Loan Approval prediction

Introduction

Automation of loan approval process is one of the most common problems which are used for machine learning practice purposes. To get a loan person or business has to get approval from bank who is lending them money based on information bank is asking you to submit while applying a loan. This process is usually seen as slow when done by humans and also in certain levels it can be inaccurate as well. Putting machine to decide or either give you a risk score, can be seen as faster and unbiased way of doing this same decision.

In a heart of the problem it's a binary type ”yes/no” decision where person/business either will get allowed or disallowed for the loan. This problem can be evaluated also in other ways. One of them is giving each applicant a risk score and using this risk score we can do the binary divide based on the banks current situation. Problem could be also seen as 3-way prediction where our machine learning model gives us a likelihood of approval as a number between 0 and 1 and we cut these predictions in 3 groups of ”yes/maybe/no”. In basics this access somewhat similarly as a risk score but it also can give human a possibility to override the mathematical solution in uncertain cases.

This third option of 3-way grouping is the one which we're exploring in this example. Will work with artificially created dataset and we explore 1) how well our model works in general binary form prediction, 2) how big part of the predictions will end up into this new third category of ”maybe” and 3) how well our model can predict the uncertain inside the borders for the third category ie. is this method useful in the first place at all? Dataset was found from Kaggle.com (link in references) and for modeling needs XGBoost was used to create the predictions.

Dataset and preprocessing

This dataset was artificially created and the it could be used for all three prediction problems introduced above. It has 20,000 observations from different applications with total of 33 features to model the 2 possible target features with. These features include your basic information like age, income, credit score, job status, education, work experience etc. But also some information from earlier loan payments, loan purpose, assets, liabilities and so on. It also has information how much loan is applied for, what is interest rate and also couple ratios like debt to income ratio.

Most of the variabes are already numerical but there are few categorical variables mixed in. Job status, education, marital status, home ownership and loan purpose had to be changed into numerical form. Some of the numerical values also needed alternation into logarithmic version to make them more suitable for future modeling. As the dataset was artificial, it didn't have too many difficult bits like missing values and therefore it was good to go fairly quickly.

Modeling

This part was performed in RStudio with XGBoost method. This model gives us a prediction between 0 to 1 which we can rework in a way we are interested of. As the dataset was not split before downloading it we had to split it ourselves into training and testing datasets with 80%/20% split respectively. We also performed a vague hyperparameter tuning which basically contained 3 values for learning rate (0.15, 0.20, 0.25) and 3 values for max depth of the model (4, 5, 6). Evaluation metric was AUC.

For this time the main interest was to group these predictions in 3 groups. For the testing purposes it was decided that loan was approved if the prediction score was above 0.6, not approved when below 0.4 and everything inbetween was considered being in third group of ”maybe”. We also modified predictions into a binary form where above 0.5 meant approving and below not approving and at the end we filtered few datasets from the testing dataset to compare the results and answer the questions set in introduction.

The best model was achieved by following parameters:

- 0.15 learning rate
- 4 max depth
- 96 rounds

Results and analysis

In general with binary prediction in mind this model performed really well, having a 95,9% accuracy in testing dataset. The error types were also fairly in line (type I error 2,4%, type II error 11,2%). The testing dataset was slightly unbalanced however having roughly 25% applications being positive so this might have an effect on type II error being percentually bigger. In this case type I error can be the more detrimental however, since banks probably would like to avoid applying loans for people who are not valid for the loan in their eyes.

The total number of incorrect predictions out of the 4,000 cases (observations) test dataset was then 177. When dividing these predictions into three groups we got total of 141 cases where the prediction from the model landed inbetween of values 0.4 and 0.6. This already rings a bell as the cases that were unclear based on the model is smaller than the incorrect predictions from the same model. When investigated further we found that only 57 cases of the 177 incorrect predictions landed inbetween of this third category of ”maybe” which is only 32% of all them. This means that 120 of the incorrect predictions did get higher, or lower, predictions score and would’ve end up going through without a question if this model would’ve been used without consideration with these limitations.

A closer look to these predictions showed that the incorrect predictions got scores from 0.03 all the way to 0.93. This confirms that even with widening of the ”maybe” range, we couldn’t catch all the incorrect cases from the testing dataset. There are of course also 84 cases where prediction score did land in the ”maybe” range but were predicted correctly in binary grouping. In the banks eyes I don’t see a problem with this as the group in question is only 2% of all the customers and assumption is that the ”correct” answer will be delivered either way.

Concerns

With given results created some concerns. First of all it's good to raise up the issue of the data being artificially made which mean that the results can be only interpret from practice point of view. It also can be affecting on the prediction accuracy and it shouldn't be trusted that in real life situation model could predict approval with nearly 96% accuracy. Most likely however even with real life scenario model will get results where predictions will be incorrect and the predictions score can vary as highly as in this example it did as well. It means that creating automation for the approval system is questionable and in most cases there should be human consideration within the process as well. From this point of view the risk score would probably be better indicator for application process than predicting in binary or 3-way form.

The main issue however which should be raised with this type of a problem should be that the training dataset in real life will contain old human decided cases of loan approvals and the reality is that training our model with this kind of dataset will always lead into situation where model can't predict correctly if loan should be approved or not. Since this inconsistancy is built in to the real world data, it's not possible to ask a model to give us perfect results even with this kind of more conservative 3-way grouping style. It's also good to mention that with this model all the variables were equal pre-modeling phase while assuming this dataset was a real world case, some of the decisions were made with different kind of prioritization, both in good and bad. Our model couldn't catch these anomalies and to have closer look to why did some predictions end up being so badly incorrect, the best way to analyze this would be having reasoning added to why each individual application was approved or not approved. We obviously don't have this kind of data as this was artificially created.

Conclusion

In this project we did model a 3-way grouping of loan applications from artificial dataset and compared how well it did perform with factors set in before hand. The results were that it did perform well but also did leave room to improve and end up into a realization of that dataset has quite a lot of impact which our prediction as well and since it can be assumed that there are biases in the earlier loan approvals, this will show as errors in our model predicting future cases. For this reason it might be better to make model predict a risk score instead which can be used as evaluation metric when doing a human decision of each individual loan approval case.