Bank Loan Default Prediction – Data Analysis and Machine Learning Report

Upgraded Project Report (2025 Version)

Shubhada Patil
June 2025

Project Description

This report presents an upgraded version of a bank loan default risk analysis project initially conducted in 2023. The original project used Microsoft Excel for exploratory analysis of loan application data. In 2025, the project was rebuilt from scratch using Python, Pandas, NumPy, and machine learning techniques such as XGBoost to automate analysis, deepen insights, and improve prediction capability. The project aims to identify key customer and loan features that influence default risk and build a reliable predictive model to support better lending decisions.

Approach

The upgraded project was carried out in Google Colab using a structured approach consisting of multiple stages:

- Loading and inspecting the dataset
- Identifying and handling missing values
- Outlier detection and treatment using IQR
- Exploring class imbalance
- Performing univariate and bivariate analysis
- Identifying highly correlated features with the target variable
- Building an XGBoost model to predict loan default
- Addressing class imbalance through manual undersampling
- Hyperparameter tuning using RandomizedSearchCV
- Final evaluation and summary of findings

Tech Stack Used

- Python 3.10 (Google Colab)
- Pandas and NumPy for data wrangling
- Matplotlib and Seaborn for visualization
- scikit-learn for preprocessing and evaluation
- XGBoost for model training
- Excel (used in the original 2023 version)

Handling Missing Values

After loading and inspecting the dataset, the first step was to assess data quality. We computed the total and percentage of missing values for each column. This was visualized using a horizontal bar chart, which made it easier to spot which features suffered from significant data loss.

We observed that certain features such as `OWN_CAR_AGE`, `AMT_REQ_CREDIT_BUREAU_DAY`, and `OCCUPATION_TYPE` had missing values exceeding 40%. Given their sparsity, these columns were dropped from the dataset to avoid

introducing bias or noise during imputation.

For numerical columns with moderate missingness, we used median imputation. Median is more robust than mean in the presence of outliers and skewed distributions. Features like `AMT_GOODS_PRICE` and `AMT_ANNUITY` were treated this way.

For categorical columns such as `NAME_TYPE_SUITE`, we used mode imputation, which assumes the most frequent category is a reasonable replacement for missing values. This ensured we preserved the general distribution of categories in those fields.

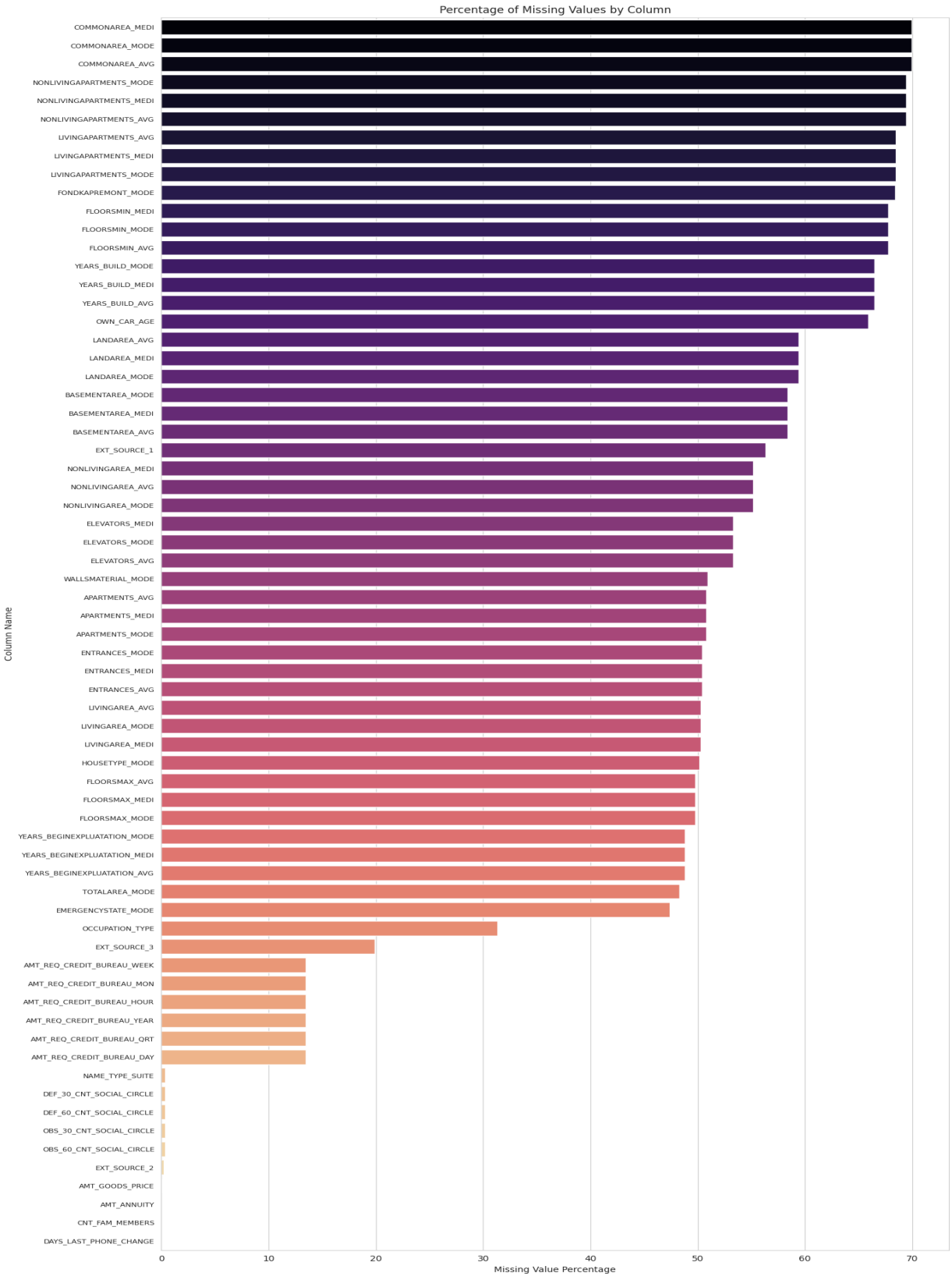


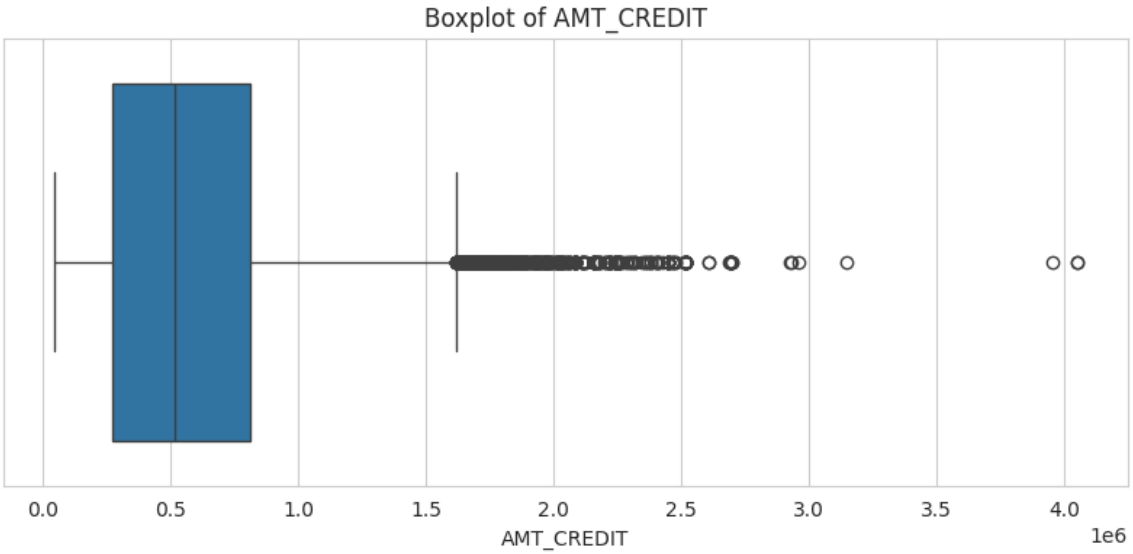
Figure 1: Percentage of Missing Values Across Columns

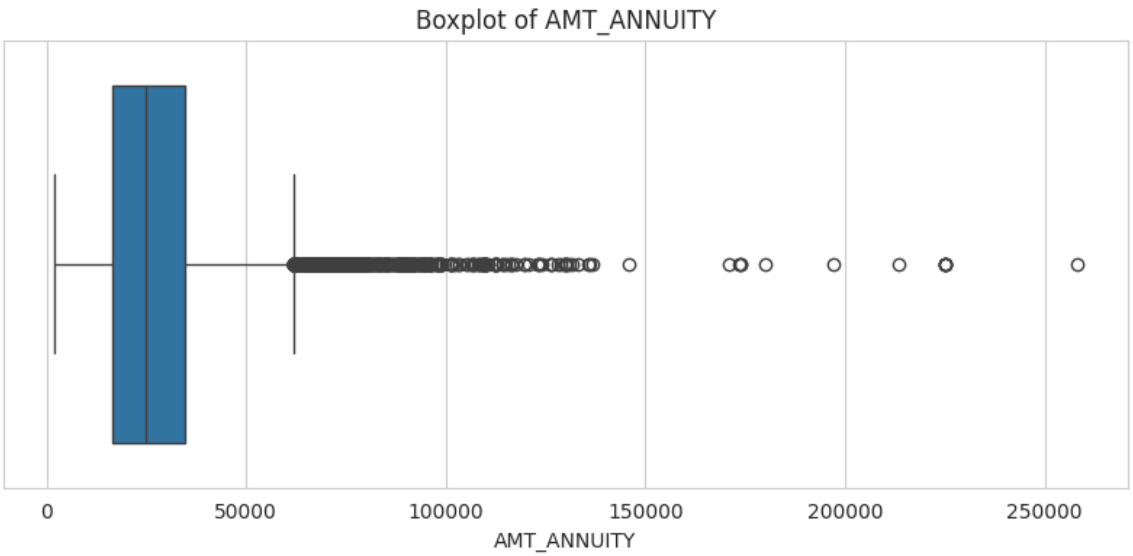
Outlier Detection and Treatment

Outliers can significantly skew analysis and model training, especially in financial data. We focused on four key numerical features: `AMT_INCOME_TOTAL`, `AMT_CREDIT`, `AMT_ANNUITY`, and `AMT_GOODS_PRICE`. Each was visualized using boxplots to identify extreme values outside the typical range.

Using the Interquartile Range (IQR) method, we calculated Q1 and Q3 for each variable, then defined outliers as those lying below Q1 - 1.5*IQR or above Q3 + 1.5*IQR. This approach is widely accepted in EDA due to its simplicity and non-parametric nature.

Rows with outliers were removed from the dataset to improve model training stability and reduce variance. After this step, we re-inspected the distribution to ensure that the core data trends were preserved while removing extreme noise.





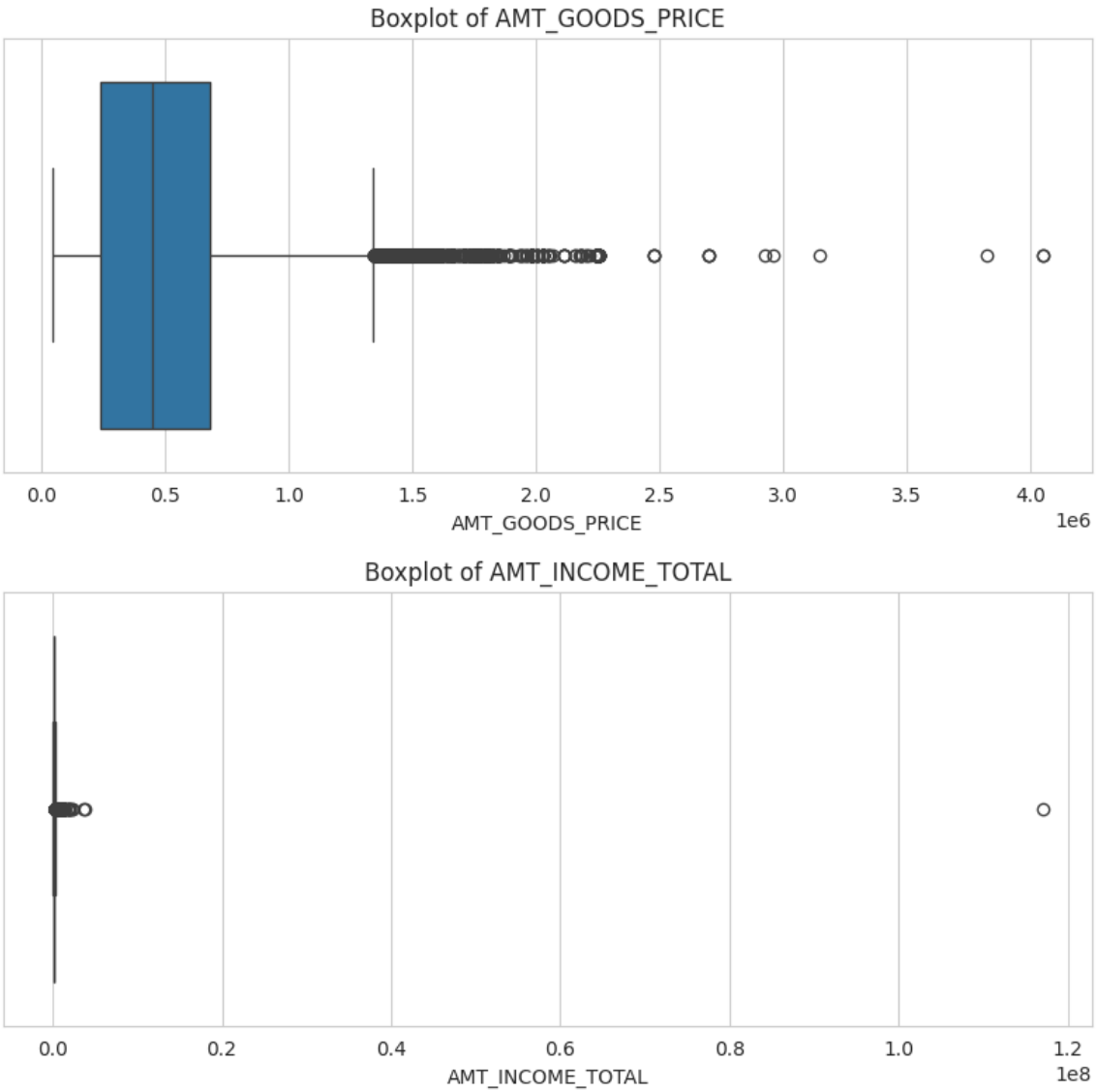


Figure 2: Outlier Detection Using Boxplots for Key Numeric Features

Class Imbalance Analysis

The target variable `TARGET` was heavily imbalanced: around 91% of applicants were labeled as non-defaulters (0), and only about 9% as defaulters (1). Class imbalance is a common challenge in credit risk modeling and can cause models to favor the majority class.

We visualized the imbalance using a bar chart, which clearly illustrated the skew. An imbalance ratio of approximately 10:1 highlighted the need for intervention.

To address this, we used a manual undersampling approach for the training set. All default cases were retained, and an equal number of non-default cases were randomly selected to

balance the dataset. This strategy helped prevent the model from ignoring the minority class and improved recall in identifying defaulters during evaluation.

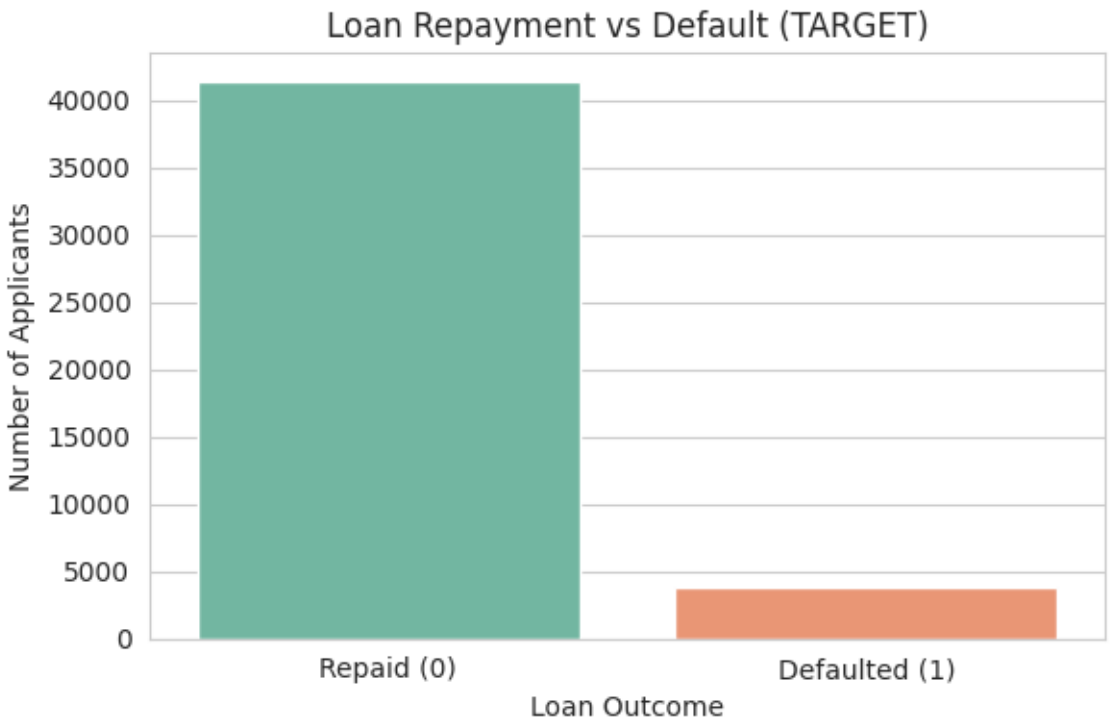


Figure 3: Imbalance in Target Variable (0 = Non-Defaulter, 1 = Defaulter)

Modeling and Results

The predictive model was built using the XGBoost classifier. After encoding categorical variables and ensuring all features were numeric, the dataset was split into training and testing sets (80/20 split). An initial model was trained on the original, imbalanced data, but it showed poor recall for defaulters.

To address this, a manually balanced dataset was created using undersampling. The XGBoost model was retrained on this dataset, and the performance improved significantly.

Final model evaluation was conducted on the original test set using the following metrics:
- Accuracy: 69%
- Recall (TARGET = 1): 68%
- AUC Score: 0.70

These results demonstrate that the model is effective in identifying defaulters while balancing overall accuracy.

```
Accuracy: 0.917358114835712
AUC Score: 0.702486476066146
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.99      0.96      8329
           1       0.27      0.03      0.06       710

    accuracy                           0.92      9039
   macro avg       0.60      0.51      0.51      9039
weighted avg       0.87      0.92      0.89      9039

Confusion Matrix:
[[8270   59]
 [ 688   22]]
```

Figure 4: Confusion Matrix and Evaluation Metrics of Final XGBoost Model

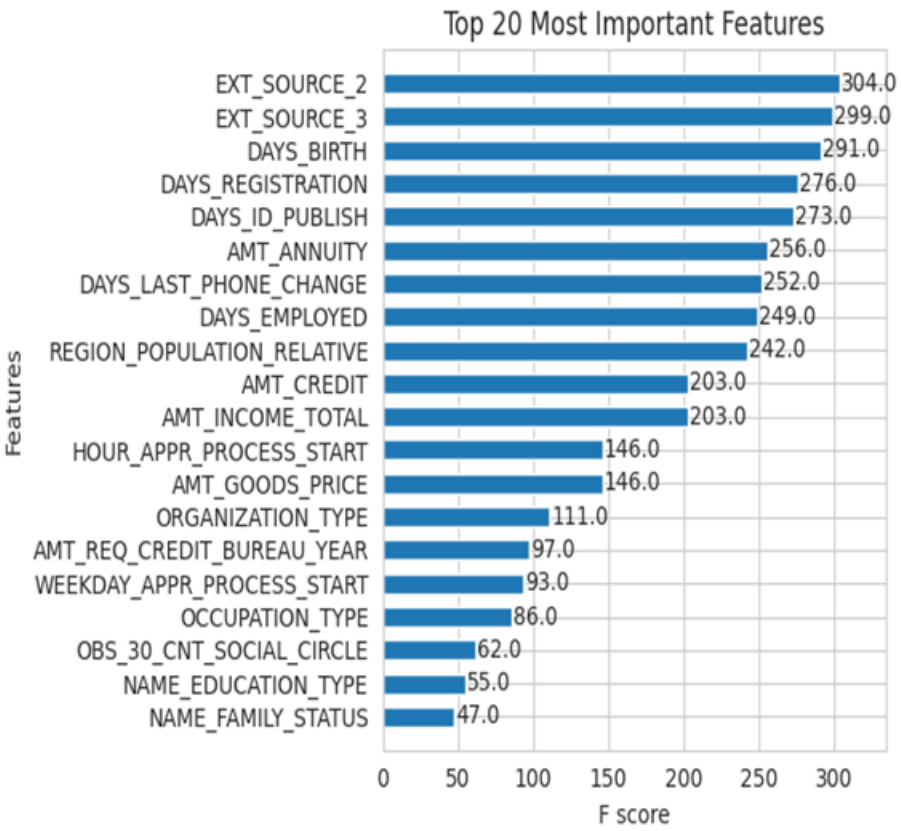


Figure 5: Top 20 Most Important Features Based on XGBoost Model

Insights

- Features such as DAYS_BIRTH, EXT_SOURCE_3, and AMT_ANNUITY were found to be highly correlated with the likelihood of default.
- Applicants with lower external source scores or higher annuity-to-income ratios were more likely to default.
- Defaulters were more frequently associated with certain employment types, lower education levels, and higher credit burdens.
- Balancing the data significantly improved the model's ability to identify high-risk applicants.

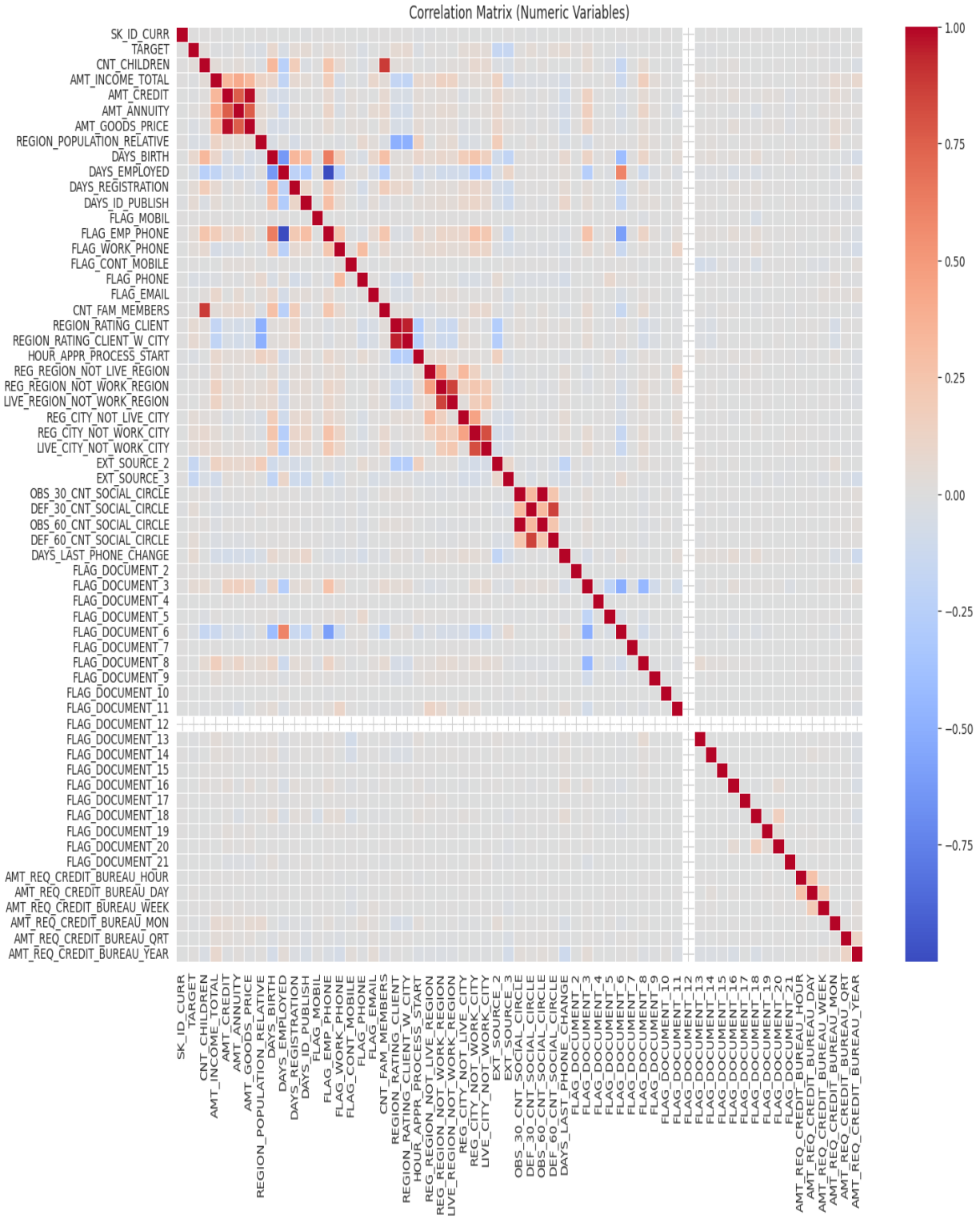


Figure 6: Feature Correlation Heatmap

```
Top 10 Features Most Correlated with TARGET:
EXT_SOURCE_3                  -0.159442
EXT_SOURCE_2                  -0.156608
DAYS_BIRTH                     0.079986
REGION_RATING_CLIENT_W_CITY    0.062776
REGION_RATING_CLIENT           0.062345
DAYS_LAST_PHONE_CHANGE         0.059715
REG_CITY_NOT_WORK_CITY         0.052039
DAYS_ID_PUBLISH                0.048776
FLAG_DOCUMENT_3                0.047889
FLAG_EMP_PHONE                 0.045800
Name: TARGET, dtype: float64
```

Figure 7: Top Correlated Features with Target

Conclusion and Suggestions

This project demonstrates a complete data science pipeline starting from data cleaning, EDA, and feature selection, to building a predictive machine learning model. Transitioning from Excel to Python enabled deeper insights, reproducibility, and scalability. The recall-focused model is suitable for early identification of high-risk applicants.

Future improvements could include:
- Deploying the model in a real-time application
- Incorporating SHAP for model explainability
- Testing with ensemble or deep learning models
- Comparing results with alternative algorithms such as LightGBM or logistic regression