……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Name:** Koshal Kumar

**Email address:** koshalini0921@gmail.com

**Contact number:** 8210004520

**Anydesk address: not installed yet**

**Years of Work Experience: 0**

**Date: 23rd Sep 2021**

……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Self Case Study -1: Google Analytics Customer Revenue Prediction**

“After you have completed the document, please submit it in the classroom in the pdf format.”

Please check this video before you get started: <https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg>

# **Overview**

\*\*\* Write an overview of the case study that you are working on. ***(MINIMUM 200 words)*** \*\*\*

1. **Introduction** :- “The 80/20 rule has proven true for many businesses–only a small percentage of customers produce most of the revenue. As such, marketing teams are challenged to make appropriate investments in promotional strategies.”

In simple words, only a fraction of people are the potential consumer of a product through which the companies generate revenues. And we have to find/ identify those potential customers and give the data to the company so that they can promote their products to the right person or the right locality. In this way any company could save a lot of money because they are investing on only 20% of the people now.

1. **Business Problem** :-For this Kaggle competition R-studio, Google cloud and Kaggle are partnered together to demonstrate the business impact that data analysis can have. For this competition, we will be getting data from google merchandise store, which is also known as Gstore where Google swag is sold. So, using the Gstore customer dataset, we have to build some machine learning model that could identify potential customers.

The outcome of this competition would be less expense on non potential customers and more on potential customers. This would make the marketing team more productive and thus revenue of the company will increase.

1. **ML formulation of this business problem** :-
   1. We have seen that only 20% of the people from the crowd generate revenue for the company. So, we have to classify those 20% from the crowd. This it’s a binary classification problem.
   2. Furthermore, we can also identify and calculate the revenue that can be generated by each individual. Thus, it can also be further formulated as a Regression problem.
2. **Business constraints** :-
   1. The cost of classification error would depend on whether it’s a Type I error or a Type II error. Type I error is not that fatal, But Type II error can cost a huge loss to the company.
   2. It would be great to have some probability associated with the classification, so that we can choose a threshold a/c to need.
   3. No strict latency concerns.
   4. Interpretability is partially important.
3. **Dataset Column Analysis**

## 

* ***fullVisitorId***- A unique identifier for each user of the Google Merchandise Store, i.e., an unique code to identify each unique user.
* ***channelGrouping*** - The channel via which the user came to the Store i.e., nature of the channel like organic, referral or paid etc.
* ***date*** - The date on which the user visited the Store.
* ***device*** - The specifications for the device used to access the Store(every details of the device including browser detail).
* ***geoNetwork*** - This section contains information about the geography of the user(like country or any particular region).
* ***socialEngagementType*** - Engagement type, either "Socially Engaged" or "Not Socially Engaged".i.e., either the user is socially active or not.
* ***totals*** - This section contains aggregate values across the session i.e., information about the visit.
* ***trafficSource*** - This section contains information about the Traffic Source from which the session originated i.e., which website or other source.
* ***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.

1. **Performance metric :-** 
   1. We can use Root Mean Squared Error (**RMSE**)

RMSE = sqrt((1/n)\*(yi-yi\_)\*\*2)

* 1. Confusion matrix :- through this matrix we can minimize Type II error.

# 

# **Research-Papers/Solutions/Architectures/Kernels**

\*\*\* Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. It is mandatory to write a brief description about that paper. Without understanding of the resource please don’t mention it\*\*\*

1. The objective of this paper is to find the best algorithm among several ML algorithms to understand which method is best suited to this kind of dataset.

In order to achieve the objective they went through the following process :-

1. First they preprocess the data collected from the kaggle.
2. Then they implemented several regression algo such as linear regression, polynomial regression, Decision tree regression, Random forest Regression, Perceptron regression and Neural Network.
3. Finally, they compared the performance by using learning curves and cross validation for each of the implementations.

As the training dataset is of 24GB, the researchers have used a ‘chunksize’ parameter in order to load a part of the dataset. ‘Hits’ and ‘customDimensions’ columns are censored by Google thus they dropped these columns. ‘visitId’ column also acted as the index of the dataset, so they dropped it too. Now, they have 4 json columns ‘device’, ‘geoNetwork’, ‘trafficSource’ and ‘totals’. They flattened the json columns and concatenated with the original dataset. Now, they have a total of 57 columns. This whole process took 90 minutes on training data. They replaced ‘not available in the demo dataset’, ‘unknown.unknown’, ‘ ’, ‘None’, ’nan’, ’(none)’ and ‘(not set)’ with **nan.** They also removed the columns that are more than 50% empty. After doing these preprocessing tasks, they left with 20 columns including the target column. Now, for the small percentage of empty data, they used imputation and replaced all empty values in the categorical columns with the most frequent values and all empty values in numerical columns with the mean of that column. Now they have their final dataset and with that they hypertuned the different model using GridSearch.

Although the polynomial Regression had the least error while learning, still it can’t be an optimal choice due to time consumption during training, they suggested. If their goal was to get a simple model which had less consumption then they suggest using linear regression. They found that both Decision Tree and Random Forest are best suited for large amounts of data.

They concluded that choosing a model depends on the user requirements and dataset characteristics.

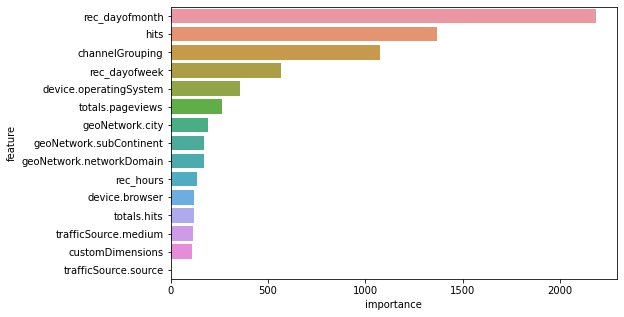
<https://drive.google.com/file/d/1YMz7BamS6Ky0cydiiZ14daCOl0zS-w-1/view?usp=sharing>

2. **Basics of Google Analytics**

* The basic output of this kernel is that they want to show a clear way and necessary steps needed to be taken for similar analysis tasks.
* The very first thing they did was univariate analysis of transaction revenue column and then performed some bivariate analysis along with columns where they found following information:-
  + “There are 4000 records for 3710 users.
  + A very low percentage of users contribute to generating revenue.
  + Users generating revenue use Macintosh more than other users
  + Their channel grouping is referral rather than organic search
  + They use more Chrome
  + They usually visit more than once
  + Their network domain is less unknown.unkown
  + They have higher time on site
  + They have higher session quality dimension(\*)
  + Their total hits a lot higher
  + They connect less by mobile devices or tablets and use more desktop
  + They are mostly from US, rather than Europe, Asia or Africa
  + They are usually connecting from California and NewYork
  + They usually connect directly and less through google”
  + They decided to delete columns which have more than 50% missing value.
  + They found total.newVisits column not informative.
  + They found 24 columns with constant value.
* Now this is the feature engineering part. Here first they handled the date and time column by disintegrating them to day, hour and months.
* Then they converted the categorical columns to numerical type.
* Now comes the training part. They just demonstrated the LGBM Regression algorithm with hyperparameters num\_leaves=31, learning\_rate=0.03, n\_estimators=100, subsample=0.9, colsample\_bytree=0.9 and random\_state=34.

Then they printed the feature importance, which were as follow

| feature | importance |  |
| --- | --- | --- |
| 22 | rec\_dayofmonth | 2184.051979 |
| 2 | hits | 1370.217010 |
| 0 | channelGrouping | 1077.901001 |
| 20 | rec\_dayofweek | 566.035995 |
| 5 | device.operatingSystem | 354.673996 |
| 16 | totals.pageviews | 265.623993 |
| 13 | geoNetwork.city | 192.095993 |
| 9 | geoNetwork.subContinent | 173.393005 |
| 14 | geoNetwork.networkDomain | 172.865005 |
| 21 | rec\_hours | 135.039993 |



Link of the kernel :- <https://www.kaggle.com/oceands/basics-of-google-analytics>

**3. Gstore Customer Revenue Report**

This notebook is divided into 5 segments :-

1. **Problem Statement**
2. **Dataset Understanding**
3. **Exploration**
4. **Visitor Profile**
5. **Baseline Model**

* According to this notebook, the aim is to analyze a Gstore Customer dataset to predict revenue per customer.
* Insights from EDA :-
  + They found the dataset had 903653 rows and 55 raw columns.
  + There are 14 columns with missing values.

Device Attributes :-

* Number of visitors on the desktop is 3 times the number of visitors on mobile and 62 times the number of visitors on tablets. But the mean of revenue generated is almost the same by mobile and tablet.
* Number of chrome users is approx 17 times the number of firefox users but revenue generated by firefox is 4 times chrome users. Number of Windows users is 13 times the number of Chrome OS users but Chrome OS users generated more than any other OS users.

GeoNetwork Atribute :-

* The African population visited the site least but generated the revenue most.

Traffic Attribute :-

* The source of the maximum traffic campaign is Data share Promo.
* The most traffic source medium is Organic and Referral.

Visits by day, month and day attribute :-

* The most busy day is 28th November 2016.
* The most revenue was generated on 5th April 2017 which was 848.456 Million.
* The most busy month was November.
* The most busy days were Tuesday and Wednesday.
* April generated the most revenue.
* On weekdays, Tuesday and Wednesday generated the most revenues.

Visit Number Frequency :-

* Most of the visitors never came again.
* Only 9996 visitors made the transaction.

Preprocessing :-

* Dropped the columns with constant values.
* Dropped Ids and other non relevant columns.

Then they tried LGBM algorithm with following parameters and values :-

* "objective" : "regression",
* "metric" : "rmse",
* "num\_leaves" : 40,
* "learning\_rate" : 0.005,
* "bagging\_fraction" : 0.6,
* "feature\_fraction" : 0.6,
* "bagging\_frequency" : 6,
* "bagging\_seed" : 42,
* "verbosity" : -1,
* "seed": 42

The model stopped after 4000 rounds.

At 3998, train\_rmse=1.505, valid\_rmse=1.66372.

They found feature importance in following order :-

feature split gain

14 totals.pageviews 18837 34.535168

13 totals.hits 19867 18.466582

1 visitNumber 12029 7.681866

12 geoNetwork.subContinent 2683 5.119800

26 WoY 15872 4.132430

8 geoNetwork.country 3425 3.504271

24 trafficSource.source 4441 3.066060

9 geoNetwork.metro 6627 2.664815

7 geoNetwork.continent 1119 2.655823

30 visit\_hour 11519 2.633657

Then they used CatBoost Regressor with following parameters, which are as follow :-

iterations=1000,

learning\_rate=0.05,

depth=10,

eval\_metric='RMSE',

random\_seed = 42,

bagging\_temperature = 0.2,

od\_type='Iter',

metric\_period = 50,

od\_wait=20

This model ran for 300 rounds.

Val\_rmse = 1.61188, train\_rmse = 1.52371

Link for the kernel :- <https://www.kaggle.com/yatnam/gstore-customer-revenue-report>

**4. Simple Exploration + Baseline - GA Customer Revenue**

* The objective of this kernel is to explore the gstore dataset and then make inference with it. Then to get started they built a baseline model which was LGBM.
* They explained the dataset and initial features like others. Then they inferred the following information :-
  + They inferred that the ratio of revenue generating customers with no revenue is in the ratio of 1.3% .
  + In the training set out of 903653, 714167 are unique visitors. Number of common visitors in the train and test set is 7679.
  + There were 19 columns with constant values.
  + The distribution plot of both number of visitors and number of revenue generators for the device browser attribute.
  + Desktop users had a higher counter of non-zero revenue generator, than mobile device users.
  + Windows had more users than macintosh while macintosh had more number of non zero revenue generator.
  + Chrome OS also had a higher % of non zero revenue generator.
  + Among the mobile OS, ios had most stake for non zero revenue generator.
  + They has data from 1st August 2016 to 31st july 2017 in their training dataset.
  + There was a hike in visitors count in Nov 2016 but the revenue didn’t increase during that time.
  + America had the highest number of visitors and highest number of revenue generators.
  + Asia and Europe had good numbers of visitors but not the revenue generators.
  + Youtube was the primal source of traffic for visitors but not for revenue generator.
  + Then googleplex had the highest ratio of revenue generator and visitors.
  + “Coun plot shows decreasing nature i.e., we have a very high total count for loss number of hits and page views for visitors transaction and the overall count decreases when the number of hits per visitors transaction increases.”
* After this whole EDA, They did some preprocessing :-
  + They removed campaignCode and sessionId columns.
  + They imputed 0 for missing target values.
  + They encoded the categorical variables and converted the numerical variables to float type.
* After this preprocessing, they applied the ML algorithms.
  + They applied LGBM with following parameters :-
    - "objective" : "regression",
    - "metric" : "rmse",
    - “num\_leaves" : 30,
    - “min\_child\_samples" : 100,
    - "learning\_rate" : 0.1,
    - "bagging\_fraction" : 0.7,
    - "feature\_fraction" : 0.5,
    - "bagging\_frequency" : 5,
    - "bagging\_seed" : 2018,
    - "verbosity" : -1

After 254 rounds they got train rmse: 1.69087 and validation rmse was 1.70.

# 

# **First Cut Approach**

\*\*\* Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** \*\*\*

\*\*\* When you are doing the basic EDA and building the First Cut Approach you should not refer any blogs or papers \*\*\*

1. Since, we have 57 columns after flattening the 4 json columns and removing unnecessary columns, we will first drop unwanted and unnecessary columns. We will follow the following process :-
   1. First, we will print the frequency of each category (for categorical variable) from each column. Then we can drop the columns that have just one category.
   2. Then, we can observe for null values. We will first fill those nan values with 0 (for numerical features), after that we will fill nan values with unique values (i.e., for categorical features).
   3. After that we will do bivariate analysis by plotting column\_i vs target variable. Through the plot we will try to figure out “if the nan values have any significance for the target variable”.

If the variables have significance then, we will keep it otherwise drop it.

Now, we will prepare the target variable. Since, we have a classification problem, but our target variable contains continuous data. So, will replace nan with 0 and nonzero values with 1.

Now, we have a complete dataset. Thus we can split it into train and test.

**Models** :-

We have plenty of algorithms to apply for classification. Some of the powerful algo we will try are XGBoost classifier (lightGBM), Decision tree classifier and Random Forest.

After the classification we will work on a regression problem.

**Notes when you build your final notebook**:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument “X” (a single data points i.e 1\*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
   1. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
   2. so in your final notebook, you need to pass only those two values
   3. def final(X):

preprocess data i.e data cleaning, filling missing values etc

compute features based on this X

use pre trained model

return predicted outputs

final([time, location])

* 1. in the instructions, we have mentioned two functions one with original values and one without it
  2. final([time, location]) # in this function you need to return the predictions, no need to compute the metric
  3. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y\_predict)

1. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data
2. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
3. Check this live session: <https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models>