455dc5

BENGALURU CAMPUS

**PROJECT TITLE:**

**CREDIT WORTHINESS EVALUATION**

**PROJECT REPORT FINAL**

Unlocking Financial Confidence: Your Trustworthy Partner in Credit Worthiness Evaluation.

**PROJECT DETAILS**

|  |  |
| --- | --- |
| Batch details | DSE AUGUST 23, Bangalore |
| Team members | JANANE M  JAYESH SURENDRA KUMBHAR  MUNIR SHETH  SRUTHI BHANDARE  VASUDEV R NAIR |
| Domain of Project | FINANCE |
| Proposed project title | **CREDIT WORTHINESS EVALUATION** |
| Group Number | Group 07 |
| Team Leader | VASUDEV R NAIR |
| Mentor Name | SAI SOURABH REDDY |

Date: 22– 02- 2024

Signature of the Mentor Signature of the Team Leader

ABSTRACT

This research focuses on the development of a predictive model leveraging machine learning algorithms to assess an individual's creditworthiness, specifically their ability to repay a loan. The binary target variable categorizes individuals as either "Unable to pay the loan" (labeled as 1) or "Able to pay the loan" (labeled as 0). The dataset comprises a diverse set of features, including gender, salary, details about the individual's residence, familial information, marital status, and more.

The primary objective is to employ sophisticated machine learning techniques to analyze these features systematically and identify patterns indicative of financial stability or risk. By doing so, the model aims to provide financial institutions with a robust tool for making informed decisions on loan approvals, ultimately contributing to more accurate assessments of an individual's creditworthiness.

Through the careful examination of various attributes, our research seeks to enhance the accuracy and efficiency of credit evaluations. The findings from this study have the potential to revolutionize the loan approval process, allowing financial institutions to optimize risk management and better serve their customers

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | ABSTRACT | i |
|  | LIST OF TABLES | v |
|  | LIST OF FIGURES | vi |
|  | LIST OF ABBREVIATIONS | vii |
| 1 | INTRODUCTION |  |
| 1.1 | Problem Definition |  |
| 1.2 | Motivation / Challenges |  |
| 2 | LITERATURE SURVEY |  |
| 2.1 | Survey |  |
| 2.2 | Findings |  |
| 2.3 | Objectives of the work |  |
| 3 | PROPOSED WORK |  |
| 3.1 | Architecture of the System |  |
| 3.2 | Phases |  |
| 3.3 | Algorithm Design |  |
| 3.3.1 | Decision Tree |  |
| 3.3.2 | Random Forest |  |
| 3.3.3 | Logistic Regression |  |
| 3.3.4 | Hybrid Classifier |  |
| 3.4 | Data Set Description |  |
| 4 | Results & Inferences |  |
| 4.1 | Confusion Matrix |  |
| 4.2 | Results & Discussion |  |
| 4.3 | Results Obtained |  |
| 5 | Conclusion & Future Enhancement |  |
| 6 | References |  |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NO | TITLE | PAGE NO |
| 1 | Results comparison from dataset-1 | 22 |
| 2 | Results comparison from dataset-2 | 22 |

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO | TITLE | PAGE NO. |
| 1 | The architecture of the system | 10 |
| 2 | Sigmoid function | 14 |
| 3 | Ensemble Method | 15 |
| 4 | Stacking Classifier | 16 |
| 5 | Statistical Description | 16 |
| 6 | Confusion Matrix | 19 |
| 7 | Pie-Chart | 23 |
| 8 | Heat Map | 24 |
| 9 | Accuracy Comparison Plot for Dataset – 1 | 24 |
| 10 | Accuracy Comparison Plot for Dataset – 2 | 25 |

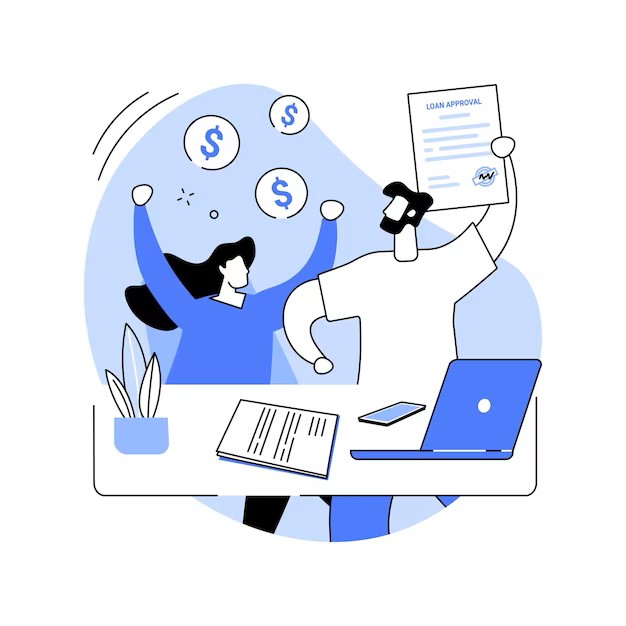
LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| FIGURE NO | TITLE | PAGE NO. |
| 1 | The architecture of the system | 10 |
| 2 | Sigmoid function | 14 |
| 3 | Ensemble Method | 15 |
| 4 | Stacking Classifier | 16 |
| 5 | Statistical Description | 16 |
| 6 | Confusion Matrix | 19 |
| 7 | Pie-Chart | 23 |
| 8 | Heat Map | 24 |
| 9 | Accuracy Comparison Plot for Dataset – 1 | 24 |
| 10 | Accuracy Comparison Plot for Dataset – 2 | 25 |

1. **INTRODUCTION**

In the dynamic landscape of the financial industry, credit card fraud poses a significant threat, resulting in substantial financial losses annually. To counteract this challenge, there has been a paradigm shift from a reactive, post-incident investigation approach to a proactive, predictive approach. This transformation involves the implementation of sophisticated fraud detection algorithms aimed at providing early warnings and supporting fraud investigators in their efforts to curb fraudulent activities.

This case study is designed to provide a practical demonstration of applying Exploratory Data Analysis (EDA) techniques within the context of real-world business scenarios. EDA is a crucial phase in data analysis that involves the exploration and visualization of datasets to extract meaningful insights and patterns. In the context of credit card fraud detection, EDA serves as a foundational step to understand the characteristics of the data and identify potential features that can contribute to the development of robust predictive models.



Key Objectives:

1. **EXPLORATORY DATA ANALYSIS (EDA**)

Utilize various EDA techniques to gain insights into the structure, distribution, and relationships within the dataset.

Visualize patterns, trends, and anomalies that may inform the subsequent steps in the analysis.

1. **CLASSIFICATION MODELS:**

Apply classification models to predict whether a customer is likely to default on credit card payments.

Explore the use of machine learning algorithms for predictive modeling to enhance fraud detection capabilities.

1. **RISK MITIGATION:**

Contribute to the ongoing efforts in the financial industry to mitigate the risks associated with credit card fraud.

Provide a practical demonstration of how data-driven approaches can enhance decision-making and preemptively address potential fraud instances.

By combining the power of Exploratory Data Analysis and classification models, this case study aims to showcase the practical application of data science techniques in the financial sector. Through a comprehensive understanding of the dataset and the deployment of predictive models, the goal is to contribute to the development of proactive strategies for identifying and preventing credit card fraud, ultimately safeguarding financial institutions and their customers.

## **BUSINESS PROBLEM UNDERSTANDING**

The loan providing companies find it hard to give loans to people due to their inadequate or missing credit history. Some consumers use this to their advantage by becoming a defaulter. Let us consider your work for a consumer finance company that specialises in lending various types of loans to customers. You must use Exploratory Data Analysis (EDA) to analyse the patterns present in the data which will make sure that the loans are not rejected for the applicants capable of repaying.

When the company receives a loan application, the company has to rights for loan approval based on the applicant’s profile. Two types of risks are associated with the bank’s or company’s decision:

If the aspirant is likely to repay the loan, then not approving the loan tends in a business loss to the company

If the a is aspirant not likely to repay the loan, i.e. he/she is likely to default/fraud, then approving the loan may lead to a financial loss for the company.

The data contains information about the loan application.

When a client applies for a loan, there are four types of decisions that could be taken by the bank/company:

**1.Approved**

**2.Cancelled**

**3. Refused**

**4. Unused offer**: The loan has been cancelled by the applicant but at different stages of the process.

In this comprehensive case study, the primary focus lies on leveraging Exploratory Data Analysis (EDA) techniques to delve into the intricate interplay between consumer attributes and loan-specific factors, unraveling their collective impact on the likelihood of loan default. By meticulously scrutinizing and visualizing the dataset, we aim to discern patterns, uncover correlations, and gain profound insights into the intricate relationships that govern the creditworthiness of clients.

The EDA process will entail a thorough examination of consumer-centric characteristics and loan attributes, allowing us to draw connections between these variables and the propensity for loan default. Through visual representations and statistical analyses, we endeavor to elucidate the underlying dynamics, shedding light on potential risk factors and discernible trends.

Furthermore, this case study extends beyond the realms of exploration, delving into the realm of predictive analytics. By employing advanced classification models, we seek to anticipate and predict the creditworthiness of clients. The models will be trained on the amalgamated insights derived from EDA, empowering us to build robust frameworks for determining whether a client is deemed worthy or poses a potential default risk.

## **BUSINESS GOAL**

In this insightful case study, the primary objective is to discern patterns that serve as reliable indicators of an applicant's likelihood to repay their loan installments. The strategic identification of these patterns holds the key to informed decision-making, paving the way for actions such as approving or denying the loan, adjusting loan amounts, or determining interest rates. By employing a combination of Exploratory Data Analysis (EDA) and sophisticated Classification techniques, the study seeks to unravel the intrinsic characteristics that distinguish applicants capable of fulfilling their repayment commitments. The overarching goal is to enhance the decision-making process, ensuring that deserving applicants are not unjustly rejected while simultaneously mitigating the risk of default. This case study is poised to contribute valuable insights and predictive capabilities that empower stakeholders in making sound lending decisions based on a thorough understanding of applicant profiles.

This dual-pronged approach, encompassing both exploratory insights and predictive modeling, aims to provide a comprehensive understanding of the intricate relationships between consumer attributes, loan features, and creditworthiness. Ultimately, the goal is to equip stakeholders in the financial domain with actionable insights, enabling informed decision-making and fostering a proactive stance in managing the risk associated with loan defaults.

1. **ABOUT THE DATASET**

'application\_data.csv' serves as a comprehensive repository encapsulating pertinent information about clients at the moment of loan application. The dataset intricately captures the nuances related to payment difficulties, offering insights into whether an applicant is currently experiencing challenges in meeting their payment obligations. This valuable compilation of client-specific data encompasses a diverse range of attributes, facilitating a holistic understanding of the financial landscape at the time of application. By delving into this dataset, stakeholders gain a nuanced perspective on applicants' financial health, thereby enabling informed decision-making processes and the formulation of targeted strategies to address payment difficulties when necessary. The dataset acts as a critical resource in the quest for a thorough comprehension of the dynamics surrounding payment behaviors within the scope of loan applications.

**TABLE 1 - DATA DICTIONARY FOR PREVIOUS APPLICATION**

|  |  |  |
| --- | --- | --- |
| **Slno** | **Col name** | **Description** |
| 1 | SK\_ID\_CURR | ID of loan in our sample |
| 2 | TARGET | 1. Unable to pay the loan   0 - Able to pay the loan |
| 3 | NAME\_CONTRACT\_TYPE | Identification if loan is cash or revolving |
| 4 | CODE\_GENDER | Gender of the client |
| 5 | FLAG\_OWN\_CAR | Does the client own a car |
| 6 | FLAG\_OWN\_REALTY | Does the client has a property or not |
| 7 | CNT\_CHILDREN | Number of children the client has |
| 8 | AMT\_INCOME\_TOTAL | Income of the client |
| 9 | AMT\_CREDIT | total amount of money that is borrowed by the client |
| 10 | AMT\_ANNUITY | Amount paid / year |
| 11 | AMT\_GOODS\_PRICE | For consumer loans it is the price of the goods for which the loan is given |
| 12 | NAME\_TYPE\_SUITE | Who accompanied client when applying for the previous application |
| 13 | NAME\_INCOME\_TYPE | Clients income type (businessman, working, maternity leave,…) |
| 14 | NAME\_EDUCATION\_TYPE | highest education level the client achieved |
| 15 | NAME\_FAMILY\_STATUS | Marital status of the client |
| 16 | NAME\_HOUSING\_TYPE | What is the housing situation of the client (renting, living with parents, ...) |
| 17 | REGION\_POPULATION\_RELATIVE | Popoulation normalized value, higher value indicates more population in the area |
| 18 | DAYS\_BIRTH | Client's age in days at the time of application |
| 19 | DAYS\_EMPLOYED | How many days before the application the person started current employment |
| 20 | DAYS\_REGISTRATION | How many days before the application did client change his registration |
| 21 | DAYS\_ID\_PUBLISH | How many days before the application did client change the identity document with which he applied for the loan |
| 22 | OWN\_CAR\_AGE | Age of client's car |
| 23 | FLAG\_MOBIL | Did client provide mobile phone (1=YES, 0=NO) |
| 24 | FLAG\_EMP\_PHONE | Did client provide work phone (1=YES, 0=NO) |
| 25 | FLAG\_WORK\_PHONE | Did client provide home phone (1=YES, 0=NO) |
| 26 | FLAG\_CONT\_MOBILE | Was mobile phone reachable (1=YES, 0=NO) |
| 27 | FLAG\_PHONE | Did client provide home phone (1=YES, 0=NO) |
| 28 | FLAG\_EMAIL | Did client provide email (1=YES, 0=NO) |
| 29 | OCCUPATION\_TYPE | What kind of occupation does the client have |
| 30. | CNT\_FAM\_MEMBERS | How many family members does client have |
| 31 | REGION\_RATING\_CLIENT | Our rating of the region where client lives (1,2,3) |
| 32 | REGION\_RATING\_CLIENT\_W\_CITY | Our rating of the region where client lives with taking city into account (1,2,3) |
| 33 | WEEKDAY\_APPR\_PROCESS\_START | On which day of the week did the client apply for the loan |
| 34 | HOUR\_APPR\_PROCESS\_START | Approximately at what hour did the client apply for the loan |
| 35 | REG\_REGION\_NOT\_LIVE\_REGION | Flag if client's permanent address does not match contact address (1=different, 0=same, at region level) |
| 36 | REG\_REGION\_NOT\_WORK\_REGION | Flag if client's permanent address does not match work address (1=different, 0=same, at region level) |
| 37 | LIVE\_REGION\_NOT\_WORK\_REGION | Flag if client's contact address does not match work address (1=different, 0=same, at region level) |
| 38 | REG\_CITY\_NOT\_LIVE\_CITY | Flag if client's permanent address does not match contact address (1=different, 0=same, at city level) |
| 39 | REG\_CITY\_NOT\_WORK\_CITY | Flag if client's permanent address does not match work address (1=different, 0=same, at city level) |
| 40 | LIVE\_CITY\_NOT\_WORK\_CITY | Flag if client's contact address does not match work address (1=different, 0=same, at city level) |
| 41 | ORGANIZATION\_TYPE | Type of organization where client works |
| 42 | EXT\_SOURCE\_1 | Normalized score from external data source |
| 43 | EXT\_SOURCE\_2 | Normalized score from external data source |
| 44 | EXT\_SOURCE\_1 | Normalized score from external data source |
| 45 | APARTMENTS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 46 | BASEMENTAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 47 | YEARS\_BEGINEXPLUATATION\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 48 | YEARS\_BUILD\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 49 | COMMONAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 50 | ELEVATORS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 51 | ENTRANCES\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 52 | FLOORSMAX\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 53 | FLOORSMIN\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 54 | LANDAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 55 | LIVINGAPARTMENTS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 56 | LIVINGAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 57 | NONLIVINGAPARTMENTS\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 58 | NONLIVINGAREA\_AVG | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 59 | APARTMENTS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 60 | BASEMENTAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 61 | YEARS\_BEGINEXPLUATATION\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 62 | YEARS\_BUILD\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 63 | COMMONAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 64 | ELEVATORS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 65 | ENTRANCES\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 66 | FLOORSMAX\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 67 | FLOORSMIN\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 68 | LANDAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 69 | LIVINGAPARTMENTS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 70 | LIVINGAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 71 | NONLIVINGAPARTMENTS\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 72 | NONLIVINGAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 73 | APARTMENTS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 74 | BASEMENTAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 75 | YEARS\_BEGINEXPLUATATION\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 76 | YEARS\_BUILD\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 77 | COMMONAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 78 | ELEVATORS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 79 | ENTRANCES\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 80 | FLOORSMAX\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 81 | FLOORSMIN\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 82 | LANDAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 83 | LIVINGAPARTMENTS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 84 | LIVINGAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 85 | NONLIVINGAPARTMENTS\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 86 | NONLIVINGAREA\_MEDI | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 87 | FONDKAPREMONT\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 88 | HOUSETYPE\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 89 | TOTALAREA\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 90 | WALLSMATERIAL\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 91 | EMERGENCYSTATE\_MODE | Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor |
| 92 | OBS\_30\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings with observable 30 DPD (days past due) default |
| 93 | DEF\_30\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings defaulted on 30 DPD (days past due) |
| 94 | OBS\_60\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings with observable 60 DPD (days past due) default |
| 95 | DEF\_60\_CNT\_SOCIAL\_CIRCLE | How many observation of client's social surroundings defaulted on 60 (days past due) DPD |
| 96 | DAYS\_LAST\_PHONE\_CHANGE | How many days before application did client change phone |
| 97 | FLAG\_DOCUMENT\_2 | Did client provide document 2 |
| 98 | FLAG\_DOCUMENT\_3 | Did client provide document 3 |
| 99 | FLAG\_DOCUMENT\_4 | Did client provide document 4 |
| 100 | FLAG\_DOCUMENT\_5 | Did client provide document 5 |
| 101 | FLAG\_DOCUMENT\_6 | Did client provide document 6 |
| 102 | FLAG\_DOCUMENT\_7 | Did client provide document 7 |
| 103 | FLAG\_DOCUMENT\_8 | Did client provide document 8 |
| 104 | FLAG\_DOCUMENT\_9 | Did client provide document 9 |
| 105 | FLAG\_DOCUMENT\_10 | Did client provide document 10 |
| 106 | FLAG\_DOCUMENT\_11 | Did client provide document 11 |
| 107 | FLAG\_DOCUMENT\_12 | Did client provide document 12 |
| 108 | FLAG\_DOCUMENT\_13 | Did client provide document 13 |
| 109 | FLAG\_DOCUMENT\_14 | Did client provide document 14 |
| 110 | FLAG\_DOCUMENT\_15 | Did client provide document 15 |
| 111 | FLAG\_DOCUMENT\_16 | Did client provide document 16 |
| 112 | FLAG\_DOCUMENT\_17 | Did client provide document 17 |
| 113 | FLAG\_DOCUMENT\_18 | Did client provide document 18 |
| 114 | FLAG\_DOCUMENT\_19 | Did client provide document 19 |
| 115 | FLAG\_DOCUMENT\_20 | Did client provide document 20 |
| 116 | FLAG\_DOCUMENT\_21 | Did client provide document 21 |
| 117 | AMT\_REQ\_CREDIT\_BUREAU\_HOUR | Credit Bureau is an agency that collects and maintains credit information on individuals. |
| 118 | AMT\_REQ\_CREDIT\_BUREAU\_DAY | Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application) |
| 119 | AMT\_REQ\_CREDIT\_BUREAU\_WEEK | Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application) |
| 120 | AMT\_REQ\_CREDIT\_BUREAU\_MON | Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application) |
| 121 | AMT\_REQ\_CREDIT\_BUREAU\_QRT | Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application) |
| 122 | AMT\_REQ\_CREDIT\_BUREAU\_YEAR | Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application) |

# **5. DATA PREPARATION**

Data preparation involves transforming raw data into a form that is more appropriate for modeling.

Preparing data may be the most important part of a predictive modeling project and the most time-consuming, although it seems to be the least discussed. Instead, the focus is on machine learning algorithms, whose usage and parameterization has become quite routine.

Practical data preparation requires knowledge of data cleaning, feature selection data transforms, dimensionality reduction, and more.

**4.1. FEATURE SELECTION**

Feature selection is the process of reducing the number of input variables when developing a predictive model.

It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

Statistical-based feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable. These methods can be fast and effective, although the choice of statistical measures depends on the data type of both the input and output variables.

As such, it can be challenging for a machine learning practitioner to select an appropriate statistical measure for a dataset when performing filter-based feature selection.

There are two main types of feature selection techniques: supervised and unsupervised, and supervised methods may be divided into wrapper, filter and intrinsic.

Filter-based feature selection methods use statistical measures to score the correlation or dependence between input variables that can be filtered to choose the most relevant features.

Statistical measures for feature selection must be carefully chosen based on the data type of the input variable and the output or response variable.

Feature selection methods are intended to reduce the number of input variables to those that are believed to be most useful to a model in order to predict the target variable.Some predictive modeling problems have a large number of variables that can slow the development and training of models and require a large amount of system memory. Additionally, the performance of some models can degrade when including input variables that are not relevant to the target variable.

The difference has to do with whether features are selected based on the target variable or not. Unsupervised feature selection techniques ignores the target variable, such as methods that remove redundant variables using correlation. Supervised feature selection techniques use the target variable, such as methods that remove irrelevant variables..

Another way to consider the mechanism used to select features which may be divided into wrapper and filter methods. These methods are almost always supervised and are evaluated based on the performance of a resulting model on a hold out dataset.

Wrapper feature selection methods create many models with different subsets of input features and select those features that result in the best performing model according to a performance metric. These methods are unconcerned with the variable types, although they can be computationally expensive. RFE is a good example of a wrapper feature selection method.

Some predictive modeling problems have a large number of variables that can slow the development and training of models and require a large amount of system memory. Additionally, the performance of some models can degrade when including input variables that are not relevant to the target variable.

Feature selection is also related to dimensionally reduction techniques in that both methods seek fewer input variables to a predictive model. The difference is that feature selection select features to keep or remove from the dataset, whereas dimensionality reduction create a projection of the data resulting in entirely new input features. As such, dimensionality reduction is an alternate to feature selection rather than a type of feature selection.

**4.1.1. Numerical Input, Numerical Output**

This is a regression predictive modeling problem with numerical input variables.

The most common techniques are to use a correlation coefficient, such as Pearson’s for a linear correlation, or rank-based methods for a nonlinear correlation.

• Pearson’s correlation coefficient (linear).

• Spearman’s rank coefficient (nonlinear)

**4.1.2. Numerical Input, Categorical Output**

This is a classification predictive modeling problem with numerical input variables. This might be the most common example of a classification problem. Again, the most common techniques are correlation based, although in this case, they must take the categorical target into account.

• ANOVA correlation coefficient (linear).

• Kendall’s rank coefficient (nonlinear).

Kendall does assume that the categorical variable is ordinal.

**4.1.3. Categorical Input, Numerical Output**

This is a regression predictive modeling problem with categorical input variables. This is a strange example of a regression problem (e.g. you would not encounter it often). Nevertheless, you can use the same “Numerical Input, Categorical Output” methods (described above), but in reverse.

**4.1.4 Categorical Input, Categorical Output**

This is a classification predictive modeling problem with categorical input variables. The most common correlation measure for categorical data is the [chi-squared test](https://machinelearningmastery.com/chi-squared-test-for-machine-learning/). You can also use mutual information (information gain) from the field of information theory.

* Chi-Squared test (contingency tables).
* Mutual Information.

In fact, mutual information is a powerful method that may prove useful for both categorical and numerical data, e.g. it is agnostic to the data types.

**4.2. FEATURE ENGINEERING**

**4.2.1. REQUIREMENT OF SCALING FOR CLASSIFICATION ALGORITHMS**

**DECISION TREES AND RANDOM FORESTS:**

These algorithms are not particularly sensitive to feature scaling. They make decisions based on threshold values for individual features, and the relative scale of features does not impact their performance significantly.

**K-NEAREST NEIGHBORS (KNN):**

KNN is sensitive to the scale of features because it relies on distances between data points. Features with larger scales may dominate the distance calculation, leading to biased results. It is often recommended to scale features before using KNN.

**SUPPORT VECTOR MACHINES (SVM) :**

SVMs can be sensitive to the scale of features, especially in the case of non-linear kernels. Feature scaling is generally recommended to ensure that all features contribute equally to the model.

**LOGISTIC REGRESSION :**

Logistic regression is not inherently sensitive to feature scaling. However, scaling may help in terms of convergence speed during optimization.

**NAIVE BAYES:** Naive Bayes algorithms are generally not sensitive to feature scaling. They are based on probabilities and independence assumptions.

**ENSEMBLE METHODS (E.G., ADABOOST, GRADIENT BOOSTING) :**

Ensemble methods like AdaBoost and Gradient Boosting are generally not very sensitive to feature scaling. However, it can depend on the specific base learner used within the ensemble methods.

**4.2.2. MULTICOLLINAERITY REMOVAL FOR CLASSIFICATION ALGORITHMS**

Multicollinearity, which refers to the high correlation between independent variables in a regression model, can be an issue for various machine learning algorithms, including classification algorithms. Here's a brief overview of the impact of multicollinearity on different types of classification algorithms:

**LOGISTIC REGRESSION:**

Logistic regression is particularly sensitive to multicollinearity. High correlation between independent variables can lead to unstable coefficient estimates, making it challenging to interpret the influence of individual features.

**LINEAR SUPPORT VECTOR MACHINES (SVM):**

Linear SVMs can also be affected by multicollinearity, as they rely on linear combinations of features. Multicollinearity may result in less stable and less interpretable models.

**DECISION TREES AND RANDOM FORESTS:**

Decision trees and random forests are less affected by multicollinearity. Decision trees make decisions based on individual features, and random forests can handle correlated features to some extent.

**K-NEAREST NEIGHBORS (KNN):**

KNN can be sensitive to multicollinearity, as it relies on distances between data points. High correlation between features may distort the distance metric and affect the performance of KNN.

**NAIVE BAYES:**

Naive Bayes algorithms assume independence between features, so they are less impacted by multicollinearity. However, extreme multicollinearity may still affect model performance.

**ENSEMBLE METHODS (E.G., ADABOOST, GRADIENT BOOSTING):**

Ensemble methods, like random forests or gradient boosting, are generally robust to multicollinearity. They can handle correlated features to some extent due to the combination of multiple weak learners.

**4.2.3. THE CLASS IMBALANCE PROBLEM HANDLING**

Several classification algorithms are capable of handling imbalanced classes. These algorithms are designed to address the challenge posed by datasets where the number of instances in one class is significantly lower than the other. Here are some algorithms that are known for their ability to handle imbalanced classes:

**RANDOM FOREST:**

Random Forest is an ensemble learning algorithm that builds multiple decision trees and combines their predictions. It is robust to imbalanced datasets and can handle them effectively.

**ADABOOST:**

AdaBoost is an ensemble method that combines weak learners to create a strong learner. It can be used with various base classifiers and is known for its ability to handle imbalanced classes.

**GRADIENT BOOSTING:**

Gradient Boosting is another ensemble technique that builds trees sequentially, with each tree correcting errors made by the previous ones. It is effective for imbalanced datasets.

**XGBOOST:**

XGBoost is an optimized version of gradient boosting and is widely used for imbalanced classification tasks. It includes regularization terms to control overfitting and is efficient in handling skewed class distributions.

**SUPPORT VECTOR MACHINES (SVM) WITH CLASS WEIGHTS:**

SVMs can be adapted for imbalanced classes by assigning different weights to classes. This allows the algorithm to penalize misclassifications of the minority class more heavily.

**LOGISTIC REGRESSION WITH CLASS WEIGHTS**:

Logistic Regression can be adjusted by assigning different weights to classes. This helps the model give more importance to the minority class.

**NAIVE BAYES:**

Naive Bayes algorithms, such as Gaussian Naive Bayes or Bernoulli Naive Bayes, are known for their simplicity and can perform reasonably well on imbalanced datasets.

**ENSEMBLE TECHNIQUES IN GENERAL:**

Ensemble methods, in general, are often effective for handling imbalanced classes because they combine predictions from multiple models, mitigating the impact of skewed class distributions.

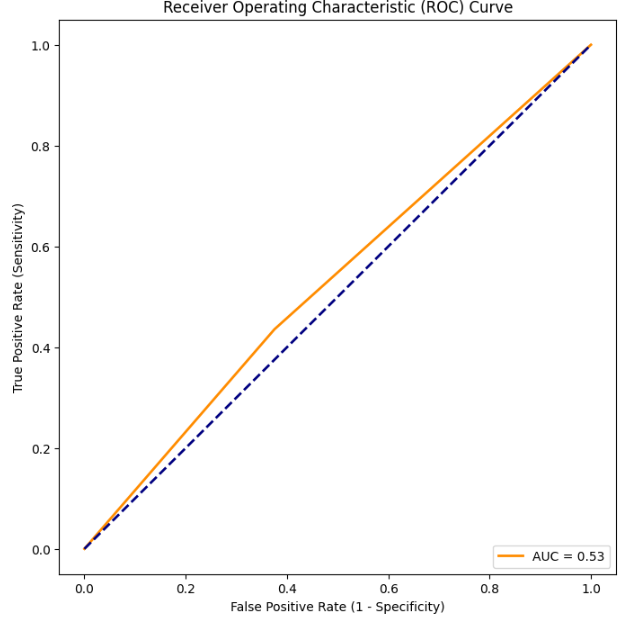
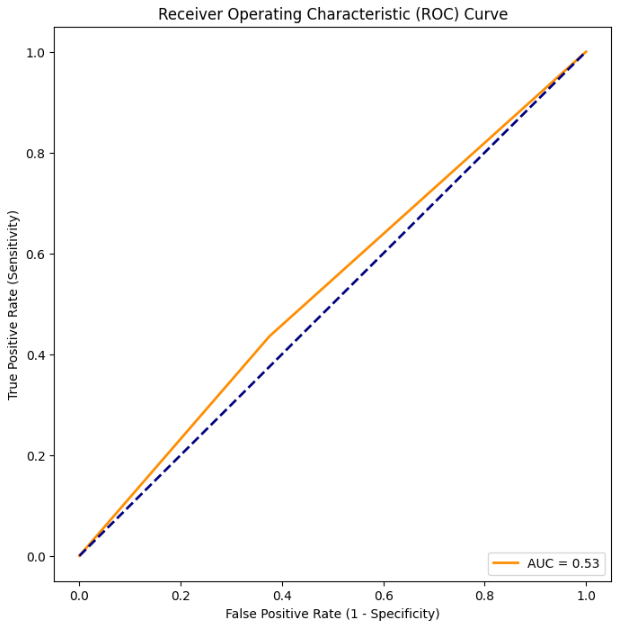
1. **MODEL BUILDING**

**MODEL - LOGISTIC REGRESSION**

Simple logistic regression, by itself, does not inherently address the issue of class imbalance. Logistic regression is designed to model the relationship between independent variables and the log-odds of the dependent variable, and its primary focus is on estimating coefficients that provide the best fit to the observed data.

If you have a class imbalance problem (i.e., one class has significantly fewer samples than the other), logistic regression may be biased toward the majority class. This is because the model aims to maximize the likelihood of the observed data, and if one class is dominant, the model might be more inclined to predict instances as belonging to the majority class.

**MODEL 1- LOGISTIC REGRESSION WITH CLASS WEIGHTS**

True Positive Rate (TPR): 0.4240890688259109

True Negative Rate (TNR): 0.624709042076992

Precision: 0.09086965950986771

Negative Predictive Value (NPV): 0.9246058036305816

F1 Score: 0.16547640510568995

Accuracy: 0.6084059878269452

False Positive Rate (FPR): 0.37529095792300804

False Negative Rate (FNR): 0.5759109311740891

**MODEL 2- LOGISTIC REGRESSION WITH OVERSAMPLING**

which is the best oversampling strategy?

Table : Different kinds of smoting and its applications

|  |  |
| --- | --- |
| METHOD | ADVANTAGES |
| Random Oversampling (RandomOverSampler) | Advantages: Simple and easy to implement.  Considerations: May lead to overfitting on the minority class if the oversampling is too aggressive. |
| SMOTE (Synthetic Minority Over-sampling Technique) | Advantages: Generates synthetic samples by interpolating between existing minority class instances.  Considerations: May be sensitive to noisy data and may not perform well if the feature space is high-dimensional. |
| ADASYN (Adaptive Synthetic Sampling): | Advantages: Similar to SMOTE but adapts the sampling density according to the local distribution of the minority class.  Considerations: Can be computationally more expensive than SMOTE. |
| BorderlineSMOTE: | Advantages: Focuses on borderline instances, which are samples that are close to the decision boundary.  Considerations: May be less sensitive to noisy data than standard SMOTE. |
| SMOTE-ENN (SMOTE with Edited Nearest Neighbors): | Advantages: Combines SMOTE with the removal of noisy samples using ENN.  Considerations: ENN may remove informative minority class samples. |

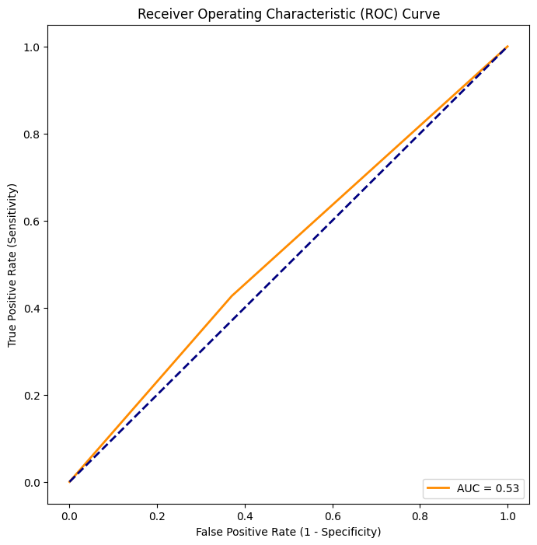
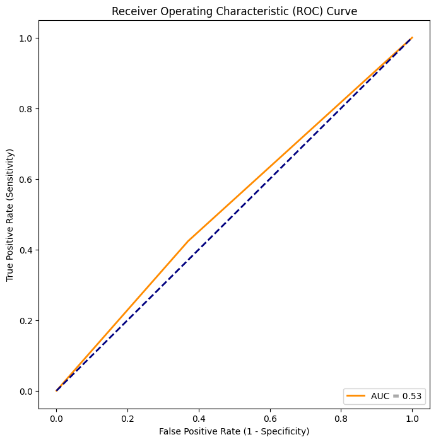
 

Fig ROC AUC using SMOTE by increasing minority class by 30% left side- Training performance, right-side Testing performance

|  |  |  |
| --- | --- | --- |
| ACTUAL/ PREDICTED | 0 | 1 |
| 0 | 34890 | 20960 |
| 1 | 2788 | 2152 |

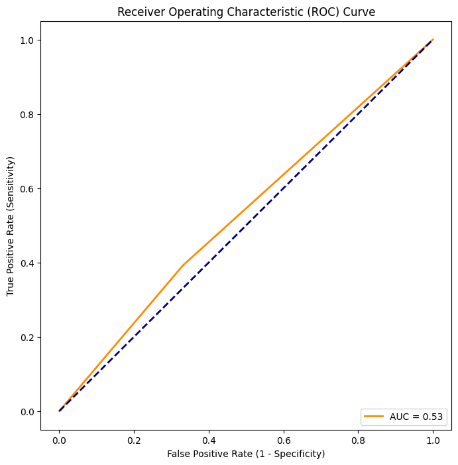
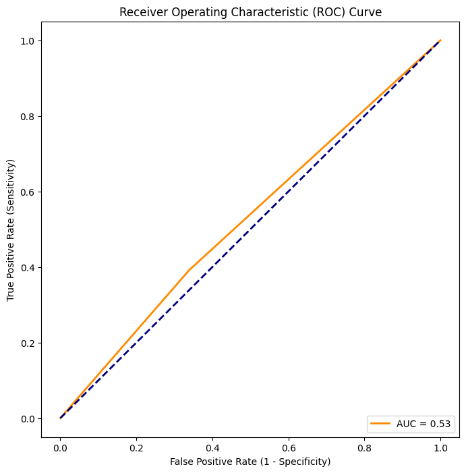
 

Fig : ROC AUC using SMOTEENN by increasing minority class by 30% left side- Training performance, right-side Testing performance

|  |  |  |
| --- | --- | --- |
| ACTUAL/ PREDICTED | 0 | 1 |
| 0 | 36922 | 18928 |
| 1 | 3005 | 1935 |

True Positive Rate (TPR): 0.43562753036437246

True Negative Rate (TNR): 0.624709042076992

Precision: 0.09311180339217723

Negative Predictive Value (NPV): 0.9260045649981421

F1 Score: 0.16920924375408086

Accuracy: 0.6093436420463892

False Positive Rate (FPR): 0.37529095792300804

False Negative Rate (FNR): 0.5643724696356275

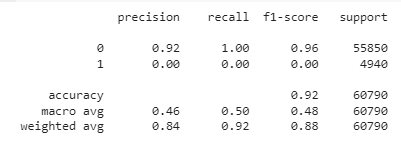
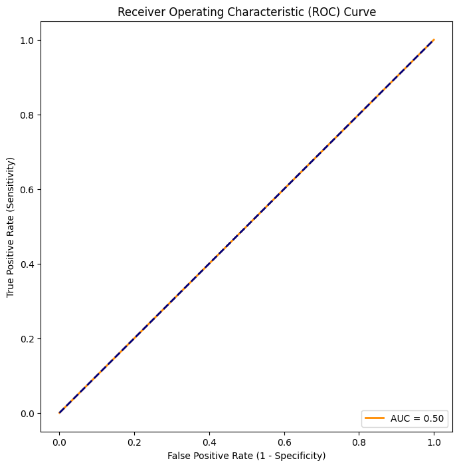
A TPR of 0.4356 indicates that approximately 43.56% of the users who are actually likely to default were correctly identified by the model. In other words, the model is able to capture a portion of the users who are at risk of defaulting.

**MODEL - NAIVE BAYES**

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes independence among features and calculates the probability of a class given the input features. Widely used for text classification and spam filtering, it is computationally efficient and performs well in high-dimensional spaces.

**MODEL -1 - GAUSSIAN NAIVE BAYES**

* Normal Distribution (Gaussian): The algorithm assumes that the features within each class follow a Gaussian (normal) distribution. Checking the distribution of your features, especially if they deviate significantly from normality, can be helpful.
* Homoscedasticity: The algorithm assumes that the variance of each feature is the same across different classes. Checking for homoscedasticity (equal variance) can be important, especially if certain features have significantly different variances in different classes.
* Missing Values: Naive Bayes assumes that missing values are missing completely at random. If your dataset has missing values, it's essential to handle them appropriately or assess whether they follow a random pattern.
* Outliers: Outliers can affect the mean and standard deviation calculations in Gaussian Naive Bayes. Checking for and handling outliers may be necessary.
* Class Priors: Naive Bayes assumes that the prior probabilities of different classes are known. If you have imbalanced classes, it's important to consider how the algorithm might be influenced by the prior probabilities and whether adjustments are needed

An AUC (Area Under the Receiver Operating Characteristic Curve) value of 0.50 typically indicates that the model is performing no better than random chance. The ROC curve is a graphical representation of the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at various threshold settings. An AUC of 0.50 corresponds to a diagonal line in the ROC space, which is the line of no discrimination.

In practical terms:

A model with an AUC of 0.50 is not making accurate predictions, and its performance is equivalent to random guessing.

The model's ability to distinguish between positive and negative instances is no better than chance.

If you encounter an AUC of 0.50, it suggests that the model is not effectively discriminating between the classes, and you may need to revisit your model, features, or data preprocessing to improve its performance.

**MODEL - DECISION TREE CLASSIFIER**

Decision Tree is a versatile and interpretable supervised machine learning algorithm used for both classification and regression tasks. It works by recursively partitioning the data into subsets based on the values of input features. The goal is to create a tree-like structure where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents the final prediction.

**ASSUMPTIONS**

1. Feature Relevance:

Features (variables) included in the dataset should be relevant to the task at hand. Including irrelevant or redundant features can introduce noise and negatively impact model performance.

1. Feature Types:

Machine learning models, including Decision Trees, can handle a mix of numerical and categorical features. Ensure that your features are appropriately encoded for the model to interpret them correctly.

Missing Data:

Deal with missing values appropriately. Decision Trees can handle missing values, but imputing or addressing missing data can enhance model performance.

1. Outliers:

Decision Trees are robust to outliers, but extreme outliers can impact the splits made in the tree. It's essential to understand the nature of outliers in your data and decide whether to treat or keep them.

1. Variable Scale:

Decision Trees are not sensitive to the scale of numerical variables. However, for other models that are sensitive to scale (e.g., linear models), standardization or normalization may be necessary.

1. Independence of Observations:

The observations (rows) in your dataset are assumed to be independent. This means that the value of one observation should not be influenced by the values of other observations.

1. Homoscedasticity (for Regression):

In regression tasks, homoscedasticity assumes that the variance of the residuals is constant across all levels of the independent variable. Decision Trees are less concerned with this assumption compared to linear regression models.

1. Target Variable Definition:

Clearly define and understand the nature of your target variable. Whether it's binary classification, multi-class classification, or regression, the model's task depends on the type of target variable.