**INSY 5378**

**PROJECT 1:**

Pokemon Go! Analytics

Team - 2

Dhanashri Ostwal (dhanashrivilas.ostwal@mavs.uta.edu) – 1001277328

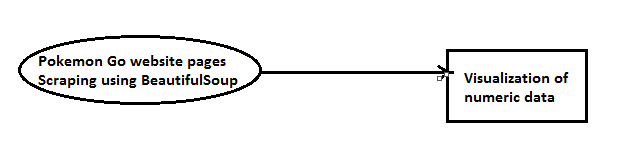
Priyanka Sekhar (priyanka.sekhar@mavs.uta.edu) - 1001215254

Saranya Ravichandran (saranya.ravichandran@mavs.uta.edu) -1001160582

**AIM OF THE PROJECT**

Pokémon Go! became a very famous augmented reality game in the last summer. In this project, we want to understand the success of the app. Specifically, the purposes of this project are (1) to be familiar with web scraping using BeautifulSoup and (2) to visualize the numeric data.

**MODEL**



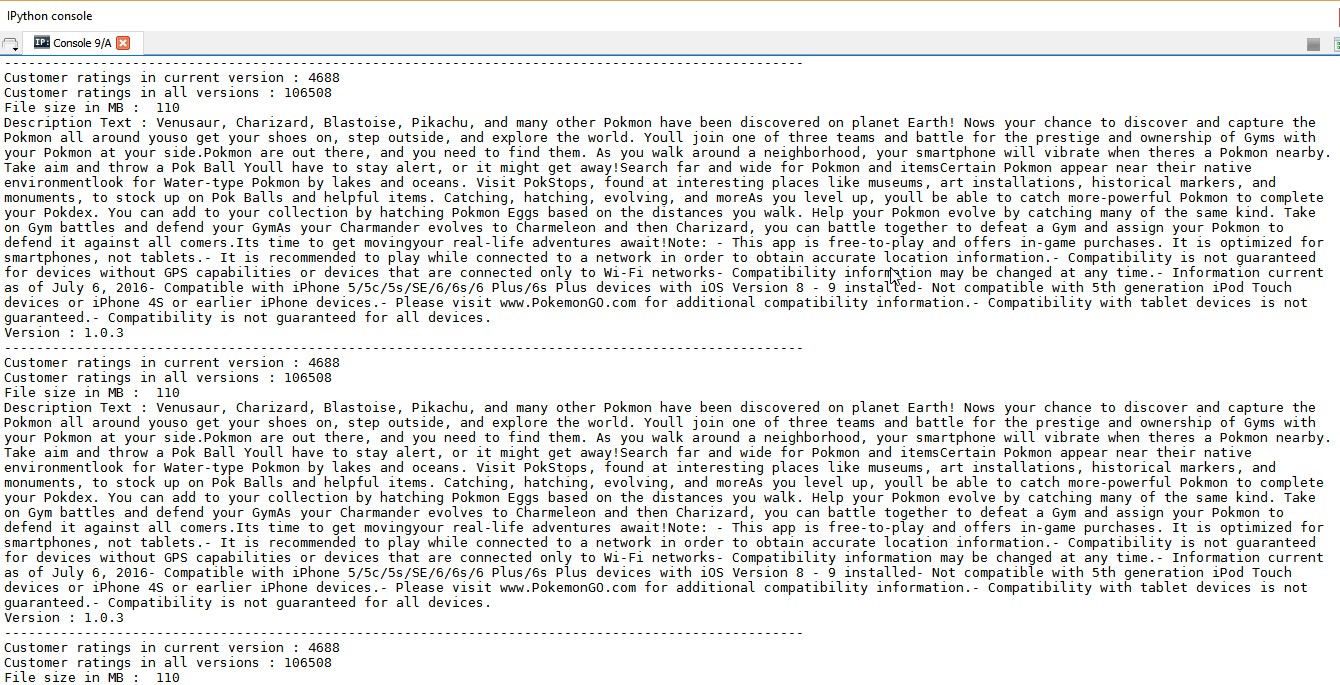
Our system analyzes the Pokémon Go data and gives a new and meaningful perspective on the visualization of the data.

**Part A and C**

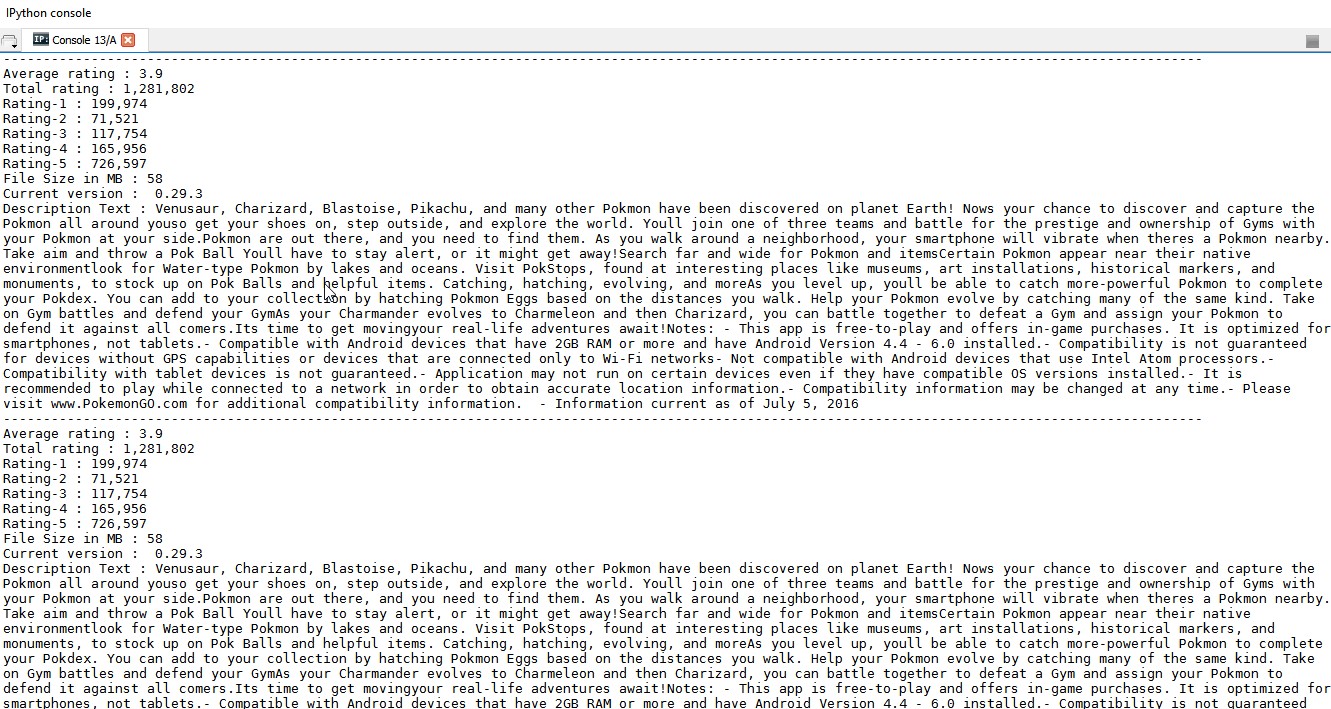
**Data scraping from given pages of Pokémon Go for iOS and Android:**

We scrape the given pages of the Pokémon Go website. We use BeautifulSoup to scrape the given web pages. Different data is scraped for iOS and Android.

iOS scraped data-



Android scraped data-



**Part B and D**

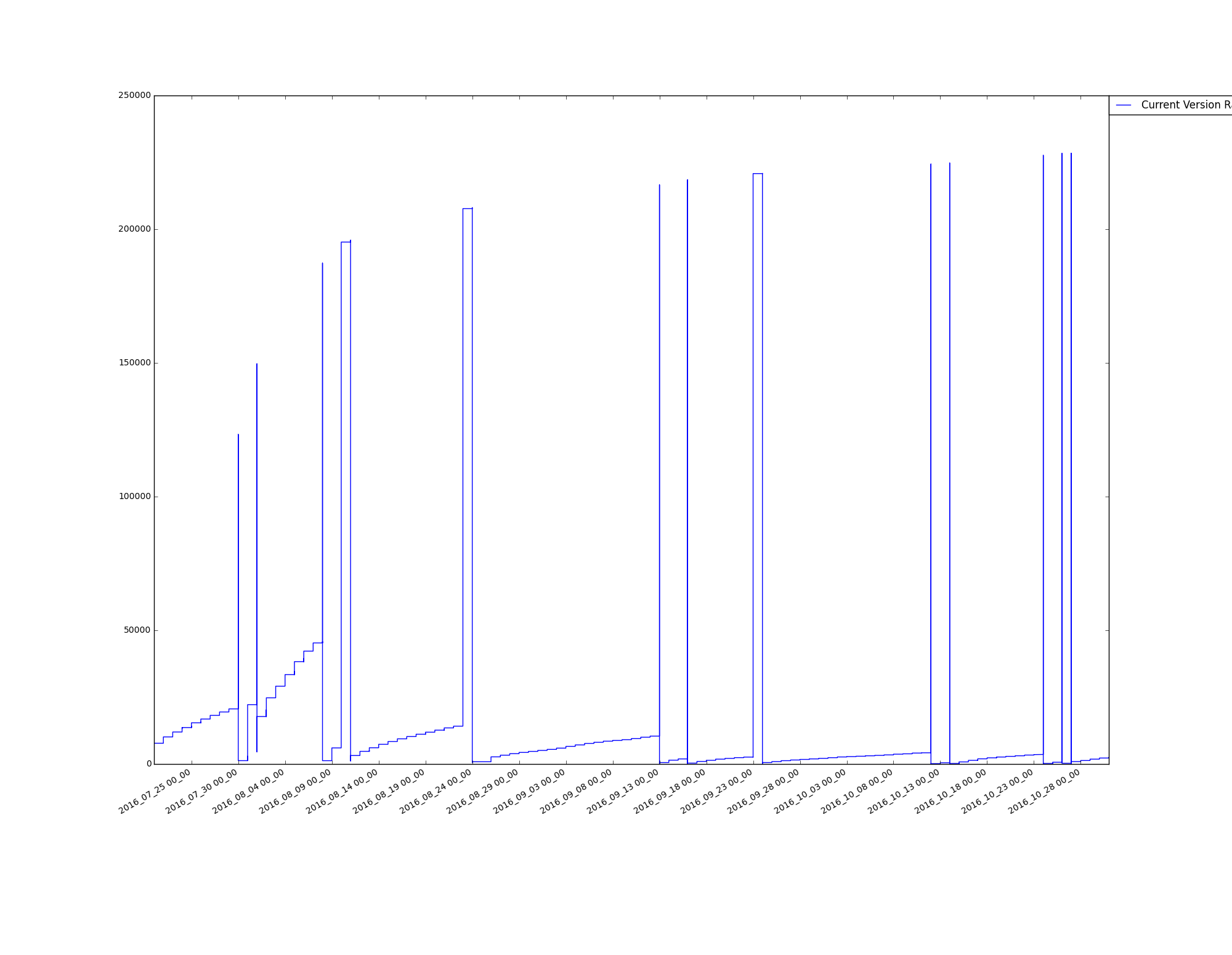
**Creation of Graph:**

In this step, from all the numeric data that we collected, we create a graph where X-axis is for the dates (2016/07/21 to 2016/10/31) and Y-axis is for the numeric data (e.g., number of ratings (Current and All Versions), and file size for iOS and average rating, number of total ratings, number of ratings for 1-5 stars, file size for Android).

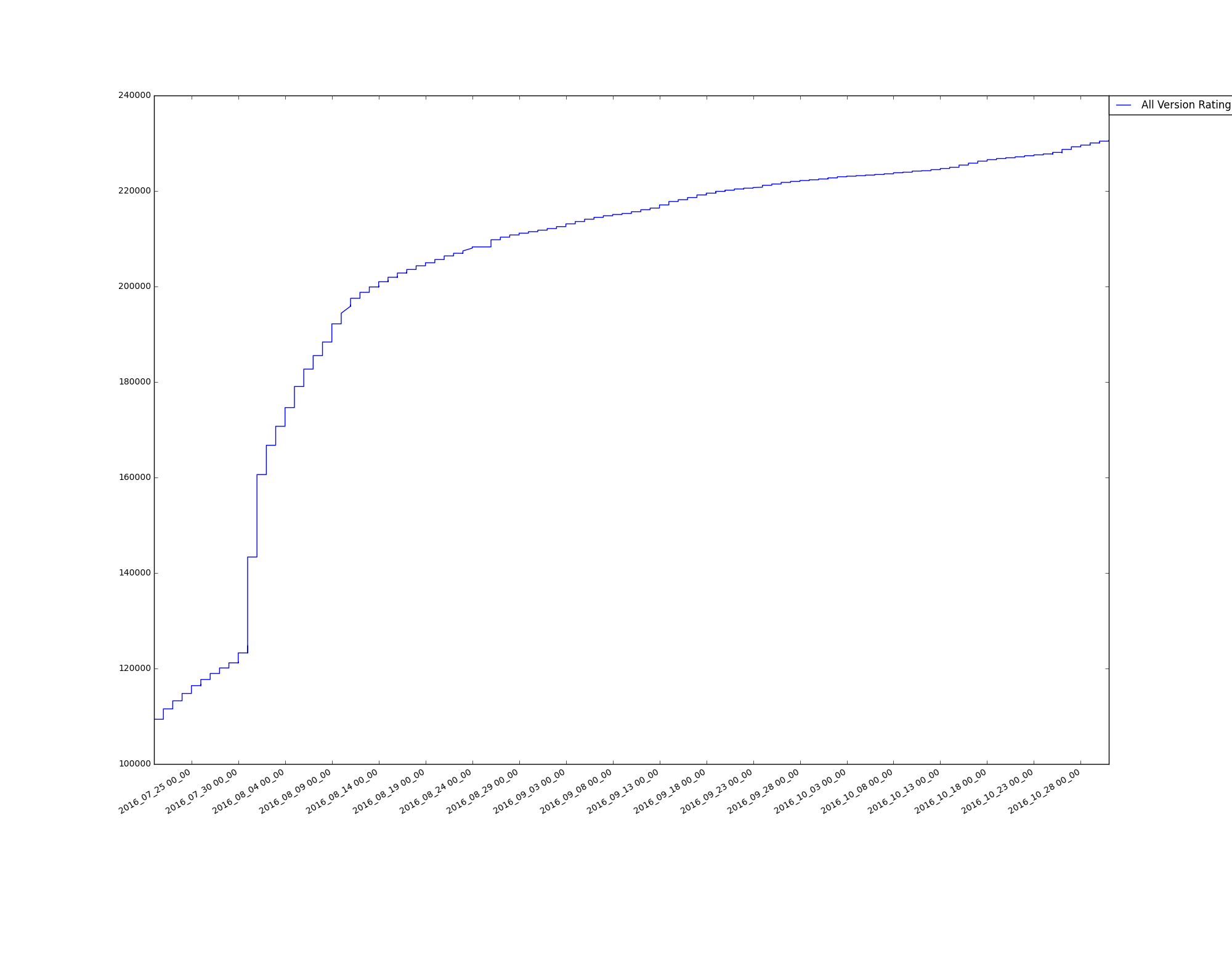
Library Matplotlib is used for the graph creation. All the graphs are displayed separately here.

iOS Graphs-

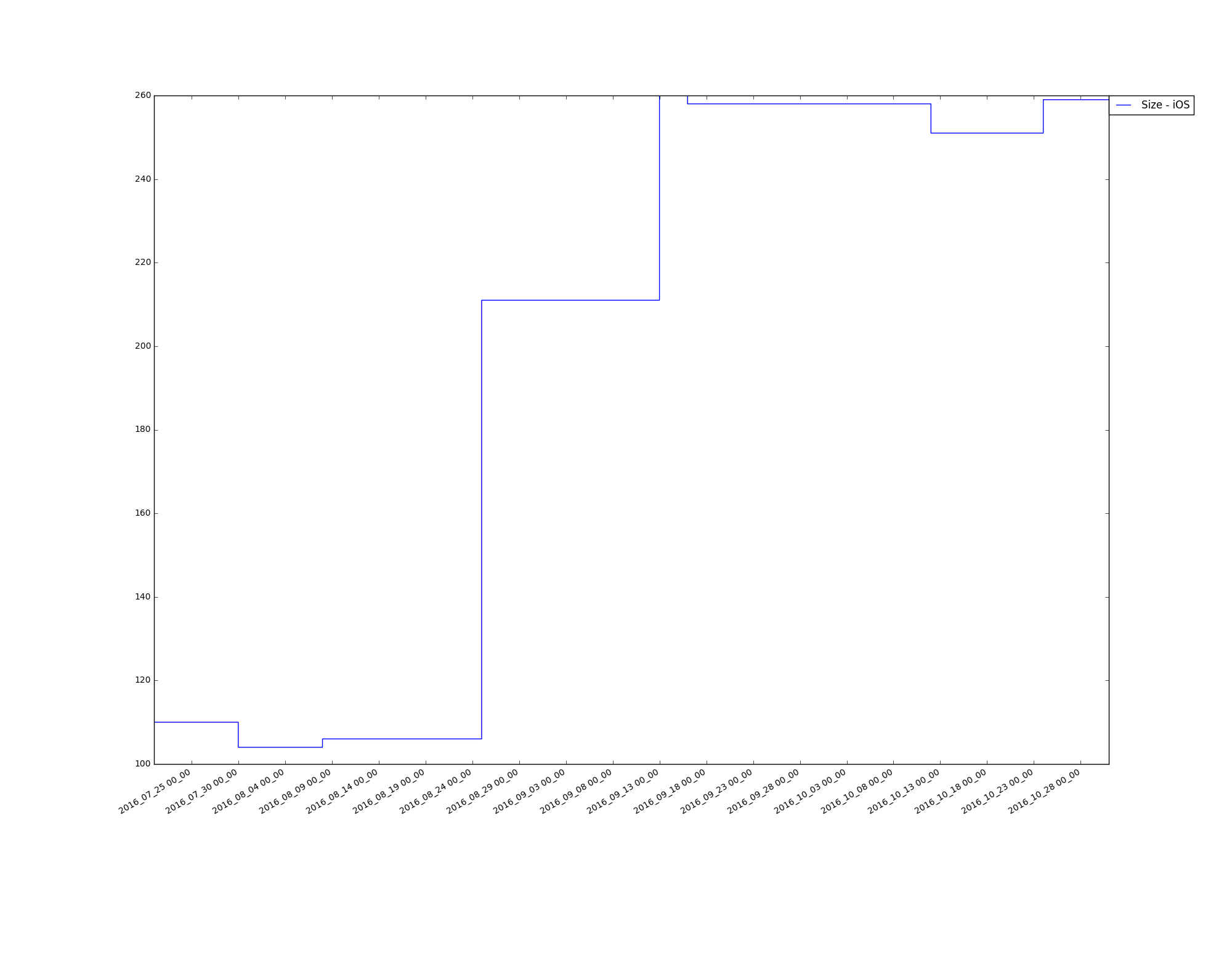
Current Version Rating -



All Version Rating -

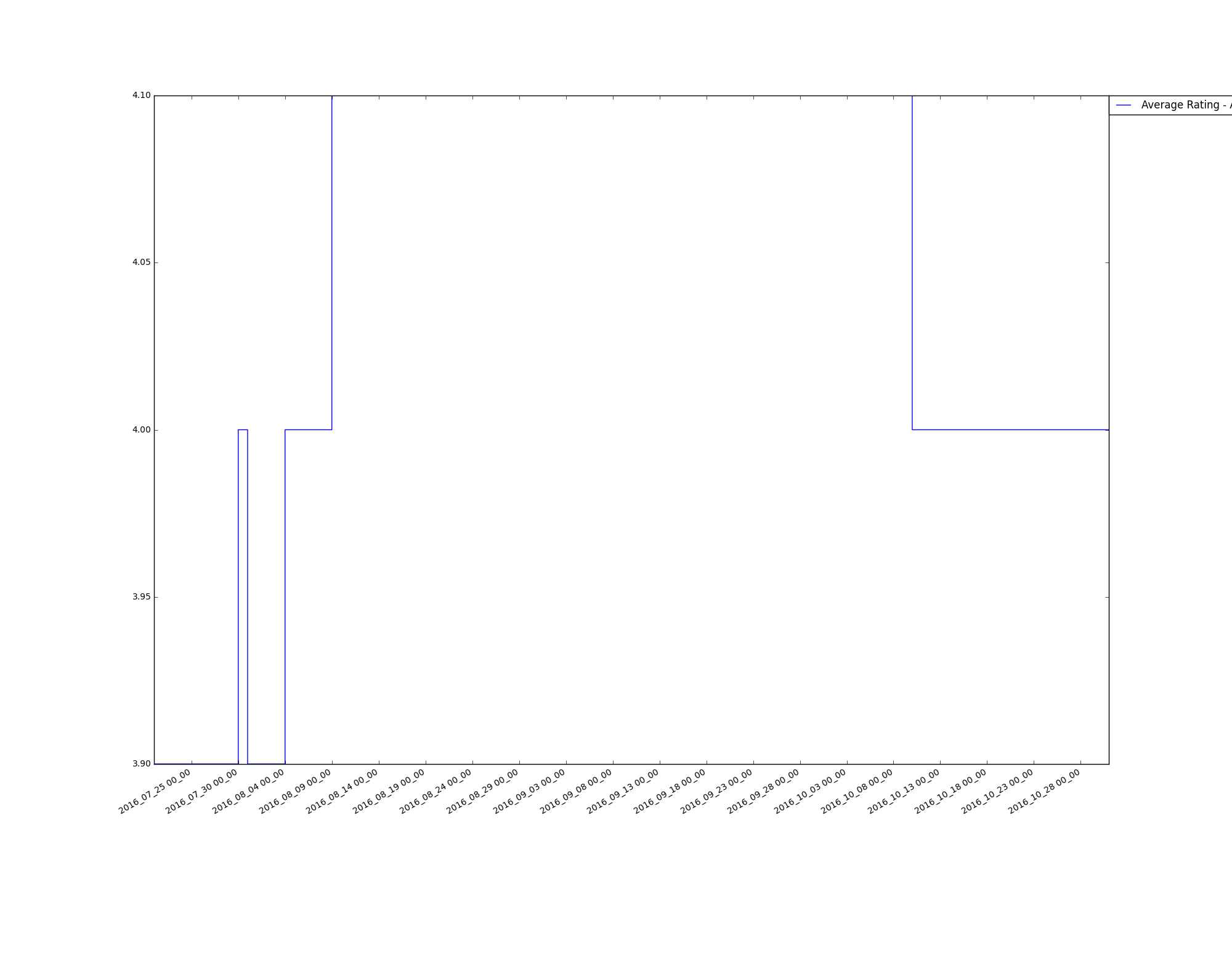


Size –

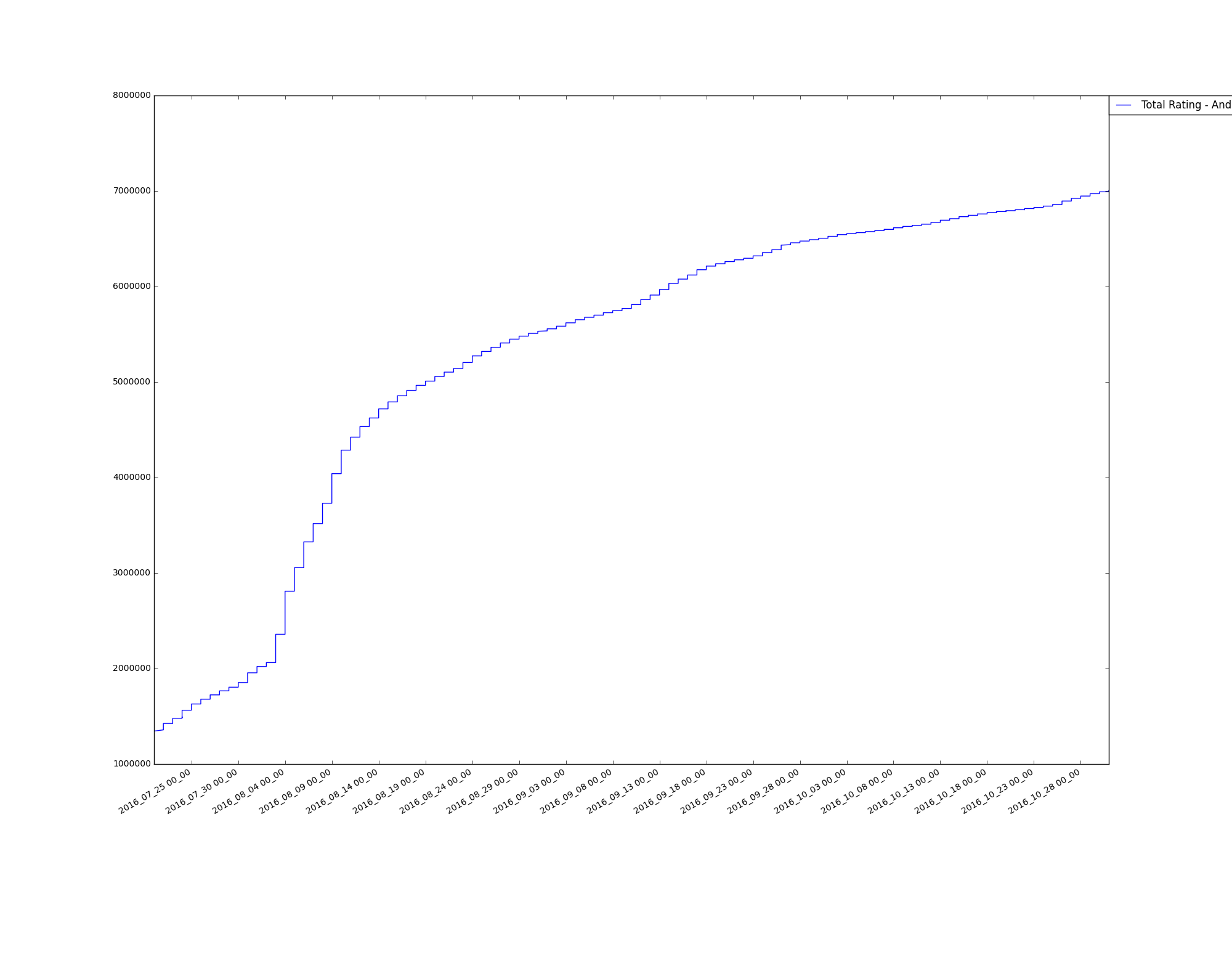


Android-

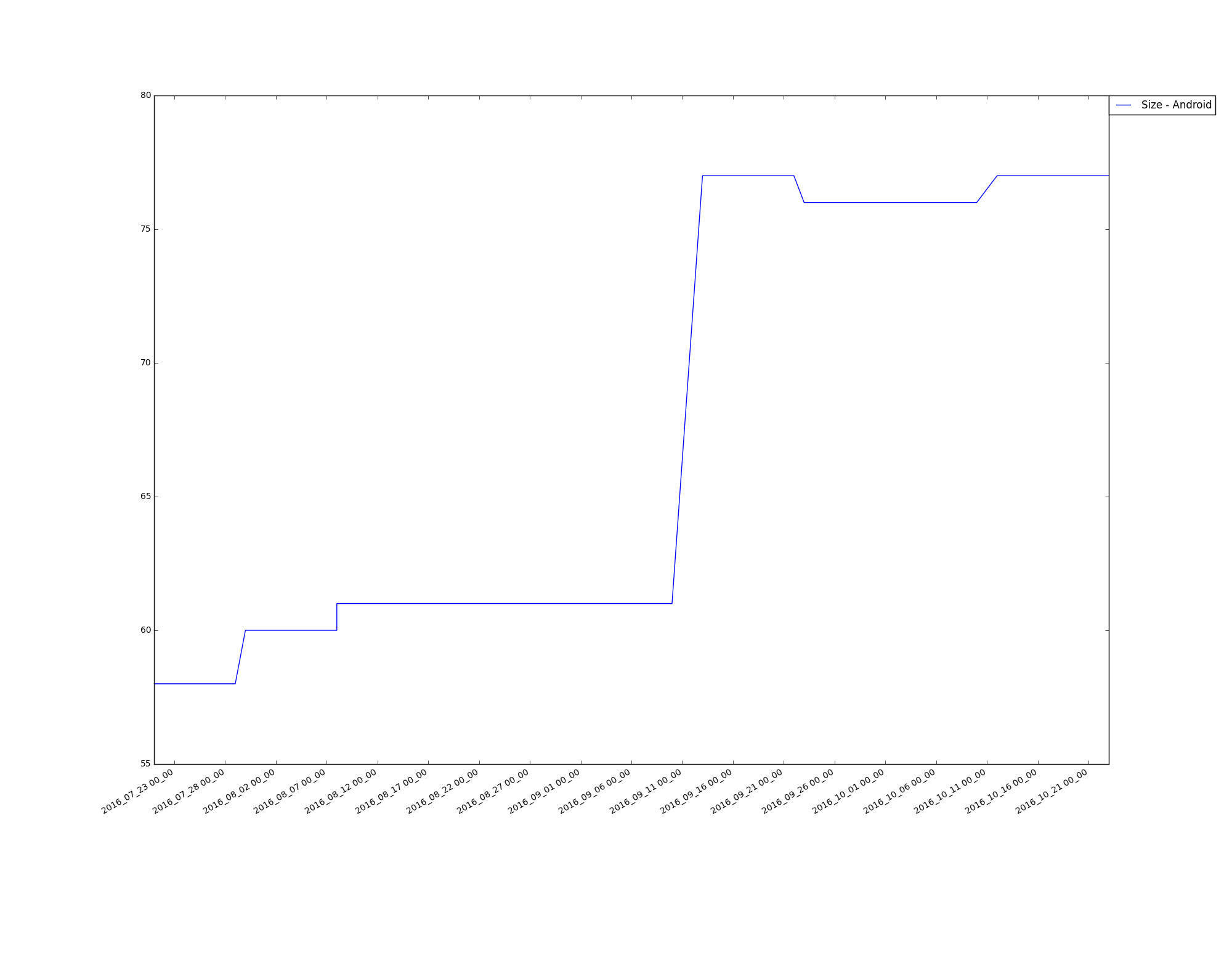
Average Rating -



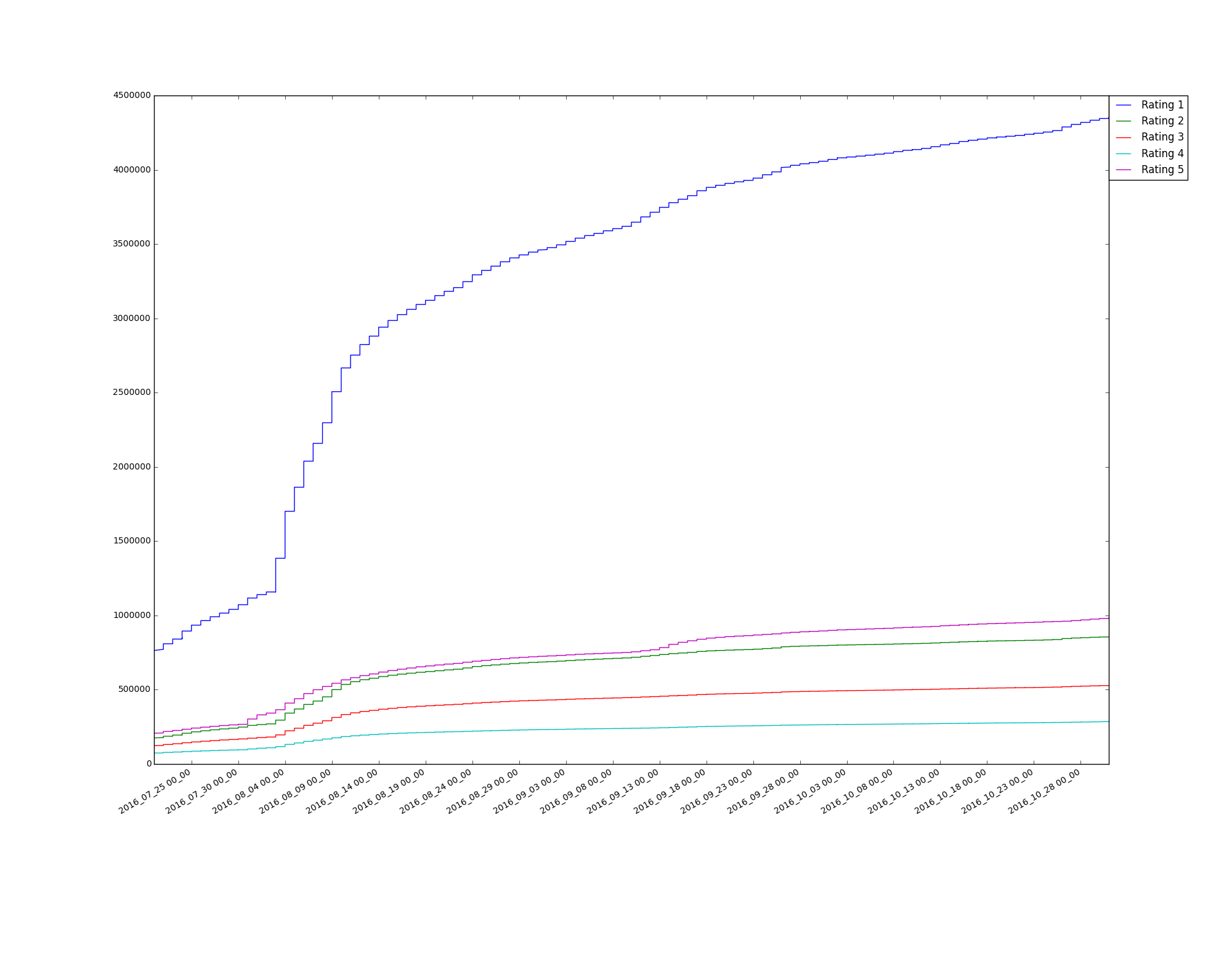
Total Rating -



Size –



Rating Scale –

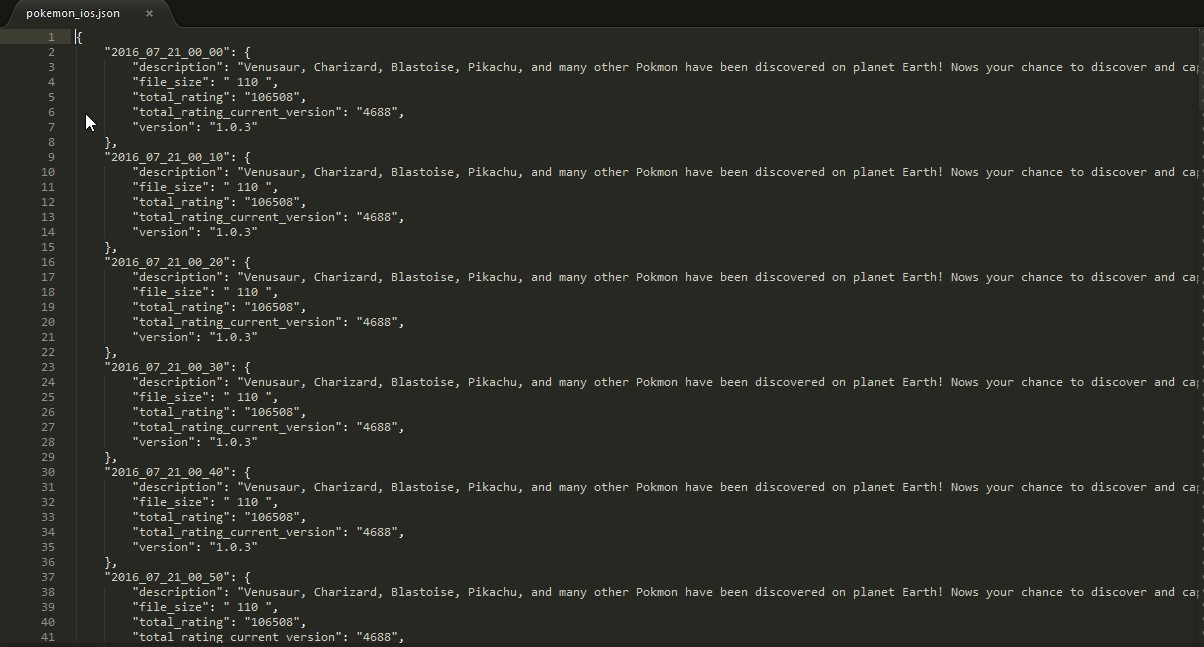


**Part E**

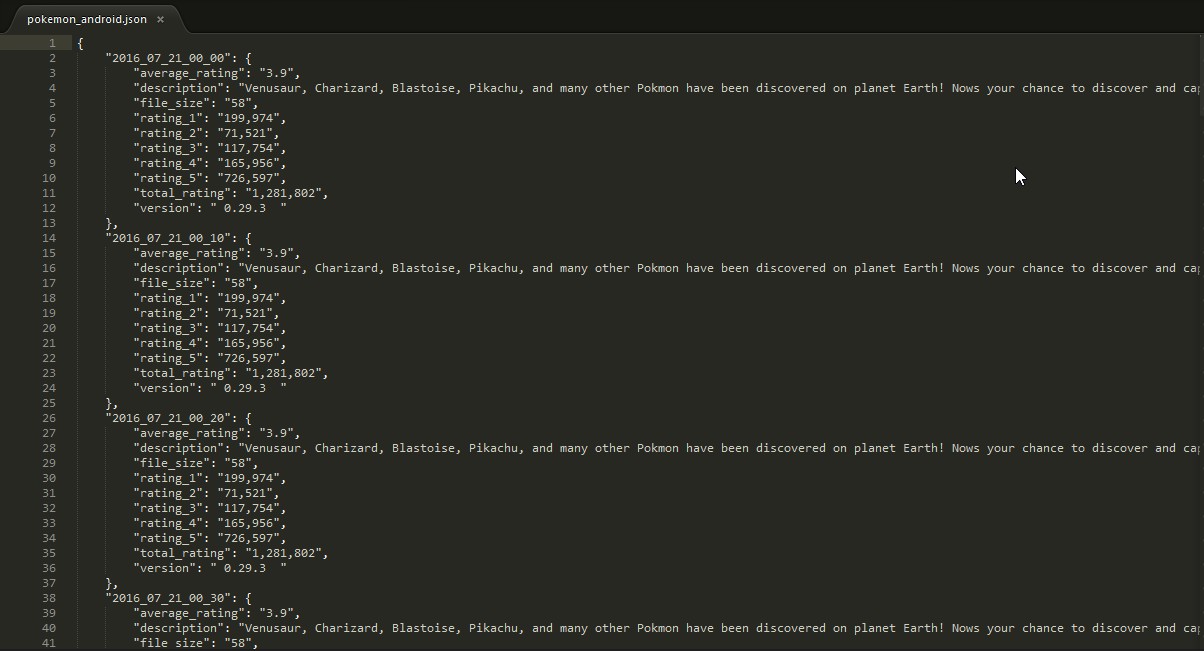
**Construction of JSON files:**

Two files are constructed, namely, pokemon\_android.json” for Android and the other “pokemon\_ios.json” for iOS data. The JSON file’s keys are “YYYY\_MM\_DD\_HH\_MM”, where YYYY is year, MM is month, DD is day, HH is hour, MM is minute, and PLATFORM is either iOS or android. And values are dictionaries of the collected data for the corresponding HTML file

iOS JSON file –



Android JSON file –



.**Part F**

**Deep Learning - Extraction of tag lines with corresponding probabilities:**

This step helps us in understanding the screenshots of the app. There are multiple images in each app page. We downloaded all the screenshot images from iOS and Android web pages. For each image we were able to extract the tags with the corresponding probabilities using the TensorFlow.

Image 1-



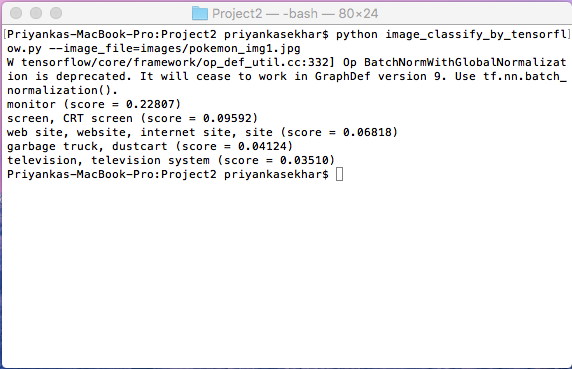


Image 2 -



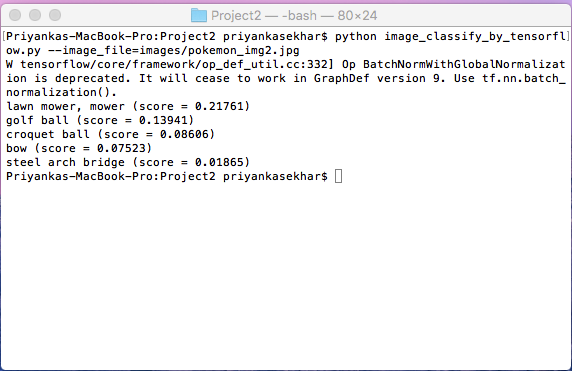


Image 3-



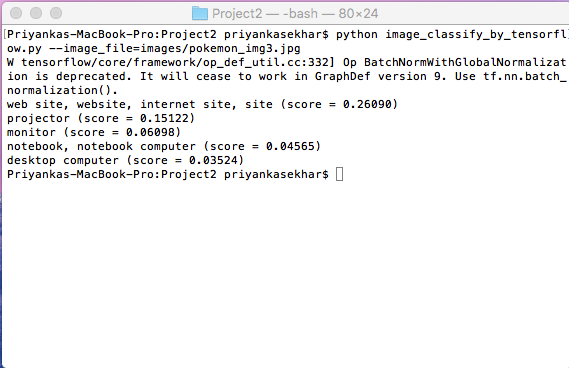


Image 4 -



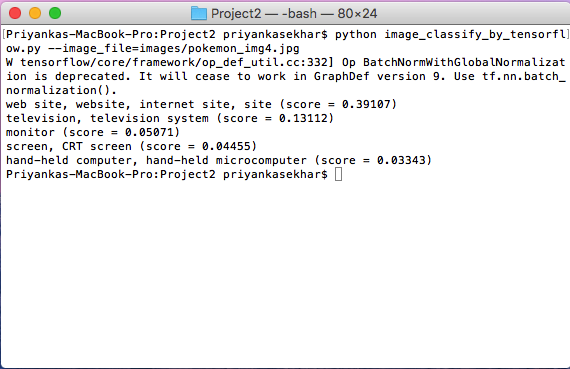
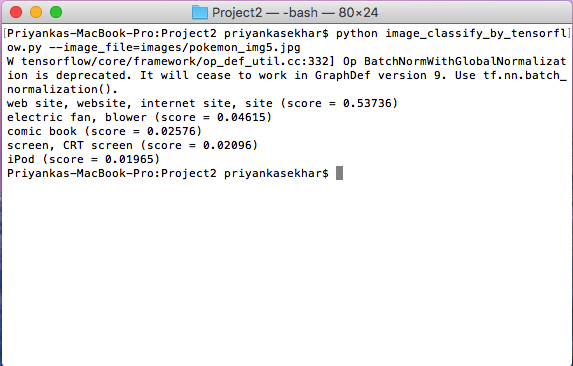


Image 5 -

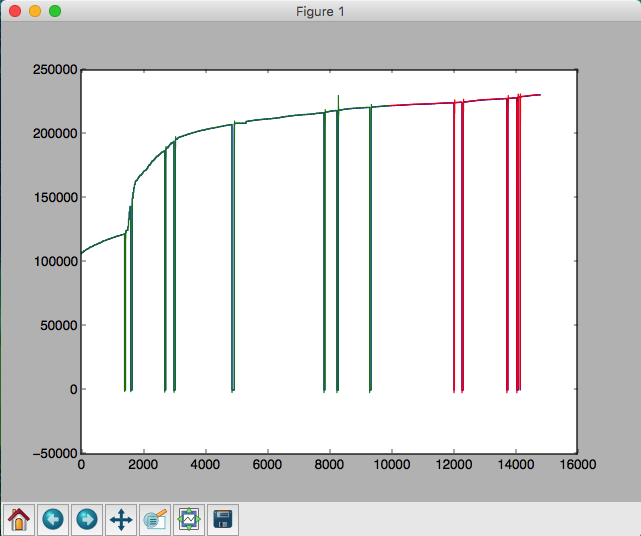


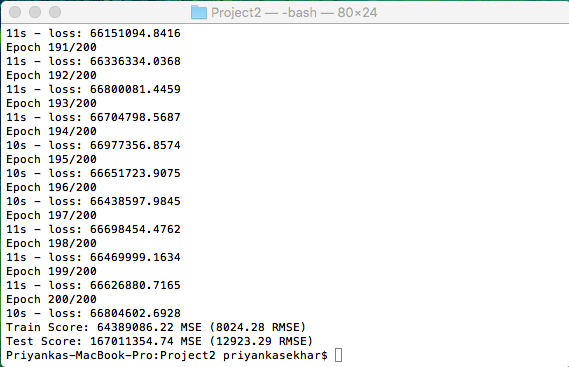


**Part G**

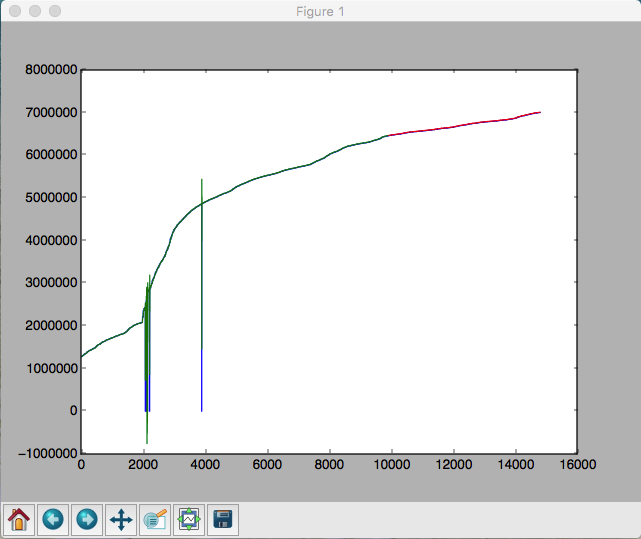
**Predictive model in sklearn:**

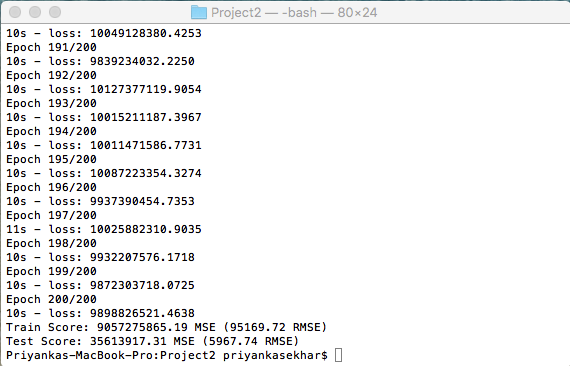
iOS –

****

****

Android –



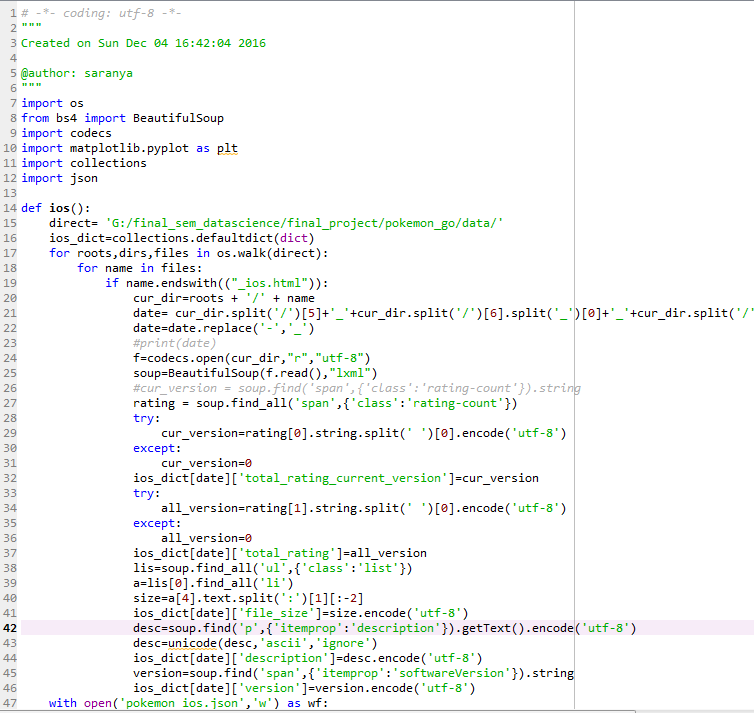


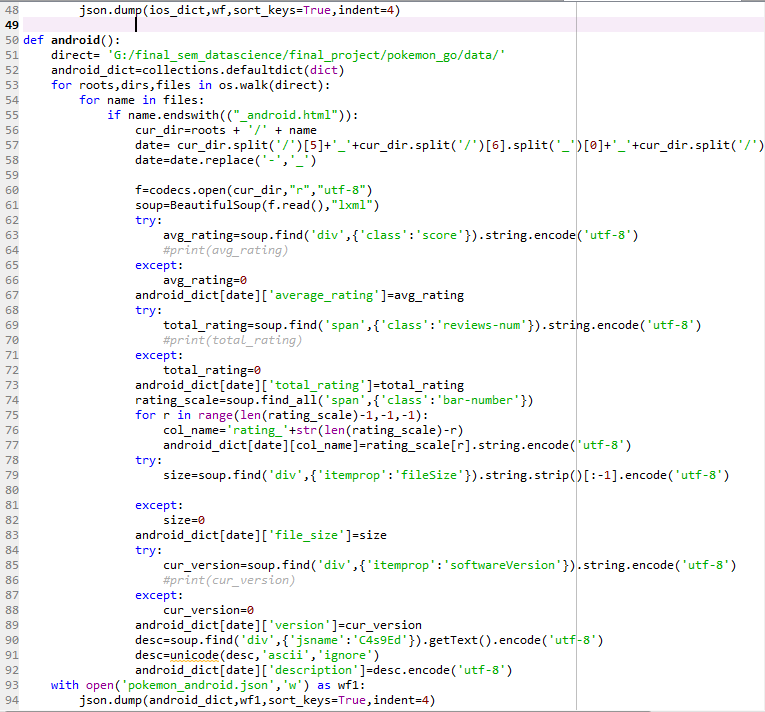
Note: The above prediction model will provide a way for us to make predictions about future dates. Thus we can find the predictions for Nov 1, 2016 11:50 am using the same and also factoring in the root mean-squared error for each future prediction

**PYTHON SCRIPTS**

**Part A, C, E, F**

**pokemon\_scraping.py**

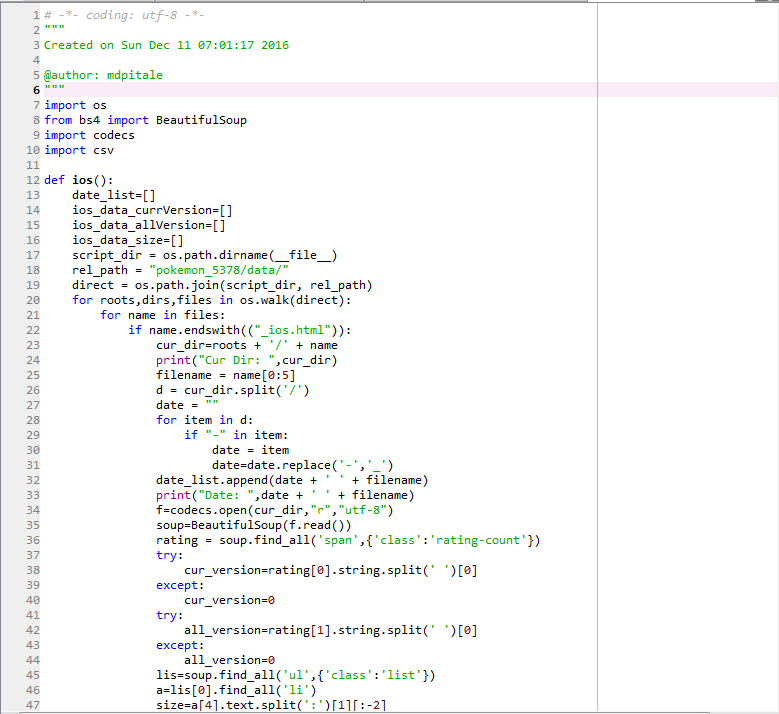
****

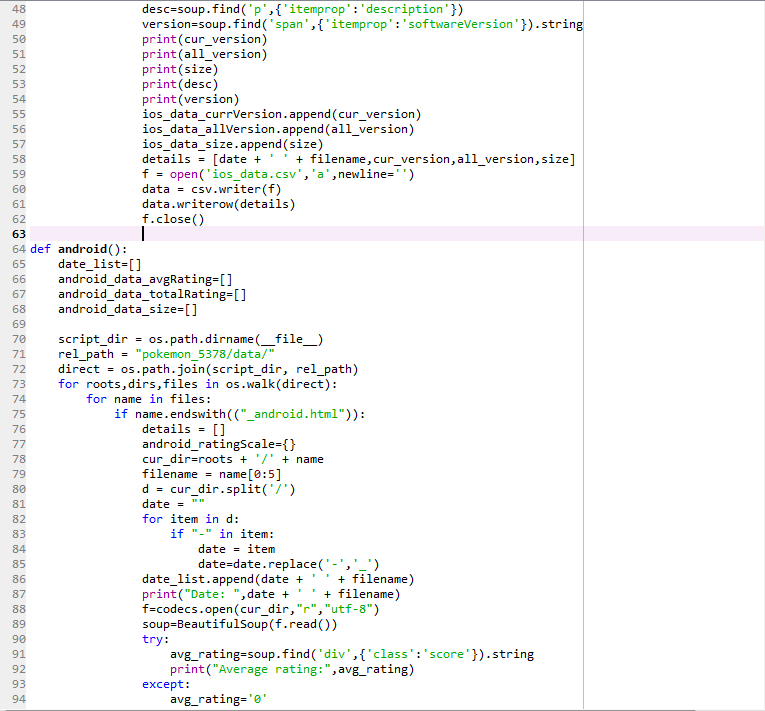
****

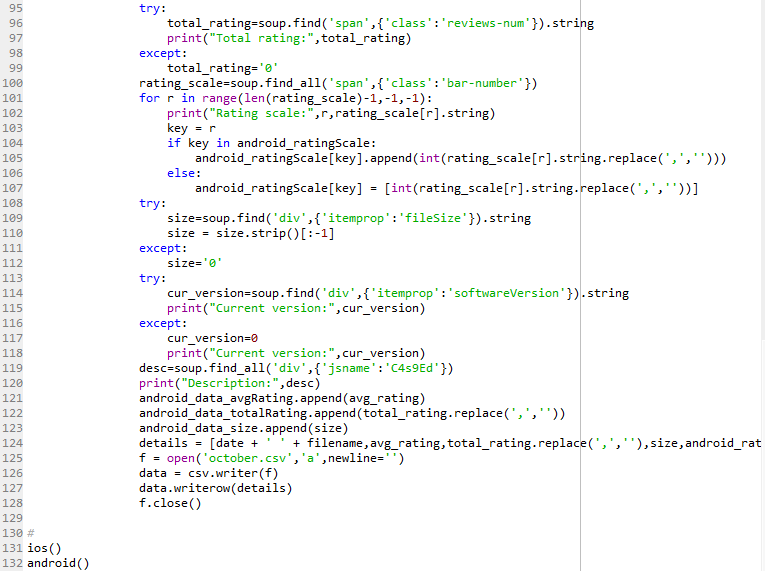
****

**Part B, D**

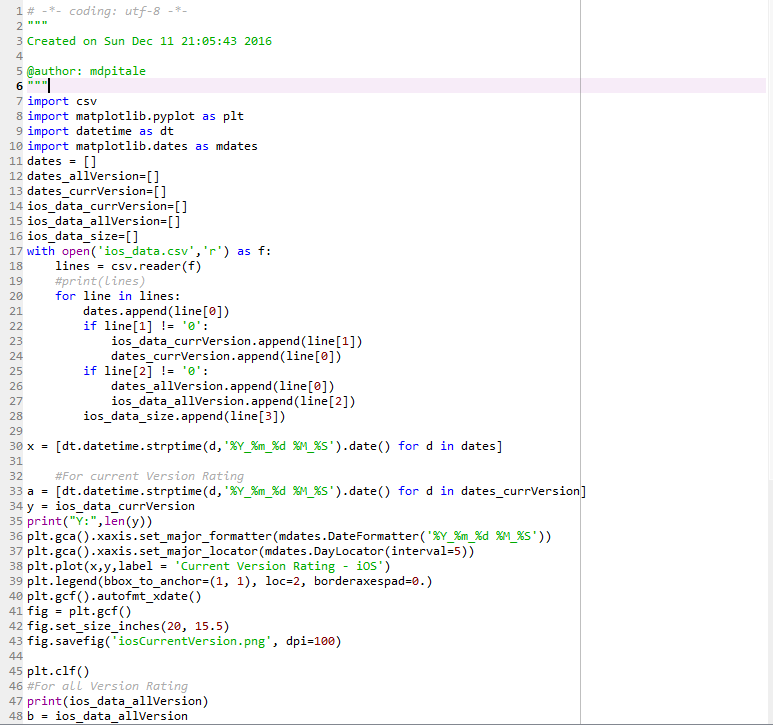
**pokemonWriteFile.py**

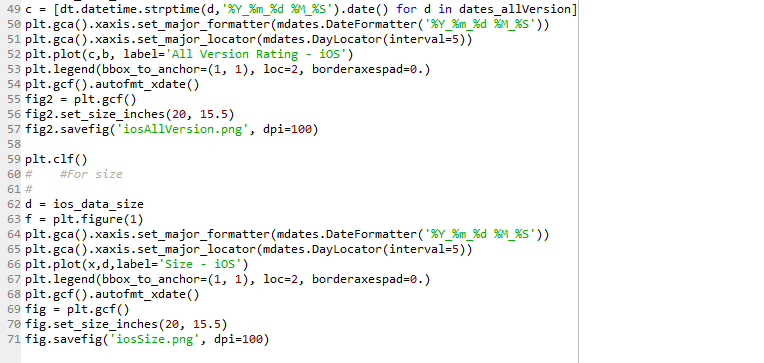
****

****

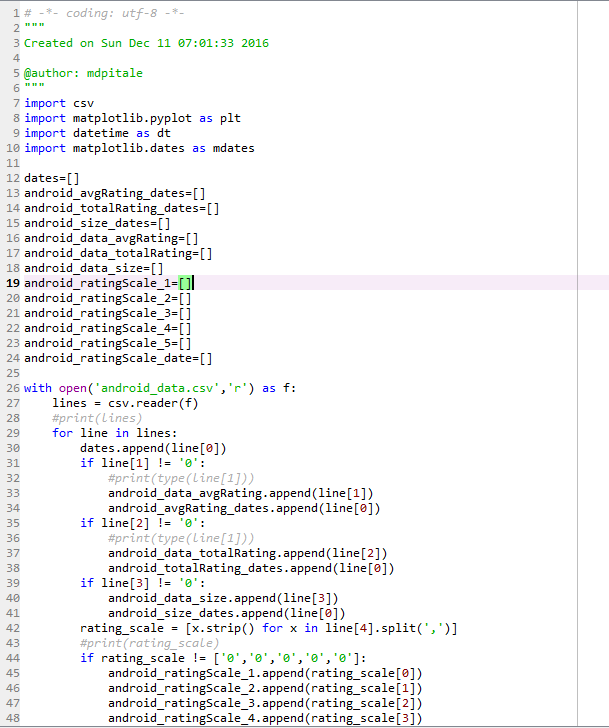
****

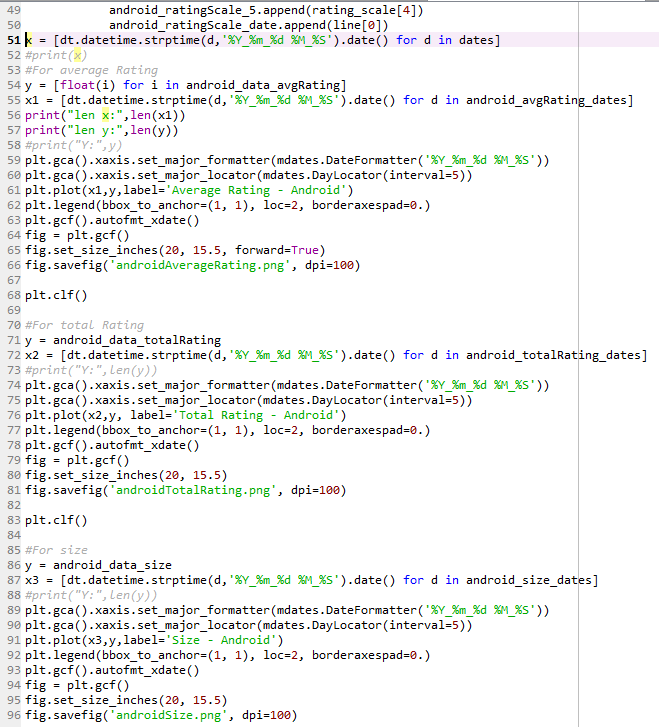
**pokemonReadFileForGraphiOS.py**

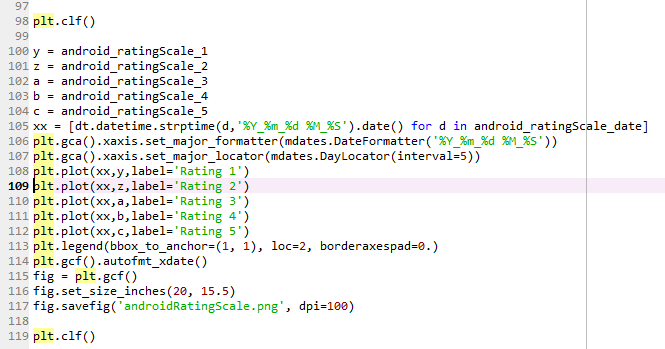
****

****

**pokemonReadFileForGraph.py**

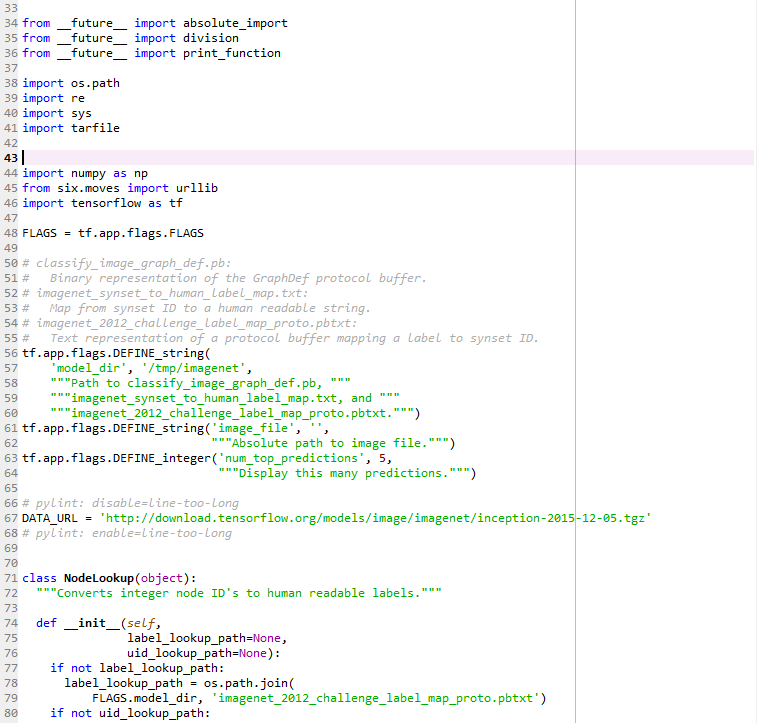
****

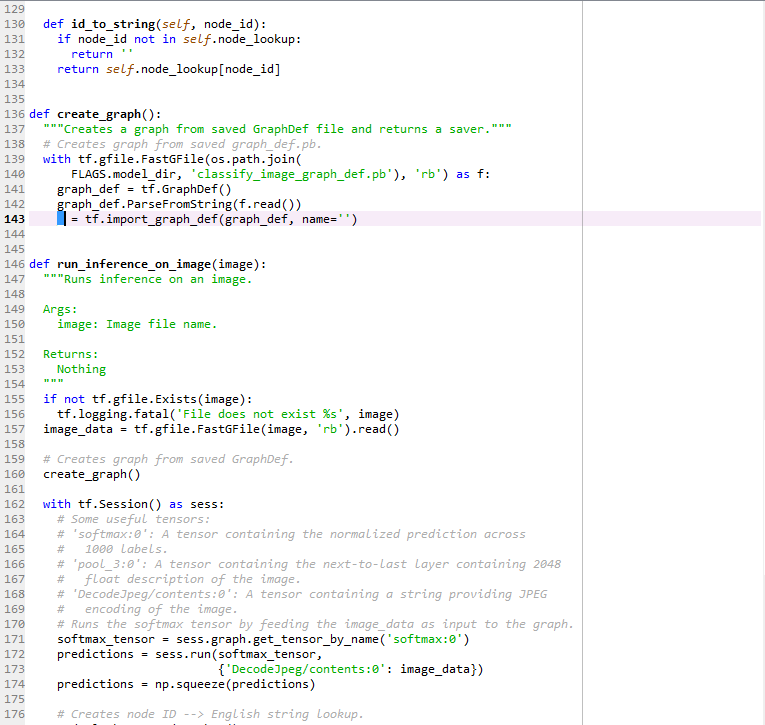
****

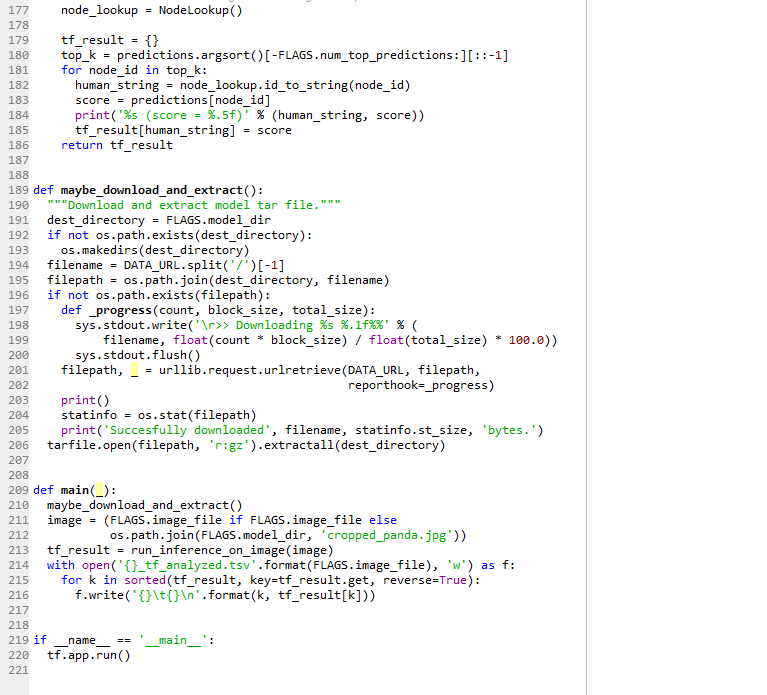
****

**Part F**

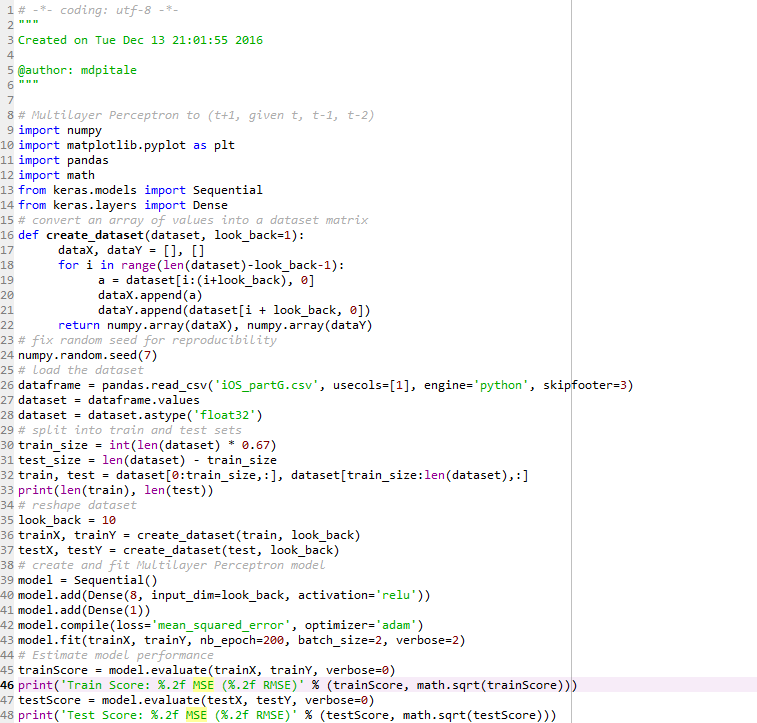
**image\_classify\_by\_tensorflow.py**

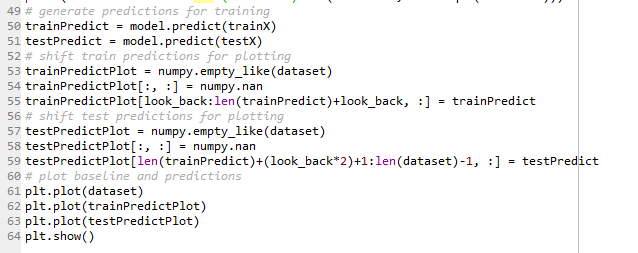






**Part G**





***Note:*** We have all the scripts separate due to the reason that because of the amount of data, the machines that we were using were not powerful enough and used to go out of memory. Therefore, the scripts are separate and the data has been written to a csv file and then read. We have attached the csv files as well.

**CONCLUSION**

We presented a system for one of the most popular game, Pokémon Go! Analytics. We used the BeautifulSoup, Python, Matplotlib and TensorFlow and scraped the Pokémon Go iOS and Android website, visualized the numeric data received by scraping and conducted a simple deep learning task with TensorFlow