CS145: Project 3 | Analysis of NYC Motor Vehicle Collisions

Collaborators:

Please list the names and SUNet IDs of your collaborators below:

- *Cullen O'Connell, cullenoc*
- *Eastan Giebler, eastan*

Project Overview

The main question we are looking to answer is **"What are the contributing factors (ie. locality, accident details, surrounding environment) that influence the severity (injuries/deaths) of motor vehicle collisions in New York City?"**

The main way we will answer this question is by examining the New York Police Department's motor vehicle collisions dataset. This dataset contains entries for individual collisions, detailing things like the time, location, contributing factors and vehicles, number of injured/killed, etc. Severity in our project will be the measure of the number of people injured or killed. Hence, we will examine how each of the factors relates to the severity of collisions.

We will primarily focus on collisions that have well formed metadata (that is, complete listing of contributing factors, vehicles, time, location) in the data analysis portion, but less so in the model as missing metadata might have expresivity. Although we mainly focus on exploring the collisions dataset, we will also explore another New York dataset of interest, the citibike stations dataset. To increase the project depth, we will engineer new factors based on locality of items from this other dataset.

Therefore, to answer our main question, we will also explore the following subquestions.

- How does time influence the collisions?
- Common contributing factors and how they influence severity of a collision.
- How does severity scale with the number of vehicles involved in a collision?
- To what extent does the street(s) impact the severity of the accident?
- Can we correlate collision data with citibike stations (proximity)?

Analysis of Dataset

Dataset summary

We are using 2 datasets for this project:

- `new_york_mv_collisions` (459 MB): Main database containing entries for motor vehicle collisions within New York City since 2012.
- `new_york_citibike` (235 KB): Information about citibike trips and stations in NYC. We will only focus on the `citibike_stations` table

Motor Vehicle Collisions Dataset (459 MB)

The `new_york_mv_collisions` dataset includes a single table that's 459 MB large containing rows that each contain the details of a specific motor vehicle collision in NYC. It contains information about individual vehicle collision incidents since 2012, detailing things such as the timestamp, borough, street names, contributing factors for each vehicle, number of individuals injured or killed, vehicle type, location, etc.

New York Citibike Dataset (235 KB)

The `new_york_citibike` dataset contains two tables, `citibike_stations` and `citibike_trips`. The table of interest, `citibike_stations`, is a 235 KB table detailing each of the current 1799 citibike stations in NYC, describing their names and cross streets, latitude, longitude, and bike statuses. The main keys we are using to relate the Citibike dataset to the collisions dataset are the street names and latitude/longitude.

Data Exploration

Dependencies and Setup

In [ ]:
```
# Run this cell to authenticate yourself to BigQuery
from google.colab import auth
auth.authenticate_user()
project_id = "eg-cs145-project-22"
```

In [ ]:
```
# Run this cell to authenticate yourself to BigQuery
from google.colab import auth
auth.authenticate_user()
project_id = "cs145nycfinal"
```

In [ ]:
```
# Initialize BigQuery client
from google.cloud import bigquery
client = bigquery.Client(project=project_id)
```

In [ ]:
```
# Installations
!pip install -U plotly
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: plotly in /usr/local/lib/python3.8/dist-packages (5.5.0)
Collecting plotly
  Downloading plotly-5.11.0-py2.py3-none-any.whl (15.3 MB)
     |████████████████████████████████| 15.3 MB 974 kB/s
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.8/dist-packages (from plotly) (8.1.0)
Installing collected packages: plotly
  Attempting uninstall: plotly
    Found existing installation: plotly 5.5.0
    Uninstalling plotly-5.5.0:
      Successfully uninstalled plotly-5.5.0
Successfully installed plotly-5.11.0
```

In [ ]:
```
#Import various python packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Preview Data

This section just gives us a very basic preview of the underlying data in our tables, for easy future reference.

Preview the `nypd_mv_collisions` dataset

In [ ]:
```
%%bigquery --project $project_id

SELECT * FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions` LIMIT 5
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[84]:

	borough	contributing_factor_vehicle_1	contributing_factor_vehicle_2	contributing_factor_vehicle_3	contributing_factor_vehicle_4	contributing_factor_vehicle_5
0	None	Passing or Lane Usage Improper	Unspecified	None	None	None
1	None	Unspecified	Unspecified	None	None	None
2	None	Unspecified	Unspecified	Unspecified	None	None
3	None	Backing Unsafely	Unspecified	None	None	None
4	None	Following Too Closely	Unspecified	None	None	None

5 rows × 28 columns

Preview the `citibike_stations` dataset

In [ ]:
```
%%bigquery --project $project_id

SELECT * FROM `bigquery-public-data.new_york_citibike.citibike_stations` LIMIT 5
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[85]:

	station_id	name	short_name	latitude	longitude	region_id	rental_methods	capacity	eightd_has_key_dispenser	num_bikes_available	num_bikes_disa
0	305	E 58 St & 3 Ave	6762.02	40.760958	-73.967245	71	CREDITCARD, KEY	0	False	0	
1	368	Carmine St & 6 Ave	5763.03	40.730386	-74.002150	71	CREDITCARD, KEY	0	False	0	
2	403	E 2 St & 2 Ave	5593.02	40.725029	-73.990697	71	CREDITCARD, KEY	0	False	0	
3	415	Pearl St & Hanover Square	4993.02	40.704718	-74.009260	71	CREDITCARD, KEY	0	False	0	
4	3066	Tompkins Ave & Hopkins St	4850.04	40.699576	-73.947084	71	CREDITCARD, KEY	0	False	0	

Collisions Dataset Analysis

This section focuses on analysing the NYC collisions dataset and making observations about the features. We will first analyze the dataset with respect to the timestamp feature to get an idea for how factors such as time of day or year affect the collisions.

Hourly Analysis

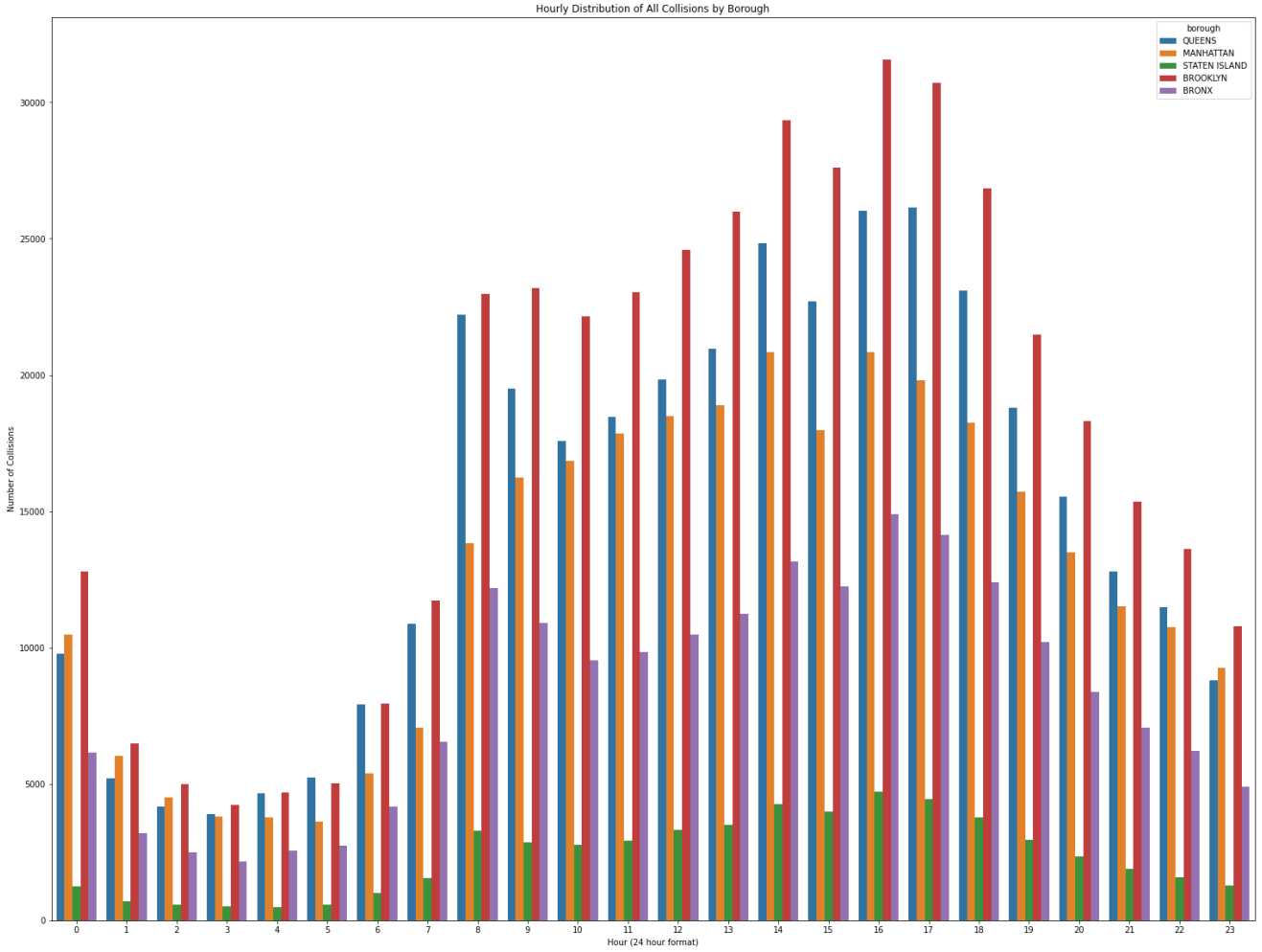
We group the collisions by hour to see how time of day impacts collisions.

In [ ]:
```
%%bigquery hourly --project $project_id

SELECT EXTRACT(HOUR FROM timestamp) as hour, COUNT(*) as num_crash, borough
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
WHERE borough IS NOT NULL
GROUP BY EXTRACT(HOUR FROM timestamp), borough
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

In [ ]:
```
plt.figure(figsize = (26, 20))

ax = sns.barplot(data=hourly, x='hour', y='num_crash', hue='borough')
ax.set(title="Hourly Distribution of All Collisions by Borough", xlabel="Hour (24 hour format)", ylabel="Number of Collisions")
```

Out[107]:
```
[Text(0, 0.5, 'Number of Collisions'),
 Text(0.5, 0, 'Hour (24 hour format)'),
 Text(0.5, 1.0, 'Hourly Distribution of All Collisions by Borough')]
```



This visualization helps us to see when and where the colllisions are occuring in NYC throughout a day. From this analysis we can see that collisions begin to spike up at 8am each day. Also, most the accidents are occuring from 2pm-6pm.

Conceptually, this makes sense as we can assume that NYC traffic and the number of drivers is the worst at morning rush hour and when people are getting off work (8am and 2-6pm respectively).

We see all the boroughs follow relatively similar patterns throughout the day. However, points along the histogram where the changes in the height of the different regions' bars diverge suggest subtle traffic pattern differences across boroughs, possibly influenced by differences in land use (commercial, retail, residential, etc) which lends itself to traffic at different times. For instance, at 8 AM, traffic in the Bronx is at its peak and declines for a few hours while traffic in Manhattan, known for its businesses, and Queens continues to mount through the workday. These different patterns could influence severity of collisions.

Monthly Analysis

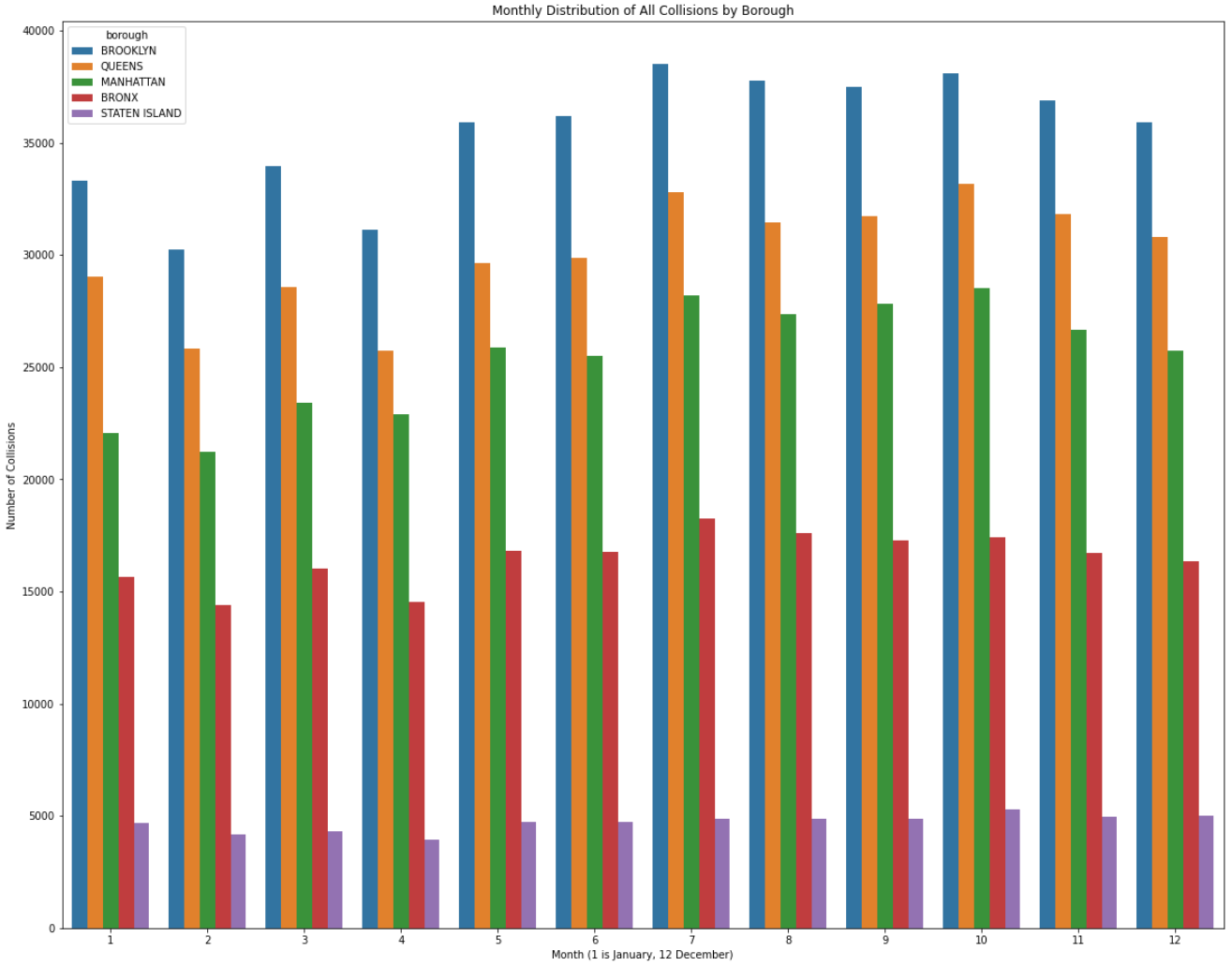
We group the collisions by month to see which times of year are the worst for collisions.

In [ ]:
```
%%bigquery monthly --project $project_id

SELECT EXTRACT(MONTH FROM timestamp) as month, COUNT(*) as num_crash, borough
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
WHERE borough IS NOT NULL
GROUP BY EXTRACT(MONTH FROM timestamp), borough
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

In [ ]:
```
plt.figure(figsize = (20, 16))


ax = sns.barplot(data=monthly, x='month', y='num_crash', hue='borough')
ax.set(title="Monthly Distribution of All Collisions by Borough", xlabel="Month (1 is January, 12 December)", ylabel="Number of C
```

Out[110]:
```
[Text(0, 0.5, 'Number of Collisions'),
 Text(0.5, 0, 'Month (1 is January, 12 December)'),
 Text(0.5, 1.0, 'Monthly Distribution of All Collisions by Borough')]
```



This visualization reveals that collisions are not particularly affected by the time of year. There is a small uptick in the number of collisions during the summer months. This could be caused by a larger number of people visiting New York during the summer months for vacation, travel, or work. As we are not incorporating general trip volume data, we cannot definitively say how the rate of collisions per trip taken varies by season.

This seems counterintuitive as one might assume the winter and cold weather causes more dangerous road conditions, which may lead to more collisions during the winter months.

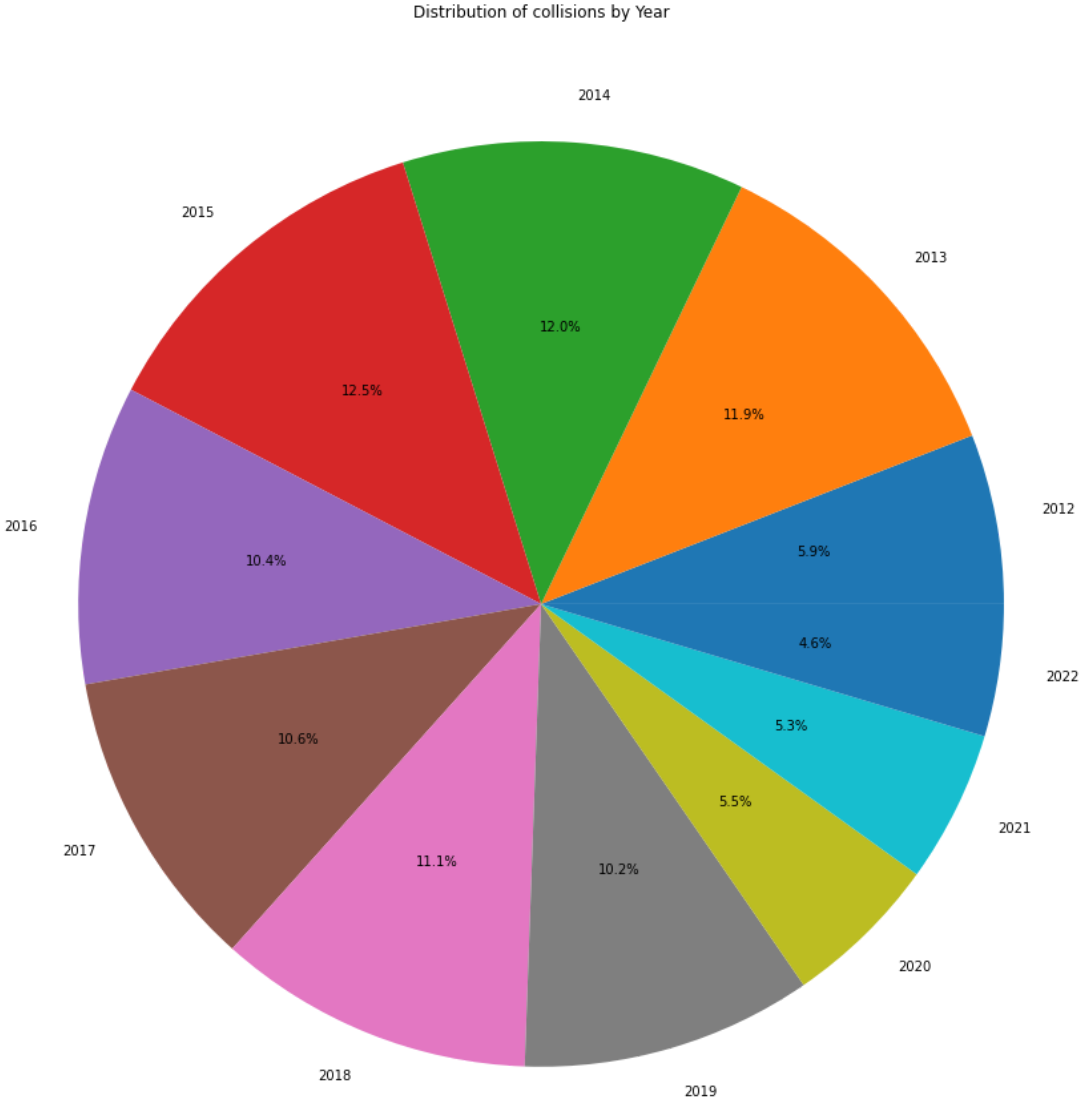
Yearly distribution

We group the collisions by year to see the make up of the collisions dataset. Furthermore, we only take the entries that have boroughs and valid geolocations. This is to get a better idea of the actual data that the model will be able to use.

In [ ]:
```
%%bigquery yearly --project $project_id

SELECT EXTRACT(YEAR FROM timestamp) as year, COUNT(*) as num_collisions
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
WHERE borough IS NOT NULL AND latitude != 0 AND longitude !=0
GROUP BY EXTRACT(YEAR FROM timestamp)
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

In [ ]:
```
plt.figure(figsize = (16, 16))

total = 1.0*np.sum(yearly.num_collisions)

yearly.num_collisions = yearly.num_collisions / total
yearly = yearly.sort_values('year')

colors = sns.color_palette('bright')[0:yearly.size]

plt.pie(yearly.num_collisions, labels=yearly.year, autopct='%1.1f%%')
plt.title("Distribution of collisions by Year")
plt.show()
```



Crash Severity Influences

Since we are trying to find out what influences the severity of collisions in NYC, we will now do a more granular analysis of the correlations between the various attributes of the collisions dataset and the number of people injured/killed.

Firstly, we will group the collisions by street names to see which streets are have the most casualties from collisions.

In [ ]:
```
%%bigquery --project $project_id

SELECT street, SUM(number_of_persons_killed + number_of_persons_injured) as casualties
FROM (
  SELECT on_street_name as street, number_of_persons_killed, number_of_persons_injured FROM `bigquery-public-data.new_york_mv_col
  UNION ALL
  SELECT off_street_name as street, number_of_persons_killed, number_of_persons_injured FROM `bigquery-public-data.new_york_mv_co
  UNION ALL
  SELECT cross_street_name as street, number_of_persons_killed, number_of_persons_injured FROM `bigquery-public-data.new_york_mv_
)
WHERE street IS NOT NULL
GROUP BY street
HAVING COUNT(*) > 5  # min number of instances
ORDER BY casualties DESC
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[92]:

	street	casualties
0	BROADWAY	7087
1	ATLANTIC AVENUE	5875
2	BELT PARKWAY	5864
3	LINDEN BOULEVARD	5423
4	3 AVENUE	5096
...	...	...
16041	100 OVERLOOK TERRACE	0
16042	132-50 METROPOLITAN AVENUE	0
16043	859 HENDRIX STREET	0
16044	700 PACIFIC STREET	0
16045	hugh grant circle	0

16046 rows × 2 columns

As we can see, the streets in NYC with the most collision casualties are notable main streets such as Broadway spanning west Manhattan, 3rd Ave on east Manhattan, and Atlantic Ave in Brooklyn. This makes intuitive sense because each of these main streets are quite long and span through busy, central areas of their respective boroughs.

Thus, it makes sense that they would have more casualties as they have many more intersections with other cross streets throughout the city. Furthermore, at each of these intersections, people are likely trying to get onto or use the main street, and so the increased throughput and traffic on the main streets leads to more casualties.

Furthermore, we hypothesize that being on a main street or a street of interesting might be correlated with an increase in probability that the collision will involve casualties.

Casualties on a Street-by-street Basis

Hence, we will investigate the streets with the highest rate of casualty for pedestrians vs cyclists vs motorists (the average number of casualties per collision in relation to that street). It's important to note that we enforced a minimum number of collision instances for each street to filter out outlier streets that have small numbers of reports yet high casualties. These analyses produced particularly interesting results.

In [ ]:
```
%%bigquery --project $project_id

SELECT street, AVG(number_of_cyclist_injured + number_of_cyclist_killed) as casualty_rate
FROM (
  SELECT on_street_name as street, number_of_cyclist_killed, number_of_cyclist_injured FROM `bigquery-public-data.new_york_mv_col
  UNION ALL
  SELECT off_street_name as street, number_of_cyclist_killed, number_of_cyclist_injured FROM `bigquery-public-data.new_york_mv_co
  UNION ALL
  SELECT cross_street_name as street, number_of_cyclist_killed, number_of_cyclist_injured FROM `bigquery-public-data.new_york_mv_
)
WHERE street IS NOT NULL
GROUP BY street
HAVING COUNT(*) > 25  # min number of instances
ORDER BY casualty_rate DESC
LIMIT 20
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[39]:

	street	casualty_rate
0	97 EAST DRIVE	0.896552
1	101 EAST DRIVE	0.847458
2	101 WEST DRIVE	0.829268
3	79 WEST DRIVE	0.763158
4	85 EAST DRIVE	0.689655
5	71 EAST DRIVE	0.661017
6	WEST DRIVE	0.571429
7	EAST DRIVE	0.507042
8	WEST DRIVE	0.491803
9	CENTER DRIVE	0.464286
10	EAST DRIVE	0.453333
11	TRANSVERSE ROAD NUMBER FOUR	0.377358
12	BERKELEY PLACE	0.235294
13	WHIPPLE STREET	0.225806
14	TRANSVERSE ROAD NUMBER TWO	0.210526
15	WEST 120 STREET	0.202899
16	ARION PLACE	0.189189
17	SOUTH 4 STREET	0.187166
18	QUEENSBORO BRIDGE LOWER	0.183673
19	37 DRIVE	0.178571

The results of the bicyclist casualty rate query reveal an intriguing pattern. The top 11 streets where the bicyclist casualty rate (casualties/collison) is the highest are all related to East, West, and Central Drive. Interestingly, the first 6 streets all have full addresses. Looking these locations up on Google Maps, one can see that all these streets are in Brooklyn's Prospect Park. Similarly, streets 6-10 all correspond to Central Park!

There is a rather straightforward explanation for these findings. Prospect and Central Park are two massive parks in the middle of a busy metropolis. More importantly, these parks are popular routes for bicyclists. Hence, whenever these bike routes intersect with drivable roads, they create the perfect storm for motor vehicles to collide with bicyclists.

In [ ]:
```
%%bigquery --project $project_id

SELECT street, AVG(number_of_pedestrians_injured + number_of_pedestrians_killed) as casualty_rate
FROM (
  SELECT on_street_name as street, number_of_pedestrians_killed, number_of_pedestrians_injured FROM `bigquery-public-data.new_yor
  UNION ALL
  SELECT off_street_name as street, number_of_pedestrians_killed, number_of_pedestrians_injured FROM `bigquery-public-data.new_yo
  UNION ALL
  SELECT cross_street_name as street, number_of_pedestrians_killed, number_of_pedestrians_injured FROM `bigquery-public-data.new_
)
WHERE street IS NOT NULL
GROUP BY street
HAVING COUNT(*) > 25  # min number of instances
ORDER BY casualty_rate DESC
LIMIT 20
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[38]:

	street	casualty_rate
0		0.500000
1	EAMES PLACE	0.482759
2	TILDEN STREET	0.282051
3	THWAITES PLACE	0.279070
4	EINSTEIN LOOP	0.269231
5	MOHEGAN AVENUE	0.269231
6	NATIONAL STREET	0.268657
7	PARKCHESTER ROAD	0.266667
8	WEST DRIVE	0.255639
9	BRIGHTON 1 STREET	0.250000
10	FINDLAY AVENUE	0.250000
11	BRITTON AVENUE	0.243243
12	WEST 195 STREET	0.242857
13	HAMPTON STREET	0.242647
14	91 PLACE	0.241758
15	JUSTICE AVENUE	0.240741
16	91 PLACE	0.238095
17	ARLINGTON PLACE	0.236842
18	HAMPTON STREET	0.236842
19	ASCH LOOP	0.234375

The pattern around pedestrian casualties is much less straightforward. However, if we look at each of the top 1-7 streets with the highest pedestrian casualty rates, we begin to see some commonalities between these seemingly random streets around New York. Each of these streets with high pedestrian casualty rates are very short, usually no longer than 4 blocks. Despite being shorter streets, these streets share the characteristic that they intersect with many high-traffic main streets. Additionally, some of the intersections are multistreet intersections involving more than 2 streets.

Hence, we hypothesize that these short streets are high risk for pedestrians because they intersect busy streets, often in awkward intersections. Thus, this leads to the collisions being more likely to involve pedestrians who are walking through these high risk areas.

In [ ]:
```
%%bigquery --project $project_id

SELECT street, AVG(number_of_motorist_injured + number_of_motorist_killed) as casualty_rate
FROM (
  SELECT on_street_name as street, number_of_motorist_killed, number_of_motorist_injured FROM `bigquery-public-data.new_york_mv_c
  UNION ALL
  SELECT off_street_name as street, number_of_motorist_killed, number_of_motorist_injured FROM `bigquery-public-data.new_york_mv_
  UNION ALL
  SELECT cross_street_name as street, number_of_motorist_killed, number_of_motorist_injured FROM `bigquery-public-data.new_york_m
)
WHERE street IS NOT NULL
GROUP BY street
HAVING COUNT(*) > 50  # min number of instances
ORDER BY casualty_rate DESC
LIMIT 20
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[37]:

	street	casualty_rate
0	153 LANE	0.830189
1	Whitestone Expy	0.818182
2	PROSPECT EXPRESSWAY RAMP	0.818182
3	PITMAN AVENUE	0.814815
4	BRONX RIVER PARKWAY RAMP	0.794872
5	PROSPECT EXPRESSWAY EAST	0.773196
6	MARMION AVENUE	0.768421
7	MICKLE AVENUE	0.762712
8	DEREIMER AVENUE	0.746269
9	JACKIE ROBINSON PKWY	0.728736
10	VAN WYCK EXPWY	0.710555
11	VERRAZANO BRIDGE UPPER	0.707006
12	BELT PARKWAY	0.696882
13	EAST 230 STREET	0.696429
14	231 STREET	0.693642
15	ATKINS AVENUE	0.691630
16	NORTH CHANNEL BRIDGE	0.689655
17	MANGIN AVENUE	0.688525
18	132 AVENUE	0.674419
19	GOWANUS EXPY (BQE)	0.674157

Here, the pattern for motorist casualties is more straightforward to see. The street with the highest motorist casualty rate, 153rd lane, happens to be a blind intersection with the of ramp of the Belt Parkway and JFK Expressway. Similarly, the other streets at the top of the list are also other expressways, highway ramps, or cross streets of highway ramps.

We see that the most dangerous streets for motorists are the highways and the off and on ramps. We hypothesize that the ramps are even more dangerous than the highways themselves. Our reasoning is that the ramps are where motorists are transitioning from the high speed expressway onto urban roads, and so the large speed differential between the two leads to collisions that cause more casualties.

Motor Vehicle Types

Next, we investigate how the types of motor vehicles might correlate with casualties.

The first thing we will investigate is which vehicles cause the most severe collisions. That is, we are examining the average number of casualties in a collision when there are is a non-zero number of casualties.

In [ ]:
```
%%bigquery --project $project_id
SELECT vehicle_type_code1, AVG(number_of_persons_killed + number_of_persons_injured) as severity
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
WHERE number_of_persons_killed > 0 OR number_of_persons_injured > 0
GROUP BY vehicle_type_code1
HAVING COUNT(*) > 30  # adjustable
ORDER BY severity DESC
LIMIT 20
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[57]:

	vehicle_type_code1	severity
0	Chassis Cab	1.759036
1	AMBULANCE	1.674121
2	3-Door	1.632653
3	FIRE TRUCK	1.602041
4	Bus	1.588187
5	BUS	1.579339
6	Ambulance	1.569364
7	School Bus	1.500000
8	Tractor Truck Gasoline	1.461538
9	Tanker	1.455446
10	van	1.451220
11	4 dr sedan	1.430556
12	2 dr sedan	1.412731
13	Carry All	1.406091
14	PASSENGER VEHICLE	1.394063
15	SPORT UTILITY / STATION WAGON	1.391241
16	Sedan	1.386749
17	Station Wagon/Sport Utility Vehicle	1.383438
18	Refrigerated Van	1.382979
19	Flat Bed	1.356467

As we can see, the types of vehicles that cause the most severe collision casualties are the larger motor vehicles. This would make sense as the larger vehicles are bulkier and have more mass so they are likely to cause more severe damage to people.

Interesting, ambulances and fire trucks are among the most injurious vehicles involved in NYC collisions. This perhaps makes sense due to the urgency of the emergency vehicles when they drive around NYC. As they must drive more aggressively to save lives at the scene, it seems that emergency vehicles are more often involved in injurious collisions.

Number of Vehicles vs. Casualties

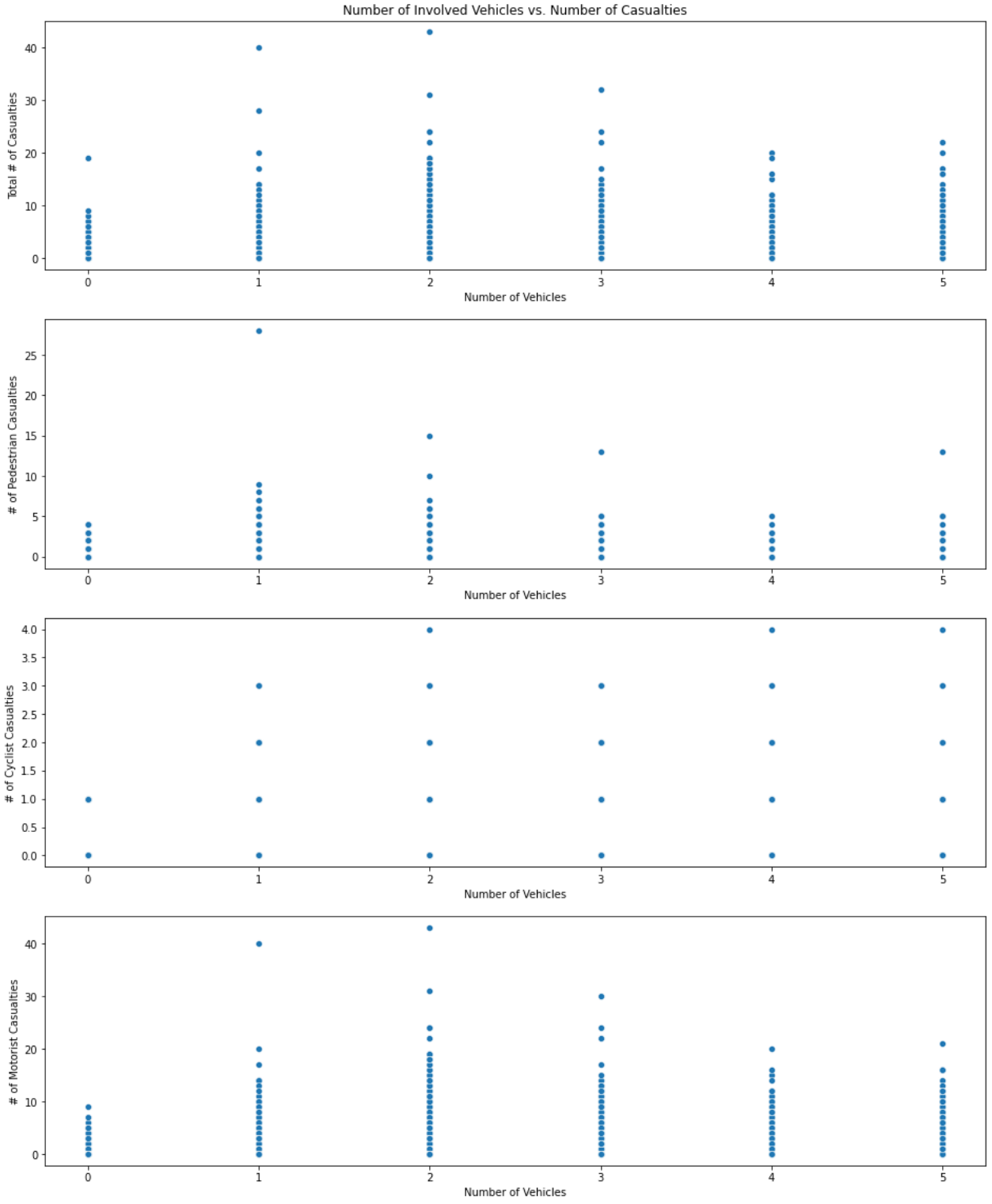
Next, we will examine how the number of vehicles involved in a collision correlates with the numbers of casualties for each of the measured groups.

In [ ]:
```
%%bigquery collision_breakdowns --project $project_id

SELECT number_of_persons_injured + number_of_persons_killed as total_casualties,
number_of_cyclist_injured + number_of_cyclist_killed as cyclist_casualties,
number_of_motorist_injured + number_of_motorist_killed as motorist_casualties,
number_of_pedestrians_injured + number_of_pedestrians_killed as ped_casualties,
CASE WHEN vehicle_type_code1 IS NOT NULL THEN 1 ELSE 0 END +
CASE WHEN vehicle_type_code2 IS NOT NULL THEN 1 ELSE 0 END +
CASE WHEN vehicle_type_code_3 IS NOT NULL THEN 1 ELSE 0 END +
CASE WHEN vehicle_type_code_4 IS NOT NULL THEN 1 ELSE 0 END +
CASE WHEN vehicle_type_code_5 IS NOT NULL THEN 1 ELSE 0 END AS num_vehicles
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

In [ ]:
```
fig, ax = plt.subplots(4, 1, figsize=(16,20))

sns.scatterplot(data=collision_breakdowns, x='num_vehicles', y='total_casualties', ax=ax[0]).set(title='Number of Involved Vehicl
sns.scatterplot(data=collision_breakdowns, x='num_vehicles', y='ped_casualties', ax=ax[1]).set(xlabel="Number of Vehicles", ylabe
sns.scatterplot(data=collision_breakdowns, x='num_vehicles', y='cyclist_casualties', ax=ax[2]).set(xlabel="Number of Vehicles", y
sns.scatterplot(data=collision_breakdowns, x='num_vehicles', y='motorist_casualties', ax=ax[3]).set(xlabel="Number of Vehicles",
```

Out[82]:
```
[Text(0, 0.5, '# of Motorist Casualties'), Text(0.5, 0, 'Number of Vehicles')]
```



As we can see from the scatterplots, the distributions for each type of party involved in the collision looks relatively similar. The exception is with the bicyclist casualties with respect to scale. The maximum number of bicyclists involved in a give collision is 4, unlike the other person groups.

The number of casualties involved in each collision tends to trend upward as the number of motor vehicles increases from 0 to 2. However, when the number of listed vehicles is 2+, the number of casualties no longer increases linearly with the number of vehicles.

These distributions gives us a high level proxy for the size of a collision. That is, collisions involving more casualties and more vehicles can be interpreted as being more severe. Correlating the number of vehicles involved with the number of casualties is a useful feature to investigate further.

So, we will further bucket the total casualties alongside the number of vehicles involved to get an idea for the distribution of collisions based on their number of casualties.

In [ ]:
```
%%bigquery vehicles --project $project_id

SELECT num_vehicles, num_casualties, COUNT(*) as num_instances
FROM (
  SELECT number_of_persons_injured + number_of_persons_killed AS num_casualties,
  CASE WHEN vehicle_type_code1 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code2 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_3 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_4 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_5 IS NOT NULL THEN 1 ELSE 0 END AS num_vehicles
  FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
)
GROUP BY num_vehicles, num_casualties
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

In [ ]:
```
# plt.figure(figsize = (20, 20))

# ax = sns.scatterplot(data=vehicles, x='num_vehicles', y='num_casualties', size=np.log(vehicles.num_instances), sizes=(20,5000))
# ax.set(title='Correlation between Number of Involved Vehicles and Number of people injured', xlabel="Number of Vehicles", ylabe
fig = px.scatter(vehicles, x="num_vehicles", y="num_casualties",
             size=np.log(vehicles["num_instances"].to_list()), size_max=40,
                    height=1000,
                    labels={
                              "num_vehicles": "Number of Vehicles",
                              "num_casualties": "Number of Casualties"
                       },
                       title="Correlation between Number of Involved Vehicles and Casualty distribution")

fig.show()
```

Notice that the size of the bubbles are scaled logarithmically to make the graph more readable as most collisions involve small numbers of casualties. However, this data provides more granular analysis of the relation between the number of vehicles and the number of casualties in a collision.

It further supports the idea that as the number of vehicles increases from 0 to 2, the number of casualties are also increasing in a positive relation. Although this trend does not continue when the number of involved vehicles is 2+, this insight makes the number of involved vehicles a promising engineered feature.

Collision Contributing Factors

In this section, we preview the contributing factors of the first vehicle in collisions to get a sense of the most injurious contributing factors for collisions.

In [ ]:
```
%%bigquery --project $project_id
SELECT contributing_factor_vehicle_1, AVG(number_of_persons_killed + number_of_persons_injured) as danger_rate
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
WHERE number_of_persons_killed > 0 OR number_of_persons_injured > 0
GROUP BY contributing_factor_vehicle_1
HAVING COUNT(*) > 15  # adjustable
ORDER BY danger_rate DESC
LIMIT 20
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[5]:

	contributing_factor_vehicle_1	danger_rate
0	Shoulders Defective/Improper	2.277778
1	Unsafe Speed	1.656126
2	Using On Board Navigation Device	1.621622
3	Traffic Control Device Improper/Non-Working	1.618557
4	Physical Disability	1.611791
5	Brakes Defective	1.598933
6	Traffic Control Disregarded	1.557003
7	Alcohol Involvement	1.556675
8	Accelerator Defective	1.549383
9	Drugs (illegal)	1.544737
10	Drugs (Illegal)	1.541353
11	Cell Phone (hand-Held)	1.503185
12	Fatigued/Drowsy	1.484279
13	Following Too Closely	1.478448
14	Tire Failure/Inadequate	1.476619
15	Unsafe Lane Changing	1.464387
16	Other Vehicular	1.462611
17	Cell Phone (hand-held)	1.450000
18	Reaction to Uninvolved Vehicle	1.442388
19	Driver Inexperience	1.429465

When collisions occur, the responding officer will list the "contributing factors" of each vehicle with the intent of describing the cause of the accident. Hence, it makes intuitive sense that the contributing factors in a collision have a high correlation with how injurious an incident is. No/tight shoulders, especially in a busy city, leads to more stressful and dangerous road conditions, so it makes sense that this factor has a correlation with high casualty rates. Similarly, unsafe speeds cause more disasterous crashes which would tend to involve more casualties aswell.

Spatial Analysis of collisions in NYC

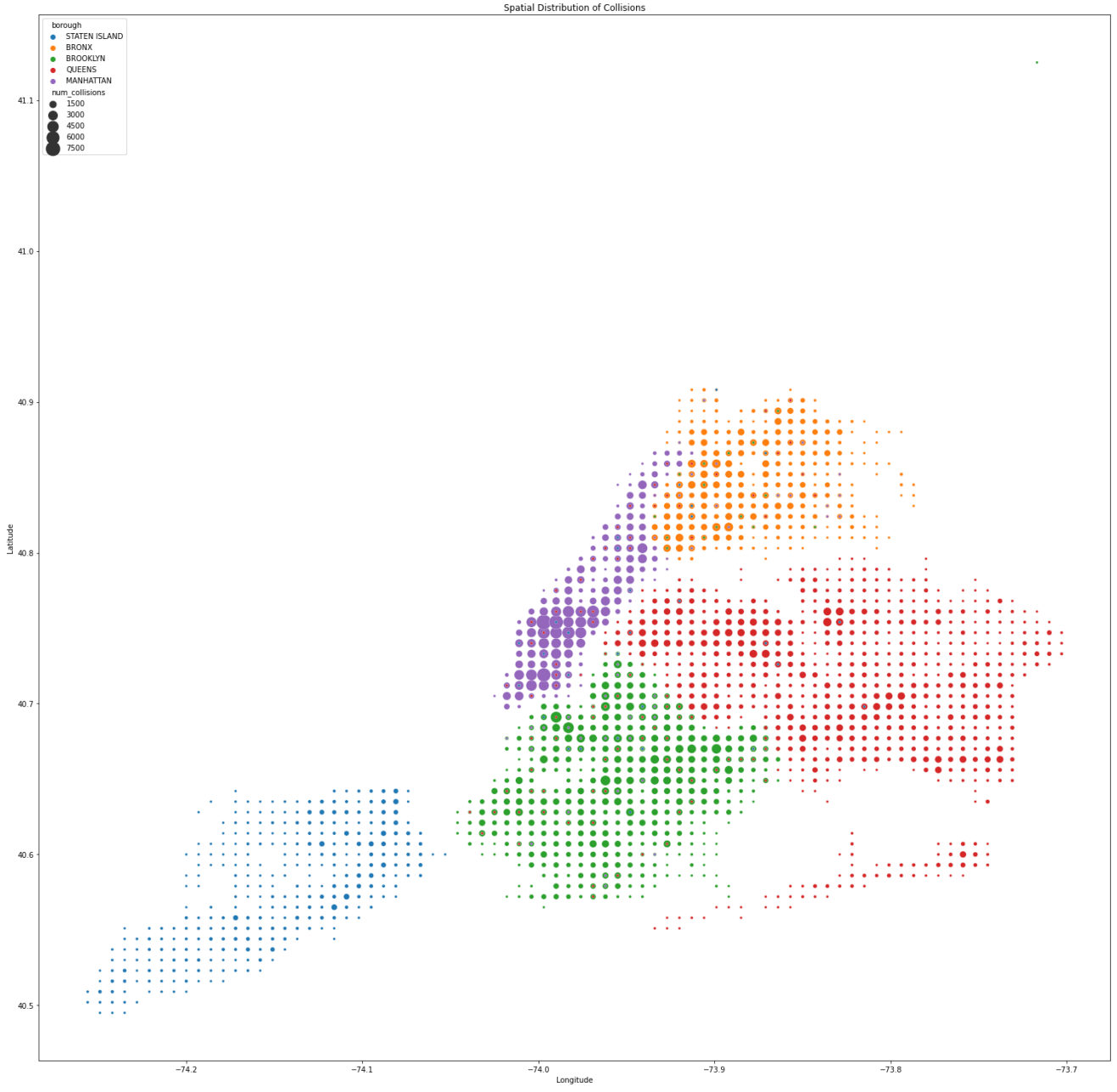
In this section, we will examine the spatial distribution of the collisions in NYC by creating a bubble chart heat map of the collisions by geo-location. This will allow us to get a high level overview of where accidents are occuring.

In [ ]:
```
%%bigquery collisions --project $project_id
SELECT borough, latitude, longitude
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
WHERE borough IS NOT NULL AND latitude != 0 AND longitude !=0
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

In [ ]:
```
step = 0.007

collisions["latBin"] = np.floor(collisions.latitude / step) * step
collisions["lonBin"] = np.floor(collisions.longitude / step) * step
groups = collisions.groupby(["latBin", "lonBin", "borough"]).count()

groups.rename(columns={'latitude': 'num_collisions'}, inplace=True)

plt.figure(figsize = (26, 26))

ax = sns.scatterplot(x='lonBin', y='latBin', hue='borough', data=groups, size='num_collisions', sizes=(12, 400))
ax.set(title='Spatial Distribution of Collisions', xlabel="Longitude", ylabel="Latitude")
```

Out[86]:
```
[Text(0, 0.5, 'Latitude'),
 Text(0.5, 0, 'Longitude'),
 Text(0.5, 1.0, 'Spatial Distribution of Collisions')]
```



For reference, here is a map of NYC (image from Alamy)

Danger to the parties involved in a collision

In this section we examine the lethality rates (Fatalities ÷ (Injuries + Fatalities)) for different parties involded in accidents where at least one person is harmed or injured.

In [ ]:
```
%%bigquery --project $project_id
SELECT ROUND((SUM(number_of_motorist_killed) / (SUM(number_of_motorist_injured) + SUM(number_of_motorist_killed)))*100, 2) AS mot
ROUND((SUM(number_of_cyclist_killed) / (SUM(number_of_cyclist_injured) + SUM(number_of_cyclist_killed)))*100, 2) AS cyclist_letha
ROUND((SUM(number_of_pedestrians_killed) / (SUM(number_of_pedestrians_injured) + SUM(number_of_pedestrians_killed)))*100, 2) AS p
ROUND((SUM(number_of_persons_killed) / (SUM(number_of_persons_injured) + SUM(number_of_persons_killed)))*100, 2) AS general_letha
FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions`
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[9]:

	motorist_lethality_rate	cyclist_lethality_rate	pedestrian_lethality_rate	general_lethality_rate
0	0.27	0.41	1.3	0.47

Pedestrians are evidently, and perhaps intuitively, least able to mitigate the damage from a harmful motor vehicle collision than cyclists. All numbers are percentages.

Citibike Stations Dataset Analysis

This section focuses on analysing the citibike stations dataset in conjunction with the collisions dataset and making observations about the features. We will first clean and extract data from the citibike stations dataset.

Extract the names of the streets a citibike station is on from station name

In [ ]:
```
%%bigquery station_streets --project $project_id
WITH station_names_split AS(
    SELECT station_id, SPLIT(name, ' & ') AS streets
    FROM `bigquery-public-data.new_york_citibike.citibike_stations`),
    station_street_names AS(
    SELECT station_id, UPPER(streets[OFFSET(0)]) AS street1,
    IF(ARRAY_LENGTH(streets) > 1, UPPER(streets[OFFSET(1)]), NULL)as street2
    FROM station_names_split
    )

SELECT a.station_id, name, latitude, longitude, region_id, a.capacity, num_bikes_available, num_bikes_disabled,
num_docks_available, num_docks_disabled, is_installed, is_renting, is_returning, b.street1, b.street2
FROM `bigquery-public-data.new_york_citibike.citibike_stations` a, station_street_names b
WHERE a.station_id = b.station_id
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Convert citibike station street names to the unabbreviated uppercase format used for street names in the motor vehicle collisions table. Create a SQL table in the current project with columns from street names added to the original citibike stations dataset for further querying.

In [ ]:
```
# adapted from user PeteCrosier at https://community.esri.com/t5/python-questions/street-name-abbreviation-replacement-script-wor
def street_name_fix(StreetName): # make sure to cite this
    suffix = {
                    'PKWY': 'PARKWAY',
                    'TER': 'TERRACE',
                    'SQ': 'SQUARE',
                    'AVE': 'AVENUE',
                    'RD': 'ROAD',
                    'CIR': 'CIRCLE',
                    'DR': 'DRIVE',
                    'LN': 'LANE',
                    'CT': 'COURT',
                    'PL': 'PLACE',
                    'ST': 'STREET',
                    'BLVD': 'BOULEVARD'}
    prefix = {
                    'N': 'NORTH',
                    'E': 'EAST',
                    'S': 'SOUTH',
                    'W': 'WEST'
                                    }

    StreetName = StreetName.strip().rstrip('.')

    try:
        if StreetName.split()[0] in list(prefix.keys()):
          return '{} {} {}'.format(prefix[StreetName.split()[0]], ' '.join(StreetName.split()[1:-1]), suffix[StreetName.split()[-
        else:
          return '{} {}'.format(' '.join(StreetName.split()[:-1]), suffix[StreetName.split()[-1]])

    except IndexError:
        return StreetName

    except KeyError:
        return StreetName

for ind in station_streets.index:
    station_streets.at[ind, 'street1'] = street_name_fix(station_streets.at[ind, 'street1'])
    if station_streets.at[ind, 'street2'] is not None:
      station_streets.at[ind, 'street2'] = street_name_fix(station_streets.at[ind, 'street2'])
    #print((station_streets.at[ind, 'street1'], station_streets.at[ind, 'street2']))

station_streets.to_gbq(destination_table='new_york_citibike.citibike_station_streets', if_exists='replace', project_id=project_id
```

```
100%|██████████| 1/1 [00:00<00:00, 6043.67it/s]
```

To find which collisions occured near a citibike station, we first determine whether a motor vehicle collision took place at any of the following locations: on a street, just off of a street, or with a cross street which has a citibike station. This step is necessary to approximate whether a collision may have taken place near a citibike station, reducing the number of collisions under consideration, before computing the geographic distance of such collision to each citibike

station. A column is added to the collisions table to record whether a collision took place near a citibike station, with collisions which occured within 250m of a citibike station are marked as 1, all other collisions as 0.

In [ ]:
```
%%bigquery engineered_mv_collisions --project $project_id
WITH citibike_station_locations AS(
  SELECT ST_GeogPoint(longitude, latitude) AS location
  FROM `bigquery-public-data.new_york_citibike.citibike_stations`
  WHERE latitude IS NOT NULL AND longitude IS NOT NULL
  ),
  collisions_on_station_streets AS(
  SELECT DISTINCT a.unique_key AS unique_key, 1 AS on_citibike_street
  FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions` a, `new_york_citibike.citibike_station_streets` b
  WHERE b.street1 IN (UPPER(a.cross_street_name), UPPER(a.off_street_name), UPPER(a.on_street_name))
  OR b.street2 IN (UPPER(a.cross_street_name), UPPER(a.off_street_name), UPPER(a.on_street_name))
  ),
  all_collisions AS(
  SELECT x.*, IF(y.unique_key IS NOT NULL, 1, 0) as on_citibike_street
  FROM `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions` x
  LEFT JOIN collisions_on_station_streets y
  ON x.unique_key = y.unique_key
  )

SELECT * EXCEPT(on_citibike_street), IF (
  a.on_citibike_street = 1
  AND (SELECT COUNT(*) AS count
  FROM citibike_station_locations b
  WHERE ST_DISTANCE(ST_GEOGPOINT(a.longitude, a.latitude), b.location) < 250) > 0, 1, 0
) AS citibike_factor
FROM all_collisions a
WHERE a.longitude IS NOT NULL
AND a.latitude IS NOT NULL
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Add the newly engineered collisions table to the current database. This table will be useful when training models.

In [ ]:
```
engineered_mv_collisions.to_gbq(destination_table='new_york_mv_collisions.engineered_mv_collisions', if_exists='replace', project
```

```
100%|██████████| 1/1 [00:00<00:00, 6335.81it/s]
```

We then examine the percentage of collisions which occured near a citibike station, what percentage of those collisions were injurious (person(s) either injured or killed), and what percentage of all collisions, regardless of location, were injurious. Collisions near citibike stations make up a small but far from insignifant portion of all collisions and do have a markedly higher likelihood of being injurious.

In [ ]:
```
%%bigquery --project $project_id
WITH collisions_near_stations AS(
  SELECT*
    FROM `new_york_mv_collisions.engineered_mv_collisions`
    WHERE citibike_factor = 1
  )


SELECT (SELECT COUNT(*) FROM collisions_near_stations) / COUNT(*) AS Percent_of_Col_Near_Stations,
 (SELECT COUNT(*) FROM collisions_near_stations WHERE number_of_persons_injured + number_of_persons_killed > 0) /
 (SELECT COUNT(*) FROM collisions_near_stations) AS Col_Near_Stations_Injurious_Percent,
 (SELECT COUNT(*) FROM `new_york_mv_collisions.engineered_mv_collisions` WHERE number_of_persons_injured + number_of_persons_kill
 COUNT(*) AS All_Col_Injurious_Percent
FROM `new_york_mv_collisions.engineered_mv_collisions`
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[91]:

	Percent_of_Col_Near_Stations	Col_Near_Stations_Injurious_Percent	All_Col_Injurious_Percent
0	0.119138	0.267592	0.219722

Data Prediction

Create the Database to store our models

In [ ]:
```
# Run this cell to create a dataset to store your model, or create in the UI

model_dataset_name = 'ny_collision'

dataset = bigquery.Dataset(client.dataset(model_dataset_name))
dataset.location = 'US'
client.create_dataset(dataset)
```

Out[92]:
```
Dataset(DatasetReference('eg-cs145-project-22', 'ny_collision'))
```

Train the basic model

We train on all data from years outside of 2018 and 2019. We use 2018 as our test set and 2019 as our validation set as each of these years accounts for ~10% of the data.

We choose indicators such as time of day, location, types of vehicles involved, external factors, and motorist/cyclist/pedestrian induced factors as the features for our model. The query:

In [ ]:
```
%%bigquery --project $project_id
SELECT
  IF(number_of_persons_injured + number_of_persons_killed > 0, 1, 0) AS label,
  borough AS feature1,
  EXTRACT(HOUR FROM timestamp) AS feature2,
  EXTRACT(DAY FROM timestamp) AS feature3,
  EXTRACT(MONTH FROM timestamp) AS feature4,
  contributing_factor_vehicle_1 AS feature5,
  contributing_factor_vehicle_2 AS feature6,
  contributing_factor_vehicle_3 AS feature7,
  contributing_factor_vehicle_4 AS feature8,
  contributing_factor_vehicle_5 AS feature9,
  vehicle_type_code1 AS feature10,
  vehicle_type_code2 AS feature11,
  vehicle_type_code_3 AS feature12,
  vehicle_type_code_4 AS feature13,
  vehicle_type_code_5 AS feature14,
  zip_code AS feature15
FROM
  `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions` a
WHERE EXTRACT(YEAR FROM timestamp) NOT IN (2018, 2019)
LIMIT 10
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[93]:

	label	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	feature11	feature12	feature13	feature
0	0	None	17	14	10	Passing or Lane Usage Improper	Passing or Lane Usage Improper	None	None	None	Sedan	Taxi	None	None	No
1	0	None	2	24	9	Pavement Slippery	Unspecified	None	None	None	Station Wagon/Sport Utility Vehicle	Pick-up Truck	None	None	No
2	0	None	15	15	9	Other Vehicular	Other Vehicular	None	None	None	Sedan	Sedan	None	None	No
3	0	None	17	30	9	Unspecified	None	None	None	None	Sedan	None	None	None	No
4	0	None	20	6	9	Unspecified	None	None	None	None	Sedan	None	None	None	No
5	0	None	12	2	9	Unsafe Lane Changing	Unspecified	None	None	None	Tractor Truck Diesel	Sedan	None	None	No
6	0	None	17	22	8	Unspecified	None	None	None	None	Sedan	None	None	None	No
7	0	None	8	12	9	Driver Inattention/Distraction	Unspecified	None	None	None	Sedan	Sedan	None	None	No
8	0	None	11	8	9	Unspecified	Unspecified	None	None	None	Sedan	None	None	None	No
9	1	None	16	9	8	Following Too Closely	Unspecified	None	None	None	Sedan	Sedan	None	None	No

Now, we train the model which will be trying to predict whether a collision has casualties. That is, the label we are trying to predict is boolean true when the number of people injured or killed is non-zero.

In [ ]:
```
%%bigquery --project $project_id

CREATE OR REPLACE MODEL `ny_collision.collision_model` -- we'll call our model 'collision_model'
OPTIONS(model_type='logistic_reg') AS
SELECT
  IF(number_of_persons_injured + number_of_persons_killed > 0, 1, 0) AS label,
  borough AS feature1,
  EXTRACT(HOUR FROM timestamp) AS feature2,
  EXTRACT(DAY FROM timestamp) AS feature3,
  EXTRACT(MONTH FROM timestamp) AS feature4,
  contributing_factor_vehicle_1 AS feature5,
  contributing_factor_vehicle_2 AS feature6,
  contributing_factor_vehicle_3 AS feature7,
  contributing_factor_vehicle_4 AS feature8,
  contributing_factor_vehicle_5 AS feature9,
  vehicle_type_code1 AS feature10,
  vehicle_type_code2 AS feature11,
  vehicle_type_code_3 AS feature12,
  vehicle_type_code_4 AS feature13,
  vehicle_type_code_5 AS feature14,
  zip_code AS feature15
FROM
  `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions` a
WHERE EXTRACT(YEAR FROM timestamp) NOT IN (2018, 2019)
```

```
Query is running:   0%|          |
```

Out[22]:
```
—
```

Training stats:

In [ ]:
```
%%bigquery --project $project_id

# Run cell to view training stats

SELECT
  *
FROM
  ML.TRAINING_INFO(MODEL `ny_collision.collision_model`)
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[23]:

	training_run	iteration	loss	eval_loss	learning_rate	duration_ms
0	0	5	0.436679	0.431811	0.8	14142
1	0	4	0.439983	0.435855	3.2	13061
2	0	3	0.451495	0.448991	1.6	14059
3	0	2	0.476090	0.477037	0.8	12871
4	0	1	0.500975	0.504521	0.4	12565
5	0	0	0.551277	0.553799	0.2	10357

Evaluating the model on the test set:

In [ ]:
```
%%bigquery --project $project_id

SELECT
  *
FROM
  ML.EVALUATE(MODEL `ny_collision.collision_model`, (
SELECT
  IF(number_of_persons_injured + number_of_persons_killed > 0, 1, 0) AS label,
  borough AS feature1,
  EXTRACT(HOUR FROM timestamp) AS feature2,
  EXTRACT(DAY FROM timestamp) AS feature3,
  EXTRACT(MONTH FROM timestamp) AS feature4,
  contributing_factor_vehicle_1 AS feature5,
  contributing_factor_vehicle_2 AS feature6,
  contributing_factor_vehicle_3 AS feature7,
  contributing_factor_vehicle_4 AS feature8,
  contributing_factor_vehicle_5 AS feature9,
  vehicle_type_code1 AS feature10,
  vehicle_type_code2 AS feature11,
  vehicle_type_code_3 AS feature12,
  vehicle_type_code_4 AS feature13,
  vehicle_type_code_5 AS feature14,
  zip_code AS feature15
FROM
  `bigquery-public-data.new_york_mv_collisions.nypd_mv_collisions` a
WHERE EXTRACT(YEAR FROM timestamp) = 2018)
)
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[24]:

	precision	recall	accuracy	f1_score	log_loss	roc_auc
0	0.710328	0.269411	0.833865	0.390655	0.435137	0.744588

Looking at the performance metrics of the model, we can see that we can achieve seemingly great accuracy of 0.834 with this model. This number is a bit misleading as only 21% of the dataset is collision where a non-zero number of people were injured or killed. Thus, if our model only predicted negative labels, deciding that no people were injured or killed, for every collision, that model would have an accuracy around 0.79. However, it's undeniably promising that the model performs better than that baseline at 0.834. Furthermore, the metrics are promising because our ROC AUC is good. At 0.745, it means the model does a decent job distinguish between positive and negative examples.

Precision is the number of true positives, divided by the number of total positive predictions. The prediction rate is 0.71, so when the model predicts that a collision had casualties, it's usually correct. However, since recall is the total number of positive labels we correctly predicted, we can see that the model is missing alot of collisions that had casualties as recall is low. These statistics combined means that our model is making lots of negative predictions, and being conservative when it comes to predicting a positive labels on collisions.

Hence, we will try to improve the model by adding more engineered features.

##Improving model with engineered features

Now, we add the engineered features to the model.

The first engineered feature we add describes whether a collision took place near a citibike station.

The second engineered feature is the number of reported vehicles involved in a collision.

The third engineered feature is a semantic representation of the intersection related to the collision. That is, the feature is constructed via an alphabetically ordered concatination of the main street and cross street.

In [ ]:
```
%%bigquery --project $project_id

CREATE OR REPLACE MODEL `ny_collision.collision_model_v2` -- we'll call our model 'collision_model_v2'
OPTIONS(model_type='logistic_reg') AS
SELECT
  IF(number_of_persons_injured + number_of_persons_killed > 0, 1, 0) AS label,
  borough AS feature1,
  EXTRACT(HOUR FROM timestamp) AS feature2,
  EXTRACT(DAY FROM timestamp) AS feature3,
  EXTRACT(MONTH FROM timestamp) AS feature4,
  contributing_factor_vehicle_1 AS feature5,
  contributing_factor_vehicle_2 AS feature6,
  contributing_factor_vehicle_3 AS feature7,
  contributing_factor_vehicle_4 AS feature8,
  contributing_factor_vehicle_5 AS feature9,
  vehicle_type_code1 AS feature10,
  vehicle_type_code2 AS feature11,
  vehicle_type_code_3 AS feature12,
  vehicle_type_code_4 AS feature13,
  vehicle_type_code_5 AS feature14,
  zip_code AS feature15,
  citibike_factor AS feature16,
  CASE WHEN vehicle_type_code1 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code2 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_3 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_4 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_5 IS NOT NULL THEN 1 ELSE 0 END AS feature17,
  CONCAT(LEAST(GREATEST(off_street_name, on_street_name), cross_street_name), GREATEST(GREATEST(off_street_name, on_street_name),
FROM
  `new_york_mv_collisions.engineered_mv_collisions` a
WHERE EXTRACT(YEAR FROM timestamp) NOT IN (2018, 2019)
```

```
Query is running:   0%|          |
```

Out[32]:

Training stats with engineered feature:

In [ ]:
```
%%bigquery --project $project_id

# Run cell to view training stats

SELECT
  *
FROM
  ML.TRAINING_INFO(MODEL `ny_collision.collision_model_v2`)
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[33]:

	training_run	iteration	loss	eval_loss	learning_rate	duration_ms
0	0	5	0.436927	0.448552	0.8	12736
1	0	4	0.439767	0.451964	3.2	13618
2	0	3	0.452274	0.461586	1.6	13362
3	0	2	0.477139	0.484503	0.8	12289
4	0	1	0.502949	0.509107	0.4	13063
5	0	0	0.553319	0.556588	0.2	11127

Evaluating the model with engineered features on the test set:

In [ ]:
```
%%bigquery --project $project_id

SELECT
  *
FROM
  ML.EVALUATE(MODEL `ny_collision.collision_model_v2`, (
SELECT
  IF(number_of_persons_injured + number_of_persons_killed > 0, 1, 0) AS label,
  borough AS feature1,
  EXTRACT(HOUR FROM timestamp) AS feature2,
  EXTRACT(DAY FROM timestamp) AS feature3,
  EXTRACT(MONTH FROM timestamp) AS feature4,
  contributing_factor_vehicle_1 AS feature5,
  contributing_factor_vehicle_2 AS feature6,
  contributing_factor_vehicle_3 AS feature7,
  contributing_factor_vehicle_4 AS feature8,
  contributing_factor_vehicle_5 AS feature9,
  vehicle_type_code1 AS feature10,
  vehicle_type_code2 AS feature11,
  vehicle_type_code_3 AS feature12,
  vehicle_type_code_4 AS feature13,
  vehicle_type_code_5 AS feature14,
  zip_code AS feature15,
  citibike_factor AS feature16,
  CASE WHEN vehicle_type_code1 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code2 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_3 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_4 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_5 IS NOT NULL THEN 1 ELSE 0 END AS feature17,
  CONCAT(LEAST(GREATEST(off_street_name, on_street_name), cross_street_name), GREATEST(GREATEST(off_street_name, on_street_name),
FROM
  `new_york_mv_collisions.engineered_mv_collisions` a
WHERE EXTRACT(YEAR FROM timestamp) = 2018)
)
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[34]:

	precision	recall	accuracy	f1_score	log_loss	roc_auc
0	0.717013	0.270182	0.833982	0.392473	0.433522	0.74613

As we can see, adding our engineered features slightly increases the model's performance. However, since the recall of the model is still quite low, we can expect that the model is not great for generating predictions over collisions missing casualty data.

##Using the trained model to predict on the validation set

In [ ]:
```
%%bigquery --project $project_id

SELECT
  unique_key,
  predicted_label
FROM
  ML.PREDICT(MODEL `ny_collision.collision_model_v2`, (
SELECT
  unique_key AS unique_key,
  IF(number_of_persons_injured + number_of_persons_killed > 0, 1, 0) AS label,
  borough AS feature1,
  EXTRACT(HOUR FROM timestamp) AS feature2,
  EXTRACT(DAY FROM timestamp) AS feature3,
  EXTRACT(MONTH FROM timestamp) AS feature4,
  contributing_factor_vehicle_1 AS feature5,
  contributing_factor_vehicle_2 AS feature6,
  contributing_factor_vehicle_3 AS feature7,
  contributing_factor_vehicle_4 AS feature8,
  contributing_factor_vehicle_5 AS feature9,
  vehicle_type_code1 AS feature10,
  vehicle_type_code2 AS feature11,
  vehicle_type_code_3 AS feature12,
  vehicle_type_code_4 AS feature13,
  vehicle_type_code_5 AS feature14,
  zip_code AS feature15,
  citibike_factor AS feature16,
  CASE WHEN vehicle_type_code1 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code2 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_3 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_4 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_5 IS NOT NULL THEN 1 ELSE 0 END AS feature17,
  CONCAT(LEAST(GREATEST(off_street_name, on_street_name), cross_street_name), GREATEST(GREATEST(off_street_name, on_street_name),
FROM
  `new_york_mv_collisions.engineered_mv_collisions` a
WHERE EXTRACT(YEAR FROM timestamp) = 2019
))
LIMIT 10
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[35]:

	unique_key	predicted_label
0	4147196	0
1	4151087	0
2	4099287	0
3	4164277	0
4	4075860	0
5	4111875	0
6	4115289	0
7	4102094	0
8	4190825	1
9	4187799	0

A count of the number of collisions in validation which the model labeled as probably injurious is produced to verify that the model is not merely giving all incidents the same label. For reference, there were 194114 collisions in 2019.

In [ ]:
```
%%bigquery --project $project_id

SELECT
  COUNT(*) AS labeled_injurious
FROM
  ML.PREDICT(MODEL `ny_collision.collision_model_v2`, (
SELECT
  unique_key AS unique_key,
  IF(number_of_persons_injured + number_of_persons_killed > 0, 1, 0) AS label,
  Borough AS feature1,
  EXTRACT(HOUR FROM timestamp) AS feature2,
  EXTRACT(DAY FROM timestamp) AS feature3,
  EXTRACT(MONTH FROM timestamp) AS feature4,
  contributing_factor_vehicle_1 AS feature5,
  contributing_factor_vehicle_2 AS feature6,
  contributing_factor_vehicle_3 AS feature7,
  contributing_factor_vehicle_4 AS feature8,
  contributing_factor_vehicle_5 AS feature9,
  vehicle_type_code1 AS feature10,
  vehicle_type_code2 AS feature11,
  vehicle_type_code_3 AS feature12,
  vehicle_type_code_4 AS feature13,
  vehicle_type_code_5 AS feature14,
  zip_code AS feature15,
  citibike_factor AS feature16,
  CASE WHEN vehicle_type_code1 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code2 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_3 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_4 IS NOT NULL THEN 1 ELSE 0 END +
  CASE WHEN vehicle_type_code_5 IS NOT NULL THEN 1 ELSE 0 END AS feature17,
  CONCAT(LEAST(GREATEST(off_street_name, on_street_name), cross_street_name), GREATEST(GREATEST(off_street_name, on_street_name),
FROM
  `new_york_mv_collisions.engineered_mv_collisions` a
WHERE EXTRACT(YEAR FROM timestamp) = 2019
))
WHERE predicted_label = 1
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

Out[36]:

	labeled_injurious
0	15225

Conclusion

We will wrap up by answering each of our subquestions with explanations based on our analysis.

How do common contributing factors influence severity of a collision?

From the insights gained in the data exploration section, we see that the time of day (hour) is the biggest influencing factor for collisions. Month and year had less of an influence on the number of collisions. The visualizations we created showed us that there is an uptick in collisions at 8am each day with the most number of collisions occuring from 2pm-6pm. Conceptually, this made sense as we can assume that NYC traffic and the number of drivers are at the worst during morning rush hour and when people are getting off work (8am and 2-6pm respectively). The hourly analysis also revealed differences in collision patterns between the boroughs. For instance, at 8 AM, traffic in the Bronx is at its peak and declines for a few hours while traffic in Manhattan, known for its businesses, and Queens continues to mount through the workday.

Furthermore, we also believe that the time of day influences how injurious the collision will be. Although we did not perform dataset analysis related to this idea, we believe this is true simply because there are people on the streets of NYC, whether on bikes, in cars, or walking, during the day. Hence, it makes sense that the patterns we observed in the number of collisions through the day might carry the same expressivity with relation to the occurence of casualties in a collsion.

How do common contributing factors influence severity of a collision?

Road and intersection maintenance issues were among the most dangerous factors, with shoulders defective and improper being most sever and traffic control device improper/non-working being 4th most severe. This is understandble, as even when abiding by traffic laws and driving carefully, a poorly maintained road or point of traffic flow becomes dangerous as drivers have to take corrective actions to ensure everyone's safety. The danger of unsafe speed is perhaps self explanatory. Roads designed for certains speeds do not allow room to manuver at much higher speeds. Interestingly, use of navigation device is much higher than texting or cell-phone use when driving as an indicator of severity. Perhaps this is because one uses their navigation device in an unfamiliar area where they are more prone to mistakes where as the typical driver texting or calling while driving has probably driven the area before.

How does severity scale with the number of vehicles involved in a collision?

From our exploration, we saw that the number of casualties involved in each collision tends to trend upward as the number of motor vehicles increases from 0 to 2. However, when the number of involved vehicles is 2+, the number of casualties no longer increases linearly with the number of vehicles. The relation between casualties and number of vehicles is a high level proxy for the size of a collision. That is, collisions involving more casualties and more vehicles can be

interpreted as being more severe.

There were limitations in our analysis as we did not go into more granular detail around the vehicle types or locations around the city. These granular analyses could have shed more light on possible engineered features with more correlation to casualty presence of a collision. Another limitation is the sheer volume of collisions happening despite a maximum of 5 involved vehicles being reported. Hence, the collision data is compressed into only 5 categories and we lose information about collisions involving 5+ vehicles.

To what extent does the street(s) impact the severity of the accident?

In our data exploration, we examined how specific streets might have influence on the severity of collisions. One of our first investigations was with regards to the relations between street names and casualty rates of motorists, cyclists, and pedestrians. We can conclude from these explorations that the street a collision occurs on does have correlation with the casualty rate for a specific person group. Specifically, we found strong evidence that streets within Prospect Park and Central Park are particularly injurious for cyclists. Additionally, our analysis supports the idea that highway on/off ramps are the most dangerous streets for motorists. However, our analysis around the relation of streets to pedestrian casualties is much less concrete, and we cannot make definitive conclusions about how streets affect pedestrian casualties.

What does the precense of Citibike stations tell us?

Our query analysis and model demonstrate the the precense of citibike station is a meaningful indicator of severity of collisions in an area. What our data does not tell us is causality. Citibike likely builds stations in place they foresee having a strong base of riders, which are likely the most densely populated areas which are still within biking range of businesses and attractions. Commercial business and dense populated together create traffic, which increases potential for accidents. However, Citibike stations are useful as feature because they overtly tell the model that collisions near them are more severe and implicitly tell the model both that the area is both busy and frequented by cyclists and pedestrian on their way to the station, two groups which experience more harm from collisions.

Summary

The contributing factors and vehicle types are the strongest indicators of accident severity. When the features representing either of those two categories were removed from the model, its ability to accurately predict labels degenerated significantly. Adding engineered features such as proximity to a citibike station did improve the model but they are by no means as essential to the model, and therefore in determining the severity of a crash. This is reasonable, as locations, times, and proximity to citibike stations may all correlate with number of cars on the road, speeds at which people are driving, etc. The types of vehicles involved in a crash determine the amount of energy that goes into the crash and are strong indicators of how many individual people might be affected by a crash. A bus, for instance, holds tens of passengers, and first responder vehicles such as firetrucks or ambulances carry a crew a several first responders and have to respond to emergencies at high speeds, so the collision force and number of people in the collision will be great.