The solution to Credit card fraud detection

By Mohammed

NOTE

- Please note that I will send only one notebook with phase 3 of the project because it turned out the steps in completing phase 3 were short and I did not feel the need to have separate notebooks and presentations.

Feature selection

- I decided to not drop any features as I thought all the features might be important in giving me the best performing model. It turned out that indeed having all the features was helpful in giving the best performing model.

Shortlist Promising models

- Due to the heavy computational resources, I tried couple of models such as logistic, random forest and xgboost. It turned out that the random forest and xgboost were the best models that gave exceptional results. But Xgboost was a bit better in terms of recall and precision. So I did tuning on both random forest and xgboost. Xgboost has a higher recall score so I moved on to solve the problem with xgboost.

The model

- I went on to select a threshold with the xgboost. I wanted to have the recall be at least 90 and maximize the precision as much as possible. I wanted to catch as many fraudulent cases as possible.

The results for validation set

- I obtained the following: Best Threshold: 0.7585858585858586
- Best Recall Score: 0.9090909090909091
- Best Precision Score: 0.569620253164557 I reached the objective of having a score of 90 on recall which means the system correctly identifies 91% of fraudulent transactions. For the precision 0.57 means that out of all transactions flagged as fraudulent, around 57% are actually fraudulent.

The results on the test set and conclusion

- I obtained the results for the recall and precision: XGBoost with Best Threshold - Precision: 0.6194029850746269
- XGBoost with Best Threshold - Recall: 0.8469387755102041 These are good scores not a huge drop for the recall score in here. There is no sign of overfitting or underfitting. In conclusion, I have met the business objective.