Untitled6

October 23, 2018

In [1]:
```
import os
import json
import pandas as pd
import numpy as np
from pandas.io.json import json_normalize
from IPython.display import display
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
import gc
import lightgbm as lgb
import xgboost as xgb
import lightgbm
from sklearn.model_selection import GroupKFold
from sklearn import preprocessing
import datetime as datetime
from datetime import timedelta, date
from sklearn.model_selection import train_test_split, KFold
import warnings
warnings.simplefilter("ignore")
import time
import plotly.graph_objs as go
import plotly.tools as tools
import plotly.plotly as py
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
from matplotlib import pyplot
```

```
/Users/vivek/anaconda3/lib/python3.6/site-packages/lightgbm/__init__.py:45: FutureWarning: Sta
This means that in case of installing LightGBM from PyPI via the ``pip install lightgbm`` comma
Instead of that, you'll need to install the OpenMP library, which is required for running Light
You can install the OpenMP library by the following command: ``brew install libomp``.
  "You can install the OpenMP library by the following command: ``brew install libomp``.", Futu
```

Task 1 : Cleaning

Load our training and test data frames into main memory. FullVisitorId column converted to string format. 4 columns (device, geoNetwork, totals, trafficSource) are in JSON format. So we normalize those columns.

```
In [2]: def load_df(csv_path='train.csv', nrows=None):
            JSON_COLUMNS = ['device', 'geoNetwork', 'totals', 'trafficSource']

            df = pd.read_csv(csv_path,
                             converters={column: json.loads for column in JSON_COLUMNS},
                             dtype={'fullVisitorId': str},
                             nrows=nrows)

            for column in JSON_COLUMNS:
                column_as_df = json_normalize(df[column])
                column_as_df.columns = [f"{column}.{subcolumn}" for subcolumn in column_as_df.
                df = df.drop(column, axis=1).merge(column_as_df, right_index=True, left_index=
            print(f"Loaded {os.path.basename(csv_path)}. Shape: {df.shape}")
            return df

In [3]: train_df = load_df('train.csv',1_00_000)
        test_df = load_df("test.csv",1_00_000)

Loaded train.csv. Shape: (100000, 55)
Loaded test.csv. Shape: (100000, 53)


In [4]: train_df.describe()

Out[4]:                date       visitId    visitNumber  visitStartTime
        count  1.000000e+05  1.000000e+05  100000.000000    1.000000e+05
        mean   2.016536e+07  1.484710e+09       2.274210    1.484710e+09
        std    4.694358e+03  9.166865e+06       9.260941    9.166865e+06
        min    2.016081e+07  1.470896e+09       1.000000    1.470899e+09
        25%    2.016111e+07  1.478832e+09       1.000000    1.478832e+09
        50%    2.016121e+07  1.481667e+09       1.000000    1.481667e+09
        75%    2.017050e+07  1.493684e+09       1.000000    1.493684e+09
        max    2.017080e+07  1.501657e+09     389.000000    1.501657e+09

In [5]: train_df['totals.transactionRevenue'] = train_df['totals.transactionRevenue'].astype(fl
        train_df['totals.transactionRevenue'] = train_df['totals.transactionRevenue'].fillna(0)

In [6]: train_df.head()

Out[6]:   channelGrouping      date        fullVisitorId  \
        0  Organic Search  20160902  1131660440785968503
        1  Organic Search  20160902   377306020877927890
        2  Organic Search  20160902  3895546263509774583
        3  Organic Search  20160902  4763447161404445595
        4  Organic Search  20160902    27294437909732085
```

```
                      sessionId  socialEngagementType     visitId  \
0  1131660440785968503_1472830385  Not Socially Engaged  1472830385
1   377306020877927890_1472880147  Not Socially Engaged  1472880147
2  3895546263509774583_1472865386  Not Socially Engaged  1472865386
3  4763447161404445595_1472881213  Not Socially Engaged  1472881213
4    27294437909732085_1472822600  Not Socially Engaged  1472822600

   visitNumber  visitStartTime device.browser            device.browserSize  \
0            1      1472830385         Chrome  not available in demo dataset
1            1      1472880147        Firefox  not available in demo dataset
2            1      1472865386         Chrome  not available in demo dataset
3            1      1472881213     UC Browser  not available in demo dataset
4            2      1472822600         Chrome  not available in demo dataset

          ...          trafficSource.adwordsClickInfo.isVideoAd  \
0         ...                                               NaN
1         ...                                               NaN
2         ...                                               NaN
3         ...                                               NaN
4         ...                                               NaN

  trafficSource.adwordsClickInfo.page trafficSource.adwordsClickInfo.slot  \
0                                 NaN                                 NaN
1                                 NaN                                 NaN
2                                 NaN                                 NaN
3                                 NaN                                 NaN
4                                 NaN                                 NaN

   trafficSource.campaign trafficSource.campaignCode  \
0              (not set)                         NaN
1              (not set)                         NaN
2              (not set)                         NaN
3              (not set)                         NaN
4              (not set)                         NaN

  trafficSource.isTrueDirect trafficSource.keyword trafficSource.medium  \
0                        NaN       (not provided)              organic
1                        NaN       (not provided)              organic
2                        NaN       (not provided)              organic
3                        NaN      google + online              organic
4                       True       (not provided)              organic

  trafficSource.referralPath trafficSource.source
0                        NaN               google
1                        NaN               google
2                        NaN               google
3                        NaN               google
```

```
4                          NaN                google

[5 rows x 55 columns]
```

As we can see some columns are containing constant values and some are containing Nan. Lets first remove constant columns since they are of no use in our modelling.

```
In [7]: const_cols = [c for c in train_df.columns if train_df[c].nunique(dropna=False)==1 ]
        const_cols

Out[7]: ['socialEngagementType',
         'device.browserSize',
         'device.browserVersion',
         'device.flashVersion',
         'device.language',
         'device.mobileDeviceBranding',
         'device.mobileDeviceInfo',
         'device.mobileDeviceMarketingName',
         'device.mobileDeviceModel',
         'device.mobileInputSelector',
         'device.operatingSystemVersion',
         'device.screenColors',
         'device.screenResolution',
         'geoNetwork.cityId',
         'geoNetwork.latitude',
         'geoNetwork.longitude',
         'geoNetwork.networkLocation',
         'totals.visits',
         'trafficSource.adwordsClickInfo.criteriaParameters']

In [8]: len(const_cols)

Out[8]: 19
```

There are 19 columns which contain constant values. So remove them.

```
In [9]: train_df = train_df.drop(const_cols, axis=1)
        test_df = test_df.drop(const_cols, axis=1)

In [10]: train_df.head()

Out[10]:   channelGrouping      date        fullVisitorId  \
          0  Organic Search  20160902  1131660440785968503
          1  Organic Search  20160902   377306020877927890
          2  Organic Search  20160902  3895546263509774583
          3  Organic Search  20160902  4763447161404445595
          4  Organic Search  20160902    27294437909732085

                                sessionId     visitId  visitNumber  visitStartTime  \
```

```
0  1131660440785968503_1472830385  1472830385           1      1472830385
1   377306020877927890_1472880147  1472880147           1      1472880147
2  3895546263509774583_1472865386  1472865386           1      1472865386
3  4763447161404445595_1472881213  1472881213           1      1472881213
4    27294437909732085_1472822600  1472822600           2      1472822600

  device.browser device.deviceCategory  device.isMobile         ...          \
0         Chrome               desktop           False         ...
1        Firefox               desktop           False         ...
2         Chrome               desktop           False         ...
3     UC Browser               desktop           False         ...
4         Chrome                mobile            True         ...

  trafficSource.adwordsClickInfo.isVideoAd  \
0                                      NaN
1                                      NaN
2                                      NaN
3                                      NaN
4                                      NaN

  trafficSource.adwordsClickInfo.page trafficSource.adwordsClickInfo.slot  \
0                                 NaN                                  NaN
1                                 NaN                                  NaN
2                                 NaN                                  NaN
3                                 NaN                                  NaN
4                                 NaN                                  NaN

  trafficSource.campaign trafficSource.campaignCode  \
0             (not set)                        NaN
1             (not set)                        NaN
2             (not set)                        NaN
3             (not set)                        NaN
4             (not set)                        NaN

  trafficSource.isTrueDirect trafficSource.keyword trafficSource.medium  \
0                        NaN       (not provided)              organic
1                        NaN       (not provided)              organic
2                        NaN       (not provided)              organic
3                        NaN      google + online              organic
4                       True       (not provided)              organic

  trafficSource.referralPath trafficSource.source
0                        NaN               google
1                        NaN               google
2                        NaN               google
3                        NaN               google
4                        NaN               google
```

```
[5 rows x 36 columns]
```

Let's analyse which columns are in our training set and not in our test set.

```
In [11]: print("Variables in train but not in test : ", set(train_df.columns).difference(set(te

Variables in train but not in test :  {'totals.transactionRevenue', 'trafficSource.campaignCode
```

So there are 2 columns : campaign code and transactionRevenue which are in our training data and not in test data. TransactionRevenue is the target variable so we have to keep it. campaign code can be removed from our training dataset.

```
In [12]: train_df.drop('trafficSource.campaignCode', 1, inplace=True)
         train_df.drop('sessionId', 1, inplace=True)
         test_df.drop('sessionId', 1, inplace=True)
         orig_df = train_df.copy()
```

Some values are missing from our training and test data sets. Let's impute those values.

```
In [13]: cols_with_missing = [col for col in train_df.columns
                                        if train_df[col].isnull().any()]
         cols_with_missing

Out[13]: ['totals.bounces',
          'totals.newVisits',
          'totals.pageviews',
          'trafficSource.adContent',
          'trafficSource.adwordsClickInfo.adNetworkType',
          'trafficSource.adwordsClickInfo.gclId',
          'trafficSource.adwordsClickInfo.isVideoAd',
          'trafficSource.adwordsClickInfo.page',
          'trafficSource.adwordsClickInfo.slot',
          'trafficSource.isTrueDirect',
          'trafficSource.keyword',
          'trafficSource.referralPath']

In [14]: len(cols_with_missing)

Out[14]: 12

In [15]: def RunEncoder():
             cat_cols = ["channelGrouping", "device.browser",
                         "device.deviceCategory", "device.operatingSystem",
                         "geoNetwork.city", "geoNetwork.continent",
                         "geoNetwork.country", "geoNetwork.metro",
                         "geoNetwork.networkDomain", "geoNetwork.region",
                         "geoNetwork.subContinent", "trafficSource.adContent",
                         "trafficSource.adwordsClickInfo.adNetworkType",
                         "trafficSource.adwordsClickInfo.gclId",
```

```
                        "trafficSource.adwordsClickInfo.page",
                        "trafficSource.adwordsClickInfo.slot", "trafficSource.campaign",
                        "trafficSource.keyword", "trafficSource.medium",
                        "trafficSource.referralPath", "trafficSource.source",
                        'trafficSource.adwordsClickInfo.isVideoAd', 'trafficSource.isTrueDire
                        'visitId','totals.bounces','totals.newVisits','totals.pageviews']

                for col in cat_cols:
                    lbl = preprocessing.LabelEncoder()
                    lbl.fit(list(train_df[col].values.astype('str')) + list(test_df[col].values.as
                    train_df[col] = lbl.transform(list(train_df[col].values.astype('str')))
                    test_df[col] = lbl.transform(list(test_df[col].values.astype('str')))

                num_cols = ["totals.hits", "totals.pageviews", "visitNumber", "visitStartTime", '

                for col in num_cols:
                    train_df[col] = train_df[col].astype(float)
                    test_df[col] = test_df[col].astype(float)
```

In [16]:
```
RunEncoder()
```

In [17]:
```
train_df.head()
```

Out[17]:

	channelGrouping	date	fullVisitorId	visitId	visitNumber	\
0	4	20160902	1131660440785968503	13696	1.0	
1	4	20160902	377306020877927890	15217	1.0	
2	4	20160902	3895546263509774583	14899	1.0	
3	4	20160902	4763447161404445595	15230	1.0	
4	4	20160902	27294437909732085	13417	2.0	

	visitStartTime	device.browser	device.deviceCategory	device.isMobile	\
0	1.472830e+09	12	0	False	
1	1.472880e+09	17	0	False	
2	1.472865e+09	12	0	False	
3	1.472881e+09	40	0	False	
4	1.472823e+09	12	1	True	

	device.operatingSystem	...	\
0	17	...	
1	7	...	
2	17	...	
3	6	...	
4	1	...	

	trafficSource.adwordsClickInfo.gclId	\
0	8098	
1	8098	
2	8098	

```
3                                   8098
4                                   8098

   trafficSource.adwordsClickInfo.isVideoAd  \
0                                         1
1                                         1
2                                         1
3                                         1
4                                         1

   trafficSource.adwordsClickInfo.page  trafficSource.adwordsClickInfo.slot  \
0                                    6                                    3
1                                    6                                    3
2                                    6                                    3
3                                    6                                    3
4                                    6                                    3

   trafficSource.campaign  trafficSource.isTrueDirect  trafficSource.keyword  \
0                       3                           1                      5
1                       3                           1                      5
2                       3                           1                      5
3                       3                           1                    285
4                       3                           0                      5

   trafficSource.medium  trafficSource.referralPath  trafficSource.source
0                     5                        1386                    74
1                     5                        1386                    74
2                     5                        1386                    74
3                     5                        1386                    74
4                     5                        1386                    74

[5 rows x 34 columns]
```

Task 2 : HeatMap and Plots Generation

Let's now create a heatmap of all columns taking into consideration correlation matrix.

```
In [18]: def add_time_features(df):
             df['date'] = pd.to_datetime(df['date'], format='%Y%m%d', errors='ignore')
             df['year'] = df['date'].apply(lambda x: x.year)
             df['month'] = df['date'].apply(lambda x: x.month)
             df['day'] = df['date'].apply(lambda x: x.day)
             df['weekday'] = df['date'].apply(lambda x: x.weekday())
             df['visitStartTime_'] = pd.to_datetime(df['visitStartTime'],unit="s")
             df['visitStartTime_year'] = df['visitStartTime_'].apply(lambda x: x.year)
             df['visitStartTime_month'] = df['visitStartTime_'].apply(lambda x: x.month)
             df['visitStartTime_day'] = df['visitStartTime_'].apply(lambda x: x.day)
             df['visitStartTime_weekday'] = df['visitStartTime_'].apply(lambda x: x.weekday())
             return df
```

```
In [19]: corr_train = train_df.copy()
         train_df = add_time_features(train_df)
         test_df = add_time_features(test_df)

In [20]: train_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 43 columns):
channelGrouping                                 100000 non-null int64
date                                            100000 non-null datetime64[ns]
fullVisitorId                                   100000 non-null object
visitId                                         100000 non-null int64
visitNumber                                     100000 non-null float64
visitStartTime                                  100000 non-null float64
device.browser                                  100000 non-null int64
device.deviceCategory                           100000 non-null int64
device.isMobile                                 100000 non-null bool
device.operatingSystem                          100000 non-null int64
geoNetwork.city                                 100000 non-null int64
geoNetwork.continent                            100000 non-null int64
geoNetwork.country                              100000 non-null int64
geoNetwork.metro                                100000 non-null int64
geoNetwork.networkDomain                        100000 non-null int64
geoNetwork.region                               100000 non-null int64
geoNetwork.subContinent                         100000 non-null int64
totals.bounces                                  100000 non-null float64
totals.hits                                     100000 non-null float64
totals.newVisits                                100000 non-null float64
totals.pageviews                                100000 non-null float64
totals.transactionRevenue                       100000 non-null float64
trafficSource.adContent                         100000 non-null int64
trafficSource.adwordsClickInfo.adNetworkType    100000 non-null int64
trafficSource.adwordsClickInfo.gclId            100000 non-null int64
trafficSource.adwordsClickInfo.isVideoAd        100000 non-null int64
trafficSource.adwordsClickInfo.page             100000 non-null int64
trafficSource.adwordsClickInfo.slot             100000 non-null int64
trafficSource.campaign                          100000 non-null int64
trafficSource.isTrueDirect                      100000 non-null int64
trafficSource.keyword                           100000 non-null int64
trafficSource.medium                            100000 non-null int64
trafficSource.referralPath                      100000 non-null int64
trafficSource.source                            100000 non-null int64
year                                            100000 non-null int64
month                                           100000 non-null int64
day                                             100000 non-null int64
weekday                                         100000 non-null int64
visitStartTime_                                 100000 non-null datetime64[ns]
```

```
visitStartTime_year                              100000 non-null int64
visitStartTime_month                             100000 non-null int64
visitStartTime_day                               100000 non-null int64
visitStartTime_weekday                           100000 non-null int64
dtypes: bool(1), datetime64[ns](2), float64(7), int64(32), object(1)
memory usage: 32.1+ MB
```

In [21]:
```
corr_mat = corr_train.corr()
corr_mat
```

Out[21]:
```
                                              channelGrouping      date  \
channelGrouping                                      1.000000 -0.219457
date                                                -0.219457  1.000000
visitId                                             -0.214953  0.879864
visitNumber                                         -0.039646  0.007521
visitStartTime                                      -0.206453  0.884425
device.browser                                       0.108538 -0.075138
device.deviceCategory                               -0.219050  0.139765
device.isMobile                                     -0.238215  0.152053
device.operatingSystem                               0.036194 -0.006849
geoNetwork.city                                     -0.008405  0.063876
geoNetwork.continent                                 0.024573  0.006682
geoNetwork.country                                  -0.035648 -0.006708
geoNetwork.metro                                    -0.015806  0.081203
geoNetwork.networkDomain                             0.118330 -0.079355
geoNetwork.region                                    0.020495  0.043394
geoNetwork.subContinent                              0.068772 -0.042649
totals.bounces                                      -0.039894 -0.020614
totals.hits                                         -0.063704 -0.019232
totals.newVisits                                    -0.075733  0.031759
totals.pageviews                                    -0.070173 -0.002280
totals.transactionRevenue                           -0.004261 -0.001749
trafficSource.adContent                             -0.022756 -0.072255
trafficSource.adwordsClickInfo.adNetworkType        -0.034681 -0.032218
trafficSource.adwordsClickInfo.gclId                -0.029820 -0.014860
trafficSource.adwordsClickInfo.isVideoAd            -0.034683 -0.032185
trafficSource.adwordsClickInfo.page                 -0.034674 -0.032220
trafficSource.adwordsClickInfo.slot                 -0.034379 -0.031803
trafficSource.campaign                              -0.180634  0.011005
trafficSource.isTrueDirect                           0.459128 -0.048668
trafficSource.keyword                                0.277461 -0.182715
trafficSource.medium                                 0.820530 -0.098672
trafficSource.referralPath                          -0.576718  0.058212
trafficSource.source                                 0.932689 -0.259641

                                               visitId  visitNumber  \
channelGrouping                              -0.214953    -0.039646
```

date	0.879864	0.007521
visitId	1.000000	0.007441
visitNumber	0.007441	1.000000
visitStartTime	0.984184	0.004475
device.browser	-0.060166	-0.042754
device.deviceCategory	0.144204	-0.038334
device.isMobile	0.156521	-0.040382
device.operatingSystem	-0.013735	0.004439
geoNetwork.city	0.044134	0.003271
geoNetwork.continent	-0.012381	-0.065560
geoNetwork.country	0.007930	0.054393
geoNetwork.metro	0.067324	0.019227
geoNetwork.networkDomain	-0.091470	-0.047906
geoNetwork.region	0.019908	-0.002800
geoNetwork.subContinent	-0.041535	-0.027428
totals.bounces	-0.003890	0.021572
totals.hits	-0.027440	0.039546
totals.newVisits	0.035405	0.253634
totals.pageviews	0.009025	0.024393
totals.transactionRevenue	0.001337	0.029682
trafficSource.adContent	-0.077683	-0.024722
trafficSource.adwordsClickInfo.adNetworkType	-0.049645	0.002456
trafficSource.adwordsClickInfo.gclId	-0.026725	0.002463
trafficSource.adwordsClickInfo.isVideoAd	-0.049628	0.002460
trafficSource.adwordsClickInfo.page	-0.049735	0.002445
trafficSource.adwordsClickInfo.slot	-0.049206	0.002600
trafficSource.campaign	0.010320	-0.006404
trafficSource.isTrueDirect	-0.052487	-0.183824
trafficSource.keyword	-0.183006	0.003727
trafficSource.medium	-0.098922	-0.035906
trafficSource.referralPath	0.047618	-0.020893
trafficSource.source	-0.258301	-0.047765

	visitStartTime	device.browser	\
channelGrouping	-0.206453	0.108538	
date	0.884425	-0.075138	
visitId	0.984184	-0.060166	
visitNumber	0.004475	-0.042754	
visitStartTime	1.000000	-0.054519	
device.browser	-0.054519	1.000000	
device.deviceCategory	0.139118	0.285575	
device.isMobile	0.151480	0.272805	
device.operatingSystem	-0.014525	0.171948	
geoNetwork.city	0.039891	0.031814	
geoNetwork.continent	-0.003126	0.052541	
geoNetwork.country	0.003946	-0.012112	
geoNetwork.metro	0.058873	-0.009579	
geoNetwork.networkDomain	-0.083180	0.109305	

geoNetwork.region	0.020022	0.077593
geoNetwork.subContinent	-0.035483	0.036155
totals.bounces	-0.010754	-0.057074
totals.hits	-0.031346	-0.075660
totals.newVisits	0.028309	-0.131675
totals.pageviews	0.001718	-0.068470
totals.transactionRevenue	0.000379	-0.023413
trafficSource.adContent	-0.078221	-0.022528
trafficSource.adwordsClickInfo.adNetworkType	-0.047342	-0.010799
trafficSource.adwordsClickInfo.gclId	-0.024079	-0.029304
trafficSource.adwordsClickInfo.isVideoAd	-0.047321	-0.010854
trafficSource.adwordsClickInfo.page	-0.047407	-0.010937
trafficSource.adwordsClickInfo.slot	-0.047086	-0.011516
trafficSource.campaign	0.008574	-0.028068
trafficSource.isTrueDirect	-0.043364	0.115209
trafficSource.keyword	-0.180860	0.003693
trafficSource.medium	-0.090603	0.063643
trafficSource.referralPath	0.049100	0.116223
trafficSource.source	-0.249542	0.138060

	device.deviceCategory \
channelGrouping	-0.219050
date	0.139765
visitId	0.144204
visitNumber	-0.038334
visitStartTime	0.139118
device.browser	0.285575
device.deviceCategory	1.000000
device.isMobile	0.945925
device.operatingSystem	-0.084144
geoNetwork.city	0.070056
geoNetwork.continent	-0.002335
geoNetwork.country	0.004093
geoNetwork.metro	0.069419
geoNetwork.networkDomain	-0.006709
geoNetwork.region	0.082687
geoNetwork.subContinent	-0.025277
totals.bounces	-0.025440
totals.hits	-0.022319
totals.newVisits	-0.028986
totals.pageviews	-0.008993
totals.transactionRevenue	-0.020504
trafficSource.adContent	-0.090812
trafficSource.adwordsClickInfo.adNetworkType	-0.092311
trafficSource.adwordsClickInfo.gclId	-0.082645
trafficSource.adwordsClickInfo.isVideoAd	-0.092341
trafficSource.adwordsClickInfo.page	-0.092265
trafficSource.adwordsClickInfo.slot	-0.093945

```
trafficSource.campaign                                 0.013255
trafficSource.isTrueDirect                            -0.061888
trafficSource.keyword                                 -0.199060
trafficSource.medium                                  -0.160634
trafficSource.referralPath                             0.227207
trafficSource.source                                  -0.227627

                                                   device.isMobile  \
channelGrouping                                          -0.238215
date                                                      0.152053
visitId                                                   0.156521
visitNumber                                              -0.040382
visitStartTime                                            0.151480
device.browser                                            0.272805
device.deviceCategory                                     0.945925
device.isMobile                                           1.000000
device.operatingSystem                                   -0.134512
geoNetwork.city                                           0.055277
geoNetwork.continent                                     -0.013259
geoNetwork.country                                        0.003616
geoNetwork.metro                                          0.057533
geoNetwork.networkDomain                                 -0.009810
geoNetwork.region                                         0.068896
geoNetwork.subContinent                                  -0.022839
totals.bounces                                           -0.027312
totals.hits                                              -0.025560
totals.newVisits                                         -0.028801
totals.pageviews                                         -0.009040
totals.transactionRevenue                                -0.021612
trafficSource.adContent                                  -0.083468
trafficSource.adwordsClickInfo.adNetworkType             -0.087556
trafficSource.adwordsClickInfo.gclId                     -0.076245
trafficSource.adwordsClickInfo.isVideoAd                 -0.087597
trafficSource.adwordsClickInfo.page                      -0.087502
trafficSource.adwordsClickInfo.slot                      -0.088340
trafficSource.campaign                                    0.009048
trafficSource.isTrueDirect                               -0.073441
trafficSource.keyword                                    -0.206355
trafficSource.medium                                     -0.176099
trafficSource.referralPath                                0.240214
trafficSource.source                                     -0.246458

                                                   device.operatingSystem  \
channelGrouping                                                  0.036194
date                                                            -0.006849
visitId                                                         -0.013735
visitNumber                                                      0.004439
visitStartTime                                                  -0.014525
```

```
device.browser                                        0.171948
device.deviceCategory                                -0.084144
device.isMobile                                      -0.134512
device.operatingSystem                                1.000000
geoNetwork.city                                       0.094182
geoNetwork.continent                                  0.108971
geoNetwork.country                                   -0.066546
geoNetwork.metro                                      0.026843
geoNetwork.networkDomain                              0.112824
geoNetwork.region                                     0.143948
geoNetwork.subContinent                              -0.014218
totals.bounces                                       -0.069038
totals.hits                                          -0.054600
totals.newVisits                                     -0.088274
totals.pageviews                                     -0.067159
totals.transactionRevenue                            -0.019098
trafficSource.adContent                              -0.005112
trafficSource.adwordsClickInfo.adNetworkType          0.002262
trafficSource.adwordsClickInfo.gclId                 -0.014625
trafficSource.adwordsClickInfo.isVideoAd              0.002221
trafficSource.adwordsClickInfo.page                   0.002118
trafficSource.adwordsClickInfo.slot                   0.002284
trafficSource.campaign                                0.019343
trafficSource.isTrueDirect                            0.116705
trafficSource.keyword                                -0.096161
trafficSource.medium                                  0.067980
trafficSource.referralPath                            0.113594
trafficSource.source                                  0.037027

                                                  geoNetwork.city  \
channelGrouping                                        -0.008405
date                                                    0.063876
visitId                                                 0.044134
visitNumber                                             0.003271
visitStartTime                                          0.039891
device.browser                                          0.031814
device.deviceCategory                                   0.070056
device.isMobile                                         0.055277
device.operatingSystem                                  0.094182
geoNetwork.city                                         1.000000
geoNetwork.continent                                    0.023754
geoNetwork.country                                     -0.094925
geoNetwork.metro                                        0.793767
geoNetwork.networkDomain                                0.060147
geoNetwork.region                                       0.695027
geoNetwork.subContinent                                -0.033893
totals.bounces                                         -0.013139
totals.hits                                            -0.002748
```

```
totals.newVisits                                    -0.033128
totals.pageviews                                    -0.005282
totals.transactionRevenue                           -0.006361
trafficSource.adContent                             -0.019704
trafficSource.adwordsClickInfo.adNetworkType        -0.032611
trafficSource.adwordsClickInfo.gclId                -0.028176
trafficSource.adwordsClickInfo.isVideoAd            -0.032532
trafficSource.adwordsClickInfo.page                 -0.032600
trafficSource.adwordsClickInfo.slot                 -0.032212
trafficSource.campaign                               0.010953
trafficSource.isTrueDirect                           0.068693
trafficSource.keyword                               -0.118140
trafficSource.medium                                 0.037206
trafficSource.referralPath                           0.093450
trafficSource.source                                -0.008439

                                                       ...          \
channelGrouping                                        ...
date                                                   ...
visitId                                                ...
visitNumber                                            ...
visitStartTime                                         ...
device.browser                                         ...
device.deviceCategory                                  ...
device.isMobile                                        ...
device.operatingSystem                                 ...
geoNetwork.city                                        ...
geoNetwork.continent                                   ...
geoNetwork.country                                     ...
geoNetwork.metro                                       ...
geoNetwork.networkDomain                               ...
geoNetwork.region                                      ...
geoNetwork.subContinent                                ...
totals.bounces                                         ...
totals.hits                                            ...
totals.newVisits                                       ...
totals.pageviews                                       ...
totals.transactionRevenue                              ...
trafficSource.adContent                                ...
trafficSource.adwordsClickInfo.adNetworkType           ...
trafficSource.adwordsClickInfo.gclId                   ...
trafficSource.adwordsClickInfo.isVideoAd               ...
trafficSource.adwordsClickInfo.page                    ...
trafficSource.adwordsClickInfo.slot                    ...
trafficSource.campaign                                 ...
trafficSource.isTrueDirect                             ...
trafficSource.keyword                                  ...
trafficSource.medium                                   ...
```

```
trafficSource.referralPath                          ...
trafficSource.source                                ...

                                                    trafficSource.adwordsClickInfo.gclId  \
channelGrouping                                                                -0.029820
date                                                                           -0.014860
visitId                                                                        -0.026725
visitNumber                                                                     0.002463
visitStartTime                                                                 -0.024079
device.browser                                                                 -0.029304
device.deviceCategory                                                          -0.082645
device.isMobile                                                                -0.076245
device.operatingSystem                                                         -0.014625
geoNetwork.city                                                                -0.028176
geoNetwork.continent                                                            0.107233
geoNetwork.country                                                             -0.099464
geoNetwork.metro                                                               -0.050692
geoNetwork.networkDomain                                                        0.038810
geoNetwork.region                                                              -0.010074
geoNetwork.subContinent                                                         0.038008
totals.bounces                                                                 -0.039456
totals.hits                                                                    -0.028333
totals.newVisits                                                               -0.030866
totals.pageviews                                                               -0.040769
totals.transactionRevenue                                                       0.000744
trafficSource.adContent                                                         0.292169
trafficSource.adwordsClickInfo.adNetworkType                                    0.845477
trafficSource.adwordsClickInfo.gclId                                            1.000000
trafficSource.adwordsClickInfo.isVideoAd                                        0.845755
trafficSource.adwordsClickInfo.page                                             0.845704
trafficSource.adwordsClickInfo.slot                                             0.837555
trafficSource.campaign                                                         -0.417179
trafficSource.isTrueDirect                                                     -0.040477
trafficSource.keyword                                                           0.110654
trafficSource.medium                                                            0.097472
trafficSource.referralPath                                                     -0.080687
trafficSource.source                                                            0.052256

                                                    trafficSource.adwordsClickInfo.isVideoAd
channelGrouping                                                                    -0.034683
date                                                                               -0.032185
visitId                                                                            -0.049628
visitNumber                                                                         0.002460
visitStartTime                                                                     -0.047321
device.browser                                                                     -0.010854
device.deviceCategory                                                              -0.092341
device.isMobile                                                                    -0.087597
device.operatingSystem                                                              0.002221
```

geoNetwork.city	-0.03253
geoNetwork.continent	0.12410
geoNetwork.country	-0.11476
geoNetwork.metro	-0.05907
geoNetwork.networkDomain	0.04677
geoNetwork.region	-0.01001
geoNetwork.subContinent	0.04327
totals.bounces	-0.04877
totals.hits	-0.03504
totals.newVisits	-0.03687
totals.pageviews	-0.05120
totals.transactionRevenue	0.00039
trafficSource.adContent	0.38721
trafficSource.adwordsClickInfo.adNetworkType	0.99995
trafficSource.adwordsClickInfo.gclId	0.84575
trafficSource.adwordsClickInfo.isVideoAd	1.00000
trafficSource.adwordsClickInfo.page	0.99975
trafficSource.adwordsClickInfo.slot	0.99124
trafficSource.campaign	-0.48250
trafficSource.isTrueDirect	-0.05047
trafficSource.keyword	0.12881
trafficSource.medium	0.11336
trafficSource.referralPath	-0.09384
trafficSource.source	0.06077

	trafficSource.adwordsClickInfo.page \
channelGrouping	-0.034674
date	-0.032220
visitId	-0.049735
visitNumber	0.002445
visitStartTime	-0.047407
device.browser	-0.010937
device.deviceCategory	-0.092265
device.isMobile	-0.087502
device.operatingSystem	0.002118
geoNetwork.city	-0.032600
geoNetwork.continent	0.124064
geoNetwork.country	-0.114721
geoNetwork.metro	-0.059082
geoNetwork.networkDomain	0.046665
geoNetwork.region	-0.010151
geoNetwork.subContinent	0.043257
totals.bounces	-0.048753
totals.hits	-0.035067
totals.newVisits	-0.036870
totals.pageviews	-0.051157
totals.transactionRevenue	0.000391
trafficSource.adContent	0.387505

```
trafficSource.adwordsClickInfo.adNetworkType                          0.999706
trafficSource.adwordsClickInfo.gclId                                  0.845704
trafficSource.adwordsClickInfo.isVideoAd                              0.999756
trafficSource.adwordsClickInfo.page                                   1.000000
trafficSource.adwordsClickInfo.slot                                   0.990115
trafficSource.campaign                                               -0.482348
trafficSource.isTrueDirect                                           -0.050408
trafficSource.keyword                                                 0.128756
trafficSource.medium                                                  0.113341
trafficSource.referralPath                                           -0.093824
trafficSource.source                                                  0.060763

                                                  trafficSource.adwordsClickInfo.slot  \
channelGrouping                                                      -0.034379
date                                                                 -0.031803
visitId                                                              -0.049206
visitNumber                                                           0.002600
visitStartTime                                                       -0.047086
device.browser                                                       -0.011516
device.deviceCategory                                                -0.093945
device.isMobile                                                      -0.088340
device.operatingSystem                                                0.002284
geoNetwork.city                                                      -0.032212
geoNetwork.continent                                                  0.123133
geoNetwork.country                                                   -0.113953
geoNetwork.metro                                                     -0.058464
geoNetwork.networkDomain                                              0.047369
geoNetwork.region                                                    -0.010269
geoNetwork.subContinent                                               0.042941
totals.bounces                                                       -0.046618
totals.hits                                                          -0.033464
totals.newVisits                                                     -0.036382
totals.pageviews                                                     -0.049231
totals.transactionRevenue                                             0.000522
trafficSource.adContent                                               0.381714
trafficSource.adwordsClickInfo.adNetworkType                          0.991004
trafficSource.adwordsClickInfo.gclId                                  0.837555
trafficSource.adwordsClickInfo.isVideoAd                              0.991244
trafficSource.adwordsClickInfo.page                                   0.990115
trafficSource.adwordsClickInfo.slot                                   1.000000
trafficSource.campaign                                               -0.478313
trafficSource.isTrueDirect                                           -0.050384
trafficSource.keyword                                                 0.127889
trafficSource.medium                                                  0.112376
trafficSource.referralPath                                           -0.093025
trafficSource.source                                                  0.060246

                                                  trafficSource.campaign  \
```

channelGrouping	-0.180634
date	0.011005
visitId	0.010320
visitNumber	-0.006404
visitStartTime	0.008574
device.browser	-0.028068
device.deviceCategory	0.013255
device.isMobile	0.009048
device.operatingSystem	0.019343
geoNetwork.city	0.010953
geoNetwork.continent	-0.026785
geoNetwork.country	0.029951
geoNetwork.metro	0.003788
geoNetwork.networkDomain	-0.003271
geoNetwork.region	0.020574
geoNetwork.subContinent	-0.025354
totals.bounces	0.020529
totals.hits	0.005072
totals.newVisits	0.031104
totals.pageviews	0.024144
totals.transactionRevenue	-0.005237
trafficSource.adContent	-0.278024
trafficSource.adwordsClickInfo.adNetworkType	-0.482490
trafficSource.adwordsClickInfo.gclId	-0.417179
trafficSource.adwordsClickInfo.isVideoAd	-0.482508
trafficSource.adwordsClickInfo.page	-0.482348
trafficSource.adwordsClickInfo.slot	-0.478313
trafficSource.campaign	1.000000
trafficSource.isTrueDirect	0.058991
trafficSource.keyword	-0.001222
trafficSource.medium	-0.182770
trafficSource.referralPath	0.112122
trafficSource.source	-0.146500

	trafficSource.isTrueDirect \
channelGrouping	0.459128
date	-0.048668
visitId	-0.052487
visitNumber	-0.183824
visitStartTime	-0.043364
device.browser	0.115209
device.deviceCategory	-0.061888
device.isMobile	-0.073441
device.operatingSystem	0.116705
geoNetwork.city	0.068693
geoNetwork.continent	0.140594
geoNetwork.country	-0.115606
geoNetwork.metro	0.014914

geoNetwork.networkDomain	0.141731
geoNetwork.region	0.128550
geoNetwork.subContinent	0.074241
totals.bounces	-0.043376
totals.hits	-0.091115
totals.newVisits	-0.614322
totals.pageviews	-0.050776
totals.transactionRevenue	-0.045571
trafficSource.adContent	-0.027772
trafficSource.adwordsClickInfo.adNetworkType	-0.050466
trafficSource.adwordsClickInfo.gclId	-0.040477
trafficSource.adwordsClickInfo.isVideoAd	-0.050479
trafficSource.adwordsClickInfo.page	-0.050408
trafficSource.adwordsClickInfo.slot	-0.050384
trafficSource.campaign	0.058991
trafficSource.isTrueDirect	1.000000
trafficSource.keyword	-0.233993
trafficSource.medium	0.609524
trafficSource.referralPath	-0.036608
trafficSource.source	0.435115

	trafficSource.keyword \
channelGrouping	0.277461
date	-0.182715
visitId	-0.183006
visitNumber	0.003727
visitStartTime	-0.180860
device.browser	0.003693
device.deviceCategory	-0.199060
device.isMobile	-0.206355
device.operatingSystem	-0.096161
geoNetwork.city	-0.118140
geoNetwork.continent	-0.077054
geoNetwork.country	0.037297
geoNetwork.metro	-0.079138
geoNetwork.networkDomain	0.010903
geoNetwork.region	-0.123616
geoNetwork.subContinent	-0.007277
totals.bounces	-0.019875
totals.hits	-0.032668
totals.newVisits	0.019020
totals.pageviews	-0.043250
totals.transactionRevenue	0.019408
trafficSource.adContent	0.069812
trafficSource.adwordsClickInfo.adNetworkType	0.128802
trafficSource.adwordsClickInfo.gclId	0.110654
trafficSource.adwordsClickInfo.isVideoAd	0.128814
trafficSource.adwordsClickInfo.page	0.128756

```
trafficSource.adwordsClickInfo.slot                        0.127889
trafficSource.campaign                                    -0.001222
trafficSource.isTrueDirect                                -0.233993
trafficSource.keyword                                      1.000000
trafficSource.medium                                      -0.182265
trafficSource.referralPath                                -0.502552
trafficSource.source                                       0.321503

                                                  trafficSource.medium  \
channelGrouping                                               0.820530
date                                                         -0.098672
visitId                                                      -0.098922
visitNumber                                                  -0.035906
visitStartTime                                               -0.090603
device.browser                                                0.063643
device.deviceCategory                                        -0.160634
device.isMobile                                              -0.176099
device.operatingSystem                                        0.067980
geoNetwork.city                                               0.037206
geoNetwork.continent                                          0.073413
geoNetwork.country                                           -0.062387
geoNetwork.metro                                              0.020222
geoNetwork.networkDomain                                      0.080594
geoNetwork.region                                             0.050462
geoNetwork.subContinent                                       0.059617
totals.bounces                                               -0.011817
totals.hits                                                  -0.028749
totals.newVisits                                             -0.049612
totals.pageviews                                             -0.028849
totals.transactionRevenue                                    -0.007768
trafficSource.adContent                                       0.074382
trafficSource.adwordsClickInfo.adNetworkType                  0.113363
trafficSource.adwordsClickInfo.gclId                          0.097472
trafficSource.adwordsClickInfo.isVideoAd                      0.113368
trafficSource.adwordsClickInfo.page                           0.113341
trafficSource.adwordsClickInfo.slot                           0.112376
trafficSource.campaign                                       -0.182770
trafficSource.isTrueDirect                                    0.609524
trafficSource.keyword                                        -0.182265
trafficSource.medium                                          1.000000
trafficSource.referralPath                                   -0.431187
trafficSource.source                                          0.746644

                                                  trafficSource.referralPath  \
channelGrouping                                                   -0.576718
date                                                               0.058212
visitId                                                            0.047618
visitNumber                                                       -0.020893
```

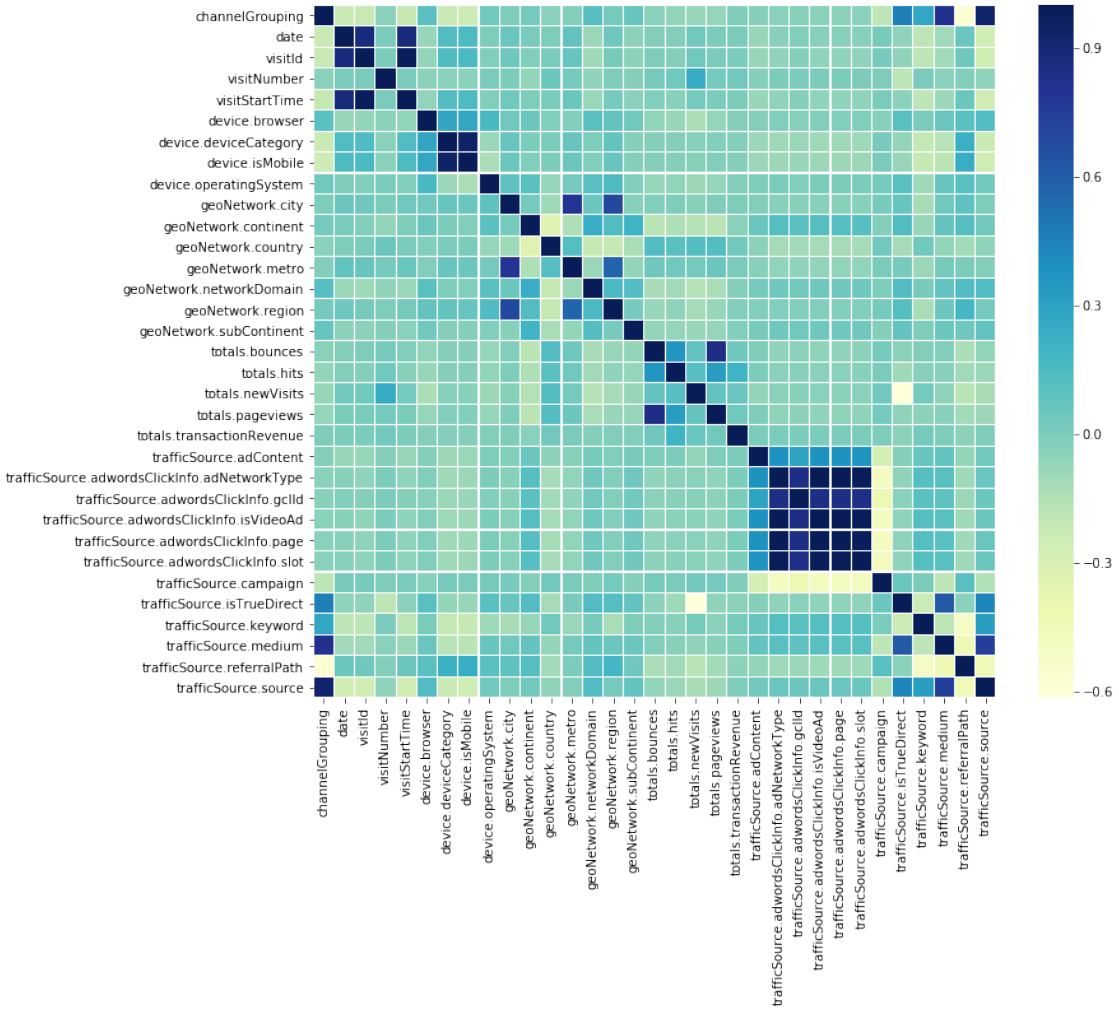
visitStartTime	0.049100
device.browser	0.116223
device.deviceCategory	0.227207
device.isMobile	0.240214
device.operatingSystem	0.113594
geoNetwork.city	0.093450
geoNetwork.continent	0.125781
geoNetwork.country	-0.106029
geoNetwork.metro	0.003459
geoNetwork.networkDomain	0.137448
geoNetwork.region	0.181985
geoNetwork.subContinent	0.045878
totals.bounces	-0.129265
totals.hits	-0.097983
totals.newVisits	-0.171241
totals.pageviews	-0.109226
totals.transactionRevenue	-0.041578
trafficSource.adContent	-0.061573
trafficSource.adwordsClickInfo.adNetworkType	-0.093842
trafficSource.adwordsClickInfo.gclId	-0.080687
trafficSource.adwordsClickInfo.isVideoAd	-0.093847
trafficSource.adwordsClickInfo.page	-0.093824
trafficSource.adwordsClickInfo.slot	-0.093025
trafficSource.campaign	0.112122
trafficSource.isTrueDirect	-0.036608
trafficSource.keyword	-0.502552
trafficSource.medium	-0.431187
trafficSource.referralPath	1.000000
trafficSource.source	-0.449347

	trafficSource.source
channelGrouping	0.932689
date	-0.259641
visitId	-0.258301
visitNumber	-0.047765
visitStartTime	-0.249542
device.browser	0.138060
device.deviceCategory	-0.227627
device.isMobile	-0.246458
device.operatingSystem	0.037027
geoNetwork.city	-0.008439
geoNetwork.continent	0.034345
geoNetwork.country	-0.050672
geoNetwork.metro	-0.018633
geoNetwork.networkDomain	0.138995
geoNetwork.region	0.026760
geoNetwork.subContinent	0.083987
totals.bounces	-0.058798

```
totals.hits                                          -0.078856
totals.newVisits                                     -0.120260
totals.pageviews                                     -0.090836
totals.transactionRevenue                            -0.005958
trafficSource.adContent                               0.039877
trafficSource.adwordsClickInfo.adNetworkType          0.060775
trafficSource.adwordsClickInfo.gclId                  0.052256
trafficSource.adwordsClickInfo.isVideoAd              0.060778
trafficSource.adwordsClickInfo.page                   0.060763
trafficSource.adwordsClickInfo.slot                   0.060246
trafficSource.campaign                               -0.146500
trafficSource.isTrueDirect                            0.435115
trafficSource.keyword                                 0.321503
trafficSource.medium                                  0.746644
trafficSource.referralPath                           -0.449347
trafficSource.source                                  1.000000

[33 rows x 33 columns]
```

In [22]:
```
f, ax = plt.subplots(figsize=(12, 10))
sns.heatmap(corr_mat, ax=ax, cmap="YlGnBu", linewidths=0.1)
```

Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a5cb1d438>
```



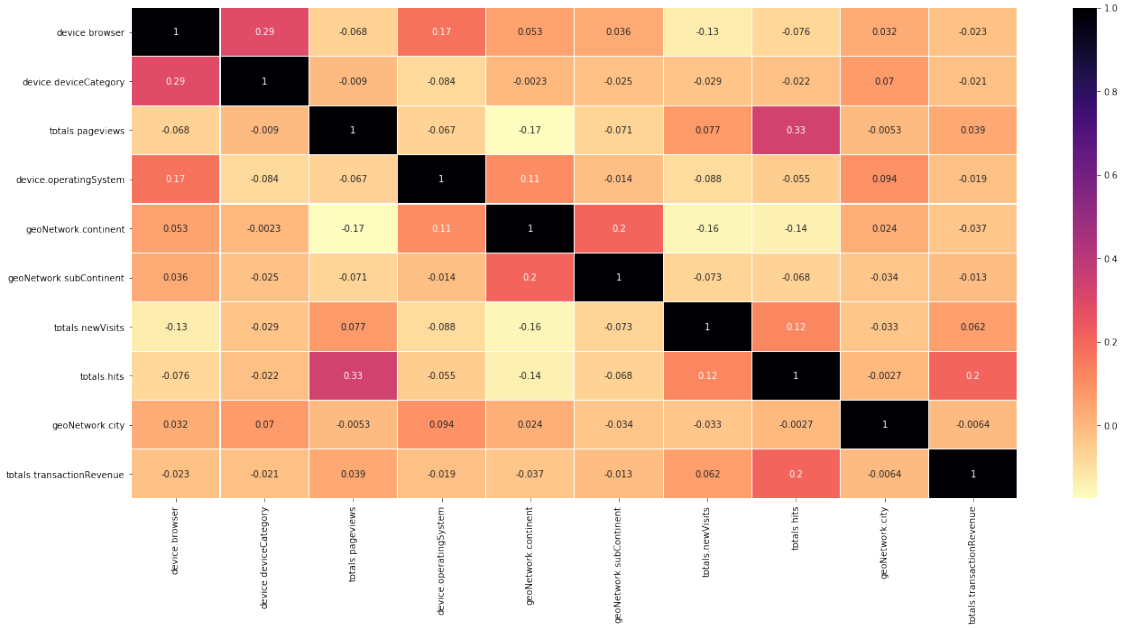
Let's create a heatmap involving only subset of columns.

```
In [23]: cat_cols1 =  ['device.browser', 'device.deviceCategory', 'totals.pageviews',
                       'device.operatingSystem', 'geoNetwork.continent','geoNetwork.subContinent
                       'totals.hits','geoNetwork.city','totals.transactionRevenue']
         new_train = train_df[cat_cols1].copy()
         new_train['totals.transactionRevenue'] = new_train['totals.transactionRevenue'].fillna
         new_train.head()

Out[23]:    device.browser  device.deviceCategory  totals.pageviews  \
         0              12                     0               0.0
         1              17                     0               0.0
         2              12                     0               0.0
         3              40                     0               0.0
         4              12                     1               0.0

            device.operatingSystem  geoNetwork.continent  geoNetwork.subContinent  \
```

```
0                  17                  3                  21
1                   7                  5                   1
2                  17                  4                  19
3                   6                  3                  16
4                   1                  4                  13

   totals.newVisits  totals.hits  geoNetwork.city  totals.transactionRevenue
0               0.0         1.0              189                        0.0
1               0.0         1.0              459                        0.0
2               0.0         1.0              233                        0.0
3               0.0         1.0              459                        0.0
4               1.0         1.0              459                        0.0
```

```
In [24]: corr_mat = new_train.corr()
         f, ax = plt.subplots(figsize=(22, 10))
         sns.heatmap(corr_mat,annot=True, ax=ax, cmap="magma_r", linewidths=0.1)
```

```
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x1a5c393240>
```



It is showing direct positive correlation between page views and total hits. The more page views mean more page hits.

```
In [25]: def ImputeMissingValues(train):
             for df in [train]:
                 df['trafficSource.adContent'].fillna('N/A', inplace=True)
                 df['trafficSource.adwordsClickInfo.slot'].fillna('N/A', inplace=True)
                 df['trafficSource.adwordsClickInfo.page'].fillna(0.0, inplace=True)
                 df['trafficSource.adwordsClickInfo.isVideoAd'].fillna('N/A', inplace=True)
```

```
                df['trafficSource.adwordsClickInfo.adNetworkType'].fillna('N/A', inplace=True)
                df['trafficSource.adwordsClickInfo.gclId'].fillna('N/A', inplace=True)
                df['trafficSource.isTrueDirect'].fillna('N/A', inplace=True)
                df['trafficSource.referralPath'].fillna('N/A', inplace=True)
                df['trafficSource.keyword'].fillna('N/A', inplace=True)
                df['totals.bounces'].fillna(0.0, inplace=True)
                df['totals.newVisits'].fillna(0.0, inplace=True)
                df['totals.pageviews'].fillna(0.0, inplace=True)
                return df

        orig_df = ImputeMissingValues(orig_df)
        orig_df = add_time_features(orig_df)
```

```
In [26]: orig_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 43 columns):
channelGrouping                                 100000 non-null object
date                                            100000 non-null datetime64[ns]
fullVisitorId                                   100000 non-null object
visitId                                         100000 non-null int64
visitNumber                                     100000 non-null int64
visitStartTime                                  100000 non-null int64
device.browser                                  100000 non-null object
device.deviceCategory                           100000 non-null object
device.isMobile                                 100000 non-null bool
device.operatingSystem                          100000 non-null object
geoNetwork.city                                 100000 non-null object
geoNetwork.continent                            100000 non-null object
geoNetwork.country                              100000 non-null object
geoNetwork.metro                                100000 non-null object
geoNetwork.networkDomain                        100000 non-null object
geoNetwork.region                               100000 non-null object
geoNetwork.subContinent                         100000 non-null object
totals.bounces                                  100000 non-null object
totals.hits                                     100000 non-null object
totals.newVisits                                100000 non-null object
totals.pageviews                                100000 non-null object
totals.transactionRevenue                       100000 non-null float64
trafficSource.adContent                         100000 non-null object
trafficSource.adwordsClickInfo.adNetworkType    100000 non-null object
trafficSource.adwordsClickInfo.gclId            100000 non-null object
trafficSource.adwordsClickInfo.isVideoAd        100000 non-null object
trafficSource.adwordsClickInfo.page             100000 non-null object
trafficSource.adwordsClickInfo.slot             100000 non-null object
trafficSource.campaign                          100000 non-null object
trafficSource.isTrueDirect                      100000 non-null object
```

```
trafficSource.keyword                              100000 non-null object
trafficSource.medium                               100000 non-null object
trafficSource.referralPath                         100000 non-null object
trafficSource.source                               100000 non-null object
year                                               100000 non-null int64
month                                              100000 non-null int64
day                                                100000 non-null int64
weekday                                            100000 non-null int64
visitStartTime_                                    100000 non-null datetime64[ns]
visitStartTime_year                                100000 non-null int64
visitStartTime_month                               100000 non-null int64
visitStartTime_day                                 100000 non-null int64
visitStartTime_weekday                             100000 non-null int64
dtypes: bool(1), datetime64[ns](2), float64(1), int64(11), object(28)
memory usage: 32.1+ MB
```

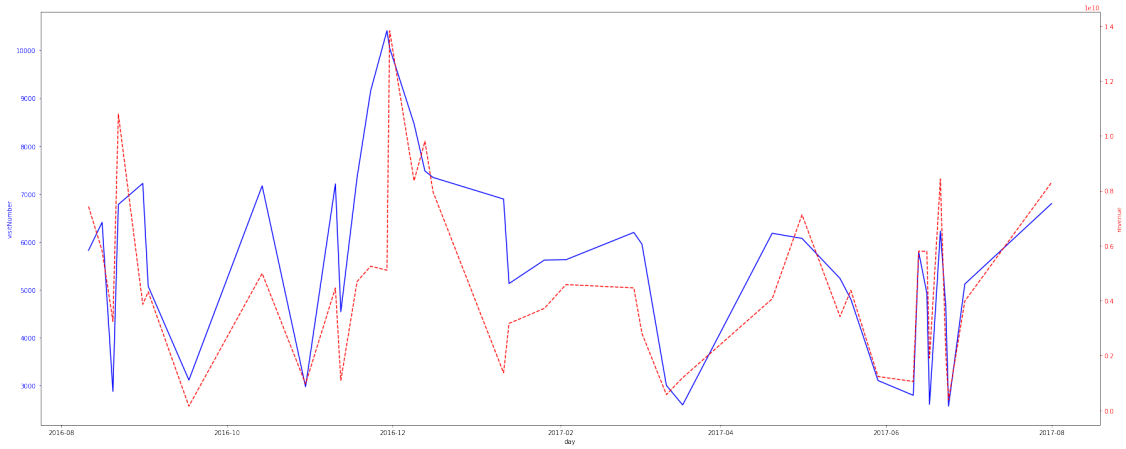
In [27]:
```
orig_df["date"] = pd.to_datetime(orig_df["date"],format="%Y%m%d")
orig_df["visitStartTime"] = pd.to_datetime(orig_df["visitStartTime"],unit='s')
revenue_datetime_df = orig_df[["totals.transactionRevenue" , "date", "totals.pageviews
revenue_datetime_df["totals.transactionRevenue"] =revenue_datetime_df["totals.transact
revenue_datetime_df["totals.pageviews"] =revenue_datetime_df["totals.pageviews"].asty
revenue_datetime_df.head()
```

Out[27]:

	totals.transactionRevenue	date	totals.pageviews
0	0	2016-09-02	1
1	0	2016-09-02	1
2	0	2016-09-02	1
3	0	2016-09-02	1
4	0	2016-09-02	1

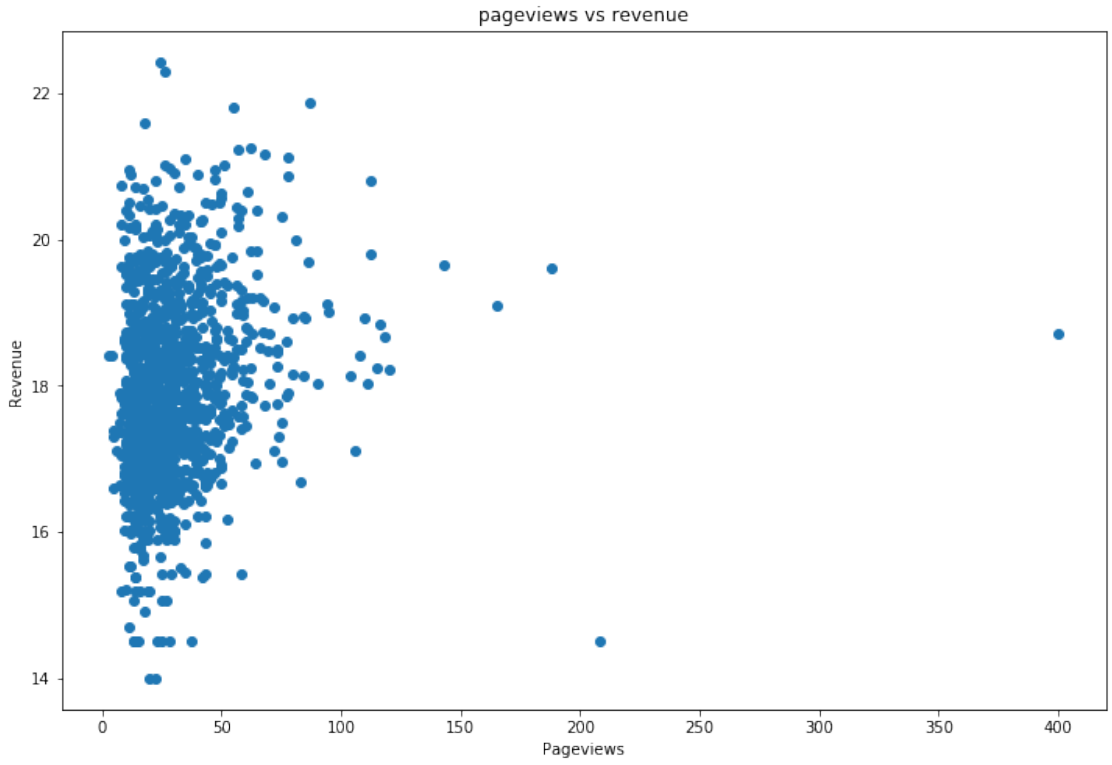
In [28]:
```
total_revenue_daily_df = revenue_datetime_df.groupby(by=["date"],axis=0).sum()
total_visitNumber_daily_df = orig_df[["date","visitNumber"]].groupby(by=["date"],axis=
```

In [29]:
```
datetime_revenue_visits_df = pd.concat([total_revenue_daily_df,total_visitNumber_daily
fig, ax1 = plt.subplots(figsize=(25,10))
t = datetime_revenue_visits_df.index
s1 = datetime_revenue_visits_df["visitNumber"]
ax1.plot(t, s1, 'b-')
ax1.set_xlabel('day')
ax1.set_ylabel('visitNumber', color='b')
ax1.tick_params('y', colors='b')
ax2 = ax1.twinx()
s2 = datetime_revenue_visits_df["totals.transactionRevenue"]
ax2.plot(t, s2, 'r--')
ax2.set_ylabel('revenue', color='r')
ax2.tick_params('y', colors='r')
fig.tight_layout()
```



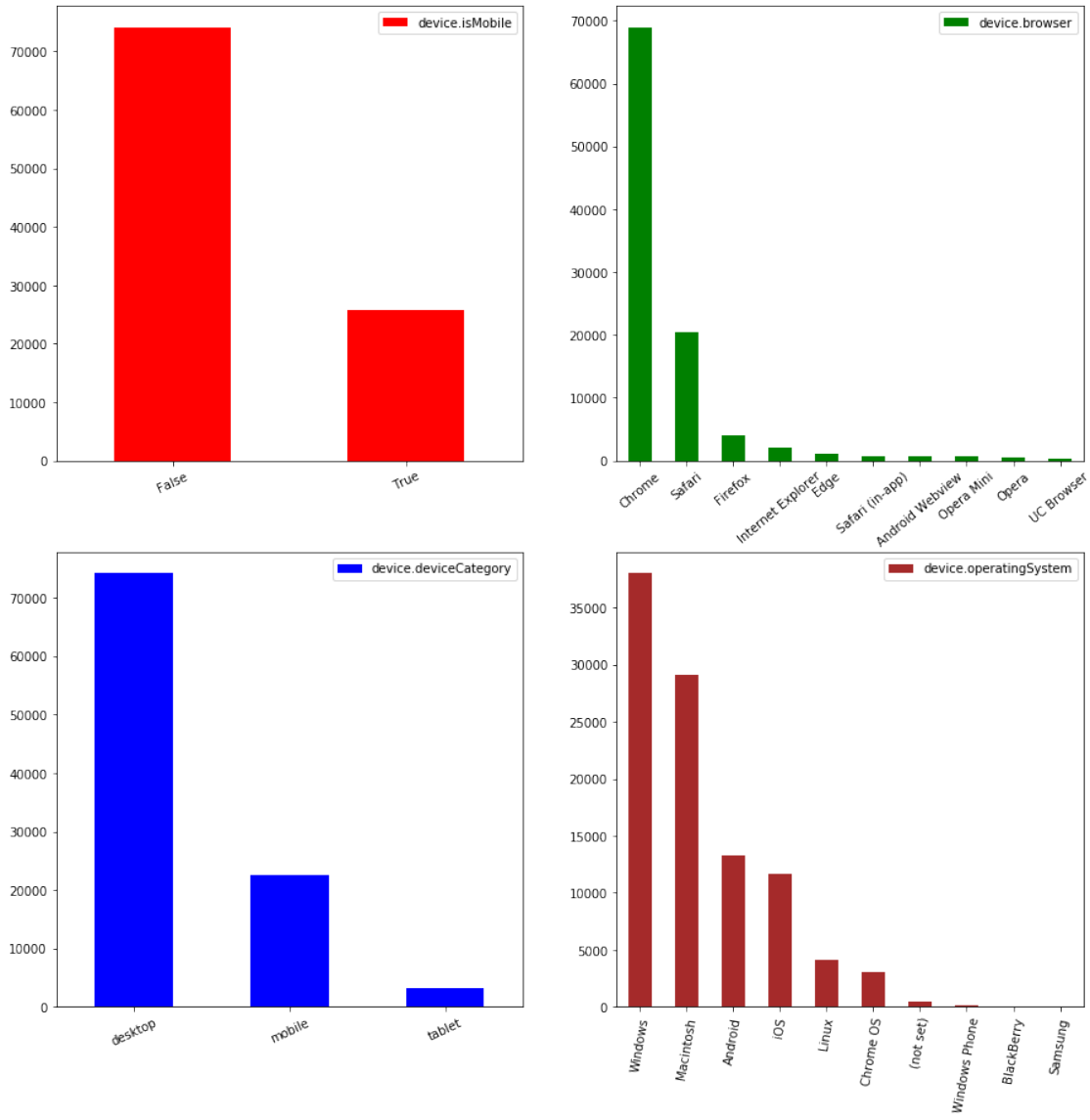
The figure above shows that when there is more visit the revenue is more in those days. Also we can see the peak at around December which signifies that the number of visits are more in December and revenue is more as well. This signifies that people tend to make more visits and purchase lot of items during Christmas as compared to other days.

```
In [30]: revenue_datetime_df["totals.transactionRevenue"] = np.log1p(revenue_datetime_df["total
         revenue_datetime_df["totals.transactionRevenue"] = revenue_datetime_df[revenue_dateti
         plt.figure(figsize=(12, 8))
         plt.title('pageviews vs revenue');
         plt.scatter(revenue_datetime_df['totals.pageviews'], revenue_datetime_df["totals.trans
         plt.xlabel('Pageviews');
         plt.ylabel('Revenue');
```



The plot above signifies that more pageviews signifies more natural log of transaction revenue.

```
In [31]: fig, axes = plt.subplots(2,2,figsize=(15,15))
         orig_df["device.isMobile"].value_counts().plot(kind="bar",ax=axes[0][0],rot=25,legend=
         orig_df["device.browser"].value_counts().head(10).plot(kind="bar",ax=axes[0][1],rot=40
         orig_df["device.deviceCategory"].value_counts().head(10).plot(kind="bar",ax=axes[1][0]
         orig_df["device.operatingSystem"].value_counts().head(10).plot(kind="bar",ax=axes[1][1

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1a5ade7b38>
```

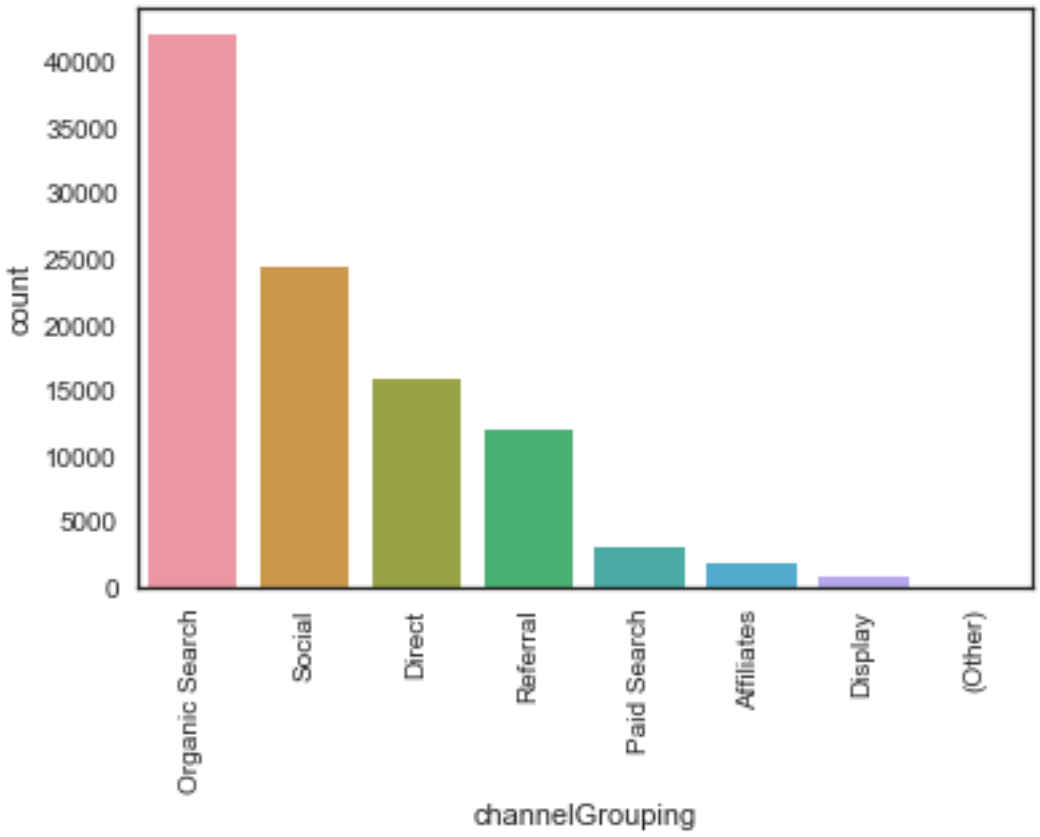


Operating systems results : TOP 1 => Windows - 38%, TOP 2 => Macintosh - 27%, TOP 3 => Android - 15%, TOP 4 => iOS - 11%, TOP 5 => Linux - 4%

Percentual of Device category: desktop 73%, mobile 23%, tablet 3%

Browser results : TOP 1 - CHROME - 69%, TOP 2 - SAFARI - 20%, TOP 3 - FIREFOX - 3%

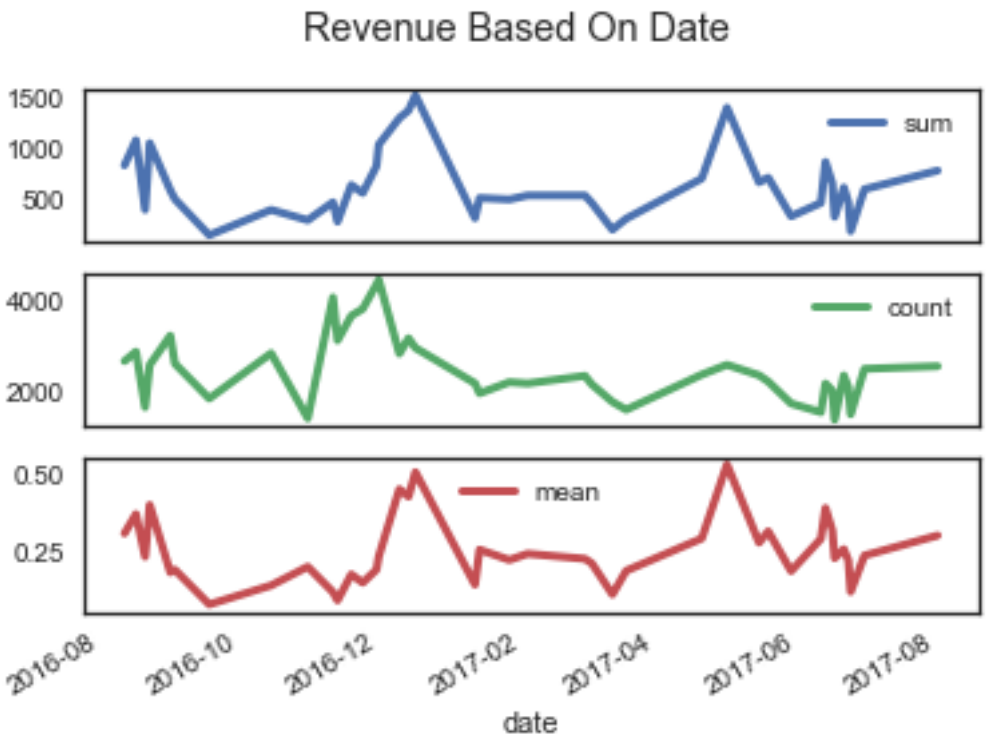
```
In [32]: sns.set(style="white")
         titanic = sns.load_dataset("titanic")
         sns.countplot(orig_df['channelGrouping'],order=orig_df['channelGrouping'].value_counts
         plt.xticks(rotation=90);
```



ChannelGrouping Results: TOP 1 => Organic Search - 43%, TOP 2 => Social - 24%, TOP 3 => Direct - 15%, TOP 4 => Referral - 12%, TOP 5 => Paid Search - 3%

```
In [33]: # plotting natural log of transactional revenue according to dates
         new_copy = orig_df.copy()
         new_copy.loc[:, "totals.transactionRevenue_ln"] = np.log1p(orig_df["totals.transactio
         new_copy.groupby("date")["totals.transactionRevenue_ln"].agg(['sum', 'count', 'mean'])
```

```
Out[33]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x1a5a8624a8>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x1a5a8200b8>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x1a5a773630>],
               dtype=object)
```



Task 3 : Geography Plots
Let's do clustering of geography plots.

```
In [34]: def horizontal_bar_chart(cnt_srs, color):
             trace = go.Bar(
                 y=cnt_srs.index[::-1],
                 x=cnt_srs.values[::-1],
                 showlegend=False,
                 orientation = 'h',
                 marker=dict(
                     color=color,
                 ),
             )
             return trace
```

```
In [35]: orig_df["totals.transactionRevenue"] = orig_df["totals.transactionRevenue"].astype('fl
         cnt_srs = orig_df.groupby('geoNetwork.country')['totals.transactionRevenue'].agg(['siz
         cnt_srs.columns = ["count", "count of non-zero revenue", "mean"]
         cnt_srs = cnt_srs.sort_values(by="count", ascending=False)
         trace1 = horizontal_bar_chart(cnt_srs["count"].head(5), 'rgba(128,0,128, 1.0)')
         trace2 = horizontal_bar_chart(cnt_srs["count of non-zero revenue"].head(5), 'rgba(128
         trace3 = horizontal_bar_chart(cnt_srs["mean"].head(5), 'rgba(128,0,128, 1.0)')

         cnt_srs = orig_df.groupby('geoNetwork.continent')['totals.transactionRevenue'].agg(['s
```

```
cnt_srs.columns = ["count", "count of non-zero revenue", "mean"]
cnt_srs = cnt_srs.sort_values(by="count", ascending=False)
trace4 = horizontal_bar_chart(cnt_srs["count"].head(5), 'rgba(128,0,128, 1.0)')
trace5 = horizontal_bar_chart(cnt_srs["count of non-zero revenue"].head(5), 'rgba(128
trace6 = horizontal_bar_chart(cnt_srs["mean"].head(5), 'rgba(128,0,128, 1.0)')

cnt_srs = orig_df.groupby('geoNetwork.subContinent')['totals.transactionRevenue'].agg
cnt_srs.columns = ["count", "count of non-zero revenue", "mean"]
cnt_srs = cnt_srs.sort_values(by="count", ascending=False)
trace7 = horizontal_bar_chart(cnt_srs["count"].head(10), 'rgba(128,0,128, 1.0)')
trace8 = horizontal_bar_chart(cnt_srs["count of non-zero revenue"].head(10), 'rgba(128
trace9 = horizontal_bar_chart(cnt_srs["mean"].head(10), 'rgba(128,0,128, 1.0)')
```

In [36]:
```
fig = tools.make_subplots(rows=3, cols=3, vertical_spacing=0.04,
                          subplot_titles=["Country - Count", "Country - Non-zero Rever
                                          "Continent - Count", "Continent - Non-zero R
                                          "SubContinent - Count", "SubContinent - Non-
                                          ])
```

```
This is the format of your plot grid:
[ (1,1) x1,y1 ]  [ (1,2) x2,y2 ]  [ (1,3) x3,y3 ]
[ (2,1) x4,y4 ]  [ (2,2) x5,y5 ]  [ (2,3) x6,y6 ]
[ (3,1) x7,y7 ]  [ (3,2) x8,y8 ]  [ (3,3) x9,y9 ]
```

In [37]:
```
fig.append_trace(trace1, 1, 1)
fig.append_trace(trace2, 1, 2)
fig.append_trace(trace3, 1, 3)
fig.append_trace(trace4, 2, 1)
fig.append_trace(trace5, 2, 2)
fig.append_trace(trace6, 2, 3)
fig.append_trace(trace7, 3, 1)
fig.append_trace(trace8, 3, 2)
fig.append_trace(trace9, 3, 3)
```

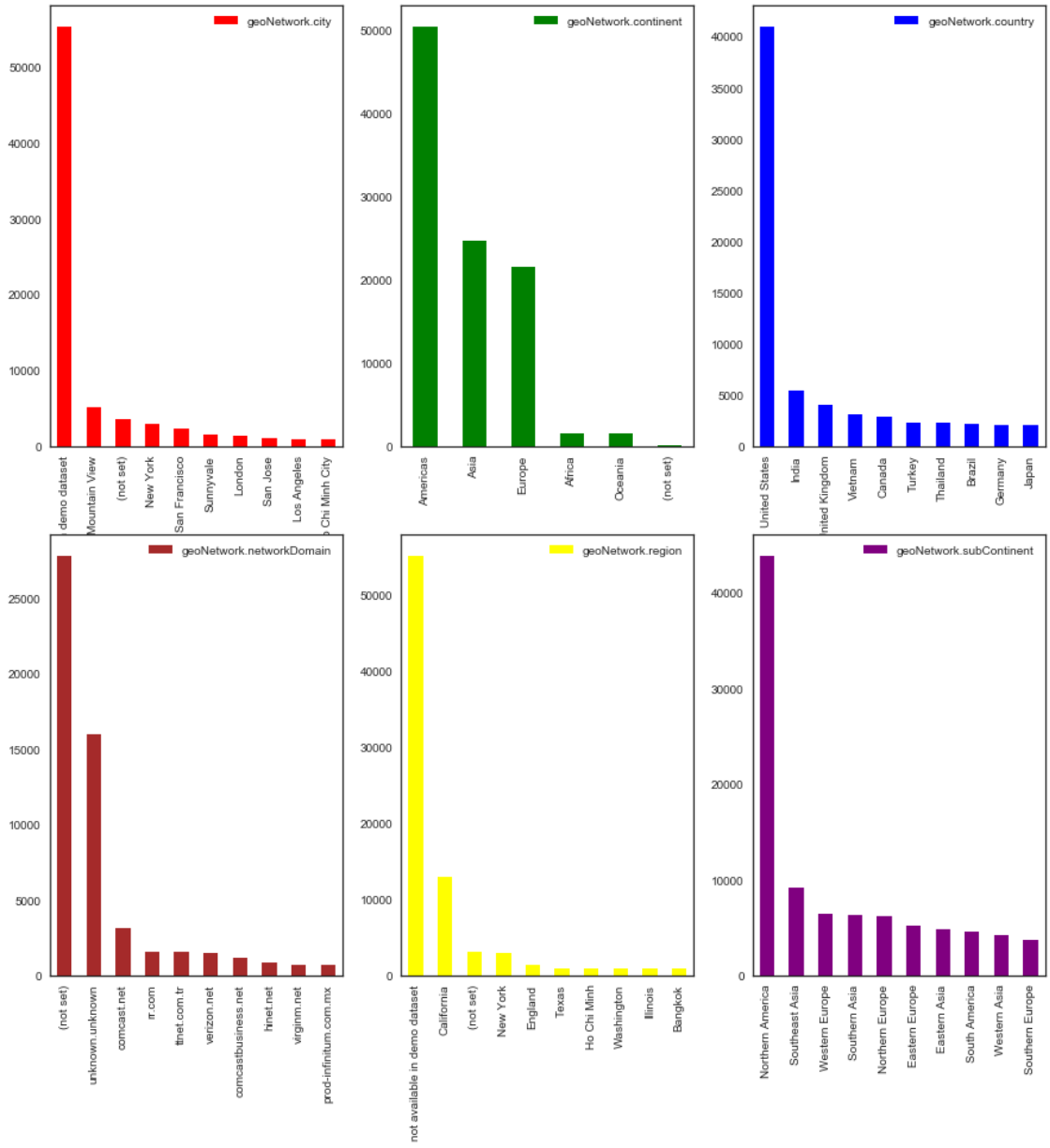
In [38]:
```
fig['layout'].update(height=1200, width=1200, paper_bgcolor='rgb(233,233,233)', title=
py.iplot(fig, filename='Geography-plots')
```

Out[38]: <plotly.tools.PlotlyDisplay object>

United States has the largest transaction revenue in terms of country results. In terms of continent Americas is at the top. Northern Anerica has maximum transaction revenue in subcontinents.

In [39]:
```
fig, axes = plt.subplots(2,3,figsize=(15,15))
orig_df["geoNetwork.city"].value_counts().head(10).plot(kind="bar",ax=axes[0][0],rot=9
orig_df["geoNetwork.continent"].value_counts().head(10).plot(kind="bar",ax=axes[0][1]
orig_df["geoNetwork.country"].value_counts().head(10).plot(kind="bar",ax=axes[0][2],ro
orig_df["geoNetwork.networkDomain"].value_counts().head(10).plot(kind="bar",ax=axes[1
orig_df["geoNetwork.region"].value_counts().head(10).plot(kind="bar",ax=axes[1][1],rot
orig_df["geoNetwork.subContinent"].value_counts().head(10).plot(kind="bar",ax=axes[1]
```

```
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x1a35e52f60>
```



Sub-Continents Results : TOP 1 => Northern America - 44%, TOP 2 => Southeast Asia - 8%, TOP 3 => Western Europe - 6%, TOP 4 => Southern Asia - 6%, TOP 5 => Northern Europe - 6%

Task 4 : Buying Score Probability Function

```
In [40]: df_new = train_df[['fullVisitorId','visitNumber','totals.transactionRevenue']].copy()
         df_new['totals.transactionRevenue'] = df_new['totals.transactionRevenue'].astype(bool)
         agg_dict = {}
         agg_dict['visitNumber'] = "max"
         agg_dict['totals.transactionRevenue'] = "sum"
```

```
df_new = df_new.groupby('fullVisitorId').agg(agg_dict).reset_index()
df_new[df_new['totals.transactionRevenue'] > 1].head
df_new['buy_probability'] = df_new['totals.transactionRevenue']/df_new['visitNumber']
top_clients = df_new.sort_values('buy_probability', ascending= False)['fullVisitorId']
print("Top 10 users who are most likely to buy from GStore are :\n" , top_clients.valu
```

```
Top 10 users who are most likely to buy from GStore are :
 ['653069947099114599' '2518447979646501002' '9710480501909231921'
 '8186007457246709564' '3614707430894059857' '6275380876231092642'
 '144847938814859371' '5970035247923497864' '1881458907401964229'
 '3112141012835314636']
```

Task 5 : Addition of External Data Set

I have taken external data set from https://www.kaggle.com/satian/exported-google-analytics-data. There are 4 files : Test_external_data.csv, Test_external_data_2.csv, Train_external_data.csv, Train_external_data_2.csv. I have used only 2 of them. They contain 6 columns : Client Id, Sessions, Avg_Session_Duration, Bounce_Rate, Revenue Transactions, Goal Conversion Rate. It really helps in making the accurate predictions of the total revenue per user. My rank improves from around 1200 to 846 after incorporating this data set into my training model.

Let's work on external data set. I merged the columns of external data set into our existing data sets.

```
In [41]: # load external dataset
         train_data = pd.read_csv('Train_external_data.csv', low_memory=False, skiprows=6, dtyp
         test_data = pd.read_csv('Test_external_data.csv', low_memory=False, skiprows=6, dtype=

         for df in [train_data,test_data]:
             df["visitId"] = df["Client Id"].apply(lambda x: x.split('.', 1)[1]).astype(str)

         cat_cols = ['Revenue','Sessions','Avg. Session Duration', 'Bounce Rate', 'Transactions
                     'visitId']

         # did label encoding on the external data set
         for col in cat_cols:
             lbl = preprocessing.LabelEncoder()
             lbl.fit(list(train_data[col].values.astype('str')) + list(test_data[col].values.as
             train_data[col] = lbl.transform(list(train_data[col].values.astype('str')))
             test_data[col] = lbl.transform(list(test_data[col].values.astype('str')))

         train_df.info()
         train_new = train_df.merge(train_data, how="left", on="visitId")
         test_new = test_df.merge(test_data, how="left", on="visitId")

         for df in [train_new, test_new]:
             df.drop("Client Id", axis = 1, inplace=True)
```

```
        # imputing Nan values
        for df in [train_new, test_new]:
            df["Sessions"] = df["Sessions"].fillna(0)
            df["Avg. Session Duration"] = df["Avg. Session Duration"].fillna(0)
            df["Bounce Rate"] = df["Bounce Rate"].fillna(0)
            df["Revenue"] = df["Revenue"].fillna(0)
            df["Transactions"] = df["Transactions"].fillna(0)
            df["Goal Conversion Rate"] = df["Goal Conversion Rate"].fillna(0)
            df['trafficSource.adContent'].fillna('N/A', inplace=True)
            df['trafficSource.adwordsClickInfo.slot'].fillna('N/A', inplace=True)
            df['trafficSource.adwordsClickInfo.page'].fillna(0.0, inplace=True)
            df['trafficSource.adwordsClickInfo.isVideoAd'].fillna('N/A', inplace=True)
            df['trafficSource.adwordsClickInfo.adNetworkType'].fillna('N/A', inplace=True)
            df['trafficSource.adwordsClickInfo.gclId'].fillna('N/A', inplace=True)
            df['trafficSource.isTrueDirect'].fillna('N/A', inplace=True)
            df['trafficSource.referralPath'].fillna('N/A', inplace=True)
            df['trafficSource.keyword'].fillna('N/A', inplace=True)
            df['totals.bounces'].fillna(0.0, inplace=True)
            df['totals.newVisits'].fillna(0.0, inplace=True)
            df['totals.pageviews'].fillna(0.0, inplace=True)

        del train_df
        del test_df
        train_df = train_new
        test_df = test_new
        del train_new
        del test_new
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 43 columns):
channelGrouping                                 100000 non-null int64
date                                            100000 non-null datetime64[ns]
fullVisitorId                                   100000 non-null object
visitId                                         100000 non-null int64
visitNumber                                     100000 non-null float64
visitStartTime                                  100000 non-null float64
device.browser                                  100000 non-null int64
device.deviceCategory                           100000 non-null int64
device.isMobile                                 100000 non-null bool
device.operatingSystem                          100000 non-null int64
geoNetwork.city                                 100000 non-null int64
geoNetwork.continent                            100000 non-null int64
geoNetwork.country                              100000 non-null int64
geoNetwork.metro                                100000 non-null int64
geoNetwork.networkDomain                        100000 non-null int64
geoNetwork.region                               100000 non-null int64
geoNetwork.subContinent                         100000 non-null int64
```

```
totals.bounces                                      100000 non-null float64
totals.hits                                         100000 non-null float64
totals.newVisits                                    100000 non-null float64
totals.pageviews                                    100000 non-null float64
totals.transactionRevenue                           100000 non-null float64
trafficSource.adContent                             100000 non-null int64
trafficSource.adwordsClickInfo.adNetworkType        100000 non-null int64
trafficSource.adwordsClickInfo.gclId                100000 non-null int64
trafficSource.adwordsClickInfo.isVideoAd            100000 non-null int64
trafficSource.adwordsClickInfo.page                 100000 non-null int64
trafficSource.adwordsClickInfo.slot                 100000 non-null int64
trafficSource.campaign                              100000 non-null int64
trafficSource.isTrueDirect                          100000 non-null int64
trafficSource.keyword                               100000 non-null int64
trafficSource.medium                                100000 non-null int64
trafficSource.referralPath                          100000 non-null int64
trafficSource.source                                100000 non-null int64
year                                                100000 non-null int64
month                                               100000 non-null int64
day                                                 100000 non-null int64
weekday                                             100000 non-null int64
visitStartTime_                                     100000 non-null datetime64[ns]
visitStartTime_year                                 100000 non-null int64
visitStartTime_month                                100000 non-null int64
visitStartTime_day                                  100000 non-null int64
visitStartTime_weekday                              100000 non-null int64
dtypes: bool(1), datetime64[ns](2), float64(7), int64(32), object(1)
memory usage: 32.1+ MB
```

```
In [42]: train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100001 entries, 0 to 100000
Data columns (total 49 columns):
channelGrouping                                     100001 non-null int64
date                                                100001 non-null datetime64[ns]
fullVisitorId                                       100001 non-null object
visitId                                             100001 non-null int64
visitNumber                                         100001 non-null float64
visitStartTime                                      100001 non-null float64
device.browser                                      100001 non-null int64
device.deviceCategory                               100001 non-null int64
device.isMobile                                     100001 non-null bool
device.operatingSystem                              100001 non-null int64
geoNetwork.city                                     100001 non-null int64
geoNetwork.continent                                100001 non-null int64
geoNetwork.country                                  100001 non-null int64
```

```
geoNetwork.metro                                    100001 non-null int64
geoNetwork.networkDomain                            100001 non-null int64
geoNetwork.region                                   100001 non-null int64
geoNetwork.subContinent                             100001 non-null int64
totals.bounces                                      100001 non-null float64
totals.hits                                         100001 non-null float64
totals.newVisits                                    100001 non-null float64
totals.pageviews                                    100001 non-null float64
totals.transactionRevenue                           100001 non-null float64
trafficSource.adContent                             100001 non-null int64
trafficSource.adwordsClickInfo.adNetworkType        100001 non-null int64
trafficSource.adwordsClickInfo.gclId                100001 non-null int64
trafficSource.adwordsClickInfo.isVideoAd            100001 non-null int64
trafficSource.adwordsClickInfo.page                 100001 non-null int64
trafficSource.adwordsClickInfo.slot                 100001 non-null int64
trafficSource.campaign                              100001 non-null int64
trafficSource.isTrueDirect                          100001 non-null int64
trafficSource.keyword                               100001 non-null int64
trafficSource.medium                                100001 non-null int64
trafficSource.referralPath                          100001 non-null int64
trafficSource.source                                100001 non-null int64
year                                                100001 non-null int64
month                                               100001 non-null int64
day                                                 100001 non-null int64
weekday                                             100001 non-null int64
visitStartTime_                                     100001 non-null datetime64[ns]
visitStartTime_year                                 100001 non-null int64
visitStartTime_month                                100001 non-null int64
visitStartTime_day                                  100001 non-null int64
visitStartTime_weekday                              100001 non-null int64
Sessions                                            100001 non-null float64
Avg. Session Duration                               100001 non-null float64
Bounce Rate                                         100001 non-null float64
Revenue                                             100001 non-null float64
Transactions                                        100001 non-null float64
Goal Conversion Rate                                100001 non-null float64
dtypes: bool(1), datetime64[ns](2), float64(13), int64(32), object(1)
memory usage: 37.5+ MB
```

Task 6 : Predictive Model

I have used LGBMRegressor with KFold incorporating various factors and adding some more features related to dates.

```
In [43]: def add_time_features(df):
             df['date'] = pd.to_datetime(df['date'], format='%Y%m%d', errors='ignore')
             df['year'] = df['date'].apply(lambda x: x.year)
             df['month'] = df['date'].apply(lambda x: x.month)
```

```
        df['day'] = df['date'].apply(lambda x: x.day)
        df['weekday'] = df['date'].apply(lambda x: x.weekday())
        df['visitStartTime_'] = pd.to_datetime(df['visitStartTime'],unit="s")
        df['visitStartTime_year'] = df['visitStartTime_'].apply(lambda x: x.year)
        df['visitStartTime_month'] = df['visitStartTime_'].apply(lambda x: x.month)
        df['visitStartTime_day'] = df['visitStartTime_'].apply(lambda x: x.day)
        df['visitStartTime_weekday'] = df['visitStartTime_'].apply(lambda x: x.weekday())
        return df

    date_features = [#"year","month","day","weekday",'visitStartTime_year',
        "visitStartTime_month","visitStartTime_day","visitStartTime_weekday","visitStartTi

    # add date-time features to train into our model.
    train_df = add_time_features(train_df)
    test_df = add_time_features(test_df)

    # add more features related to hits, pageviews, revenue.
    for df in [train_df, test_df]:
        df['hits/pageviews'] = (df["totals.pageviews"]/(df["totals.hits"])).apply(lambda x
        df['is_high_hits'] = np.logical_or(df["totals.hits"]>5,df["totals.pageviews"]>5).a
        df["Revenue"] = np.log1p(df["Revenue"])

    # drop columns which are insignificant
    no_use = ['visitStartTime', "date", "fullVisitorId","visitId", 'trafficSource.referral
              'visitStartTime_year','totals.transactionRevenue']
    train_df['totals.transactionRevenue'] = np.log1p(train_df['totals.transactionRevenue']
    X = train_df.drop(no_use, axis=1)
    y = train_df['totals.transactionRevenue']
    X_test = test_df.drop([col for col in no_use if col in test_df.columns], axis=1)
```

In [44]:
```
params = {"objective" : "regression",
          "metric" : "rmse",
          "min_child_samples": 20,
          "reg_alpha": 0.033948965191129526,
          "reg_lambda": 0.06490202783578762,
          "num_leaves" : 34,
          "learning_rate" : 0.019732018807662323,
          "subsample" : 0.876,
          "colsample_bytree" : 0.85,
          "subsample_freq ": 5
         }

# run KFold with LGBMRegressor
n_fold = 5
folds = KFold(n_splits=n_fold, random_state=42)
model = lgb.LGBMRegressor(**params, n_estimators = 10000, nthread = 4, n_jobs = -1)
```

In [45]:
```
# submit our predicted results
prediction = np.zeros(len(test_df))
```

```
        for fold_n, (train_index, valid_index) in enumerate(folds.split(X)):
            print('Fold', fold_n, 'started at', time.ctime())
            X_train, X_valid = X.iloc[train_index], X.iloc[valid_index]
            y_train, y_valid = y.iloc[train_index], y.iloc[valid_index]

            model.fit(X_train, y_train,
                      eval_set=[(X_train, y_train), (X_valid, y_valid)], eval_metric='rmse'
                      verbose=500, early_stopping_rounds=100)
        y_pred = model.predict(X_test, num_iteration=model.best_iteration_)
        prediction += y_pred
        prediction /= n_fold
        actual_score = model.best_score_
        submission = test_df[['fullVisitorId']].copy()
        submission.loc[:, 'PredictedLogRevenue'] = prediction
        grouped_test = submission[['fullVisitorId', 'PredictedLogRevenue']].groupby('fullVisi
        grouped_test.to_csv('submit1.csv',index=False)
```

```
Fold 0 started at Tue Oct 23 07:46:21 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[129]        training's rmse: 1.56111        valid_1's rmse: 1.63239
Fold 1 started at Tue Oct 23 07:46:24 2018
Training until validation scores don't improve for 100 rounds.
[500]        training's rmse: 1.3979        valid_1's rmse: 1.56313
Early stopping, best iteration is:
[485]        training's rmse: 1.40197        valid_1's rmse: 1.56258
Fold 2 started at Tue Oct 23 07:46:31 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[188]        training's rmse: 1.48462        valid_1's rmse: 1.71596
Fold 3 started at Tue Oct 23 07:46:34 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[148]        training's rmse: 1.50969        valid_1's rmse: 1.79924
Fold 4 started at Tue Oct 23 07:46:37 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[158]        training's rmse: 1.48033        valid_1's rmse: 1.85294
```

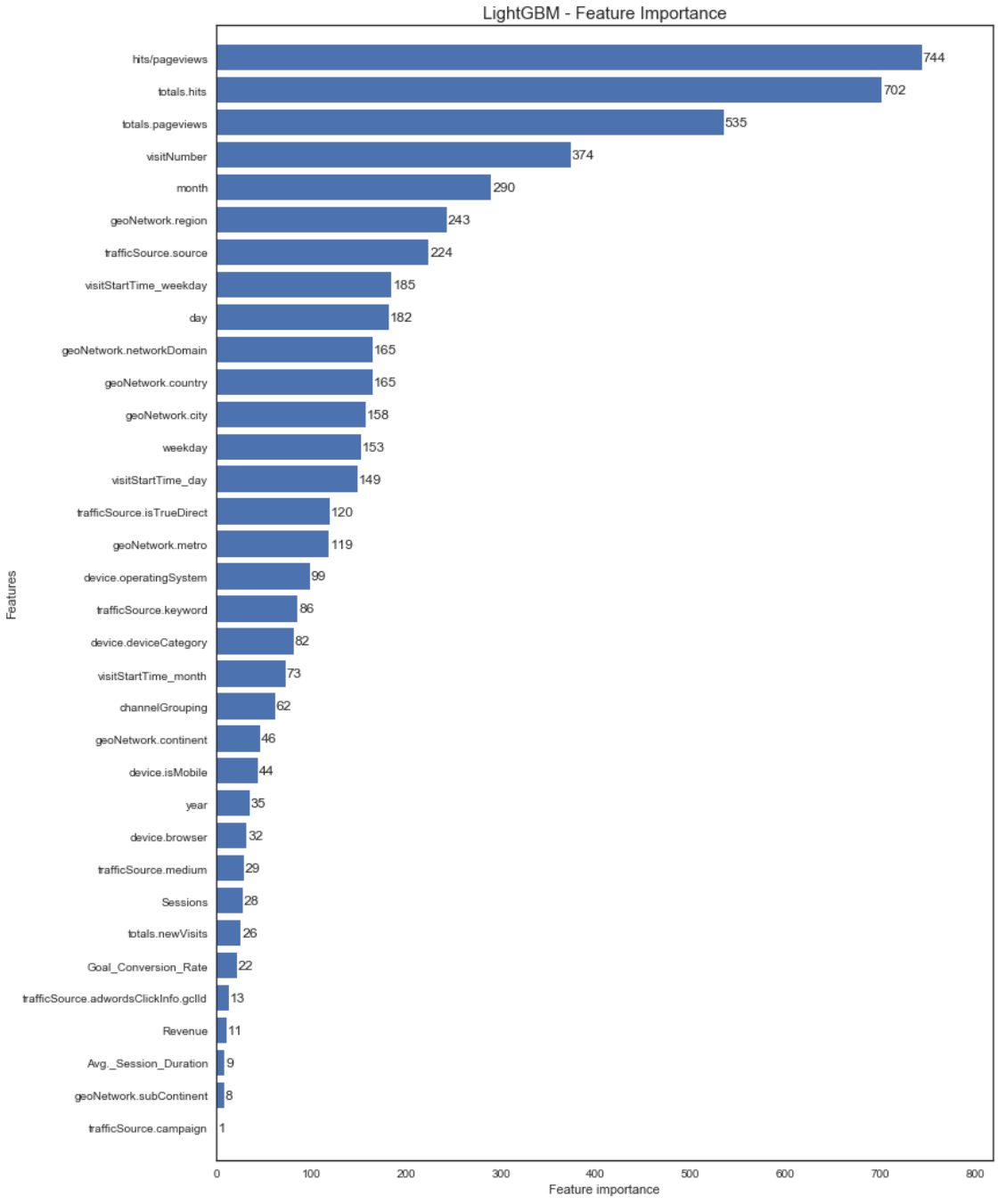
Task 7 : Permutation Test

```
In [46]: import eli5
         from eli5.sklearn import PermutationImportance

         perm = PermutationImportance(model, random_state=1).fit(X, y)
         eli5.show_weights(perm, feature_names = X.columns.tolist())
```

```
Out[46]: <IPython.core.display.HTML object>
```

Above Table shows the importance of each features.

```
In [47]: fig, ax = plt.subplots(figsize=(12,18))
         lgb.plot_importance(model, max_num_features=50, height=0.8, ax=ax)
         ax.grid(False)
         plt.title("LightGBM - Feature Importance", fontsize=15)
         plt.show()
```



```
In [48]: from sklearn.metrics import mean_squared_error
         def score(data, y):
             validation_res = pd.DataFrame(
             {"fullVisitorId": data["fullVisitorId"].values,
              "transactionRevenue": data["totals.transactionRevenue"].values,
              "PredictedLogRevenue": np.expm1(y)})

             validation_res = validation_res.groupby("fullVisitorId")["transactionRevenue", "Pr
             return np.sqrt(mean_squared_error(np.log1p(validation_res["transactionRevenue"].va
                                               np.log1p(validation_res["predictedRevenue"].value

In [49]: def RunModel():
             prediction = np.zeros(len(test_df))
             for fold_n, (train_index, valid_index) in enumerate(folds.split(X)):
                 print('Fold', fold_n, 'started at', time.ctime())
                 X_train, X_valid = X.iloc[train_index], X.iloc[valid_index]
                 y_train, y_valid = y.iloc[train_index], y.iloc[valid_index]

                 model.fit(X_train, y_train,
                         eval_set=[(X_train, y_train), (X_valid, y_valid)], eval_metric='rmse'
                         verbose=500, early_stopping_rounds=100)
             y_pred = model.predict(X_test, num_iteration=model.best_iteration_)
             prediction += y_pred
             prediction /= n_fold
             actual_score = model.best_score_
             return actual_score

In [50]: def Permute(col):
             scorelist = []
             for i in range(3):
                 X[col] = np.random.permutation(X[col])
                 score = RunModel()
                 scorelist.append(score['training']['rmse'])
             return np.mean(scorelist)
```

Let's run the permutation test by random shuffling column and compare their rmse with the actual model.

```
In [51]: selected_col = 'visitNumber'
         val = Permute(selected_col)

Fold 0 started at Tue Oct 23 07:47:05 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[164]        training's rmse: 1.52434        valid_1's rmse: 1.63032
Fold 1 started at Tue Oct 23 07:47:08 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[282]        training's rmse: 1.47319        valid_1's rmse: 1.56438
```

```
Fold 2 started at Tue Oct 23 07:47:13 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[180]        training's rmse: 1.49461        valid_1's rmse: 1.71552
Fold 3 started at Tue Oct 23 07:47:17 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[147]        training's rmse: 1.509        valid_1's rmse: 1.79711
Fold 4 started at Tue Oct 23 07:47:20 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[161]        training's rmse: 1.48146        valid_1's rmse: 1.84826
Fold 0 started at Tue Oct 23 07:47:25 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[136]        training's rmse: 1.55519        valid_1's rmse: 1.63453
Fold 1 started at Tue Oct 23 07:47:28 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[292]        training's rmse: 1.46972        valid_1's rmse: 1.56529
Fold 2 started at Tue Oct 23 07:47:33 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[173]        training's rmse: 1.50016        valid_1's rmse: 1.71716
Fold 3 started at Tue Oct 23 07:47:37 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[139]        training's rmse: 1.52048        valid_1's rmse: 1.79652
Fold 4 started at Tue Oct 23 07:47:40 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[158]        training's rmse: 1.48628        valid_1's rmse: 1.85002
Fold 0 started at Tue Oct 23 07:47:43 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[170]        training's rmse: 1.51888        valid_1's rmse: 1.63378
Fold 1 started at Tue Oct 23 07:47:46 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[344]        training's rmse: 1.45371        valid_1's rmse: 1.56635
Fold 2 started at Tue Oct 23 07:47:51 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[178]        training's rmse: 1.49562        valid_1's rmse: 1.71438
Fold 3 started at Tue Oct 23 07:47:55 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[138]        training's rmse: 1.5202        valid_1's rmse: 1.79831
```

```
Fold 4 started at Tue Oct 23 07:47:58 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[161]        training's rmse: 1.48289        valid_1's rmse: 1.85447
```

```
In [52]: selected_col = 'totals.pageviews'
         val = Permute(selected_col)
```

```
Fold 0 started at Tue Oct 23 07:48:02 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[133]        training's rmse: 1.56612        valid_1's rmse: 1.63244
Fold 1 started at Tue Oct 23 07:48:04 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[332]        training's rmse: 1.45683        valid_1's rmse: 1.56498
Fold 2 started at Tue Oct 23 07:48:09 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[199]        training's rmse: 1.48256        valid_1's rmse: 1.71385
Fold 3 started at Tue Oct 23 07:48:14 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[166]        training's rmse: 1.49274        valid_1's rmse: 1.80034
Fold 4 started at Tue Oct 23 07:48:18 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[168]        training's rmse: 1.47593        valid_1's rmse: 1.87255
Fold 0 started at Tue Oct 23 07:48:22 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[146]        training's rmse: 1.5499        valid_1's rmse: 1.64136
Fold 1 started at Tue Oct 23 07:48:26 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[292]        training's rmse: 1.47221        valid_1's rmse: 1.57087
Fold 2 started at Tue Oct 23 07:48:31 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[206]        training's rmse: 1.47845        valid_1's rmse: 1.71442
Fold 3 started at Tue Oct 23 07:48:35 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[180]        training's rmse: 1.48401        valid_1's rmse: 1.79652
Fold 4 started at Tue Oct 23 07:48:38 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
```

```
[176]        training's rmse: 1.46901        valid_1's rmse: 1.87375
Fold 0 started at Tue Oct 23 07:48:42 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[144]        training's rmse: 1.55415        valid_1's rmse: 1.63781
Fold 1 started at Tue Oct 23 07:48:45 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[274]        training's rmse: 1.48924        valid_1's rmse: 1.56767
Fold 2 started at Tue Oct 23 07:48:50 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[188]        training's rmse: 1.49369        valid_1's rmse: 1.71545
Fold 3 started at Tue Oct 23 07:48:54 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[127]        training's rmse: 1.5362        valid_1's rmse: 1.80557
Fold 4 started at Tue Oct 23 07:48:57 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[174]        training's rmse: 1.47513        valid_1's rmse: 1.87052
```

```
In [53]: selected_col = 'totals.hits'
         val = Permute(selected_col)
```

```
Fold 0 started at Tue Oct 23 07:49:01 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[97]        training's rmse: 1.70998        valid_1's rmse: 1.72146
Fold 1 started at Tue Oct 23 07:49:04 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[320]        training's rmse: 1.60418        valid_1's rmse: 1.61662
Fold 2 started at Tue Oct 23 07:49:09 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[277]        training's rmse: 1.56055        valid_1's rmse: 1.79085
Fold 3 started at Tue Oct 23 07:49:13 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[147]        training's rmse: 1.61393        valid_1's rmse: 1.88845
Fold 4 started at Tue Oct 23 07:49:17 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[177]        training's rmse: 1.56877        valid_1's rmse: 1.95868
Fold 0 started at Tue Oct 23 07:49:21 2018
Training until validation scores don't improve for 100 rounds.
```

```
Early stopping, best iteration is:
[106]        training's rmse: 1.69635        valid_1's rmse: 1.72023
Fold 1 started at Tue Oct 23 07:49:23 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[240]        training's rmse: 1.62663        valid_1's rmse: 1.62276
Fold 2 started at Tue Oct 23 07:49:28 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[279]        training's rmse: 1.56036        valid_1's rmse: 1.79378
Fold 3 started at Tue Oct 23 07:49:35 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[162]        training's rmse: 1.60252        valid_1's rmse: 1.88776
Fold 4 started at Tue Oct 23 07:49:40 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[168]        training's rmse: 1.57704        valid_1's rmse: 1.95529
Fold 0 started at Tue Oct 23 07:49:46 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[88]        training's rmse: 1.72098        valid_1's rmse: 1.72183
Fold 1 started at Tue Oct 23 07:49:49 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[395]        training's rmse: 1.57661        valid_1's rmse: 1.62339
Fold 2 started at Tue Oct 23 07:49:56 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[256]        training's rmse: 1.56976        valid_1's rmse: 1.78726
Fold 3 started at Tue Oct 23 07:50:01 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[135]        training's rmse: 1.6263        valid_1's rmse: 1.8854
Fold 4 started at Tue Oct 23 07:50:04 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[189]        training's rmse: 1.55709        valid_1's rmse: 1.95484
```

```
In [54]: def Permute(col1, col2):
             scorelist = []
             for i in range(3):
                 X[col1] = np.random.permutation(X[col1])
                 X[col2] = np.random.permutation(X[col2])
                 score = RunModel()
                 scorelist.append(score['training']['rmse'])
             return np.mean(scorelist)
```

```
In [55]: col1 = 'totals.hits'
         col2 = 'totals.pageviews'
         val = Permute(col1, col2)

Fold 0 started at Tue Oct 23 07:50:08 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[104]        training's rmse: 1.70085        valid_1's rmse: 1.721
Fold 1 started at Tue Oct 23 07:50:11 2018
Training until validation scores don't improve for 100 rounds.
[500]        training's rmse: 1.55201        valid_1's rmse: 1.61372
Early stopping, best iteration is:
[724]        training's rmse: 1.50255        valid_1's rmse: 1.61204
Fold 2 started at Tue Oct 23 07:50:21 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[241]        training's rmse: 1.57467        valid_1's rmse: 1.7898
Fold 3 started at Tue Oct 23 07:50:25 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[140]        training's rmse: 1.61959        valid_1's rmse: 1.88964
Fold 4 started at Tue Oct 23 07:50:28 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[265]        training's rmse: 1.52336        valid_1's rmse: 1.95775
Fold 0 started at Tue Oct 23 07:50:33 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[97]        training's rmse: 1.70995        valid_1's rmse: 1.72207
Fold 1 started at Tue Oct 23 07:50:35 2018
Training until validation scores don't improve for 100 rounds.
[500]        training's rmse: 1.5515        valid_1's rmse: 1.62555
Early stopping, best iteration is:
[617]        training's rmse: 1.52478        valid_1's rmse: 1.62397
Fold 2 started at Tue Oct 23 07:50:44 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[202]        training's rmse: 1.58897        valid_1's rmse: 1.79023
Fold 3 started at Tue Oct 23 07:50:48 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[154]        training's rmse: 1.60797        valid_1's rmse: 1.88654
Fold 4 started at Tue Oct 23 07:50:51 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[173]        training's rmse: 1.57087        valid_1's rmse: 1.95863
Fold 0 started at Tue Oct 23 07:50:56 2018
Training until validation scores don't improve for 100 rounds.
```

```
Early stopping, best iteration is:
[101]        training's rmse: 1.70139        valid_1's rmse: 1.71939
Fold 1 started at Tue Oct 23 07:50:59 2018
Training until validation scores don't improve for 100 rounds.
[500]        training's rmse: 1.54756        valid_1's rmse: 1.61417
Early stopping, best iteration is:
[642]        training's rmse: 1.51455        valid_1's rmse: 1.61225
Fold 2 started at Tue Oct 23 07:51:08 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[229]        training's rmse: 1.57822        valid_1's rmse: 1.79169
Fold 3 started at Tue Oct 23 07:51:13 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[151]        training's rmse: 1.61116        valid_1's rmse: 1.88223
Fold 4 started at Tue Oct 23 07:51:16 2018
Training until validation scores don't improve for 100 rounds.
Early stopping, best iteration is:
[154]        training's rmse: 1.58234        valid_1's rmse: 1.95829
```