ML Project Presentation 3

Team: Sober ML Engineers

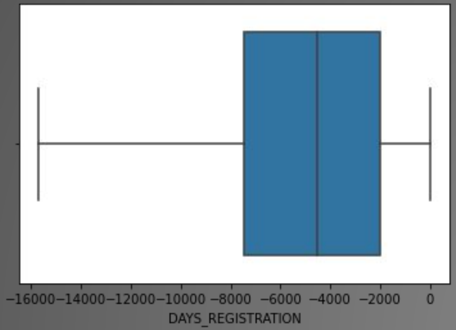
Project: Home Loan Default Risk Prediction

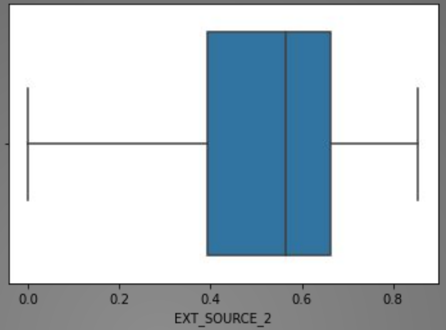
Chinthan Chandra
Shashank Shekhar

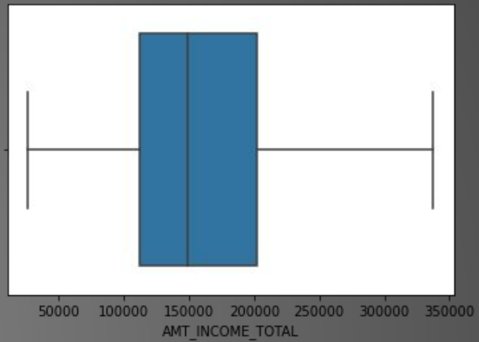
Outlier Correction

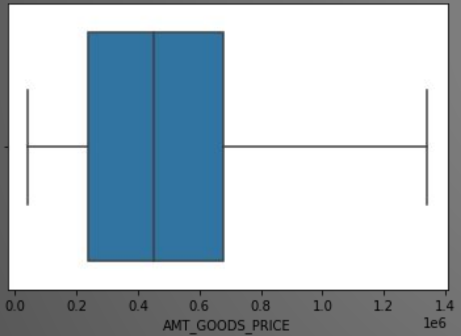
For the outliers in all the numerical columns, we have corrected the outliers to the respective whiskers, i.e the negative outliers are replaced with the value of lower whiskers and the positive outliers are replaced with the values of upper whiskers.

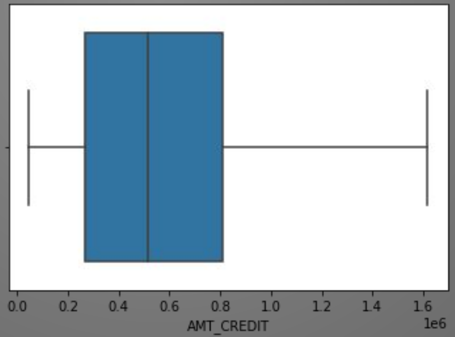
Last time we had removed a part of the outliers, based upon the last project review we thought of trying this.

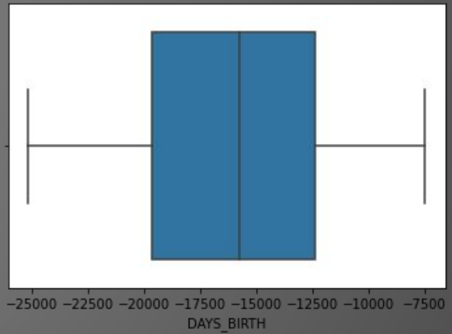

−16000
−14000
−12000
−10000
−8000
−6000
−4000
−2000
0
DAYS_REGISTRATION


0.0
0.2
0.4
0.6
0.8
EXT_SOURCE_2


50000
100000
150000
200000
250000
300000
350000
AMT_INCOME_TOTAL


0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e6
AMT_GOODS_PRICE


0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6
AMT_CREDIT


−25000
−22500
−20000
−17500
−15000
−12500
−10000
−7500
DAYS_BIRTH

Resampling Data

The data was imbalanced, so we planned to do sampling. We tried SMOTE, which is synthetically minority oversampling technique. In this the minority samples are synthetically produced to the given ratio. We have used the sampling ratio as 0.3.

We wanted to oversample on the minority class and undersample on the majority class, but the undersampler was taking too much time to run.

We decided to only oversample and check results.

Models tried with sampling and Outlier correction

- Logistic regression
- Decision Trees
- XGB
- LGB

The results obtained are discussed in the next few slides.

Logistic Regression

Logistic Regression was the basic linear model we had used to do our first submission, so we thought this would be a good model to start off with.

We tried logistic regression using the lbfgs solver and iterations capped at 10000.

This gave us a score of 0.52785 on the leaderboard.

We tried to tweak the regularisation parameter C and increase iterations to achieve better results, and we got a kaggle score of 0.59305, which was impressive.

So we tried to change solvers and do hyperparameter tuning, but the result didn’t seem to improve on validation data.

Decision Trees

This was one of the models which gave good results with resampling, so we tried this with resampling.

The results didn’t increase the score. We tried hyperparameter tuning by changing the max_depth, the splitting criterion as entropy instead of default gini etc.

None of it seemed to increase the score over 0.52.

XGB

This was an ensemble method, so we thought trying this with the resampled data would give optimal results.

The parameters used were :

```
classifier=xgboost.XGBClassifier(learning_rate =0.02,
n_estimators=10000,
objective= 'binary:logistic',
nthread=8,
eval_metric="logloss",
scale_pos_weight=2,
seed=42)
```

This gave a score of 0.56952 on kaggle.

With a little bit of hyperparameter tuning we would have obtained a little better results but the model running time was not fast enough to check the results. So we decided to not try the tuning.

LGBM

This model took relatively less time to run amongst the ensemble methods.

So we tried hyperparameter tuning on this extensively. The highest score we achieved on kaggle was 0.58979.

After a lot of tries in the limited time, the results did not seem to improve more than 0.60629.

So we dropped the sampling and outlier detection.

Best Result

The best result was obtained by hyperparameter tuning the LightGB model with no outlier removal or correction and no sampling.

The results we obtained were:

```
Accuracy for the train(total) data for lgb is: 0.9364166549230886
f1 score for the train(total) data for lgb is: 0.8408228435960925
kaggle: 0.60775
```

This was our best submission, we initially tweaked the estimators, the bagging frequency, the regularisation parameters, used cross-entropy loss, etc.

Things done after last checkpoint

- Data sampling
- Outlier correction
- Parameter tuning for LightGB.

These were the few things we could with the limited time we had because of end sems.

Things learnt in the Project

It’s the unexpected models which give better results at times.

Outliers need not always make the model bad, if there is good amount of them they matter a lot in training the model.

Ensemble methods are the good models, but trying them directly without hyperparameter tuning will not give good results.

No matter what the model is, every model requires hyperparameter tuning, so never give up when you get results which you didn’t expect on the default parameters.

Validation is important. Gives you insights on the model hyperparameter tuning.

Thank you!