Project 1

Habibur Rahman Dipto

kazi.dipto@northsouth.edu
North South University
Department of Electrical and Computer Engineering

June 30, 2021

Abstract

In our modern day life, tabular data is the most popular type of data to keep record of day to day life task. Though we have a huge collection of tabular data there is no new SOTA model to understand the tabular data more. Even a regular tree based machine learning model is good for these type of problem. Since people don't have to worry about the model that much, the main focus/challenge is to feature engineer the data. And try to find some meaningful value from the data.

I. Methodology

The problem I am discussing here is IEEE-CIS Fraud Detection. It's hosted in kaggle. The problem here is, from a large dataset of customer transaction figuring out whether the transaction is fraud or not. For this competition Vesta offered us a huge dataset containing 435 features of 590540 records. I mapped all the missing value with a large negative number, text with discrete number. Since a most common practice for ML/DL work is to feed all the data into the largest model possible and check what are the accuracy it can get without any feature engineering. I chose a vanilla FCNNs (Fully Connected Neural Networks) with relu activation function and dropout of .5. It was a 6 layer architecture and the final layer had only one neuron (Since the output is a binary number is Fraud 0/1). The learning rate was .001 and the optimzer was Adam. The loss function i used is BCE (Binary Cross Entropy).

The accuracy I hit is 88.34% which is quite good. Next I tried to feature engineer our data. In order to feature engineer we need to understand the data first. There was a brief discussion in kaggle about the dataset. I go through it first and try to understand the nature of the data and what it wants to tell. Most of the columns are independent but there are some columns which shares the result among themselves. One of the group of columns V_n which is basically Vesta engineered rich features, including ranking, counting, and other entity relations. So I apply correlation there and find some subgroup of V are highly correlated. We then applied mean encoding among those subgroups to extract their correlated information and concat the column with training dataset and removed those subgroups. The same technique we used for columns C_n which are for counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked. The columns D_n are keeping track of the past transaction in ms which is technically a very large number. And for any basic ml/dl problem if any feature gets a very large range of value the activation threshold gets smooth for that feature. So we converted that into day. The final shape of training data is 590540x263. We split our data into 75/25 train, val set. We then fed data into LGBoost classifier first because it's a tree based classifier, it's faster than xgb. We used ROC AUC score as evaluation metrics. So we fine tune our classifier hyperparameter and got 94.74%. Finally we use xgb with that hyperparameter and got 95.18%

II. Result Analysis

Without feature engineering the model get 88% accuracy. And with feature engineering and some fine tuning we get around 95%. By observing those result we can say feature engineering is very important since you there are not many new implementation of ml/dl model to extract the feature from tabular. So we need to manually feature engineer in order to boost the result.

i. Conclusion

The current model gives us a very good result. But to go further we can use GridSearch or other feature engineering. Or maybe we can use introduce FCNN with LSTM to keep track of necessary information and discard which are not important. We can feature engineer further in V_n and M_n columns as well.

III. Acknowledgements

I would humbly like to thank both Upskill and IM for the discussion about why there are limitations of using NN in tabular dataset.