**Table of Contents**

[1. Business Problem 2](#__RefHeading___Toc1438_4083667713)

[1.1 Introduction 2](#__RefHeading___Toc1440_4083667713)

[1.2 Problem Statement 2](#__RefHeading___Toc1442_4083667713)

[1.3. Business Constraints 2](#__RefHeading___Toc1444_4083667713)

[1.4. Benefits of Modelling, Customer Metrics 2](#__RefHeading___Toc1446_4083667713)

[2 Type of Machine Leaning Problem 3](#__RefHeading___Toc1448_4083667713)

[3 Exploratory Data Analysis 3](#__RefHeading___Toc1450_4083667713)

[3.1 Data Fields 3](#__RefHeading___Toc1452_4083667713)

[3.1 Load Data and Format fields 3](#__RefHeading___Toc1454_4083667713)

[3.2 Analyse Data 3](#__RefHeading___Toc1456_4083667713)

[3.3 Check Feature Values 4](#__RefHeading___Toc1458_4083667713)

[3.3 Null Processing 4](#__RefHeading___Toc1460_4083667713)

[3.4 Univariate Analysis for Numeric fields 4](#__RefHeading___Toc1462_4083667713)

[3.4.1 Analysis of target variable: totals.totalTransactionRevenue 4](#__RefHeading___Toc1464_4083667713)

[3.6 Bivariate analysis 5](#__RefHeading___Toc1466_4083667713)

[3.7 Time Series Analysis 5](#__RefHeading___Toc1468_4083667713)

[4 Feature Engineering 5](#__RefHeading___Toc1470_4083667713)

[4.2 Encode Categorical features 6](#__RefHeading___Toc1472_4083667713)

[5 prepare data for Time Series 6](#__RefHeading___Toc1474_4083667713)

[5.1 Find Feature Importance 6](#__RefHeading___Toc1476_4083667713)

[6 Models 6](#__RefHeading___Toc1478_4083667713)

[6.0 Data Modelling 6](#__RefHeading___Toc1480_4083667713)

[Hyperparameter Optimization 7](#__RefHeading___Toc1484_4083667713)

[LightGBM Model 8](#__RefHeading___Toc1540_4083667713)

[6.6 Model Scores 8](#__RefHeading___Toc1486_4083667713)

[7.0 Model Analysis 9](#__RefHeading___Toc1510_4083667713)

[7.0.0 Objective 9](#__RefHeading___Toc1512_4083667713)

[7.1.1 LBG classification model 9](#__RefHeading___Toc1514_4083667713)

[Conclusion of Data Analysis: 9](#__RefHeading___Toc1522_4083667713)

[7.3 LBG Regression model 10](#__RefHeading___Toc1524_4083667713)

[7.3.2 LBG Regression Data Analysis: Remove outliers 10](#__RefHeading___Toc1526_4083667713)

[7.4 Conclusion 10](#__RefHeading___Toc1528_4083667713)

[8 Model Deployment API 10](#__RefHeading___Toc1530_4083667713)

[Objective and Function Description 10](#__RefHeading___Toc1532_4083667713)

[process\_data(in\_point) 10](#__RefHeading___Toc1534_4083667713)

[regr\_metric(test\_sampl,y\_pred) 11](#__RefHeading___Toc1536_4083667713)

[9 Model Deployment on Cloud 11](#__RefHeading___Toc1538_4083667713)

Google Analytics Customer Revenue Prediction

This project is an attempt to understand the end-to-end implementation of Data Science project from project definition, EDA,preprocessing,featurisation,modelling and deployment. Performing these activities will help gain practical exposure of implenting a data science project.

# 1. Business Problem

## 1.1 Introduction

Online marketing is a billion-dollar industry. Companies spend a lot of money by targeting users who visit their website once, to encourage them to buy products from them. But major part of the revenue comes from only a small percentage of users. Instead of targeting everyone who visits the website, marketing budget can better be utilized If they target only those users who are most likely to purchase a product in the future

## 1.2 Problem Statement

Google has provided Merchandise customer dataset and no.of transactions per customer.Using this data we need to predict future revenue created by customers. If we build a predictive model using G-store data set to predict revenue per customer. This prediction can be used to identify high revenue customers.This helps in better use of marketing budget.

## 1.3. Business Constraints

1. Customers producing revenue is sparse. Only a small percentage of customers produce revenue.
2. Customer visits follow a time pattern (time dependent).
3. No strict latency concerns.
4. Interpretability is useful for business analysis.

## 1.4. Benefits of Modelling, Customer Metrics

We use a data driven approach to identify features which help in better prediction. The marketing team can use this information to more effectively utilise the advertising budget.

After implementing the predictive model.following metrics can be used to measure benefits

1. **Improve Customer Experience**. This can be measured by percentage increase customer visits to store and repeat visits.  
   visits = No of unique fullVisitorId visiting the store(in time step)  
   repeat visit = No of fullVisitorId visiting storeafter fixed time.
2. Better use of marketing budget on high value customers. This can be measured by **average revenue per customer.**  
   average revenue = sum(totalRevenue)/No of fullVisitorId having transaction > 0

# **2 Type of Machine Leaning Problem**

This is a Time Series regression problem. We are predicting the natural log of sum of all transactions per user to predict total revenue per customer.

# **3 Exploratory Data Analysis**

In EDA we find relationship,correlations, feature distributions, within the data. Data quality,missing values, anamolies and outliers is also checked.

In this project following tasks were done for EDA

## **3.1 Data Fields**

**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** - The specifications for the device used to access the Store.  
**geoNetwork** - This section contains information about the geography of the user.  
**sessionId** - A unique identifier for this visit to the store.  
**socialEngagementType** - Engagement type, either "Socially Engaged" or "Not Socially Engaged".  
**totals** - This section contains aggregate values across the session.  
**trafficSource** - This section contains information about the Traffic Source from which the session origina ted.  
**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, we should use a combination of fullVisitorId and visitId.  
**visitNumber** - The session number for this user. For the first session,this is set to 1.  
**visitStartTime** - The timestamp (expressed as POSIX time)

Each row in dataset represents a visit to store.We are predicting log of total revenue per user. So not all rows in test.csv corresponds to a row in submisssion file.But all unique **fullVisitorIds** corresponds to a row in submission.

## **3.1 Load Data and Format fields**

Here we load data from csv files. The fields device , geoNetwork, totals , trafficSource are JSON fields. JSON format is used to store simple data structures into a file and load them into programs.

Here we deserialize JSON and convert it into flat fields using functions from json library.

Since the data is quite big we need a VM with sufficent memory and disk space to store the data.

So selection of proper hardware and software is vital for the success of the project. The VM used here is google Compute engine with of type e2-highmen-4 ( 4 vCPU, 32 GB RAM) and 500 GB disk space with debain10 and tf 2.0.

## **3.2 Analyse Data**

Data analysis is part of EDA where we try to get overview of data.From Data exploration we understand the characteristics of data like data size,data types. We compute summary statistics of data like mean variance, quantiles

## 3.3 Check Feature Values

For Numeric features we get summary statistics for each column using **describe()** function.This function displays the min.max. Quantiles,mean and variance.For some columns the std =0 incicating that these columns have constant values which are not useful in modelling. So these columns can be removed.

For Categorical features we check the frequencies of categorical features using **describe(include=['O'])** command. This command shws that for some columns we have some columns with unique=1.( single value or Constants).Constants are not useful in modelling and these features can be removed.

Thus using EDA features can be pruned and transformed to improve models significantly.

## 3.3 Null Processing

EDA also involves performing data quality analysis to detemine missing values in the dataset. For Numerical Features we check count of missing values. Features having large percentage of missing values are not useful for modelling and can be dropped. For other Numerical Features having small percentage of nulls, we replace nulls with numeric features.

For Categorical Features nulls are replaced with mode (most frequent value) of the feature.

## 3.4 Univariate Analysis for Numeric fields

We use univariate analysis to identify if data contains outliers.It also helps to identify features which need data transformation or feature engineering. Using data visualization, we can see distribution of data, identiy skewness in data and identify outliers.

### **3.4.1 Analysis of target variable: totals.totalTransactionRevenue**

The target value is continuous in case of regression problems. Regression is a statistical for estimating relationship between a dependent variable and one or more independent variables. For regression model we use peformance metrics

* Mean Absolute percentage error(MAPE):To measure model performance we can use MAPE.to know on average how good is the model with predictions.
* Mean Squared error(MSE): shows how well the forecasting model performs with outliers. To check if there is not much of a error margin between predicted and actual value

From parametric distribution of target variable using distplot() ,we observe

1. field deviates from the normal distribution.
2. Has appreciable positive skewness.
3. Show peakedness.
4. Target field follows log normal distribution

The Q-Q plot of target variable with log normal gives a 45 degee line which shows that the target has a log normal distribution. Logarithmic transformation is a technique to convert skewed distribution to a normal distribution.

From the plot it is observed that in field "totals.totalTransactionRevenue" 99% of values is less than 16990000.0. and 99.9 of values is less than 300570000.0. There are roughly 1700 values with greater than 300570000.0

From univariate analysis we conclude that for maximum benefit one needs to focus on top 0.01 percentile for high value segment and top 10 percetile for top segment.   
For categorical features the focus should be top few values.

It is observed that data is sparse since most points do not contibute.This aspect has to be handled in modelling the data.

## 3.6 Bivariate analysis

Using **.corr()** method we can compute Pearson correlation between every variable and the target. Using correlation analysis we can get collinear relation between independent variables. Since linear regression assumes no collinearity among independent features, features having high correlation can be dropped since they dont add any value.

## 3.7 Time Series Analysis

Obervations recorded at regular time intervals are termed as Time series. The frequency can be houly,daily,weekly,monthly,annual.

By creating features from time variable, machinelearning algorithms can be used for time series problems.Machine Learning for time series is a powerful technique to identify hidden complexities compared to classical methods.

Analysis of time series data over time is done by grouping data using frequency.

Time series consists of following components

1. Trend: observed when there is increase or decrease in slope.
2. Seasonality: repeated patterns observed at regular intervals due to seasonal factors.
3. Cyclic pattern does not follow any calendar frequency.Unlike seasonality they are affected by business and socio-economic factors.

Stationary time series properties are not time dependent.So statistical properties like mean, variance

and autocorrelation are constant over time.It is east to forecast for stationary series.So using tranform to make series stationary is important.

Autocorrelation is correraltion of series with its lag. ACF can be used for prediction using the lags.

Moving average is the average of rolling window of selected width.Choosing a good window width can nullify seasonality. An expanding window contains prior values also. For a time series problem validation data has to be choosen such that data order is preserved.

From time series plot we conclude that periods showing the peaks in time series have to utilised for more gains to the store.

# 4 Feature Engineering

According to Andrew Ng "applied machinelearning is basically feature engineering”.

Feature engineering consists of

* Feature Construction: is adding new features to exiting data
* Feature Selection: is selection of most important features

For time series problems we use feature constrcution from time data so that the model can identify hidden time patterns.

The features extracted are

• Window statistics:

• Max/min value of series in a window

• Average/median value in a window

• Window variance etc.

To generate use level features we group by field fullVisitorId.

Date and time features:

• Minute of an hour, hour of a day, day of the week, and so on

• Is this day a holiday? Maybe there is a special event? Represent that as a boolean

feature

**It is important that to note that any tranformation that is done for training data, the same transformation is also required for test data.**

## 4.2 Encode Categorical features

**Label encoding** assigns an integer to each unique category in a categorical variable.No new columns are created. When Cardinality of Features is high label encoding is used. The disadvantage of Label encoding is it gives arbitrary order to the variables.

In **one hot encoding** a new column is created for each unique value.The disadvantage is number of dimensions increases when cardinality is high.

# 5 Prepare data for Time Series

The test dataset provided has dates from 1-may-2018 to 15-oct-2018. The time window is 168 days. This time window can be used to create the train datasets for time series.

The prediction time frame is 1-dec-2018 to 31-jan-2019. The difference between the train and prediction dates is 46 days ( 15-oct-2018,1-dec-2018 ). This gap is maintained between train and test for creation of test dataset for time series. The prediction time window is 62 days. This window is used for test dataset.  
Customers present in both train and test are identified as common customers.

Using above criterion, we create 4 sets of data to train the prediction model.

## 5.1 Find Feature Importance

Feature importance is the importance or contribution for the supervised task ( classification or regression).Using this the effectiveness of feature engineering can be obtained.The feature importance graph shows the feature importance for around 140 variables in the input dataset.

# 6 Models

## 6.0 Data Modelling

We use two stage modeling approach and Pose Problem as “classification and regression”.  
1- predict probability of returning of customer using all train set.The idea is to classify non-zero transactions first and use that for better results for revenues.data imbalance can be handled using under-sampling/ oversampling.  
2- predict amount of transactions for those customers who returned.So, in regression task train was filtered only by customers who had a session in 62- day window.

This competition is a regression problem about predicting customer revenue. From EDA we observe that training data has very few non zero transactions. So it is a good idea to first run a classifier to identify non-zero customers.The classifier predicts the purchase probability.Then we use regression model trained only on the non-zero customers for better prediction.

The approach here is to try different combinations of the models and select the combination with the best prediction for implementation.

Logistic + Linear regression

First model used is logistic classification and Linear regression.Logistics regression tries to find linear hyperplane to separate the two classes.

SGDClassifier, SGDRegressor model

Stochastic Gradient Descent (SGD) classifier basically implements a plain SGD learning routine. Using loss= ’ĺog’ gives logistic regression.

SGDRegressor uses loss='squared\_loss’

DT-Classifier, Support Vector Regressor

The preformance of SVCClassifier was not good since training time is n^2. So for large datasets the performance is not good.So DT Classifier was used instead.

Gradient Descent (SGD) classifier basically implements a plain SGD learning routine. Using loss= ’ĺog’ gives logistic regression.

SGDRegressor uses loss='squared\_loss’

**Linear classifiers try to find a hyperplane to a hyperplane to separate the two classes. When data is not linearly separable the performance of linear classifier is not good.**

DecisionTree model

Decision Trees are interpretable and easily capture non-linear patterns and non-parametric. Decision Trees splits data based on best differentiator which minimises cost metric. The metric for clasification is entropy or the gini index. For regression it is mean squared error. Decision tree structure consists of root,branch and leaf nodes.

Random Forest Model

Random Forest is a collection of decorrelated decision trees. Random forest is a bagging techique.This is an ensemble of learners to create model of low variance. Random Forest ovecomes the overfit tendency of decision trees.

### **Hyperparameter Optimization**

Hyperparameters are algorithm settings that can be adjusted to optimise performance. The techniques used for optimising are RandomizedSearchCV and GridSearchCV. RandomizedSearchCV is used when there are large number of hyperparameters. For tree based models hyperparameters are grouped into 4 categories.

1.Parameters that affect the structure and learning of decision trees

2.Parameters that affect the training speed

3.Parameters for better accuracy

4.Parameters to combat overfitting

The hyperparameters for RF model are

Criterion: tells how good is the splitting of a node of a decision tree.

max\_depth: number of levels in each tree

max\_features: number of features considered, at the most, while splitting a node.

min\_samples\_leaf: number of samples that a leaf can store, at the least.

min\_samples\_split: minimum number of samples in a node for splitting of the node.

n\_estimators: number of trees in ensemble

bootstrap : method for sampling data points (with or without replacement)

### LightGBM Model

Gradient boosting uses weak learners(decision trees). The trees are built sequentially.

- The first tree learns to fit the target variable

- The second tree learns to fit the residual error between prediction of first learner and ground truth.

- The next tree learns using residual of previous step and so on.

The trees are trained by propagating the gradients of errors. LBGM trains fast and uses less memory. Unlike XGBoost LBGM uses leaf-wise growth. This increases growth and reduces error. Leaf-wise algorithm can reduce more loss than a level-wise algorithm.

The Control parameters of LBGM are

max\_depth : maximum depth of tree. Handles model overfitting.

min\_data\_in\_leaf : minimum number of records a leaf. Handles overfitting.

num\_leaves : Maximum tree leaves for base learners. higher value results in deeper trees.

min\_child\_samples : minimum number of data points in child (leaf) node. very important parameter to prevent overfitting.

feature\_fraction: select % of parameters randomly used in each iteration for building trees.

• bagging\_fraction : fraction of data used for each iteration. used to speed up training and avoid overfitting.

• early\_stopping\_round : speed up analysis. Model will stop training if one metric of one validation data doesn’t improve in last early\_stopping\_round rounds. This reduces excessive iterations.

• lambda : lambda specifies regularization. Typical value ranges from 0 to 1.

• min\_gain\_to\_split : minimum gain to make a split. controls number of useful splits in tree.

• boosting : defines the type of algorithm you want to run, default=gdbt.

• gbdt : traditional Gradient Boosting Decision Tree

• rf : random forest

• dart : Dropouts meet Multiple Additive Regression Trees

• goss : Gradient-based One-Side Sampling

Learning\_rate (shrinkage): slow learning turn requiring more trees to be added to the ensemble. gives regularisation effect. Reduces influence of individual tree and improves model

## 6.6 Model Scores

|  |  |  |  |
| --- | --- | --- | --- |
| **Model type** | **mean\_square error** | **mean\_absolute error** | **kaggle** |
| Logistic regression,  Linear regression | 9.24998 | 8.891554 | 9.44293 |
| SGDClassifier,  SGDRegressor | 3.8762e+25 | 9.7391e+23 | 1114008574477370000000000 |
| DT-Classifier,  Support Vector Regression | 2.11168 | 0.29602 | 0.8896 |
| DecisionTreeClassifier,  DecisionTreeRegressor | 2.1249005 | 0.25609 | 0.88771 |
| RandomForestRegressor, RandomForestClassifier | 2.110571 | 0.25809 | 0.88726 |
| LGBMClassifier,  LGBMRegressor | 2.1098 | 0.263 | 0.88457 (21st position) 0.88437 (17th position) 0.88397 (14th position) |

# 7.0 Model Analysis

## 7.0.0 Objective

The objective of Post Training analysis is to analyse input data of models to improve performance. This will ensure that data is of good quality and improve model performance. This approach is called **Data Centric approach**.

Source : <https://www.deeplearning.ai/wp-content/uploads/2021/06/MLOps-From-Model-centric-to-Data-centric-AI.pdf>   
Credit : Andrew Ng

## **7.1.1 LBG classification model**

The Classifier Feature importance shows the important features are

* year\_max
* date\_max
* TrafficSource\_ReferralPath\_max
* channelGrouping\_OS\_max
* year\_min
* date\_diff

We analyse these features in more detail to understand model performance. For the classifier to be effective,the feature plots should not overlap.

### Conclusion of Data Analysis:

The Feature Plots shows that classifier features have overlap hence the model in not effective in classifying data. To overcome this new features have to designed without overlap so model can better classify the data.

The Feature importance graph shows, temporal features like year\_max, date\_max are important for classifier. Also the number of features in the graph are also less. So more features can be included

to separate data into classes, five different querys were tried on input data.These queries to separate data into classes are not effective due to overlap of data. To make model more effetive more data can be collected. Such that this data helps with classification

## **7.3 LBG Regression model**

### **7.3.2 LBG Regression Data Analysis: Remove outliers**

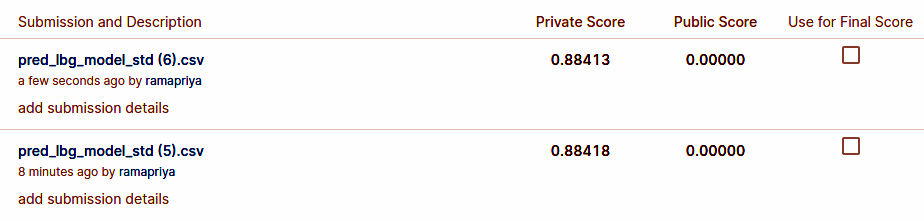
we can use standard deviation as a cut-off for identifying outliers. From Gaussian distribution remove values are more than a fixed Standard Deviations from the Mean.Values outside this can be considered outliers.

By selecting range of 6 sigma, lower cutoff value is less than min value for difference.This means no data points are removed.

## **7.4 Conclusion**

Using cutoff of 6 sigma for outlier the kaggle score is 0.88413.  
Using cutoff of 5 sigma for outlier the kaggle score is 0.88418.  
 Since removal of data points reduces the score(using 5 sigma).We can conclude there are no outliers in data for regression model. Using 6 sigma no datapoints are removed since lower range is less than minimum value.

Below is screenhot of kaggle scores (16th position in leaderboard)



# **8 Model Deployment API**

The notebook ¨Final.ipynb” contains the deployment API. A sample test point is extacted from the test dataset and used as input for deployment API.

## **Objective and Function Description**

The objective here is to develop deployment API for the best performing ML model. The models developed earlier are saved using joblib python module. The functions are

* process\_data
* regr\_metric

## process\_data(in\_point)

**input** : csv file with sample data point(s).  
**output** : predictions for the input data.  
**processing**: This function takes the input data sample and produces prediction for the input data.The input data is pre-processed to handle nulls.Then helper functions perform feature engineering to get additional features. These features undergo feature transformation. The transformed features are used to compute the final prediction.  
**display**: The functions has display for process time

## regr\_metric(test\_sampl,y\_pred)

**input** : input data ,model prediction.  
**output** : model metrics mean\_squared\_error and mean\_absolute\_error.  
**processing**: This function takes input data sample and model prediction for this data.Using these two inputs this function computes the metrics for the regression model like mean\_squared\_error and mean\_absolute\_error.  
**display**: mean\_squared\_error , mean\_absolute\_error

# **9 Model Deployment on Cloud**

The next step is to deploy the model on Cloud. The cloud platform used here is GCP (Google Cloud Platform) . The Flask gramework is used to used to develop the web API for the model.

The Google AppEngine service has to be enabled for model deployment. For deployment of model on App Engine, following modules are required

1) main.py: containing the flask web API

2) index.html: This is the html form containing the API for accepting the file input. The html should be placed in /templates folder.

3) app.yaml : containing the modules to be intalled for the project.

The process of deploying the model is as follows

- Upload code to app engine

- place html in /templates folder

- open cloud shell and run command “gloud init” to set the project configuration

- run command”gloud app deploy”to deploy the project code.

- go to project url generated. The home page has an option to upload the input file

- upload the input file and press submit.

- The model process input and displays the prediction in new page.

The model deployment video is available in url <https://youtu.be/jgBiqw5Q-B8>

# **9 References**

[http://www.AppliedAICourse.com](http://www.AppliedAICourse.com/)

[http://www.kaggle.com](http://www.AppliedAICourse.com/)

<https://www.kaggle.com/c/ga-customer-revenue-prediction/discussion/82614>

linkedin : [www.linkedin.com/in/ramapriyakp](http://www.linkedin.com/in/ramapriyakp)