PROCESS REFLECTION

Arpit Sidana
Hamsika Venkataramanan
Karan Puri
Kavisha Shah
Malavika Mahalingam

The following is our internal reflection based on our project:

Hypothesis 1:

Informal reflection - Issues/roadblocks and how we overcame them

The sheer size of the data led to a high processing time while trying to write and optimize code. Attempts at working with limited number of rows, say first 50,000 yielded no data left for our analysis of the average active user percentage by state question. This was primarily due to the fact that multiple merges were required to map user id and state and this resulted in significant loss of data, more so when we read in limited rows. The only workaround to decrease processing time by a little was achieved by coding in the console.

How did your analysis questions change from when you first framed them to your final result?

our original question was to determine total number of active users per state but we realized this number might not be a good interpretation due to different sizes of states and data available per state and hence we switched to active user percentage per state.

What coding issues did you have to overcome?

It took time to optimize code for the choropleth map as searches for heatmap returned advanced visualizations beyond the scope of this project and thus did not necessarily help address the questions we wished to answer.

Hypothesis 2:

The hypothesis involves comparing the performance of popular food joints based on ratings across different states. Since the data set was very large we encountered a number of challenges during testing our hypothesis. To start with the hypothesis, 'food joints' was a very broad term and narrowing this term was challenging as there were vast number of entries for the same food joint. For eg. McDonalds could be listed as 'Breakfast' and as 'Fast Food'. There could be a large number of local restaurants/food joints which may be popular but would not show up due to its small volume or geographic concentration. We observed some inconsistency in listing the names of the restaurants eg. McDonalds listed as McD, Mcd or McDonalds'. In addition, there was some difficulty in segregating data on a state wise level to calculate presence of popular food joints across the State. By using a sample of the data set and running it, we were able to optimize our code, however due to the size of the data it was time consuming.

How did your analysis questions change from when you first framed them to your final result?

The question more or less remained the same.

What coding issues did you have to overcome?

There were a number of issues faced for testing this hypothesis:

To count the total number of chains in a state: To obtain this count, we faced problems with the index as every time we had to apply groupby the index had to be changed to set it according to the output we needed for the following steps. Hence we had to reset the index using reset_index to do further calculations. We faced issues while determining and changing the axis for plotting the graph for variation in rating for popular chains in different states. We also faced a problem when groupby altered all the columns according to the function applied whereas we wanted a change in a specific column only. As a result, we had to had define a new data frame and match the index of the new data frame to the original one and add it the original data frame conduct our analysis.

Hypothesis 3:

What roadblocks did we encounter? How did we get past them?

Review.json which was used to perform sentiment analysis for our third business question was extremely huge ~1.9GB in size. Due to the size of the base dataset, any operation performed on it took hours to run. Despite selecting only the required columns from the dataset, we could not perform the analysis for the entire data. We also faced memory issues while extracting data from the forecasting of API. To improve the run time and prevent memory issues, we filtered data for the year 2015. Even this did not help minimize the code run time. After trying multiple combination of filters, we finally filtered data for January 2015. This could be a limitation in our final model that tries to predict the sentiment for a review based on that day's temperature and ratings captured as stars.

How did your analysis questions change from when you first framed them to your final result?

The questions more or less remained the same. We added a comparison of users' star ratings and the sentiment score we created to check how our algorithm did in bucketing reviews as positive, negative or neutral.

What coding issues did you have to overcome?

One of the issues we faced was finding the number of positive and negative words in each comment including the duplicates. We initially used a set-intersection approach. However, the result did not give us the duplicate values. We went back to using the conventional for-loop to get the count of duplicate good or bad words in every comment.