Week 10 - Data Centric AI

Aritra Ray

Maria Alice Fagundes Vieira

- **Three (new) ways that you used to improve the data**

The three ways we used to improve the data were:

Change #1: Instead of dropping the duplicates on the customer code column and use only the last instance we will keep those duplicates since it could capture some patterns such as if a client buys product x first, it will likely buy y product next.

Change #2: Add the date column as one of the features. For that, we will calculate the time since purchase using the month we are trying to predict: June 2016. For this transformation to make sense, we will also keep the first transformation, since the time line of purchase matters now, we will keep the duplicate clients' purchases instead of only keeping the last one.

Change #3: We will scale our features using MinMaxScaler from sklearn.preprocessing before training. This could improve convergence and the stability of the model. We were between using MinMaxScaler or StandardScaler, however, since our data is not normal distributed, we will use MinMaxScaler

- **Compare performance metrics between the model from this week with your best model from Week 9 using the updated validation dataset.**

After comparing the performance of the model from this week and the best model from week 9, the model **2** performed better. This is the model that keep the duplicate customer codes and adds the data feature on it, showing how many months has passed since the last purchase. Below is a table with performance metrics:

Dataset	Type	Avg ROC AUC	Avg F1 Score
train	train	0.888038	0.110746
train	test	0.883099	0.201405
train_1	train	0.885656	0.111385
train_1	test	0.883373	0.207875
train_2	train	0.885926	0.111536
train_2	test	0.883623	0.212467
train_3	train	0.885885	0.111541
train_3	test	0.883519	0.205022

- **What are some insights you can provide based on the test and training errors of the selected model?**

The ROC AUC scores for both the training and test sets are relatively close across all datasets, indicating that the model generalizes well in terms of distinguishing between classes.

The differences between training and test ROC AUC scores are small, with the largest difference being for the original "train" dataset (0.004939). For other datasets (train_1, train_2, and train_3), the differences are even smaller, ranging from 0.002283 to 0.002366. This suggests that there is minimal overfitting in terms of ROC AUC, as the model performs similarly on both the training and test sets.

The F1 Score shows a significant difference between the training and test sets across all datasets. The F1 score is consistently higher on the test set compared to the training set. This could indicate that while the model is able to distinguish between classes (as seen with ROC AUC), it struggles with class imbalance or threshold calibration during training, leading to lower F1 scores on the training data.

Overall the model shows good generalization in terms of ROC AUC but struggles with achieving a balanced precision-recall tradeoff during training. There might be underlying issues related to class imbalance or threshold calibration that need further investigation.