Machine Learning

CS6375.004 Final Project Report

**Google Analytics Customer Revenue Prediction**

## Team Members:

Rama Narayan Lakshmanan (rxl174430)

Shubham Raosaheb Kharde (sxk173732)

Rohit Seetepalli (rxs170631)

# 1. Introduction:

Marketing teams are challenged to make appropriate investments in promotional strategies as only small percentage of customers bring the most revenue (80%-20% rule). In this challenge, RStudio in collaboration with Google cloud and Kaggle aims to demonstrate this business impact thorough data analysis. The data that they have provided for this challenge reflects this revenue model. As part of this challenge, we are required to predict how much GStore customers will spend. GStore is an online store that sells google merchandise. The aim of this project is to train a machine learning model that best predicts revenue per customer with good metrics. We will use ensemble techniques like different types of boosting and random forests to come up with the best model. Results of the techniques with various parameters will be summarized and the best model will be presented.

# 2. Problem Definition and Algorithm:

## 2.1 Task Definition

The problem is to predict the natural log of sum of all transactions per user. It can be represented as:

The dataset that we use is from Kaggle’s Google Analytics Customer Revenue Prediction challenge [1]. There are two aspects to the challenge:

* Public leaderboard – Revenue prediction for customers during the same timeframe of 5/1/18 to 10/15/18 given in test dataset
* Private Leaderboard – Revenue prediction for customers during a future timeframe of 12/1/18 to 1/31/19.

In this project, we are solving the public leaderboard challenge.

Input to trained Model – Test.csv

Output – Prediction column with predicted log revenue

## 2.2 Algorithm Definition

The following is the algorithm that we used to get the best model. It has three main steps – preprocessing, training and testing.

* Pre-processing
  + Split train\_v2.csv into 16 different chunks so that it becomes manageable for processing
  + For each chunk,
    - Flatten all json columns – "device", "trafficSource", "geoNetwork", "totals", “customDimensions”, “hits”
    - Identify inconsistent columns between the chunks and for every column:
      * Find correlation between target and inconsistent column.
      * Drop in case of low correlation or add the column with default values.
    - Identify mixed types in features and convert them into a single data type based on the semantics of the column.
    - Resolve NaN, None and null values :
      * Replace them with default values based on their datatype
  + Concatenate all chunks into a single file.
  + Remove features that have constant values.
    - Apply filter: Number of unique values = 1
  + Parse datetime features:
    - Split datetime features that are given in POSIX format to day, month, year and hour.
    - Categorize hour into one of the following – “early morning”, “morning”, “afternoon”, “evening”, “night”, “midnight”.
  + Drop the datetime features that have been processed.
  + Parse the url features by just taking their host names and drop the processed feature.
  + Encode category data to be numeric.
  + Remove highly correlated features.
  + Scale the features using standard scaler.
  + Perform Principle Component Analysis (PCA) for further dimensionality reduction.
  + Log scale the target column.
* Training
  + Since dataset is huge, we will use the following strategy
    - Randomly sample 10 % of data
    - Fine tune the models on this data and get the best model for each algorithm.
    - Train and tune the top models on the whole dataset and find the best model.
  + Split the sampled dataset into training and testing in 70:30 ratio.
  + Gradient boosting:
    - Build a PARAM grid to tune the model with following parameters in a 5 fold cross validation.
      * loss
      * learning rate
      * number of estimators
      * max depth

Use cross validator to find the best model with the algorithm’s default scoring metric.

* + Random Forest Model
    - Build a PARAM grid to tune the model with following parameters in a 5 fold cross validation.
      * Max features
      * Warm start
      * Number of estimators
      * Max depth

Use cross validator to find the best model with the algorithm’s default scoring metric.

* + Adaboost:
    - Build a PARAM grid to tune the model with following parameters in a 5 fold cross validation.
      * loss
      * learning rate
      * number of estimators
      * random state

Use cross validator to find the best model with the algorithm’s default scoring metric.

* + LGBM:
    - Tune the model in a 5 fold cross validation.
  + XGBoost:
    - Tune the model with custom parameters.
  + Find best models in each algorithm.
* Testing
  + Regression Metrics:
    - RMSE – Root Mean Square error
    - R-Squared to determine the goodness of the fit
    - MAE – Mean Absolute Error
    - Explained Variance
    - MSE –Mean Square Error
  + Use the top models on the testing data.
  + We have predicted the log revenue. To get the competition metric do the following:
    - Inverse log the prediction column
    - Group the prediction by “fullVisitorId”
    - Apply log again
  + Get the regression metrics on this data.
  + Based on the metrics obtained, decide on the best model. Our goal is to choose a model with highest R-Squared and least RMSE value.

As this an ongoing competition, we list out the steps on how to prepare and upload the results to Kaggle.

* Run the preprocessing on test.csv
* Run the chosen best model on the preprocessed test data.
* Select only “fullVisitorId” column and “prediction” column.
* Export it as csv and upload it to Kaggle website.

# 3. Technology Used

* Python
* Scikit-learn library
* MS Excel for data exploration

# 4. Experimental Evaluation

## 4.1 Methodology

### 4.1.1 Dataset Description

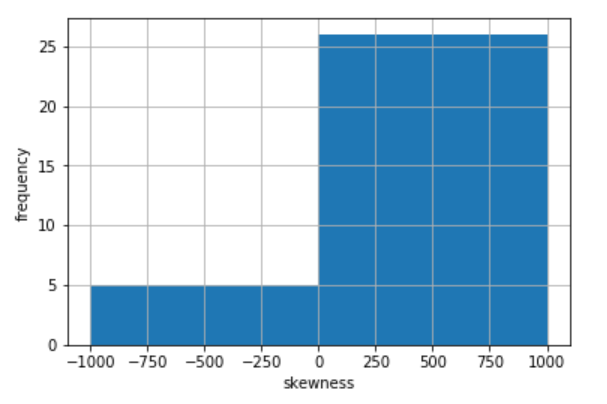
There are two datasets provided:

* train\_v2.csv
  + No of records – 1,708,344
  + No of features – 14, 6 of which are deep nested json blobs which we will expand during preprocessing.
  + Label column – transactionRevenue nested within “totals” feature.
* test\_v2.csv
  + No of records – 401,589
  + No of features – 14, 6 of which are deep nested json blobs which we will expand during preprocessing.

We have two major challenges that makes this dataset interesting. The first challenge is to flatten the deep nested json blob features. The second challenge is the humongous training data and right skewed target data. We have 1,708,344 records of which 1,689,830 records have transaction revenue of 0 which is 98.91 % of data. This is in accordance with the theme of this challenge that smaller number of people bring more revenue.

#### Data Distribution:

There are many non-numeric columns. Before pre-processing, we just had a look at the skewness of numeric columns. Some of them are left skewed and some are right skewed as observed in the below histogram.



### 4.1.2 Data Preprocessing

#### Data Flattening:

There are six json blobs to flatten:

* customDimension
* device
* geoNetwork
* hits
* totals
* trafficSource

##### customDimesion

Has index and value. Flatten it to customDimensions\_index and customDimensions\_value. We are not interested in index. So drop it.

##### Device

Flatten it to device\_{attributes}

##### geoNetwork

Flatten it to geoNetwork\_{attributes}

##### hits

Flatten it to hits\_{attributes}. Drop the json blobs that remain after flattening.

##### totals

Flatten it to totals\_{attributes}. “totals\_transactionRevenue” is the target column

##### trafficSource

Flatten it to trafficSource\_{attributes}. There is one more blob “trafficSource\_adwordsClickInfo”. Flatten it again to trafficSource\_adwordsClickInfo\_{attributes}

#### Feature Engineering

##### Inconsistent features

Since we are processing in chunks, not all chunks have all the features. Identify the inconsistent features and apply the following steps for each inconsistent column.

* Find correlation between inconsistent column and target
* Drop if the correlation is very less or add the column with default value in the chunks that doesn’t have them.

At the end of this process, total features – 120 (excluding target column)

##### Constant Features

Drop all the features which have a constant value for all records. The features with constant values are:

*['device\_browserVersion', 'device\_browserSize', 'device\_operatingSystemVersion', 'device\_mobileDeviceBranding', 'device\_mobileDeviceModel', 'device\_mobileInputSelector', 'device\_mobileDeviceInfo', 'device\_mobileDeviceMarketingName', 'device\_flashVersion', 'device\_language', 'device\_screenColors', 'device\_screenResolution', 'geoNetwork\_cityId', 'geoNetwork\_latitude', 'geoNetwork\_longitude', 'geoNetwork\_networkLocation', 'trafficSource\_adwordsClickInfo\_criteriaParameters', 'hits\_appInfo.screenDepth', 'hits\_contentGroup.contentGroup4', 'hits\_contentGroup.contentGroup5', 'hits\_page.searchCategory', 'hits\_time']*

Total Features Dropped: 22

Remaining features: 98

##### Incorrect Features

Drop the following features

*'hits\_social.socialInteractionNetworkAction'* - has only “:” and “unknown” as values

*trafficSource\_adwordsClickInfo\_targetingCriteria* - has only “{}” and “unknown” as values

Total Features Dropped: 2

Remaining features: 96

##### Redundant Features

Drop the following redundant features:

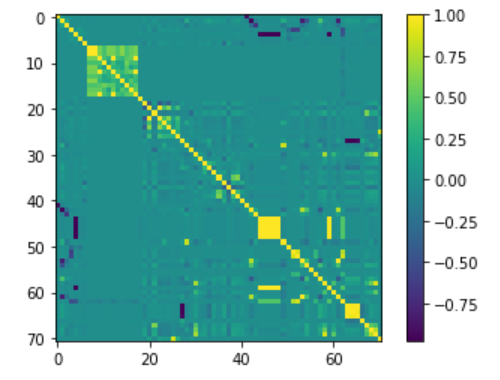
* *"customDimensions\_index"* – It’s key for another column *"customDimensions\_value"*
* *"hits\_page.pagePath" , "hits\_page.pagePathLevel1" , "hits\_page.pagePathLevel2" ,"hits\_page.pagePathLevel3","hits\_page.pagePathLevel4"* – All this information about the page is aggregated in *“hits\_page.pageTitle”*
* *“trafficSource\_referralPath”* - information is covered in *“trafficSource\_source”*
* *“hits\_appInfo.screenName*” - same as *“hits\_appInfo.landingScreenName”*
* “*trafficSource\_adwordsClickInfo\_gclId*” - not required as its unique id for all clicks.

Total Features Dropped: 9

Remaining features: 87

##### Correlated Features

We found pearson correlation between the features and dropped features that have a correlation of more than 0.8 for positive correlation and less than -0.8 for negative correlation.



Darker shade of yellow indicates high positive correlation and darker shade of blue indicates high negative correlation.

Total Features Dropped: 17

Remaining features: 70

##### Feature addition

###### Date Features

Split the date features as month, day and year. Drop the date feature once you split

Date feature – “date”

New features added – 3

Features dropped – 1

Total features - 72

###### DateTime Features

Split the datetime features as month, day, year and hour in UTC. Categorize hour as

* 12am – 3.59am – Midnight
* 4am - 6.59am – EarlyMorning
* 7am – 11.59am – Morning
* 12pm – 3.59pm – Afternoon
* 4pm – 6.59pm – evening
* 7pm – 11.59pm – night

Date feature – “visitStartTime”

New features added – 4

Features dropped – 1

Total features – 75

###### Additional Features

We added one additional feature.

*“isEntryExitSame”* – Boolean to determine if the landing screen the same as the exit screen of the user’s visit

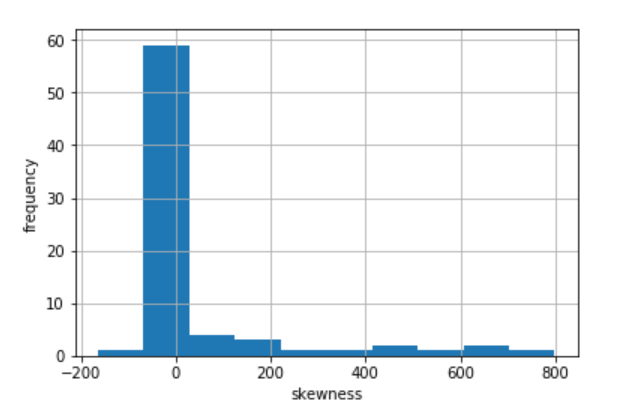
Total Features after feature engineering – 76

###### Final Feature Description

|  |  |
| --- | --- |
| Features | Description |
| channelGrouping | The Default Channel Group associated with an end user's session for this View. |
| customDimensions\_value | The value of the custom dimension. |
| date\_day | date in day |
| date\_month | date in month |
| date\_year | date in year |
| device\_browser | The browser used (e.g., "Chrome" or "Firefox"). |
| device\_deviceCategory | The type of device (Mobile, Tablet, Desktop). |
| device\_isMobile | If the user is on a mobile device, this value is true, otherwise false. |
| device\_operatingSystem | The operating system of the device (e.g., "Macintosh" or "Windows"). |
| fullVisitorId | The unique visitor ID (also known as client ID). |
| geoNetwork\_city | Users' city, derived from their IP addresses or Geographical IDs. |
| geoNetwork\_continent | The continent from which sessions originated, based on IP address. |
| geoNetwork\_country | The country from which sessions originated, based on IP address. |
| geoNetwork\_metro | The Designated Market Area (DMA) from which sessions originate. |
| geoNetwork\_networkDomain | The domain name of user's ISP, derived from the domain name registered to the ISP's IP address. |
| geoNetwork\_region | The region from which sessions originate, derived from IP addresses. In the U.S., a region is a state, such as New York. |
| geoNetwork\_subContinent | The sub-continent from which sessions originated, based on IP address of the visitor. |
| hits\_appInfo.exitScreenName | The exit screen of the session. |
| hits\_appInfo.landingScreenName | The landing screen of the session. |
| hits\_contentGroup.contentGroup1 | The content group on a property. A content group is a collection of content that provides a logical structure that can be determined by tracking-code or page-title/URL regex match, or predefined rules. |
| hits\_contentGroup.contentGroup3 | The content group on a property. A content group is a collection of content that provides a logical structure that can be determined by tracking-code or page-title/URL regex match, or predefined rules. |
| hits\_contentGroup.contentGroupUniqueViews3 | number of unique content group views |
| hits\_dataSource | The data source of a hit. By default, hits sent from analytics.js are reported as "web" and hits sent from the mobile SDKs are reported as "app". |
| hits\_eCommerceAction.action\_type | action type |
| hits\_eCommerceAction.option | This field is populated when a checkout option is specified. For example, a shipping option such as option = 'Fedex'. |
| hits\_eCommerceAction.step | This field is populated when a checkout step is specified with the hit. |
| hits\_eventInfo.eventCategory | The event category. |
| hits\_eventInfo.eventLabel | The event label. |
| hits\_exceptionInfo.isFatal | If the exception was fatal, this is set to true. |
| hits\_hitNumber | The sequenced hit number. For the first hit of each session, this is set to 1. |
| hits\_hour | The hour in which the hit occurred (0 to 23). |
| hits\_isEntrance | If this hit was the first pageview or screenview hit of a session, this is set to true. |
| hits\_isExit | If this hit was the last pageview or screenview hit of a session, this is set to true. |
| hits\_isInteraction | If this hit was an interaction, this is set to true. If this was a non-interaction hit (i.e., an event with interaction set to false), this is false. |
| hits\_item.currencyCode | The local currency code for the transaction. |
| hits\_latencyTracking.domainLookupTime | The total time (in milliseconds) all samples spent in DNS lookup for this page. |
| hits\_latencyTracking.domInteractiveTime | The time (in milliseconds), including the network time from users' locations to the site's server, the browser takes to parse the document (DOMInteractive). |
| hits\_latencyTracking.domLatencyMetricsSample | Sample set (or count) of pageviews used to calculate the averages for site speed DOM metrics. |
| hits\_latencyTracking.pageDownloadTime | The total time (in milliseconds) to download this page among all samples. |
| hits\_latencyTracking.pageLoadSample | The sample set (or count) of pageviews used to calculate the average page load time. |
| hits\_latencyTracking.pageLoadTime | Total time (in milliseconds), from pageview initiation (e.g., a click on a page link) to page load completion in the browser, the pages in the sample set take to load. |
| hits\_latencyTracking.redirectionTime | The total time (in milliseconds) all samples spent in redirects before fetching this page. If there are no redirects, this is 0. |
| hits\_latencyTracking.serverConnectionTime | Total time (in milliseconds) all samples spent in establishing a TCP connection to this page |
| hits\_latencyTracking.serverResponseTime | The total time (in milliseconds) the site's server takes to respond to users' requests among all samples; this includes the network time from users' locations to the server. |
| hits\_latencyTracking.speedMetricsSample | The sample set (or count) of pageviews used to calculate the averages of site speed metrics. |
| hits\_minute | The minute in which the hit occurred (0 to 59). |
| hits\_page.hostname | The hostname of the URL. |
| hits\_page.pageTitle | The page title. |
| hits\_page.searchKeyword | If this was a search results page, this is the keyword entered. |
| hits\_promotionActionInfo.promoIsClick | True if the Enhanced Ecommerce action is a promo click. |
| hits\_promotionActionInfo.promoIsView | True if the Enhanced Ecommerce action is a promo view. |
| hits\_referer | The referring page, if the session has a goal completion or transaction. If this page is from the same domain, this is blank. |
| hits\_social.hasSocialSourceReferral | A string, either Yes or No, that indicates whether sessions to the property are from a social source. |
| hits\_social.socialNetwork | The social network name. This is related to the referring social network for traffic sources; e.g., Google+, Blogger. |
| isEntryExitSame | Is landing screen same as exit screen |
| totals\_bounces | Total bounces (for convenience). For a bounced session, the value is 1, otherwise it is null. |
| totals\_hits | Total number of hits within the session. |
| totals\_newVisits | Total number of new users in session (for convenience). If this is the first visit, this value is 1, otherwise it is null. |
| totals\_sessionQualityDim | An estimate of how close a particular session was to transacting, ranging from 1 to 100, calculated for each session |
| totals\_timeOnSite | Total time of the session expressed in seconds. |
| totals\_totalTransactionRevenue | Total transaction revenue, expressed as the value passed to Analytics multiplied by 10^6 |
| totals\_transactions | Total number of ecommerce transactions within the session. |
| trafficSource\_adContent | The ad content of the traffic source. Can be set by the utm\_content URL parameter. |
| trafficSource\_adwordsClickInfo\_isVideoAd | True if it is a Trueview video ad. |
| trafficSource\_adwordsClickInfo\_targetingCriteria | Google Ads targeting criteria for a click. There are multiple types of targeting criteria, but should have only one value for each criterion. |
| trafficSource\_campaign | The campaign value. Usually set by the utm\_campaign URL parameter. |
| trafficSource\_isTrueDirect | True if the source of the session was Direct |
| trafficSource\_keyword | The keyword of the traffic source, usually set when the trafficSource.medium is "organic" or "cpc". Can be set by the utm\_term URL parameter. |
| trafficSource\_medium | The medium of the traffic source. Could be "organic", "cpc", "referral", or the value of the utm\_medium URL parameter. |
| trafficSource\_source | The source of the traffic source. Could be the name of the search engine, the referring hostname, or a value of the utm\_source URL parameter. |
| visitId | An identifier for this session |
| visitNumber | The session number for this user. If this is the first session, then this is set to 1. |
| visitStartTime\_day | Visit start time in day |
| visitStartTime\_hour | Visit start time in hour |
| visitStartTime\_month | Visit start time in month |
| visitStartTime\_year | Visit start time in year |

Table content reference - <https://support.google.com/analytics/answer/3437719?hl=en>

#### Scaling:

We encoded all category data except “fullVisitorId” (Id column) to numeric and calculated skewness. Some features are right skewed and some are left skewed as observed in the following histogram.

We performed standard scaling on the data.

## 4.1.3 Algorithm Selection and Model Evaluation

### Regression Techniques used

We tried the following ensemble techniques

* Gradient Boosting
* Random Forests
* Adaboost
* LGBM
* XGBoost

### Model evaluation metrics

*RMSE =*

*MAE =*

*R-Squared =*

*Where SSE =*

*SSTO =*

*Explained Variance =*

*MSE =*

## 4.2 Experimental Results and Analysis

### 4.2.1. Gradient Boosting Regressor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GB Parameters | RMSE | MSE | MAE | EX Var | R-Squared |
| PCA-50  10% Training data  Default Parameters | 2.5924 | 6.7207 | 0.4965 | -0.4515 | -0.47412 |
| Without PCA  10% Training data  Default Parameters | 0.1015 | 0.0103 | 0.0118 | 0.9977 | 0.9977 |
| Without PCA  10% Training data  CV\_folds = 5 | param\_grid=[ {'loss': [‘ls’, ‘lad’], 'learning\_rate': [0.01, 0.1], 'n\_estimators': [100, 300], 'max\_depth': [3, 5]}] | | | | |
| 0.1019 | 0.0103 | 0.0119 | 0.9977 | 0.9977 |
| Best\_ Params = { ‘loss’:’ls’, 'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100 }  Best\_CV\_Score = 0.9998 | | | | |
| Without PCA  Whole Training data  Default Parameters | 0.0340 | 0.0011 | 0.0037 | 0.9997 | 0.9997 |
| Without PCA  Whole Training data  n\_estimators = 1000  Learning\_rate = 0.01  Max\_depth = 5 | 0.0339 | 0.0011 | 0.0036 | 0.9997 | 0.9997 |

### 4.2.2. Random Forest Regressor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RF Parameters | RMSE | MSE | MAE | EX Var | R-Squared |
| PCA-50  10% Training data  Default Parameters | 2.2156 | 4.9091 | 0.4182 | -0.0608 | -0.0767 |
| Without PCA  10% Training data  Default Parameters | 0.1023 | 0.0104 | 0.0118 | 0.9977 | 0.9977 |
| Without PCA  10% Training data  CV\_folds = 5 | param\_grid = [ { 'max\_features': ['auto', 'sqrt'], 'warm\_start': [True, False], 'n\_estimators': [400, 800], 'max\_depth': [10, 20] } ] | | | | |
| 0.1191 | 0.0141 | 0.0165 | 0.9968 | 0.9968 |
| Best\_Params = {'max\_depth': 10, 'max\_features': 'auto', 'n\_estimators': 400, 'warm\_start': True}  Best\_CV\_Score = 0.9998 | | | | |
| Without PCA  Whole Training data  Default Parameters | 0.0350 | 0.0012 | 0.0037 | 0.9997 | 0.9997 |
| Without PCA  Whole Training data  n\_estimators = 1000  Max\_depth = 20 | 0.0347 | 0.0012 | 0.0037 | 0.9997 | 0.9997 |

# 4.2.3. AdaBoost Regressor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AB Parameters | RMSE | MSE | MAE | EX Var | R-Squared |
| PCA-50  10% Training data  Default Parameters | 2.1552 | 4.6450 | 1.0058 | 0.0619 | -0.0188 |
| Without PCA  10% Training data  Default Parameters | 0.1144 | 0.0130 | 0.0127 | 0.9971 | 0.9971 |
| Without PCA  10% Training data  CV\_folds = 5 | param\_grid=[ { ‘loss’:[‘linear’, ‘square’], 'learning\_rate': [0.01, 0.1, 1.0], 'n\_estimators': [100, 300], ‘random\_state’: [None,42] } ] | | | | |
| 0.1159 | 0.0134 | 0.0127 | 0.9970 | 0.9970 |
| Best\_Params = { ‘loss’:’linear’, 'learning\_rate': 1.0, 'n\_estimators': 100, ‘random\_state’:None }  Best\_CV\_Score = 0.9994 | | | | |
| Without PCA  Whole Training data  Default Parameters | 0.0707 | 0.0050 | 0.0070 | 0.9989 | 0.9989 |
| Without PCA  Whole Training data  n\_estimators = 50  learning\_rate = 1.0 | 0.0771 | 0.0059 | 0.0077 | 0.9987 | 0.9986 |

# 4.2.4. Microsoft Light GBM Regressor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AB Parameters | RMSE | MSE | MAE | EX Var | R-Squared |
| PCA-50  10% Training data  Default Parameters | 2.1388 | 4.5746 | 0.4019 | 0.0016 | -0.0033 |
| Without PCA  10% Training data  Default Parameters | 0.1021 | 0.0104 | 0.0118 | 0.9977 | 0.9977 |
| Without PCA  10% Training data  CV\_folds = 5 | lgbm\_params = { 'boosting\_type': 'gbdt', 'objective': 'regression', 'metric': 'rmse', 'n\_jobs':-1,  "learning\_rate": 0.05,"num\_leaves": 31, "max\_depth": 5, "reg\_alpha": 0.05, "reg\_lambda": 0.1 } | | | | |
| 0.0346 | 0.0011 | 0.0038 | 0.9997 | 0.9997 |
| Best\_CV\_Score = 0.0202  Optimal\_rounds = 173 | | | | |
| Without PCA  Whole Training data  Default Parameters | 0.0343 | 0.0011 | 0.0037 | 0.9997 | 0.9997 |
| Without PCA  Whole Training data  n\_estimators = 50  learning\_rate = 1.0 |  |  |  |  |  |

# 4.2.5. XGBoost

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AB Parameters | RMSE | MSE | MAE | EX Var | R-Squared |
| PCA-50  10% Training data  Default Parameters | 2.5118 | 6.3091 | 0.4740 | -0.3645 | -0.3838 |
| Without PCA  10% Training data  Default Parameters | 0.1015 | 0.0103 | 0.0119 | 0.9977 | 0.9977 |
| Without PCA  Whole Training data  Default Parameters |  |  |  |  |  |
| Without PCA  Whole Training data  n\_estimators = 50  learning\_rate = 1.0 |  |  |  |  |  |

# 5. Future Work

In this project, we restricted our techniques to ensemble learning algorithms. We recommend to use other powerful learners like deep learning models and SVM to get better results.

# 6. Conclusion

The enormity of dataset was a huge challenge as we required a lot of compute to process it. We tackled it by splitting them into chunks, processing on each chunk and combining them together. Preprocessing was important as we had a lot of features with deep nested json blobs. Flattening the json columns, removing inconsistent columns, dropping constant and highly correlated columns were part of the preprocessing we did. We tried PCA for dimensionality reduction but had to disregard it as it didn’t give better results. Of all the models we ran, Gradient Boosting gave better results in terms of RMSE. Kaggle’s public leaderboard metric was RMSE of predicted log revenues. We scored 0.0339 RMSE using Gradient Boosting.

# 7. Contribution of team members

|  |  |  |
| --- | --- | --- |
| **RAMA NARAYAN LAKSHMANAN** | **SHUBHAM RAOSAHEB KHARDE** | **ROHIT SEETEPALLI** |
| Dataset Flattening | Random Forest Training | Initial Data Analysis |
| Feature Engineering | LGBM training | Dataset Partition |
| Train data preprocessing | Gradient Boosting training | PCA |
| Test Data Preprocessing | XGBoost training | Adaboost Training |
| Project Report | Result aggregation in Report | Source Code Preparation |

# 8. References

[1] <https://www.kaggle.com/c/ga-customer-revenue-prediction>

https://support.google.com/analytics/answer/3437719?hl=en

Stack overflow for fixing errors