Machine Learning Engineer Nanodegree

Capstone Proposal

Pingili Lohith Oct 27, 2018

Proposal

Domain Background:

The Pareto principle (also known as the 80/20 rule) states that, for many events, roughly 80% of the effects come from 20% of the causes. The 80/20 rule has proven true for many businesses– only a small percentage of customers produce most of the revenue. It's a challenge in such businesses to invest in promotional strategies that target the small fraction of customers who drive the revenue.

Related research:

https://hbr.org/2017/02/ai-is-going-to-change-the-8020-rule

https://www.swrve.com/images/uploads/whitepapers/swrve-monetization-report-2016.pdf

Problem Statement:

Predict revenue per customer by analyzing a Google Merchandise Store (also known as GStore, where Google swag is sold) customer dataset. Source.

Datasets and Inputs:

The Kaggle competition has the train and test datasets. Train dataset has 903653 rows with 12 columns and the test dataset has 804683 rows with 12 columns. There are multiple sub-columns for 4 of the attributes. 'Transaction revenue' sub-column from 'Totals' attribute in the test dataset is held back by Kaggle for evaluating the submissions. The description of the attributes:

- fullVisitorId- A unique identifier for each user of the Google Merchandise Store.
- channelGrouping - The channel via which the user came to the Store.
- date - The date on which the user visited the Store.
- device - The specifications for the device used to access the Store.

- geoNetwork - This section contains information about the geography of the user.
- sessionId - A unique identifier for this visit to the store.
- socialEngagementType - Engagement type, either "Socially Engaged" or "Not Socially Engaged".
- totals - This section contains aggregate values across the session.
- trafficSource - This section contains information about the Traffic Source from which the session originated.
- visitId - An identifier for this session. This is part of the value usually stored as the _utmb cookie. This is only unique to the user. For a completely unique ID, you should use a combination of fullVisitorId and visitId.
- visitNumber - The session number for this user. If this is the first session, then this is set to 1.
- visitStartTime - The timestamp (expressed as POSIX time).

Solution Statement:

As the problem statement requires the prediction of revenue per customer, the solution is a regression model. Various regression algorithms like Linear Regression, SVM, Decision Tress, Adaboost, XGBoost should be applied to find the best performing algorithm.

Benchmark Model:

A simple linear regression yielded a RMSE of 1.98 and a XGBoost with default parameters yielded a RMSE of 1.75. (RMSE was calculated after taking natural logarithm of (y predicted +1), 1 is added as most of the revenue is 0).

Evaluation Metrics:

Kaggle evaluates a model in this competition based on RMSE score calculated for predictions based on the test data. This would be helpful in comparing the performance of my model with the leaderboard for the competition. For model selection RMSE or R2 score will be used on a validation set to gauge the performance as compared to the benchmark model.

Project Design:

Programming language: Python 3.6

Workflow:

- **Import** datasets and flatten the json fields into a csv file for easier readability. Drop any redundant columns.

- **Visualization** of data to understand if any correlation exists, predictability of various attributes etc.
- **Splitting of data into train and test**: 'Transaction revenue' sub-column from 'Totals' attribute in the test dataset is held back by Kaggle for evaluating the submissions. So, train data is split into development and validation sets to evaluate the performance of models before submitting the best model to Kaggle for evaluation based on the held back test dataset.
- **Feature engineering** if applicable: PCA if the evaluation metrics are in favor.
- **Model selection**: XGBoost performed best in the initial testing. I intend to use other alogorithms like lgboost, catboost, SVM, Artificial Neural Networks etc to find the best performing model.
- **Parameter tuning** by using GridSearchCV to fine tune the model.
- **Testing**: Testing the model with the Kaggle RMSE score.