Udacity Capstone Proposal

Elo Merchant Category Recommendation

Helping understand customer loyalty

Domain Background

Customer loyalty and churn prediction are widely studied areas of machine learning across all industries. Companies need to understand what attributes/features can help categorize loyal customers. It's a common understanding in business that loyal customers are the most profitable customers, therefore companies will try and do what is necessary to avoid churn for these customers' wherever possible.

Common machine learning use cases are churn prediction, customer journey analysis, recommendation systems for upsell/cross-sell opportunities, and many more.

Problem Statement

For my capstone, I will be tapping into the "Elo Merchant Category Recommendation" competition available on Kaggle to delve into a real-word problem which contains implications for a supervised modelling/regression approach to calculate customer signal.

Elo, one of the largest payment brands in Brazil, has built partnerships with merchants in order to offer promotions or discounts to cardholders. They have built machine learning models to understand the most important aspects and preferences in their customers' lifecycle, from food to shopping. But so far none of them is specifically tailored for an individual or profile.

For my capstone, I will develop algorithms to identify and serve the most relevant opportunities to individuals, by uncovering signal in customer loyalty. This input will improve customers' lives and help Elo reduce unwanted campaigns, to create the right experience for customers.

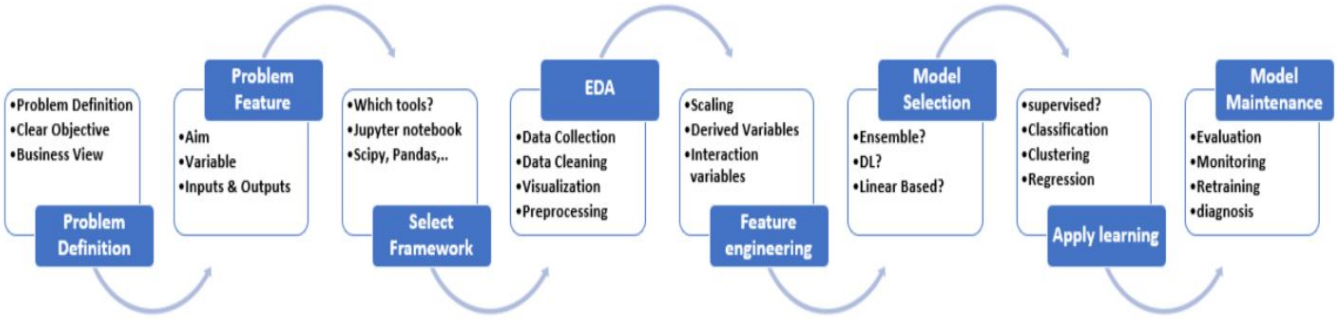
Datasets and Inputs

Provided are the available datasets for the competition:

- train.csv - the training set
- test.csv - the test set
- sample_submission.csv - a sample submission file in the correct format - contains all `card_id`s you are expected to predict for.
- historical_transactions.csv - up to 3 months' worth of historical transactions for each `card_id`
- merchants.csv - additional information about all merchants / `merchant_id`s in the dataset.
- new_merchant_transactions.csv - two months' worth of data for each `card_id` containing ALL purchases that `card_id` made at `merchant_id`s that were *not visited in the historical data*.

Solution Statement

Provided is an overview of the approach that will be used to encapsulate all aspects of the machine learning lifecycle required for this capstone:



As the problem definition and inputs have already been defined, I will go over the remaining steps in the process.

Select Framework: For this project I will spinning up a virtual environment where I'll be developing the code based in python within a jupyter notebook. This is a flexible approach which also translates well for model maintenance & production, as the model can be spun in a docker container or a flask application for a model with front-end implications. Pandas will be the underlying library used to load and manipulate the required datasets, and sklearn for all modelling needs.

EDA: I will conduct various EDA techniques such as deriving summary statistics as well as distributions across several variables to determine if any outliers exist that can be candidate for removal. I will use the seaborn package to determine if any correlation exist across different features in reference to predicting the target variable.

Feature Engineering: This will involve working with the data to derive net new features that will be used as input to the supervised model. Dummy encoding will be used to convert the categorical features to a numeric representation, and features will be generated based on the aggregates/summary statistics for the variables in the new_merchant and historical_transactions.

Model Selection: For this exercise I'll be using a XGBRegressor to employ the benefits of a tree based learning model, in combination with GridSearch to optimize the values given for the model's hyperparameters.

Baseline Model

For the purposes of this project I'll be using the XGBRegressor model to predict the outcome for the target variable.

The implementation of XGBoost offers several advanced features for model tuning, computing environments and algorithm enhancement. It is capable of performing the three main forms of

gradient boosting (Gradient Boosting (GB), Stochastic GB and Regularized GB) and it is robust enough to support fine tuning and addition of regularization parameters.

As refernced in https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

Advantages for using XGBoost:

1. **Regularization: To help reduce overfitting**
2. **Parallel Processing**
3. **High Flexibility**
4. **Handling Missing Values**
5. **Tree Pruning**
6. **Built-in Cross-Validation**
7. **Continue on Existing Model**

Evaluation Metrics

Submissions are scored on the root mean squared error. RMSE(Root Mean Squared Error) is

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

defined as: where y^ is the predicted loyalty score for each card_id, and y is the actual loyalty score assigned to a card_id.

Project Design

Upon running the baseline model, feature importance plots will be utilized to trim out the irrelevant features so that training time can be reduced. In addition, additional feature engineering tasks will be completed to derive net new features that could deem useful to the model outcome. The research and resources available to tune an XGBoost model is extensive. I will be utilizing this sources to improve the baseline results in order to create a model ready for production.

The initial kaggle score was and RMSE of 3.908

Efforts will be made to improve this baseline result and reduce the overall RMSE.