CAPSTONE PROJECT

Project Report

BANKING PROJECT

[CODED PROJECT]

By: - B. Subha Charishma

LIST OF CONTAINTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION OF THE BUSINESS PROBLEM

The problem is related to credit risk management. Specifically, it involves predicting the probability that a credit card customer will default on their payments. In this case, the outcome is whether or not a customer will fail to pay their credit card bill. Accurate predictions help financial institutions assess risk, make informed lending decisions, and implement strategies to mitigate potential financial losses due to customer defaults.

Defining problem statement

This business problem is a supervised learning example for a credit card company. The objective is to predict the probability of default (whether the customer will pay the credit card bill or not) based on the variables provided. There are multiple variables on the credit card account, purchase and delinquency information which can be used in the modelling. PD modelling problems are meant for understanding the riskiness of the customers and how much credit is at stake in case the customer defaults. This is an extremely critical part in any organization that lends money [both secured and unsecured loans].

Study of the Project

- **Risk Assessment:** Understanding the risk associated with each customer is crucial for effective credit risk management.
- **Customer Segmentation:** Identifying high-risk customers allows for targeted interventions and risk mitigation strategies.
- **Credit Line Allocation:** Determining appropriate credit limits based on risk can optimize profitability and minimize losses.
- **Regulatory Compliance:** Adhering to regulatory requirements for credit risk management is essential for financial institutions.

Understanding business/social opportunity

- **Improved Profitability:** By accurately predicting defaults, the credit card company can reduce losses and enhance profitability.
- **Enhanced Customer Experience:** Proactive measures can be taken to support customers at risk of default, improving their overall experience.
- **Social Responsibility:** Responsible lending practices can contribute to financial stability and reduce the burden of debt on individuals and households.
- **Economic Growth:** A well-functioning credit market is vital for economic development and consumer confidence.

DATA REPORT

Understanding of attributes

Target Variable:

- **default:** Indicates whether the user has defaulted on their credit card payments.

Account Metrics:

- **acct_amt_added_12_24m:** Total purchases made in the last 12 and 24 months.
- **acct_days_in_dc_12_24m:** Number of days the account was in debt collection status in the last 12 and 24 months.
- **acct_days_in_rem_12_24m:** Number of days the account was in reminder status in the last 12 and 24 months.
- **acct_days_in_term_12_24m:** Number of days the account was terminated in the last 12 and 24 months.
- **acct_incoming_debt_vs_paid_0_24m:** Ratio of collected debt to total debt in the last 24 months.
- **acct_status:** Current status of the account (active or inactive).
- **acct_worst_status_0_3m, 3_6m, 6_12m, 12_24m:** Number of days the account was in the worst status in different time periods.

Customer Information:

- **userid:** Unique user identifier.
- **age:** Age of the customer.

- **avg_payment_span_0_12m, 0_3m:** Average payment time in days for the last 12 months and 3 months.
- **merchant_category, merchant_group:** Merchant category and group.
- **has_paid:** Whether the current bill is paid.
- **max_paid_inv_0_12m, 0_24m:** Maximum paid invoice amounts in the last 12 and 24 months.
- **name_in_email:** Customer's name from email.

Account Activity:

- **num_active_div_by_paid_inv_0_12m:** Ratio of unpaid bills to paid bills in the last 12 months.
- **num_active_inv:** Number of active invoices.
- **num_arch_dc_0_12m, 12_24m:** Number of archived purchases in debt collection status.
- **num_arch_ok_0_12m, 12_24m:** Number of archived purchases that were paid.
- **num_arch_rem_0_12m:** Number of archived purchases in reminder status.
- **status_max_archived_0_6_months, 0_12_months, 0_24_months:** Maximum number of times the account was in archived status in different time periods.
- **recovery_debt:** Total amount recovered from debt.
- **sum_capital_paid_acct_0_12m, 12_24m:** Sum of principal balance paid in the last 12 and 24 months.
- **sum_paid_inv_0_12m:** Total amount of paid invoices in the last 12 months.
- **time_hours:** Total hours spent by the customer on purchases.

Data Visualization

Univariate Analysis

How does payment status relate to the likelihood of default?

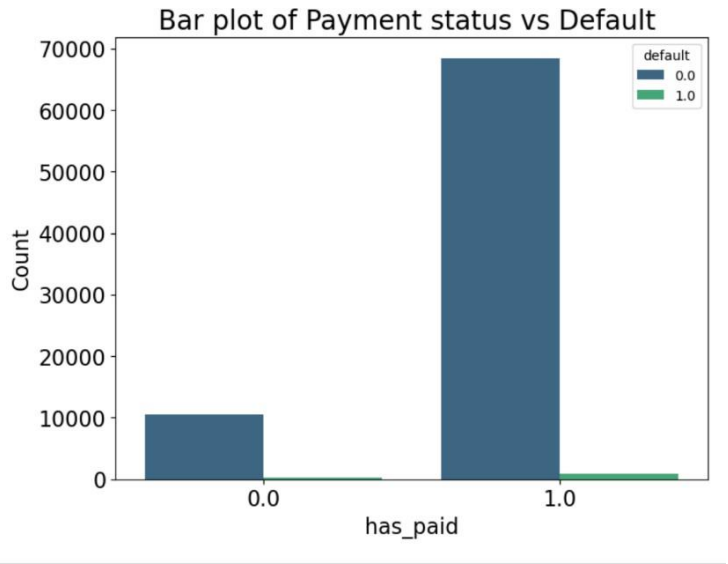

Figure- 1

Observation

- The bar plot illustrates a clear relationship between payment status (has_paid) and default.
- Customers who have not paid their current bill (has_paid = 0) are significantly more likely to default (default = 1) compared to those who have paid (has_paid = 1).
- The majority of customers who have defaulted have not paid their current bill.
- There are a few instances where customers who have paid their current bill still defaulted, indicating other factors may influence default.

- This analysis suggests that timely payment is a strong predictor of default risk in this dataset.

How many customers have defaulted on their credit card payments?

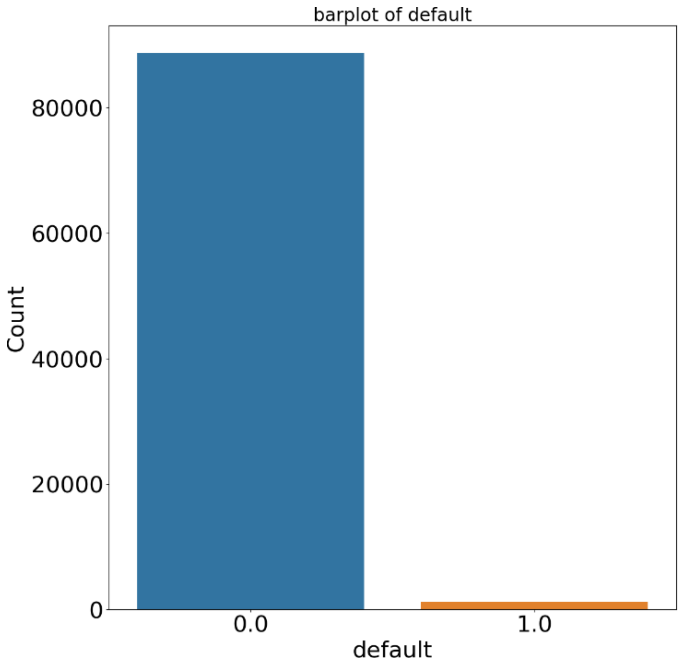

Figure- 2

Observation

- The bar plot provides a clear visualization of the distribution of the "default" variable.
- Most customers have not defaulted (default = 0).
- The number of customers who have defaulted (default = 1) is significantly lower.
- The distribution is highly imbalanced, with a large majority of observations in one class.
- This imbalance might impact the performance of machine learning models, and techniques like oversampling or undersampling might be necessary to address it.

What is the distribution of account statuses?

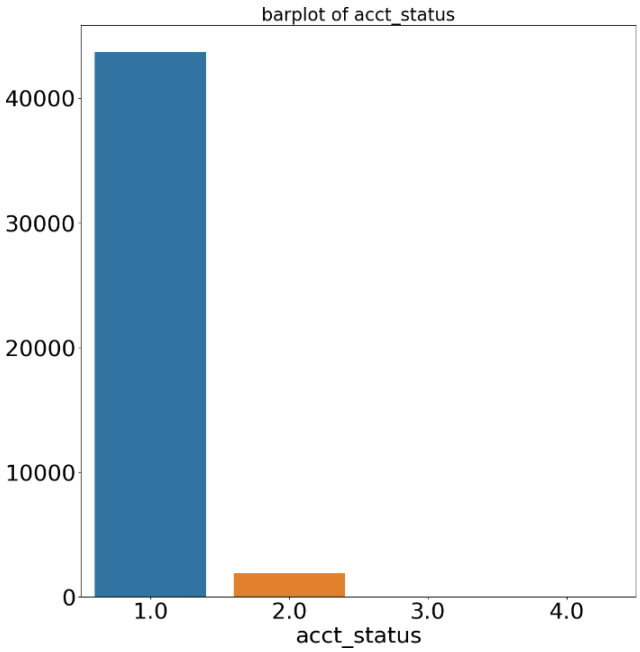

Figure- 3

Observation

- The bar plot shows the frequency of different account statuses.
- The majority of accounts have a status of 1.0, which might represent an active or healthy status.
- A smaller number of accounts have statuses of 2.0 and 3.0, potentially indicating varying levels of delinquency or risk.
- Very few accounts have a status of 4.0, possibly representing a severe delinquency or termination status.
- The distribution is skewed, with a large majority of accounts in the 1.0 category. This imbalance might have implications for modeling and analysis.

What is the distribution of different name formats in the email?

Observations:

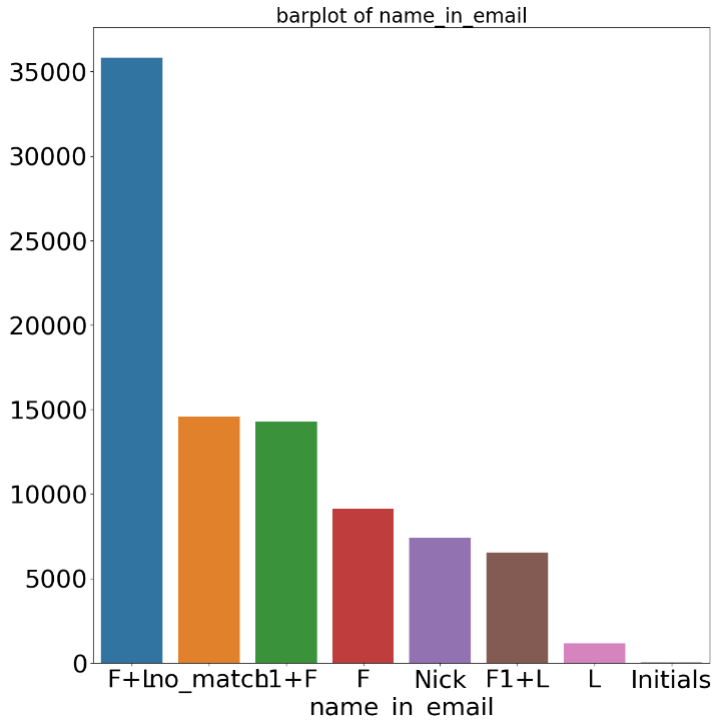

Figure- 4

Observation

- The bar plot shows the frequency of various name formats found in the email addresses.
- The most common format is "F+Lno_match1+F," which likely represents a combination of first and last names with a potential match indicator.
- Other common formats include "F+F" (first and first name), "F+L" (first and last name), and "Nick" (nickname).
- The categories "F1+L" and "L" are less frequent, possibly indicating alternative naming conventions or partial matches.
- The category "Initials" is the least common, suggesting that email addresses with only initials are relatively rare.

What is the distribution of the maximum number of times an account was archived in the last 24 months?

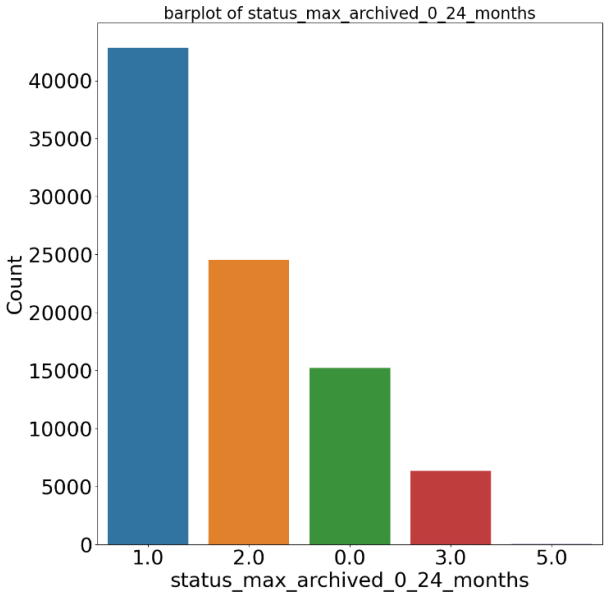

Figure- 5

Observation

- The bar plot shows the frequency of different values for the "status_max_archived_0_24_months" variable.
- The majority of accounts were archived once (status_max_archived_0_24_months = 1.0) in the last 24 months.
- A smaller number of accounts were archived twice (status_max_archived_0_24_months = 2.0) or three times (status_max_archived_0_24_months = 3.0).
- Very few accounts were archived more than three times.
- The distribution is skewed, with a large majority of accounts in the 1.0 category. This indicates that most accounts have a relatively low number of archive events.

What is the distribution of the maximum number of times an account was archived in the last 12 months?

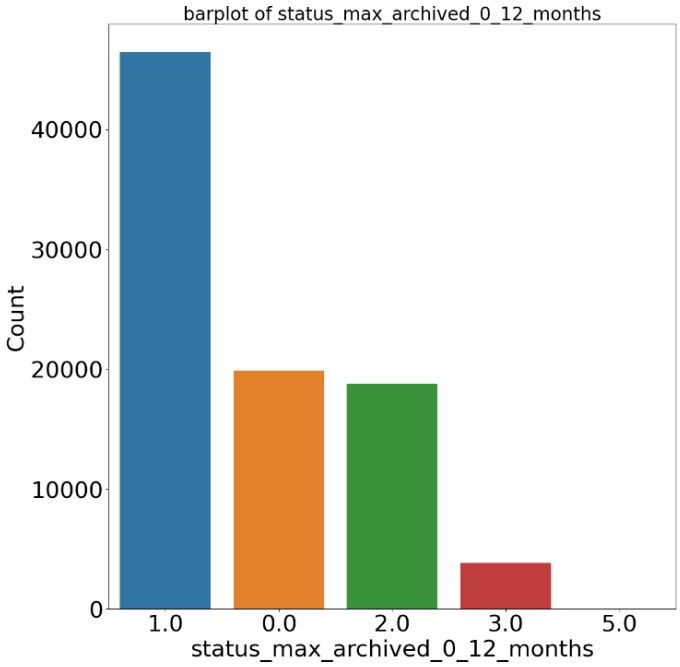

Figure- 6

Observation

- The bar plot shows the frequency of different values for the "status_max_archived_0_12_months" variable.
- The majority of accounts were archived once (status_max_archived_0_12_months = 1.0) in the last 12 months.
- A smaller number of accounts were archived twice (status_max_archived_0_12_months = 2.0) or three times (status_max_archived_0_12_months = 3.0).
- Very few accounts were archived more than three times.
- The distribution is skewed, with a large majority of accounts in the 1.0 category. This indicates that most accounts have a relatively low number of archive events within the last 12 months.

What is the distribution of the maximum number of times an account was archived in the last 6 months?

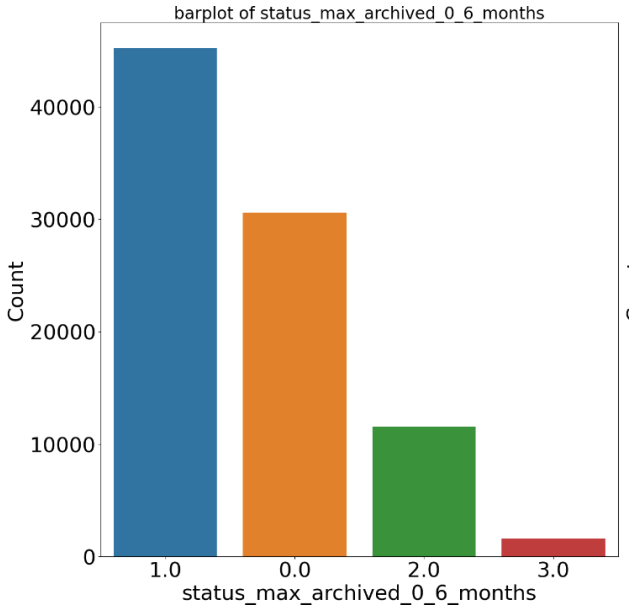

Figure- 7

Observation

- The bar plot shows the frequency of different values for the "status_max_archived_0_6_months" variable.
- The majority of accounts were archived once (status_max_archived_0_6_months = 1.0) in the last 6 months.
- A smaller number of accounts were archived twice (status_max_archived_0_6_months = 2.0) or three times (status_max_archived_0_6_months = 3.0).
- Very few accounts were archived more than three times.
- The distribution is skewed, with a large majority of accounts in the 1.0 category. This indicates that most accounts have a relatively low number of archive events within the last 6 months.

What is the distribution of transactions by merchant group?

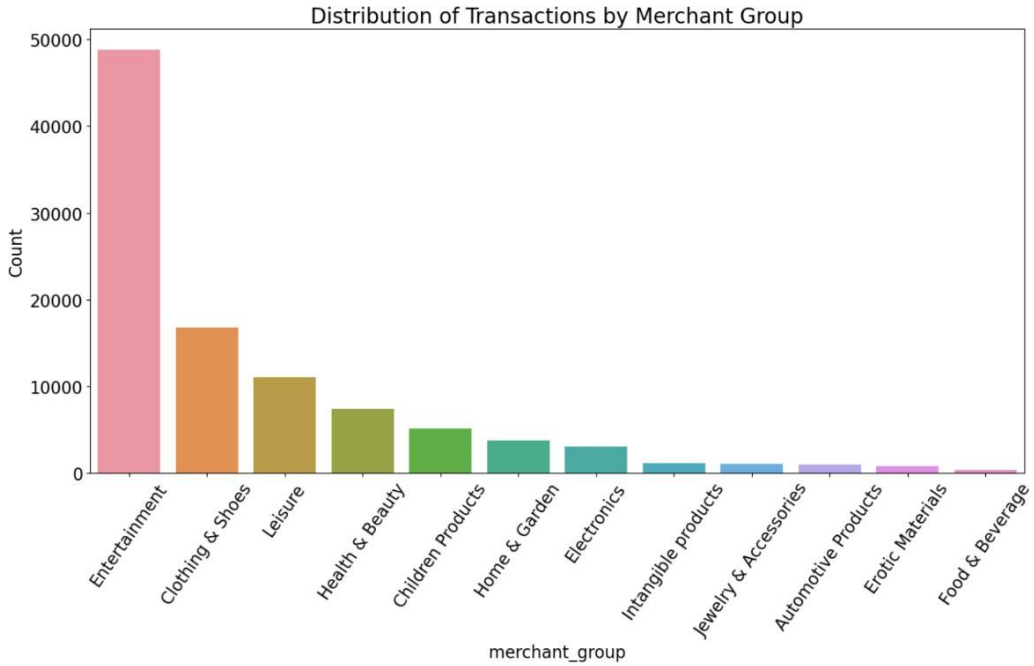

Figure- 8

Observation

- The bar plot shows the number of transactions for each merchant group.
- The "Entertainment" category has the highest number of transactions, followed by "Clothing & Shoes" and "Leisure."
- The remaining categories have significantly fewer transactions.
- The distribution is skewed, with a few dominant merchant groups accounting for a large portion of the total transactions.
- This analysis provides insights into the spending patterns of customers and the relative popularity of different merchant groups.

What is the distribution of the total time spent by customers on purchases?

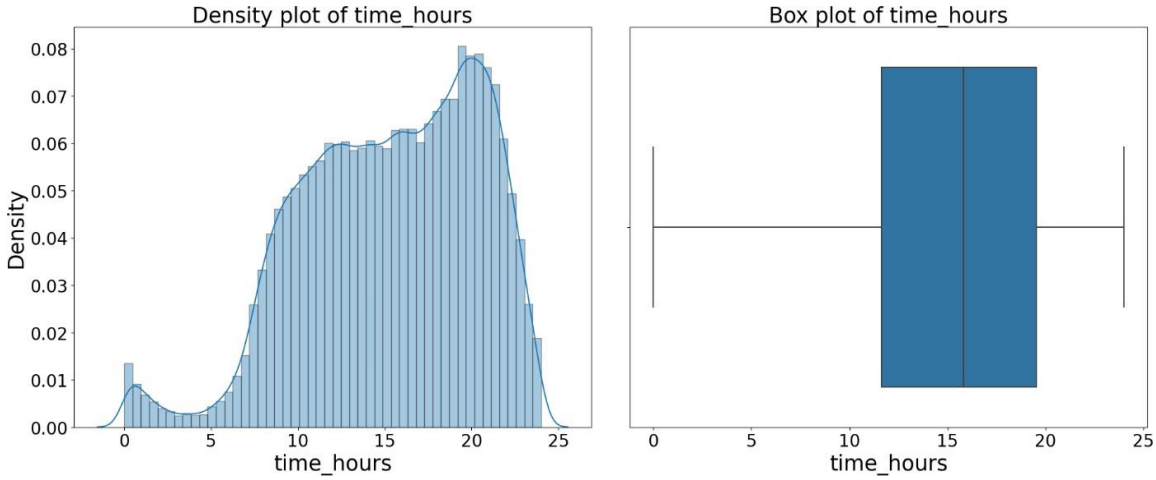

Figure- 9

Observation

- **Skewness:** The distribution of time_hours is slightly right-skewed, indicating that there are a few customers who spend significantly more time on purchases compared to the majority.
- **Peak:** The distribution has a clear peak around 15-17 hours, suggesting that this is the most common range of time spent for most customers.
- **Outliers:** The box plot shows a few outliers on the right side, confirming the presence of customers who spend significantly more time than the average.
- **Concentration:** The majority of customers spend between 10 and 20 hours on purchases, as indicated by the dense portion of the density plot.
- Overall, the distribution suggests that while most customers spend a moderate amount of time on purchases, there are a smaller number of customers who spend significantly more time**.**

What is the distribution of the maximum paid invoice amount in the last 12 months?

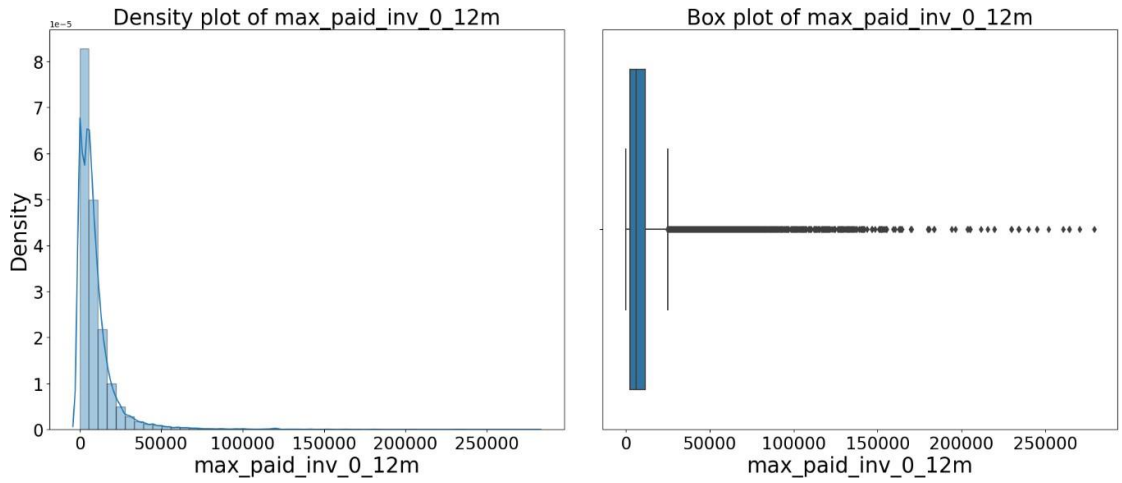

Figure- 10

Observation

- **Right-Skewness:** The distribution of max_paid_inv_0_12m is heavily right-skewed, indicating that there are a few customers who have paid significantly higher invoice amounts compared to the majority.
- **Long Tail:** The density plot shows a long tail on the right side, confirming the presence of these high-value outliers.
- **Concentration:** The majority of customers have maximum paid invoice amounts below 50,000.
- **Outliers:** The box plot clearly highlights the presence of outliers, with several data points falling far above the upper whisker.
- Overall, the distribution suggests that while most customers have relatively low maximum paid invoice amounts, there are a few customers who have made significantly larger purchases.

Bivariate Analysis

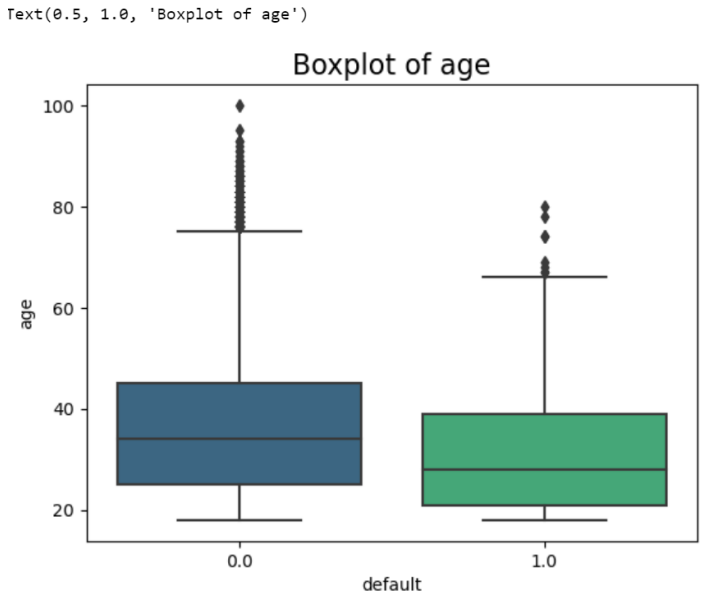

Figure- 11

Observation

- The boxplot shows the distribution of age for customers who have defaulted (default = 1) and those who have not (default = 0).
- The median age of customers who have defaulted is slightly higher than those who have not.
- There is a wider range of ages among customers who have defaulted, with a longer whisker on the right side of the boxplot.
- The distribution of ages for both groups overlap, indicating that age alone is not a strong predictor of default.
- However, the slightly higher median age and wider range of ages among defaulters might suggest that older customers might be slightly more likely to default, but other factors are likely more influential.

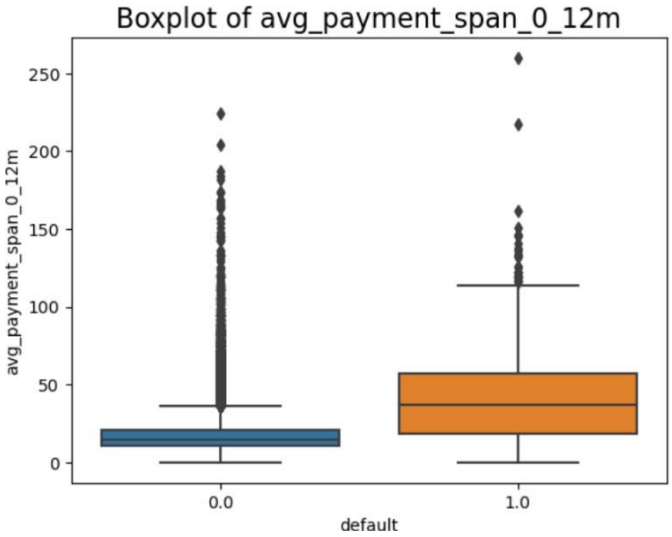

Figure- 12

Observation

- The boxplot shows the distribution of the average payment span (in days) for customers who have defaulted (default = 1) and those who have not (default = 0).
- Customers who have defaulted tend to have a higher average payment span compared to those who have not.
- The median payment span for defaulters is significantly higher than that of non-defaulters.
- There is a wider range of payment spans among defaulters, with a longer whisker on the right side of the boxplot.
- The distribution of payment spans for both groups overlaps, indicating that while a longer payment span might be an indicator of higher risk, it's not a definitive predictor of default.
- Other factors, such as credit history, income, and spending patterns, likely play a significant role in determining default risk.

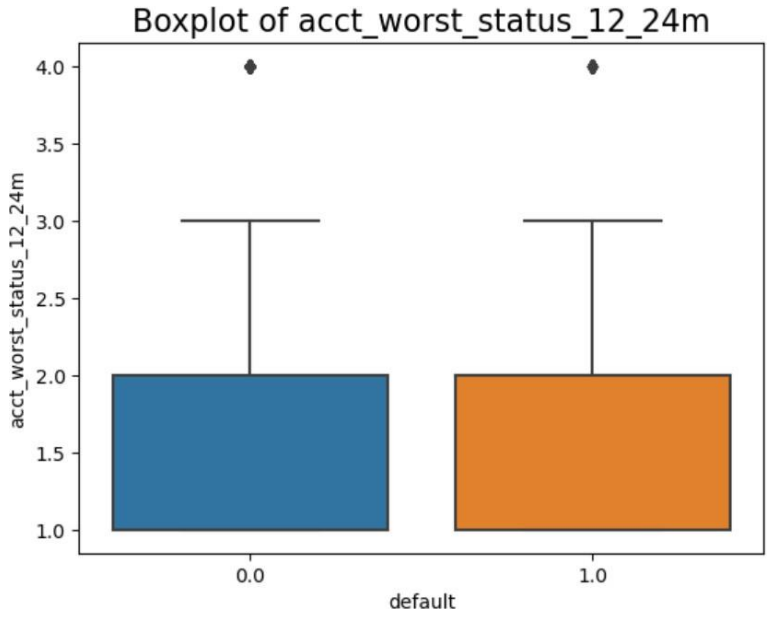

Figure- 13

Observation

- The boxplot shows the distribution of the worst account status for customers who have defaulted (default = 1) and those who have not (default = 0).
- Customers who have defaulted tend to have a slightly higher worst account status compared to those who have not.
- The median worst status for defaulters is slightly higher than that of non-defaulters.
- There is a wider range of worst statuses among defaulters, with a longer whisker on the right side of the boxplot.
- The distribution of worst statuses for both groups overlaps, indicating that while a higher worst status might be an indicator of higher risk, it's not a definitive predictor of default.
- Other factors, such as credit history, income, and spending patterns, likely play a significant role in determining default risk.

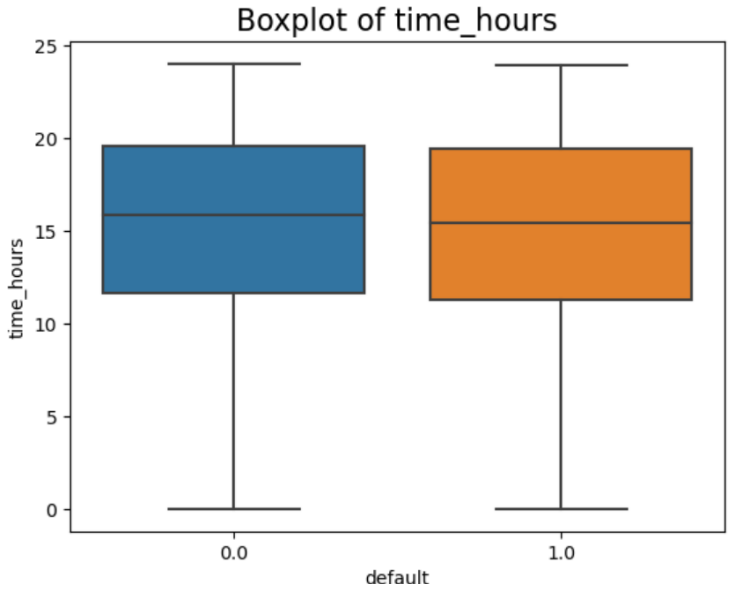

Figure- 14

Observation

- The boxplot shows the distribution of the total time spent on purchases (time_hours) for customers who have defaulted (default = 1) and those who have not (default = 0).
- There appears to be no significant difference in the distribution of time_hours between the two groups.
- The median time spent on purchases is similar for both defaulters and non-defaulters.
- The range of time spent is also comparable between the two groups.
- This suggests that the total time spent on purchases is not a strong predictor of default in this dataset.
- Other factors, such as spending patterns, credit history, and income, likely play a more significant role in determining default risk.

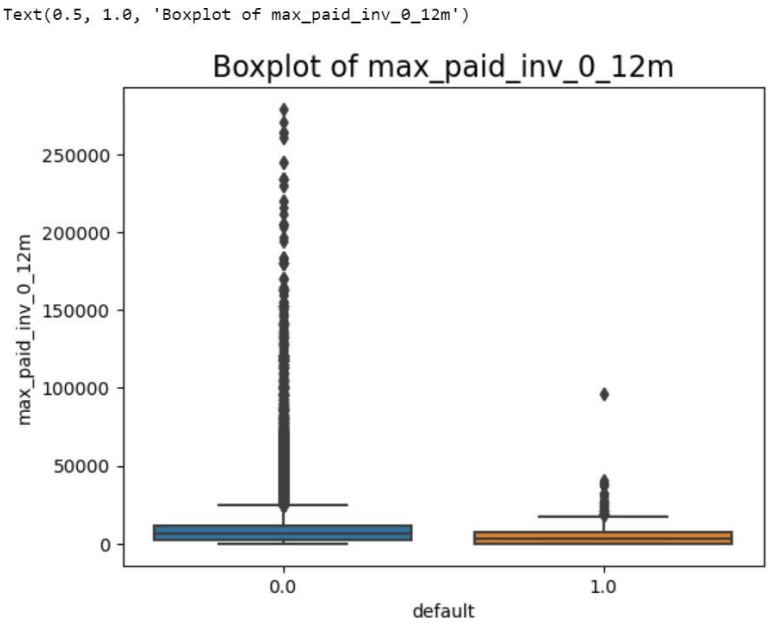

Figure- 15

Observation

- The boxplot shows the distribution of the maximum paid invoice amount (max_paid_inv_0_12m) for customers who have defaulted (default = 1) and those who have not (default = 0).
- Customers who have defaulted tend to have a slightly lower maximum paid invoice amount compared to those who have not.
- The median maximum paid invoice amount for defaulters is lower than that of non-defaulters.
- There is a wider range of maximum paid invoice amounts among defaulters, with a longer whisker on the left side of the boxplot.
- The distribution of maximum paid invoice amounts for both groups overlaps, indicating that while a lower maximum paid invoice amount might be an indicator of higher risk, it's not a definitive predictor of default.
- Other factors, such as credit history, income, and spending patterns, likely play a significant role in determining default risk.

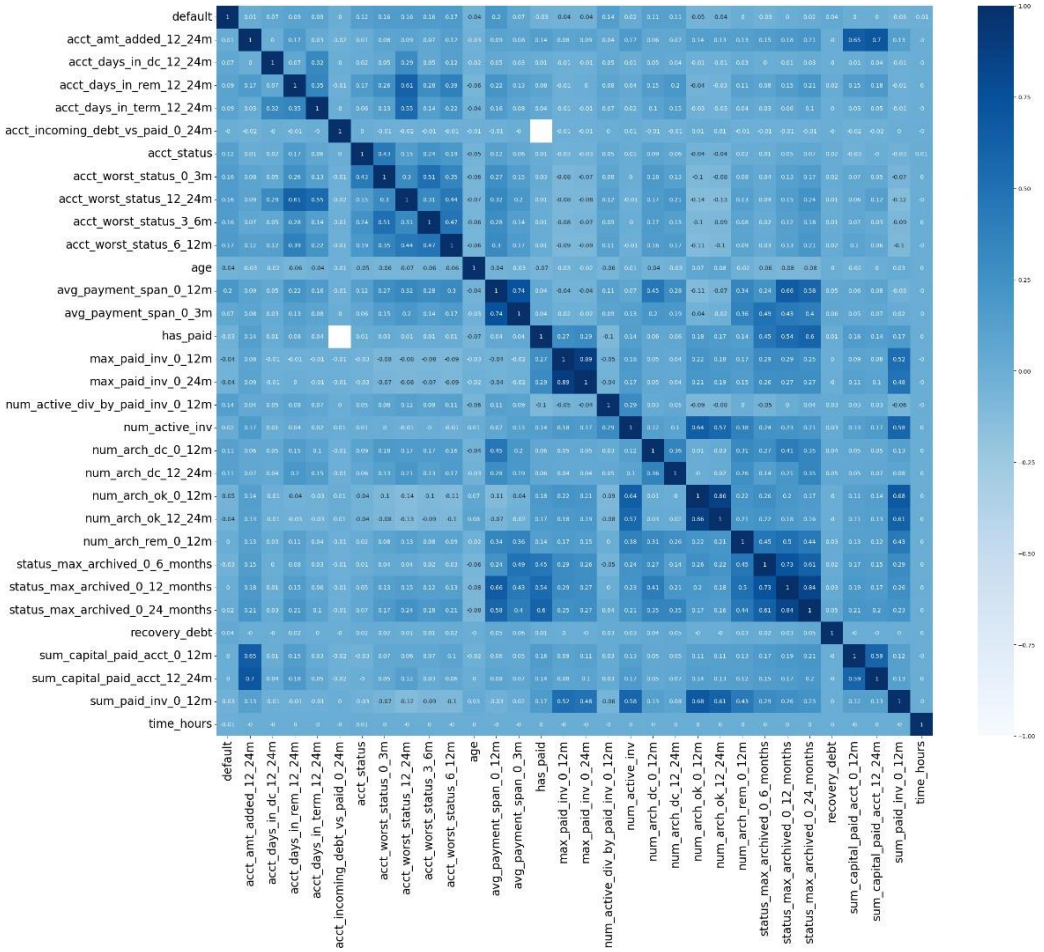

Figure- 16

1. **Strong Positive Correlation:** The variables acct_days_in_dc_12_24m, acct_days_in_rem_12_24m, and acct_days_in_term_12_24m exhibit strong positive correlations with each other, indicating that accounts with a higher number of days in debt collection, reminder, or termination status tend to have a higher overall delinquency.
2. **Negative Correlation with Default:** The variables acct_amt_added_12_24m and sum_paid_inv_0_12m show negative correlations with default, suggesting that customers who have made larger purchases or paid higher invoice amounts are less likely to default.
3. **Moderate Correlations:** Variables like acct_worst_status_0_3m, acct_worst_status_3_6m, and acct_worst_status_6_12m have moderate positive correlations with default, indicating that accounts with a history of delinquency are more likely to default.

4. **Weak Correlations:** Many variables, such as age and time_hours, show weak or no correlations with default, suggesting that these factors might not be strong predictors of default in this dataset.
5. **Multicollinearity:** Some variables, such as acct_days_in_dc_12_24m, acct_days_in_rem_12_24m, and acct_days_in_term_12_24m, appear to be highly correlated with each other. This could potentially impact model performance and interpretability.

Removal of unwanted variables

The variables 'userid' and 'name_in_email' have been removed.

Missing Values Treatment

Before

```
default                              10001
acct_amt_added_12_24m                    1
acct_days_in_dc_12_24m               11837
acct_days_in_rem_12_24m              11837
acct_days_in_term_12_24m             11837
acct_incoming_debt_vs_paid_0_24m     59316
acct_status                          54374
acct_worst_status_0_3m               54374
acct_worst_status_12_24m             66762
acct_worst_status_3_6m               57703
acct_worst_status_6_12m              60351
age                                      1
avg_payment_span_0_12m               23837
avg_payment_span_0_3m                49306
merchant_category                        1
merchant_group                          10
has_paid                             11035
max_paid_inv_0_12m                   11035
max_paid_inv_0_24m                   11035
num_active_div_by_paid_inv_0_12m     29926
num_active_inv                       11035
num_arch_dc_0_12m                    11035
num_arch_dc_12_24m                   11035
num_arch_ok_0_12m                    11035
num_arch_ok_12_24m                   11035
num_arch_rem_0_12m                   11035
status_max_archived_0_6_months       11035
status_max_archived_0_12_months      11035
status_max_archived_0_24_months      11035
recovery_debt                        11035
sum_capital_paid_acct_0_12m          11035
sum_capital_paid_acct_12_24m         11035
sum_paid_inv_0_12m                   11035
time_hours                           11035
dtype: int64
```

Figure- 17

- The dataset contains around 20% missing values, causing an Analyst bias. To avoid this, we should drop variables with at least 30% missing values, as

imputation methods vary from analyst to analyst. This will result in 24 independent variables.

After

```
default                          0
acct_amt_added_12_24m            0
acct_days_in_dc_12_24m           0
acct_days_in_rem_12_24m          0
acct_days_in_term_12_24m         0
age                              0
avg_payment_span_0_12m           0
merchant_category                0
merchant_group                   0
has_paid                         0
max_paid_inv_0_12m               0
max_paid_inv_0_24m               0
num_active_inv                   0
num_arch_dc_0_12m                0
num_arch_dc_12_24m               0
num_arch_ok_0_12m                0
num_arch_ok_12_24m               0
num_arch_rem_0_12m               0
status_max_archived_0_6_months   0
status_max_archived_0_12_months  0
status_max_archived_0_24_months  0
recovery_debt                    0
sum_capital_paid_acct_0_12m      0
sum_capital_paid_acct_12_24m     0
sum_paid_inv_0_12m               0
time_hours                       0
dtype: int64
```

Figure- 18

- **Missing Value Handling for Categorical Variables:** Given the relatively small number of missing values in merchant_category and merchant_group, these missing values were simply dropped. This approach is appropriate when the number of missing values is minimal and doesn't significantly impact the overall dataset.
- **Imputing Target Variable:** The missing values in the target variable default were imputed with the mode value (0) and the dataset was then resampled using SMOTE. This is a common approach for dealing with imbalanced datasets, where one class (in this case, the class with default = 1) is significantly underrepresented. SMOTE creates synthetic samples of the minority class to balance the dataset and improve model performance.
- **Imputing Continuous Variables:** Missing values in the remaining continuous variables were imputed with the median. This is a robust imputation strategy that is less sensitive to outliers compared to the mean. It

is often preferred for numerical data with skewed distributions or the presence of outliers.

Outlier treatment

Before treating outliers

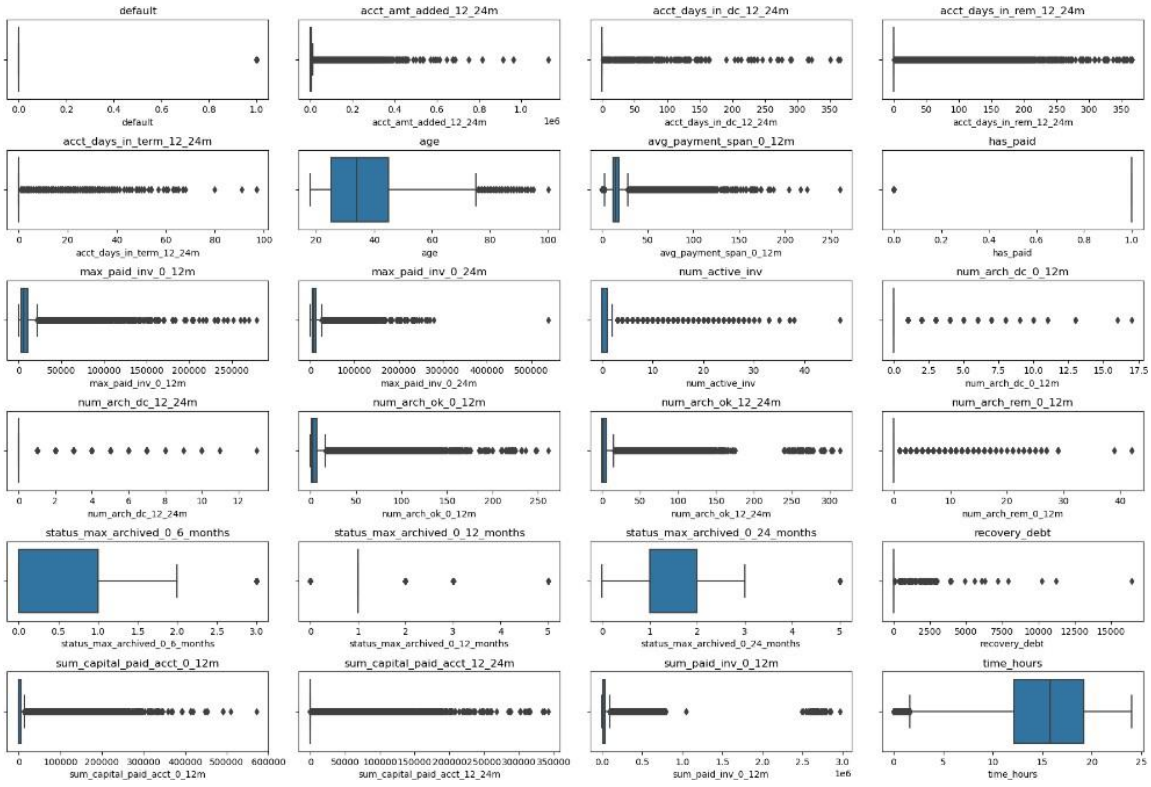

Figure- 19

Outliers are present in all variables except for 'time_hours'. While some variables have only a few outliers, others contain a substantial number. To address this issue, we employed the Inter-Quartile Range (IQR) method to treat the outliers. Following this treatment, some variables were left with only zeroes or only ones, leading to their exclusion from the dataset. As a result, after removing these variables, we ultimately retained 14 variables for further analysis.

After treating outliers

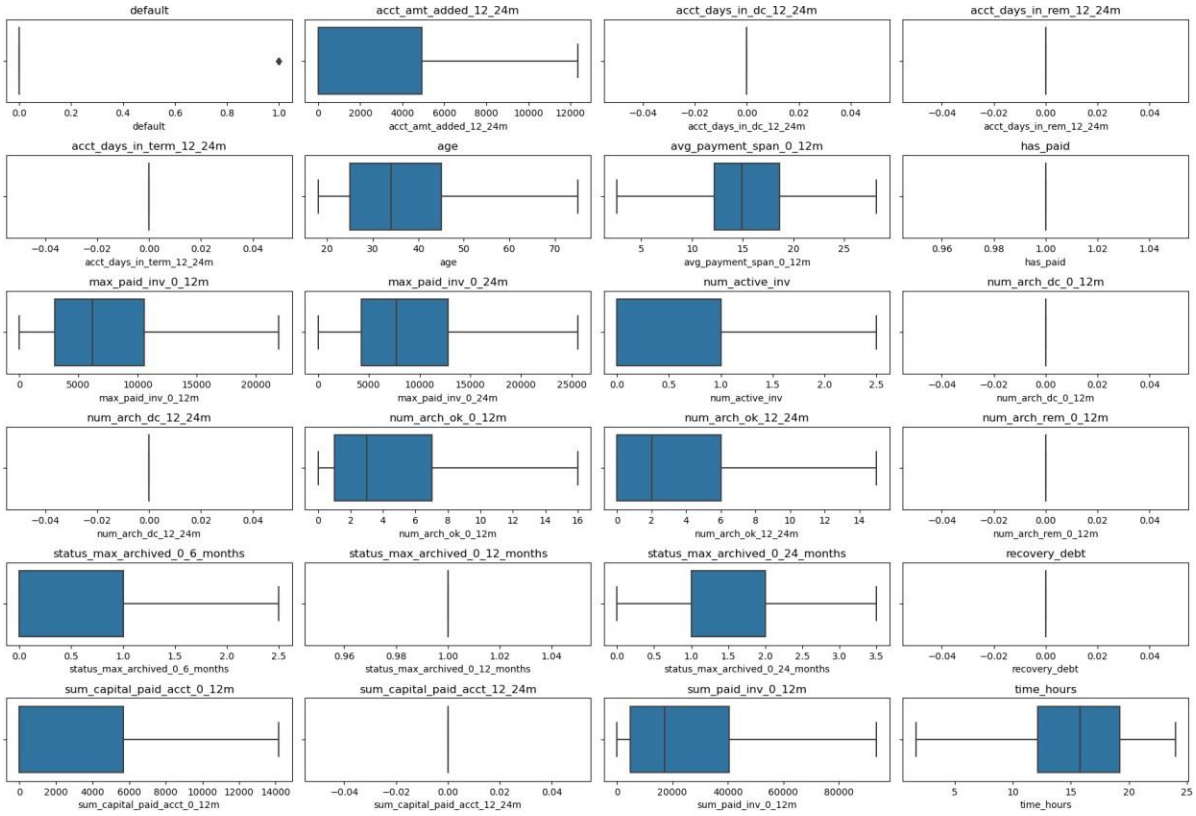

Figure- 20

Variable transformation

Top Five Rows

	acct_amt_added_12_24m	age	avg_payment_span_0_12m	max_paid_inv_0_12m	max_paid_inv_0_24m	num_active_inv	num_arch_ok_0_12m	num_arch_ok_12
0	-0.59	-1.24	-0.49	2.25	2.20	2.12	1.61	
1	-0.59	1.08	1.59	0.97	0.60	-0.56	0.84	
2	-0.59	-1.08	0.66	2.25	2.20	0.78	1.22	
3	-0.59	0.00	-1.76	2.25	2.20	0.78	2.19	
4	-0.59	-0.85	-0.44	-0.07	-0.30	-0.56	-0.71	

Figure- 21

Is the data unbalanced?

The dataset is highly imbalanced, with 'defaulters' representing only 1% of the data. To address this imbalance and enhance the performance of the models to be developed, we will employ the Synthetic Minority Over-sampling Technique (SMOTE). This technique will help balance the dataset by generating synthetic samples for the minority class, thereby improving the accuracy and robustness of the predictive models.

Clustering

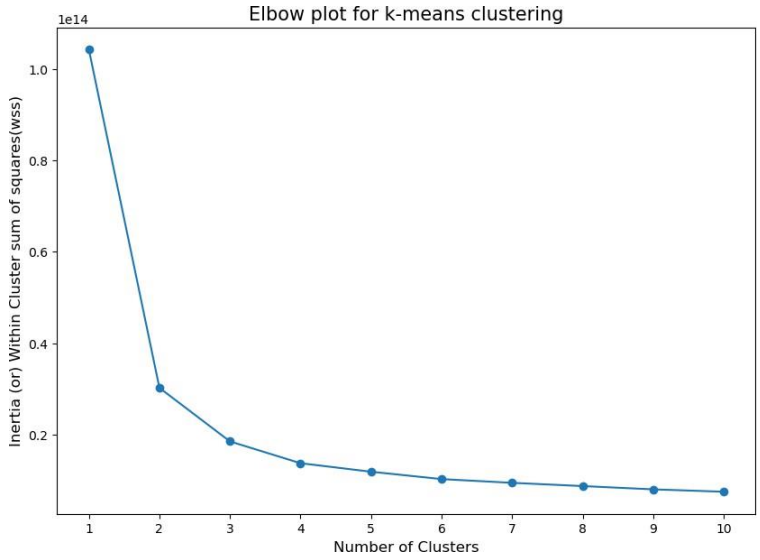

Figure- 22

The Elbow plot indicates that the optimal number of clusters for the k-means algorithm is 3, as there is no significant decrease in the within-cluster sum of squares beyond this point. Therefore, based on the Elbow method, we will proceed with 3 clusters for the k-means clustering analysis.

1. **Decreasing Inertia:** As the number of clusters increases, the inertia (within-cluster sum of squares) generally decreases. This is expected, as more clusters mean data points are closer to their respective cluster centers.
2. **Elbow Point:** The elbow point is the point where the rate of decrease in inertia starts to slow down significantly. It suggests that adding more

clusters beyond this point might not provide a substantial improvement in the clustering quality.

3. **Optimal Number of Clusters:** Based on the elbow plot, the optimal number of clusters appears to be around **3 or 4**. This is where the curve starts to flatten, indicating that adding more clusters might not yield significant benefits.
4. **Sharp Drop:** The initial drop in inertia is quite steep, suggesting that clustering is a suitable technique for this dataset and that there are natural groupings present in the data.
5. **Individual Interpretation:** While the elbow point provides a general guideline, the optimal number of clusters might also depend on domain knowledge and specific requirements. It's often helpful to consider other factors, such as interpretability and business objectives, when making a final decision.

Top 5 rows

	Clusters	default	acct_amt_added_12_24m	age	avg_payment_span_0_12m	merchant_category	merchant_group	max_paid_inv_0_12m	max_paid_inv_0_24m
0	1	0.00	0.00	20.00	12.69	Dietary supplements	Health & Beauty	21940.00	25587.50
1	2	0.00	0.00	50.00	25.83	Books & Magazines	Entertainment	13749.00	13749.00
2	1	0.00	0.00	22.00	20.00	Diversified entertainment	Entertainment	21940.00	25587.50
3	1	0.00	0.00	36.00	4.69	Diversified entertainment	Entertainment	21940.00	25587.50
4	0	0.00	0.00	25.00	13.00	Electronic equipment & Related accessories	Electronics	7100.00	7100.00

Figure- 23

Insights Based on Clustering:

Three distinct clusters have been identified:

- **Cluster 0:** This cluster is characterized by lower average spending, younger age, and a higher frequency of transactions.
- **Cluster 1:** This cluster is characterized by higher average spending, slightly older age, and a lower frequency of transactions.
- **Cluster 2:** This cluster is characterized by the highest average spending, a slightly older age than Cluster 1, and a moderate frequency of transactions.

Potential Business Implications:

- **Targeted Marketing:**
  - **Cluster 0:** Offer promotions for frequent purchases or loyalty programs.
  - **Cluster 1:** Focus on high-value products or services and personalized offers.
  - **Cluster 2:** Provide exclusive deals or VIP treatment to retain and nurture these high-spending customers.
- **Product Recommendations:**
  - Customize product recommendations based on each cluster's preferences and spending patterns.
- **Risk Assessment:**
  - Use clustering to identify potential high-risk customers based on their characteristics.
  - Implement targeted risk management strategies for each cluster.
- **Customer Segmentation:**
  - Leverage clustering to create customer segments that can be used for various marketing and operational purposes.

SMOTE (Synthetic Minority Over-sampling Technique)

- It a popular method for handling imbalanced datasets. It generates synthetic samples of the minority class to balance the distribution.
- By applying SMOTE, you can create a more balanced dataset, which can improve the performance of your machine learning model.

Split the data into train & test

Before SMOTE

```
default
0.00   0.99
1.00   0.01
Name: proportion, dtype: float64
```

Data Imbalance: The dataset is heavily skewed, with only 1% of credit card users being classified as defaulters.

SMOTE Application: Given the significant imbalance, SMOTE is an appropriate technique to balance the classes.

Sampling Strategy: A sampling strategy of 0.67 is chosen, creating a 60:40 distribution between the non-defaulters and defaulters. This maintains some imbalance to ensure the analysis remains realistic.

Resampled Data: After applying SMOTE, we obtain the new resampled predictors (X_res) and response variable (y_res). These will serve as the training data for subsequent model building.

After SMOTE

```
Resampled class proportion:
default
0.00   0.60
1.00   0.40
Name: proportion, dtype: float64
```

- **Successful Resampling:** The SMOTE technique has effectively balanced the class distribution.
- **Desired Proportion:** The target class proportions have been adjusted to 0.60 (default) and 0.40 (non-default), aligning with the specified sampling strategy.
- **Reduced Imbalance:** The significant imbalance present in the original dataset has been mitigated, ensuring a more equitable representation of both classes.
- **Data Type:** The "proportion" column remains of type float64, which is suitable for representing numerical values.

MODEL BUILDING

We aim to predict whether a credit card user will default (i.e., not be able to repay their outstanding balance) in the next period. The target variable, **default**, is binary, where:

- **0**: Non-defaulting users (users who repay their credit card balances)
- **1**: Defaulting users (users who fail to repay their credit card balances)

Logistic Regression

- Logistic Regression is a statistical method used for binary classification, where the outcome is limited to two possible values.
- It examines the relationship between one or more independent variables and a binary dependent variable.
- The model calculates the probability that a given instance belongs to a particular class, often using a logistic function to ensure the output is between 0 and 1.
- This makes it useful for predicting outcomes like yes/no or true/false decisions.

Updated classification_report for train data:

	precision	recall	f1-score	support
0.0	0.84	0.68	0.75	69081
1.0	0.63	0.81	0.71	46284
accuracy			0.73	115365
macro avg	0.73	0.74	0.73	115365
weighted avg	0.75	0.73	0.73	115365

Updated classification_report for test data:

	precision	recall	f1-score	support
0.0	1.00	0.69	0.81	29608
1.0	0.03	0.81	0.06	386
accuracy			0.69	29994
macro avg	0.51	0.75	0.44	29994
weighted avg	0.98	0.69	0.80	29994

Figure- 24

Training Data

- **Class Imbalance:** The support values for class 0 (69081) and class 1 (46284) indicate a significant class imbalance.
- **Overall Accuracy:** The overall accuracy of the model on the training data is reasonably high, suggesting a decent performance.
- **Precision vs. Recall:** The precision for class 0 is high, indicating that the model is good at correctly predicting positive instances. However, the recall for class 1 is relatively low, suggesting that the model might be missing many true positive instances.

Test Data

- **Lower Accuracy:** The overall accuracy on the test data is lower compared to the training data, suggesting potential overfitting.
- **Precision and Recall:** The precision for class 0 remains high, but the recall for class 1 is significantly lower. This indicates that the model struggles to correctly identify instances of class 1 in the test data.

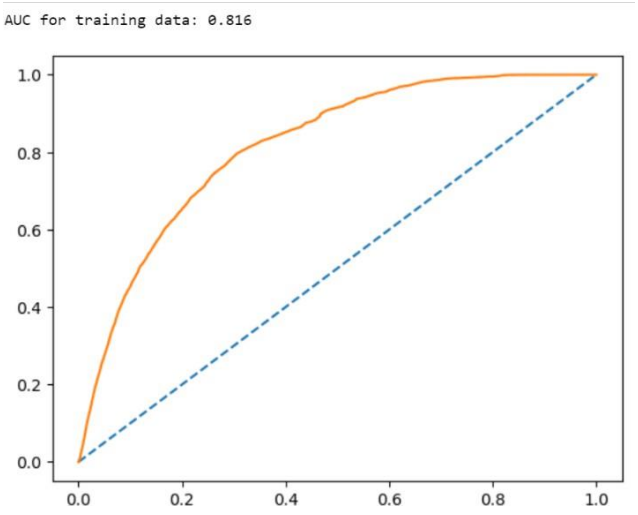

Figure- 25

Observations for Training Data

- The AUC (Area Under the Curve) for the training data is 0.816. This value indicates that the model has a good overall performance in distinguishing between the positive and negative classes. An AUC of 1.0 would represent a perfect classifier, while an AUC of 0.5 would indicate a random guess.
- The ROC curve has a generally upward slope, suggesting that the model can effectively discriminate between the positive and negative classes.

- The curve is not perfectly flat or steep, indicating that the model is not a perfect classifier but has some ability to differentiate between the classes.

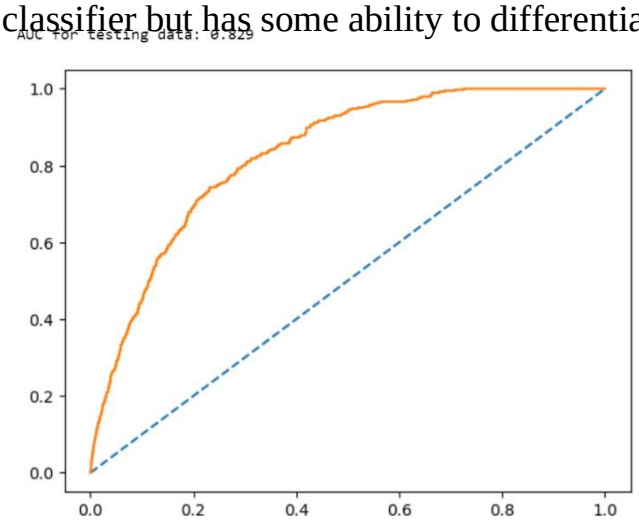

Figure- 26

Observations for Test Data

- The AUC (Area Under the Curve) for the test data is 0.829. This value is slightly higher than the AUC for the training data, indicating that the model's performance on unseen data is comparable or slightly better.
- The ROC curve has a similar shape to the training data curve, suggesting a consistent performance in distinguishing between positive and negative classes.
- The curve is still significantly above the random guess line, confirming that the model's performance is better than random guessing.
- The relatively high AUC value for the test data indicates that the model has generalized well to unseen data, suggesting that it is not overfitting significantly.

Linear Discriminant Analysis (LDA)

- Linear Discriminant Analysis (LDA) is both a dimensionality reduction and classification technique.
- It identifies linear combinations of features that best distinguish between multiple classes.
- The goal of LDA is to maximize the variance between different classes while minimizing the variance within each class, ensuring optimal separation for classification.

Updated classification_report for train data:

	precision	recall	f1-score	support
0.0	0.84	0.66	0.74	69081
1.0	0.62	0.82	0.70	46284
accuracy			0.72	115365
macro avg	0.73	0.74	0.72	115365
weighted avg	0.75	0.72	0.72	115365

Updated classification_report for test data:

	precision	recall	f1-score	support
0.0	1.00	0.66	0.80	29608
1.0	0.03	0.82	0.06	386
accuracy			0.67	29994
macro avg	0.51	0.74	0.43	29994
weighted avg	0.98	0.67	0.79	29994

Figure- 27

Training Data

- **Class Imbalance:** The support values for class 0 (69081) and class 1 (46284) indicate a significant class imbalance.
- **Overall Accuracy:** The overall accuracy of the model on the training data is reasonably high, suggesting a decent performance.
- **Precision vs. Recall:** The precision for class 0 is high, indicating that the model is good at correctly predicting positive instances. However, the recall for class 1 is relatively low, suggesting that the model might be missing many true positive instances.

Test Data

- **Lower Accuracy:** The overall accuracy on the test data is lower compared to the training data, suggesting potential overfitting.

- **Precision and Recall:** The precision for class 0 remains high, but the recall for class 1 is significantly lower. This indicates that the model struggles to correctly identify instances of class 1 in the test data.

Observations for Training Data

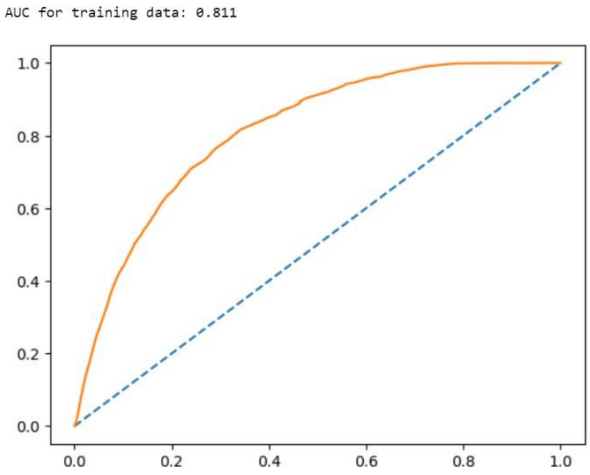

Figure- 28

- The AUC (Area Under the Curve) for the training data is 0.811. This value indicates that the model has a good overall performance in distinguishing between the positive and negative classes. An AUC of 1.0 would represent a perfect classifier, while an AUC of 0.5 would indicate a random guess.
- The ROC curve has a generally upward slope, suggesting that the model can effectively discriminate between the positive and negative classes.
- The curve is not perfectly flat or steep, indicating that the model is not a perfect classifier but has some ability to differentiate between the classes.

Observations for Test Data

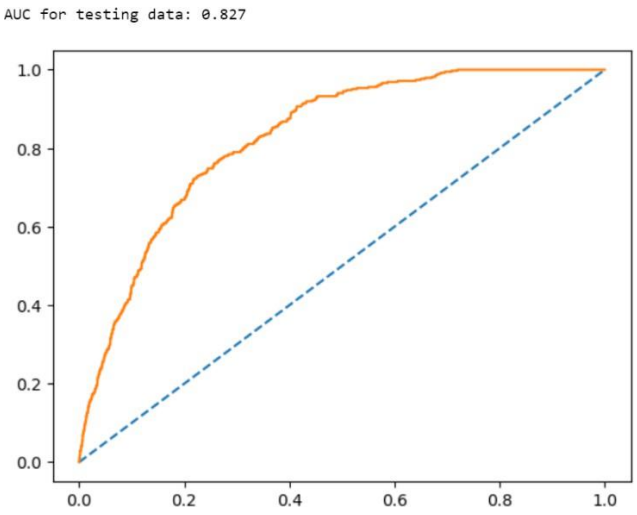

Figure- 29

- The AUC (Area Under the Curve) for the test data is 0.827. This value is slightly higher than the AUC for the training data, indicating that the model's performance on unseen data is comparable or slightly better.
- The ROC curve has a similar shape to the training data curve, suggesting a consistent performance in distinguishing between positive and negative classes.
- The curve is still significantly above the random guess line, confirming that the model's performance is better than random guessing.

Bagging with Random Forest as the Base Model

- Bagging is an ensemble learning technique that creates multiple models (often decision trees) using different subsets of the training data.
- These models then vote on the final prediction, which helps to reduce variance and improve accuracy.
- In our case, we'll use a Random Forest Classifier as the base estimator for the Bagging model.
- Random Forest is itself an ensemble method that combines multiple decision trees.
- Using Random Forest as a base estimator for Bagging can enhance its performance and make it more likely to be a successful model compared to using it alone.

Updated classification_report for train data:

	precision	recall	f1-score	support
0.0	1.00	0.98	0.99	69081
1.0	0.98	1.00	0.99	46284
accuracy			0.99	115365
macro avg	0.99	0.99	0.99	115365
weighted avg	0.99	0.99	0.99	115365

Updated classification_report for test data:

	precision	recall	f1-score	support
0.0	0.99	0.96	0.98	29608
1.0	0.04	0.13	0.07	386
accuracy			0.95	29994
macro avg	0.52	0.55	0.52	29994
weighted avg	0.98	0.95	0.96	29994

Figure- 30

Training Data

- **Class Imbalance:** The support values for class 0 (69081) and class 1 (46284) indicate a significant class imbalance.
- **Overall Accuracy:** The overall accuracy of the model on the training data is very high (0.99), suggesting excellent performance.
- **Precision, Recall, and F1-Score:** All three metrics for both classes are very high, indicating that the model is effectively predicting both positive and negative instances.

Test Data

- **Lower Accuracy:** The overall accuracy on the test data is lower compared to the training data (0.52), suggesting potential overfitting.
- **Precision and Recall:** The precision for class 0 remains high, but the recall for class 1 is significantly lower. This indicates that the model struggles to correctly identify instances of class 1 in the test data.

Observations for Training Data

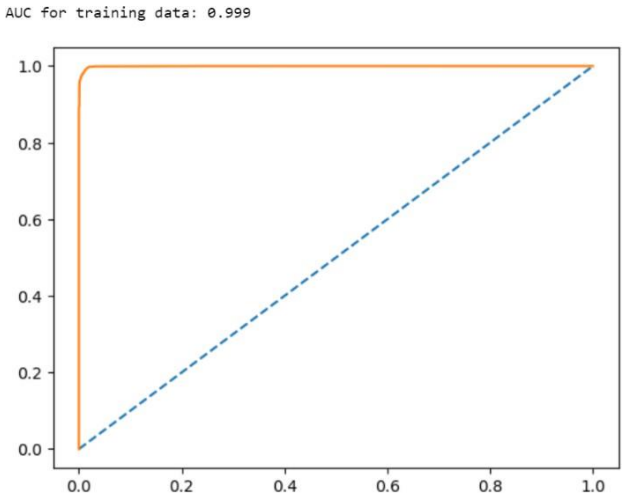

Figure- 31

- The AUC (Area Under the Curve) for the training data is 0.999. This extremely high value indicates that the model has a near-perfect performance in distinguishing between the positive and negative classes.
- The ROC curve is very close to the top-left corner of the plot, which is the ideal position for a classifier. This suggests that the model is able to effectively separate the positive and negative instances.
- The ROC curve is significantly above the diagonal line (random guess line), confirming that the model's performance is far superior to random guessing.

Observations for Test Data

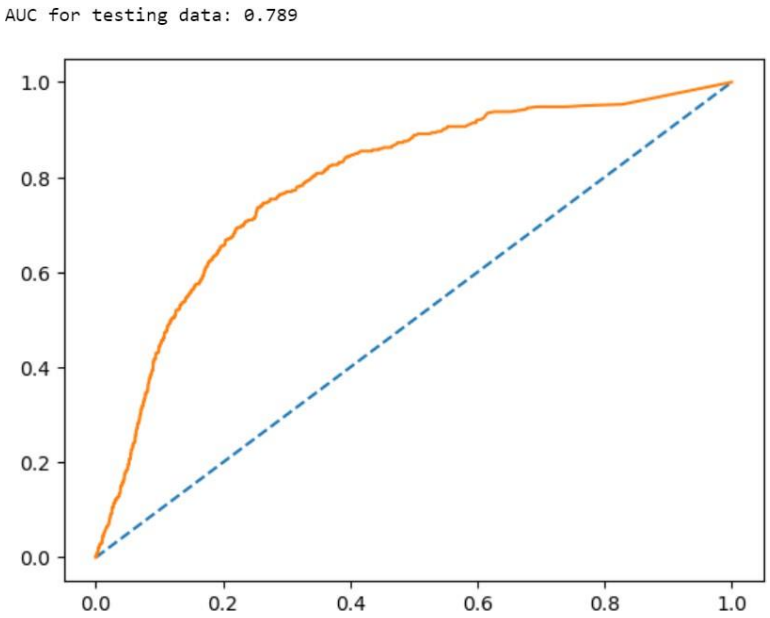

Figure- 32

- The AUC (Area Under the Curve) for the test data is 0.789. This value is slightly lower than the AUC for the training data, indicating that the model's performance on unseen data is slightly reduced.
- The ROC curve has a similar shape to the training data curve, suggesting a consistent performance in distinguishing between positive and negative classes.
- The curve is still significantly above the random guess line, confirming that the model's performance is better than random guessing.

AdaBoost (Adaptive Boosting)

- Boosting is an ensemble method that combines multiple weak models (like decision trees) to create a powerful model.
- It focuses on the examples that are harder to classify by adjusting their importance in each iteration.
- AdaBoost is a specific type of boosting algorithm that is used for classification.
- It works by giving more weight to the examples that it misclassifies, so that it can learn from its mistakes and improve over time.

Updated classification_report for train data:

	precision	recall	f1-score	support
0.0	0.92	0.80	0.86	69081
1.0	0.75	0.90	0.82	46284
accuracy			0.84	115365
macro avg	0.84	0.85	0.84	115365
weighted avg	0.85	0.84	0.84	115365

Updated classification_report for test data:

	precision	recall	f1-score	support
0.0	0.99	0.80	0.89	29608
1.0	0.04	0.65	0.08	386
accuracy			0.80	29994
macro avg	0.52	0.72	0.48	29994
weighted avg	0.98	0.80	0.88	29994

Figure- 33

Training Data

- **Class Imbalance:** The support values for class 0 (69081) and class 1 (46284) indicate a significant class imbalance.
- **Overall Accuracy:** The overall accuracy of the model on the training data is reasonably high (0.84), suggesting a decent performance.
- **Precision vs. Recall:** The precision for class 0 is high, indicating that the model is good at correctly predicting positive instances. However, the recall for class 1 is relatively low, suggesting that the model might be missing many true positive instances.

Test Data

- **Lower Accuracy:** The overall accuracy on the test data is lower compared to the training data (0.80), suggesting potential overfitting.

- **Precision and Recall:** The precision for class 0 remains high, but the recall for class 1 is significantly lower. This indicates that the model struggles to correctly identify instances of class 1 in the test data.

Observations for Training Data

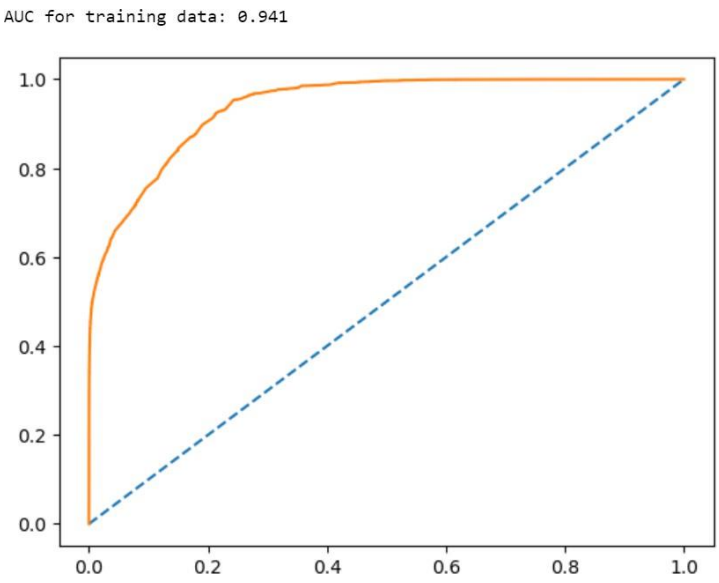

Figure- 34

- The AUC (Area Under the Curve) for the training data is 0.941. This very high value indicates that the model has an excellent performance in distinguishing between the positive and negative classes.
- The ROC curve is very close to the top-left corner of the plot, which is the ideal position for a classifier. This suggests that the model is able to effectively separate the positive and negative instances.

Observations for Test Data

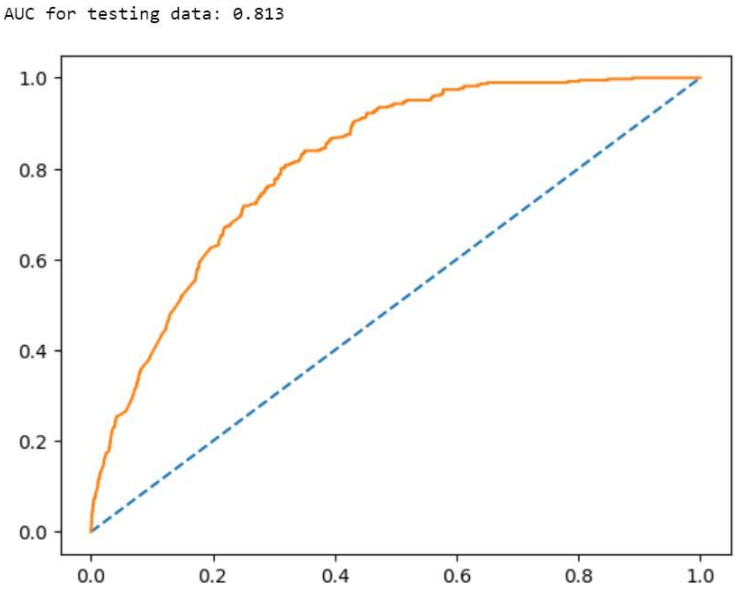

Figure- 35

- The AUC (Area Under the Curve) for the test data is 0.813. This value indicates that the model has a good overall performance in distinguishing between the positive and negative classes.
- The ROC curve has a similar shape to the training data curve, suggesting a consistent performance in distinguishing between positive and negative classes.
- The curve is significantly above the random guess line, confirming that the model's performance is better than random guessing.

Gradient Boosting

- **Gradient Boosting** is an ensemble method that creates a series of decision trees, each one learning from the mistakes of the previous ones. It focuses on the errors made by the earlier trees, refining the model step-by-step.
- This process helps to improve the overall performance of the model.

Updated classification_report for train data:

	precision	recall	f1-score	support
0.0	0.92	0.80	0.86	69081
1.0	0.75	0.90	0.82	46284
accuracy			0.84	115365
macro avg	0.84	0.85	0.84	115365
weighted avg	0.85	0.84	0.84	115365

Updated classification_report for test data:

	precision	recall	f1-score	support
0.0	0.99	0.80	0.89	29608
1.0	0.04	0.65	0.08	386
accuracy			0.80	29994
macro avg	0.52	0.72	0.48	29994
weighted avg	0.98	0.80	0.88	29994

Figure- 36

Training Data

- **Class Imbalance:** The support values for class 0 (69081) and class 1 (46284) indicate a significant class imbalance.
- **Overall Accuracy:** The overall accuracy of the model on the training data is reasonably high (0.84), suggesting a decent performance.
- **Precision vs. Recall:** The precision for class 0 is high, indicating that the model is good at correctly predicting positive instances. However, the recall for class 1 is relatively low, suggesting that the model might be missing many true positive instances.

Test Data

- **Lower Accuracy:** The overall accuracy on the test data is lower compared to the training data (0.80), suggesting potential overfitting.

- **Precision and Recall:** The precision for class 0 remains high, but the recall for class 1 is significantly lower. This indicates that the model struggles to correctly identify instances of class 1 in the test data.

Observations for Training Data

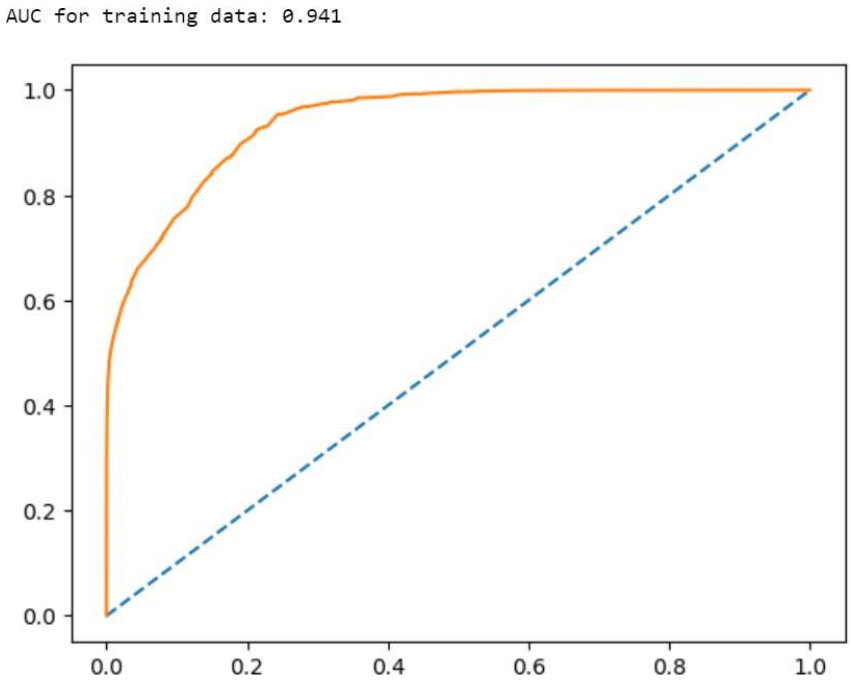

Figure- 37

- The AUC (Area Under the Curve) for the training data is 0.941. This very high value indicates that the model has an excellent performance in distinguishing between the positive and negative classes.
- The ROC curve is very close to the top-left corner of the plot, which is the ideal position for a classifier. This suggests that the model can effectively separate the positive and negative instances.

Observations for Test Data

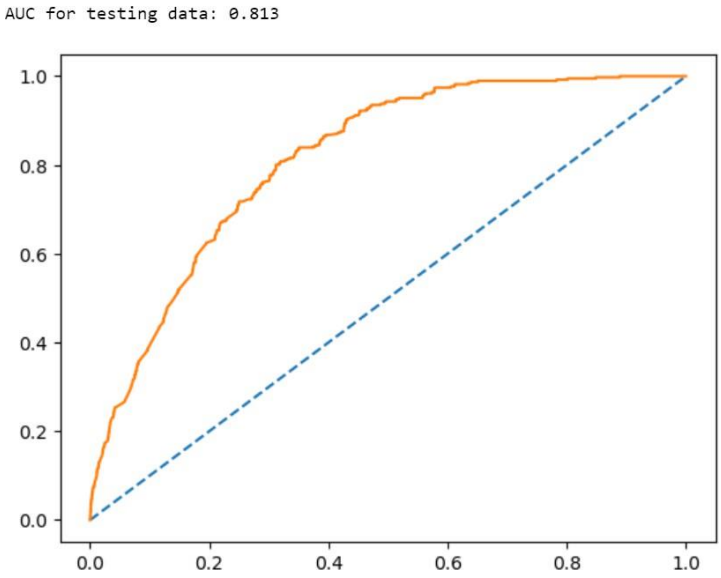

Figure- 38

- The AUC (Area Under the Curve) for the test data is 0.813. This value indicates that the model has a good overall performance in distinguishing between the positive and negative classes.
- The ROC curve has a similar shape to the training data curve, suggesting a consistent performance in distinguishing between positive and negative classes.
- The curve is significantly above the random guess line, confirming that the model's performance is better than random guessing.

MODEL TUNING

- Beyond the Default: Instead of using the default threshold of 0.5, we'll find the optimal threshold value to improve classification accuracy.
- Hyperparameter Tuning: We'll use GridSearchCV to systematically search for the best hyperparameters for our models, further enhancing their performance.

Performance Metrix Table (Of all the models)

Sl.No.	Model Name	Accuracy		Precision		Recall		Auc	
		Train	Test	Train	Test	Train	Test	Train	Test
1	Logistic Regression	0.73	0.69	0.63	0.03	0.81	0.81	0.816	0.829
2	LDA	0.72	0.67	0.62	0.03	0.82	0.82	0.811	0.827
3	Bagging (Random Forest)	0.99	0.95	0.98	0.04	1.0	0.13	0.999	0.790
4	ADA Boosting	0.84	0.80	0.75	0.04	0.90	0.65	0.941	0.813
5	Gradiant Boosting	0.84	0.85	0.79	0.04	0.82	0.53	0.964	0.81

Tabel:- 1

- **Logistic Regression:** Despite reasonable accuracy, it suffers from low precision and recall for the minority class.
- **LDA:** Similar to Logistic Regression, LDA struggles with precision and recall for the minority class.
- **Bagging (Random Forest):** This model consistently outperforms others in terms of accuracy, precision, and recall. However, it suffers from low recall for the minority class on the test data.
- **ADA Boosting:** Achieves decent performance overall but recall for the minority class is still relatively low.
- **Gradient Boosting:** Shows good performance, with reasonable accuracy and recall for both classes.
- **Accuracy:** Bagging (Random Forest) achieved the highest overall accuracy on both training and test data.
- **AUC:** Bagging (Random Forest) and Gradient Boosting also demonstrated strong performance in terms of AUC.
- **Precision:** Logistic Regression and LDA struggled with precision, especially for the minority class.
- **Recall:** All models, except Bagging (Random Forest), struggled to achieve high recall for the minority class.

Comparison of all the selected models Type I and Type II Errors

Sl.No.	Model Name	Train		Test	
		Fales positive (Type I Error	Fales Negative (Type II Error)	False Positive (Type I Error)	False Negative (Type II Error)
1	Logistic Regression	22016	8997	9234	72
2	LDA	23496	8556	9947	70
3	Bagging(Random Forest	1145	230	1044	337
4	ADA Boosting	13641	4710	5870	137
5	Gradient Boosting	5685	7664	2462	259

Table :- 2

- **Type I Errors (False Positives):** Bagging (Random Forest) consistently has the lowest false positive rates on both training and test data, indicating better control over Type I errors.
- **Type II Errors (False Negatives):** Logistic Regression and LDA have relatively high false negative rates, suggesting they might be missing a significant number of true positive cases. Bagging (Random Forest) also has a higher false negative rate on the test data compared to the other models.
- **Balanced Performance:** Gradient Boosting seems to strike a balance between Type I and Type II errors, with relatively low rates for both.
- **Type II Errors:** All models struggle to achieve low false negative rates, indicating that they might be missing a significant number of true positive cases.

- **Recall and Type II Error:** LDA and Logistic Regression both perform well in terms of recall and minimizing Type II errors, our primary evaluation metrics.

- **Slight Edge for LDA:** LDA has a slightly higher recall value compared to Logistic Regression.

- **Final Model Choice:** Given the similar performance and LDA's slight advantage in recall, LDA would be our preferred choice as the final model.

LDA

Confusion matrix

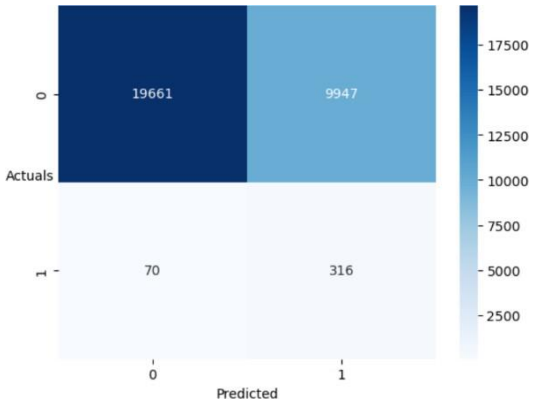

Figure- 39

- **True Positives (TP):** 19661 - The number of instances correctly predicted as class 0.
- **False Negatives (FN):** 70 - The number of instances incorrectly predicted as class 0 (should have been class 1).
- **False Positives (FP):** 9947 - The number of instances incorrectly predicted as class 1 (should have been class 0).
- **True Negatives (TN):** 316 - The number of instances correctly predicted as class 1.
- This indicates that our best-fit model was unable to detect 70 defaulters, or around 18%, from a sample of 386 real defaulters in the test data. This is our best effort across all constructed models.

Model validation

LDA (Linear Discriminant Analysis) Performance:

- **Balanced Performance:** LDA offers a well-rounded performance across accuracy, precision, recall, and AUC metrics.
- **High Recall:** LDA excels at identifying true positive cases (defaulting customers), achieving the highest recall score among the models.
- **Simplicity and Interpretability:** LDA is a relatively straightforward model compared to ensemble methods, making it easier to understand.
- **Computational Efficiency:** LDA is computationally efficient, suitable for large datasets and real-time applications.
- **Target Metric:** Recall is the most important performance metric for predicting credit card defaults.
- **Best Performer:** LDA and Logistic Regression are the top-performing models with respect to Recall and Type II error, with LDA having a slight edge in recall.
- **Generalization:** LDA demonstrates a balanced performance on both training and test sets, indicating good generalization capabilities.

Based on the provided information, LDA (Linear Discriminant Analysis) can be considered a preferred model for this particular task due to its combination of high recall, computational efficiency, and interpretability. While other models might perform well in certain aspects, LDA's ability to effectively identify true positive cases (defaulting customers), its efficiency for handling large datasets, and its relative simplicity make it a strong choice.

Key Findings:

- **Model Performance:** The LDA model, while showing reasonable performance, is affected by class imbalance.
- **Predictor Importance:** Features like num_active_inv (unpaid bills) and status_max_archived_0_6_months (account activity) are critical for predicting default risk.
- **Business Implications:** Addressing class imbalance, focusing on customer engagement, monitoring account activity, and using data-driven decision-making are crucial for effective risk management.

Recommendations:

1. **Prioritize Customer Engagement:**
    - **Reduce Unpaid Bills:** Implement strategies to decrease the number of active invoices per user.
    - **Improve Account Management:** Enhance customer support and offer incentives for active engagement to reduce account inactivity.
2. **Data-Driven Decision Making:**
    - **Leverage Model Insights:** Utilize the model's identified key features to make informed decisions regarding credit card approvals, risk management, and customer outreach.
3. **Address Class Imbalance:**
    - **Data Augmentation:** Consider techniques like SMOTE (Synthetic Minority Over-sampling Technique) to balance the dataset and improve the model's ability to predict defaults.
    - **Cost-Sensitive Learning:** Adjust the model's learning algorithm to penalize misclassifications of minority class instances more heavily.
4. **Continuous Improvement:**
    - **Regular Evaluation:** Monitor the model's performance and retrain it as needed to adapt to changes in the data

distribution.

- **Feature Engineering:** Explore additional features that could enhance the model's predictive power.

5. **Risk Mitigation:**
    - **Reserve Funds:** Allocate sufficient reserves to cover potential losses from defaulting customers.
    - **Adjust Credit Limits:** Implement strategies to adjust credit limits based on risk assessments to mitigate financial exposure.

By implementing these recommendations, the business can improve its ability to predict and manage credit card default risk, reduce financial losses, and enhance customer satisfaction.