INSTRUCTIONS

**For Static**

1. *avro\_to\_csv.py* converts avro (.avsc) file to a csv file containing all the raw data . It uses **\*\***splunk-enterprise local version so make sure to edit and add your credentials .
2. *flatten\_csv.py* takes input from the step 1 ,i.e. raw data csv and process it and flattens it . It also asks for version so please specify and also if you want version to be “3.0” then enter capital o not zero like “3.O” not “3.0” and save it with appropriate name
3. (optional) if you want to compare and merge two different versions then get 2 different flattened file from step 2 and feed them to *compare\_And\_merge.py*  to get a csv containing joint data of the two versions
4. *Smartsheep.py* it converts csv from step 2 or step 3 into a csv file that is google doc / smartsheet compatible and easily understandable and this file is fed to **web-app** to generate the webpage with different color scheme and other options

**\*\* Splunk-enterprise has limit of reading 10,000 bits by default and rest are truncated so to use the script follow these steps to avoid truncation :**

1. **Go to Splunk > etc > system > local > props.conf**
2. **Set TRUNCATE=0**
3. **Restart Splunk . In Splunk Web, go to Settings > Server controls**
4. **Select "Restart Splunk"**

**\*\* If using splunk cloud or splunk enterprise web UI then convert your .avsc file into splunk readable json file using *tojson.py* and then add it to the splunk and then perform the following search**

host="your host name"| stats values as \* by name

**and the export the results into csv file.**

**For Flask-project**

1. Install all the requirements present in requirements.txt file using “pip install requirements.txt”
2. cd to flask-project
3. directly execute python run.py or export FLASK\_APP=run.py and then flask run
4. if using linux then use gunicorn and nginx to deploy this on server