ASSIGNMENT 1: REPORT *(TEAM 06)*

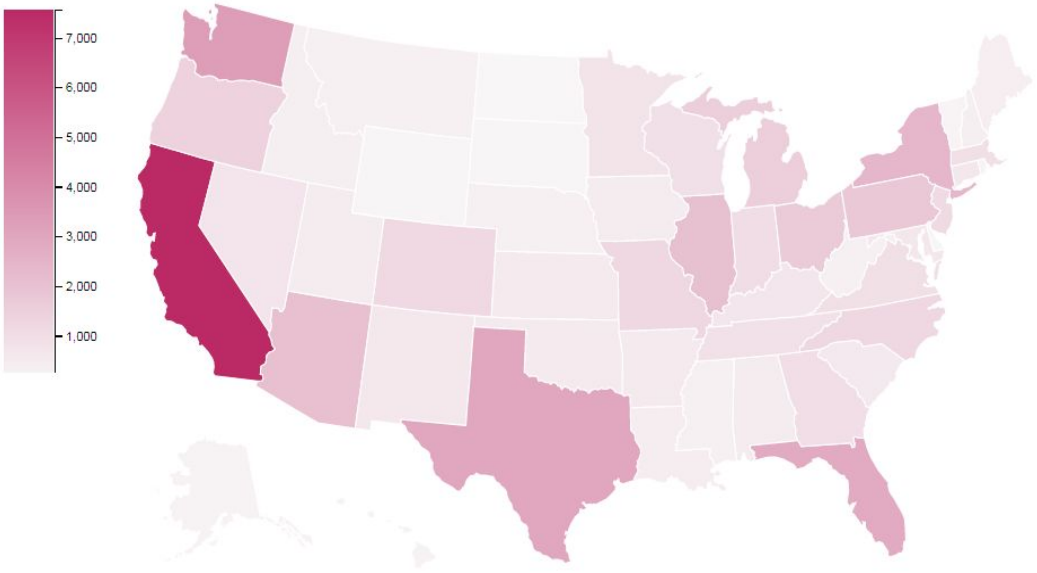


Fig 1: Choropleth of UFO sightings by state

Data Munging and Feature Extraction

We've chosen the following datasets and features to be appended to the UFO Awesome data.

1. **Census** (*text/csv*)
   Describes US population statistics from 1990-2010 measured every 10 years, aggregated into three distinct age categories
   Features extracted: Census by age-group: *children, adults, senior citizens*

2. **Geospatial Data** (*image/tiff, application/json*)
   Used to geocode areas that come up in the dataset.
   Features extracted: *elevation* (miles), *latitude, longitude*

3. **Climate** (*application/xml*)
   Contains climate data from the years of 1906-2007 per state per month
   Features extracted: *max temp, min temp, average temp, Palmer Drought Severity Index (PDSI), Precipitation Index*

4. **Airport Codes** (*text/csv*)
   This is loaded into QGIS to extract the airport that is closest to the area of sighting (uses the KNN algorithm)
   Features extracted: name, type (size), coordinates, distance

The datasets in terms of the 5V*'s:

Dataset	Volume	Veracity	Variety	Value
Census	High	Medium	Low	Med
Geospatial	Med	High	High	High
Weather	High	High	Low	High
Airport	High	High	Med	Med

Table 1: 5 (minus one) Vs

* Velocity is not considered here since we're dealing with static datasets only

The datasets were cleaned/transformed/aggregated/joined using python and pandas. Repo: https://github.com/Coldsp33d/UFO-Awesome

Observations on the Datasets

1. **Pros**
   - The data includes sightings from around the world (including India!), and the first observation is from the 18th century
   - UFO Awesome has 60K records and 6 features. This is a lot to work with, and there are many options for joining and/or performing feature engineering
2. **Cons**
   - The UFO data is difficult to load. The easiest file format to work with is the JSON and that has corrupt records which we had to write a script to fix using regular expressions
   - The data is not consistently stored (prime examples are the *description, shape,* and *duration* columns). Some degree of effort is needed to process these to formats that can be used for further analysis.
   - (Left) Joining the datasets is not straightforward. The join-columns are often incompatible, and further preprocessing is required, such as normalizing dates, lowercasing strings, and so on.

One useful takeaway from this exercise is that deciding on a consistent format for storing data (especially with datetime and string columns) helps better manage the data, and reduce the NULLs generated as a result of the join.

Data Analysis and Inferences w/ Tika Similarity

It is possible to learn a lot through the power of Tika, pandas and D3. For example:

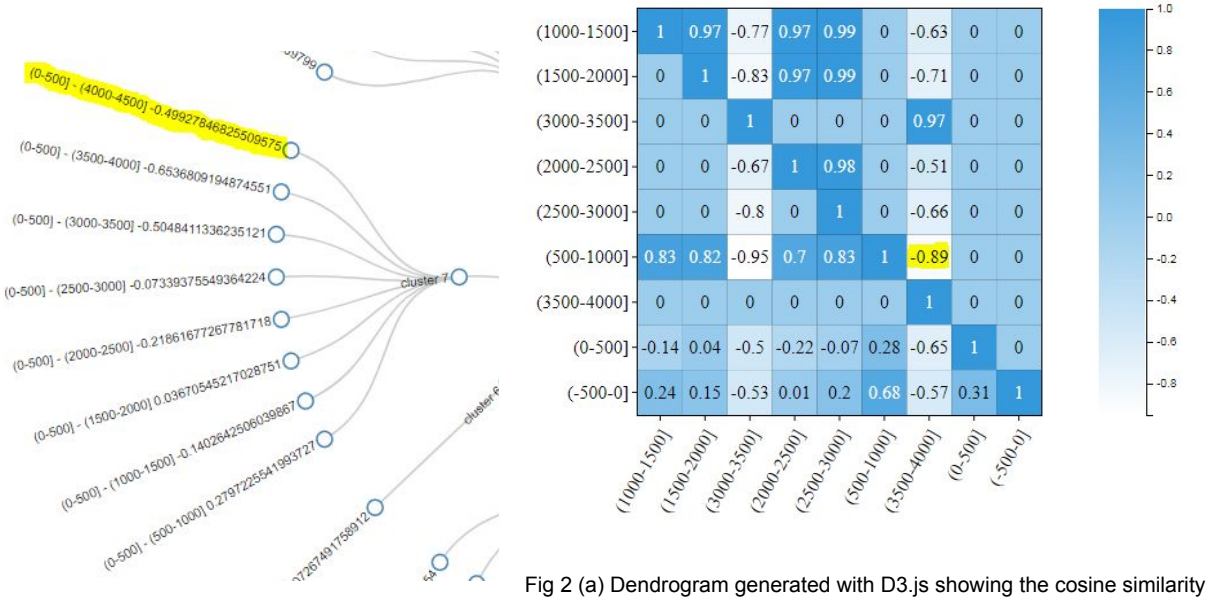


Fig 2 (a) Dendrogram generated with D3.js showing the cosine similarity b/w elevations at different levels for a single cluster, (b) D3.js similarity matrix for the same values

The figures above are the result of comparing the *climate*, *elevation*, and the *number of sightings*, using cosine similarity from Tika-Python. In Fig 2 (a), we see that there is a (dis)similarity between sightings at elevations from 0-500m versus those at 4000-4500m, based on the climate at these elevations. Possible inference: Climate changes with altitude. And so, weather/climate changes have a direct effect on UFO sightings. *Note:* For all similarity measures using cosine similarity, the data is min-max normalized.

x-coordinate	y-coordinate	Similarity_score
changing	cross	0.6
changing	cigar	0.142857
cigar	cross	0
cross	cylinder	0
cigar	disk	0
chevron	cross	0.6
changing	chevron	0.6
circle	cylinder	0
cylinder	diamond	0.142857
changing	diamond	0
cone	cross	1

Fig 3: Jaccard Similarity with Shape/Census

More inferences can be drawn by analysing seemingly unrelated columns. One example would be to contrast the *shape* of sighted UFOs against the *population* of the area at the time of sighting. One can use Jaccard similarity after binning the data into quartiles (deciles can also be used).

The result is shown in Figure 3. Possible inference: shapes with higher similarity tend to be witnessed in areas with comparable population density.

Here is a more direct analysis: Analysing the *duration* of sightings with respect to *distances from airbases.*

	airport_distance	1-5 min	1hr	<1min	>1hr
0	<5mi	5453	5393	7427	40
1	5-10mi	4238	4338	5989	28
2	10-15mi	1236	1351	1818	12
3	15-20mi	324	353	422	4
4	20-25mi	61	85	82	0
5	>25mi	23	27	30	1

The dataframe in Fig 4 has been generated using pandas. From the table, a possible inference here is that sightings of shorter duration are more likely to occur when there are air bases close by. This could mean that, in many cases, airplanes flying overhead were mistaken for UFOs (leading to false reports).

Fig 4: Correlating distance from airports with duration of sightings

In an effort to explore all aspects of the data, we have also utilised the data in the *description* column to perform similarity analysis. The procedure for cosine similarity is straightforward: First, preprocess the data (this involves lowercasing, lemmatization, removing stopwords, and removing misspellings). Next, train a TFIDF vectorizer on unigrams from these documents grouped by state. This vectorizes the descriptions and they can then directly be passed into Tika-Python for cosine similarity analysis. The result is shown in Fig 5(a). Fig 5(b) is also interesting, as it is generated through an edit distance similarity measure.

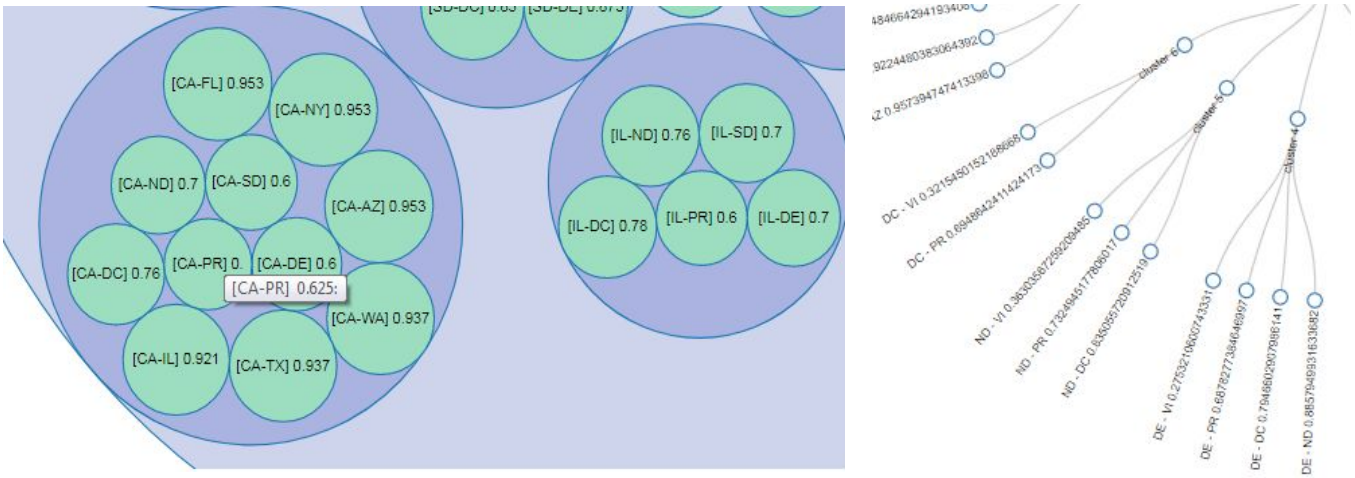


Fig 5: D3.js edit distance similarities on descriptions featurized by TF-IDF (a) Circle-packing (b) Dendrogram

It does not make much sense to use edit distance directly on the data, so it is first featured into bit strings of length 64. The input is shown in Fig 6, and the corresponding output in Fig 7.

state	description
AK	a public employee calls nuforc to report a ver...
AL	man wife children driving down long hill see...
AR	memphis faa nd rept two men witness huge blind...
AZ	father son witness strange obj streak across ...

Each bit represents whether a certain keyword is present in the description or not. The importance of the keywords are determined using the same TFIDF vectorizer as before. Edit distance is then computed on these bit strings.

Fig 6: Input description grouped by state

```
CA,11101111111111111111111011111111111111111111111111111111111111111
WA,11101111111101111001110011111111111111111111111111111111111111111
TX,11101111110101111111111111111111111111111111110111111111111111111
FL,11101111111101111111111111111011111111111111111111111111111111111
NY,11101111111101111101110111111111111111111111101111111111111111111
AZ,11101111111111111111111111111111101111111111101111111111111111111
```

Because these are length 64 bit strings, it is possible to perform a broadcasted XOR operation using NumPy and just sum the bits, and could be used as a performant alternative to the standard edit distance algorithm.

Fig 7: Featurised bit strings

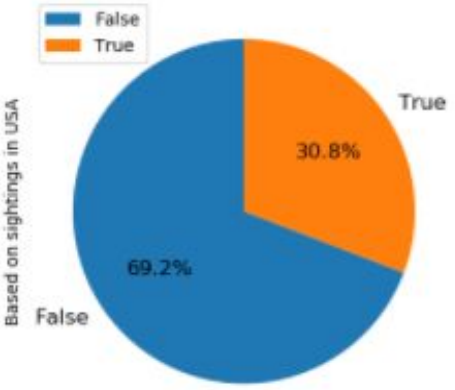
For ease of visualisation, the top 5 states by number of sightings have been compared with the bottom 5 states. California (CA) is the top state, and Puerto Rico (PR) is at the bottom. Of all the pairs in the CA-cluster, CA-PR has the lowest similarity, hinting that *description* column reflects the UFO activity of a state quite well.

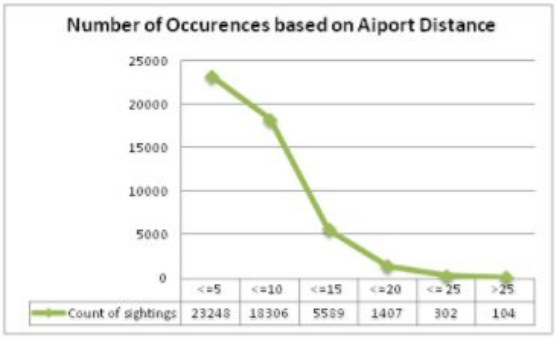
Comparing Similarity Metrics

For our dataset, the only attribute we could use edit-distance on to obtain reasonable results was the description. We tweaked the same as described above. For cosine similarity, we vectorized our data and fed that as an input to the algorithm. We partitioned the feature spaces into groups and computed jaccard similarity on these groups. Comparing the cosine and Jaccard similarities, the former is preferred because it does not require the input to be partitioned.

More Observations

1. 70% of sightings come from rural areas
   (Fig 8 on the right)

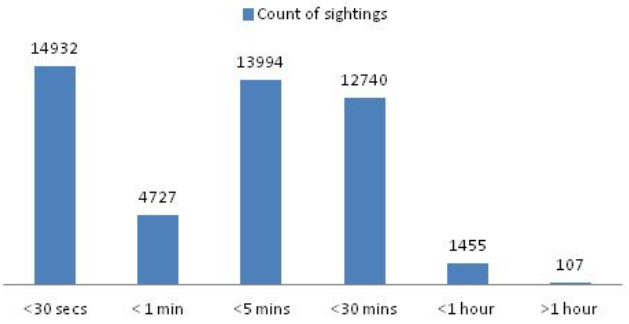




2. Almost every sighting is within 25 miles of the nearest airport along the coasts
   (Fig 9 on the left)

3. From the choropleth in Fig 1, most of the sightings were along the coast, and in relatively larger, more densely populated states.

4. Count of UFO sighting by duration (Fig 10 on the right)



5. Indirect Features
   - Age Groups
   - *Is_urban* - whether an area is part of an urban area or not
   - *zipcode* - not present in the final joined data, but is used to join the census data with the UFO data

Thoughts on Tika-Python and Tika-Similarity

1. **Pros**
   - Reasonably fast execution
   - Out-of-box integration with D3.js, minimal tweaking required
2. **Con**
   - A lot of the python code (not the code committed by Prof. Mattmann) is poorly written, difficult to read, and does not conform to PEP-8. Blatant misuse of lambdas, inefficient routines involving reduce, etc. Some python script file names even contain hyphens.
   - Repo written in python-2.7 (see https://pythonclock.org/)

Link to Tika-Similarity PR

https://github.com/chrismattmann/tika-similarity/pull/90

Members of Team Six

Akashdeep Singh, Koustav Mukherjee, Sayali Ghaisas, Shiva Deviah, Vritti Rohira