EE 510: Machine Learning Theory & Algorithms **Spring 2020**

Mini Project 3

Due: June 5 (official) / June 12, 2020, 11:59PM PT

Student Name: Andrew Huffman *Instructor Name: John Lipor*

Note: The "official" due data will be June 5, meaning I will not give feedback or answer questions after that date. However, you are welcome to take the extra week to continue working on your project, turning it in on June 12.

Problem Description

In this project, designed by Porto Seguro, a Brazilian auto insurance company, the goal was to predict the probability that a policy holder will file a claim in the next year [1]. Porto Seguro provided 57 unknown features, which belonged to three classes—binary, categorical, and continuous—and binary training targets representing whether each training policy holder filed a claim. The probability of each test policy holder filing a claim was predicted, and the predictions were evaluated using the normalized Gini coefficient.

Exploratory Data Analysis (EDA)

I found this Kaggle notebook [2] helpful for performing exploratory data analysis on this project. The most significant characteristic of this dataset is its high degree of imbalance. As show in Figure 1, the training targets are dominated by the not-filing case. I found this Medium article [3]by Baptiste Rocca informative regarding imbalanced datasets. The correlations between features are plotted in Figure 2. Figure 3 shows the training data projected into two dimensions via truncated singular value decomposition. The classes are highly mixed. Figure 4 shows the UMAP embedding of a randomly selected subset (to reduce computation time) of the training data. This embedding technique does not yield good class separation. Due to the apparently low separability between the classes and vast overrepresentation of the non-filing targets, I expect that it will be difficult to correctly predict cases when a claim is filed.

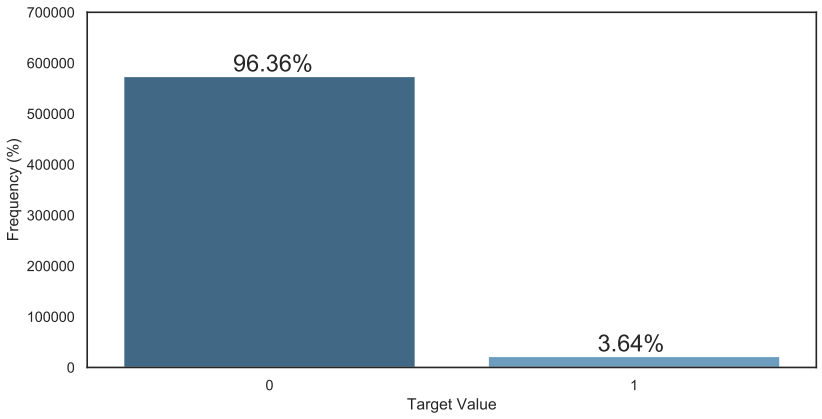

Figure 1: Target Distribution

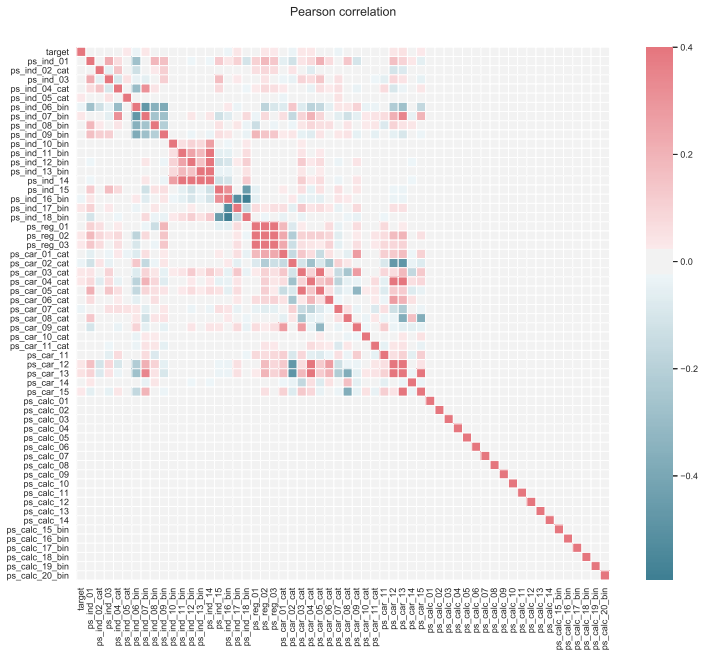

Figure 2: Feature Correlations

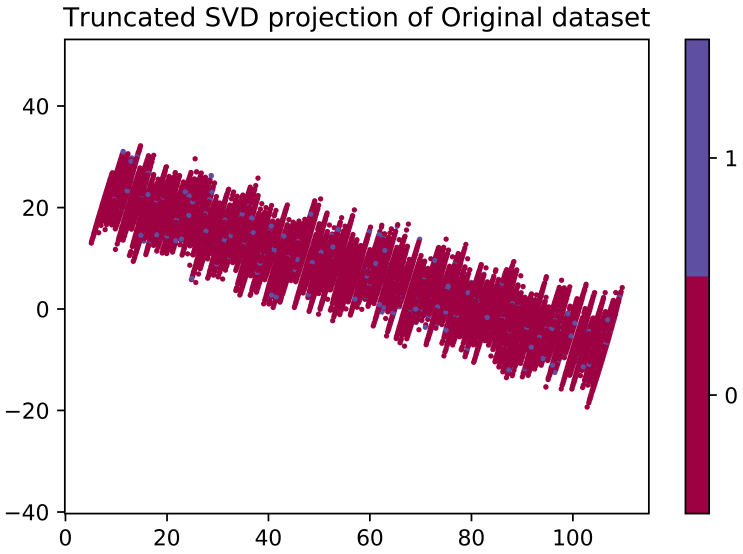

Figure 3: Truncated SVD Projection of Training Data

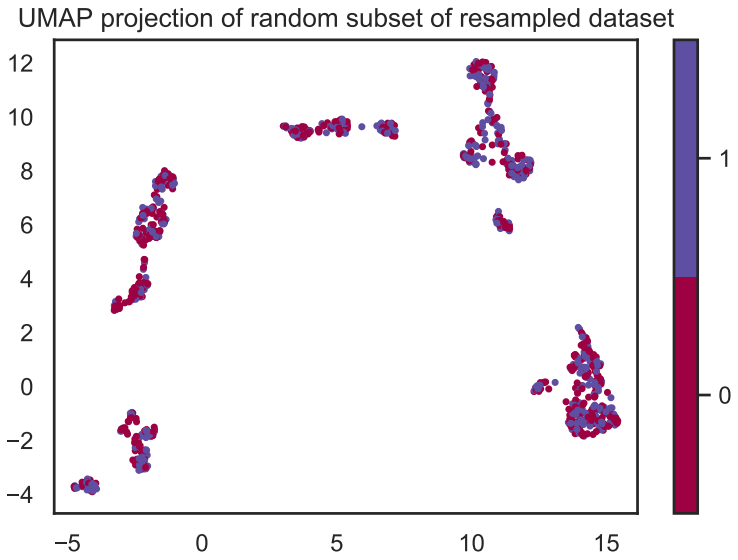

Figure 4: UMAP Embedding of Randomly Selected Subset of Training Data

Challenges

The most challenging aspects of this project were from the data. Not knowing what the features represented removed the opportunity to use intuitive reasoning in understanding the relationships represented in the data. Another challenge was that the data were highly imbalanced and not easily separable. It was easy to achieve high accuracy by predicting no claim for essentially every case, but correctly classifying cases when a claim was filed was difficult.

Approach

I tried two main approaches to this problem. First, I tried using the provided training data to train and validate my predictor without any resampling to reduce the effect of training target imbalance. Second, I tried under sampling the majority class, and then training and validating from this resampled training set. In both cases I used sklearn's standard scaler to scale training and test data to improve model performance. Originally, I tried using XGBoost but I switched to using the LightGBM gradient boosting framework for this problem because it is significantly faster and performs well relative to XGBoost results I found from Kaggle notebooks. I used a validation set to select the learning rate and number of trees to use in making test predictions. Figures 5 and 6 show model selection curves for models trained on the original dataset and a resampled dataset, respectively. The results were not drastically different, so I selected two models to make test predictions to submit. I used learning rate = 0.1 and number of leaves = 10 for the model trained on the original dataset. I used learning rate = 0.1 and number of leaves = 20 for the model trained on the resampled dataset.

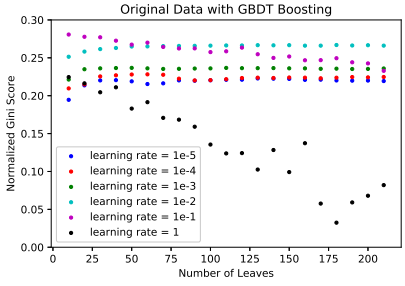

Figure 5: Model Selection Using Original Data

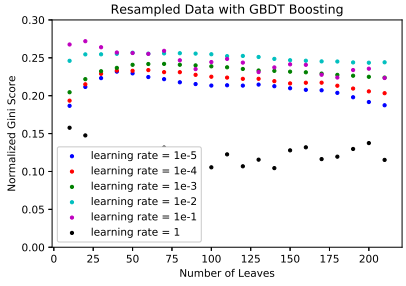

Figure 6: Model Selection Using Resampled Data

Evaluation and Summary

For this dataset choosing a good evaluation metric is not totally straightforward. The imbalance in the training data, which reflects real-world conditions, makes it easy to predict with high accuracy (greater than 96%) whether a claim will be filed. However, this accuracy is misleading in that the model can actually just never predict that a claim will not be filed and still perform well by this metric. If the priority for the model is to provide insight regarding when a policy holder will file a claim, then such a model would be of no value. If this is the priority, then a metric that emphasizes the model's ability to distinguish between classes rather than merely correctly label a larger proportion of examples is more useful. This competition used the gini coefficient to evaluate submissions. This metric is intended emphasize a model's ability to distinguish between classes more than accuracy does. In my opinion, true positive rate and sensitivity could be valuable metrics to measure a model's ability to predict the occurrence of a claim failing.

Using the original data and the model described in the previous section, yielded a test score ≈ 0.279. Using the resampled data and the model described in the previous section yielded a test score ≈ 0.271. Figures 7 and 8 show the confusion matrices (from validation data) for these two models. The model trained on the original data achieved a slightly higher gini score, but the model trained on the resampled data is much better at predicting that a claim is filed (the "1" class).

Figures 9 and 10 show importance plots by feature from the two trained models. There is a significant disparity in importance across features. Perhaps this characteristic could be used to develop a better model.

In summary, I trained two LightGBM models to predict the probability of a policy holder filing a claim. One model was trained on the original, highly imbalanced dataset. The other was trained on a resampled dataset using undersampling of the majority class. The first model yielded a slightly higher gini score, and so it was better for the purposes of the Kaggle competition. However, it essentially never predicted a claim would be filed. The second model correctly predicted around 58% of filed claims and about 62% of non-filings. Both models are limited, and more work needs to be done to train a model capable of achieving high accuracy for both filing and non-filing. To do this, I would suggest more extensive feature engineering/selection to try to uncover stronger predictive relationships in the data.

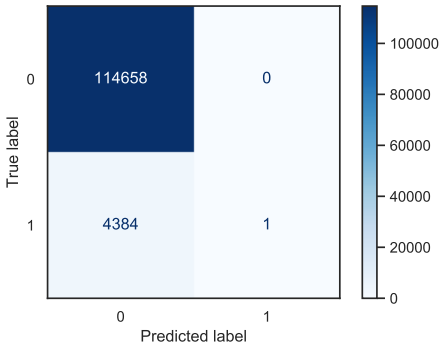

Figure 7: Confusion Matrix Using Original Data

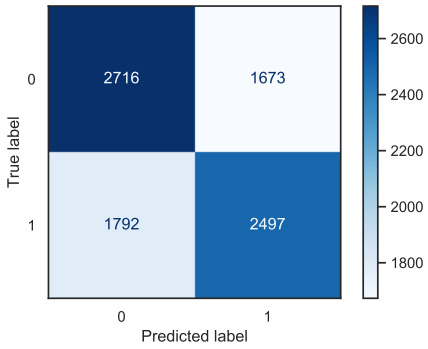

Figure 8: Confusion Matrix Using Resampled Data

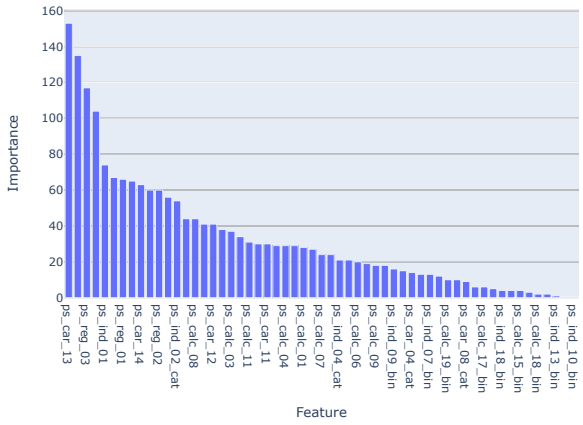

Figure 9: Feature Importance from Model Trained on Original Data

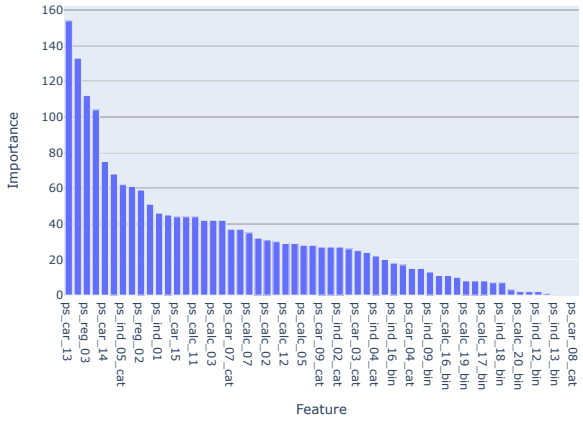

Figure 10: Feature Importance from Model Trained on Resampled Dataa

What I learned

The main skill I learned from this project was exposure to working with an imbalanced dataset. I had never worked with such a dataset previously. This taught me that is important to analyze what information performance metrics are actually conveying rather than naively thinking that high accuracy implies a good model. Depending on the nature of the data and the purposes of the model, this interpretation could be incorrect.

References

[1] (2017) Porto seguro's safe driver prediction. [Online]. Available: https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/overview/description

[2] (2017) Stratified kfold+xgboost+eda tutorial. [Online]. Available: https://www.kaggle.com/sudosudoohio/stratified-kfold-xgboost-eda-tutorial-0-281

[3] (2019) Handling imbalanced datasets in machine learning. [Online]. Available: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28