

# 6406 VISUALIZATION PROJECT PROPOSAL

## VISUALIZING AIRPORT & STATIONS DATASET

### YAMUNA JAYABALAN – B00741912

#### INTRODUCTION:

As discussed in the proposal, the project is about implementing an interactive D3 visualization tool for accessing the airport dataset collected from “<http://openflights.org/data.html>”. This dataset consists over 10,000 airports, train stations and ferry terminals operating over the globe. This data is collected and saved as a CSV file for Visualization. This project is based on the International Journal “Analysis of Airport Data using Hadoop-Hive: A Case Study”. This journal mainly states the uses of Big data and how Airport data is processed to provide various information using Big data analytics. But the concentration of this project is only on efficiently visualizing data on a very cool and interactive system.

#### MOTIVATION:

There are existing visualizations for this data, but the motivation for this implementation is to visualize this data in Google Maps, which provides all Google maps functionality to the user along with the airports and Train Stations located in the global map.

#### PRE-PROCESSING DATA:

The airport-extended dataset is taken and the rows containing “stations” are separated using python code. The separated rows are added to “stations.csv”. The “airports.csv” dataset is already provided in the flights.org website. These two datasets are used for Visualization in this project. The stations.csv has 1422 rows of data and airport.csv has 7185 rows of data.

#### Python Code:

```
import os
import re
# Traversing to file path
os.chdir("C:\\Users\\Yamuna\\Desktop\\Visualization\\Project\\preprocess\\")
# Importing dataset
import csv
with open('data.csv', 'rb') as f:
    reader = csv.reader(f)
    preproc = list(reader)
station=[]
count = 0
for i in preproc:
    if ("station" in i):
        #print i
        station.append(i)
        count = count+1
        #print station
print count
with open("stations.csv", "w") as output:
    writer = csv.writer(output, lineterminator='\n')
    writer.writerows(station)
```

## MORE ON IMPLEMENTATION USING GOOGLE MAPS:

The function used to mark the latitude and longitude on google maps is “google.maps.OverlayView()” function. Overlays are red circles placed on the map using SVG that are tied to latitude and longitude from the airport dataset, they move along with the map when user drags or perform zoom on the map.

Google Maps JavaScript API provides the OverlayView class for creating custom overlays. This OverlayView is a base class that provides methods that helps to implement custom overlays. The class also provides some methods that can be used to translate between screen coordinates and locations on the map.

### Creating Map

```
var googlemap = new google.maps.Map(d3.select("#map").node(), {
  zoom: 5,
  // assigning the zoom quotient on the first load
  center: new google.maps.LatLng(44.8807983398,-63.5085983276),
  // assigning the center coordinates when map loads first. This location has Halifax as the center.
  mapTypeId: google.maps.MapTypeId.TERRAIN
});
```

### Using Custom overlay for Visualization

- Creating a new instance of custom overlay object google.maps.OverlayView() This will create a subclass of the original overlay class.
- Creating a constructor for the custom overlay.

```
var overlay = new google.maps.OverlayView();
```

- Implementing an onAdd() method within the prototype, and attaching the custom overlay to the map. OverlayView.onAdd() is called when the map is ready for the overlay to be attached.

```
overlay.onAdd = function() {
  var layer = d3.select(this.getPanes().overlayLayer).append("div")
  .attr("class", "airports");
```

- Implementing draw() method within the prototype, and creating SVG to place the circle on the map. OverlayView.draw() will be called when the circle is displayed.

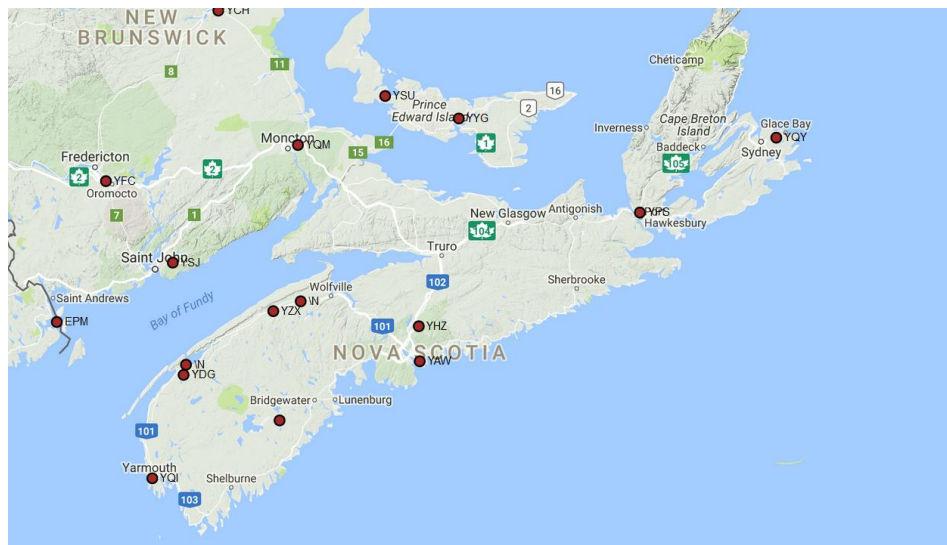
```
overlay.draw = function() {
  var projection = this.getProjection(),
  padding = 10;
```

## VISUALIZATION IMPLEMENTED & TO BE DONE

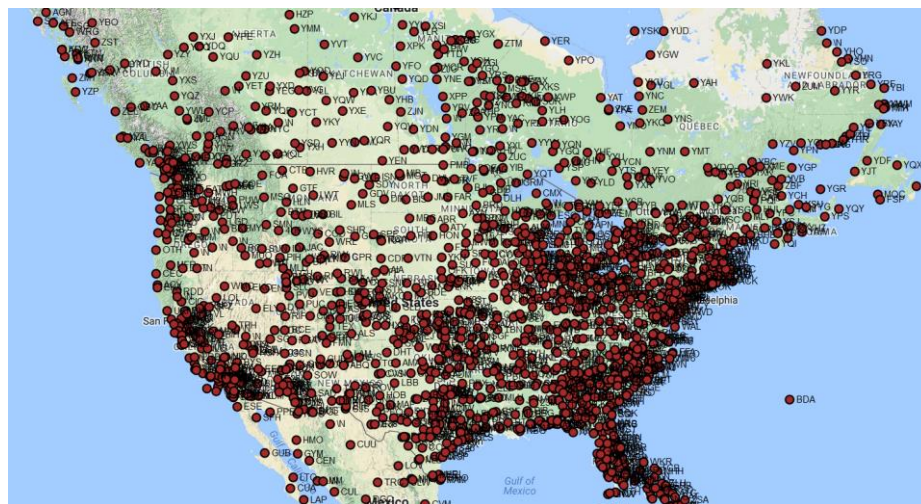
The “airports.csv” file has list of all airports globally, and the airports are plotted on the created map using latitude and longitude values. The IATA code is used to label the plots. The “stations.csv” file has a list of all the stations globally and is also plotted map shown below, it is labelled with station Name. Still working on visualizing with better User interaction for the visualization. The airport data is huge and visualizing a large dataset is very challenging.

## OUTPUTS – COMPLETED SO FAR

### Airports in Nova Scotia



### Airports in US



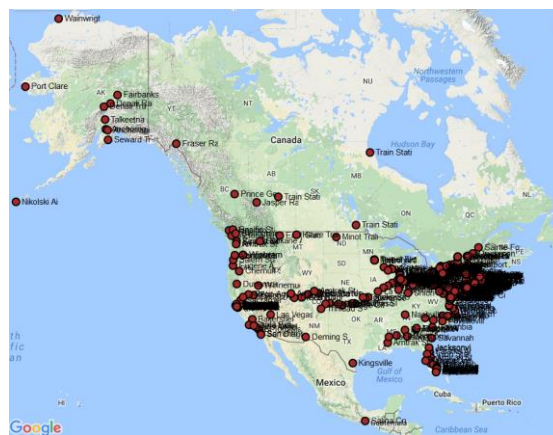
### Airports located globally



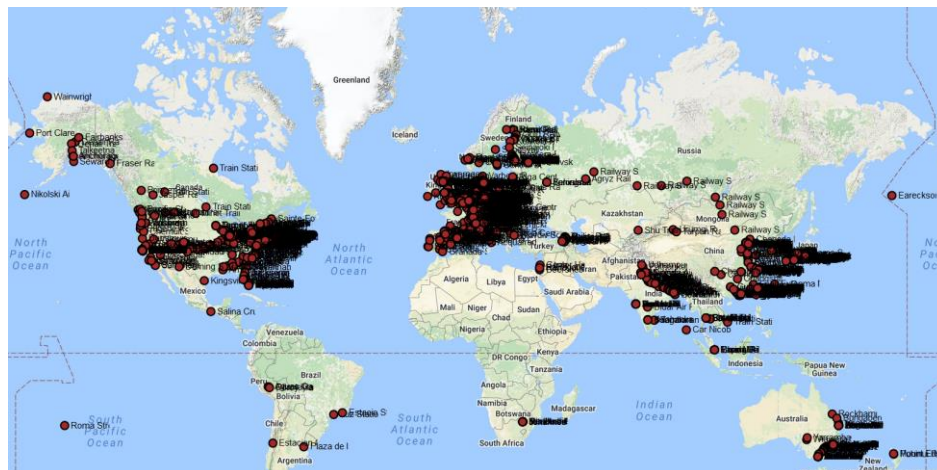
## Satellite view of Map



## Stations located in US and Canada



## Stations located across the globe



## REFERENCES

- [1] International Journal of Computer Applications, “Analysis of Airport Data using Hadoop-Hive: A Case Study”, 2016, <http://research.ijcaonline.org/ncrtit2016/number2/ncrtit201634.pdf>
- [2] Contentshare – “Airport, airline and route data”, 2017, <http://openflights.org/data.html>
- [3] Google, “Google Maps APIs”, <https://developers.google.com/maps/documentation/javascript/tutorial>