Random Forests to Predict

Car Crash Fatalities and Injuries

Jared Kinneer

Swanson School of Engineer, University of Pittsburgh

4200 Fifth Ave, Pittsburgh, PA 15260

jrk152@pitt.edu

Abstract— This research employs Random Forests, a machine learning method, to predict car crash fatalities and injuries based on data collected in New York City since 2012. Leveraging comprehensive data, the models offer precise predictions. The primary aim is to enhance emergency response resource allocation by integrating machine learning into first responders’ systems. By implementing these models, emergency responders can more quickly and accurately dispatch the correct services based on predictions generated using data collected at the time of a 911 call. This study contributes to advancing emergency management systems, highlights the benefits of machine learning in the public sector, and most importantly, reduces the human cost of car crashes.

Keywords— Random Forest, Machine Learning, Python, Decision Trees, Data, C++

1. Introduction And Motivation

The goal of this research was to determine if using random forests was a viable approach to predicting car crash fatalities and injuries in NYC. The motivation behind this research is to provide better emergency response to car crashes. In 2023 there were 48,116 injuries and 238 fatalities in NYC as a result of car accidents [1]. The idea is that if we can use data that can be collected at the time of a 911 call to predict the severity of an accident, injured persons can be cared to sooner and deaths prevented.

1. Random Forest

In this section, the mechanics behind random forests will be explained as well as the factors behind choosing that as the machine learning model for this problem.

1. Understanding Random Forests

The formal definition of a random forest according to Leo Breiman is, “A random forest is a classifier consisting of a collection of tree-structured

classifiers {h(x, Theta\_k), k = 1,...} where the {Theta\_k} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x” [2]. Now because we are using the random forest to predict a continuous output, the forest uses the average of the outputs for the decision trees in the forest.

Now for how the forest is created. First the training data is split evenly, with replacement, amongst the number of trees in the forest. Then each tree randomly selects a designated number of features from the total number of features.

Now each individual tree is assembled node by node using the following steps. The data subset for each tree is recursively split into further subsets based on one feature at a time to create a hierarchical structure. The criteria for this split and what feature is used is determined mean squared error (MSE). The MSE at each node is determined with the following formula:

MSE(m) = 1/Nm∑i ∈Dm (yi – ym)2

Where Nm is the number of data points in the subset, Dm is the set of indices for the datapoints as they are found in the original data, yi is the actual result for datapoint i and ym is the average result for all the datapoints in the set. Then the feature with the minimum MSE is selected for that level of the tree. This process happens for the left and right of each node until all features in the feature subset have an assigned node.

1. The Data

This section will explore the original dataset, preprocessing challenges, and the processed dataset that was used for random forest models.

1. Original Dataset

The original data set was comprised of just over 2 million records dating back to 2012. Each row has a combination of up to 29 features. The date, location data (borough, zip code, latitude, and longitude), contributing factors for vehicles 1 to 5, and the type of vehicle for vehicles 1 to 5.

The first step I took was calculating the number of vehicles involved in the crash, as this was not a recorded metric. This was done by counting the number of non-empty vehicle-type descriptions for each row.

After adding this new feature, and extracting relevant features, we end up with 9 features and 1 output.

1. Preprocessing Challenges and Outcomes

The first issue that was encountered in generating the predictive models was the number of features that the model was generating.

The reason for this is a process called one hot encoding. One hot encoding finds each unique value in a non-numeric feature column creates a new column for each of those unique values and uses ones and zeros to indicate if the original feature matched the new column.

Of the 9 features in the training data, 4 were non-numeric (Vehicle Type 1, Vehicle Type 2, Contributing Factor Vehicle 1, Contributing Factor Vehicle 2). After using one-hot encoding on this data, the number of features increased to 3409. This resulted in an incredibly slow training time.

After close inspection it was discovered that the entries in the vehicle type category were not standardized, leading to misspellings, incomplete words, and nonsense entries.

To combat this, I found the 40 most common entries in the vehicle type 1 and vehicle type 2 columns and put them in their own sets. I then iterated over the original data and if either of the vehicle types could not be found in their respective sets that row was skipped.

This resulted in a new data set that was 1.6 million rows rather than 2 million. And after one hot encoding, the feature space was 206 as opposed to 3409. This reduction in training data greatly increased the runtime of the model and increased its accuracy.

1. Results

This section will investigate the results of using random forest regression models. Runtimes, model accuracy, mean absolute error, and most important features. All data was collected from training and testing the models on a combination of forest sizes and maximum features allowed in each decision tree. In total 28 different combinations were tested.

A. Training Times

The first figure below shows the training times of each combination in seconds, for the fatality regression model.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training Times | | Maximum Features | | | | | | |
| 3 | 5 | 10 | 14 | 25 | 100 | 205 |
| Forest Size | 2 | 9.35 | 11.15 | 14.43 | 15.79 | 22.63 | 70.46 | 139.23 |
| 10 | 28.5 | 33.65 | 44.27 | 52.45 | 96.40 | 348.08 | 689.38 |
| 25 | 63.6 | 72.54 | 95.9 | 127.5 | 210 | 731.07 | 1497.3 |
| 100 | 229 | 254.6 | 352.9 | 445.5 | 749.7 | 2911.5 | 5951.8 |

This figure shows the training times for the injury regression model.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training Times | | Maximum Features | | | | | | |
| 3 | 5 | 10 | 14 | 25 | 100 | 205 |
| Forest Size | 2 | 12.9 | 13.65 | 15.62 | 17.75 | 22 | 63.03 | 111.41 |
| 10 | 46.7 | 48.18 | 58.92 | 66.29 | 88.51 | 286.77 | 523.70 |
| 25 | 110 | 113.4 | 134.9 | 158.9 | 204.1 | 646.13 | 1287.2 |
| 100 | 408 | 423.9 | 494.4 | 563.8 | 777.9 | 2556.2 | 5164 |

These figures show there is a linear relationship between forest size and training time, as well as maximum features and training time.

A graph of a graph with a line

Description automatically generated with medium confidence

A graph of a training

Description automatically generated with medium confidence

The two graphs above illustrate this linear relationship.

B. Mean Absolute Error.

First, what is the mean absolute error (MAE)? MAE is defined as:

* N = number of data points
* actual result for data point i
* predicted result for data point i

For the fatality regression, the average MAE across all experiments was .0013, with a variance of 2.72 \* 10^-9.

For the injury regression model, the average MAE across all experiments was .404 with a variance of 4.31\*10^-5.

1. Most Important Features

The importance of a feature can be expressed as

a percentage. This percentage is derived from how well a feature splits the data and how well it contributes to a correct prediction. This measure of well it splits and contribution to correct answer is measured node by node for all trees in the forest and then aggregated at the end.

Below are two figures that show the sum of the feature importance’s across all experiments for both models.

A graph with blue and black text

Description automatically generatedA graph with blue and white text

Description automatically generated

To recap, crash time, latitude, longitude, zip code and number of vehicles involved in the crash are the top predictors of output across both random forests.

1. Accuracy

For this research, I recorded the accuracy of the models using two different metrics. I used the MAE to derive accuracy using this equation:

This works by looking at the MAE in comparison to the total range of the data. The idea is that the closer the MAE is to the maximum range of the data the further away predictions are to be from the actual values.

The second measure of accuracy I used was R2. The R2 value of the data shows the proportion of the variance in the output variable that is explained by the feature variables. R2 values are typically in the range of 0 to 1.

For the fatality random forest model, the highest MAE-derived accuracy was %99.95481. This was for a random forest with 10 trees and a maximum of 10 features per tree. The R2 for that experiment was -.08. The best R2 achieved was.

-.0171 for a forest of size 100 with 14 features per tree. The MAE-derived accuracy for that experiment was %99.95477

For the injury random forest model, the highest MAE-derived accuracy was %98.36947. This was for a random forest with 100 trees and a maximum of 14 features per tree. The R2 for that experiment was .028. The best R2 achieved was .0352 for a forest of size 100 with 100 features per tree. The MAE-derived accuracy for that experiment was %98.35301

Overall, the R2 values increased as forest size increased for a constant number of features and decreased as features increased for a constant number of trees.

A graph of a number of injuries

Description automatically generated

A graph of a number of injuries

Description automatically generated

1. Conclusions

The first thing to discuss is the accuracy measurements for the models. The low MAE and high MAE-derived accuracy for both regression models indicate that they predict outcomes that are very close to the expected outcome. However, the extremely low, even negative, R2 values show that there is little to no correlation between the variance of the number of fatalities/injuries and the features in the data.

This means that the high accuracy is obtained due to there being a large amount of training data and an unlimited tree depth, however, no amount of data can manufacture a correlation between the output and input variables.

In conclusion, random forests can successfully predict the amount of people killed and injured in car crashes. However, more testing should be conducted to confirm the best machine-learning model for improving emergency response to car crashes.

Acknowledgment

I would like to thank Dr. Amr Mahmoud for fostering my interest in big data and algorithms, exposing me to new material, and providing support throughout the semester. I would also like to thank Will Koehrson for his tutorial for random forests in Python [3].

References

[1] Police Department (NYPD, “Motor Vehicle Collisions - Crashes,” *Cityofnewyork.us*, Apr. 28, 2014. https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95 (accessed Dec. 02, 2023).

[2] L. Breiman, “Random Forests,” Jan. 2001. Accessed: Dec. 02, 2023. [Online]. Available: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>

[3] W. Koehrsen, “Random Forest in Python,” *Medium*, Dec. 27, 2017. <https://towardsdatascience.com/random-forest-in-python-24d0893d51c0>