Transforming a Traditional Database Into a Full-fledged Computing Platform with Machine Learning and SQL

(OCIENT)™

(OCIENT)™

JASON ARNOLD

Co-founder & Distinguished Engineer

- 17 years in the database world
- Focus: Query Optimization, Query Performance, ML, and Geospatial

(OCIENT)™

GEOSPATIAL ML AND AI FOR EVERYONE

Geospatial data volumes are exploding as is the use of data warehouses.

Can modern data warehouse systems enable wider access to ML and AI tools for geospatial and general analytics?

(OCIENT)™

INTRODUCING OCIENT

Geospatial SQL

- Based on WGS84 standard spheroid model of the earth
- Support for Geographies only
- 120+ Native ST functions
- Spatial Index based on a Hilbert R-Tree index

(OCIENT)™

SPATIOTEMPORAL SQL

ANALYZING SPACE AND TIME

What are Spatiotemporal SQL Analytics?

- Analyze and track moving objects (time-stamped geospatial data)
- Query on time or location to determine the relationship of one to the other
- Ocient supports nine spatiotemporal functions natively
  - ST_LINEGETALLTIMESATPOINT
  - ST_LINEGETTIMEATPOINT
  - ST_LINEGETPOINTATTIME
  - Special versions of pre-existing functions
    - ST_LONGESTLINE
    - ST_SHORTESTLINE
    - ST_INTERSECTION

This is used to find:

- The location of a moving object at a specific point in time
- How long objects did not have signal
- The trajectory of a specific moving object over time
- The time range where an object was in a particular zone or area

LINEAR ALGEBRA IN A DATABASE

THE FOUNDATION OF MACHINE LEARNING

(OCIENT)™

Matrices as First-class Data Types

Matrices underpin of much of ML

Ocient supports Matrices as native column types - *includes support for row and column vectors*

All major functions exist natively:

- Eigenvectors and eigenvalues
- Inverse
- Transpose
- Determinant
- QR, LUP, and SVD decompositions

(OCIENT)™

LINEAR ALGEBRA IN A DATABASE

A BASIC EXAMPLE

```
Ocient> select eigen(inverse(2 * {{1,2},{3,4}} + {{5,6},{7,8}} / 2));


eigen(inverse((((2))*({{1,2},{3,4}}))+(({{5,6},{7,8}})/((2)))))
--------------------------------------------------------------------------------
[<<-1.3780529228406495, [[0.8013353799887076, -0.5982153531783962]]>>, <<0.05805292284064968,
[[0.4819655926750846 5, 0.876190143449 1]]>>]
```

(OCIENT)™

Leveraging In-database Machine Learning on Geospatial Data to Identify Objects in Motion through Zoned Regions

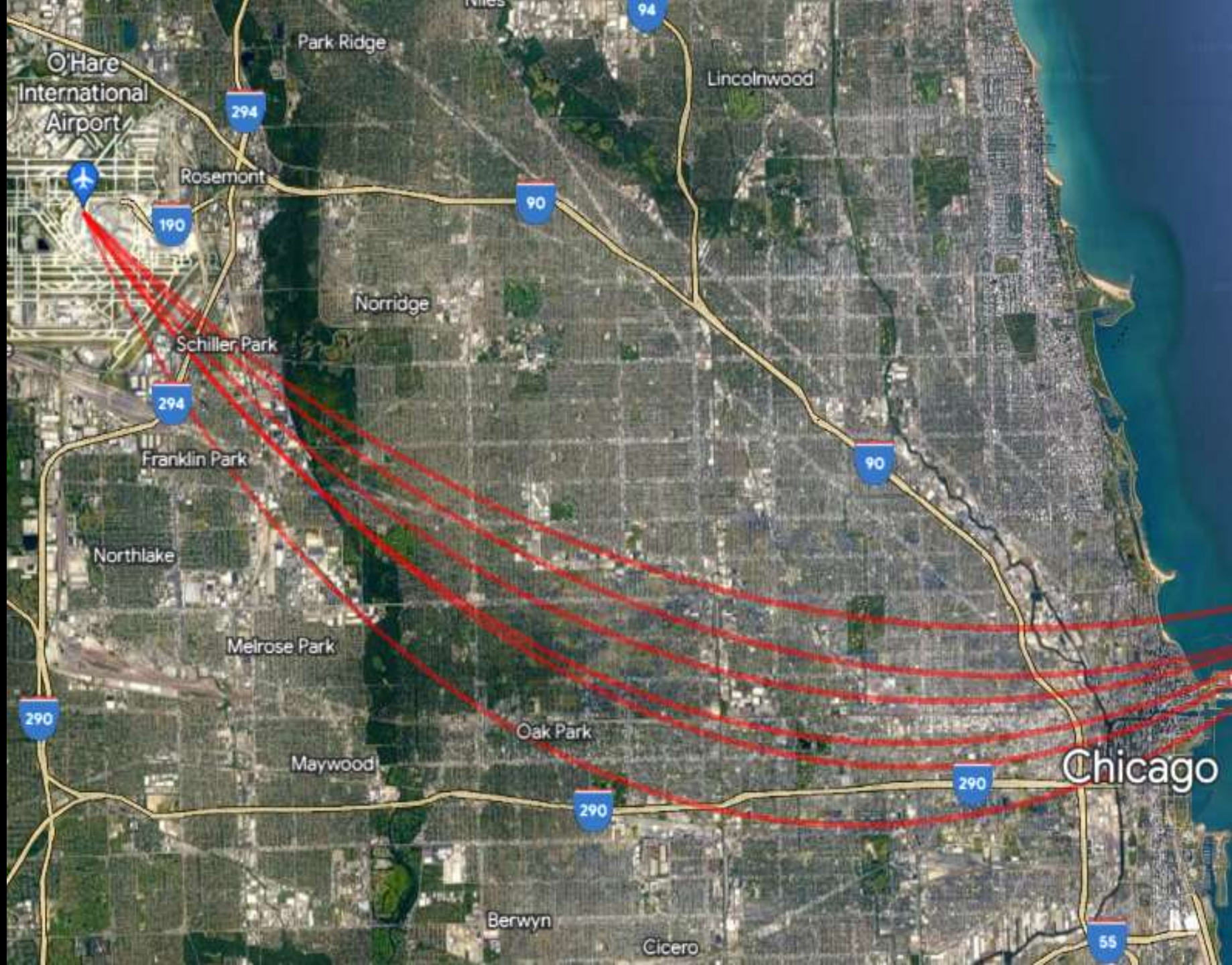
(OCIENT)™

THE DATA

Two Major Sets of Data Used

A set of "paths" denote objects in motion

- Each path is a set of points and timestamp pairs
- In this example, each path is a "flight" leaving O'Hare airport in Chicago



(OCIENT)™

THE DATA

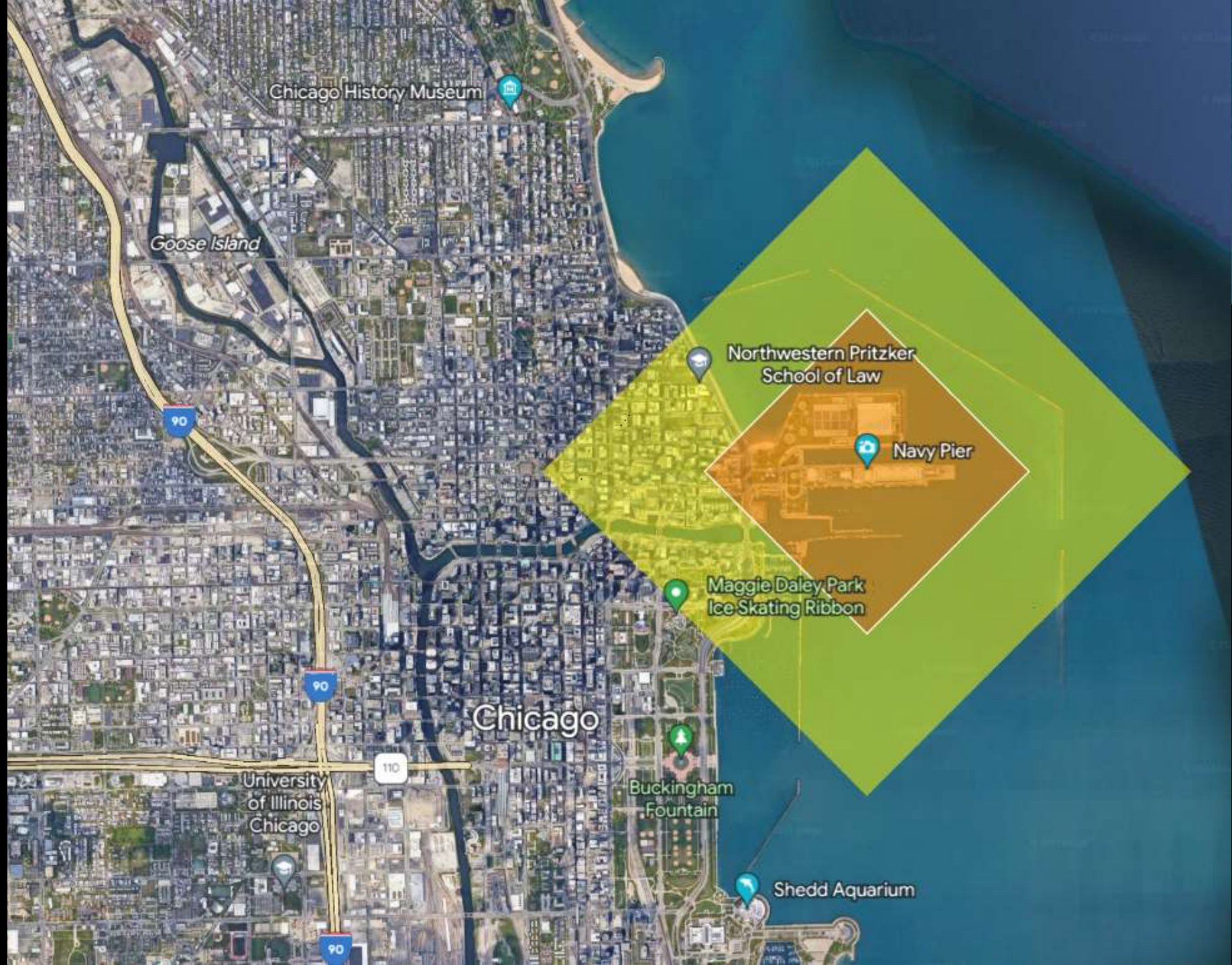
Two Major Sets of Data Used

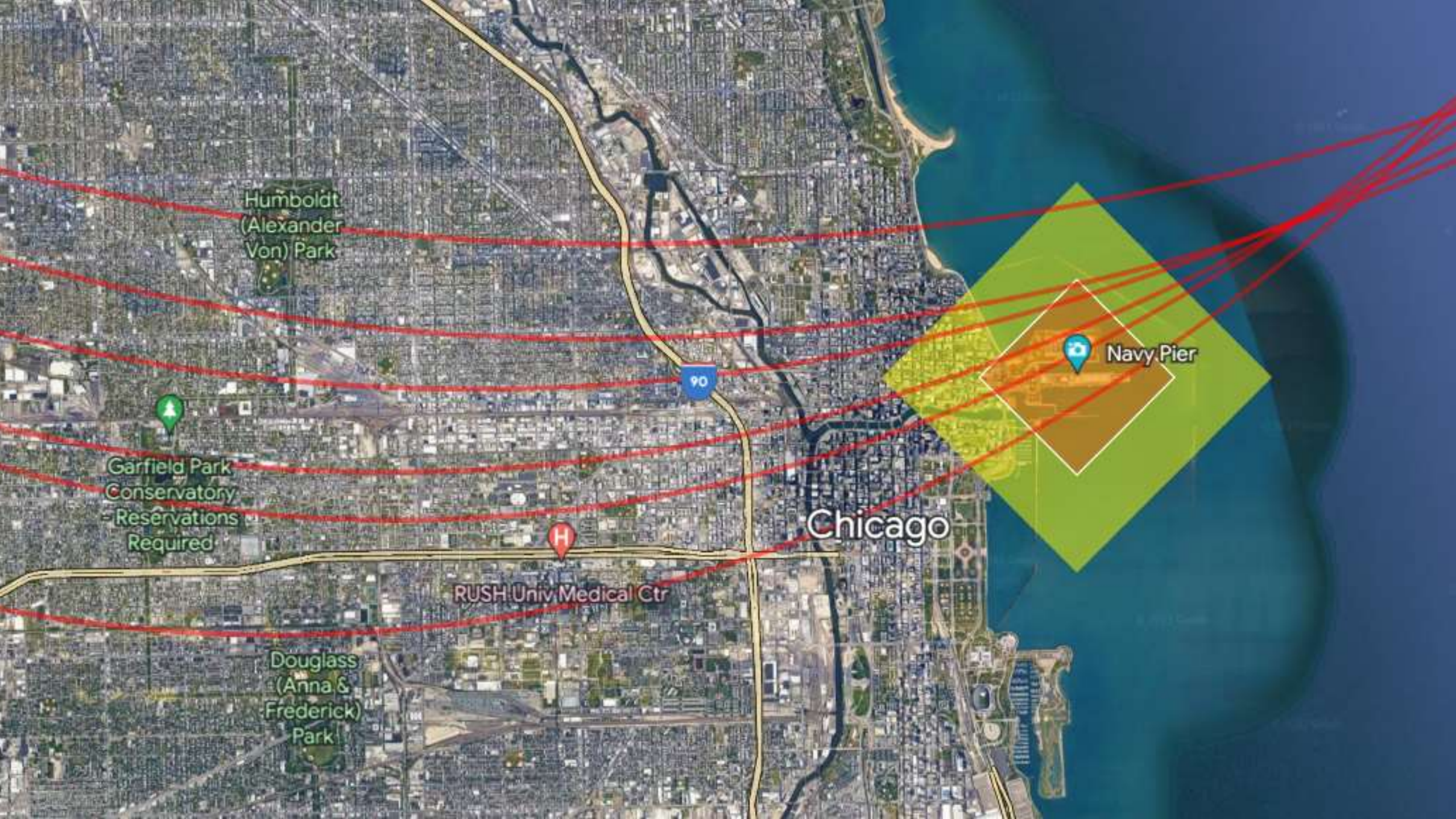
A set of "paths" denote objects in motion

- Each path is a set of points and timestamp pairs
- In this example, each path is a "flight" leaving O'Hare airport in Chicago

Two polygons representing our "red" and "yellow" zones

- In this example, the zones are airspace around Navy Pier in Chicago




Humboldt (Alexander Von) Park
90
Navy Pier
Garfield Park Conservatory - Reservations Required
Chicago
RUSH Univ Medical Ctr
Douglass (Anna & Frederick) Park

(OCIENT)™

IN-DATABASE MACHINE LEARNING

HOW DO WE IDENTIFY THESE OBJECTS?

Solving the Problem

- The goal was to predict time in red zone based on three tracking points once an object was in the yellow zone.
- A neural net model seemed like a good thing to try.
- Pre-processing with PCA seemed like it might help.
- After a couple of tries, I had a model with ~50% variance explained on a validation set

```
create mlmodel prep_the_data type principal component analysis on (
with points as (
select st_pointn(st_intersectionarray(tracking, st_polygon('POLYGON((-87.62926210044512 41.89189746763829, -87.6051 41.87391358815503, -87.58093789955488 41.89189746763829, -87.6051 41.90988641184496, -87.62926210044512 41.89189746763829))'))[1], 2) as p1,
…,
secs
from jason.yellow)
select st_x(p1), st_y(p1), st_x(p2), st_y(p2), st_x(p3), st_y(p3) from points where p1 is not null and p2 is not null and p3 is not null)

create mlmodel time_in_red_zone2 type feedforward network on (with points as (
select st_pointn(st_intersectionarray(tracking, st_polygon('POLYGON((-87.62926210044512 41.89189746763829, -87.6051 41.87391358815503, -87.58093789955488 41.89189746763829, -87.6051 41.90988641184496, -87.62926210044512 41.89189746763829))'))[1], 2) as p1,
…,
secs
from jason.yellow),
process as (select st_x(p1) as x1, st_x(p2) as x2, st_x(p3) as x3, st_y(p1) as y1, st_y(p2) as y2, st_y(p3) as y3, secs from points where p1 is not null and p2 is not null and p3 is not null)
select prep_the_data(x1, y1, x2, y2, x3, y3, 1), prep_the_data(x1, y1, x2, y2, x3, y3, 2), secs from process)
options('metrics' -> 'true', 'hiddenLayers'->'3', 'hiddenLayerSize' -> '8', 'outputs' -> '1', 'lossFunction' -> 'squared_error', 'outputActivationFunction' -> 'relu');
```

(OCIENT)™

EVALUATING OUR MODEL

Did it work?

```
WITH points AS (.....)

SELECT 1.0 - (sum(power(estimated - actual, 2.0)) / sum(power(actual - mean, 2.0)))
FROM   residuals,
       the_avg;



((1.0))-((sum(power((estimated)-(actual), (2.0))))/(sum(power((actual)-(mean), (2.0)))))
-----------------------------------------------------------------------------------------
0.5089509235121586
```

(OCIENT)™

THANK YOU

For more information about Ocient
and our capabilities visit ocient.com.