CS230

Predicting Who Will Get A Mortgage in New York

Manjinder Kandola
Stanford Graduate School of Business, MBA Candidate '19
mkandola@stanford.edu
Aris Konstantinidis
Stanford Graduate School of Business, MBA Candidate '19
arisk@stanford.edu
Dasmer Singh
Stanford Graduate School of Business, MBA Candidate '19
dasmer@stanford.edu

Abstract

There has long been a debate on whether or home mortgage application decisions are still racist, despite the passage of Fair Housing Act in 1968. Our team choose to put this debate to the test using recent advancements in Deep Learning to understand what factors are most important when loan underwriters accept or reject loan applications in NY and try to predict the outcome of a loan application. After using a deep neural network, we uncovered that applicant race continues to be the single largest factor in deciding whether or not a loan application is accepted.

Introduction

The process of underwriting home mortgage applications today is a black box to most consumers. They fill out long forms of papers, wait often for days if not weeks, and finally get a binary accept or reject to their application. There usually is no explanation as to why that decision was made. Using all that we have learned in CS230 in deep learning, our team wanted to be able to explain this block box by assessing the weights of the fields that go into the mortgage application.
We used data made publicly available by the NY Home Mortgage Disclosure Act, which requires financial institutions to disclose application data, for all NY home mortgage

applications in 2015. After cleaning the data, we had data from 155,000 applications. The inputs to our project were race of applicant, race of co-applicant, loan purpose, property type, state code and loan amount. We then use a neural network to output a predicted accept or reject to the loan application.

Our model is valuable not only to financial institutions, which can use it to at least complement their assessment process, but also to potential borrowers, who can receive feedback on the predicted outcome of their applications. Of course, we believe that banks already develop similar models, but we believe that there is a lot of room for improvement, especially if we consider the amount and types of sensitive information that needs to be processed and taken into account.

Related Work

Many before have tried to solve this problem. The Federal Reserve Bank [1] attempted a similar machine learning regression to diagnose whether mortgage decisions were fair based on gender and race. Their findings actually demonstrated that logit regressions perform better than machine learning does to predict and solve these problems.
Kay Giesecke is Professor of Management Science & Engineering at Stanford University also applied “Deep Learning to assess Mortgage Risk” [2]. His findings were that County-level unemployment rates, Zip-code level housing prices, and Lagged foreclosure and prepayment rates in zip-code were the most predictive indicators for why someone might default on a loan.
Ali Bagherpour at the University of California, Riverside is also solving the problem and claims that nonlinear and non-parametric algorithms are substantially better than the logit models in his paper “Predicting Mortgage Loan Default with Machine Learning Methods” [3]. This paper does not go on to explain the weight of the inputs, but rather focuses on the accuracy of the algorithm. We were able to use learnings from this paper such as how to normalize data and optimize algorithms in order make our models converge faster and more accurately.
Bolarinwa Akindaini at the University of Tampere in his paper “Machine Learning Applications in Mortgage Defaults Predictions”[4] explores how machine learning methods can be used to classify mortgages into in good standing, prepaid, or defaulted. He uses Logistic regression (simple and multi-class), Naive Bayes, Random forest and K-Nearest Neighbors methods. His findings also reveal that unemployment rate in the region is the largest predictor of risk of default.
Amir E. Khandani and his team at MIT look at all types of credit risk (not only specific to home-mortgage) in his paper “Consumer Credit Risk Models via Machine-Learning

Algorithms”[5]. They look at consumer transactions from customer transactions and credit bureau data from January 2005 to April 2009 to assess reasons for delinquency. Consistent with other findings, unemployment is what is the best predictor of delinquency.

Dataset and Features

We used a dataset publicly available on consumerfinance.gov that consists of 439,654 NYS home mortgage applications in 2015. Many of these mortgages were withdrawn before a decision was even made and other had incomplete data. Also, the dataset contained seven possible outcomes for each application, instead of two (originated / not originated). So, after adjusting the dataset to fit our binary classification model, and after cleaning the data, we ended up with a dataset of 155,000 rows. Each row of in our data was one application and each column had 78 different variables which described the lender, institution, applicant, loan and property. In order to avoid overfitting or making our model more excessively complicated we used just 6 variables that we hypothesized would be most descriptive: race of applicant, race of co-applicant, loan purpose, property type, state code and loan amount.

Methods

In our model, y is the 1 or 0 if the loan is accepted or rejected. $\hat{y}$ is 1 or 0 if our model predicts the loan is accepted or rejected respectively. The goal of our model is to minimize the loss function. That is, minimize how often our model predicts the loan incorrectly.

$$min(\Sigma\,(y\ -\ \hat{y}\ )^2\ )$$

In order to do so we use gradient descent on our neural network. In the experiments section, we discuss how we designed our network including architecture and how we chose our hyper parameters. There is a linear function

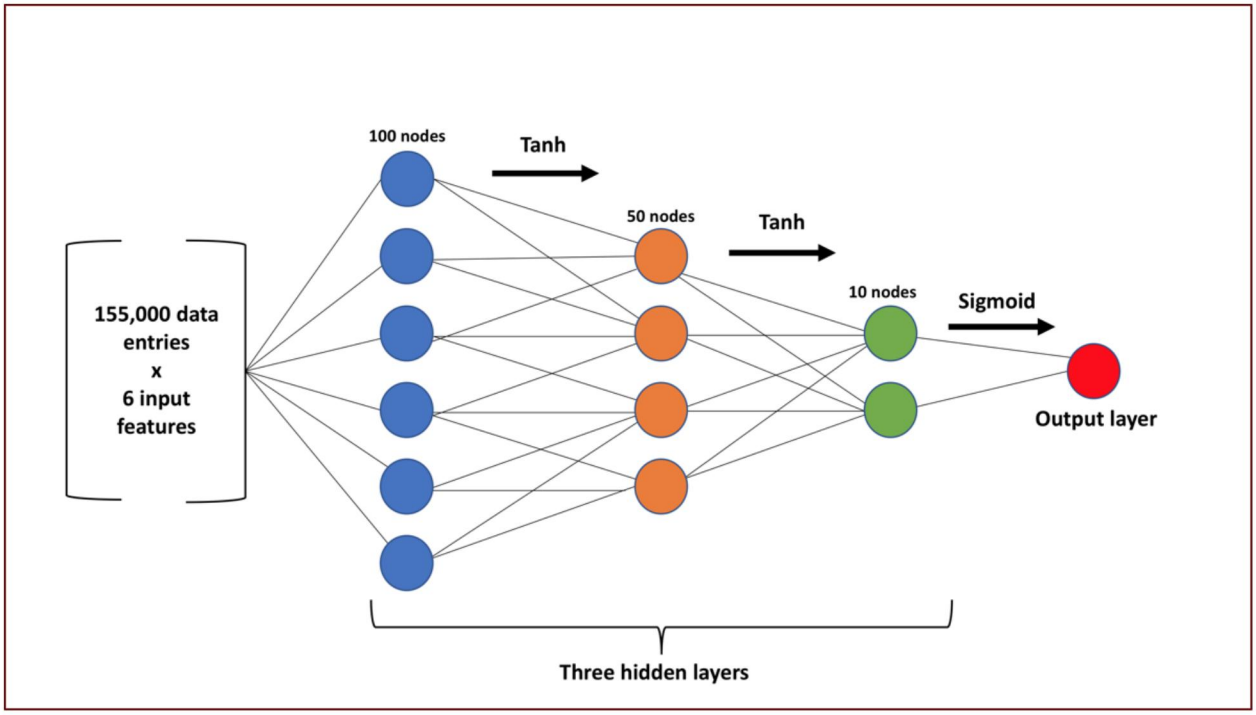
$$z\ =\ wx\ +\ b$$

in each node, where w is a set of parameters, x are the variables and b is a constant for each node. We then apply an activation function, $a\ =\ g(z)$, because our model is non-linear. When we get to an output, we figure out what the cost function is and then backwards propagate to refine our parameter values, all of which were initially randomly

generated. We repeat this cycle until we converge on an optimal set of parameters. Specifics of our model are below.

Experiments/Results/Discussion

We used a neural network to train the algorithm. Our network has 3 hidden layers. The first hidden layer has 100 nodes, the second has 50 nodes and the last one has 10 nodes. Choosing the appropriate number of hidden layers and number of nodes per hidden layer was a result of rigorous trial and error to figure out which model was the most accurate and converged the fastest. In order to be most accurate, we use a tanh activation function on all 3 hidden layers and a sigmoid activation function on the output layer so that we can easily return a binary 1 if the loan is approved or a 0 if the loan is denied.



We split our data as follows: 87% train set, 6.5% dev set, 6.5% test set.
In code, we used Keras to train and test the Neural Network. Through trial and error we tried many different values for hyper parameters. Eventually we settled on a model with Stochastic Gradient Descent decay with a learning rate of 0.05, a decay rate of 0.000001, and momentum of 0.9. The low decay rate (almost 0), basically means that that decay in the learning rate was not helping the learning converge faster, given the 0.05 learning rate. The 0.9 momentum helps accelerate gradient vectors in the right directions, thus leading to faster converging. Similarly, the data was normalized in order

to facilitate parameter convergence. This makes optimization using gradient descent more feasible. It also prevents the function from having vanishing gradient descent.

We also tried running the data through traditional logistic regression and GBM. The table below demonstrates that the neural network trained data had the highest accuracy.

	Neural Network	Log	GBM
Training Accuracy	80%	78%	79%
Test Accuracy	74%	72%	72%

Our model has an 80% accuracy on the training set and 74% accuracy on the test set. We don't know what the Bayes error on this dataset might be, but it looks as if we're not overfitting our data since the error rates of the training and test sets are not differing too much. Additional revisions to the model might include using custom features, converting categorical data to one-hot vectors (i.e. state code), and building a deeper network to further strengthen training accuracy.

Below is a chart of each variable and the relative weight in the GBM model.

var	rel.inf
applicant_race_1	76.442427
loan_purpose	10.629782
loan_amount_000s	4.868266
applicant_income_000s	3.026543
county_code	1.719624
co_applicant_race_1	1.196390
tract_to_msamd_income	1.079772
property_type	1.037197
state_code	0.000000

As the data shows, applicant race is, by far, the most important decider in whether a loan was accepted or rejected.

Conclusion/Future Work

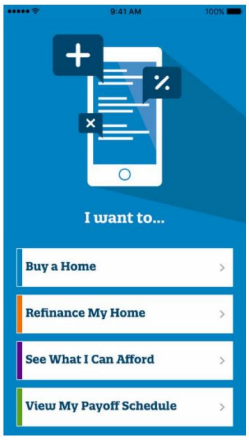
Source: New York Times [6]

The unfortunate truth is that even today, race is the most important decider of whether or not an applicant gets a loan. Interestingly enough, research (mentioned in *Related Work* section) shows that unemployment rates in the zip code contribute most to delinquency rate and that is what underwriters should focus on in making loan decisions. Still, we cannot say for sure that mortgage underwriters are racist. There may be unobserved correlations between unemployment rates and race (possibly employers are racist when firing), and thus race is used as a signal for future unemployment. If this is true, we urge underwriters to learn more about the employer and employer preferences and lay-off rates when making loan decisions instead of using race.

A New York Times article with title “The Race-Based Mortgage Penalty”[6] states “In Philadelphia, whites received 10 times as many conventional mortgage loans as African-Americans during 2015 and 2016, even though the two groups reside in the city in roughly equal numbers… Banks often claim that they deny mortgages in minority communities based on credit scores — but that claim is almost impossible to check, given that the credit scores are not publicly available.”

Race and any variables such as name that can identify race should not be an input variable at all when deciding loans to truly stop racist underwritings.

Our algorithm has many use cases. First and foremost, we hope to use our algorithm to urge policymakers to remove race as a variable that decides an approval on a loan. We also believe we can put our algorithm in an app and enable consumers to figure out whether or not they are likely to be approved for a loan without having to actually apply (and thus getting a hard credit check). A screenshot of such a solution is to the right.



Contributions

All 3 teammates contributed equally to the entire project and discussed and worked on each part together. That said, we all had ownership of different parts in order to play to our strengths and be as efficient as possible. Manjinder mainly focused on coding, building the neural network, and testing different hyperparameters. Aris focused on writing milestone pieces and building the poster. Dasmer found the data for the project, cleaned the data so that it is usable for the model and then owned this final report piece.

References

1. https://paulgp.github.io/papers/MLCreditPaper.pdf
2. https://arxiv.org/abs/1607.02470
3. http://economics.ucr.edu/job_candidates/Bagherpour-Paper.pdf
4. http://tampub.uta.fi/bitstream/handle/10024/102533/1513083673.pdf
5. http://mitsloan.mit.edu/media/Lo_ConsumerCreditRiskModels.pdf
6. https://www.nytimes.com/2018/03/07/opinion/mortage-minority-income.html