**Infinite Lambda Data Engineer Task**

**Solution by Vladimir Yovchev**

**Date: 08.12.2020**

**Plovdiv, Bulgaria**

**Dear all,**

Here are the steps I perfumed to address the task and the data presented in it. This will be also committed to GitHub as requested. It includes also – excel files and a ipynb (jupyter notebook) file of the analysis.

**Step 1:** We transfer the data in a more readable way by putting it into an Excel file

**Step 2:** When transferring the data we skip fields that do not bring any additional value and are not useful in calculations. In our case we omit the ‘description’ field as it repeats the information from the ‘column name’ field.

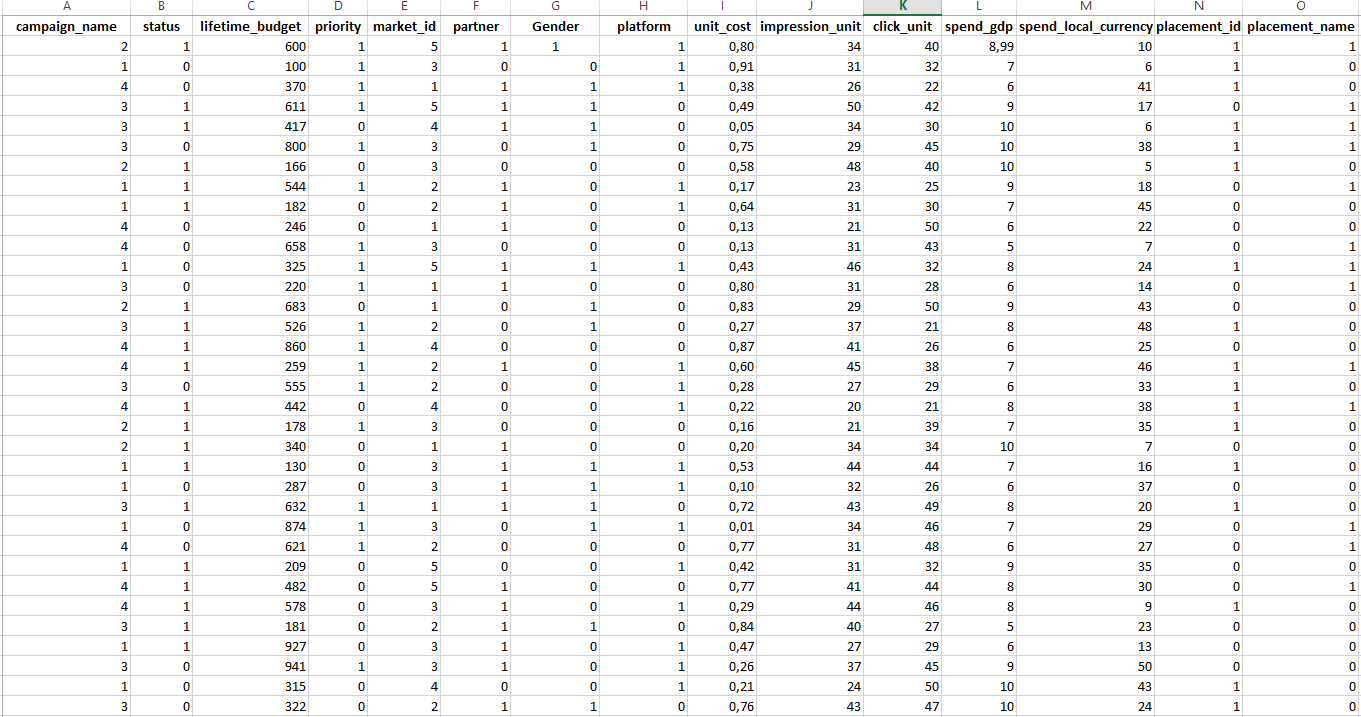
**Step 3:** We substitute the string values (campaign name – ‘summer’, status – active, platform – IOS or Android and the remaining string fields) with numerical values. We design a legend with explanations for that – see excel table ‘Data table by Vladimir’.

**Step 4:** We remove the fields that we believe do not determine any campaign/sales success and do not bring information that can provide us with any new insights. These are:

* Campaign id – the code of a campaign cannot be correlated or provide information about sales, price, number of clicks or has any relation to budget and money spent
* Country – market id field already contains information about country of origin. Therefore we do not need another field for that.
* Name – who purchase the product as a name does not bring any important information. We can instead substitute the field with ‘gender’ which can provide more insight.
* We do not have information at the moment for purchased brand. This information will be good if we can add it.
* We add three more fields for the remaining seasons in campaign name field
* We add field ‘Gender’ instead of name.

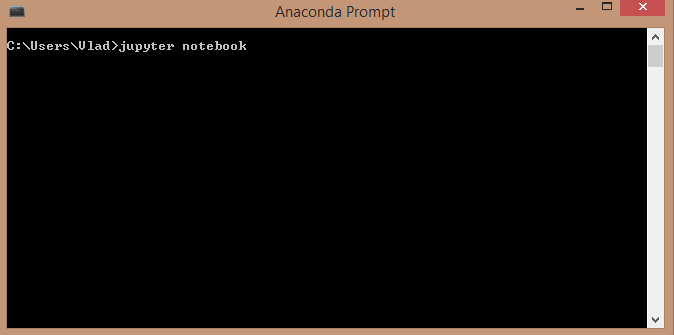
*Comment:* We try to replace string values with numerical ones as much as possible because we have to randomly generate more data. Wherever we can, we replace the string values with 1 or 0 for better and faster calculations. We will generate randomly more data using excel trying to make our data more valid.

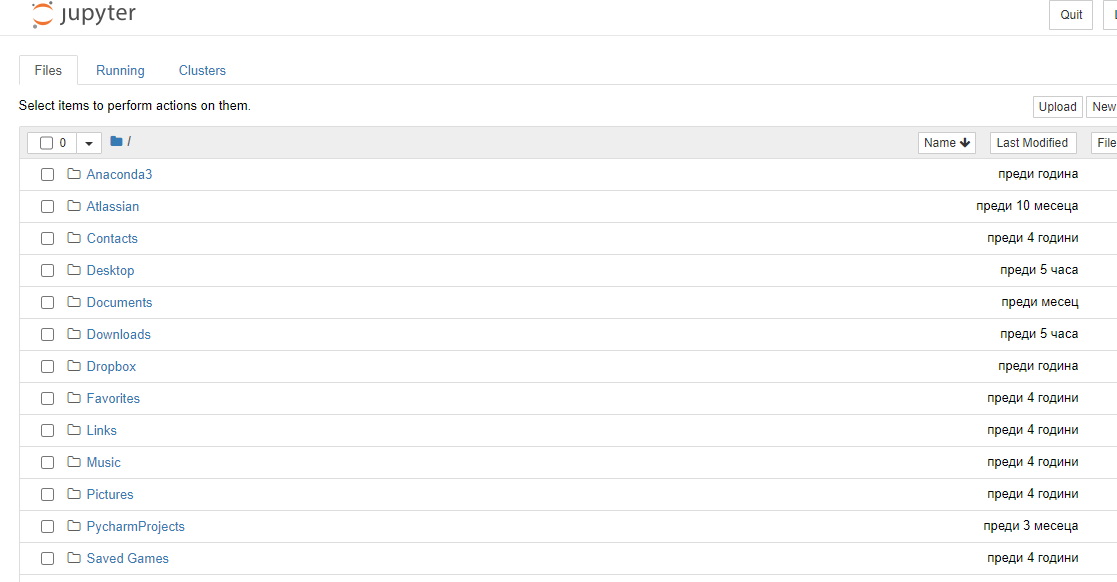
**Step 5:** After we replace the strings with values and generate 500 new observations we get the following table: see ‘New sheet’ from excel file.



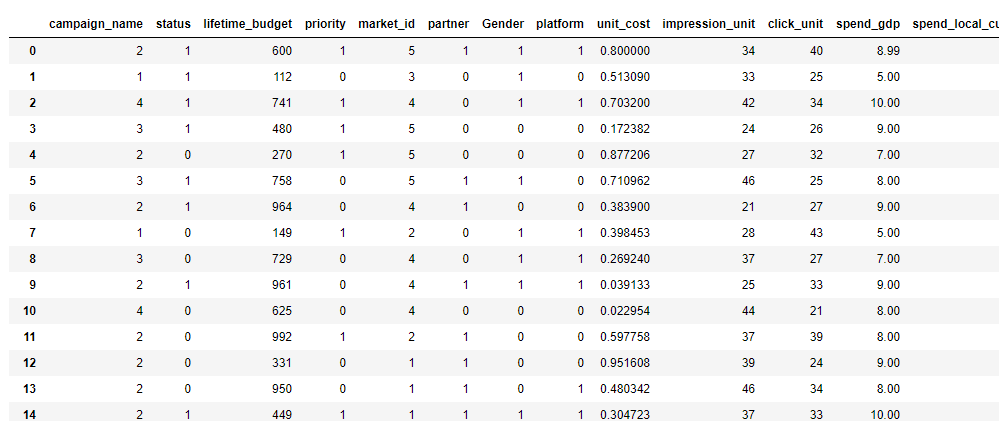
**Step 6:** We save the file as CSV(comma delimited) format so any BI API can process it easily. The new file is called ‘New data’.

**Step 7:** We run Anaconda Linux environment and load Jupyter Notebook

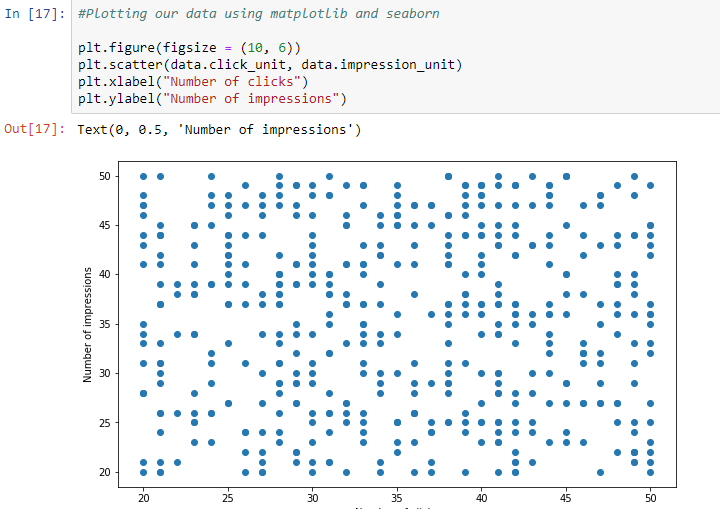


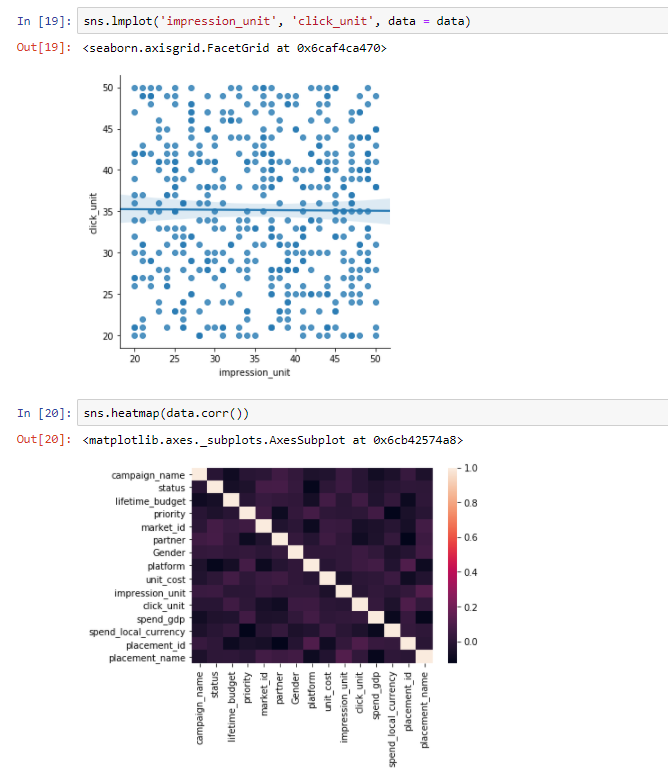


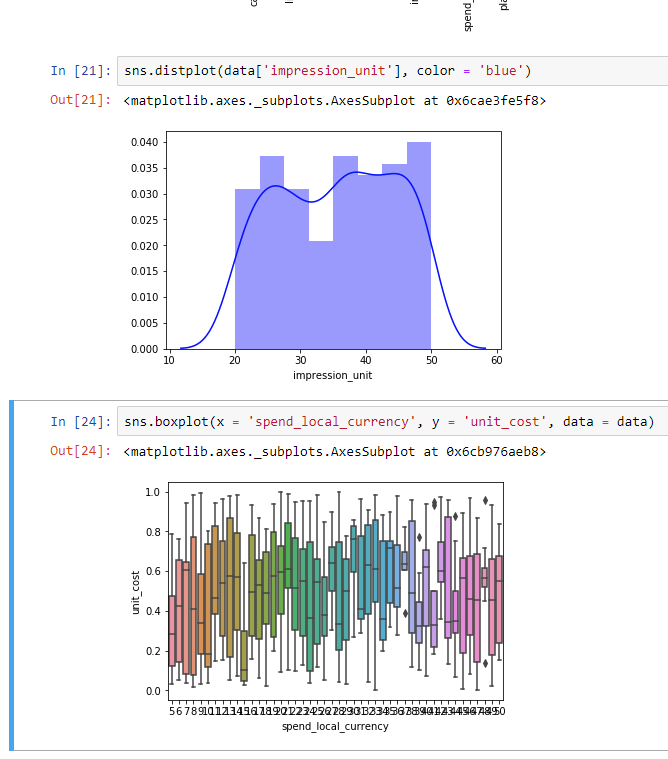
**Step 8:** We visualize the data for convenience and try to find correlation between the different factors. We are looking for a linear correlation or multiple after visualizing and then confirming if the data is normally distributed in order to apply linear or multiple regression models. We discard for convenience the fields with 0 and 1 as they require other technique – binomial distribution.



**Step 9:** Unfortunately after visualizing the data using Cartesian coordinate system for any two variables that look initially correlated such as impression\_unit and click\_unit or unit\_cost and spend\_local\_currency we see that there is no correlation between them.



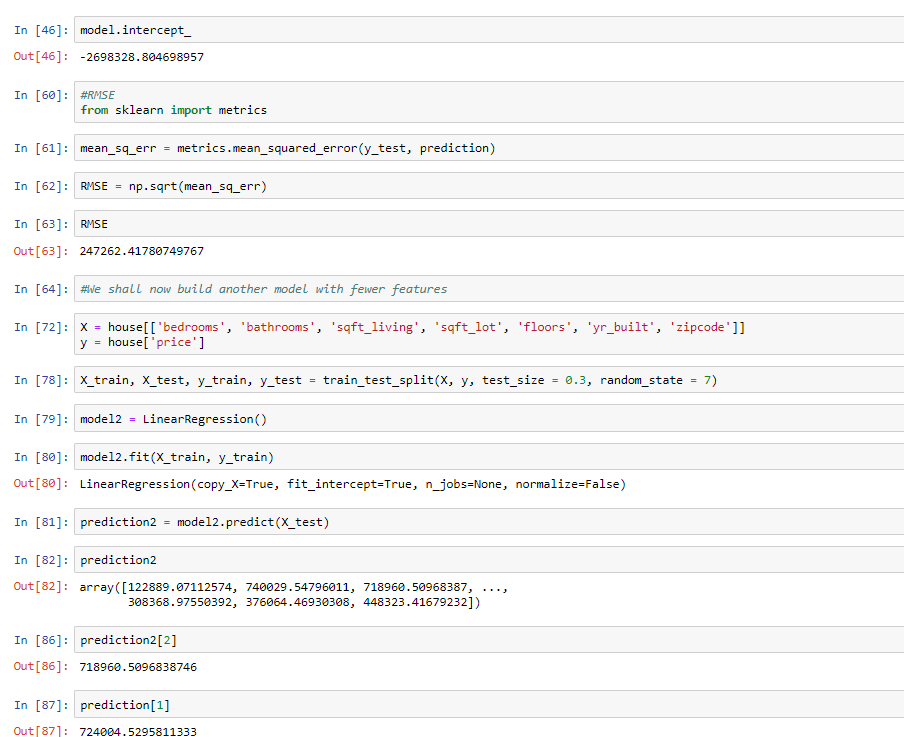




As you can see above the same is valid for how much a person spends based on the price of a product. If the price goes up the spending should go down. Again here we see no correlation.

If we had any correlation we can perform regression analysis and design a model which can predict new spending or pricing of new products based on constantly coming new data. We can also use binomial distribution for the remaining fields that we denoted with 0 and 1.

If we had enough qualified data the model could look like the one bellow predicting the price of a house:



**Conclusion:** The initial task comes with 15 rows and just one observation for each. This is not enough to do any analysis. The data should be of at least 100 real observations. For that reason we generated automatically 500 more entries. But this generation means that 99% of our data is extrapolated. This is not at all a valid data set. The maximum for extrapolation is 20% to reach reasonable results in analysis. That is why for correlated categories such as number of impressions, number of clicks or price of a product and money spend we get no relationship at all when that should not be the case.

Similar analysis can be done with Microsoft AWS. This API will allow us to compare the validity of our model with other existing models. However even if we use AWS the initial data we have is not enough to create and run a model. So the conclusion will be the same.