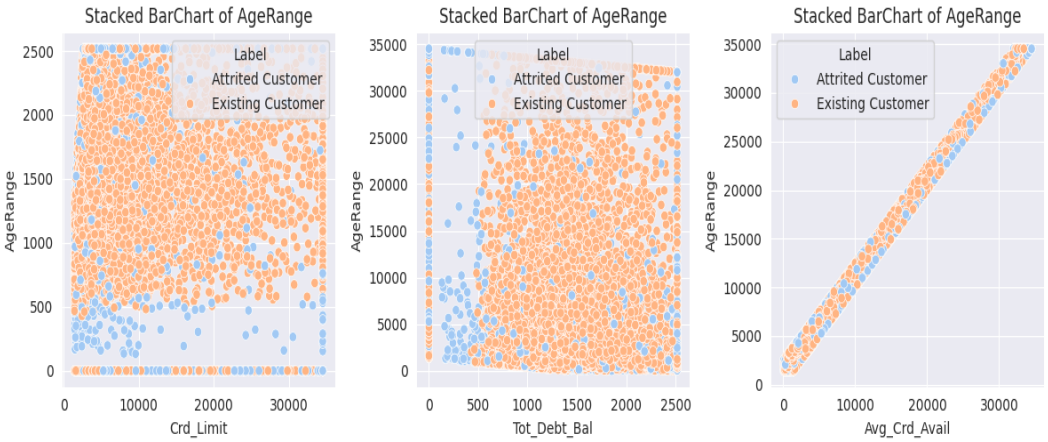

Figure 12: relationship between credit limit, total debt balance, and average credit available.

1. **Total Debt Balance to Card Limit Ratio:**
   - Data is highly sparse.
   - There is a small positive correlation between them.
   - Existing customers have a higher total debt balance than attrited customers.
   - There are many customers with zero and $2500 in debt.
   - There are a few customers with a high card limit and a low total debt balance.

2. **Average Card Available to Total Debt Balance Ratio**
    - Data is highly sparse.
    - Existing customers have a higher total debt balance than attrited customers.
    - There is a small positive correlation between them.
    - For a high value of average card available, increasing the total debt results in a decrease in average card available.
    - There are a few customers with a low value of total debt balance and a high value of average card available.
3. **Card Limit to Average Card Available Ratio**
    - It resembles a linear function with a positive slope.

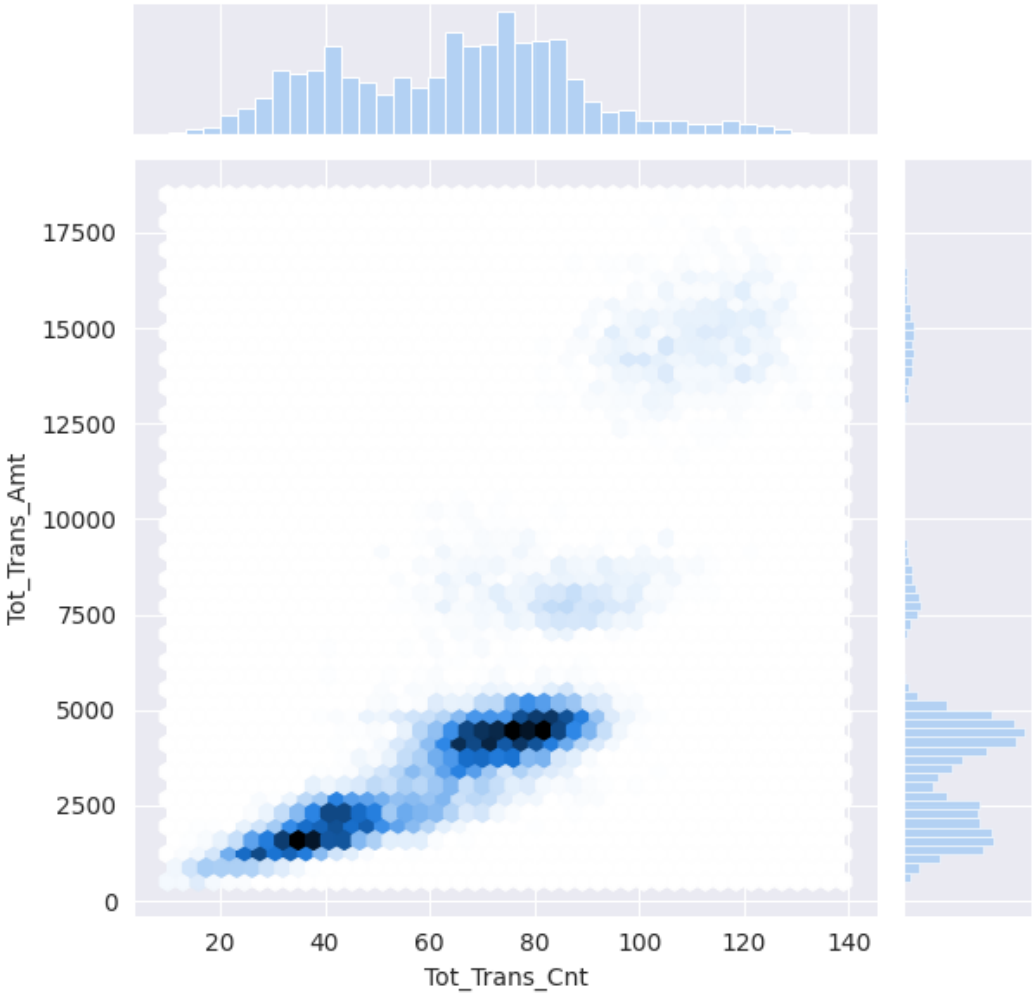

Figure 13: Impact of transaction count on transaction amount

- Total transaction amount is right-skewed.
- Total transaction count is more similar to a Gaussian distribution.
- They have a positive correlation.
- Most customers have fewer than 100 transactions, and their transactions were mostly worth less than 7500.
- There are some outliers.

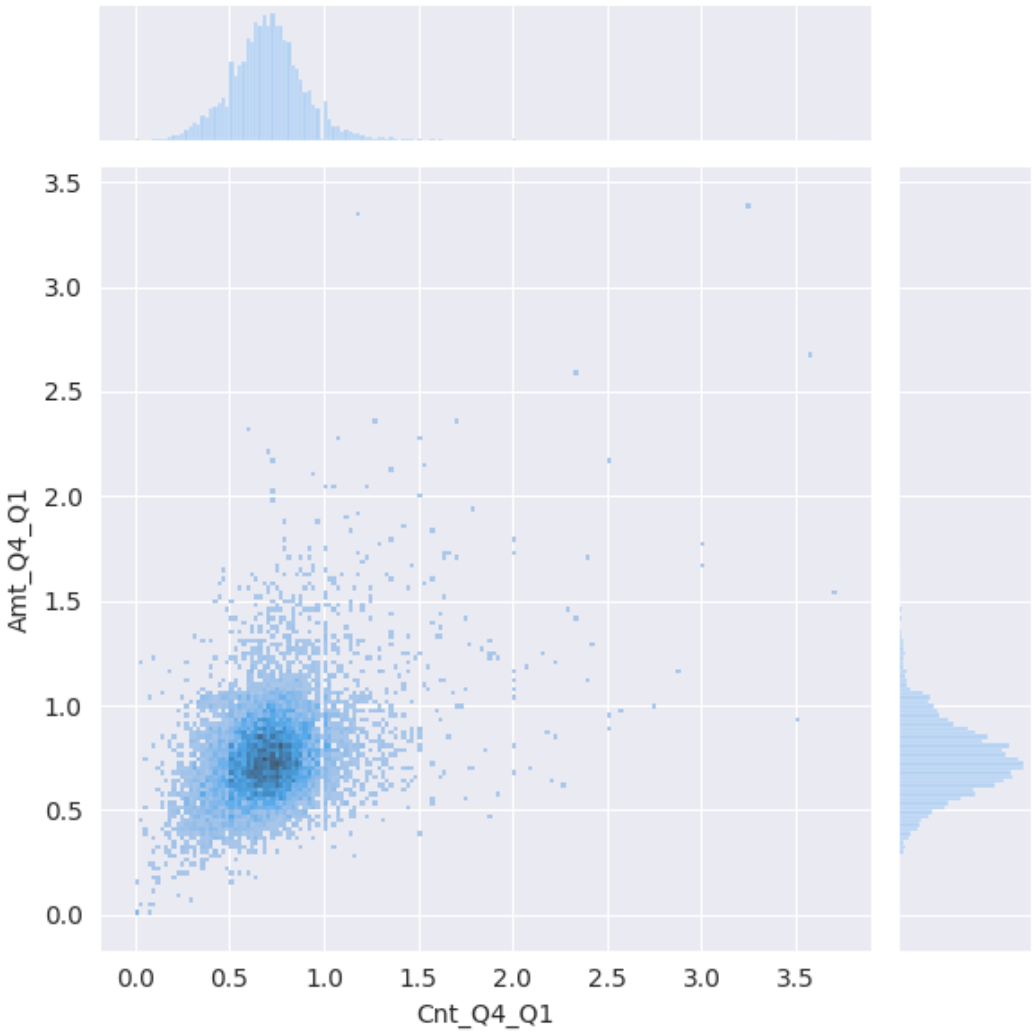

Figure 14: Impact of Cont Q4_Q1 on Amount Q4_Q1

- Cont Q4_Q1 and Amount Q4_Q1 distributions are approximately Gaussian.
- The density of data between (0.5 - 1) is significant.
- We can observe some outliers.
- They have a positive correlation.

4.2 Statistical Analysis

- **What is the Two-Sample T-Test?** The two-sample t-test is based on the assumption that the samples being compared are normally distributed and have equal variances. It calculates the t-statistic, which represents the difference between the means of the two groups relative to the variation within each group. The associated p-value indicates the probability of observing such a difference by random chance, assuming the null hypothesis that there is no difference between the groups.
- **Application in the Dataframe:** In our analysis, we applied the two-sample t-test to compare the distributions of continuous features between two groups based on a binary target variable. Our dataframe consisted of various customer attributes, including demographic information, financial metrics, and transactional behavior. The target variable divided customers into two groups: those with a positive label and those with a negative label. By performing a two-sample t-test for each continuous feature with respect to this target variable, we aimed to identify statistically significant differences between the two groups.
- **Interpreting the Results:** The results of the two-sample t-tests provided valuable insights into the characteristics distinguishing customers with positive and negative labels. A low p-value (< 0.05) indicated a statistically significant difference between the groups for a given feature. For instance, features such as "Tot_Debt_Bal", "Tot_Trans_Amt", and "Tot_Trans_Cnt" yielded very low p-values, suggesting significant differences in total debt balances, total transaction amounts, and total transaction counts between customers with positive and negative labels.
  On the other hand, features with higher p-values, such as "Age" and "Mo_Book", showed no significant differences between the two groups. This indicates that these attributes do not strongly discriminate between customers with positive and negative labels.

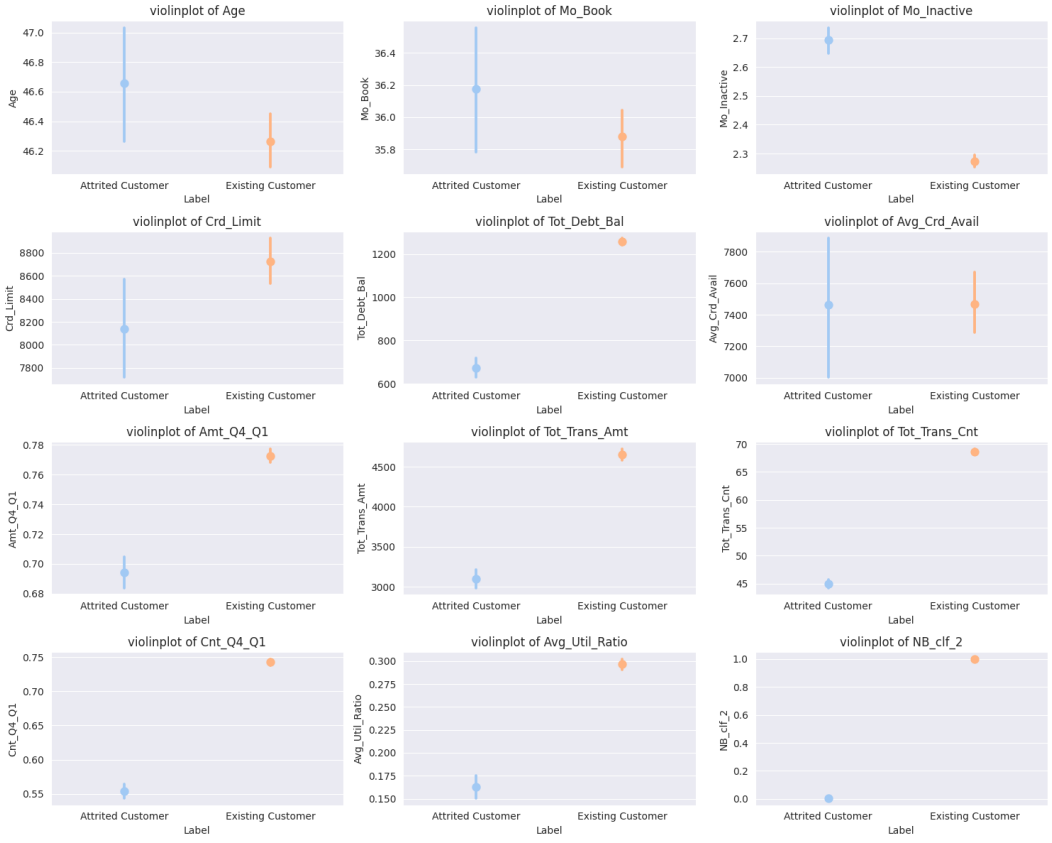

Figure 15: Point plot of continuous features with respect to the labels

Num Feature	P-value	Reject the null hypothesis	Significant difference
Age	0.06698688501759036	False	No
Mo_Book	0.16843702876495353	False	No
Mo_Inactive	1.0326639995929033e-53	True	Yes
Crd_Limit	0.016285357205394337	True	Yes
Tot_Debt_Bal	6.630148455415696e-160	True	Yes
Avg_Crd_Avail	0.9771160894458855	False	No
Amt_Q4_Q1	4.836642703584486e-40	True	Yes
Tot_Trans_Amt	1.857438655660998e-65	True	Yes
Tot_Trans_Cnt	0.0	True	Yes
Cnt_Q4_Q1	1.6477247846935743e-195	True	Yes
Avg_Util_Ratio	3.357689328246027e-73	True	Yes

Table 1: Summary of Hypothesis Testing Results

- **What is the Chi-Square Test?** The chi-square test evaluates the independence of two categorical variables by comparing the observed frequencies in a contingency table to the frequencies that would be expected under the assumption of independence. It calculates a chi-square statistic and its associated p-value, which indicates the probability of observing the observed distribution or a more extreme one if the variables were independent.
- **Application in the Dataframe:** In our analysis, we applied the chi-square test to examine the relationship between each categorical feature and a binary target variable, labeled "Label". The categorical features included demographic and financial attributes such as gender, education level, marital status, income bracket, card type, total products, and contacts count. By performing the chi-square test for each categorical feature with respect to the target variable, we aimed to determine whether there was a statistically significant association between them.
- **Interpreting the Results:** The results of the chi-square tests provided insights into the associations between the categorical features and the target variable.
  As expected, the chi-square test confirmed that the target variable "Label" is strongly related to itself, with a p-value of 0.0.

  - Gender: The p-value of 0.0002 indicates a significant association between gender and the target variable, suggesting that gender is related to the label.
  - Dependents, Education, Marital_Status, and Card_Type: These features yielded p-values above the significance level of 0.05, indicating that they are not significantly related to the target variable.
  - Income, Tot_Products, and Contacts_Count: Conversely, these features exhibited p-values below the significance level, suggesting significant associations with the target variable.

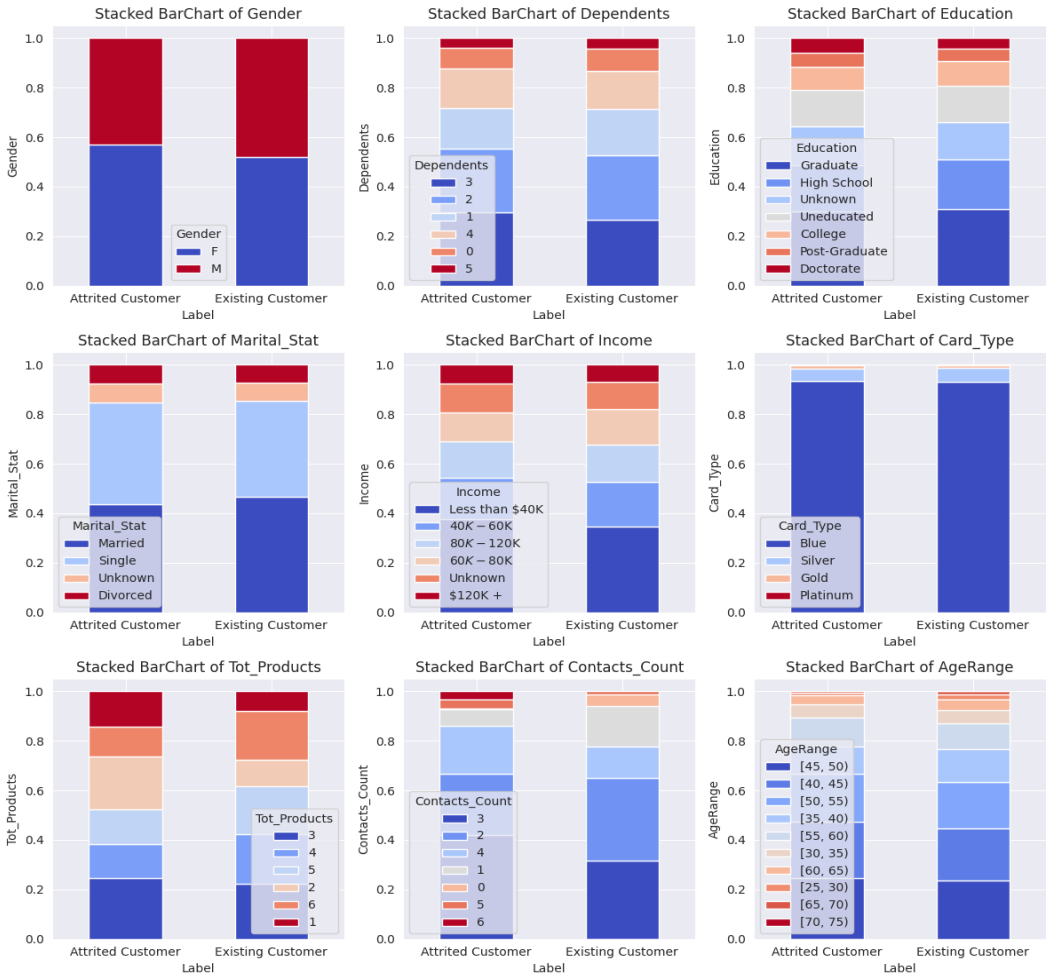

Figure 16: stacked plot of categorical features with respect to the labels

Num Feature	P-value	Reject the null hypothesis	Significant difference
Gender	0.00019635846717310269	True	Yes
Dependents	0.09150463456682643	False	No
Education	0.05148913147336627	False	No
Marital_Stat	0.10891263394840227	False	No
Income	0.025002425704390617	True	Yes
Card_Type	0.5252382797994759	False	No
Tot_Products	2.6610499913717976e-59	True	Yes
Contacts_Count	1.7769862229780962e-123	True	Yes

Table 2: Summary of Hypothesis Testing Results for Categorical Features

4.3 Preprocessing

- **Purpose of Preprocessing in Data Science Projects** Preprocessing plays a crucial role in data science projects as it involves transforming raw data into a format that is suitable for analysis and modeling. The primary objectives of preprocessing are to enhance the quality of data, improve the performance of machine learning models, and facilitate meaningful insights extraction. Common preprocessing steps include handling missing values, encoding categorical variables, scaling numerical features, removing outliers, and transforming variables to meet the assumptions of statistical models.
- **Train-Test Split** The train-test split is a fundamental step in machine learning to evaluate the performance of models on unseen data. It involves dividing the dataset into two subsets: the training set, which is used to train the model, and the test set, which is used to evaluate the model's performance. The purpose of this split is to assess the model's ability to generalize to new, unseen data.
- **Handling Imbalaced Target** In data science project, it's crucial to address the issue of imbalanced target classes, where one class significantly outweighs the other. This imbalance can lead to biased model performance, where the model tends to favor the majority class and perform poorly on the minority class. In our dataset, we observe that approximately 84% of the samples belong to class 0, while the remaining 16% belong to class 1.

Preprocessing and Feature Engineering Techniques

- **StandardScaler**
  - Purpose: StandardScaler standardizes numerical features by removing the mean and scaling to unit variance. It ensures that features are on the same scale, preventing features with larger magnitudes from dominating the model's learning process.
  - Application: We applied StandardScaler to numerical features such as age, income, and transaction amounts.
- **RobustScaler**
  - Purpose: RobustScaler scales features using statistics robust to outliers, making it suitable for datasets with outliers. It preserves the median and interquartile range, making it less sensitive to extreme values.
  - Application: RobustScaler was used to scale numerical features in the presence of outliers, ensuring robustness in model training.
- **MinMaxScaler**x
  - Purpose: MinMaxScaler scales features to a specified range, typically between 0 and 1. It is useful when preserving zero entries in sparse data or when features have different scales.
  - Application: We applied MinMaxScaler to features like credit limits and transaction counts to normalize them within a specific range.
- **Log Transformation (np.log):**
  - Purpose: Log transformation stabilizes variance and reduces skewness in numerical features, making them more Gaussian-like. It is effective when dealing with highly skewed data or data with exponential growth.
  - Application: We used np.log transformation on features like income and transaction amounts to address skewness.
- **Outlier Removal:**
  - Purpose: Outlier removal involves identifying and removing data points that deviate significantly from the rest of the dataset. This improves the robustness and generalization of machine learning models by reducing the impact of outliers.
  - Application: We removed outliers from numerical features using techniques such as z-score or interquartile range (IQR) method.
- **Adding New Features:**
  - Purpose: Adding new features enriches the dataset with additional information that may not be captured by existing features, enhancing the model's predictive power.

  - Application: New numerical features were created through mathematical operations, such as calculating ratios or differences between variables. New categorical features were generated by binning numerical variables or combining existing categorical variables.

- **One-Hot Encoding:**
  - Purpose: One-hot encoding converts categorical variables into a numerical format that can be utilized by machine learning algorithms, ensuring effective utilization of categorical information for prediction.
  - Application: We applied one-hot encoding to categorical variables with multiple categories, ensuring that the models can interpret and learn from the categorical information effectively.
- **SMOTE (Synthetic Minority Over-sampling Technique):**
  - Purpose: SMOTE addresses class imbalance by generating synthetic samples for the minority class, ensuring a balanced representation of both classes in the training data.
  - Application: SMOTE was applied to the training set to balance the class distribution, preventing bias towards the majority class and improving model performance on the minority class.

4.4 Training

Binary Classification Models: Binary classification models are supervised machine learning algorithms designed to classify instances into one of two classes based on input features. These models are widely used in various applications such as spam detection, medical diagnosis, and fraud detection. Here, we'll discuss four common binary classification models: Gaussian Naive Bayes (GaussianNB), Logistic Regression, Random Forest Classifier, and Bagging Classifier.

- **Gaussian Naive Bayes (GaussianNB):**
  - Algorithm: Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence between features.
  - Strengths: Simple and computationally efficient. Works well with high-dimensional data. Performs well with categorical features.
  - Weaknesses: Assumes feature independence, which may not hold true in real-world datasets. Sensitive to irrelevant features.
  - Application: GaussianNB is suitable for classification tasks where the features follow a Gaussian distribution and independence assumption holds, making it effective for text classification and spam filtering.
- **Logistic Regression:**
  - Algorithm: Logistic Regression is a linear model that predicts the probability of an instance belonging to a particular class using a logistic function.
  - Strengths: Provides interpretable coefficients representing feature importance. Works well with binary and linearly separable data. Robust to noise and overfitting.
  - Weaknesses: Assumes linear relationship between features and target variable. May underperform when the classes are highly imbalanced.
  - Application: Logistic Regression is widely used in binary classification tasks such as credit risk assessment, customer churn prediction, and disease diagnosis.
- **Random Forest Classifier:**
  - Algorithm: Random Forest Classifier is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes as the prediction.
  - Strengths: Robust to overfitting and noise. Handles high-dimensional data and large datasets well. Provides feature importance measures.
  - Weaknesses: Can be computationally expensive, especially for large numbers of trees. Less interpretable compared to single decision trees.
  - Application: Random Forest Classifier is suitable for a wide range of binary classification tasks, including customer segmentation, fraud detection, and medical diagnosis.

- **Bagging Classifier:**
  - Algorithm: Bagging Classifier is an ensemble method that generates multiple base classifiers using bootstrapped samples of the training data and aggregates their predictions using averaging or voting.
  - Strengths: Reduces variance and improves generalization by combining multiple models. Works well with unstable models. Parallelizable and computationally efficient.
  - Weaknesses: May not perform well with highly biased base classifiers. Limited interpretability due to aggregation of multiple models.
  - Application: Bagging Classifier is effective for binary classification tasks with high variance and instability, such as in noisy datasets or with complex decision boundaries.

Fine-Tuning of Models: Fine-tuning of models is a crucial step in the machine learning pipeline aimed at optimizing the performance of the models by selecting the best hyperparameters. The purpose of fine-tuning is to systematically search through a predefined hyperparameter space to find the combination of hyperparameters that yields the best performance on the validation set. This process helps to improve the model's generalization and predictive accuracy.

Procedure for Fine-Tuning:

1. **Hyperparameter Space Definition:** Before fine-tuning, it's essential to define a hyperparameter space, which includes the set of hyperparameters that will be tuned during the process. These hyperparameters could include regularization strength, learning rate, number of estimators, maximum depth of trees, etc.
2. **Grid Search Cross-Validation (GridSearchCV):** Fine-tuning is commonly performed using techniques like Grid Search Cross-Validation (GridSearchCV). This method exhaustively searches through the hyperparameter space, evaluating the performance of each combination of hyperparameters using cross-validation.
   For each model in consideration (e.g., Naive Bayes, Logistic Regression, Random Forest, Bagging), GridSearchCV is applied with the corresponding pipeline and associated hyperparameter grid.using cross-validation.
3. **Evaluation Metric:** The choice of evaluation metric is critical during fine-tuning. It defines how the performance of each model configuration is assessed. Common evaluation metrics for binary classification tasks include F1-score, accuracy, precision, and recall.
   In the provided code snippet, the F1-score is used as the evaluation metric, which considers both precision and recall, particularly useful for imbalanced datasets.
4. **Grid Search Execution:** GridSearchCV is executed for each pipeline (a sequence of preprocessing steps and a model) with its corresponding hyperparameter grid.
   The grid search iteratively evaluates different combinations of hyperparameters using cross-validation and selects the combination that maximizes the chosen evaluation metric.
5. **Best Model Selection:** Once grid search is complete for each pipeline, the best model from each search is selected based on the performance metric (e.g., F1-score).
   The best models are stored along with their associated names (e.g., Naive Bayes, Logistic Regression) for further evaluation and deployment.
6. **Interpretation of Results:** After fine-tuning, the best hyperparameters for each model (e.g., best parameters) and their corresponding performance scores (e.g., best score) are printed. These results provide insights into which hyperparameters contribute most to the model's performance and help in understanding the behavior of different algorithms on the dataset

4.5 Model Evaluation and Comparison

Classification Report: The classification report provides a comprehensive summary of the performance of a classification model. It includes various metrics such as precision, recall, F1-score, and support for each class (in binary classification, typically "positive" and "negative" classes). Each metric provides valuable insights into different aspects of the model's performance.

- **Precision:** Precision measures the proportion of true positive predictions among all positive predictions made by the model. It indicates the accuracy of positive predictions and is calculated as TP / (TP + FP), where TP is the number of true positives and FP is the number of false positives.
- **Recall (Sensitivity):** Recall measures the proportion of true positive predictions among all actual positive instances in the dataset. It indicates the ability of the model to correctly identify positive instances and is calculated as TP / (TP + FN), where TP is the number of true positives and FN is the number of false negatives.
- **F1-score:** The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It represents the model's accuracy in terms of both precision and recall and is calculated as 2 * (precision * recall) / (precision + recall).

index	precision	recall	f1-score	support
0	0.90	0.94	0.92	1701
1	0.59	0.43	0.50	325
Macro avg:	0.74	0.69	0.71	2026
Weighted avg:	0.85	0.86	0.85	2026
Accuracy:			0.86	2026

Table 3: Classification Report for Naive Bayes

index	precision	recall	f1-score	support
0	0.95	0.95	0.95	1701
1	0.71	0.71	0.71	325
Macro avg:	0.83	0.83	0.83	2026
Weighted avg:	0.91	0.91	0.91	2026
Accuracy:			0.91	2026

Table 4: Classification Report for Logistic Regression

index	precision	recall	f1-score	support
0	0.92	0.97	0.94	1701
1	0.78	0.55	0.65	325
Macro avg:	0.85	0.76	0.79	2026
Weighted avg:	0.90	0.90	0.90	2026
Accuracy:			0.90	2026

Table 5: Classification Report for RandomForest

index	precision	recall	f1-score	support
0	0.93	0.93	0.93	1701
1	0.65	0.65	0.65	325
Macro avg:	0.79	0.79	0.79	2026
Weighted avg:	0.89	0.89	0.89	2026
Accuracy:			0.89	2026

Table 6: Classification Report for Bagging

Interpretation of Classification Reports:

- **Naive Bayes:**
  - Precision: The model achieved a precision of 0.90 for class 0 (negative class) and 0.59 for class 1 (positive class). This indicates that 90% of the instances predicted as class 0 are truly class 0, while 59% of the instances predicted as class 1 are truly class 1.
  - F1-score: The F1-score balances precision and recall, providing a harmonic mean. It is 0.92 for class 0 and 0.50 for class 1.
  - Accuracy: The overall accuracy of the model is 86%, indicating the proportion of correctly classified instances out of all instances.
  - Support: Support represents the number of actual occurrences of each class in the dataset.
- **Logistic Regression:**
  - Precision, recall, F1-score, and accuracy for Logistic Regression are higher compared to Naive Bayes, indicating better performance in classifying both classes.
  - The precision, recall, and F1-score are 0.95, 0.95, and 0.95, respectively, for class 0, and 0.71, 0.71, and 0.71, respectively, for class 1.
  - The weighted average precision, recall, and F1-score are 0.91, indicating overall good performance.
- **Random Forest:**
  - Random Forest achieves higher precision, recall, and F1-score for class 0 compared to class 1, indicating better performance in classifying the majority class.
  - The precision, recall, and F1-score are 0.92, 0.97, and 0.94, respectively, for class 0, and 0.78, 0.55, and 0.65, respectively, for class 1.
  - The model has an overall accuracy of 90%.
- **Bagging:**
  - Bagging Classifier shows balanced performance for both classes, with similar precision, recall, and F1-score for both classes.
  - The precision, recall, and F1-score are 0.93, 0.93, and 0.93, respectively, for class 0, and 0.65, 0.65, and 0.65, respectively, for class 1.
  - The overall accuracy is 89
- **Comparison:**
  - Logistic Regression generally outperforms Naive Bayes, Random Forest, and Bagging in terms of precision, recall, and F1-score, achieving the highest values for both classes.
  - Random Forest has the highest recall for class 0 but lower recall for class 1 compared to Logistic Regression.
  - Bagging shows balanced performance for both classes, with moderate precision, recall, and F1-score for both classes.
  - The choice of the best model may also depend on other factors such as computational efficiency, interpret ability, and specific requirements of the application.

ROC AUC Curve: The ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classification model across various threshold settings. It plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The area under the ROC curve (AUC) is a single scalar value that summarizes the performance of the model across all possible threshold settings.

- **True Positive Rate (TPR):** TPR, also known as sensitivity or recall, measures the proportion of true positive predictions among all actual positive instances in the dataset. It represents the ability of the model to correctly identify positive instances.
- **False Positive Rate (FPR):** FPR measures the proportion of false positive predictions among all actual negative instances in the dataset. It represents the rate of incorrectly classifying negative instances as positive.
- **AUC (Area Under the Curve):** AUC quantifies the overall performance of the classification model. It represents the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. A higher AUC value (closer to 1) indicates better discrimination ability of the model across all threshold settings.
- **Purpose and Interpretation:** Classification Report: The classification report provides a detailed breakdown of the model's performance, helping to understand its strengths and weaknesses. Precision, recall, and F1-score are particularly useful for evaluating the trade-offs between false positives and false negatives, depending on the specific requirements of the application.
- **ROC AUC Curve:** The ROC AUC curve provides a visual representation of the model's performance across different threshold settings. It helps assess the model's ability to distinguish between positive and negative instances, regardless of the chosen threshold. A higher AUC value indicates better discrimination ability and overall performance of the model. The ROC curve also helps in selecting an optimal threshold depending on the desired balance between TPR and FPR.

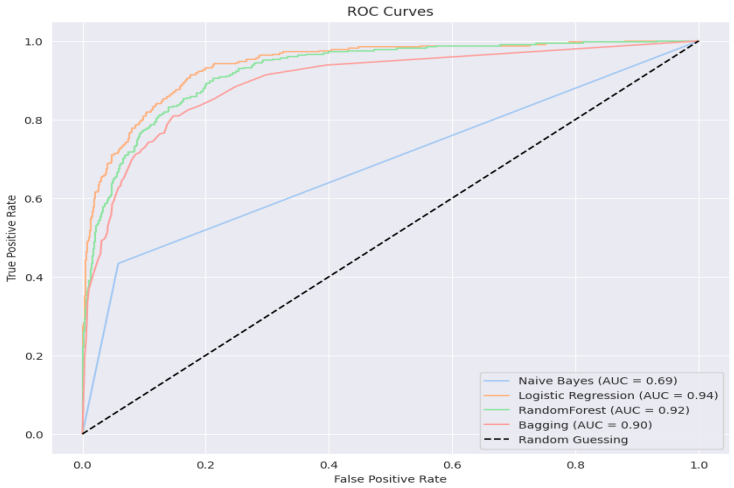

Figure 17: ROC curve

Precision-Recall Curve: The ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classification model across various threshold settings. It plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The area under the ROC curve (AUC) is a single scalar value that summarizes the performance of the model across all possible threshold settings.

- **True Positive Rate (TPR):** TPR, also known as sensitivity or recall, measures the proportion of true positive predictions among all actual positive instances in the dataset. It represents the ability of the model to correctly identify positive instances.
- **Definition and Interpretation:**

- Precision: Precision measures the accuracy of positive predictions made by the model. It is calculated as the ratio of true positive predictions to the total number of positive predictions (TP / (TP + FP)). Precision indicates how many of the predicted positive instances are actually positive.
- Recall (Sensitivity): Recall measures the ability of the model to correctly identify positive instances from all actual positive instances in the dataset. It is calculated as the ratio of true positive predictions to the total number of actual positive instances (TP / (TP + FN)). Recall indicates the completeness of the positive predictions made by the model.

- **Purpose and Usage:** The Precision-Recall curve provides insights into how well a binary classification model performs across different thresholds, particularly in scenarios with imbalanced class distributions. It helps evaluate the trade-off between precision and recall, allowing stakeholders to choose the threshold that best suits their specific requirements and constraints.
- **Advantages over ROC Curve:**
  - Sensitivity to Imbalanced Datasets: The Precision-Recall curve is more informative when dealing with imbalanced datasets compared to the ROC curve. It focuses on the performance of the model with respect to the positive class, making it particularly suitable for scenarios where the positive class is rare.
  - Interpretability: Precision and recall are more interpretable metrics compared to true positive rate (TPR) and false positive rate (FPR) used in the ROC curve. Precision directly measures the accuracy of positive predictions, while recall measures the completeness of positive predictions.
  - Useful for Decision-Making: The Precision-Recall curve helps stakeholders make informed decisions about model performance, especially when differentiating between precision-focused and recall-focused scenarios. Depending on the application, stakeholders can prioritize precision or recall based on their specific needs.

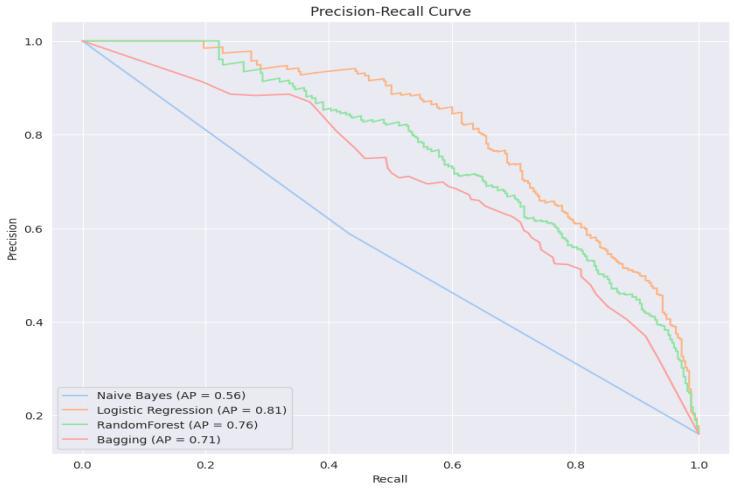

Figure 18: ROC curve

5 References

Hypothesis Testing and Chi-Square Test:

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). Introduction to Linear Regression Analysis. John Wiley & Sons.

Preprocessing Techniques:

Raschka, S. (2015). Python Machine Learning. Packt Publishing.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

Binary Classification Models:

Raschka, S., & Mirjalili, V. (2017). Python Machine Learning, 2nd Edition. Packt Publishing.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.

Fine-Tuning of Models:

Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

Model Evaluation and Comparison:

Raschka, S., & Mirjalili, V. (2017). Python Machine Learning, 2nd Edition. Packt Publishing.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

ROC AUC Curve and Precision-Recall Curve:

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240).