TEAM_02_CSCI599_HW_BIGDATA_Report

Input File: ufo_awesome.json

Output file: ufo_awesome_FINAL_OUTPUT.tsv

Data Preparation and Cleaning

We started with analyzing the UFO sightings data. The most important column identified was the location as it should be used to get the latitude and longitude using geocode API. We observed following nature of the location field in the data which were later exploited to decode the ISO region for the location.

1) Total number of sightings are 61067. Number of non-US sightings based on a cursory analysis is 9019. Out of these there are around 21k unique locations.
2) Most of the US sightings followed the notation of (county_name, state_code) For example: - Santa Cruz, CA.
3) Some US sightings are mostly random and has Freeway/Expressway names
4) There is no similar notation used in non-US sightings except for the countries Canada, Australia and some others. These countries have the country name in the location string.

Some other observations made about the datasets:

1) Some of the data is not clean. Ex: city is missing from the UFO sighting location
2) Row 19202 in TSV file is not of proper structure. (Ex: the description is there in the city column)
3) Inconsistent data in the JSON and TSV files. (Ex: Row 19202).

Finding Nearest Airport

We used the following airport dataset**: -** http://ourairports.com/data/airports.csv

This file contains around 53k airports with the longitude, latitudes, iso_country and iso_region. First thing we had to do was get the latitudes and longitudes for the UFO sightings dataset and then compare it to the airports dataset to get the closest airport. The first instinct was to compare all the locations with all the airports. But soon we could see that it will be too many comparisons. Then we found that as most of the sightings are in US and there are around 22k airports corresponding to US region, we need to have a better strategy for reducing number of comparisons. So, we came up with the following strategy.

1) For all the US sightings, we will assume that it could be related to an airport in that state or the neighboring state. For this we created a key-value pair data structure which had every state's neighboring state. Using the state codes of the neighboring states we formed ISO region codes such as US-CA (for USA California). We used these codes to narrow down on the list of airports to be compared to the UFO sighting location.
2) For non-US sightings, we assume that the sightings will be related to the airports in that country alone. While getting the latitude and longitudes from geocode API, we also fire the reverse query to capture the ISO country code of that location. This helps reducing the number of comparisons in non-US sightings.

Issues faced while getting longitude and latitudes from geocoding API

1) The OpenMaps geocode API would throw Too many requests error if overwhelmed with too many geocode requests. For this we had to store all the unique locations in pyMongo DB and then call the geocode API with a sleep of 1 second.
2) Some cities in US share names with cities in Canada and Mexico for such cases to get best results we had to include the name of country in the location query string.

3) Same city names across different states in USA. For these we had to include the state name in the location query string.

Data Sets for additional 9 features:

Dataset 1 – Meteorite Landings: Mime Type – Application/JSON

Source - https://data.nasa.gov/resource/y77d-th95.json?$limit=50000

Meteorite Landing - based on year and location we can get to know if people confused a meteorite for a UFO

Features:

I. Name of closest meteorite
II. Distance of closest meteorite to each city for that year the UFO was sighted
III. Possibility that the sighting is mistaken (sighting happened at < 50 miles of Meteorite landing)

Features Extraction & Methodology.

The dataset is in the JSON format, which we had downloaded using SODA API (by using parameter limit=50,000). The dataset is stored in file **meteorites.json**. Then we wrote a python script **meteor.py** to extract the said three features as follows. The same script merges the data with the UFO sighting dataset and the airport features. Features extracted are closest Meteor name, Meteor Distance based on the longitude and latitude of meteor landing, and the possibility that the meteor could have been confused as a UFO sighting, which is based upon the threshold on the distance and the year in which the meteor landing happened. To narrow down the number of comparisons, we indexed all the meteor landings based on the year in which it happened and then compared it to UFO sightings of that year.

Insights from the dataset and its extracted features:

1) About 10% of the total UFO sightings were found to be less than 100 miles of the nearest meteor landing in the same year.
2) The low percentage points to very less correlation in these two events.

Dataset 2 – Census Data: MIME Type – text/CSV

Input data Sources: We took Census 2000 and 2010 data from https://factfinder.census.gov/, Open data source at https://github.com/grammakov/USA-cities-and-states?files=1

We have added around 8 missing entries to this data file. Including few here.

City	County	State
Oregon	Clackamas	Oregon
Murphy	Collin	Texas
Bloomington	Hennepin	Hennepin

Features Extraction

Features extracted are Housing density, Population density, County. We imported the data using python 3 CSV library to read the respective columns from our input file. We get the county for a given city, state from the Input_CountyCitiesList.CSV. We grouped the UFO sightings data on Key of <State, County, Year>. We joined UFO dataset with this census data on the key <State, County, Year>

What we noticed about datasets and handling of issues if any:

- Problem with Alaska and District of Columbia states to get county-city info, as the structure of cities and counties does not align with the other states of America
- Puerto Rico data format, added united states suffix in the input data file to make format consistent with other states
- In the input file Input_CountyCitiesList.CSV - removed the data related to District of Columbia as the County names were empty strings and some junk data related to Postal Service

- From the total UFO sightings, there are 51547 sightings for valid 50 US states and out of which for 5883 sightings we could not map to the county due to the above-mentioned data issues.
- There are some incorrect locations of UFO sightings due to which we could not map given city, state to a county, state
  Examples like:
  - Invalid locations like "Laporte, WA"
  - Ambiguous locations like “Silver Beach, NY”
  - Spelling mistakes: Seatle [Seattle], Lewiston [Lewistown]
  - Missing detail: Hollywood - [WestHollywood], Bluff - [Pine Bluff], Tawas - [East Tawas]

Insights from the joined dataset:

- For a given Census year range, most of the sightings happen in rural areas. For the year range 1991 – 2010, rural sightings percentage is at least 74.9% ~75% (ranging up to 91.9%)
- Highly sighted UFO locations are sparsely populated
- Joined dataset suggests that rural population can mistook UFO sightings due to lack of awareness

Assumptions

Based on the source at https://www2.census.gov/geo/pdfs/reference/GARM/Ch12GARM.pdf, we define County as urban if population density (per sq. mile) is > 1000 and housing density > 500 rural otherwise.

Dataset 3 – Sci-Fi Movies: Mime Type – text/HTML

Sci-Fi Movies (e.g. Star Wars, Star Trek) released - based on the year of release of sci-fi movies and the year of the UFO sightings, we can predict if whether the UFO sighting was a delusion or not.

Features:

I. The number of sci-fi movies released in that year
II. The number of UFO sightings that took place in that year
III. ratio of number of movies released to number of sightings that took place in that year
IV. based on the ratio, if it possible that due to a high influence of sci-fi movies, people imagined aircrafts to be UFO

Source of the dataset acquired: https://en.wikipedia.org/wiki/Lists_of_science_fiction_films

Feature Extraction and Methodology:

The dataset was available in HTML format in the form of multiple tables for each year and decade. A python script - *wiki_scifi.py* was written to extract the table contents from the links for each decade. All the extracted table contents from each decade was finally inserted into a csv file - *sci-fi_database.csv*.
From the generated csv file and the original ufo_awesome.json data file, the years were extracted based on the movie release date and the UFO sighted date and their respective counts were calculated. Once we had this data, we made an ***assumption that if the ratio of the number of UFO sightings to the number of movies in that year < 2***, then there is a high probability that the sci-fi movie had an impact on the person who sighted the UFO and probably mistook what they saw, for a UFO.

Insights from the dataset and its extracted features:

- There were 18 out of 82 years in which the ratio was less than 2.
- 21.95% of the years where a UFO was reported could be a possibility of delusion where the person confused an aircraft for a UFO after watching a sci-fi movie released in the same year.
- The unintended consequence seems to be the fact that such a huge sci-fi movie watching crowd could have been the reasons for confusing a flying object to be a UFO and falsely reporting its sighting.

Difference between clusters in Jaccard, Edit Distance and Cosine Distance Similarities?

For our datasets, the clusters created for Jaccard, Edit Distance and Cosine Distance similarities are not very different. Edit Distance had clustered the data with a range of similarity scores with respect to their x-coordinates. Cosine Distance generated clusters similar to Edit Distance, but some of the features in few of the clusters lie outside the range defined for that cluster. For Jaccard Similarity, the clusters formed are same as Cosine Distance, but it is grouping the data based on equal similarity scores.

How do the resultant clusters highlight the features you extracted?

Our datasets comprise of 3 main features – Were meteorites landing assumed be a sighted UFO? Did people take a flying object as UFO after watching Sci-Fi movies? Was the region Rural? Our resultant clusters indicated that most of the sightings happened in the Rural areas, where no meteorites landed and people did not watch Sci-Fi movies. Few of the clusters denote that few rural areas might have assumed a meteorite landing with a UFO.

Advantage of using TIKA for computing similarity metric score:

Ease of use – It was very convenient to run the TIKA Similarity using TIKA, and compare the metadata as well as content by just providing in the file names.

Disadvantage of using TIKA for computing similarity metric score:

Time consuming for large files – since our dataset contained approx. 61000 rows, TIKA takes a lot of time to read the content, hence we had to read file manually and then perform TIKA operations on it.

Do UFO Sightings only occur in rural areas?

For a given Census year range, most of the sightings happened in rural areas. For the year range 1991 – 2010, rural sightings percentage is at least 74.9% ~75% (ranging up to 91.9%)

Are UFO sightings mostly (greater than 75%) occurring in areas within 25 miles of airport?

As per our dataset, we could deduce that for nearly 91% of the UFO sightings, the nearest airport is less than 25 miles away.

What do population demographics tell us about the areas in which UFOs occur?

Highly sighted UFO locations are sparsely populated.

What similarity metrics produced more accurate measurements? Why?

Edit-Distance Similarity produced the most accurate result. As shown by Dendrogram in **Flare_Dendrogram(EditDist_Similarity).pdf** (under Tika_Similarity folder), it takes in a range of similarity metric values, and thus is able to cluster the features of y-coordinates which have very close similarity metric with the x-coordinate. Jaccard Similarity is clustering the data based on equal similarity metric value, so some of the clusters formed are not taking into account a range of similarity metric values based on the x-coordinate. The cosine similarity measure is forming clusters similar to edit-distance, but many clusters being formed as in edit-distance similarity.

What clusters were revealed?

As shown by Dendrogram in **Flare_Dendrogram(EditDist_Similarity).pdf** (under Tika_Similarity), the biggest cluster formed is where population/housing density denotes that UFO sightings mostly happened in the rural areas, where people do not watch sci-fi movies, and there were not many meteorite sightings in the city or county. The major cluster suggests that rural population might have assumed UFO sightings due to lack of awareness. For the next biggest cluster, we were missing out census data before 1991, so we assumed rural area metric to be **TRUE** (since there might have been a greater number of rural areas before 1991), thus it depicts that UFO sightings again happened in rural areas. The next biggest cluster is where the meteorites were sighted in the rural areas, so people might have mistaken a falling meteor with a UFO.