Tutku Kılıçaslan
54201
ENGR 421
Homework #8

CLASSIFICATION PROBLEM: Modeling Credit Card Customer Behaviors

Objective

The main objective is to classify each customer's credit card behavior according to their various types of data. We are given a huge data set with both categorical and numerical values and a separated label set with six different types of label. Our aim is predicting the credit card behavior of the customers in the test set. At the end, we assign posterior probabilities to the customers for each type of label. Our measurement will be AUROC.

Observations

As I observed, we have lots of NA values in the label set. Addition to this, we have some type of categorical values like R4. We need to tackle both of these problems.

Brainstorm

- Most proper algorithms for our problem are **LDA or Decision Tree.**

- We can get higher accuracy also by using **different parameters**.
  Some algorithms are sensitive to parameters. Therefore, performing different parameters can give us better results.

- We can get higher accuracy by using **different input representations.**
  We can give different input to the algorithm and we can get higher accuracy by using these different results from different algorithms. For our data, use the Xgboost method to train our model with different types of input to get a higher AUROC value.

My Solution Method

First I applied the LDA model but I didn't get better results. Therefore, I ended up using xgboost.

Before applying any model, I needed to do one-hot encoding to organize some features. After that I need to handle NA values in the label set. For this we only use the data of customers who have corresponding labels for the specific label. This was reasonable because we have still enough data to train our model even after eliminating irrelevant customer data.

After eliminating irrelevant customer data, I tried to perform a validation test. For this I separated customer data with label 0 and 1 and chose 90% of them into my training set. After that I shuffled the training set before applying xgboosting.

The remaining 10% of data becomes my validation set. Again, since I have a huge amount of data, I could sacrifice %10 of my data as a validation set instead of a training set.
After applying xgboost for each label I got very high AUROC values.

I got the AUROC value of validation set for the first label as **0.8905.**
I got the AUROC value of validation set for the second label as **0.7680.**
I got the AUROC value of validation set for the third label as **0.7916.**
I got the AUROC value of validation set for the fourth label as **0.8014.**
I got an AUROC value of validation set for the fifth label as **0.7503.**
I got the AUROC value of validation set for the sixth label as **0.7943.**

Overall, xgboost gave me the best result among other algorithms. These AUROC values can be evaluated as successful since they are greater than 0.7. Even for the first label we reached almost 0.9 which means that my classifier is very successful in terms of labeling the customers. These values are for the default parameters and round number 20 as in the quick and dirty solution. Later in this report we will discuss the effect of the parameters on AUROC.

Since my algorithm works well for the validation set, I assumed that my algorithm will work well for other test sets as well. Yet, we expect to get lower AUROC levels since always training errors are less than test errors.

Xgboost algorithm gave me the AUROC values 0.7503 and 0.7680 for the fifth and second labels. This result shows me that my approach may not be the best one for these labels. This approach can be improved by using different algorithms for these columns.

Therefore, I assumed that the credit card behavior of the customers in test sets can be predicted by this model.

Parameters

Firstly, I tried changing the parameter maxdepth which defines the maximum depth of the trees to be grown. I set the parameter as default, 6 and 15 I received the values 0.89, 0.74, 0.71. Therefore, I left it as default.

After that I set the parameter nround which is the number of rounds for the xgboosting method as 10,20 and 30. As I increased the value, I got higher AUROC values. However, I kept it as 20 for computational reasons and time constraints. It sounds very reasonable since these parameters provide diversity.

I also set eta as various values but again the default value of eta gave me the higher AUROC.

Lastly, I change the validation set size. As I changed the ratio from 0.8 to 0.95 AUROC value increased as we expected. Because if we set the training size larger, our machine will have more examples. Yet, from 0.90 to 0.95, there was little change, so I kept it as 0.9 in order to decrease the processing time for training.

Improvement

In the solution key, we were given an xgboost model. The solution was quick and dirty. Since we have enough data, I tried to improve the solution by adding a validation test and calculate AUROC. This AUROC is a proxy for test set AUROC level.

I improved the AUROC values from **0.9830176** to **approximately 0.80** for the labels at average. At first it may seem like a negative improvement. However, where AUROC has almost value 1, it means that our algorithm memorizes the data and does not learn it. For a good classifier we need the value more than 0.5 and less than 1. 0.7-0.9 are very good classifiers.

Conclusion

During the semester, we have learnt to implement different algorithms. However, one algorithm does not always give the best solution. We can improve our solution by taking the advantage of diversity. In my solution, I improved the diversity of my input representation by using xgboosting. Xgboost uses the decision tree model as a base learner and trains the decision tree by using different input representations. Therefore, it includes a more diverse approach than the decision tree. Our performance measurement was **AUROC**. At the end of the implementation, I obtained **a significant improvement** in the AUROC value for the validation set. It is important that this solution may not be the best solution but it solves our problem and it is expected to predict the labels for the test set in a successful way.

Resources

- https://xgboost.readthedocs.io/en/latest/tutorials/index.html
- https://towardsdatascience.com/doing-xgboost-hyper-parameter-tuning-the-smart-way-part-1-of-2-f6d255a45dde
- https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
- https://stackoverflow.com/questions/11467855/roc-curve-in-r-using-rocr-package