YILDIZ TECHNICAL UNIVERSITY
FACULTY OF CHEMISTRY AND METALLURGY
MATHEMATICAL ENGINEERING DEPARTMENT

MULTIDISCIPLINARY DESIGN PROJECT

Machine Learning for Credit Risk Prediction: An Analysis of the Home Credit Dataset

21058024, Serhet GÖKDEMİR

Advisor: Prof. Dr. Hale GONCE KÖÇKEN

İstanbul, 2026

CONTENTS

List of Figures

LIST OF TABLES

Symbols

e	Euler's Number
θ_j	Threshold
β_i	Coefficient
X_i	Feature
P(Y=1)	Probability notation

ABBREVIATIONS

ML	Machine Learning

Preface

All analyses, observations, and progress documented in this report were carried out under supervision of advisor Prof. Dr. Hale Gonce Köçken. The work finished in accordance with the objectives and requirements of the project.

Abstract

Credit risk prediction is a fundamental problem for financial institutions, as accurate identification of high-risk applicants is essential for reducing financial losses while maintaining fair lending decisions. This study addresses the credit default prediction problem using the Home Credit Default Risk dataset, focusing exclusively on application-level information contained in the application_train.csv and application_test.csv files. The target variable exhibits a strong class imbalance, with default cases forming a small minority, which necessitates imbalance-aware modeling and evaluation strategies.

A structured preprocessing and feature engineering pipeline is employed to handle common data quality issues, including high missing value ratios, anomalous values such as extreme employment durations, and high-cardinality categorical variables. Missing values are treated using appropriate imputation strategies, rare categories are grouped, and several domain-informed ratio features are created to better capture applicants' financial capacity. After encoding and dataset alignment, multiple classification models are trained and evaluated.

Logistic regression is used as a baseline model. Although it achieves a reasonable ROC–AUC score, its performance in identifying default cases is limited due to class imbalance. Class weighting significantly improves minority-class recall without sacrificing overall discrimination, whereas SMOTE oversampling does not provide additional benefits. Decision threshold optimization based on precision–recall trade-offs is applied to better reflect credit risk considerations. Subsequently, XGBoost models are trained to capture non-linear relationships, and class weighting together with hyperparameter refinement leads to improved performance and a more balanced recall–precision profile. Feature importance analysis highlights external credit score variables and engineered financial ratios as the most influential predictors.

Overall, the study presents a complete and academically sound machine learning workflow, covering data preprocessing, feature engineering, model comparison, and threshold selection. While the achieved performance levels are moderate, the results are consistent with the limited scope of the data used and provide a meaningful baseline for credit default prediction based solely on application-level information.

1. INTRODUCTION

1.1 What is Machine Learning?

Machine Learning is the ability of a computer to learn from information and past experiences about a situation, so it can make decisions and provide solutions for similar events in the future [1]. More generally, it can be considered as a application of mathematical functions that take input variables, process the input variables and produce outputs by discovering patterns in the data [6].

By using computing methods, we become able to design systems that can learn from data by being trained via computers. The systems can learn and improve with experience, and with time developed model that can be used to predict outcomes of questions based on the previous learning process [2].

There are too many different algorithms that can be used in Machine Learning problems. The required output is the main decision criteria for which to use. Machine Learning algorithms are divided under one of these two learning types: supervised learning and unsupervised learning [2].

1.2 Supervised Learning

Supervised learning is a method of ML that learns the relationship between input variables and an output variable. Supervised learning uses this relationship to predict results for a given new, unseen data. It is the most important approach in machine learning and plays a crictical role in processing multimedia data [7]. Thus, supervised learning can be defined as learning from labeled training data.

For example, in a banking dataset classification problem, each input could be customer's previous banking activity records and the output can be "Potential Customer" or "Not Potential Customer". By training on a set of labeled examples, the system learns the patterns in the data and can classify the new inputs with an accuration rate.

In supervised learning, problems fall into two categories according to the type of their output: classification and regression.

1.2.1 Classification

Classification is a supervised learning method that is used to predict group patterns for data inputs. Classification tasks are one of the most well known tasks in machine learning applications, especially in future planning problems [3].

The field of classification covers four primary task types. These are binary classification, multi-class classification, multi-label classification, and imbalanced

classification [8]. However, in this study, we consider two primary types of classification problems: binary and multi-class classification.

Since a data set with a binary-valued target variable is dealt with, binary classification is the most suitable option for credit default risk analyis.

There are many binary classification methods. The most common methods are: logistic regression, support vector machine, decision trees, random forest, gradient boosting, k-NN, naive bayes and neural networks. The selected methods to solve mentioned problem will be explained in the next sections.

1.2.2 Regression

A regression problem can be described as a problem of predicting a numeric (continuous) label and the target variable is a numeric (continuous) value [8].

In regression, we try to understand how variables are affected by each other. For instance, we could try to understand the relationship between study time and final test scores, people's weights and heights, or we can illustrate how many transistors the semiconductor industry is capable of fitting into a circuit across time, as shown in Figure 1.1 (Moore's Law) [8].

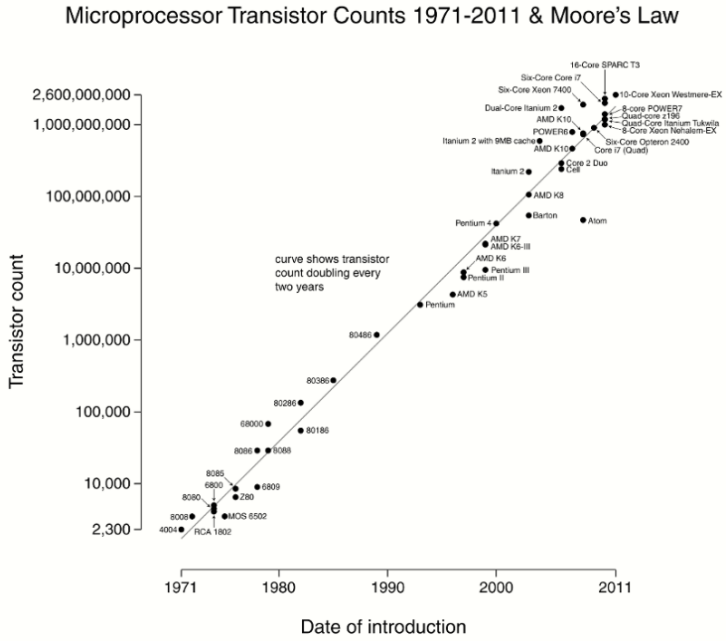

Figure 1.1 Illustration of Moore's law (screenshot).
Source: Adapted from [8].

The most well-known regression models are simple linear regression, multiple linear regression, polynomial regression [8]. In our study, classification models will be be

preferred to solve the problem. Hence, this study will not get into the details of regression models.

1.3 Unsupervised Learning

On the other hand, unsupervised learning with data that have no associated labels. Unlike supervised learning, the goal is not to detect a connection between input and outputs. Instead, we let the model to find hidden patterns or structures that human brain cannot find immediately in the data. In unsupervised learning subjects, there is no right or wrong answer. It is just a application of setting the ML algorithm, running it and observing what patterns and outcomes we get [2].

Thus, prediction is not the primary focus in unsupervised learning. Instead, we are looking to see if there is an informative way to visualize tha data, or can we discover subgroups among the variables or among the observations and more. Unsupervised learning refers to a multiple categories of methods for answering questions such as these [10].

The most common unsupervised learning methods are clustering, dimensionality reduction, density estimation, feature learning, association rule learning, and anomaly detection. Methods such as clustering, have sub-approaches, each with trade offs, as shown in Figure 2 [11]. Since this study primarily focuses on supervised learning, the details of unsupervised learning will not be discussed in detail.

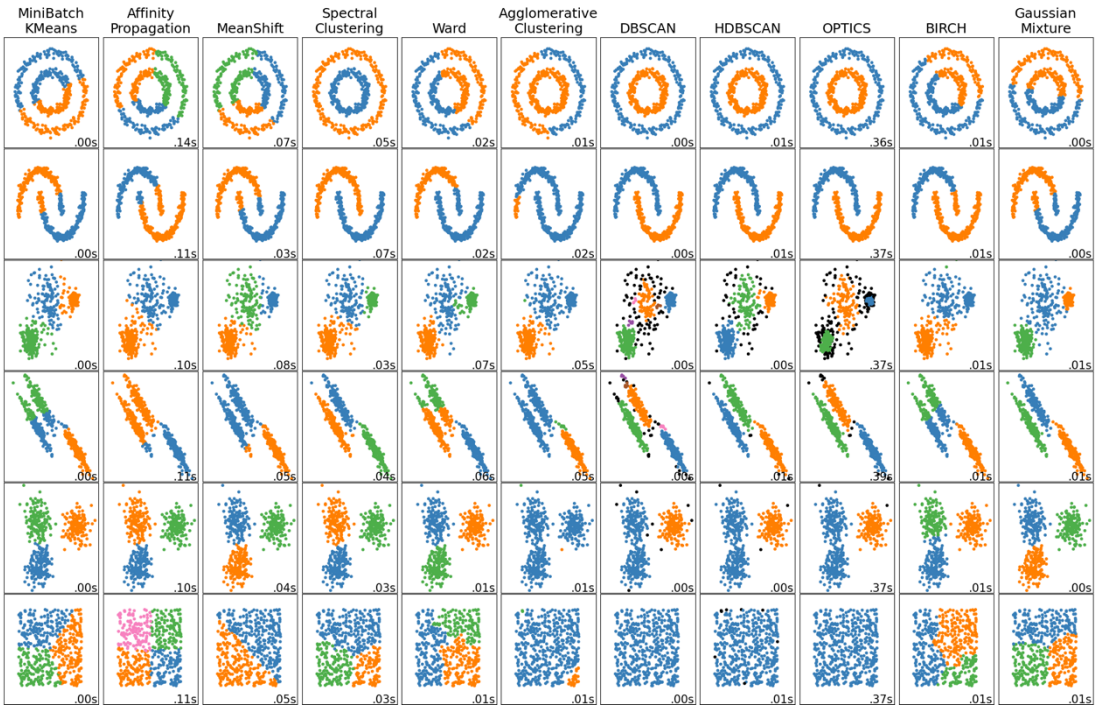

Figure 1.2 Comparison of different clustering algorithms applied to multiple synthetic datasets. Adapted from *Scikit-Learn Cluster Comparison Example*.

1.4 Credit Risk Prediction

Credit risk is a major challenge for banks, since loan defaults can lead to serious financial problems and disrupt the stability of their business flow. To protect both themselves and their customers, banks must follow strict regulations that require thorough risk assessments and transparent reportings. When making lending decisions, banks need to carefully manage risk while also making sure people have fair opportunities to access credit. Modern machine learning tools are now helping banks with this task by analyzing past data to spot patterns and better predict which loans might go unpaid.

1.5 Details Of The Problem

In 21st century's world, some people struggle to get loans due to weak or non-existent credit histories. And, unfortunately, this population is often taken advantage of by untrustworthy lenders [4].

Home Credit aims to increase financial access for the unbanked population by providing good experience. In order to make sure this group of disadvantaged people has a good loan experience, Home Credit applies a selection of alternative data —including even telecom provider and transactional information—to predict their clients' settlement abilities [4].

1.6 Selected Method to Apply

At the initial stage of this study, a detailed exploratory data analysis is done and after careful considerations, logistic regression is chosen as the classification method to be applied. It is a simple and effective baseline model that provides a reference point for later stages.

Logistic regression helps estimate the likelihood that a customer will default by learning the relationship between their characteristics and the final outcome — repayment or default. Because it is easy to interpret and computationally efficient, it allows us to understand which factors most strongly influence credit risk. This clarity is especially useful for financial institutions, which need transparent and explainable models to support their decisions within regulatory requirements.

1.7 What is Logistic Regression?

Logistic regression is a statistical model used to predict the probability of a binary outcome based on one or more independent variables. Its primary purpose in machine learning is to perform binary classification and model the relationship between the independent variables and outcome variable [5].

Logistic regression applies a logistic (sigmoid) function to a linear combination of input values. The sigmoid function is a smooth, differentiable curve defined for all real input values. It has a characteristic S-shape and ensures that predictions remain between 0 and 1. Since it has a non-negative slope and a single inflection point, it is very suitable for modeling outcomes in logistic regression.

Since a sigmoid function transforms the predictions into a value between 0 and 1, it can be interpreted as a probability. The logistic function is shown in Figure 3 [12].

$$f(z) = \frac{1}{1 + e^{-z}}$$

Here, z represents the linear combination of features. The sigmoid curve increases smoothly from 0 to 1, making it a good choice for modeling binary probability scenarios.

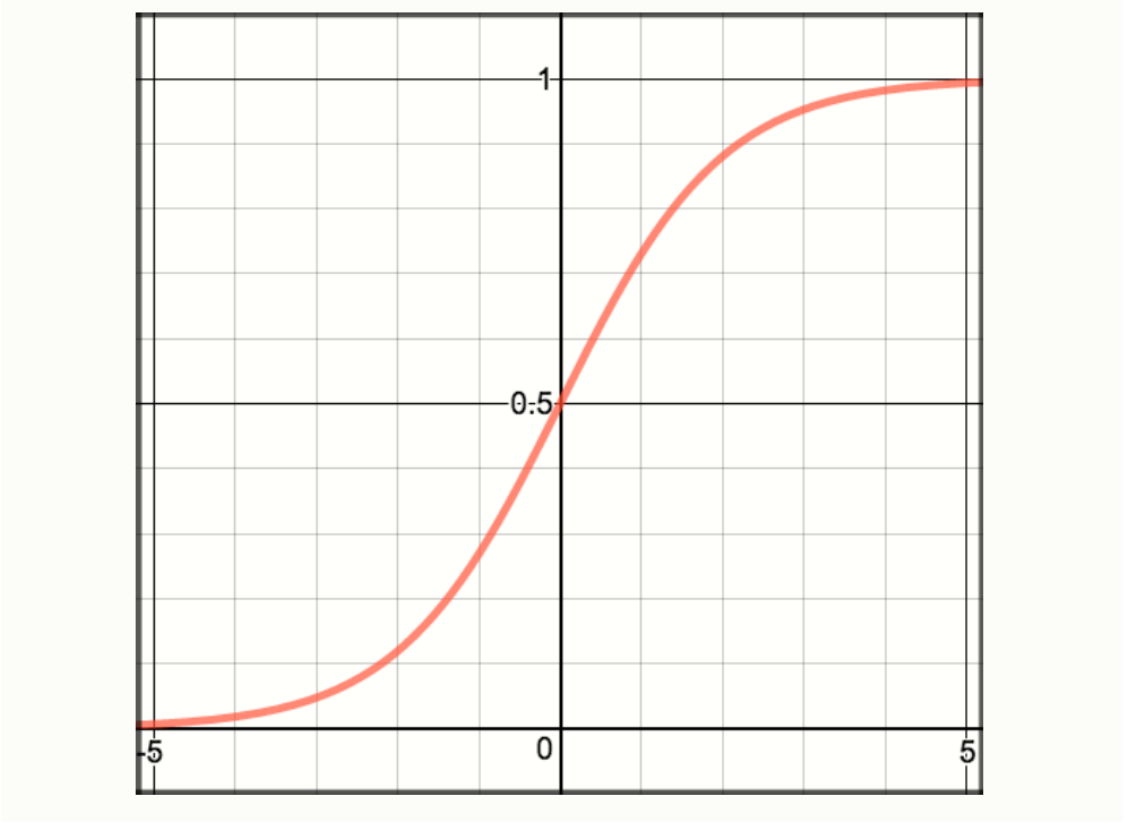

Figure 1.3 Sigmoid (logistic) function curve.
Source: Screenshot from "Logistic Regression" documentation, ML Cheatsheet (2017).

There are 3 logistic regression variants, binary logistic regression, multinomial logistic regression and ordinal logistic regression [5].

1.7.1 Binary Logistic Regression

Binary logistic regression is applied when the outcome variable has only two possible categories. Its purpose is to predict the probability that an observation belongs to one of these categories based on the input features [5].

The probability of the positive class (Y = 1) in binary logistic regression is modeled as:

$$P(Y = 1) = \frac{1}{1 + e^{-z}}$$

where $z = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$

The decision boundary is formed where P(Y = 1) = 0.5, which occurs when z = 0. Observations with z > 0 are classified as positive and observations with z < 0 are classified as negative. A typical sigmoid curve demonstrating class separation is shown in Figure 4 [12].

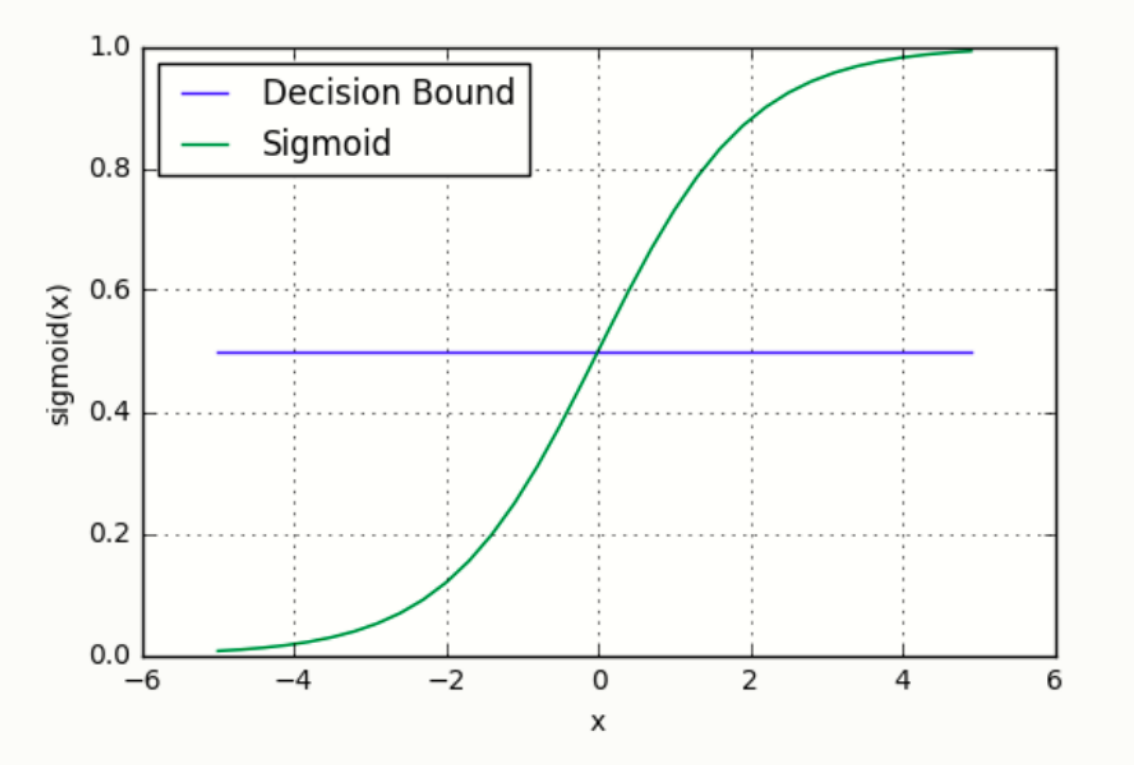

Figure 1.4 Sigmoid function and decision boundary illustrating binary class separation. Source: Screenshot adapted from ML Cheatsheet (2017).

Therefore, binary logistic regression models the log-odds of the positive class as a linear function of the input features.

1.7.2 Multiclass Logistic Regression

When the outcome variable has more than two categories without a specific order, multiclass (multinomial) logistic regression is used. The model calculates the

probabilities of an observation falling into each category relative to a reference category, using the independent variables [5].

The multiclass probabilities are defined using the softmax function:

$$P(Y = i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

where $z_i = \beta_{0,i} + \beta_{1,i}X_1 + \cdots + \beta_{n,i}X_n$

In the equation above, k represents the number of classes and z_i is the linear predictor for class i. The class with the highest probability is selected as the prediction.

1.7.3 Ordinal Logistic Regression

Ordinal logistic regression is suitable when the outcome variable has multiple ordered categories. It allows us to understand how the independent variables affect the chances of an observation being in a higher or lower category on the ordinal scale [5].

$$\text{logit}(P(Y \leq j)) = \theta_j - (\beta_1 X_1 + \cdots + \beta_n X_n), \quad j = 1, 2, \ldots, k-1.$$

Each threshold θ_j represents the boundary between ordered categories. This allows the model to capture the increasing or decreasing likelihood of being in higher-ranked categories based on the input variables.

In practice, ordinal logistic regression assumes that the relationship between each pair of outcome categories is proportional, known as the proportional odds assumption.

1.8 What Is XGBoost?:

The gradient boosting technique can be scaled to large and sparse datasets with the help of XGBoost, an optimized tree-based learning framework. Model updates and split evaluations are more accurate and stable thanks to the algorithm's sequential decision tree construction and incorporation of second-order derivatives of the loss function. Both L1 and L2 regularization terms are included in its objective function, allowing for explicit control over model complexity during training. Furthermore, it uses a weighted quantile sketch method to speed up approximate tree construction on high-dimensional data and a sparsity-aware split-finding strategy. These elements enable XGBoost to outperform traditional gradient boosting techniques in terms of speed, robustness, and overall accuracy [15].

2. INTRODUCTION OF DATASET

The dataset used in this study is from the Kaggle Competition Home Credit Default Risk[14] and has two primary files: application_train.csv and application_test.csv. The train file has the target variable showing whether the client is able to repay the loan. The test file contains the same group of features, without the target value.

2.1 Structure of the Dataset

application_train.csv
- Number of observations: 307,511
- Number of features: 122 (before feature engineering)
- Contains the binary target variable TARGET, where:
  - 0 indicates the client repaid the loan without issues.
  - 1 indicates the client experienced repayment difficulty (default).
- The target variable is highly imbalanced, with only 8.07% of the samples labeled as default.

- application_test.csv
  - Number of observations: 48,744
  - Number of features: 121
  - Does not include the TARGET column and is used for model evaluation or submission purposes.

2.2 Feature Types and General Characteristics

The dataset includes a wide range of client-related information, which can be grouped into the following categories:

- Demographic Information:
  Examples include age, gender, number of family members, and education level. Age is represented using the variable DAYS_BIRTH, which shows the client's age in negative days.

- Financial and Income Information:
  These features describe a client's economic situation and loan capacity, such as total income (AMT_INCOME_TOTAL), credit amount (AMT_CREDIT), annuity amount (AMT_ANNUITY), and price of goods (AMT_GOODS_PRICE).

- External Credit Scores:
  Three variables (EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3) summarize the client's credit worthiness based on scoring systems. In the next chapters, it is concluded that those variables are three of the strongest predictors in the study

- Housing and Property Information:
  Includes building characteristics, apartment measurements, and living conditions. Many of these variables have high number of missing values (mostly above 60%).

- Employment and Registration Information:
  Variables such as DAYS_EMPLOYED, DAYS_REGISTRATION, and DAYS_LAST_PHONE_CHANGE describe employment duration and registration history. Some of these variables contain special sentinel values (e.g., 365243 in DAYS_EMPLOYED), which show missing or anomalous entries.

- Document Flags:
  A set of binary indicators showing if the client submitted certain documents. Most of these flags are uninformative due to extremely low positive counts but a few of them are able to provide meaningful informations.

- Categorical Variables:
  The dataset contains multiple nominal and ordinal categorical features such as occupation type, housing type, income type, and contract type. These variables required encoding operations during the preprocessing stage.

2.3 Data Quality and Challenges

Some of the characteristics of the dataset show complexity and require careful handling during preprocessing and analysis:

1. High Missing Value Rates:
   Many housing related variables had missing ratios above 60-70%. Those variables were not removed entirely. Missing indicator flags were created to preserve possible signal instead.

2. Imbalanced Target Distribution:
   Only a small portion of clients defaulted on their loans. That imbalance influences model performance highly. This problem requires specific techniques such as class weighting or sampling strategies.

3. Sentinel Values and Anomalies:
   Some variables contained non natural placeholder values (e.g., DAYS_EMPLOYED = 365243). Those are needed to be replaced with proper missing values.

4. Mixed Data Types:
   The dataset combines numeric, categorical and binary flag variables. This diversity situation requires very careful feature engineering and encoding decisions.

5.

3. EXPLORATORY DATA ANALYSIS

3.1 Overview of Data Structure

The training dataset has 307,511 rows and 122 features. The test dataset has 48,744 rows and 121 features. The only difference between them is appearance of the TARGET variable.

A large portion of the variables are numerical, accompanied by several categorical features representing demographic, financial, social and housing related attributes. Initial analysis revealed that variation in data distributions, missing patterns and potential anomalies that required careful consideration in data preprocessing step.

3.2 Target Variable Distribution

The TARGET is highly imbalanced. Only 8.07 percent of rows correspond to clients who defaulted on their loans. This imbalance affects highly the model performance, which means it must be considered in training and evaluation steps.

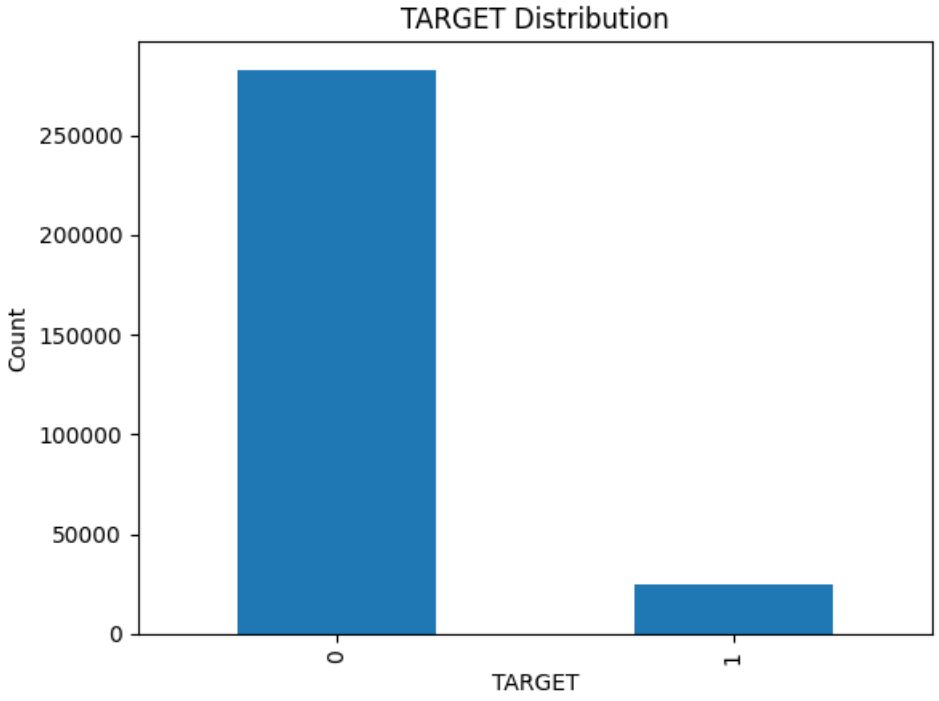

Figure 3.1 Histogram of TARGET distribution

3.3 Analysis of Missing Values

The existence of numerous features with exceptionally high missing rates, particularly those pertaining to housing and property characteristics, is one of the dataset's most important features.

A number of variables have missing ratios greater than 60%, such as COMMONAREA, LIVINGAPARTMENTS, NONLIVINGAPARTMENTS, and YEARS_BUILD groups. These characteristics probably indicate situations where clients have incomplete housing documentation or reside in rental units. During preprocessing, missing-value indicator variables were added rather than completely eliminating these variables.

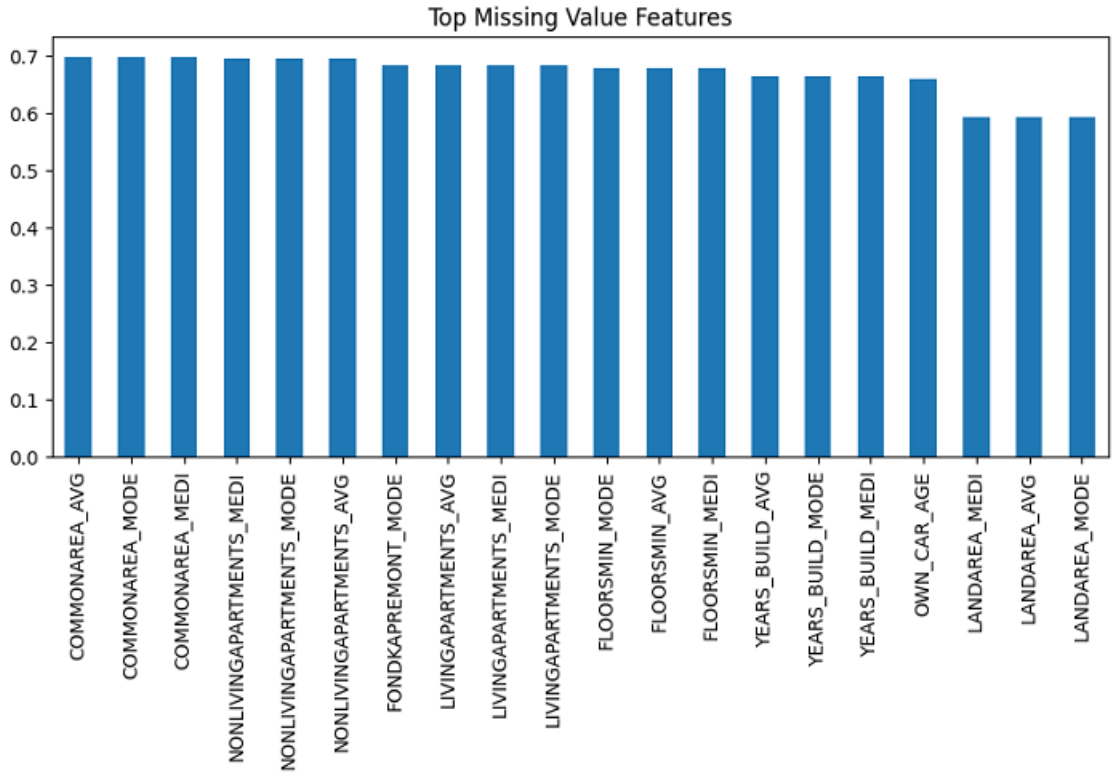

Figure 3.2 Bar chart of top features by missing percentage

Important conclusions about absent patterns include:

1. The missingness pattern of high-missing housing variables may still provide predictive information.

2. Certain features, like DAYS_EMPLOYED = 365243, had significant sentinel values that were replaced with appropriate missing labels.

3. Rare-category analysis is necessary because the majority of document-related variables were binary flags with incredibly low positive rates.

3.4 Numerical Feature Exploration

3.4.1 Age and Employment Duration

The age variable (DAYS_BIRTH) and employment duration variable (DAYS_EMPLOYED) exhibited clear and interpretable patterns after transformation.

- Age is negatively encoded in days. After conversion to years, the average client age is approximately 44.

- Employment duration contained anomalous sentinel values affecting over 50,000 entries. These values were marked and treated as missing.

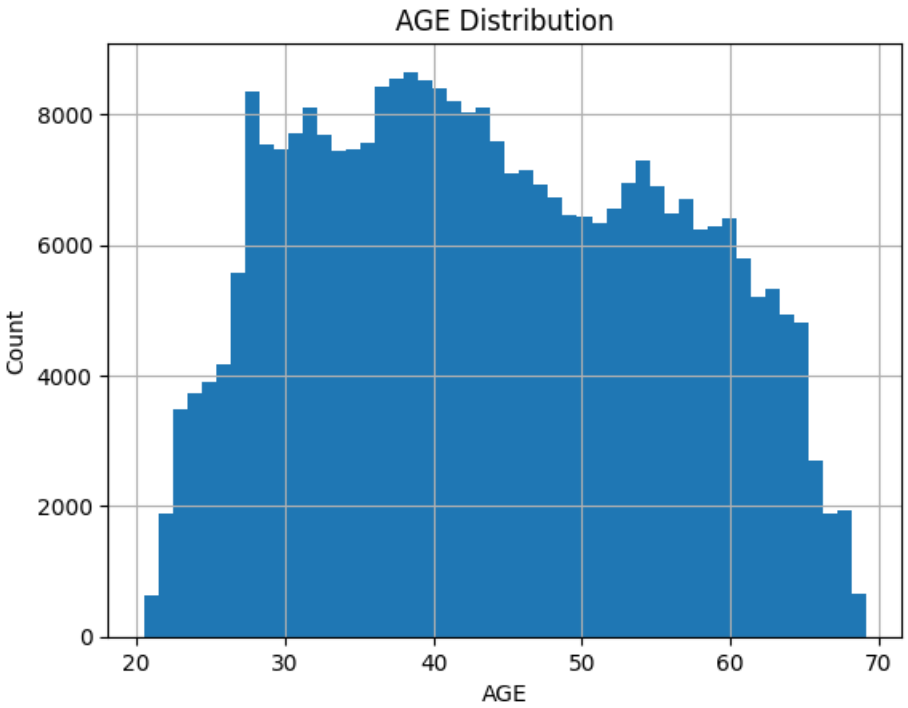

Figure 3.3 Age histogram

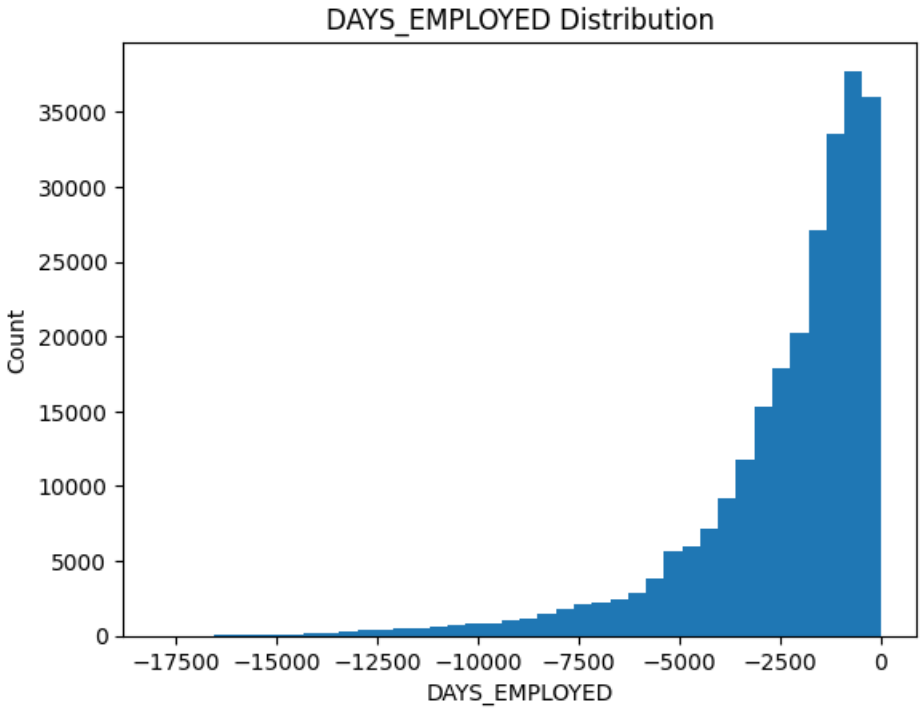

Figure 3.4 Employment duration distribution

Default rates were observed to decrease consistently with increasing age groups.

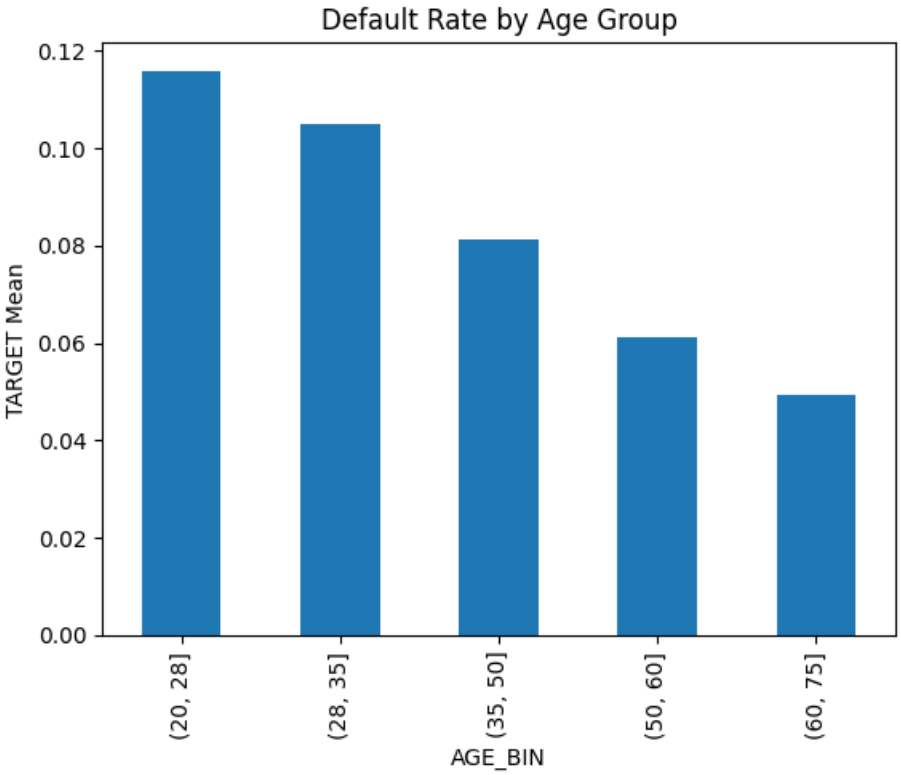

Figure 3.5 Default rate by AGE_BIN

3.4.2 Financial Variables

Important financial features such as income, credit amount, annuity, and goods price displayed right-skewed distributions with several outliers. To better capture relationships, several ratio-based features (e.g., CREDIT_INCOME_RATIO, ANNUITY_INCOME_RATIO, PAYMENT_RATE) were created.

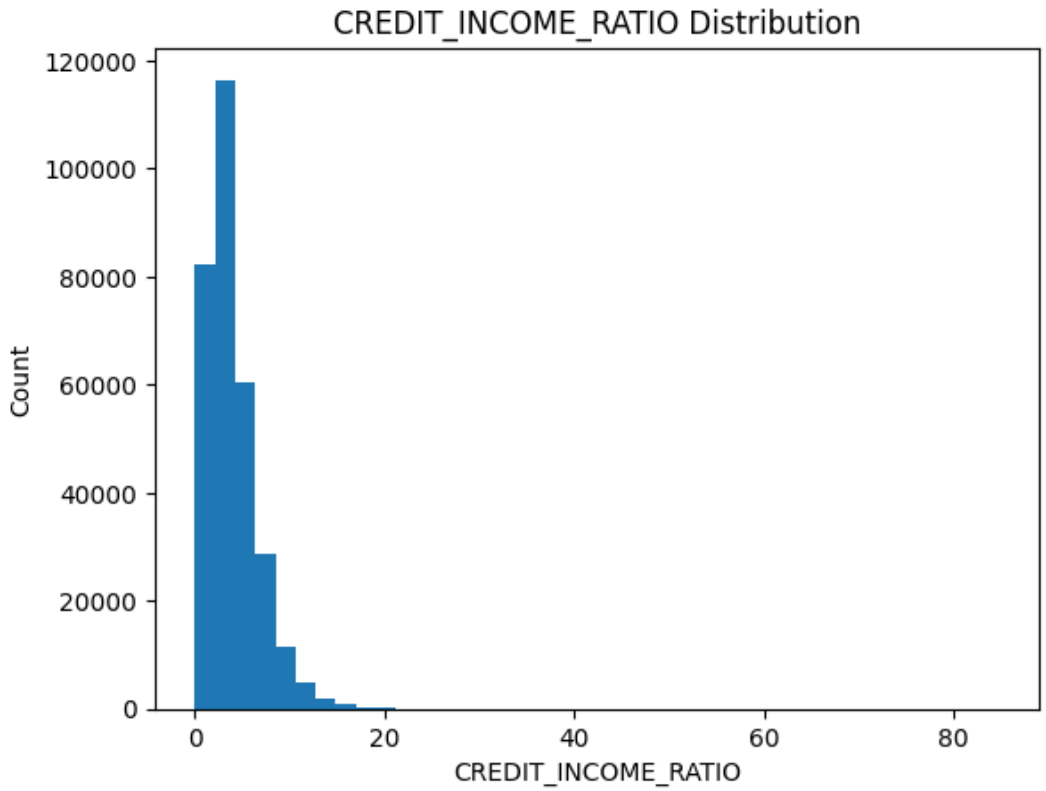

Figure 3.6 Distribution of CREDIT_INCOME_RATIO

These engineered ratios shows relatively stronger correlations with TARGET variable, when it's compared to raw financial amounts.

3.5 Categorical Feature Analysis

A number of categorical factors, including family status, income type, housing type, education level and gender are inspected. Some of the important findings are:

1. The default rate is higher for clients with less education.

2. Applicants with inconsistent work or those who live in rented apartments typically have higher risk levels.

3. Some of the categories have extremely small sample sizes.

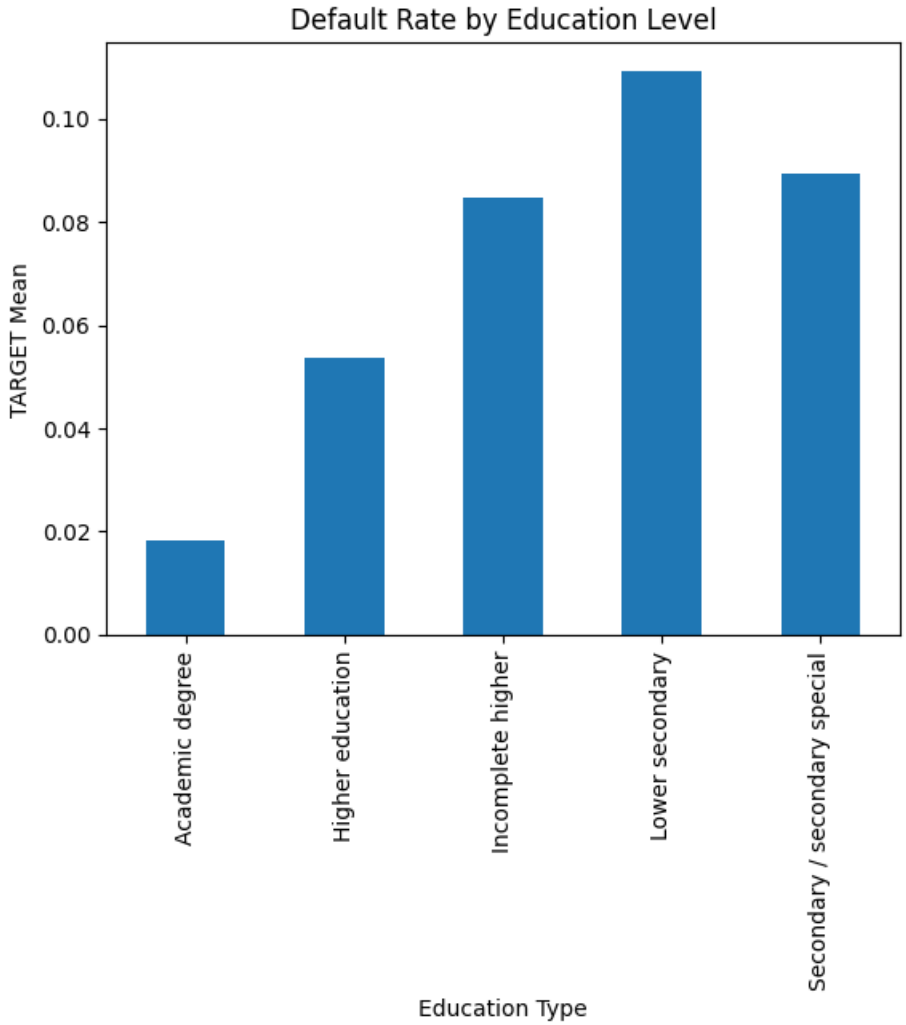

Figure 3.7 Default rate by NAME_EDUCATION_TYPE

With the exception of a few flags exhibiting minor deviations in default tendency, document-related categorical features (FLAG_DOCUMENT_X) were found to be largely uninformative due to extremely low counts.

3.6 Correlation Analysis

The majority of financial variables have only a weak correlation with the TARGET variable, according to correlation analysis. Later on in the modeling process, however, the EXT_SOURCE scores showed the strongest correlations and became the most significant predictors.

- EXT_SOURCE_1: moderate negative correlation
- EXT_SOURCE_2: weak but significant negative correlation
- EXT_SOURCE_3: moderate negative correlation

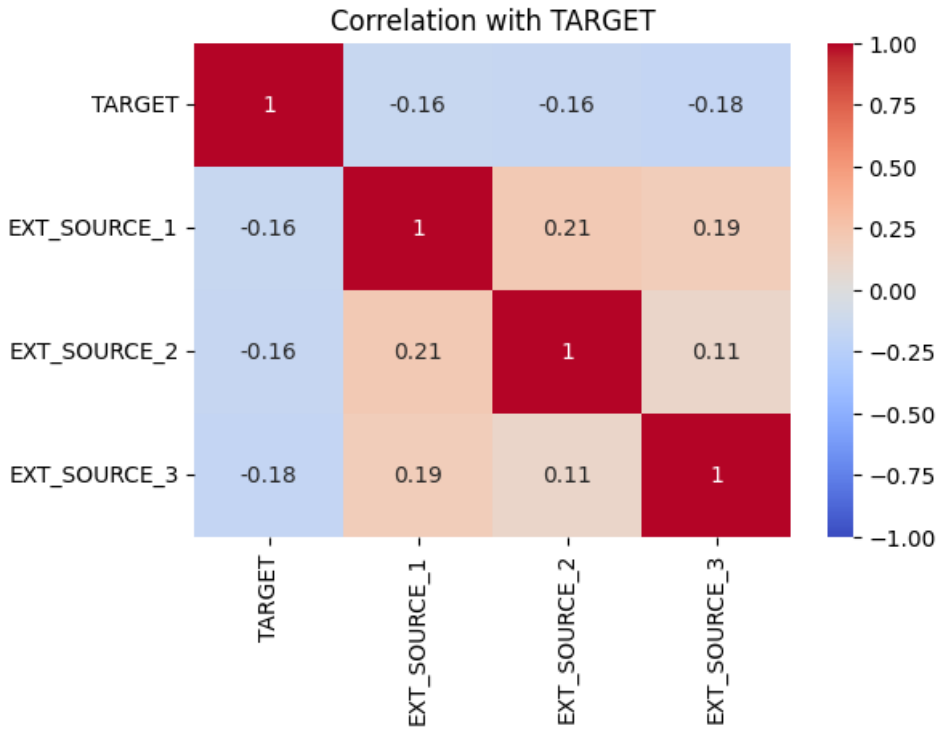

Figure 3.8 Correlation heatmap of key numerical features

These external score variables are consistent with established financial risk modeling literature and indicate creditworthiness effectively.

3.7 Insights from Exploratory Data Analysis

Several crucial insights from the exploratory analysis informed feature engineering and modeling:

1. Because of the dataset's extreme imbalance, modeling requires careful metric selection and threshold adjustments.

2. The use of missing-indicator encoding is justified by the significant missingness patterns found in many housing-related variables.

3. There are definite correlations between default behavior and age, length of employment, and external credit score variables.

4. Compared to raw values, financial ratios are better at capturing customer behavior.

5. Rare-category problems and necessary grouping or one-hot encoding plague a number of categorical features.

6. The significance of anomaly handling is emphasized by the existence of sentinel values.

7. Among the best predictors and essential to model performance are the EXT_SOURCE variables.

8. No single variable strongly predicts default; the model must rely on multivariate patterns.

4. FEATURE ENGINEERING

4.1. Age Feature Construction

The client's age in days is represented by the dataset's DAYS_BIRTH, a negative number. This variable was changed to an age feature that is easier to understand.

- DAYS_BIRTH's absolute value was divided by 365 to determine AGE.
- The results are rounded to the closest integer.
- To ensure accuracy, summary statistics were reviewed.

This change improved interpretability and made it possible to more clearly analyze age-related patterns.

4.2. Employment Anomaly Flag

The dataset provider uses the anomalous placeholder value (365243) in the variable DAYS_EMPLOYED to denote missing data. The actions listed below were carried out:

1. To show that the anomaly was present, a binary flag called EMPLOYED_ANOM was made.
2. Missing values were substituted for every instance of the anomalous value.
3. During the preprocessing phase, employment duration was subsequently imputed.

This procedure prevented statistical measure distortion while preserving valuable information.

4.3. Missing-Value Indicator Features
Extremely high missing rates were found for a number of variables, most of which were connected to housing characteristics. Missingness was encoded as an informative signal rather than eliminating these predictors entirely:

- A list of high-missing columns was found.
- To indicate missing observations, a corresponding _MISS indicator was developed for every column.
- The original columns were later eliminated, but these indicators were kept for modeling.

Because structural missingness is known to have predictive value in credit scoring datasets, this method enables the model to learn from it.

4.4. Removal of Rare Document Flags

The dataset had a series of binary features related to documents (FLAG_DOCUMENT_x). Most of these indicators seemed to appear very few times. In order to reduce sparsity:

- The frequency of each document flag was measured.
- Features with fewer than 500 positive occurrences were removed from both training and test sets.

This step reduces noise and prevents the model from overfitting to extremely rare events.

4.5. Grouping Rare Categories

Several categorical variables contained a long tail of categories with very low frequency. To stabilize model training, rare categories were grouped:

- ORGANIZATION_TYPE
- OCCUPATION_TYPE
- NAME_TYPE_SUITE

For each of these variables, categories occurring fewer than 200 times were replaced with the label “Other”.
This reduces dimensionality and helps one-hot encoding produce balanced columns.

4.6. Creation of Financial Ratio Features

Credit risk prediction often benefits from ratio-based features. Therefore, several domain-informed ratios were engineered:

- CREDIT_INCOME_RATIO: AMT_CREDIT / AMT_INCOME_TOTAL
- ANNUITY_INCOME_RATIO: AMT_ANNUITY / AMT_INCOME_TOTAL
- ANNUITY_CREDIT_RATIO: AMT_ANNUITY / AMT_CREDIT
- CREDIT_TERM: AMT_CREDIT / AMT_ANNUITY (approximate loan term)
- PRICE_TO_INCOME: AMT_GOODS_PRICE / AMT_INCOME_TOTAL

These ratios capture affordability, repayment burden, and credit behavior, which are strong predictors of default.

4.7. Age Binning

To capture non-linear age effects, the AGE feature was discretized into five groups:

- 20–28
- 28–35
- 35–50
- 50–60
- 60+

This transformation allows the model to capture risk differences between demographic segments.

4.8. Winsorization of Outliers

Two ratio features contained extreme upper-tail values. A soft-capping strategy was applied:

- The 99.9th percentile was computed for each variable.
- Values above this threshold were capped to the percentile value.

This reduces the influence of extreme cases without removing observations.

4.9. Removal of Original High-Missing Columns

After generating missingness indicators, the original high-missing variables were removed. Keeping only the _MISS flags preserves information while reducing dimensionality and sparsity.
Additionally, the original time-based variables used to generate AGE and employment features (DAYS_BIRTH, DAYS_EMPLOYED) were removed to avoid redundancy.

4.10. Identification of Numerical and Categorical Variables

The dataset was separated into numerical and categorical feature groups:

- Numerical features: integer and float types (excluding the target).
- Categorical features: string/object variables.

This classification guided the imputation and encoding strategy.

4.11. Missing Value Imputation

Imputation methods were applied consistently to both train and test sets:

- Numerical features → imputed with median values.
- Categorical features → imputed with mode values.

Median and mode imputation maintain distribution stability and avoid introducing artificial relationships.

4.12. One-Hot Encoding of Categorical Features

Categorical features were transformed using one-hot encoding:

- Dummy variables were created with drop_first=True.
- Train and test sets were encoded separately.

Because one-hot encoding can produce mismatched columns, additional alignment was required.

4.13. Alignment of Train and Test Feature Spaces

After encoding, the feature sets differed slightly. To ensure compatibility:

1. Columns present in train but missing in test were added to test as zeros.
2. Columns present in test but missing in train were removed from test.
3. The test set was reordered to match the train column order exactly.

This guarantees that both datasets have identical feature structure for modeling.

4.14. Export of Final Processed Datasets

The final engineered datasets were saved as:

- train_fe.csv
- test_fe.csv

5. BASE MODEL: LOGISTIC REGRESSION

The preprocessed datasets (train_fe.csv and test_fe.csv) were loaded. The training dataset contains 307,511 samples and 217 engineered features. A stratified split was performed to preserve the original class distribution:

The dataset was split into training and validation subsets using a stratified sampling strategy. Specifically, 80% of the data was allocated to the training set (X_train, y_train) and 20% to the validation set (X_val, y_val). Stratification was applied based on the target variable to preserve the original class distribution, and a fixed random seed (random_state = 42) was used to ensure reproducibility of the split.

The target distribution remained consistent between splits:

- Training: 8.07% default (positive class)
- Validation: 8.07% default

The variable AGE_BIN was dropped because it is categorical. All numeric features were standardized.

5.1 Implementation of Logistic Regression

A logistic regression model was initialized with a maximum of 1000 iterations using the *liblinear* solver and a fixed random seed of 42. The model was trained on the scaled training dataset. Predictions and class probability estimates for the positive class were then generated on the scaled validation dataset.

Performance metrics are: ROC-AUC: 0.7507

Classification Report:

Table 5.1 Classification report of the first implementation

Class	Precision	Recall	F1-score
0	0.92	1.00	0.96
1	0.56	0.01	0.03

The baseline model reaches a reasonable ROC-AUC score of **0.7507**, but its ability to detect the minority class is extremely poor:

- Recall for class 1 is **0.01**
- Accuracy is misleading due to class imbalance

- Model is biased toward predicting class 0

This confirms that imbalance-handling methods are required.

5.2 Imbalance Handling Approaches

5.2.1 Class Weight Adjustment

A logistic regression model was initialized with a maximum of 1000 iterations using the *liblinear* solver, **with the class weights automatically adjusted to balance the class distribution**. A fixed random seed of 42 was used for reproducibility. The model was trained on the scaled training dataset.

Performance: ROC-AUC: 0.7505

Classification Report:

Table 5.2 Classification report after Class Weight Adjustment

Class	Precision	Recall	F1-score
0	0.96	0.69	0.81
1	0.16	0.68	0.26

- Recall for class 1 increased from 0.01 → 0.68
- Precision decreased as expected
- ROC-AUC remained stable
- Model shifted toward catching more risky customers

This confirms that class weighting is an effective strategy for this dataset.

5.2.2 Decision Threshold Optimization

Threshold tuning was performed using the precision–recall curve:

Precision, recall, and decision threshold values were computed based on the validation labels and the predicted scores in order to construct the **precision–recall curve**.

A threshold providing recall ≥ 0.60 and the best available precision was selected:

Best Threshold = 0.5501

So the performance at new threshold is as follows:

Table 5.3 Performance at new threshold

Class	Precision	Recall	F1-score
0	0.96	0.76	0.85
1	0.18	0.60	0.28

Confusion Matrix:

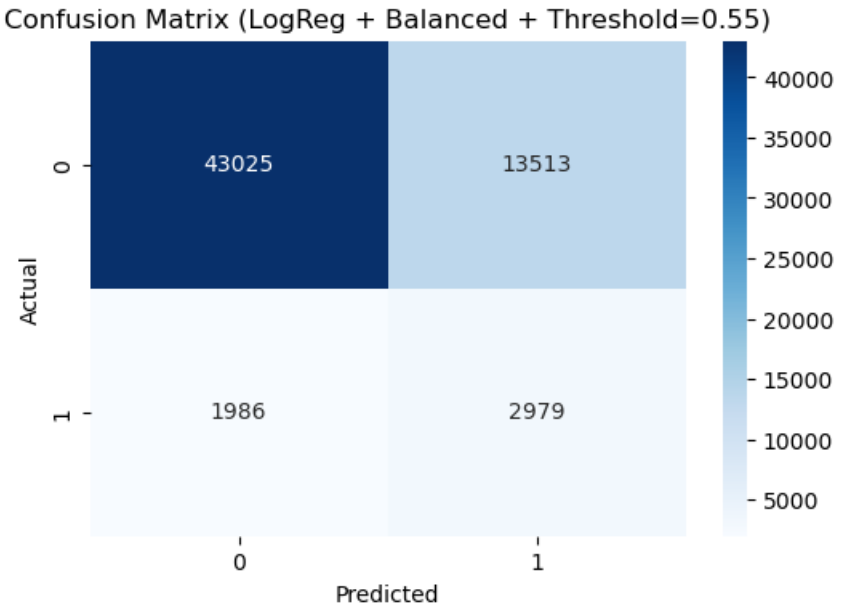

Figure 5.1 Confusion Matrix

- Recall slightly decreased (0.68 → 0.60)
- Precision increased (0.16 → 0.18)
- F1-score for class 1 improved
- Accuracy increased from 0.69 → 0.75

Threshold tuning produced a more balanced and business-appropriate model.

5.2.3 ROC Curve

A ROC curve was generated for model comparison.

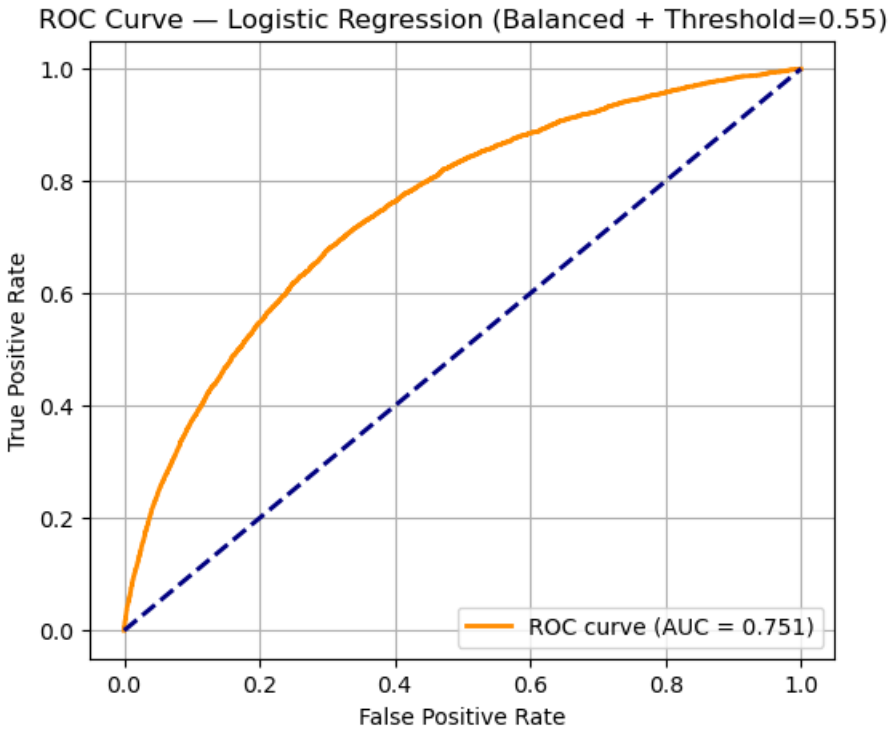

Figure 5.2 ROC Curve for Logistic Regression

ROC-AUC remained stable at 0.75, as ROC is threshold-independent.

5.2.4 SMOTE Oversampling

The SMOTE technique was applied to the training data to address class imbalance. Using a fixed random seed (random_state = 42), the minority class was oversampled to generate a balanced training set by creating synthetic samples based on feature-space similarities. Class ratio is became perfectly balanced (0.50 / 0.50).

After applying of SMOTE, ROC-AUC: 0.7324 is the result. Recall dropped again to 0.02, showing clear performance degradation.

SMOTE negatively impacted Logistic Regression because:

- The method creates synthetic samples based on nearest neighbors
- Logistic Regression is sensitive to linear boundary distortion
- The resulting noise reduces generalization

As a result, SMOTE is not suitable in this context.

5.3 Interpretation of the Base Model

Class Imbalance Remains the Core Challenge:

- Baseline model heavily favored the majority class.
- Class weighting was significantly more effective than SMOTE.

Class Weighting Provided the Best Trade-Off:

- Recall for class 1 improved drastically.
- ROC-AUC remained stable.
- Model became suitable for “risk-averse” financial screening.

Threshold Adjustment Improved Practical Usability:

- Better alignment with business requirements.
- Fewer false negatives without excessive false positives.

ROC Curve Confirms Moderate Discriminative Power:

- AUC ≈ 0.75 indicates useful but improvable performance.

SMOTE Was Not Effective Here:

- Degraded validation performance.
- Generated noise harmful for linear classifiers.

5.4 Model Evaluation for Logistic Regression

Logistic Regression is validated as a solid baseline model for this project:

- Fast training time
- High interpretability
- Stable performance after balancing and threshold tuning

However, due to the dataset’s non-linear structure and large number of engineered features, more complex models are expected to perform better.

6. FINAL MODEL: XGBOOST

6.1 What Is XGBoost?

The gradient boosting technique can be scaled to large and sparse datasets with the help of XGBoost, an optimized tree-based learning framework. Model updates and split evaluations are more accurate and stable thanks to the algorithm's sequential decision tree construction and incorporation of second-order derivatives of the loss function. Both L1 and L2 regularization terms are included in its objective function, allowing for explicit control over model complexity during training. Furthermore, it uses a weighted quantile sketch method to speed up approximate tree construction on high-dimensional data and a sparsity-aware split-finding strategy. These elements enable XGBoost to outperform traditional gradient boosting techniques in terms of speed, robustness, and overall accuracy [15].

6.2 Implementation of XGBoost

A first XGBoost model was trained using moderately tuned parameters inspired by prior industry examples and Kaggle discussions. The desired objective was to establish a strong benchmark before applying additional optimization.

An XGBoost classifier was initialized with **200 estimators**, a **maximum tree depth of 4**, and a **learning rate of 0.05**. Subsampling and column sampling ratios were both set to **0.8** to introduce randomness and reduce overfitting. The model was trained using **logarithmic loss as the evaluation metric**, with a fixed random seed of **42** for reproducibility.

Validation Performance:

- ROC-AUC: 0.7630
- Recall for minority class remained very low (≈ 0.01)

This indicated that while XGBoost had improved overall separability compared to logistic regression, the model was still insufficient in identifying defaulting users due to the class imbalance.

6.3 Improved XGBoost Model with Class Weight Adjustment

To address the imbalance, scale_pos_weight was introduced. This parameter forces the model to pay more attention to the minority class by increasing the cost of misclassifying positive samples.

An XGBoost classifier was initialized with **300 estimators** and a **maximum tree depth of 5**, while the learning rate was kept at **0.05**. Subsampling and column sampling ratios were both set to **0.8**. To address class imbalance, **the scale_pos_weight parameter was**

set based on the ratio of negative to positive samples in the training data. The model was trained using **logarithmic loss as the evaluation metric**, with a fixed random seed of **42** for reproducibility.

Validation Performance:

- ROC-AUC: 0.7677
- Minority class recall increased dramatically to 0.68
- Majority class recall decreased—expected trade-off

This model produced the best balance between AUC and sensitivity to high-risk customers.

6.4 Final XGBoost Model and Hyperparameter Refinement

A final refined model was trained using additional stability-focused hyperparameters such as min_child_weight, gamma, and regularization terms.

An XGBoost classifier was configured with **350 estimators** and a **learning rate of 0.05**, using a **maximum tree depth of 4**. The model employed **subsample = 0.9** and **colsample_bytree = 0.8** to improve generalization. Additional regularization and complexity control were introduced through **min_child_weight = 10**, **L2 regularization (reg_lambda = 1.0)**, and **L1 regularization disabled (reg_alpha = 0.0)**, while **gamma was set to 0.0**.

To handle class imbalance, **scale_pos_weight was fixed at 7**. The model was trained using **logarithmic loss as the evaluation metric**, with the **hist tree method** selected for computational efficiency and **parallel processing enabled (n_jobs = −1)**. A fixed random seed of **42** was used to ensure reproducibility.

Results (Default Threshold = 0.50):

- ROC-AUC: 0.7678
- Precision (class 1): 0.23
- Recall (class 1): 0.51
- Accuracy: 0.82

This model stands as the best-performing model in the project.

6.5 Decision Threshold Optimization

Because XGBoost, like most classifiers, outputs probabilities rather than direct labels, the final performance depends significantly on the chosen classification threshold.

To find an optimal threshold that maximizes both sensitivity and specificity, Youden's J statistic (tpr-fpr) was applied.

The false positive rates, true positive rates, and corresponding decision thresholds were computed on the validation set to construct the ROC curve. **The Youden's J statistic was calculated as the difference between the true positive rate and the false positive rate**, and **the optimal decision threshold was selected as the threshold that maximizes this statistic**.

Optimal Threshold Identified: 0.3618

This new threshold improved the model's ability to detect risky customers.

6.6 Performance with Optimized Threshold

Applying the custom threshold produced a more balanced, risk-sensitive model.
Validation Results (Threshold = 0.3618):

- ROC-AUC: 0.7678 (unchanged, threshold-independent)
- Recall (class 1): 0.72
- Precision (class 1): 0.17
- Accuracy: 0.68
- Confusion Matrix:

Output is as follows:

The confusion matrix obtained on the validation set indicates **38,452 true negatives**, **18,086 false positives**, **1,386 false negatives**, and **3,579 true positives**.

This performance indicates that the model successfully captures the majority of high-risk customers (TARGET=1), at the cost of generating more false alarms.
Given the domain (credit risk prediction), this trade-off is acceptable and even desirable.

6.7 Feature Importance Analysis

To interpret the model's behavior, the top 20 features were extracted from the final XGBoost model.

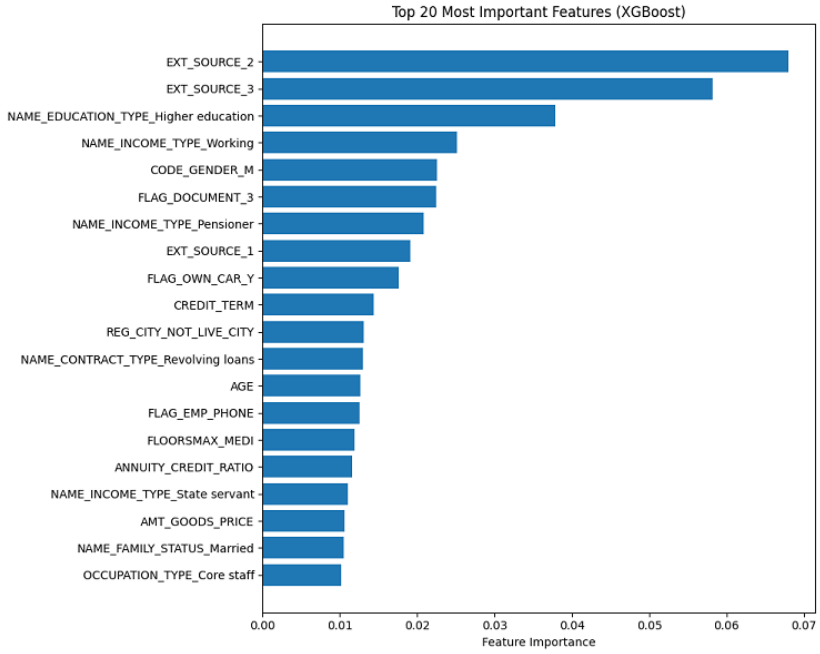

Figure 5.3 Top 20 Most Important Features (XGBoost)

Key influential features included:

- EXT_SOURCE_2, EXT_SOURCE_3, EXT_SOURCE_1
- Education level
- Income type
- CREDIT_TERM
- FLAG_DOCUMENT_3
- AGE
- ANNUITY_CREDIT_RATIO

These results align with common findings in credit scoring literature, confirming the importance of external credit bureau scores and income-related variables.

6.8 Model Evaluation for XGBoost

The XGBoost model demonstrated substantial improvements over the baseline logistic regression model:

1. Higher ROC-AUC (≈ 0.768)
2. More effective handling of class imbalance via scale_pos_weight

3. Successful threshold tuning with strong recall performance (≈ 0.72)

4. Enhanced interpretability through feature importance analysis

5. Balanced trade-off aligned with real-world credit risk requirements

This model is considered the final model for the project and provides a strong performance benchmark for future model extensions.

Moreover, XGBoost with class weighting and threshold optimization, achieves the project objective of identifying clients with a higher probability of credit default. It provides a reliable, interpretable, and domain-appropriate solution for the credit scoring task.

8. CONCLUSION

This study successfully builds and evaluates credit default prediction models on the Home Credit dataset, progressing from an interpretable baseline (logistic regression) to a stronger non-linear classifier (XGBoost). The core challenge throughout the project is the **strong class imbalance** (only 8.07% defaults), which makes accuracy misleading and forces the modeling process to prioritize imbalance-aware evaluation and threshold decisions. The baseline logistic regression reached **ROC-AUC ≈ 0.7507**, but initially failed to detect defaults (minority recall ≈ 0.01), indicating that naive training under default settings is not appropriate for the credit risk setting. Among the imbalance strategies tested for the baseline, **class weighting** was the most effective, substantially improving minority recall while keeping ROC-AUC essentially stable, whereas **SMOTE oversampling** reduced ROC-AUC and degraded recall, suggesting that synthetic sampling introduced boundary distortions or noise that harmed generalization in this configuration.

The final XGBoost model improves discrimination to **ROC-AUC ≈ 0.7678**, representing a measurable gain over logistic regression and confirming the benefit of tree-boosting for capturing non-linear relationships in engineered credit features. However, these ROC-AUC values also indicate that performance is **useful but not exceptional**. In particular, the project's best AUC remains below what is typically achieved by top-performing solutions on this competition, and the report's own results show that improvements in "real-world usefulness" came mainly from **operating-threshold selection** (e.g., Youden's J threshold ≈ 0.3618 increasing minority recall to ≈ 0.72) rather than from a dramatic gain in separability. This outcome is consistent with the fact that, under severe imbalance, models can maintain similar AUC while their recall/precision trade-off shifts substantially when the threshold changes.

A principal limitation that explains why ROC-AUC and confusion matrices did not become "excellent" is **data coverage**: the report explicitly states that the project uses only the **two primary CSV files** (application_train.csv and application_test.csv). In the Kaggle Home Credit setting, stronger performance typically depends on incorporating additional relational tables (e.g., bureau history, previous applications, installment/payment behavior) and aggregating them into applicant-level features. By restricting the feature space to application-level information, the model cannot fully leverage historical credit and behavioral signals that often provide the largest incremental predictive power. In other words, the models here are learning from a meaningful but incomplete view of the customer, which naturally caps achievable discrimination.

Beyond dataset coverage, the results also reflect inherent trade-offs between interpretability, robustness, and sensitivity to defaults. The project's feature importance analysis indicates that **EXT_SOURCE variables** dominate predictive influence, along with interpretable socio-economic factors (education, income type) and affordability-

related ratios. This alignment with credit-scoring intuition is a strength, but it also suggests that much of the available signal is concentrated in a limited subset of features, making large AUC gains difficult without additional external/historical data sources.

In conclusion, the project provides a coherent, academically structured ML pipeline and reaches a **moderate but meaningful** performance level, with XGBoost plus class weighting and threshold optimization offering the best domain-aligned operating behavior. The most important explanation for not obtaining "outstanding" ROC-AUC is the intentional simplification of the data input to only two core CSV files, which limits access to richer behavioral and credit-history signals. Despite this limitation, the study demonstrates correct handling of missingness and anomalies, justifies feature engineering decisions, and clearly shows how imbalance strategies (class weighting and thresholding) materially change the practical usefulness of the model in a credit risk screening context.

8. REFERENCES

[1] TÜBİTAK. (n.d.). *Artificial Intelligence and Machine Learning*. TÜBİTAK Encyclopedia. Retrieved October 23, 2025, from https://ansiklopedi.tubitak.gov.tr/ansiklopedi/yapay_zeka_ve_makine_ogrenmesi

[2] Bell, J. (2022). What is machine learning?. Machine learning and the city: applications in architecture and urban design, 207-216.

[3] Soofi, A. A., & Awan, A. (2017). Classification techniques in machine learning: applications and issues. Journal of Basic & Applied Sciences, 13, 459-465.

[4] Kaggle. (n.d.). *Home Credit Default Risk Competition*. Retrieved October 23, 2025, from https://www.kaggle.com/competitions/home-credit-default-risk/overview

[5] Buhl, N. (2023, November 27). *Logistic regression: Definition, use cases, implementation*. Encord. Retrieved October 23, 2025, from https://encord.com/blog/what-is-logistic-regression/

[6] Baloglu, O., Latifi, S. Q., & Nazha, A. (2022). What is machine learning?. *Archives of Disease in Childhood-Education and Practice*, *107*(5), 386-388.

[7] Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In Machine learning techniques for multimedia: case studies on organization and retrieval (pp. 21-49). Berlin, Heidelberg: Springer Berlin Heidelberg.

[8] Alnuaimi, A. F., & Albaldawi, T. H. (2024). An overview of machine learning classification techniques. In *BIO Web of Conferences* (Vol. 97, p. 00133). EDP Sciences.

[9] Saleh, H., & Layous, J. (2022). Machine Learning-Regression. Suriye Üniversitesi, Yüksek Uygulamalı Bilimler ve Teknoloji Enstitüsü, Elektronik ve Mekanik Sistemler Anabilim Dalı, 23.

[10] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Unsupervised learning. In *An introduction to statistical learning: with applications in Python* (pp. 503-556). Cham: Springer International Publishing.

[11] Scikit-Learn Developers. (2025). Comparison of clustering algorithms on toy datasets. In Scikit-Learn Documentation (Version 1.7). Retrieved from https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

[12] ML Cheatsheet. (2017). Logistic Regression. Retrieved from https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

[13] Kaggle. (2018). Home Credit Default Risk [Dataset]. https://www.kaggle.com/competitions/home-credit-default-risk

[14] Home Credit Group, & Kaggle. (2018). *Home Credit Default Risk* [Data set]. Kaggle. https://www.kaggle.com/c/home-credit-default-risk

[15] Chen, T. (2016). XGBoost: A Scalable Tree Boosting System. *Cornell University*.

9. APPENDIX

Appendix A. RESUME

Name Surname	Serhet Gökdemir
Birth Date	19.11.2001
Birth Place	Diyarbakır
High School	2016 – 2020 Diyarbakır Anadolu Lisesi
Internships	Turkcell Teknoloji Araştırma ve Geliştirme A.Ş– Istanbul/Hybrid
	Ziraat Teknoloji A.Ş. – Istanbul – (Ongoing)