CSE587A Data Intensive Computing - Project Phase 3 Report

Title: Data-Driven Loan Default Prediction and Financial Analysis

Team members: Juseung Lee, Venkata Amballa

1. Summary of Phase 1

Please refer to the 'report_phase1.pdf' for the full report of Phase 1.

1.1. Problem Statement

Loan defaults raise significant issues in the financial sector that affect lenders, borrowers, and the overall economic environment. It results in financial losses for lenders and impedes borrowers' financial stability. Moreover, loan defaults can also lead to a loss of confidence in the lending ecosystem and can disrupt credit access for individuals and businesses. The main objective of this project is to develop predictive models that can address these challenges by accurately predicting key aspects of the risk of loan issuance.

1.2. Data Source and Description

The loan default dataset used in this project is available on the Kaggle [1]. The loan default dataset provides a comprehensive overview of loan repayment patterns across various regions throughout 2019. This dataset is a significant resource for analyzing and understanding loan repayment behaviors. It includes several deterministic factors such as loan_purpose, loan_amount, property_value, and Credit_Score. In this dataset, the 'Status' feature indicates the loan status that value 1 identifies loan granted and value 0 identifies loan not granted. Using the data from the 'Status' feature, we can make assumptions about whether a loan applicant will default on the loan. For example, 'Status=0' indicates the loan is not sanctioned and the loan applicant will default on the loan.

1.3. Data Cleaning/Processing

- **Normalizing numerical features**

  We applied robust scaling by using 'RobustScaler()' to the numerical features in the dataset to maintain all the features in a consistent range.

1.4. Cleaned Data

Cleaned data has been saved to ‘Loan_Default_cleaned.csv’ in phase 1. We renamed ‘Loan_Default_cleaned_downsampled.csv’, and we added a new file named ‘Loan_Default_cleaned_all.csv’ that contains all cleaned data without downsampling.

1.5. Changes we’ve made in Phase 2

We made the following changes in the phase 1 code:

- Total_units: We realized that total_units is a scaler, so we converted it to an integer.
- Age: This value is one-hot encoded. We resolved this by considering this feature as an integer.

2. Phase 2

Please refer to the ‘report_phase2.pdf’ for the full report of Phase 2.

2.1. Algorithm - Decision Tree

We selected a Decision Tree as our final model based on the following criteria:

1. **Model Complexity:** Decision Trees are renowned for their simplicity, as they select only the most relevant features.
2. **Interpretability:** Among machine learning models, Decision Trees stand out for their interpretability, making it easy to visualize the decisions made by the model.
3. **Performance:** As a non-parametric model, Decision Trees exhibit exceptional speed, allowing for efficient processing

In terms of tuning the model, we applied robust scaling by using ‘RobustScaler()’ to normalize numerical features in Phase 1 and split the data into training and testing sets to prevent overfitting.

3. Phase 3

In this phase, we’ve built a data product based on Phase 1 and Phase 2 that enables a user to interact with it and gain insight into the problem statement we’ve set out to solve.

3.1. Tuning our Model

In terms of tuning the model, we decided to use our baseline model as it achieves better accuracy. However, we used cross-validation scoring to select the best model that generalizes well for our dataset.

3.2. Deploying Our ML Model

3.2.1. Preprocessing

To utilize the preprocessing logic developed in Phase 1, which includes handling missing or NULL values, removing unnecessary columns, scaling numerical features, and encoding categorical variables, we generated pickle files for the RobustScaler and OneHotEncoder. This allows us to perform the same preprocessing steps consistently, ensuring that we can reproduce the results achieved in Phase 2.

3.2.2. Model Loading

Similarly, we generated a pickle file for our best model, the Decision Tree. We utilized pickle files in our preprocessing steps to prepare the form input before submitting it to our model for prediction. This model generates outputs based on the learned patterns from our training in Phase 2.

3.2.3. API Endpoint for Result

The generated prediction has to be sent back as a response to the user. The ***/submit*** endpoint is designed for this purpose, this receives the model's predictions and sends them to the front end.

3.3. Further Improvements

We acknowledge that there could be a possibility of introducing historical, representational, or other forms of bias during the data collection phase. Further work is required to address these types of biases to make our model more robust and reliable.

3.4. Recommendations related to our problem statement

The target users for our system are banking officials in a financial institution that lends loans to people. Our system can help the target users while deciding whether to give a loan or not. The target users can gain insights about loan default based on their input data, and they can be provided a visualization of age distribution by loan status while interacting in the user-interface. Our system makes it easier for financial institutions to make decisions about the approval of loan applications since it can provide valuable insights into patterns that can inform risk assessment for financial institutions. It can be used for improving decision-making processes and strengthening financial stability.

3.5. Extending this Project

Future developments could include integration with real-time data further enhance the capabilities of our model. Additionally, we can use advanced ML techniques like Reinforcement Learning to dynamically adapt to shifting market conditions. Moreover, developing a mobile app that allows loan officers to anticipate the risk associated with loan defaults.

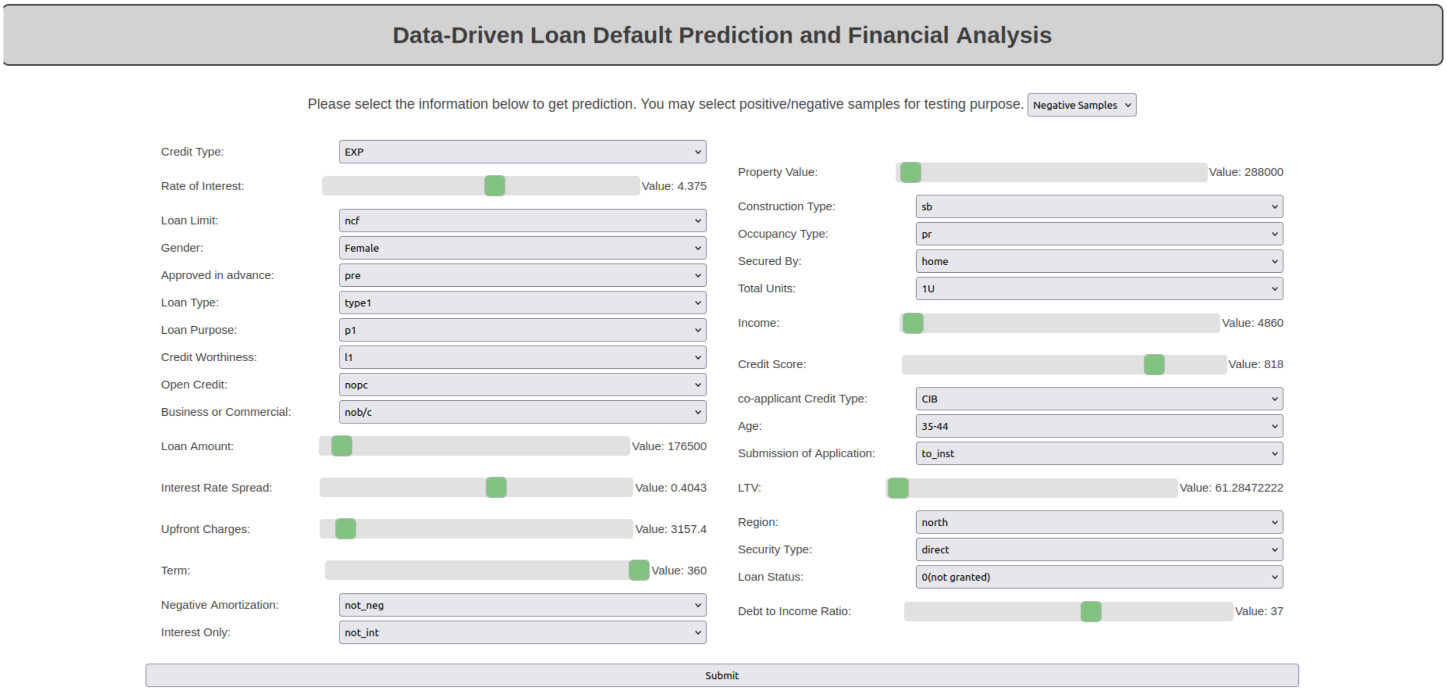
3.6. Usage

Please follow the instructions to run our code: (1) run ***export FLASK_APP=app*** in the src/phase3, and (2) run ***flask run*** in the src/phase3. We used Flask along with Javascript, HTML, CSS, and Python. In app.py, our pickled model is loaded for providing prediction results to a user.

Note: An additional **README.md** file is provided along with our project code, that outlines the steps required to successfully run our project.

3.7. User Interface

This is our landing page, and the following screenshots show after a user submits input data and clicks a button to view a plot that provides some insight to a user.

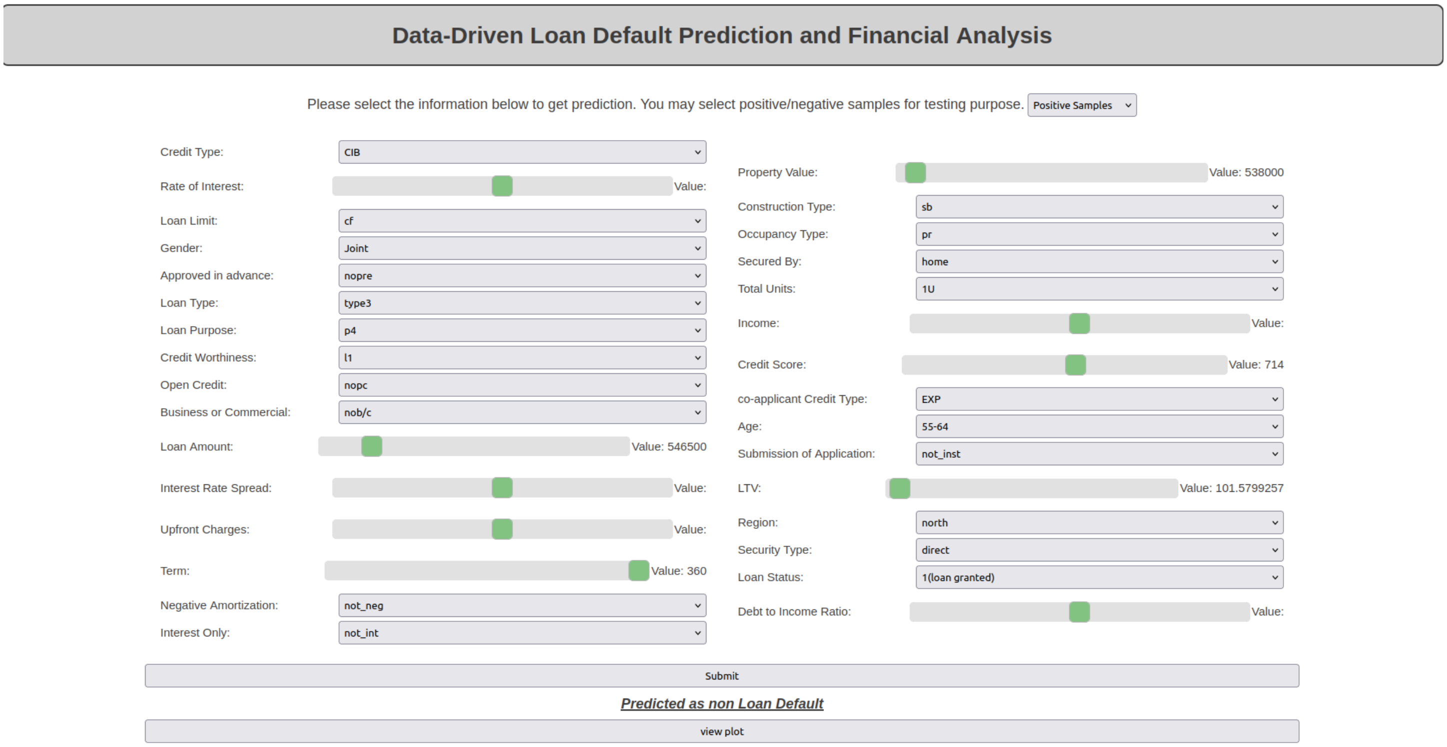


Data-Driven Loan Default Prediction and Financial Analysis

Please select the information below to get prediction. You may select positive/negative samples for testing purpose. Negative Samples

Credit Type: EXP
Rate of Interest: Value: 4.375
Loan Limit: ncf
Gender: Female
Approved in advance: pre
Loan Type: type1
Loan Purpose: p1
Credit Worthiness: l1
Open Credit: nopc
Business or Commercial: nob/c
Loan Amount: Value: 176500
Interest Rate Spread: Value: 0.4043
Upfront Charges: Value: 3157.4
Term: Value: 360
Negative Amortization: not_neg
Interest Only: not_int

Property Value: Value: 288000
Construction Type: sb
Occupancy Type: pr
Secured By: home
Total Units: 1U
Income: Value: 4860
Credit Score: Value: 818
co-applicant Credit Type: CIB
Age: 35-44
Submission of Application: to_inst
LTV: Value: 61.28472222
Region: north
Security Type: direct
Loan Status: 0(not granted)
Debt to Income Ratio: Value: 37



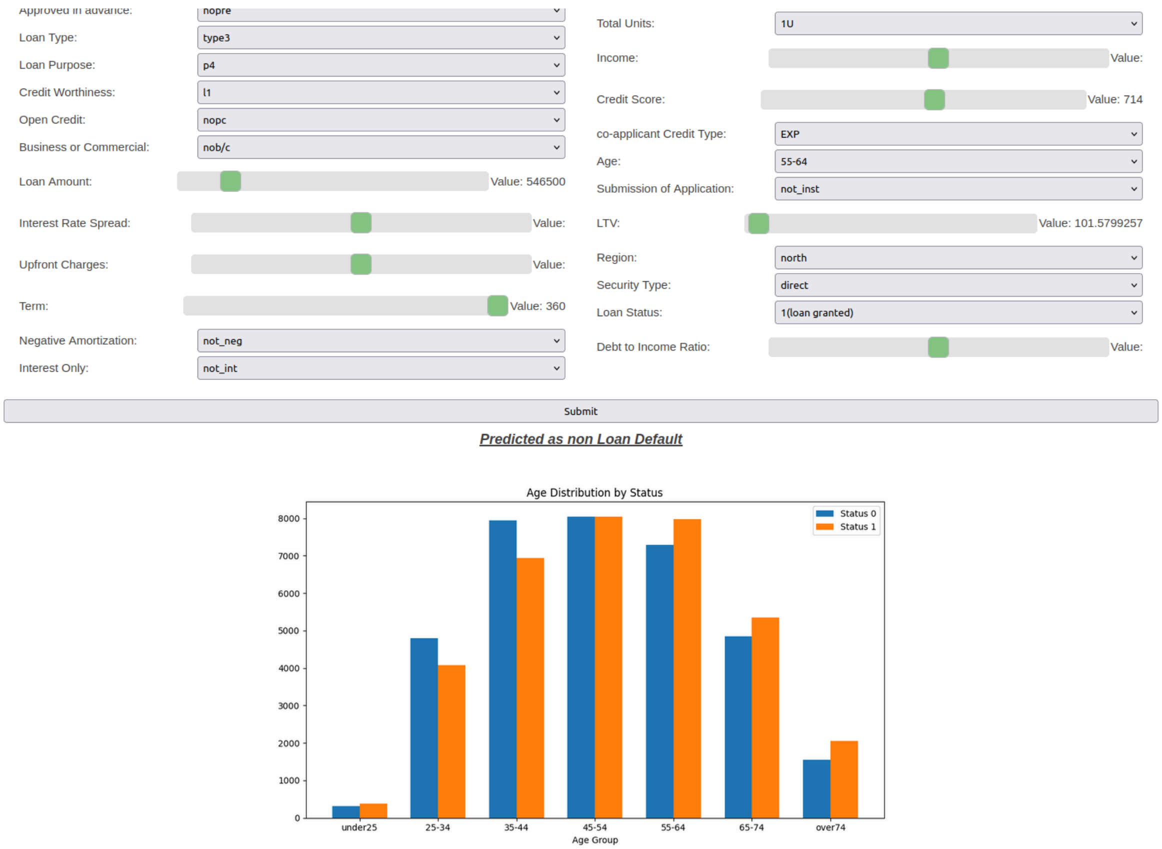
Data-Driven Loan Default Prediction and Financial Analysis

Please select the information below to get prediction. You may select positive/negative samples for testing purpose. Positive Samples

Credit Type: CIB
Rate of Interest: Value:
Loan Limit: cf
Gender: Joint
Approved in advance: nopre
Loan Type: type3
Loan Purpose: p4
Credit Worthiness: l1
Open Credit: nopc
Business or Commercial: nob/c
Loan Amount: Value: 546500
Interest Rate Spread: Value:
Upfront Charges: Value:
Term: Value: 360
Negative Amortization: not_neg
Interest Only: not_int

Property Value: Value: 538000
Construction Type: sb
Occupancy Type: pr
Secured By: home
Total Units: 1U
Income: Value:
Credit Score: Value: 714
co-applicant Credit Type: EXP
Age: 55-64
Submission of Application: not_inst
LTV: Value: 101.5799257
Region: north
Security Type: direct
Loan Status: 1(loan granted)
Debt to Income Ratio: Value:

Predicted as non Loan Default



Approved in advance: nopre
Loan Type: type3
Loan Purpose: p4
Credit Worthiness: l1
Open Credit: nopc
Business or Commercial: nob/c
Loan Amount: Value: 546500
Interest Rate Spread: Value:
Upfront Charges: Value:
Term: Value: 360
Negative Amortization: not_neg
Interest Only: not_int

Total Units: 1U
Income: Value:
Credit Score: Value: 714
co-applicant Credit Type: EXP
Age: 55-64
Submission of Application: not_inst
LTV: Value: 101.5799257
Region: north
Security Type: direct
Loan Status: 1(loan granted)
Debt to Income Ratio: Value:

Predicted as non Loan Default

3.8. Demo

Our recorded demo is included in the zip file. It is a brief demo of our product that explains as follows:

- how it works in the user-interface
- how a user could input their own data in the user-interface
- how it gives feedback and visualization to a user in the user-interface
- how a user could use it to help them solve a problem statement we've set out to solve

It presents the walk-through of our user-interface to show how it communicates with a user.

4. References

[1] https://www.kaggle.com/datasets/yasserh/loan-default-dataset