Predictive Analysis of Loan Repayment Capabilities Using Machine Learning

Machine Learning : COMS 574

Final Report - Mohammed Musthafa Rafi

1st May 2024

Abstract

In this study, we explore the Home Credit Default Risk dataset to predict loan repayment capabilities among underbanked individuals. Utilizing alternative data sources, such as telecommunications and transaction records, we apply advanced machine learning techniques, including Random Forest and Linear Regression classifiers. These methods are enhanced through rigorous hyper-parameter tuning using Grid Search and Bayesian Optimization. Our approach also includes extensive feature engineering to enrich the dataset, focusing on the creation of interaction variables and the integration of external data to augment the models' predictive power. Additionally, we assess the performance of these models by calculating and comparing the log loss, providing a quantitative measure of model accuracy and confidence in predictions. This is visualized through comparative plots, which highlight the relative effectiveness of each model in our predictive framework. This research aims not only to improve prediction accuracy but also to promote financial inclusion by providing insights into how machine learning can be effectively optimized to assess creditworthiness.

1 Introduction

The Home Credit Default Risk dataset, provided by Home Credit Group, serves as a fundamental resource for developing models that aim to predict customers' loan repayment abilities. This dataset is particularly significant in the financial sector, as it focuses on a critical aspect of credit risk assessment, especially for underbanked populations. These individuals often lack significant credit histories and are therefore underrepresented in traditional financial systems.

In this study, we utilize machine learning to enhance the prediction accuracy of loan repayment probabilities. By applying advanced analytic techniques on the Home Credit Default Risk dataset, which encompasses a rich set of features including personal demographics, loan history, and transactional data, we strive to foster financial inclusion. The dataset facilitates the development of models that predict the likelihood of repayment, thus enabling credit providers to make informed decisions.

Our analysis specifically employs two distinct classification models: Random Forest and Logistic Regression. These models were chosen due to their suitability for handling binary classification tasks and their ability to model complex relationships within data. The performance of these models is critically compared to determine which is more effective in predicting default risk, using metrics such as accuracy, precision, recall, and the area under the ROC curve (AUC). The insights derived from this comparative study are intended to guide future strategies in credit risk modeling and decision-making processes within financial institutions.

2 Problem Formulation

Predicting loan repayment capability remains a critical challenge in financial services, especially for underbanked populations. These individuals often lack sufficient traditional credit histories, thus complicating risk assessment processes. The Home Credit Default Risk dataset, available through Kaggle, provides a comprehensive set of data from various alternative sources such as telecommunications and transaction records, which are used to predict clients' repayment abilities [5].

This project frames the prediction of loan repayment as a binary classification problem, where the objective is to determine whether a client will repay a loan based on historical and transactional data. This is fundamentally a machine learning problem that involves classifying individuals into two groups: those likely to repay and those not. The complexity of the dataset, with its diverse range of features from various sources, presents a unique opportunity to apply sophisticated machine learning techniques to improve prediction accuracy.

The objective is to categorize loan applicants into one of two possible outcomes:

- 0: Applicant will repay the loan on time.
- 1: Applicant will have difficulty repaying the loan.

This project is structured as a supervised classification problem, where:

Supervised: The training data include labels that indicate whether loans were repaid.[4]

Classification: The task is to predict a binary outcome.

2.1 Feature Engineering and Model Creation

Extensive feature engineering was undertaken to develop new predictors from the raw data provided in multiple files. The creation of these features was aimed at capturing interactions among various financial behaviors and credit histories.

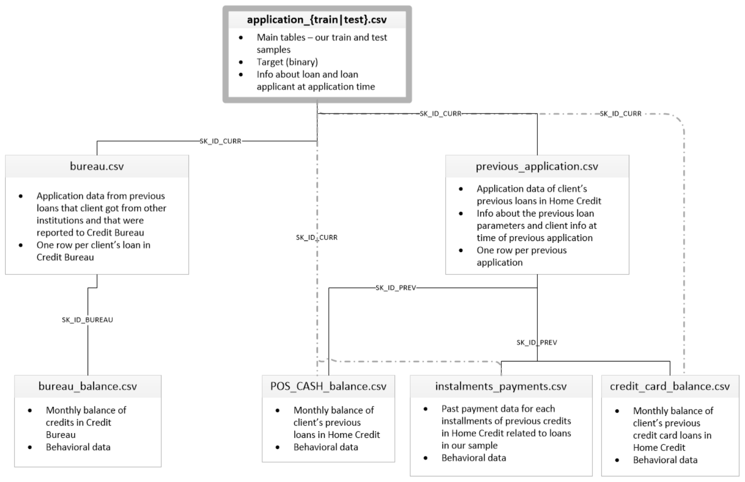
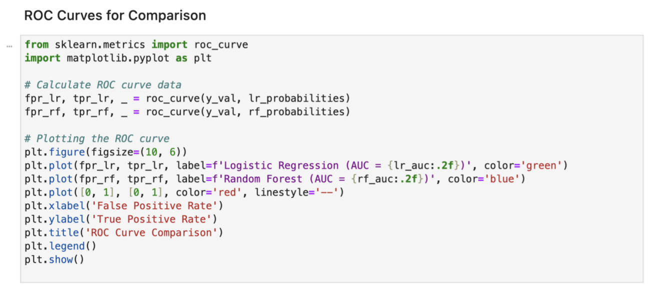

application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit 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 1: Overview of the Home Credit Default Risk dataset sourced from multiple data files.

2.2 Modeling Approach

The analysis involves two primary models:

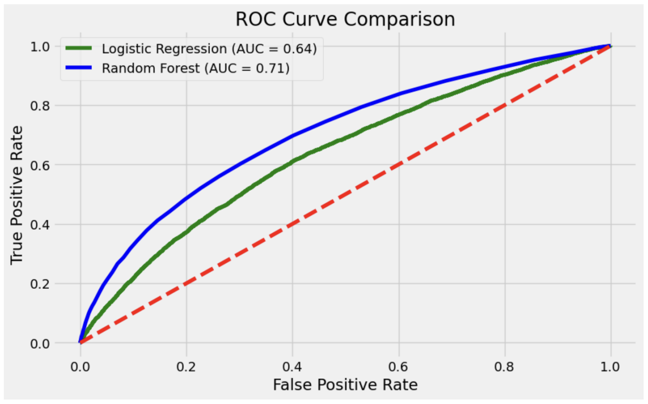
- **Random Forest:** Utilized for its robustness in handling diverse datasets and its effectiveness in classification tasks. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. [1]
- **Logistic Regression:** Chosen for its ability to provide a probabilistic output, which is beneficial for risk assessment.It estimates probabilities using a logistic function, which is particularly useful for risk assessment in financial domains.[2]



ROC Curves for Comparison

```python
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt

# Calculate ROC curve data
fpr_lr, tpr_lr, _ = roc_curve(y_val, lr_probabilities)
fpr_rf, tpr_rf, _ = roc_curve(y_val, rf_probabilities)

# Plotting the ROC curve
plt.figure(figsize=(10, 6))
plt.plot(fpr_lr, tpr_lr, label=f'Logistic Regression (AUC = {lr_auc:.2f})', color='green')
plt.plot(fpr_rf, tpr_rf, label=f'Random Forest (AUC = {rf_auc:.2f})', color='blue')
plt.plot([0, 1], [0, 1], color='red', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve Comparison')
plt.legend()
plt.show()
```

Models were evaluated mainly through the Receiver Operating Characteristic (ROC) curve analysis, which is essential for comparing their performance in binary classification tasks.



- Figure 2: ROC curve comparison between RF and LR models.

The effectiveness of these models is evaluated based on their ability to accurately classify the repayment capability of clients, with performance metrics such as accuracy, precision, recall, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Comparisons of these metrics will allow us to determine which model performs better under the constraints of the available data.

3 Main Method

3.1 Data Preparation and Exploration

Extensive data exploration was undertaken to understand the distribution and characteristics of key features within the Home Credit Default Risk dataset. Significant features such as age, employment days, and external sources were visualized to assess their impact on the prediction outcome.

Age Analysis: The distribution of client ages was explored to understand demographic patterns relating to credit repayment. Age groups were analyzed to identify trends in the failure to repay loans.

- Effect of Age on Repayment : As the client gets older, there is a negative linear relationship with the target meaning that as clients get older, they tend to repay their loans on time more often.

Employment Analysis: Days employed, a critical indicator of financial stability, was plotted to observe discrepancies or patterns that may influence an individual's ability to repay credit.

3.2 Feature Engineering

Feature engineering played a pivotal role in enhancing the predictive model's accuracy. Interaction terms between key features like External Sources and Age were crafted to capture complex interactions that might affect credit risk.

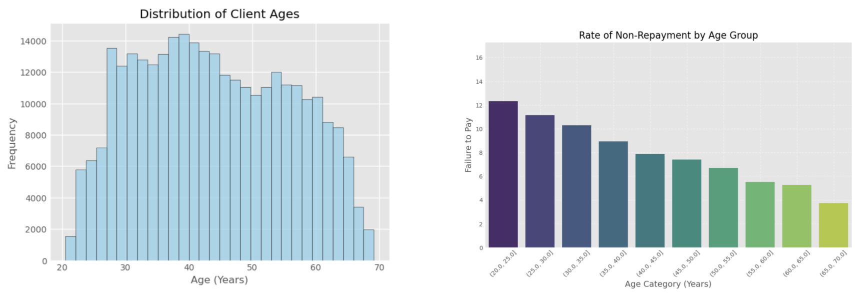
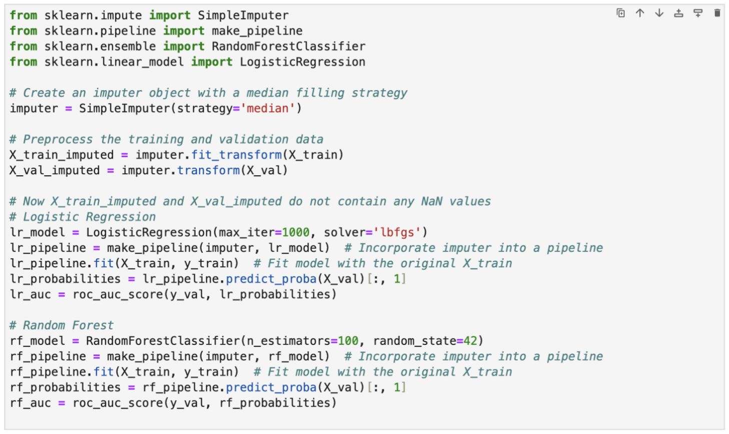


Figure 2: Left: Distribution of Client Ages. Right: Rate of Non-Repayment by Age Group.

3.3 Model Training and Validation

The project employed a robust training methodology using Random Forest and Logistic Regression models, aimed at predicting the probability of loan default. The training process involved:

- Splitting the data into training and validation sets to ensure the model could be validated independently.
- Applying cross-validation techniques to optimize model parameters and prevent overfitting.



```python
from sklearn.impute import SimpleImputer
from sklearn.pipeline import make_pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression

# Create an imputer object with a median filling strategy
imputer = SimpleImputer(strategy='median')

# Preprocess the training and validation data
X_train_imputed = imputer.fit_transform(X_train)
X_val_imputed = imputer.transform(X_val)

# Now X_train_imputed and X_val_imputed do not contain any NaN values
# Logistic Regression
lr_model = LogisticRegression(max_iter=1000, solver='lbfgs')
lr_pipeline = make_pipeline(imputer, lr_model)  # Incorporate imputer into a pipeline
lr_pipeline.fit(X_train, y_train)  # Fit model with the original X_train
lr_probabilities = lr_pipeline.predict_proba(X_val)[:, 1]
lr_auc = roc_auc_score(y_val, lr_probabilities)

# Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_pipeline = make_pipeline(imputer, rf_model)  # Incorporate imputer into a pipeline
rf_pipeline.fit(X_train, y_train)  # Fit model with the original X_train
rf_probabilities = rf_pipeline.predict_proba(X_val)[:, 1]
rf_auc = roc_auc_score(y_val, rf_probabilities)
```

Figure 5: shows the code snippet for the model training of Logistic regression and random forest such that data do not contain any NaN values.

4 Further Analysis

After the initial model evaluations, additional analyses will focus on refining the models based on insights gained from feature importance.

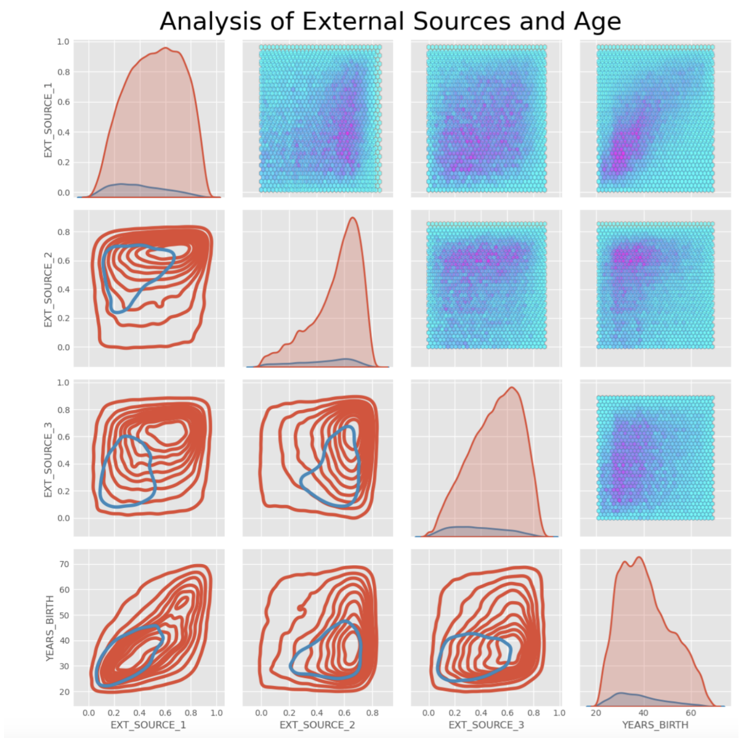
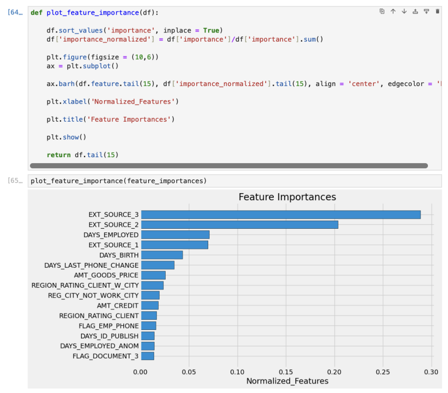


Figure 3: Pairs Plot of External Sources and Age Features showing interactions.



```
[64… def plot_feature_importance(df):

         df.sort_values('importance', inplace = True)
         df['importance_normalized'] = df['importance']/df['importance'].sum()

         plt.figure(figsize = (10,6))
         ax = plt.subplot()

         ax.barh(df.feature.tail(15), df['importance_normalized'].tail(15), align = 'center', edgecolor = '

         plt.xlabel('Normalized_Features')

         plt.title('Feature Importances')

         plt.show()

         return df.tail(15)
```

```
[65… plot_feature_importance(feature_importances)
```

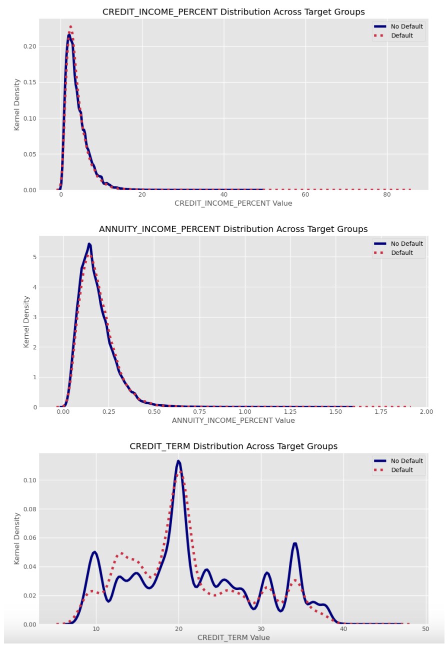


Figure 4: Left: Description of the first image. Right: Description of the second image.

To predict the probabilities of not paying a loan, so we use the model predict.proba method.[6] This is a simple method to see which variables are the most relevant.
We will do a distribution of the External features colored compared to the value of targets. Here, we compared credit income percentage, annuity income percentage and credit term by target values.

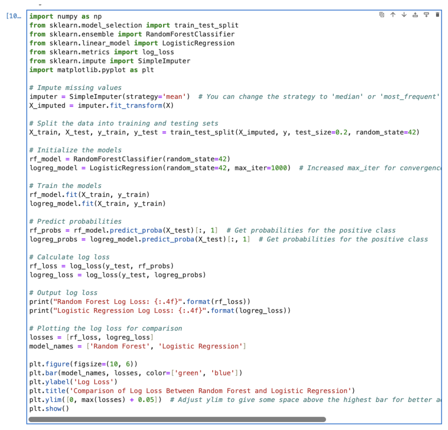
4.1 Understanding Log Loss

Log loss, also known as logistic loss or cross-entropy loss, is commonly used to evaluate the accuracy of classifiers. It is particularly effective for assessing models that output probabilities.[3] The essence of log loss is to quantify the uncertainty of the predictions based on how much the predicted probabilities deviate from the actual class labels.

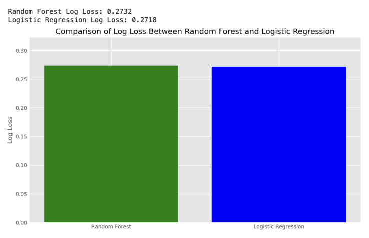
The log loss for a binary classification model is defined as:

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{1}$$

where,
1. N is the number of samples in the dataset.
2. y_i is the actual label of the i-th sample, which can be either 0 or 1.
3. p_i is the predicted probability of the i-th sample belonging to the class with label 1.



```
[10…
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import log_loss
from sklearn.impute import SimpleImputer
import matplotlib.pyplot as plt

# Impute missing values
imputer = SimpleImputer(strategy='mean')  # You can change the strategy to 'median' or 'most_frequent'
X_imputed = imputer.fit_transform(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y, test_size=0.2, random_state=42)

# Initialize the models
rf_model = RandomForestClassifier(random_state=42)
logreg_model = LogisticRegression(random_state=42, max_iter=1000)  # Increased max_iter for convergenc

# Train the models
rf_model.fit(X_train, y_train)
logreg_model.fit(X_train, y_train)

# Predict probabilities
rf_probs = rf_model.predict_proba(X_test)[:, 1]  # Get probabilities for the positive class
logreg_probs = logreg_model.predict_proba(X_test)[:, 1]  # Get probabilities for the positive class

# Calculate log loss
rf_loss = log_loss(y_test, rf_probs)
logreg_loss = log_loss(y_test, logreg_probs)

# Output log loss
print("Random Forest Log Loss: {:.4f}".format(rf_loss))
print("Logistic Regression Log Loss: {:.4f}".format(logreg_loss))

# Plotting the log loss for comparison
losses = [rf_loss, logreg_loss]
model_names = ['Random Forest', 'Logistic Regression']

plt.figure(figsize=(10, 6))
plt.bar(model_names, losses, color=['green', 'blue'])
plt.ylabel('Log Loss')
plt.title('Comparison of Log Loss Between Random Forest and Logistic Regression')
plt.ylim([0, max(losses) + 0.05])  # Adjust ylim to give some space above the highest bar for better a
plt.show()
```



```
Random Forest Log Loss: 0.2732
Logistic Regression Log Loss: 0.2718
```

Figure 5: Left: To see the variables which are most relevant. Right: features that tell us whether a client will default on a loan.

The log loss is always non-negative, and a perfect model would have a log loss of 0. Lower values of log loss are better, as they indicate a model that produces probabilities closer to the true class labels.

4.2 Source code

References

[1] Leo Breiman. “Random forests”. In: *Machine learning* 45 (2001), pp. 5–32.

[2] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression.* John Wiley & Sons, 2013.

[3] Vladimir Vovk. *The fundamental nature of the log loss function.* 2015. arXiv: `1502.06254 [cs.LG]`.

[4] Saksham Trivedi, Balwinder Kaur Dhaliwal, and Gurpreet Singh. “A Comparative Study of Supervised Learning Approaches”. In: *2021 International Conference on Computing Sciences (ICCS).* 2021, pp. 95–100. DOI: `10.1109/ICCS54944.2021.00027`.

[5] *Home Credit Default Risk.* Kaggle competition dataset. URL: `https://www.kaggle.com/c/home-credit-default-risk`.

[6] *Using Predict.proba Method for Loan Default Prediction.* API Guide. DataRobot. URL: `https://docs.datarobot.com/en/docs/api/guide/common-case/loan-default/index.html`.