Google Analytics Customer Revenue Prediction

Predict how much GStore customers will spend

Featured · 3 days to go · regression, tabular data

Project Report

Xiaohe Yu, Xie Qin, Yao Li

Abstract. This report is about building a Google Analytics Customer Revenue Prediction for Kaggle competition. The Google Analytics Customer Revenue Prediction is to analyze Google Merchandise Store customers' dataset to predict revenue per customer. The prediction framework is composed by four mainly parts: data preprocessing, feature engineering, modeling and evaluation. Finally, the results show a good performance of the prediction method.

Keywords: Kaggle, Revenue Prediction.

1. Introduction

The 80/20 rule can be proved true by many businesses that 20% of customers produce 80% of the revenue. Because of that, marketing team need to make appropriate investments in promotional strategies.

In this Kaggle competition, we're challenged to analyze the Google Merchandise Store (also known as GStore, where Google swag is sold) customer dataset to predict revenue per customer. Hopefully, the outcome will be more actionable operational changes and a better use of marketing budgets for those companies who choose to use data analysis on top of GA data.

In this project, we are requested to use data set provided by Google store to predict customs' transactions in the future time period of December 1st 2018 through January 31st 2019 .the total natural log revenue per customer and use Root Mean Squared Error to judge the prediction.

The reminder of the report is organized as follows. Section 2 describes our

approach. In Section 3, experimental results are presented. Finally, the report is concluded in Section 4.

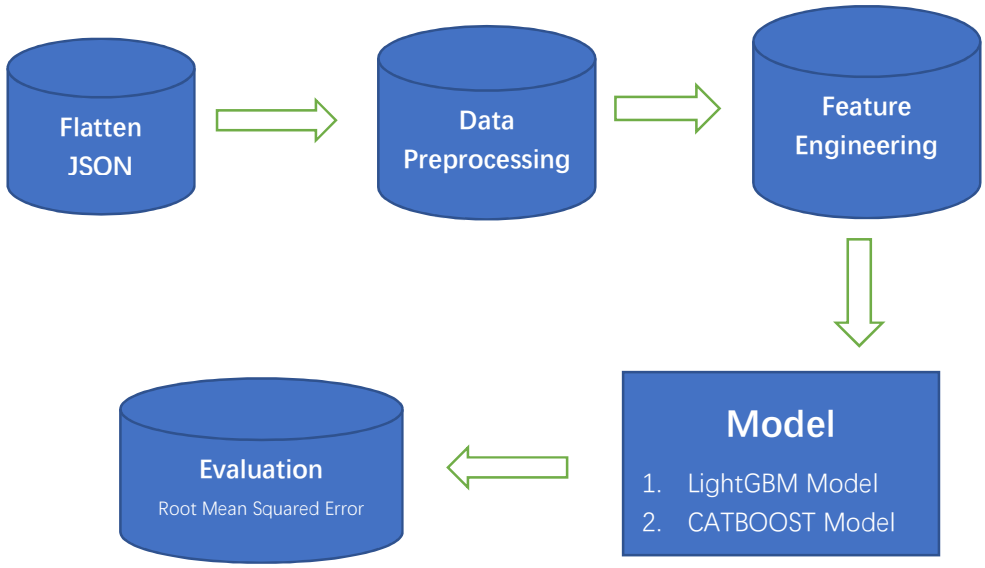

Figure 1

2. Our Approach

As shown in Figure 1, This is the architecture of our Google Analytics Customer Revenue Prediction system. It shows that our prediction system mainly consists of five parts, Flatten Json, Data Preprocessing, Feature Engineer, Model Selection and Evaluation. The details of each parts will be showed in the following sections.

Data Set

According to the Kaggle competition, we decide to use the data set provided by Google Store which named train_v2.csv and test_v2.csv. train_v2.csv contains user transactions from August 1st 2016 to April 30th 2018 and test_v2.csv contains user transactions from May 1st 2018 to October 15th 2018. The data fields of data set is shown in the Chart1.

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.
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).
hits	This row and nested fields are populated for any and all types of hits. Provides a record of all page visits.
customDimensions	This section contains any user-level or session-level custom dimensions that are set for a session. This is a repeated field and has an entry for each dimension that is set.
totals	This set of columns mostly includes high-level aggregate data.

Chart 1 Data field
(from https://www.kaggle.com/c/ga-customer-revenue-prediction/data)

Data Preprocessing

Before we train the model, we need to preprocess the dataset. The preprocessing steps are shown in the below.

1. Flatten JSON. First, we flatten JSON subfield into separate columns in order to convert raw data into the format we can use and normalize the JSON columns using json_normalize function.
2. Clearing data. In row dataset, there are 14 columns of features. So, we will check the shape of the train and test databases and delete columns which don't exist in test dataset. We also drop those columns that have one unique value and too many unique values. Next we remove rare values in each column and categorize such rare values as "others". And some model irrelevant fields such as timestamp information are excluded from our training set.

Feature engineering

In order to have a better prediction, we can't use all the data in the training dataset, we need extract the features from data fields. We select some important fields and combinate them as new composite fields and convert non-numerical fields into numerical based on Labelencoder.

1. Feature combination. According to the Google store's dataset, there are 14 data fields. So we need first to figure out which feature will have more influence on the prediction. We use the method provided by Ashish Patel in Kaggle Kernels to display the feature importance.

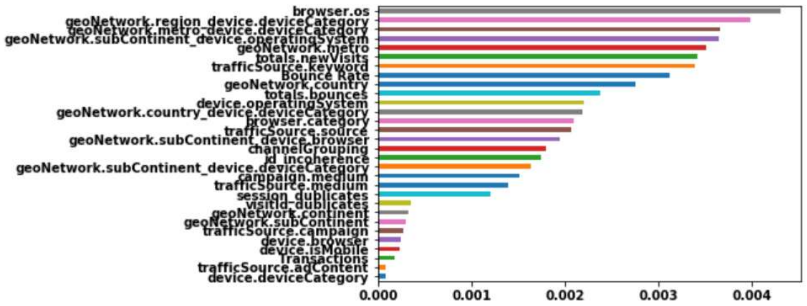

Figure 2 feature importance

(from Ashish Patel https://www.kaggle.com/ashishpatel26/1-23-pb-first-try-to-think)

So we merge four subcolumns of device, geoNetwork, trafficSource.ad, trafficSource into each one new column.

2. Labelencoder. we convert non-numerical fields into numerical based on Labelencoder. In particular, at first, we also want to use on hot encode techinic. But we found that when the number of categories is large, the feature space becomes very large. In this case, we need use PCA to reduce the dimension. Considering of this, we decide just use labelencoder technic in our base revision, and in improved revision we will use one hot encoding+PCA.

Model Selection.

In this project, we have chosen the LightGBM and Catboost as our classifier. Each method is trained and tested separately first and then stacked together to improve the prediction accuracy. In the validation stage, we use K-Fold to validate the classifier to avoid overfitting.

Comparing evaluations of lightGBM, catboost and mixed model, we find that the prediction accuracy of the classifier of mixed model is little better than the classifier just using one model. So we adjust the model parameters and each

model's weight of mixed model. Finally, we find that 0.8 of Catboost and 0.2 of LightGBM can give the best prediction accuracy.

As for iterative convergence, we use K-fold to validate our model. For LightGBM, we use 4-Fold validation with 1200 iterations and set 30 rounds of early stopping. For CatBoost, we use 4-Fold validation with 500 iterations and set 30 rounds of early stopping.

Evaluation

As for model validation, the Root Mean Square Error(RMSE) is selected as the model accuracy measurement to score the model we submit. For each fullVisitorId in the test set, we should predict the natural log of Google Store customers' total revenue in PredictedLogRevenue. The submission file should contain a header and have the official format. RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

where y hat is the natural log of the predicted revenue for a customer and y is the natural log of the actual summed revenue value plus one.

3. Experiment Results and Analysis

Because of the large dataset provided by Google Store, CatBoost model takes 50mins to run and LightGBM takes 30mins to run. After tens of testing, we found that the Catboost performed better than the LightGBM between each model. Two of representative results are shown in the below.

Figure3 Lgboost single model

Figure4 Catboost single model

Then we stack LightGBM and CATBOOST together to reduce variances. After training two models we get the folds' RMSE. As for LightGBM, first fold RMSE is

6.64483e-07, second fold RMSE is 6.61787e-07 , third fold RMSE is 6.49045e-07 and forth fold RMSE is 6.47187e-07. As for CATBOOST, first fold bestTest is 0.0004915930885, second fold bestTest is 2.812065844e-08 , third fold bestTest is 2.756866884e-08 and forth fold bestTest is 4.903307428e-05. And stacking two models together with proper weight, the accuracy is further improved. The best results we got so far is 0.083.

Figure5 0.8*CAT+0.2LGB

We also find one feature named TotalRevenue will have a big influence on the prediction accuracy, so we test it and results are in the below.

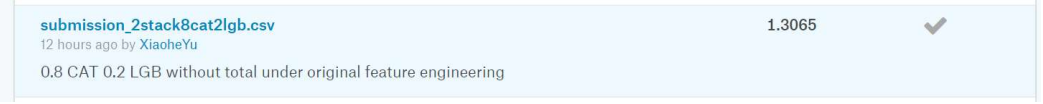

Figure6 without TotalRevenue

Figure7 with TotalRevenue

4 Conclusions

In this project report, we show our work in Kaggle competition of Google Analytics Customer Revenue Prediction. We build a revenue prediction system. It mainly performs four steps to predict and our strategy works well. We will extract more useful features and try to figure out further strategies to improve our model accuracy. We will focus on the learning to regression approaches in the future.