Tide Business Data Scientist Interview Task Report

Abdul Waheed
abdulwaheed1513@gmail.com

1 Problem Statement: Receipt Matching Data Science Challenge

Our customers want the ability to automatically match receipt images to the associated transaction within the tide app. For this we have used an external supplier to extract data from the receipt images, (for example, the date of the transaction on the receipt, the name of the merchant, the transaction amount). Each of the data items returned from the external supplier was compared against the same data from the tide transaction, and a 'transaction-receipt' matching vector was created. The elements of the matching vector will be produced in different ways depending on the underlying data types being matched (for example for strings it may be a fuzzy matching score, for transaction amounts it may be an absolute difference of the amounts on the receipt and the transaction, for dates/times it may be a time delta, some may be discretized measures of confidence).
Since the data extraction from the receipt image is not always perfect (for example the incorrect string is extracted for the merchant name) we want to build a model to learn which matching features are the most successful. The ultimate goal is to match a single receipt to the correct transaction given a number of possible transactions however, given real world considerations, we want to sort the possible transactions for a given receipt in order of likelihood of being the correct transaction. So in the app when the customer takes a picture of a receipt, the app provides a list of transactions likely to match the receipt, with the one we think is correct at the top of the list. 'Success' in this context means that the correct transaction for the given receipt is at the top of the list, (note, if the correct matching is not in the data for a given receipt 'success' is not possible).

1.0.1 Target Variable

From the problem statement I've created the label by comparing the $matched_transaction_id$ with $feature_transaction_id$. The dataset found to be highly imabalenced which is shown below.

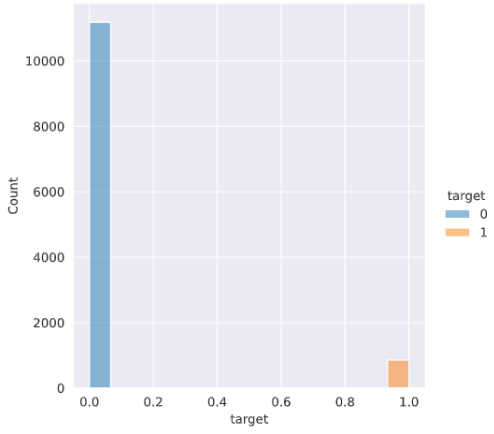

Figure 1: Class labeld distribution

1.1 Features

Dataset has 10 feature fields all numerical and and non-nan/missing values. A statistical description of the same is attached here with.

	DateMappingMatch	AmountMappingMatch	DescriptionMatch	DifferentPredictedTime	TimeMappingMatch	PredictedNameMatch	ShortNameMatch	DifferentPredictedDate	PredictedAmountMatch	PredictedTimeCloseMatch
count	12034.000000	12034.000000	12034.000000	12034.000000	12034.000000	12034.000000	12034.000000	12034.000000	12034.000000	12034.000000
mean	0.217901	0.031660	0.021522	0.986455	0.013877	0.024215	0.037893	0.753532	0.001005	0.076533
std	0.384535	0.122611	0.116995	0.115597	0.116987	0.128646	0.190945	0.430972	0.020134	0.265860
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
max	1.000000	0.900000	0.800000	1.000000	1.000000	0.800000	1.000000	1.000000	0.600000	1.000000

Figure 2: Feature Description

2 Experiments and Results

I have trained 5 different model wich are available in scitkit-learn and xgboost. The training and evaluation configuration is given bellow:

- StratefiedKFold validation with K=5
- Precision, Recall, F1, Accuracy and ACU metrics are used to to asses the model, since data is highly imabalanced it's important to note that one metric may not be enough to tell everything about the model, but since the risk for misclassifying the **mistmached** class as **matched** we have to have high recall for that.
- Manually Calibrated the class weight to handle class imabalance.
- Results of each model is presented in table 1

Model	Accuracy	Precision	Recall	F1	AUC
H2O	96.36	92.22	79.66	84.64	77.33
SVM	95.98 ± 0.31	77.68 ± 2.33	59.08 ± 32.36	68.38 ± 2.81	70.02 ± 2.87
RF	96.09 ± 0.33	79.67 ± 3.76	60.02 ± 32.52	68.88 ± 2.48	74.95 ± 3.47
XGBoost	95.83 ± 0.54	74.96 ± 6.71	63.92 ± 36.26	68.37 ± 3.28	75.52 ± 3.19
Ensemble	96.13 ± 0.27	82.78 ± 3.94	66.51 ± 35.35	68.03 ± 2.16	75.40 ± 2.47
TabNet	95.88 ± 0.44	79.32 ± 3.75	74.56 ± 30.58	66.26 ± 4.29	73.68 ± 3.16

Table 1: Experiment Results. H2O is an ensemble method and 100 models were trained which is very compute and memory intensive due to which cross validation could not be done rather I trained it for 80:20 split. Rest of the models in table are trained with 5-fold validation and mean with std metric value is being reported here.

3 Conclusion

I have trained total 6 models among all H2O Ensemble model yeilds best result. Among cross validation results suprisingly TabNet which is neural model yields best recall value and much higher than the XGBoost, RandomForest and Ensemble methods. Throughout the results we observe that recall value is has high std which is caused by the fact that data is highly unbalanced and few misclassification in minority class can make huge difference. All the models are trained with fixed set parameters (few tuned manually), automated parameter tunning may yield singficant improvement in the peroformance. Further, there were very minimal information about how these parameters are obtained which restriced us to do the any type of feature engineering, although I believe that a good feature engineering can improve the result further.

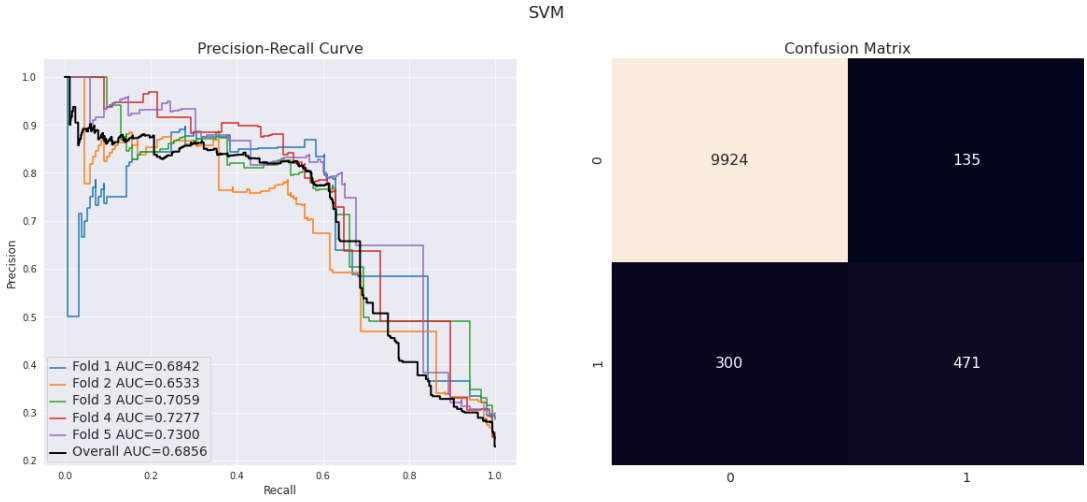

Figure 3: Support Vector Machine (SVM) Precision-Recall curve (left) and Confusion Matrix over all 5 folds combined

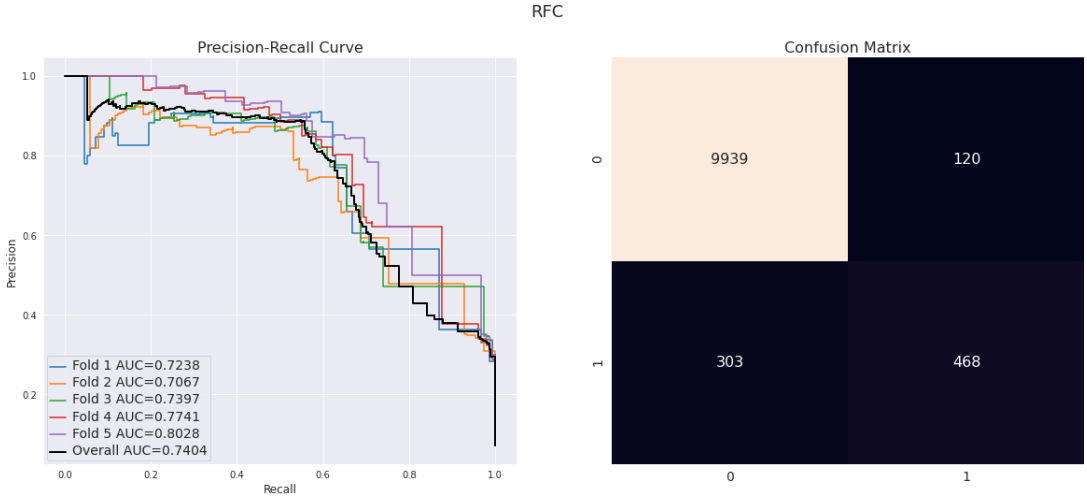

Figure 4: Random Forest Classifier (RFC) Precision-Recall curve (left) and Confusion Matrix over all 5 folds combined

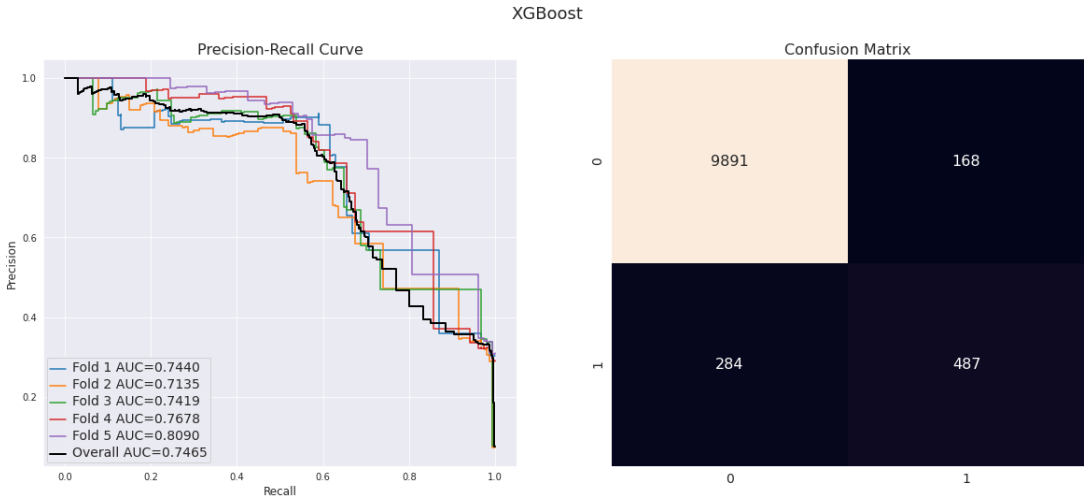

Figure 5: XGBoost Precision-Recall curve (left) and Confusion Matrix over all 5 folds combined.

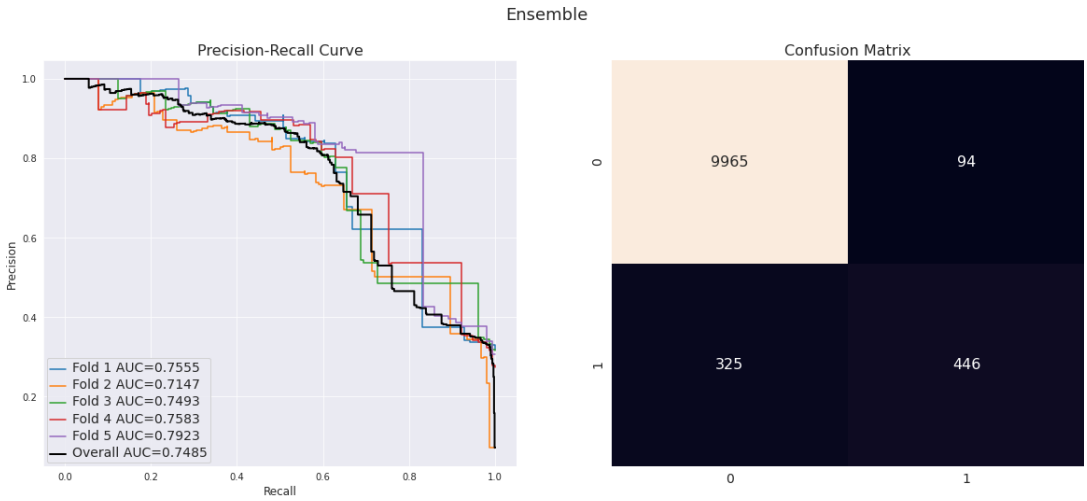

Figure 6: Ensemble Precision-Recall curve (left) and Confusion Matrix over all 5 folds combined.

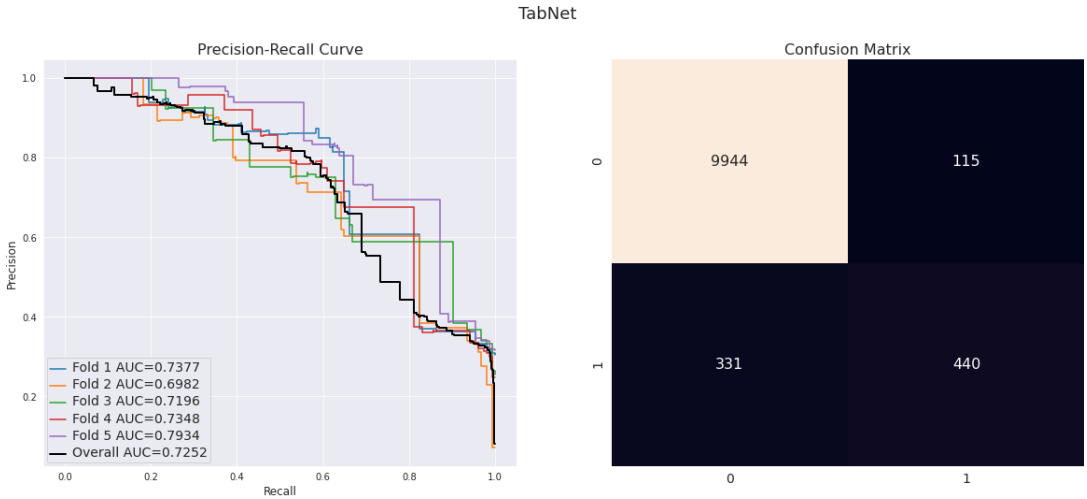

Figure 7: TabNet Precision-Recall curve (left) and Confusion Matrix over all 5 folds combined.

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

A Appendix

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.