# AIS VESSELS TRAFFIC DATA

Final Report – Scalable Data Analysis

DECEMBER 16, 2017

RAVI SHANKAR
01647513

**Introduction:** I am working on the AIS Vessels Traffic Data, there is a large amount of trajectory data available, this data is taken from webpage of Marine Cadastre – Vessels traffic data which is given below.

https://marinecadastre.gov/ais/

**Data Set Description:** The data is divided into the hierarchy of Year, and sub hierarchy of Zones and by Months, even Yearly data is also available to download, Data contains the Broadcast points, Vessel and Voyages Information.

In the starting I worked on the single vessel data and then I expanded my data to work on the three months of data, which contains at most 54 million of records.

The Broadcast points contains the MMSI which represents each vessel and one vessel contains multiple voyages, it contains the Latitude and longitude for each point, the data is recorded on every minute. It also contains the date and time of the movement of the ship.

Fig-1 is the screenshot of the dataset.

| | BaseDateTime | COG | Heading | MMSI | ROT | ReceiverID | ReceiverType | SOG | Status | VoyageID | lat | lon |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2011-01-01T00:00:00 | 254 | 511 | 367609189 | 128 | 01NFIS1 | r | 0 | 0 | 1 | 40.641045 | -74.164090 |
| 1 | 2011-01-01T00:00:00 | 338 | 146 | 367993089 | 127 | 01NFIS1 | r | 2 | 15 | 2 | 41.167430 | -73.174177 |
| 2 | 2011-01-01T00:00:00 | 329 | 114 | 247207450 | 0 | 01NFIS1 | r | 0 | 5 | 3 | 40.670333 | -74.083333 |
| 3 | 2011-01-01T00:00:00 | 157 | 511 | 367030180 | 128 | 01NFIS1 | r | 24 | 15 | 4 | 40.563197 | -74.019465 |
| 4 | 2011-01-01T00:00:00 | 192 | 210 | 366912510 | 0 | 01NFIS1 | r | 0 | 0 | 5 | 40.669623 | -74.037770 |
| 5 | 2011-01-01T00:00:00 | 336 | 339 | 367407028 | 0 | 003669959 | r | 16 | 15 | 6 | 41.037577 | -73.127367 |
| 6 | 2011-01-01T00:00:00 | 14 | 14 | 367718405 | 0 | 05NNNE1 | r | 10 | 0 | 7 | 39.726543 | -75.503247 |
| 7 | 2011-01-01T00:00:00 | 0 | 267 | 367680500 | 0 | 003669730 | b | 0 | 5 | 8 | 36.945733 | -76.332193 |
| 8 | 2011-01-01T00:00:00 | 218 | 216 | 367406050 | 0 | 2003669982 | b | 7 | 0 | 9 | 40.785910 | -73.919427 |
| 9 | 2011-01-01T00:00:00 | 265 | 511 | 367333406 | 128 | 003669983 | b | 9 | 10 | 10 | 40.641042 | -74.155382 |
| 10 | 2011-01-01T00:00:00 | 0 | 511 | 368608000 | 128 | 003669730 | b | 0 | 5 | 11 | 37.166360 | -76.610020 |
| 11 | 2011-01-01T00:00:00 | 268 | 275 | 369074439 | 0 | 003669984 | b | 0 | 0 | 12 | 40.730680 | -74.013873 |
| 12 | 2011-01-01T00:00:00 | 0 | 511 | 366649058 | 128 | 05SOAK1 | r | 0 | 0 | 13 | 34.198033 | -77.955633 |
| 13 | 2011-01-01T00:00:00 | 214 | 228 | 866860249 | 127 | 01NFIS1 | r | 2 | 0 | 14 | 40.659090 | -74.045832 |
| 14 | 2011-01-01T00:00:00 | 192 | 191 | 371257000 | 0 | 05RTUC1 | r | 21 | 0 | 15 | 38.815657 | -74.055347 |

**Goals:** There is so much work is already done before on this AIS Vessels traffic data. My goal for this project is to figure out the stops of the trajectory data and to segment the trajectory data into multiple sub trajectories.

I used the Python 3 libraries, PySpark and Pandas and tool I used is Jupyter Notebook.

**Calculation and Sub trajectories:** Initially I worked on single vessels data, where I had some calculation to find out the stops and segmented the trajectory. Later on I worked on the three months data, where I followed the following steps to find out the trajectory.

1- I sorted out the whole data based on MMSI and the BaseDateTime.
2- Then I calculated Distance throughout the consecutive rows by taking latitude and longitude.
3- I calculated the time difference through the given BaseDateTime, and converted into the total hours.
4- And through the Distance and total hours, I calculated the speed into Km/h.
5- Then calculated the stops through the given speed, so if the speed goes below threshold there I can split the data into multiple datasets by assigning the multiple id's, which is named as subVoyagaId

**Results:**

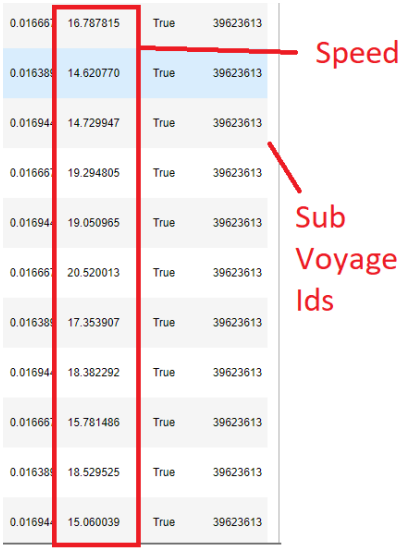



Fig-2 As the result I have shown the calculated speed column and making the ids to call them the subVoyagaId. This shows the speed is greater than 5km/h and separating it through the id's.

**Scalability:**

A part for scalability, I used the python libraries i.e., Pandas and PySpark, to see the performance that which one is faster in terms of processing.

In the comparison PySpark is much faster than the pandas. As per my calculated results Pandas took 4.38 minutes but the PySpark took 2.83 minutes to calculate and write results onto multiple CSV files and each CSV file were around 25MB, as compared to Pandas which was taking long to write the output on the CSV which was around 5 to 10 minutes to write it.

```
54582826   00:03:00    0.050000  0.238712   False     40207249
54582827   00:03:00    0.050000  0.133515   False     40207250
54582828   00:03:00    0.050000  0.143250   False     40207251
54582829   00:03:01    0.050278  0.180693   False     40207252
54582830   00:06:00    0.100000  0.028850   False     40207253
54582831   00:06:00    0.100000  0.019234   False     40207254
54582832   00:03:00    0.050000  0.152940   False     40207255
54582833   00:03:00    0.050000  0.031868   False     40207256
54582834   00:03:00    0.050000  0.033806   False     40207257
54582835   00:08:59    0.149722  0.015078   False     40207258
54582836   00:03:00    0.050000  0.044757   False     40207259
54582837   00:06:00    0.100000  0.022649   False     40207260
54582838   00:06:01    0.100278  0.006210   False     40207261
54582839   00:03:00    0.050000  0.329572   False     40207262
54582840   00:06:00    0.100000  0.141377   False     40207263
54582841   00:02:00    0.033333  0.184370   False     40207264

[54582842 rows x 24 columns]
262.3845628388226
```

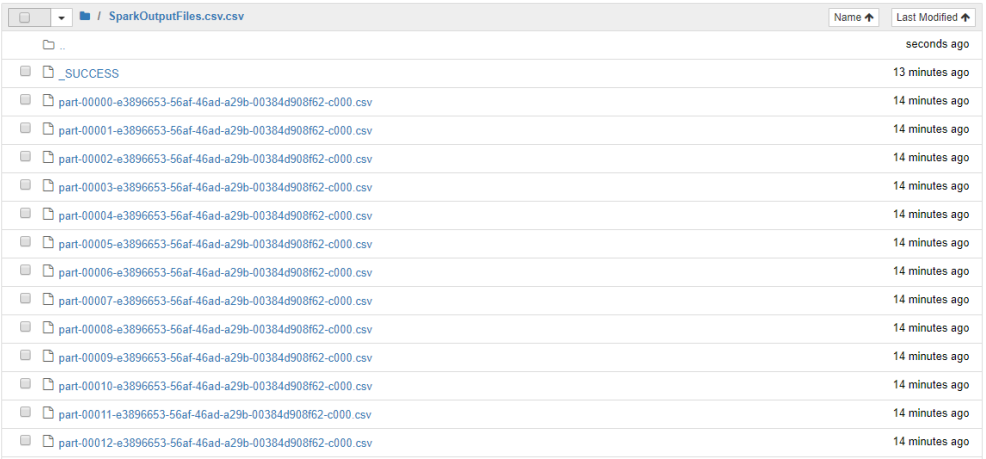
Fig-3 As talked about Pandas is taking around 262.384563 seconds which are around 4.38 minutes.

Fig-4 And the image below is showing that PySpark took 172.994621 seconds, which becomes around 2.38 minutes.

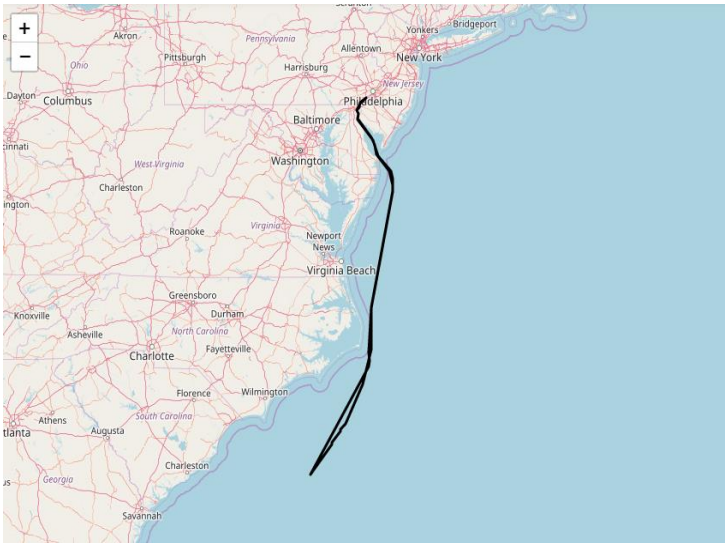
```
result_frame.write.format("com.databricks.spark.csv")

print('total time taken:', time.time() - starttime)
```

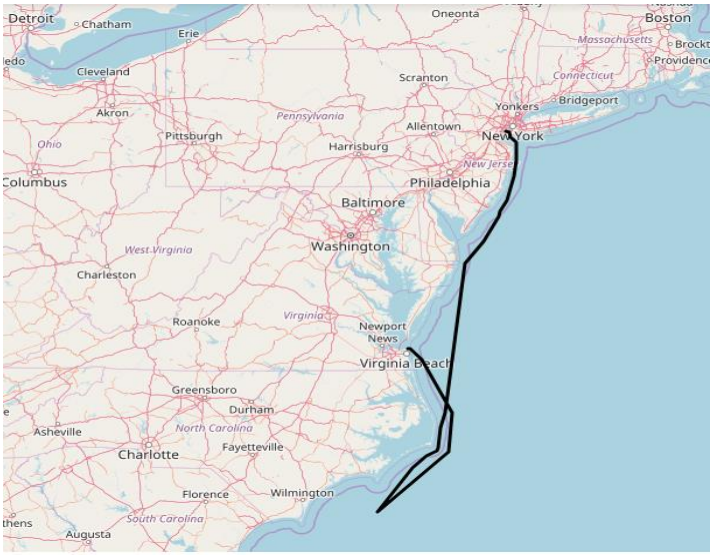
```
total time taken: 172.99462127685547
```

Fig -5 These are the output files written through the spark.



**Plotting the Sub Trajectory Fig – 6**: Below trajectories are plotted by using sub voyage id




References:


1- https://stackoverflow.com/questions/34295642/spark-add-new-column-to-dataframe-with-value-from-previous-row
2- https://stackoverflow.com/questions/37967070/using-a-shift-function-within-an-apply-function-to-compare-rows-in-a-pandas-da
3- https://spark.apache.org/docs/1.6.1/api/python/pyspark.sql.html#pyspark.sql.Column.when
4- https://stackoverflow.com/questions/10715519/conditionally-fill-column-values-based-on-another-columns-value-in-pandas
5- https://datascience.stackexchange.com/questions/8549/how-do-i-set-get-heap-size-for-spark-via-python-notebook
6- https://stackoverflow.com/questions/39401729/plot-latitude-longitude-points-from-dataframe-on-folium-map-ipython/43713646
7- https://stackoverflow.com/questions/26601576/reset-cumsum-if-over-limit-python/26605026#26605026
8- https://spark.apache.org/docs/2.1.1/api/python/_modules/pyspark/sql/functions.html