In [1]:

```
!pip install lightgbm
```

```
Collecting lightgbm
  Downloading lightgbm-3.1.0-py2.py3-none-win_amd64.whl (751 kB)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\rudra\anaconda3\lib\site-packages
(from lightgbm) (0.22.1)
Requirement already satisfied: numpy in c:\users\rudra\anaconda3\lib\site-packages (from lightgbm)
(1.18.1)
Requirement already satisfied: scipy in c:\users\rudra\anaconda3\lib\site-packages (from lightgbm)
(1.4.1)
Requirement already satisfied: joblib>=0.11 in c:\users\rudra\anaconda3\lib\site-packages (from
scikit-learn!=0.22.0->lightgbm) (0.14.1)
Installing collected packages: lightgbm
Successfully installed lightgbm-3.1.0
```

In [3]:

```
import pandas as pd
import lightgbm as lgb
import datetime
```

In [17]:

```
store_df = pd.read_csv('preprocessed_gstoredata.csv',low_memory = False)
```

In [18]:

```
store_df.head()
```

Out[18]:

	channelGrouping	date	fullVisitorId	visitId	visitNumber	visitStartTime	device.browser	device.operatingSystem	device.isMob
0	4	2016-09-02	80509	1472830385	1	1472830385	11	16	Fal
1	4	2016-09-02	269007	1472880147	1	1472880147	16	7	Fal
2	4	2016-09-02	277678	1472865386	1	1472865386	11	16	Fal
3	4	2016-09-02	339713	1472881213	1	1472881213	46	6	Fal
4	4	2016-09-02	194517	1472822600	2	1472822600	11	1	Tr

5 rows × 31 columns

In [19]:

```
store_df.shape
```

Out[19]:

```
(903653, 31)
```

In [20]:

```
#splitting dataset

train_df = store_df[pd.to_datetime(store_df['date']).dt.date < datetime.date(2017,4,1)]
eval_df = store_df[pd.to_datetime(store_df['date']).dt.date >= datetime.date(2017,4,1)]
```

In [21]:

```
train_df.shape
```

Out[21]:

```
(633210, 31)
```

In [22]:

```
eval_df.shape
```

Out[22]:

```
(270443, 31)
```

In [23]:

```
#target labels

train_y = train_df['totals.transactionRevenue'].astype(float).values
eval_y = eval_df['totals.transactionRevenue'].astype(float).values
```

In [25]:

```
#getting the features by dropping the unncecessary columns

train_X =
train_df.drop(['date','fullVisitorId','visitId','visitStartTime','totals.transactionRevenue'],axis
=1)
eval_X =
eval_df.drop(['date','fullVisitorId','visitId','visitStartTime','totals.transactionRevenue'],axis=
1)
```

In [27]:

```
def train_lightgbm(train_X, train_y, eval_X, eval_y):

    #initializing dataset

    lgtrain = lgb.Dataset(train_X, label = train_y)

    lgeval = lgb.Dataset(eval_X, label = eval_y)


    #setting the hyperparameters
    params = {
        "objective":"regression",
        "metric":"rmse",
        "num_leaves" : 30,
        "min_child_samples" : 100,
        "learning_rate" : 0.1,
        "bagging_fraction": 0.7,
        "feature_fraction" : 0.5,
        "bagging_seed" :2018,
        "verbosity" : -1
    }

    #training the model

    model = lgb.train(params, lgtrain, 1000, valid_sets = [lgeval], early_stopping_rounds = 100, ve
rbose_eval = 100)

    return model

model =  train_lightgbm(train_X, train_y, eval_X, eval_y)
```

```
Training until validation scores don't improve for 100 rounds
[100] valid_0's rmse: 6.48845e+07
[200] valid_0's rmse: 6.48533e+07
Early stopping, best iteration is:
[140] valid_0's rmse: 6.4822e+07
```

In [28]:

```
index_val = 0

actual_X_val = eval_X.reset_index(drop=True).iloc[index_val]

actual_y_val = eval_y[index_val]
```

In [29]:

```
actual_X_val
```

Out[29]:

```
channelGrouping                              2
visitNumber                                  1
device.browser                              11
device.operatingSystem                       1
device.isMobile                           True
device.deviceCategory                        1
geoNetwork.continent                         3
geoNetwork.subContinent                     21
geoNetwork.country                         204
geoNetwork.region                          375
geoNetwork.metro                            93
geoNetwork.city                            648
geoNetwork.networkDomain                 24932
totals.hits                                  1
totals.pageviews                             1
totals.bounces                               1
totals.newVisits                             1
trafficSource.campaign                       0
trafficSource.source                         0
trafficSource.medium                         0
trafficSource.keyword                       11
trafficSource.isTrueDirect                True
trafficSource.adwordsClickInfo.page          0
trafficSource.adwordsClickInfo.slot          0
trafficSource.adwordsClickInfo.adNetworkType 1
trafficSource.adwordsClickInfo.isVideoAd     1
Name: 0, dtype: object
```

In [30]:

```
actual_y_val
```

Out[30]:

```
0.0
```

In [ ]:

```

```

In [ ]:

```

```