# Google Analytics Customer Revenue Prediction
# Predict how much GStore customers will spend

**Problem Description**

The goal of the project is to predict how much money customers will spend at Gstore, an online retail store. We apply regression algorithms to predict the value of the natural log of the sum of the total revenue of all transactions per user.

$Yuser = \sum transactionuseri$

$Targetuser = \ln (Yuser+1)$

**Introduction**

The 80-20 rule states that 80 percent of a business' revenue comes from 20 percent of the customers. Knowledge of customer profile and their spending behavior can be valuable to a business by helping in formulating and implementing efficient and effective marketing strategies. Gstore, an online retail store, collected transactional data for each visit to the store by customers. The goal is to utilize the data to build a model that will predict the total revenue per customer. The following sections details the steps that we undertook to build the model. These steps are data preprocessing, model learning, results and discussion.

**Preprocessing**

The original dataset comprises of 12 attributes, namely fullVisitorId, channelGrouping, date, device, geoNetwork, sessionId, socialEngagementType, totals, trafficSource, visitId, visitNumber and visitStartTime. Each record in the dataset corresponds to a single visit to the store.
Some of the attributes had nested attributes in json format; these are device, geoNetwork, totals and trafficSource. We used the 'python json package*' to flatten out those attributes. The resulting dataset now had 50 attributes. Some of the features had nominal values, and we converted the values to ordinal where applicable. We then ensured all the values for all features were converted to numeric by the use of LabelEncoder and pandas.get_dummies that convert categorical variable into dummy/indicator variables. For the LabelEncoder we used those attributes which had large number of unique values, which are

1. networkDomain 28064
2. gclId 17774

3. keyword 3659
4. referralPath 1475
5. city 649
6. visitNumber 384
7. source 380
8. region 376
9. date 366
10. hits 274
11. country 222
12. pageviews 214
13. metro 94
14. browser 54
15. adContent 44
16. subContinent 23
17. operatingSystem 20
18. campaign 10

We then generated index from the weekdays, month, year, day and from timestamps to see if we can visualize any pattern. We removed some attributes that do not contribute in predicting the target value. These attributes had datatypes as boolean, object and datetime. The next step involved addressing the missing values by filling in with the mean of the values in that attribute. A record in the dataset represents a visit to the store by the customer. However, the goal is to predict the total revenue per customer. We therefore aggregated all the transactions for each customer to get the total revenue. For all the other attributes, the aggregation resulted in 'sub attributes' and these were the max, min, count, sum and mean for all the transaction records for each customer. Each row in the resulting dataset now represented the aggregated transactions per customer. At this stage, the data is ready to be used for model building. We saved the preprocessed training data to applications.csv file. The preprocessed testing data is saved as test_agg.csv.

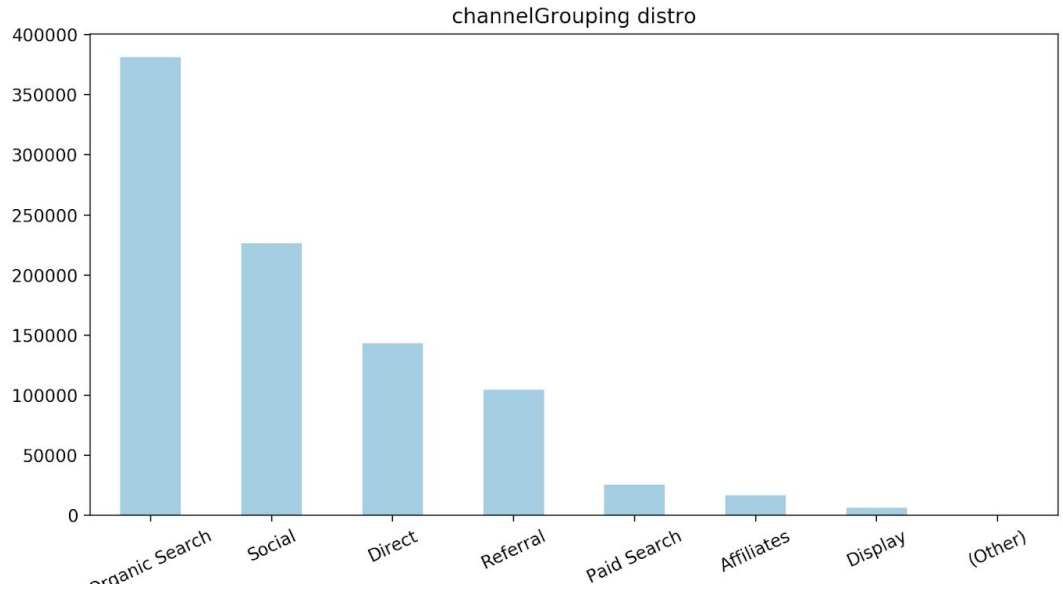
**Model Learning**

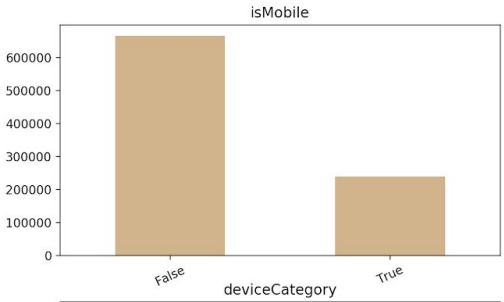
Since the target value for the dataset is continuous, we employed regression to build the model. We applied three different regression algorithms and these are Random Forest regression, Decision Tree Regressor and RANSAC regressor. Since different algorithms perform differently, the rationale behind using different regression algorithms is to ensure that we build the model that best reflects the patterns in the data. In addition to directly applying the regression algorithms on the dataset, we also applied dimensionality reduction techniques prior to using regression. The dataset provided is already is already split into training and testing sets. However we merged both the sets to get the aggregate values like mean, min, max, count and sum. The reason behind getting the aggregate values was to analyze each customer and get

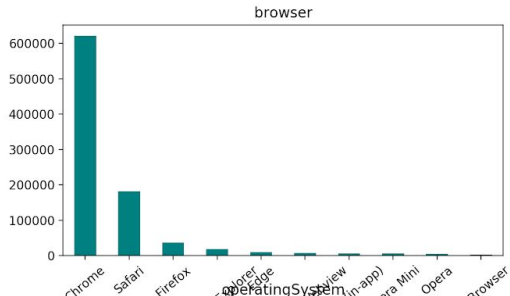
predictions on customer level. We performed regression six times for the aggregated dataset with and without dimensionality reduction technique to visualise the effects in accuracy scores.

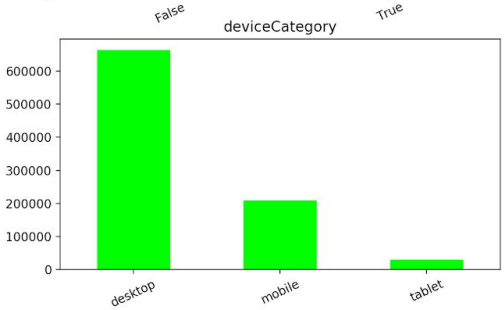
## Results

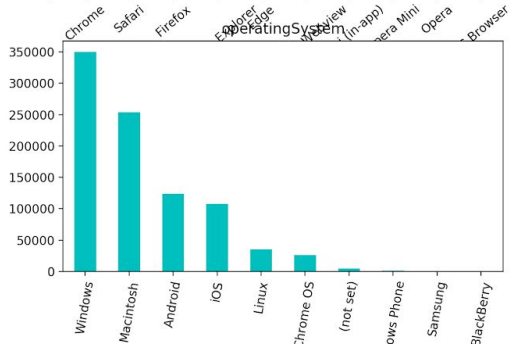
Prior to applying the regression algorithms on the dataset, we performed analysis of the distributions of a select attributes. These attributes are channelGrouping, deviceCategory, Operating System, Browser and Region. We also performed an analysis on how the numberOfVisits varies with the totalRevenue over time. The visualizations of the distributions of these attributes give insight into the predictive power of the attributes. The diagrams below show the visualizations of the distributions.



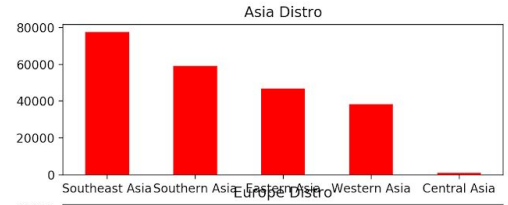

isMobile
600000
500000
400000
300000
200000
100000
0
False
True

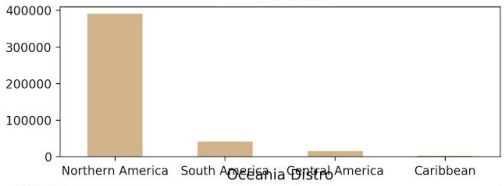

browser
600000
500000
400000
300000
200000
100000
0
Chrome
Safari
Firefox
Opera
Opera Mini
Opera Browser

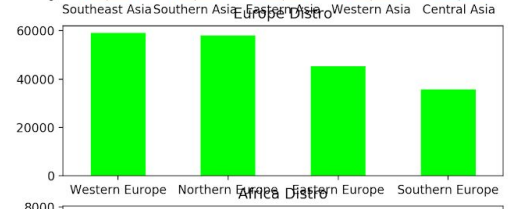

deviceCategory
600000
500000
400000
300000
200000
100000
0
desktop
mobile
tablet

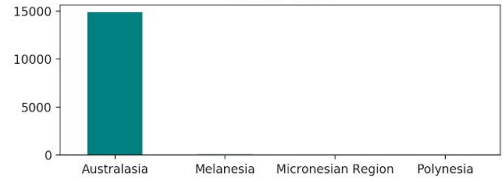

operatingSystem
350000
300000
250000
200000
150000
100000
50000
0
Windows
Macintosh
Android
iOS
Linux
(not set)
Samsung
BlackBerry

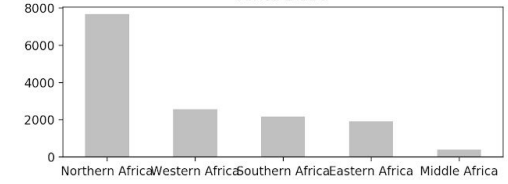

Global Distributions
400000
300000
200000
100000
0
Americas
Asia
Europe
Oceania
Africa
(not set)

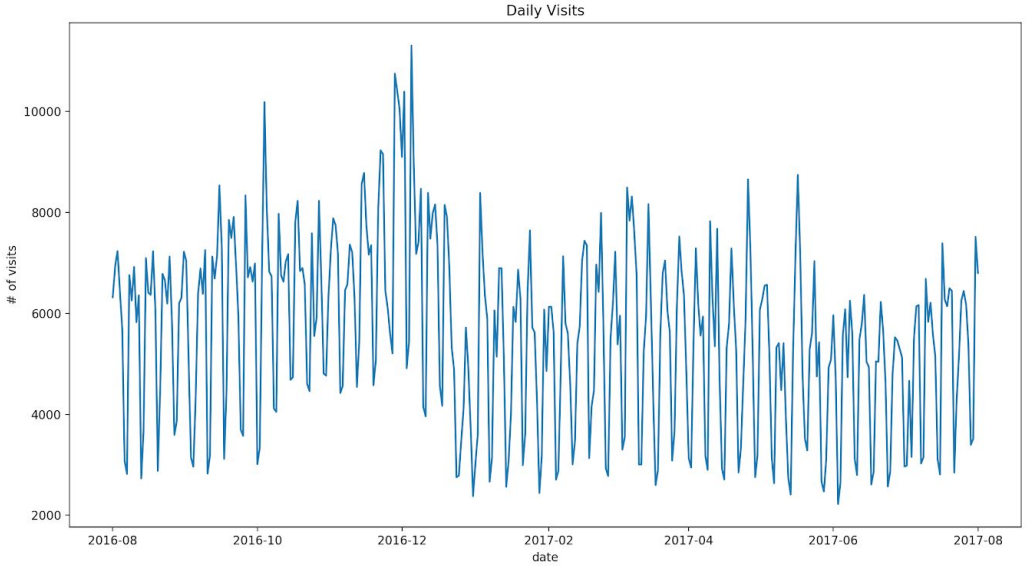

Asia Distro
80000
60000
40000
20000
0
Southeast Asia
Southern Asia
Eastern Asia
Western Asia
Central Asia

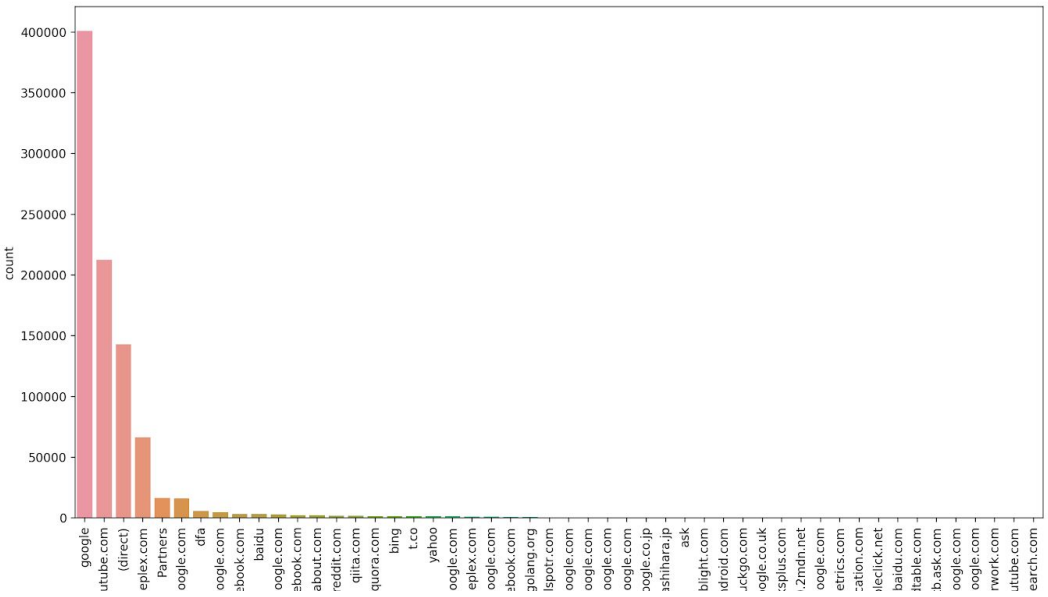

America Distro
400000
300000
200000
100000
0
Northern America
South America
Central America
Caribbean

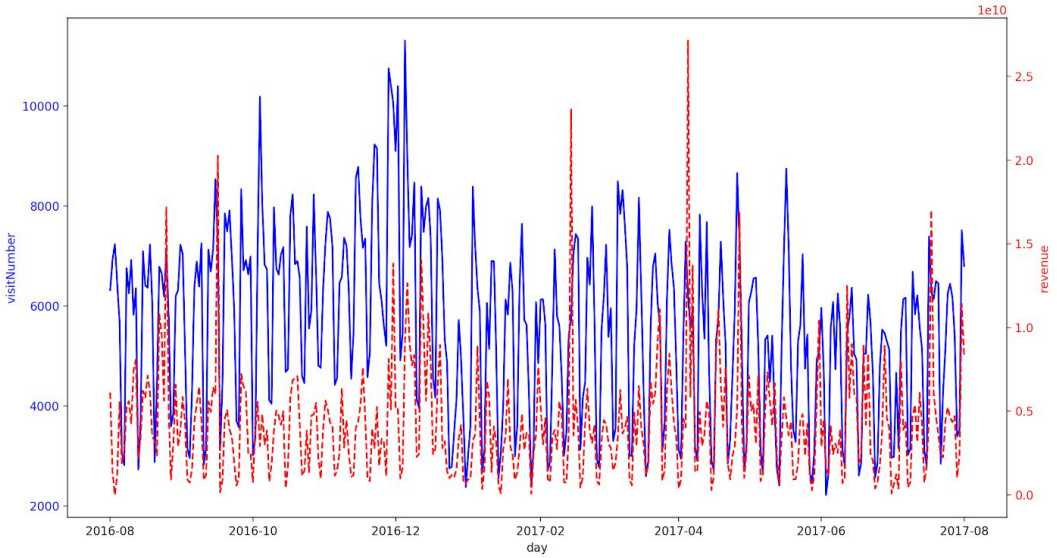

Europe Distro
60000
40000
20000
0
Western Europe
Northern Europe
Eastern Europe
Southern Europe


Oceania Distro
15000
10000
5000
0
Australasia
Melanesia
Micronesian Region
Polynesia


Africa Distro
8000
6000
4000
2000
0
Northern Africa
Western Africa
Southern Africa
Eastern Africa
Middle Africa


Daily Visits
# of visits
10000
8000
6000
4000
2000
2016-08
2016-10
2016-12
2017-02
2017-04
2017-06
2017-08
date


count
400000
350000
300000
250000
200000
150000
100000
50000
0
google
(direct)
Partners
bing
t.co
yahoo
ask

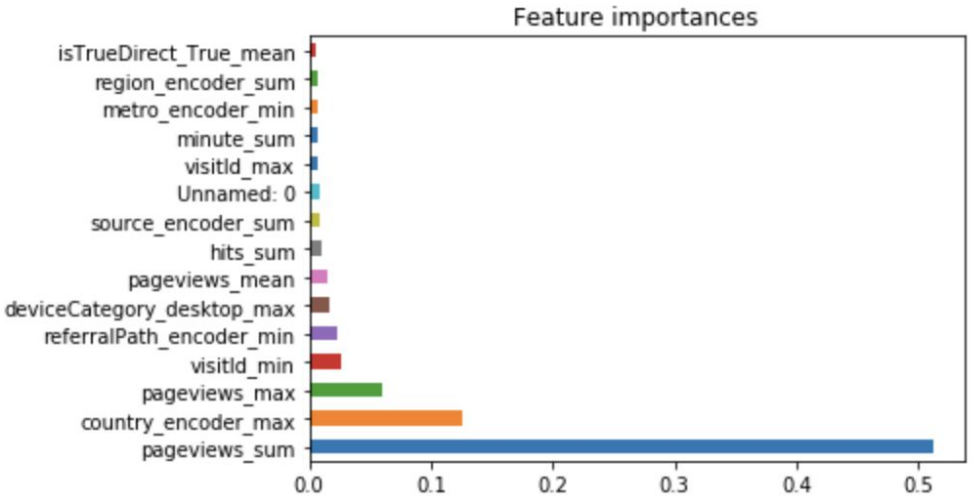



**Regression results**

The diagrams below show the results obtained after running each regression algorithm. The results include the Mean Squared Error(MSE) and the R2 values. There is also a visualization of the ranking of the attributes using feature importance.

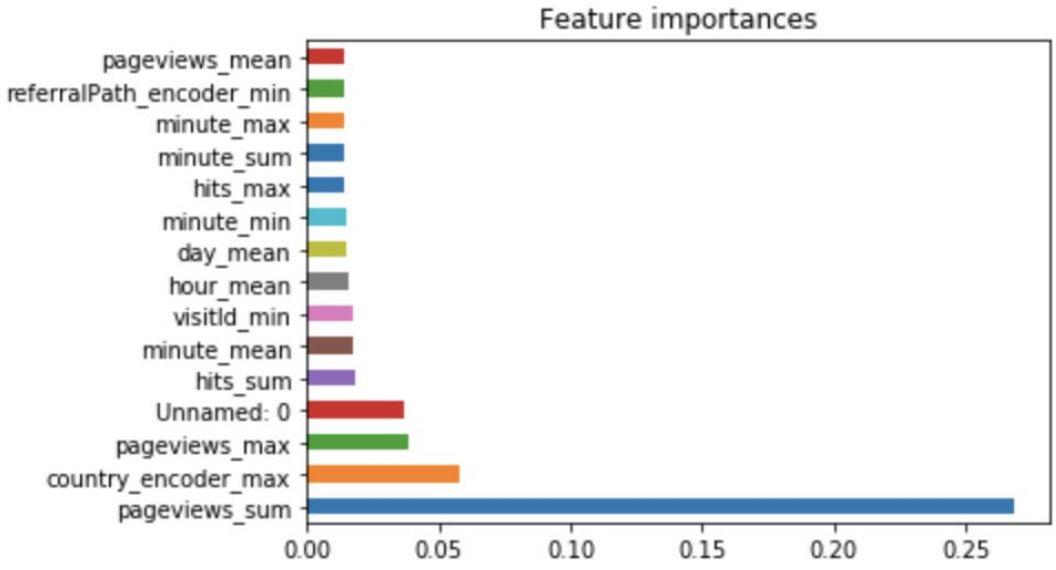
**DecisionTreeRegressor**

--without dimensionality reduction technique--
DecisionTreeRegressor
MSE train: 2.218, test: 3.042
R^2 train: 0.491, test: 0.325
--- Time taken is 24.3293240070343 seconds ---



**RandomForestRegressor**

RandomForestRegressor
MSE train: 0.523, test: 2.977
R^2 train: 0.880, test: 0.340
--- Time taken is 124.93984007835388 seconds ---




**RANSACRegressor**

RANSAC Regressor
MSE train: 4.419, test: 4.574
R^2 train: -0.014, test: -0.014
--- Time taken is 32.525099992752075 seconds ---

We applied principal component analysis as the dimensionality reduction. We re-ran the regressors using principal components with a total explained variance ratio of 80 %. The diagram below shows the results obtained for the DecisionTreeRegressor and RandomForestRegressor.

```
RandomForestRegressor
MSE train: 2.935, test: 4.471
R^2 train: 0.326, test: 0.008
--- Time taken is 8.986267805099487 seconds ---
DecisionTreeRegressor
MSE train: 4.031, test: 4.175
R^2 train: 0.075, test: 0.074
--- Time taken is 0.2446222305297851 6 seconds ---
```

**Discussion**

From the preliminary analysis of the distribution of attributes, we decided that a simple linear model might not be the best model that accurately captures the pattern in the dataset. This informed our decision to use the DecisionTreeRegressor and RandomforestRegressor to build the model. However, we still used RANSAC Regressor to compare the results with the other 2 regressors. From the results above, RANSAC has the highest MSE and lowest R2 values. This confirms that a linear model is not the best fit for this particular dataset. The DecisionTreeRegressor has relatively consistent MSE and R2 values for both the training and testing datasets. The time taken by DecisionTreeRegressor is also lowest in comparison to other regression algorithms. As expected, RandomForestRegressor has the highest running time because it runs the base regressor several times.The RandomForestRegressor has the best MSE and R2 values for the training set, but the metrics are relatively worse for the testing set. These results indicate that there might be overfitting. In fact, these were the best results obtained after we adjusted the parameters, in particular the max_depth of the tree. We concluded that lowering the max_depth parameter any further does not address the overfitting issue. From the results above, we also conclude that applying principal component analysis does not improve the accuracy of the model. In light of these results, we based our final model for the project on the RandomForestRegressor since it had the best MSE and R2 values on the testing set.

**Limitations**

We tried to implement the K-fold cross-validation as well to evaluate performance of different models but couldn't get any result out of it. We concluded that it may be due to large size of the dataset.