LOAN DEFAULT PROJECT

BY

APOSTOLOS

STAVROU

TABLE OF CONTENTS

ABSTRACT 3
INTRODUCTION 3
GOAL 4
METHODS 4
RESULTS 6
APPENDIX 8

ABSTRACT

The primary goal of this project is to build a model that can predict whether a loan will default based on a number of selected features. Due to heavy class imbalance the SMOTE oversampling technique will be applied and then the data will be split into training, test and validation subparts. The model was trained on a deep neural network based on PyTorch library, with 3 hidden layers. Data was preprocessed and measures were taken in order to ensure robust and accurate classifications on a rather uniformly distributed dataset.

INTRODUCTION

In finance, a loan is in default when the borrower fails to meet its' legal obligations or conditions. That can happen when a home buyer stops making payments on a mortgage or when a corporation or government fails to pay a bond after the agreed maturity date is due. One of the primary objectives of companies with financial loan services is to decrease payment defaults while ensuring that the individuals are paying back their loans as expected. Using deep learning techniques and tuning hyperparameters the model will be able to predict which loan may default based on various factors such as income, credit score etc. The dataset itself includes variables such as age, income, loan amount, credit score, etc. It is unusually balanced and uniformly distributed across the board with our primary feature (loan defaulting or not) is greatly imbalanced as most of the loans do not end up defaulting and an oversampling technique will be required.

GOAL

The primary objectives of this project are the following:

1) Perform Exploratory Data Analysis which includes handling missing data, encode categorical variables to provide better readability on results, update data types to correct ones, etc.

2) Build a suitable model for predicting loan defaults by calculating the most importance features to use, split the dataset into train, test and validation subsets and pick appropriate loss and optimizing methods.

3) Finetune relevant hyperparameters and test various machine learning techniques to pick the most optimal one.

METHODS

This section includes the various methods and techniques used to clean and organize the original dataset to be eventually usable for analysis and predictions.

Data Preprocessing:

1. No missing values found.
2. Categorical variables encoded: Education, Employment Type, Marital Status, Loan Purpose.
3. Remove Loan ID column as it's an identifier and not a predictive feature.
4. Check for outliers using boxplots and interquartile range.
5. Scale numerical variables using Min Max scaler to improve machine learning calculations.

6. No multicollinearity spotted on the correlation matrix.
7. Target variable does not correlate with any of the features.

Feature Selection:

Random Forest was used to select the most relevant features to predict the target variable. Such features include Interest Rate, Loan Amount, Age, Credit Score, Months Employed, DTI ratio.

Split and Resampling:

SMOTE Resampling technique used to balance both classes as defaulted loans were underrepresented. 70-15-15 Split used for training, testing and validation.

Classification Analysis:

Creation of a neural network with 3 hidden layers, all using ReLU activation function, Binary Cross Entropy Loss for Binary Classification, Adam optimizer. Due to false negatives (model incorrectly predicting a defaulting loan will not default) being more important for our case, a threshold of 0.3 has been applied.

RESULTS

Since the primary objective of the model is to predict whether a loan will default or not with the focus being heavily on correctly predicting defaulting loans, the main metric it should focus on is recall (true positives / (true positives + false negatives).

First, the results of the model without any additional weights or threshold adjustments are shown:

```
Accuracy: 0.6703, Precision: 0.2097, Recall: 0.6643, F1 Score: 0.3188, AUC-ROC: 0.6677
Specificity: 0.6710
```

These results are relatively balanced; however, 66% recall is not satisfactory for this case. Next outcome shown is adjusted with weights where the weight is equal to count of negative class divided by count of positive class (equal to 7.6) and a higher weight for the positive class means the model will try harder to minimize errors for defaults:

```
Accuracy: 0.8020, Precision: 0.2803, Recall: 0.4494, F1 Score: 0.3453, AUC-ROC: 0.6489
Specificity: 0.8484
```

This particular weight adjustment improves every metric except for recall which is not optimal for its purpose. By manually adjusting the weight to the value of 15 (double its original) the recall metric improves marginally.

```
Accuracy: 0.6186, Precision: 0.1940, Recall: 0.7241, F1 Score: 0.3060, AUC-ROC: 0.6644
Specificity: 0.7577
```

Last but not least by applying SMOTE resampling to account for the class imbalance and introducing thresholds (0.4 and 0.3 respectively) the products become:

```
Accuracy: 0.5513, Precision: 0.1782, Recall: 0.7929, F1 Score: 0.2910, AUC-ROC: 0.6563
Specificity: 0.5196
```

And,

```
Accuracy: 0.4164, Precision: 0.1531, Recall: 0.8885, F1 Score: 0.2612, AUC-ROC: 0.6214
Specificity: 0.3544
```

A whopping 89% recall can be achieved at the expense of missing on some potentially beneficial loans.

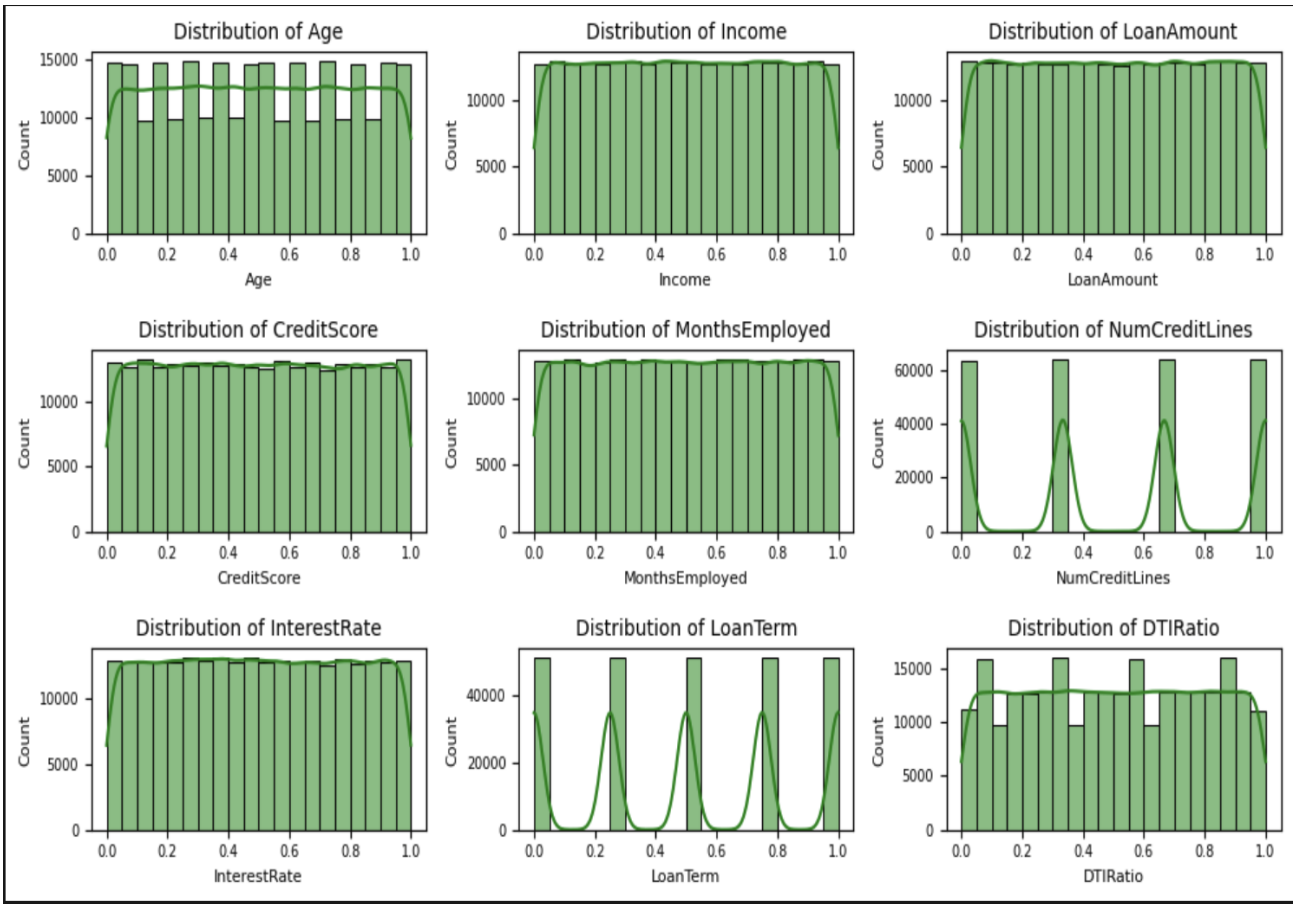
For clarification purposes these metrics indicate the following:

- Accuracy (~42%): The model correctly predicts the outcome for barely half of all samples.
- Precision (~15%): Among all loans predicted as defaults, only 15% are actual defaults. The model generates many false positives.
- Recall (~89%): The model identifies about 89% of all actual loan defaults, showing it is effective at minimizing missed defaults.
- F1 Score (~26%): The balance between precision and recall is obviously low.
- AUC-ROC (~62%): The model has moderate ability to distinguish between defaults and non-defaults, better than random guessing.
- Specificity (~35%): The model correctly identifies about a third of all actual non-defaults.

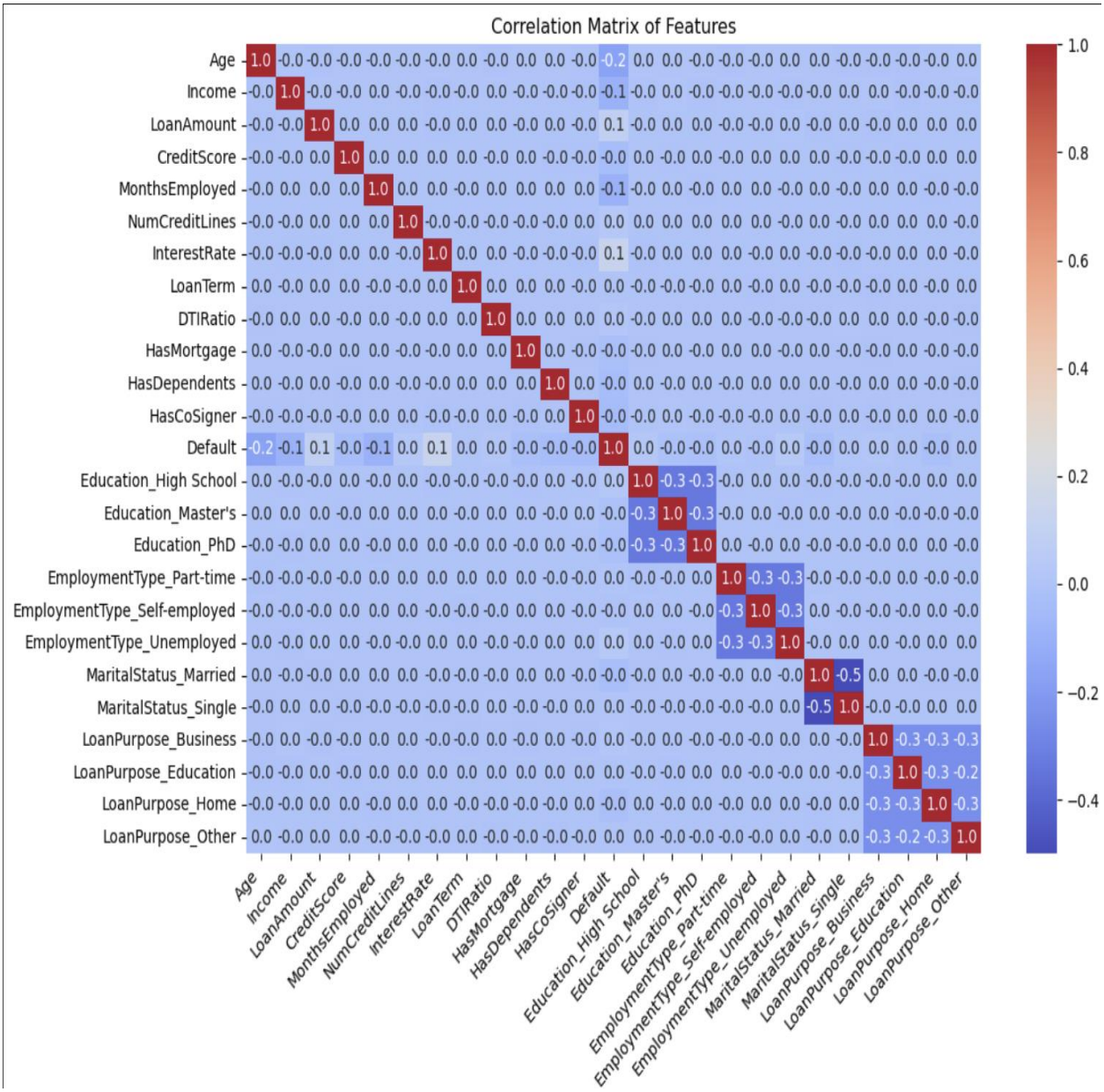
These results of course highlight a trade-off: the model focuses on catching defaults at the expense of incorrectly labeling non-defaults.

APPENDIX

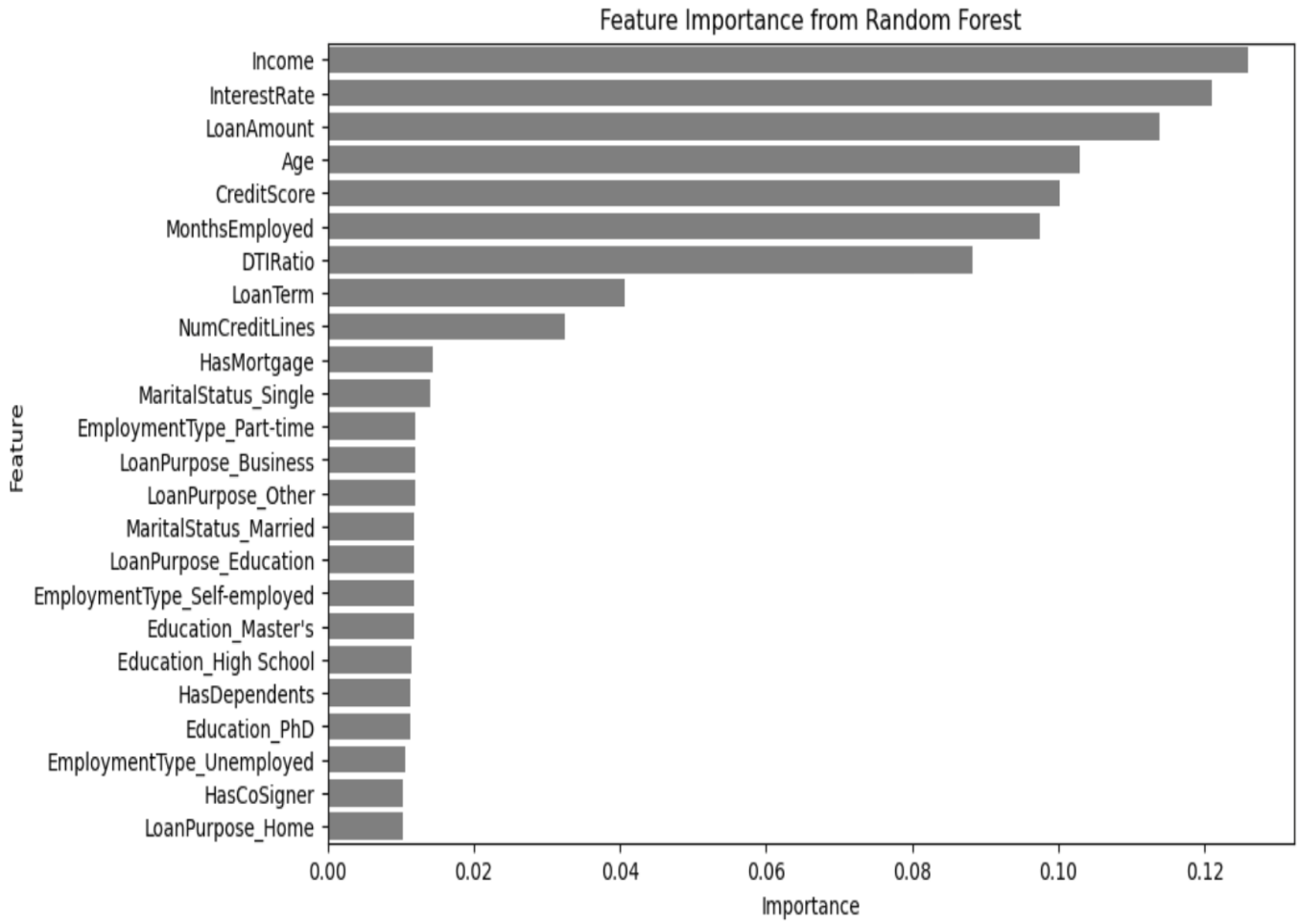
Distribution of numerical variables:



Correlation Matrix:



Feature Selection:


Feature Importance from Random Forest
Income
InterestRate
LoanAmount
Age
CreditScore
MonthsEmployed
DTIRatio
LoanTerm
NumCreditLines
HasMortgage
MaritalStatus_Single
EmploymentType_Part-time
LoanPurpose_Business
LoanPurpose_Other
MaritalStatus_Married
LoanPurpose_Education
EmploymentType_Self-employed
Education_Master's
Education_High School
HasDependents
Education_PhD
EmploymentType_Unemployed
HasCoSigner
LoanPurpose_Home
Feature
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Importance