

INSY 5378-002: Data Science: A Programming Approach

Spring 2017

Group Project 2: Pokémon Go! Analytics



Members:

|  |  |  |
| --- | --- | --- |
| **Name** | **Major** | **Email** |
| Anish Grandhi | Information Systems | anish.grandhi@mavs.uta.edu |
| Meghna Goswami | Information Systems | meghna.goswami@mavs.uta.edu |
| Vinothkumaar Udhayakumar | Industrial Engineering | vinothkumaar.udhayakumar@mavs.uta.edu |

Table of Contents

[I. Introduction 3](#_Toc481773102)

[II. High level Codes Description 4](#_Toc481773103)

[III. Data exploration 5](#_Toc481773104)

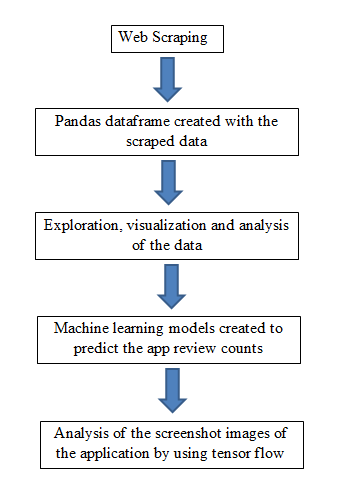
[IV. Prediction Model 11](#_Toc481773105)

[V. Deep Learning 12](#_Toc481773106)

[VI. References 17](#_Toc481773107)

# Introduction

Pokemon Go is an augmented reality, free-to-play game which was initially released on July 6,2016. It soon became an instant hit amongst user groups of all ages and was downloaded by several users on their Android and iOS mobile devices. In this project, we aim to understand the success of this game. The following steps have been taken in order to fulfil our aim:



# High level Codes Description

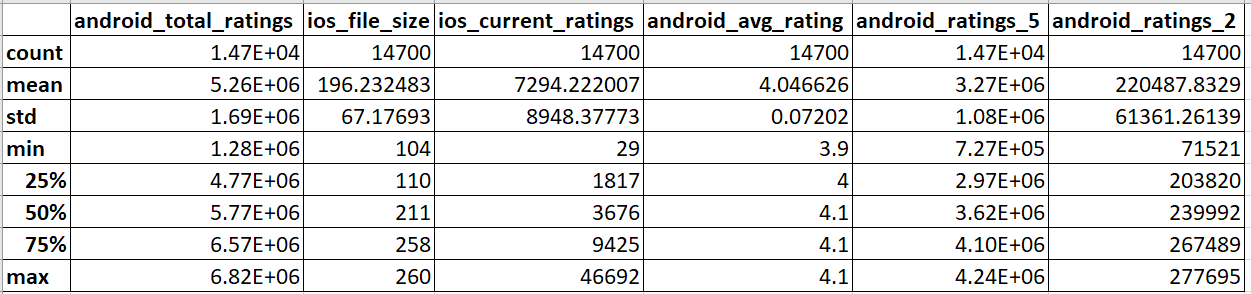
* Scraping and storing.py – This file is used to scrape all the data from the HTML files in the directory using lxml. Once the data is scraped and loaded into memory, it is loaded into pandas dataframe. Using pandas inbuilt functions, the dataframe is stored in json, csv, xlsx formats.
* Data Exploration.py – This file loads all the data from the json file and then into pandas dataframe. Pandas describe function is used to find the information such as mean/min/max etc. Scatter matrix is plot and Pearson’s correlation coefficient between the top 20 pairs are printed. Plot for file sizes, android ratings and ios ratings are plot separately for clarity.
* Regression Models.py – This file loads the data from json to pandas dataframe. The missing values are filled using pandas interpolate function which automatically fills the gaps with the average of previous and succeeding values. Various regression models were tried and we found that ElasticNet model gives the best accuracy after cross validating through sklearn’s time series split method.
* Extract links and store images.py – This file is used to scrape all the links of screenshot images from the HTML files.

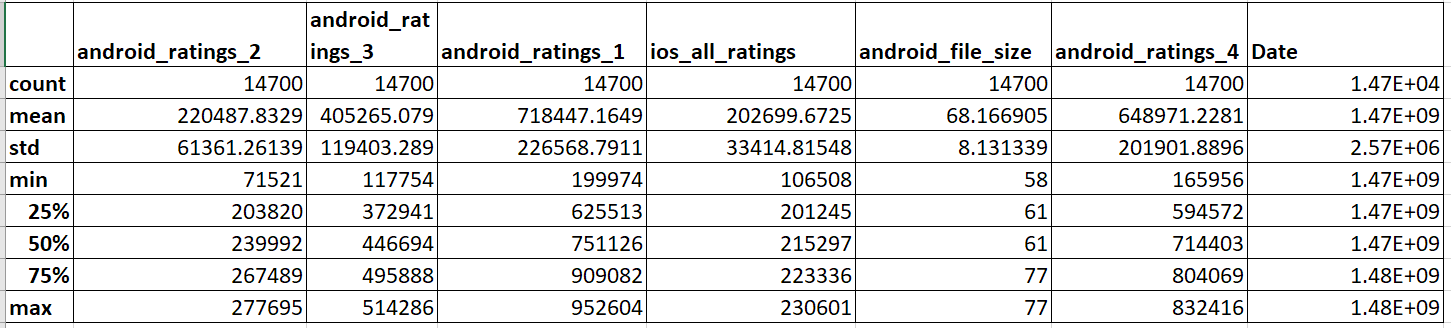
# Data exploration

After creating the Pandas dataframe, we used pandas inbuilt functions to explore the data:

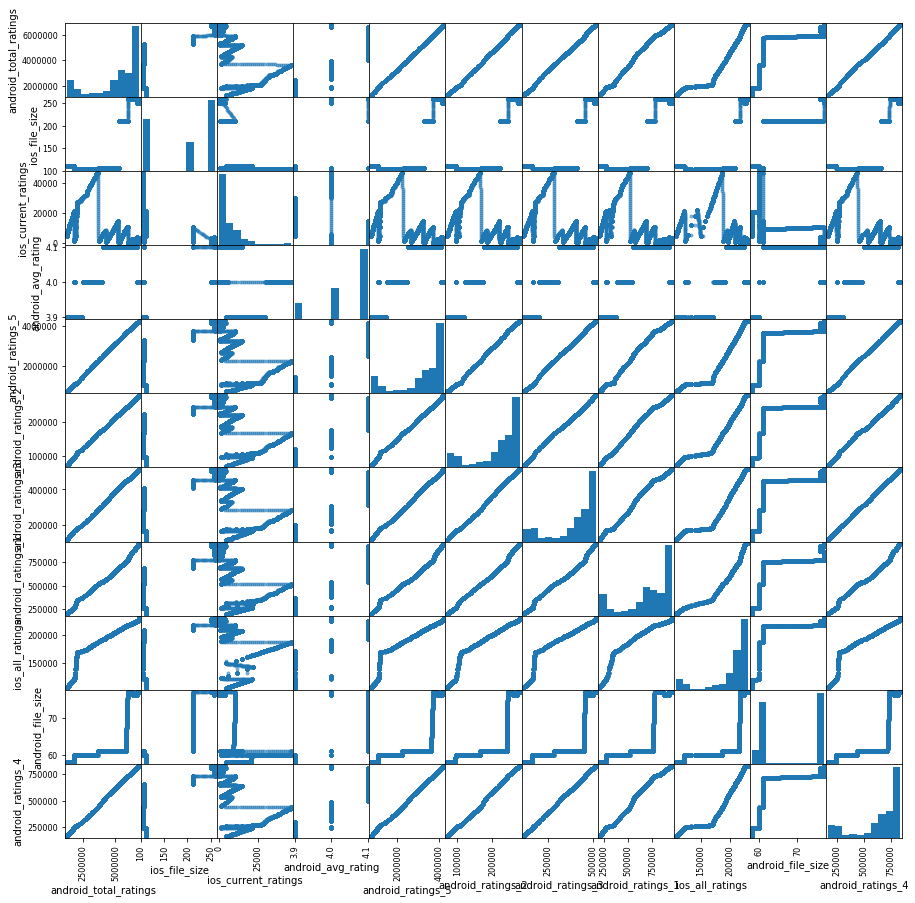
* Used dataframe.describe() method to find the count/mean/std/min/25%/50%75%/max

values for each 11 variables scraped from the html files and a date column that has been added for prediction.





* Used scatter\_matrix() method to find correlations between variables.



From the above scatter\_matrix(), we can see that a lot of variables have positive correlations between them – namely the android ratings. This is obvious because all the individual android ratings add up to the total ratings for the android. The iOS Current ratings has no trend because it resets to zero once the new version is released in the app store.

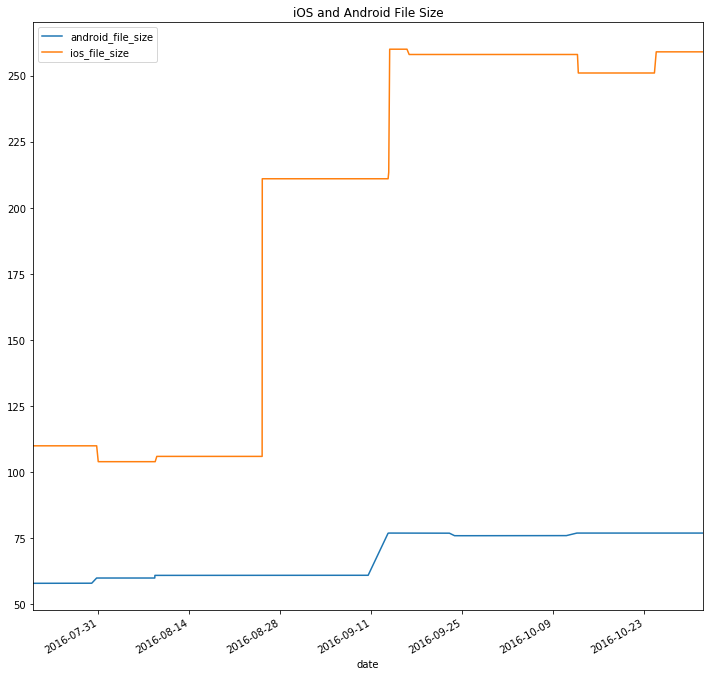
To find the individual value of the correlations between them, we use Pearson’s correlation method in the next step.

* For the top 20 highly correlated identified pairs, we calculated the Pearson’s correlation coefficients.

|  |  |  |
| --- | --- | --- |
| Top Absolute Correlations | | |
| android\_ratings\_3 | android\_ratings\_4 | 0.99989 |
| android\_total\_ratings | android\_ratings\_5 | 0.999845 |
|  | android\_ratings\_4 | 0.999713 |
| android\_ratings\_5 | android\_ratings\_4 | 0.999694 |
| android\_total\_ratings | android\_ratings\_2 | 0.999663 |
| android\_ratings\_5 | android\_ratings\_3 | 0.999599 |
| android\_total\_ratings | android\_ratings\_3 | 0.999567 |
| android\_ratings\_2 | android\_ratings\_3 | 0.999488 |
| android\_ratings\_5 | android\_ratings\_2 | 0.999482 |
| android\_ratings\_2 | android\_ratings\_4 | 0.999399 |
| android\_total\_ratings | android\_ratings\_1 | 0.994736 |
| android\_ratings\_2 | android\_ratings\_1 | 0.994189 |
| android\_ratings\_1 | android\_ratings\_4 | 0.992961 |
| android\_ratings\_5 | android\_ratings\_1 | 0.992891 |
| android\_ratings\_3 | android\_ratings\_1 | 0.992342 |
| android\_ratings\_2 | ios\_all\_ratings | 0.968107 |
| android\_ratings\_5 | ios\_all\_ratings | 0.964302 |
| android\_ratings\_3 | ios\_all\_ratings | 0.963321 |
| android\_total\_ratings | ios\_all\_ratings | 0.963248 |
| ios\_all\_ratings | android\_ratings\_4 | 0.962669 |

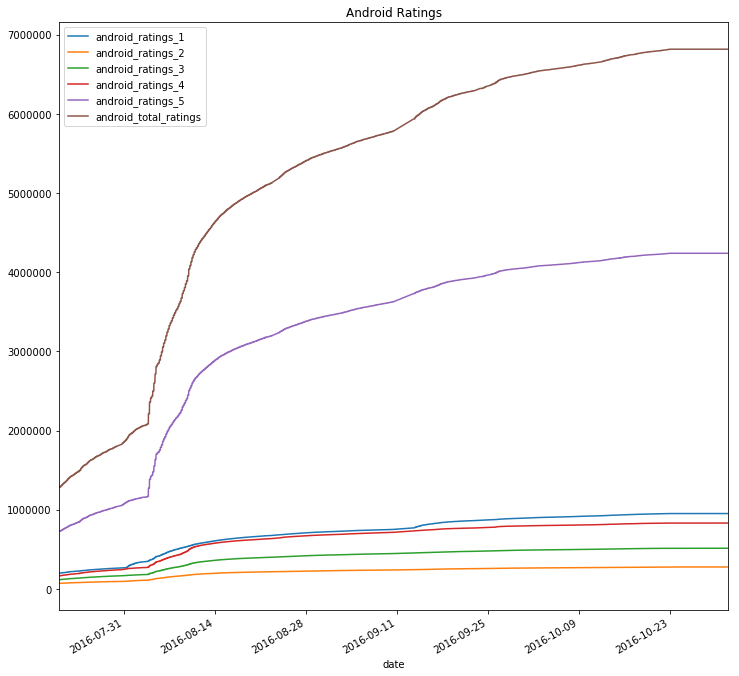
* Used dataframe.plot() to create time series graphs for each of the given 11

variables.



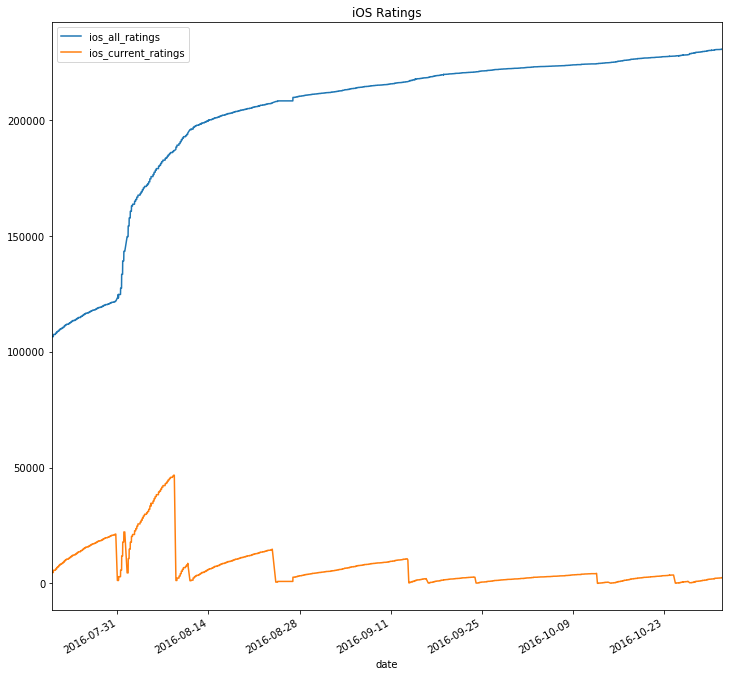
Plot showing time series of iOS and Android file sizes

As the time increases, the android file size shows little increase while the iOS file size increases dramatically.



Plot showing time series of Android Ratings

While the ratings 1-4 have increased a little over time, the 5-star ratings and total ratings increase a lot. It could mean that a lot people have enjoyed the game.



Plot showing time series of iOS Ratings

While the iOS all ratings have shown an increasing trend, the current ratings have no trend due to resetting to zero with every new version of the app that comes out in the app store.

# Prediction Model

For the missing values, we used pandas interpolate function which automatically fills the missing values with the average of previous and succeeding values. Using cross validation, we found out that both interpolate and filling the missing values with the previous values resulted in the same accuracy of the model.

* For the android model, we took the 'Date', 'android\_file\_size', 'android\_avg\_rating', 'android\_ratings\_5' as independent variables and ‘android\_total\_ratings’ as dependant variable. As we had to predict the value for a future date, date remains as an independent variable and after running a couple of trials, we decided upon the other independent variables as they add a lot of weight in predicting the total ratings.

In the cross validation, we used sklearn’s time series split. The kfold method does not work well with time series models as we lose the value of time when the method splits the data into multiple folds. Using time series split has given us good accuracy for the above-mentioned variables.

**The accuracy for various models are –**

Linear Regression – 78.94%

Ridge – 79.7%

Lasso – 78.9%

ElasticNet – 91.082%

Since the accuracy is high for ElasticNet, we decided to use ElasticNet for prediction.

**The predicted value for 2016/11/01 is 6831355.35.**

* For the iOS model, we took 'Date', 'android\_ratings\_2', 'android\_ratings\_4', 'android\_ratings\_5' as independent variables and ‘ios\_all\_ratings’ as dependant variable. Taking ’ios\_current\_ratings’ did not make sense as the current ratings start from zero once the next version of the app is launched in the app store. The other variables have been taken after a lot of trial and testing.

For unknown reasons and lack of time, we could not understand why sklearn’s time series split method did not work well with the ios model. The accuracy of the model was coming out in negative. The least negative accuracy percentage was with ElasticNet (-694%) compared to Lasso, Ridge, Linear Regression.

**The predicted value for 2016/11/01 is 224440.37**

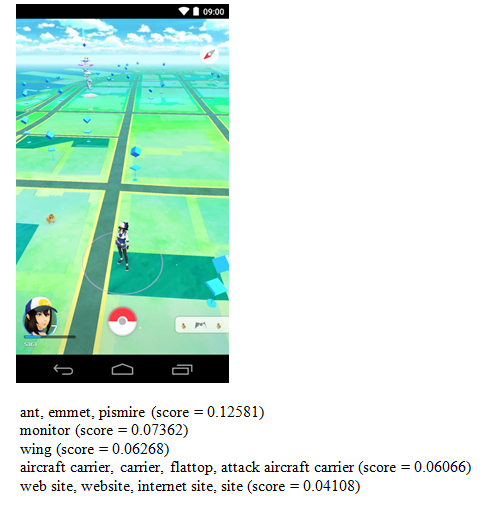
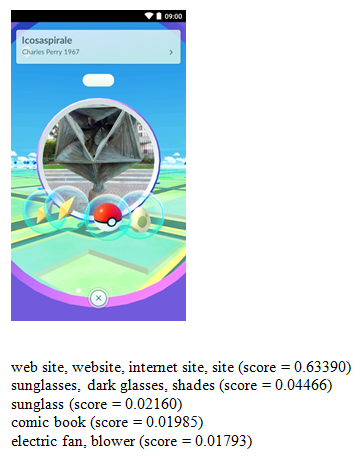
# Deep Learning

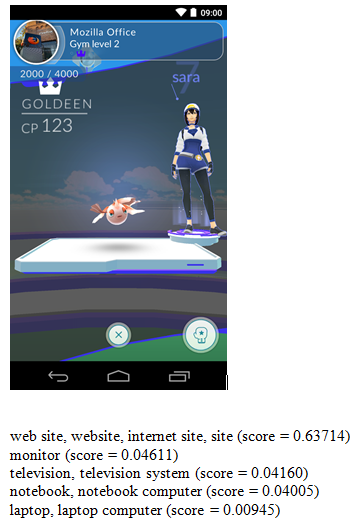
We extracted unique screenshots and used tensorflow to extract tags and probabilities.

Number of unique images in iOS app pages – 14

Number of unique images in android app pages - 5

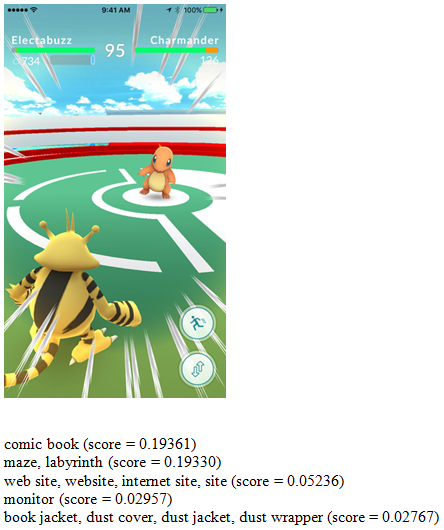
**For Android app images:**

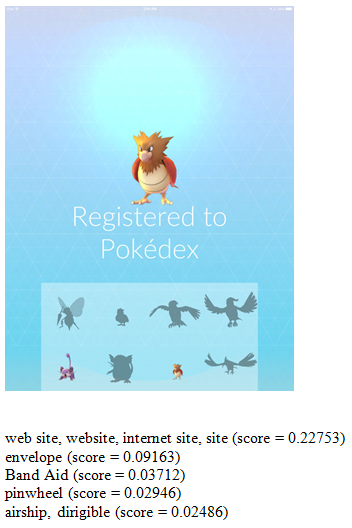
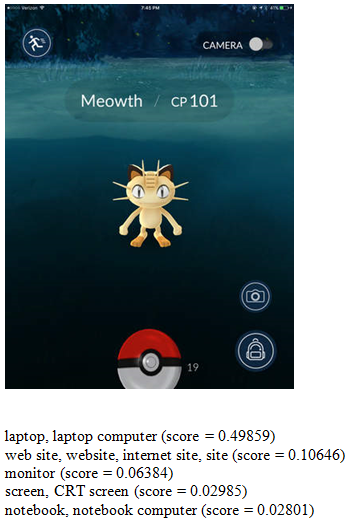
 

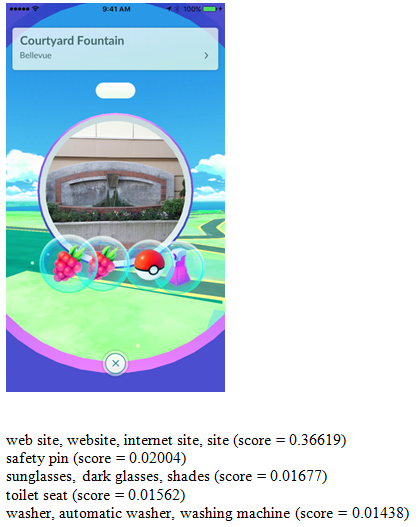
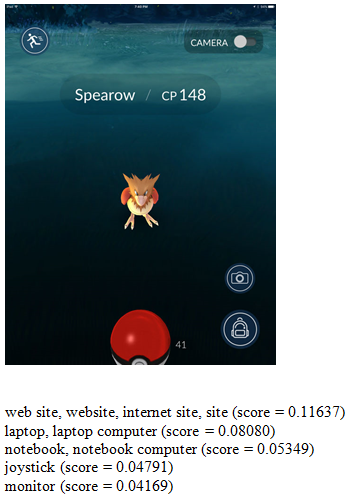
 

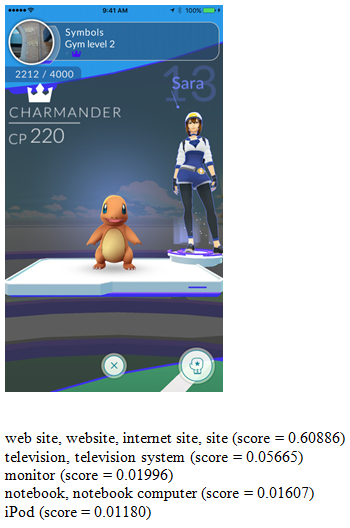


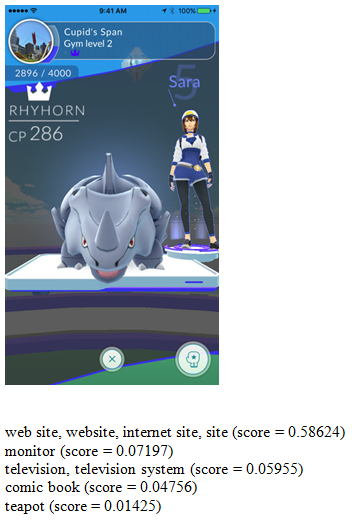
**For iOS app images:**

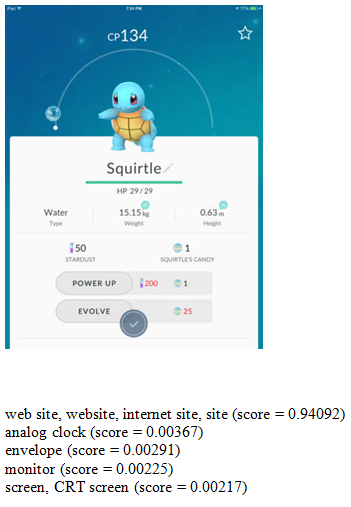
 

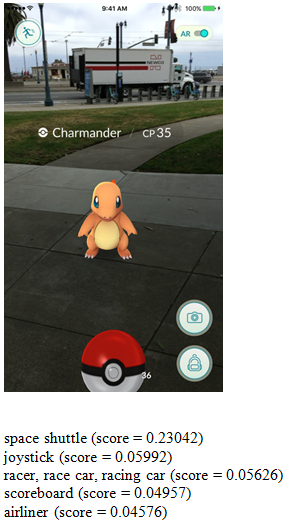
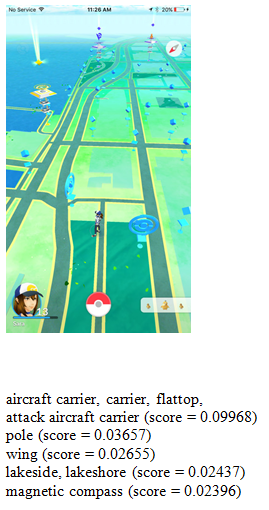
 



# References

* urlretrieve - http://stackoverflow.com/questions/22676/how-do-i-download-a-file-over-http-using-python

Regression models –

* http://machinelearningmastery.com/evaluate-performance-machine-learning-algorithms-python-using-resampling/
* http://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/
* http://stackoverflow.com/questions/41753795/sklearn-timeseriessplit-cross-val-predict-only-works-for-partitions

Get highly correlated pairs –

* http://stackoverflow.com/questions/17778394/list-highest-correlation-pairs-from-a-large-correlation-matrix-in-pandas
* os.walk Reference - http://stackoverflow.com/questions/5817209/browse-files-and-subfolders-in-python
* lxml scraping - http://python-guide-pt-br.readthedocs.io/en/latest/scenarios/scrape/