Windward:

DATA ANALYSIS TASK

Idan Fonea

The task is to identify a specific type of dock according to AIS data

The analyze Algorithm steps:

1. Getting the data:

Read the CSV data to a Pandas Dataframe

Grouping by MMSI, instantiate ship Models

2. cleaning

Each ship record that is inconsistent (had encountered changes) in the size, Class, and distances, are discarded

Ships with a mismatch in the size and distances are discarded

3. analyzing ships:

Each ship, when **speed is 0** (stopped) for **at least 4 (configurable) hours**, was counted as docked for that period. The docking locations were saved

4. analyzing docking locations:

Each saved docking location, that its distance is larger than the configurable resolution, instantiates a dock Model

The dock model has:

Latitude, longitude and heading

And from the docked ships we can learn: the min/max size/width of the ships in can consist and the classes of ships docked

In addition I used Google Static Map API to get a satellite image and the

Elevation API (which was according to the documentation to give the depth, but only gives sea level according to my observations)

About the data

General info about the given data

[56057 rows x 12 columns]

Columns:

Time, MMSI, Latitude, Longitude, Speed, Heading, Class, Size, DistanceToBow, DistanceToStern, DistanceToPort

Date format: "yyyy-MM-dd hh:mm:ss"

```
len (set(df['MMSI']))
```

There are 327 different MMSI

```
max(df['Latitude'])
```

52.010829999999999

```
min(df['Latitude'])
```

51.844949999999997

```
max(df['Longitude'])
```

4.5812230000000005

```
min(df['Longitude'])
```

3.9503330000000001

Stats:

30 ships has metadata consistency problem

109 ships are not well measured

215 ships are bigger the 200

Extra: Histogram tool

The docks can be briefly identified using a histogram plot of all data's longitude and latitude, assuming that a position should be occurring a lot more where it represents a dock (a ship transmitted a lot from the same position: the location can be obtained by comparing the occurrences of the specific coordinate. Figure [1] and Figure [2]

A docking state can be flagged also where the heading of ship equals to one of the occurring heading in the heading histogram Figure [3]

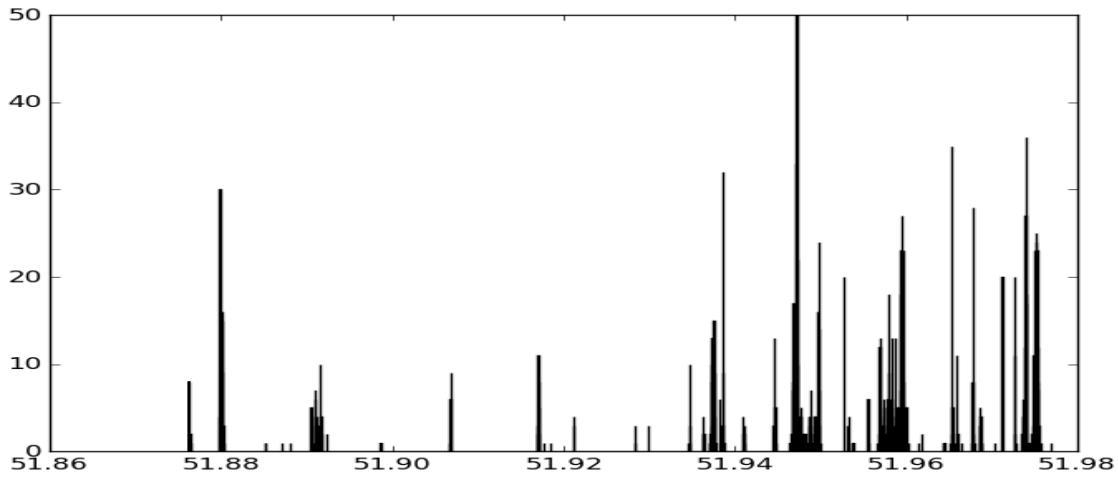


Figure 1histogram of ships longitudes

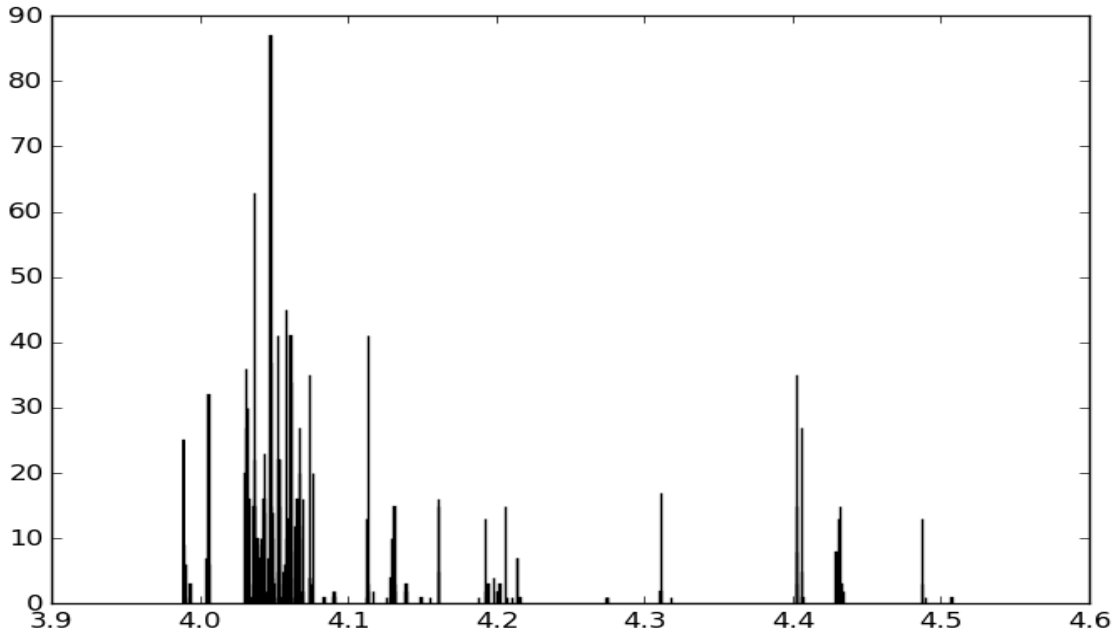


Figure 2 histogram of ships latitudes

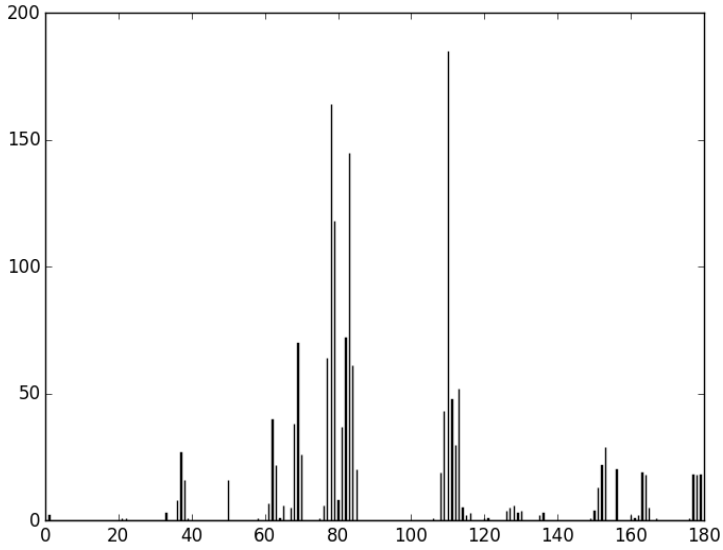


Figure 3 Heading angle of all ships

System design and assumptions

Data Integrity tests:

1. Consistency of data that is entered manually: 'MMSI','Class', 'Size', 'DistanceToBow', 'DistanceToStern', 'DistanceToPort', 'DistanceToStarboard'
2. size == distance to bow + distance to stern
   width < size

Configuration:

All system configuration and parameters controlled via the configuration module

Assumptions

Docking locations assumptions:

1. Docking meaning speed =0
2. minimal docking time: 4 hours (configurable)
3. heading = constant on docking (not used)
4. Elevation (depth) google api – only sea level acquired, can disqualify terrain errors
5. What google api: locations (optional)
6. satellite image with orientation [2]

Relevant ship assumptions:

1. minimal size of ship = 250m (configurable)
2. not marked as classes: “High speed craft”, “Military or law”, “Fishing”, “Passenger”, “Pleasure”
3. are consistent in entered parameters (mmsi, class, size and distances) meaning the AIS was set professionally and is reliable

A talk about distances and Angles

x amount of size (in meters) is equivalent to :

ra = 6378134 m , rb = 6356752.31 m (Wikipedia[1])

a. earth is sphere:
b. earth is ellipsoid- [out of scope]

we will talk only on the sphere case

r = (ra + rb) /2 = 6367443.155 m

$\Delta x = (\Delta long/360) * pi * r$

$\Delta y = (\Delta lat/360) * pi * r$

a * Angle= Distance

(Or in short, multiply by a = 55566.424 do get the distance between two angles)

Now let's calculate the Google maps zoom size:

We want to see about 5 cm : 200 m ratio (sized ship), meaning : 200/0.05 =times: 4000

Where zoom = 1, the closest zoom we have is 2^ZOOM = 12: 2^12 = 4096.

Meaning a fraction of times 4096 of each axis comparing to full map: zoom=1 ->[-180:180],[-90:90]

Given L is the latitude axis length, L / 4096 = Δy = a * Δlat

$\Delta limit\_latitude, \Delta limit\_logitude$ = Lx, Ly / (a*4096)

a = 55566.424, Ly = 40007860 m, Lx =40075016 m

let's determine minimal angular resolution: let say 100 m [0.1?], (minimal angle of where we wish to classify an object in space. Well take it slightly less than the ships size)

Meaning :

$\Delta lat\_reolution$ = 100/ (555.66424 * 2^ZOOM) = 10/ 2276000.72704 =4.39 *10^-5[degrees]

Meaning we need only 5 numbers after a decimal point accuracy

REMARK- it turned out I could not predict the Google Api Zoom scale, but this exercise did a great deal to understand and asses the resolution

Results:

On the setting of resolution = 0.001647 degrees: the system detected 58 docks.

Here are the first 4 out of total 58 desired docks found (can be maybe calibrated to more, with the cost of duplicates by setting the resolution)

more images can be found at:

https://picasaweb.google.com/116781616540240633183/PredictedDocks

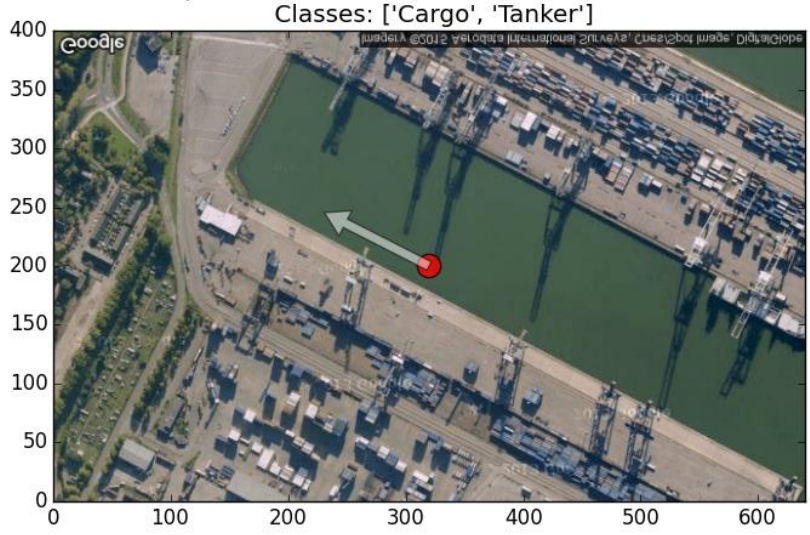
The code:

https://github.com/forye/docks_images

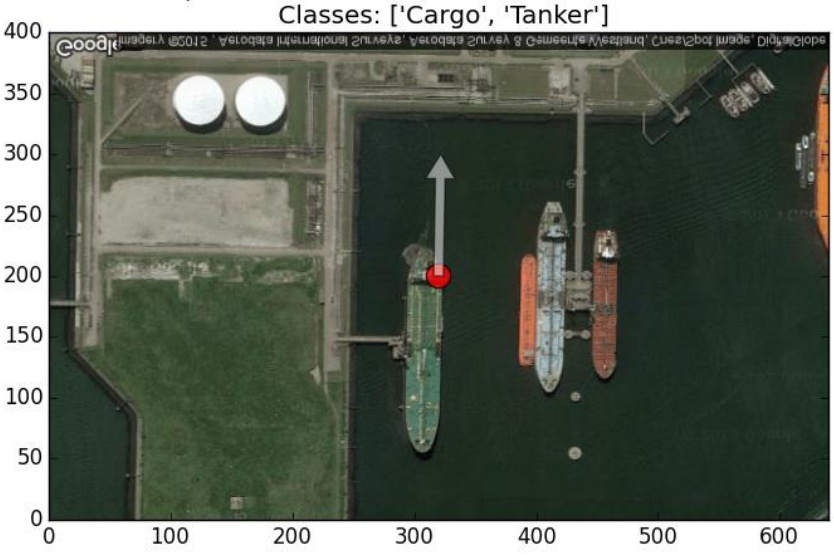
After installing all the correct module, run get_docks_candidates.py, to replicate the results,

Play with the configuration at config\configuration.py

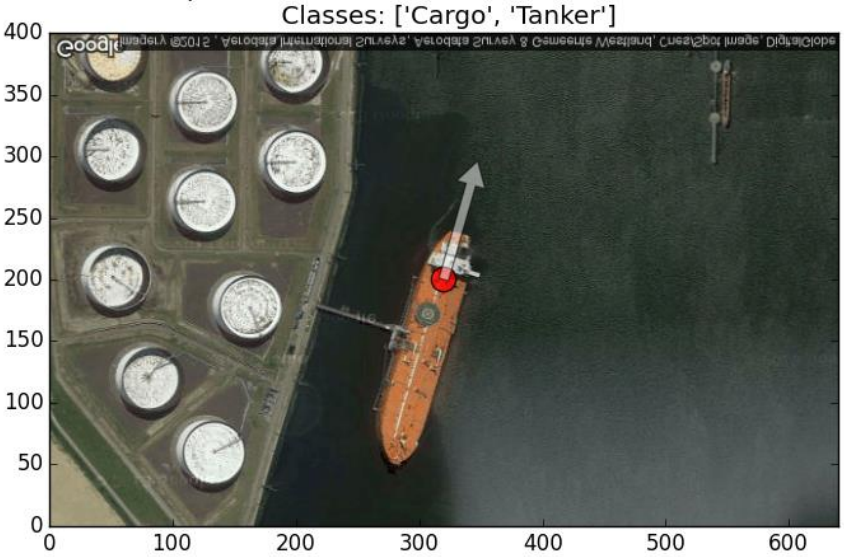
Longitude:4.43041 Latitude:51.89093 Heading:63 Elevation:0.644188463688
Docked Ships Count: 49 size: 254 to 300m width: 30.0 to 48.0
Classes: ['Cargo', 'Tanker']



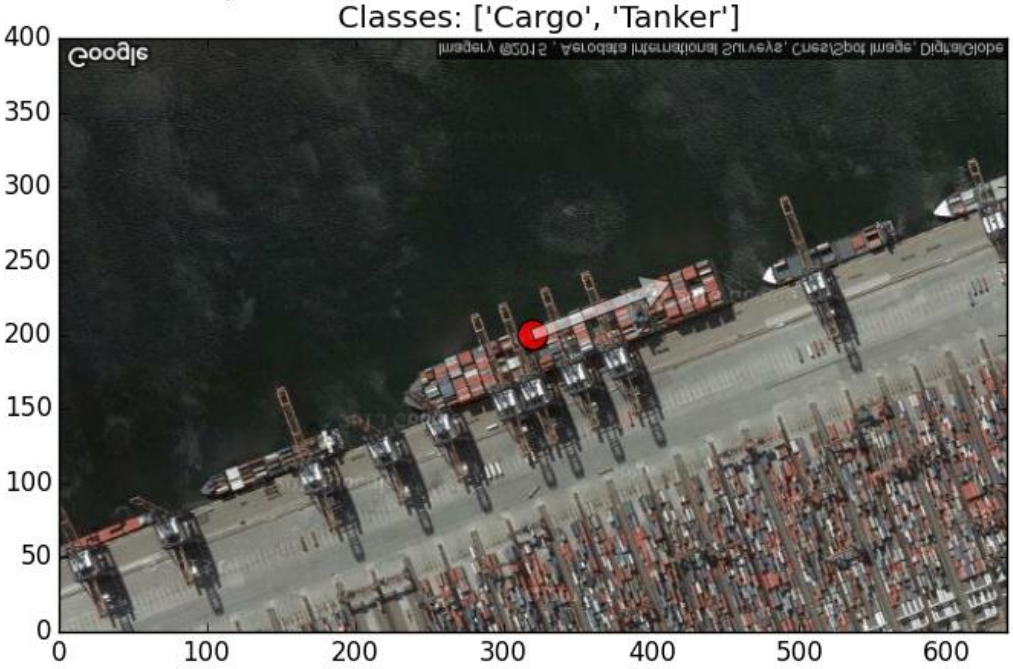
Longitude:4.138867 Latitude:51.95333 Heading:359 Elevation:0
Docked Ships Count: 85 size: 254 to 396m width: 30.0 to 54.0
Classes: ['Cargo', 'Tanker']



Longitude:4.160717 Latitude:51.94468 Heading:344 Elevation:0
Docked Ships Count: 43 size: 254 to 396m width: 30.0 to 54.0
Classes: ['Cargo', 'Tanker']



Longitude:4.036292 Latitude:51.97386 Heading:291 Elevation:0
Docked Ships Count: 56 size: 254 to 400m width: 30.0 to 60.0
Classes: ['Cargo', 'Tanker']



references

[1] Wikipedia

https://en.wikipedia.org/wiki/Earth

[2] Google Static Maps API

https://developers.google.com/maps/documentation/static-maps/

[3] Google Elevation API

https://developers.google.com/maps/documentation/elevation/intro

[4] GitHub public repository: the code, resources and results

https://github.com/forye/docks_images.git