Google Analytics Customer Revenue Prediction

Ashish Kumar
axk171531@utdallas.edu

Varun Mehrotra
vxm170830@utdallas.edu

Abstract – Marketing team currently face difficulty in making correct strategy for generating more revenue per customer. The 80/20 rule has proven true for many businesses i.e. only a small number of customers generate most of the revenue. So, marketing teams can make appropriate advertisement and promotional strategies for attracting right segment of user.

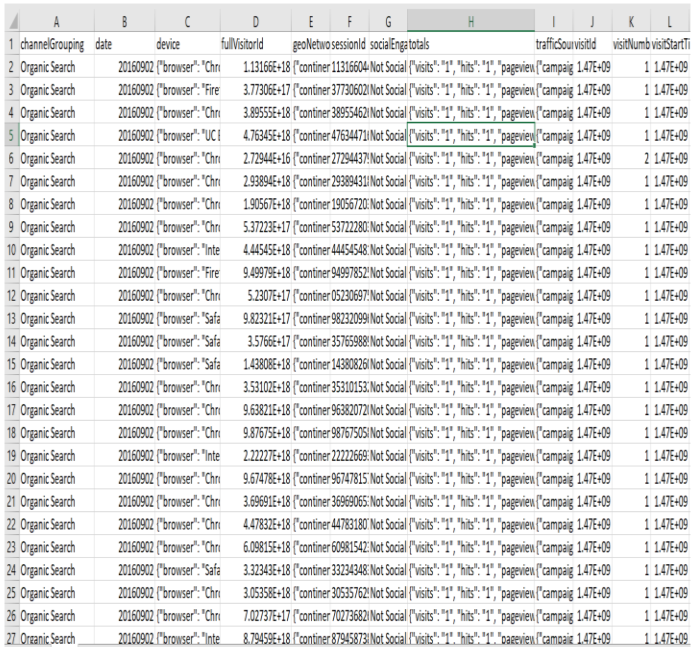
In this project we analyze a Google Merchandise Store (also known as GStore, where Google swag is sold) customer dataset to predict revenue per customer using different machine learning techniques. We are predicting natural log of the sum of all transactions per user. Hopefully, the outcome will be more actionable operational changes and a better use of marketing budgets for those companies who choose to use data analysis on top of Google Analytics data.

I. INTRODUCTION

The drop in the cost of Internet of Things (IOT) devices like microcontroller units and sensors is fueling the rise in available data. The recent advances in machine learning, data science and computing power helps us in getting valuable insights from these vast amounts of data. In this project we are analysing the GStore data to gain useful insights the revenue per customer which will help marketing team in strategizing on which customer segment to focus like customizing the user experience with relevant offers that are tailored to the right audience.

In marketing world, there is a popular principle called Pareto Principle, which is also known as 80/20 rule. It basically says that there's an 80-to-20 relationship between effects and their causes. In terms of business, we can say that about 80% of the company sales comes from 20% of the customer. So, an analyst can use this principle to come up with smart and effective business strategies for the company, by analysing the data they can discover the best customers, locate customer geographically, tailor best experience for premium customers, manage operational costs.

In this project, as we have to predict revenue per customer, so we applied various regression machines learning techniques such as Random Forests, Extreme Gradient Boosting. We also perform exploratory analysis of data and perform standardization and Principal Component Analysis (PCA) etc. Random Forests is an ensemble learning technique that is based on decision trees and we create new datasets by varying the instances from original dataset (cross validation). PCA is a mathematical tool from applied linear algebra used for extracting relevant information from confusing datasets by reducing complex data to lower dimensions.



	A	B	C	D	E	F	G	H	I	J	K	L
1	channelGrouping	date	device	fullVisitorId	geoNetwo	sessionId	socialEnga	totals	trafficSour	visitId	visitNumb	visitStartTi
2	Organic Search	20160902	{"browser": "Chr	1.13166E+18	{"continer	11316604	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
3	Organic Search	20160902	{"browser": "Fire	3.77306E+17	{"continer	37730602	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
4	Organic Search	20160902	{"browser": "Chr	3.89555E+18	{"continer	38955462	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
5	Organic Search	20160902	{"browser": "UC	4.76345E+18	{"continer	47634471	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
6	Organic Search	20160902	{"browser": "Chr	2.72944E+16	{"continer	27294437	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	2	1.47E+09
7	Organic Search	20160902	{"browser": "Chr	2.93894E+18	{"continer	29389431	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
8	Organic Search	20160902	{"browser": "Chr	1.90567E+18	{"continer	19056720	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
9	Organic Search	20160902	{"browser": "Chr	5.37223E+17	{"continer	53722280	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
10	Organic Search	20160902	{"browser": "Inte	4.44545E+18	{"continer	44454548	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
11	Organic Search	20160902	{"browser": "Fire	9.49979E+18	{"continer	94997852	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
12	Organic Search	20160902	{"browser": "Chr	5.2307E+17	{"continer	05230697	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
13	Organic Search	20160902	{"browser": "Safa	9.82321E+17	{"continer	98232099	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
14	Organic Search	20160902	{"browser": "Safa	3.5766E+17	{"continer	35765988	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
15	Organic Search	20160902	{"browser": "Safa	1.43808E+18	{"continer	14380826	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
16	Organic Search	20160902	{"browser": "Chr	3.53102E+18	{"continer	35310153	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
17	Organic Search	20160902	{"browser": "Chr	9.63821E+18	{"continer	96382072	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
18	Organic Search	20160902	{"browser": "Chr	9.87675E+18	{"continer	98767505	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
19	Organic Search	20160902	{"browser": "Inte	2.22227E+18	{"continer	22222669	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
20	Organic Search	20160902	{"browser": "Chr	9.67478E+18	{"continer	96747815	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
21	Organic Search	20160902	{"browser": "Chr	3.69691E+18	{"continer	36969065	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
22	Organic Search	20160902	{"browser": "Chr	4.47832E+18	{"continer	44783180	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
23	Organic Search	20160902	{"browser": "Chr	6.09815E+18	{"continer	60981542	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
24	Organic Search	20160902	{"browser": "Safa	3.32343E+18	{"continer	33234348	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
25	Organic Search	20160902	{"browser": "Chr	3.05358E+18	{"continer	30535762	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
26	Organic Search	20160902	{"browser": "Chr	7.02737E+17	{"continer	70273682	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09
27	Organic Search	20160902	{"browser": "Inte	8.79459E+18	{"continer	87945873	Not Social	{"visits": "1", "hits": "1", "pageview	{"campaig	1.47E+09	1	1.47E+09

Fig 1: Snapshot of dataset

There were 14 columns in the original dataset with some column in JSON format and many columns with zero variance and null values. We performed pre-processing and remove unwanted columns. The dataset description and pre-processing techniques applied are discussed in the following sections.

II. DATASET DESCRIPTION

The dataset chosen for the project is taken from - https://www.kaggle.com/c/ga-customer-revenue-prediction/data

The dataset contains transaction done at Google store merchandise in year Aug 2016- Jul 2017. The dataset has 903653 rows. The format of the data is CSV with JSON embedded columns. It has the following 14 attributes:

- *fullVisitorId*- it is a unique identifier for each customer who visits the Google Store.
- *channelGrouping* – it is the channel used by the user to visit the store.
- *date* – date on which user visited the store.
- *device* – used device specification that is used to access the store.
- *geoNetwork* – user geographical information is contained in this section.
- *socialEngagementType* – this describes whether user is "Socially Engaged" or "Not Socially Engaged".
- *totals* - this contains total aggregate across the sections, including transaction revenue.
- *trafficSource* - information from where the session was generated, i.e, traffic source
- *visitId* – it is an identifier for this session. Its value is usually stored as the _utmb cookie. It is unique to the user. For a completely unique ID, a combination of fullVisitorId and visitId can be used.
- *visitNumber* – it provides information about how many times user visited, i.e., the session number for this user. It is set to 1 for the first session.
- *visitStartTime* - The timestamp (expressed as POSIX time).
- *hits* – this section along with nested fields are populated for any and all types of hits. It provides a record of all page visits.
- *customDimensions* – It contains any user-level or session-level custom dimensions that are set for a session. It has an entry for each dimension that is set and is repeated.

III. PREPROCESSING TECHNIQUES

The downloaded dataset from Kaggle contains 14 columns with some columns is in JSON format. Those columns are following: device, geoNetwork, totals, trafficSource and adwordsClickInfo.

So, we first parsed the json columns using json parser and expanded all the columns in multiple json format. The sub columns after flattening json columns are described below:

Device: browser, browserVersion, browserSize, operatingSystem, operatingSystemVersion, isMobile, mobileDeviceBranding, mobileDeviceModel, mobileInputSelector, mobileDeviceInfo, mobileDeviceMarketingName, flashVersion, language, screenColors, screenResolutiondemo, deviceCategory.

geoNetwork: continent, subContinent, country, region, metro, city, cityId, networkDomain, latitude, longitude, networkLocation.
totals: visits, hits, pageviews.
trafficSource: campaign, source, medium, keyword.
adwordsClickInfo: criteriaParameters, isTrueDirect

After flattening the json columns, our new dataset contains **56 columns**. Now we have done exploratory analysis the new dataset.

TABLE 1
Dataset Statistic

Number of unique customers visited the store	714167
Number of instances with revenue greater than zero	11515

We plotted different plots as shown in figure 2 to figure 6.

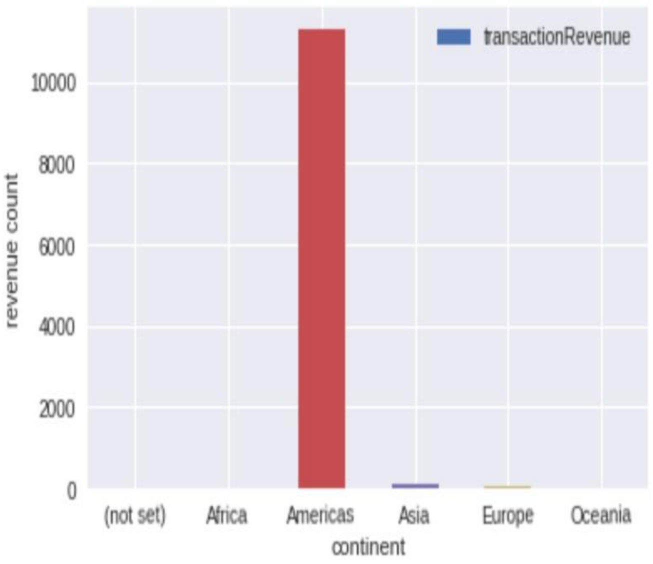


Fig 2: revenue count greater than 0 vs. continents

In figure 2 we first plotted revenue count of all transaction greater than 0 vs continents from which user accessed it. As seen from the plot the maximum number of transactions are originating from America in the given dataset.
Similarly, in Figure 3 we have plotted revenue count vs operating system from which transaction originate and we can see from Macintosh OS most of the transaction has started.

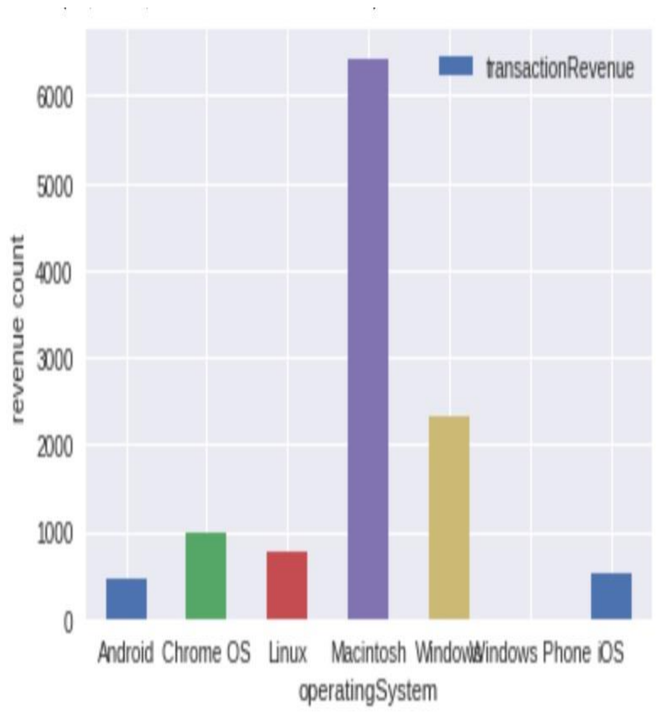


Fig 3: revenue count greater than 0 vs. operating system

In figure 4, we have plotted total revenue vs month, and we can infer from the plot that maximum revenue is generated in August, December and April may be because of holiday season (Summer, winter) and back to college sale.

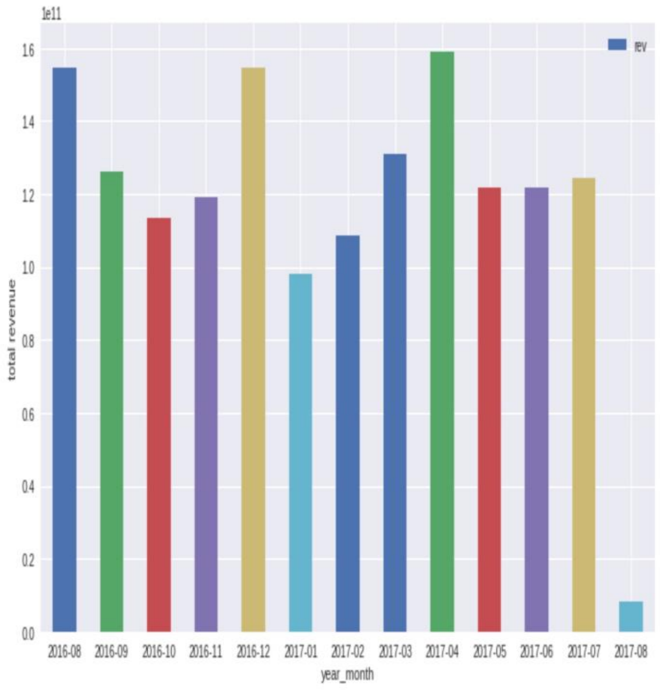


Fig 4: total revenue in billion vs month (Aug-2016 to Aug-2017)

In figure 5, we plotted total hits vs month and we can infer that during the end of year number of hits on the GStore has increased.

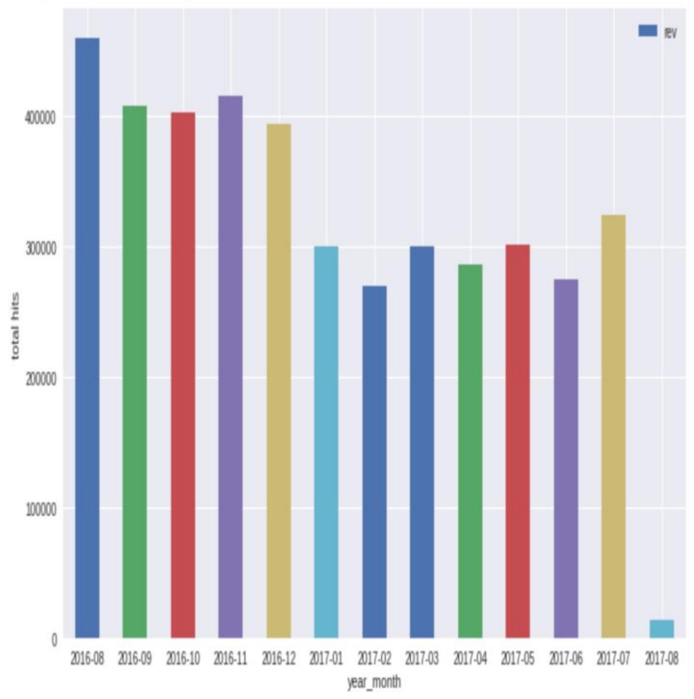


Fig 5: total hits vs month (Aug-2016 to Aug-2017)

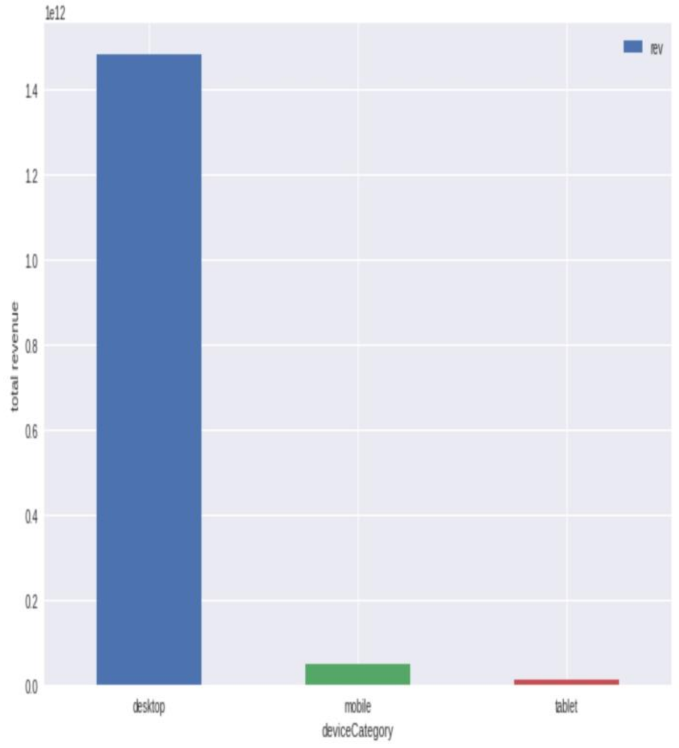


Fig 6: total revenue in billion vs device

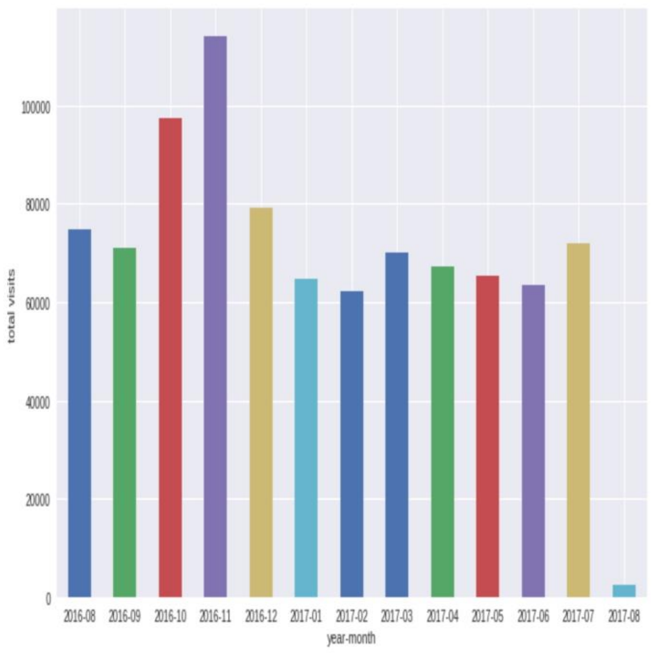


Fig 7: total visits vs year-month

The initial dataset consists of 56 columns, out of which we removed columns with zero variance and columns with more than 97% null values. This leads to the dataframe with 36 columns (903653, 36). Further, as part of data engineering we added few more features to the dataset, i.e., we generated year, month and dayofweek columns from date and performed data exploratory analysis. Further on, we used dataframe fillna method to replace all null or NaN with zero.

Initials all the columns were object type, so we changed some columns like transactionRevenue, pageviews, hits, bounces, newVisits to float type. Inorder to convert categorical data such as object or boolean data type, we used factorizer to convert them into integer data type.

Categorical features are transformed into numerical using Label Encoder and continuous numerical features are scaled using StandardScaler. The StandardScaler standardizes features by scaling to unit variance and/or removing the mean using column summary statistics on the samples in the training set.

After this we performed Primary Component Analysis (PCA) which is used for dimensionality reduction. It is a method to find a rotation such that the first coordinate has largest variance possible. PCA can be computed by eigen value decomposition or a data covariance matrix or SVD of a data matrix. After performing PCA we take 20 dimension which describes 97.547587 % variance in the data.

IV. PROPOSED SOLUTION

This project aims at predicting the transaction revenue at the google store using the set of features. Since the task is related to predicting the revenue which can be any real number, each feature in the feature vector has its own characteristic so a regression model is used to learn the important characteristics in the feature vector. We have used a set of 3 machine learning based regressors for prediction. The three methods are Linear Regressor, Light GBM and eXtreme Gradient Boosting.

Linear Regressor is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. These models are often fitted using the least squares approach, but they may also be fitted in other ways, like by minimizing the "lack of fit" in some other norm or by minimizing a penalized version of the least squares cost function.

LightGBM is a relatively new algorithm. It is a gradient boosting framework that applies decision tree based learning. It is quite fast, distributed, high-performing. It is used for many machine learning task. It grows tree vertically as compare to other boosting algorithm which grows tree horizontally. In lightGBM, tree grows leaf-wise while in other gradient boosting algorithms tree grows either level wise or depth wise. Advantages of Light GBM is faster training speed and higher efficiency, lower memory usage, better accuracy and compatibility with large datasets.

XGBoost (eXtreme Gradient Boosting) is one of the implementation of Gradient Boosting concept. Gradient Boosting is a technique used for regression and classification problems. This produces a prediction model in the form of an ensemble of weak prediction models. The weak models are typically decision trees. This model builds the model in a stage wise manner and generalizes them by allowing optimization of an arbitrary differentiable loss function. XGBoost uses a more regularized model formalization to control over-fitting. So, it gives better performance but it takes more training time than light GBM.

V. RESULTS

All the models are evaluated with different types of error metrics.

The different types of regression metrics are:

RMSE (Root Mean Square Error) - It represents the sample standard deviation of the differences between predicted values and observed values (called residuals). Mathematically, it is calculated using this formula:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - yi^{\wedge})^2}{N}}$$

MAE (Mean Absolute Error) – It is the average of the absolute difference between the predicted values and observed value. It is a linear score. Its mathematical formula is :

$$MAE = \sum_{i=0}^{N-1} |y_i - y_i{}^{\wedge}|$$

Variance Score (R Squared (R^2)) – It explains how the independent selected variables explain the variability in dependent variables. It is often used for explanatory purpose. Mathematically it is given as:

$$R^2 = 1 - \frac{\sum_{i=0}^{N-1}(y_i - yi^{\wedge})^2}{\sum_{i=0}^{N-1}(y_i - \bar{y})^2}$$

Where N is test data set size y_i is actual magnitude, $yi^{\wedge}$ is predicted magnitude and $\bar{y}$ is average magnitude.

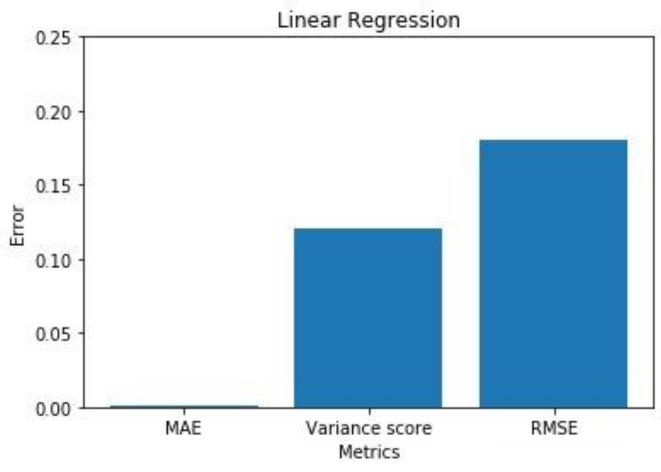


Fig 8: Linear Regression vs error metrics

We have experimented there models- linear regression, light gradient boosting and XGB, with different parameters such as k folds, max iterations, max depth, learning rate and max no of trees and regularization parameter using Grid search. Our analysis shows that XGB and Light GBM performs equally well with RMSE 0.15.

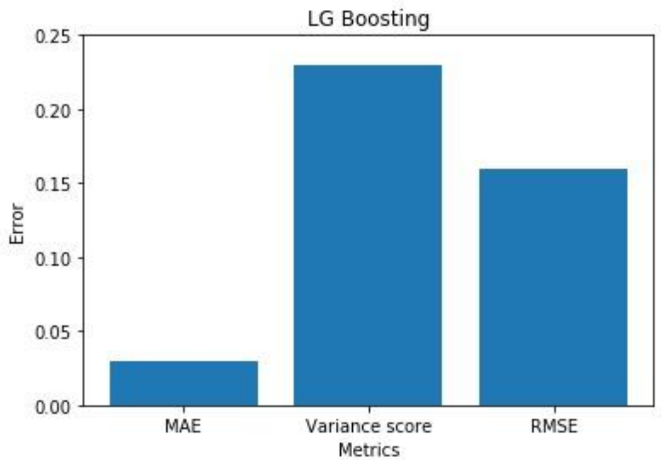


Fig 9: Light GBM vs error metrics

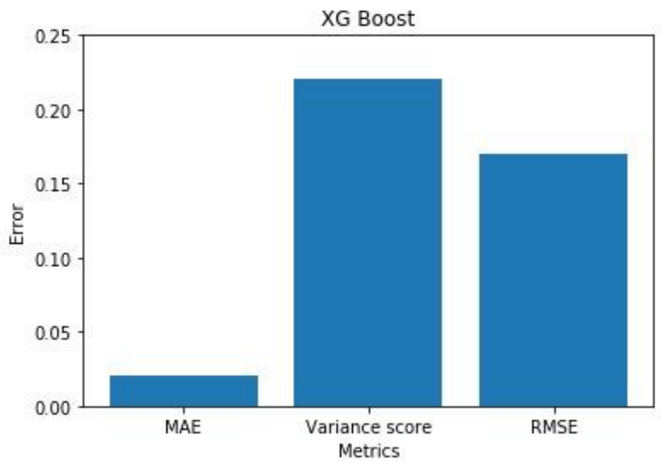


Fig 10: XGBoost vs error metrics

VI. CONCLUSION

Our analysis shows that the Light GBM and XGB performs equally well with RMSE 0.15. Both the models are based gradient boosting and boosted trees are derived by optimizing the objective function. They are ensemble of weak regressor which gets stronger in a sequential manner but linear regression is just build once and never modified as other boosting models so it doesn’t perform that well.

VII. CONTRIBUTION

Varun : Data Collection, formatting, model training (Linear Regressor, XGB)

Ashish :

Data pre-processing and model training (Light GBM, XGB)

VIII. REFERENCES

Tianqi Chen, Carlos Guestrin. XGBoost: A Scalable Tree Boosting System

Guolin Ke1 , Qi Meng , Thomas Finley , Taifeng Wang , Wei Chen , Weidong Ma , Qiwei Ye , Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics

L. Breiman. Bagging predictors. Machine learning

Google Analytics Customer Revenue Prediction
https://www.kaggle.com/c/ga-customer-revenue-prediction

Scikit learn modules
https://scikit-learn.org/stable/modules