# NYC Crash Analysis

## How Time and Location Affect the Number of Crashes in NYC

Andrew Dharmaputra
Student ID: 1213935
Github repo with commit

August 24, 2022

## 1 Topic and Dataset Introduction

This report aims to aid authorities responsible for handling crashes by informing them of when and where crashes are more likely to occur in New York City (NYC). The report uses a crash dataset that is externally acquired from the NYC Open Data titled Motor Vehicle Collisions – Crashes [1]. A couple baseline models, random forest, and gradient boosting regressions are implemented to predict the number of crashes based on time and location, but the latter two are preferred due to their good performance and flexibility on non-normally distributed datasets.

The timeline is 2017 to 2019, because for 2016, it is potentially too old to be applicable nowadays; and from 2020 onwards, Covid-19 changed the entire structure of the streets, changing the applicability of this report's findings. As NYC slowly returns to normal in 2022, this report will hopefully become more feasible and practical to be used as a guidance for the authorities for future crashes.

This report will use the season (derived from the date of the crash), the daylight (derived from the time of the crash), the average speed based on time (derived from the NYC Taxi and Limousine Commission (TLC) data) [2], and the location of borough as attributes; and the number of crashes as labels.

Firstly, the season is used as an attribute because each season can have an influence on the number of vehicles on the streets, i.e., whether more people preferring to hail a taxi rather than taking a public transport due to the summer heat or winter cold.

Secondly, the daylight accounts for visibility. In this report, the day is defined as the period where artificial light is not needed for outdoor visibility, whereas the night is defined as the period where artificial light is required. This distinction between day and night can be achieved by looking at the time when civil twilight begins/ends. The twilight data is scraped from dateandtime.com [3], which follows the steps outlined in the article about web scraping by Vandany Lubis [4].

Thirdly, the speed indicates the average speed of all taxi trips in a given 24-hour time period, derived from the yellow TLC data. This is useful particularly to infer the busyness of the road (in other words, less speed means busier road, and vice versa) which is directly correlated to the number of crashes.

Lastly, the location attribute consists of each borough instead of each zone, because the zones are geographically small and most trips will go through multiple zones, so using the borough is more feasible in real life in this scenario.

# 2 Data Preprocessing and Analysis

## 2.1 TLC Data

The 2019 yellow TLC dataset is used to analyse the average speed of vehicles in a 24-hour time frame. Although it does not represent the entire population, neither 2018, 2017, nor any green TLC datasets are used, because it is not worth sacrificing computational time and memory since they are highly likely to be similar to the 2019 yellow TLC data. The data for the average speed of vehicles other than taxis are also not possible to be gathered. However, this data should be a good representation to measure the busyness of the roads at a certain time.

Extreme outliers present in this data, many of them due to the values not recorded properly. There are trips whose drop off time is earlier than pick up time, trips which last multiple days, and trips with unrealistically small/large/negative distance. As a result, there are infinite, negative, invalid, and unrealistically high or low speed values which need to be removed.

Since the average speed data are skewed by outliers, their left and right tails deviate significantly from a normal distribution. Therefore, using standard deviations to detect outliers is not the best method since the data's distribution is unknown. The alternative method to calculate outlier thresholds is by using the interquartile range (IQR). However, after further checking and analysing the data distribution, using 1.5×IQR as threshold is quite strict, and it does result in removing values that are perfectly valid, such as when a trip is mostly spent 80km/h on a highway. As a result, the outlier thresholds will be 2×IQR, which allows more leeway but is still fairly strict to ensure the final result is not skewed.

After the outliers are removed, the data then may or may not be randomly sampled (depending on the processing capability and memory of the machine running the code, or the patience of the user), for its average to be calculated on a minute-by-minute basis.

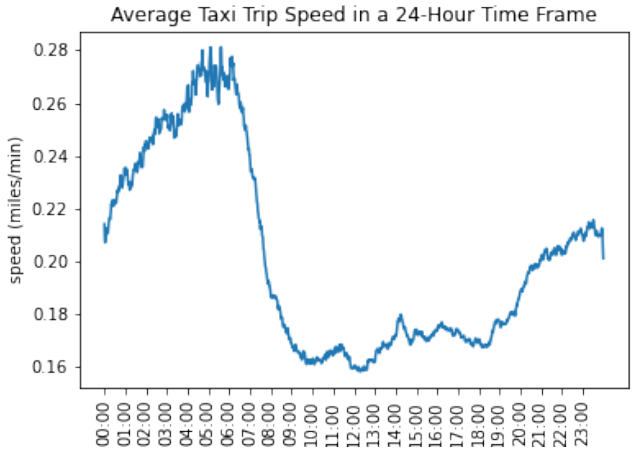



Figure 1: Average Taxi Trip Speed in a 24-hour Time Frame

According to NYC TLC, their "rush hour" is defined as the time period from 4pm to 8pm on weekdays due to the surcharge they impose if trips occur during those hours [5]. However, as seen in Figure 1, the true impact of rush hour in NYC is hardly seen. There is a slight dip from about 09:00 to 13:00, but the difference is only around 0.01 miles/min, which is negligible. This is because the average speed data takes into account not only the weekdays, but all days. It is likely that the trends for weekdays and weekends/public holidays are different, but even on weekdays, the difference between normal and

rush hour during the day is much smaller compared to the difference between any time of day and 4 AM. Hence, it is more feasible in this case to treat each day as having a "rush day".

This continuous data is then discretised into three bins based on the observed values in Figure 1:

- Normal: 08:00 - 20:00
- Fast: 20:00 - 02:00, 07:00 - 08:00
- Empty: 02:00 - 07:00

## 2.2 Twilight Data

Although the timeline of this research is from 2017 to 2019, only the 2019 twilight data is used. This is because the pattern is unchanging every year – and all of 2017, 2018, and 2019 are not leap years, so the 2019 data is valid to all of those three years. The purpose of this data is to create a distinction between night and day for the crash data.

## 2.3 Crash Data

The crash data is filtered such that it only consists of crashes from 2017 to 2019.

From this data, the speed data, and the twilight data, information about a crash's season, daylight, and speed can be inferred from the crash's date and time. The crash's location coordinate, which consists of longitude and latitude, is then aggregated with the TLC taxi zones data to determine the borough in which the crash had occurred.

### 2.3.1 Time

Fortunately in this data, every crash has its own date and time recorded. Although the time values are a bit skewed towards even numbers (e.g. 13:00, 16:00 occurs much more often than 13:42, 12:51), it does not impose a problem towards the final result in daylight and speed differentiation, since they are all merely rounded to the nearest hour. What imposes a problem, however, is the suspiciously large number of crashes which occurred at 00:00. The most likely explanation as to why this could happen is because crashes whose time are not recorded are automatically allocated at the time 00:00.

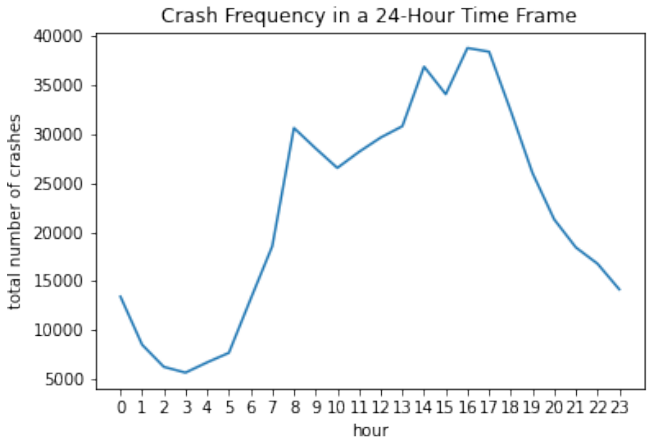



Figure 2: Crash Frequency in a 24-Hour Time Frame

However, this ends up not being a problem anyway. As seen in Figure 2, only a small amount of crashes has its time unrecorded, and the value at 00:00 does not deviate noticeably from the trend. In addition, the deviation becomes more negligible after the data is aggregated for analysis in section 3.

Another important trend found from the above figures is that the crash frequency is almost inversely proportional to the average speed of vehicles. This is probably due to their direct correlation to the number of vehicles travelling outside, which is then directly correlated to the human sleeping hours. At the early AM hours, where most people are asleep, the average speed is at its highest and the number of crashes is the opposite. Likewise, the number of crashes peaks at the afternoon, where most people are awake and busy.

### 2.3.2 Location

What is not fortunate, unfortunately, is that not all crashes has its location coordinate recorded. Out of 674056 total crashes, only 621346 has its location recorded. As a result, there are slightly less than 8% of data which has missing location values, and therefore data imputation is required. Since the data is categorical, one of the best ways to impute missing data is by creating a random sample based on the distribution of the known data. This, however, assumes that there are no significant interactions between "borough" and other features.

### 2.3.3 Aggregation

The next step is to "groupby" the crash count by its features: season, daylight, speed, and location. However, this does not result in all possible value combinations for the features since "count" only has values 1 or above. In order to make this data complete, all possible combinations of season, daylight, speed, and location need to be included in the data with 0 count values. Not all combinations are possible. For example; summer, night, normal and spring, night, normal is not possible from April 13 until August 29, because the day ends later than 8 PM (and obviously starts earlier than 8 AM). These combinations are ensured to be removed from the final dataset. The time length for each combination is also inconsistent. On the $1^{st}$ of January; winter, day, empty only lasts from 06:49 to 07:00 whereas winter, day, normal lasts from 08:00 until the end of the day. Since it is not a good idea to normalise the number of crashes based on time, there will inevitably be skewed 0 values in the response variable.

As a result, the final data shape consists of 30765 instances, with four columns of predictor variables, and one column of response variable.

## 3 Modelling and Post-Modelling Analysis

Before implementing models, the nominal features (season, daylight, and location) are transformed into ones and zeros, and the ordinal feature (speed) is transformed into 0 for normal, 1 for fast, and 2 for empty. After splitting into 75% train and 25% test sets, a dummy regressor is then used as a baseline whose prediction always outputs a constant value that is the most common, 0.

As seen in Figure 3, the distribution of crashes is not normal, and is difficult to infer, which takes any type of linear models out of consideration. Poisson regression could be implemented here, but it is not a good idea since the mean and variance of the data is drastically different, indicating that the data distribution is not Poisson. However, despite Poisson regression not being an ideal model, it decently outperforms the baseline model. Therefore, Poisson regression will be used as the new baseline model.

The two preferred models to be used for this data are random forest and gradient boosting regressions. Considering the distribution of this data is unknown, implementing models who are not sensitive to the

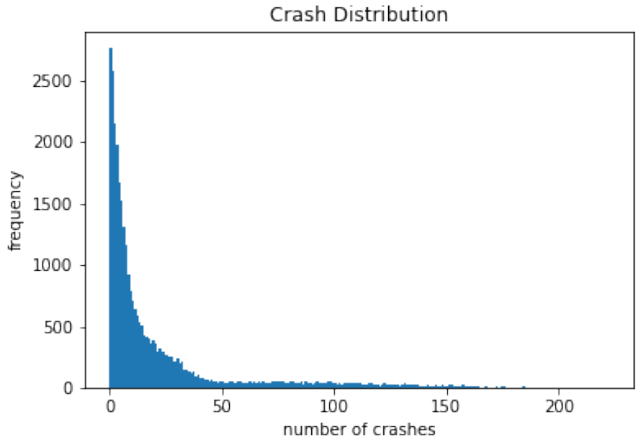



Figure 3: Crash Distribution

data distribution is desirable. In addition, both random forest and gradient boost are fairly robust to overfitting. Considering there are only four features (expanded to 12 after encoding nominal features), overfitting is a big risk for many statistical models, making these two models ideal to use in this situation. Additionally, their performance is excellent, although it comes with a trade off that they both are slow to train. However, it is not a problem since the training size is reasonably small.

The main idea of the random forest algorithm from COMP30027 [6] is that it uses an ensemble of multiple regression trees, built using different bagged training datasets, to find the best fit. The difference is, instead of using random forest as a classifier, this model is used to predict continuous data, making it a regressor, using the mean squared error as the criterion for each node split.

Likewise, the gradient boosting algorithm also does a regression based on an ensemble of regression trees. The difference is, unlike random forest, this model builds its trees by using an instance manipulation method called "boosting" [6]. The model aims to minimise the gradient of the mean squared error in each step, similar to gradient descent, with the learning rate set to 0.1 to minimise the risk of overfitting. The number of trees in both random forest and gradient boosting are set to 500, with their maximum depth set to 10 to improve accuracy and minimise variance.

### 3.1 Model Performance

The evaluation metrics used to evaluate model performance are mean absolute error (MAE) and root mean squared error (RMSE). They are both used since they are generally easy to interpret. RMSE penalises values further from the mean more than MAE, so in this dataset where the values are right-skewed and variance is high, it is expected that the RMSE values for all models are much higher than the MAE.

The summarised performance of each model can be seen on the table below.

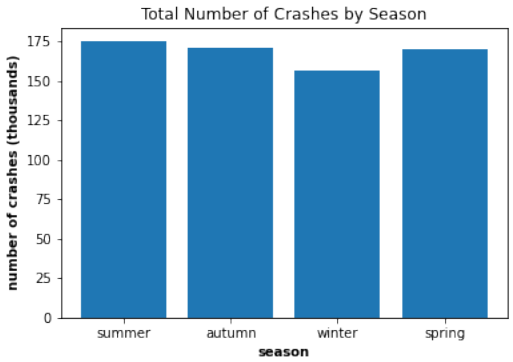
| Model | True Mean | True Var | Predicted Mean | Predicted Var | MAE | RMSE |
|---|---|---|---|---|---|---|
| Dummy | 21.918 | 1171.855 | 0 | 0 | 21.918 | 40.648 |
| Poisson | | | 21.866 | 553.020 | 12.274 | 18.207 |
| Random Forest | | | 21.904 | 1071.390 | 5.752 | 10.086 |
| Gradient Boost | | | 21.905 | 1071.556 | 5.753 | 10.086 |

Table 1: Model Performance

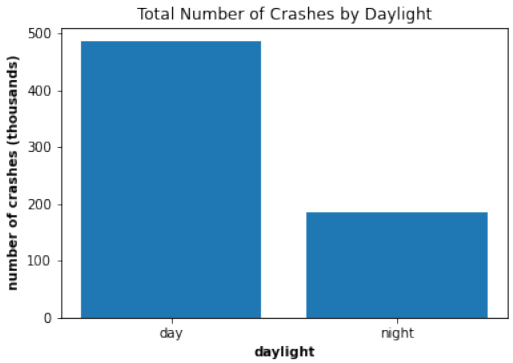
Ultimately, both random forest and gradient boosting are the two best and preferred models, and in addition, they both have extremely similar performance as well.

### 3.2 Feature Analysis

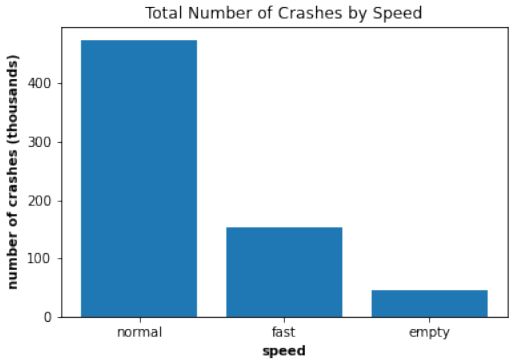
The distribution of the four features can be seen in Figure 4 below. According to the plots, the distribution of crashes according to season are nearly uniform, meaning that this attribute is potentially not useful in predicting the number of crashes. Meanwhile, the number of crashes based on speed, daylight, and borough are distributed in a certain way. It can be seen that the number of crashes that occurred during the day is much higher compared to the night. The speed plays a more significant role, where crashes are much more likely to happen during normal hours, i.e., from 08:00 to 20:00. Lastly, more crashes are likely to occur at Brooklyn or Queens, and unlikely to occur at Staten Island.



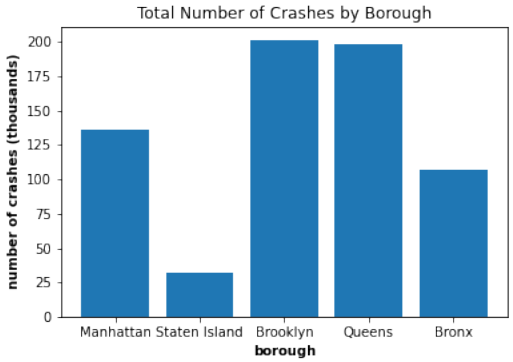

(a) fig 1




(b) fig 2




(c) fig 3




(d) fig 4

Figure 4: Distribution of Features in the Crash Dataset

These distributions follow a noticeable pattern where the number of crashes is directly proportional

to the number of vehicles travelling outside. Ultimately, the main takeaway from these graphs is that according to the crime data, the highest number of crashes on average, in theory, happens during the day, during normal (08:00 - 20:00) hours, and in Brooklyn or Queens.

The models (random forest and gradient boosting) are then used to find out the most important features. The feature importance scores according to the two models are averaged, which results in the following the top three features: the speed, the Staten Island borough, and the daylight. The results are not surprising, given the trends of the graphs in Figure 4.

The top 10 predictions on the test set by gradient boost (both models perform very similarly, so it doesn't matter which one to choose) is as follows ("Count" denotes the number of crashes):

| Season | Daylight | Speed | Borough | Count |
|---|---|---|---|---|
| Summer | Day | Normal | Brooklyn | 137 |
| Summer | Day | Normal | Queens | 130 |
| Spring | Day | Normal | Brooklyn | 129 |
| Spring | Day | Normal | Queens | 125 |
| Autumn | Day | Normal | Brooklyn | 119 |
| Autumn | Day | Normal | Queens | 111 |
| Winter | Day | Normal | Brooklyn | 97 |
| Winter | Day | Normal | Queens | 96 |
| Summer | Day | Normal | Manhattan | 92 |
| Spring | Day | Normal | Manhattan | 85 |

Table 2: Top 10 Predicted Crashes

As seen on the table, the trend is also not surprising. There are nearly equal amount of each season in the top 10; and the daylight, speed, and location follows the trend from Figure 4 such that most values, if not all, in the top 10 are day, normal, and Brooklyn/Queens.

## 4 Discussion and Recommendations

There are numerous other factors causing vehicle crashes other than time and location. For example, the education level of the drivers, the in-depth weather information (e.g., wind speed/gust, rain, and snow), the quality of road facilities, how busy the streets are, and many more. Although finding all of these information is possible, it is arguably impractical in real life to implement an accurate research of those data by the authorities responsible for handling crashes, since it could be a huge waste of time or/and resources compared to simply knowing the time and the location.

Fortunately, many of those factors are correlated with time and location. Take busyness as an example. Streets are busier during the day than at night, and when it is busy, the probability of a crash happening will be higher. The busyness of the road is arguably a better feature to be used to predict the number of crashes, but it is rather impractical for the authorities to predict exactly how busy the roads are. A much better way to predict crashes is by predicting time and location, because they are directly correlated to the busyness of the roads.

Once the time and location are known, authorities then are able to allocate their budget and their priorities accordingly based on the findings from this report. For example, summer, day, normal, Brooklyn is predicted to have the most number of crashes: 137 crashes with 6 MAE (rounded to the nearest integer). That means, they are recommended allocate more budget and employees to handle the crashes in Brooklyn and less budget/employees to the time and location in which the number of

crashes is much lower (at night in Staten Island, for example). By optimally allocating resources, authorities are then able to handle the crashes more efficiently, which is particularly beneficial to the crash victims, as well as preventing unnecessary wasted budget, saving more money in the process.

One shortcoming of this project is the fact that there is no direct correlation between the weather and the time and location. Another recommendation is to aggregate the data with the weather data such that there will be three aspects (time, location, and weather) used to estimate the number of crashes. However, in order to find the past weather data, the daily data might not be the best idea due to the nature of weather that it changes quickly, and accurate hourly weather data are obscure and therefore difficult and challenging to find.

# 5 Conclusion

In conclusion, this report finds that the highest number of crashes, on average, occurs during the day, during normal (08:00 - 20:00) hours, and in Brooklyn or Queens – whereas the season does not contribute much to the number of crashes. This finding is backed by the visualisations of the distribution of each feature as well as the results of the statistical modellings.

The report then concludes three recommendations:

1. Authorities are advised to focus on gathering information about time and location to predict the number of crashes
2. Authorities should allocate budgets and priorities accordingly based on this report's findings on how time and location affect the number of crashes
3. Authorities are recommended to find the past hourly weather data in addition to time and location to increase the accuracy of the predicted number of crashes

## References


[1] NYC Open Data. *Motor Vehicle Collisions - Crashes*. https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95/data. Accessed: 2022-08-22.

[2] New York City Taxi and Limousine Commission. *TLC Trip Record Data*. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Accessed: 2022-08-22.

[3] dateandtime.info. *Sunrise and sunset times, day length in New York, New York, USA*. https://dateandtime.info/citysunrisesunset.php?id=5128581. Accessed: 2022-08-22.

[4] Vandany Lubis. *How to Scrape Table from Website using Python*. https://medium.com/analytics-vidhya/how-to-scrape-a-table-from-website-using-python-ce90d0cfb607. Accessed: 2022-08-14.

[5] New York City Taxi and Limousine Commission. *Taxi Fare - TLC*. https://www1.nyc.gov/site/tlc/passengers/taxi-fare.page. Accessed: 2022-08-22.

[6] Hasti Samadi and Ling Luo. *Classifier Combination*. COMP30027 Lecture. 2022.