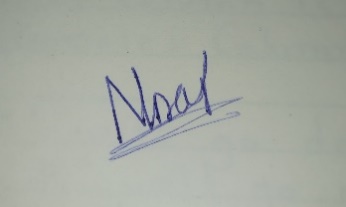
**CAPSTONE PROJECT FINAL REPORT**

PROJECT TITLE: **Home Credit Risk Analysis**

|  |  |
| --- | --- |
| Batch details | **PGP-DSE FT-PUNE-Aug-2023** |
| Team members | **Ziya Ahmad**  **Niraj Chaudhari**  **Ashutosh Chavan** |
| Domain of Project | **Finance and Risk Analytics** |
| Proposed project title | **Home Credit Risk Analysis** |
| Group Number | **2** |
| Team Leader | **Niraj Chaudhari** |
| Mentor Name | **Mr. Subramanian P V** |

Date: 06-03-2024

**** 

Signature of Mentor Signature of Team Leader

**Table of Contents**

[**1)** **Industry Review** 4](#_Toc158063464)

[**1.1.** **Current Practices in the Financial Industry** 4](#_Toc158063465)

[**1.2.** **Challenges in Loan Default Prediction** 4](#_Toc158063466)

[**1.3.** **Use of Data Science and Machine Learning** 4](#_Toc158063467)

[**1.4.** **Relevant Research and Literature** 4](#_Toc158063468)

[**1.5.** **Conclusion** 5](#_Toc158063469)

[**2)** **Dataset and Domain** 5](#_Toc158063470)

[**2.1.** **Data Dictionary** 5](#_Toc158063472)

[**2.2.** **Pre-processing Data Analysis** 6](#_Toc158063473)

[**2.3.** **Data pre-processing** 7](#_Toc158063474)

[**2.3.1.** **Data type conversion** 7](#_Toc158063480)

[**2.3.2.** **Check and treat for missing values** 8](#_Toc158063481)

[**2.3.3.** **Prepare a comprehensive dataset with the required fields** 14](#_Toc158063482)

[**2.3.4.** **Checking and treatment of null values in the Comprehensive File, newly created** 15](#_Toc158063483)

[**2.3.5.** **Check and remove all duplicate rows** 15](#_Toc158063484)

[**2.4.** **Alternate sources of data that can supplement the core dataset (at least 2-3 columns)** 16](#_Toc158063485)

[**2.5.** **Problem Justification** 16](#_Toc158063486)

[**2.5.1.** **Problem Statement** 16](#_Toc158063489)

[**2.5.2.** **Project Objective** 16](#_Toc158063490)

[**3)** **Data Exploration (EDA)** 17](#_Toc158063491)

[**3.1.** **Relationship with the target variable** 17](#_Toc158063493)

[**3.2.** **Checking for multi-collinearity** 21](#_Toc158063494)

[**3.3.** **Checking for distribution of variables** 22](#_Toc158063495)

[**3.4.** **Checking for presence of outliers and its treatment** 27](#_Toc158063496)

[**3.5.** **Checking for statistical significance of variables** 30](#_Toc158063497)

[**3.5.1.** **Two-sample Statistical t-test:** 30](#_Toc158063504)

[**3.6.** **Checking for class imbalance and its treatment** 33](#_Toc158063505)

[**4)** **Feature Engineering** 33](#_Toc158063506)

[**4.1.** **Transformations required** 33](#_Toc158063508)

[**4.2.** **Scaling the data** 34](#_Toc158063509)

[**4.3.** **Feature selection** 35](#_Toc158063510)

[**4.4.** **Dimensionality Reduction** 36](#_Toc158063511)

[**5)** **Assumptions for base model (Logistic Regression)** 36](#_Toc158063512)

[**5.1.** **Logistic Regression assumptions** 36](#_Toc158063518)

[**6)** **Building & Evaluation of ML Model** 41](#_Toc158063519)

[**6.1.** **Performance Metrics for our base model** 41](#_Toc158063526)

[**6.2.** **Steps in building the Logistic Regression Model** 42](#_Toc158063527)

[**6.2.1.** **Split the Dataset for training and testing** 42](#_Toc158063536)

[**6.2.2.** **Measure of Model Performance using original data** 42](#_Toc158063537)

[**6.2.3.** **Measure of Model Performance after applying SMOTE on Training data** 45](#_Toc158063538)

[**6.2.4.** **Tasks planned for future** 47](#_Toc158063539)

**7) Building Multiple model**

**7.1 K-Nearest Neighbour**………………………………………………………………………………………………….47.

**7.1.1 Measure of Model Performance after applying SMOTE on Training data…………..47**

**7.2 Decision Tree Mode**l……………………………………………………………………………………………………50

**7.2.1 Measure of Model Performance on originaldata**…………………………………………………50

**7.2.2 Measure of Model Performance after applying SMOTE on Training data**…………..52

**7.3 XG BOOST Model**

**7.3.1 Measure of Model Performance on originaldata**……………………………………………….54

**7.3.2 Measure of Model Performance after applying SMOTE on Training data**………….57

**7.4 ADA BOOST Model**

**7.4.1 Measure of Model Performance after applying SMOTE on Training data**………….64

**7.5 Gradient Boosting Model**

**7.5.1 Measure of Model Performance after applying SMOTE on Training data**………… 66

**8 Model Comparisons**

**8.1 On Basis of recall** …………………………………………………………………………………..72

**8.2 Comparison on the basis of Weighted Avg(Recall)**…………………………………….73

**8..3 Final Model (XG BOOST)**:…………………………………………………………………………74

**9. Feature Importance**…………………………………………………………………………………………………..7**6**

**10. Project Conclusion**

[**10)** **Appendix** 47](#_Toc158063540)

[**8.1.** **Data Dictionary** 47](#_Toc158063548)

[**8.2.** **Missing values** 58](#_Toc158063549)

# **Industry Review**

Loans have always been an important part of people’s lives for quite some time now. Everyone has different reasons for borrowing a loan. It could be to buy a dream car or a home, to set up a business, or to buy some products. Even wealthy people prefer taking loans overspending their cash to get tax benefits and to keep the cash available for future unexpected and unconventional expenses.

Loans are also as important to Lenders as they are for Borrowers. Almost all Banking Organizations make most of their revenues from the interests generated through loans. However, the caveat here is that the lenders make a profit only if the loan gets repaid. The Lending Organizations are faced with the tough task of analysing the risk associated with each client. Therefore, it is important to identify the risky behaviours of clients and make educated decisions.

## **Current Practices in the Financial Industry**

* Traditional methods for loan approval involve assessing applicants' credit scores, income, debt-to-income ratios, and other financial indicators.
* Banks often use predefined rules and thresholds to determine eligibility for loans.
* The reliance on historical data and fixed criteria may lead to suboptimal predictions, especially in rapidly changing economic conditions.

## **Challenges in Loan Default Prediction**

* Financial institutions face challenges in accurately predicting loan defaults due to dynamic economic factors and evolving customer behaviours.
* Traditional models may struggle to adapt to changing conditions, leading to higher default rates during economic downturns.
* Data quality issues and limited features in traditional models may hinder accurate predictions.

## **Use of Data Science and Machine Learning**

* Data science and machine learning offer opportunities to enhance loan default prediction by leveraging advanced algorithms.
* Predictive models can analyse a broader range of features, including non-traditional data sources such as social media activity and transaction history.
* Machine learning models can continuously learn and adapt to evolving patterns, improving predictive accuracy over time.

## **Relevant Research and Literature**

* Research studies have explored the effectiveness of machine learning algorithms such as decision trees, random forests, and neural networks in predicting loan defaults.
* Feature engineering, model interpretability, and ensemble methods are key areas of focus in the literature.
* Some studies emphasize the importance of explainability and transparency in machine learning models for regulatory compliance.

## **Conclusion**

* The financial industry is undergoing a transformation in risk management practices, with an increasing emphasis on leveraging data science and machine learning.
* Challenges in traditional loan approval methods highlight the need for more adaptive and data-driven approaches.
* Our project aims to contribute to this evolving landscape by developing a predictive model for loan default using advanced machine learning techniques.

# **Dataset and Domain**



## **Data Dictionary**

There are six files related to the housing loan with common key, ID of loan connecting all of them to the main data file, application file. Data dictionary is available in the Appendix section.

1. **application\_train.csv**
2. This is the main table with TARGET.
3. Static data for all applications. One row represents one loan in our data sample.
4. **bureau.csv**
5. All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in our sample).
6. For every loan in our sample, there are as many rows as number of credits the client had in Credit Bureau before the application date.
7. **bureau\_balance.csv**
8. Monthly balances of previous credits in Credit Bureau.
9. This table has one row for each month of history of every previous credit reported to Credit Bureau – i.e the table has (#loans in sample \* # of relative previous credits \* # of months where we have some history observable for the previous credits) rows.
10. **POS\_CASH\_balance.csv**
11. Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
12. This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample \* # of relative previous credits \* # of months in which we have some history observable for the previous credits) rows.
13. **credit\_card\_balance.csv**
14. Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
15. This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample \* # of relative previous credit cards \* # of months where we have some history observable for the previous credit card) rows.
16. **previous\_application.csv**
17. All previous applications for Home Credit loans of clients who have loans in our sample.
18. There is one row for each previous application related to loans in our data sample.
19. **installments\_payments.csv**
20. Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.
21. There is a) one row for every payment that was made plus b) one row each for missed payment.
22. One row is equivalent to one payment of one installment OR one installment corresponding to one payment of one previous Home Credit credit related to loans in our sample.



**Fig,2,1.** Structure of data

## **Pre-processing Data Analysis**

We need to process the following files described above for our application:

1. application\_train.csv
2. bureau.csv
3. bureau\_balance.csv
4. credit\_card\_balance.csv
5. installments\_payments.csv
6. POS\_CASH\_balance.csv
7. previous\_application.csv

We have one more file, HomeCredit\_columns\_description.csv which describes the above files.

**Summary of the above Data Files:**

|  |  |  |  |
| --- | --- | --- | --- |
| # | **File Name** | **Number of rows** | **Number of columns** |
| 1 | application\_train.csv | 307511 | 122 |
| 2 | bureau.csv | 1716428 | 17 |
| 3 | bureau\_balance.csv | 27299925 | 3 |
| 4 | credit\_card\_balance.csv | 3840312 | 23 |
| 5 | installments\_payments.csv | 13605401 | 8 |
| 6 | POS\_CASH\_balance.csv | 10001358 | 8 |
| 7 | previous\_application.csv | 1670214 | 37 |

## **Data pre-processing**



### **Data type conversion**

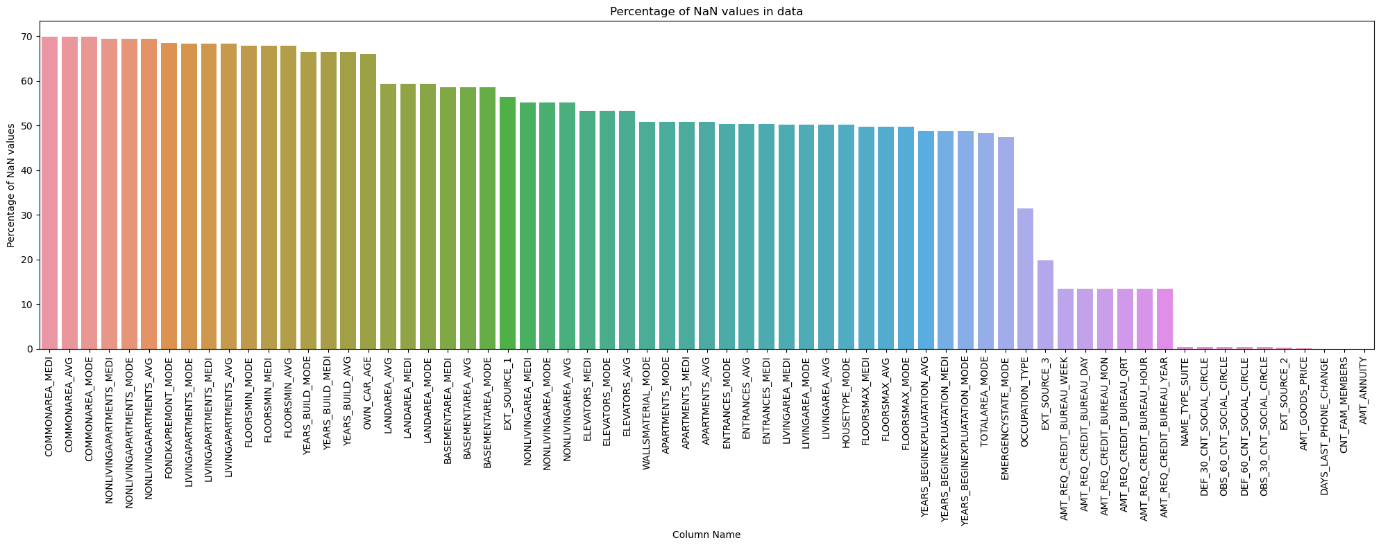
**File 1: application\_train.csv**

There are 307511 rows and 122 columns in this file. The data type of the following 35 variables is specified as numeric and we need to convert them to categorical variables:

1. TARGET
2. FLAG\_MOBIL
3. FLAG\_EMP\_PHONE
4. FLAG\_WORK\_PHONE
5. FLAG\_CONT\_MOBILE
6. FLAG\_PHONE
7. FLAG\_EMAIL
8. REGION\_RATING\_CLIENT
9. REGION\_RATING\_CLIENT\_W\_CITY
10. REG\_REGION\_NOT\_LIVE\_REGION
11. REG\_REGION\_NOT\_WORK\_REGION
12. LIVE\_REGION\_NOT\_WORK\_REGION
13. REG\_CITY\_NOT\_LIVE\_CITY
14. REG\_CITY\_NOT\_WORK\_CITY
15. LIVE\_CITY\_NOT\_WORK\_CITY
16. FLAG\_DOCUMENT\_2
17. FLAG\_DOCUMENT\_3
18. FLAG\_DOCUMENT\_4
19. FLAG\_DOCUMENT\_5
20. FLAG\_DOCUMENT\_6
21. FLAG\_DOCUMENT\_7
22. FLAG\_DOCUMENT\_8
23. FLAG\_DOCUMENT\_9
24. FLAG\_DOCUMENT\_10
25. FLAG\_DOCUMENT\_11
26. FLAG\_DOCUMENT\_12
27. FLAG\_DOCUMENT\_13
28. FLAG\_DOCUMENT\_14
29. FLAG\_DOCUMENT\_15
30. FLAG\_DOCUMENT\_16
31. FLAG\_DOCUMENT\_17
32. FLAG\_DOCUMENT\_18
33. FLAG\_DOCUMENT\_19
34. FLAG\_DOCUMENT\_20
35. FLAG\_DOCUMENT\_21

### **Check and treat for missing values**

**File 1: application\_train.csv**

There are 307511 rows and 122 columns in this file. There are 67 features with missing values in this file as given in the Appendix.

**Fig.2.2.** Missing values % in the file, application\_train.csv

1. **If the percentage of missing values in a column exceeds the threshold value of 60%, then the column is dropped.**

There are 17 columns having missing values exceeding the threshold value. They are dropped and given below:

| **#** | **Column** |
| --- | --- |
| 1 | COMMONAREA\_AVG |
| 2 | COMMONAREA\_MEDI |
| 3 | COMMONAREA\_MODE |
| 4 | NONLIVINGAPARTMENTS\_AVG |
| 5 | NONLIVINGAPARTMENTS\_MEDI |
| 6 | NONLIVINGAPARTMENTS\_MODE |
| 7 | FONDKAPREMONT\_MODE |
| 8 | LIVINGAPARTMENTS\_AVG |
| 9 | LIVINGAPARTMENTS\_MEDI |
| 10 | LIVINGAPARTMENTS\_MODE |
| 11 | FLOORSMIN\_AVG |
| 12 | FLOORSMIN\_MEDI |
| 13 | FLOORSMIN\_MODE |
| 14 | YEARS\_BUILD\_AVG |
| 15 | YEARS\_BUILD\_MEDI |
| 16 | YEARS\_BUILD\_MODE |
| 17 | OWN\_CAR\_AGE |

1. **Falls below the threshold value, missing values are imputed with median or mode depending on the data type being numerical or categorical.**

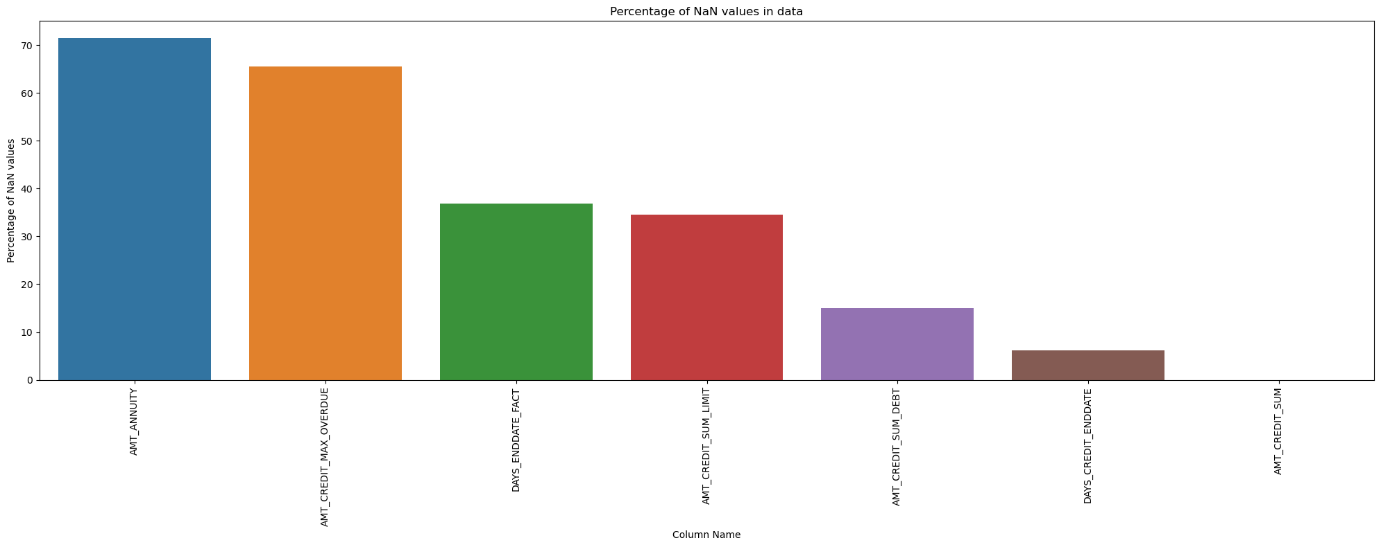
We have imputed with mode for the following 5 categorical variables:

1. WALLSMATERIAL\_MODE
2. HOUSETYPE\_MODE
3. EMERGENCYSTATE\_MODE
4. OCCUPATION\_TYPE
5. NAME\_TYPE\_SUITE

For the rest of 45 numerical variables, we shall use median for imputation

**File 2: File bureau.csv**

There are 1716428 rows and 17 columns in this file. There are 7 features with missing values as given in the Appendix.



**Fig.2.3.** Missing values % in the file, bureau.csv

1. **If the percentage of missing values in a column exceeds the threshold value of 60%, then the column is dropped.**

There are 2 columns having missing values exceeding the threshold value. They are dropped and given below:

|  |
| --- |
| AMT\_ANNUITY |
| AMT\_CREDIT\_MAX\_OVERDUE |

**b) Falls below the threshold value, missing values are imputed with median or mode depending on the data type being numerical or categorical.**

We have imputed with median for the following 5 numerical variables:

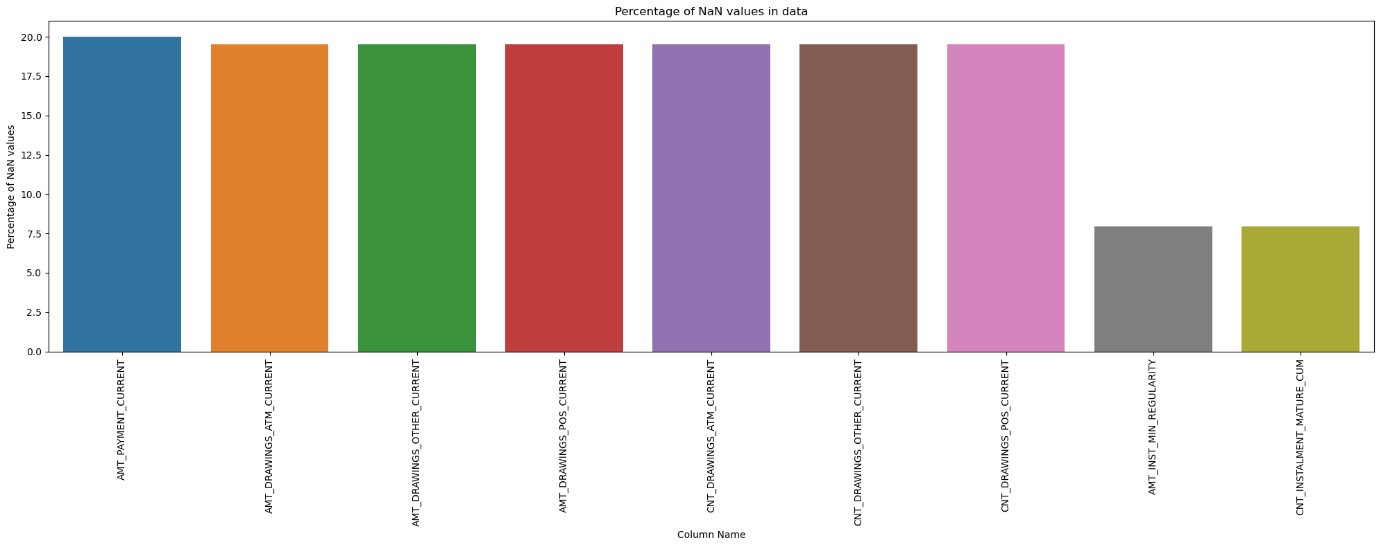
1. DAYS\_ENDDATE\_FACT
2. AMT\_CREDIT\_SUM\_LIMIT
3. AMT\_CREDIT\_SUM\_DEBT
4. DAYS\_CREDIT\_ENDDATE
5. AMT\_CREDIT\_SUM

**File 3: bureau\_balance.csv**

Thereare27299925rowsand3 columnsinthisfile**.** There are no missing values in this file

**File 4: credit\_card\_balance.csv**

There are 3840312 rows and 23 columns in this file. There are 9 features with missing values in this file as given in the Appendix.



**Fig.2.4.** Missing values % in the file, credit\_card\_balance.csv

1. **If the percentage of missing values in a column exceeds the threshold value of 60%, then the column is dropped.**

We observe that there are no columns having missing values above the threshold value.

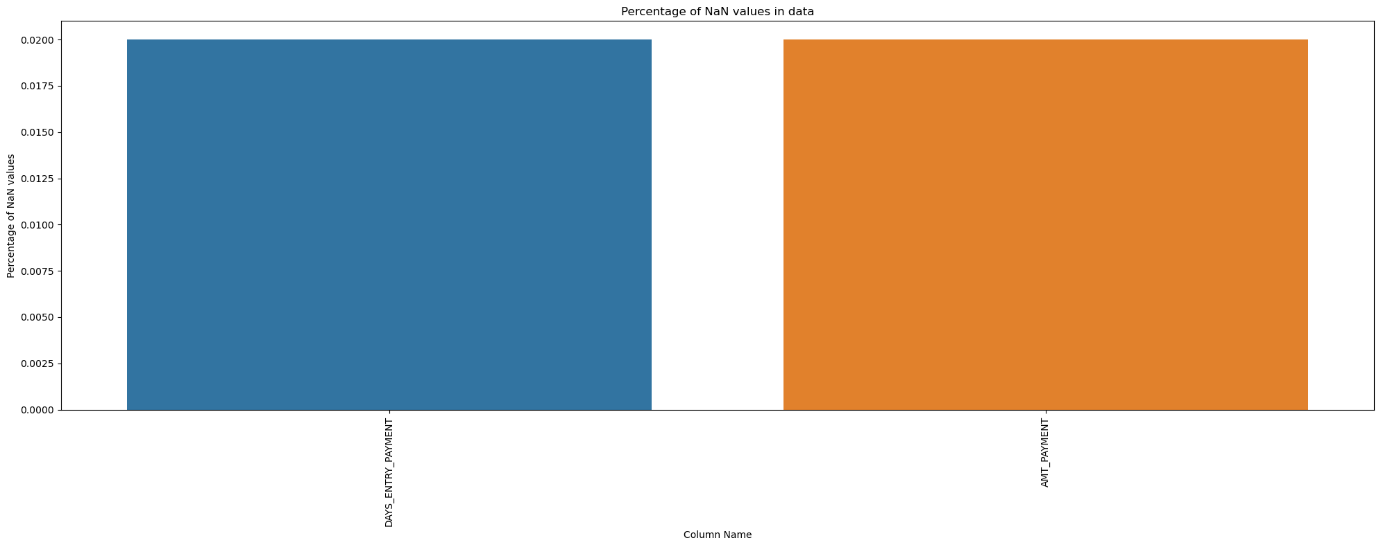
1. **Falls below the threshold value, missing values are imputed with median or mode depending on the data type being numerical or categorical.**

We have imputed with median for the following 9 numerical variables:

1. AMT\_PAYMENT\_CURRENT
2. AMT\_DRAWINGS\_ATM\_CURRENT
3. AMT\_DRAWINGS\_OTHER\_CURRENT
4. AMT\_DRAWINGS\_POS\_CURRENT
5. CNT\_DRAWINGS\_ATM\_CURRENT
6. CNT\_DRAWINGS\_OTHER\_CURRENT
7. CNT\_DRAWINGS\_POS\_CURRENT
8. AMT\_INST\_MIN\_REGULARITY
9. CNT\_INSTALMENT\_MATURE\_CUM

**File 5: installments\_payments.csv**

There are 13605401 rows and 8 columns in this file. There are 2 features with missing values in this file as given in the Appendix.



**Fig.2.5.** Missing values % in the file, installments\_payments.csv

1. **If the percentage of missing values in a column Exceeds the threshold value of 60%, then the column is dropped.**

We observe that there are no columns having missing values above the threshold value.

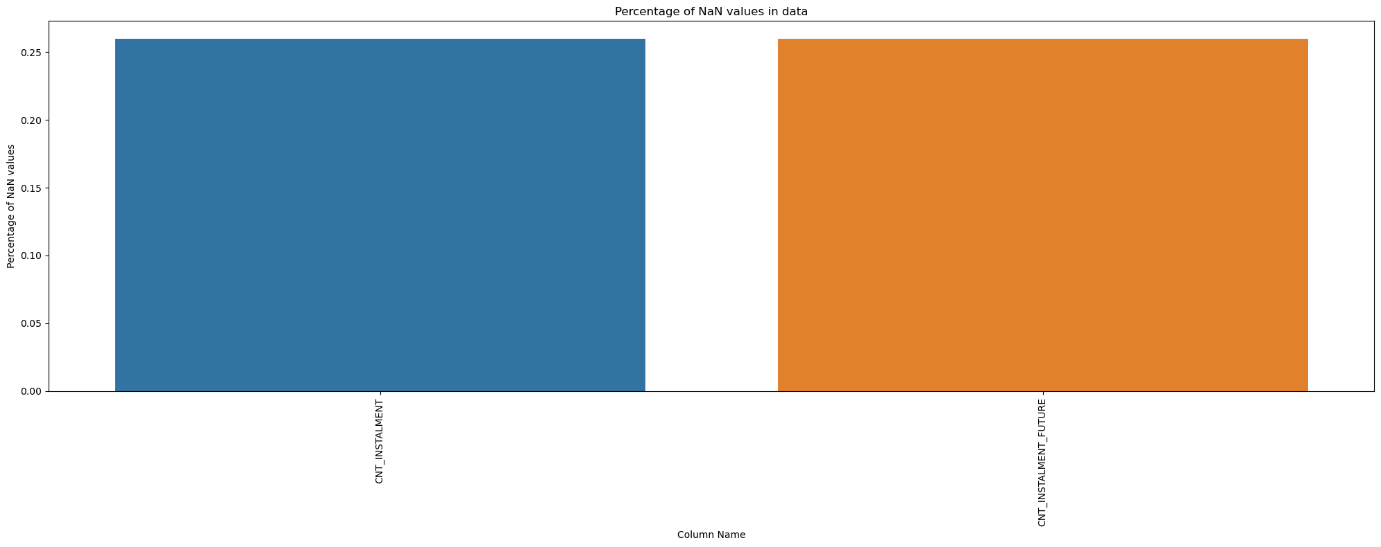
1. **Falls below the threshold value, missing values are imputed with median or mode depending on the data type being numerical or categorical.**

We have imputed with median for the following 2 numerical variables:

1. AMT\_PAYMENT
2. DAYS\_ENTRY\_PAYMENT

**File 6: POS\_CASH\_balance.csv**

There are 10001358 rows and 8 columns in this file. There are two features with missing values in this file as given in the Appendix.



**Fig.2.6.** Missing values % in the file POS\_CASH\_balance.csv

1. **If the percentage of missing values in a column exceeds the threshold value of 60%, then the column is dropped.**

We observe that there are no columns having missing values above the threshold value.

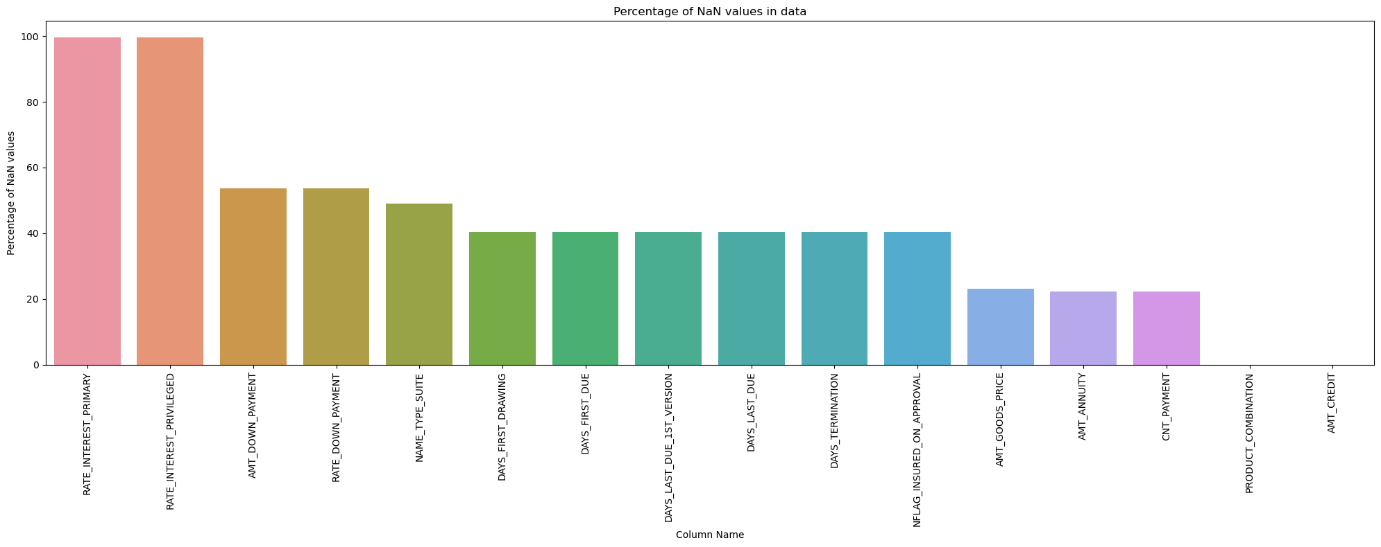
1. **Falls below the threshold value, missing values are imputed with median or mode depending on the data type being numerical or categorical.**

We have imputed with median for the following 2 numerical variables:

1. CNT\_INSTALMENT
2. CNT\_INSTALMENT\_FUTURE

**File 7: previous\_application.csv**

There are 1670214 rows and 37 columns in this file. There are 16 features with missing values in this file as given in the Appendix.



**Fig.2.7.** Missing values % in the file, previous\_application.csv

1. **If the percentage of missing values in a column exceeds the threshold value of 60%, then the column is dropped.**

We observe that there are two columns having missing values above the threshold value. They are given below:

1. RATE\_INTEREST\_PRIMARY

2. RATE\_INTEREST\_PRIVILEGED

1. **Falls below the threshold value, missing values are imputed with median or mode depending on the data type being numerical or categorical.**

We have imputed with median for the following 12 numerical variables:

1. AMT\_DOWN\_PAYMENT
2. RATE\_DOWN\_PAYMENT
3. DAYS\_FIRST\_DRAWING
4. DAYS\_FIRST\_DUE
5. DAYS\_LAST\_DUE
6. DAYS\_LAST\_DUE\_1ST\_VERSION
7. DAYS\_TERMINATION
8. NFLAG\_INSURED\_ON\_APPROVAL
9. AMT\_GOODS\_PRICE
10. AMT\_ANNUITY
11. CNT\_PAYMENT
12. AMT\_CREDIT

We have imputed with mode for the following 2 categorical variables:

1. NAME\_TYPE\_SUITE
2. PRODUCT\_COMBINATION

**After treating null values, in the features, we have the following summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **File name** | **Number of rows** | **Number of columns** |
| 1 | application\_train.csv | 307511 | 122 |
| clean\_application\_train.csv | 105 |
| 2 | bureau.csv | 1716428 | 17 |
| clean\_bureau.csv | 15 |
| 3 | bureau\_balance.csv | 27299925 | 3 |
| clean\_bureau\_balance.csv | 3 |
| 4 | credit\_card\_balance.csv | 3840312 | 23 |
| clean\_credit\_card\_balance.csv | 23 |
| 5 | installments\_payments.csv | 13605401 | 8 |
| clean\_installments\_payments.csv | 8 |
| 6 | POS\_CASH\_balance.csv | 10001358 | 8 |
| clean\_POS\_CASH\_balance.csv | 8 |
| 7 | previous\_application.csv | 1670214 | 37 |
| clean\_previous\_application.csv | 35 |

### **Prepare a comprehensive dataset with the required fields**

The following fields were derived from other related files, application\_train.file using the key field, SK\_ID\_CURR:

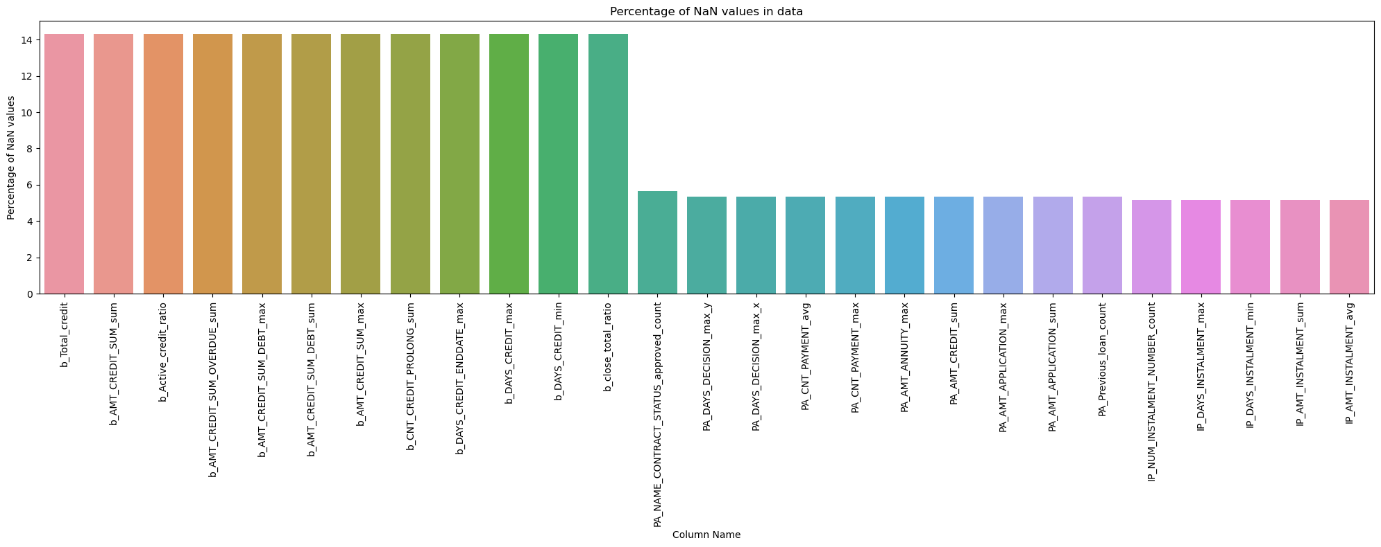
| # | Derived field name | Description |
| --- | --- | --- |
| 1 | b\_Total\_credit | Total number of credits taken by applicant with other banks. |
| 2 | b\_Active\_credit\_ratio | Ratio of number of credits still active from other banks to the overall credits taken by applicant |
| 3 | b\_close\_total\_ratio | Ratio of number of credits still active from other banks to the closed credits taken by applicant |
| 4 | b\_DAYS\_CREDIT\_min | How many days before current application did client apply for last Credit |
| 5 | b\_DAYS\_CREDIT\_max | How many days before current application did client apply for first Credit |
| 6 | b\_DAYS\_CREDIT\_ENDDATE\_max | Out of all the client’s previous credits, we are taking the one which has maximum time duration left till fulfillment of the credit, it’s a column with no of days |
| 7 | b\_CNT\_CREDIT\_PROLONG\_sum | How many times was the Credit Bureau credit prolonged(total) |
| 8 | b\_AMT\_CREDIT\_SUM\_sum | Total sum of all credits taken by applicant |
| 9 | b\_AMT\_CREDIT\_SUM\_max | Maximum credit taken by applicant of all credit’s applicant |
| 10 | b\_AMT\_CREDIT\_SUM\_DEBT\_sum | Total current debt on credit bureau credit |
| 11 | b\_AMT\_CREDIT\_SUM\_DEBT\_max | Maximum current debt on credit bureau credit |
| 12 | b\_AMT\_CREDIT\_SUM\_OVERDUE\_sum | Total Current amount overdue on Credit Bureau credit |
| 13 | PA\_Previous\_loan\_count | No of previous loan from home credit |
| 14 | PA\_AMT\_ANNUITY\_max | Annuity=Amount of installment for each month. We are taking maximum installment |
| 15 | PA\_AMT\_APPLICATION\_sum | Sum of all the credit previously taken by the client, this is applied credit value, actual disburse amount could differ slightly |
| 16 | PA\_AMT\_APPLICATION\_max | Maximum amount of the credit previously taken by the client |
| 17 | PA\_AMT\_CREDIT\_sum | Actual disbursed credit amount |
| 18 | PA\_NAME\_CONTRACT\_STATUS\_approved\_count | How many previous applications for loan were accepted |
| 19 | PA\_CNT\_PAYMENT\_max | What was the maximum term out of all the previous credits.  CNT\_PAYMENT tells us about the duration of the credit repayment |
| 20 | PA\_CNT\_PAYMENT\_avg | What was the average term length for the previous credits |
| 21 | PA\_DAYS\_DECISION\_max | No of days since the decision about first application was taken |
| 22 | PA\_DAYS\_DECISION\_max | No of days since the decision of about recent application was taken |
| 23 | IP\_NUM\_INSTALMENT\_NUMBER\_count | No of total installments for all the loan |
| 24 | IP\_DAYS\_INSTALMENT\_max | No of days since the last installments, it could be of anyone of the multiple loans taken by client |
| 25 | IP\_DAYS\_INSTALMENT\_min | No of days since the First installments, it could be of anyone of the multiple loans taken by client |
| 26 | IP\_AMT\_INSTALMENT\_sum | Overall payment done by client for all the loans through installments |
| 27 | IP\_AMT\_INSTALMENT\_avg | Average installments by client for all the loans through installments |
| 28 | AT\_CREDIT\_INCOME\_PERCENT | The percentage of the credit amount relative to a client's income |
| 29 | AT\_ANNUITY\_INCOME\_PERCENT | The percentage of the loan annuity relative to a client's income |
| 30 | AT\_CREDIT\_TERM | The length of the payment in months (since the annuity is the monthly amount due) |
| 31 | AT\_DAYS\_EMPLOYED\_PERCENT | The percentage of the days employed relative to the client's age |

Now, we have 307511 rows and 136 columns in our comprehensive dataset.

### **Checking and treatment of null values in the Comprehensive File, newly created**

Again, check for null values because the applicant may not have availed any previous loan or defaulted on any loan.

There are 27 features with missing values in this file as given in the Appendix.



**Fig.2.8.** Missing values % in the comprehensive data

**Treatment of Null values**

These Null values have appeared because for the given applicant don’t have record entries in other data files. So, we replaced the missing values with zeros as it is the most appropriate method.

### **Check and remove all duplicate rows**

There are no duplicates in the comprehensive dataset.

Our dataset has 307511 rows. We had 134 independent variables. Using a popular feature selection algorithm, Recursive Feature Elimination, or RFE for short, we selected the 15 independent variables from this dataset that are more or most relevant in predicting the target variable. Details of this selection is explained later. Apart from the target variable, TARGET, the 15 independent variables selected are given below:

| # | Name of the variable | Source |
| --- | --- | --- |
| 1 | AMT\_ANNUITY | Original data |
| 2 | DAYS\_BIRTH | Original data |
| 3 | DAYS\_REGISTRATION | Original data |
| 4 | DAYS\_ID\_PUBLISH | Original data |
| 5 | EXT\_SOURCE\_2 | Original data |
| 6 | EXT\_SOURCE\_3 | Original data |
| 7 | b\_DAYS\_CREDIT\_max | Derived data |
| 8 | PA\_AMT\_CREDIT\_sum | Derived data |
| 9 | PA\_DAYS\_DECISION\_max\_y | Derived data |
| 10 | IP\_DAYS\_INSTALMENT\_min | Derived data |
| 11 | IP\_AMT\_INSTALMENT\_sum | Derived data |
| 12 | IP\_AMT\_INSTALMENT\_avg | Derived data |
| 13 | AT\_ANNUITY\_INCOME\_PERCENT | Derived data |
| 14 | AT\_CREDIT\_TERM | Derived data |
| 15 | AT\_DAYS\_EMPLOYED\_PERCENT | Derived data |

## **Alternate sources of data that can supplement the core dataset (at least 2-3 columns)**

In a related study Oladeebo (2008), examined socio-economic factors influencing loan repayment among small scale farmers in Ogbomoso agricultural zone of Oyo State of Nigeria. Hence, socio-demographic factors such as education, employment, community safety and social support may be considered.

## **Problem Justification**



### **Problem Statement**

There are lots of people who do not particularly have a prior credit history, for example students, small businessmen, etc. who need credits, be it for studies, or for setting up some sort of businesses. Without adequate credit history, the lending organizations find it difficult to lend credits to such people, as these loans could be associated with high risks. In these kinds of situations, some lending organizations even tend to exploit the borrowers by asking for too high of an interest rate.

There is another subset of people, who do have prior credit history, which could be with the same organization or some other organizations. However, going through that historical data could be very time consuming and redundant. This would scale up even further as the number of applicants increases.

For such cases, if there could be a way through which the lending organization could predict or estimate the borrower’s repayment capability, the process could be streamlined and be made effective for both the lender and the borrower. It could save resources both in terms of humans and time.

### **Project Objective**

* We can identify that it is a Supervised Learning Classification problem, which contains the training data points along with their Class Labels. Here, the Class Labels represent whether a given applicant is a Defaulter or not. Thus, for a given application of a client, using the given features, we need to predict the Class Label associated with that client.
* We also realize that it is a Binary Classification problem, that is it contains just 2 classes, viz. Positive (1) and Negative (0).
* The main objective is to identify the potential Defaulters based on the given data about the applicants.
* The probability of classification is essential because we want to be very sure when we classify someone as a Non-Defaulter, as the cost of making a mistake can be very high to the company.
* The dataset provided is an imbalanced dataset. Thus, we would need to address this imbalance wherever required, as some ML algorithms are sensitive to data imbalance.

# **Data Exploration (EDA)**



## **Relationship with the target variable**

##### We check the relationship with each of the 15 independent variables with the target variable.

|  |  |
| --- | --- |
| **Fig.3.1.**Distribution of the variable, AMT\_ANNUITY and relationship with the target variable | **Fig.3.2**.Distribution of the variable, DAYS\_BIRTH and relationship with the target variable |
| **Fig.3.3.**Distribution of the variable, DAYS\_REGISTRATION and relationship with the target variable | **Fig.3.4.**Distribution of the variable, DAYS\_ID\_PUBLISH and relationship with the target variable |

|  |  |
| --- | --- |
| **Fig.3.5.**Distribution of the variable, EXT\_SOURCE\_2 and relationship with the target variable | **Fig.3.6.**Distribution of the variable, EXT\_SOURCE\_3 and relationship with the target variable |
| **Fig.3.7**.Distribution of the variable, b\_DAYS\_CREDIT\_max and relationship with the target variable | **Fig.3.8.**Distribution of the variable, PA\_AMT\_CREDIT\_sum and relationship with the target variable |
| **Fig.3.9.**Distribution of the variable, PA\_DAYS\_DECISION\_max\_y and relationship with the target variable | **Fig.3.10**.Distribution of the variable, IP\_DAYS\_INSTALMENT\_min and relationship with the target variable |
| **Fig.3.11.**Distribution of the variable, IP\_AMT\_INSTALMENT\_sum and relationship with the target variable | **Fig.3.12.**Distribution of the variable, IP\_AMT\_INSTALMENT\_avg and relationship with the target variable |
| **Fig.3.13.**Distribution of the variable, AT\_ANNUITY\_INCOME\_PERCENT and relationship with the target variable | **Fig.3.14.**Distribution of the variable, AT\_CREDIT\_TERM and relationship with the target variable |
| **Fig.3.15.**Distribution of the variable, AT\_DAYS\_EMPLOYED\_PERCENT and relationship with the target variable |  |

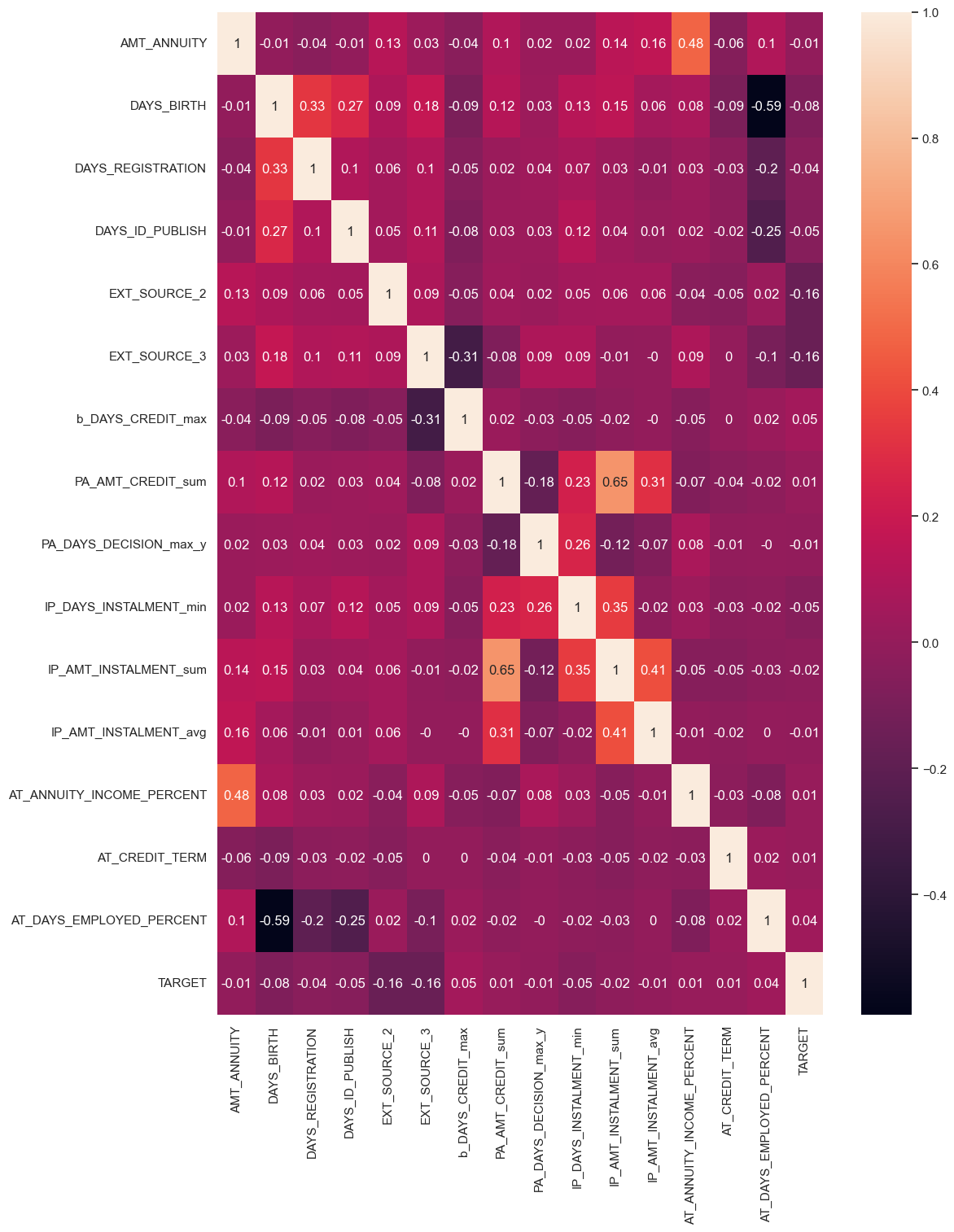
**Observations**

* 1. The AMT\_ANNUITY for Defaulter is almost same as that of Non-Defaulters. There are many outliers for both Defaulters and Non-Defaulters.
  2. The DAYS\_BIRTH now converted to years since birth indicating the Age of Defaulters is usually lesser than the Non-Defaulters. All the quantiles of ages of Defaulters is lesser than Non-Defaulters. The 75th percentile value of Non-Defaulters is around 54 years while for Defaulters it is near to 49 years. There are no outliers for both Defaulters and Non-Defaulters. **These observations imply that the Defaulters are usually younger than Non-Defaulters.**
  3. The DAYS\_REGISTRATION for Defaulter is almost same as that of Non-Defaulters. There are many outliers for both Defaulters and Non-Defaulters.
  4. The DAYS\_ID\_PUBLISH indicates the number of days before the application did client change the identity document with which he applied for the loan. DAYS\_ID\_PUBLISH for Defaulter is slightly less than Non-Defaulters. There are no outliers for both Defaulters and Non-Defaulters.
  5. The EXT\_SOURCE\_2 for defaulter is less than the Non-Defaulters. It is interesting to note that the median value for defaulters is almost equal to or lower than 25th percentile values of Non-Defaulters. There are no outliers for both Defaulters and Non-Defaulters.
  6. The EXT\_SOURCE\_3 for defaulter is less than the Non-Defaulters. It is interesting to note that the median value for defaulters is almost equal to or lower than 25th percentile values of Non-Defaulters. There are outliers for Non-Defaulters.
  7. The b\_DAYS\_CREDIT\_max for defaulter is more than the Non-Defaulters. There are no outliers for both Defaulters and Non-Defaulters.
  8. The PA\_AMT\_CREDIT\_sum for defaulter is almost same as that of Non-Defaulters. There are no outliers for both Defaulters and Non-Defaulters.
  9. The PA\_DAYS\_DECISION\_max\_y for defaulter is visibly more than that of Non-Defaulters. It is interesting to note that the Whisker end of Defaulter is almost sane as that of Q3 of Non-Defaulter. There are outliers for both Defaulters and Non-Defaulters.
  10. The IP\_DAYS\_INSTALMENT\_min for defaulter is less than the Non-Defaulters. There are outliers for both Defaulters and Non-Defaulters.
  11. The IP\_AMT\_INSTALMENT\_sum for defaulter is almost same as that of Non-Defaulters. There are outliers for both Defaulters and Non-Defaulters.
  12. The IP\_AMT\_INSTALMENT\_avg for defaulter is almost same as that of Non-Defaulters. There are outliers for both Defaulters and Non-Defaulters.
  13. The AT\_ANNUITY\_INCOME\_PERCENT for defaulter is almost same as that of Non-Defaulters. There are outliers for both Defaulters and Non-Defaulters.
  14. The AT\_CREDIT\_TERM for defaulter is almost same as that of Non-Defaulters. There are outliers for both Defaulters and Non-Defaulters.
  15. The AT\_DAYS\_EMPLOYED\_PERCENT for defaulter is almost same as that of Non-Defaulters. There are outliers for both Defaulters and Non-Defaulters.

## **Checking for multi-collinearity**

Multicollinearity is a statistical phenomenon that occurs when two or more independent variables in a regression model are highly correlated with each other. In other words, multicollinearity indicates a strong linear relationship among the predictor variables. This can create challenges in the regression analysis because it becomes difficult to determine the individual effects of each independent variable on the dependent variable accurately.

We shall check for the correlation among numeric independent variables. A correlation matrix is a common tool used to compare the coefficients of correlation between different features (or attributes) in a dataset. It allows us to visualize how much (or how little) correlation exists between different variables.



**Fig.3.16.** Correlation heatmap

**Observations**

**We observe the correlation coefficients between the pair of these independent variables < 0.7 indicating low to moderate correlation!**

**Multicollinearity** does not affect Tree based algorithms such as Tree based algorithms, but affects the Parametric algorithms like linear, logistic, Naive Bayes etc.

## **Checking for distribution of variables**

|  |  |
| --- | --- |
| **Fig.3.17.** Histogram for the variable, AMT\_ANNUITY | **Fig.3.18.** Histogram for the variable, DAYS\_BIRTH |
| **Fig.3.19.** Histogram for the variable, DAYS\_REGISTRATION | **Fig.3.20.** Histogram for the variable, DAYS\_ID\_PUBLISH |
| **Fig.3.21.** Histogram for the variable, EXT\_SOURCE\_2 | **Fig.3.22.** Histogram for the variable, EXT\_SOURCE\_3 |

|  |  |
| --- | --- |
| **Fig.3.23.** Histogram for the variable, b\_DAYS\_CREDIT\_max | **Fig.3.24.** Histogram for the variable, PA\_AMT\_CREDIT\_sum |
| **Fig.3.25.** Histogram for the variable, PA\_DAYS\_DECISION\_max\_y | **Fig.3.26.** Histogram for the variable, IP\_DAYS\_INSTALMENT\_min |
| **Fig.3.27.** Histogram for the variable, IP\_AMT\_INSTALMENT\_sum | **Fig.3.28.** Histogram for the variable, IP\_AMT\_INSTALMENT\_avg |

|  |  |
| --- | --- |
| **Fig.3.29.** Histogram for the variable, AT\_ANNUITY\_INCOME\_PERCENT | **Fig.3.30.** Histogram for the variable, AT\_CREDIT\_TERM |
| **Fig.3.31.** Histogram for the variable, AT\_DAYS\_EMPLOYED\_PERCENT |  |
|  |  |

**Observations**

1. For the variable, **AMT\_ANNUITY**, we observe that

range(1615.5, 258025.5)

Mean: 27108.48784108536

Median: 24903.0

Skewness: 1.5798237246955467

Kurtosis: 7.7077559537369025

The distribution is positively skewed (skewed to the right).

The distribution has heavier tails (more extreme values).

1. For the variable, **DAYS\_BIRTH**, we observe that

range(21.0, 69.0)

Mean: 43.93864609721278

Median: 43.0

Skewness: 0.1151343538689001

Kurtosis: -1.0472266784228774

The distribution is approximately symmetric.

The distribution has lighter tails (less extreme values).

1. For the variable, **DAYS\_REGISTRATION** we observe that

range(0.0, 68.0)

Mean: 13.659439824916833

Median: 12.0

Skewness: 0.5885810266508744

Kurtosis: -0.3217250717568527

The distribution is positively skewed (skewed to the right).

The distribution has lighter tails (less extreme values).

1. For the variable, **DAYS\_ID\_PUBLISH**, we observe that

range(0.0, 20.0)

Mean: 8.198575010324834

Median: 9.0

Skewness: -0.34966073846663426

Kurtosis: -1.0891039551894959

The distribution is approximately symmetric.

The distribution has lighter tails (less extreme values).

1. For the variable, **EXT\_SOURCE\_2**, we observe that

range(8.173616518884397e-08, 0.8549996664047012)

Mean: 0.514503354322039

Median: 0.5659614260608526

Skewness: -0.7959400362910248

Kurtosis: -0.26225703458375627

The distribution is negatively skewed (skewed to the left).

The distribution has lighter tails (less extreme values).

1. For the variable, **EXT\_SOURCE\_3,** we observe that

range(0.0005272652387098, 0.8960095494948396)

Mean: 0.5156949091808121

Median: 0.5352762504724826

Skewness: -0.53769669691455

Kurtosis: -0.04869212165958725

The distribution is negatively skewed (skewed to the left).

The distribution has lighter tails (less extreme values).

1. For the variable, **b\_DAYS\_CREDIT\_max**, we observe that

range(-2922.0, 0.0)

Mean: -419.25515184822655

Median: -239.0

Skewness: -2.241816733472214

Kurtosis: 5.523059055120931

The distribution is negatively skewed (skewed to the left).

The distribution has heavier tails (more extreme values).

1. For the variable, **PA\_AMT\_CREDIT\_sum**, we observe that

range(0.0, 41461128.0)

Mean: 902685.9269682546

Median: 383778.0

Skewness: 5.077017601460826

Kurtosis: 50.975539142218324

The distribution is positively skewed (skewed to the right).

The distribution has heavier tails (more extreme values).

1. For the variable, **PA\_DAYS\_DECISION\_max\_y**, we observe that

range(0.0, 8.0)

Mean: 1.228359310723844

Median: 1.0

Skewness: 2.0723345505200146

Kurtosis: 4.771499859727258

The distribution is positively skewed (skewed to the right).

The distribution has heavier tails (more extreme values).

1. For the variable, **IP\_DAYS\_INSTALMENT\_min**, we observe that

range(0.0, 8.0)

Mean: 4.131913980312899

Median: 4.0

Skewness: 0.06045041625911667

Kurtosis: -1.3808581068332964

The distribution is approximately symmetric.

The distribution has lighter tails (less extreme values).

1. For the variable, **IP\_AMT\_INSTALMENT\_sum**, we observe that

range(0.0, 2504589.66)

Mean: 17443.52688302848

Median: 11954.42052631579

Skewness: 18.351061680164857

Kurtosis: 1108.8648749117804

The distribution is positively skewed (skewed to the right).

The distribution has heavier tails (more extreme values).

1. For the variable, **IP\_AMT\_INSTALMENT\_avg**, we observe that

range(0.0002238846153846, 1.8759649122807016)

Mean: 0.1809289211110611

Median: 0.1628333333333333

Skewness: 1.5120203936578784

Kurtosis: 5.451605611925493

The distribution is positively skewed (skewed to the right).

The distribution has heavier tails (more extreme values).

1. For the variable, **AT\_ANNUITY\_INCOME\_PERCENT**, we observe that

range(0.0167896919664934, 0.1581142857142857)

Mean: 0.053695336136833555

Median: 0.05

Skewness: 1.1170129589310844

Kurtosis: 0.5276272337419181

The distribution is positively skewed (skewed to the right).

The distribution has lighter tails (less extreme values).

1. For the variable, **AT\_CREDIT\_TERM**, we observe that

range(0.0167896919664934, 0.1581142857142857)

Mean: 0.053695336136833555

Median: 0.05

Skewness: 1.1170129589310844

Kurtosis: 0.5276272337419181

The distribution is positively skewed (skewed to the right).

The distribution has lighter tails (less extreme values).

1. For the variable, **AT\_DAYS\_EMPLOYED\_PERCENT**, we observe that

range(-47.48966324275127, 0.7288114904178704)

Mean: -2.9201348735243022

Median: 0.088644630713269

Skewness: -1.7598345904832133

Kurtosis: 1.4141048013133508

The distribution is negatively skewed (skewed to the left).

The distribution has lighter tails (less extreme values).

## **Checking for presence of outliers and its treatment**

|  |  |  |
| --- | --- | --- |
| **Fig.3.32.** Boxplot for the variable, AMT\_ANNUITY | **Fig.3.33.** Boxplot for the variable, DAYS\_BIRTH | **Fig.3.34.** Boxplot for the variable, DAYS\_REGISTRATION |
| **Fig.3.35.** Boxplot for the variable, DAYS\_ID\_PUBLISH | **Fig.3.36.** Boxplot for the variable, EXT\_SOURCE\_2 | **Fig.3.37**. Boxplot for the variable, EXT\_SOURCE\_3 |
| **Fig.3.38.** Boxplot for the variable, b\_DAYS\_CREDIT\_max | **Fig.3.39.** Boxplot for the variable, PA\_AMT\_CREDIT\_sum | **Fig.3.40.** Boxplot for the variable, PA\_DAYS\_DECISION\_max\_y |
| **Fig.3.41.** Boxplot for the variable, IP\_DAYS\_INSTALMENT\_min | **Fig.3.42.** Boxplot for the variable, IP\_AMT\_INSTALMENT\_sum | **Fig.3.43.** Boxplot for the variable, IP\_AMT\_INSTALMENT\_avg |
| **Fig.3.44.** Boxplot for the variable, AT\_ANNUITY\_INCOME\_PERCENT | **Fig.3.45.** Boxplot for the variable, AT\_CREDIT\_TERM | **Fig.3.46.** Boxplot for the variable, AT\_DAYS\_EMPLOYED\_PERCENT |

**Observations**

There are No outliers in the following 4 variables as revealed by the boxplots:

1. DAYS\_BIRTH
2. DAYS\_ID\_PUBLISH
3. EXT\_SOURCE\_2
4. IP\_DAYS\_INSTALMENT\_min

#### There are outliers in the following 11 variables as revealed by the boxplots:

1. AMT\_ANNUITY
2. DAYS\_REGISTRATION
3. EXT\_SOURCE\_3
4. b\_DAYS\_CREDIT\_max
5. PA\_AMT\_CREDIT\_sum
6. PA\_DAYS\_DECISION\_max\_y
7. IP\_AMT\_INSTALMENT\_sum
8. IP\_AMT\_INSTALMENT\_avg
9. IP\_ANNUITY\_INCOME\_PERCENT
10. AT\_CREDIT\_TERM
11. AT\_DAYS\_EMPLOYED\_PERCENT

**When do you use winsorization technique?**

When you want to keep the data distribution intact. Winsorization can be useful when you want to address outliers but still maintain the original data distribution to some extent. When you have some tolerance for outliers: Winsorization allows you to set a threshold for how extreme values should be replaced, making it a flexible approach.

We will use winsorization technique using IQR method.

We cap the 11 variables affected by outliers:

* + 1. AMT\_ANNUITY
    2. DAYS\_REGISTRATION
    3. EXT\_SOURCE\_3
    4. b\_DAYS\_CREDIT\_max
    5. PA\_AMT\_CREDIT\_sum
    6. PA\_DAYS\_DECISION\_max\_y
    7. IP\_AMT\_INSTALMENT\_sum
    8. IP\_AMT\_INSTALMENT\_avg
    9. IP\_ANNUITY\_INCOME\_PERCENT
    10. AT\_CREDIT\_TERM
    11. AT\_DAYS\_EMPLOYED\_PERCENT

## **Checking for statistical significance of variables**



### **Two-sample Statistical t-test:**

A two-sample t-test is performed when you want to compare the means of two independent groups to determine if there is a statistically significant difference between them. In the context of one numeric and one categorical variable with two categories, the goal is to assess whether the means of the numeric variable are different between the two categories of the categorical variable.

**Objective**

Determine if there is a statistically significant difference in the means of the numeric variable between the two categories of the categorical variable.

**Null Hypothesis (H0):**

The means of the numeric variable are equal between the two categories.

**Alternative Hypothesis (H1):**

The means of the numeric variable are not equal between the two categories.

**Why Perform a Two-Sample T-Test:**

1. To test whether there is evidence to reject the null hypothesis and conclude that there is a significant difference in the means of the numeric variable between the two categories.
2. The t-test assesses whether the observed difference in means is statistically significant, considering the variability within each group.

**Assumption of t test:**

One of the important assumptions of t test is The distribution is approximately normal.

**Central Limit Theorem:**

For a large sample size, the population is approximately normally distributed, regardless of the distribution of the population one samples from. If the population has mean, μ and standard deviation, σ, then x has mean μ and standard deviation σ/√ n.

If the population is skewed, then the distribution of sample mean looks more and more normal when it gets larger.

Our dataset has more than 300K rows and hence the distribution is approximately normal.

**Summary:**

A two-sample t-test helps you assess whether the means of a numeric variable differ significantly between two categories of a categorical variable, providing insights into potential group differences.

| **#** | **Variable** | **T-Statistic** | **P Value** | **Remarks** |
| --- | --- | --- | --- | --- |
| 1 | AMT\_ANNUITY | 5.320847 | 1.033575e-07 | Reject the null hypothesis: There is a significant difference in means of two classes. |
| 2 | DAYS\_BIRTH | 43.533171 | 0.0 | Reject the null hypothesis: There is a significant difference in means of two classes. |
| 3 | DAYS\_REGISTRATION | 23.202349 | 5.46e-119 | Reject the null hypothesis: There is a significant difference in means of two classes. |
| 4 | DAYS\_ID\_PUBLISH | 28.514876 | 1.31e-178 | Reject the null hypothesis: There is a significant difference in means of two classes |
| 5 | EXT\_SOURCE\_2 | 90.053506 | 0.0 | Reject the null hypothesis: There is a significant difference in means of two classes. |
| 6 | EXT\_SOURCE\_3 | 87.10705 | 0.0 | Reject the null hypothesis: There is a significant difference in means of two classes. |
| 7 | b\_DAYS\_CREDIT\_max | -35.21855 | 3.6e-271 | Reject the null hypothesis: There is a significant difference in means of two classes. |
| 8 | PA\_AMT\_CREDIT\_sum | -3.115856 | 0.001834 | Reject the null hypothesis: There is a significant difference in means of two classes. |
| 9 | PA\_DAYS\_DECISION\_max\_y | 7.266284 | 3.7e-13 | Reject the null hypothesis: There is a significant difference in means of two classes |
| 10 | IP\_DAYS\_INSTALMENT\_min | 25.88620 | 1.4e-147 | Reject the null hypothesis: There is a significant difference in means of two classes |
| 11 | IP\_AMT\_INSTALMENT\_sum | 11.355365 | 7.07e-30 | Reject the null hypothesis: There is a significant difference in means of two classes |
| 12 | IP\_AMT\_INSTALMENT\_avg | 11.084198 | 1.52e-28 | Reject the null hypothesis: There is a significant difference in means of two classes |
| 13 | AT\_ANNUITY\_INCOME\_PERCENT | -8.586181 | 9.03e-18 | Reject the null hypothesis: There is a significant difference in means of two classes |
| 14 | AT\_CREDIT\_TERM | -7.026797 | 2.1e-12 | Reject the null hypothesis: There is a significant difference in means of two classes |
| 15 | AT\_DAYS\_EMPLOYED\_PERCENT | 0.800502 | 0.423420 | Fail to reject the null hypothesis: There is no significant difference in means of two classes. |

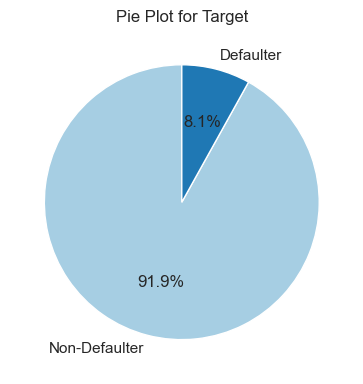
**Inference**

Except for the variable, AT\_DAYS\_EMPLOYED\_PERCENT, there is a significant difference in the means of two classes. But there is no significant difference in the ratio of days employed to income between the two groups. Hence, we will not consider the variable, AT\_DAYS\_EMPLOYED\_PERCENT for further processing.

In summary, for most of the features, there is a significant difference in means between the two groups, indicating potential distinctions between the groups for these variables. However, for the feature AT\_DAYS\_EMPLOYED\_PERCENT, there is no significant difference in means. These results provide insights into the characteristics that vary significantly between the two groups in your dataset.

Usually, the **Mann-Whitney U test** is used when the data is ordinal or when the assumptions of the t-test are not met. We will not use Mann-Whitney U test as the assumptions of t test are met.

## **Checking for class imbalance and its treatment**



**Observations**

1. We have 307511 rows and 16 columns and all are numeric columns.
2. Our target variable, is having two classes 0 (Non-defaulter) and 1 (Defaulter) with count and % as follows:

* Non-Defaulter - 282,686 (92%)
* Defaulter - 24,825 (8%)

1. Our dataset is not balanced.

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don’t add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the **Synthetic Minority Oversampling Technique, or SMOTE for short.**

# **Feature Engineering**



## **Transformations required**

We have converted absolute value of days into years by dividing by 365 for the following variables:

1. DAYS\_BIRTH
2. DAYS\_REGISTRATION
3. DAYS\_ID\_PUBLISH
4. PA\_DAYS\_DECISION\_max\_y
5. IP\_DAYS\_INSTALMENT\_min

## **Scaling the data**

Feature scaling is a data preprocessing technique used to transform the values of features or variables in a dataset to a similar scale. The purpose is to ensure that all features contribute equally to the model and to avoid the domination of features with larger values.

Feature scaling becomes necessary when dealing with datasets containing features that have different ranges, units of measurement, or orders of magnitude. In such cases, the variation in feature values can lead to biased model performance or difficulties during the learning process.

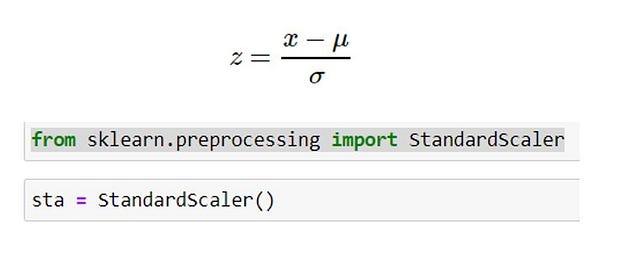
There are several common techniques for feature scaling, including standardization, normalization, and min-max scaling. These methods adjust the feature values while preserving their relative relationships and distributions.

By applying feature scaling, the dataset’s features can be transformed to a more consistent scale, making it easier to build accurate and effective machine learning models. Scaling facilitates meaningful comparisons between features, improves model convergence, and prevents certain features from overshadowing others based solely on their magnitude.

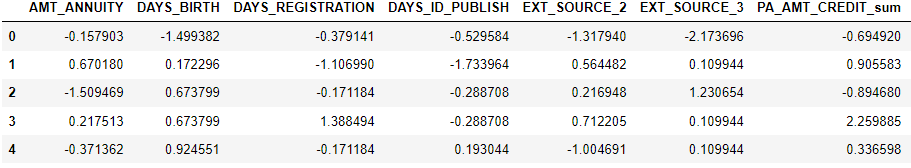
We have used Standard Scaler for our data.

**Standardization**

[StandardScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html) standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation. Standardization can be helpful in cases where the data follows a Gaussian distribution (or Normal distribution). However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

Here, μ is the mean (average) and σ is the standard deviation from the mean; standard scores (also called z scores) of the samples are calculated as follows:

First five rows of the columns, after Scaling:



## **Feature selection**

Our dataset has 307511 rows. We had 134 independent variables.

**Recursive Feature Elimination (RFE)**

Recursive Feature Elimination, or RFE for short, is a popular feature selection algorithm in a dataset that are more or most relevant in predicting the target variable. RFE applies a backward selection process to find the best combination of features.

This is done as follows:

Builds a model based on all features and calculates the importance of each feature in the model. It ranks the features and removes the feature(s) with the least importance iteratively based on model evaluation metrics such as accuracy ratio, recall.

Using the RFECV class, we achieved the feature selection by performing cross-validation evaluation of different numbers of features and automatically selecting the number of features that resulted in the best mean score.

The RFECV is configured by specifying the minimum number of features via the “min\_features\_to\_select” argument (defaults to 1) and we can also specify the type of cross-validation and scoring to use via the “cv” (defaults to 5) and “scoring” arguments (uses accuracy for classification).

We selected the 15 independent variables by performing RFE:

**Original features:**

1. AMT\_ANNUITY
2. DAYS\_BIRTH
3. DAYS\_REGISTRATION
4. DAYS\_ID\_PUBLISH
5. EXT\_SOURCE\_2
6. EXT\_SOURCE\_3

Six of the features selected by RFE are features from application train data and the remaining features are manually derived fields from other datasets, indicating our manual feature engineering was successful.

**Derived Features:**

1. b\_DAYS\_CREDIT\_max
2. PA\_AMT\_CREDIT\_sum
3. PA\_DAYS\_DECISION\_max\_y
4. IP\_DAYS\_INSTALMENT\_min
5. IP\_AMT\_INSTALMENT\_sum
6. IP\_AMT\_INSTALMENT\_avg
7. AT\_ANNUITY\_INCOME\_PERCENT
8. AT\_CREDIT\_TERM
9. AT\_DAYS\_EMPLOYED\_PERCENT

We will not consider the variable, AT\_DAYS\_EMPLOYED\_PERCENT. As per our t test, the p-value is 0.4234 and we fail to accept the null hypothesis and conclude that there is no significant difference in the means of two classes for this variable,

Now, we have 307511 rows and 15 columns (14 independent variables and 1 target variable) in our dataset.

## **Dimensionality Reduction**

Dimensionality reduction is used to reduce the number of input variables in the training data, by reducing the dimension of your feature set. When a model has a high number of features, it is naturally more complex leading to a higher chance of overfitting and decrease in accuracy.

While both feature selection and dimensionality reduction aim to reduce the number of features, they differ in their approach. Common dimensionality reduction techniques include Principal Component Analysis (PCA), Factor Analysis and Linear Discriminant Analysis (LDA). We have applied Feature selection to reduce the number of features.

**We are not going to apply dimensionality reduction techniques such as PCA, Factor Analysis and LDA.**

# **Assumptions for base model (Logistic Regression)**



## **Logistic Regression assumptions**

1. Binary logistic regression requires the target / dependent variable to be binary. For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome (such as Success etc..).
2. Only the meaningful variables should be included.
3. The predictor variables should not be correlated to each other meaning the model should have little or no multicollinearity.
4. The independent variables are linearly related to the log odds.
5. Logistic regression requires quite a large number of observations.

**Checking the assumptions:**

* 1. Binary logistic regression requires the target / dependent variable to be binary.

For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome. In our case, we have only two levels or a binary class since the target variable is loan default or not, which denotes "Loan default" or "No Loan default". Hence this assumption is satisfied.

* 1. Only the meaningful variables should be included.

It is ensured that there are no unwanted variables selected for model building.

* 1. Absence of multicollinearity

**Multicollinearity** occurs when two or more predictor variables are highly correlated with each other, such that they do not provide unique or independent information in the regression model.

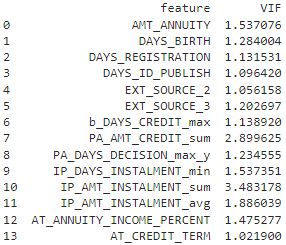
If the degree of correlation is high enough between predictor variables, it can cause problems when fitting and interpreting the regression model.

The most straightforward way to detect multicollinearity in a regression model is by calculating a metric known as the variance inflation factor, often abbreviated VIF.

VIF measures the strength of correlation between predictor variables in a model. It takes on a value between 1 and positive infinity.

We use the following rules of thumb for interpreting VIF values:

* VIF = 1: There is no correlation between a given predictor variable and any other predictor variables in the model.
* VIF between 1 and 5: There is moderate correlation between a given predictor variable and other predictor variables in the model.
* VIF > 5: There is severe correlation between a given predictor variable and other predictor variables in the model.



**Observations:**

There is no multi-collinearity as the VIF of each of the nine features is less than the threshold value of 5.

* 1. The independent variables are linearly related to the log odds

We need to check the assumption of independent variables are linearly related to the log odds. One way to checking this is to plot the independent variables in question and look for an image that resembles a S-shaped curve.

|  |  |
| --- | --- |
| **Fig.5.1.** Log odds linear plot for AMT\_ANNUITY | **Fig.5.2.** Log odds linear plot for DAYS\_BIRTH |
| **Fig.5.3.** Log odds linear plot for DAYS\_REGISTRATION | **Fig.5.4.** Log odds linear plot for DAYS\_ID\_PUBLISH |
| **Fig.5.5.** Log odds linear plot for EXT\_SOURCE\_2 | **Fig.5.6.** Log odds linear plot for EXT\_SOURCE\_3 |
| **Fig.5.7.** Log odds linear plot for b\_DAYS\_CREDIT\_max | **Fig.5.8.** Log odds linear plot for PA\_AMT\_CREDIT\_sum |
| **Fig.5.9.** Log odds linear plot for PA\_DAYS\_DECISION\_max\_y | **Fig.5.10.** Log odds linear plot for IP\_DAYS\_INSTALMENT\_min |
| **Fig.5.11.** Log odds linear plot for IP\_AMT\_INSTALMENT\_sum | **Fig.5.12**. Log odds linear plot for IP\_AMT\_INSTALMENT\_avg |
| **Fig.5.13.** Log odds linear plot for AT\_ANNUITY\_INCOME\_PERCENT | **Fig.5.14**. Log odds linear plot for AT\_CREDIT\_TERM |

* 1. Logistic regression requires quite a large number of observations.

A general rule of thumb is that at least 10 per variable. We have 307511 observations per each variable; so we have approximately 34k observations per variable.

**Observations**

**Hence all the assumptions of the Logistic Regression are satisfied.**

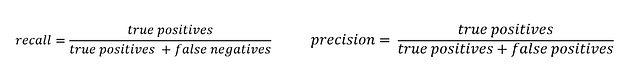
# **Building & Evaluation of ML Model**

Logistic Regression is one of the basic and popular algorithms to solve a classification problem. Logistic regression is easier to implement, interpret, and very efficient to train. Hence, we have used Logistic Regression as our base model.



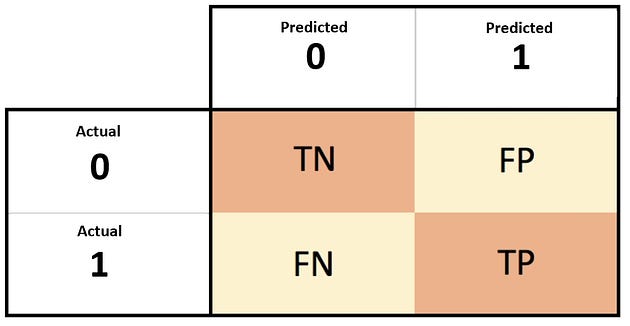
## **Performance Metrics for our base model**

Since the dataset available with us is an Imbalanced Dataset, we cannot simply use Accuracy as a metric for evaluating the performance of the model. There are some metrics that work well with imbalanced datasets, as mentioned below:

* **ROC-AUC Score:** This metric is insensitive to class imbalance. It works by ranking the probabilities of prediction of the positive class label and calculating the Area under the ROC Curve which is plotted between True Positive Rates and False Positive Rates for each threshold value.
* **Recall Score:** It is the ratio of the True Positives predicted by the model and the total number of Actual Positives. It is also known as True Positive Rate.
* **Precision Score:** It is the ratio of True Positives and the Total Positives predicted by the model.

**Fig.6.1**. Recall and Precision Score Formulae

* **Confusion Matrix:** The confusion matrix helps us to visualize the mistakes made by the model on each of the classes, be it positive or negative. Hence, it tells us about misclassifications for both classes.



**Fig.6.2**. Confusion Matrix Formulae

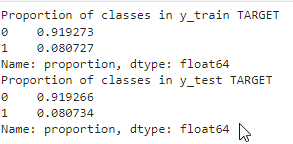
In the context of predicting whether a user will default on housing loan or not, both precision and recall are useful. However, which one to prioritize depends on the business objective. If the goal is to minimize false positives (i.e., users who are predicted to default but do not), then **precision is more important.** On the other hand, if the goal is to minimize false negatives (i.e., users who are predicted not to default but do), then **recall is more important.**

**We shall use recall as the measure of performance.**

## **Steps in building the Logistic Regression Model**



### **Split the Dataset for training and testing**



**Fig.6.2**. Confusion Matrix Formulae

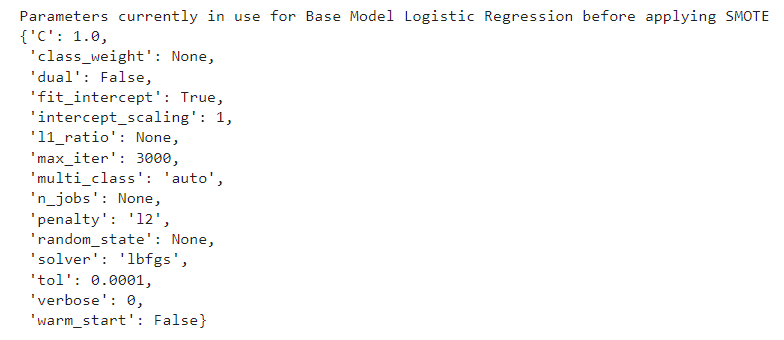
**Observations:**

We have split the dataset containing 307511 rows and 16 columns using the stratify parameter while splitting data in the ratio, 70%:30%. This parameter is only taking the output label as an argument. Hence, we must pass the target variable (y) as the parameter for stratify = y.

1. Proportion of classes in the target variable in the full dataset: 92%:8%.
2. We observe the same proportion of classes are maintained in the training and test datasets.

### **Measure of Model Performance using original data**

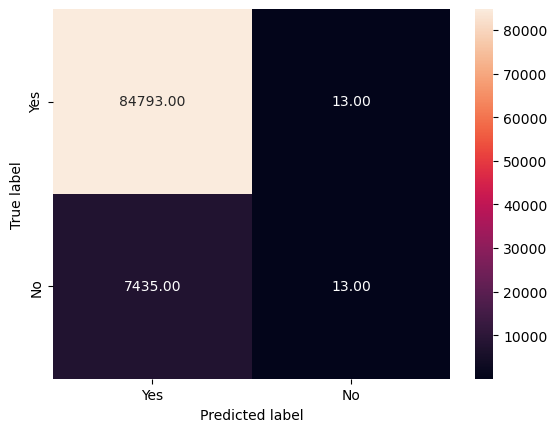
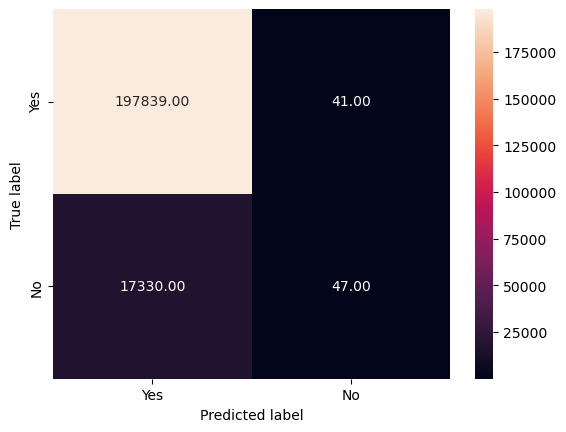
1. **Using Default model parameters**



**Fig.6.3**. Logistic Regression model parameters using default values

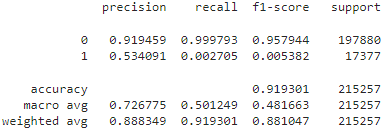
**Confusion Matrix**

Train Data Test Data



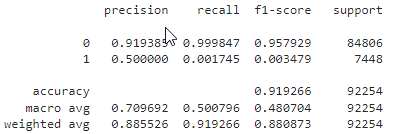
**Fig.6.4. Confusion Matrix for training and test data**

**Performance on Train data**



**Fig.6.5. Performance metrics for training data**

**Performance on Test data**

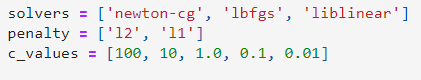


**Fig.6.6**. Performance metrics for test data

**When we compare Recall for training data (0.270%) and test data (0.175%), we observe that the model performance is very bad. Let us check if hyper-parameter tuning helps.**

1. **Using model parameters after hyper-parameter tuning**

GridSearchCV is a technique for finding the optimal parameter values from a given set of parameters in a grid. Refer: <https://holypython.com/log-reg/logistic-regression-optimization-parameters/>

****

**Fig.6.7**. Range of values for critical parameters in GridSearchCV()

**Critical parameters for Logistic Regression:**

* Solver provides options to choose solver algorithm for optimization. We need to choose one from the available list comprising lbfgs (Limited-memory BFGS), liblinear, newton-cg, sag, saga.
* Penalty defines penalization norms. Options include l1, l2, elasticnet and none.
* C parameter signifies strength of the regularization and takes a positive float value. Smaller the C stronger the regularization will be.

****

**Fig.6.8**. Recall value after hyper-parameter tuning

We observe that even after hyper-parameter tuning (C to 100 from 1, solver to ‘newton-cg’ from lbfgs), recall value is 0.2647% only. So, hyper-parameter tuning did not help in improving the performance measure.

[Imbalanced datasets](https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/)pose a common challenge for machine learning practitioners in binary classification problems. This scenario frequently arises in practical business applications like[fraud detection,](https://www.analyticsvidhya.com/blog/2022/03/exploratory-data-analysis-eda-credit-card-fraud-detection-case-study/) [spam filtering](https://www.analyticsvidhya.com/blog/2021/06/automated-spam-e-mail-detection-modelusing-common-nlp-tasks/), rare disease discovery, and hardware fault detection. To address this issue, one popular technique is Synthetic Minority Oversampling Technique (SMOTE). SMOTE is specifically designed to tackle imbalanced datasets by generating synthetic samples for the minority class

**Dealing with Imbalanced Data**

Resampling data is one of the most preferred approaches to deal with an imbalanced dataset. There are broadly two types of methods for this

* Under-sampling
* Over-sampling

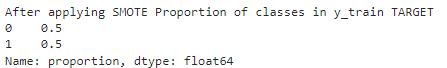
In most cases, oversampling is preferred over under-sampling techniques. The reason being, in under-sampling we tend to remove instances from data that may be carrying some important information. In this article, I am specifically covering some special data augmentation oversampling techniques: SMOTE and its related counterparts.

**SMOTE: Synthetic Minority Oversampling Technique**

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.

**Create training Data using SMOTE**

**Fig.6.8**. Applying SMOTE on Training data



**Fig.6.9**. Proportion of classes in the Target variable in the Training data after SMOTE

**Observations**

1. We have split the dataset containing 307511 rows and 16 columns using the stratify parameter while splitting data in the ratio, 70%:30%. This parameter is only taking the output label as an argument. So, we have to pass the target variable (y) as the parameter for stratify = y.

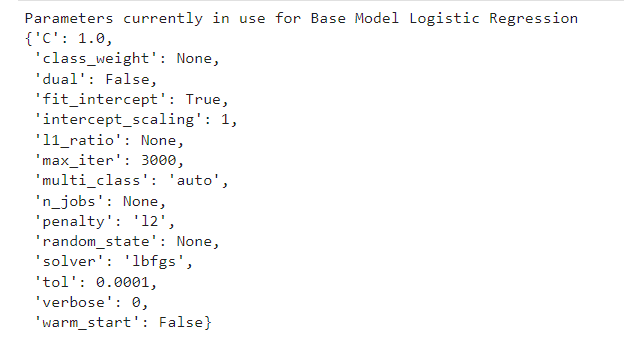
2. Train dataset contains 452296 rows and 16 columns.

3. Proportion of classes in the target variable in the SMOTE Train dataset: 50%:50%.

4. Proportion of classes in the target variable in the SMOTE Train dataset: 92%:8%.

### **Measure of Model Performance after applying SMOTE on Training data**

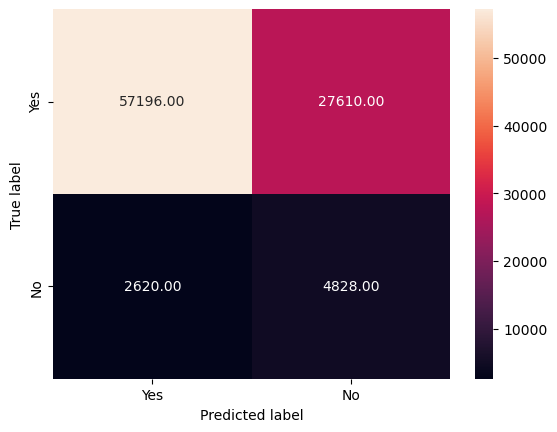
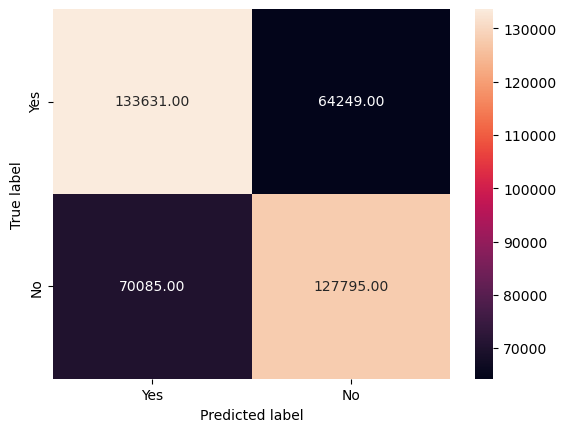
1. **Using Default model parameters**



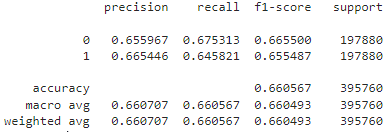
**Fig.6.10**. Default parameter values for Logistic Regression after applying SMOTE

**Confusion Matrix:**

**Train Data Test Data**

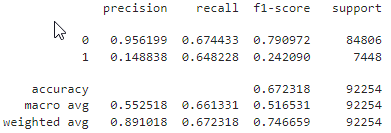
**Fig.6.8**. Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.6.9**. Performance metrics for training data

**Performance on Test Performance**



**Fig.6.10**. Performance metrics for test data

When we compare Recall for training data (64.58%) and test data (64.82%), we observe that the model performance has improved a lot. There is no model overfit as the measures for both training and test data are in the same range.

1. **Using model parameters after hyper-parameter tuning**



We observe that even after hyper-parameter tuning (C to 0.01 from 1.0, penalty to l1 from l2, solver to ‘liblinear’ from lbfgs), recall value is 64.63% from 64.58%. So, hyper-parameter tuning did marginally help in improving the performance measure.

### **Tasks planned for future**

1. We need to try other models such as Decision Tree, Random Forest, KNN, XGBoost etc.
2. We will also tune the model parameters using hyper-parameter tuning method.
3. Choose the best model and write our recommendations based on the variable importance plots for the best model.
4. **More ML Models**

**7.1. K-Nearest Neighbours :**

K-Nearest Neighbors (KNN) is a simple yet effective machine learning algorithm used for both classification and regression tasks. It belongs to the category of instance-based learning, where the model memorizes the training instances and makes predictions based on the similarity of new instances to those in the training set.

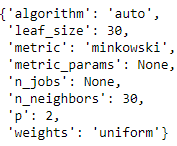
In the context of classification, KNN works by finding the K nearest neighbors of a data point based on a distance metric (e.g., Euclidean distance) and assigning the majority class among those neighbors as the predicted class for the data point. The value of K is a hyperparameter that needs to be tuned based on the dataset and the problem at hand.

KNN is a non-parametric and lazy learning algorithm, meaning it does not make any assumptions about the underlying data distribution and does not learn a specific model during training. Instead, it performs computation at the time of prediction.

One of the key considerations when using KNN is the choice of distance metric, as it heavily influences the algorithm's performance. Additionally, KNN can be sensitive to the scale of the features, so it's often beneficial to normalize or standardize the data before applying the algorithm.

### **7.1.1** **Measure of Model Performance after applying SMOTE on Training data**

**1. Model parameters used for KNN**

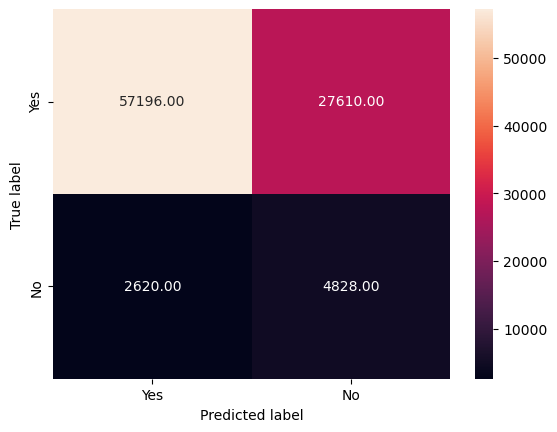
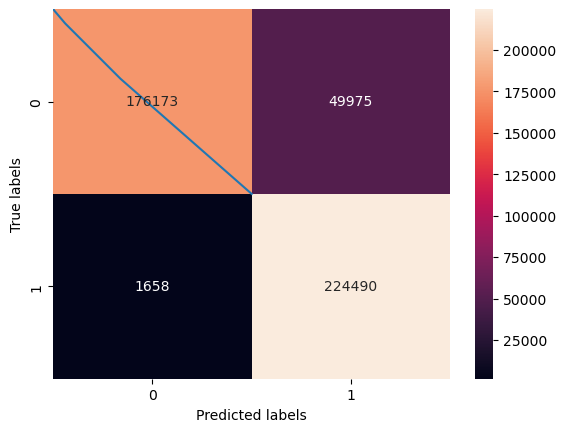


**Fig.7.1**. Parameter values used for KNN Model after applying SMOTE obtained using

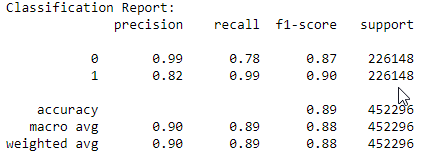
Hyperparameter tuning.

**Confusion Matrix:**

**Train Data Test Data**

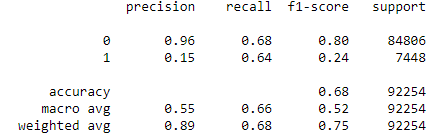
**Fig.7.2**. Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.7.3**. Performance metrics for training data

**Performance on Test Performance**



**Fig.7.4**. Performance metrics for test data

**Training Performance:**

* **Accuracy:** 91.88%
* **Precision (Class 1):** 50.00%
* **Recall (Class 1):** 99.00%
* **F1-score (Class 1):** 90.00%

The model has high accuracy but low recall for Class 1, indicating that it is good at predicting Class 0 but performs poorly for Class 1.

**Test Performance:**

* **Accuracy:** 60.91%
* **Precision (Class 1):** 12.00%
* **Recall (Class 1):** 64.00%
* **F1-score (Class 1):** 21.00%

The model's accuracy is lower on the test data compared to the training data, and it has a low precision but higher recall for Class 1. This means that the model is better at identifying Class 1 instances but at the cost of more false positives.

**Overall Summary:**

* The model trained on SMOTE data shows improved performance in terms of recall for the minority class (Class 1) compared to the model trained without SMOTE.
* However, the model still has limitations, as indicated by the low precision and overall modest performance metrics on the test data.

**7.2. Decision Tree Model**

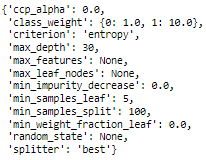
A decision tree is a popular and easy-to-understand model used in machine learning for both classification and regression tasks. It's a tree-like structure where:

* Each internal node represents a "decision" based on a feature.
* Each branch represents the outcome of the decision.
* Each leaf node represents the final decision or prediction.

The goal of a decision tree is to create a model that predicts the value of a target variable based on several input variables. It does this by recursively splitting the data into subsets based on the input features, with each split maximizing the homogeneity of the target variable within the subsets.

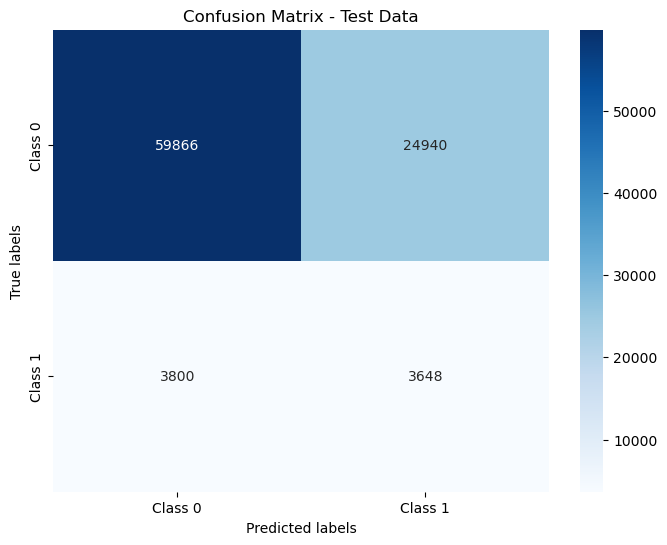
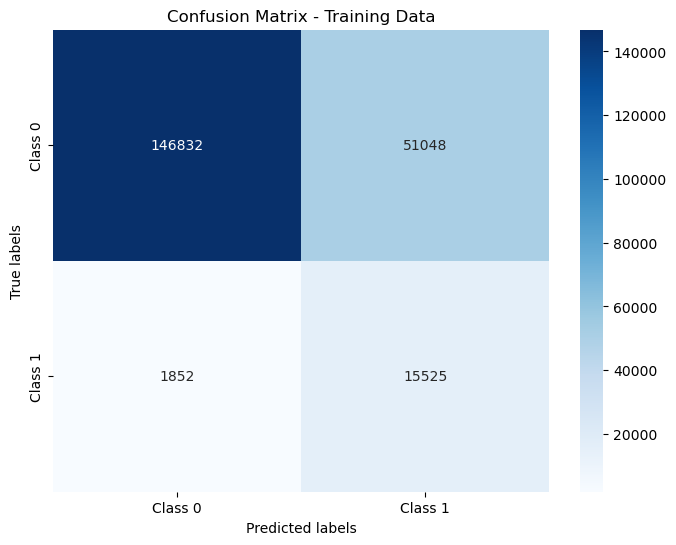
### **Measure of Model Performance after applying SMOTE on Training data**

**1. Model parameters used for Decision Tree model**

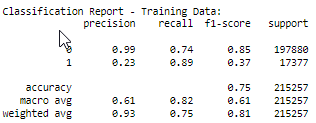


**Fig.7.2.1**. Parameter values used for Decision Tree Model on Imbalanced Data

**Confusion Matrix:**

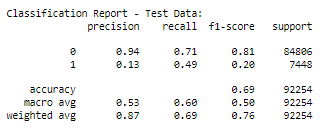
**Train Data Test Data****Fig.7.2**.2 Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.7.2**.3 Performance metrics for training data

**Performance on Test Performance**



**Fig.7.2**.4 Performance metrics for test data

1. **Training Performance**:
   * **Accuracy**: 75.4%
   * **Precision (Class 1)**: 23.3%
   * **Recall (Class 1)**: 89.3%
   * **F1 Score (Class 1)**: 37.0%

The model performs well in terms of recall for Class 1 (Defaulters), indicating that it can correctly identify a high percentage of defaulters among all actual defaulters. However, the precision is relatively low, suggesting that there are a significant number of false positives among the predicted defaulters. The F1 score, which is the harmonic mean of precision and recall, provides a balance between these two metrics.

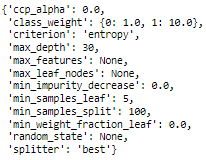
1. **Test Performance**:
   * **Accuracy**: 68.8%
   * **Precision (Class 1)**: 12.8%
   * **Recall (Class 1)**: 49.0%
   * **F1 Score (Class 1)**: 20.2%

The test performance shows similar trends to the training performance, with relatively low precision and F1 score for Class 1. This indicates that the model struggles to generalize to unseen data, particularly in correctly identifying defaulters. The recall is higher than precision, suggesting that the model is better at identifying actual defaulters but may also predict a large number of false positives.

In summary, while the model using class weights improves recall for Class 1, it comes at the cost of lower precision and overall performance on the test data. Further tuning of the model's hyperparameters or exploration of different algorithms may be necessary to improve its performance, especially in correctly identifying defaulters while minimizing false positives.

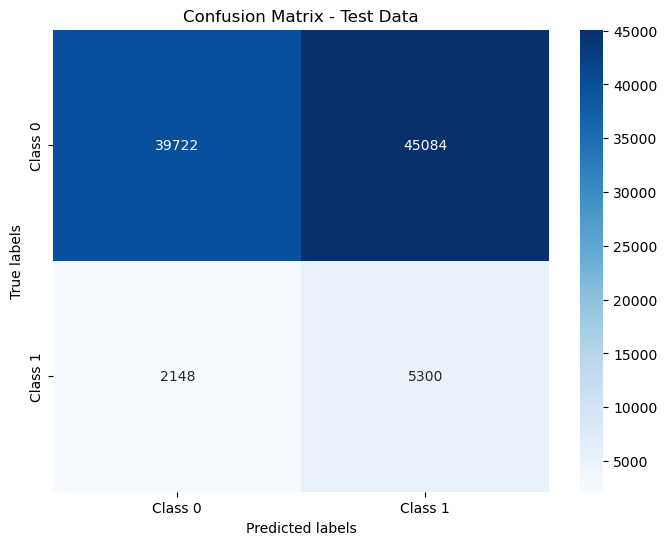
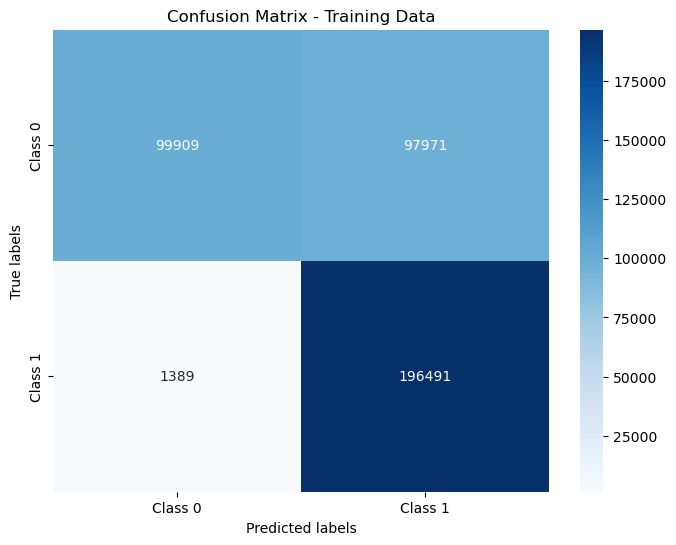
### **7.2.2 Measure of Model Performance after applying SMOTE on Training data**

**1. Model parameters used for Decision Tree model after SMOTE**

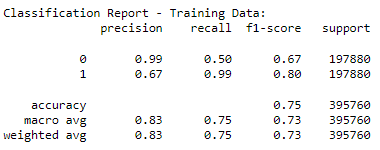


**Fig.7.2**.**5** Parameter values used for Decision Tree Model on SMOTE Data

**Confusion Matrix:**

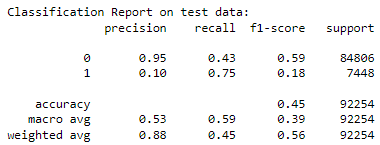
**Train Data Test Data****Fig.7.2**.**6** Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.7.2**.3 Performance metrics for training data

**Performance on Test Performance**



**Fig.7.2**.4 Performance metrics for test data

**Training Performance (SMOTE Data):**

* **Accuracy**: 73.00%
* **Precision (Class 1):** 65.00%
* **Recall (Class 1):** 99.00%
* **F1-score (Class 1):** 79.00%

**Test Performance (SMOTE Data):**

* **Accuracy**: 45.00%
* **Precision (Class 1):** 10.00%
* **Recall (Class 1):** 75.00%
* **F1-score (Class 1):** 18.00%

**Summary:**

* The model trained on SMOTE data shows improved recall for the minority class (Class 1) in the test data compared to the model trained without SMOTE.
* However, there is still room for improvement, especially in the F1-score for Class 1, which indicates a trade-off between precision and recall.

7.3 **Extreme Gradient Boosting**

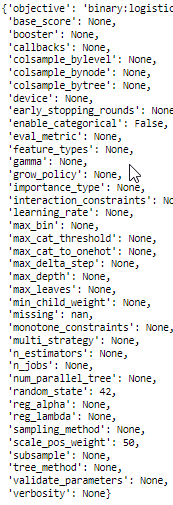
XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that is widely used for regression and classification tasks. It belongs to a class of ensemble learning algorithms known as gradient boosting machines. XGBoost is known for its speed and performance, often outperforming other algorithms on a wide range of datasets.

Here are some key features of XGBoost:

1. **Gradient Boosting:** XGBoost is based on the principle of gradient boosting, which builds an ensemble of weak learners (typically decision trees) sequentially, with each tree correcting the errors of its predecessor.
2. **Regularization:** XGBoost has built-in regularization techniques that help prevent overfitting, such as L1 and L2 regularization, which add penalty terms to the loss function.
3. **Handling Missing Values:** XGBoost can handle missing values in the dataset, eliminating the need for imputation techniques.
4. **Parallel Processing:** XGBoost is designed for efficiency and speed, with support for parallel and distributed computing, making it suitable for large datasets.
5. **Tree Pruning:** XGBoost uses a technique called tree pruning to remove splits that have little impact on improving the model's performance, which helps reduce complexity and overfitting.
6. **Cross-Validation:** XGBoost supports built-in cross-validation capabilities, making it easier to tune hyperparameters and evaluate the model's performance.
7. **Feature Importance:** XGBoost provides insights into feature importance, helping to identify which features are most influential in making predictions.

### **Measure of Model Performance on original Data**

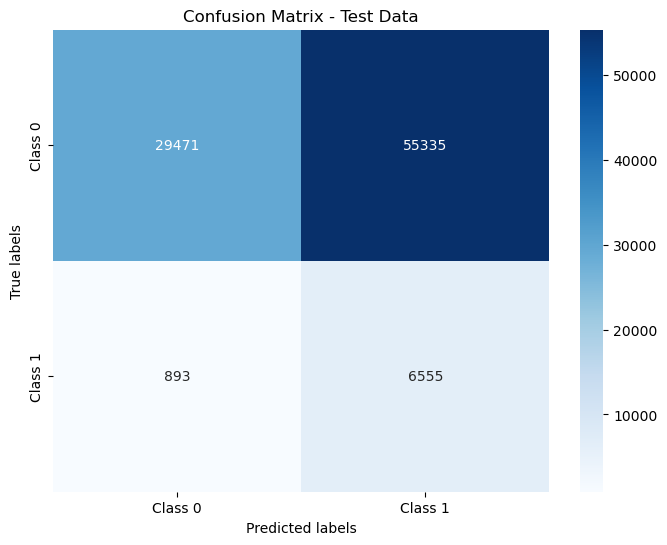
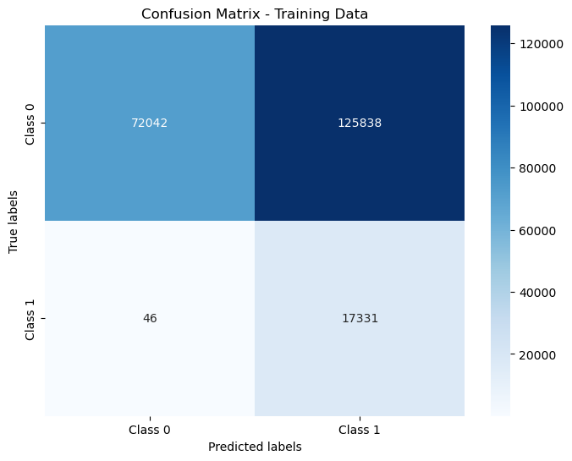
**1. Model parameters used for XGBoost model on imbalanced data**



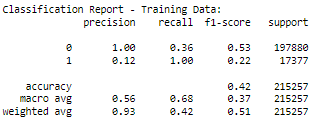
**Fig.7.3**.**1** Parameter values used for Decision Tree Model on Imbalanced Data

**Confusion Matrix:**

**Train Data Test Data**

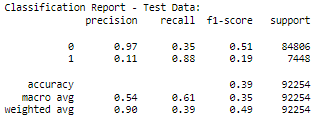
**Fig.7.3**.**2** Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.7.3**.**3** Performance metrics for training data

**Performance on Test Performance**



**Fig.7.3**.**4** Performance metrics for test data

**Training Performance**:

* + **Accuracy**: 42.0%
  + **Precision (Class 1)**: 12.0%
  + **Recall (Class 1)**: 88%
  + **F1 Score (Class 1)**: 19.0%

1. Despite the class imbalance, the model achieves an impressive recall of 1.00 for Class 1, meaning it correctly identifies all instances of the minority class in the training data.
2. The precision for Class 0 is high at 1.00, indicating that the model makes very few false positive predictions for the majority class.

**Test Performance**:

* + **Accuracy**: 39.0%
  + **Precision (Class 1)**: 11.0%
  + **Recall (Class 1)**: 88.0%
  + **F1 Score (Class 1)**: 19.0%

1. The model maintains a high recall of 0.88 for Class 1 on the test data, demonstrating its ability to generalize well and identify the minority class instances.
2. The precision for Class 0 remains strong at 0.97, indicating a low rate of false positives for the majority class.

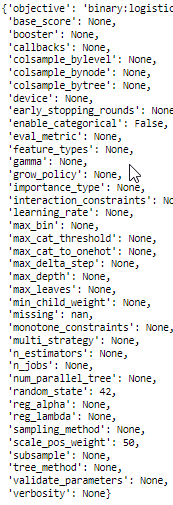
The XGBoost model trained with the **scale\_pos\_weight** parameter shows promising results, especially considering the importance of recall for Class 1.

**Overall Summary:**

* The XGBoost model with **scale\_pos\_weight** effectively addresses the class imbalance and achieves high recall for the minority class, which was a key objective.
* Despite the high recall for Class 1, there is a trade-off with precision, as the model tends to have a higher false positive rate for Class 1.

### **Measure of Model Performance after applying SMOTE on Training data**

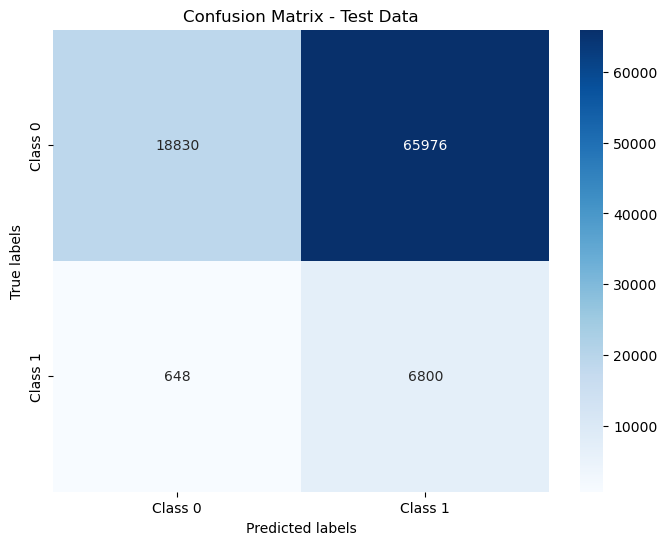
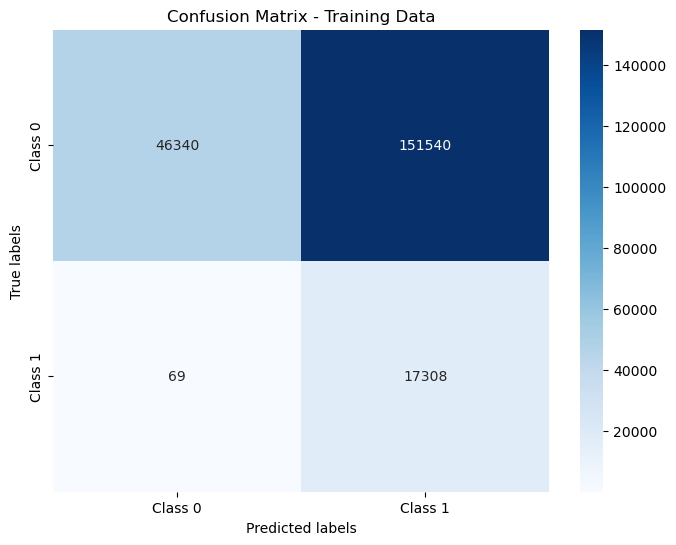
**1. Model parameters used for XGBoost model on SMOTE data**



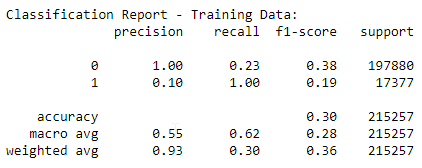
**Fig.7.3**.**5** Parameter values used for Decision Tree Model on SMOTE Data

**Confusion Matrix:**

**Train Data Test Data**

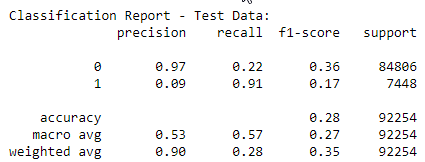
**Fig.7.3**.**6** Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.7.3**.**7** Performance metrics for training data

**Performance on Test Performance**



**Fig.7.3**.**8** Performance metrics for test data

**Training Performance**:

* + **Accuracy**: 42.0%
  + **Precision (Class 1)**: 12.0%
  + **Recall (Class 1)**: 88%
  + **F1 Score (Class 1)**: 19.0%

Despite the class imbalance, the model achieves an impressive recall of 1.00 for Class 1, meaning it correctly identifies all instances of the minority class in the training data.

The precision for Class 0 is high at 1.00, indicating that the model makes very few false positive predictions for the majority class.

**Test Performance**:

* + **Accuracy**: 39.0%
  + **Precision (Class 1)**: 11.0%
  + **Recall (Class 1)**: 88.0%
  + **F1 Score (Class 1)**: 19.0%
  1. **ADA Boost Model**

### **Measure of Model Performance after applying SMOTE on Training data**

**1. Model parameters used for ADABoost model on SMOTE data**

AdaBoost (Adaptive Boosting) is an ensemble learning method that combines multiple weak classifiers to create a strong classifier. It was originally formulated by Yoav Freund and Robert Schapire in 1996. AdaBoost is particularly useful for binary classification problems and is known for its high accuracy and ability to handle complex decision boundaries.

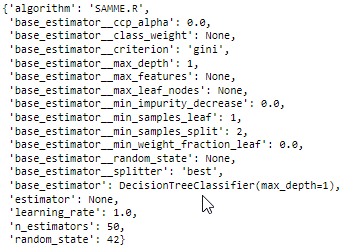
Here's a brief overview of how AdaBoost works:

1. **Initialization**: AdaBoost assigns equal weights to each sample in the training dataset.
2. **Iteration**: For each iteration, AdaBoost trains a weak classifier on the training data. A weak classifier is a simple model that performs slightly better than random chance.
3. **Weighted Training**: During training, AdaBoost adjusts the weights of incorrectly classified samples to focus more on those samples in the next iteration. This process is repeated for a predefined number of iterations or until a perfect classifier is achieved.
4. **Combination of Classifiers**: The final prediction is made by combining the predictions of all the weak classifiers. The weight of each classifier's prediction is based on its accuracy.

AdaBoost is effective because it focuses on difficult-to-classify examples, improving the overall performance of the ensemble. It is also resistant to overfitting and can handle noisy data well. However, AdaBoost can be sensitive to outliers in the dataset.

Top of Form

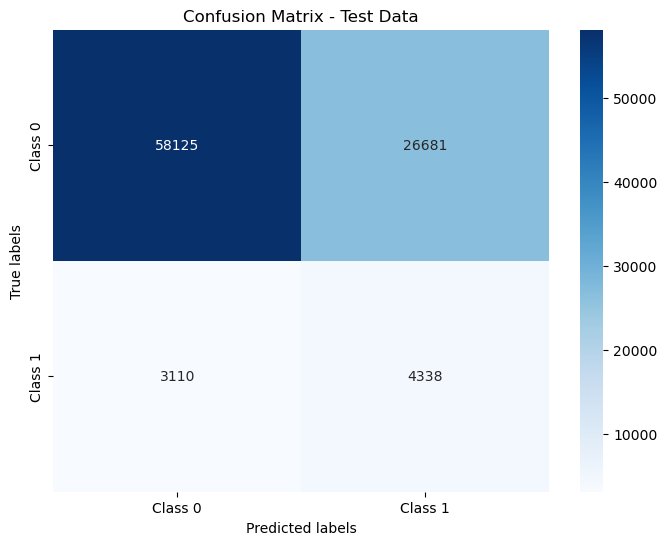
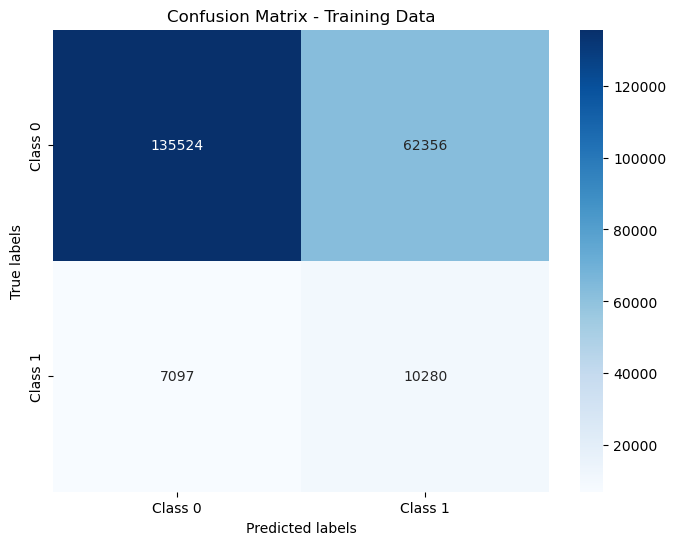
**1. Model parameters used for ADA Boost model on SMOTE data**



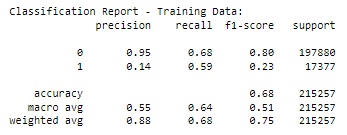
**Fig.7.4**.**1** Parameter values used for ADA BOOST on SMOTE Data

**Confusion Matrix:**

**Train Data Test Data**

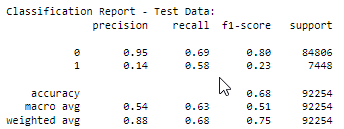
**Fig.7.4**.**2** Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.7.4**.**4** Performance metrics for training data

**Performance on Test Performance**



**Fig.7.4**.**5** Performance metrics for test data

**Performance on Training Data (SMOTE):**

* **Accuracy**: 68.0%
* **Precision (Class 1):** 14.0%
* **Recall (Class 1):** 59.0%
* **F1 Score (Class 1):** 23.0%

Despite the class imbalance, the model achieves a recall of 59.0% for Class 1, meaning it correctly identifies 59.0% of actual Class 1 instances in the training data.

The precision for Class 0 is high at 95.0%, indicating that the model makes very few false positive predictions for the majority class.

**Performance on Test Data (SMOTE):**

* **Accuracy**: 68.0%
* **Precision (Class 1):** 14.0%
* **Recall (Class 1):** 58.0%
* **F1 Score (Class 1):** 23.0%

The model maintains a recall of 58.0% for Class 1 on the test data, demonstrating its ability to generalize well and identify the minority class instances.

The precision for Class 0 remains strong at 95.0%, indicating a low rate of false positives for the majority class.

**Overall Summary:**

* The AdaBoost model trained on SMOTE data achieves a balanced performance with similar results on both training and test datasets.
* The model shows a moderate ability to identify Class 1 instances (minority class), with a recall of 59.0% on the training data and 58.0% on the test data.
  1. **Gradient Boosting Model**

Gradient Boosting is a powerful machine learning technique used for both regression and classification tasks. It is an ensemble method that builds a strong model by combining the predictions of multiple weak models, typically decision trees. Gradient Boosting, including the popular implementation known as Gradient Boosting Machine (GBM), works by sequentially adding models to an ensemble, where each new model corrects errors made by the previous models.

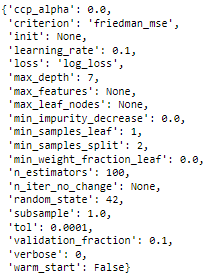
1. Base Learners: Gradient Boosting starts by initializing the model with a single base learner, usually a decision tree with a small depth.
2. Sequential Training: It then trains additional base learners sequentially. Each new learner is trained to correct the errors made by the combination of the existing learners.
3. Gradient Descent: The key idea behind Gradient Boosting is to optimize the loss function of the model using gradient descent. It calculates the gradient of the loss function with respect to the model's prediction and updates the model in the negative gradient direction to minimize the loss.
4. Combination of Models: The final prediction is made by combining the predictions of all the base learners, usually by taking a weighted sum of their predictions.
5. Regularization: Gradient Boosting includes regularization techniques to prevent overfitting, such as controlling the number of base learners (trees) and the learning rate (shrinkage).

Gradient Boosting is known for its high predictive accuracy and is widely used in various machine learning competitions and real-world applications.

### **7.5.1 Measure of Model Performance after applying SMOTE**

Top of Form

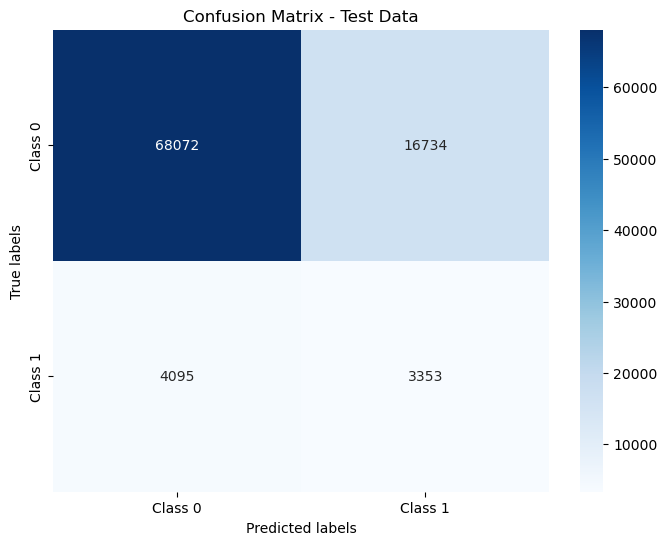
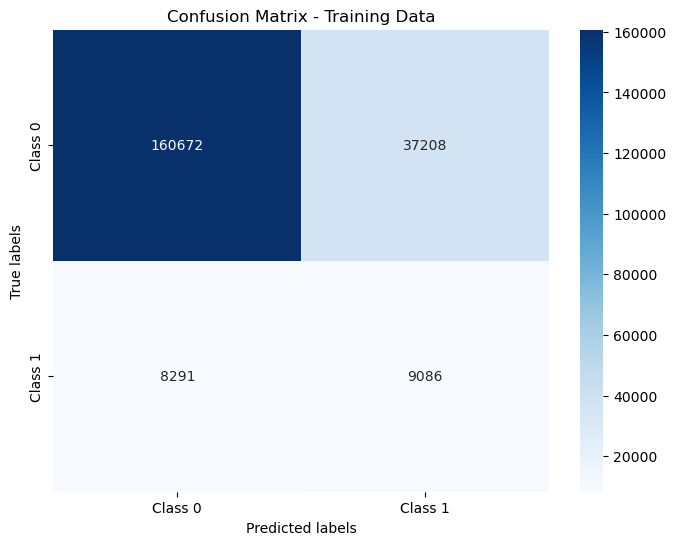
**1. Model parameters used for Gradient Boosting model on SMOTE data**



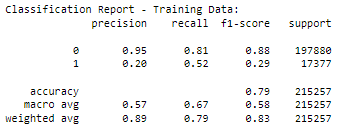
**Fig.7.5**.**1** Parameter values used for Gradient BOOST on SMOTE Data

**Confusion Matrix:**

**Train Data Test Data**

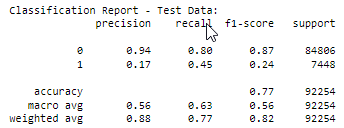
**Fig.7.5**.**2** Confusion Matrix for training and test data

**Performance on Train Performance**



**Fig.7.5**.**3** Performance metrics for training data

**Performance on Test Performance**



**Fig.7.5**.**4** Performance metrics for test data

**Performance on Training Data (SMOTE):**

* **Accuracy**: 79.0%
* **Precision (Class 1):** 20.0%
* **Recall (Class 1):** 52.0%
* **F1 Score (Class 1):** 29.0%

Despite the class imbalance, the model achieves a recall of 52.0% for Class 1, meaning it correctly identifies 52.0% of actual Class 1 instances in the training data.

The precision for Class 0 is high at 95.0%, indicating that the model makes very few false positive predictions for the majority class.

**Performance on Test Data (SMOTE):**

* **Accuracy**: 77.0%
* **Precision (Class 1):** 17.0%
* **Recall** **(Class 1):** 45.0%
* **F1 Score (Class 1):** 24.0%

The model maintains a recall of 45.0% for Class 1 on the test data, demonstrating its ability to generalize well and identify the minority class instances.

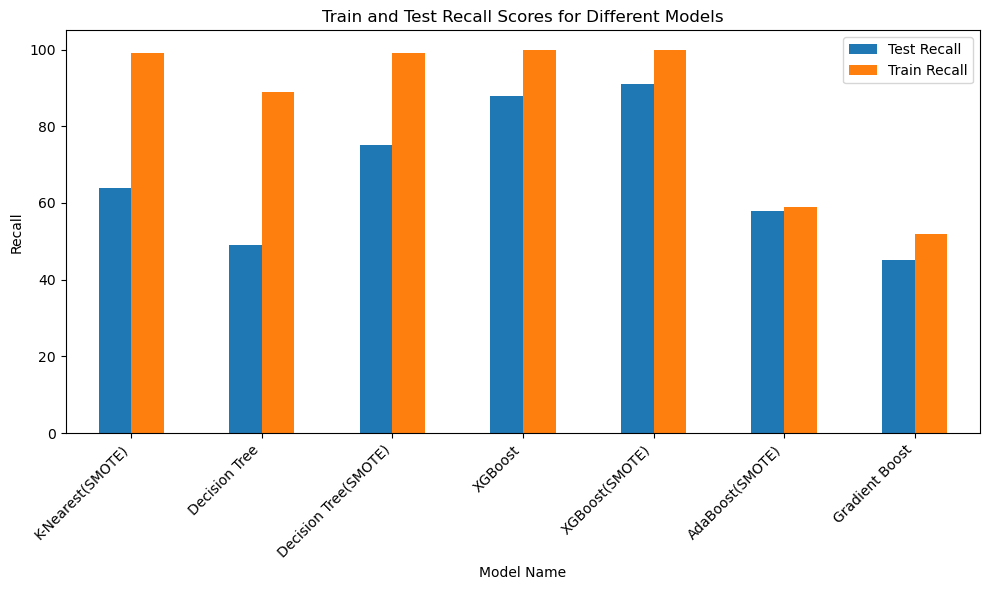
The precision for Class 0 remains strong at 94.0%, indicating a low rate of false positives for the majority class.

**Overall Summary:**

The Gradient Boosting model trained on SMOTE data achieves a balanced performance with similar results on both training and test datasets. The model shows a moderate ability to identify Class 1 instances (minority class), with a recall of 52.0% on the training data and 45.0% on the test data.

1. **Model Comparisons**

**8.1 On Basis of recall**



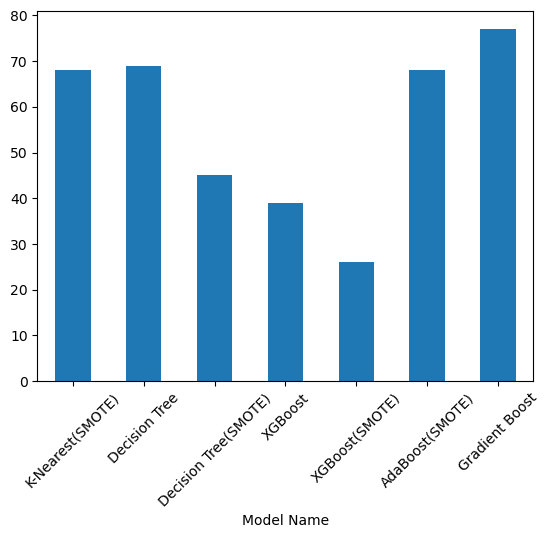
**K-Nearest(SMOTE) and Decision Tree(SMOTE):** These models trained with Synthetic Minority Over-sampling Technique (SMOTE) exhibit higher recall scores on the test dataset compared to their counterparts without SMOTE. This suggests that SMOTE has effectively balanced the class distribution, resulting in better model performance, particularly in correctly identifying positive cases (e.g., credit defaults).

**XGBoost and XGBoost(SMOTE):** Both versions of the XGBoost model demonstrate high recall scores on the test dataset, with XGBoost(SMOTE) slightly outperforming the base XGBoost model. This indicates that XGBoost is effective in capturing complex relationships within the data, and applying SMOTE further enhances its performance in handling imbalanced classes.

**Decision Tree and AdaBoost(SMOTE):** These models show relatively lower recall scores on the test dataset compared to other models. This suggests that Decision Tree and AdaBoost models might not capture the complexities of the data as effectively as other ensemble methods like XGBoost and Gradient Boosting.

**Gradient Boost(SMOTE):** Despite having lower recall scores on both test and train datasets compared to other models, Gradient Boosting achieves the highest weighted average recall on the test dataset. This indicates that while Gradient Boosting may not perform as well on individual classes, it achieves a good overall balance between precision and recall, resulting in a higher weighted average recall.

**8.2. Comparison on the basis of Weighted Avg(Recall)**



**Decision Tree**(SMOTE): Despite achieving high recall scores for individual classes, the weighted average recall is relatively low (45%). This indicates that while the model performs well in identifying certain classes, it struggles with others, possibly leading to an imbalance in performance across different groups.

**XGBoost**(SMOTE): This model has the lowest weighted average recall (26%) among all models. Despite achieving high recall scores for specific classes, the overall performance suffers due to the imbalanced distribution of classes or misclassification in certain groups.

**K-Nearest**(SMOTE): Despite having a weighted average recall of 68%, which is higher than some other models, it still indicates room for improvement. The model may struggle with class imbalances or misclassification in specific groups, affecting its overall performance.

**Gradient Boost**: Despite having relatively lower recall scores for individual classes, the model achieves the highest weighted average recall (77%). This suggests that the model effectively balances precision and recall across different classes, resulting in a better overall performance.

**8.3 Final Model (XG BOOST):**

In our project, we are tasked with detecting credit defaulters, which is a critical task in the financial industry. Our objective is to build a model that can accurately identify individuals who are likely to default on their credit obligations.

To achieve this, we have trained several models, including XGBoost models with and without SMOTE data. SMOTE (Synthetic Minority Over-sampling Technique) is a method used to address class imbalance, which is common in credit default prediction where the number of defaulters is typically much lower than non-defaulters.

Upon evaluating the models, we found that the XGBoost model without SMOTE data has a recall value of 88%, which is the second highest among all models tested. The model with the highest recall (91%) is also an XGBoost model but trained on SMOTE data. Despite its higher recall, we have decided to choose the model without SMOTE data as our final model.

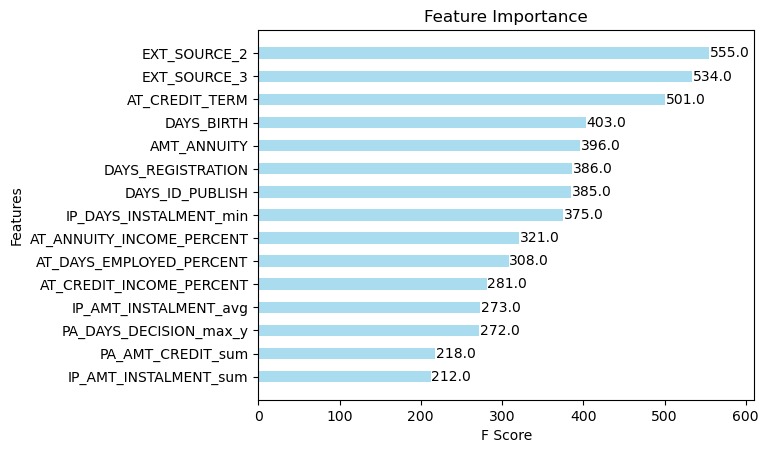
The reason for this decision is that, in our context, recall is more important than precision. Recall, also known as sensitivity, measures the ability of the model to correctly identify all actual positive cases (defaulters) in the dataset. A high recall indicates that the model can capture a large portion of the defaulters, which is crucial for our objective of identifying potential defaulters.

While the model trained on SMOTE data has a higher recall, we prefer the model without SMOTE data because it still achieves a relatively high recall of 88% while avoiding the potential drawbacks of SMOTE, such as introducing synthetic data that may not accurately represent the underlying distribution of the minority class.

In summary, we have chosen the XGBoost model without SMOTE data as our final model because it provides a good balance between high recall and avoiding potential issues associated with SMOTE, aligning with our project objective of detecting credit defaulters.

**9. Feature Importance**

Feature importance analysis is a critical aspect of predictive modeling in machine learning, offering valuable insights into the relevance and impact of individual features on the target variable.This is feature importance table of XGBoost Model.

****

**Fig.9.1 Feature Importance**

EXT\_SOURCE\_3 (0.144843) and EXT\_SOURCE\_2 (0.127533) appear to be the most influential features. These features likely correspond to external credit rating sources and could be crucial indicators of creditworthiness.

AT\_CREDIT\_TERM (0.095446) also has a relatively high importance value. This feature might represent the duration of the credit term, which could be significant in assessing the risk associated with the loan.

Features such as DAYS\_BIRTH (0.057707) and DAYS\_REGISTRATION (0.045652) have moderate importance values, indicating that age and the duration since registration might also play a role in predicting credit risk.

Other features, like AMT\_ANNUITY, DAYS\_ID\_PUBLISH, and PA\_DAYS\_DECISION\_max\_y, have relatively lower importance values but still contribute to the model's predictive power.

**10. Project conclusion**

In conclusion, our capstone project on loan default detection has culminated in the development of an XGBoost model with a recall of 88%. This model represents a significant achievement in addressing the critical task of identifying potential loan defaulters in the financial industry.

Throughout the project, our main focus has been on maximizing recall, as it is crucial to correctly identify as many defaulters as possible to mitigate financial risks for lenders. The model's high recall of 88% means that it can effectively capture a large portion of actual defaulters, providing valuable insights for lenders to take proactive measures.

The implications of this model are substantial. By leveraging machine learning algorithms like XGBoost, lenders can enhance their risk assessment processes, leading to more informed lending decisions. This could result in reduced default rates, improved portfolio performance, and ultimately, a more sustainable and profitable lending business.

In summary, our XGBoost model for loan default detection represents a significant step forward in leveraging data-driven approaches to mitigate financial risks. By prioritizing recall and developing a robust machine learning model, we have contributed to the advancement of risk management practices in the financial sector.

In addition to its direct impact on risk management practices, our XGBoost model can also benefit other stakeholders in the financial ecosystem. For regulators, this model can serve as a tool for monitoring and ensuring compliance with lending regulations, ultimately contributing to a more stable financial system.

Furthermore, the insights gained from our model can inform policy decisions aimed at promoting responsible lending practices and protecting consumers. By identifying patterns and factors associated with loan default, our model can help policymakers develop targeted interventions and support programs for at-risk borrowers.

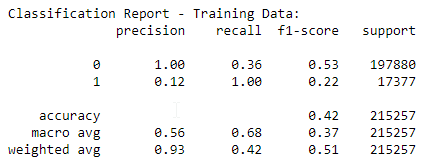
Overall, our capstone project underscores the transformative potential of machine learning in the financial industry. By harnessing the power of data and advanced analytics, we have developed a model that not only improves risk management for lenders but also has broader implications for financial stability and consumer protection.

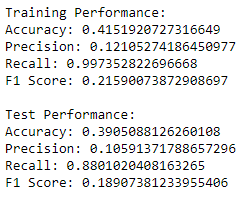
# **11. Appendix**



## **9.1 Data Dictionary**

| **1)** | **application\_train.csv** | |
| --- | --- | --- |
| **#** | **Column** | **Description** |
| 1 | *SK\_ID\_CURR* | ID of loan in our sample |
| 2 | **TARGET** | **Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)** |
| 3 | NAME\_CONTRACT\_TYPE | Identification if loan is cash or revolving |
| 4 | CODE\_GENDER | Gender of the client |
| 5 | FLAG\_OWN\_CAR | Flag if the client owns a car |
| 6 | FLAG\_OWN\_REALTY | Flag if client owns a house or flat |
| 7 | CNT\_CHILDREN | Number of children the client has |
| 8 | AMT\_INCOME\_TOTAL | Income of the client |
| 9 | AMT\_CREDIT | Credit amount of the loan |
| 10 | AMT\_ANNUITY | Loan annuity |
| 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 was accompanying client when he was applying for the loan |
| 13 | NAME\_INCOME\_TYPE | Clients income type (businessman, working, maternity leave,…) |
| 14 | NAME\_EDUCATION\_TYPE | Level of highest education the client achieved |
| 15 | NAME\_FAMILY\_STATUS | Family status of the client |
| 16 | NAME\_HOUSING\_TYPE | What is the housing situation of the client (renting, living with parents, ...) |
| 17 | REGION\_POPULATION\_RELATIVE | Normalized population of region where client lives (higher number means the client lives in more populated region) |
| 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\_3 | 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 | Number of enquiries to Credit Bureau about the client one hour before application |
| 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) |

****

****

| **2)** | **bureau.csv** | |
| --- | --- | --- |
| **#** | **Column** | **Description** |
| 1 | SK\_ID\_CURR | ID of loan in our sample - one loan in our sample can have 0,1,2 or more related previous credits in credit bureau |
| 2 | SK\_BUREAU\_ID | Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application) |
| 3 | CREDIT\_ACTIVE | Status of the Credit Bureau (CB) reported credits |
| 4 | CREDIT\_CURRENCY | Recoded currency of the Credit Bureau credit |
| 5 | DAYS\_CREDIT | How many days before current application did client apply for Credit Bureau credit |
| 6 | CREDIT\_DAY\_OVERDUE | Number of days past due on CB credit at the time of application for related loan in our sample |
| 7 | DAYS\_CREDIT\_ENDDATE | Remaining duration of CB credit (in days) at the time of application in Home Credit |
| 8 | DAYS\_ENDDATE\_FACT | Days since CB credit ended at the time of application in Home Credit (only for closed credit) |
| 9 | AMT\_CREDIT\_MAX\_OVERDUE | Maximal amount overdue on the Credit Bureau credit so far (at application date of loan in our sample) |
| 10 | CNT\_CREDIT\_PROLONG | How many times was the Credit Bureau credit prolonged |
| 11 | AMT\_CREDIT\_SUM | Current credit amount for the Credit Bureau credit |
| 12 | AMT\_CREDIT\_SUM\_DEBT | Current debt on Credit Bureau credit |
| 13 | AMT\_CREDIT\_SUM\_LIMIT | Current credit limit of credit card reported in Credit Bureau |
| 14 | AMT\_CREDIT\_SUM\_OVERDUE | Current amount overdue on Credit Bureau credit |
| 15 | CREDIT\_TYPE | Type of Credit Bureau credit (Car, cash,...) |
| 16 | DAYS\_CREDIT\_UPDATE | How many days before loan application did last information about the Credit Bureau credit come |
| 17 | AMT\_ANNUITY | Annuity of the Credit Bureau credit |

| **3)** | **bureau\_balance.csv** | |
| --- | --- | --- |
| **#** | **Column** | **Description** |
| 1 | SK\_BUREAU\_ID | Recoded ID of Credit Bureau credit (unique coding for each application) - use this to join to CREDIT\_BUREAU table |
| 2 | MONTHS\_BALANCE | Month of balance relative to application date (-1 means the freshest balance date) |
| 3 | STATUS | Status of Credit Bureau loan during the month (active, closed, DPD0-30,… [C means closed, X means status unknown, 0 means no DPD, 1 means maximal did during month between 1-30, 2 means DPD 31-60,… 5 means DPD 120+ or sold or written off ] ) |

| **4)** | **POS\_CASH\_balance.csv** | |
| --- | --- | --- |
| **#** | **Column** | **Description** |
| 1 | SK\_ID\_PREV | ID of previous credit in Home Credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) |
| 2 | SK\_ID\_CURR | ID of loan in our sample |
| 3 | MONTHS\_BALANCE | Month of balance relative to application date (-1 means the information to the freshest monthly snapshot, 0 means the information at application - often it will be the same as -1 as many banks are not updating the information to Credit Bureau regularly ) |
| 4 | CNT\_INSTALMENT | Term of previous credit (can change over time) |
| 5 | CNT\_INSTALMENT\_FUTURE | Installments left to pay on the previous credit |
| 6 | NAME\_CONTRACT\_STATUS | Contract status during the month |
| 7 | SK\_DPD | DPD (days past due) during the month of previous credit |
| 8 | SK\_DPD\_DEF | DPD during the month with tolerance (debts with low loan amounts are ignored) of the previous credit |

| **5)** | **credit\_card\_balance.csv** | |
| --- | --- | --- |
| **#** | **Column** | **Description** |
| 1 | SK\_ID\_PREV | ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) |
| 2 | SK\_ID\_CURR | ID of loan in our sample |
| 3 | MONTHS\_BALANCE | Month of balance relative to application date (-1 means the freshest balance date) |
| 4 | AMT\_BALANCE | Balance during the month of previous credit |
| 5 | AMT\_CREDIT\_LIMIT\_ACTUAL | Credit card limit during the month of the previous credit |
| 6 | AMT\_DRAWINGS\_ATM\_CURRENT | Amount drawing at ATM during the month of the previous credit |
| 7 | AMT\_DRAWINGS\_CURRENT | Amount drawing during the month of the previous credit |
| 8 | AMT\_DRAWINGS\_OTHER\_CURRENT | Amount of other drawings during the month of the previous credit |
| 9 | AMT\_DRAWINGS\_POS\_CURRENT | Amount drawing or buying goods during the month of the previous credit |
| 10 | AMT\_INST\_MIN\_REGULARITY | Minimal installment for this month of the previous credit |
| 11 | AMT\_PAYMENT\_CURRENT | How much did the client pay during the month on the previous credit |
| 12 | AMT\_PAYMENT\_TOTAL\_CURRENT | How much did the client pay during the month in total on the previous credit |
| 13 | AMT\_RECEIVABLE\_PRINCIPAL | Amount receivable for principal on the previous credit |
| 14 | AMT\_RECIVABLE | Amount receivable on the previous credit |
| 15 | AMT\_TOTAL\_RECEIVABLE | Total amount receivable on the previous credit |
| 16 | CNT\_DRAWINGS\_ATM\_CURRENT | Number of drawings at ATM during this month on the previous credit |
| 17 | CNT\_DRAWINGS\_CURRENT | Number of drawings during this month on the previous credit |
| 18 | CNT\_DRAWINGS\_OTHER\_CURRENT | Number of other drawings during this month on the previous credit |
| 19 | CNT\_DRAWINGS\_POS\_CURRENT | Number of drawings for goods during this month on the previous credit |
| 20 | CNT\_INSTALMENT\_MATURE\_CUM | Number of paid installments on the previous credit |
| 21 | NAME\_CONTRACT\_STATUS | Contract status (active signed,...) on the previous credit |
| 22 | SK\_DPD | DPD (Days past due) during the month on the previous credit |
| 23 | SK\_DPD\_DEF | DPD (Days past due) during the month with tolerance (debts with low loan amounts are ignored) of the previous credit |

| **6)** | **previous\_application.csv** | |
| --- | --- | --- |
| **#** | **Column** | **Description** |
| 1 | SK\_ID\_PREV | ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loan applications in Home Credit, previous application could, but not necessarily have to lead to credit) |
| 2 | SK\_ID\_CURR | ID of loan in our sample |
| 3 | NAME\_CONTRACT\_TYPE | Contract product type (Cash loan, consumer loan [POS] ,...) of the previous application |
| 4 | AMT\_ANNUITY | Annuity of previous application |
| 5 | AMT\_APPLICATION | For how much credit did client ask on the previous application |
| 6 | AMT\_CREDIT | Final credit amount on the previous application. This differs from AMT\_APPLICATION in a way that the AMT\_APPLICATION is the amount for which the client initially applied for, but during our approval process he could have received different amount - AMT\_CREDIT |
| 7 | AMT\_DOWN\_PAYMENT | Down payment on the previous application |
| 8 | AMT\_GOODS\_PRICE | Goods price of good that client asked for (if applicable) on the previous application |
| 9 | WEEKDAY\_APPR\_PROCESS\_START | On which day of the week did the client apply for previous application |
| 10 | HOUR\_APPR\_PROCESS\_START | Approximately at what day hour did the client apply for the previous application |
| 11 | FLAG\_LAST\_APPL\_PER\_CONTRACT | Flag if it was last application for the previous contract. Sometimes by mistake of client or our clerk there could be more applications for one single contract |
| 12 | NFLAG\_LAST\_APPL\_IN\_DAY | Flag if the application was the last application per day of the client. Sometimes clients apply for more applications a day. Rarely it could also be error in our system that one application is in the database twice |
| 13 | NFLAG\_MICRO\_CASH | Flag Micro finance loan |
| 14 | RATE\_DOWN\_PAYMENT | Down payment rate normalized on previous credit |
| 15 | RATE\_INTEREST\_PRIMARY | Interest rate normalized on previous credit |
| 16 | RATE\_INTEREST\_PRIVILEGED | Interest rate normalized on previous credit |
| 17 | NAME\_CASH\_LOAN\_PURPOSE | Purpose of the cash loan |
| 18 | NAME\_CONTRACT\_STATUS | Contract status (approved, cancelled, ...) of previous application |
| 19 | DAYS\_DECISION | Relative to current application when was the decision about previous application made |
| 20 | NAME\_PAYMENT\_TYPE | Payment method that client chose to pay for the previous application |
| 21 | CODE\_REJECT\_REASON | Why was the previous application rejected |
| 22 | NAME\_TYPE\_SUITE | Who accompanied client when applying for the previous application |
| 23 | NAME\_CLIENT\_TYPE | Was the client old or new client when applying for the previous application |
| 24 | NAME\_GOODS\_CATEGORY | What kind of goods did the client apply for in the previous application |
| 25 | NAME\_PORTFOLIO | Was the previous application for CASH, POS, CAR, … |
| 26 | NAME\_PRODUCT\_TYPE | Was the previous application x-sell o walk-in |
| 27 | CHANNEL\_TYPE | Through which channel we acquired the client on the previous application |
| 28 | SELLERPLACE\_AREA | Selling area of seller place of the previous application |
| 29 | NAME\_SELLER\_INDUSTRY | The industry of the seller |
| 30 | CNT\_PAYMENT | Term of previous credit at application of the previous application |
| 31 | NAME\_YIELD\_GROUP | Grouped interest rate into small medium and high of the previous application |
| 32 | PRODUCT\_COMBINATION | Detailed product combination of the previous application |
| 33 | DAYS\_FIRST\_DRAWING | Relative to application date of current application when was the first disbursement of the previous application |
| 34 | DAYS\_FIRST\_DUE | Relative to application date of current application when was the first due supposed to be of the previous application |
| 35 | DAYS\_LAST\_DUE\_1ST\_VERSION | Relative to application date of current application when was the first due of the previous application |
| 36 | DAYS\_LAST\_DUE | Relative to application date of current application when was the last due date of the previous application |
| 37 | DAYS\_TERMINATION | Relative to application date of current application when was the expected termination of the previous application |
| 38 | NFLAG\_INSURED\_ON\_APPROVAL | Did the client requested insurance during the previous application |

| **7)** | **installments\_payments.csv** | |
| --- | --- | --- |
| **#** | **Column** | **Description** |
| 1 | SK\_ID\_PREV | ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) |
| 2 | SK\_ID\_CURR | ID of loan in our sample |
| 3 | NUM\_INSTALMENT\_VERSION | Version of installment calendar (0 is for credit card) of previous credit. Change of installment version from month to month signifies that some parameter of payment calendar has changed |
| 4 | NUM\_INSTALMENT\_NUMBER | On which installment we observe payment |
| 5 | DAYS\_INSTALMENT | When the installment of previous credit was supposed to be paid (relative to application date of current loan) |
| 6 | DAYS\_ENTRY\_PAYMENT | When was the installments of previous credit paid actually (relative to application date of current loan) |
| 7 | AMT\_INSTALMENT | What was the prescribed installment amount of previous credit on this installment |
| 8 | AMT\_PAYMENT | What the client actually paid on previous credit on this installment |

## **Missing values**

**File name 1: application\_train.csv**

| **#** | **Column** | **NaN Count** | **NaN Percentage** |
| --- | --- | --- | --- |
| 1 | COMMONAREA\_AVG | 214865 | 69.8723 |
| 2 | COMMONAREA\_MEDI | 214865 | 69.8723 |
| 3 | COMMONAREA\_MODE | 214865 | 69.8723 |
| 4 | NONLIVINGAPARTMENTS\_AVG | 213514 | 69.433 |
| 5 | NONLIVINGAPARTMENTS\_MEDI | 213514 | 69.433 |
| 6 | NONLIVINGAPARTMENTS\_MODE | 213514 | 69.433 |
| 7 | FONDKAPREMONT\_MODE | 210295 | 68.3862 |
| 8 | LIVINGAPARTMENTS\_AVG | 210199 | 68.355 |
| 9 | LIVINGAPARTMENTS\_MEDI | 210199 | 68.355 |
| 10 | LIVINGAPARTMENTS\_MODE | 210199 | 68.355 |
| 11 | FLOORSMIN\_AVG | 208642 | 67.8486 |
| 12 | FLOORSMIN\_MEDI | 208642 | 67.8486 |
| 13 | FLOORSMIN\_MODE | 208642 | 67.8486 |
| 14 | YEARS\_BUILD\_AVG | 204488 | 66.4978 |
| 15 | YEARS\_BUILD\_MEDI | 204488 | 66.4978 |
| 16 | YEARS\_BUILD\_MODE | 204488 | 66.4978 |
| 17 | OWN\_CAR\_AGE | 202929 | 65.9908 |
| 18 | LANDAREA\_AVG | 182590 | 59.3767 |
| 19 | LANDAREA\_MEDI | 182590 | 59.3767 |
| 20 | LANDAREA\_MODE | 182590 | 59.3767 |
| 21 | BASEMENTAREA\_AVG | 179943 | 58.516 |
| 22 | BASEMENTAREA\_MEDI | 179943 | 58.516 |
| 23 | BASEMENTAREA\_MODE | 179943 | 58.516 |
| 24 | EXT\_SOURCE\_1 | 173378 | 56.3811 |
| 25 | NONLIVINGAREA\_AVG | 169682 | 55.1792 |
| 26 | NONLIVINGAREA\_MEDI | 169682 | 55.1792 |
| 27 | NONLIVINGAREA\_MODE | 169682 | 55.1792 |
| 28 | ELEVATORS\_AVG | 163891 | 53.296 |
| 29 | ELEVATORS\_MEDI | 163891 | 53.296 |
| 30 | ELEVATORS\_MODE | 163891 | 53.296 |
| 31 | WALLSMATERIAL\_MODE | 156341 | 50.8408 |
| 32 | APARTMENTS\_AVG | 156061 | 50.7497 |
| 33 | APARTMENTS\_MEDI | 156061 | 50.7497 |
| 34 | APARTMENTS\_MODE | 156061 | 50.7497 |
| 35 | ENTRANCES\_AVG | 154828 | 50.3488 |
| 36 | ENTRANCES\_MEDI | 154828 | 50.3488 |
| 37 | ENTRANCES\_MODE | 154828 | 50.3488 |
| 38 | LIVINGAREA\_AVG | 154350 | 50.1933 |
| 39 | LIVINGAREA\_MEDI | 154350 | 50.1933 |
| 40 | LIVINGAREA\_MODE | 154350 | 50.1933 |
| 41 | HOUSETYPE\_MODE | 154297 | 50.1761 |
| 42 | FLOORSMAX\_AVG | 153020 | 49.7608 |
| 43 | FLOORSMAX\_MEDI | 153020 | 49.7608 |
| 44 | FLOORSMAX\_MODE | 153020 | 49.7608 |
| 45 | YEARS\_BEGINEXPLUATATION\_AVG | 150007 | 48.781 |
| 46 | YEARS\_BEGINEXPLUATATION\_MEDI | 150007 | 48.781 |
| 47 | YEARS\_BEGINEXPLUATATION\_MODE | 150007 | 48.781 |
| 48 | TOTALAREA\_MODE | 148431 | 48.2685 |
| 49 | EMERGENCYSTATE\_MODE | 145755 | 47.3983 |
| 50 | OCCUPATION\_TYPE | 96391 | 31.3455 |
| 51 | EXT\_SOURCE\_3 | 60965 | 19.8253 |
| 52 | AMT\_REQ\_CREDIT\_BUREAU\_DAY | 41519 | 13.5016 |
| 53 | AMT\_REQ\_CREDIT\_BUREAU\_HOUR | 41519 | 13.5016 |
| 54 | AMT\_REQ\_CREDIT\_BUREAU\_MON | 41519 | 13.5016 |
| 55 | AMT\_REQ\_CREDIT\_BUREAU\_QRT | 41519 | 13.5016 |
| 56 | AMT\_REQ\_CREDIT\_BUREAU\_WEEK | 41519 | 13.5016 |
| 57 | AMT\_REQ\_CREDIT\_BUREAU\_YEAR | 41519 | 13.5016 |
| 58 | NAME\_TYPE\_SUITE | 1292 | 0.4201 |
| 59 | DEF\_30\_CNT\_SOCIAL\_CIRCLE | 1021 | 0.332 |
| 60 | DEF\_60\_CNT\_SOCIAL\_CIRCLE | 1021 | 0.332 |
| 61 | OBS\_30\_CNT\_SOCIAL\_CIRCLE | 1021 | 0.332 |
| 62 | OBS\_60\_CNT\_SOCIAL\_CIRCLE | 1021 | 0.332 |
| 63 | EXT\_SOURCE\_2 | 660 | 0.2146 |
| 64 | AMT\_GOODS\_PRICE | 278 | 0.0904 |
| 65 | AMT\_ANNUITY | 12 | 0.0039 |
| 66 | CNT\_FAM\_MEMBERS | 2 | 0.0007 |
| 67 | DAYS\_LAST\_PHONE\_CHANGE | 1 | 0.0003 |

**File name 2: bureau.csv**

| **#** | **Column** | **NaN Count** | | **NaN Percentage** |
| --- | --- | --- | --- | --- |
| 1 | AMT\_ANNUITY | | 1226791 | 71.4735 |
| 2 | AMT\_CREDIT\_MAX\_OVERDUE | | 1124488 | 65.5133 |
| 3 | DAYS\_ENDDATE\_FACT | | 633653 | 36.917 |
| 4 | AMT\_CREDIT\_SUM\_LIMIT | | 591780 | 34.4774 |
| 5 | AMT\_CREDIT\_SUM\_DEBT | | 257669 | 15.0119 |
| 6 | DAYS\_CREDIT\_ENDDATE | | 105553 | 6.1496 |
| 7 | AMT\_CREDIT\_SUM | | 13 | 0.0008 |

**File name 3: credit\_card\_balance.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **NaN Count** | **NaN Percentage** |
| 1 | AMT\_PAYMENT\_CURRENT | 767988 | 19.9981 |
| 2 | AMT\_DRAWINGS\_ATM\_CURRENT | 749816 | 19.5249 |
| 3 | AMT\_DRAWINGS\_OTHER\_CURRENT | 749816 | 19.5249 |
| 4 | AMT\_DRAWINGS\_POS\_CURRENT | 749816 | 19.5249 |
| 5 | CNT\_DRAWINGS\_ATM\_CURRENT | 749816 | 19.5249 |
| 6 | CNT\_DRAWINGS\_OTHER\_CURRENT | 749816 | 19.5249 |
| 7 | CNT\_DRAWINGS\_POS\_CURRENT | 749816 | 19.5249 |
| 8 | AMT\_INST\_MIN\_REGULARITY | 305236 | 7.9482 |
| 9 | CNT\_INSTALMENT\_MATURE\_CUM | 305236 | 7.9482 |

**File name 4: installments\_payments.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **NaN Count** | **NaN Percentage** |
| 1 | AMT\_PAYMENT | 2905 | 0.0214 |
| 2 | DAYS\_ENTRY\_PAYMENT | 2905 | 0.0214 |

**File name 5: POS\_CASH\_balance.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **NaN Count** | **NaN Percentage** |
| **1** | CNT\_INSTALMENT\_FUTURE | 26087 | 0.2608 |
| **2** | CNT\_INSTALMENT | 26071 | 0.2607 |

**File name 6: previous\_application.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **NaN Count** | **NaN Percentage** |
| 1 | RATE\_INTEREST\_PRIMARY | 1664263 | 99.6437 |
| 2 | RATE\_INTEREST\_PRIVILEGED | 1664263 | 99.6437 |
| 3 | AMT\_DOWN\_PAYMENT | 895844 | 53.6365 |
| 4 | RATE\_DOWN\_PAYMENT | 895844 | 53.6365 |
| 5 | NAME\_TYPE\_SUITE | 820405 | 49.1198 |
| 6 | DAYS\_FIRST\_DRAWING | 673065 | 40.2981 |
| 7 | DAYS\_FIRST\_DUE | 673065 | 40.2981 |
| 8 | DAYS\_LAST\_DUE | 673065 | 40.2981 |
| 9 | DAYS\_LAST\_DUE\_1ST\_VERSION | 673065 | 40.2981 |
| 10 | DAYS\_TERMINATION | 673065 | 40.2981 |
| 11 | NFLAG\_INSURED\_ON\_APPROVAL | 673065 | 40.2981 |
| 12 | AMT\_GOODS\_PRICE | 385515 | 23.0818 |
| 13 | AMT\_ANNUITY | 372235 | 22.2867 |
| 14 | CNT\_PAYMENT | 372230 | 22.2864 |
| 15 | PRODUCT\_COMBINATION | 346 | 0.0207 |
| 16 | AMT\_CREDIT | 1 | 0.0001 |