A SEMI-LAZY FLIGHT TRAJECTORY PREDICTION FOR AIR TRAFFIC MANAGEMENT

Ashwini Ravi (A0119980W)
Raghavendhra Balaraman (A0123443R)
Vishnu Gowthem Thangaraj (A0134525L)

GROUP 10

Project Background

Problem Statement

- Advancements in real-time big data analysis are changing the course of flight as we know it.
- Imagine if the pilot could augment their decision-making process with “Real time business intelligence”
- A semi-lazy algorithm that delivers a real-time flight profile to the pilot, helping them make flights more efficient and reliable on time

Data Set

Data Source : Kaggle Flight Quest Challenge

Data Size : 2.6GB

It contains all the flight information & weather info for domestic US flights during November 26, 2012 - December 9, 2012 (14 days)

The dataset contains the following types of data to aid our analysis

1. Flight History – 22 Million Records
2. Flight Plan– 0.9 Million Records
3. Weather Information – 80,000 Records

Input/Output

- Input - Flight Information

Latitude	Longitude	Departure Airport	Arrival Airport	Start Time	End Time

- Weather Input - AirSigmet Information

Airsigmet Area	Hazard Type	Hazard Severity

- Output - Predicted Trajectory

Original Trajectory	Predicted Trajectory	Speed Knots	Hazard Severity	Movement Direction Degrees

Technologies & Challenges

To store the contents to a scalable database to facilitate easy querying :

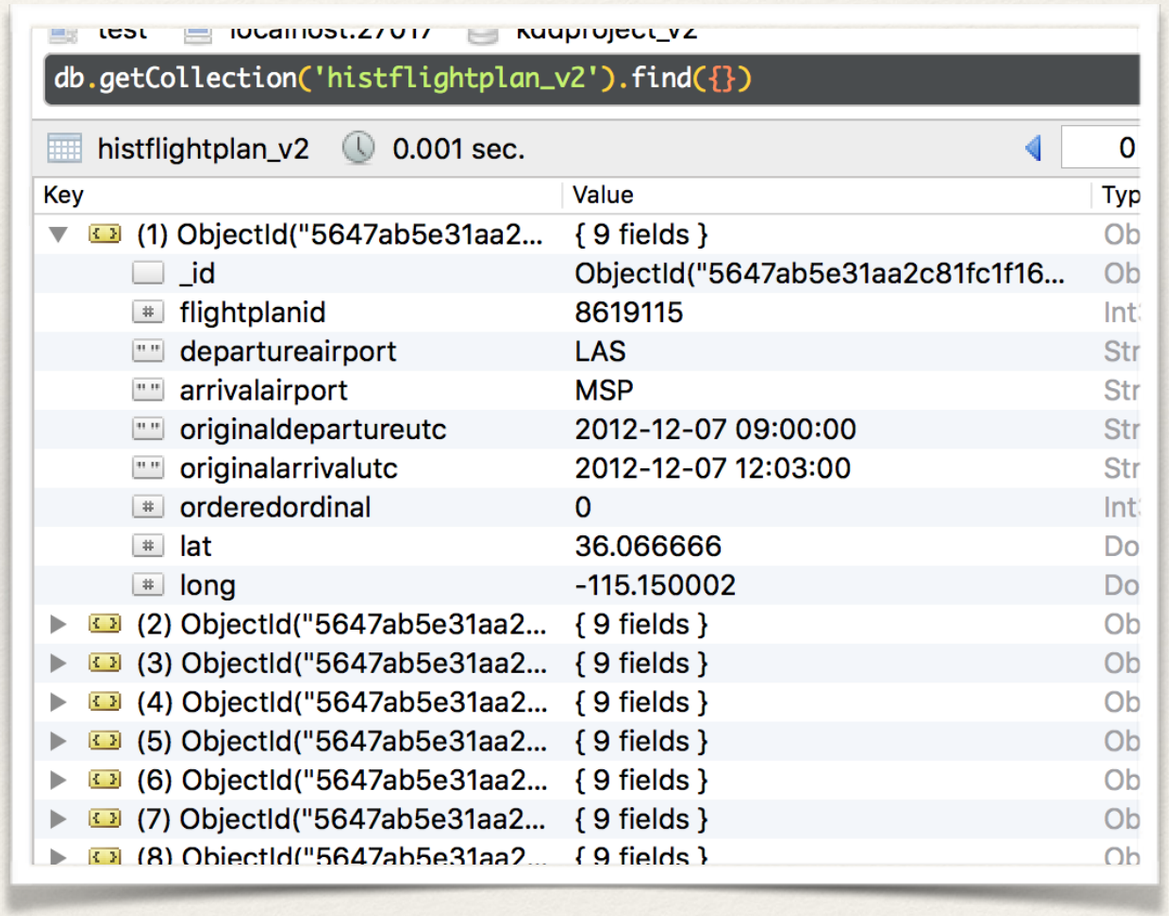
- We loaded the dataset into SQLITE3, considering its light weight and easy portable nature.
- We then shifted to MYSQL - another popular open source relational DB choice.

However the data was so large that querying took considerable time and reduced the performance.

Solution - MongoDB

Why MongoDB?

- Leading NoSQL database.
- Take advantage of its **JSON** format documents and dynamic schemas.
- Better performance due to Indexing.



Platform

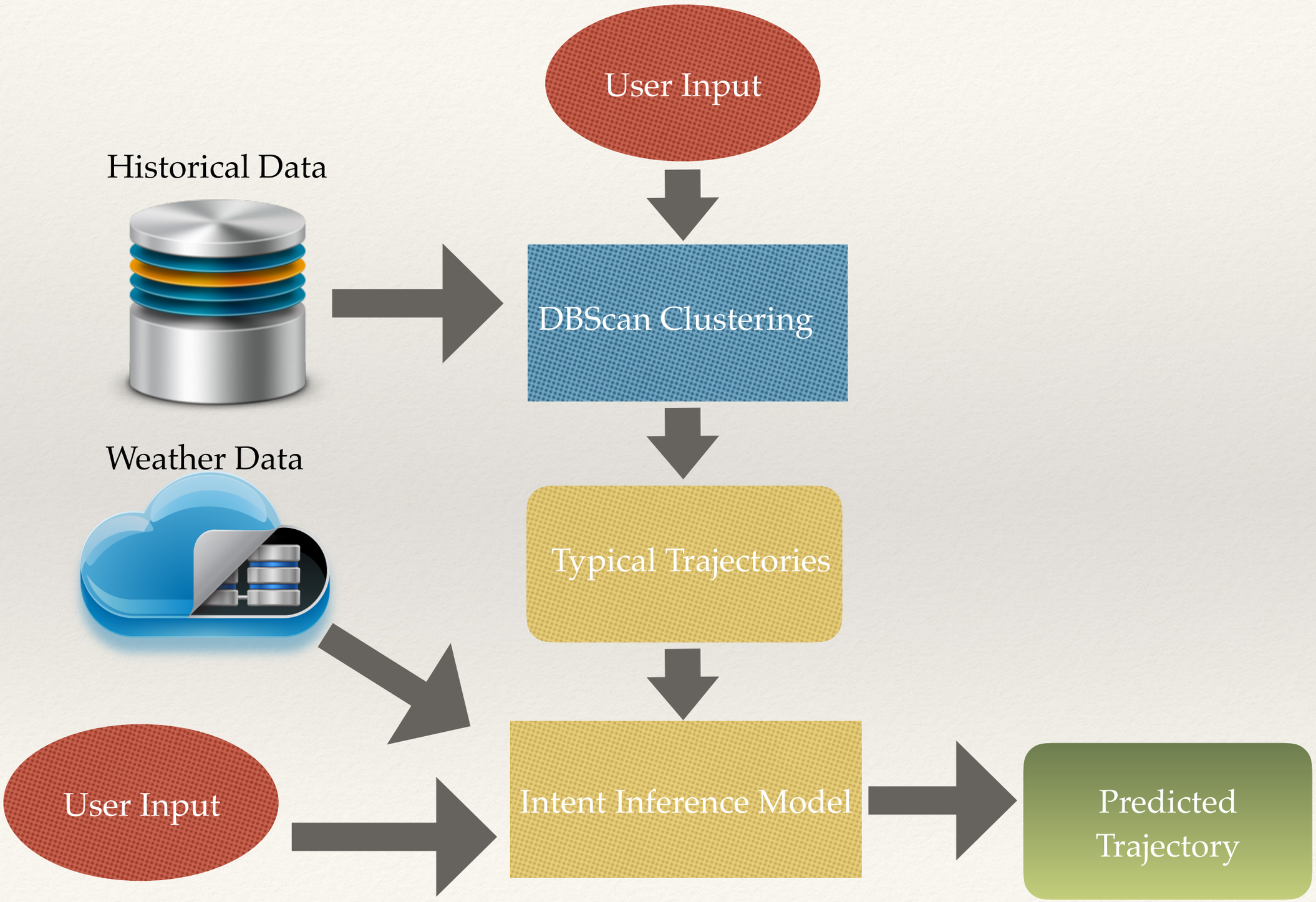
- **Configuration :** OS X El Capitan, 2.6 GHz Intel Core i5, 8 GB
- **DB :** MongoDB (DB Client - Robomongo)
- **Programming Language :** Python
- **Python packages :** pandas, pymongo, numpy, matplotlib, networkx

Shortcomings of Existing Models

1. **Constant Model** - Does not handle dynamic change in data.
2. **Lazy learning** on existing models infers a very simplistic prediction
3. **Eager learning** on existing model increases the computation cost/not a pragmatic solution.

A semi-lazy paradigm on flight trajectory models strikes a fair balance between accuracy and performance.

Proposed Solution



Implementation

Data Pre-Processing

Sampling :

Sample the trajectories, to obtain time series of equal length for each aircraft.

$$P_{ij}' = P_{ij}, j = \text{round}((j'*L_i)/L), j'=1,2\ldots,L$$

- Historical trajectories are sampled to equal length of the input flight plan
- The time duration of the flight are sampled to identify the time at any point in the trajectory.

DBSCAN Clustering Solution

Why DBSCAN?

Fits well for our high density, spatio-temporal data.

Does not need to know the number of types before clustering

Method :

The historical trajectory data are fed into DBSCAN algorithm whose cluster determines the Source-Destination trajectories.

The typical trajectories are built by considering clusters that are similar to the input trajectory.

Clustering Output

```
Cluster:0
8636286,8767258,8785505,8800831,9088690,9102183,9128975,9159820,9177645,9193601,9307236,9358583,9598048,9617735,9642780,9665194,9680150,970061
4,8783442,9175654,9697494,9654742
Cluster:  1
8643405,8657674,8673054,8814722,9420888,9432147,9445306,8711155,8962525
Cluster:  2
8697838,8711308,8737716,9034893,9048872
Cluster:  3
8834770,8857274,8886670,9028133,9227505,9250233
Cluster:  4
8915170,8936497,8967159,9551196,9566274
Cluster:  5
9467418,9479693,9505405

Centroid  0
(37.6273535834917, -92.61746431809526)
Centroid  1
(37.55649009111807, -92.70794983764732)
Centroid  2
(37.57339475781138, -92.66957424923632)
Centroid  3
(37.491390816629995, -92.76980974068454)
Centroid  4
(37.52591692751105, -92.73339662335135)
Centroid  5
(37.597580632855816, -92.64381728633758)

The input historical trajectory belongs to cluster [ 0 ,   2 ,   3 ,   5 ]
```

Abstract Typical Trajectories by Clustering

Field	Example
Flight Number	AAR315
Time	2012-11-27 15:07:28
X Coordinate	36.0666656494141
Y Coordinate	-115.150001525879

- Typical Trajectory Library, $T = \{T_t, t=1,\ldots.n)$
- In the library, each trajectory record is : $T_t = \{p_{tj}, j=1,\ldots l_t\}$
- Each point : $p_{tj} = \{x_{jtk}, k=1,\ldots,w\}$

The above expressions mean that the trajectory library contains *n* trajectories, each trajectory T_t contains l_t

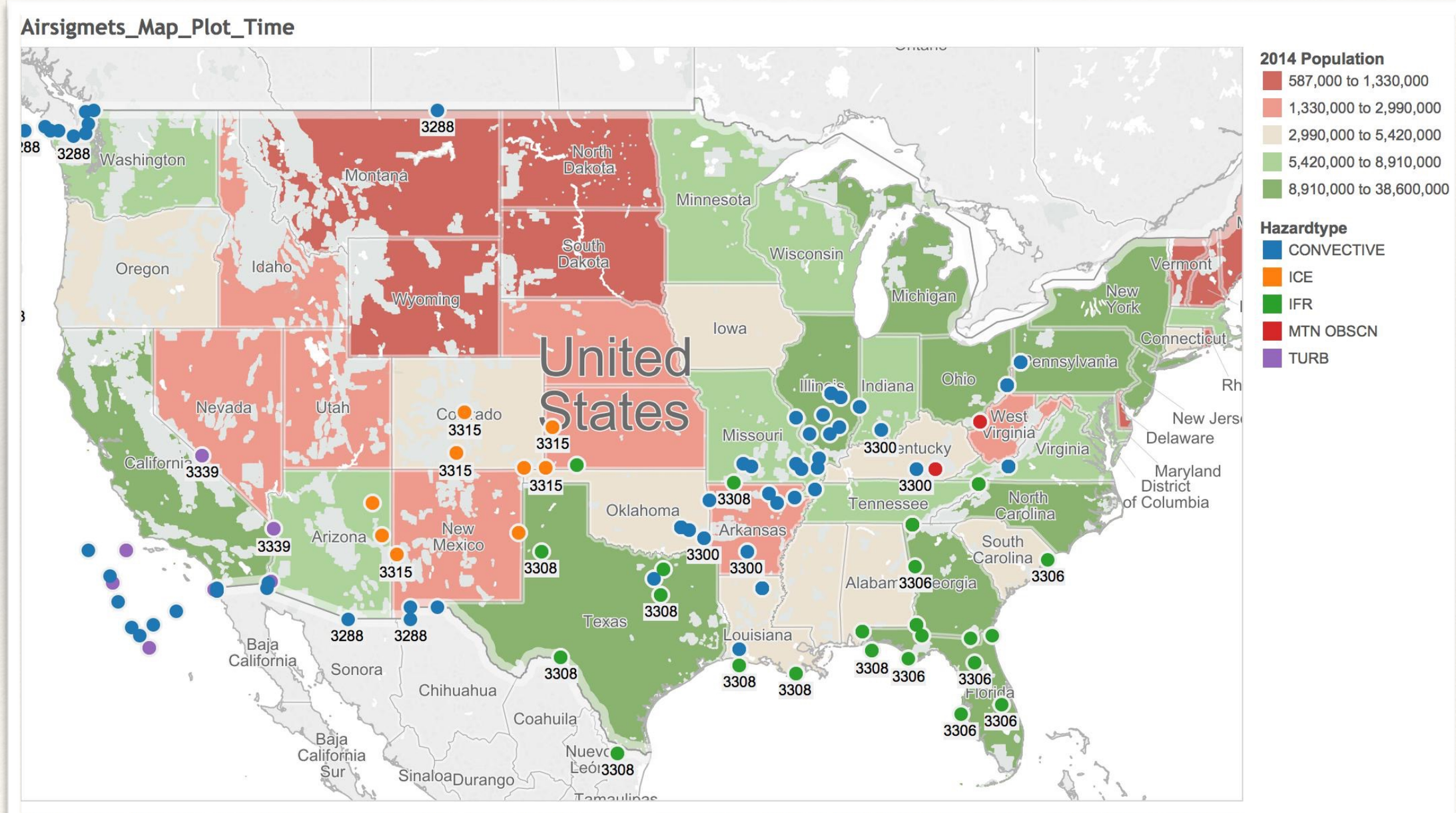
Intent Inference Path Prediction

Why use AIRSIGMETS? (Airmets,Sigmets)

- Airman's significant weather advisory information available to the aircraft.
- **Area :** Recorded every 4 hours.
- Types of hazards and their severity :

Hazard Type	Convective, Ice, Turbulence, IFR, MTN OBSCN
Hazard Severity	Moderate, Extreme, None

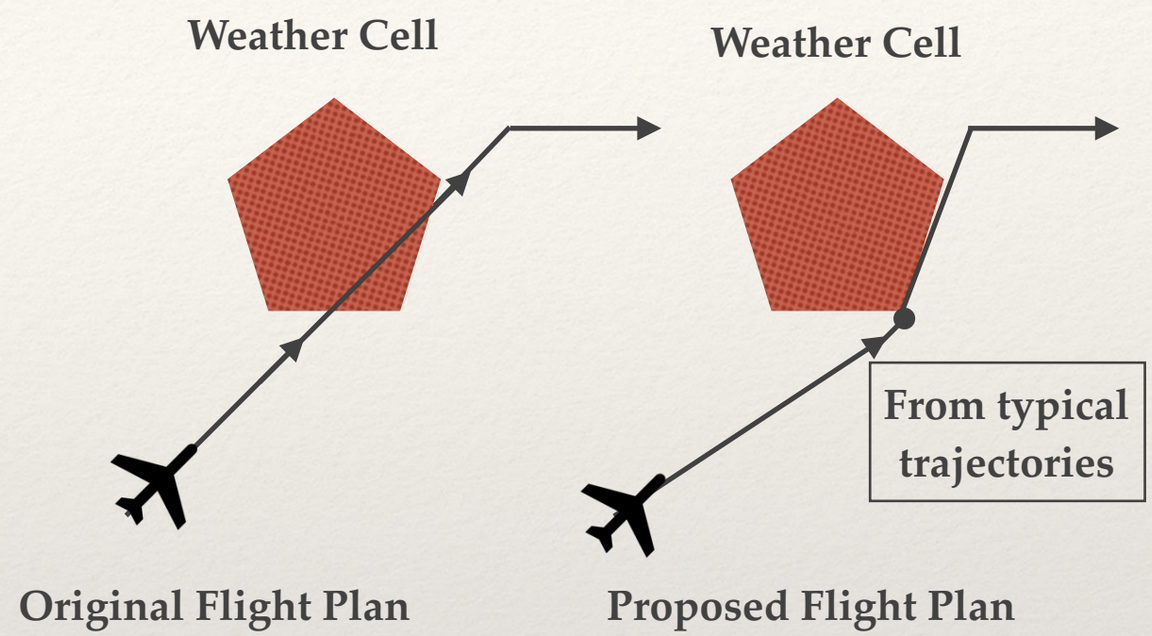
Data Visualization - Airsigmets

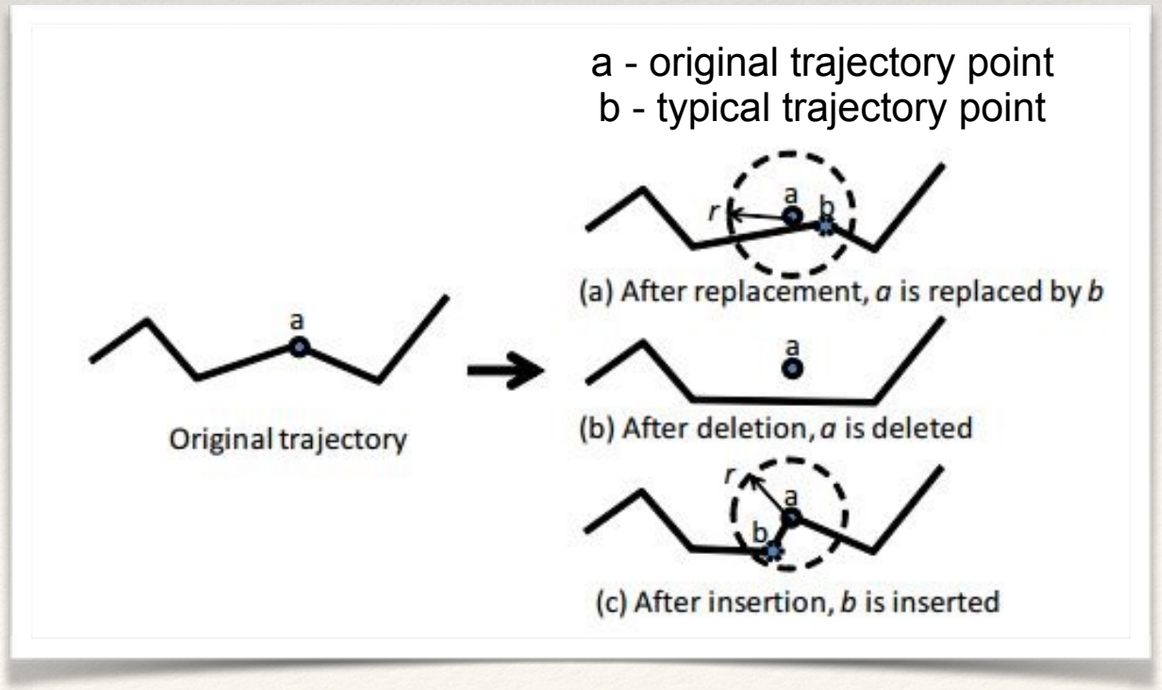


Map based on Airsigmets across United States. Color shows details about Hazardtype. The numbers below are labeled by Airsigmetid. Map coloring shows populous areas. The data is filtered on Timevalidfrom and Timevalid for sigmets.

Intent-based Model - Solution

- For each point in input trajectory, its corresponding time is compared against the dynamic airsigmet time ranges.
- All the input trajectory points that falls within the airsigmet area is recommended with an alternate point information using the typical trajectory data:
  - The typical trajectory points that had deviated a similar airsigmet area is considered and recommended.
- The flight continues on its flight plan trajectory after avoiding the weather cell.
- The same process repeats until the flight reaches its destination.





Results

Result Analysis : DFW-ORD

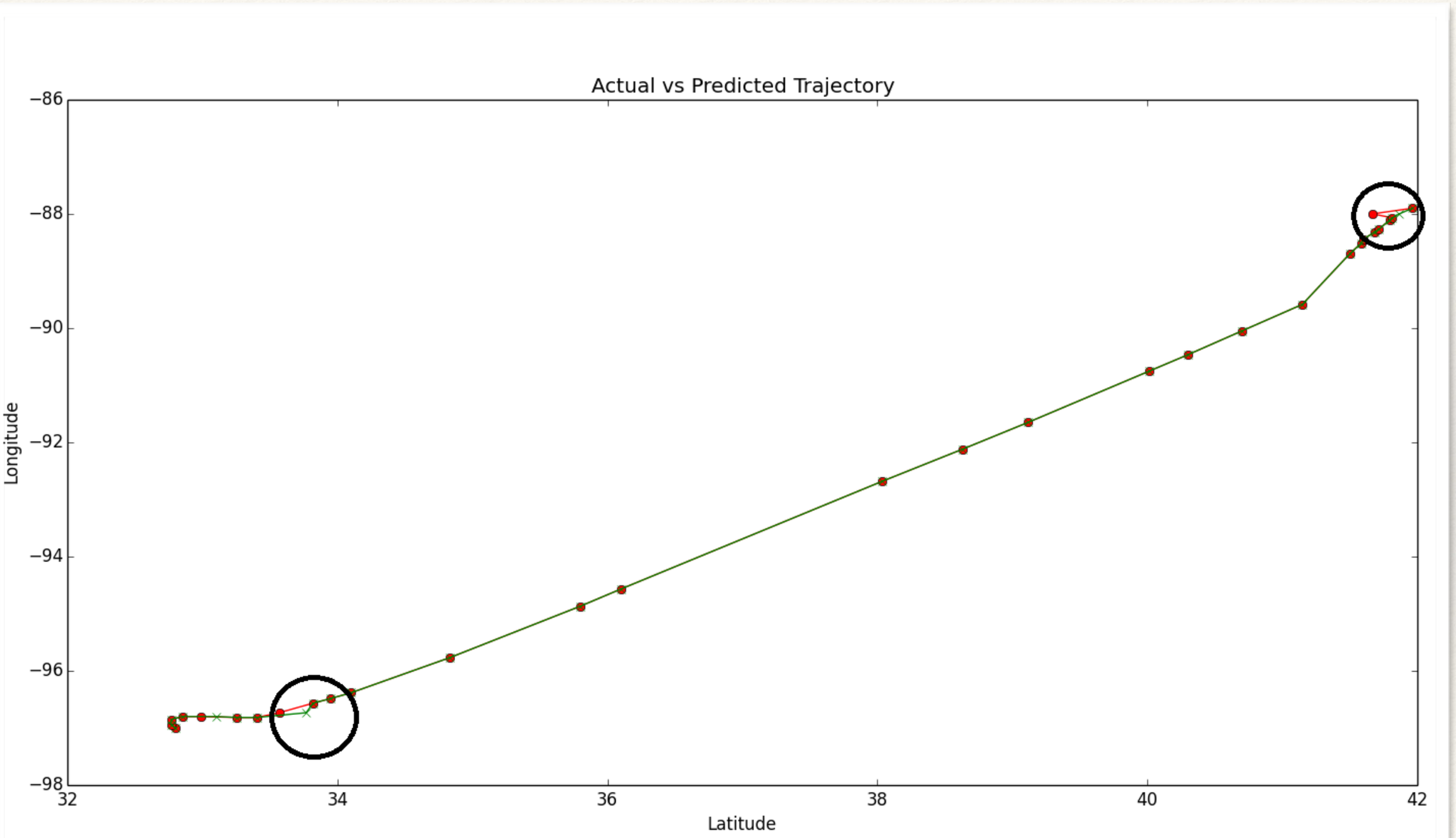
Input Flight Plan

Latitude	Longitude
32.79999924	-97
32.76666641	-96.94999695
32.76666641	-96.84999847
32.84999847	-96.80000305
32.98333359	-96.80000305
33.25	-96.81666565
33.4000015259	-96.8166656494
33.5666656494	-96.7333297729
33.9500007629	-96.4833297729
34.8333320618	-95.7666702271
35.7999992371	-94.8666687012
36.0999984741	-94.5666656494
38.03333282	-92.68333435
38.63333511	-92.1166687
39.11666489	-91.65000153
40.01666641	-90.75
40.29999924	-90.46666718
40.70000076	-90.05000305
41.15000153	-89.58333588

Predicted Flight Path

Latitude	Longitude
32.79999924	-97
32.76666641	-96.94999695
32.76666641	-96.84999847
32.84999847	-96.80000305
32.98333359	-96.80000305
33.25	-96.81666565
33.4000015258789	-96.8166656494141
33.8166656494141	-96.5666656494141
34.0999984741211	-96.3833312988281
34.9833335876465	-95.6500015258789
35.6500015258789	-95.0
36.533332824707	-94.1333312988281
38.03333282	-92.68333435
38.63333511	-92.1166687
39.11666489	-91.65000153
40.01666641	-90.75
40.29999924	-90.46666718
40.70000076	-90.05000305
41.15000153	-89.58333588

Actual vs Predicted Trajectory



Performance Benchmarking

Benchmark	Semi-lazy Approach	Eager Learning
Querying all trajectories between Departure & Arrival	6m12.036s	Total time : **30 mins & Force-Quit**
Sampling & identifying Typical trajectories :	4m0.494s	
Trajectory Prediction : (Intent Based Model)	1m20.165s	
Total Time	**11m32.695s**	

Future Scope

- **Conflict Detection :** Finding conflicts to manage air traffic.
- **Air Space Health Management :** Measuring the robustness of an air space given different weather data against historical data.
- **Extending other weather factors** such as wind speed,wind directions etc.
- **Supervised learning** - Model to learn from the best practices.

Thank you !