**Group 17: Pokemon Go Project Report**

**Web Scraping:**

We have used *Beautiful soup* module for scraping the data from the given html pages. Since the files were present in nested folders, we used *os.walk* method in python to read the Android and iOS HTML files located inside each of the folders from Jul-21,2016 to Oct-31,2016.

We created an exhaustive list of the all the android and iOS webpages seen alternatively. After scraping the respective fields for Android and iOS pages using *Beautiful Soup*, string manipulation is performed to retrieve the exact value from the webpages.

**Data Organization:**

A dictionary with values from the webpages is passed as a value to the outer dictionary with *Datetime* object being the key. A counter is used to pass both android and iOS values into the *Datetime* object and thus fetch the values concurrently.

This dictionary is then put into the Pandas data frame and then transposed before they are converted to excel and csv files. For converting to JSON files we have used *csv.Dictreader* function with input being the excel file generated from transposed pandas dataframe.

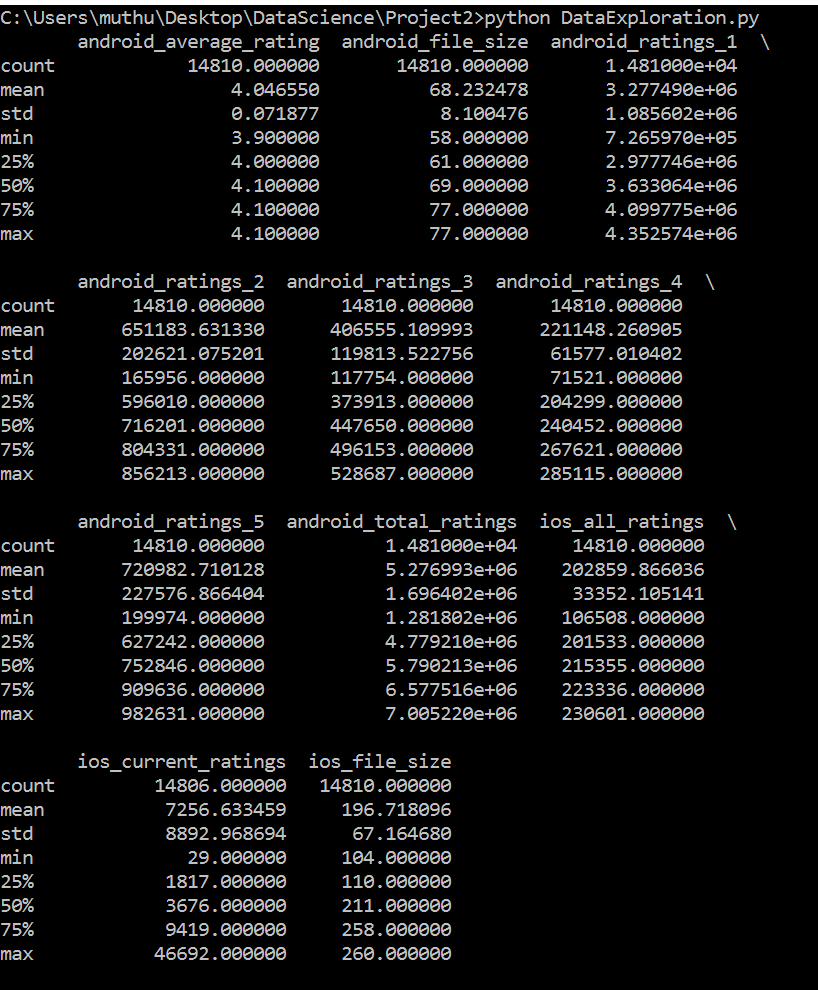
**Data Exploration:**

The data collected through Web Scraping had lot of missing values. Hence, we had two options to handle this issue for our data pre-processing. They were

i)Removing the entire row including the datetime object even if 1 out of 11 values were missing

ii) Replacing the missing values with the mean of closest data points for the particular cell.

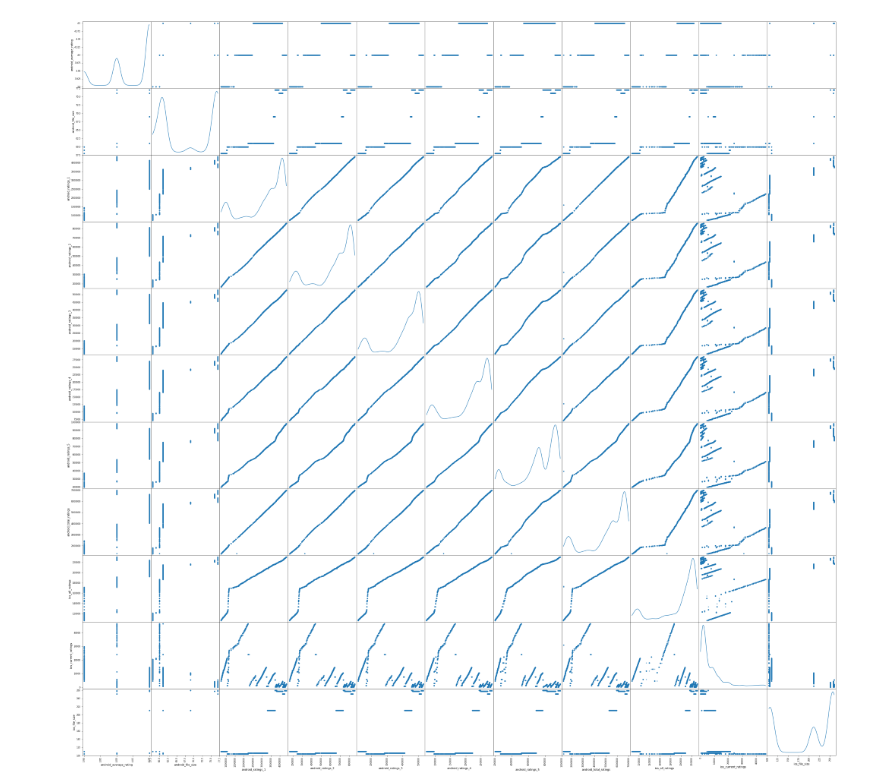
Dataframe’s ‘describe’ gave the following result.



Please find the python code for the above steps attached below:

****

Scatter Matrix:



Correlation:

From the scatter matrix, some correlations are clearly identified and are summarized as follows.

1) Positive correlation between android\_rating\_1 and android\_rating\_2

2) Positive correlation between android\_rating\_1 and android\_rating\_3

3) Positive correlation between android\_rating\_1 and android\_rating\_4

4) Positive correlation between android\_rating\_1 and android\_rating\_5

5) Positive correlation between android\_rating\_1 and android\_total\_ratings

6) Positive correlation between android\_rating\_1 and ios\_all\_ratings

7) Positive correlation between android\_rating\_2 and android\_rating\_3

8) Positive correlation between android\_rating\_2 and android\_rating\_4

9) Positive correlation between android\_rating\_2 and android\_rating\_5

10) Positive correlation between android\_rating\_2 and android\_total\_ratings.

11) Positive correlation between android\_rating\_2 and ios\_all\_ratings.

12) Positive correlation between android\_rating\_3 and android\_rating\_4.

13) Positive correlation between android\_rating\_3 and android\_rating\_5.

14) Positive correlation between android\_rating\_3 and android\_total\_ratings.

15) Positive correlation between android\_rating\_3 and ios\_all\_ratings.

16) Positive correlation between android\_rating\_4 and android\_rating\_5.

17) Positive correlation between android\_rating\_4 and android\_total\_ratings.

18) Positive correlation between android\_rating\_4 and ios\_all\_ratings.

19) Positive correlation between android\_rating\_5 and android\_total\_ratings.

20) Positive correlation between android\_rating\_5 and ios\_all\_ratings.

21) Positive correlation between android\_total\_ratings and ios\_all\_ratings.

The *Numpy* module’s corrcoef() function is used to calculate the correlation among the identified variables and the results are conveniently put in a dictionary which is transformed to pandas and eventually created an excel sheet. Attached is the excel sheet containing the values.



**Time-series graphs:**

The time series graph is constructed using *Matplotlib*’s pyplot module and we have also used add \_subplot function to accommodate date and time in the x-axis. Also, we have used matplotlib’s dates class and used date2num function to convert dates to floating point values to feed into the plot function. Also, we have used to\_pydatetime method to make it compatible with date2num method of dates class. The issue was resolved by converting the x axis data as numpy datetime object. The results are captured in a word document attached below.



**Prediction Model:**

The predicted android total ratings is 6622149.72112044

The predicted iOS total ratings is 6622149.72112044

The steps performed are attached in this document.



**Deep Learning using Tensor Flow:**

We have conducted Image scraping for all the Android and iOS HTML pages and identified that there were unique screenshots in Android HTML pages and 17 unique images in iOS HTML pages.

Please find the code attached below, that was used for scraping the images from the respective webpages.

****

Once the 22 unique screenshot images from the Android and iOS HTML pages are scrapped, *Tensor Flow* module is used to perform Deep learning to the images. This was done to extract tags from each of the images with their corresponding probabilities. Following python code was used for the same.



While the iOS images returned tags with probabilities, the Android images were throwing errors when Tensor Flow was performed for the images. Plea

se find the downloaded images and their tags with probabilities attached in the document below:



References:

1. <https://www.stackoverflow.com>
2. <https://stats.stackexchange.com/questions/275276/linearregression-vs-ridge-vs-lasso>
3. <https://discuss.analyticsvidhya.com/t/comparison-between-ridge-linear-and-lasso-regression/8213/3>
4. <https://www.quora.com>
5. <https://www.daniweb.com/programming/software-development/threads/482270/plot-date-and-time-xaxis-versus-a-value-yaxis-using-data-from-txt-file>