AML Group-14 Project Proposal V1.0

Members: Harsh Benahalkar (hb2776) | Shan Hui (sh4477) | Quinn Booth (qab2004) | Steven Chase (sc4859) | Siddharth Vijay (sv2637)?

1 Background and context

We intend to attempt a Kaggle competition from 2018 titled "Elo Merchant Category Recommendation." This was held on behalf of a Brazilian company, Elo, which garners the market on payment, dealing with a large amount of customer and transaction data. The task is to utilize this data in order to predict a customer's loyalty score. This is important to generate a targeted consumer experience and provide promotions/deals to the right people.

2 Identification and description of the data set(s)

Dataset link: Elo Merchant Category Recommendation

The dataset consists of multiple CSVs containing customer data: historical_transactions.csv, merchants.csv, new_merchant_transactions.csv, and train/test.csv . We would need to group these by customer to make tangible predictions. The data provided in the dataset is fictitious and generated and not real world data. ~20000 rows in the historical_transactions dataset.

Historical_transactions.csv : All the transaction records of cards matching those in the train and test CSVs (at least 3 months of records per card). Columns include card_id, month_lag, installments, purchase_amount, purchase_date, state_id and 3 extensions (Category _1, 2 and 3).

Merchants.csv : Some information about merchants, including group, category, subsector IDs and other identifiers. Data on sales (avg_sales, avg_purchases), location (city_id, state_id), and activity (active_months).

New_merchant_transactions.csv : Transactions of new merchants where users have made purchases for the first time within two months. Categories include card_id (for inter-dataset linkage), city_id, installments, purchase_amound, purchase_date, city_id, etc.

train/test.csv's : Train and test sets, including card_id to match with the other CSVs, a target loyalty score and 3 extensions (feature_1, 2 and 3).

3 Proposed ML techniques

3.1 Data Pre-processing

3.1.1 Reducing noise of data: isolation forest – using binary trees to detect anomalies in our data (this method is good because we have a very large dataset), etc.

3.1.2 Sampling by methods like bootstrap, stratified sampling etc.

3.2 Feature Engineering and Visualization

3.4 Building Models

3.4.1 Implementation of different models like XGBoost, Lightgbm and NN models etc. Compare the performance and stacking models for prediction.

3.4.2 Hyperparameter tuning techniques: grid Search, random search etc.

3.5 Performance Evaluation

3.5.1 We will use RMSE as this is how submissions for the original competition were scored and we want to see how ours measures up.