

Springboard DSC Capstone Project 2

Google Analytics Customer Revenue Prediction

Neha Jain Surana

Nov 2018

Introduction

Client

Data and its acquisition

Data Wrangling

- Parse Json columns
- Delete non useful columns
- Fill up missing values
- Check validity of geo related data

Data Exploration

- Analyze the target variable
- Analyze the visitor Ids
- Analyze the date and visitStartTime
- Analyze visitNumber - visit number of a visitor
- Analyze channelGrouping
- Analyze device fields
- Analyze geo fields
- Analyze new visits
- Analyze hits
- Statistical Inference

Data Modeling

- Theory and Approach followed
- Overfitting and Underfitting
- Model evaluation criteria
- Data and code organization
- First (Baseline) Model
 - Baseline Results
 - Held out test set
 - Linear Regression

Model Comparisons

Recommendations for the client

Conclusions and Future Analysis

Introduction

The primary purpose of any business is to make profits for its owners and stakeholders. Also, the 80/20 rule has proven true for many businesses—only a small percentage of customers produce most of the revenue.

For an e-store like Google merchandise store, it is nothing different. As such, marketing teams are challenged to make appropriate investments in promotional strategies. But for that, they need as accurate a prediction as possible, of the revenue per customer.

Although, the main prediction is not for future, i.e. we predict revenues for the visits already done. But there might be a second phase of the project where we predict for future dates.

Client

Here the client is the company Google and its specific e-store called Google Merchandise Store (GStore). We hope to provide highly accurate revenue predictions per customer visiting the GStore. The outcome should be more actionable operational changes and a better use of marketing budgets. Also, it would serve as an evidence for those companies who choose to use data analysis on top of GA data, because the current data comes from the same.

Data and its acquisition

The data is provided by the client in the form of csv file. There is some additional test data that can be used to evaluate and report the algorithm success. The main data file has around 900 thousand rows corresponding to each customer visit to the GStore. There are multiple columns which contain JSON blobs of varying depth. In one of those JSON columns, totals, the sub-column transactionRevenue contains the revenue information we are trying to predict.

Loading the data will not be very simple because the json blobs need to be parsed. I am using the `read_csv` function provided by the `python-pandas` library to load the data into `dataframe`. `Json converters` are used to parse the `Json` fields. The result is a table like structure that can be used for processing.

Some of the Data Fields:

- **fullVisitorId** - A unique identifier for each user of the Google Merchandise Store.
- **channelGrouping** - The channel via which the user came to the Store.
- **date** - The date on which the user visited the Store.
- **device** - The specifications for the device used to access the Store.
- **geoNetwork** - This section contains information about the geography of the user.
- **sessionId** - A unique identifier for this visit to the store.
- **socialEngagementType** - Engagement type, either "Socially Engaged" or "Not Socially Engaged".
- **totals** - This section contains aggregate values across the session.
- **trafficSource** - This section contains information about the Traffic Source from which the session originated.
- **visitId** - An identifier for this session. This is part of the value usually stored as the `utmb` cookie. This is only unique to the user. For a completely unique ID, you should use a combination of `fullVisitorId` and `visitId`.

- **visitNumber** - The session number for this user. If this is the first session, then this is set to 1.
- **visitStartTime** - The timestamp (expressed as POSIX time).
- **hits** - This row and nested fields are populated for any and all types of hits. Provides a record of all page visits.
- **customDimensions** - This section contains any user-level or session-level custom dimensions that are set for a session. This is a repeated field and has an entry for each dimension that is set.
- **totals** - This set of columns mostly includes high-level aggregate data.

Data Wrangling

Metadata about number of unique and null values for each column, helps to decide what clean up and preprocessing of the data is needed.

Parse Json columns

Columns ('device', 'geoNetwork', 'totals', 'trafficSource') were found to be json blobs. Parsed these columns and added them back to the train/test data. Train data has 55 columns and 903653 rows. The data types are: bool(1), int64(4), object(50)

Delete non useful columns

Columns with 1 unique value and zero null value are not useful in any way and we delete them.

		data type	unique values	null values
	socialEngagementType	object	1	0
	device.browserSize	object	1	0
	device.browserVersion	object	1	0
	device.flashVersion	object	1	0
	device.language	object	1	0
	device.mobileDeviceBranding	object	1	0
	device.mobileDeviceInfo	object	1	0
	device.mobileDeviceMarketingName	object	1	0
	device.mobileDeviceModel	object	1	0
	device.mobileInputSelector	object	1	0
	device.operatingSystemVersion	object	1	0
	device.screenColors	object	1	0
	device.screenResolution	object	1	0
	geoNetwork.cityId	object	1	0
	geoNetwork.latitude	object	1	0
	geoNetwork.longitude	object	1	0
	geoNetwork.networkLocation	object	1	0
	totals.visits	object	1	0
trafficSource.adwordsClickInfo.criteriaParameters		object	1	0

Fill up missing values

The columns having only one unique value other than nulls, can be treated as boolean. The null can be treated as True or False depending on the other value present. But from the perspective of fitting the model it doesn't matter.

Column	: Old values	: New value
s		
totals.bounces	: ['1' nan]	: [1 0]
totals.newVisits	: ['1' nan]	: [1 0]
trafficSource.adwordsClickInfo.isVideoAd	: [nan False]	: [1 0]
trafficSource.campaignCode	: [nan '11251kjhkvahf']	: [1 0]
trafficSource.isTrueDirect	: [nan True]	: [0 1]

The columns having multiple values and nulls, have to filled up for missing values.

Column	: Unique values	: Null values
totals.pageviews	: 213	: 100
(Very few null values. null values might indicate no page view. so fill up m issing by '0's.)		
totals.transactionRevenue	: 5332	: 892138
(Huge number of null values. Null revenue means '0', so replace by it.)		
trafficSource.adwordsClickInfo.page	: 8	: 882193
(Huge number of null values. So might not be important variable. It's an in t, so convert it to int data type. Filling up null with '0's.)		
trafficSource.adContent	: 44	: 892707
trafficSource.adwordsClickInfo.adNetworkType	: 2	: 882193
trafficSource.adwordsClickInfo.slot	: 2	: 882193
trafficSource.keyword	: 3659	: 502929
trafficSource.referralPath	: 1475	: 572712
(Huge number of null values. So might not be important variable. It's a stri ng value. Filling up nulls by 'Unknown')		
trafficSource.adwordsClickInfo.gclID	: 17774	: 882092
(It's alpha numeric strings. Filling up nulls by 'Unknown')		

Check validity of geo related data and clean up if needed

- Find out unique geo combinations.
- There are values like '(not set)' and 'not available in demo dataset'. We can't do much about them.

- Divided the 6 geo columns into 2 parts. The unique combinations of country, sub-continent, continent seems to be fine. There are 222 such rows. Eyeballing them also looks good.
- The other unique combinations for city, metro, region seems to have some invalid data. Same city is associated with different regions. This can happen when same city name occurs in two places. So we further validate such combinations.
- We found that many of such combinations don't have valid match with country. We delete 3765 such rows. For e.g.

geoNetwork.city	geoNetwork.metro	geoNetwork.region	geoNetwork.country	geoNetwork.subContinent	geoNetwork.continent	counts
Abu Dhabi	(not set)	Abu Dhabi	United States	Northern America	Americas	1
Amsterdam	(not set)	North Holland	United States	Northern America	Americas	7
Amsterdam	(not set)	North Holland	Ireland	Northern Europe	Europe	2
Amsterdam	(not set)	North Holland	United Kingdom	Northern Europe	Europe	2
Amsterdam	(not set)	North Holland	Norway	Northern Europe	Europe	1
Amsterdam	(not set)	North Holland	Germany	Western Europe	Europe	1
Ankara	(not set)	Ankara	Ireland	Northern Europe	Europe	1
Ann Arbor	Detroit MI	Michigan	Ireland	Northern Europe	Europe	2
Ann Arbor	Detroit MI	Michigan	France	Western Europe	Europe	1
Ashburn	Washington DC (Hagerstown MD)	Virginia	China	Eastern Asia	Asia	2
Atlanta	Atlanta GA	Georgia	Puerto Rico	Caribbean	Americas	2

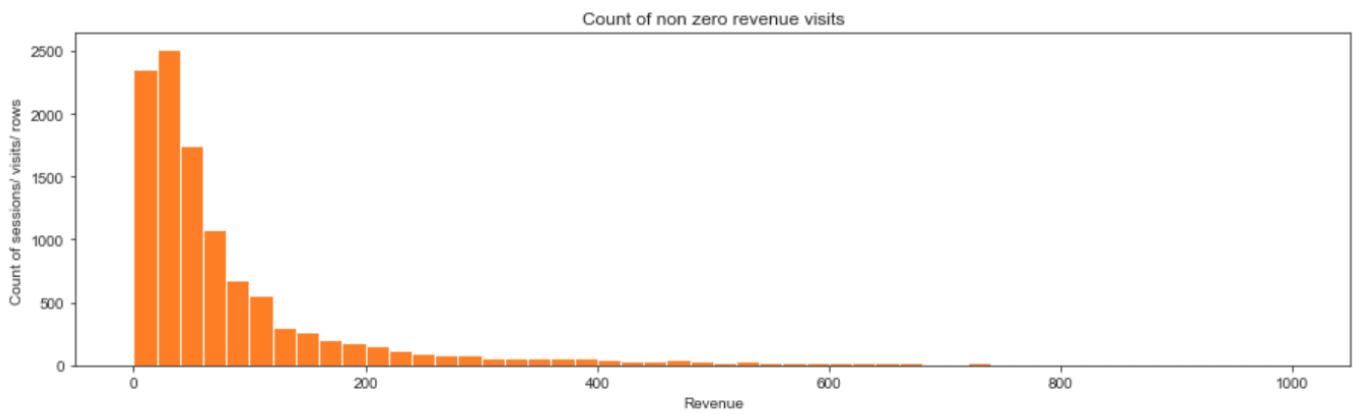
Data Exploration

Let us explore all the fields and their relation to the target variable.

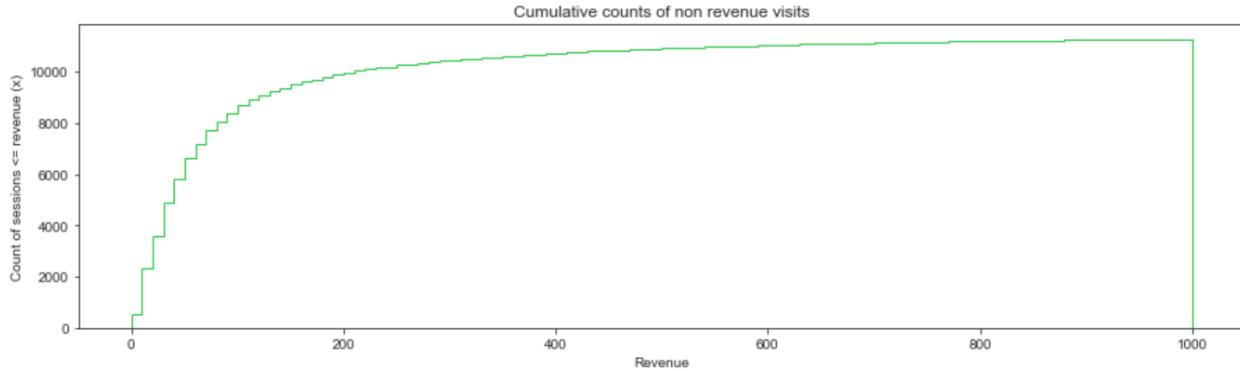
Analyze the target variable

The given transaction revenues data is actually the original revenue multiplied by a million. So actual revenues are millionth of the given numbers. The reason behind is to keep the precision of the numbers upto certain length, so that the values are not rounded while being transformed in multiple systems.

The percent of revenue generating sessions is only 1.27%. And that means most of the visits don't generate any revenue.



This gives a rough picture that there are large number of small valued transactions and very few large valued.



This picture better shows that most of the non zero transactions have revenue of less than 200. In fact we see below that 90% of all revenues are less than 279. The median revenue is around \$50.

Few things noticed by comparing revenue generating visits vs no-revenue visits. So these all could serve as good features as they have differences in values for revenue vs no-revenue visits.

- Bounces are all zeros for revenue generating visits
- Campaign code are all ones
- Maximum visitNumber is 315 vs 395
- Minimum pageview is 2 vs 0
- Maximum pageviews is 469 vs 429
- Maximum adwordclickinfo page is 1 vs 14

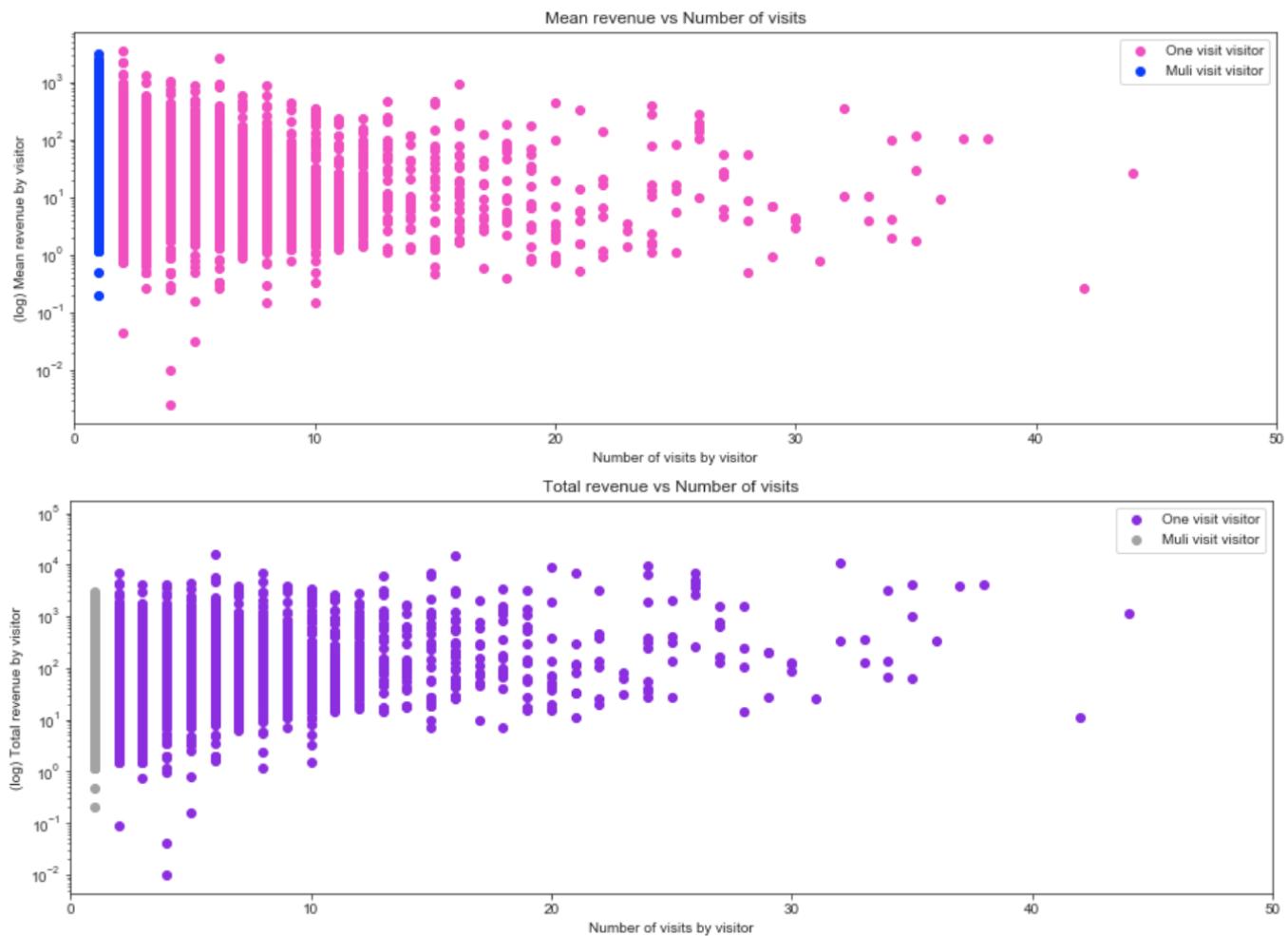
Analyze the visitorids

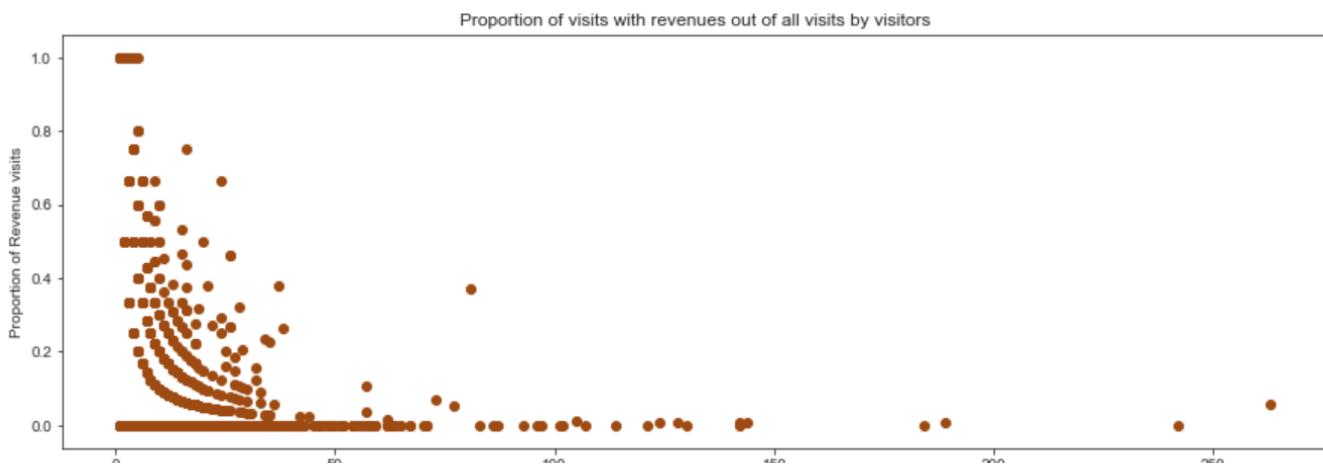
There are 714167 unique visitor ids for the total of 903653 visits. This tells that there will be huge number of visitors who visit the store only once.

Percentage of unique users for:

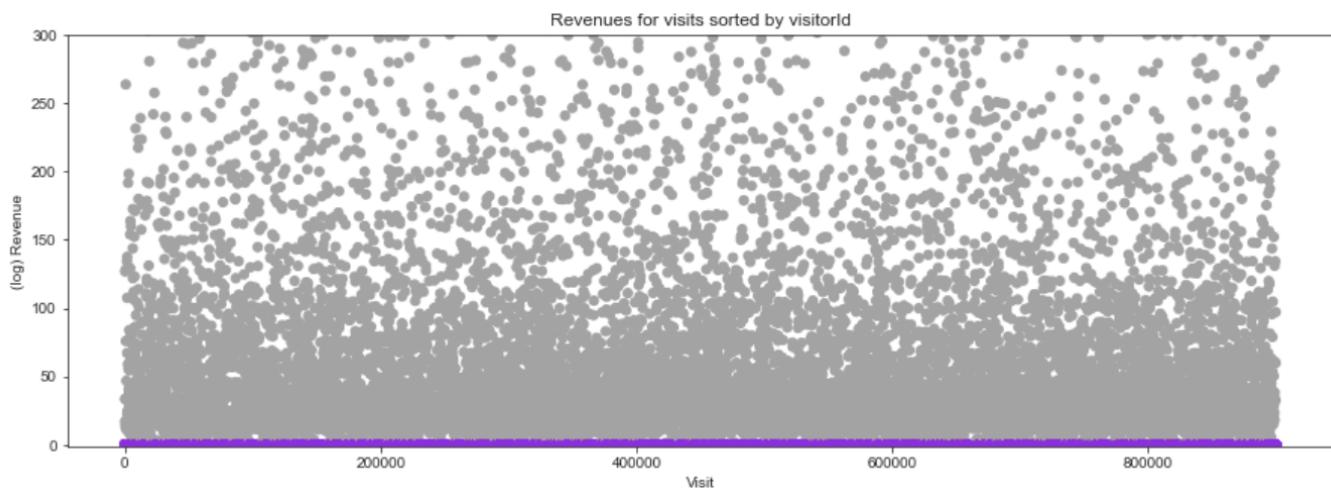
- revenue visits 88.25992150021806
- non revenue visits 80.88129190712083
- all visits 80.25376491296694

Percent for unique visitors is slightly more for revenue visits as compared to non revenue visits which is little weird because we would expect more visits by some visitor when they buy something.





Above all analysis means that when the number of visits by a visitor increases it necessarily doesn't mean that more revenue will be generated



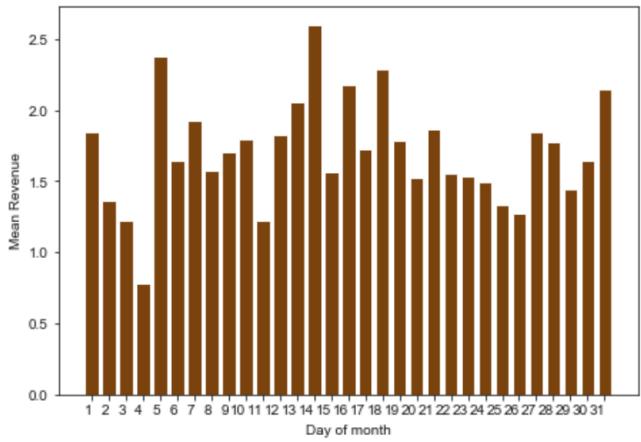
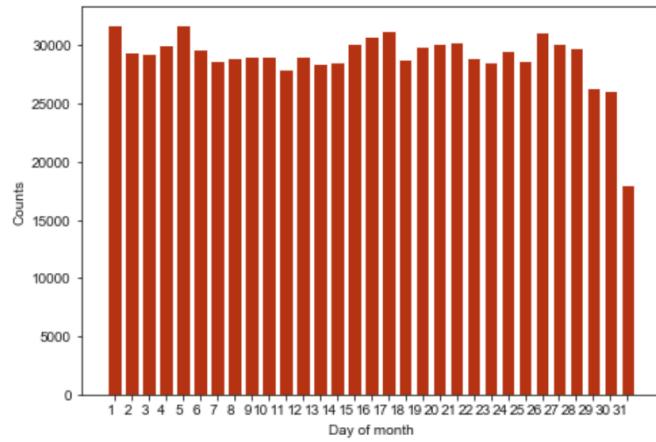
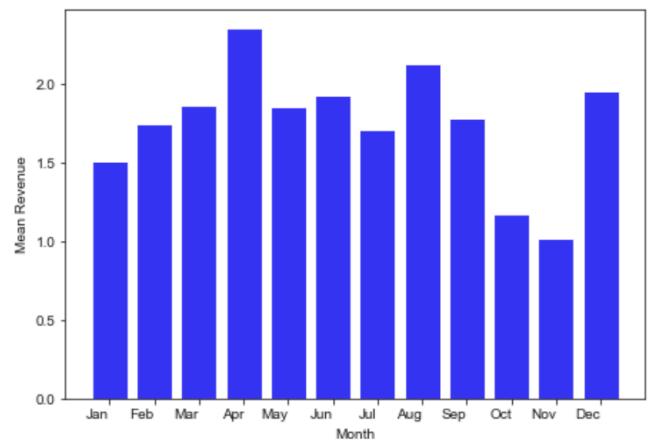
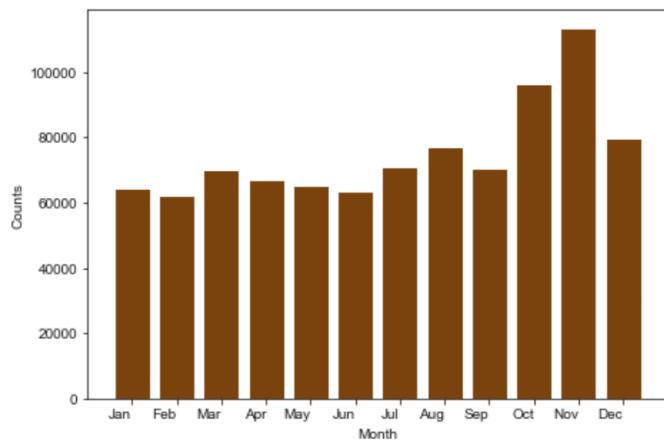
Understanding about visitorids and revenues

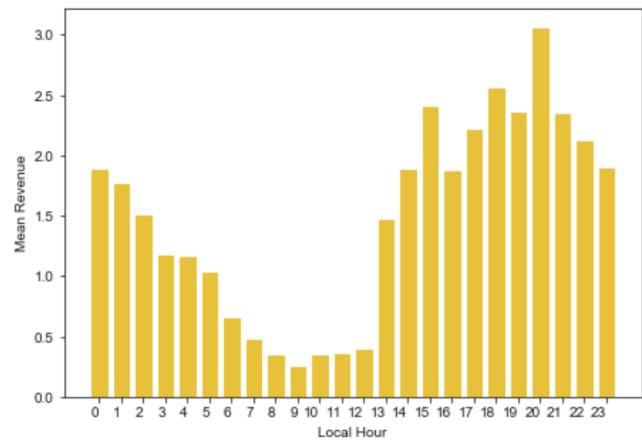
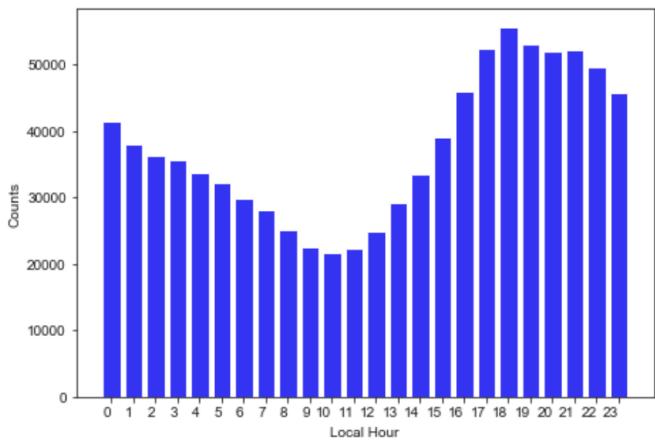
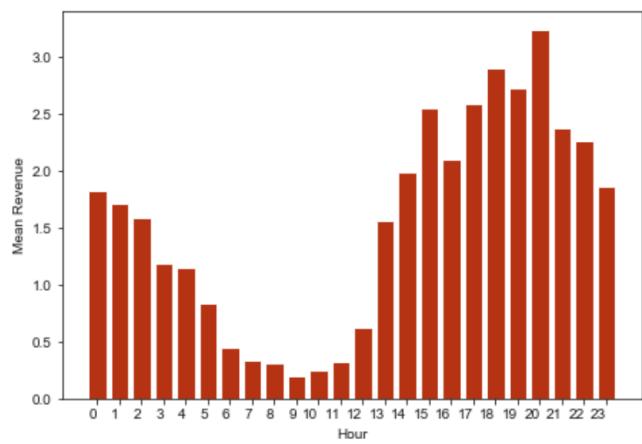
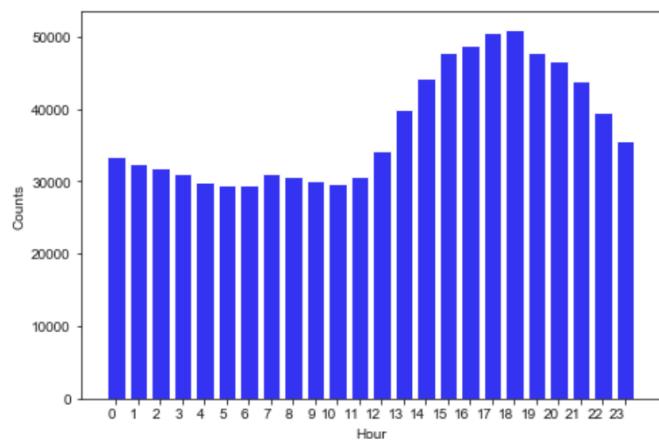
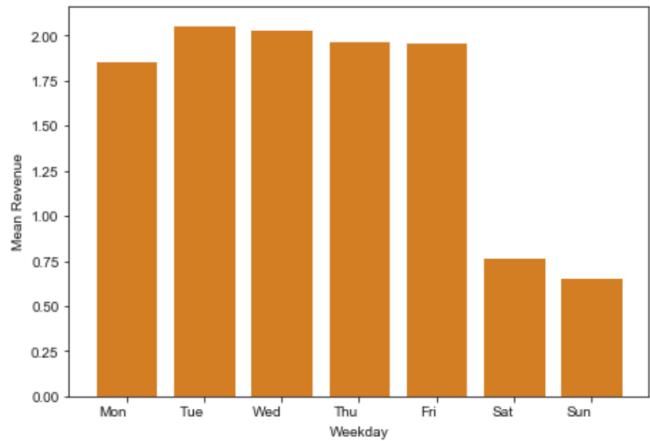
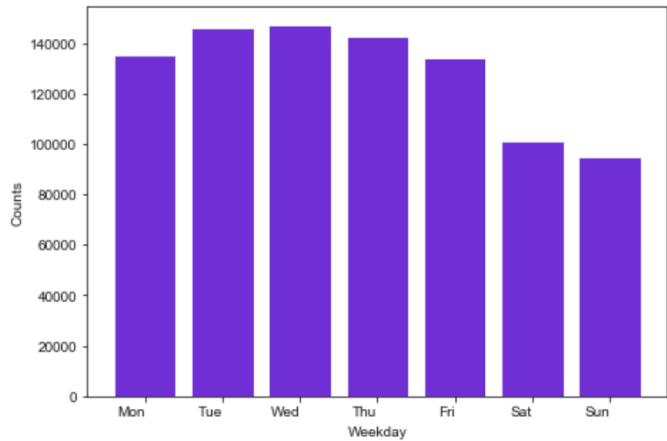
- Revenues are evenly spread for all the visits
- Proportion of revenue visits decreases as the number of visits by the visitor increases. That is, if a visitor visits the site multiple times, it doesn't mean he/she is buying more and more.
- Total revenue increases slightly if the number of visits increase
- Some visitors have generated revenue multiple times in multiple visits. Maximum is 30 revenue transactions by a visitor.

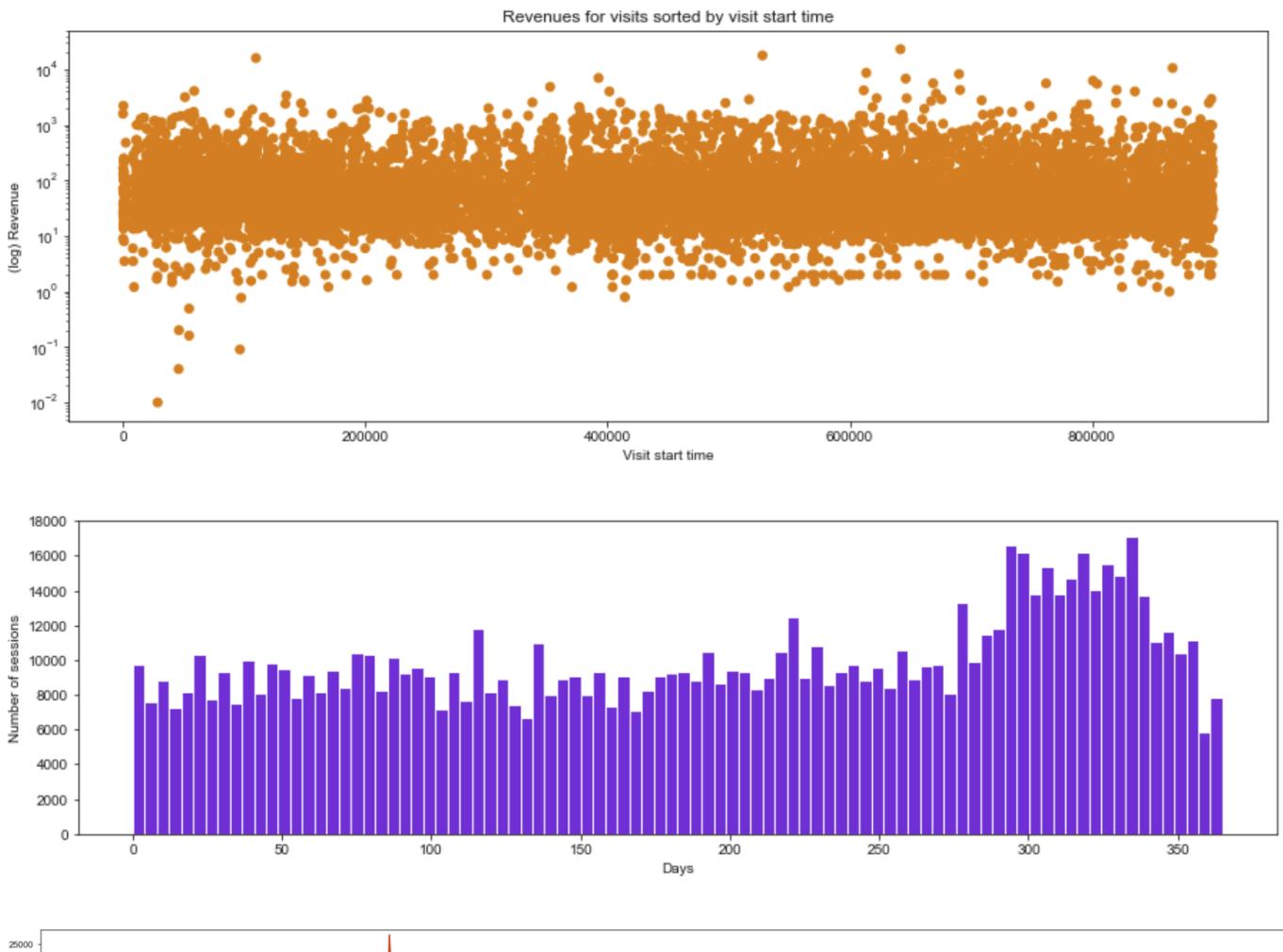
Analyze the date and visitStartTime

We parse the Date and VisitStartTime into following components and visualize them with respect to the revenue. Also, show below are some interesting charts.

- Year
- Month
- Day (of the month)
- Day (of the year, Jan 1 is day 1)
- Weekday
- Hour
- Local hour (local hour based on timezone)







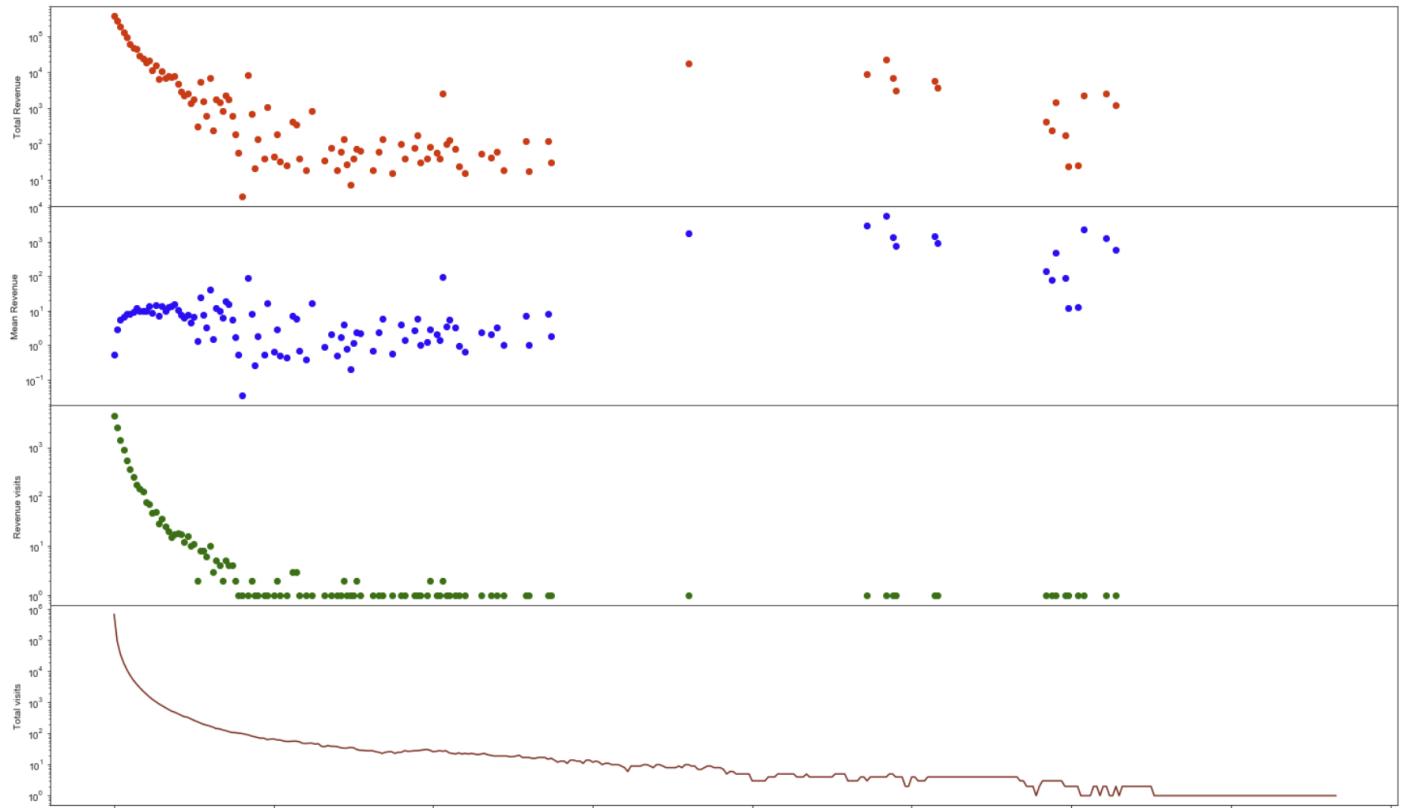
Understanding about date and visitStartTime

- The visit counts increase in the festival months. But the mean revenue fluctuates in those months.
- Both the visit counts and mean revenues drop on weekends. May be people are busy doing other stuff then buying things online.
- Assuming large portion of visitors are from North America, the 24 hour pattern shows mean revenue increase in day time, maximizes around 8pm and then drops.
- We dont see any pattern for day of the month mean revenue. And the visit counts are pretty uniform. It means all days of the month are same as far as visiting the GStore is considered.
- We see weekly patterns for visits and therefore mean and total revenues. And although the visits surge around 50-120 days the mean revenues surge around 120-150 days.

Analyze visitNumber - visit number of a visitor

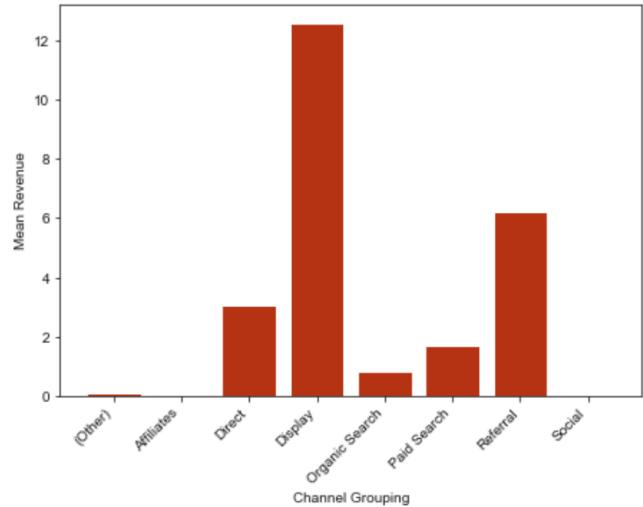
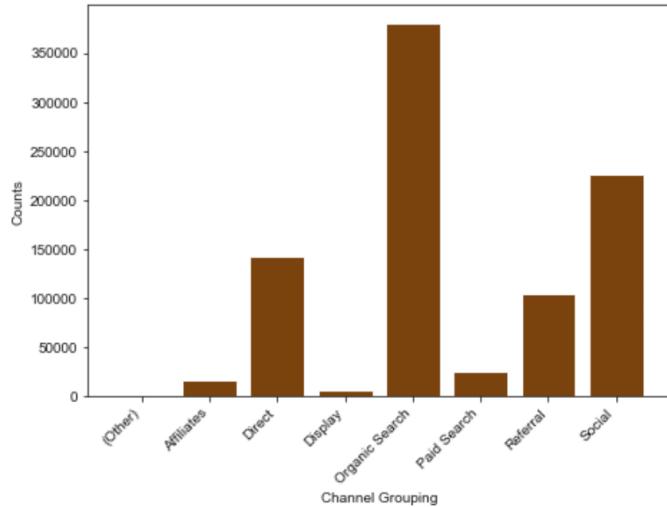
Understanding about visitNumber

- Most of the visitors visit less than 5 times. For e.g. number of visit first or second time is around 1 million and it drops very fast. Very few visitors visit for more than 50th time or more.
- Same is the case for visits that produce revenue.
- It is little strange that there are very few revenue visits such that it is 150-250th visit for that visitor.
- Mean revenues are uniformly distributed from 0.1 to 10 for visit numbers 1 to 150. And it is significantly higher for greater visit numbers. It indicates that often more number of visits are needed before high valued purchase.

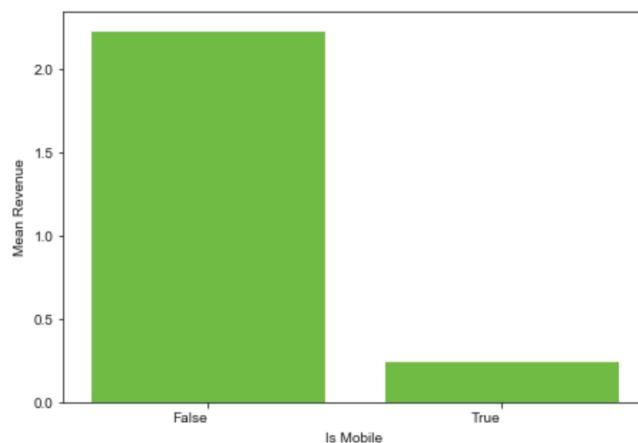
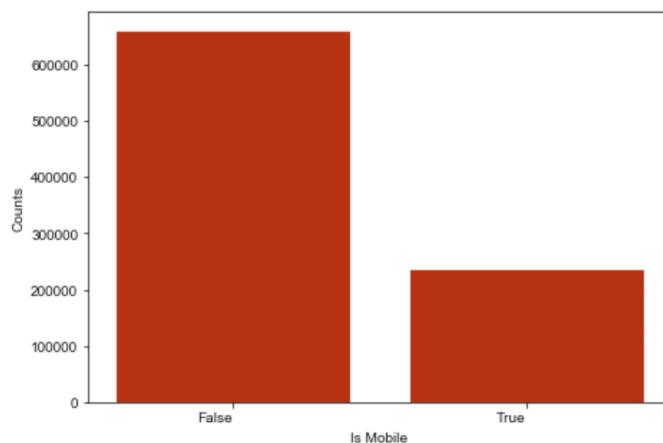
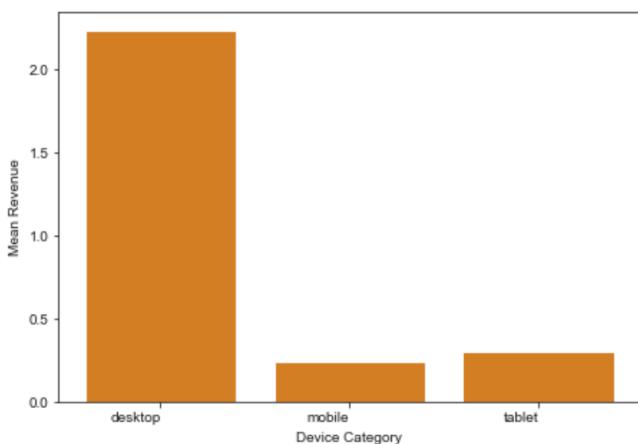
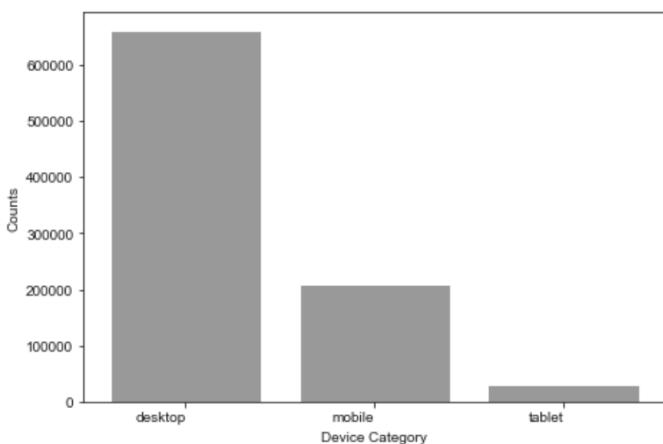
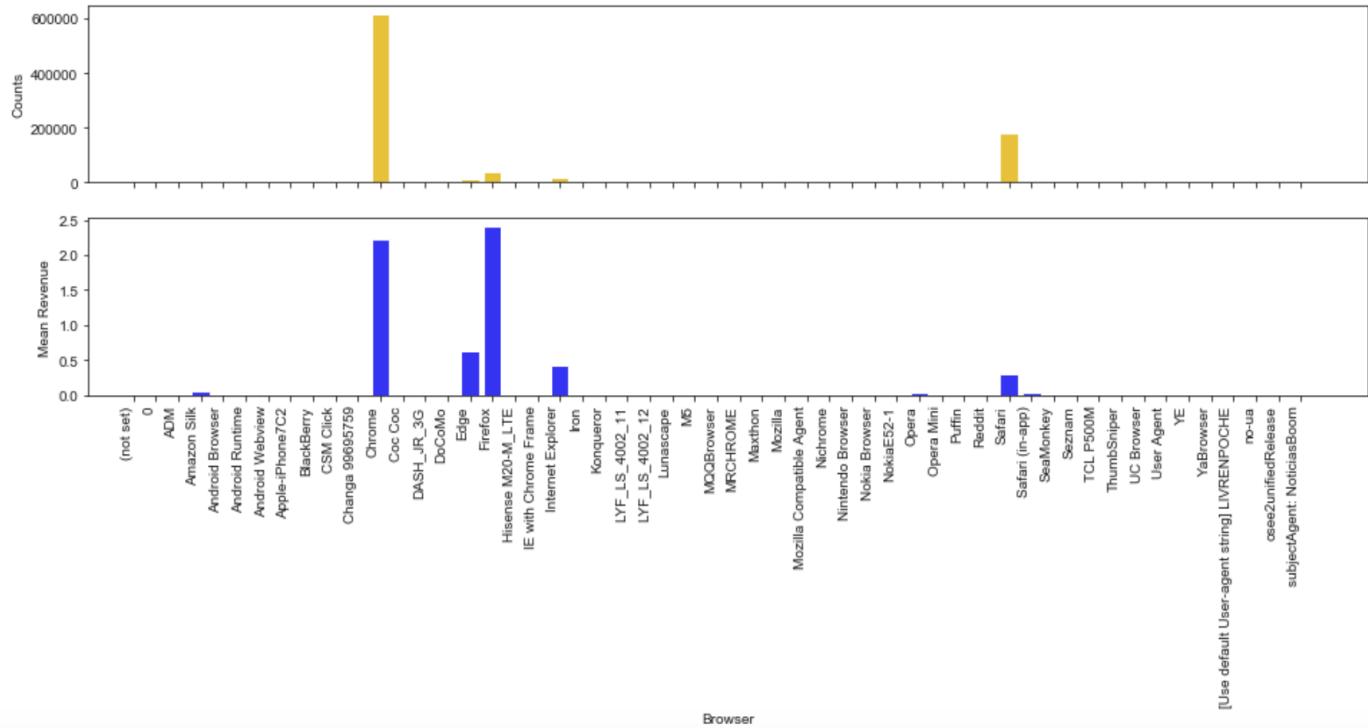


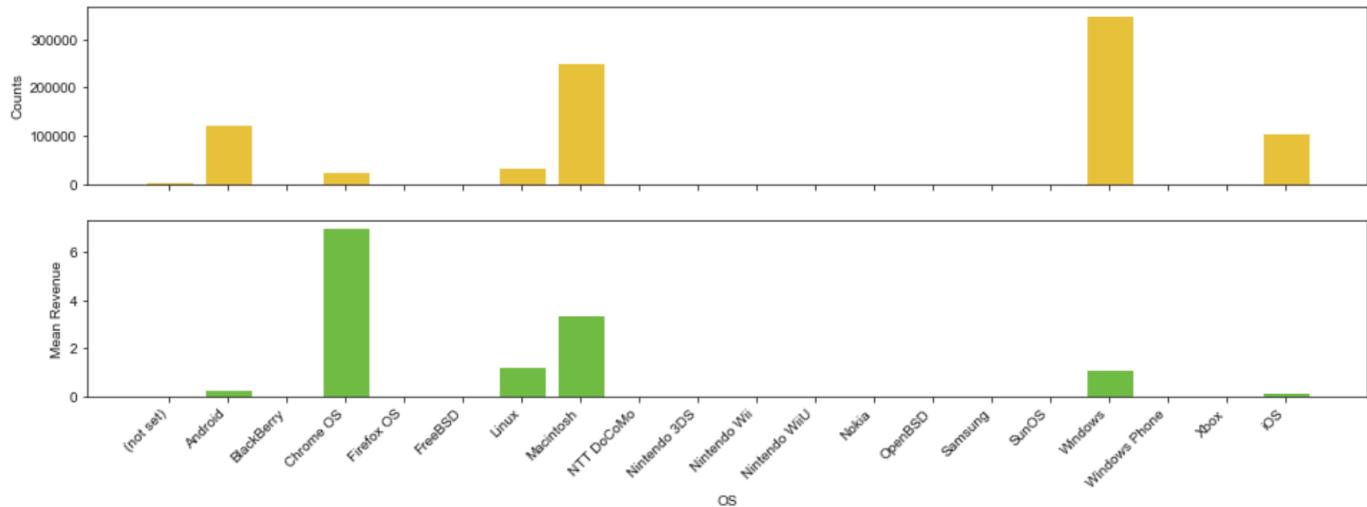
Analyze channelGrouping

The counts and mean revenues for different channels are almost inversely proportional. That is, the mean revenue is high if a channel grouping is less frequent. For e.g. 'Display' channel.

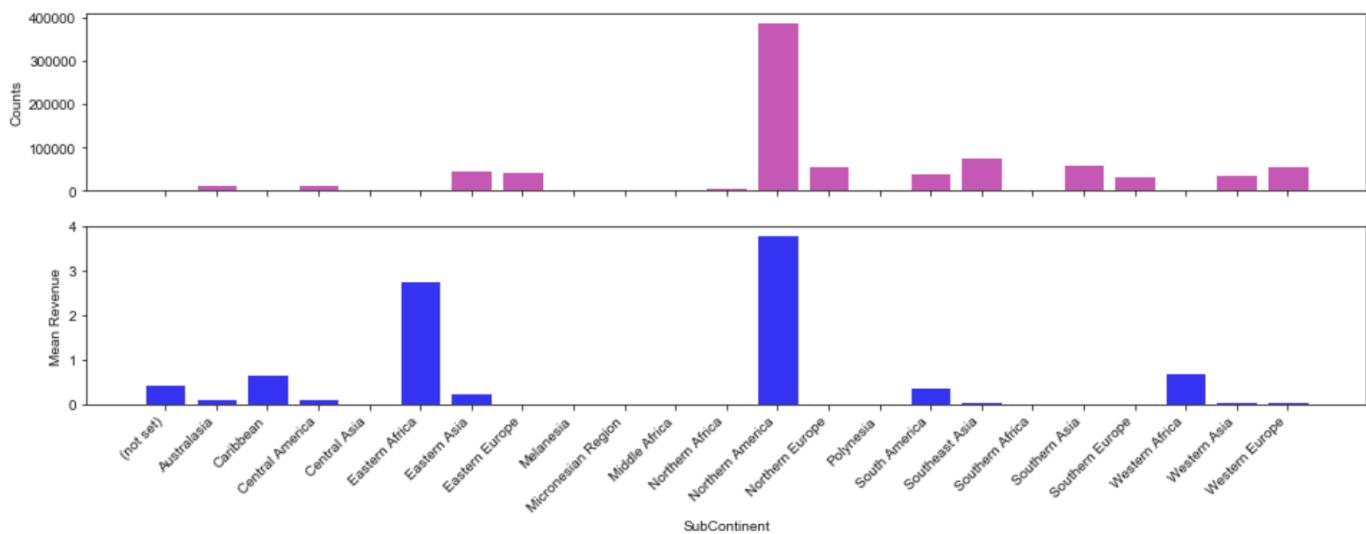
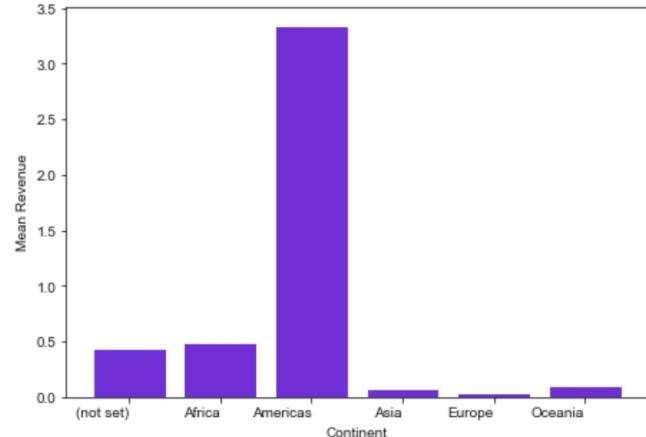
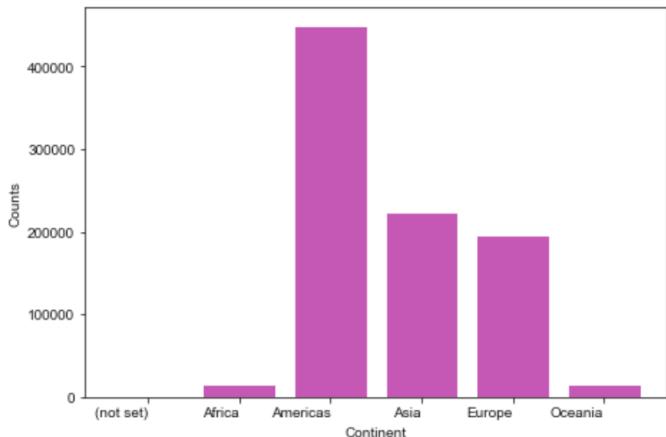


Analyze device fields

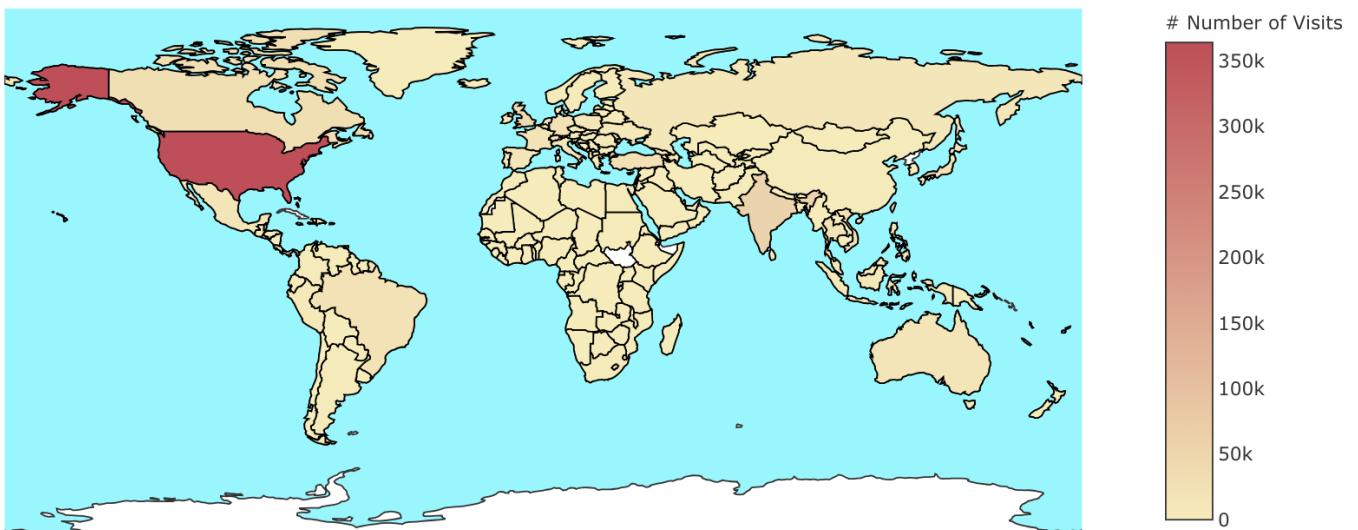




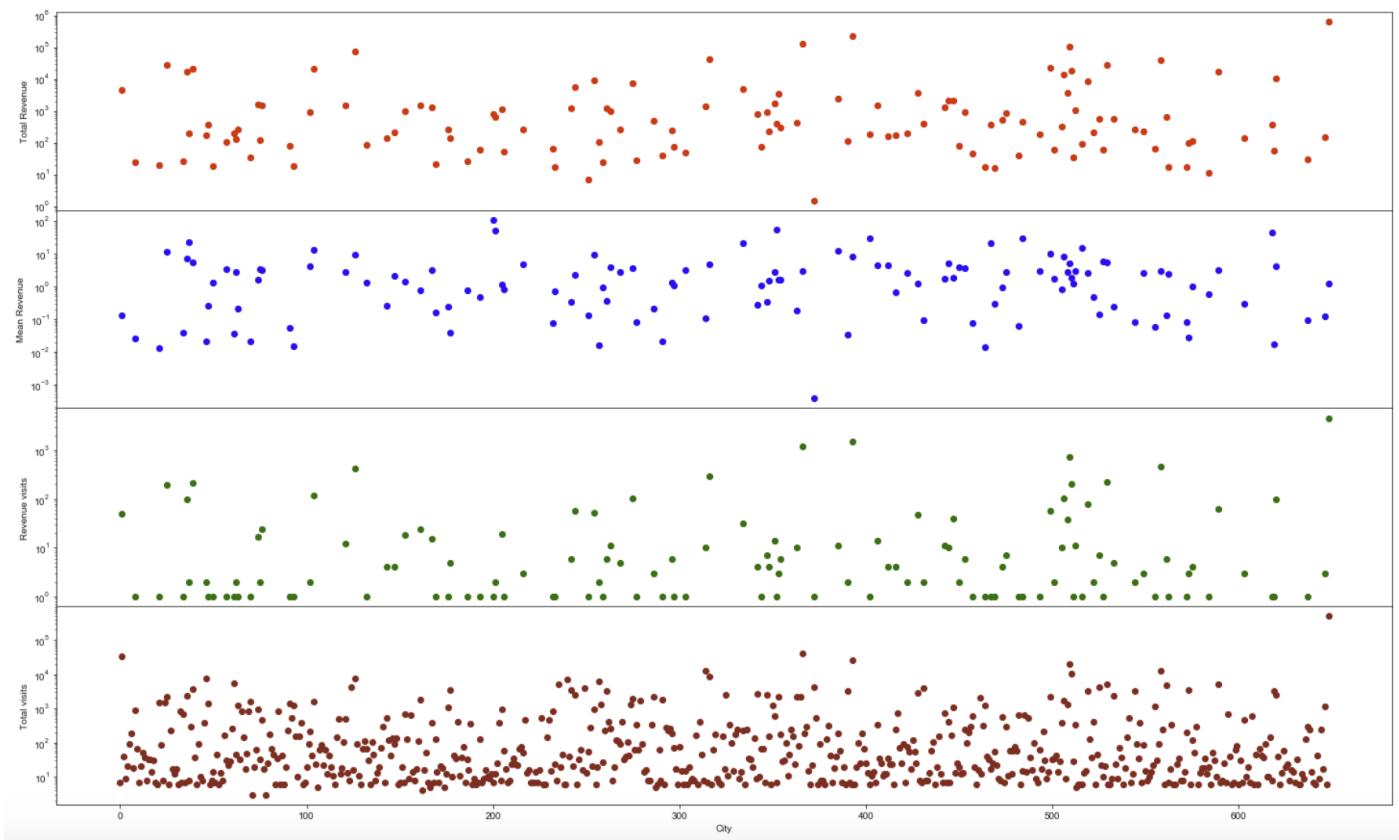
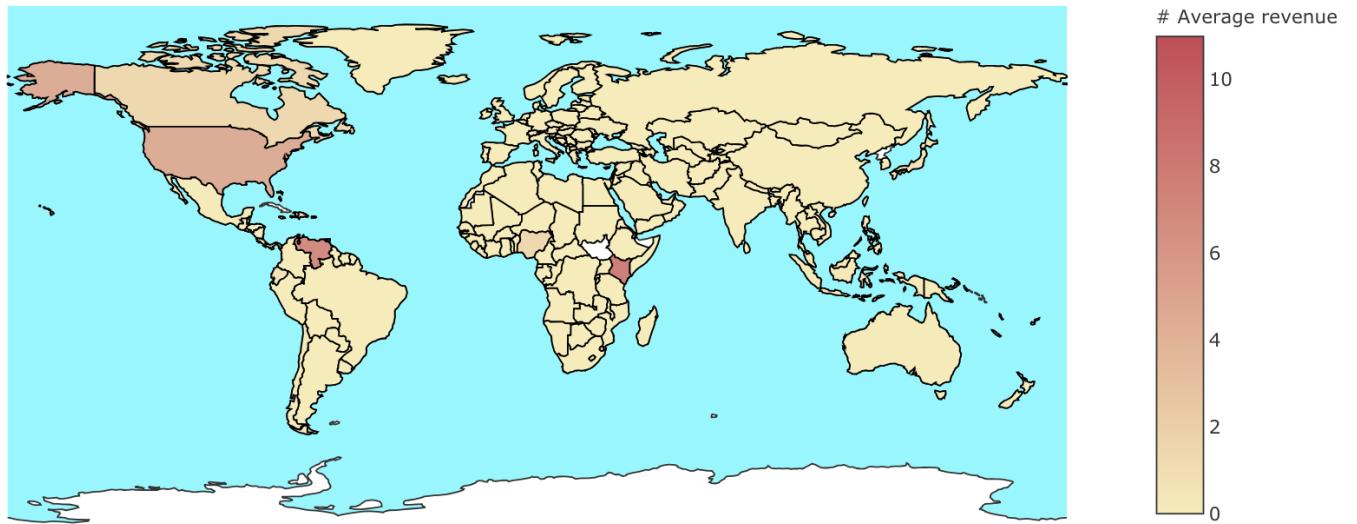
Analyze Geo fields



Visit by countries

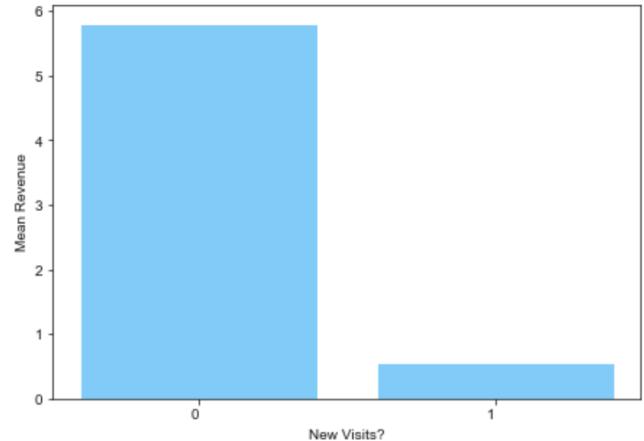
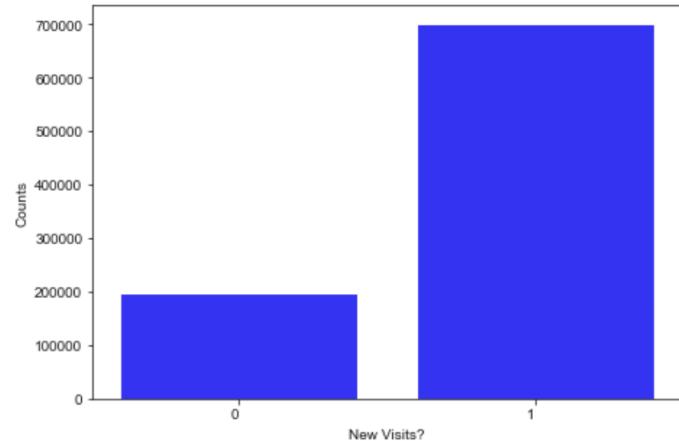


Average revenue by countries



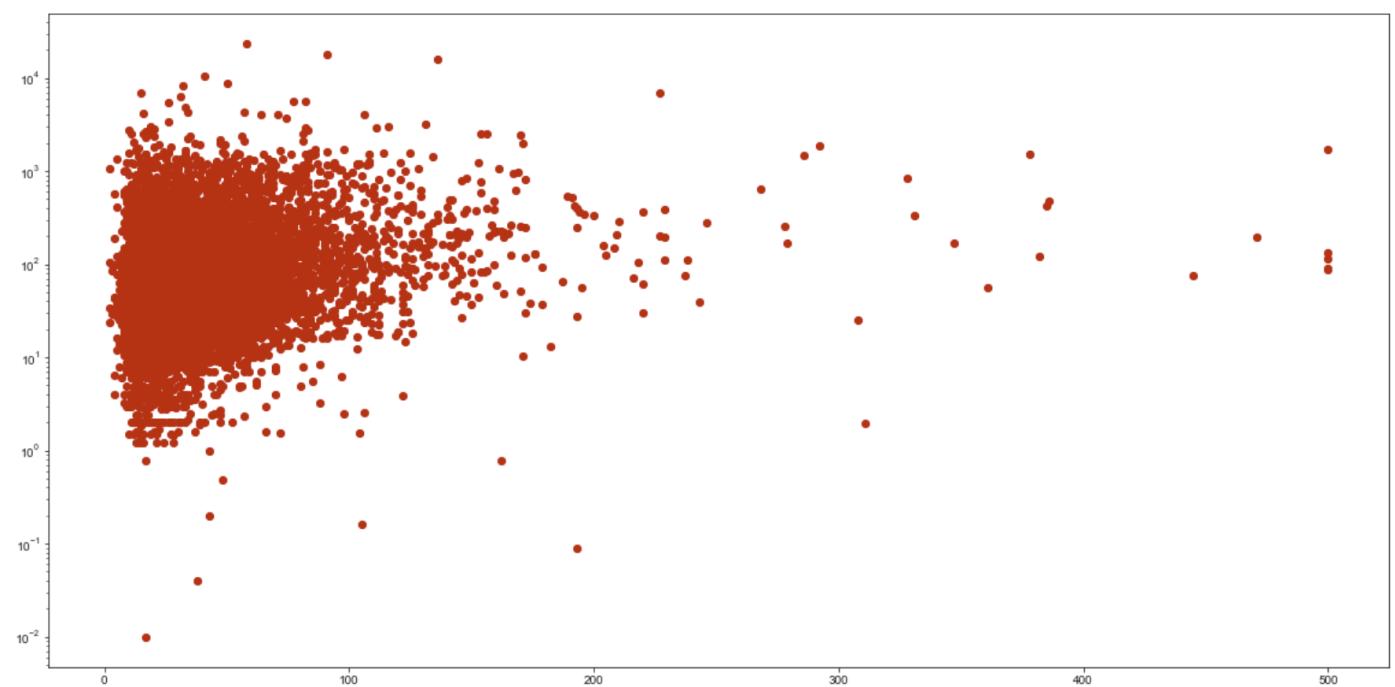
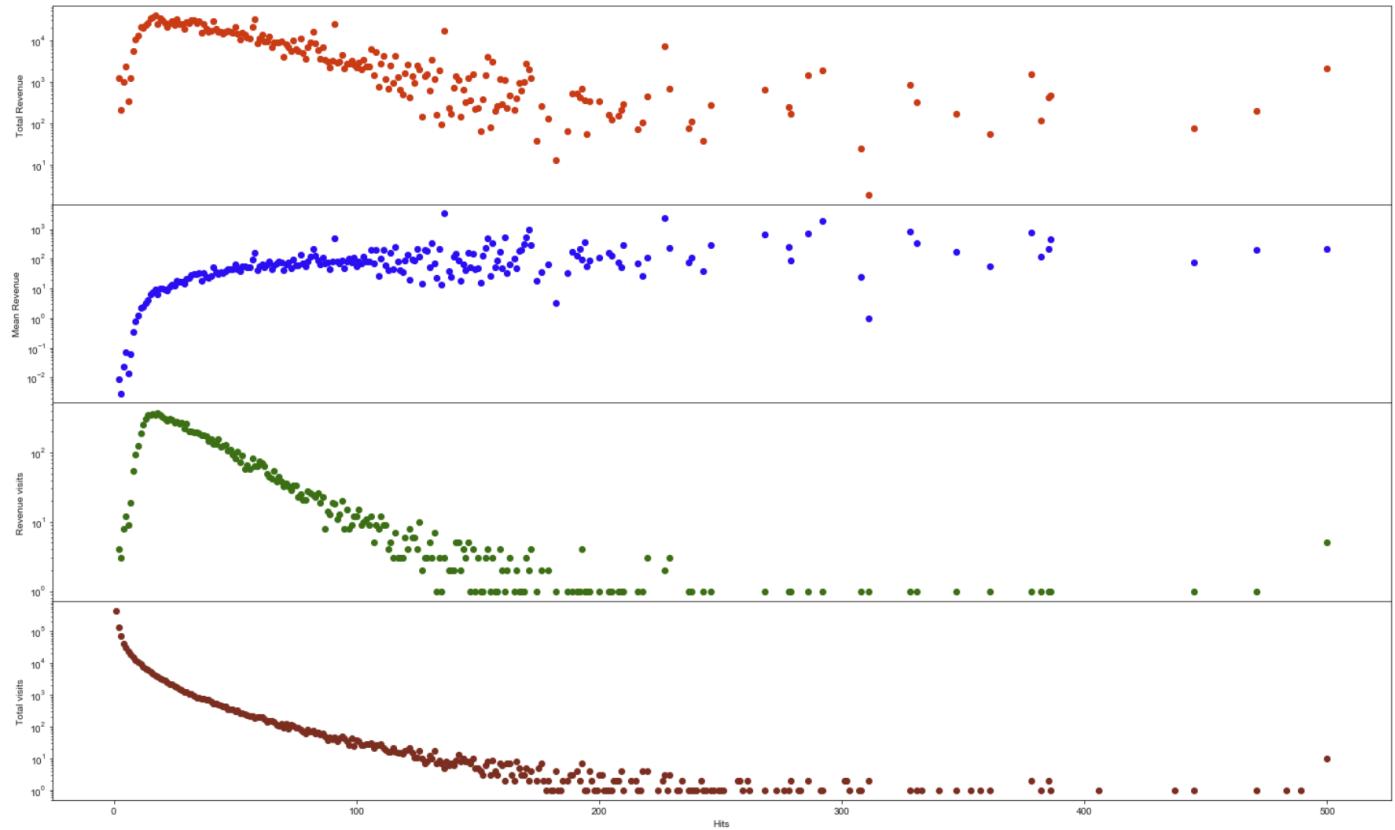
Analyze new visits

More than 70% visits are new visits or first time visits. But as expected the mean revenue from old visitors is high as compared to new visitors.



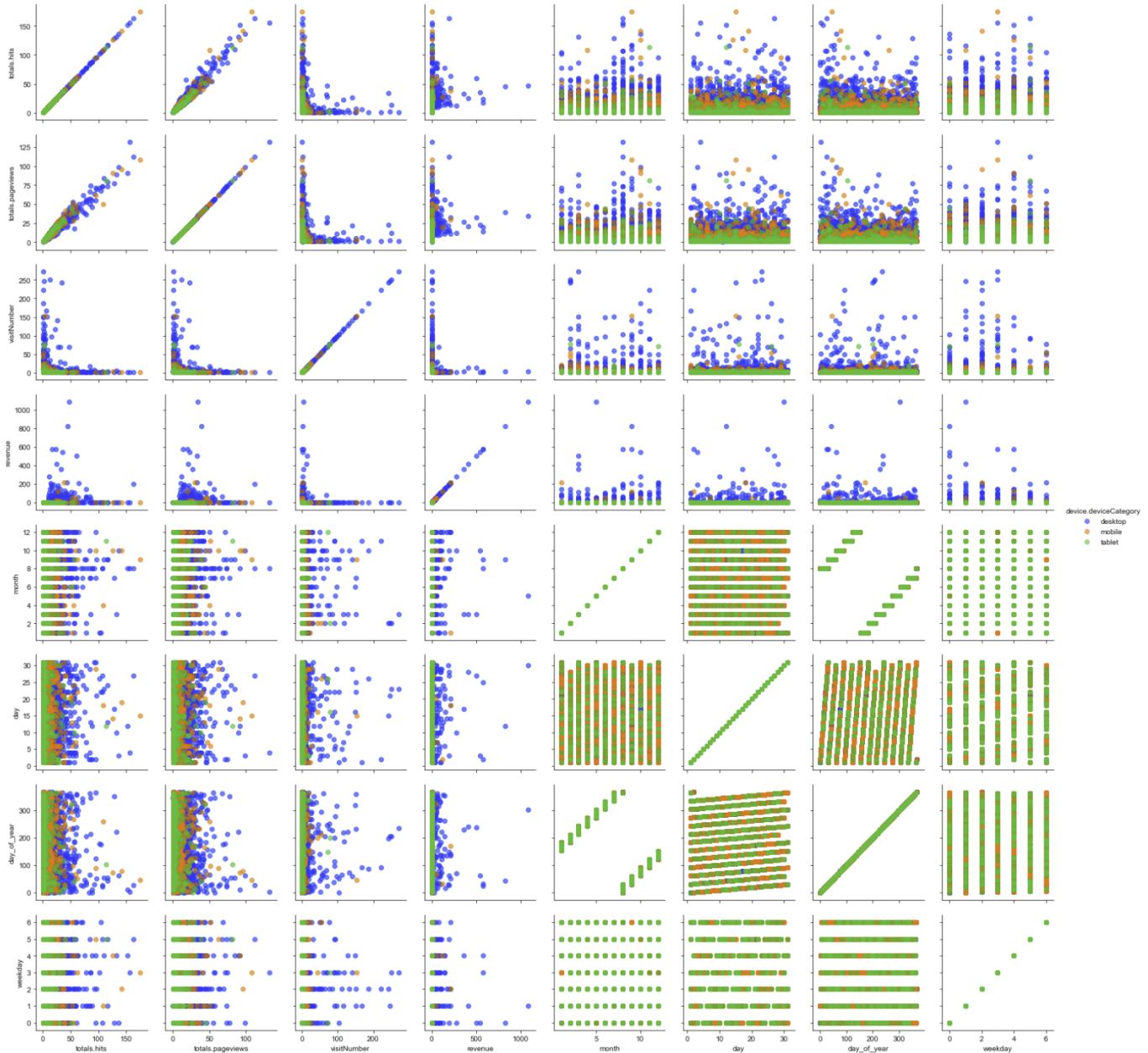
Analyze hits

Scatter plot indicates that hits below 100 get huge number of transactions as compared to other hit values. Page views has similar distribution. And it shows that hits and page views are linearly related.



Statistical Inference

Out of all the given variables, we find out what relations, if any, exist between different pairs. Specially the relation with the target variable. We only take a random sample for the plots.



Hypothesis Testing

By looking at all these plots, one thing is clear that there is linear relationship between the hits and the page views. Logically also it makes sense. The more number of pages are browsed it will produce more hits/downloads. Now let us see if the correlation is statistically significant. If the correlation is high we can try omitting one of the variable in our feature matrix to train the machine learning models. Such features produce multi-collinearity issues while model fitting.

Formally, the correlation coefficient, r , tells us about the strength and direction of the linear relationship between x and y . However, the reliability of the linear model also depends on how many observed data points are in the sample. We need to look at both the value of the correlation coefficient r and the sample size n , together. The hypothesis test lets us decide whether the value of the population correlation coefficient ρ is "close to zero" or "significantly different from zero". We decide this based on the sample correlation coefficient r and the sample size n .

Null Hypothesis: $H_0: \rho = 0$

Alternate Hypothesis: $H_a: \rho \neq 0$

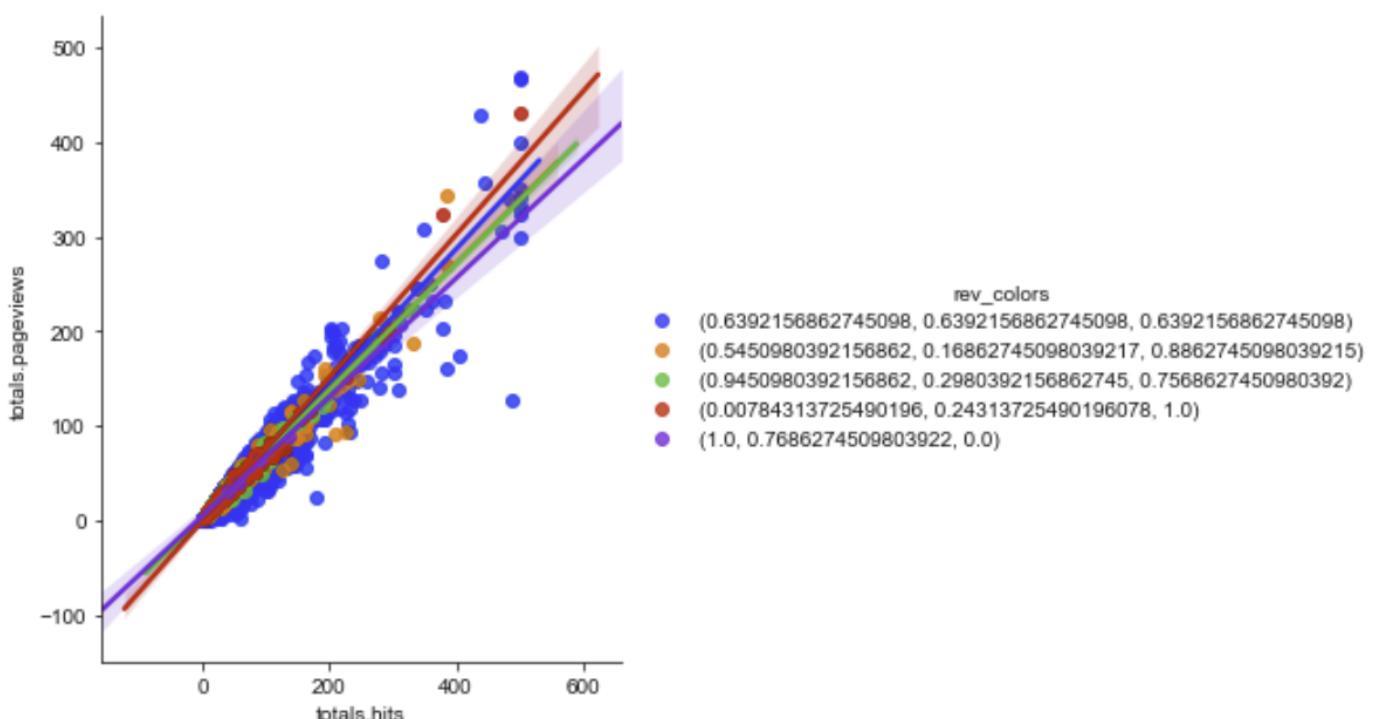
OR

Null Hypothesis H_0 : The population correlation coefficient IS NOT significantly different from zero. There IS NOT a significant linear relationship(correlation) between x and y in the population.

Alternate Hypothesis H_a : The population correlation coefficient IS significantly DIFFERENT FROM zero. There IS A SIGNIFICANT LINEAR RELATIONSHIP (correlation) between x and y in the population.

at significance level ($\alpha = 0.01$)

We will accept or reject the Null hypothesis based on the p values. We will apply this to all the feature pairs and find the highest correlated features. And we can compare that with our previous results too.



Conclusion

The above p value is zero and we reject the null hypothesis that the population correlation is zero. The variables hits and pageViews have very high statistically significant correlation of 0.98 at significance level of 1%.

Modeling

Theory and Approach followed

Our goal is to develop a model that describes the relationship between the features in the data and the target class (product category). So far we have got good understanding of the data and we can apply the machine learning algorithms and find out the best model. Technically speaking, this is Supervised Regression ML problem. Here we have the training data with the actual labels. We need to model that and use the model to predict labels for new data.

We will start with a quick and dirty baseline model. And improve upon our model in iterative fashion. Following are some ideas to keep in mind.

Overfitting and Underfitting

When the model fits the training data too perfectly but not the test data, it is called overfitting. Such a model will not be useful for predictions. While at the other extreme end our model might fit the training data very poorly. Consequently, it can't fit test data any better and will give inaccurate predictions. Generally its a trade off between overfitting and underfitting. This is also called **Bias-Variance trade-off**. When the model fits too well to the training data (a subset of population) it has low Bias, that is, it mimics the data very well. But it might fail miserably in mimicking the test data (another subset of population). Or it has variance between different subsets of data. As we try to make this variance low, it will increase the Bias.

In our case, the number of data points is very large (900 thousand) as compared to the number of features. This will help in avoiding overfitting. We are going to always compare train and test accuracy to see how well the model fits. We should also design train-test split of the data wisely to regulate the bias-variance trade off.

Model Evaluation Criteria

Evaluating our model closely relates with how the model is going to be used by the client.

Root Mean Squared Error (RMSE): This is the evaluation criteria used by the client (Google). So this is our main criterion to compare and improve the models. It is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2},$$

where \hat{y} is the natural log of the predicted revenue for a customer and y is the natural log of the actual summed revenue value plus one.

Data and code organization

- Data files generated and saved instead of data pipelines in the runtime

Although data pipelines are generally used to transform train and test data sets. In this project the data is huge and we don't want to run all the steps of datapipeline while development. Moreover there is not much of data transformation. Mainly we do the Standard Scaling and save the resulting data as a file in

the /data folder. This file is used for training of most the models. Similiarly the cleaned data files are also saved into the /data folder.

- **Python functions written to apply the DRY principle**

Since lot of things are repetitive like fitting the data, evaluating the results etc. we have written Python functions are used multiple times.

- **Fitted Models are also saved**

Again because the data is huge we don't want to run the (timeconsuming) model fitting whenever we restart the Jupyter notebook. So we do save the fitted models in the /model-repo directory.

First (Baseline) Models

Model Comparisons

Recommendations for the client

Conclusions and Future analysis