Loan Approval Prediction using Neural Networks: Phase 1

Cornell Stokes

April 2025

1 Phase 1: Introduction

I selected this dataset because it presents a real-world problem that is relevant to financial institutions and individuals alike. The decision of whether or not a loan application is approved has significant implications, and using machine learning to predict these outcomes can help improve decision-making and increase efficiency.

The goal is to create a model that classifies whether a loan will be approved (1) or denied (0) based on features such as age, income, employment experience, credit score, and more.

2 Dataset Source

The dataset used for this project was obtained from [**insert source here – e.g., Kaggle or UCI ML Repository**]. It contains over 1,000 records and includes both categorical and numerical features.

2.1 Data Overview

The dataset contains the following input features:

- `person_age`
- `person_gender`
- `person_education`
- `person_income`
- `person_emp_exp`
- `person_home_ownership`
- `loan_amnt`
- `loan_intent`
- `loan_int_rate`
- `loan_percent_income`
- `cb_person_cred_hist_length`
- `credit_score`
- `previous_loan_defaults_on_file`

This also contains the link to my HTML code files: GitHub Repository - exportToHTML
The output label is:

- `loan_status` – where 1 indicates approval and 0 indicates rejection

2.2 Feature Distribution

Histograms were created to visualize the distribution of each numeric feature. These plots provide insight into the range and distribution of data.

Figure 1: Histograms for numeric features after normalization

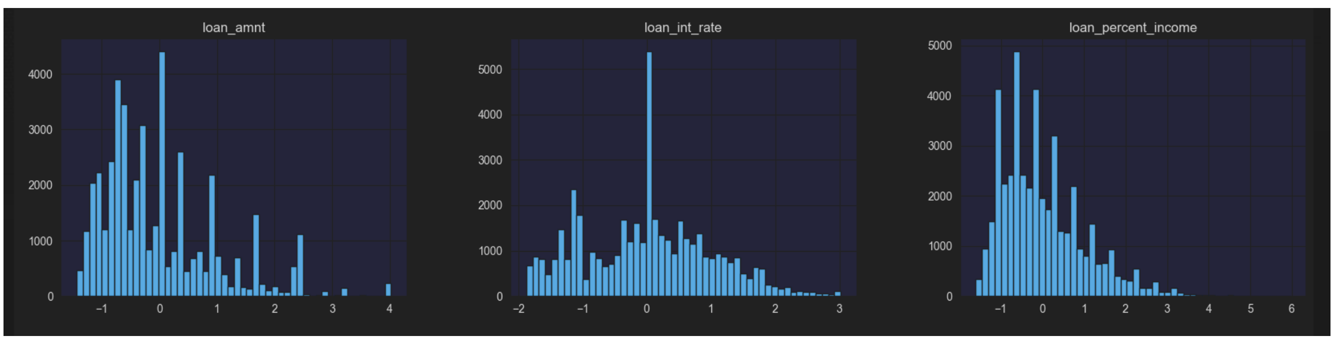

Figure 2: Histograms for numeric features after normalization

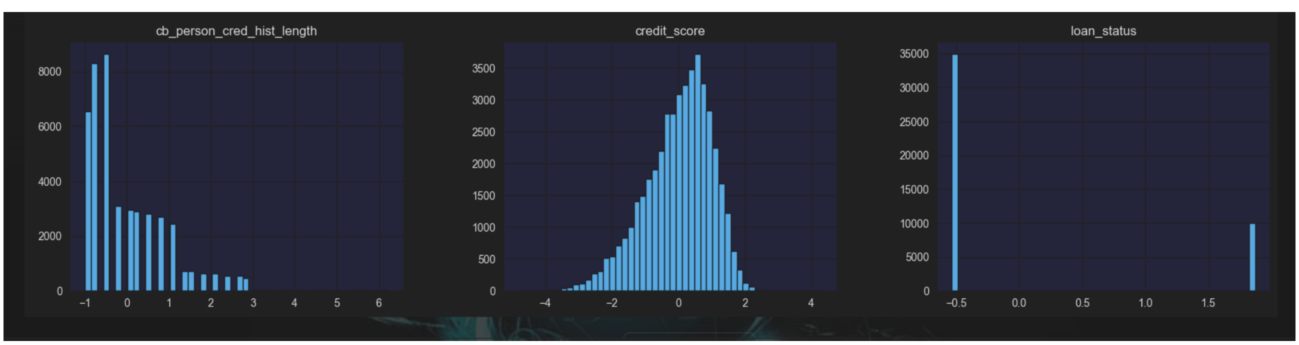

Figure 3: Histograms for numeric features after normalization

2.3 Output Label Distribution

To understand whether the dataset is balanced or not, the distribution of the output label `loan_status` was calculated:

- Approved (1): XX%
- Denied (0): XX%

2.4 Data Normalization

Only numeric features were selected for normalization. Categorical features like gender, home ownership, loan intent, and education were excluded. The normalization process was done using `StandardScaler` from `sklearn`, which transforms the features to have a mean of 0 and a standard deviation of 1.

- **Why normalize?** Neural networks perform better when numeric inputs are scaled uniformly.
- **How?** Used `StandardScaler.fit_transform()` on selected columns.

2.5 Descriptive Statistics Table

Feature	Count	Mean	Std	Min	25%	
person_age	45000	-1.191343e-16	1.000011e+00	-1.284388e+00	-6.226885e-01	-2
person_income	45000	-4.294836e-17	1.000011e+00	-8.992491e-01	-4.117681e-01	-1
person_emp_exp	45000	1.073709e-17	1.000011e+00	-8.922841e-01	-7.273619e-01	-2
loan_amnt	45000	1.263187e-17	1.000011e+00	-1.438388e+00	-7.257784e-01	-2
loan_int_rate	45000	-2.779012e-16	1.000011e+00	-1.875471e+00	-8.112750e-01	1.
loan_percent_income	45000	-9.094947e-17	1.000011e+00	-1.602141e+00	-7.994934e-01	-2
cb_person_cred_hist_length	45000	1.957940e-17	1.000011e+00	-9.968632e-01	-7.391085e-01	-4
credit_score	45000	-9.627065e-16	1.000011e+00	-4.810296e+00	-6.267188e-01	1.
loan_status	45000	1.212660e-16	1.000011e+00	-5.345225e-01	-5.345225e-01	-5

3 Conclusion

Phase 1 was about getting the data ready for the rest of the project. Since my data set had both words and numbers in columns, I needed them split to get an accurate representation for the scaling in my dataset. This help me with the rest of the phases and its balancing (overall imbalance).