Loan Approval Prediction using Neural Networks

Cornell Stokes

April 2025

Abstract

This project explores the use of neural networks for predicting loan approval decisions based on applicant data. By analyzing both numerical and categorical features such as age, income, credit score, and loan intent, the model aims to classify applications as either approved or denied. The data set used is real-world and unbalanced, making it a practical case for preprocessing, normalization, and model evaluation. The final results provide insights into the effectiveness of neural networks in financial decision-making scenarios. The items in the dataset consisted of both specific numbers and text that needed to be hot-coded for the purpose of this project.A challenge in this dataset is the inherent imbalance in the dataset, where the number of approved loans outweights the number of denied ones. This imbalance affects model training and evaluation.

This project consists of multiple phases, including normalization, model training, and evaluation. TensorFlow and Keras is used to implement both logistic regression and multilayer neural networks. This contains phases 1, 3, and 4 where phase 2 will not be included as it handles overfitting the training data. Metrics such as precision, recall, accuracy, and F1-score were used to evaluate model performance. Additionally, multiple network models were tested to get the best accuracy score among the phases.

Additionally, feature importance was analyzed by isolating the contribution of each feature to the overall model accuracy. This helped identify which features were most predictive in the loan decision process. The final results demonstrate that neural networks can effectively imitate complex decision boundaries in financial datasets, specifically for loan status, when properly tuned and evaluated.

Contents

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 magentaData$_s$*ource*. 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: magentaGitHub 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

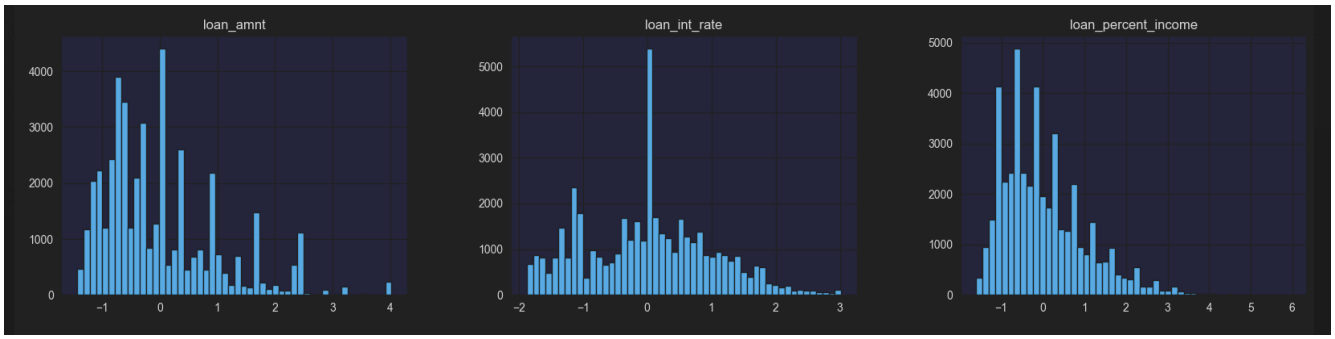

Figure 2: Histograms for numeric features after normalization

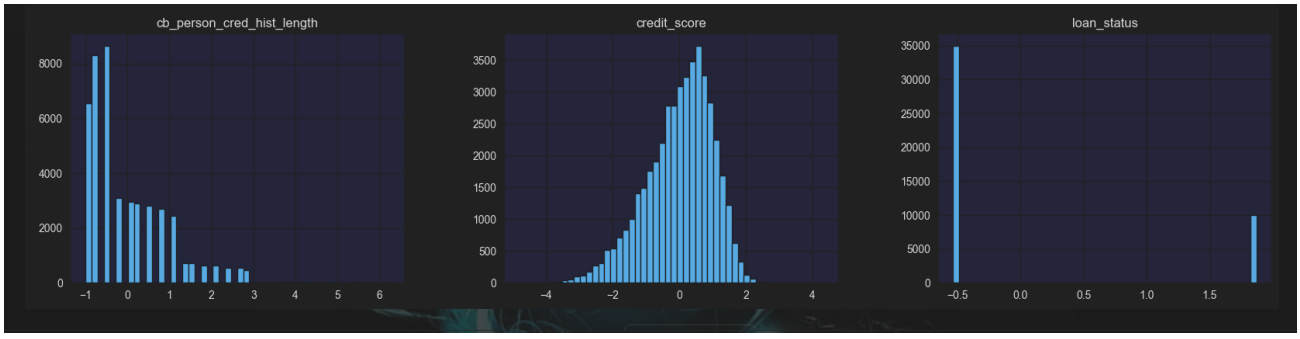

Figure 3: Histograms for numeric features after normalization

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%	50%	75%	Max
person_age	45000	-1.191343e-16	1.000011e+00	-1.284388e+00	-6.226885e-01			
person_income	45000	-4.294836e-17	1.000011e+00	-8.992491e-01	-4.117681e-01			
person_emp_exp	45000	1.073709e-17	1.000011e+00	-8.922841e-01	-7.273619e-01			
loan_amnt	45000	1.263187e-17	1.000011e+00	-1.438388e+00	-7.257784e-01			
loan_int_rate	45000	-2.779012e-16	1.000011e+00	-1.875471e+00	-8.112750e-01			
loan_percent_income	45000	-9.094947e-17	1.000011e+00	-1.602141e+00	-7.994934e-01			
cb_person_cred_hist_length	45000	1.957940e-17	1.000011e+00	-9.968632e-01	-7.391085e-01			
credit_score	45000	-9.627065e-16	1.000011e+00	-4.810296e+00	-6.267188e-01			
loan_status	45000	1.212660e-16	1.000011e+00	-5.345225e-01	-5.345225e-01			

3 Phase 3: Data Split

First, the loan data was randomly shuffled, then split into training, validation, and testing periods. For the training and testing split, the data was split 80 percent for training, and 20 percent for testing. Once this was finished, I then divided it into 64 percent for training, 16 percent for validation, and 20 percent for testing.

3.1 Data Normalization

When determining the balance of the data, I noticed that the data was heavily imbalanced. By taking the numbers of 0's and 1's, which dictate the loan status, this was confirmed.

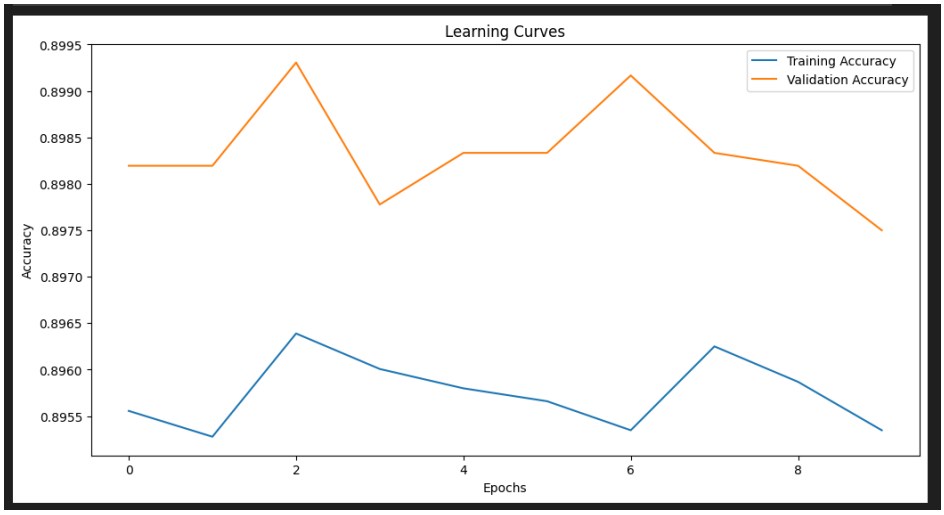

Figure 4: Distribution of approved vs. denied loans

From the chart above, we can observe that the dataset is noticeably imbalanced, with a significantly larger portion of loans being denied (class 0) compared to those approved (class 1).

To account for this imbalance during model evaluation, I will use performance metrics beyond accuracy, including precision, recall, and F1-score. These metrics provide a more complete view of how well the model handles both classes.

3.2 Model Comparison

Model Comparison Results

	Model	Acc. on Training Set	Acc. on Validation Set	Precision	Recall	F1-score
0	Neural network model (64, 32, 16, 8, 1)	92.40 %	91.29 %	86.55 %	74.33 %	0.7998
1	Neural network model (32, 16, 8, 1)	91.96 %	91.29 %	85.41 %	75.54 %	0.8017
2	Neural network model (16, 8, 1)	91.29 %	91.42 %	83.26 %	76.19 %	0.7957
3	Neural network model (8, 1)	90.64 %	91.11 %	82.92 %	76.09 %	0.7936
4	Neural network model (4, 1)	90.30 %	90.33 %	80.84 %	75.84 %	0.7826
5	Neural network model (2, 1)	89.64 %	89.79 %	78.76 %	73.93 %	0.7627
6	Logistic Regression	89.83 %	N/A	78.26 %	74.84 %	0.7651

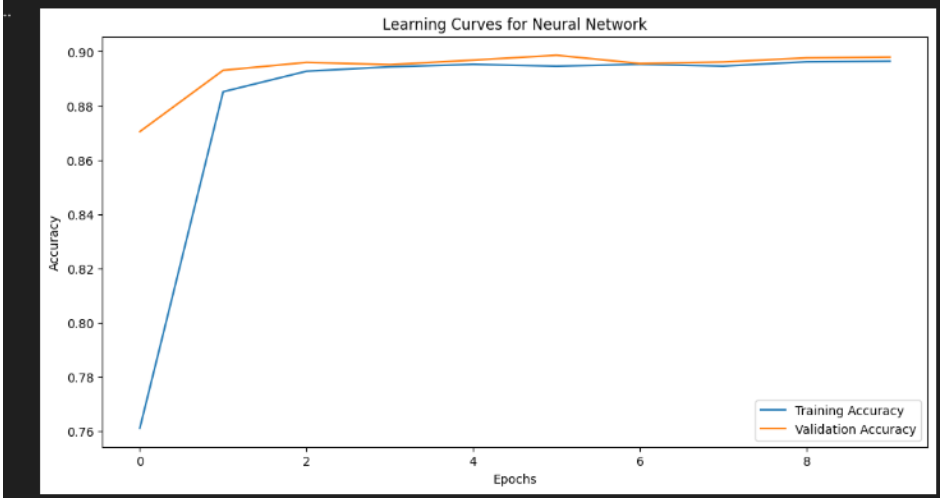

Figure 5: Model Comparison Table

4 Phase 4: Evaluating Feature Impact on Model Performance

For this phase, I compared the dataset with two models: a model using all columns, and another model where less important features were removed iteratively.

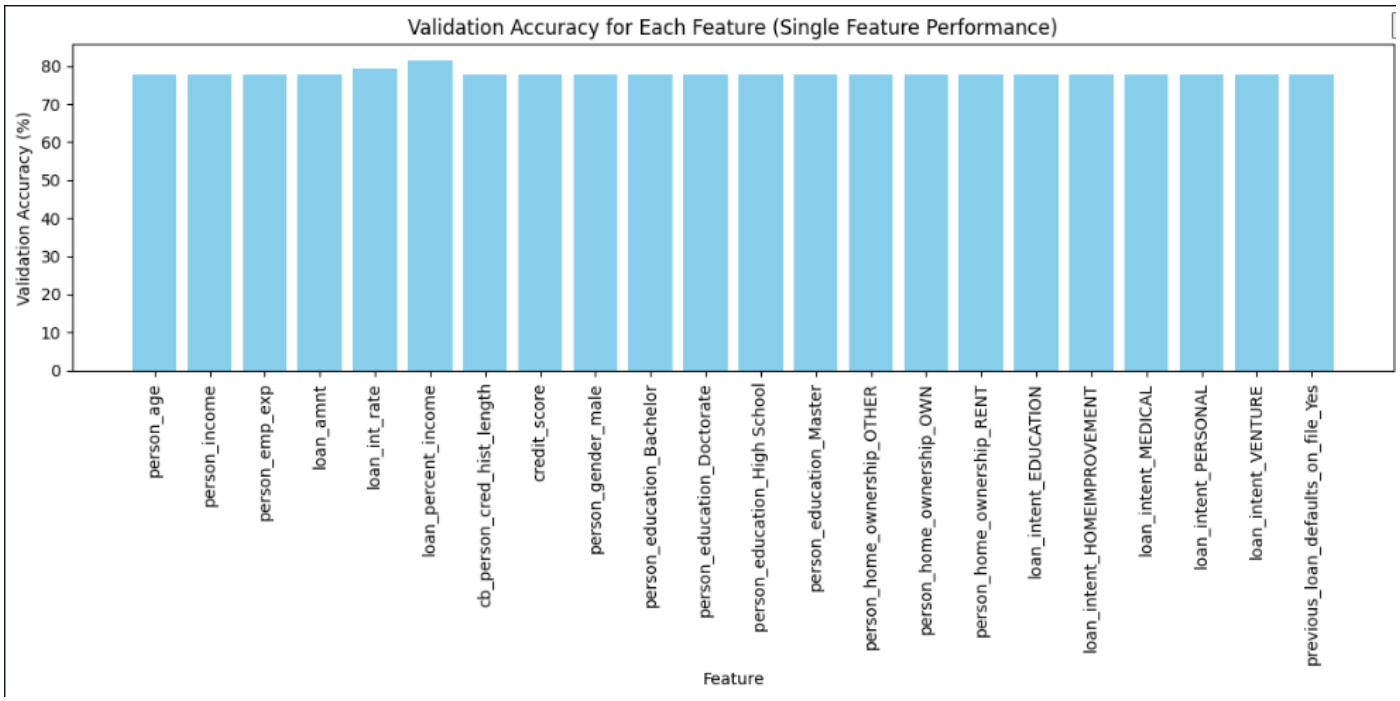

Figure 6: Validation accuracy by feature

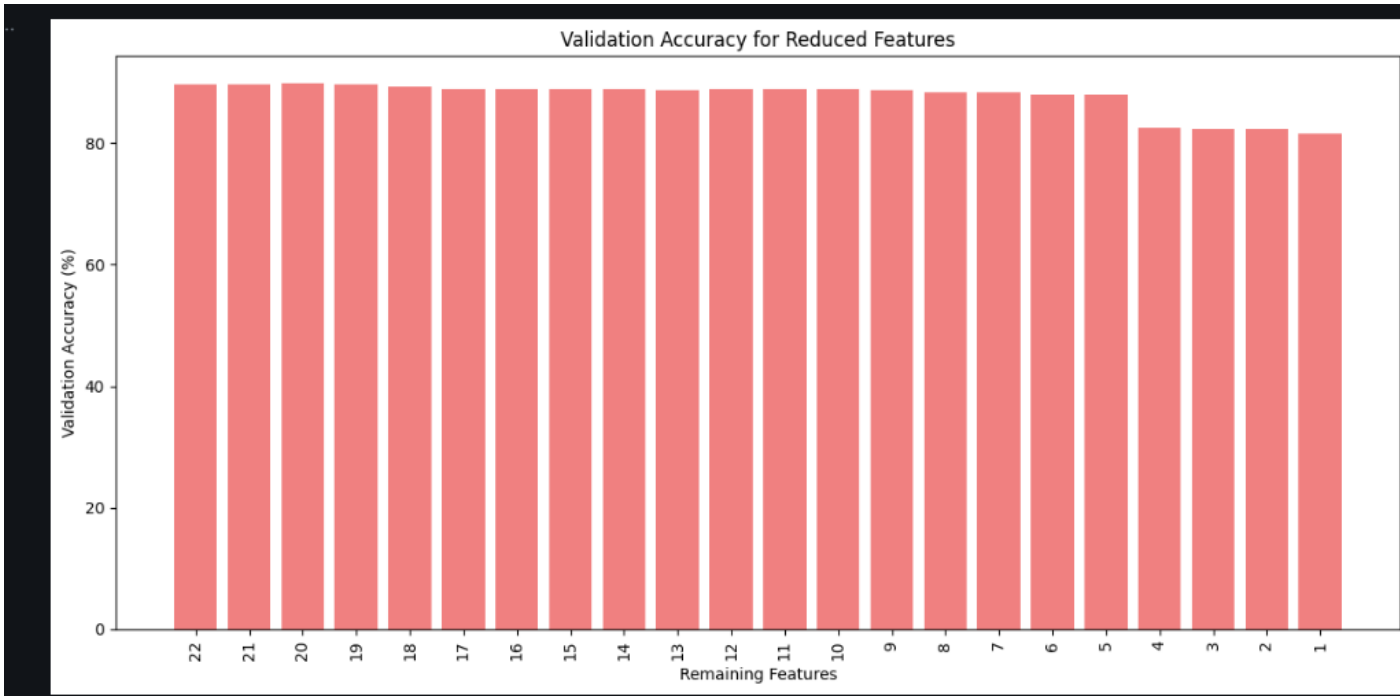

Figure 7: Model accuracy after removing low-impact features

5 Conclusion

Trying to predict the approval of an individual can depend on a lot of factors, especially as a binary classification problem. It takes a lot of financial data to properly predict if an individual will be able to get a loan. Throughout the project, I used logistic regress and neural networks to address the challenge of predicting loan approval status for individuals.

The processing steps, like one-hot encoding for categorical variables, were important for ensuring the neural network models could learn from the data. By experimenting with different features that impacted the accuracy the most, like income, loan amount, credit score, and loan interest rate, I was able to extract a significant impact on the prediction accuracy. These features were analyzed individually and in combination to determine their contribution to the models performance.

The results demonstrated that neural networks, particularly with optimized architectures, outperformed logistic regression in terms of precision, recall, and F1 score. However, there was an importance in addressing the data's imbalance, as it directly influenced the models ability to generalize. Techniques like model checkpointing, validation splits, and learning curve analysis further enhanced the training process.

Additionally, feature importance analysis using Shapley values provided valuable insights to the contribution of individual features, enabling a better understanding of the decision-making process. This is important for financial applications, where transparency and fairness is important.

In conclusion, this project focuses on machine learning in the case of financial decision-making while emphasizing the importance of data processing, feature selection, and model evaluation.