Classifying Loan Applications by Applying Machine Learning Algorithms

Final Project Report

Chichi Christine, Daniel Albert

Intro to Machine Learning CS 4347

Texas State University

December 1, 2019

1. INTRODUCTION

Lending money to good entities encourages entrepreneurship and stimulates economic growth. Banks worry about loans to individuals and small businesses due to the possibility of default. This concern leads banks to limit how much money can be extended and to whom. It is important to make good decisions about which entity should receive a loan from a financier. Good loans lead to increased profit and a more stable economy for everyone. Applying machine learning may help us reduce mistakes and bias during the process. Machine learning (ML) can help companies make good decisions, reduce risk and manage uncertainty by providing tools for loan management.

In the past, some methods of assessing creditworthiness were used. Letters of reference and vouches were replaced gradually by analysis of multiple financial factors. For the entirety of the 20th century, "scorecards" were the norm. These involved a human analyst reviewing a number of factors which the bank had selected to determine creditworthiness. The number of individual factors has steadily risen to match the progression of technology.

These factors and weights are quantifiable and thus can be determined via algorithm. Credit scoring is a good example of this. Past defaults, large numbers of inquiries, and unpaid accounts form a picture of the debtor's quality, which can then be used to predict how they may act in the future. Banks use similar factors to credit scoring agencies, but also include factors that credit scoring agencies may not legally include. These weights may include those which are categorical (married status, gender, etc.) or quantitative (income, amount requested, age, etc.).

For the past twenty years, lenders and fixed-income investors have become interested in using machine learning models. The Federal Reserve Bank of St Louis wrote in 1998 that automated mortgage credit scoring systems were used to find patterns in the credit histories of groups of individuals as reflected in credit bureau records [1]. Since then, these tools have continued to evolve. Using them to reduce risk helps companies make better financial decisions both internally and externally. By automating their analysis processes, companies can save money and make more accurate, precise decisions which benefit debtors as well as lenders.

Modern ML models used in loan management can be explained and incorporated with automation tools that increase efficiency of validation and monitoring. In addition, they could lead to better credit risk management and business performance. ZestFinance provides modern innovations in this field in the form of credit scoring models that incorporate artificial intelligence [2]. ZestFinance analysts use data such as historical loan performance data, credit bureau data, debt to income ratio and other unique borrower characteristics to build a machine learning model that can be explained. That model increased number of borrower approvals while reducing credit losses. When statistical anomalies occur, the software throws a flag and people can adjust the decision-making process. The model is retrained every few months using collected data.

2. PROBLEM DESCRIPTION

Financial companies have to decide who to lend to and they also need to apply software and automation to reduce cost, reduce the load of work placed on humans, and reduce mistakes or bias in lending. By implementing logistic regression, random forests, neural networks or gradient tree boosting we will be able to increase efficiency of predictions about loan approval.

2.1 Data, Theory and Application

Banks or credit unions usually consider customer information such as job, salary, income, age, gender, marital status, location (urban or rural), collateral, existing client (yes or no), number of years as client, total debt, account balance. Credit information refers to the total loan amount applied for, purpose, amount of the monthly payment, interest rate, etc.

Credit history is payment history and delinquencies or payment delays, amount of current debt, number of months in payment arrears, length of credit history, time since last credit, types of credit in use. Bank account behavior includes average monthly savings amount, maximum and minimum levels of balance, trends in payments, trends in balance, number of missed payments, times the applicant exceeded credit limit, times the applicant changed home address.

2.2 Data Description

Our dataset comes from https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/

Variable	Description
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)
Dependents	Number of dependents
Education	Education (Graduate/ Under Graduate)
Self_Employed	Self-employed (Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Co-applicant income
LoanAmount	Loan amount in thousands

Loan_Amount_Term	Term of loan in months
Credit_History	Credit history meets guidelines
Property_Area	Urban/ Semi Urban/ Rural
LoanStatus	Loan approved (Y/N)

Table 1. Description of Variables in the Data Set

Exploratory Data Analysis (EDA) is the process of figuring out what the data can tell us and we use EDA to find patterns, relationships, or anomalies to inform our analysis. Our dataset has dimensions 614 x 13. Balance of classes is 0.687 of tuples are LoanStatus = ‘Y’. We used Seaborn’s pairplot function and sns.heatmap to look at relationships between variables.

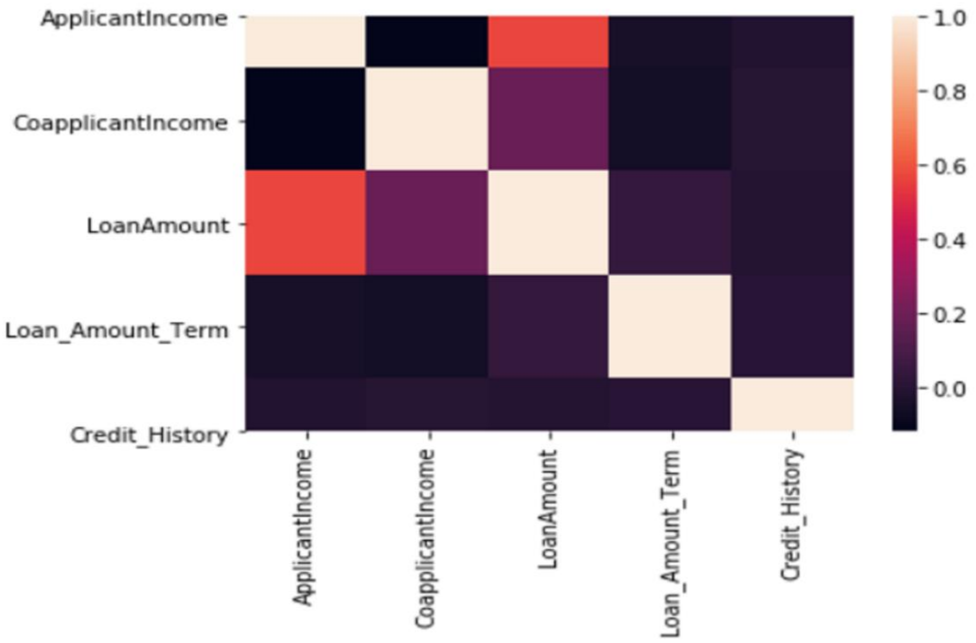

Figure 1. Sns heatmap showing relationships between some variables

Proj2.ipynb has the program, pairplot and heatmap. There is a positive correlation between applicant income and loan amount. There were no outliers in the data.

2.3 Theory and Application

GitHub link to code: https://github.com/mlp12/Final-Project-Report.git

Logistic regression is a statistical method that analyzes a dataset to identify the relationship between “predictors” that are independent variables and an outcome or dependent variable. Our output is an estimate of the probability of granting a loan given the input variables. “z” is a function that acts on relevant input data. For example, we try to find the coefficients α_n where n = 0, 1, 2, 3, 4.

$$z = \alpha_0 + \alpha_1 * \text{education} + \alpha_2 * \text{income} + \alpha_3 * \text{credit history} + \alpha_4 * \text{credit duration}. \tag{1}$$

Probability (P) of a bad loan is:

$$P(\text{Bad Loan}) = \exp(-z)/\ (1 + \exp(-z)) \tag{2}$$

$$P(\text{Good Loan}) = 1 - P\ (\text{Bad Loan}) \tag{3}$$

$$P(\text{Good Loan}) = 1/(\ 1 + \exp(-z)) \tag{4}$$

Target y is binary i.e., granted means P = 1, not granted means P = 0. We train the classification model with historical data where the outcome is already known. Use cross-entropy as a loss function to compare predictions to actual results. Find coefficients that minimize loss. Minimize loss using an optimization algorithm like gradient descent.

We prepared the data and did logistic regression analysis of the data. Please see the Appendix. We filled in missing values (NA) with reasonable values such as the mean, most frequent value for that column or zero. We replaced categorical values with numeric equivalents through onehot encoding.

```
lrobj = LogisticRegression(solver='liblinear', class_weight='balanced')
```

The “balanced” mode uses the values of y to automatically adjust class weights inversely proportional to class frequencies.

Random Forest

A random forest consists of many individual decision trees that operate as an ensemble. Each individual tree in the random forest gives us a class prediction and the class with the most votes becomes our model's prediction. The process is to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree [3]. The trees should be uncorrelated with each other. There needs to be a signal between features and the outcome so that models are not doing random guessing. Random forests can classify large data sets as long as relevant features are put into the model.

Advantages: Ensemble learning prevents overfitting of data. Bagging enables random forest to work well on relatively small datasets.

We used the sklearn RandomForestClassifier model. N_estimators = 200 which is the number of trees in the forest. Criterion is gini to measure the quality of a split. High Gini index indicates the node gave closer to a 50/50 split of tuples.

Neural Network

A neural network recognizes patterns in data. The node combines each input from the data with a weight, that either amplifies or dampens that input. Weighted input results in a guess. Network's guess - ground truth = error. The program walks the error back over its model, adjusting weights to the extent that they contributed to the error.

We prepared the data, used StandardScaler on numerical features, and replaced categorical values with numeric equivalents through onehot encoding. The layers consist of Keras Sequential model and Dense library. From [4], Dense implements output = activation(dot(input, kernel) + bias).

ReLu is the activation function for hidden layers. It is a binary classification problem therefore sigmoid is the activation function for the output layer. Number of nodes in input layer is 100, 60 in a hidden layer and 60 in a second hidden layer. Dropout was 0.2 to prevent overfitting [5]. Loss is binary_crossentropy.

3. RESULTS

3.1 Results from Logistic Regression

Using sklearn models we fit a model to the data that helped us predict labels for each tuple with accuracy = 1. There may be a linear relationship between input variables and the output. Confusion matrix contains these values:

[[Actual N Predicted N, Actual N Predicted Y],

[Actual Y Predicted N, Actual Y Predicted Y]]

Confusion matrix was

```
array([[ 46, 0],
        [ 0, 108]])
```

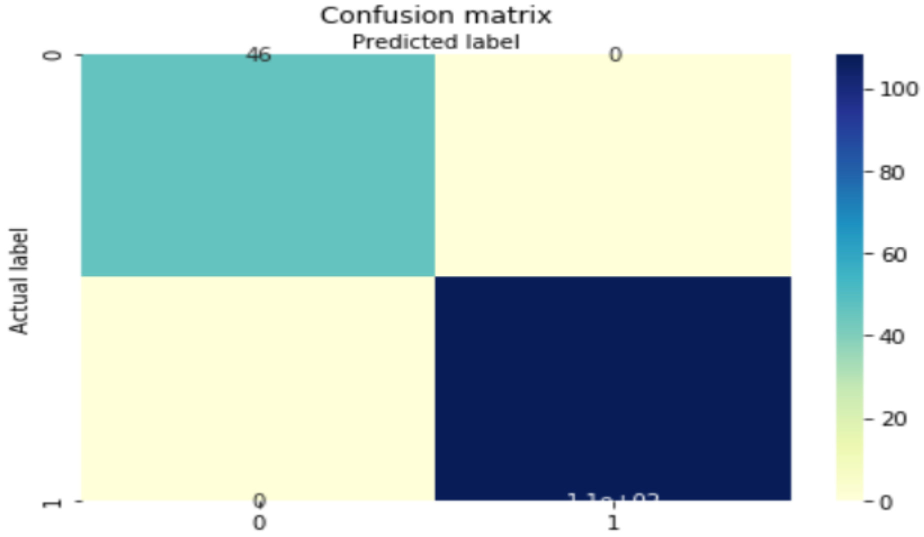

Figure 2. Plot of Confusion Matrix

3.2 Theoretical Properties and Results from Neural Network

Initial accuracy was low so we tried to find out how to improve the model. SGD gave better results than Adam for this network. According to Chengwei [6], SGD with nesterov is recommended for shallower networks. SGD randomly picks one data point or a mini batch from the whole data set at each iteration, to update parameters.

Nesterov Accelerated Gradient, we apply the velocity *vt* to the parameters θ to compute interim parameters $\tilde{\theta}$. Compute the gradient using the interim parameters. Momentum method alone can be slower since the optimization path taken exhibits large oscillations. Nesterov Accelerated Gradient's correction avoids the oscillations. Details of programs and results are in proj.ipynb, proj2.ipynb, model2 and proj2Ver3.ipynb submitted with this report.

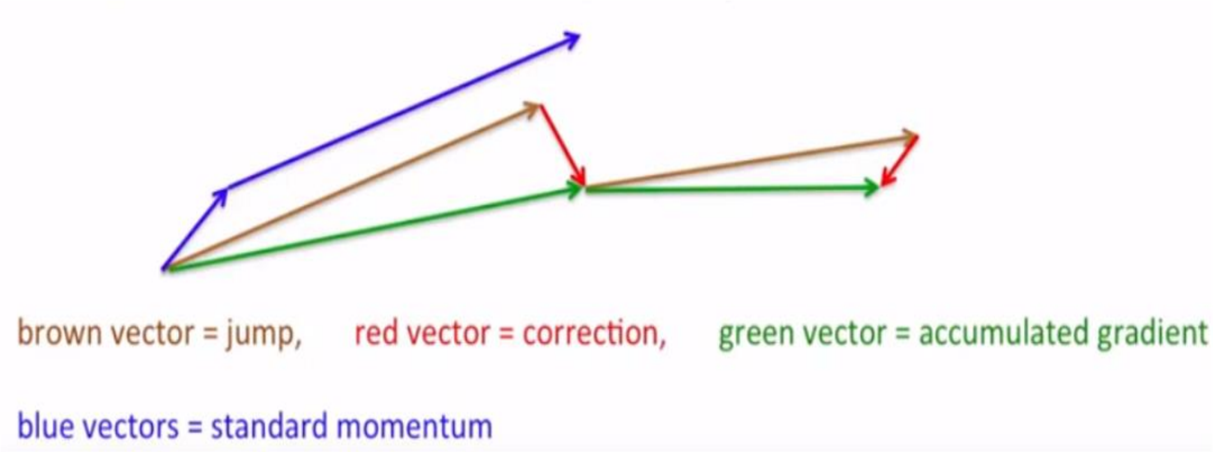
brown vector = jump,
red vector = correction,
green vector = accumulated gradient
blue vectors = standard momentum

Figure 3. Nesterov method provides less overshooting.

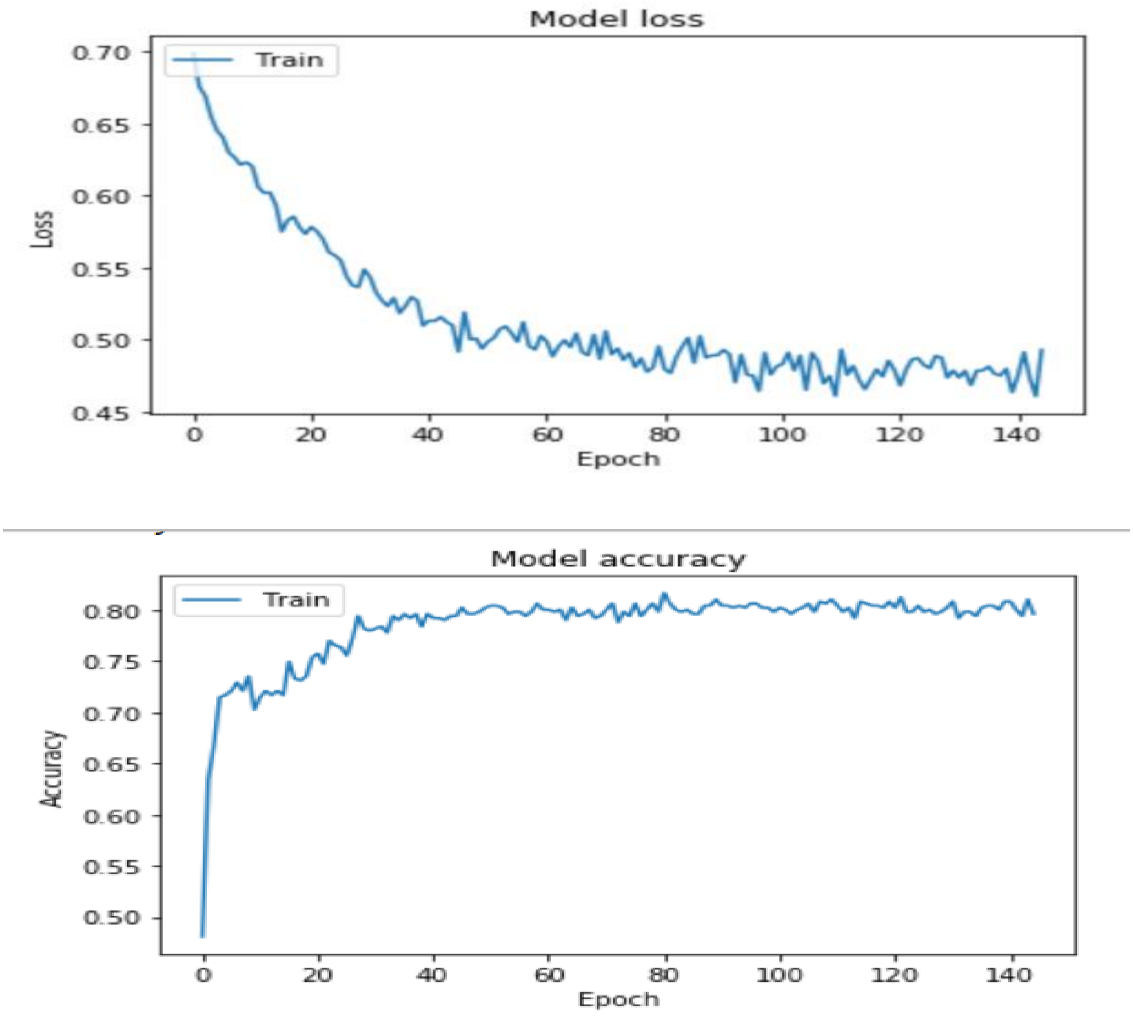

Figure 4. Training Loss and Accuracy

Test accuracy was 0.804. Looking at the list from the Random Forest program, we identified some less relevant features. We removed less relevant features: Gender and Dependents, Self_Employed, Married; and fit the model a second time. Test accuracy increased to 0.854.

Precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. Recall is the ratio tp / (tp + fn) where fn is the number of false negatives. The recall is the ability to find all the positive samples. The F-1 score can be interpreted as a weighted harmonic mean of the precision and recall.

Precision = 0.88, Recall = 0.72, F-1 score = 0.75

Further reading revealed that Neural Networks give higher accuracy when more data is available [7]. For instance, a dataset with 5000 or more tuples.

3.3 Results from Random Forest

Initial accuracy was 0.772. We removed Gender, Married, Self_Employed and Dependents; and increased the number of estimators to 200. Accuracy rose to 0.829.

4. CONCLUSIONS AND FUTURE WORK

Logistic regression, neural networks and random forests were used to predict loan application outcomes. Logistic regression gave the best predictions of classes with the highest accuracy. There may be a linear relationship between input variables and the output. Having more data points increases accuracy of neural networks (NN) and random forests. Fine-tuning parameters of the model and removing less relevant features improved the predictions from NN and Random Forests. Standard scaling of numerical features improves performance of logistic regression and NN. In industry, companies are deploying machine learning systems based on neural networks or gradient tree boosting for loan approval tasks. Automation is involved in monitoring models, doing analysis of results and alerting humans about anomalies.

<u>Future Work:</u>

- Reduce bias in lending by removing biased inputs when training the model. Do not use features such as race, gender or marital status for building the model.
- Lenders can also set up a separate project that lends to and supports female business owners. This is because many women work and have the additional responsibility of taking care of children.

- Explainable ensembles are preferred to obscure ones.
- Humans should also monitor external factors, e.g., a recession or natural disaster so that they know when to choose a different response than the one being suggested by a computer.
- Continue automation of processes in order to reduce the load of work placed on humans and reduce cost.

5. REFERENCES

[1] Federal Reserve Bank of St. Louis, “How mortgage lenders are using automated credit scoring,” St. Louis, MO, *Bridges*, Winter 1998, [Online]. Available: https://www.stlouisfed.org/publications/bridges/winter-1998/how-mortgage-lenders-are-using-automated-credit-scoring

[2] ZestFinance, “ZestFinance and Prestige Financial Services deploy first AI-powered credit scoring model in subprime auto lending industry,” *Cision PR Newswire, Sept 27,* 2018. [Online]. Available: https://www.prnewswire.com/news-releases/zestfinance-and-prestige-financial-services-deploy-first-ai-powered-credit-scoring-model-in-subprime-auto-lending-industry-300720115.html

[3] T. Yiu, “Understanding Random Forest, How the Algorithm Works and Why it Is So Effective,” Towards Data Science, June 12, 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[4] R. Khandelwal, “Building Neural Network using Keras for Classification,” Medium, Jan 6, 2019. [Online]. Available: https://medium.com/datadriveninvestor/building-neural-network-using-keras-for-classification-3a3656c726c1

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, June, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[6] Chengwei, “Quick Notes on How to Choose Optimizer in Keras,” Dlology, Feb, 2018. [Online]. Available: https://www.dlology.com/blog/quick-notes-on-how-to-choose-optimizer-in-keras/

[7] J. Klaas, "*Machine Learning for Finance*," Packt Publishing, 2019. [Online]. Available: https://catalog.library.txstate.edu/search/a?searchtype=X&searcharg=finance+neural+networks&submit=Search&searchscope=1&SORT=DX