API User Manual

Part I: Internet Usage Plot

Given a user ID, the API plots the time-series of the internet usage volume for the specific user along with some extra options (Figure 1).

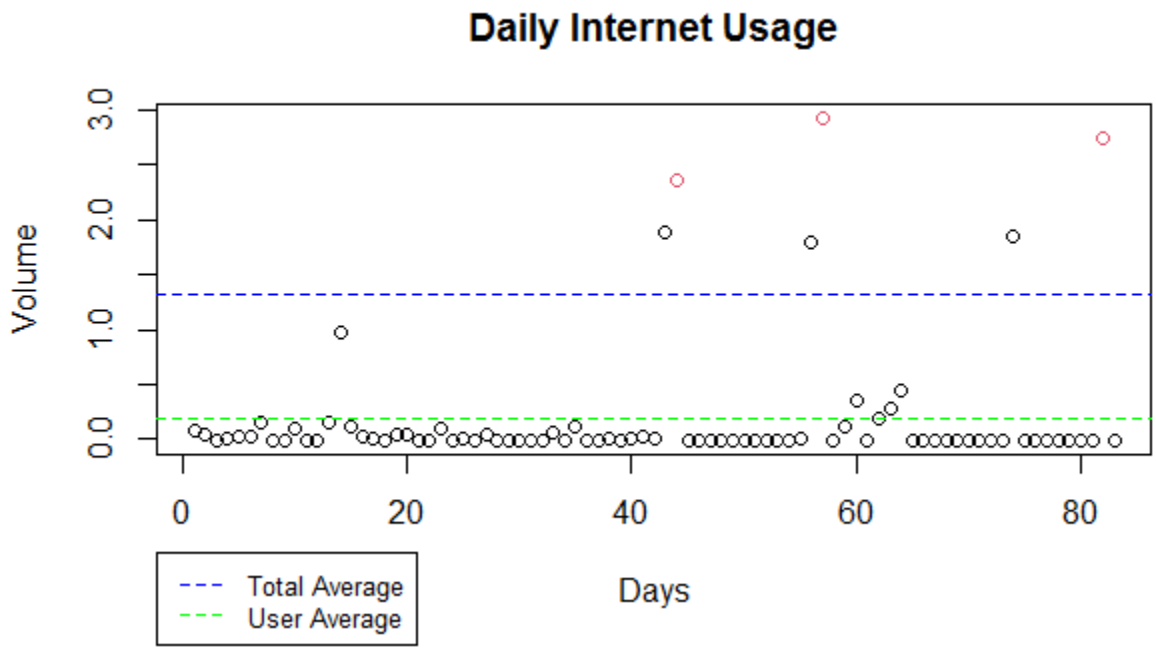

Figure 1. Internet Usage Volume Plot for the User "-06FoayTOXJ8-".

As it can be seen from the picture above, average internet usage of all users (blue line) and the average usage of the specific user (green line) are added to the plot.

Notice: The legend of the plot works without a problem when I run the code in R, but I've had problems when I ran the API on the server showing the legend. So, for now let's note what blue and green lines mean in the plots till I manage to fix the problem for the API.

Considering the "Three-Sigma Rule of Thumb", 99.7% of the values shall fall between [mean-3*σ, mean+3*σ]. Any values outside of these boundaries are considered abnormal. Using this rule, the points that are colored in red, indicate that the internet usage of the user for that day is abnormal. Note that I've only considered the right hand

side of the boundary as it was the case with our dataset (I haven't seen any abnormal usage in the negative side but it can be easily added to the code).

The API can be tested at:

http://51.89.111.146:8000/plot?ID=-06FoayTOXJ8-

And the user ID can be changed to any ID that is available in the dataset.

Part II: Predictive API for a Single User

At this point I've only used TBATS, ARIMA and SES models to test the API on the server, as the size of the other models stored on the disk were mostly exceeding 20mb. But I can bring them to you, so that you can upload them on your own server. The R codes to train and store these 3 models can be found on "Storing_Models.ipynb" file. Yet again the API can be tested at:

http://51.89.111.146:8000/Pred?ID=-06FoayTOXJ8-

And just like the plotting API, the user ID can be changed to any ID that is available in the dataset.

Notice: When I run the code in R, the output looks like this, but running the API on the server ignores the column names and returns 10 vectors with 4 dimensions. For now let's keep in mind that the order of the output vectors are shown below:

Actual	TBATS	ARIMA	SES
-0.01397663	-0.1099907	-0.08964646	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326
-0.01397663	0.1869711	0.19409675	0.1977326

Part III: Predictive API for a set of Users

Didn't include this one in the repository, as I had some problems that need to be addressed. Mainly the way I present the "set" of users to the function. There are a handful of ways this procedure can be carried out but each one has its own pros and cons. So we need to discuss this before finalizing this section.

Part IV: More Info

- A list of unique user IDs are gathered in “User_IDs.txt” to make things easier.
- The API is created using “plumber” package. If you had difficulties calling the API with Python, this video here will help:
  https://www.youtube.com/watch?v=Z2Aofr4UIFY
- At this point I’ve only used the “forecast” library in R to predict the future values to make things easier. So you need to install this package before running the codes. When you confirm we are on the right track and we discuss the details, I will add other packages to the API as well.