Chapter 1 : Tasks Accomplished and Scripts Developed

D3 visualizations:

Landing page: [website]/teams/team_12.html

- **Static Visualizations[10]:-**
  - teams/team_12/**team_12.html**
  - teams/team_12/html/**world-sightings.html**
  - teams/team_12/html/**population.html**
  - teams/team_12/html/**airports.html**
  - teams/team_12/html/**cancer.html**
  - teams/team_12/html/**airpollution.html**
  - teams/team_12/html/**top_cities.html**
  - teams/team_12/html/**Airport.html**
  - teams/team_12/html/**Shape.html**
  - teams/team_12/html/**UFO_sightings.html**
  - teams/team_12/html/**Location.html**
- **Dynamic Visualizations[2]:-**
  - teams/team_12/html/**pie.html**
  - teams/team_12/html/**time.html**

  Each page on the website has accurate descriptions of the tasks, shows results, visualizations and records inferences from these visualizations. (please refer to these links for inferences from each visualization)

  Note: Dynamic Visualizations try to fetch data from solr first, and in case Solr is down/not up and running/index files not loaded, they use small static data to keep the visualizations active.
- **Data Preparation and cleaning code:-**
  - **jsonGenerator.py**
    - Used to generate different json files used as input to d3 visualizations.
  - **Helper.py**
    - Used to clean extracted data where ever necessary, for eg - json to tsv converter, data ordering etc.

Solr Injesting

- **For D3:-**
  - **solr_add_json.sh**

    Curls solr to fetch v2.json and runs fetch_data.py to fetch json objects for shapes and time-months for the two dynamic D3 visualizations
  - **fetch_data.py**

    Fetches and cleans v2.json to only pick relevant shape related information, month and time related information and sorts the results in

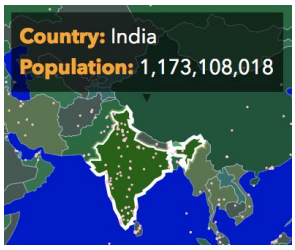
desired formats for dynamic D3 . Stores the extracted information as json objects in files in ufo.usc.edu/teams/teams_12/data. This data is then fetched for visualization.

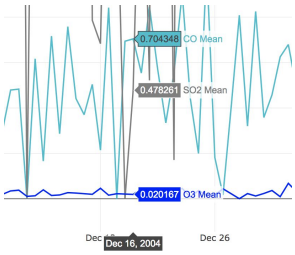
- **For Imagecat:-**
  - **Images_abs_path.py**
    Writes the absolute paths of all the UFO images to a file

Chapter 2 : Inferences

Other Inferences:

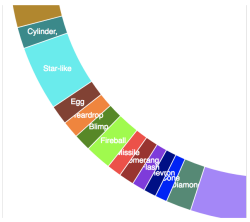
1. **Why did you select your 10 D3 visualizations?**
   The main objective was to make sure we covered all our features extracted and portray results in the best way they make inferences. To accomplish this, we kept in mind, our initial assumptions about UFO sightings and our different datasets. This helped us filter out problem statements we wanted to address and questions we wanted to answer. Thus we chose the following:



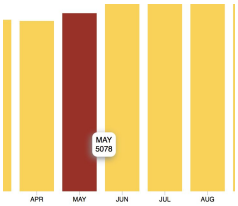


   - World map - to show all UFO sightings in our records
   - Time Series - to show how Pollution from 3 types of gases was influencing UFO sightings and reports
   - USA Map for Airport Density - A chloropeth map showing the sighting counts as percentages in color and plots some UFO sightings unique to regions based on these counts.
   - Cancer, Races and States - A timeline map of Cancer instances in Hispanics, Whites and all races across different US states over a few years. This interactive chart not only shows instance counts, but also allows active filtration of states so one can observe results for a few or all states.
   - Population and USA - This is a map of the US on which we've plot locations of UFO sightings. Each point is one of the four colors(yellow, orange, red, maroon) based on the population counts of the county at that time.

- Top Cities - This interactive graph shows sighting counts over 10 years in the Top 3 cities where UFOs have been sighted.
- Airport Word Cloud - This plots the different airports closest to sightings across the world and helps us understand and make inferences about sightings close to airports



- Shapes and States - this hierarchical cluster of shapes and US states helps us understand which particular shapes and objects are sighted in which states.
- Compound Bar Chart - This shows the counts of sightings in different countries across the world in different years (from 2001 to 2010)
- Location Word Cloud - this visualization utilizes our NER feature for location to show different locations across the world that have been involved with UFO insights and analysis as observed from descriptions of sightings.
- Shapes Pie Chart - This dynamic pie chart gives us an overview of the shapes and objects observed in the sightings.
- Time and Month plot - This interactive dynamic bar chart defines different months and times of days with counts of UFO sightings for us to observe which months see the most traffic and what times of the day are best to observe a UFO.



2. **How are they answering and showing off your features from assignments 1 and 2 and the work you did?**
   These visualizations cover 30 out of 35 of our dataset features. By covering most details, we see that it helps show of our inferences from the previous assignments and also validates or invalidates some of our assumptions. For eg, in the first assignment when we merged pollution with ufo, we thought all three gases (CO, SO2, O3) will have a huge impact on UFO sightings. From our Air Pollution graph, we see that SO2 has the maximum impact and CO, O3 have very less impact.

3. **Did Image Space allow you to find any similarity between UFO sightings images that previously was not easily discernible based on the text captions and object identifications you did?**

Yes, it was able to identify similarity between the images with "potential" UFO sightings with much better accuracy than Tika text captioning and object identification.

Our previous technique on object detection and captioning was successful in categorizing images with a range of similar objects and captions, but there existed cases where images that appeared very similar had little overlap in the suggested captions and identified objects. Consider the example below, to elaborate on the differences:-

[nematode, nematode worm, roundworm, missile, ladle]

[monitor, screen, CRT screen, notebook, notebook computer]

Both the list of identified objects have no overlap, where as the images look pretty similar. But when we used imagespace, it was rightly able to identify the similarity between the two images established higher accuracy in collectivizing the results.

4. **Also include your thoughts about Image Space and ImageCat – what was easy about using them? What wasn't?**

Easy:

1. Once we understand the system, it is really easy to use
2. Modularity-maintaining separate dockers for all the services, changes can be easily incorporated
3. Directory structures were neat
4. Gave better results than Image captioning/Object detection tools

Difficult:

1. Understanding the code took some time and also there were a lot of tools and technologies used, hard time trying to link everything.
2. Existing documentation can be improved and updated
3. Steep learning curve requires more time investment in learning the tools and understanding the code than actually using them to produce results.

Extra Credit 2 Answers:

1. **What types of images are similar? How is SMQTK different than FLANN? Write this information in your report to receive extra credit.**

   Both tools had good accuracy and were successfully able to group similar images.
   These were the same images for which we got the results like ["nematode", "roundworm", "parachute" etc] for object detection and captions like "a man flying through the air while riding a snowboard" when we ran Tike Image captioning/ Object Identification tools. But on the other hand, these images were shown similar by SMQTK:




   We observed that SMQTK and FLANN had better performances than object detection and captioning techniques. However, SMQTK performs better than FLANN or at least as good as FLANN in most cases. SMQTK is slightly faster and has more accuracy (though just by a little) than FLANN.
   We believe that this difference arises due to the underlying difference in algorithm and pipeline implementation. SMQTK evaluates similarity based on near duplicates where as FLANN uses K-means clustering to compare each image with every other image thereby constructing a dictionary {id:"" , features:"", distance:""} for each and every image (and thus is a little slower).