Predicting unsafe bank loans using machine learning algorithms

Tim16_21: Nikola Vujačić, Ilija Grk, Slobodan Zelić

1. Motivation

The main motivation behind choosing this subject was more or less based on the fact that we wanted to implement everything that we learned on this subject on something practical. In other words we wanted to find a subject that has strong connections with our possible future jobs if wanted to go on this path.

Statistical analysis and predictions are highly used in banking and due to Slobodan recently geting his driving licence we kinda found the middle ground and chose the subject that links those two things together.

2. Research questions

There were quite a few questions that needed to be answered. Most of them were based on the data set itself (which is only logical considering that we are making a highly important prediction based on a high number of data set columns). Most important of those questions were:

1. Is the dataset complete and does it contain any faults ?

2. Because of the large number of columns, do they all equally affect our final score, or do some columns affect our score more (if so which one and by how much)?

3. Is column reduction or addition necessary?

4. Model selection, what models to pick and which ones performed the best?

5. What metrics to chose, and what results did they give in corresponding to different models?

3. Related work

Considering that the data set is public and we had no means of collecting similar data set on our own, there have been prior projects done with the same data set. Also, considering that the goal of the project is also publicly known in advance, different projects already exist on https://www.kaggle.com. They are mostly focused on the same goal as us.

Generally speaking the main goal is to optimize the data set and use a fitting model on the set itself. Optimize it properly and you did 95% of the work.

4. Methodology

We mostly used methods that we previously used on our homeworks with additional preprocessing done on the data set as it is far more complex than the regular ones.

First we made sure that our data set contains no null values considering the fact that they can affect our metric accuracy by a huge amount. We decided the best way to deal with those values is just to drop them as they only made up to 4% of a huge 200 000 row data set.

We also focused on droping unecessary columnns to reduce our component number and achieve higher metric accuracy. We dropped the columns ither through data extraction or column combinations as some dual column values could actually be added into a single value.

We also focused on the logic itself as we found there was no way that a bank would issue a loan to a person that is 121 years old so we exchanged those values to something more viable as we thought it was an unintentional mistake made in the process of collecting data (in other words a typo)

We also dropped columns that have a variance of 1 meaning that if they get dropped they do not affect our metrics score considering that every row contains the same value.

After all the columns have been preprocessed, we normalized each column

and focused on model selection.

The algorithms we used were mostly the most commonly used ones. For some of them we used hyperparametrizations to find the proper arguments while some were manualy trained simply due to the fact that our machines could not really handle constant hyperparametrization as in some instances the program would run for more than 12 hours.

5. Discussion

Model performance did vary quite a bit although all of the models have been performing relatively well.

Even the binary oriented Logistical Regression had an accuracy score of 0.77 which is decent considering the multi variable nature of the data set.

Most of the other algorithms including DTC, RFC, LGB had a varied accuracy score ranging from 0.7 to 0.9.

KNN was a close second having an accuracy score of 0.94 although the computing time is quite extensive and long, so in raw performance i would hardly consider it a top contender.

XGB performance was excellent toping at 0.992 accuracy score toping even the AdaBoosting and Voting models. The closest model to the XGB performance was actually a Voting ensemble with XGB at its core providing an accuracy score of 0.97 but yet again it is pulled back by the processing speed.

Even when using more advanced metrics like f1_score XGB retains its 0.98 score severly beating any other model or ensmeble by quite a bit.

Comparing the results of every model in multiple tests we have come to a conclusion that nothing beats XGB in performance and accuracy and as such it is best suited for the subject at hand.

6. References

1. https://www.kaggle.com

2. John. D. Kelleher Fundamentals of Machine Learning for Predictive Data Anayltics (Algorithms, Worked Examples, and Case Studies)

3. https://scikit-learn.org/stable/