Loan Approval Prediction with Machine-Learning

Yong-Sung Masuda

December 20, 2024

Purpose:

Loan applications are approved or denied by financial institutions based on a set of disclosed criteria (credit history length, loan amount, loan intent, etc.) but with undisclosed decision-making processes. The result of a loan application likely varies based on the institution it was submitted to and the loan officer tasked with its review. Since the process is not strictly defined, predicting the outcome of a loan application is a problem well suited to a machine-learning solution. A well-trained prediction model may be used by financial institutions for various purposes, including application triage, automatic application approvals, and targeted marketing campaigns. Such a prediction model may also be of service to potential applicants in order for them to gauge what outcome to expect and perhaps adjust their loan application before submission.

Dataset:

The dataset used for this model was sourced from Kaggle.com, specifically from season 4, episode 10 of their playground series. This dataset was generated from a deep learning model trained on a loan approval prediction dataset with similar feature distributions. The dataset includes a labeled training set with 58,645 samples and an unlabeled test set of 39,098 samples. The training set is divided into three subsets:

- Training set of 41,051 samples (70%)
- Validation set of 8,797 samples (15%)
- Private test set of 8,797 samples (15%)

The training set is used to fit models to the data during the model selection phase. The validation set is used to evaluate the models for selection during hyperparameter tuning and is also used for early stopping. The private test set is used to ensure the selected model is not overfit to the training and validation sets, and is also used to compute additional performance metrics.

Data Preprocessing:

The parameters of each data sample are preprocessed as either categorical or numerical. Categorical parameters are one-hot encoded and numerical parameters are regularized to be a floating-point decimal between 0 and 1.

The categorical parameters are:

- Person home ownership (rent, own, or mortgage)
- Loan intent (education, medical, personal, etc.)
- Loan grade (A, B, C, D, or E)
- Person default on file (yes or no)

The numerical parameters are:

- Person age
- Person income
- Loan amount
- Loan interest rate
- Loan percent of income
- Person credit history length
- Person employment length

Model Architecture and Hyperparameter Tuning:

The neural network implemented has two hidden layers each with ReLU activation and dropout regularization. The sigmoid activation function is applied to the final layer, resulting in an output between 0 and 1 indicating the probability of loan approval. Bayesian optimization of hyperparameters is implemented with the Optuna optimization framework. The hyperparameters tuned and their range limits are:

- Sizes of each hidden layer (16 to 256 neurons)
- Dropout rate (10% - 50%)
- Learning rate (1e-5 – 1e-1)
- Batch size (16 – 256 samples)

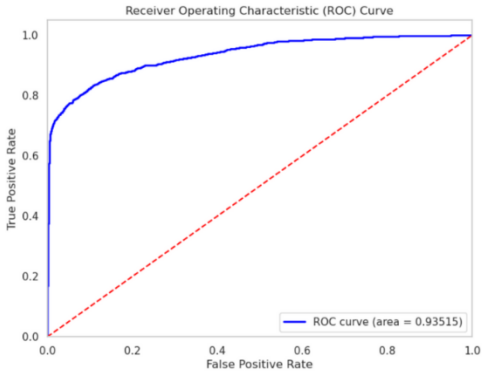
A total of 20 trials were conducted, with each trial trained for a maximum of 100 epochs. Early stopping was implemented with a patience of 7 epochs. The trial that yielded the best results was trial 3. The hyperparameters from this trial are used to train the final model with all of the available data (training, validation, and private test set combined) for 100 epochs with early stopping (best results were achieved after epoch 87). The complete results of the hyperparameter tuning are presented in the table on the next page.

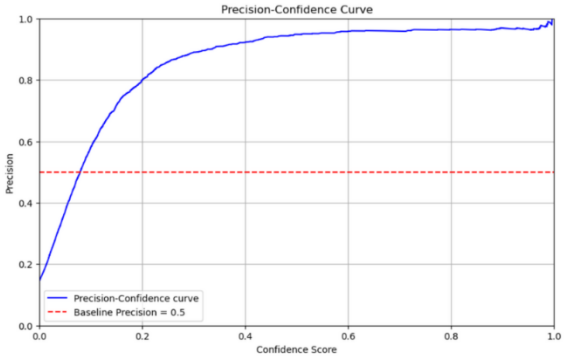
Hyperparameter Tuning Results

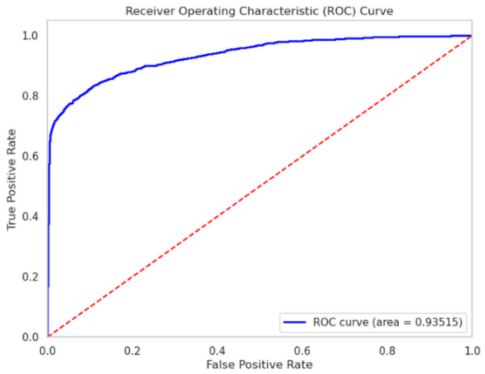
Trial	Hidden Size 1	Hidden Size 2	Dropout Rate	Learning Rate	Batch Size	Accuracy
0	130	153	0.345697	1.08E-05	222	0.932477
1	34	201	0.117724	0.020561	256	0.94248
2	71	239	0.387681	0.022064	48	0.939525
3	50	48	0.118661	0.00138	187	0.94464
4	223	248	0.290679	4.34E-05	25	0.941685
5	135	98	0.310409	6.72E-05	129	0.944299
6	238	233	0.45224	0.027946	217	0.943276
7	182	40	0.382687	2.40E-05	166	0.940093
8	134	226	0.249731	0.097402	17	0.857906
9	223	199	0.446721	1.62E-05	148	0.941116
10	22	19	0.105617	0.001333	102	0.944186
11	83	72	0.201061	0.000366	105	0.94248
12	91	95	0.201337	0.000684	166	0.942253
13	183	115	0.182077	0.000143	105	0.941457
14	171	63	0.306992	0.002488	194	0.942139
15	60	139	0.256108	0.003559	134	0.94339
16	109	93	0.163457	0.000145	80	0.943162
17	49	53	0.242589	9.80E-05	187	0.943276
18	156	161	0.343845	0.006289	139	0.944527
19	158	171	0.484996	0.007074	253	0.942139

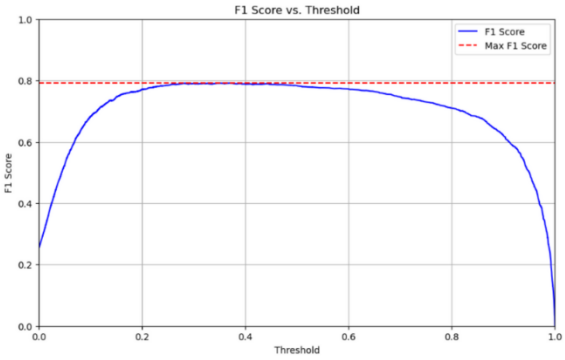
Evaluation:

The best model selected from hyperparameter tuning achieves a prediction accuracy of 94.46% on the validation set (used for model selection) and an accuracy of 94.57% on the private test set. The similar results achieved on the validation set and private test set confirms that the model is not overfit to the training and validation data, and is able to extrapolate. Additional evaluations are performed on the results from the private test set.









These graphics are not available for the final model since it is trained on the entire training set and the labels for the test set are unavailable. To evaluate the final model, predictions on the unlabeled test set are submitted to Kaggle where they are segmented into two sets in order to provide two AUROC scores, one being displayed until the competition deadline, and the other being published after. The pre-deadline score achieved by the final model is 0.93740 and the post-deadline score is 0.93313.

Acknowledgements:

The code for this project was developed while consulting various AI chatbots powered by Large Language Models (LLMs), including Claude Sonnet 3.5, Gemini 1.5 Flash, and ChatGPT 3.5. While the models did not produce directly usable code, they were instrumental in drafting code and providing usage examples for Python library functions. The human contributors behind the training data for these models, likely users of platforms such as GitHub and Stack Overflow, deserve recognition for their indirect support.