MSAI 349 - Project Report

Google Analytics Customer Revenue Prediction

Mayank Malik and Amit Adate

1 Introduction and Objective

This project is our contribution to a live kaggle competition by Google. According to Google, The 80/20 rule has proven true for many businesses only a small percentage of customers produce most of the revenue. As such, Googles marketing teams are challenged to make appropriate investments in promotional strategies.

Therefore, Google gave the task to predict the total revenue generated per customer based on the customer dataset of a Google Analytics Merchandise Store. So that the outcome will lead to more actionable operational changes and finally result in a better use of marketing budgets for them.

In short, we have tackled a regression task

2 The Data

We are provided with the training dataset of 1.7 million observations with 14 columns. However, a few columns are in JSON format and they required considerable preprocessing to convert into normal flattened format. Each JSON column had many features in itself . We wrote a python script to flattening these JSON into columns. The total number of features after converting these json columns was 60 in number.

Since, this is an online store, each observation actually is a session when a user comes online at the store to purchase something. However, a user mkay or may not cause any transaction, he could just visit the store for browsing purposes. When a user does not purchase anything , the **column totals.transactionRevenue** (Dependant feature or target feature that we need to predict) is 0 , else it generates some revenue (non zero). Also, each observation has other attributes / columns such as : device.browser , device.language, device.operatingSystem city, country, continent, date, visitStartTime, latitude , longitude etc.

2.1 Partitioning of data

The kaggle competition creators provided 1.7 million observations in the training set and 0.4 million observations in the testing set. Additionally, we have divided the training set into training and validation sets to perform 5 fold cross validation. Hence, we finally have 3 datasets: training, validation and testing

2.2 Libraries and tools

For most of the work, we used sklearn, Keras library and weka tool. Most of the times, we used GridSearchCV for Hyperparameter tuning in our models.

2.3 Features that have missing values

For feature, missing more than 50% of data, we removed the feature. Other, we imputed the mean for continuous and treated missing values like a new category for categorical features.

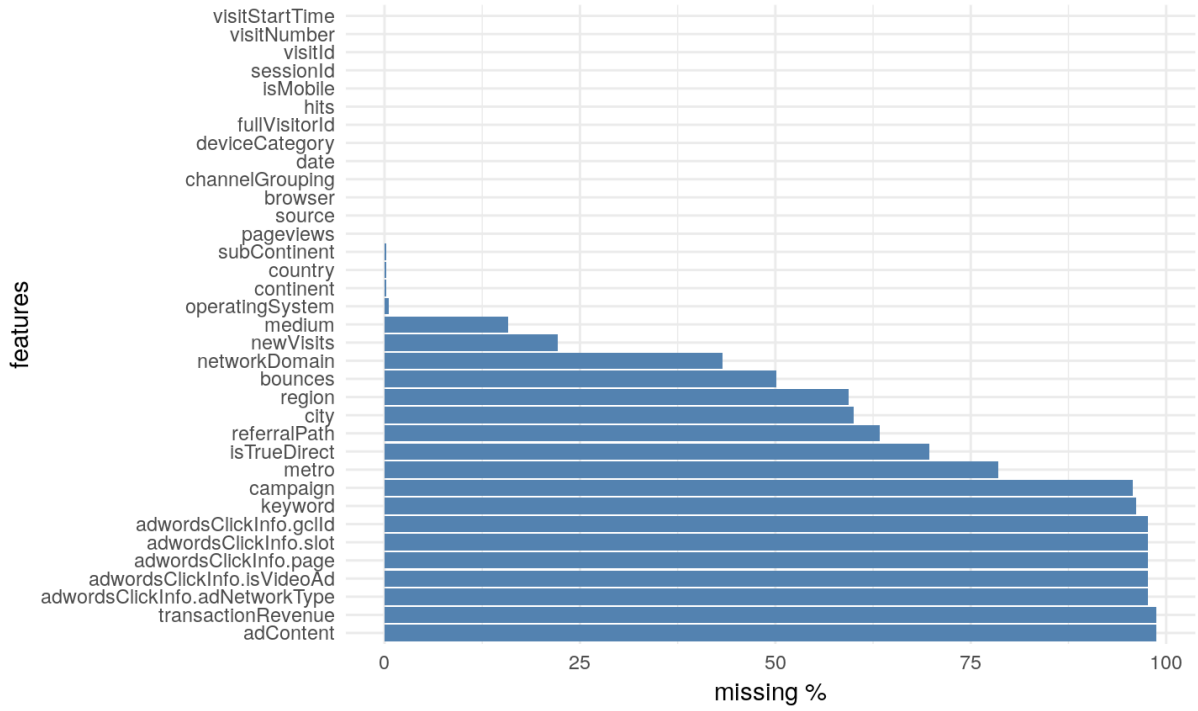

Figure 1: Missing Values Percentage among Features

3 Results and Analysis

Since we are predicting the natural log of sum of all transactions revenue of the user, we summed up the transaction revenue at user level and took natural log. Now, based on the analysis we have done, we first started with some basic regression techniques for the baseline results.

Baselines	
Techniques Used	**RMSE**
Linear Regression	8.3
Polynomial Regression (degree 3)	4.49
Decision Tree	5.077
Random Forest	3.1
XGBoost	2.9

In the project status report last month, we reported MSE and not RMSE. However, we are reporting all our results in RMSE in this final report and analysis.

3.1 Key Challenge Faced

Although we were happy to see the convincing results from our regression techniques (as discussed above) and also we reached among the top 20 percentile in the contest until then, we realized that we had serious issue in terms of data imbalance. In our dataset, The dependant variable (**totals.transactionRevenue**) is highly imbalanced. The transaction revenue is present only in 1.3 percent of all the sessions(or observations). For the rest, it is 0. As every observation in the dataset is an online session for the purchase in the online store, it means that only 1.3 percent of all the sessions resulted in a purchase of a product and in the rest sessions, customers didn't purchase anything.

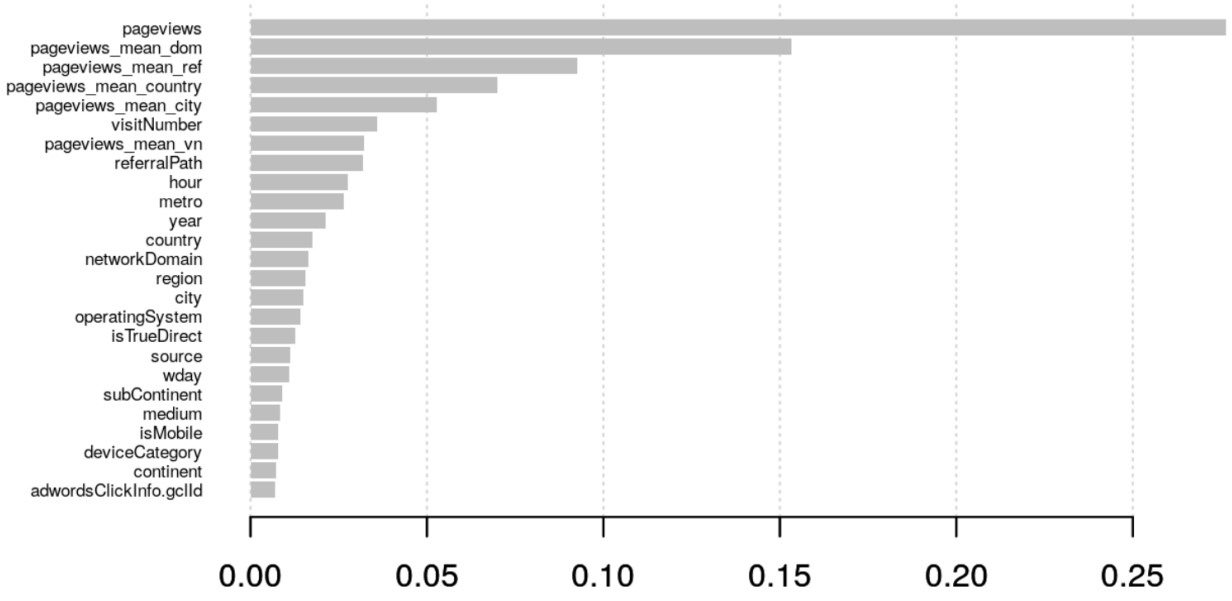

Figure 2: Feature Importance by XGBoost

As imbalanced data substantially compromises the learning process, we had a detailed discussion with Professor Downey on 21st November. We finalized two strategies to tackle the issue:

3.1.1 Resampling the data

As we had a lot of data - 1.7 million observations, it made sense to go for undersampling technique. So, we started removing records from the data where we had zero revenue and simultaneously started evaluating . Undersampling technique definitely improved the results. The best RMSE scores we got were:

3.1.2 Classification + Regression

This strategy helped us get into the top 5 percentile in the contest. We believed the main issue we have in this challenge is not to predict transaction revenues but more to get those zeros (zero revenue) right when there was no transaction present. Since less than 1.3 % of the sessions have a

After Resampling	
Techniques Used	**RMSE**
Linear Regression	1.55
Decision Tree Regressor	2.13
Adaboost Regressor	2.35
Random Forest Regressor	1.65
Keras Sequential Model	1.52
XGBoost	1.48

non-zero revenue, we believed it was important to find those 1.3 % acc. The idea of this analysis is to classify non-zero transactions first and do the regression analysis (predict revenue) only on those non-zero transactions classified by our classifier. And , ofcourse, our regressor is trained on non zero revenue data only.

Classifiers used to predict transaction revenue present (label - 1) and transaction revenue not present (zero transaction revenue) - (label - 0). Moreover, we used undersampling techniques to improve our classification problem :

Classification Imporovement		
Classifier Used	**Accuracy**	**AUC**
Adaboost	99.42 %	73%
Logistic Regression (with regularization)	99.12 %	62%
Support Vector Machine (no Kernel used)	99.18 %	64%
Support Vector Machine (with RBF Kernel)	99.29 %	69%

The very high accuracy is because of the imbalanced dataset. Therefore, we used AUC to compare the results. **Also, we used RBF kernel with SVM to improve the SVM performance and regularization with logistic regression for better generalization. With logistic regression, we optimised parameter C, which controls the inverse of the regularization strength, with GridSearchCV.**

After classifying the data into transaction revenue present (label - 1) and transaction revenue not present (zero transaction revenue) - (label - 0), we predicted the transaction revenue only on classified data with label 1 (or transaction revenue present) using a Regressor, which was trained on the data that contained non zero transaction revenue.

Experiments on Classified Label - 1 Data	
Techniques Used	**RMSE**
Random Forest	1.4
XGBoost	1.14

4 Conclusion

Analysing our results, we conclude that although we considered this project to be a simple regression task, we encountered a big challenge of the data being imbalanced. First, we tried traditional technique of resampling(undersampling) with regression to improve the results. But, the best results we got when we first classified non zero transaction observations as they were only 1.3 % of the total dataset and then predicted (regression) the revenue for them. Our best ranking on Kaggle was [143 / 3350] top 5 percentile.