American Express Campus Analyze This 2018

Team Name: Give_us_some_credit
Leaderboard Rank: 26th

Team Details:
Neha Mokmod (navya.mokmod@gmail.com**)**
Tannmay Yadav (tanmayy98@gmail.com**)**

Estimation Technique Used:

1. Decision trees- Light GBM

- We Implemented Decision Trees with Light gradient Boosting Machine variant in order to model the given classification problem.
- Light GBM gave us better accuracy on our local test set as well as on the Leaderboard.
- LightGBM also handle categorical features by taking the input of feature names.
- It does not convert to one-hot coding, and is much faster than one-hot coding. LGBM uses a special algorithm to find the split value of categorical features

2. Bayesian Optimization:

- Optimizing hyperparameters with bayesian optimization helped us get the best value of the hyper-parameters.
- Bayesian error is the minimum error which cannot be improved further and thus using Bayesian Optimization on our model gave the best result.

Strategy to decide final list

1. Threshold Strategy using probabilistic sorting :

- From training data we inferred that almost 75.73% of the total population are not defaulters.
- Bank has a budget of $50000, implying that the maximum evaluation possible is 10000
- To ensure best candidate's applications are processed first:
- We set a threshold of 0.75 on the prediction probability such that only candidates whose probability of not defaulting is greater than 0.75 will be accounted as eligible candidate
- Since most people do not default our assumption helps us find the best candidates.
- For obtaining eligible candidates list
- We calculate the probability of each candidate to be a defaulter
- Sort them in decreasing order with respect to defaulting probability and thus obtain our final submission list

2. Missing Data Imputation:

- We selected variables where medians were imputed because filling in zero directly, because the data was not available and it was logically not making sense to fill in zeros
- and some of the variables like mvar45, 44,39,25,26,28,29,12 were imputed with zeroes as it was logically aligned with the overall behavior of the dataset

Details of each Variable used in the logic/model/strategy:

1. Feature Importance:

Generating null importances distributions:

- These are created fitting the model over several runs on a shuffled version of the target variable. This shows how the model can make sense of a feature irrespective of the target.
    - for each feature test the actual importance:
        - Computing the probability of the actual importance wrt the null distribution.
        - comparing the actual importance to the mean and max of the null importances. This gave us sorted list of a feature importance that allowed to see major features in the dataset.

2. Selection of variables:

Dropped the following variables mvar43, mvar35,mvar19,mvar20 by analysing the split importance and information gain obtained from above strategy

3. Feature Engineering :

Engineered new features by merging the related variables, to reduce the number of features in the model so as to prevent the model from overfitting.

- Engineered Variables:
    - mvar3_4_5 = mvar3+ mvar4+ mvar5
    - mvar11=mvar12-mvar11
    - mvar27= mvar27-mvar25
    - mar40=mar40/mvar31
    - mvar41= mvar41/mvar30
    - mvar45= mvar45+ mvar46

4. Correlation Analysis:

Correlated features have decaying importances once one of them is used by the model. The chosen feature will have strong importance and its correlated suite will have decaying importances.

Correlation :

- (mvar 18,mvar17)= 0.93,
- (mvar27, mvar 7) = 0.91
- Removed highly correlated variables mvar 17, mvar16, mvar 7 = 1.0

Reasons for Technique(s) Used

- When we fit a decision tree to a training dataset, the top few nodes on which the tree is split are essentially the most important variables within the dataset and feature selection is completed automatically!

- To overcome scale differences between parameters - Such variable transformations are not required with decision trees because the tree structure will remain the same with or without the transformation.

- Decision trees do not require any assumptions of linearity in the data. Thus, we can use them in scenarios where we know the parameters are nonlinearly related.

- Then we chose to implement Decision trees for classification which was giving better results than the previous methods :
- XGBoost (0.7832)
- CatBoost (0.7534)
- AdaBoost (0.7452)
- LightGBM (0.7873)

- We also tried implementing other predictive methods to model given Problem statement like Support Vector Machines, Random Forest Classifier, Logistic Regression, Deep Neural Network Models to compare the predictive ability and the accuracy of the model
- Amongst all these variants of predictive models LightGBM remained unbeatable!