

New York Police Department Injury Report

Presented by
Lucky Trang, Ron Tsang
Stephanie Nguyen, Paul Yi

Intro

Body

CON

REC

Abstract

According to the World Health Organization, approximately, 1.35 million people die each year from car collisions.

Continue

Data Source

Git Hub

Abstract (cont.)

- The NYPD started to collect this data at the beginning of 2012, it is manually run every month and reviewed by the Traffic stat unit.
- With the 2016-2019 NYPD Motor Vehicle Collisions data set (162.24 MB), we plan on predicting future injuries from car accidents and providing possible solutions to prevent these injuries.

Data Source

Data set
measurement is
162.24 in Mega
Bytes.

ML
Azure

Elastic
Search

ML Azure Experimental Specification

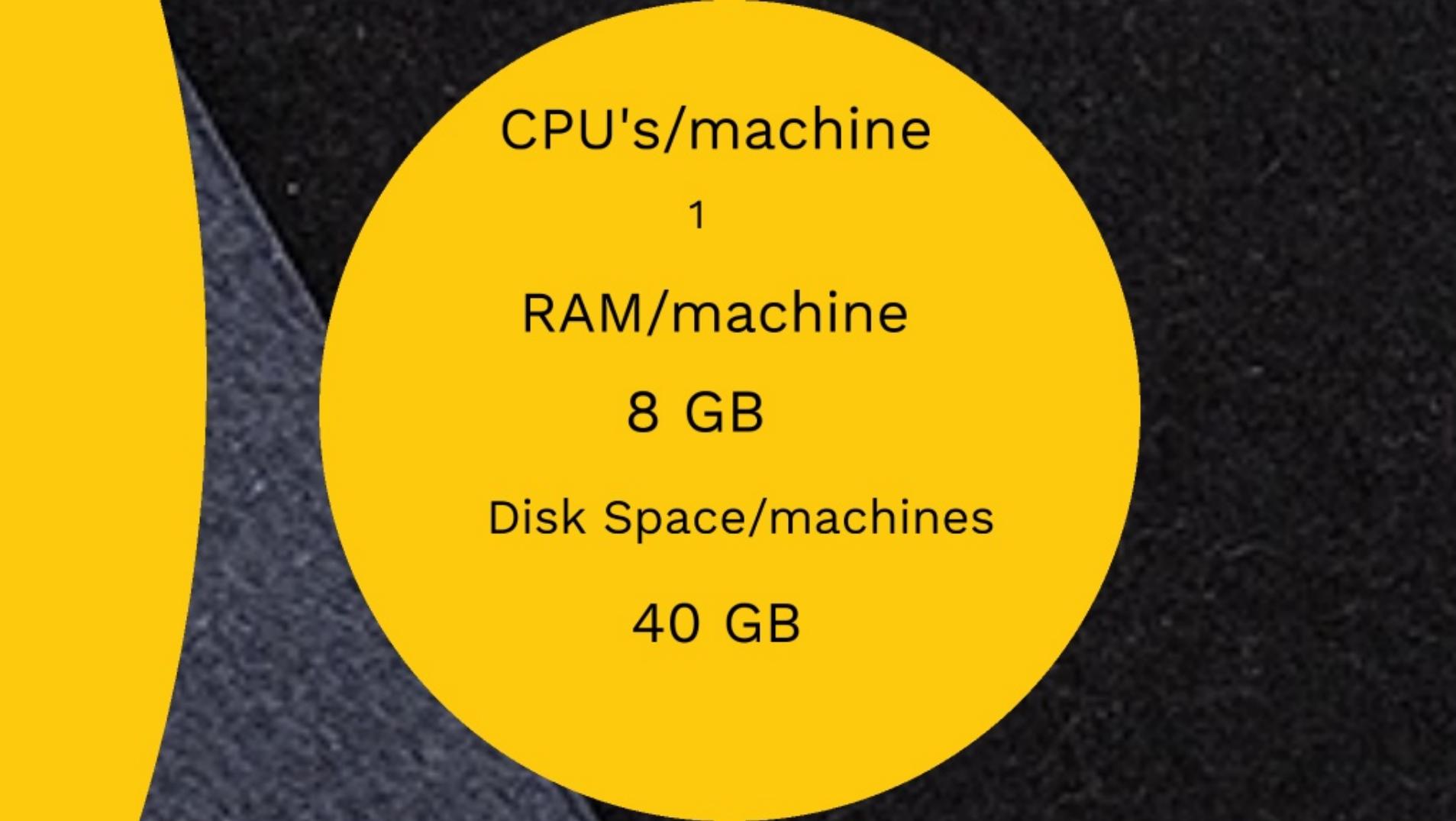
Limitation on Data-set

- 150 MB or more

Speed

- Depends, usually it runs fast

Continued



CPU's/machine

1

RAM/machine

8 GB

Disk Space/machines

40 GB

Elastic Search & Kibana Specification

Memory

64 GB of RAM (recommended)
32 GB and 16 GB machines are commonly used.

CPU's

You should choose a modern processor with multiple cores.

Disks

Disks

Disks are important for all clusters, and doubly so for indexing-heavy clusters. (such as those that ingest log data).



Git Hub Link

Link here

New York Police Department Injury Report

Presented by
Lucky Trang, Ron Tsang
Stephanie Nguyen, Paul Yi

Intro

Body

CON

REC

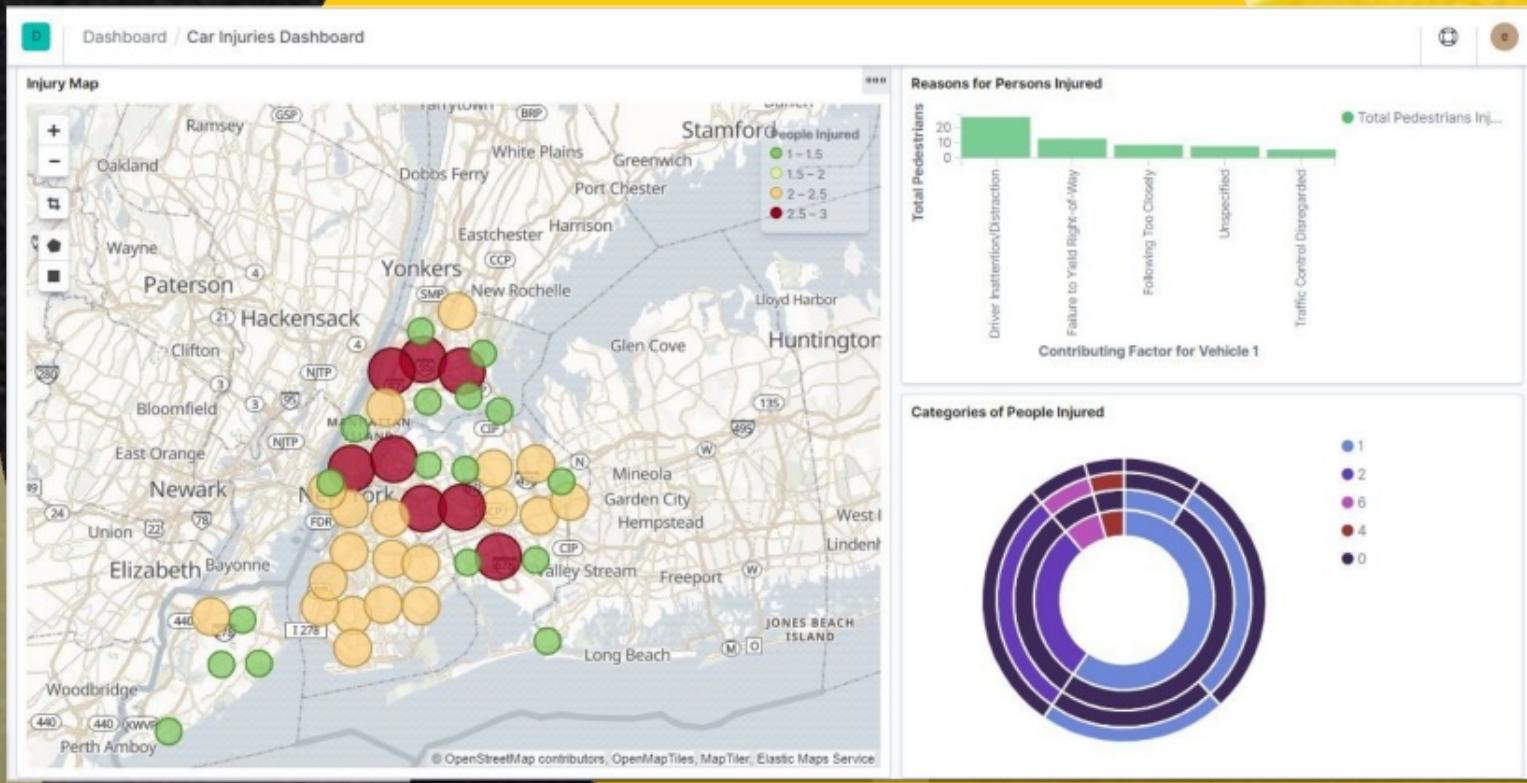
ML Azure and ES Kibana Tutorials

- ML Azure Data-set has total of 17 modules.
- Geo Points and Visualizations are used to locate the areas of accident.

Elastic
Search
Visualization

ML
Azure
Tutorial

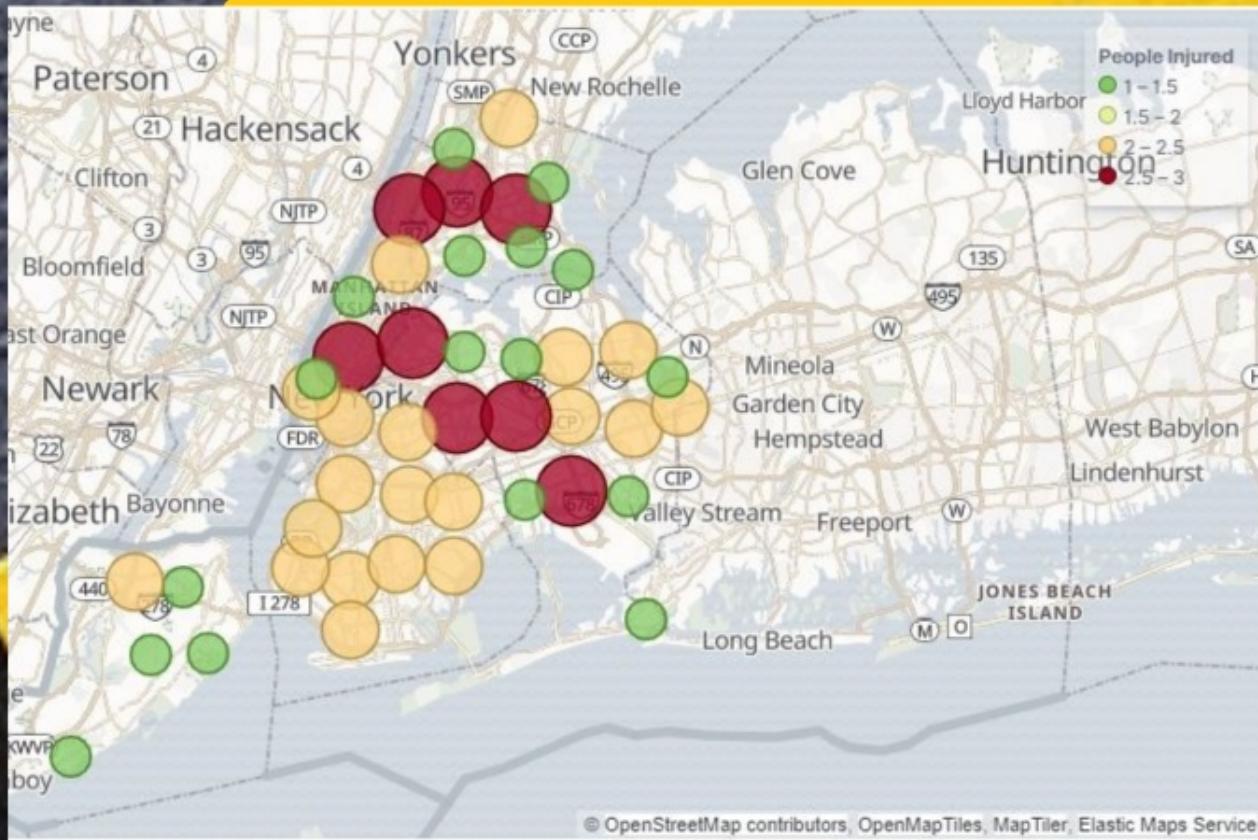
Elastic Search Dashboard



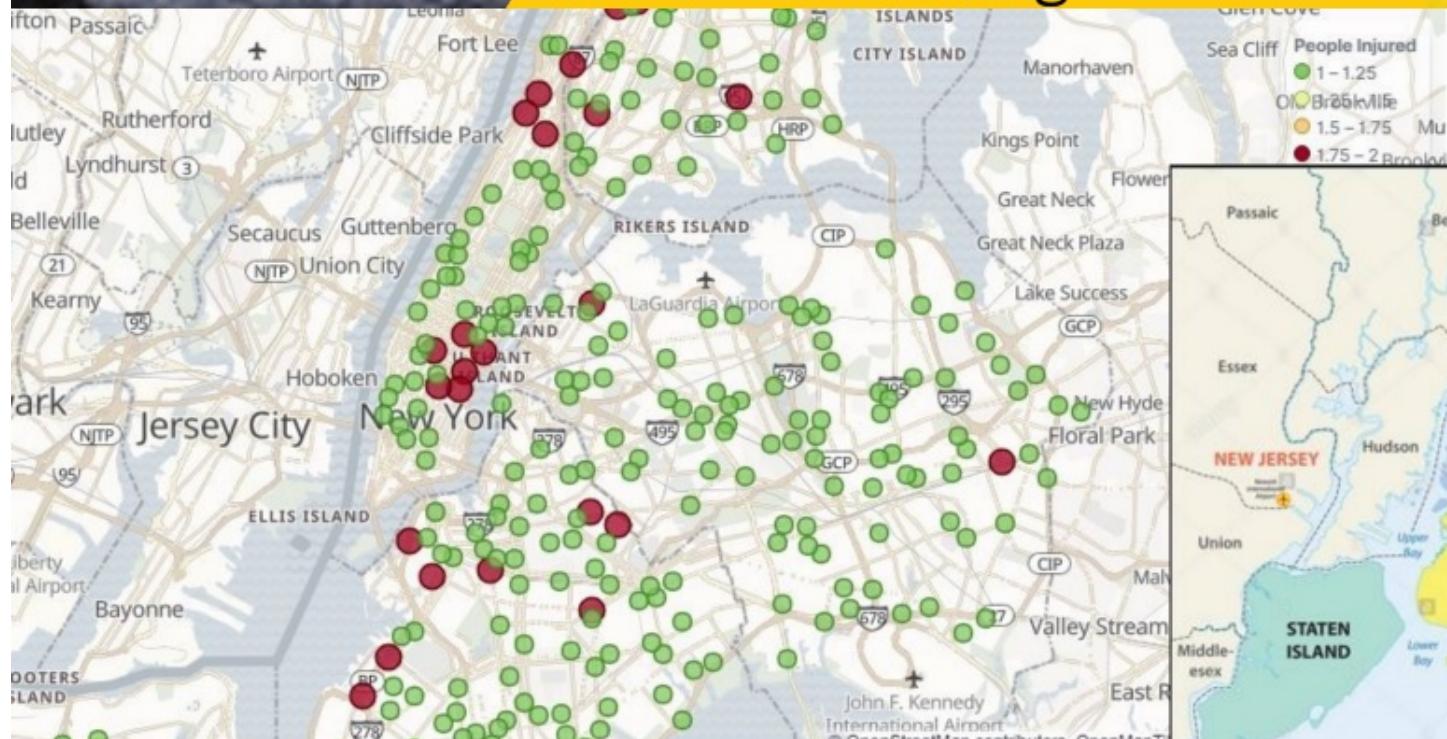
Elastic
Search
Geo Po

Elastic
Search
Map

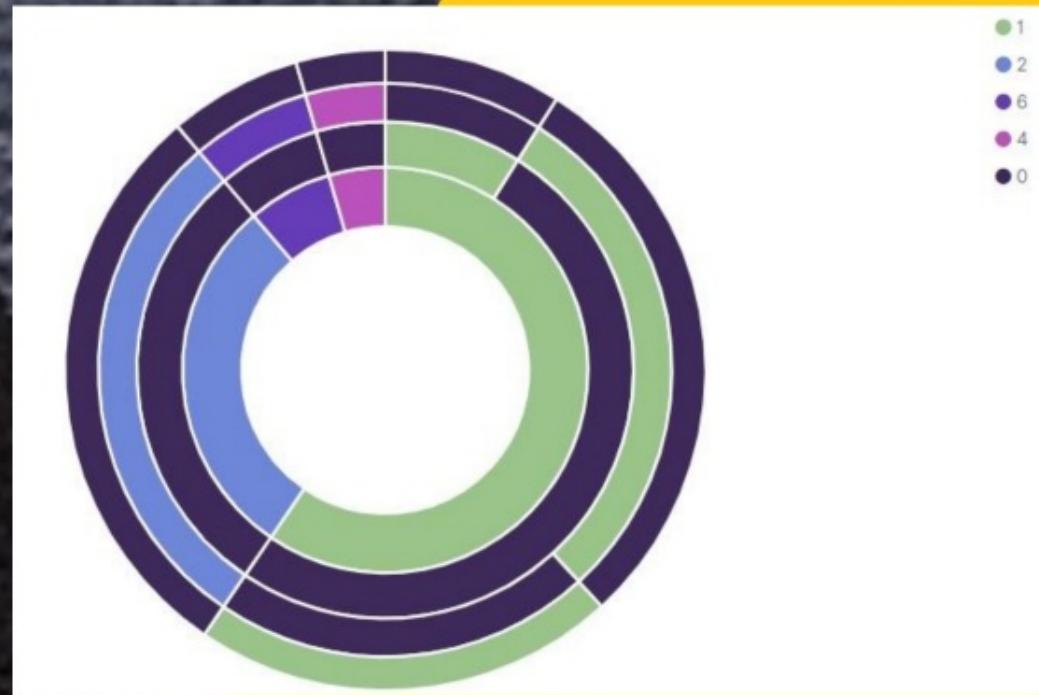
Elastic Search Map



Elastic Search Map - Boroughs



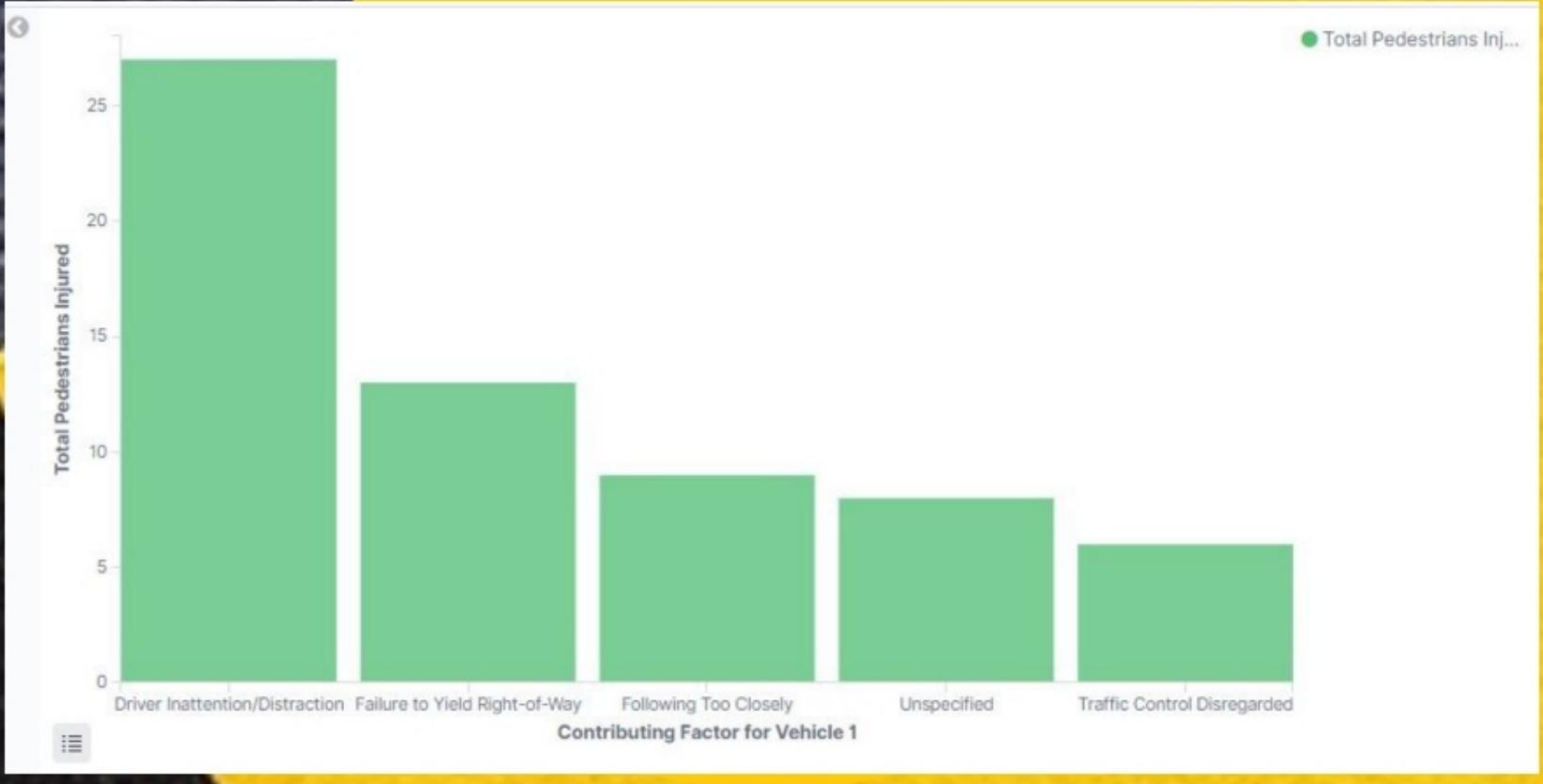
Elastic Search - Categories of Injuries



Total: 137

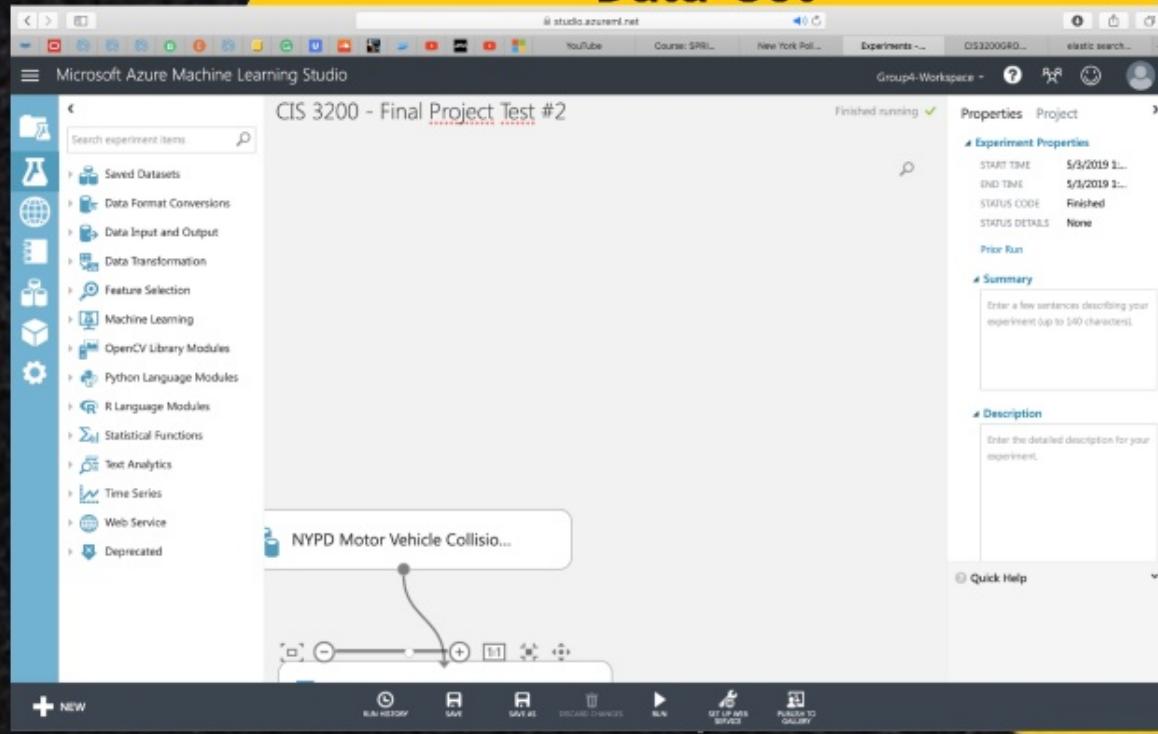
- Cyclists:
 - 1 - 15.09%
- Motorists
 - 1 - 49.06%
 - 2 - 29.21%
 - 4 - 4.49%
 - 6 - 6.74%
- Pedestrians:
 - 1 - 35.85%

Elastic Search - Contribution Factors



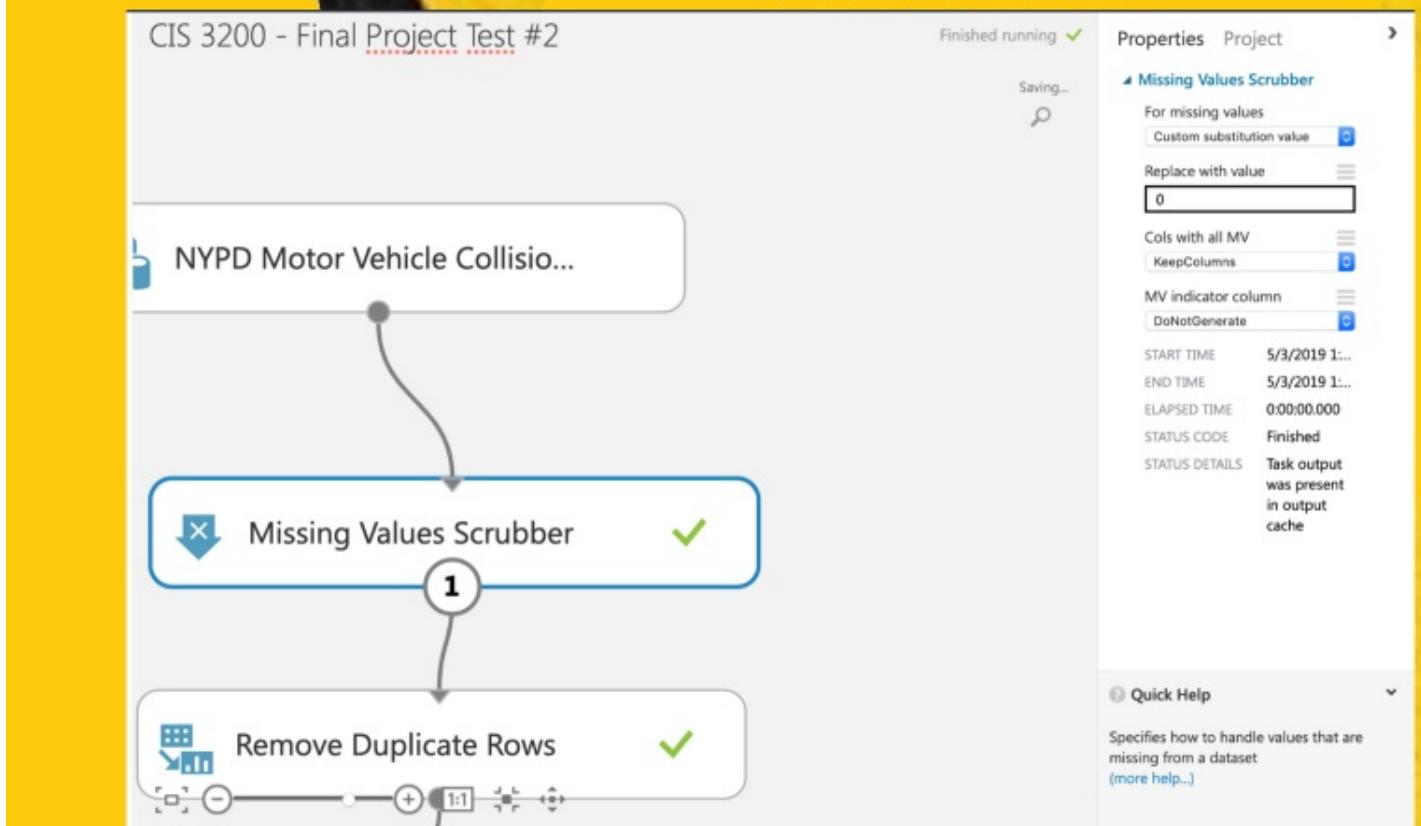
ML Azure Tutorial

Step 1 Upload the Data-Set



Step 2

Step 2 Missing Values Scrubber module



Step 3

Step 3 Remove duplicate rows module

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there's a sidebar with various modules like Saved Datasets, Data Format Conversions, and Machine Learning. The main workspace shows a flowchart titled "CIS 3200 - Final Project Test #2". A "Select columns" module is currently selected, displaying a list of columns: DATE, LOCATION, ON STREET NAME, CROSS STREET NAME, BOROUGH, LATITUDE, LONGITUDE, CONTRIBUTING FACTOR VEHICLE 1, CONTRIBUTING FACTOR VEHICLE 2, VEHICLE TYPE CODE 1, VEHICLE TYPE CODE 2, and CONTRIBUTING FACTOR VEHICLE 3. Below this, a "Select Columns in Dataset" module is shown with a green checkmark. To the right, the "Remove Duplicate Rows" module is open, showing its properties. Under "Selected columns", it lists: Column names: DATE,LOCATION,ON STREET NAME,CROSS STREET NAME,BOROUGH,LATITUDE,LONGITUDE,CONTRIBUTING FACTOR VEHICLE 1,CONTRIBUTING FACTOR VEHICLE 2,VEHICLE TYPE CODE 1,VEHICLE TYPE CODE 2,CONTRIBUTING FACTOR VEHICLE 3. It also includes a "Launch column selector" button and a checkbox for "Retain first duplicate row". The properties pane shows the following details:

- Start Time: 5/3/2019 1:00:00 AM
- End Time: 5/3/2019 1:00:00 AM
- Elapsed Time: 00:00:00.000
- Status Code: Finished
- Status Details: Task output was present in output cache

A "Quick Help" section at the bottom explains that the module removes duplicate rows from a dataset.

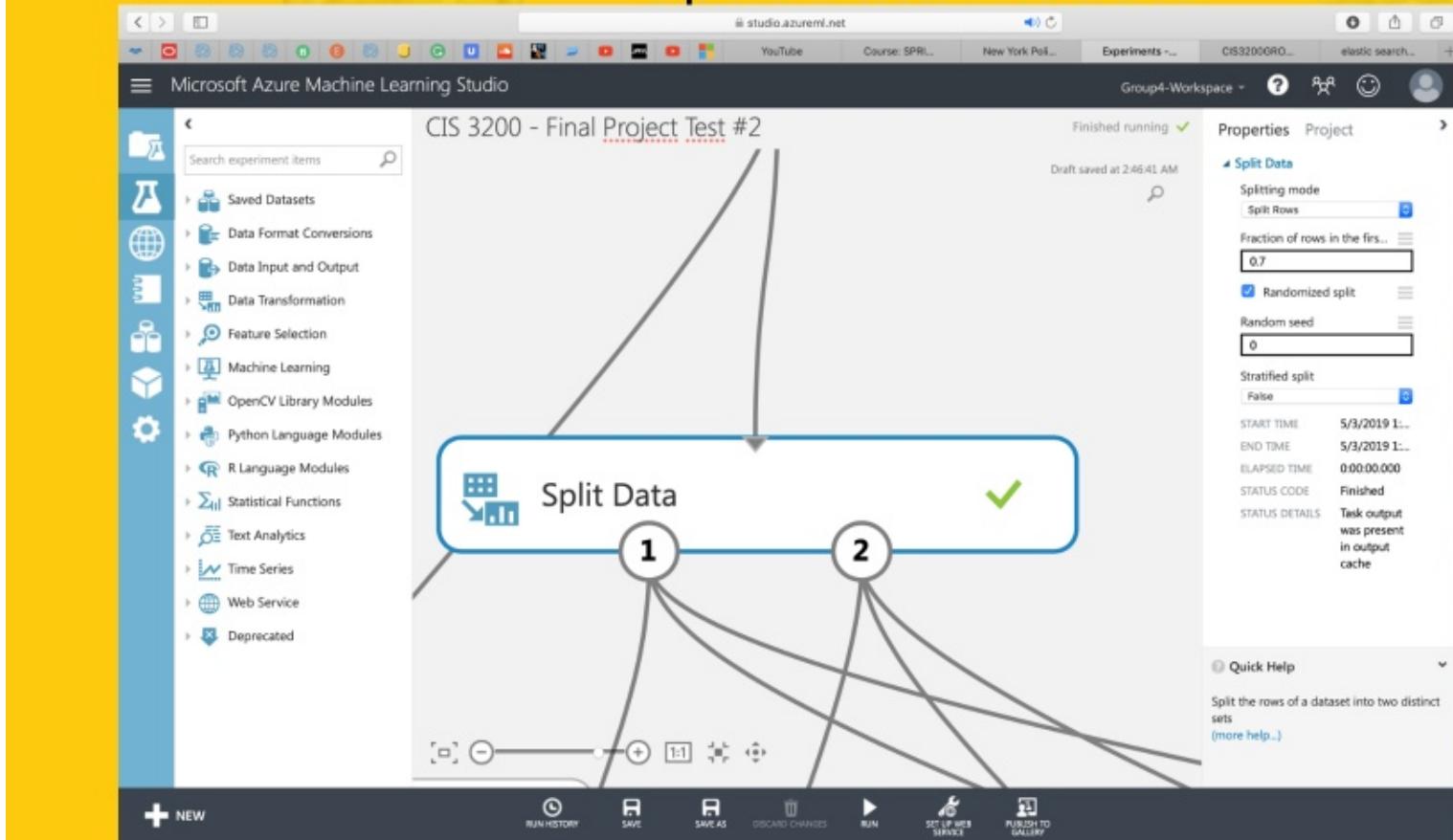
Step 4

Step 4 Select columns in dataset

The screenshot shows the Microsoft Azure Machine Learning Studio interface. The left sidebar contains various project items like Saved Datasets, Data Input and Output, and Feature Selection. The main area is titled 'CIS 3200 - Final Project Test #2' and shows a 'Select columns' dialog. This dialog has tabs for 'BY NAME' and 'WITH RULES', with 'WITH RULES' selected. It includes an 'Exclude' section and a list of columns: ON STREET NAME, CROSS STREET NAME, OFF STREET NAME, CONTRIBUTING FACTOR VEHICLE 2, CONTRIBUTING FACTOR VEHICLE 3, CONTRIBUTING FACTOR VEHICLE 4, CONTRIBUTING FACTOR VEHICLE 5, VEHICLE TYPE CODE 2, VEHICLE TYPE CODE 3, VEHICLE TYPE CODE 4, VEHICLE TYPE CODE 5, LATITUDE, LONGITUDE, and UNIQUE KEY. To the right of the dialog is a properties panel titled 'Select Columns in Dataset' with sections for 'Selected columns', 'START TIME', 'END TIME', 'ELAPSED TIME', 'STATUS CODE', and 'STATUS DETAILS'. The status details note that 'Task output was present in output cache'. At the bottom are buttons for 'RUN', 'SAVE', 'DISCARD CHANGES', and 'PUBLISH TO GALLERY'.

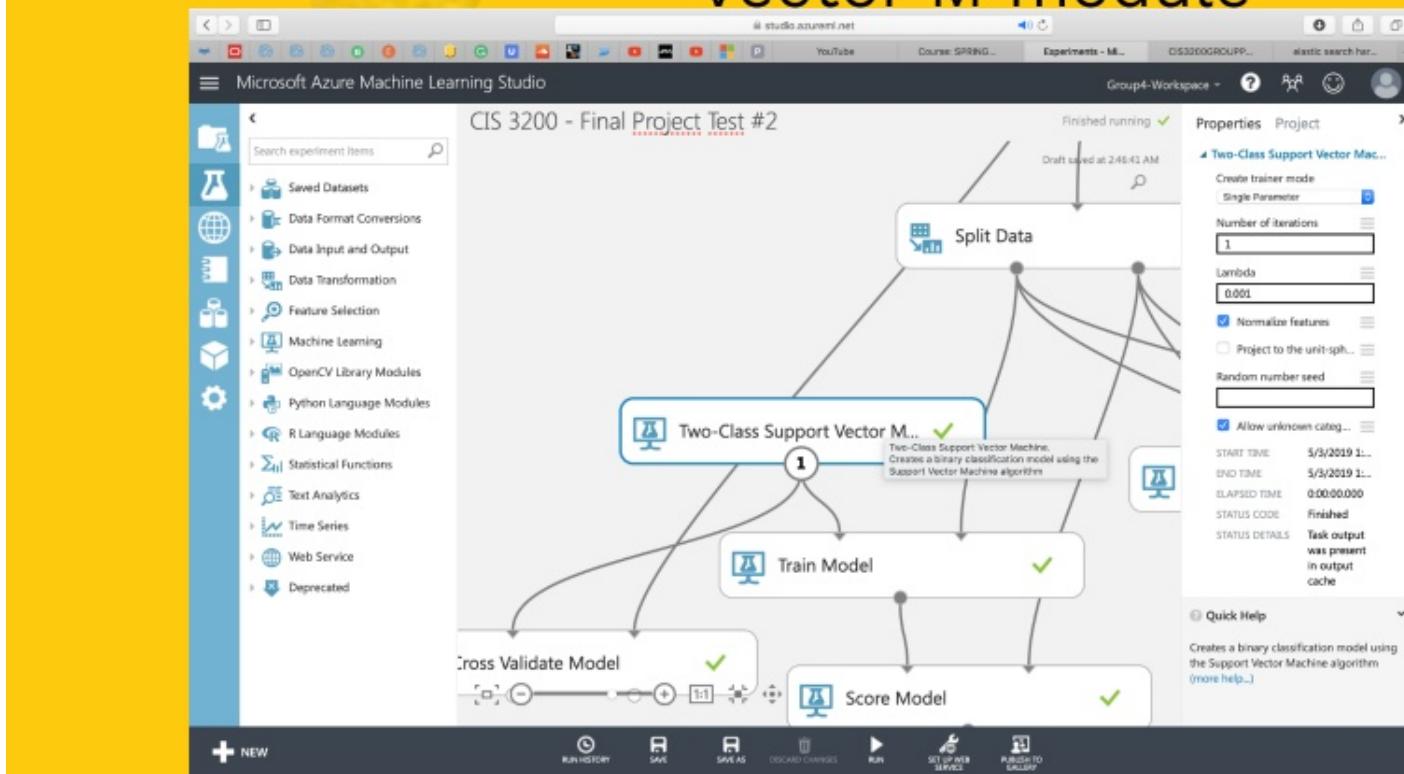
Step 5

Step 5 Split Data module



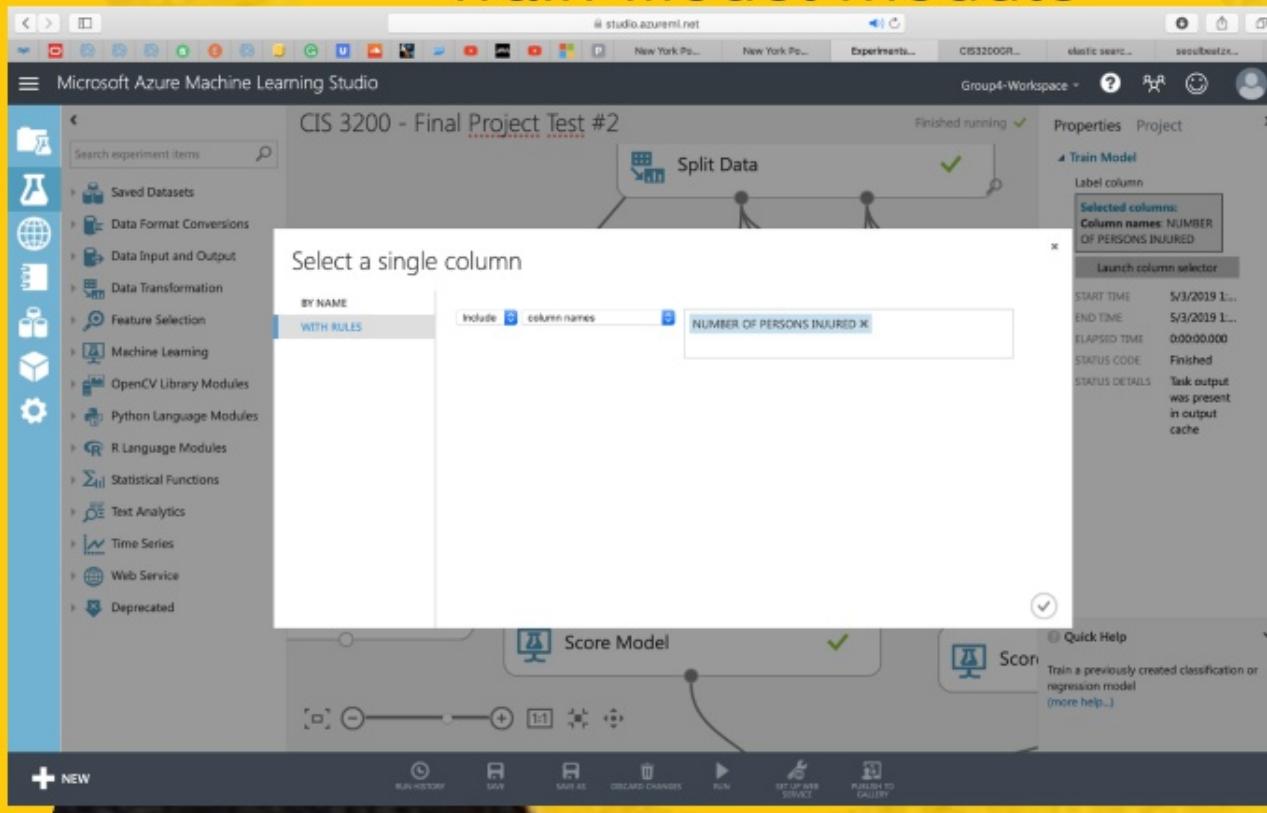
Step 6

Step 6 Two-Class Support Vector M module



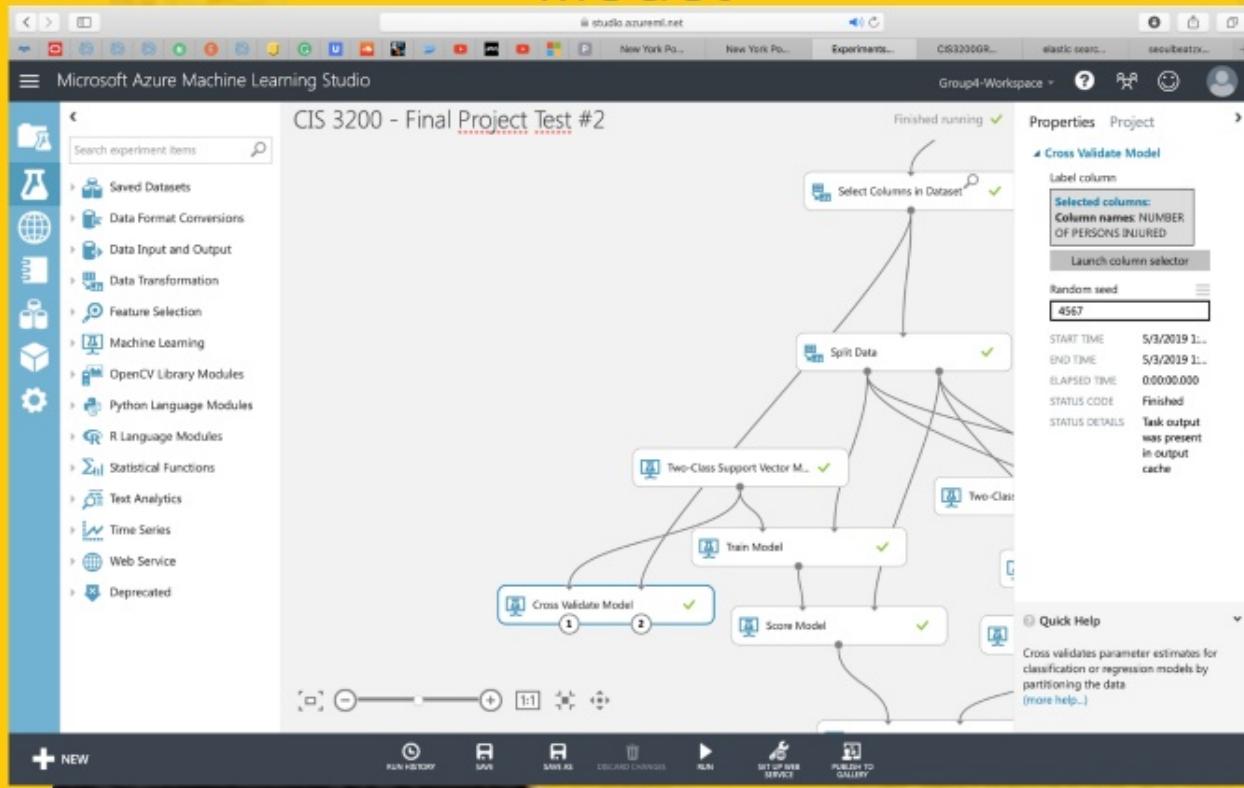
Step 7

Step 7 Train Model module



Step 8

Step 8 Cross Validate Model



Step 9

Step 9

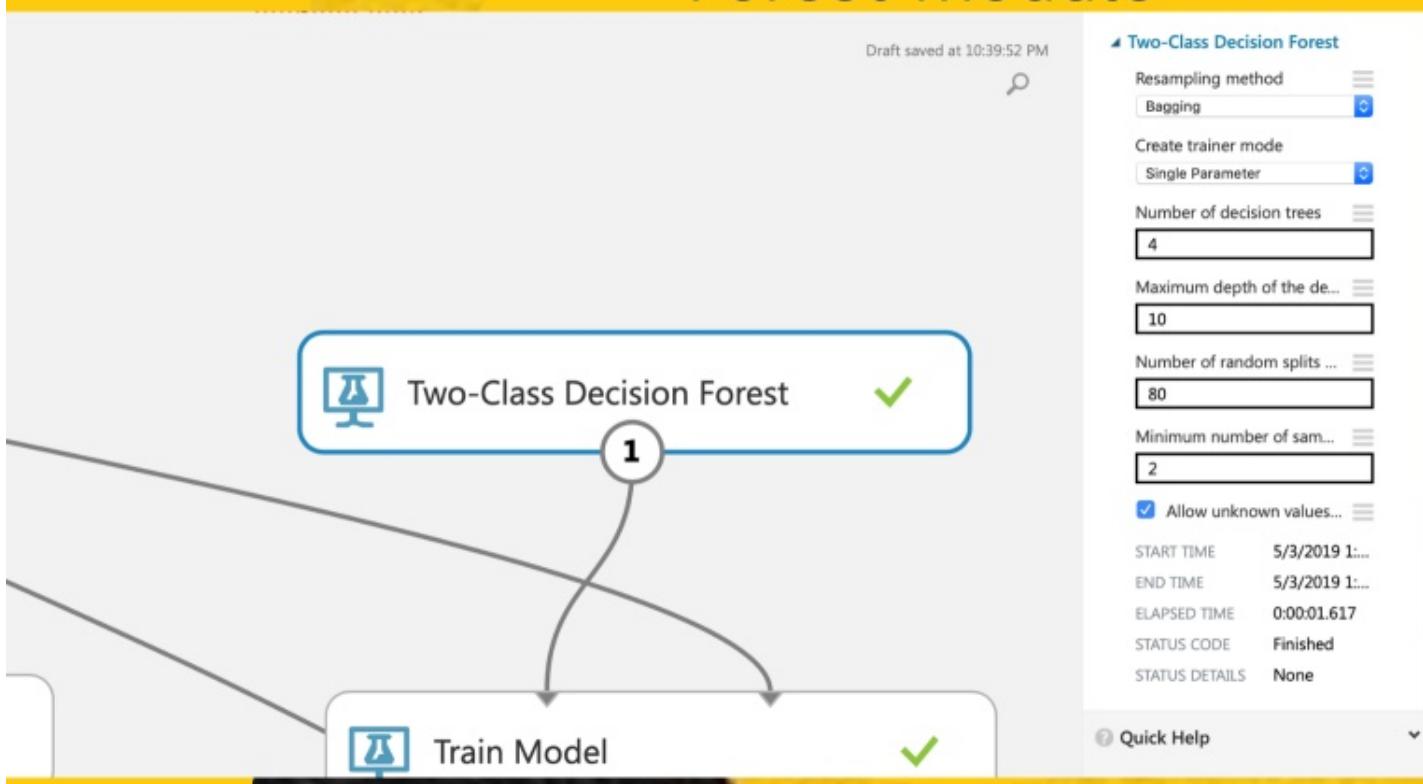
Two class locally deep support module

The screenshot shows a software interface for configuring a machine learning model. On the left, there is a flowchart-like diagram with nodes and arrows. One node is highlighted with a blue border and contains the text "Two-Class Locally-Deep Sup..." followed by a green checkmark. A large number "1" is enclosed in a circle next to this node. Arrows point from this node to another node below it, which also has a green checkmark and contains the text "Train Model". To the right of the diagram is a detailed configuration panel titled "Two-Class Locally-Deep Sup...". It includes the following settings:

- Create trainer mode: Single Parameter
- Depth of the tree: 3
- Lambda W: 0.1
- Lambda Theta: 0.01
- Lambda Theta Prime: 0.01
- Sigmoid sharpness: 1
- Number of iterations: 15000
- Feature normalizer: Min-Max normalizer
- Random number seed: (empty input field)

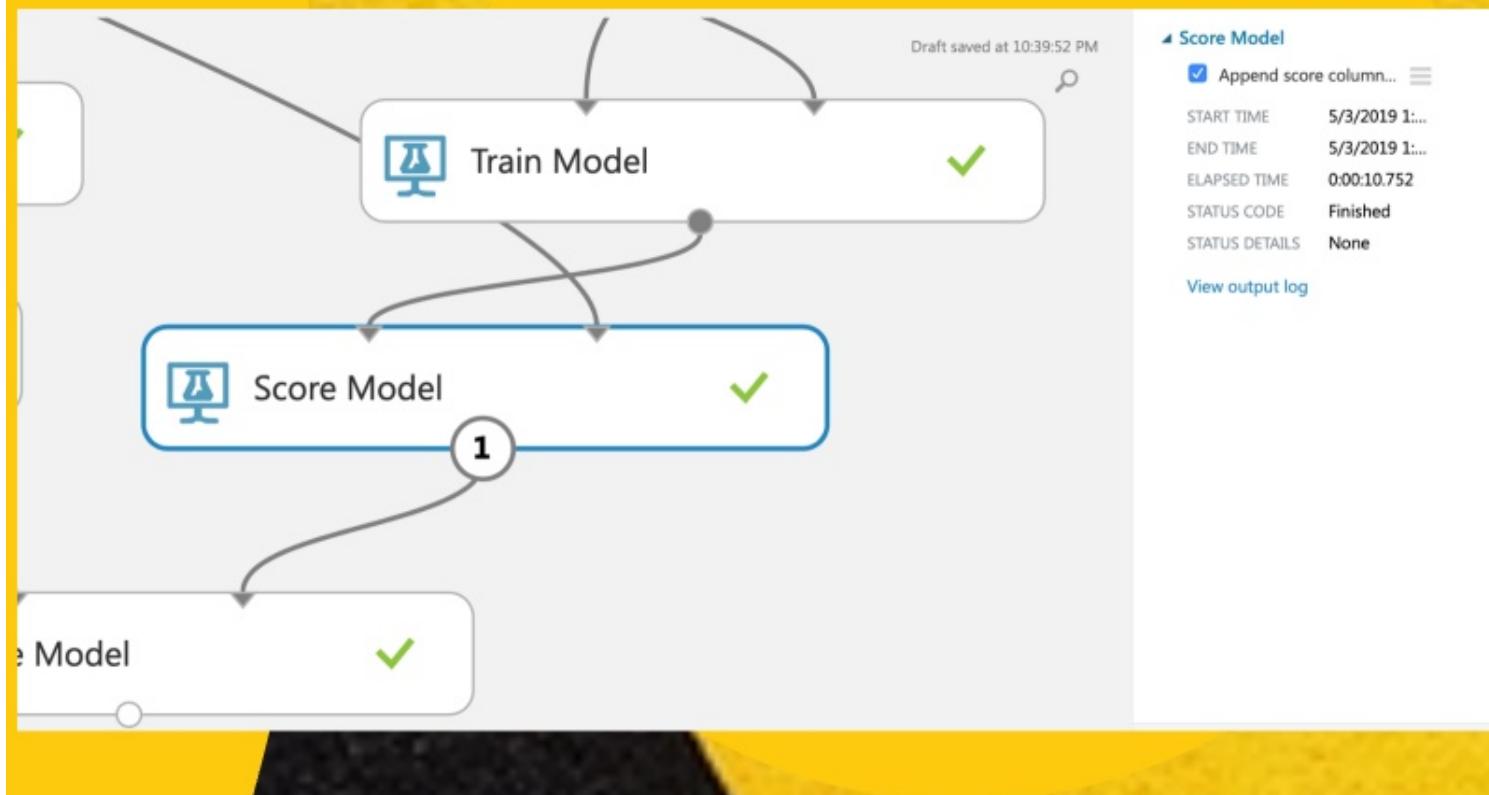
On the far right, the text "Step 10" is visible.

Step 10 Two class Decision Forest module



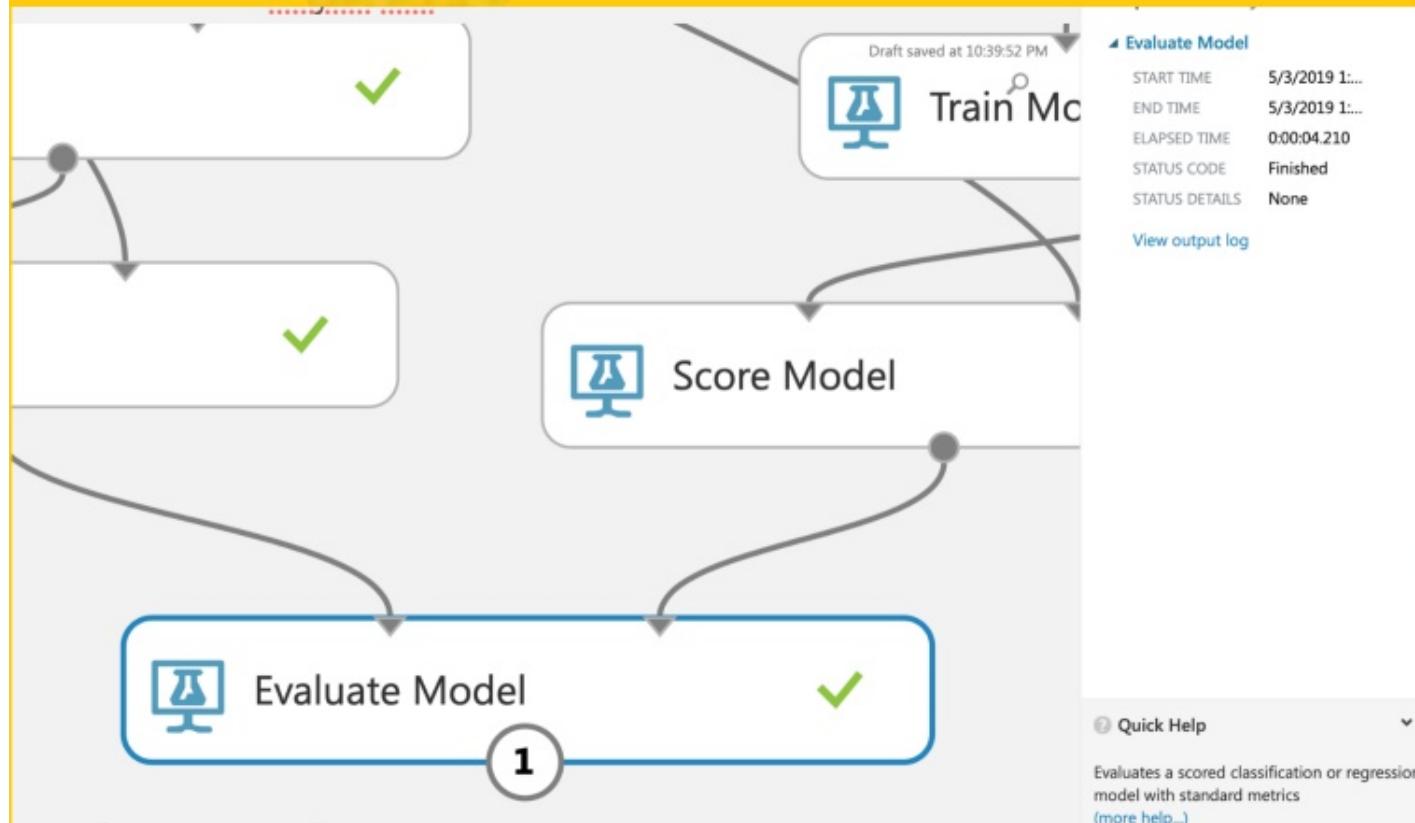
Step 11

Step 11 Score model module



Step 12

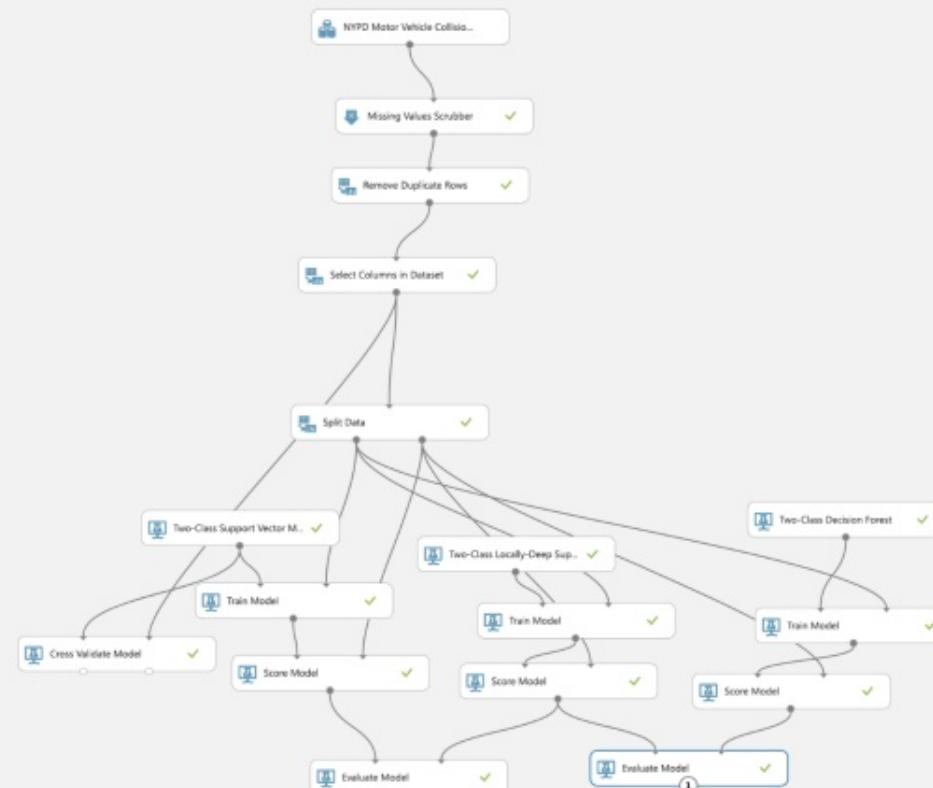
Step 12 Evaluate Model module



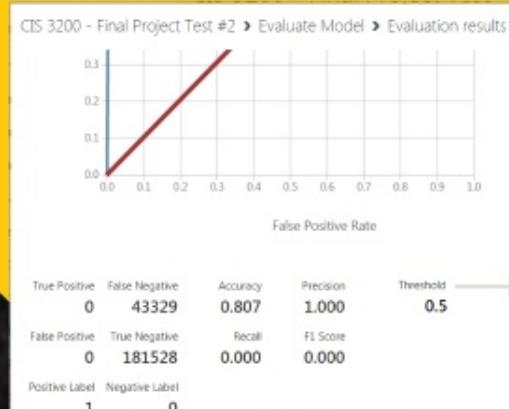
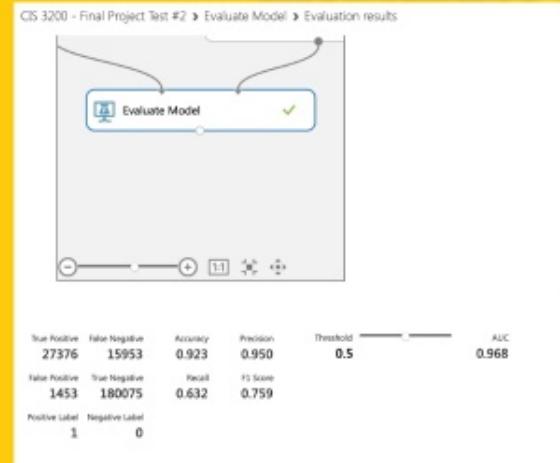
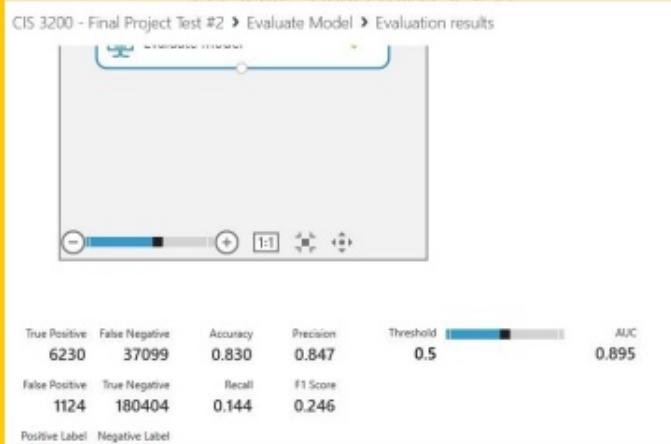
Overview of ML Azure model

Draft saved at 10:39:52 PM

Calculate
Accuracy



Calculate Accuracy



Calculate Accuracy

- **True Positive** - Correctly predicted a person was injured
- **True Negative** - Correctly predicted a person was not injured
- **False Positive** - A person was predicted to be injured but actually was not injured
- **False Negative** - A person was predicted to not be injured but was actually injured

Calculate Accuracy

Precision: Of all persons that were labeled as injured, how many were actually injured

Recall: Of all the persons that were injured, how many were labeled

Calculate
Accuracy

Two Class Support Vector Machine Model

CIS 3200 - Final Project Test #2 > Evaluate Model > Evaluation results



Two Class
Locally-Deep
Support Vector
Machine Model

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
6230	37099	0.830	0.847	0.5	0.895
False Positive	True Negative	Recall	F1 Score		
1124 180404 0.144 0.246					
Positive Label		Negative Label			

Two Class Locally-Deep Support Vector Machine Model

CIS 3200 - Final Project Test #2 > Evaluate Model > Evaluation results



True Positive	False Negative	Accuracy	Precision	Threshold	AUC		
27376	15953	0.923	0.950	0.5	0.968		
False Positive	True Negative	Recall	F1 Score				
1453	180075	0.632	0.759				
Positive Label	Negative Label						
1	0						

Two Class Decision Forest

Two Class Decision Forest

CIS 3200 - Final Project Test #2 ➤ Evaluate Model ➤ Evaluation results



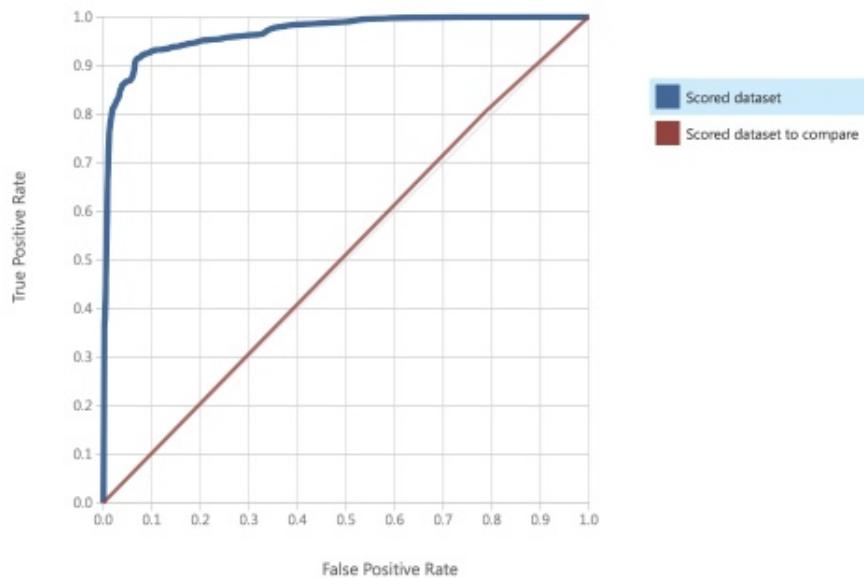
True Positive	False Negative	Accuracy	Precision	Threshold	AUC		
0	43329	0.807	1.000	0.5	0.510		
False Positive	True Negative	Recall	F1 Score				
0	181528	0.000	0.000				
Positive Label	Negative Label						
1	0						

Predictive Result

We used classification as a result for better Area Under the Curve(AUC)

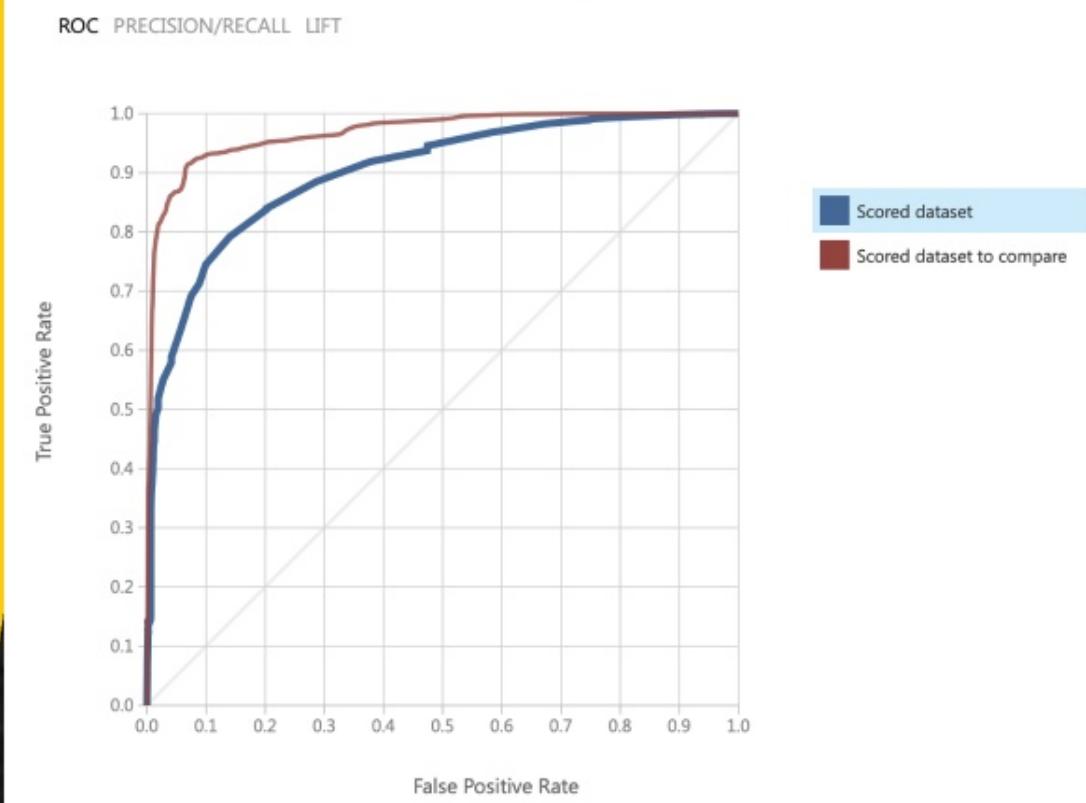
CIS 3200 - Final Project Test #2 > Evaluate Model > Evaluation results

ROC PRECISION/RECALL LIFT



Actual
Result

Actual Result of Data prediction



New York Police Department Injury Report

Presented by
Lucky Trang, Ron Tsang
Stephanie Nguyen, Paul Yi

Intro

Body

CON

REC

Our conclusion based on our experiments

Two-class Locally-Deep Support
Vector Machine Algorithm

- Highest AUC: 96.8%
- Accuracy: 92.3%
- Precision: 95%
- Recall: 63.2%

Top
Reasons for
Vehicle
Accidents

Suggestions

Top Reasons for Injuries and accidents

1. Driver inattention / Distraction
2. Failure to yield right of way
3. Following too closely (tailgating)
4. Disregarding Traffic control



Areas to Avoid

1. Manhattan
 - Times Square
 - Tourists
 - Advertisement
 - Colorful billboards
2. Bronx
3. Brooklyn

New York Police Department Injury Report

Presented by
Lucky Trang, Ron Tsang
Stephanie Nguyen, Paul Yi

Intro

Body

CON

REC

Our Recommendation

Our conclusion to personal injuries report is that we need to emphasize on how much we have to pay attention to our surroundings while driving or walking and possibly avoiding Manhattan, Bronx and Brooklyn red marked areas on a certain predicted time, location and date given by Kibana Geo Point analysis.

References

References

1. Data Source URL

<https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95>

2. Git Hub URL

Continue

More Reference

3. ML Azure Link

<https://studio.azureml.net/Home/ViewWorkspaceCached/10181d8294b7443b9ba7bb75e6bf4fb#Workspaces/Experiments/Experiment/10181d8294b7443b9ba7bb75e6bf4fb.d-f-id.9b8f7e213308472aa03ac999de873033/ViewExperiment>

New York Police Department Injury Report

Presented by
Lucky Trang, Ron Tsang
Stephanie Nguyen, Paul Yi

Intro

Body

CON

REC