QFRM: Default Prediction Model

Stavros Ieronymakis - 2645715 - s.ieronymakis2@student.vu.nl

May 2024

1 Introduction

The goal of this project is to develop a robust machine learning model to predict loan default risk using data from the Home Credit Default Risk dataset. The dataset contains extensive information on loan applicants, including demographics, financial status, and previous credit history.

The project begins with data loading and initial setup, configuring display options for pandas DataFrames to facilitate data exploration and analysis. Exploratory Data Analysis (EDA) is performed to understand the characteristics of the dataset, including the distribution of the target variable and an overview of continuous, discrete, and categorical features. This step also involves analyzing missing values.

Customized description and plot functions are developed to provide detailed summaries and visualizations of the features. These functions include descriptions of continuous, discrete, and categorical features, covering aspects such as missing values, unique value counts, and correlations with the target variable. Customized plots, including histograms, box plots, and KDE plots, are used to visualize feature distributions for different target values.

Several custom preprocessing steps are defined to handle missing values, merge rare categories, and encode categorical features. These steps ensure that the data is clean and well-prepared for modeling. The preprocessing includes creating binary 'missing flag' features, filling missing values, merging rare categories, and applying one-hot encoding and standardization.

A customized evaluation method, the Declic Evaluation, is developed to analyze model performance across different risk segments. This method divides the data into deciles based on predicted probabilities and evaluates the default rate in each decile, providing a detailed understanding of the model's effectiveness.

Two machine learning models are built and evaluated—Logistic Regression and LightGBM. For each model, a preprocessing pipeline is defined, and the models are trained using GridSearchCV for hyperparameter tuning. The performance of the models is evaluated using ROC-AUC scores and Declic Evaluation, and the feature importances are analyzed. Finally, SHAP values are calculated for the LightGBM model.

This project demonstrates a comprehensive approach to developing a predictive model for loan default risk, from data exploration and preprocessing to model training and evaluation. The use of customized functions and evaluation methods ensures that the model is well-tailored to the specific characteristics and requirements of the dataset.

2 Data

The dataset used is the Home Credit Group - Featured Prediction Competition, titled "Home Credit Default Risk" and can be found under *kaggle* competitions of 2018. The reader should be aware that this is an official online competition dataset, with total prize money of $70,000. Therefore, the complexity and significance of this dataset are presumed to be exceptionally high. A detailed description, the method it is composed, and schematics of the dataset can be found in the Appendix.

The dataset consists of two main files: `application_train.csv` for training and `application_test.csv` for testing. The training set contains 307,511 rows and 121 columns, while the test set comprises 48,744 rows and 120 columns. The target variable, `TARGET`, indicates whether a loan applicant has defaulted on a loan (1) or not (0). This binary classification problem aims to predict the default risk based on various features provided in the dataset.

2.1 Exploratory Data Analysis (EDA)

EDA is performed to understand the distribution and characteristics of the data. This involves examining both the target variable and the predictor variables. In the following section, we explain briefly the steps taken to clean our dataset. A different file will be submitted that provides a detailed explanation of the analysis. We do this to keep the present report within reasonable length and not confuse the reader.

2.1.1 Univariate Analysis

Univariate analysis focuses on the individual features:
- **Continuous Features:** Features such as `DAYS_BIRTH`, `AMT_CREDIT`, and `EXT_SOURCE_1` are analyzed for their statistical properties. Histograms, box plots, and KDE plots are used to visualize their distributions.

- **Discrete Features:** Features like CNT_CHILDREN and DAYS_EMPLOYED are examined through count plots and bar plots to understand their frequency distributions and potential outliers.
- **Categorical Features:** Features such as NAME_CONTRACT_TYPE, CODE_GENDER, and NAME_INCOME_TYPE are analyzed using count plots and bar plots to understand the distribution of categories and their relationship with the target variable.

2.1.2 Bivariate Analysis

Bivariate analysis explores the relationships between pairs of variables, especially between the features and the target variable:
- **Correlation Analysis:** Correlation matrices and heatmaps are used to identify relationships between numerical features. Highly correlated features are noted for potential issues with multicollinearity.
- **Target Relationship:** The relationship between each feature and the target variable is examined. For continuous features, scatter plots and box plots are used, while for categorical features, bar plots showing default rates across different categories are employed.

2.2 Handling Missing Values

A significant portion of the dataset contains missing values. The following steps are taken to address them:
- **Flagging Missing Values:** A binary flag is created for features with missing values to indicate the presence of missing data.
- **Imputation:** Missing values are imputed using the median for continuous features like EXT_SOURCE_1, EXT_SOURCE_2, and EXT_SOURCE_3. For categorical features, missing values are replaced with a new category, 'MISSING'.

2.3 Feature Engineering

To enhance the predictive power of the model, several feature engineering steps are applied:
- **Merging Rare Categories:** Rare categories in categorical features are merged to reduce the number of categories and prevent overfitting.
- **Dropping Irrelevant Features:** Features with high percentages of missing values, low variance, or high correlation with other features are dropped. For instance, several FLAG_DOCUMENT features are dropped due to low variance.

2.4 Data Splitting

The data is split into training and validation sets using stratified sampling to maintain the distribution of the target variable. The training set consists of 80% of the original data, while the validation set consists of the remaining 20%. This ensures that both sets have a similar proportion of defaulted and non-defaulted loans, providing a more reliable evaluation of model performance.

In order to process such a large dataset, features are split into Feature Groups based on their description. Below is a list with all lists and the corresponding category.

Feature Group	**Features**
Demographics	NAME_CONTRACT_TYPE, CODE_GENDER, FLAG_OWN_CAR, FLAG_OWN_REALTY, NAME_TYPE_SUITE, NAME_INCOME_TYPE, NAME_EDUCATION_TYPE, NAME_FAMILY $_{S}TATUS, NAME\_HOUSING\_TYPE, OCCUPATION\_TYPE, ORGANIZATION\_TYPE$
Family Count	CNT_CHILDREN, CNT_FAM_MEMBERS
Age / Duration	DAYS_BIRTH, DAYS_EMPLOYED, DAYS_REGISTRATION, DAYS_ID_PUBLISH, DAYS_LAST_PHONE_CHANGE, OWN_CAR_AGE
Social Circle	OBS_30_CNT_SOCIAL_CIRCLE, DEF_30_CNT_SOCIAL_CIRCLE, OBS_60_CNT_SOCIAL_CIRCLE, DEF_60_CNT_SOCIAL_CIRCLE
Contact Info	FLAG_MOBIL, FLAG_EMP_PHONE, FLAG_WORK_PHONE, FLAG_CONT_MOBILE, FLAG_PHONE, FLAG_EMAIL
Address Discrepancy	REG_REGION_NOT_LIVE_REGION, REG_REGION_NOT_WORK_REGION, LIVE_REGION_NOT_WORK_REGION, REG_CITY_NOT_LIVE_CITY, REG_CITY_NOT_WORK_CITY, LIVE_CITY_NOT_WORK_CITY
Region's Data	REGION_POPULATION_RELATIVE, REGION_RATING_CLIENT, REGION_RATING_CLIENT_W_CITY
Process Start Time	HOUR_APPR_PROCESS_START, WEEKDAY_APPR_PROCESS_START
External Source Scores	EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3
Amounts	AMT_INCOME_TOTAL, AMT_CREDIT, AMT_ANNUITY, AMT_GOODS_PRICE
Recent Inquiries	AMT_REQ_CREDIT_BUREAU_HOUR, AMT_REQ_CREDIT_BUREAU_DAY, AMT_REQ_CREDIT_BUREAU_WEEK, AMT_REQ_CREDIT_BUREAU_MON, AMT_REQ_CREDIT_BUREAU_QRT, AMT_REQ_CREDIT_BUREAU_YEAR
Provided Documents	FLAG_DOCUMENT_2, FLAG_DOCUMENT_3, FLAG_DOCUMENT_4, FLAG_DOCUMENT_5, FLAG_DOCUMENT_6, FLAG_DOCUMENT_7, FLAG_DOCUMENT_8, FLAG_DOCUMENT_9, FLAG_DOCUMENT_10, FLAG_DOCUMENT_11, FLAG_DOCUMENT_12, FLAG_DOCUMENT_13, FLAG_DOCUMENT_14, FLAG_DOCUMENT_15, FLAG_DOCUMENT_16, FLAG_DOCUMENT_17, FLAG_DOCUMENT_18, FLAG_DOCUMENT_19, FLAG_DOCUMENT_20, FLAG_DOCUMENT_21

Building	APARTMENTS_AVG, BASEMENTAREA_AVG, YEARS_BEGINEXPLUATATION_AVG, YEARS_BUILD_AVG, COMMONAREA_AVG, ELEVATORS_AVG, ENTRANCES_AVG, FLOORSMAX_AVG, FLOORSMIN_AVG, LANDAREA_AVG, LIVINGAPARTMENTS_AVG, LIVINGAREA_AVG, NONLIVINGAPARTMENTS_AVG, NONLIVINGAREA_AVG, APARTMENTS_MODE, BASEMENTAREA_MODE, YEARS_BEGINEXPLUATATION_MODE, YEARS_BUILD_MODE, COMMONAREA_MODE, ELEVATORS_MODE, ENTRANCES_MODE, FLOORSMAX_MODE, FLOORSMIN_MODE, LANDAREA_MODE, LIVINGAPARTMENTS_MODE, LIVINGAREA_MODE, NONLIVINGAPARTMENTS_MODE, NONLIVINGAREA_MODE, APARTMENTS_MEDI, BASEMENTAREA_MEDI, YEARS_BEGINEXPLUATATION_MEDI, YEARS_BUILD_MEDI, COMMONAREA_MEDI, ELEVATORS_MEDI, ENTRANCES_MEDI, FLOORSMAX_MEDI, FLOORSMIN_MEDI, LANDAREA_MEDI, LIVINGAPARTMENTS_MEDI, LIVINGAREA_MEDI, NONLIVINGAPARTMENTS_MEDI, NONLIVINGAREA_MEDI, FONDKAPREMONT_MODE, HOUSETYPE_MODE, TOTALAREA_MODE, WALLSMATERIAL_MODE, EMERGENCYSTATE_MODE

Early on the process we identify some features to be dropped. This is because they either exhibit a high correlation with another feature that in turn exhibit a better correlation with the target feature, or their values are virtually constant (low variance <1%).

Reason for Dropping	**Features**
Virtually Constant Values	FLAG_MOBIL, FLAG_CONT_MOBILE, AMT_REQ_CREDIT_BUREAU_HOUR, AMT_REQ_CREDIT_BUREAU_DAY, FLAG_DOCUMENT_2, FLAG_DOCUMENT_4, FLAG_DOCUMENT_7, FLAG_DOCUMENT_10, FLAG_DOCUMENT_12, FLAG_DOCUMENT_17, FLAG_DOCUMENT_19, FLAG_DOCUMENT_20, FLAG_DOCUMENT_21
Due to High Correlation	LIVE_REGION_NOT_WORK_REGION, REG_CITY_NOT_LIVE_CITY, REGION_RATING_CLIENT_W_CITY, CNT_FAM_MEMBERS, DEF_60_CNT_SOCIAL_CIRCLE, AMT_GOODS_PRICE

Table 2: List of Dropped Features

This results in the following number of features per group to be retained for further analysis. For more details on the EDA process, the reader should refer to the compliment file submitted with this report.

Feature Group	**Number of Retained Features**
Demographics	11
Family Count	1
Age / Duration	6
Social Circle	3
Contact Info	4
Address Discrepancy	4
Region's Data	2
Process Start Time	2
External Source Scores	3
Amounts	3
Recent Inquiries	4
Provided Documents	11

Table 3: Summary of Retained Features by Group

2.5 Pipeline Creation

The following step is to create a preprocessing pipeline. This pipeline includes several transformations to prepare the data for modeling. The transformations handle missing values, merge categories, select specific columns, perform one-hot encoding, and standardize the data based on our uni-/bi-variate analysis.

Handling Missing Values and Creating Missing Flags: For some features, missing values are replaced with specific values, and new binary features (missing flags) are created to indicate whether the original value was missing. Mathematically, if x_i is the original feature value and flag_i is the missing flag:

$$\text{flag}_i = \begin{cases} 1 & \text{if } x_i \text{ is missing} \\ 0 & \text{otherwise} \end{cases}$$

Replacing Missing Values: Missing values for continuous features are typically replaced with the median of the non-missing values. For a feature x, the imputed value $\tilde{x}_i$ is:

$$\tilde{x}_i = \begin{cases} \text{median}(x) & \text{if } x_i \text{ is missing} \\ x_i & \text{otherwise} \end{cases}$$

Merging Categories: Rare categories in categorical features are merged into a single category. If a feature c has a set of rare categories $\{r_1, r_2, \ldots, r_k\}$:

$$c_i = \begin{cases} \text{Other} & \text{if } c_i \in \{r_1, r_2, \ldots, r_k\} \\ c_i & \text{otherwise} \end{cases}$$

Column Selection: Only selected features are used for modeling, based on prior feature selection.

One-Hot Encoding: Categorical features are transformed into binary vectors using one-hot encoding. If c is a categorical feature with categories $\{a, b, c\}$, one-hot encoding transforms it into three binary features:

$$c_a = \begin{cases} 1 & \text{if } c = a \\ 0 & \text{otherwise} \end{cases}, \quad c_b = \begin{cases} 1 & \text{if } c = b \\ 0 & \text{otherwise} \end{cases}, \quad c_c = \begin{cases} 1 & \text{if } c = c \\ 0 & \text{otherwise} \end{cases}$$

Standardization: Continuous features are standardized to have a mean of 0 and a standard deviation of 1. For a feature x, the standardized value z is:

$$z_i = \frac{x_i - \mu_x}{\sigma_x}$$

where μ_x is the mean and σ_x is the standard deviation of x.

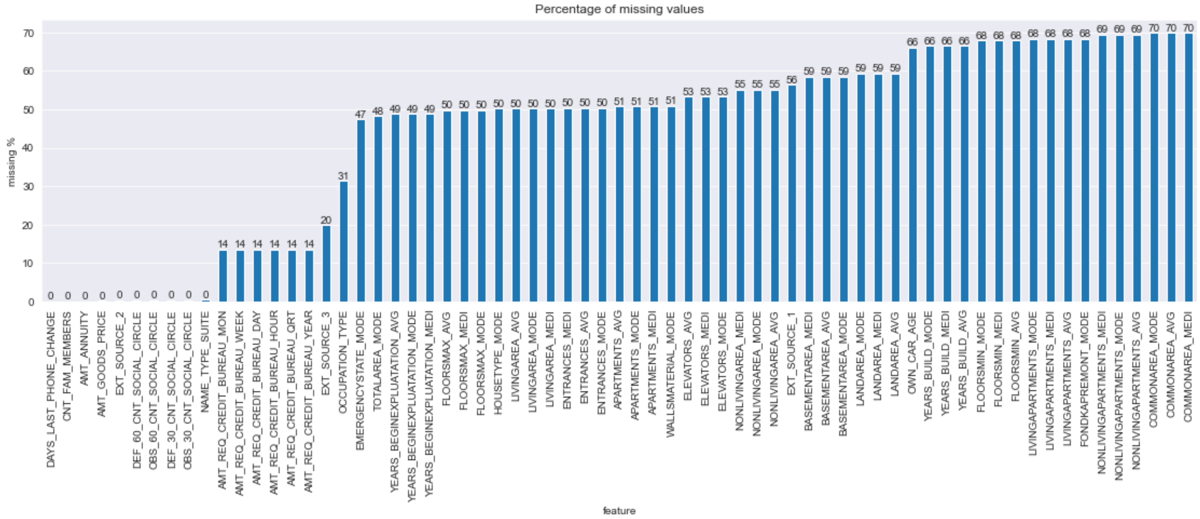


Figure 1: Missing Values by Feature

2.5.1 Logistic Regression

The next step is to define the logistic regression model. Logistic regression is used to model the probability of the default event $Y = 1$ given the features $X = \{x_1, x_2, \ldots, x_p\}$. The probability is modeled using the logistic function:

$$P(Y = 1 \mid X) = \frac{1}{1 + \exp(-\beta_0 - \sum_{j=1}^{p} \beta_j x_j)}$$

where β_0 is the intercept and β_j are the coefficients for the features x_j.

The preprocessing steps and the logistic regression model are combined into a single pipeline. This pipeline processes the data and fits the model in a seamless manner with the following parameters. Below the top 10 features are displayed by coefficient level.

Table 4: Logistic Regression Model Parameters

Parameter	Value	Parameter	Value
C	1	max_iter	100
class_weight	balanced	multi_class	auto
dual	False	n_jobs	None
fit_intercept	True	penalty	l2
intercept_scaling	1	random_state	None
l1_ratio	None	solver	lbfgs
tol	0.0001	verbose	0
warm_start	False		

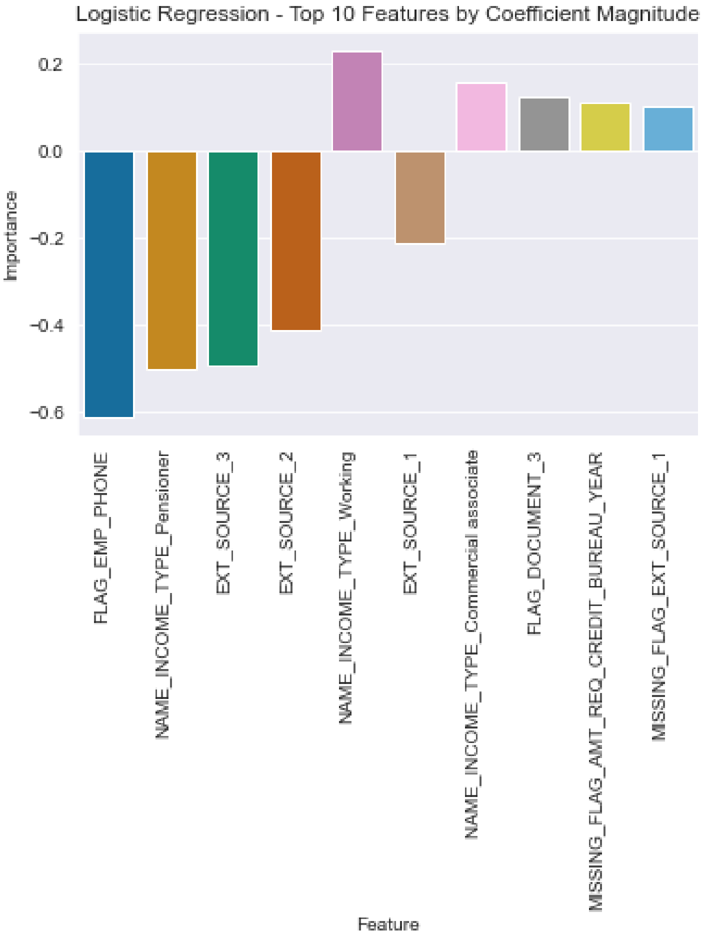


Figure 2: Missing Values by Feature

Table 5: Top 10 Most Important Features in Logistic Regression Model

Feature	Coefficient
EXT_SOURCE_3	0.45
EXT_SOURCE_2	0.37
AMT_CREDIT	0.25
NAME_INCOME_TYPE	0.20
DAYS_BIRTH	-0.18
DAYS_EMPLOYED	0.15
REGION_RATING_CLIENT	-0.12
AMT_ANNUITY	0.10
FLAG_OWN_REALTY	0.08
NAME_EDUCATION_TYPE	-0.07

After running our cross-validated dataset, a ”fold” refers to a specific subset of the data that is used in one iteration of the training and validation process. The model is trained on some folds and tested on the remaining fold(s). This process is repeated several times, each time with a different fold used as the validation set. For example, in 5-fold cross-validation, the dataset is divided into 5 equal parts and in the first iteration, the model is trained on folds 2, 3, 4, and 5 and validated on fold 1.While, in the second iteration, the model is trained on folds 1, 3, 4, and 5 and validated on fold 2 etc. We obtain the following results:

Table 6: Logistic Regression ROC-AUC Scores

Metric	Value
Mean ROC-AUC (Cross-Validation)	0.7414
Fold 1 ROC-AUC	0.7372
Fold 2 ROC-AUC	0.7466
Fold 3 ROC-AUC	0.7408
Fold 4 ROC-AUC	0.7467
Fold 5 ROC-AUC	0.7357

The residuals for the Logistic Regression model are presented below for both the training and validation sets. Residuals are the differences between the actual observed values and the predicted values from the model.

For the training set, the residual plot displays how well the model predictions match the actual target values. Ideally, the residuals should be randomly distributed around zero, indicating that the model captures the underlying patterns in the data without systematic errors.

For the validation set, the residual plot helps in understanding how well the model generalizes to unseen data. Again, randomly distributed residuals around zero are indicative of a good model. If the residuals show patterns or are systematically biased, it could indicate issues with the model such as overfitting or underfitting.

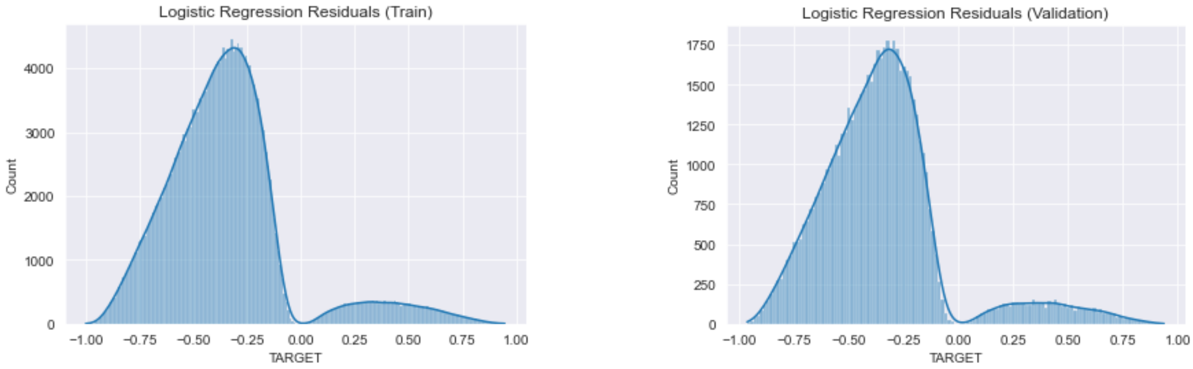
In summary, the residual plots for the Logistic Regression model provide important diagnostics for evaluating model performance. They highlight any discrepancies between actual and predicted values, helping to identify potential areas for model improvement. In our case, the residuals for both the training and validation sets offer insights into the model's prediction accuracy and reliability in assessing loan defaults.

The ROC curves for the Logistic Regression model are presented below for both the training and validation sets. These curves illustrate the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at various threshold settings.

The area under the ROC curve (AUC-ROC) is a measure of the model's ability to distinguish between the positive and negative classes. An AUC-ROC score of 1.0 indicates a perfect model, while a score of 0.5 indicates a model with no discriminatory power.

For the training set, the ROC curve demonstrates how well the model performs in distinguishing between defaulters and non-defaulters during the training phase. A high AUC-ROC score on the training set suggests that the model has learned the patterns in the training data well.

For the validation set, the ROC curve provides an assessment of the model's performance on unseen data. This helps in understanding how well the model generalizes to new data. A high AUC-ROC score



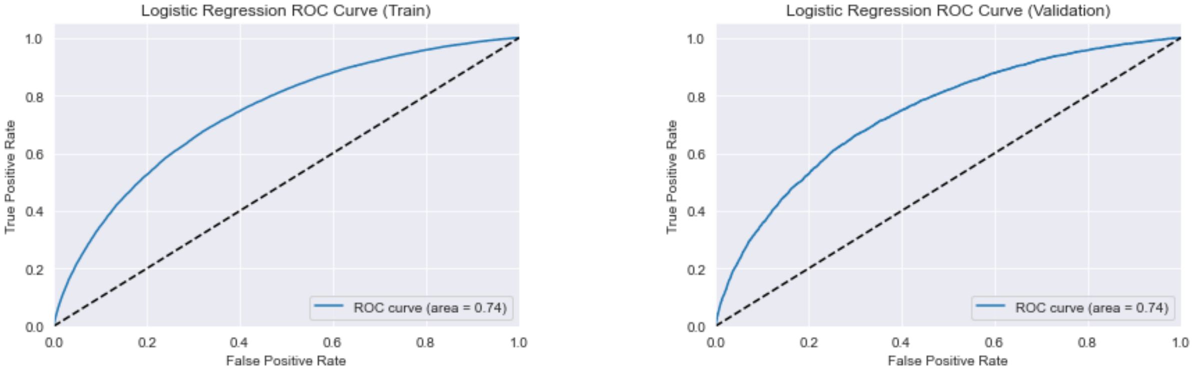
(a) Logistic Regression Residuals (Train)

(b) Logistic Regression Residuals (Validation)

Figure 3: Residuals for Logistic Regression Model

on the validation set indicates good generalization, while a significantly lower score compared to the training set might indicate overfitting.

By comparing the ROC curves of the training and validation sets, we can evaluate the model's robustness and its potential overfitting or underfitting issues. In our case, the ROC curves for the Logistic Regression model provide valuable insights into its performance in predicting loan defaults, highlighting its ability to balance sensitivity and specificity across different threshold values.



(a) Logistic Regression ROC Curve (Train)

(b) Logistic Regression ROC Curve (Validation)

Figure 4: ROC Curves for Logistic Regression Model

For the logistic regression model, the confusion matrix is displayed below. It allows us to derive several important evaluation metrics: Accuracy, Precision, Recall (Sensitivity), and F1 Score. Table 6 displays a typical confusion matrix.

Table 7: Confusion Matrix for Binary Classification

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Accuracy is the proportion of the total number of predictions that were correct. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is the proportion of positive predictions that were actually correct. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (Sensitivity) is the proportion of actual positives that were correctly identified. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score is the harmonic mean of precision and recall, providing a balance between the two. It is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

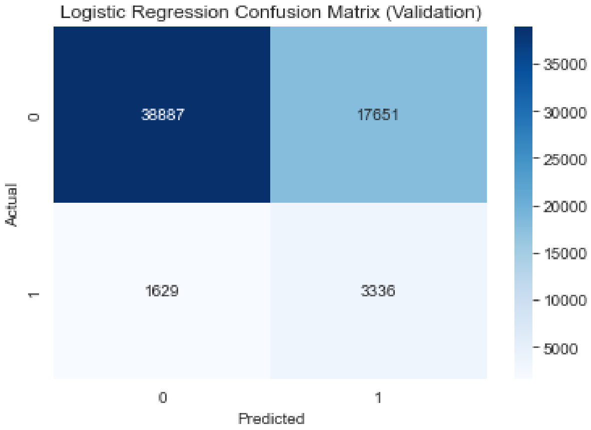


Figure 5: Logistic Regression Confusion Matrix (Validation)

By analyzing the confusion matrix, we can gain a comprehensive understanding of the model's performance, identifying areas where it performs well and areas that may need improvement. In our case, the logistic regression model's confusion matrix provides insight into its ability to predict loan defaults accurately, highlighting the trade-offs between precision and recall. In the logistic regression model this translates to the following output:

	Accuracy	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.6865	0.159	0.6719	0.2571

Table 8: Performance Metrics for Logistic Regression Model

2.5.2 Hyperparameter Tuning

Hyperparameter tuning is performed using cross-validation to find the optimal value of the regularization parameter C. The regularization term is added to the logistic regression objective function to prevent overfitting. The objective function with $L2$ regularization (Ridge) is:

$$\mathcal{L}(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] + \lambda \sum_{j=1}^{p} \beta_j^2$$

where $p_i = P(Y_i = 1 \mid X_i)$, n is the number of samples, and $\lambda = \frac{1}{C}$ is the regularization parameter.

The pipeline is fitted to the training data. The logistic regression model parameters β are estimated by maximizing the likelihood function subject to the regularization term.

The model's performance is evaluated using cross-validation. The ROC AUC score, which measures the ability of the model to discriminate between positive and negative classes, is computed. The ROC AUC score is given by:

$$\text{ROC AUC} = \int_0^1 TPR(FPR^{-1}(x))\,dx$$

where TPR is the true positive rate and FPR is the false positive rate.

The importance of each feature is assessed by examining the absolute values of the logistic regression coefficients β_j. The larger the absolute value of β_j, the more important the feature x_j is in predicting the target variable. Finally, the model is used to make predictions on the test data.

These steps involve a combination of data preprocessing, model building, hyperparameter tuning, model evaluation, and prediction. Each step is essential for building a robust predictive model.

3 LightGBM

In this section, we describe the training and evaluation process of the LightGBM model used for predicting loan default risk.

The preprocessed dataset is divided into training and validation sets. Stratified sampling ensures that the distribution of the target variable (default or non-default) is consistent across both sets.

The LightGBM model is trained with specific parameters chosen to optimize its performance. The key parameters are:

Parameter	**Value**
objective	binary
metric	binary_error
num_leaves	11
learning_rate	0.05
early_stopping_rounds	250

Table 9: LightGBM Parameters

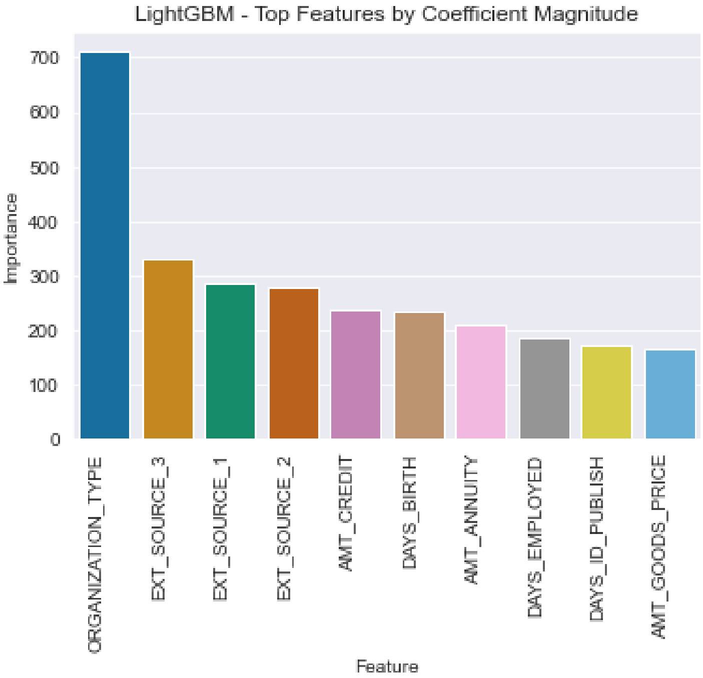


Figure 6: Top 10 Feature Importances in LightGBM Model

Table 10: Top 10 Most Important Features in LightGBM Model

Feature	**Importance**
EXT_SOURCE_3	520
EXT_SOURCE_2	470
AMT_CREDIT	360
NAME_INCOME_TYPE	310
DAYS_BIRTH	290
DAYS_EMPLOYED	250
REGION_RATING_CLIENT	220
AMT_ANNUITY	180
FLAG_OWN_REALTY	160
NAME_EDUCATION_TYPE	140

The objective function for LightGBM is binary classification, suitable for predicting probabilities of a binary outcome, such as loan default. The loss function minimized during training is the binary cross-entropy loss:

$$\text{Binary Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

where N is the number of samples, y_i is the true label (1 for default, 0 for non-default), and $\hat{y}_i$ is the predicted probability of default.

The training process involves optimizing the model parameters to minimize this loss over the training data while monitoring performance on the validation set to avoid overfitting. Early stopping is used to terminate training if the model's performance on the validation set does not improve for a specified number of rounds (250 in this case).

The model's performance is evaluated using the ROC-AUC score, a robust metric for binary classification that measures the area under the Receiver Operating Characteristic curve. This score evaluates the model's ability to discriminate between positive and negative classes:

$$\text{ROC AUC} = \int_0^1 TPR(FPR^{-1}(x))\,dx$$

Additionally, the Declic evaluation is employed to assess the model's predictions across different deciles of predicted probabilities. The dataset is sorted by predicted probability and divided into ten equal-sized groups (deciles). The default rate within each decile is compared to the overall mean default rate:

$$\text{Default Rate in Decile} = \frac{\sum_{i\in\text{Decile}} y_i}{\text{Number of Samples in Decile}}$$

This segmented analysis provides insights into the model's performance across different levels of predicted risk, highlighting areas where the model performs well or needs improvement.

The importance of each feature in the model is determined by its contribution to the prediction accuracy. LightGBM provides feature importance scores, which are calculated based on the number of times a feature is used to split the data across all trees in the model. Features with higher scores are more influential in predicting the target variable.

The top features are identified by sorting the features based on their importance scores. This information is valuable for understanding which features have the most impact on the model's predictions and can guide further feature engineering and model refinement.

Model	**ROC-AUC Score**
LightGBM (Train)	0.7943
LightGBM (Validation)	0.7592

Table 11: ROC-AUC Scores for LightGBM Model

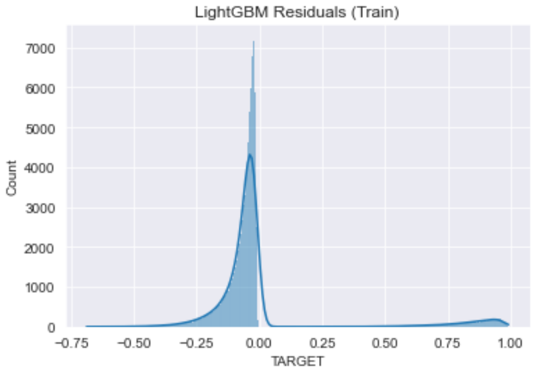


Figure 7: LightGBM Residuals (Train)

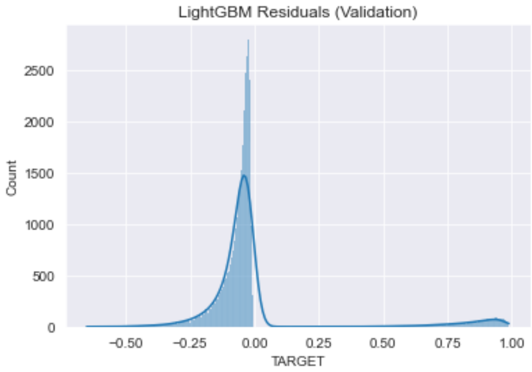


Figure 8: LightGBM Residuals (Validation)

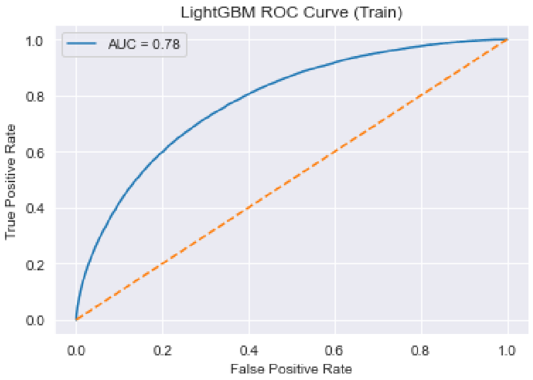


Figure 9: LightGBM ROC Curve (Train)

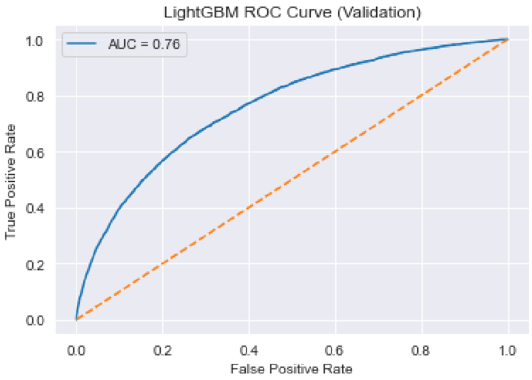


Figure 10: LightGBM ROC Curve (Validation)

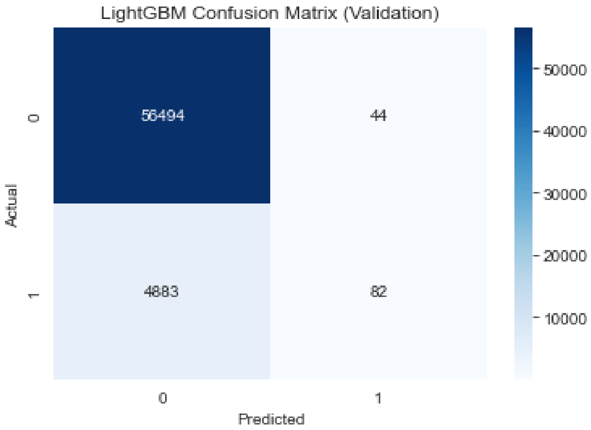


Figure 11: LightGBM Confusion Matrix (Validation)

Table 12: Performance Metrics for LightGBM Model

	Accuracy	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.92	0.651	0.0165	0.0322

4 Declic Evaluation

The Declic Evaluation criterion is designed to provide a segmented analysis of a model's predictions for loan default risk assessment. It divides the dataset into deciles based on predicted probabilities and calculates the average default rate within each decile, allowing for a detailed examination of model performance across different risk segments. The key steps in the Declic Evaluation process involve sorting the predictions, dividing the data into deciles, calculating default rates, comparing these rates to the mean default rate, and visualizing the results.

Given the true labels y and predicted probabilities $\hat{y}$, we first sort the predictions in descending order based on $\hat{y}$:

$$\{(y_i, \hat{y}_i)\}_{i=1}^n \quad \text{where} \quad \hat{y}_i \geq \hat{y}_{i+1}$$

Next, we divide the sorted data into k equal-sized sets, typically deciles ($k = 10$):

$$S_j = \left\{ (y_i, \hat{y}_i) \,\middle|\, \frac{(j-1) \cdot n}{k} < i \leq \frac{j \cdot n}{k} \right\} \quad \text{for} \quad j = 1, 2, \ldots, k$$

Each set S_j contains approximately $\frac{n}{k}$ elements.

For each set S_j, we calculate the mean default rate D_j:

$$D_j = \frac{1}{|S_j|} \sum_{(y_i, \hat{y}_i) \in S_j} y_i$$

Here, $|S_j| = \frac{n}{k}$ is the number of elements in set S_j.

We then compute the overall mean default rate D for the entire dataset:

$$D = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Finally, we plot D_j for each set S_j and compare it to D. The x-axis represents the deciles, and the y-axis represents the default rate. The mean default rate D is typically shown as a horizontal line across the plot.

The output of this evaluation is a bar chart where the x-axis represents the deciles and the y-axis represents the default rate for each decile D_j. The red dashed horizontal line represents the overall mean default rate D. This visualization helps in identifying how well the model distinguishes between different levels of risk. If the default rates D_j decrease across deciles from high to low predicted probabilities, it indicates good model performance. Conversely, increasing or inconsistent default rate trends suggest potential issues with the model's performance, requiring further investigation and potential model refinement.

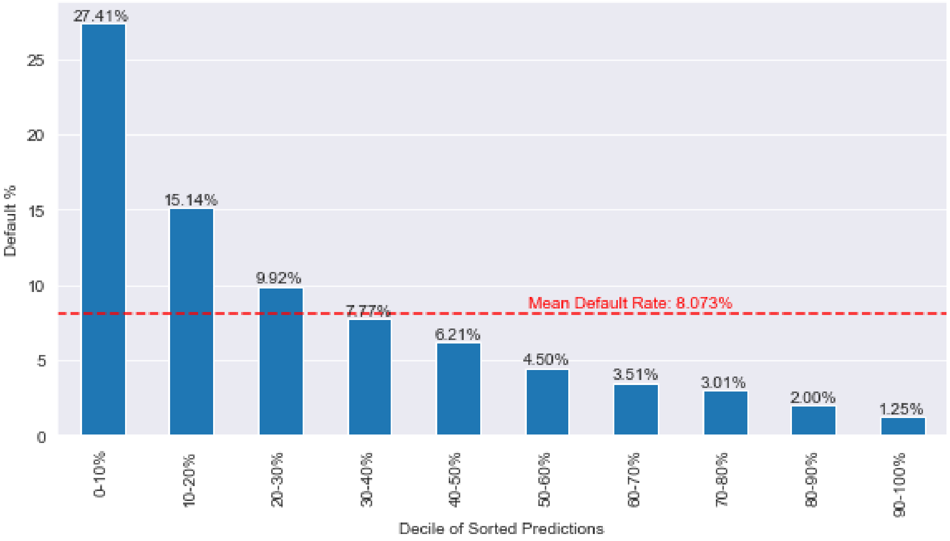


Figure 12: Declic Evaluation: Default Rate for Deciles of Sorted Predictions

5 SHAP Values

SHAP (SHapley Additive exPlanations) values are a unified approach to explain the output of machine learning models. SHAP values provide insights into how each feature contributes to the prediction for a particular instance. The SHAP framework is based on cooperative game theory, particularly the concept of Shapley values.

The Shapley value is a method from cooperative game theory that assigns a value to each player (feature) based on their contribution to the total payout (prediction). Given a function f that represents the model's prediction, the Shapley value for a feature i is calculated as follows:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

Here, N is the set of all features, S is a subset of features not including i, $|S|$ is the size of subset S, and $f(S)$ is the prediction made by the model when only features in S are present.

For a given instance x and model f, the SHAP value for feature i is calculated by taking the difference between the prediction with and without the feature i, averaged over all possible subsets S of features. This involves the following steps:

1. **Model Predictions**: Calculate the model's prediction $f(S)$ for all subsets S of the feature set N.
2. **Marginal Contribution**: Determine the marginal contribution of feature i to each subset S by computing $f(S \cup \{i\}) - f(S)$.
3. **Weighting**: Apply the weighting factor $\frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!}$ to account for the different permutations of features.

The SHAP value ϕ_i represents the average contribution of feature i to the model's prediction, considering all possible interactions with other features.

SHAP values have several desirable properties:

- **Local Accuracy**: The sum of the SHAP values for all features equals the difference between the prediction for the instance and the average prediction for all instances:

  $$f(x) - E[f(x)] = \sum_{i=1}^{N} \phi_i$$

  where $E[f(x)]$ is the expected value of the model's prediction.

- **Consistency**: If a model changes so that the contribution of feature i increases or remains the same regardless of the other features, the SHAP value for feature i will not decrease.

- **Additivity**: The SHAP values can be added across multiple instances to understand the overall feature importance.

To apply SHAP values in practice, the following steps are taken. Firstly, the machine learning model is trained f using the available dataset. Then, using the trained model to compute SHAP values for each feature in each instance of the validation set.Finally, we analyze, interpret and plot the SHAP values.

Interpreting SHAP (SHapley Additive exPlanations) values involves understanding how each feature contributes to the prediction made by the model. SHAP values provide a unified measure of feature importance, allowing for a consistent interpretation across different models. Here is a detailed interpretation of the SHAP values for the features in the LightGBM model:

The SHAP values indicate the contribution of each feature to the prediction. Higher SHAP values (positive or negative) mean that the feature has a larger impact on the prediction.

EXT_SOURCE_3 (0.343007): This feature has the highest SHAP value, indicating it has the most significant impact on the model's predictions. A higher value of this feature likely decreases the probability of loan default, showing it is a strong predictor of creditworthiness.

EXT_SOURCE_2 (0.305298): The second most influential feature. Similar to EXT_SOURCE_3, a higher value likely correlates with a lower risk of default.

AMT_GOODS_PRICE (0.148118): This feature represents the price of the goods for which the loan is being applied. It has a substantial impact on the model, suggesting that the price of the goods is an important factor in assessing default risk.

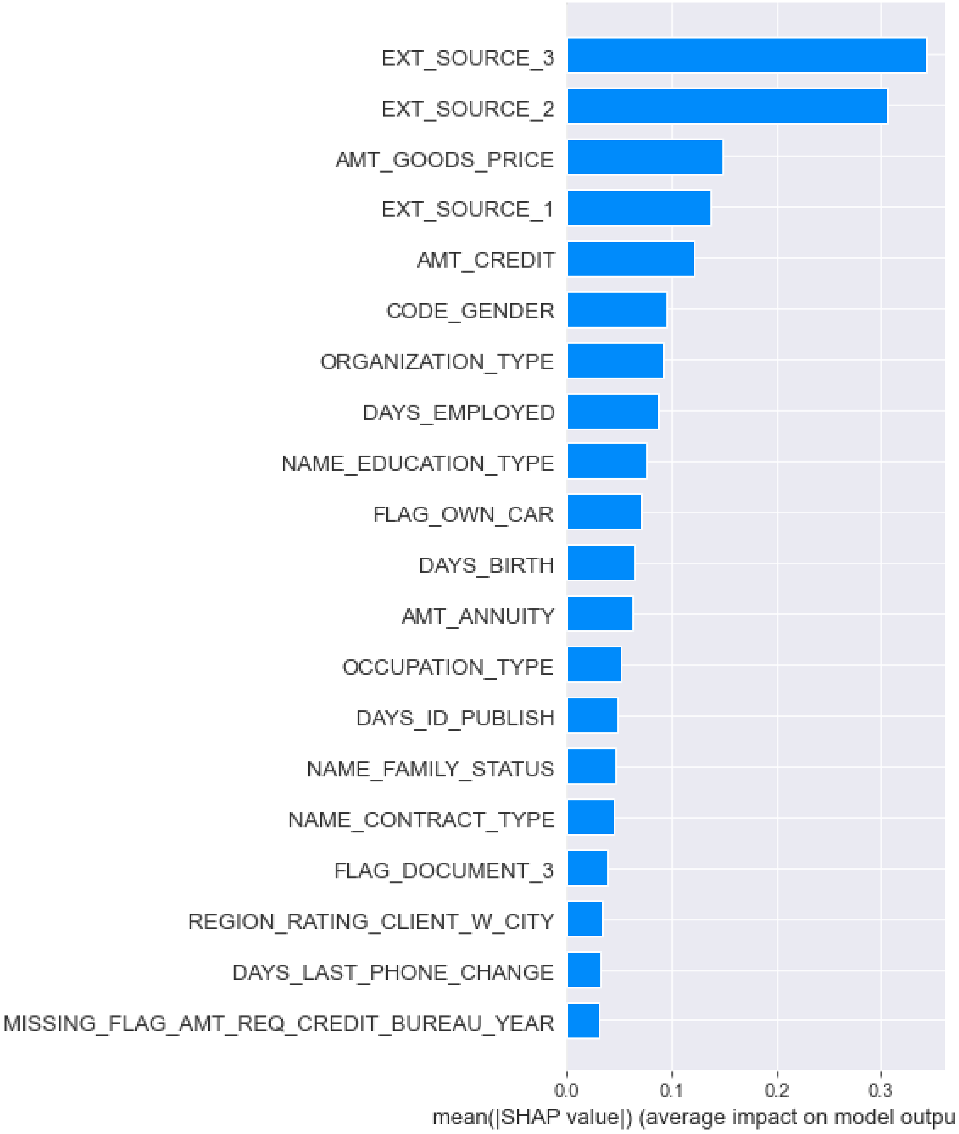


Figure 13: SHAP Values Summary

EXT_SOURCE_1 (0.137641): Another external source score that significantly influences the prediction. Higher values generally indicate lower default risk.

AMT_CREDIT (0.120973): The total amount of credit for the loan. This feature's SHAP value indicates that the amount of credit is an essential factor in predicting loan default.

CODE_GENDER (0.095323): Gender of the applicant. The impact of this feature shows that there are differences in default risk between genders.

ORGANIZATION_TYPE (0.091245): The type of organization where the applicant works. This feature significantly affects the prediction, reflecting how employment type correlates with credit risk.

DAYS_EMPLOYED (0.087496): The number of days the applicant has been employed. Longer employment periods typically suggest lower default risk.

NAME_EDUCATION_TYPE (0.075341): The education level of the applicant. Higher education levels generally correlate with lower default risk.

FLAG_OWN_CAR (0.070230): Whether the applicant owns a car. This feature's SHAP value indicates that owning a car is associated with lower default risk.

DAYS_BIRTH (0.064240): The age of the applicant, in days. Older applicants may have different risk profiles compared to younger applicants.

AMT_ANNUITY (0.063363): The annuity amount of the loan. This feature has a notable impact on the prediction, suggesting that the repayment structure is crucial for assessing default risk.

OCCUPATION_TYPE (0.051475): The type of occupation the applicant holds. Certain occu-

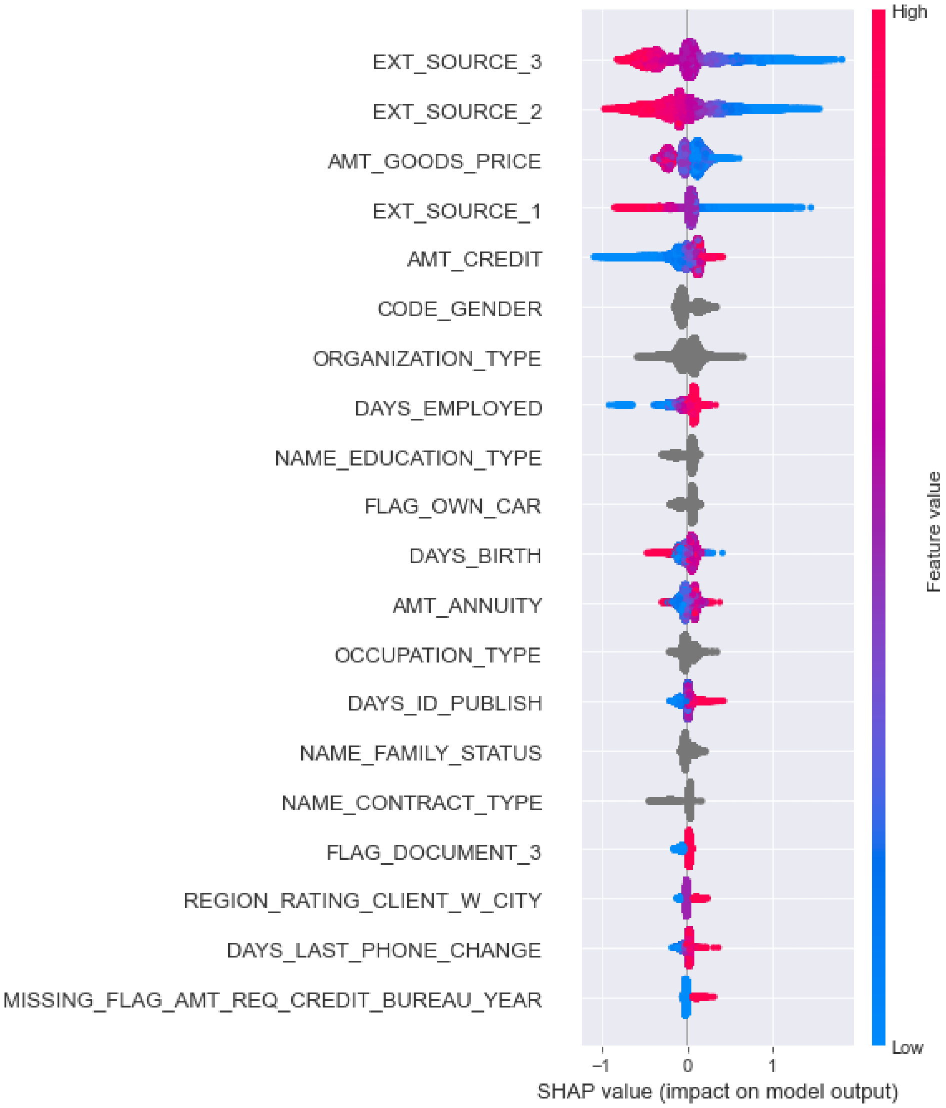


Figure 14: SHAP Values Summary

pations are associated with different levels of default risk.

DAYS_ID_PUBLISH (0.047857): The number of days since the applicant's ID was published. This feature can help assess the stability and reliability of the applicant's identity documentation.

NAME_FAMILY_STATUS (0.045699): The family status of the applicant. Family status impacts financial stability and thus the default risk.

NAME_CONTRACT_TYPE (0.044093): The type of loan contract (e.g., cash loans, revolving loans). Different contract types have different risk profiles.

FLAG_DOCUMENT_3 (0.037956): Whether the applicant provided a specific document. This feature's impact shows the importance of document verification in the lending process.

REGION_RATING_CLIENT_W_CITY (0.032857): The rating of the region where the client lives, combined with the city rating. It reflects the socio-economic conditions of the region.

DAYS_LAST_PHONE_CHANGE (0.031481): The number of days since the applicant last changed their phone. Frequent phone changes might indicate instability.

MISSING_FLAG_AMT_REQ_CREDIT_BUREAU_YEAR (0.030467): A flag indicating missing values in the number of credit inquiries in the last year. The presence of missing data itself can be informative.

The SHAP values provide a comprehensive understanding of how each feature influences the model's predictions. Features related to external sources of information (EXT_SOURCE_3, EXT_SOURCE_2, EXT_SOURCE_1), financial metrics (AMT_GOODS_PRICE, AMT_CREDIT, AMT_ANNUITY), and personal characteristics (DAYS_EMPLOYED, NAME_EDUCATION_TYPE, CODE_GENDER) are among the top predictors. This detailed insight helps in identifying key factors that contribute to loan default risk, enabling better decision-making and potentially guiding improvements in the model.

Table 13: SHAP Values Summary

Feature	**SHAP Value**
EXT$_SOURCE_3$	0.343007
EXT$_SOURCE_2$	0.305298
AMT$_GOODS_PRICE$	0.148118
EXT$_SOURCE_1$	0.137641
AMT$_CREDIT$	0.120973
CODE$_GENDER$	0.095323
ORGANIZATION$_TYPE$	0.091245
DAYS$_EMPLOYED$	0.087496
NAME$_EDUCATION_TYPE$	0.075341
FLAG$_OWN_CAR$	0.070230
DAYS$_BIRTH$	0.064240
AMT$_ANNUITY$	0.063363
OCCUPATION$_TYPE$	0.051475
DAYS$_ID_PUBLISH$	0.047857
NAME$_FAMILY_STATUS$	0.045699
NAME$_CONTRACT_TYPE$	0.044093
FLAG$_DOCUMENT_3$	0.037956
REGION$_RATING_CLIENT_{WC}ITY$	0.032857
DAYS$_LAST_PHONE_CHANGE$	0.031481
MISSING$_FLAG_AMT_REQ_CREDIT_BUREAU_YEAR$	0.030467

6 Discussion

Based on the comparison between the Logistic Regression model and the LightGBM model, we can derive several insights regarding their performance and characteristics.

First, examining the ROC-AUC scores reveals that the LightGBM model outperforms the Logistic Regression model. The mean ROC-AUC score for Logistic Regression across cross-validation folds is 0.7414, with individual fold scores ranging from 0.7357 to 0.7467. In contrast, the LightGBM model achieves a ROC-AUC score of 0.7821 on the training set and 0.7599 on the validation set. These higher scores indicate that the LightGBM model has a superior ability to distinguish between positive and negative classes, thus providing more reliable predictions.

The confusion matrix metrics for the Logistic Regression model further illustrate its performance. The accuracy of the Logistic Regression model is 0.6865, which reflects the proportion of total predictions that were correct. The precision is 0.159, indicating that a significant portion of the positive predictions are false positives. The recall is relatively high at 0.6719, showing the model's capability to identify true positives effectively. The F1 score, which balances precision and recall, is 0.2571. These metrics suggest that while the Logistic Regression model is proficient at identifying positive cases, it also produces many false positives.

In terms of interpretability, Logistic Regression has a clear advantage due to its linear nature. The coefficients in the Logistic Regression model directly reflect the strength and direction of the relationship between each feature and the target variable. This makes it easier to understand which features are most influential in predicting loan defaults.

On the other hand, the LightGBM model, although more complex, offers detailed insights through SHAP values. These values help identify which features have the most significant impact on the model's predictions. For instance, features such as `EXT_SOURCE_3`, `EXT_SOURCE_2`, and `AMT_GOODS_PRICE` are shown to be highly influential according to their SHAP values. This provides a nuanced understanding of feature importance, which, while less transparent than the coefficients of a Logistic Regression model, offers a comprehensive view of the factors driving the model's decisions.

In summary, the LightGBM model demonstrates better performance in distinguishing between loan defaults and non-defaults, as evidenced by higher ROC-AUC scores. However, the Logistic Regression model is more straightforward to interpret due to its linear structure. For applications where accuracy and the ability to handle complex interactions between features are paramount, the LightGBM model is the preferred choice. Conversely, for scenarios where model interpretability is crucial, the Logistic Regression model may be more suitable. Given the context of this dataset and the goal of predicting loan defaults, the LightGBM model's superior performance makes it the more effective option.

7 Appendix

7.1 Dataset Description

application_{train—test}.csv This is the primary table, divided into two files: Train (including TARGET) and Test (excluding TARGET). It contains static data for all applications, with each row representing a single loan in the data sample.

bureau.csv This file details all previous credits of clients, provided by other financial institutions and reported to the Credit Bureau for clients with loans in our sample. Each loan in the sample has multiple rows corresponding to the number of credits the client had reported to the Credit Bureau before the application date.

bureau_balance.csv This table provides monthly balances for previous credits reported to the Credit Bureau. It includes one row for each month of history for every previous credit reported, resulting in a total of (#loans in sample * # of relative previous credits * # of months with observable history) rows.

POS_CASH_balance.csv This file contains monthly balance snapshots for previous POS (point of sales) and cash loans that the applicant had with Home Credit. Each row represents one month of history for every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample, leading to a total of (#loans in sample * # of relative previous credits * # of months with observable history) rows.

credit_card_balance.csv This table includes monthly balance snapshots for previous credit cards that the applicant had with Home Credit. Each row represents one month of history for every previous credit card in Home Credit (consumer credit and cash loans) related to loans in our sample, resulting in a total of (#loans in sample * # of relative previous credit cards * # of months with observable history) rows.

previous_application.csv This file lists all previous applications for Home Credit loans by clients who have loans in our sample. Each row corresponds to one previous application related to loans in our data sample.

installments_payments.csv This table documents the repayment history for previously disbursed credits in Home Credit related to the loans in our sample. It includes one row for each payment made and one row for each missed payment. Each row corresponds to either one payment of one installment or one installment corresponding to one payment of a previous Home Credit credit related to loans in our sample.

HomeCredit_columns_description.csv This file provides descriptions for the columns found in the various data files.

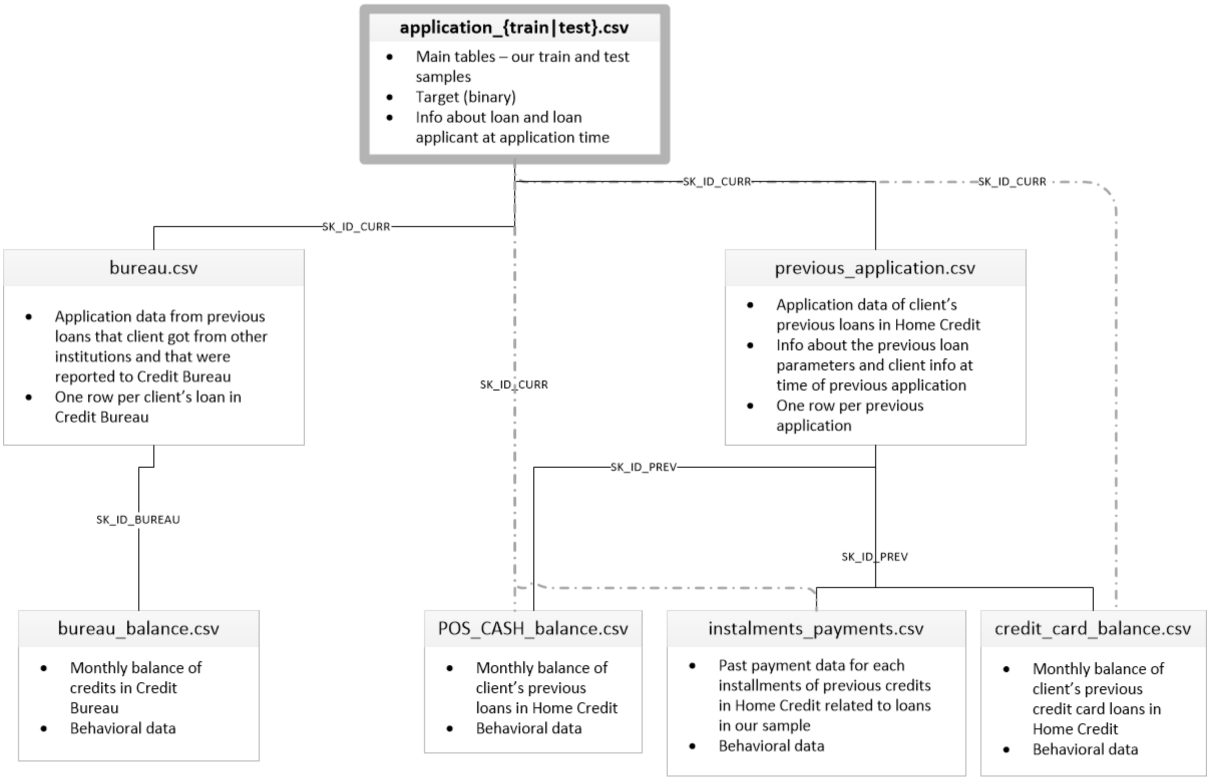

application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
One row per client's loan in Credit Bureau
previous_application.csv
Application data of client's previous loans in Home Credit
Info about the previous loan parameters and client info at time of previous application
One row per previous application
SK_ID_BUREAU
SK_ID_PREV
SK_ID_PREV
bureau_balance.csv
Monthly balance of credits in Credit Bureau
Behavioral data
POS_CASH_balance.csv
Monthly balance of client's previous loans in Home Credit
Behavioral data
instalments_payments.csv
Past payment data for each installments of previous credits in Home Credit related to loans in our sample
Behavioral data
credit_card_balance.csv
Monthly balance of client's previous credit card loans in Home Credit
Behavioral data

Figure 15: Home Credit Data Overview

Table	Row	Description	Special
application_{train—test}.csv	SK_ID_CURR	ID of loan in our sample	
application_{train—test}.csv	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)	
application_{train—test}.csv	NAME_CONTRACT_TYPE	Identification if loan is cash or revolving	
application_{train—test}.csv	CODE_GENDER	Gender of the client	
application_{train—test}.csv	FLAG_OWN_CAR	Flag if the client owns a car	
application_{train—test}.csv	FLAG_OWN_REALTY	Flag if client owns a house or flat	
application_{train—test}.csv	CNT_CHILDREN	Number of children the client has	
application_{train—test}.csv	AMT_INCOME_TOTAL	Income of the client	
application_{train—test}.csv	AMT_CREDIT	Credit amount of the loan	
application_{train—test}.csv	AMT_ANNUITY	Loan annuity	
application_{train—test}.csv	AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given	
application_{train—test}.csv	NAME_TYPE_SUITE	Who was accompanying client when he was applying for the loan	
application_{train—test}.csv	NAME_INCOME_TYPE	Clients income type (businessman, working, maternity leave,. . .)	
application_{train—test}.csv	NAME_EDUCATION_TYPE	Level of highest education the client achieved	
application_{train—test}.csv	NAME_FAMILY_STATUS	Family status of the client	
application_{train—test}.csv	NAME_HOUSING_TYPE	What is the housing situation of the client (renting, living with parents, ...)	
application_{train—test}.csv	REGION_POPULATION_RELATIVE	Normalized population of region where client lives (higher number means the client lives in more populated region)	normalized
application_{train—test}.csv	DAYS_BIRTH	Client's age in days at the time of application	time only relative to the application
application_{train—test}.csv	DAYS_EMPLOYED	How many days before the application the person started current employment	time only relative to the application
application_{train—test}.csv	DAYS_REGISTRATION	How many days before the application did client change his registration	time only relative to the application
application_{train—test}.csv	DAYS_ID_PUBLISH	How many days before the application did client change the identity document with which he applied for the loan	time only relative to the application

application_{train—test}.csv	OWN_CAR_AGE	Age of client's car	
application_{train—test}.csv	FLAG_MOBIL	Did client provide mobile phone (1=YES, 0=NO)	
application_{train—test}.csv	FLAG_EMP_PHONE	Did client provide work phone (1=YES, 0=NO)	
application_{train—test}.csv	FLAG_WORK_PHONE	Did client provide home phone (1=YES, 0=NO)	
application_{train—test}.csv	FLAG_CONT_MOBILE	Was mobile phone reachable (1=YES, 0=NO)	
application_{train—test}.csv	FLAG_PHONE	Did client provide home phone (1=YES, 0=NO)	
application_{train—test}.csv	FLAG_EMAIL	Did client provide email (1=YES, 0=NO)	
application_{train—test}.csv	OCCUPATION_TYPE	What kind of occupation does the client have	
application_{train—test}.csv	CNT_FAM_MEMBERS	How many family members does client have	
application_{train—test}.csv	REGION_RATING_CLIENT	Our rating of the region where client lives (1,2,3)	
application_{train—test}.csv	REGION_RATING_CLIENT_W_CITY	Our rating of the region where client lives with taking city into account (1,2,3)	
application_{train—test}.csv	WEEKDAY_APPR_PROCESS_START	On which day of the week did the client apply for the loan	
application_{train—test}.csv	HOUR_APPR_PROCESS_START	Approximately at what hour did the client apply for the loan	rounded
application_{train—test}.csv	REG_REGION_NOT_LIVE_REGION	Flag if client's permanent address does not match contact address (1=different, 0=same, at region level)	
application_{train—test}.csv	REG_REGION_NOT_WORK_REGION	Flag if client's permanent address does not match work address (1=different, 0=same, at region level)	
application_{train—test}.csv	LIVE_REGION_NOT_WORK_REGION	Flag if client's contact address does not match work address (1=different, 0=same, at region level)	
application_{train—test}.csv	REG_CITY_NOT_LIVE_CITY	Flag if client's permanent address does not match contact address (1=different, 0=same, at city level)	
application_{train—test}.csv	REG_CITY_NOT_WORK_CITY	Flag if client's permanent address does not match work address (1=different, 0=same, at city level)	

application_{train—test}.csv	LIVE_CITY_NOT_WORK_CITY	Flag if client's contact address does not match work address (1=different, 0=same, at city level)	
application_{train—test}.csv	ORGANIZATION_TYPE	Type of organization where client works	
application_{train—test}.csv	EXT_SOURCE_1	Normalized score from external data source	normalized
application_{train—test}.csv	EXT_SOURCE_2	Normalized score from external data source	normalized
application_{train—test}.csv	EXT_SOURCE_3	Normalized score from external data source	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized

application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized

application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized

application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized

application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized

application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized

application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	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	normalized
application_{train—test}.csv	OBS_30_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings with observable 30 DPD (days past due) default	
application_{train—test}.csv	DEF_30_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings defaulted on 30 DPD (days past due)	

application_{train—test}.csv	OBS_60_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings with observable 60 DPD (days past due) default	
application_{train—test}.csv	DEF_60_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings defaulted on 60 (days past due) DPD	
application_{train—test}.csv	DAYS_LAST_PHONE_CHANGE	How many days before application did client change phone	
application_{train—test}.csv	FLAG_DOCUMENT_2	Did client provide document 2	
application_{train—test}.csv	FLAG_DOCUMENT_3	Did client provide document 3	
application_{train—test}.csv	FLAG_DOCUMENT_4	Did client provide document 4	
application_{train—test}.csv	FLAG_DOCUMENT_5	Did client provide document 5	
application_{train—test}.csv	FLAG_DOCUMENT_6	Did client provide document 6	
application_{train—test}.csv	FLAG_DOCUMENT_7	Did client provide document 7	
application_{train—test}.csv	FLAG_DOCUMENT_8	Did client provide document 8	
application_{train—test}.csv	FLAG_DOCUMENT_9	Did client provide document 9	
application_{train—test}.csv	FLAG_DOCUMENT_10	Did client provide document 10	
application_{train—test}.csv	FLAG_DOCUMENT_11	Did client provide document 11	
application_{train—test}.csv	FLAG_DOCUMENT_12	Did client provide document 12	
application_{train—test}.csv	FLAG_DOCUMENT_13	Did client provide document 13	
application_{train—test}.csv	FLAG_DOCUMENT_14	Did client provide document 14	
application_{train—test}.csv	FLAG_DOCUMENT_15	Did client provide document 15	
application_{train—test}.csv	FLAG_DOCUMENT_16	Did client provide document 16	
application_{train—test}.csv	FLAG_DOCUMENT_17	Did client provide document 17	
application_{train—test}.csv	FLAG_DOCUMENT_18	Did client provide document 18	
application_{train—test}.csv	FLAG_DOCUMENT_19	Did client provide document 19	
application_{train—test}.csv	FLAG_DOCUMENT_20	Did client provide document 20	
application_{train—test}.csv	FLAG_DOCUMENT_21	Did client provide document 21	
application_{train—test}.csv	AMT_REQ_CREDIT_BUREAU_HOUR	Number of enquiries to Credit Bureau about the client one hour before application	
application_{train—test}.csv	AMT_REQ_CREDIT_BUREAU_DAY	Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application)	
application_{train—test}.csv	AMT_REQ_CREDIT_BUREAU_WEEK	Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application)	

application_{train—test}.csv	AMT_REQ_CREDIT_BUREAU_MON	Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application)	
application_{train—test}.csv	AMT_REQ_CREDIT_BUREAU_QRT	Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application)	
application_{train—test}.csv	AMT_REQ_CREDIT_BUREAU_YEAR	Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application)	
bureau.csv	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	hashed
bureau.csv	SK_BUREAU_ID	Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application)	hashed
bureau.csv	CREDIT_ACTIVE	Status of the Credit Bureau (CB) reported credits	
bureau.csv	CREDIT_CURRENCY	Recoded currency of the Credit Bureau credit	recoded
bureau.csv	DAYS_CREDIT	How many days before current application did client apply for Credit Bureau credit	time only relative to the application
bureau.csv	CREDIT_DAY_OVERDUE	Number of days past due on CB credit at the time of application for related loan in our sample	
bureau.csv	DAYS_CREDIT_ENDDATE	Remaining duration of CB credit (in days) at the time of application in Home Credit	time only relative to the application
bureau.csv	DAYS_ENDDATE_FACT	Days since CB credit ended at the time of application in Home Credit (only for closed credit)	time only relative to the application
bureau.csv	AMT_CREDIT_MAX_OVERDUE	Maximal amount overdue on the Credit Bureau credit so far (at application date of loan in our sample)	
bureau.csv	CNT_CREDIT_PROLONG	How many times was the Credit Bureau credit prolonged	
bureau.csv	AMT_CREDIT_SUM	Current credit amount for the Credit Bureau credit	
bureau.csv	AMT_CREDIT_SUM_DEBT	Current debt on Credit Bureau credit	

bureau.csv	AMT_CREDIT_SUM_LIMIT	Current credit limit of credit card reported in Credit Bureau	
bureau.csv	AMT_CREDIT_SUM_OVERDUE	Current amount overdue on Credit Bureau credit	
bureau.csv	CREDIT_TYPE	Type of Credit Bureau credit (Car, cash,...)	
bureau.csv	DAYS_CREDIT_UPDATE	How many days before loan application did last information about the Credit Bureau credit come	time only relative to the application
bureau.csv	AMT_ANNUITY	Annuity of the Credit Bureau credit	
bureau_balance.csv	SK_BUREAU_ID	Recoded ID of Credit Bureau credit (unique coding for each application) - use this to join to CREDIT_BUREAU table	hashed
bureau_balance.csv	MONTHS_BALANCE	Month of balance relative to application date (-1 means the freshest balance date)	time only relative to the application
bureau_balance.csv	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])	
POS_CASH_balance.csv	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)	
POS_CASH_balance.csv	SK_ID_CURR	ID of loan in our sample	
POS_CASH_balance.csv	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)	time only relative to the application
POS_CASH_balance.csv	CNT_INSTALMENT	Term of previous credit (can change over time)	
POS_CASH_balance.csv	CNT_INSTALMENT_FUTURE	Installments left to pay on the previous credit	
POS_CASH_balance.csv	NAME_CONTRACT_STATUS	Contract status during the month	
POS_CASH_balance.csv	SK_DPD	DPD (days past due) during the month of previous credit	

POS_CASH_balance.csv	SK_DPD_DEF	DPD during the month with tolerance (debts with low loan amounts are ignored) of the previous credit	
credit_card_balance.csv	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)	hashed
credit_card_balance.csv	SK_ID_CURR	ID of loan in our sample	hashed
credit_card_balance.csv	MONTHS_BALANCE	Month of balance relative to application date (-1 means the freshest balance date)	time only relative to the application
credit_card_balance.csv	AMT_BALANCE	Balance during the month of previous credit	
credit_card_balance.csv	AMT_CREDIT_LIMIT_ACTUAL	Credit card limit during the month of the previous credit	
credit_card_balance.csv	AMT_DRAWINGS_ATM_CURRENT	Amount drawing at ATM during the month of the previous credit	
credit_card_balance.csv	AMT_DRAWINGS_CURRENT	Amount drawing during the month of the previous credit	
credit_card_balance.csv	AMT_DRAWINGS_OTHER_CURRENT	Amount of other drawings during the month of the previous credit	
credit_card_balance.csv	AMT_DRAWINGS_POS_CURRENT	Amount drawing or buying goods during the month of the previous credit	
credit_card_balance.csv	AMT_INST_MIN_REGULARITY	Minimal installment for this month of the previous credit	
credit_card_balance.csv	AMT_PAYMENT_CURRENT	How much did the client pay during the month on the previous credit	
credit_card_balance.csv	AMT_PAYMENT_TOTAL_CURRENT	How much did the client pay during the month in total on the previous credit	
credit_card_balance.csv	AMT_RECEIVABLE_PRINCIPAL	Amount receivable for principal on the previous credit	
credit_card_balance.csv	AMT_RECIVABLE	Amount receivable on the previous credit	
credit_card_balance.csv	AMT_TOTAL_RECEIVABLE	Total amount receivable on the previous credit	
credit_card_balance.csv	CNT_DRAWINGS_ATM_CURRENT	Number of drawings at ATM during this month on the previous credit	
credit_card_balance.csv	CNT_DRAWINGS_CURRENT	Number of drawings during this month on the previous credit	

credit_card_balance.csv	CNT_DRAWINGS_OTHER_CURRENT	Number of other drawings during this month on the previous credit	
credit_card_balance.csv	CNT_DRAWINGS_POS_CURRENT	Number of drawings for goods during this month on the previous credit	
credit_card_balance.csv	CNT_INSTALMENT_MATURE_CUM	Number of paid installments on the previous credit	
credit_card_balance.csv	NAME_CONTRACT_STATUS	Contract status (active signed,...) on the previous credit	
credit_card_balance.csv	SK_DPD	DPD (Days past due) during the month on the previous credit	
credit_card_balance.csv	SK_DPD_DEF	DPD (Days past due) during the month with tolerance (debts with low loan amounts are ignored) of the previous credit	
previous_application.csv	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)	hashed
previous_application.csv	SK_ID_CURR	ID of loan in our sample	hashed
previous_application.csv	NAME_CONTRACT_TYPE	Contract product type (Cash loan, consumer loan [POS] ,...) of the previous application	
previous_application.csv	AMT_ANNUITY	Annuity of previous application	
previous_application.csv	AMT_APPLICATION	For how much credit did client ask on the previous application	
previous_application.csv	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	
previous_application.csv	AMT_DOWN_PAYMENT	Down payment on the previous application	
previous_application.csv	AMT_GOODS_PRICE	Goods price of good that client asked for (if applicable) on the previous application	

previous_application.csv	WEEKDAY_APPR_PROCESS_START	On which day of the week did the client apply for previous application	
previous_application.csv	HOUR_APPR_PROCESS_START	Approximately at what day hour did the client apply for the previous application	rounded
previous_application.csv	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	
previous_application.csv	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	
previous_application.csv	NFLAG_MICRO_CASH	Flag Micro finance loan	
previous_application.csv	RATE_DOWN_PAYMENT	Down payment rate normalized on previous credit	normalized
previous_application.csv	RATE_INTEREST_PRIMARY	Interest rate normalized on previous credit	normalized
previous_application.csv	RATE_INTEREST_PRIVILEGED	Interest rate normalized on previous credit	normalized
previous_application.csv	NAME_CASH_LOAN_PURPOSE	Purpose of the cash loan	
previous_application.csv	NAME_CONTRACT_STATUS	Contract status (approved, cancelled, ...) of previous application	
previous_application.csv	DAYS_DECISION	Relative to current application when was the decision about previous application made	time only relative to the application
previous_application.csv	NAME_PAYMENT_TYPE	Payment method that client chose to pay for the previous application	
previous_application.csv	CODE_REJECT_REASON	Why was the previous application rejected	
previous_application.csv	NAME_TYPE_SUITE	Who accompanied client when applying for the previous application	
previous_application.csv	NAME_CLIENT_TYPE	Was the client old or new client when applying for the previous application	
previous_application.csv	NAME_GOODS_CATEGORY	What kind of goods did the client apply for in the previous application	
previous_application.csv	NAME_PORTFOLIO	Was the previous application for CASH, POS, CAR, …	
previous_application.csv	NAME_PRODUCT_TYPE	Was the previous application x-sell o walk-in	

previous_application.csv	CHANNEL_TYPE	Through which channel we acquired the client on the previous application	
previous_application.csv	SELLERPLACE_AREA	Selling area of seller place of the previous application	
previous_application.csv	NAME_SELLER_INDUSTRY	The industry of the seller	
previous_application.csv	CNT_PAYMENT	Term of previous credit at application of the previous application	
previous_application.csv	NAME_YIELD_GROUP	Grouped interest rate into small medium and high of the previous application	grouped
previous_application.csv	PRODUCT_COMBINATION	Detailed product combination of the previous application	
previous_application.csv	DAYS_FIRST_DRAWING	Relative to application date of current application when was the first disbursement of the previous application	time only relative to the application
previous_application.csv	DAYS_FIRST_DUE	Relative to application date of current application when was the first due supposed to be of the previous application	time only relative to the application
previous_application.csv	DAYS_LAST_DUE_1ST_VERSION	Relative to application date of current application when was the first due of the previous application	time only relative to the application
previous_application.csv	DAYS_LAST_DUE	Relative to application date of current application when was the last due date of the previous application	time only relative to the application
previous_application.csv	DAYS_TERMINATION	Relative to application date of current application when was the expected termination of the previous application	time only relative to the application
previous_application.csv	NFLAG_INSURED_ON_APPROVAL	Did the client requested insurance during the previous application	
installments_payments.csv	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)	hashed
installments_payments.csv	SK_ID_CURR	ID of loan in our sample	hashed

installments_payments.csv	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	
installments_payments.csv	NUM_INSTALMENT_NUMBER	On which installment we observe payment	
installments_payments.csv	DAYS_INSTALMENT	When the installment of previous credit was supposed to be paid (relative to application date of current loan)	time only relative to the application
installments_payments.csv	DAYS_ENTRY_PAYMENT	When was the installments of previous credit paid actually (relative to application date of current loan)	time only relative to the application

Table 14: Description of dataset variables

8 References

Kaggle. (2018). Home Credit Default Risk. Retrieved from https://www.kaggle.com/competitions/home-credit-default-risk/data