**Automobile Loan Default Prediction:**

**Developing a Model to Identify High-Risk Borrowers**

*A project report submitted to ICT Academy of Kerala*

*in partial fulfillment of the requirements*

*for the certification of*

**CERTIFIED SPECIALIST**

**IN**

**DATA SCIENCE & ANALYTICS**

submitted by

**Adarsh T Saji**

**Aathira V Sajive**

**Gretty Jaison**

**Sruthi S S**

**Thoufira P C**

**Thushara K S**



**ICT ACADEMY OF KERALA**

**THIRUVANANTHAPURAM, KERALA, INDIA**

**July 2023**

**List of Figures**

**Fig 5.1.1 :** Distribution Plot

**Fig 5.1.2 :** Count Plot

**Fig 5.1.3 :** Count Plot of categorical columns

**Fig 5.1.4 :** Pie Chart (target variable)

**Fig 5.2.1 :** Bivariate analysis with Client Income

**Fig 5.2.2.a :** Bivariate analysis of Default against numerical columns

**Fig 5.2.2.b :** Bivariate analysis using barplots

**Fig 5.3.1 :** Heatmap

**Fig 5.3.2.a :** Multivariate Bar plots

**Fig 6.3.1 :** Outlier Detection Using Box Plots

**Fig 9.1 :** Website hosting

**Fig 9.2 :** Website hosting result

**List of Abbreviations**

1. KNN : k-Nearest Neighbors
2. SVM : Support Vector Machine
3. EDA : Exploratory Data Analysis
4. MAD : Mean Absolute Deviation
5. IQR : Interquartile Range
6. DT : Decision Tree
7. ML : Machine Learning
8. RF : Random Forest
9. CV : Cross-Validation
10. HTML : HyperText Markup Language
11. CSS : Cascading Style Sheets

**Table of Contents**

**Page No.**

**Abstract 5**

**1. Problem Definition 6**

**1.1 Overview 6**

**1.2 Problem Statement 7**

**2. Introduction 8**

**3. Literature Survey 9**

**4. Dataset Description 10**

**5. Exploratory Data Analysis 11**

**6. Data Preprocessing 25**

**7. Model Building 38**

**8. Hyperparameter Tuning 42**

**9. Website Hosting 43**

**10. Result 45**

**11. Conclusion 46**

**References 47**

**Abstract**

In a world driven by data, the need to come up with more efficient and faster machine learning models is becoming more important as each day passes. The exponential growth in the demand for automated decision making has only fuelled this need. One prominent task that falls within this domain is the Classification problem.The objective of this project is to come up with a simple but effective Classification model for The automobile loan default dataset. This model can be used within the Banking sector to tackle the problem of whether a person is capable of repaying automobile loans.

The Automobile Loan Default Dataset provides information whether a customer might default if a loan is provided. This dataset aims to facilitate an in-depth analysis of the factors influencing loan defaults and enable the development of predictive models to identify potential defaulters. The problem at hand involves splitting a dataset into Train and Test datasets to address the task effectively. Model building is to be done on Train Dataset and the Model testing is to be done on Test Dataset. The ultimate goal is to predict the "Default" level, indicating whether an automobile loan is likely to result in default.

Our approach tries to integrate some of the most prominently used classification algorithms tailored for our problem which is the classification whether a person might default on automobile loans or not. Our model aims to achieve superior performance in terms of Accuracy score, precision score, recall score and f1 score

**1. Problem Definition**

**1.1 Overview**

Defaulting on a loan refers to the failure to make the required payments as per the loan agreement. It can have serious consequences for both the borrower and the lender. By developing a reliable classification model, we aim to identify individuals who are at a higher risk of defaulting on their automobile loans.

To achieve this, we need to analyze a dataset containing relevant features or variables that can help us predict loan default. These features can include demographic information, financial indicators, credit history, employment details, and other relevant factors.

The ultimate goal is to create a model that can accurately classify individuals into two categories: those who are likely to default on their automobile loans and those who are not. This model can be used by lenders and financial institutions to assess the creditworthiness of loan applicants, make informed decisions, and minimize potential losses due to defaults.

To accomplish the task, we will employ machine learning techniques and algorithms to train a classification model using historical data. We will evaluate the model's performance, fine-tune it as necessary, and ultimately deploy it to make predictions on new, unseen data.

By solving this problem, we can help lenders improve their risk assessment processes, make informed lending decisions, and mitigate potential financial risks associated with automobile loans.

**1.2 Problem Statement**

The problem statement focuses on determining whether a given individual is capable of repaying an automobile loan based on various factors. The goal is to build a predictive model that can accurately classify individuals as either capable or not capable of repaying the loan.

To address this problem, we need to consider multiple factors or features that can provide insights into an individual's repayment capacity. These factors may include demographic information (such as age, gender, and marital status), financial indicators (such as income and expenses), credit history (including credit score and previous loan repayment behavior), employment details (such as job stability and income source), and potentially other relevant factors.

The predictive model will be trained on historical data that includes information about individuals who have previously taken automobile loans, including whether they successfully repaid the loan or defaulted. By analyzing this data, we aim to identify patterns and relationships between the available features and the loan repayment outcome.

The model's task is to learn from these patterns and relationships and then apply that knowledge to predict the loan repayment capability for new, unseen individuals. This prediction will be based on the features of the individual, providing a measure of their creditworthiness.

The successful development of such a predictive model can have significant benefits for lenders and financial institutions. It can help them make more informed decisions when evaluating loan applications, manage risk effectively, and minimize potential losses due to loan defaults. Additionally, it can assist individuals by providing insights into their own repayment capacity and enabling them to make more informed financial decisions.

To achieve the goal, we will utilize machine learning algorithms and techniques, such as classification algorithms, feature engineering, and model evaluation. The model will be trained, validated, and fine-tuned using appropriate data splitting and cross-validation techniques. The performance of the model will be assessed based on various evaluation metrics, such as accuracy, precision, recall, and F1 score.

By successfully addressing this problem, we aim to provide lenders and individuals with a reliable tool for assessing loan repayment capability, enhancing the overall loan approval process, and promoting responsible lending practices.

**2. Introduction**

In a world where the automobile sector plays a pivotal role in enhancing the growth of the society, by providing individuals with more efficient and convenient ways to commute, the demand for financing options such as automobile loans for acquiring these vehicles is also on the rise. While automobile loans provide a pathway toward vehicle ownership, it also provides financial institutions the challenge of assessing the creditworthiness of each loan applicant.

Predicting whether an applicant might default on their automobile loan is a critical problem faced by most banks and other lending organisations.Default prediction involves analyzing various factors and patterns in historical loan data to build a reliable model capable of accurately classifying potential defaulters. This is where the power of machine learning comes into play.

The dataset that we use for this project is from a banking sector, the datasets contains information such as occupation, income, education, family members,etc. which can be used to identify whether an applicant is likely to Default.Training a machine learning model using comprehensive historical data can empower financial institutions to proactively address potential loan defaulters with greater precision and confidence.

In this project, we aim to harness the potential of machine learning algorithms to develop an accurate and robust classification model that can effectively identify individuals who might be at risk of defaulting on their automobile loans. By doing so, we can help financial institutions make informed decisions, minimize financial losses, and ensure responsible lending practices.

**3. Literature Survey**

Loan default is a major problem for financial institutions, as it can lead to significant losses. In recent years, there has been a growing interest in using data mining techniques to predict loan defaults. This is because data mining can be used to identify patterns in data that may not be easily visible to the naked eye.

There have been a number of studies that have used data mining techniques to predict loan defaults. One of the most well-known studies is the study by Tariq et al. (2016). This study used a data mining methodology called SEMMA to develop a model to predict loan defaults. The study found that the SEMMA model was able to achieve an accuracy of 85% in predicting loan defaults.

Another study that has used data mining techniques to predict loan defaults is the study by Madaan et al. (2017). This study compared the performance of two machine learning algorithms, Random Forest and Decision Trees, in predicting loan defaults. The study found that the Random Forest algorithm was able to achieve a higher accuracy than the Decision Trees algorithm.

Finally, the study by Kwofie et al. (2018) used logistic regression to predict the probability of loan default in a microfinance company. The study found that the logistic regression model was able to explain 60% of the variability in loan default status.

The literature review shows that there is a growing body of research on the use of data mining techniques to predict loan defaults. The studies that have been conducted have shown that data mining can be a valuable tool for predicting loan defaults. However, there is still more research that needs to be done in this area.

**4. Data Set Description**

The Dataset used for thisproject contains financial information about clients who had applied for an automobile loan and whether they had defaulted. This dataset belongs to a banking sector that is struggling to mark profits due to an increase in defaults in the vehicle loan category. The company aims to determine the client’s loan repayment abilities and understand the relative importance of each parameter contributing to a borrower’s ability to repay the loan.

The Dataset contains 40 columns with 39 columns containing information that can be used to measure the capability of a client to repay the loan.

Description for each columns:

* ID: Client Loan application ID.
* Client\_Income: Client Income in dollars($).
* Car\_Owned: Any Car owned by client before applying for the loan for another car (0 means No and 1 means otherwise).
* Bike\_Owned: Any bike owned by client (0 means No and 1 means otherwise).
* Active\_Loan: Any other active loan at the time of application of loan (0 means No and 1 means otherwise).
* House\_Own: Any house owned by client (0 means No and 1 means otherwise).
* Child\_Count: Number of children the client has.
* Credit\_Amount: Credit amount of the loan in dollars($).
* Loan\_Annuity: Loan annuity in dollars($).
* Accompany\_Client: Who accompanied the client when the client applied for the loan.
* Client\_Income\_Type: Clients income type.
* Client\_Education: Highest level of education achieved by client.
* Client\_Marital\_Status: Marital status of client (D- Divorced, S- Single, M- Married, W- Widowed).
* Client\_Gender: Gender of the Client.
* Loan\_Contract\_Type: Whether the loan is Cash Loan(CL) or Revolving Role(RL).
* Client\_Housing\_Type: Client Housing situation.
* Population\_Region\_Relative: Relative population of the region where the client is living. Higher value means the client is living in a more populated area.
* Age\_Days: Age of the client at the time of application submission.
* Employed\_Days: Days before the application, the client started earning.
* Registration\_Days: Days before the loan application, the client changed his/her registration.
* ID\_Days: Days before the loan application, the client changed his/her identity document with which the loan was applied.
* Own\_House\_Age: Age of Client's house in years.
* Mobile\_Tag: Mobile Number provided by Client (1 means Yes and 0 means No).
* Homephone\_Tag: home phone Number provided by Client (1 means Yes and 0 means No).
* Workphone\_Working: Was work phone number reachable (1 means Yes and 0 means No).
* Client\_Occupation: Client Occupation type.
* Client\_Family\_Members: Number of family members a client has.
* Cleint\_City\_RatingL Client city rating. 3 denotes best and 2 denotes good and 1 denotes average.
* Application\_Process\_Day: Day of the week on which the client applied for the loan (0-Sun, 1-Mon,2-Tues, 3-Wed, 4-Thrus,5-Fri, 6-Sat).
* Application\_Process\_Hour: hour of the day on which client applied for the loan.
* Client\_Permanent\_Match\_Tag: Indication if client contact address does not match permanent address.
* Client\_Contact\_Work\_Tag: Indication if client work address does not match contact address.
* Type\_Organization: Type of organization where a client works.
* Score\_Source\_1: Score sourced from another source. This is a normalized score.
* Score\_Source\_2: Score sourced from another source. This is a normalized score.
* Score\_Source\_3: Score sourced from another source. This is a normalized score,
* Social\_Circle\_Default: How many friends/family members of clients defaulted on any loan payment in the last 60 days.
* Phone\_Change: How many days before the loan application, the client changed his/her phone.
* Credit\_Bureau: Total number of enquiries in last year
* Default: 1 means the client defaulted on loan payments and 0 means otherwise (Target variable).

**5. Exploratory Data Analysis**

Exploratory Data Analysis. It is a critical initial step in the data analysis process, where data analysts and data scientists examine, summarize, and visualize the main characteristics of a dataset. The primary goal of EDA is to gain insights into the data, understand its underlying structure, and identify patterns, trends, anomalies, and relationships between variables.

In this project we perform EDA on all records that are significant for a person to default. EDA is basically performed by plotting various charts to identify any pattern or insights that might be during preprocessing and model creation. Exploratory Data Analysis (EDA) was conducted in three comprehensive stages: Univariate analysis, Bivariate analysis, and Multivariate analysis. Each phase involved distinct techniques to uncover essential insights into the data's individual characteristics, inter-variable relationships, and complex interactions. We employed Distribution plots, box plots, bar plots, heatmaps, scatterplots to better visualize the distributions and relationships.

Overall, the EDA helped to identify important trends and patterns in the data, which informed subsequent analysis and modeling efforts. By gaining a deeper understanding of the data, we were able to develop more accurate predictive models and help financial institutions to decide whether automobile loans should be provided for a particular client.

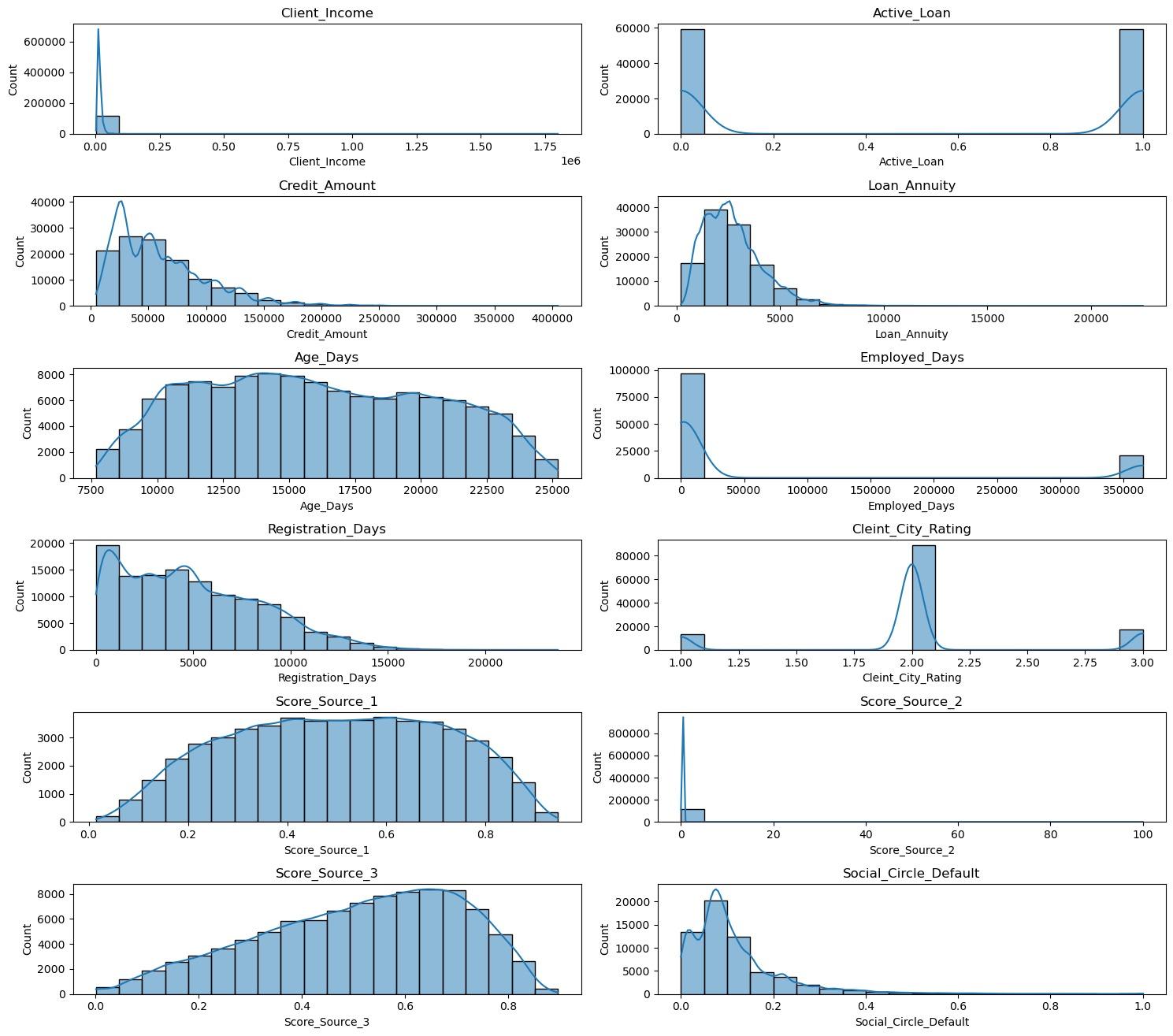
**5.1. Univariate Analysis**

Univariate analysis is a type of statistical analysis that focuses on examining and describing a single variable at a time. It is used to understand the distribution, central tendency, variability, and other important characteristics of a single variable without considering the relationship with other variables. Univariate analysis is often the first step in data analysis and is useful for gaining insights into the individual features of the dataset.

Univariate analysis is useful when it comes to studying individual variables, as it can be used to understand the distribution if it is a numerical variable or the frequency of occurrence if it is a categorical variable. Outliers can also be identified during these analyses.

In this dataset we perform Univariate analysis on every column other than the ID column as it is not essential for building our classification model. A use case for this mode of analysis can be seen if we look into the distribution plot of Client\_Income where we can find that it is a left skewed dataset; all possible outliers can also be identified from this plot. Before plotting the graphs we had initially identified and separated all the numerical and categorical columns for the ease of plotting and analysis. The plots are shown below:

**5.1.1 Distribution Plot**

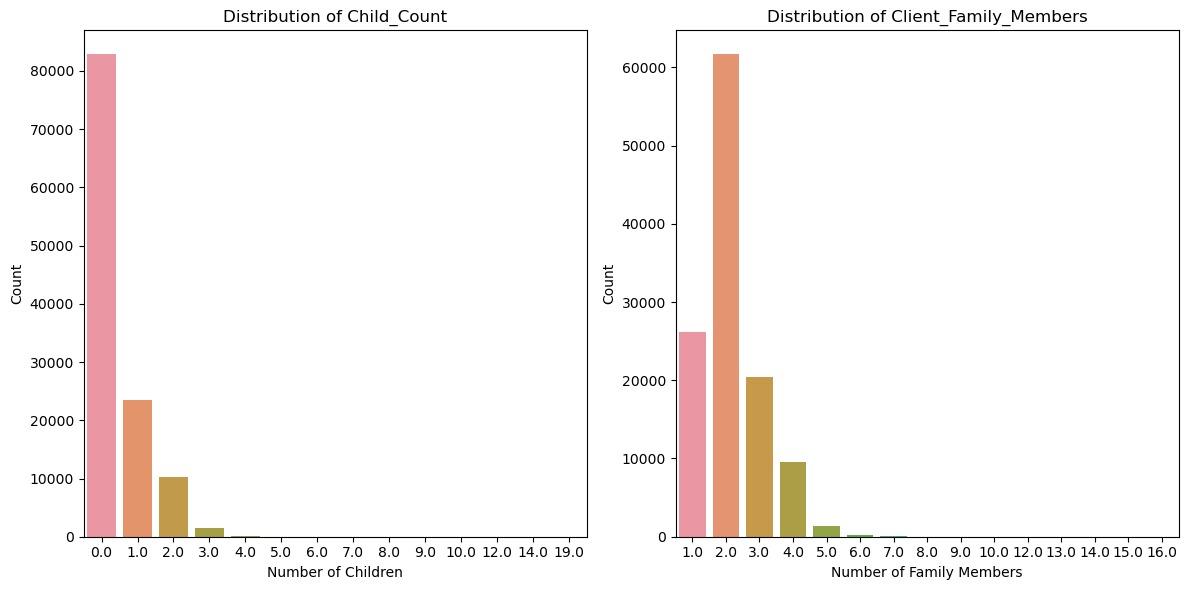


**Fig 5.1.1**

The inferences from the above plots are:

* Client\_Income: The distribution of client income is right-skewed, indicating that a majority of clients have lower incomes.
* Active\_Loan: Most clients have an active loan, as indicated by a spike in the "1" category. The distribution is imbalanced.
* Credit\_Amount: The distribution of credit amounts is heavily right-skewed, indicating a higher concentration of lower credit amounts.
* Loan\_Annuity: The distribution of loan annuities appears to be right-skewed, with higher annuities being less frequent.
* Age\_Days: The distribution of client age in days shows a somewhat uniform distribution.
* Employed\_Days: The distribution of days employed by clients is right-skewed, with a concentration of clients employed for fewer days.
* Registration\_Days: The distribution of registration days shows a similar pattern to the Employed\_Days variable.
* Cleint\_City\_Rating: The majority of clients have a city rating around 2.0 to 2.5, with fewer clients having lower or higher ratings.
* Score\_Source\_1, Score\_Source\_2, Score\_Source\_3: These variables have similar distributions, but Score Source 3 is more variable than the other two sources. Score Source 1 is the most popular scoring source.

**5.1.2 Count Plot**

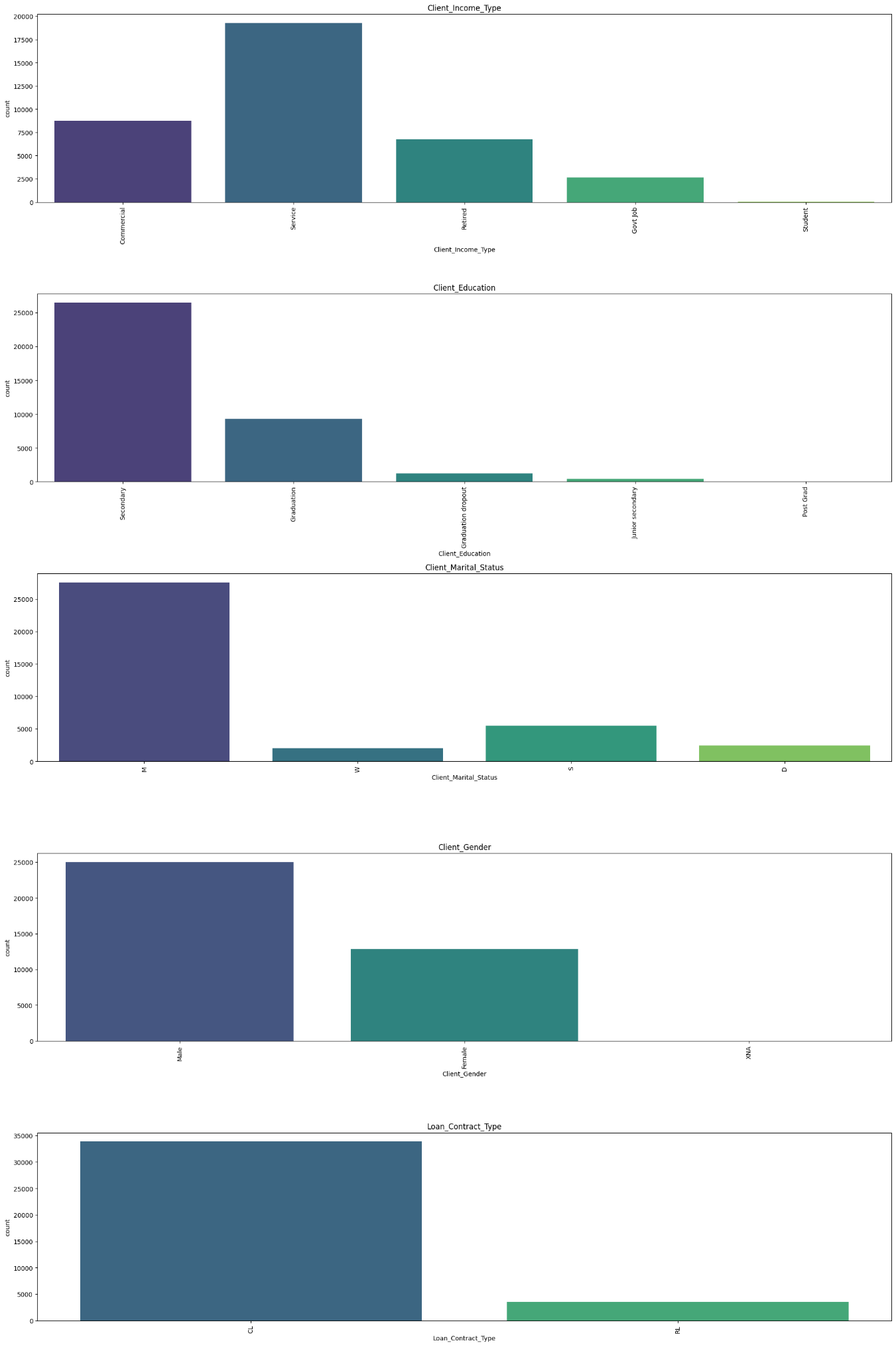
****

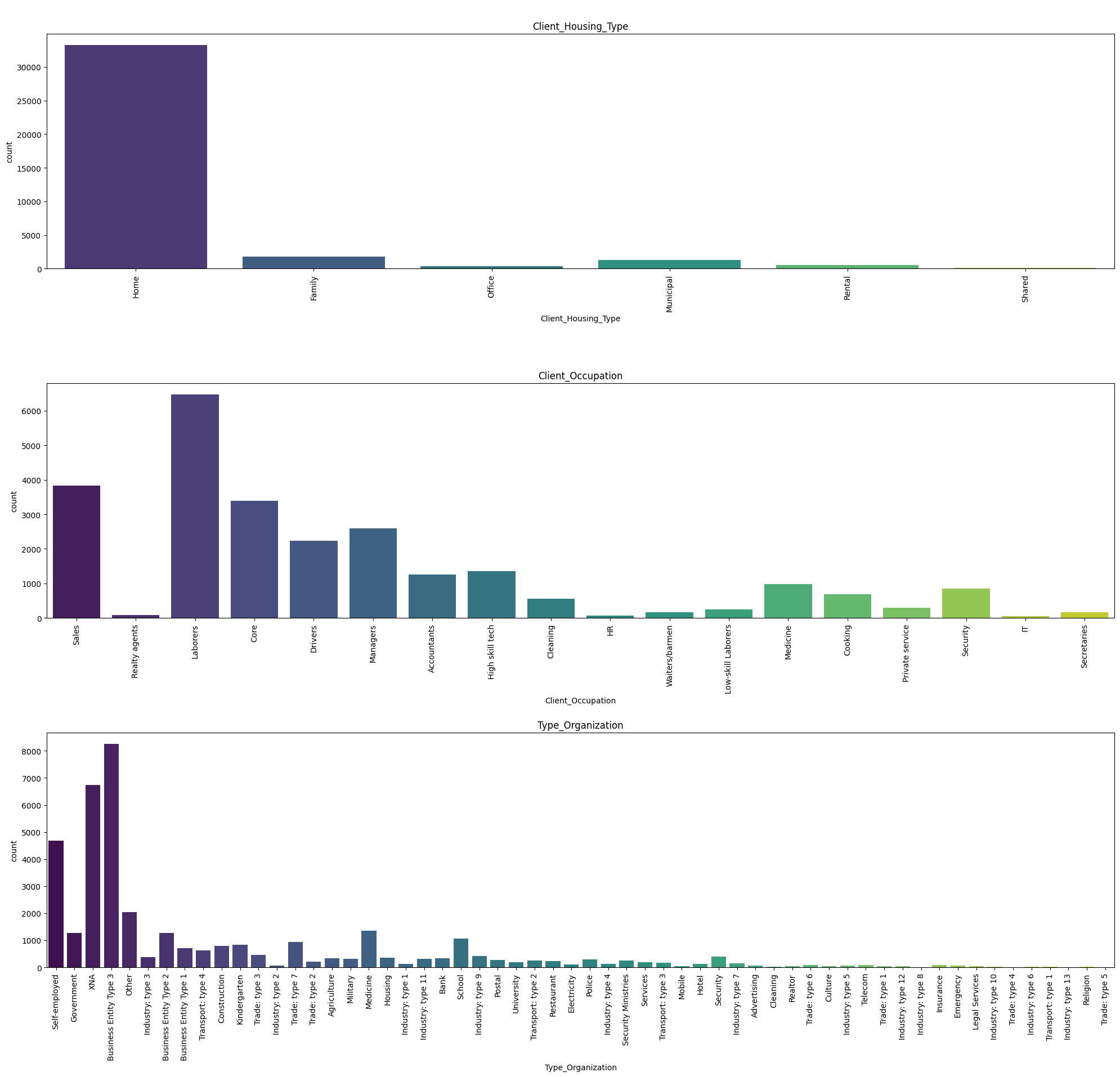
**Fig 5.1.2**

Inference :

* Child\_Count: The majority of clients have no children, with a few having one or more children.
* Client\_Family\_Members: The majority of clients have 1 or 2 family members, with a few having more.

**5.1.3 Count Plot of categorical columns**

****

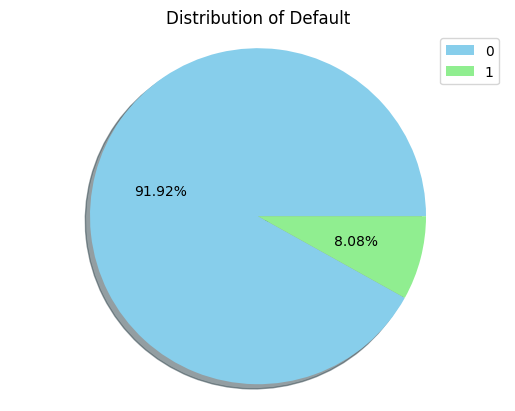
****

**Fig 5.1.3**

The inference from the count plots are:

* Client\_Income\_Type: Most clients have income from "Working" or "Commercial associate" sources.
* Client\_Education: A significant number of clients have completed Secondary / secondary special education, followed by Higher education
* Client\_Marital\_Status: The majority of clients are married or single. Client\_Gender: There are more female clients than male clients.
* Loan\_Contract\_Type: Most loans have a "Cash loans" contract type, with fewer "Revolving loans."
* Client\_Housing\_Type: A significant number of clients live in House/apartment housing type.
* Client\_Occupation: The majority of clients have "Laborers" as their occupation.
* Type\_Organization: Most clients are associated with "Business Entity Type 3" or "Self-employed" organizations.

**5.1.4 Pie Chart (target variable)**

****

**Fig 5.1.4**

Inference from the pie chart is:

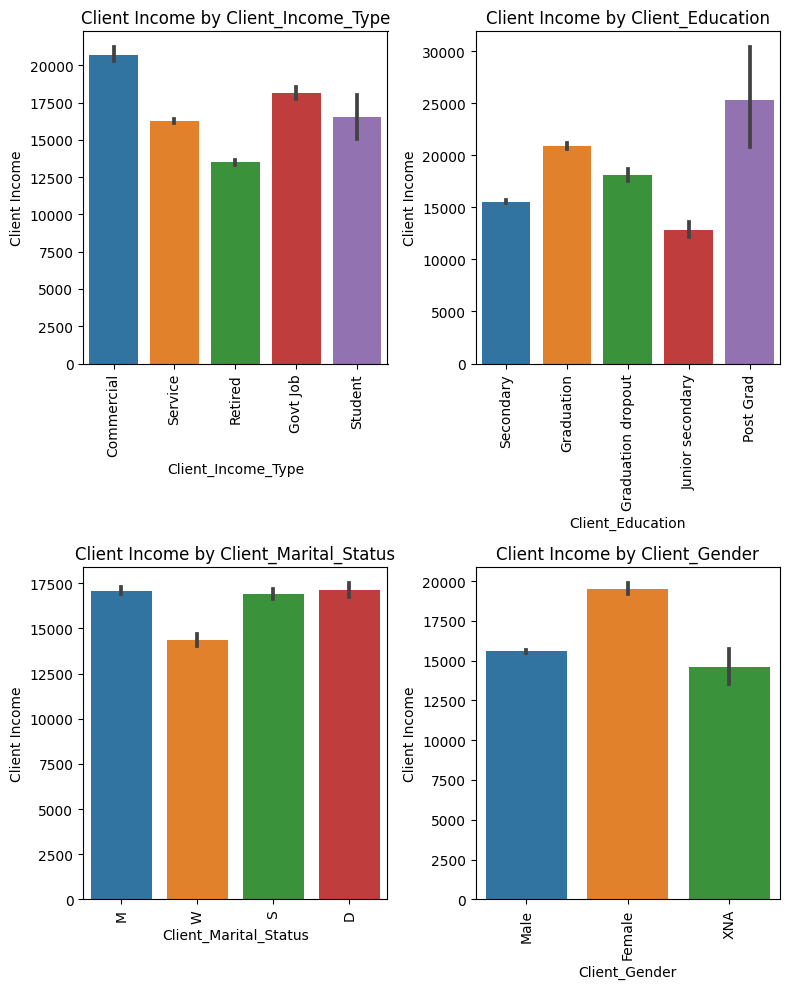
* Unbalanced distribution with 8.08% defaults and 91.92% no defaults, so a Model which predicts no default for all records will have 92% accuracy

**5.2 Bivariate Analysis**

Bivariate analysis is a statistical analysis technique that involves studying the relationship between two variables simultaneously. Unlike univariate analysis, which focuses on a single variable, bivariate analysis examines how two variables are related or interact with each other. It is a crucial step in understanding the association, correlation, or dependency between two variables in a dataset.

In this project we had initially identified important features that might provide great insights into the data we have. so the bivariate analysis is carried forward by plotting these features against each other.

**5.2.1 Bivariate analysis with Client Income**

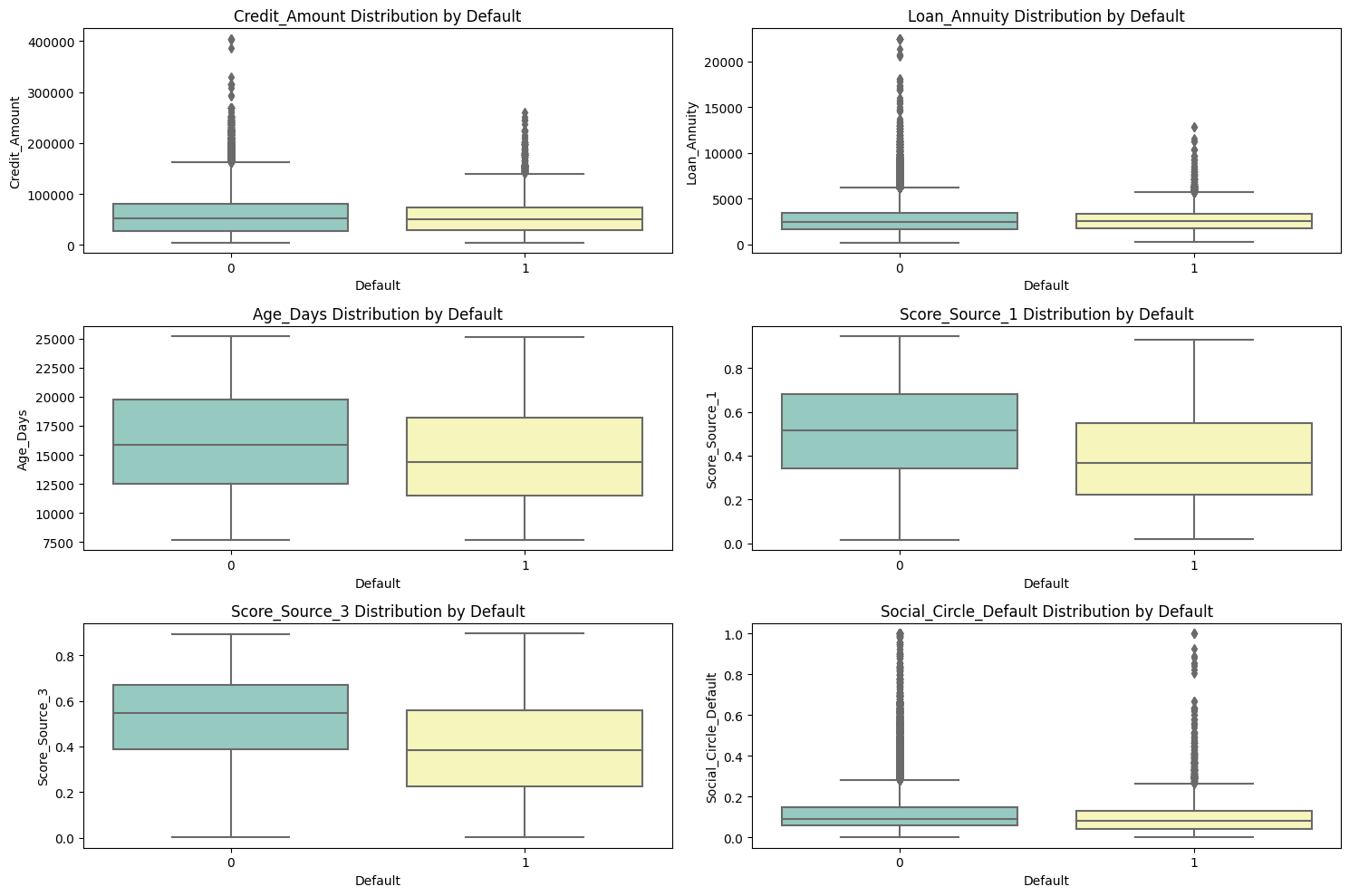
****

**Fig 5.2.1**

Inferences from these barplots are:

* Maternity\_leave tends to be at higher average income but with large variation between minimum and maximum values.
* Unemployed has the lowest average income
* Clients with post graduation tend to have higher average income while junior\_secondnary education has the lowest Average\_Income.
* Female clients have higher average income.

**5.2.2.a Bivariate analysis of Default against numerical columns**

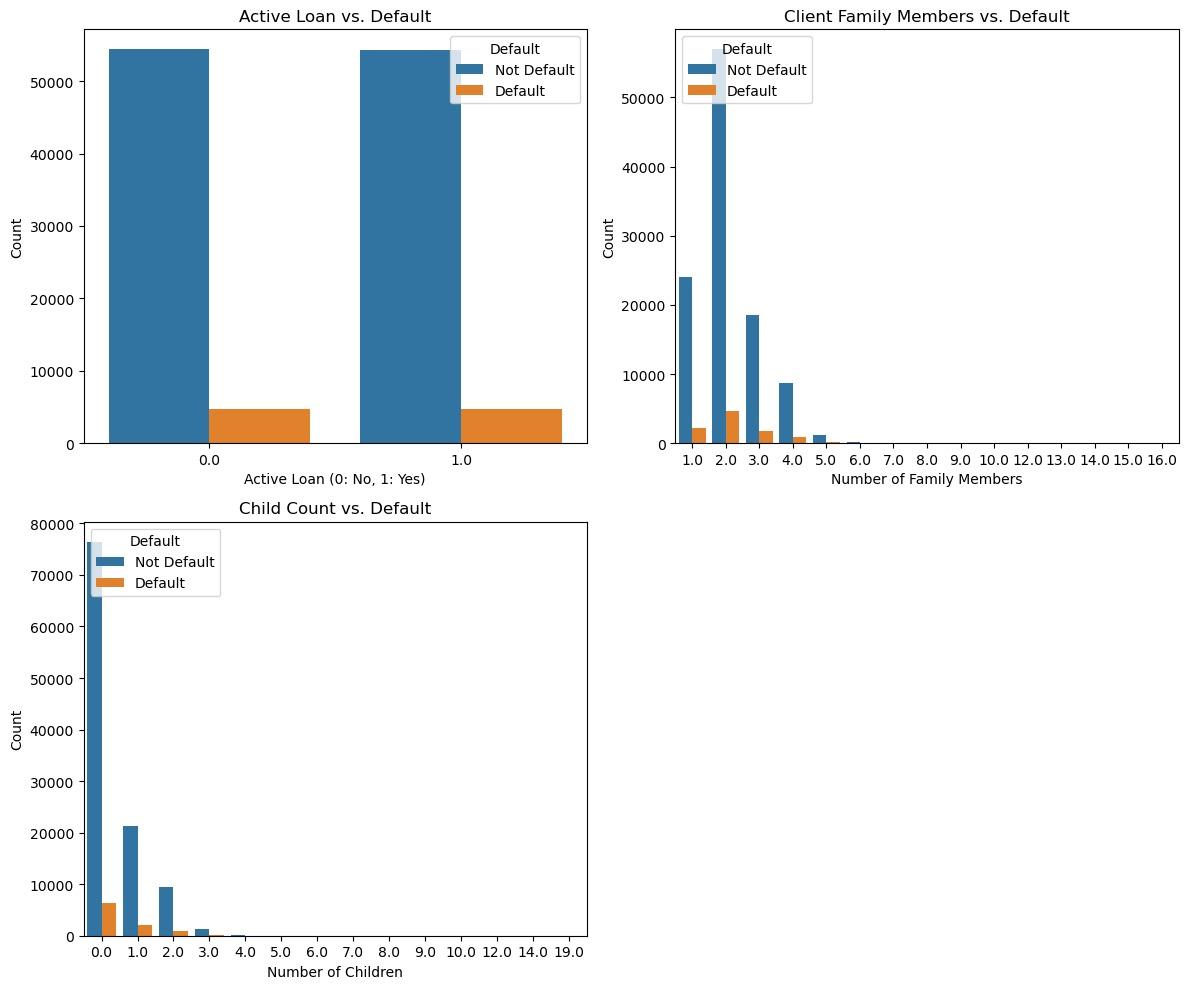
****

**Fig 5.2.2.a**

Inferences from the box plots:

* Credit\_Amount Distribution by Default: Clients who have taken larger credit amounts are more likely to default. This is consistent with the intuition that larger loans represent higher financial obligations, and clients might struggle to repay them.
* Loan\_Annuity Distribution by Default: Clients with higher loan annuity payments have a higher probability of defaulting. Higher annuity payments indicate larger monthly loan obligations, and clients may find it challenging to manage such high payments consistently.
* Age\_Days Distribution by Default: The Age\_Days variable could be a useful predictor of whether a client is likely to default on a loan payment. Clients who are younger or older than the majority may be more likely to default.
* Score\_Source\_1 Distribution by Default: Clients with Source Score 1 increases, the likelihood of default increases. It suggests that "Score\_Source\_1" (which likely represents a credit score or risk assessment) is a useful predictor of loan default, with higher scores indicating higher credit risk.
* Score\_Source\_3 Distribution by Default: Clients with higher values of "Score\_Source\_3" are more likely to default. Similar to "Score\_Source\_1," "Score\_Source\_3" seems to be another relevant predictor of default behavior, with higher scores indicating higher credit risk.
* Social\_Circle\_Default Distribution by Default: The "Social\_Circle\_Default" feature is a useful predictor of whether a client is likely to default on a loan payment.

**5.2.2.b Bivariate analysis using barplots**



**Fig 5.2.2.b**

Inferences from the bar plots::

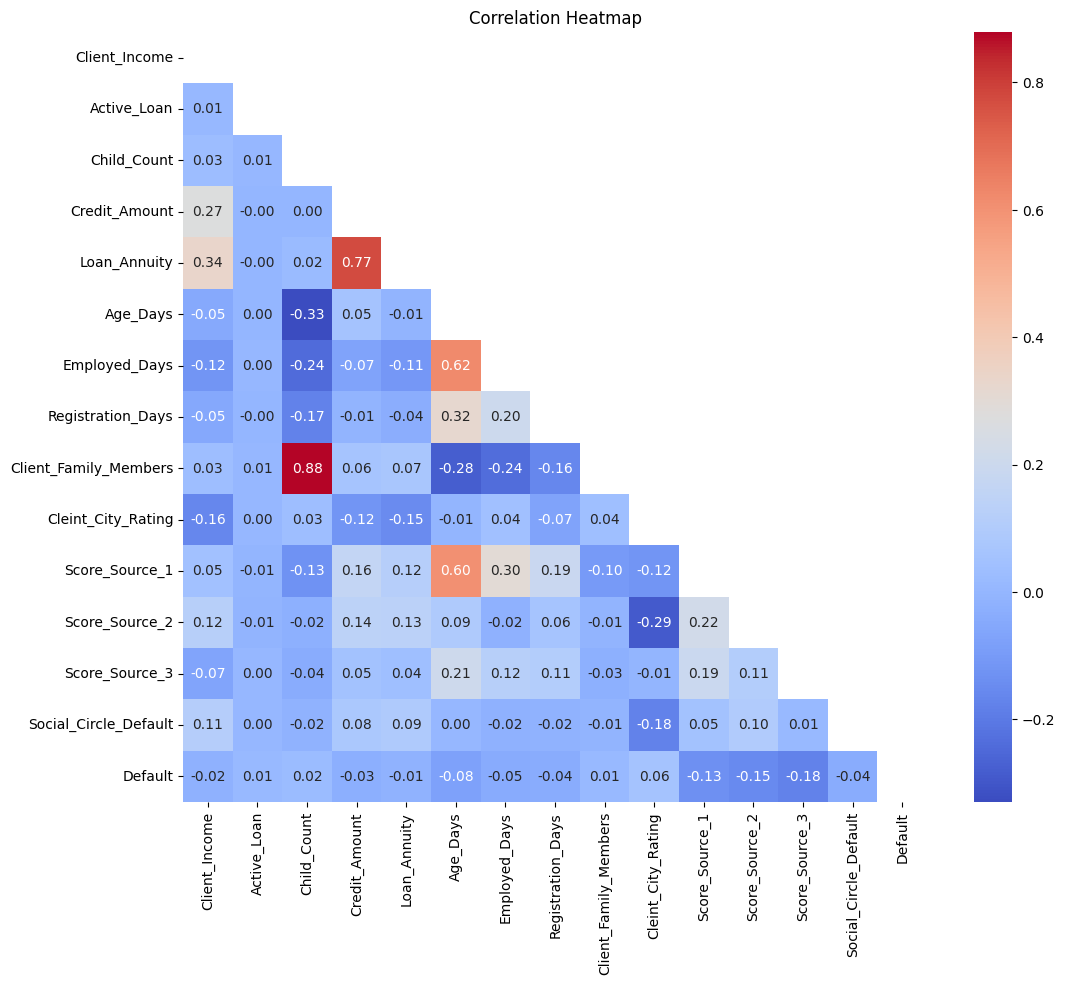
* Client\_Family\_Members by Default: In "Client\_Family\_Members" feature, Clients who have more family members by default may be more likely to default, as they may have more financial obligations or may be more likely to be exposed to financial hardship.
* Child\_Count Distribution by Default: The number of children a client has does not seem to have a strong impact on the likelihood of default. The distribution of defaulters and non-defaulters appears to be relatively similar across different child count categories.
* Active\_Loan Distribution by Default: Clients who have an active loan are more likely to default on their loans compared to those without an active loan. This suggests that having multiple ongoing loans might increase the risk of defaulting.

**5.3 Multivariate Plot**

Multivariate analysis is a statistical technique used to analyze the relationship between multiple variables simultaneously. Unlike univariate and bivariate analysis, which focus on a single variable or two variables at a time, multivariate analysis involves studying the interactions and dependencies between three or more variables in a dataset.

Here multivariate analysis is carried out using heat maps which can be used to identify presence of collinearity between each column and plots with multiple columns are also taken to understand any underlying relationships.

**5.3.1 Heatmap**

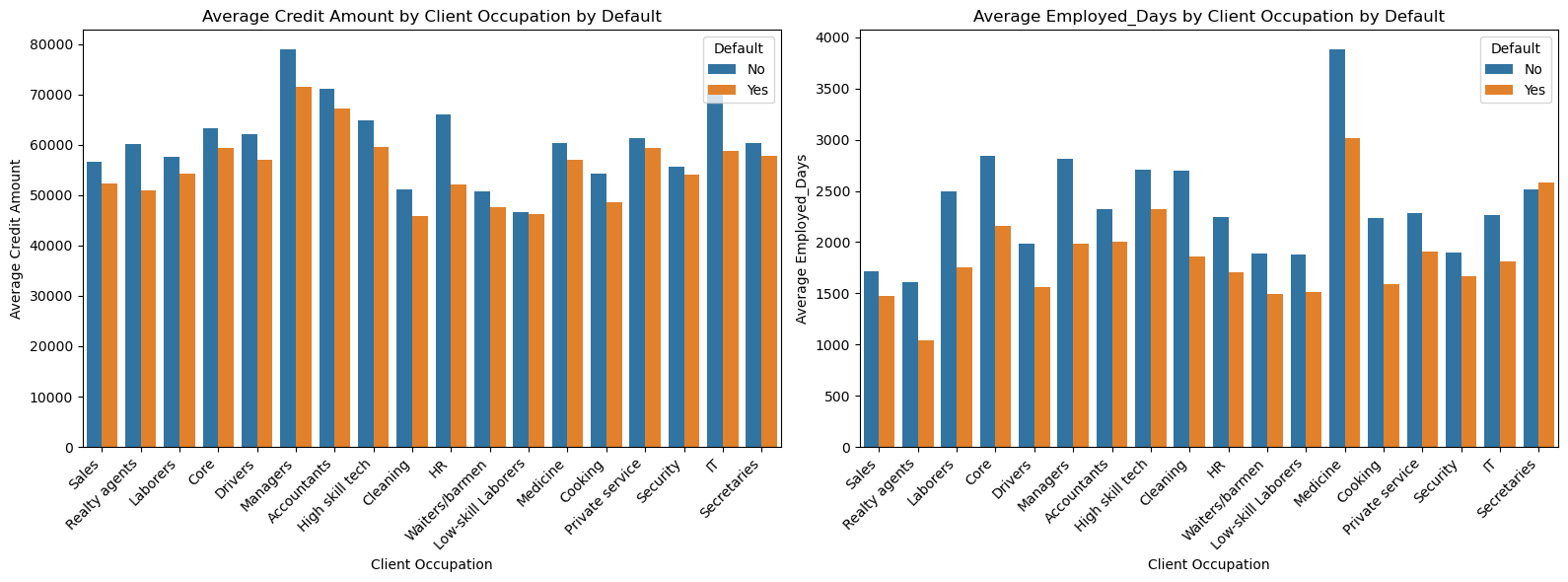
****

**Fig 5.3.1**

Inference from heatmap:

* Credit amount and Loan annuity exhibit high positive correlation which means they are exhibiting high collinearity
* Child count and Client family members also exhibit high positive correlation
* The rightmost column of the heatmap shows the correlation coefficients between each numerical column and the 'Default' column. This provides insights into how each numerical feature relates to the target variable ('Default'). Positive correlations suggest that the feature has a positive impact on the likelihood of default, while negative correlations indicate a negative impact.

**5.3.2.a Multivariate Bar plots**

****

**Fig 5.3.2.a**

Inference from barplot:

Average Employed\_Days by Client Occupation by Default:

* Clients who work in the health sector tend to have a high number of regular working days. There are also higher average working days for clients who default and work in this sector.
* In any event, real estate agents tend to have the lowest average working days regardless of whether or not they are in default.

Average Credit Amount by Client Occupation by Default:

* The plot shows the average credit amount taken by clients belonging to different occupations. We can observe that certain occupations tend to have higher average credit amounts than others.
* 'Managers' and 'IT field' have relatively higher average credit amounts, while 'Laborers' have lower average credit amounts.
* Occupations with higher average credit amounts and lower default rates might be considered less risky for lenders, while occupations with lower average credit amounts and higher default rates could be perceived as riskier.

**6. Data Preprocessing:**

Data preprocessing is the process of cleaning and transforming raw data into a format that is ready for analysis. It is an important step in the data mining process, as it can improve the accuracy and efficiency of the analysis. Data preprocessing can help to improve the accuracy of analysis by removing errors and inconsistencies from the data. This can lead to more accurate results and better decision-making.Data preprocessing can help to increase the efficiency of analysis by reducing the size of the dataset and by converting the data into a format that is more suitable for analysis. This can lead to faster and more efficient analysis.Data preprocessing can help to improve the interpretability of results by removing noise from the data and by summarizing the data into a smaller number of features. This can make the results of the analysis more understandable and easier to communicate.There are many different data preprocessing techniques that can be used to clean and transform data. Some of the most common techniques include:

**6.1 Handling Missing Values**:

Dealing with missing data is essential as many algorithms cannot handle missing values. Techniques for handling missing values include imputation (filling missing values with estimated ones) using mean, median, mode, or regression, or removing rows or columns with missing data.Missing values in a dataset can occur for various reasons. Some of the common reasons include:

**Data Entry Errors:** Human errors during data collection or data entry can lead to missing values. For example, a data entry operator might skip a field, or a sensor might fail to record a reading.

**Incomplete Data Collection:** Sometimes, certain information might not be collected or recorded due to the unavailability of sources or restrictions in data collection.

**Non-Response:** In surveys or questionnaires, some respondents might choose not to answer certain questions, leading to missing values in those fields.

**Data Processing Issues:** Errors can occur during data processing, transformation, or integration, resulting in missing values.

Handling missing values is crucial in building accurate and reliable machine learning models for automobile loan default prediction. Appropriate techniques like data imputation, removing incomplete records, or using specialized algorithms that can handle missing data need to be applied during the data preprocessing phase to ensure the quality and integrity of the predictive model. Additionally, domain expertise and understanding of the loan application process can help in interpreting the reasons behind missing values and making informed decisions on how to handle them.Here are some common techniques for handling missing values:

**Deletion of Missing Data:**

**Listwise Deletion:** Removing entire rows that contain at least one missing value. While this is simple, it can lead to a significant loss of data.

**Pairwise Deletion:** Using available data for each specific analysis or calculation, ignoring missing values for individual calculations. This can lead to varying sample sizes for different analyses.

**Mean, Median, or Mode Imputation:** Replacing missing values with the mean, median, or mode of the non-missing values in the same column. This method is straightforward but may not be ideal if there are many missing values or if the variable has significant variance.

**Regression Imputation:** Predicting the missing values based on other variables through regression models. This approach can be more accurate than simple imputation methods, but it assumes a linear relationship between variables.

**K-Nearest Neighbors (KNN) Imputation:** Imputing missing values based on the values of the nearest neighbors in the feature space. This method is useful for maintaining relationships between variables but can be computationally expensive for large datasets.

**Multiple Imputation:** Generating multiple plausible imputations for each missing value, creating several complete datasets, and then averaging the results. This technique provides more accurate estimates of uncertainty in the imputed values.

**Using Domain Knowledge:**For some missing values, domain knowledge or expert judgment can be used to infer or estimate the appropriate values.

**Create Indicator/Dummy Variables:** Creating an additional binary variable that indicates whether a value is missing or not. This approach allows the model to recognize and account for the potential influence of missingness.

**Interpolation or Extrapolation:** If the data has a time-based structure, interpolation or extrapolation techniques can be used to estimate missing values based on the trend or pattern of the existing data.

The choice of the appropriate missing value handling technique depends on the nature of the data, the amount of missing data, the underlying assumptions, and the analysis objectives. It is essential to carefully consider the implications of each method and to perform sensitivity analysis to understand the potential impact of missing value handling on the final results. Additionally, it is crucial to avoid data leakage by applying the same imputation technique to both the training and testing datasets in machine learning scenarios.

Overall, handling missing values is an integral part of the data preprocessing pipeline. It ensures that data is treated appropriately, leading to more accurate and reliable results in analyses and machine learning applications. It also promotes transparency and accountability in data-driven decision-making processes.Ways we handled missing values in this dataset.

* We started by identifying all the columns in the dataset that contained missing values.
* We then explored the relationships between these columns to exploit any patterns that could help us fill the null values effectively.
* As a first step, we filled the 'Client Occupation' column with the placeholder value 'unknown' to handle its missing values. This allowed us to use this column as a reference point for filling other related columns.
* When we analyzed this column now filled with placeholders we understood that the null values were all belonging to ‘XNA’ Type of organization.
* Next we identified all the Unknown values which had XNA for its type of organization, and then filled them as XNA.
* Utilizing the 'Client Occupation' column, we grouped the data by occupation and calculated the median values for columns such as 'Client Income', 'Credit Amount', and 'Loan Annuity'.
* We had initially filled the ‘Type Organisation’ column to see whether there were any underlying patterns,but since there were none we then filled it using the mode of the column.
* We filled the ‘Client Income type’, ‘Gender’ and ‘Client Education’ with the placeholder other as we didn't want to skew the data toward any particular unique value within this column, and to account for missing data in these fields.
* To handle missing values in several other discrete columns, we grouped the data by the target column ('Default') and took the median for each group.
* Using these median values, we efficiently filled the majority of the discrete columns with missing data.
* For the remaining discrete columns with missing values like ,'Car Owned', 'Bike Owned', 'Active Loan', 'House Own', 'Accompany Client', 'Loan Contract Type' ,'Cleint City Rating' we performed mode imputation. The mode is the most frequently occurring value in the column and serves as an appropriate choice for filling missing data in these cases.
* We utilized the 'Client Family Members' column to fill missing values in the 'Child Count' column, as these two variables were related.
* The 'Client Family Members' column itself was filled using the mode of that column.
* There was a connection between 'Population Region Relative' and 'Client Housing Type'. Using this relationship, we filled the missing values in the 'Population Region Relative' column.

**6.2 Encoding Categorical Variables:**

Categorical variables are typically encoded into numerical representations so that algorithms can process them. Common methods include one-hot encoding, label encoding, and target encoding.There are different encoding techniques based on the type of data being encoded:

**Label Encoding:**

Label encoding is used for converting categorical variables into numerical values. Each category is assigned a unique integer label. However, this encoding might not be suitable for ordinal variables as it assumes an arbitrary order.

**One-Hot Encoding:**

One-hot encoding is used for categorical variables with no inherent order. Each category is converted into a binary vector, where each binary value represents the presence or absence of the category. This technique avoids imposing any ordinal relationship between categories.

**Binary Encoding:**

Binary encoding is a hybrid of label encoding and one-hot encoding. It converts each category into binary code and then represents it as a sequence of binary digits. This technique can be efficient when dealing with high-cardinality categorical variables.

**Ordinal Encoding:**

Ordinal encoding is suitable for categorical variables with a clear ordinal relationship. It assigns a numerical value to each category based on the order of their importance or ranking.

In summary, encoding categorical data is essential for transforming non-numeric information into a format that can be used by machine learning algorithms. By doing so, we enable efficient data processing, avoid biases, improve model performance, and ultimately obtain meaningful insights and predictions from the data.

Moving on to the encoding phase,

* We identified the columns that required encoding based on their data types.
* Since the Client education was an ordinal form of data and since there was a priority for people with higher education we used Ordinal Encoding.
* For the remaining categorical columns, we applied Label Encoding, assigning numerical labels to each unique category.

**6.3 Outlier detection & Outlier Handling:**

Outlier detection is a data preprocessing technique used to identify and handle observations or data points that significantly deviate from the majority of the data points in a dataset. Outliers are data points that lie far away from the central tendency of the data and can distort statistical analyses and machine learning models. Detecting and addressing outliers is crucial to ensure the accuracy and reliability of data analysis and predictive models.In certain cases, outliers or extreme values might be removed from the dataset, leaving behind missing values.

**Privacy and Confidentiality Concerns**: In some cases, data might be intentionally masked or removed to protect the privacy and confidentiality of individuals or sensitive information.

**Instrument or Sensor Failure:** In datasets involving sensors or instruments, malfunctioning equipment can result in missing data.

**Natural Causes**: For example, weather conditions or natural disasters can interfere with data collection processes, leading to missing values.

**Dependency on External Data Sources:** Datasets that rely on external sources may encounter missing values if those sources do not provide complete information.Detecting outliers is essential because they can distort the results, bias statistical estimates, and lead to inaccurate predictions.Here are some common methods for outlier detection:

**Z-Score or Standard Deviation Method:** This method calculates the z-score of each data point, which represents how many standard deviations it is away from the mean. Data points with a z-score above a certain threshold (e.g., |z-score| > 3) are considered outliers.

**IQR (Interquartile Range) Method:** The IQR is the range between the 25th and 75th percentiles of the data. Data points outside the range (Q1 - 1.5 \* IQR, Q3 + 1.5 \* IQR) are considered outliers.

**Box Plots:** Box plots visually represent the IQR method, making it easy to identify outliers as points outside the "whiskers" of the box plot.

**MAD (Median Absolute Deviation):** MAD is a robust measure of dispersion. Data points with an absolute deviation from the median above a threshold (e.g., median ± 3 \* MAD) are considered outliers.

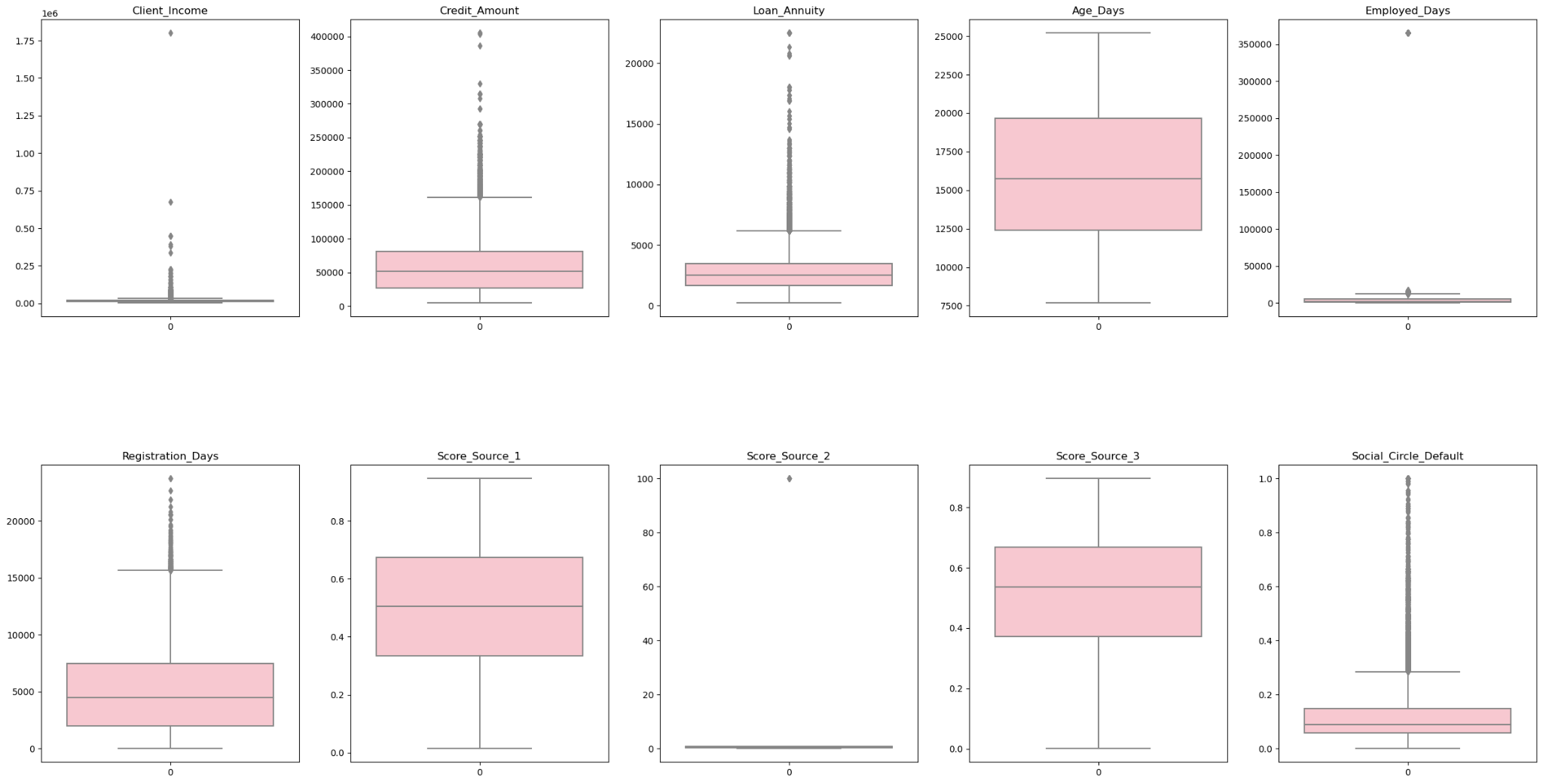
It's important to note that the choice of outlier detection method depends on the nature of the data and the specific use case. Additionally, not all outliers are necessarily errors; they can also represent valuable insights, rare events, or anomalies in the data. In some cases, outliers may need to be removed or adjusted, while in other scenarios, they may be of interest and require further investigation.Outlier detection is a crucial step in data analysis, modeling, and decision-making processes for several reasons:

Overall, outlier detection is an essential part of data exploration and preprocessing. By identifying and dealing with outliers, we can improve data quality, enhance model performance, gain valuable insights, and make more informed and accurate decisions based on the data.

We successfully handled missing values and encoded the categorical variables to prepare the data for analysis and modeling. These measures ensured the integrity of our dataset and set the foundation for obtaining meaningful insights and building predictive models for our project.After completing the data preprocessing steps, we proceeded with outlier detection to identify and handle any data points that deviated significantly from the rest of the data. Outliers can have a significant impact on statistical analyses and predictive models, so it was essential to address them appropriately.

In this project we plotted the boxplots for all the continuous values to identify all possible outliers:

**Fig 6.3.1 Outlier Detection Using Box Plots**

**Fig 6.3.1**

Inference from the plot:

* Credit Income, Credit amount and Loan Annuity columns has a large number of outliers. But from our previous analysis of the data it is clear that they are genuine outliers.
* Registration days, Employed days, social circle and source score 2 have outliers.
* Age days, source score 1 and source score 3 do not have outliers.

In our case , we identified outliers but chose not to handle them because they are all true outliers.

True Outliers: These are data points that genuinely represent extreme or rare occurrences in the data. True outliers can carry valuable information and should not be removed if they are meaningful and relevant to the problem being solved.

**Outlier Handling:**

Outlier handling is an essential step in data preprocessing to ensure that outliers do not unduly influence the results of your analysis or machine learning models. Here are some common techniques for outlier handling:

**Removal or Truncation:**

Removal: Simply removing outlier data points from the dataset. This can be risky, as it might result in loss of information, especially if the dataset is small.

Truncation: Capping the extreme values at a predefined threshold. Data points beyond the threshold are replaced with the threshold value. This approach retains the data points but limits their impact.

**Transformation:**

Logarithmic Transformation: Applying a logarithmic function to the data can compress the range and make it more symmetric. Useful when dealing with skewed distributions.

Box-Cox Transformation: A family of power transformations that can normalize the distribution and stabilize variance.

**Binning:**

Grouping data points into bins or intervals and then labeling or replacing outliers with the bin's midpoint or a predefined value.

**Imputation:**

Replacing outlier values with a central value like the mean, median, or mode. This helps maintain the data's overall distribution.

It's important to choose the appropriate outlier handling technique based on the nature of your data and the goals of your analysis. Outliers might contain valuable information or indicate data quality issues, so blindly removing them can be detrimental. Consider the context and consequences of each technique on your analysis or model performance.

Always document the outlier handling techniques used in your data preprocessing pipeline, as this transparency enhances the reproducibility and trustworthiness of your results.

In the context of our data analysis, we encountered a situation where outliers were identified in the 'Employed\_Days' column. Outliers are data points that significantly deviate from the general trend or distribution of the dataset and can potentially influence the results of our analysis. In order to mitigate their impact and maintain the integrity of our analysis, we employed an outlier handling approach.

To address the presence of outliers, we executed a two-step process.

* First, we compiled a list of the identified outliers by iterating through the 'Employed\_Days' column. Any data point exceeding the upper limit or falling below the lower limit was added to the 'outliers' list.
* Subsequently, we applied a transformation to the 'Employed\_Days' column to replace outliers with values that fall within the acceptable range. Using the NumPy library, we utilized the np.where() function to achieve this transformation.
* Data points greater than the upper limit were replaced with the upper limit value, while data points lower than the lower limit were replaced with the lower limit value.
* This transformation allowed us to retain the original distribution of data while mitigating the potential impact of outliers.

**6.4 Scaling:**

Scaling is an important preprocessing step in machine learning that aims to standardize or normalize the features of a dataset. It ensures that all features have comparable scales, which can improve the performance and convergence of many machine learning algorithms. Scaling becomes particularly crucial when features have different units or ranges, as algorithms that rely on distance measures or gradients might be affected by these variations.When to use scaling:

Many machine learning algorithms, such as k-nearest neighbors, support vector machines, and neural networks, are sensitive to feature scales. Scaling can help these algorithms perform better and converge faster.

When dealing with distance-based algorithms, such as k-means clustering, it's essential to scale the features to avoid the dominance of a single feature in the distance calculation.

Scaling is not necessary for tree-based models, like decision trees and random forests, as these models are invariant to feature scales.Scaling can be a delicate process. It is important to scale the data carefully so that the information in the data is not lost. If you scale the data too much, you may lose important information.It is important to evaluate the results of scaling to see if it has improved the accuracy and performance of the models. If scaling does not improve the accuracy of the models, then there is no need to use it.

It's important to note that some algorithms, like gradient descent-based optimization in neural networks, might require careful tuning of learning rates or other hyperparameters when using standardized features, as it can impact the optimization process. Additionally, you should scale both the training and testing datasets using the same scaling parameters obtained from the training data to avoid data leakage and ensure proper generalization.

There are several different scaling techniques commonly used in machine learning and data preprocessing. Each technique has its advantages and is suitable for different scenarios. Here are some of the most popular scaling techniques:

**Standardization (Z-score scaling):** Scales the features to have a mean of 0 and a standard deviation of 1. This method works well when features follow a roughly Gaussian distribution and when you want to preserve the ability to interpret the original feature values in terms of standard deviations.

**Normalization (Min-Max scaling):** Scales the features to a specified range, usually [0, 1]. It is useful when you want to scale all features to a common range and when the algorithm requires features to be within a specific range.

**Robust Scaling (Median and Median Absolute Deviation - MAD):** Uses the median and median absolute deviation to scale the features, making it robust to outliers in the data. This method is useful when dealing with datasets containing outliers or non-Gaussian distributions.

**Log Transformation:** Takes the logarithm of the data, which can help in reducing the impact of extreme values and make the data more symmetric.

**Ordinal Scaling:** Used for ordinal categorical variables, where the order of categories matters but the actual values do not. Assigns integers to the categories based on their order.

It's important to choose the appropriate scaling technique based on the nature of the data, the algorithm you are using, and the presence of outliers or specific distribution requirements. Always apply scaling after splitting the data into training and testing sets to avoid data leakage and ensure proper evaluation of the model's performance. Scikit-learn provides implementations for many of these scaling techniques through various preprocessing classes, making it easy to incorporate scaling into your machine learning pipelines.

**Robust Scaling:**

* The code creates an instance of the RobustScaler named rs. The RobustScaler scales features using statistics that are robust to outliers, making it suitable for datasets with extreme values.
* The RobustScaler is particularly useful when dealing with data that contains outliers.
* The RobustScaler is used to transform the data in a way that accounts for outliers, ensuring that the scaled features maintain their distribution even when dealing with extreme values.
* This process is important for improving the stability and performance of machine learning algorithms that are sensitive to the scale of input features. The scaled data can subsequently be used for training and evaluating machine learning models.

**6.5 Feature Engineering:**

Feature engineering is the process of transforming raw data into features that are more informative and useful for machine learning algorithms. It is a critical step in the machine learning process, as it can have a significant impact on the accuracy and performance of the models.It involves selecting, modifying, or creating new features that can provide more relevant and useful information for the model to make accurate predictions. Effective feature engineering can have a significant impact on model accuracy and generalization.It's important to note that feature engineering should be done carefully and thoughtfully, as creating irrelevant or redundant features can lead to overfitting or reduce the model's performance. Additionally, feature engineering is an iterative process that may require experimentation and domain expertise to find the most informative features for a specific problem.

Overall, feature engineering can be a useful tool for improving the accuracy, interpretability, and fairness of machine learning models. However, it is important to be aware of the potential drawbacks of feature engineering before using it.

* In this step we combines the information from two existing columns: 'Car\_Owned' and 'Bike\_Owned',for each row in the DataFrame, it calculates the sum of the values in the 'Car\_Owned' column and the 'Bike\_Owned' column and stores the result in the new 'Vehicle\_Owned' column.
* Then we combines the information from another two existing columns: 'Child\_Count' and 'Client\_Family\_Members',for each row in the DataFrame, it calculates the sum of the values in the 'Child\_Count' column and the 'Client\_Family\_Members' column and stores the result in the new 'Family\_Members' column.

**6.6 Feature Reduction:**

Feature reduction, also known as feature selection or dimensionality reduction, is the process of selecting a subset of relevant and significant features from the original set of features in a dataset. Manual dropping of columns is one way to perform feature reduction, where specific columns that are deemed less important or redundant are removed from the dataset.The main goal of feature reduction is to improve the performance of the model, simplify the model, and reduce computational complexity. It helps to focus on the most important features that contribute significantly to the target variable while ignoring irrelevant or redundant features.

There are two main types of feature reduction: feature selection and feature extraction.

**Feature selection** involves selecting a subset of the original features that are most relevant to the task at hand. This can be done using a variety of methods, such as correlation, information gain, and relief.

**Feature extraction** involves transforming the original features into a new set of features that are more informative. This can be done using methods such as principal component analysis (PCA) and linear discriminant analysis (LDA).

**Benefits of feature reduction:**Improved performance of machine learning models: Feature reduction can help to improve the performance of machine learning models by reducing the number of irrelevant features. This can make the models more accurate and less prone to overfitting.

Overall, feature reduction can be a useful tool for improving the performance of machine learning models, simplifying data, and reducing storage requirements. However, it is important to be aware of the potential drawbacks of feature reduction before using it.

* Based on our analysis and considerations, we have decided to drop the following columns from the dataset:

'Accompany\_Client', 'Client\_Housing\_Type', 'Population\_Region\_Relative', 'Registration\_Days','ID\_Days','Mobile\_Tag','Homephone\_Tag', 'Cleint\_City\_Rating','Client\_Permanent\_Match\_Tag','Client\_Contact\_Work\_Tag','Workphone\_Working','Score\_Source\_1','Score\_Source\_2','Score\_Source\_3', 'Social\_Circle\_Default','Phone\_Change', and 'Credit\_Bureau'.

**7. Model Building**

"Model building" refers to the process of creating and training a machine learning model using a given dataset to make predictions or classifications on new, unseen data. This process involves several steps, including data preprocessing, selecting a suitable algorithm, tuning hyperparameters, training the model, and evaluating its performance.

**7.1 Classification**

Classification is a type of supervised learning task that involves categorizing input data into one of several predefined classes or categories. The goal of a classification algorithm is to learn a mapping from input features to the corresponding class labels based on a labeled dataset. These class labels can represent different outcomes, classes, or categories that the algorithm needs to predict for new, unseen data.

**7.2 Model Creation**

The code starts by splitting your data into train, validation, and test sets using the train\_test\_split function. This process is essential for training, tuning, and evaluating your model on different datasets. Here we split the datasets into 3

1. Training Dataset
2. Test Dataset
3. Validation Dataset

Training Set: The training set is the portion of the dataset used to train the machine learning model. It consists of labeled examples where both the input features and the corresponding class labels are known.

Validation Set: The validation set is used to tune the hyperparameters of the model and to prevent overfitting. Hyperparameters are settings that are not learned during training but affect the behavior of the model (e.g., learning rate, regularization strength).

By evaluating the model's performance on the validation set, you can choose the best combination of hyperparameters that leads to optimal performance.

Test set: The testing set is a separate portion of the dataset that is not used during training or hyperparameter tuning. It is used to evaluate the final performance of the trained model and to estimate how well the model will perform on new, unseen data in real-world scenarios.

For our project we chose the Random Forest Classifier as it gave us the highest accuracy precision and f1 scores.

**7.2.1 Random Forest Classifier**

Random Forest is a powerful ensemble learning algorithm used for both classification and regression tasks in machine learning. It's composed of multiple decision trees, where each tree is trained on a random subset of the training data and makes independent predictions. The final prediction is then determined by combining the predictions of individual trees through a voting or averaging mechanism.The fundamental building blocks of a Random Forest are decision trees. Each tree is constructed using a random subset of the training data and a random subset of features.

The training data for each tree is created by bootstrapping, which involves randomly selecting samples with replacement from the original training dataset. This creates diversity among the individual trees.For each split in a decision tree, only a random subset of features is considered. This further increases the diversity among the trees and reduces the risk of overfitting.

In classification tasks, each tree "votes" for a class, and the class with the most votes is the final predicted class. In regression tasks, the predicted values from all trees are averaged to produce the final prediction.Random Forest is designed to reduce overfitting compared to individual decision trees by introducing randomness in the model creation process. This helps prevent the model from memorizing the training data.Random Forest can effectively handle high-dimensional datasets and features without requiring feature selection or dimensionality reduction techniques.

Random Forest is less sensitive to outliers compared to some other algorithms, thanks to the averaging or voting mechanism.

It is widely used due to its strong performance, versatility, and ability to handle a variety of data types and structures. It's suitable for tasks ranging from classification and regression to feature importance estimation and outlier detection.

Accuracy= 0.9262648229453039

Precision= 0.9304828678497525

Recall= 0.9262648229453039

F1 Score 0.896956135695233

we also used a few different algorithms before selecting Random forest classifier they are

* **K-Nearest Neighbors (KNN) :**

K-Nearest Neighbors (KNN) is a machine learning algorithm used for both classification and regression tasks. It's a non-parametric method that makes predictions based on the similarity of input data to the data points in a labeled training dataset.K-Nearest Neighbors is a versatile and straightforward algorithm that is used for predictive modeling tasks by leveraging the proximity of labeled data points to make predictions for new, unseen data.

Accuracy = 0.9200722140160841

Precision = 0.8732660100665061

Recall = 0.92007221401608

F1 Score = 0.8818915383334448

* **Decision Tree :**

A Decision Tree is a versatile and intuitive machine learning algorithm used for both classification and regression tasks. It models decisions or decisions-making processes using a tree-like structure, where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents an outcome or prediction.Decision trees are used in various domains due to their simplicity, interpretability, and ability to handle both discrete and continuous data. However, they can be prone to overfitting, especially when the tree is deep and the dataset is small. Techniques like pruning and using ensemble methods can help mitigate this issue.

Accuracy= 0.8696865255210898

Precision= 0.8799542502078447

Recall= 0.8696865255210898

F1 Score 0.874623036263389

* **Support Vector Machine (SVM):**

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for both classification and regression tasks. It's particularly effective for tasks where the data is not linearly separable in the feature space. SVM aims to find a hyperplane that best separates different classes while maximizing the margin between them.SVM is known for its ability to handle complex decision boundaries and perform well on a variety of datasets. It's widely used in various fields including image classification, text classification, and bioinformatics. However, SVM's performance can be sensitive to the choice of hyperparameters and kernel functions, and it might be computationally expensive for large datasets.

Accuracy= 0.9205

Precision= 0.84732025

Recall= 0.9205

F1 Score 0.8823954699297057

**8. Hyperparameter Tuning**

The hyperparameter tuning process was conducted to optimize the machine learning model's performance for predicting a target variable in the given dataset.

The project utilized RandomizedSearchCV, a popular hyperparameter tuning technique, in combination with the RandomForestRegressor algorithm. The dataset was divided into training validation and test sets for model development and evaluation.

A dictionary param\_dist defines the hyperparameters you want to tune for the RandomForestClassifier (n\_estimators, max\_depth, and criterion).A base RandomForestClassifier model is instantiated.A RandomizedSearchCV instance is created, which performs a randomized search over hyperparameter values to find the best combination of parameters. It uses cross-validation (cv=5) to evaluate performance.The RandomizedSearchCV object (rf\_random\_search) is fitted to the training data, searching for the best hyperparameters.

**Printing Tuning Results:**

We printed out all the possible combinations of hyperparameters and their accuracy using a special function. The print\_results\_random function takes the results of the hyperparameter tuning and prints out information about the best parameters and their corresponding mean and standard deviation scores across different parameter combinations. After obtaining the three best models we then fitted them on to our test set from which the best model and its hyperparameter combination are obtained. the hyperparameters where (n\_estimators=500, max\_depth=50, criterion='entropy').

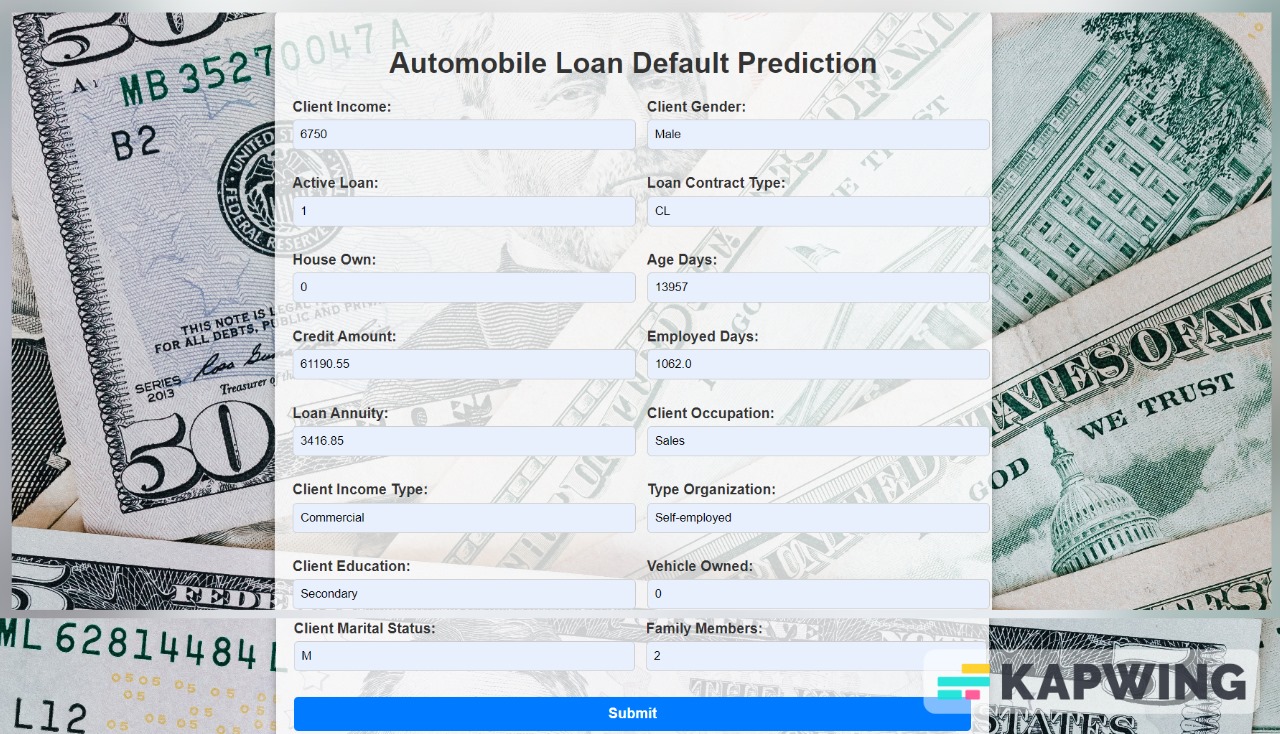
**Training and Evaluating Models**:

A final RandomForestClassifier model (rf2\_final) is created with fixed hyperparameters (n\_estimators=500, max\_depth=50, criterion='entropy').The model's performance is evaluated using the validation set (x\_val and y\_val). The predicted labels (y\_pred) are calculated, and accuracy and precision scores are calculated and printed for the rf2\_final model.

**9. Website Hosting**

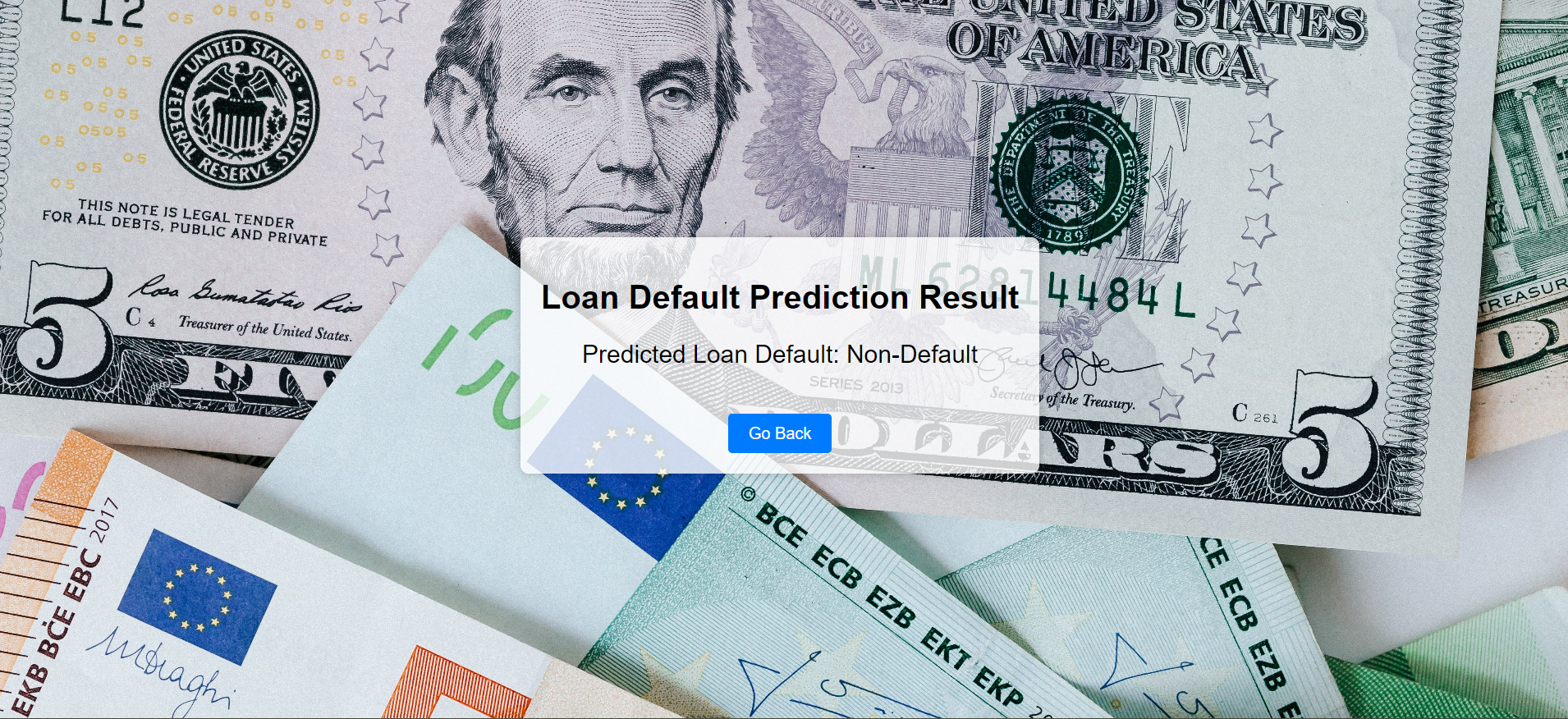
Website hosting refers to the service of storing and making web content accessible on the internet. When you create a website, the files (HTML, CSS, images, scripts, etc.) that make up your site need to be stored on a server connected to the internet so that users can access your website through their web browsers. Website hosting providers offer the infrastructure and technologies needed to store and serve these files to users when they visit your website.

We used the best model to create a flask app which can be used to predict whether a person is likely to default on automobile loans. But since our model was extremely imbalanced the app always gave a non default prediction. The screenshots of the apps are provided below.



**Fig 9.1**

and the result which was obtained by using a Random Forest Classifier model is viewed on the result page.



**Fig 9.2**

**10. Result**

In the process of developing a predictive model for automobile loan default, we employed a Random Forest classifier which demonstrated promising Accuracy , Precision,Recall and F1 score metrics during testing. However, an important observation was made regarding the model's performance on the real-world data. Due to the substantial class imbalance within the dataset, the model consistently provided 'Non-Default' predictions for both default and non-default instances when deployed as part of our Flask web application. This behavior arose from the model's challenge in effectively addressing the minority class.

**11. Conclusion**

Our endeavor to develop a predictive model for automobile loan default using a Random Forest classifier has led us to significant insights. While the model demonstrated commendable accuracy and precision in predicting loan defaults on a balanced dataset, the deployment of the model within the Flask app has highlighted the challenges associated with highly imbalanced data.

Due to the substantial class imbalance between default and non-default instances, the model's predictions within the app predominantly leaned towards the majority class. Despite its high predictive performance on a balanced dataset, the inherent nature of imbalanced data skewed the outcomes, leading to a prevalence of non-default predictions. This unanticipated outcome emphasized the importance of handling class imbalance as a critical aspect of model deployment.

Our experiment underscores the significance of employing appropriate strategies to address class imbalance, such as resampling techniques, synthetic data generation, or adjusting class weights during training. Recognizing the real-world implications of automobile loan default prediction, future endeavors should prioritize the application of techniques that ensure a more equitable representation of both default and non-default instances, ultimately leading to a more reliable and fair prediction model.

In summary, this project serves as a valuable lesson in the impact of class imbalance on model performance in real-world scenarios, emphasizing the need for continuous exploration and refinement of strategies to enhance the robustness and effectiveness of predictive models."

**References**

* Tariq, A., Khan, M. N., & Shahzad, M. (2016). Predicting Loan Default: A Case Study of Pakistan. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 686-689).
* Madaan, V., Chhabra, J. K., & Kumar, A. (2017). Predicting Loan Default Using Data Mining Techniques. Journal of Big Data, 4(1), 26.
* Kwofie, S. E., Adu, G. A., & Mensah, J. (2018). Predicting Loan Default in a Microfinance Company: A Logistic Regression Approach. Journal of Applied Mathematics, 2018.
* www.kaggle.com/