Great Learning

BENGALURU CAMPUS

PROJECT TITLE:

CREDIT WORTHINESS EVALUATION

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.

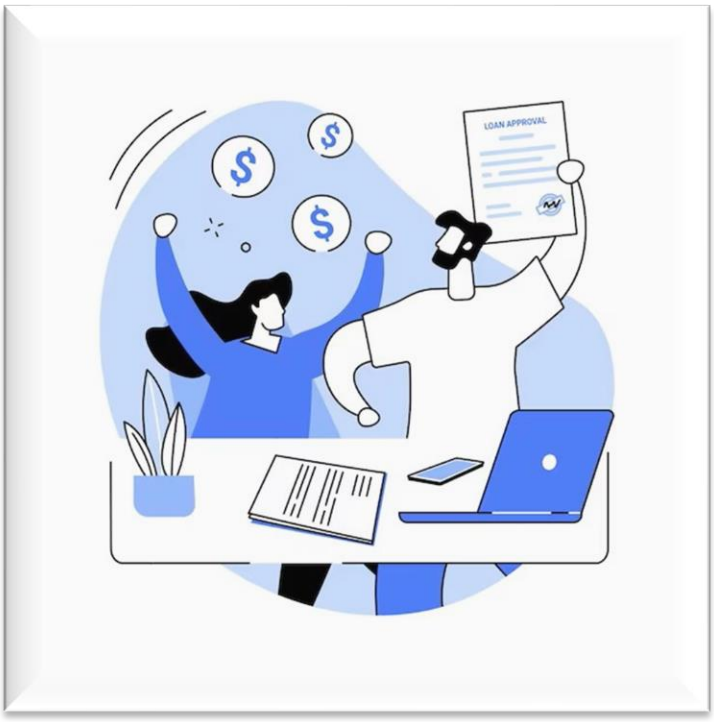
Through the careful examination of various attributes, our research seeks to enhance the accuracy and efficiency of credit evaluations. The findings from this study have the potential to revolutionize the loan approval process, allowing financial institutions to optimize risk management and better serve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

TABLE OF CONTENTS

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.

2. 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.

3. 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.

2. 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.

3. 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.

4. 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

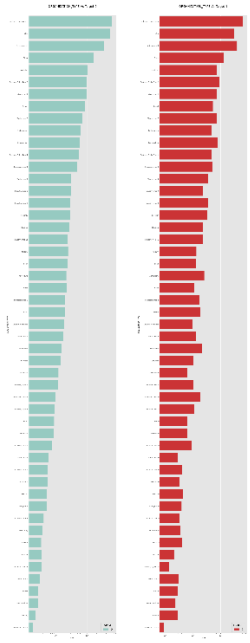
112	FLAG_DOCUMENT_17	Did client provide document 17
113	FLAG_DOCUMENT_18	Did client provide document 18
114	FLAG_DOCUMENT_19	Did client provide document 19
115	FLAG_DOCUMENT_20	Did client provide document 20
116	FLAG_DOCUMENT_21	Did client provide document 21
117	AMT_REQ_CREDIT_BUREAU_HOUR	Credit Bureau 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. EXPLORATORY DATA ANALYSIS

5.1UNIVARIANTE ANALYSIS OF CATEGORICAL COLUMNS WITH RESPECT TO TARGET VARIABLE

- It seems like Female clients applied higher than male clients for loan
- 66.6% Female clients are non-defaulters while 33.4% male clients are non-defaulters.
- 57% Female clients are defaulters while 42% male clients are defaulters.

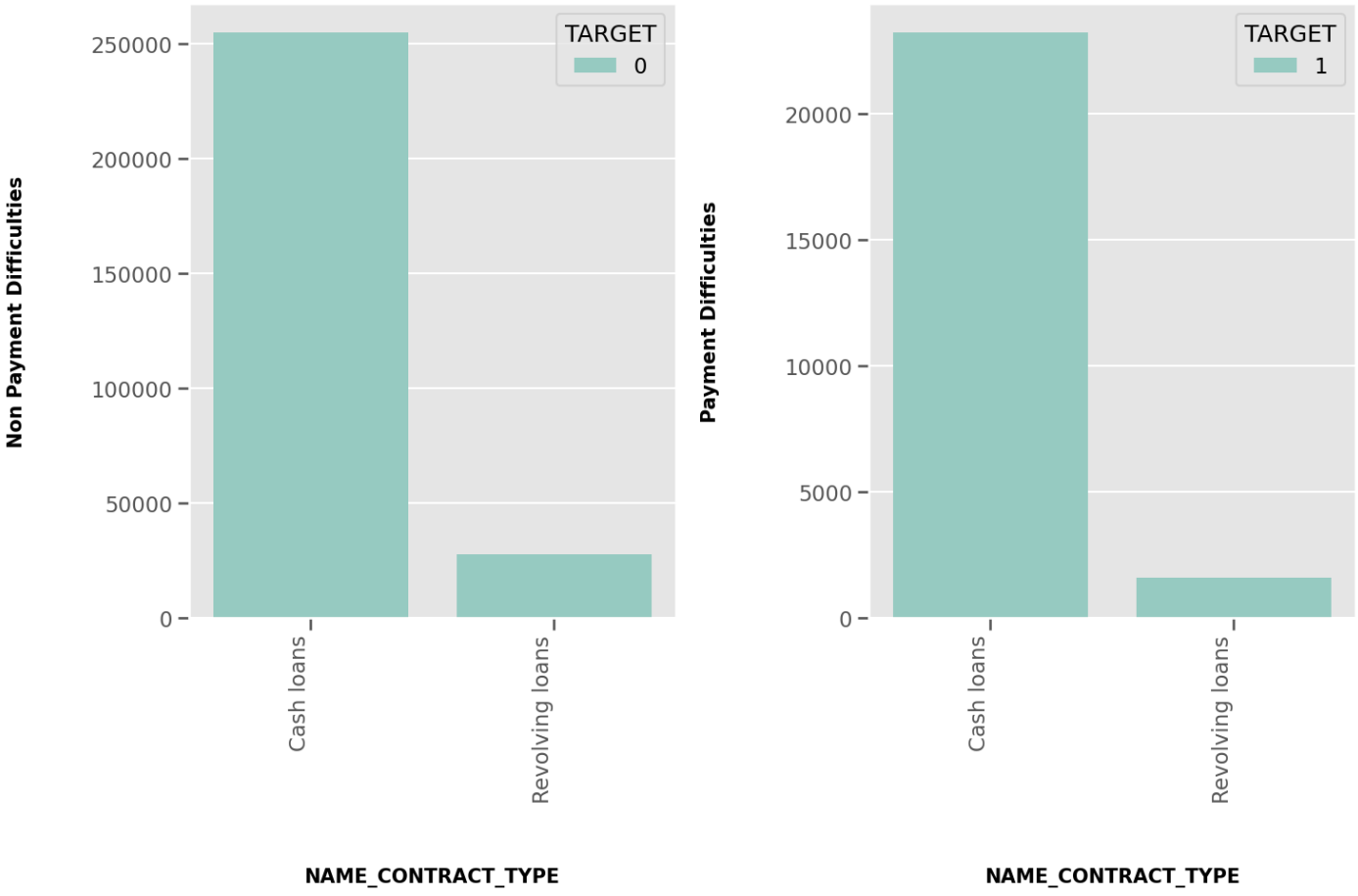
ORGANIZATION'S distribution based on Target 0 and Target 1



- (Defaulters as well as Non-defaulters) Clients with ORGANIZATION_TYPE Business Entity Type 3, Self-employed, Other ,Medicine, Government,Business Entity Type 2 applied the most for the loan as compared to others

- (Defaulters as well as Non-defaulters) Clients having ORGANIZATION_TYPE Industry: type 13, Trade: type 4, Trade: type 5, Industry: type 8 applied lower for the loan as compared to others.

SIMILARLY FOR BELOW CATOGRICAL COLUMNS

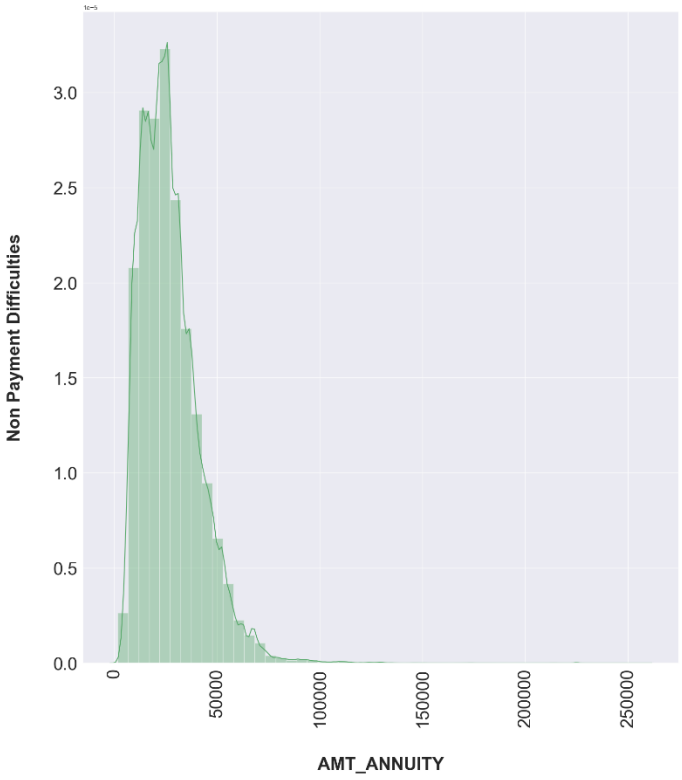


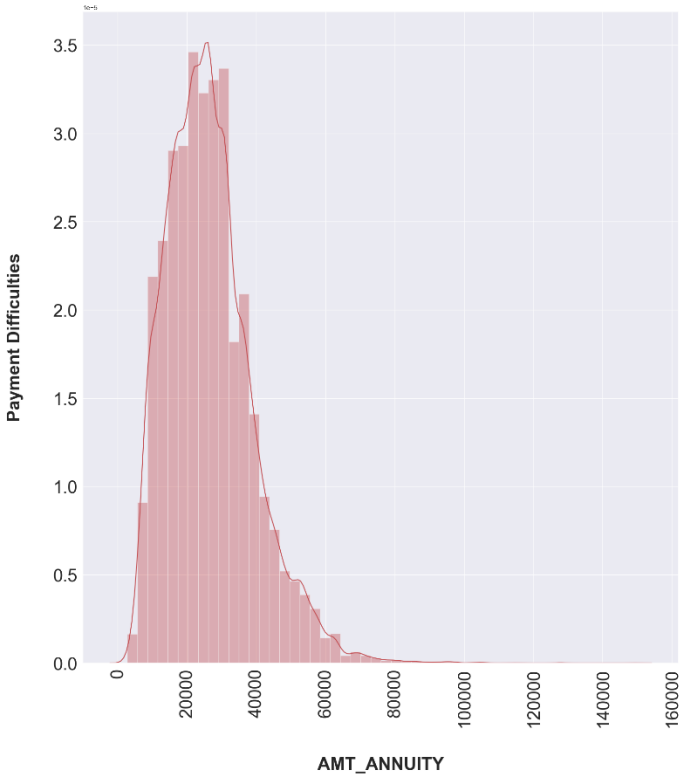
1. NAME_CONTRACT_TYPE :
    - **Most of the clients** have applied for Cash Loan while **very small proportion* have applied for Revolving loan for both Defaulters as well as Non-defaulters.
2. NAME_INCOME_TYPE:
    - Clients who applied for loans were getting income by **Working Professional** are more likely to apply for the loan, highest being the Working class category .
    - **State servant** are less likely to apply for loan .
    - **Working category** have high risk to default.
    - **State Servant** is at Minimal risk to default.
3. NAME_EDUCATION_TYPE:
    - Clients having education Secondary or Secondary Special are more likey to apply for the loan.
    - Clients having education Secondary or Secondary Special have higher risk to default.Other education types have minimal risk.
4. NAME_FAMILY_STATUS :
    - **Married Clients** seems to be applied most for the loan compared to others for both Defaulters and Non-Defaulters.
    - In case of Defaulters,Clients having single relationship are **less risky**
    - In case of Defaulters, Widows shows **Minimal risk**.
5. NAME_HOUSING_TYPE:

- From the bar chart, it is clear that Most of the clients **own a house or living in a apartment** for both Defaulters and Non-Defaulters.

6. OCCUPATION_TYPE:
   - Laborers have applied the most for the loan in case of Defaulters and Non-Defaulters.
   - Laborers being highest followed by Cleaning Staff have high risk to default.
7. WEEKDAY_APPR_PROCESS_START:
   - There is no considerable difference in days for both Defaulters and Non-defaulters.
8. FLAG_OWN_REALTY:
   - Clients who owns a realty have applied the most for the loan.

UNIVARIANTE ANALYSIS OF NUMERICAL COLUMNS WITH RESPECT TO TARGET VARIABLE

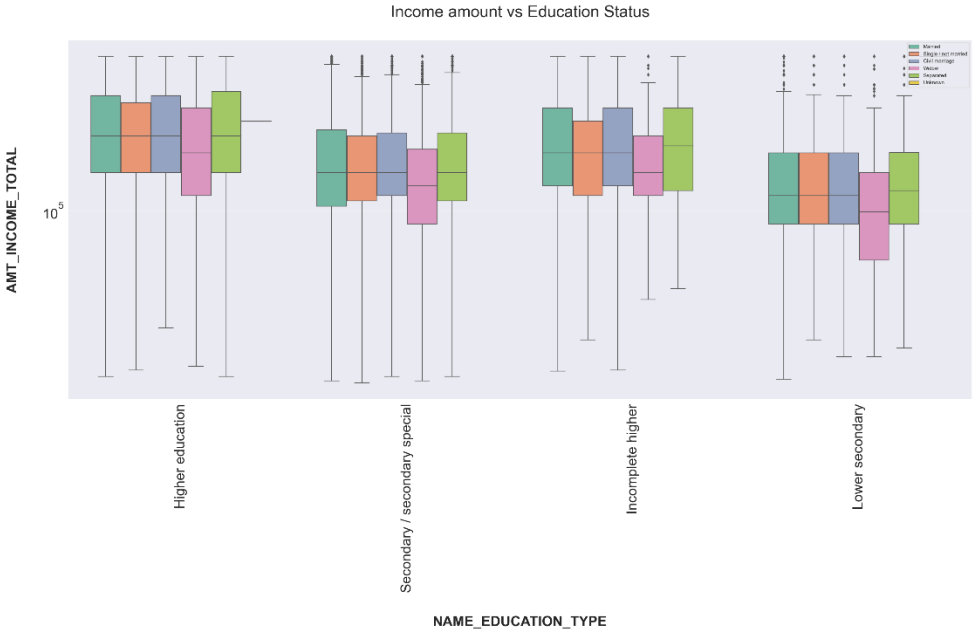




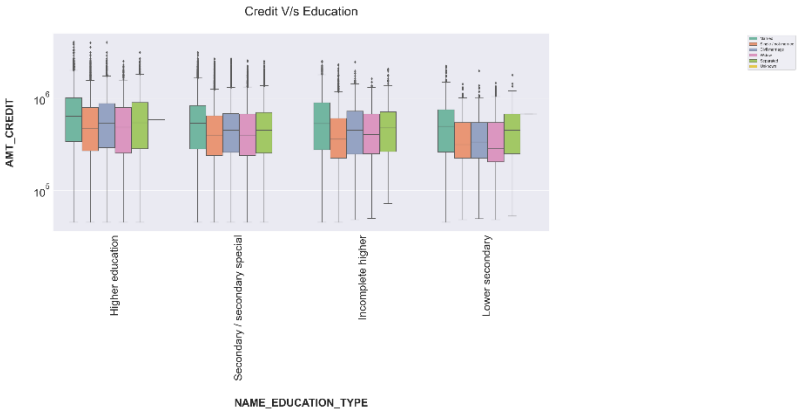
- People with target one has largely staggered income as compared to target zero. Dist. plot clearly shows that the shape in Income total, Annuity, Credit and Good Price are similar for Target 0 and similar for Target 1.

- The plots are also highlighting that people who have difficulty in paying back loans with respect to their income, loan amount, price of goods against which loan is procured and Annuity.

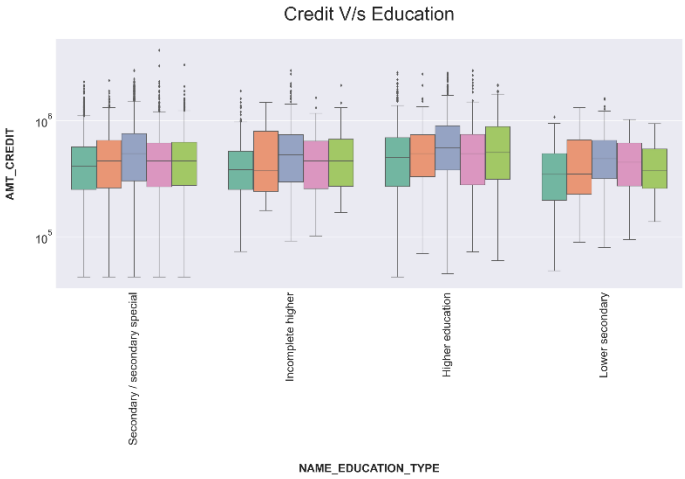
- Dist. plot highlights the curve shape which is wider for Target 1 in comparison to Target 0 which is narrower with well defined edges.



- Widow Client with Academic degree have a **very few outliers** and doesn't have First and Third quartile.Also Clients with all type of **family status having academic degree** have very **less outliers** as compared to other type of **education**.
- Income of the clients with all type of family status having rest of the education type lie Below the First quartile i.e. 25%
- Clients having Higher **Education**, Incomplete Higher Education, Lower Secondary Education and Secondary/Secondary Special have a **higher number of outliers**.
- From the above figure we can say that, some of the clients **having Higher Education tend to have highest Income compared to others.**
- Though some of the clients who **haven't completed their Higher Education** tend to have **higher Income**.
- Some of the clients having **Secondary/Secondary Sepcial Eduacation** tend to **have higher income**.



- **Income amount** for Married clients with academic degree is much lesser as compared to others.
- **(Defaulter)** Clients have relatively less income as compared to **Non-defaulters.**



- Married client with academic applied for **higher credit loan**. And doesn't have outliers.Single clients eith **academic degree have a very slim boxplot with no outliers**.

- Some of the clients with Higher Education, Incomplete Higher Education, Lower Secondary Education and Secondary/Secondary Special Eduacation are more likely to take **high amount of credit loan**.

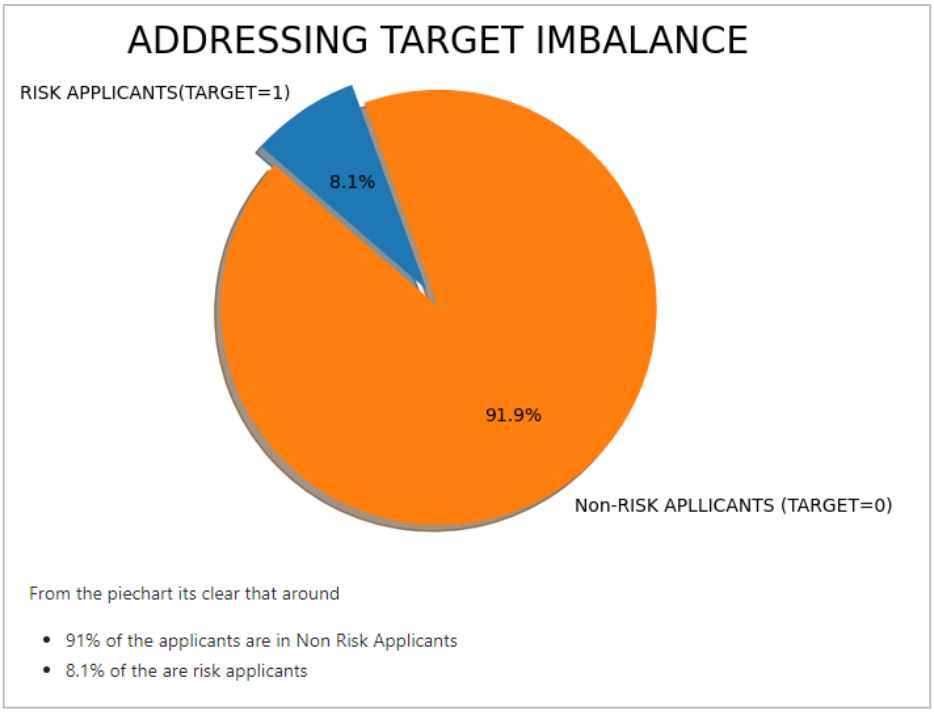
6. DATA PREPARATION

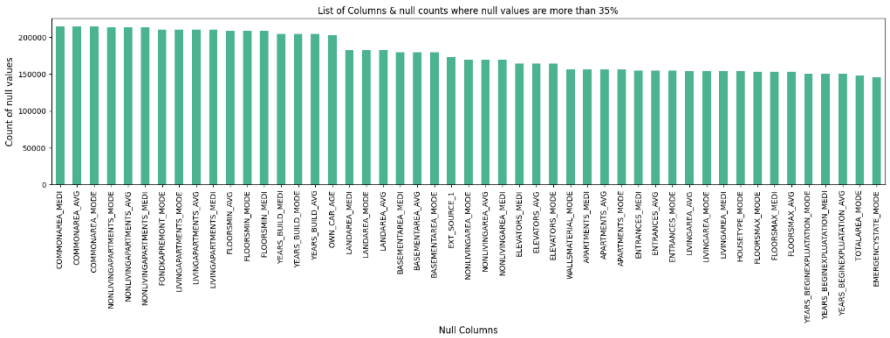
Data preparation involves transforming raw data into a form that is more appropriate for modeling.

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.

6.1. HANDING NULL VALUES





Null value columns in the data, about 50 columns in the data are having null value% >35

HANDLING DAYS COLUMN

```
df[['DAYS_BIRTH','DAYS_EMPLOYED','DAYS_REGISTRATION','DAYS_ID_PUBLISH','DAYS_LAST_PHONE_CHANGE']]
```

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_REGISTRATION	DAYS_ID_PUBLISH	DAYS_LAST_PHONE_CHANGE
0	-9461	-637	-3648.0	-2120	-1134.0
1	-16765	-1188	-1186.0	-291	-828.0
2	-19046	-225	-4260.0	-2531	-815.0
3	-19005	-3039	-9833.0	-2437	-617.0
4	-19932	-3038	-4311.0	-3458	-1106.0
...	...	...	...	...	...
307506	-9327	-236	-8456.0	-1982	-273.0
307507	-20775	365243	-4388.0	-4090	0.0
307508	-14966	-7921	-6737.0	-5150	-1909.0
307509	-11961	-4786	-2562.0	-931	-322.0
307510	-16856	-1262	-5128.0	-410	-787.0

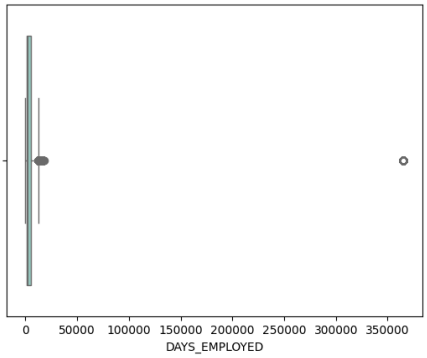
These columns were having – sign ,so we simply used abs value to resolve the issue, later part of the project we will convert these days into years

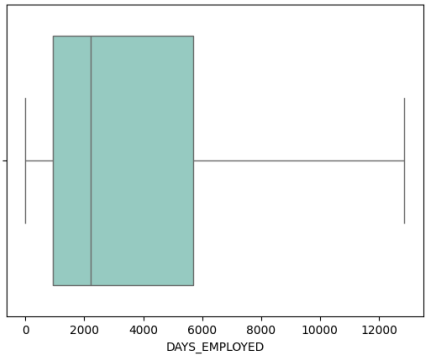
IMPUTATION ON THE REQUIRED COLUMNS

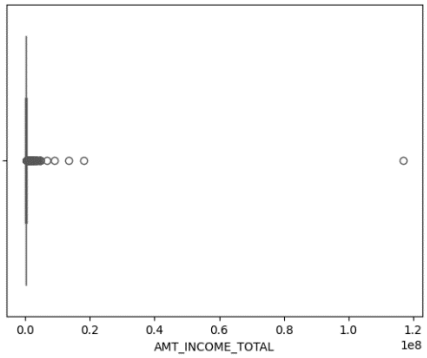
1. OCCUPATION TYPE

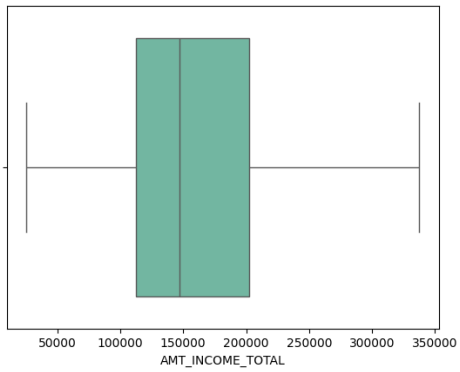
1. **OCCUPATION TYPE** IS HAVING 31% Null values, we collected other columns related to OCCUPATION_TYPE

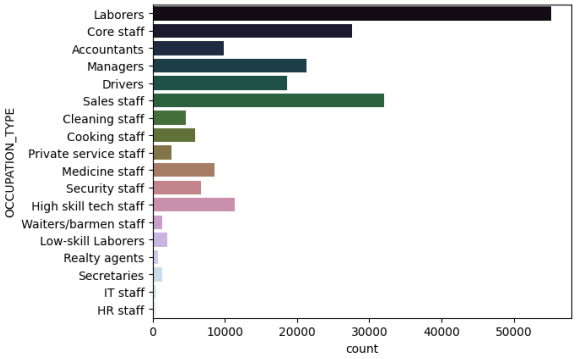
```
AMT_INCOME_TOTAL         0.000000
OCCUPATION_TYPE         31.345545
ORGANIZATION_TYPE        0.000000
NAME_INCOME_TYPE         0.000000
NAME_EDUCATION_TYPE      0.000000
DAYS_EMPLOYED            0.000000
dtype: float64
```











FILLING MISSING VALUES WITH RANDOM FOREST CLASSIFIER

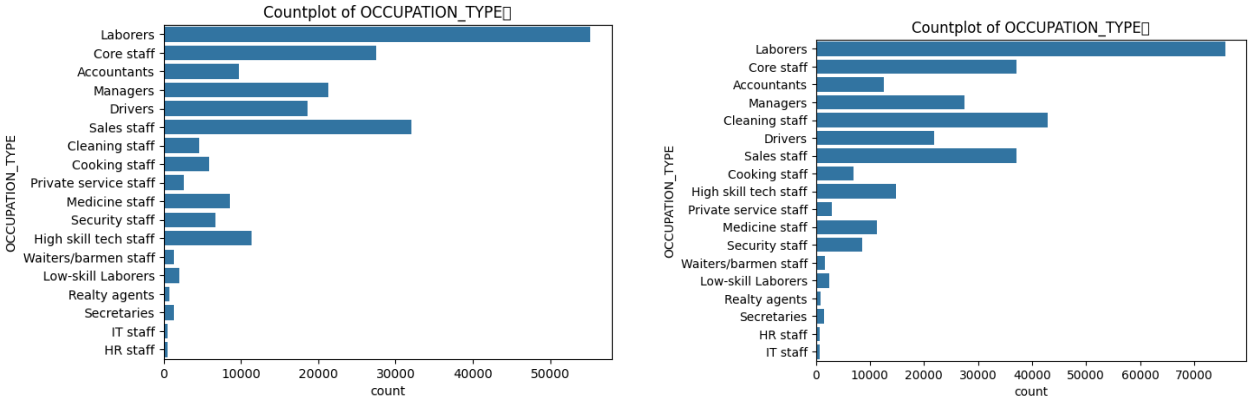
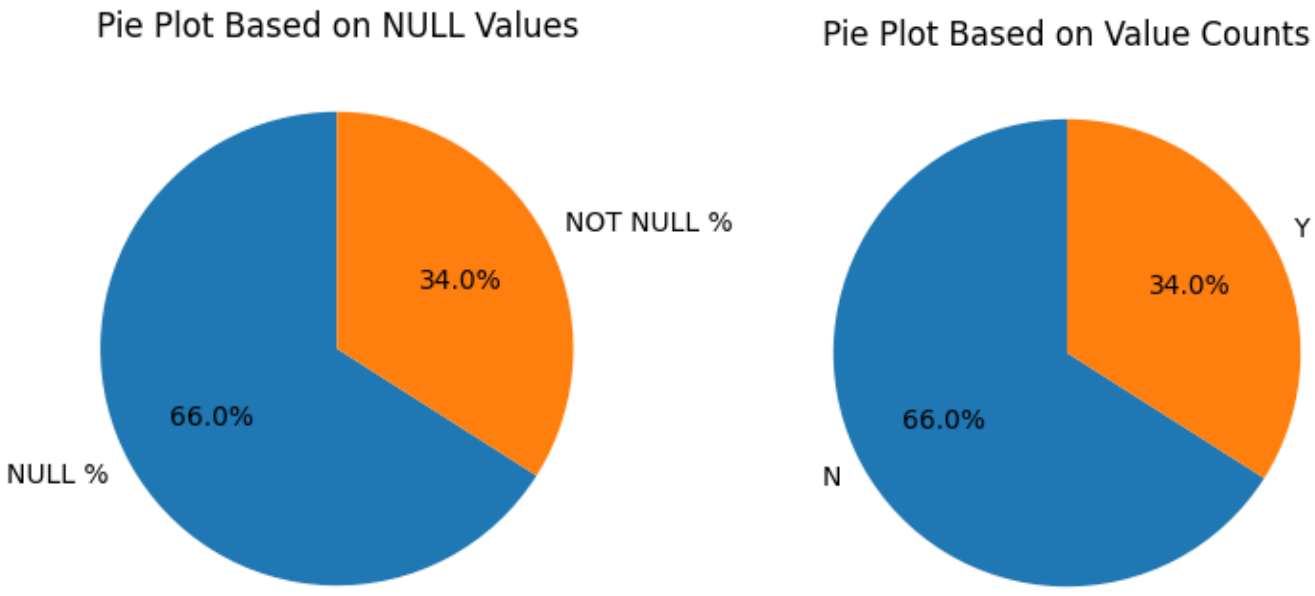


Fig Filling missing values with random forest classifier before and after imputation of missing values, the graph almost looks like the same

OWN_CAR_AGE



```
df[df['difference'] != 0][["FLAG_OWN_CAR","OWN_CAR_AGE","CODE_GENDER","DAYS_BIRTH","DAYS_EMPLOYED","OCCUPATION_TYPE","CNT_F
```

	FLAG_OWN_CAR	OWN_CAR_AGE	CODE_GENDER	DAYS_BIRTH	DAYS_EMPLOYED	OCCUPATION_TYPE	CNT_FAM_MEMBERS
30875	Y	NaN	M	16030	1889	Managers	2.0
181100	Y	NaN	F	18667	4442	Medicine staff	2.0
217391	Y	NaN	M	13502	2256	Drivers	2.0
229703	Y	NaN	F	13021	944	Accountants	3.0
236702	Y	NaN	F	10573	180	High skill tech staff	2.0

5 columns are having people are having car but their car age is not given

Liner regression for imputation

FLAG_OWN_CAR_encoded	13.2618	0.218	60.840	0.000	12.835	13.689
CODE_GENDER_encoded	-0.1168	0.075	-1.554	0.120	-0.264	0.030
DAYS_BIRTH	-5.442e-05	1.16e-05	-4.710	0.000	-7.71e-05	-3.18e-05
DAYS_EMPLOYED	5.145e-05	1.25e-05	4.126	0.000	2.7e-05	7.59e-05
OCCUPATION_TYPE_encoded	0.0033	0.009	0.366	0.714	-0.014	0.021
CNT_FAM_MEMBERS	-0.2125	0.041	-5.210	0.000	-0.292	-0.133

Rest of the columns were filled with 0 values

HANDLING OTHER MISSING VALUES <30 %

AMT_REQ_CREDIT_BUREAU

Features related to this columns are

```
"AMT_REQ_CREDIT_BUREAU_HOUR"
"AMT_REQ_CREDIT_BUREAU_DAY",
"AMT_REQ_CREDIT_BUREAU_WEEK",
"AMT_REQ_CREDIT_BUREAU_MON",
"AMT_REQ_CREDIT_BUREAU_QRT",
"AMT_REQ_CREDIT_BUREAU_YEAR"
```

Here the data was MISSING COMPLETELY AT RANDOM : Imputation Method used was MICE Imputation

Multiple Imputation by Chained Equations (MICE):

Method: Iteratively impute missing values for each variable conditional on other variables using a model. Pros: Captures relationships between variables, provides multiple imputations for uncertainty estimation. Cons: Computationally intensive.

NAME_TYPE_SUITE

```
Unaccompanied      248335
Family              40132
Spouse, partner     11363
Children             3267
Other_B              1769
Other_A               865
Group of people       271
Name: NAME_TYPE_SUITE, dtype: int64
```

This was a classification column, so we went with Random Forest Classifier

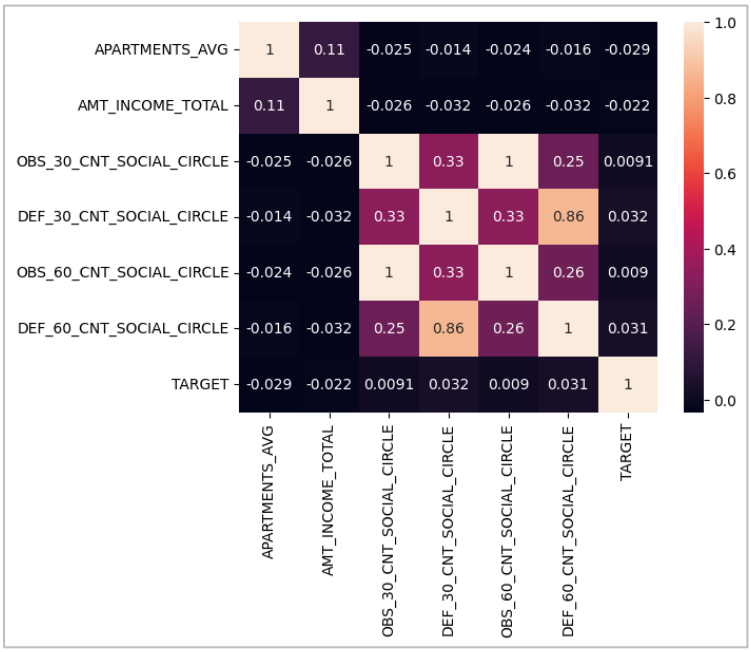
EXT_SOURCE_2

Here also Mice imputation was done

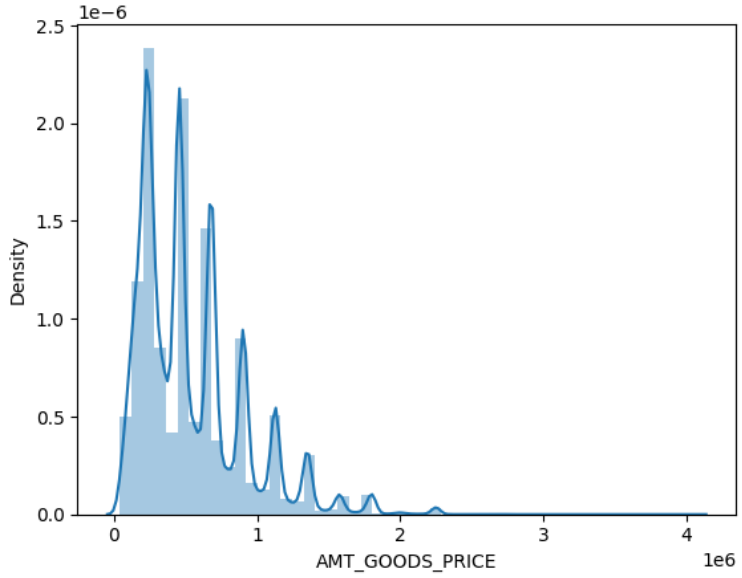
CNT_SOCIAL_CIRCLE:

How many observation of client's social surroundings defaulted on 30 DPD (days past due)

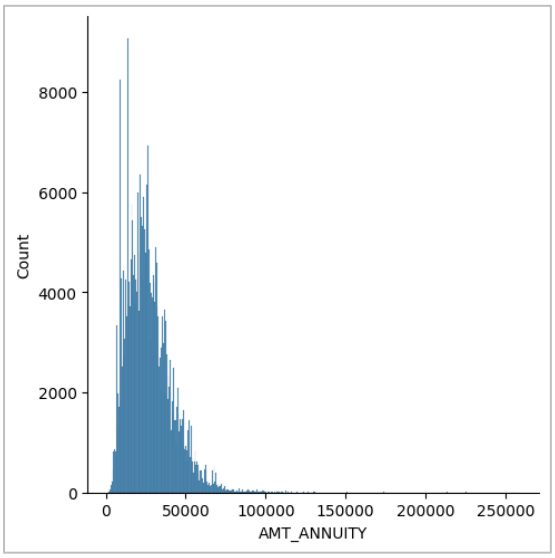
```
"APARTMENTS_AVG",
"FLAG_OWN_REALTY",
"OCCUPATION_TYPE",
"ORGANIZATION_TYPE",
"AMT_INCOME_TOTAL",
"NAME_EDUCATION_TYPE",
"OBS_30_CNT_SOCIAL_CIRCLE",
"DEF_30_CNT_SOCIAL_CIRCLE",
"OBS_60_CNT_SOCIAL_CIRCLE",
"DEF_60_CNT_SOCIAL_CIRCLE",
"TARGET"
```



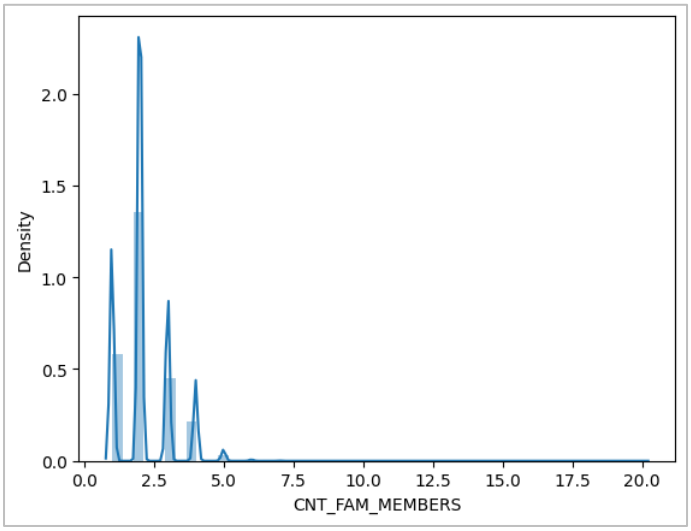
CNT_FAM_MEMBERS -



AMT_ANNUITY



CNT_FAM_MEMBERS



NULL VALUE GREATER THAN 30

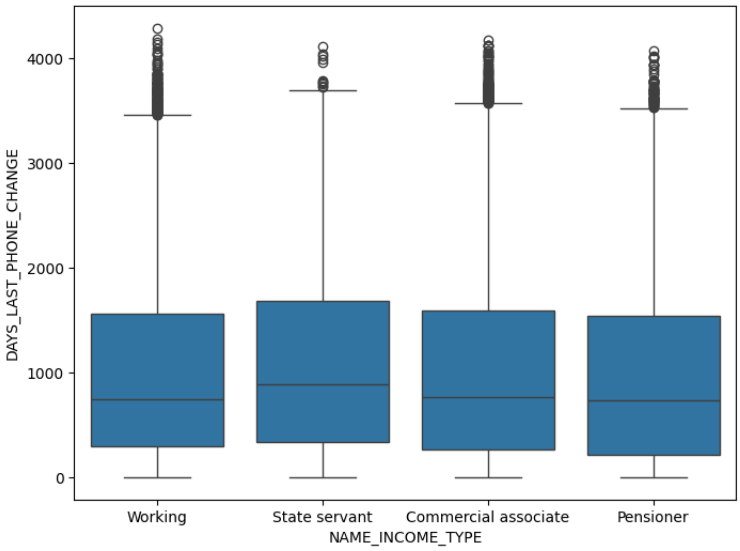
```
df["FLAG_MOBIL"].value_counts()
```

```
1    307291
0         1
Name: FLAG_MOBIL, dtype: int64
```

"DAYS_LAST_PHONE_CHANGE"

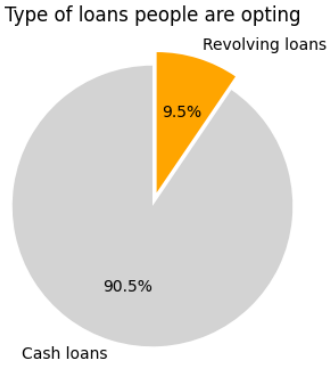
```
"DAYS_LAST_PHONE_CHANGE","FLAG_EMP_PHONE","FLAG_WORK_PHONE","FLAG_CONT_MOBILE","FLAG_PHONE"
```

	Feature	VIF
0	DAYS_LAST_PHONE_CHANGE	1.000020
1	FLAG_EMP_PHONE	1.000007
2	FLAG_WORK_PHONE	1.000006
3	FLAG_CONT_MOBILE	1.000033
4	FLAG_PHONE	1.000015

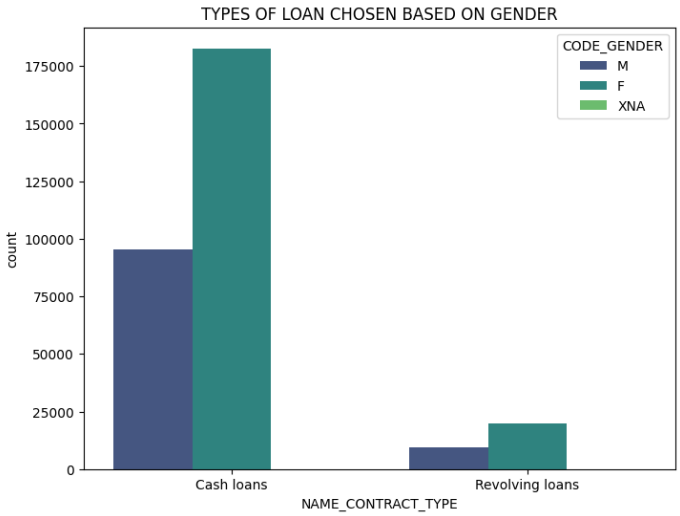


6.2. OUTLIER HANDLING

1. NAME_CONTRACT_TYPE



- almost 90% of people are opting for Cash loans
- NO outliers are present

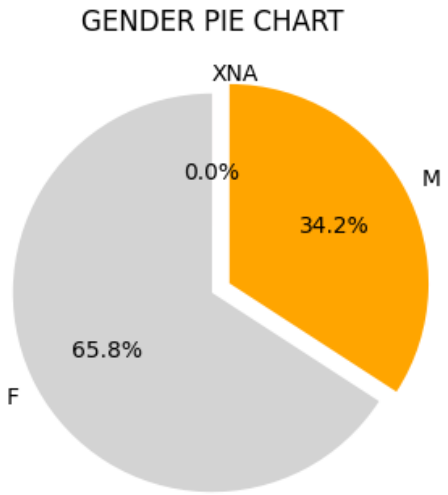


WAIT what is that XNA ??

- When I done base model, I found XNA in multiple categorical columns
- Both the cases Females are preferring to opt for loans compared to males
- based on value_counts

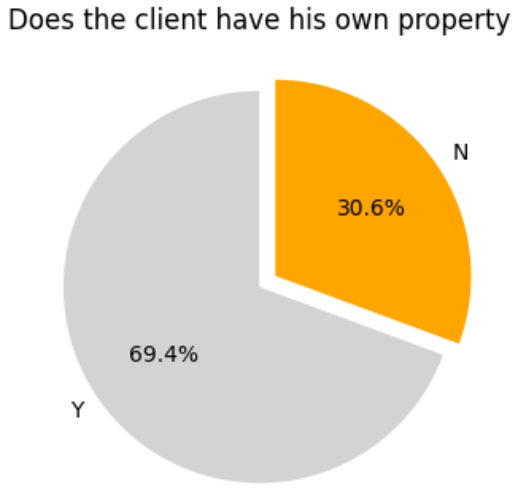
This column we have dropped cause its not giving away that much of an information

2. CODE_GENDER



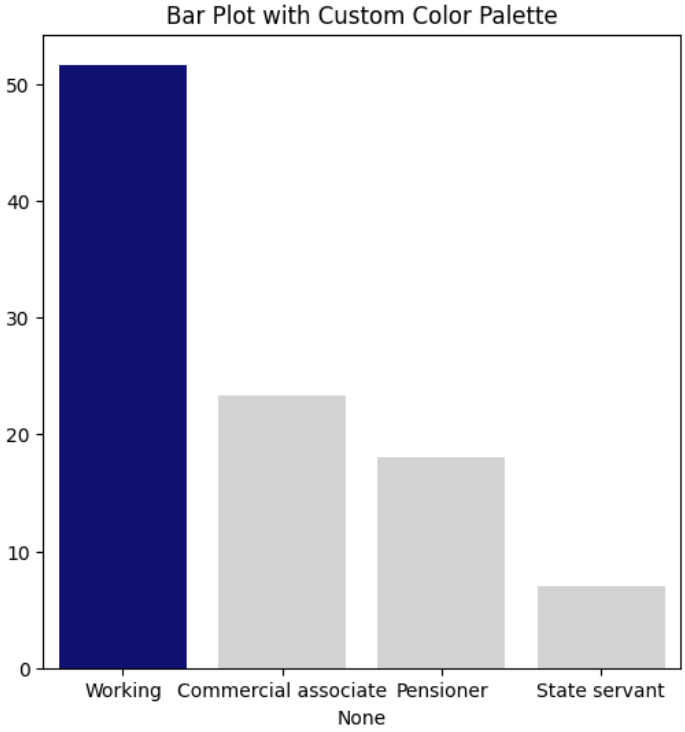
- 1/3rd of PEOPLE applied are only males
- 4 values out of total are only XNA
- before cleaning also there were no NULL values in the columns
- In the context of credit or financial datasets, finding "XNA" typically refers to a code or label used to represent "Not Applicable" or "Not Available." It's a common practice in data preprocessing to replace missing or undefined values with a placeholder like "XNA" to indicate that the information is not applicable or not provided.
- This missing value is MAR , the applicant is not ready to reveal the gender
- replacing this value the mode

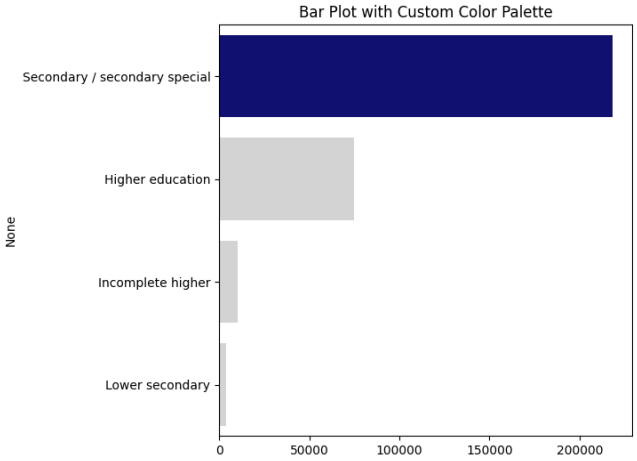
3. FLAG_OWN_REALTY

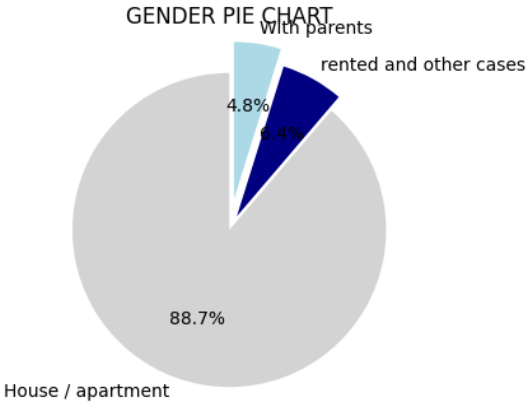


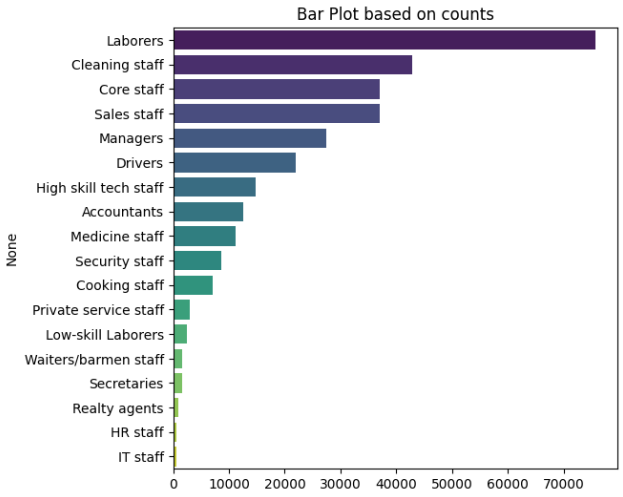
- Almost 70 % of people wons the property
- only two category
- No outliers are present
- Label encoding preffered

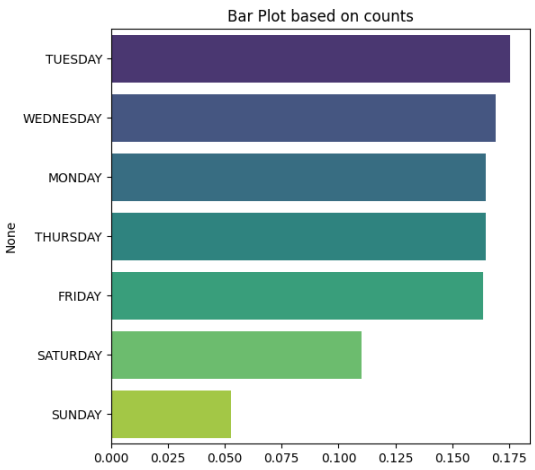
6. NAME_INCOME_TYPE

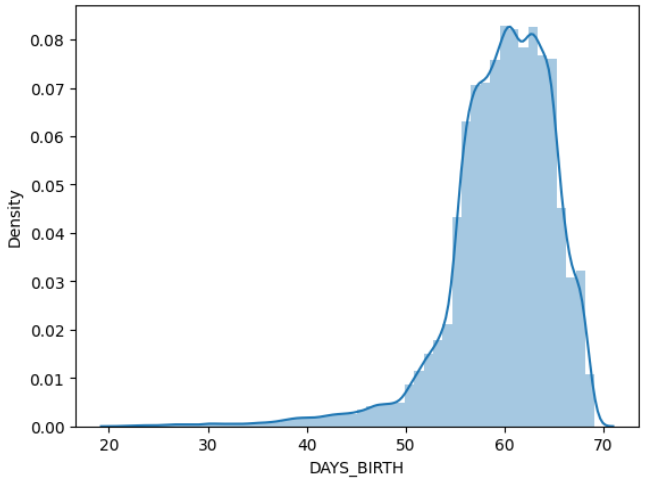












6.3. FEATURE SELECTION

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.

6.3.1. 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)

6.3.2. 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.

6.3.3. 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.

6.3.4 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. 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.

6.3.5. FEATURE ENGINEERING

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.

6.3.6. 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.

6.3.7 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.

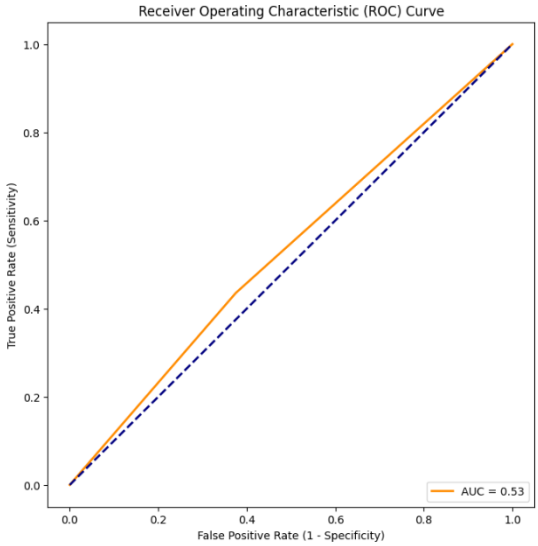
7. MODEL BUILDING

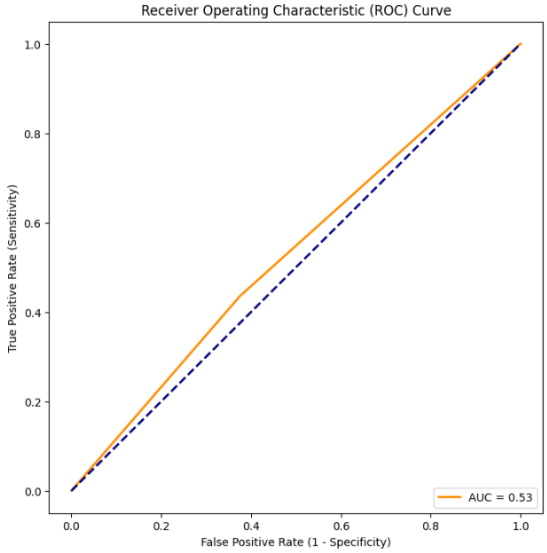
7.1 LOGISTIC REGRESSION

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
N.egative 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.

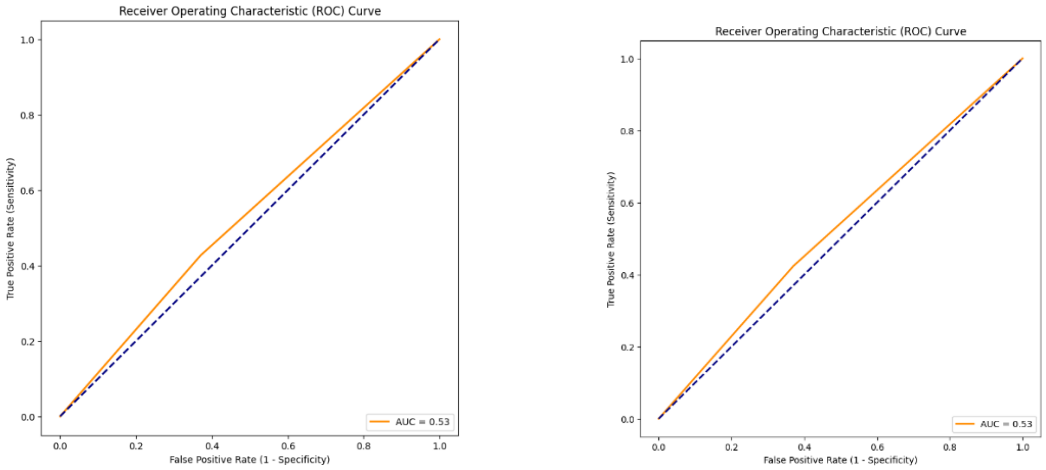


Fig ROC AUC using SMOTE by increasing minority class by 30% left side- Training performance, right-side Testing performance

ACTUAL/ PREDICTED	0	1
0	34890	20960
1	2788	2152

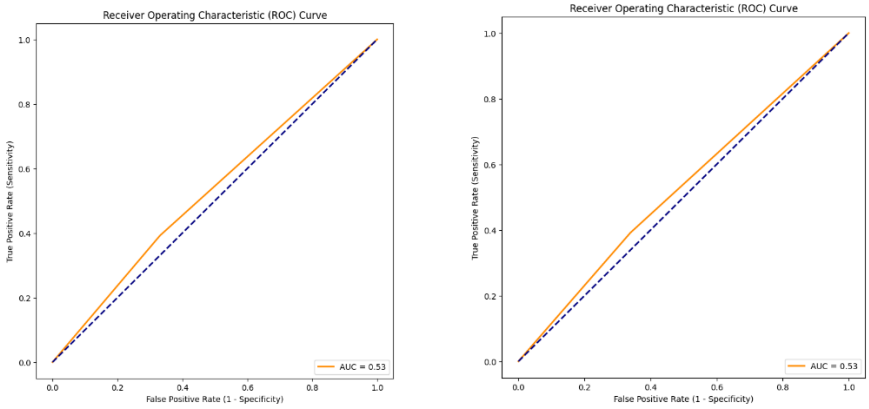


Fig : ROC AUC using SMOTEENN by increasing minority class by 30% left side- Training performance, right-side Testing performance

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.

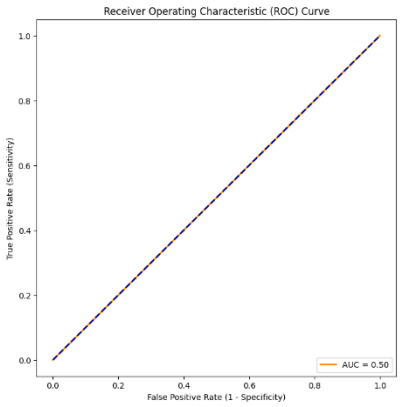
7.2 MODEL - NAIVE BAYES

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

	precision	recall	f1-score	support
0	0.92	1.00	0.96	55850
1	0.00	0.00	0.00	4940
accuracy			0.92	60790
macro avg	0.46	0.50	0.48	60790
weighted avg	0.84	0.92	0.88	60790



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.

7.3 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.

2. 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.

3. 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.

4. 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.

5. Independence of Observations:

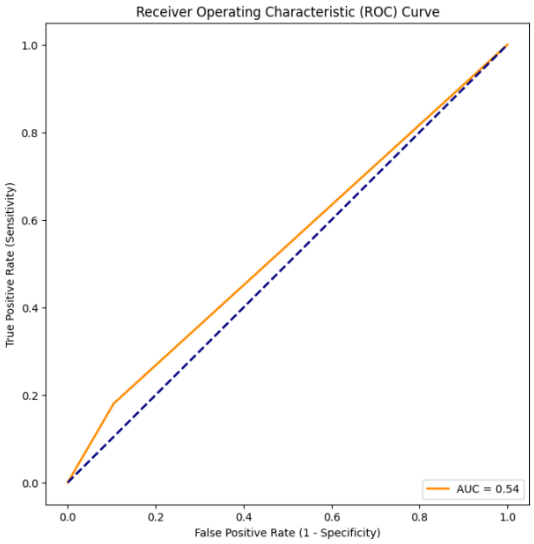
The observations (rows) in your dataset are assumed to be independent. This means that the value of one observation should not be influenced by the values of other observations.

6. 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.

7. 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.



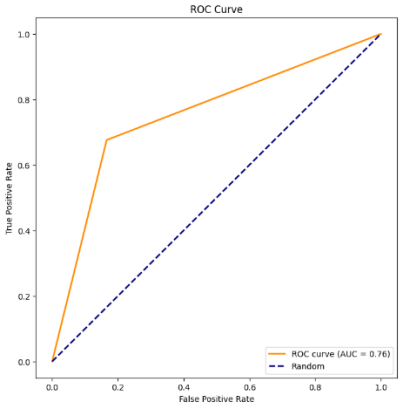

```
True Positive Rate (TPR): 0.18036437246963563

True Negative Rate (TNR): 0.8955237242614145
Precision: 0.13247100802854594
Negative Predictive Value (NPV): 0.9251072802604321
F1 Score: 0.23175569186261483
Accuracy: 0.8374074683336075
False Positive Rate (FPR): 0.10447627573858549
False Negative Rate (FNR): 0.8196356275303643
```

The model

- good for predicting non-risk users
- bad for predicting risk user
- ROC AUC of 0.53 suggests that the model is not performing well in terms of discriminating between the positive and negative classes. It's important to consider other evaluation metrics and possibly explore ways to improve the model's performance, such as tuning hyperparameters, feature engineering, or using a different algorithm.

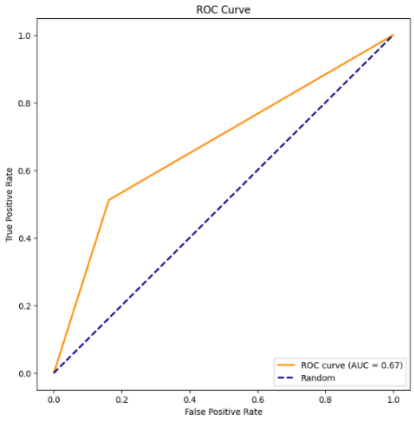
8. RESULTS OF FINAL MODEL

FOR LOGISTIC REGRESSION



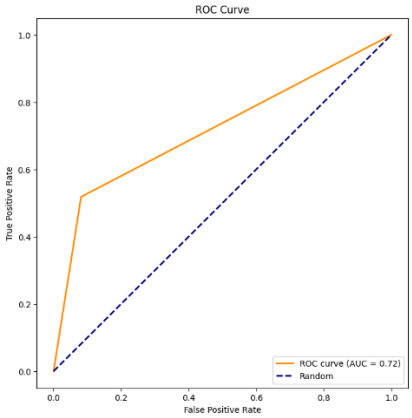

```
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.95      0.89     56698
           1       0.68      0.36      0.47     16731

    accuracy                           0.81     73429
   macro avg       0.76      0.65      0.68     73429
weighted avg       0.80      0.81      0.79     73429
```

FOR DECISION TREE

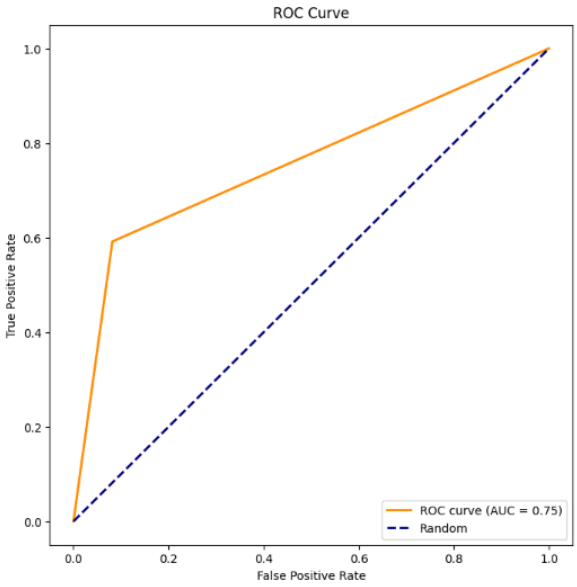


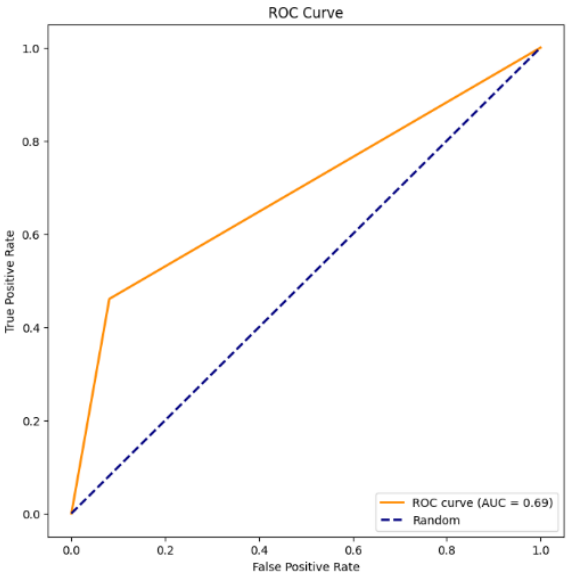

```
Accuracy :  77.66686186656499

              precision    recall  f1-score   support

           0       0.84      0.88      0.86     56698
           1       0.51      0.42      0.46     16731

    accuracy                           0.78     73429
   macro avg       0.67      0.65      0.66     73429
weighted avg       0.76      0.78      0.77     73429
```

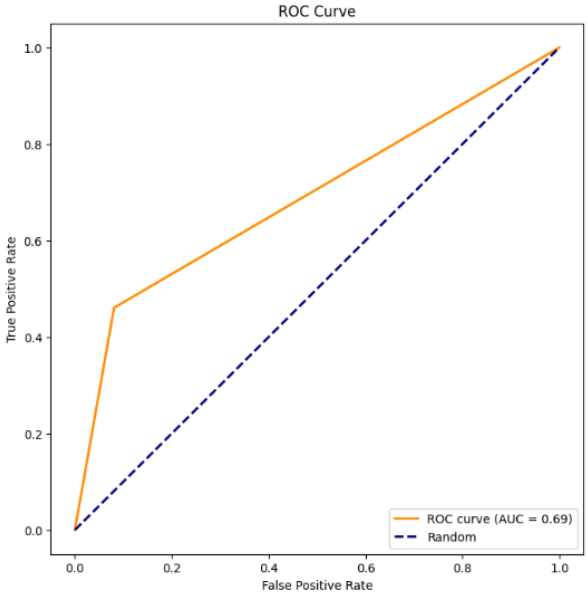
FOR RANDOM FOREST CLASSIFIER




```
Accuracy: 0.9173407112051428

Classification Report:
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     56359
           1       0.52      0.02      0.04      5086

    accuracy                           0.92     61445
   macro avg       0.72      0.51      0.50     61445
weighted avg       0.89      0.92      0.88     61445
```







These are the outputs of all boosting techniques left top – Gradient Boost, AdaBoost and XGBOOST

9. CONCLUSION

In this analysis, we aimed to evaluate credit worthiness using various classification models. The dataset consisted of [describe key characteristics of the dataset]. Our objective was to assess the predictive performance of different models and provide insights for informed decision-making in the context of credit evaluation.

We employed several classification models, including Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines. The models were evaluated using key performance metrics such as accuracy, precision, recall, F1-score, and ROC AUC.

10.FUTURE WORK: IMPLEMENTATION AS A LIVE APPLICATION

One promising avenue for future work involves transforming the credit worthiness prediction models into a live application, enhancing accessibility and usability for stakeholders. Here are key considerations and steps for this initiative:

User Interface Design:

Collaborate with user experience (UX) designers to create an intuitive and user-friendly interface. Ensure that the application is accessible to users with varying levels of technical expertise.
Integration with Existing Systems:

Explore integration possibilities with existing banking or financial systems. Seamless interaction with these systems can enhance the application's utility in real-world scenarios.
Scalability and Performance:

Optimize the application for scalability, ensuring it can handle a growing user base and an increasing volume of credit evaluation requests. Performance testing should be conducted to identify and address any bottlenecks.
Security Measures:

Implement robust security measures to protect sensitive financial data. Utilize encryption, secure data storage practices, and authentication mechanisms to ensure the confidentiality and integrity of user information.
Real-time Decision-Making:

Enhance the application to provide real-time credit worthiness decisions. This could involve optimizing model inference speed and reducing latency in response times.
Automation and Workflow Integration:

Explore opportunities for automating decision-making workflows within the application. This might include notifications to relevant stakeholders, integration with approval processes, and documentation generation.
User Feedback Mechanism:

Implement a feedback mechanism within the application to collect user insights and continuously improve model performance. Regularly update the models based on user feedback and evolving business requirements.
Compliance and Regulatory Considerations:

Ensure that the live application adheres to regulatory requirements and complies with industry standards for credit evaluation. Regularly update the application to accommodate changes in regulations.
Training and Support:

Develop training materials and provide support to end-users, including credit analysts and decision-makers. A well-supported user base is essential for the successful adoption of the application.
Monitoring and Maintenance:

Implement monitoring tools to track the application's performance, model drift, and user interactions. Establish a maintenance plan to address any issues promptly and keep the application up-to-date.
Continuous Model Improvement:

Establish a pipeline for continuous model improvement. Periodically retrain the models using updated data and explore advanced modeling techniques to enhance predictive accuracy.