# Google Analytics Customer Revenue Prediction

Kanchan Satpute, Vedant Mehta, Samarth Mistry, Neehar Yalamarti
*Texas A&M University*
College Station, Texas, United States

**Abstract**

**Customer Revenue prediction is a most sought-after challenge by large companies these days, to identify incremental opportunities for revenue. The more customer centric a company is, the more profits it brings in the long run. It is also often said that 80/20 rule is true for these businesses i.e. 80% of the revenue is brought by 20% of the customers, therefore marketing teams are challenged with tasks to identify marketing opportunities for promotional strategies. This paper aims to identify those factors that affect the revenue earned from a customer of Google for their Google Store business. These results could eventually help marketing departments of the Google store business earn greater revenue and have targeted marketing campaigns for those 20% of customers.**

## I. Overview

To fully understand the problem provided to us, we need to address the definition of Customer Analytics and explain how it is important to any business as well as how a business can use Customer Analytics to improve service and revenues significantly.[16]

Customer Analytics is referred to as the processes and technologies, a systematic examination of a specific company's customer information so as to identify patterns related to consumer behavior, taking the leverage of which, the company can design targeted marketing campaigns, customize and localize the services provided to customers in order to identify, attract and retain the most profitable customers.[17]

Customers nowadays have access to a lot of information online like the kind of products available, where is it available, what kind of product to buy, how much to pay for it. Which makes it increasingly important to use data analytics to understand the customers, their demands and hopefully model their behavior as well.

Better understanding of customers enables better decision making about how best to acquire more customers and retain them, identify high-paying customers in order to satisfy them and engage them effectively. Customer satisfaction analysis gives us information about customers who are happy with the products and services provided to them and who are much more likely to buy from you again. So, storing useful data and analyzing it can help the marketing team to do segmentation of customers and help the sales team identify leads and prospective customers.[17]

Customer analytics, particularly in e-commerce, helps firms prioritize customers. It helps derive uniqueness of each customer and hence, deliver communications, offers and online services that reverberate in increasing the revenues. A customer's similarity and differences are what makes personalization so effective and powerful in order to make a customer feel comfortable buying online.[18]

The Google merchandise store is where all the Google merchandise is sold, and the greatest opportunity to maximize revenue is by identifying those factors directly impacting revenue. The outcome of this project can ultimately be more actionable operational changes and a better use of marketing budgets for those companies who choose to use data analysis on top of the google analytics data.

# II. DATASET

This prediction challenge is part of an ongoing Kaggle competition[1] and is for a massive prize money of $45000, indicating the importance of such a challenge. The training data contains around 23 GB of data of each transaction made by a customer on the google merchandise store from August 1st, 2016 to April 30th, 2018. The test set for the data contains transactions in the future time period of December 1st, 2018 through January 31st 2019 and this not been released yet. For the purpose of this project, we have divided the training data itself to obtain training, validation and test set so as to measure the performance of our models. The features of the data and their description are stated in Table 1.

| Feature | Description |
|---|---|
| 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 | JSON column which contains the specifications for the device used to access the Store |
| geoNetwork | JSON column which contains information about the geography of the user. |
| trafficSource | JSON column which contains information about the Traffic Source from which the session originated. |
| visitId | An identifier for this session. |
| visitNumber | The session number for this user. |
| visitStartTime | The timestamp |
| hits | JSON columns which provides a record of all page visits |
| totals | JSON column which contains information about the page like bounces etc. |

*Table 1. Data Description*

The data set contains several columns which are in JSON format which are flattened to obtain the sub columns. This results in 35 columns which can affect transaction revenue. Several other information about the web page visit can be obtained from the JSON columns which are part of the Google Analytics data set, but only a selected few is taken which are significant and don't contain redundant values in the data set.

# III. METHODOLOGY

## A. Data Preprocessing

The biggest challenge here was handling a huge file as the data was of the magnitude of 1.7 million rows. These had to be imported in Python in chunks to perform pre - processing and further analysis of data. Each chunk taken was of 100,000 rows which was preprocessed by filling the NaN rows of specific columns.

Following that, feature engineering was done to identify the important and relevant features. These features were obtained by first flattening the JSON columns within the data set to obtain the several sub columns, out of which a selected few were chosen based on the significance of the columns as well as those which didn't have redundant information in them. In addition, columns with minimal information were discarded as well. The final data set consists of 35 columns, which include the ones listed in Table 1 and some others

including geographical features like continent, country; page visit details like new visits, page views; traffic source content i.e. the source from where they visit the page which can be an ad, campaign etc.

New features were also generated from existing features, e.g. a Boolean feature named Basket which signifies whether the ending screen was a Shopping cart or not, which might reveal interesting behavior about a consumer. Similar Boolean feature was also added which gave information whether that particular date was a holiday or not (considering only the US holidays, as highest revenue was generated from United States). Also, there was a feature named *"device_browser"* which contained information about the browser used by the visitor, which could give useful information but contained too many random browsers which had very less significance, so we categorized the main browsers and the rest of them were categorized as *"Miscellaneous"*. So, several features were changed in a similar way to make the dataset more manageable.

## B. Exploratory Data Analysis

Next step in the pipeline is EDA (Exploratory Data Analysis) which reveals interesting insights, patterns and statistics related to the customer, region they live in, the period in which the purchase is being made and so on, and how it impacts the transaction revenue (transaction revenue is considered because that is feature that we will be predicting in the test set)

All the data exploration has been done on approximately 100,000 rows randomly sampled from the data so as to reduce bias in the data. Hence, it can be assumed to be representing the overall characteristics of the whole population.

An abundance of information about the users can be found from the data which can be used for marketing and operational purposes by the Google Store Merchandise managers. Some of them are featured below.

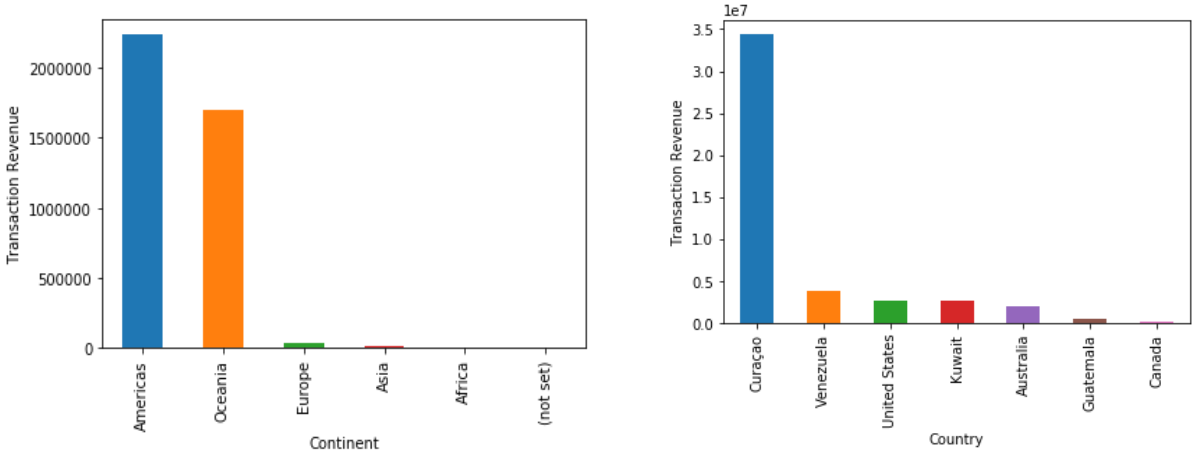


*Figure 1. a) Continent vs Transaction Revenue, b) Country vs Transaction Revenue*

As seen in Figure 1a), the Google store website is visited by customers all over the globe, majority of which are within the North and South America, with Oceania countries coming a close second. This serves as an opportunity for marketing to target specific campaigns towards Oceania for greater revenue opportunities. Although this information is only from $1/17^{th}$ of the data, it does represent a majority of the population as the data is randomly sampled. Also, if one identifies what strategy is working so better in the North and South America, and the reason, then one can devise a similar kind of strategy for other continents to increase the customer base.

In Figure 1b), Curacao is a country within the South and North America's continents. This graph can be explained because Curacao has one of the highest standards of living in the Caribbean, and has been said to have a high-income economy as defined by the World Bank.[15] It has a well-developed infrastructure built on tourism and financial services sector, which might explain such kind of behavior. If a company can identify what exact factor compels the visitor to become a buying customer, then it can develop strategies accordingly.

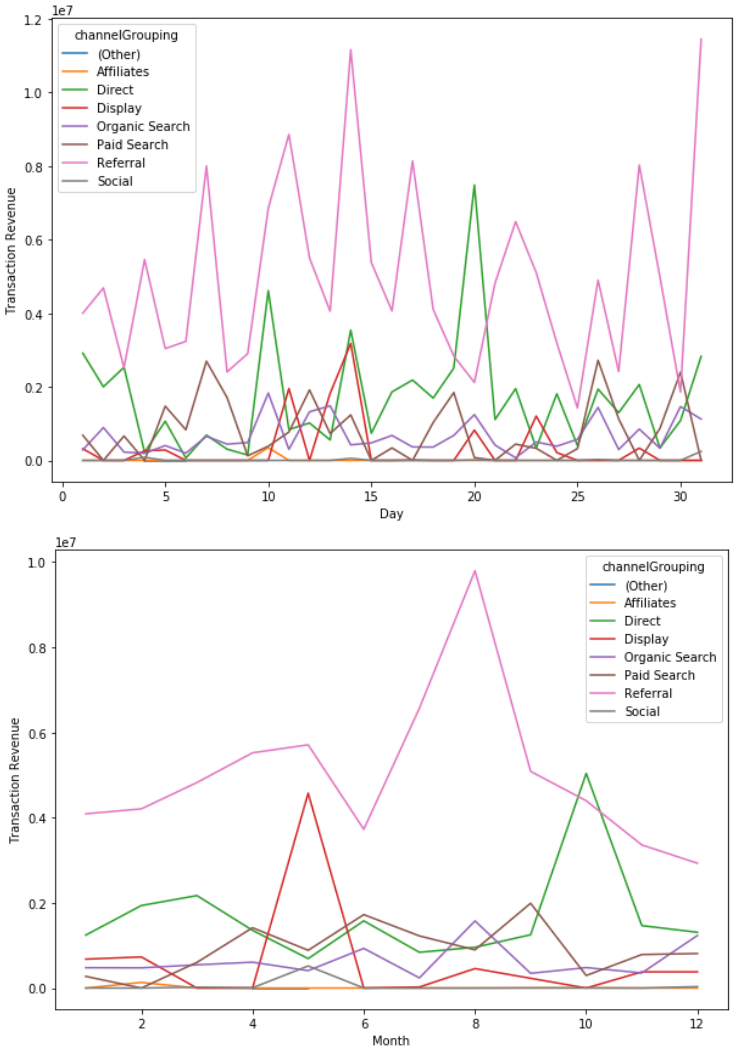


*Figure 2 a) Day vs Transaction Revenue, b) Month vs Transaction Revenue grouped by Channel source.*

Interesting results are observed from Figure 2 as transaction revenue is plotted against Day of a month and Month of a year. Channel grouping signifies the source of the customer's visit to the page. Greatest revenue is generated as customers are brought to the site via a referral link, which might have been shared by users on social platforms like Facebook, YouTube etc. Social network is considered a powerful platform of marketing. Direct traffic comes next which signifies traffic that is generated by user directly entering the URL into their browser or using a bookmark to access the website. In both these cases, it is observed from Figure 2 b) that majority of revenue is being generated in the latter half of the year. Figure 2 a) has erratic results which are hard to grab insights from and might need further investigation.

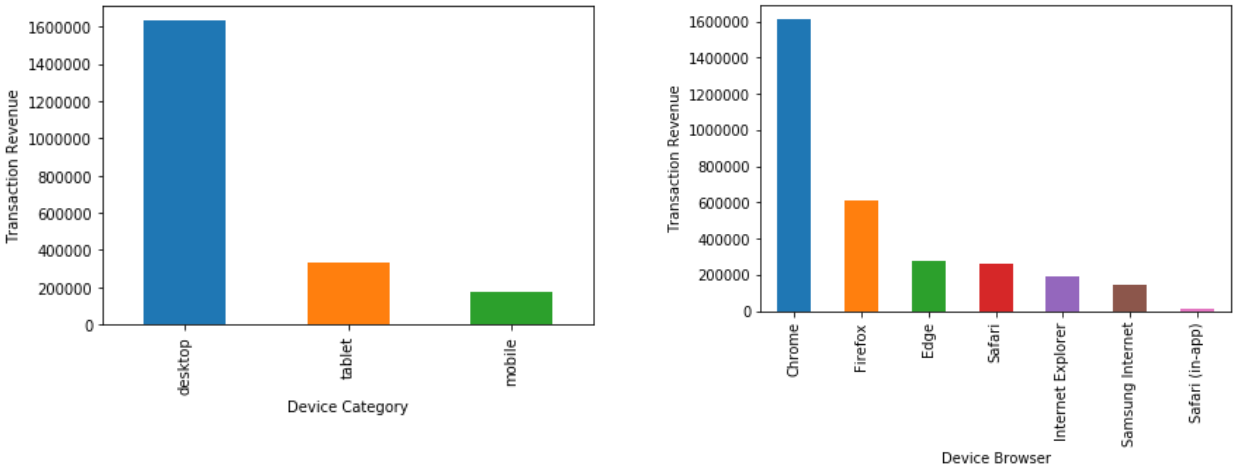


*Figure 3 a) Device Category vs Transaction Revenue, b) Device Browser vs Transaction Revenue.*

From figure 3, device sources that bring in maximum revenue is mainly the desktop. Several inferences can be taken away from here including better website development for the main website rather than focusing on the mobile app and mobile website development. Or optimizing the website further to look better on mobile devices. Chrome browser users serve as a source for the most revenue, which is expected since Chrome users are significantly larger as compared to other browsers. Firefox users can be targeted more by showing advantages of using google products like Chrome, google store etc. as google.com would be a major website visited there even though from a "non-google" owned browser.

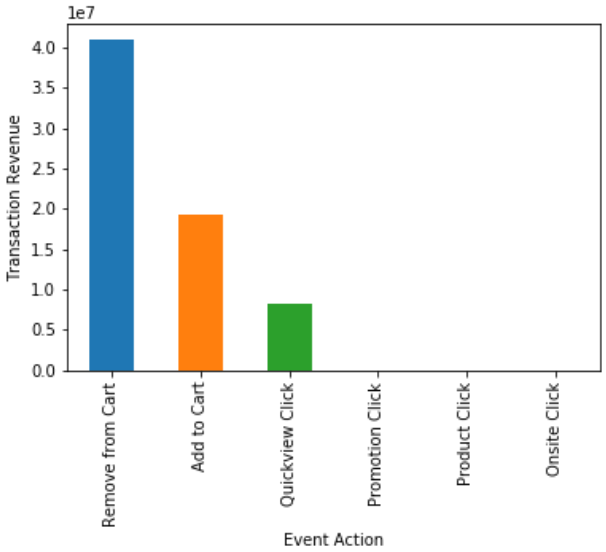


*Figure 4 Transaction Revenue vs Event Action*

When observing which event action results in maximum revenue in Figure 4, the event when a customer removes an item form a cart contributes to a greater revenue. This is interesting and is expected at the same time as this signifies a customer already has several items in his/her cart which they intend to check out with but change their mind on certain items, which they remove and then checkout. When it comes to marketing, various related items or recommended items that accompany well with the items in the cart can be shown or advertised at this page.

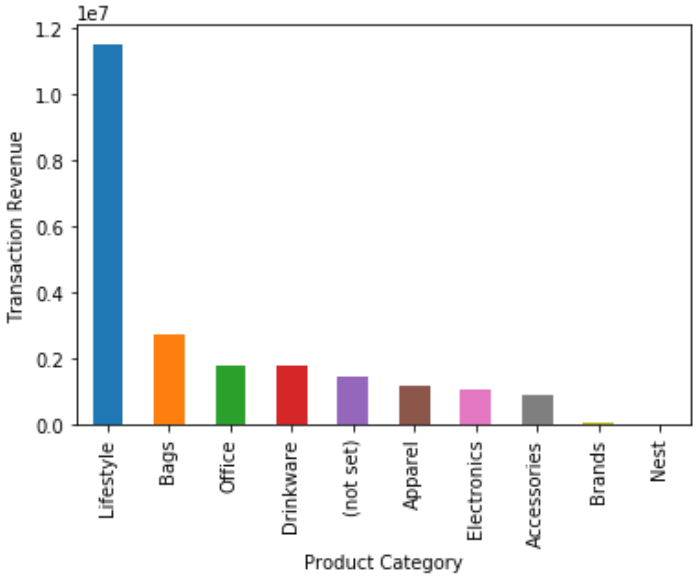


*Figure 5 Transaction Revenue vs Product Category*

Products related to lifestyle bring in the most revenue as compared to the rest by a huge margin, as seen from Figure 5. Another classic example of the 80/20 rule, which serves as greater focused strategies to be employed within the lifestyle category. This also raises concern as to the other product categories are not generating a comparable revenue compelling the company to take some actions in order to increase the revenues. More research can be done on what kind of products are customers are expecting and what kind of services are being expected for these product categories.

## C. Data Modelling

Data modelling is an integral part of the prediction procedure where the various features are fed into the machine learning models to arrive at accurate predictions. For the data modelling procedure in our project, decision trees are looked at as primary models, where we lean more towards advanced methods like boosting techniques to predict transaction revenue.

Before delving into model intricacies, a basic understanding of trees and boosting methods is warranted, as boosting are novel techniques and intuitive in their understanding. Decision tree is a model where splitting of the data is done based on features to either perform a classification or a regression task. Each split in a decision tree divides the data into one of two binary nodes. Each leaf node is allocated with a predicted value in case of a regression task. When a prediction is done with a decision tree, the data is allocated to the appropriate node, and the prediction is the average value of that leaf node.

Decision trees are a very flexible and interpretable method, but a deep decision tree results in overfitting if a tree is not pruned, and is therefore unlikely to generalize well. To avoid this shortcoming, decision trees are generally not used alone, instead, an ensemble of decision trees are used together for a prediction task. Gradient boosting decision trees is one of those methods that combines several trees and works sequentially in fitting the residuals of the predicted value at each iteration. Boosting methods are known to convert a set of weak learners into strong learners.

Three boosting methods are looked at mainly: LightGBM, XGBoost and CatBoost. These methods are widely preferred amongst the Kaggle community as well as data scientists worldwide because of their

numerous advantages. They don't require as much computing power compared to neural networks and there is wide practicality of their usage.

- **LightGBM**

The LightGBM method uses a histogram-based algorithm i.e. it creates buckets of the continuous features and converts them to discrete bins so that training is done quicker, and computation time is minimized. So as compared to other algorithms, it grows trees horizontally and not vertically i.e. it splits trees leaf wise and not level wise. As compared to other boosting algorithms, there is a significant reduction in time when working with larger data sets as it also supports parallel computing. In addition, theoretically better accuracy is obtained here since a leaf wise split approach is taken rather than the level wise approach. But leaf wise split leads to overfitting which can be taken care of by tuning the hyper-parameters.[5]

For parameter tuning, have set the number of trees to 100 and number of leaves to 30. A small learning rate of 0.1 is kept for obtaining better accuracy at each step. We have used a feature and bagging fraction of 0.5 and 0.7 respectively to avoid overfitting. This means that 50% of the features and 70% of the data will be used for each iteration for building trees. Number of boosting rounds is 1000 which is used for iterating the algorithm to obtain best results.[3][4]

- **XGBoost**

XGBoost is the short form for eXtreme Gradient Boosting. In boosting, the trees are built sequentially where each subsequent tree tries to reduce the errors of the previous tree. Since each tree learns from its previous splits and models and updates the residual errors, the tree that grows next in the sequence will learn from an updated version of the residuals. In addition, boosting makes use of trees with fewer splits. Such small trees, which are not very deep, are highly interpretable.[3][9][10]
There are several unique features of XGBoost which makes it so popular and interesting, such as:

-Regularization: XGBoost has an option to penalize based on L1 or L2 norm, which helps in preventing overfitting

-Handling Sparse data: XGBoost incorporates a sparsity-aware split finding algorithm to handle different types of sparsity

-Cache awareness: It has been designed to make optimal use of hardware

-Out-of-core computing: This particular feature optimizes available disk space and maximizes its usage which comes in handy while handling large datasets that cannot be loaded into the memory[2]

We have built XGBoost on decision trees, but it can also be built on linear models. Our objective was linear regression (*objective* = 'reg:linear'), *metric* used for validation data is 'rmse', *eta* (similar to learning rate in gradient boosted algorithms) is set to 0.001, so our learning rate is kept very low in order to avoid overshooting the optimal solution. Parameter *max_depth* is set at 10 to avoid overfitting of the data, *subsample* is set at 0.6 (which depicts the fraction of observations to be randomly sampled for building the trees). Keeping a lower value helps avoid overfitting, but very low values can also lead to under-fitting. *colsample_bytree* is also set at 0.6, so 60% of the variables will be used while building the decision trees (in each iteration). L1 regularization parameter *alpha* is set at 0.001, so penalty imposed on bigger models is very less (not much significant), because we already have removed unwanted data. So applying a higher penalty might lead to loss of information and hence under-fitting.[11]

- **CatBoost**

CatBoost converts categorical values into numbers using various statistics on combinations of categorical and numerical features, thus it can be used without any explicit preprocessing to convert categories into numbers. CatBoost, as well as all standard gradient boosting implementations, builds each new tree to approximate the gradients of the current model. However, all classical boosting algorithms suffer from overfitting caused by the problem of biased pointwise gradient estimates. Gradients used at each step are estimated using the same data points the current model was built on. However, in CatBoost, steps are performed by using unbiased estimates of the gradient, resulting in a robust and high-performance model.[7][12]

In order to avoid any under-fitting or overfitting while tuning the parameters, CatBoost algorithm has the advantage of analyzing the metric value on the validation set at each step and selecting the appropriate number of iterations. For our model, the maximum number of iterations was set to 1000 the overfitting detector stopped our model at 824th iteration to give the best RMSE value. A learning rate of 0.05 was used for reducing the gradient step.[8]

## IV. RESULTS AND DISCUSSION

Root mean squared error (RMSE) was used as the performance measure for gauging our models. The summary of results obtained is shown in Table 2.

| Model | Train RMSE | Validation RMSE | Test RMSE |
|---|---|---|---|
| LightGBM | 1.484 | 1.523 | 1.506 |
| XGBoost | 1.423 | 1.613 | 1.665 |
| CatBoost | 1.290 | 1.598 | 1.647 |

*Table 2 Model Accuracy*

The judging criteria of the model should be how well it performs and how fast does it perform.

Judging by these two criteria, LightGBM comes out as the winner, with lowest RMSE (1.506) on test dataset. LightGBM performs consistently on the training dataset and validation dataset which shows that it fits the model with minimum overfitting.

CatBoost didn't perform as good as the LightGBM, possibly because CatBoost can be taken advantage of when you have only categorical variables in the dataset. Ignoring this fact impacts the RMSE significantly. The results also point out that there might be some amount of overfitting because the training RMSE is very low but the validation RMSE increases significantly. So, we can conclude that CatBoost only performs well when we have categorical variables and we tune them properly.

XGBoost gives very comparable results to that of LightGBM and CatBoost, but the only drawback is the runtime of the algorithm. So it becomes really frustrating to do the hyper-parameter tuning. Applying GridSearch would be nothing but a nightmare, since just one combination of parameters takes approximately 40 minutes to run on a system having a basic configuration.

However, this comparison between the algorithms maybe specific to each dataset, but the general takeaway is that XGBoost is slower than the other two algorithms.[12]

Apart from the algorithms, the features that were engineered to modify the dataset helped get rid of redundant variables and clutter in each of the variables. So, according to our analysis, feature engineering played a major role in the kind of accuracy that we are getting. Data preprocessing helped include new and important features and also shrunk the dataset to a more manageable size. And hence, improved the runtime of the algorithms as well.

The importance of features according to our final LightGBM model can be seen in figure 6.

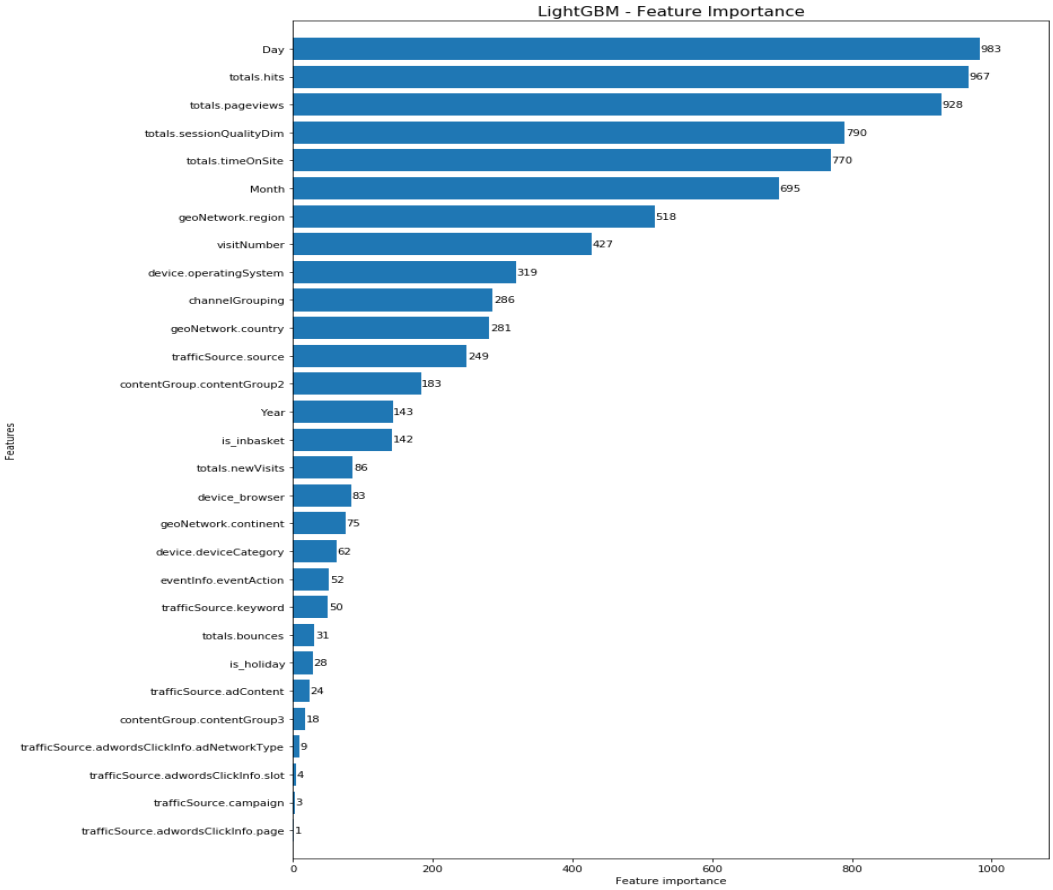


*Figure 6 Feature Importance*

In conclusion, we implemented all the algorithms by dividing the data into chunks and then pre-processed it. Due to lack of computational power, it took us a long time to do the data preprocessing and hyper-parameter tuning. Implementing these algorithms using any cloud based services like AWS might help us improve the model and its performance significantly. Next step in this project could be to implement time series forecasting analysis methods like ARIMA, moving-average, etc. for better future forecasting.

# V. References


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

[2] Which algorithm takes the crown: Light GBM vs XGBOOST? - Pranjal Khandelwal https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost

[3] LightGBM and XGBoost Explained: Keitakurita http://mlexplained.com/2018/01/05/lightgbm-and-xgboost-explained

[4] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Liu, *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

[5] A Kaggle Master Explains Gradient Boosting: Ben Gorman http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting

[6] A. V. Dorogush, V. Ershov, A. Gulin, *CatBoost: gradient boosting with categorical features support* http://learningsys.org/nips17/assets/papers/paper_11.pdf

[7] CatBoost: A machine learning library to handle categorical (CAT) data automatically: Sunil Ray https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data

[8] CatBoost: Parameter Tuning https://tech.yandex.com/catboost/doc/dg/concepts/parameter-tuning-docpage

[9] An End-to-End Guide to Understand the Math behind XGBoost: Ramya Bhaskar Sundaram https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost

[10] A Gentle Introduction to XGBoost for Applied Machine Learning: Jason Brownlee https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning

[11] Complete Guide to Parameter Tuning in XGBoost : Aarshay Jain https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python

[12] CatBoost vs. Light GBM vs. XGBoost : Alvira Swalin https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

[13] What is LightGBM, How to implement it? How to fine tune the parameters? : Pushkar Mandot https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

[14] Introduction to Boosting using LGBM: Vinay Patlolla https://www.kaggle.com/vinnsvinay/introduction-to-boosting-using-lgbm-lb-0-68357

[15] Wikipedia: Economy of Curaçao https://en.wikipedia.org/wiki/Economy_of_Cura%C3%A7ao

[16] Why customer analytics matter https://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/why-customer-analytics-matter

[17] Customer Analytics https://searchbusinessanalytics.techtarget.com/definition/customer-analytics

[18] Why Customer Analytics in E-Commerce https://www.sensiple.com/blog/why-customer-analytics-e-commerce