Evaluating the risk of traffic crash in Chicago

Introduction

Traffic crashes are among the most prevalent transportation related accidents in the United States. Each year traffic crashes causes significant amount of life losses as well as property damage. For example, in 2016 alone there were 37,461 people killed in crashes on U.S. roadways, an increase from 35,485 in 2015[1]. These fatalities involved all segments of the population: vehicle drivers, passengers, motorcyclists, pedestrians, pedalcyclists etc. In response to the need to address such issues, many U.S. cities established initiatives to improve the on-road safety and to reduce the traffic crashes, especially fatal crashes. One of the most widely acknowledged initiatives is Vision Zero. Cities that joined the Vision Zero network are the ones having a clear goal of eliminating traffic fatalities and severe injuries, strategies and plans in place or under development and key city departments engaged[2]. Currently there are 35 cities/counties committed to this initiative across the U.S..

The City of Chicago, among the 35 cities, has the most comprehensive municipal information databases. The tremendous efforts in collecting relevant data including transportation data made it possible to evaluate and even predict the risk of traffic crashes.

The primary objective of this project is to take advantage of the municipal information databases of the City of Chicago and build predictive models utilizing machine learning algorithms. The outcome of such a model should be able indicate the risk of traffic crashes by locations so that the city departments could allocate resources more accurately and efficiently in their effort of improving transportation safety. The specific purposes of the models would be to 1) predict and quantify the risk of traffic crashes by locations (i.e. road segments) within the city, and 2) find out the top features by their importance in indicating such risks.

Methodology

The entire project had two major models that are sequentially developed and implemented. The crash model is contingent to the implementation of the traffic model.

Data sources

The traffic and crash model were primarily constructed and evaluated using two datasets:

- **Illinois Highway Information System (IRIS)– Roadway Information[3]** distributed by the Illinois Department of Transportation (IDOT) as shapefiles on a yearly basis (i.e. one shapefile for each year). This dataset includes 106 variables covering aspects such as

[1] 2016 Fatal Motor Vehicle Crashes: Overview. Available at: https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812456
[2] VISION ZERO CITIES MAP. https://visionzeronetwork.org/resources/vision-zero-cities/. Accessed Sep 7, 2018.
[3] http://apps.dot.illinois.gov/gist2/

traffic volume, road conditions, roadway structures, management responsibilities etc. (See appendix or support documents for detailed list of these variables).

- **The City of Chicago Boundary**[4] distributed by the City of Chicago as a shapefile. This dataset was applied in the data preprocessing stage to define the boundary of the studied area and to select the road segments within the administrative boundary of the City of Chicago.
- **Annual Crash Reports in the City of Chicago**[5] distributed by the Illinois Department of Transportation (IDOT) as csv files. The full crash dataset includes features regarding the accident, vehicle and person. This crash model primarily ingested information contained in the accident reports and evaluated the 29 variables associated with this subset.
- **311 Maintenance Requests in the City of Chicago**[6] distributed by the City of Chicago. 311 maintenance requests datasets evaluated and ingested in this model include: potholes on the road, tree debris, tree trimming request, street light(s) out. The datasets are publicly available through the City of Chicago data portal in the csv format.

Roadways contained in the IRIS shapefiles were first processed based on the city boundary defined in the shapefile distributed by the City of Chicago to only include road segments within the city's administrative boundary.

Traffic model

Data preprocessing

Data preprocessing includes the following steps:

Step 1

Variables from the shapefiles were first manually screened in order to exclude ones that were not qualified for modeling. These variables fall into the following categories:

1) The variables only contain descriptive information that are in the form of texts. Examples are county name, road name, municipality name, etc.
2) The variables have been deprecated and only exist in earlier versions.
3) The variables contain textual information that are not applicable to be considered as either continuous or discrete/categorical variable. This categorical is specific to *DTRESS_OPP* and *DTRESS_WTH* for which values were missing for over 20% of the road segments.

Step 2

[4] https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-City/ewy2-6yfk
[5] Data source: https://github.com/stevevance/Chicago-Crash-Browser/
[6] https://data.cityofchicago.org/browse?category=Service%20Requests

Exploratory analysis subsequently excluded variables that 1) are null for over 20% of the entire dataset; 2) do not have enough variation (at least two unique values). This exclusion may vary based on the version of shapefile processed. Please check the Python scripts for more details.

Step 3

Manual inspection was performed to exclude any variables with missing values that are not represented as "null" but other values, such as 0 or 0000. This process also excluded any variables containing redundant information.

Step 4

Manual inspection was performed to exclude any variables that are irrelevant but SPECIFIC to the dataset/shapefile that include them. Such category may contain variables that, for example, are generated by GIS, have null values for more than 20% of the entire dataset or redundant as a predictor.

Model development

Considering the nature of the problem with the target variable[7] being numeric and continuous, regression algorithms were deemed appropriate.

Model development was performed in four steps:

Step 1: Screening for a subset of algorithms

Candidate algorithms were trained using default hyperparameter[8] settings and tested against a testing set resulted from a 70%/30% (training/testing) split of the entire dataset with variables retained after the preprocessing procedure. Algorithms showing significant inferior performances were excluded and no longer considered for further evaluation. Candidate algorithms evaluated in this step are: *K-nearest neighbor regression, random forest regression, gradient boosting regression, ada-boosting regression, linear regression, and support vector machine for regression (rbf and linear kernels)*. Table 1 shows the performances of the screening results from this step.

Table 1. Performance in the form of r^2 score and mean square log error developed in the first-round evaluation of candidate models[9]

Machine learning models	Performance (r^2)	Performance (mean square log error)
k-nearest neighbor regression	0.9842 ± 0.0047	0.1230 ± 0.0114

[7] A target variable is a variable whose values are to be modeled or predicted. In the case of this study, the target variable is AADT.

[8] A hyperparameter for a machine learning model/algorithm refers to a parameter whose value is set before the learning process begins. The types and values are usually specific to the model/algorithm of interest and they dictate how the model/algorithm is computationally optimized using the input data.

[9] Ten iterations with different seeds for train/test split were applied in the Step 1 evaluation. The values displayed in the table are in the format of "mean +/- standard deviation"

random forest regression	0.9885 ± 0.0036	0.0994 ± 0.0069
gradient boosting regression	0.9789 ± 0.0035	0.2319 ± 0.0083
ada-boosting regression	0.9254 ± 0.0076	1.0911 ± 0.0663
support vector machine (rbf kernel)	-0.0787 ± 0.0040	1.2669 ± 0.0392
support vector machine (linear kernel)	0.0255 ± 0.0063	0.8655 ± 0.0287

Step 1 identified that *k-nearest neighbor regression* and *random forest regression* exhibited excellent performance (i.e. highest r^2 scores and lowest mean square log error) among all the candidate machine learning algorithms and were thus selected for further evaluation.

Step 2: Feature selection

The feature selection process was intended to reduce number of predictors in the model by identifying the variables that have sufficient predicting power with regard to the target variable. Feature selection for this traffic model followed a hybrid method. Statistical methods such as "mutual information" implemented in the *sklearn* package were first applied to the variables included in the preprocessed datasets. The mutual information between the potential predictors and the target variable quantified from this step served as a general representation of the association between the two variables. Higher mutual information generally suggests higher predicting power for the variable of interest.

With the quantified mutual information between the potential predictors and the target variable, decisions on the variables to include was further made by evaluating the performance of the model with different combinations of the variables showing high mutual information values in the previous step. In addition, variables that have marginal contribution to the overall performance were not included. Note that this process also required the testing of all the candidate machine learning algorithms for the traffic model.

The variables that were selected as predictors in the final machine learning algorithms are:

SURF_WTH (surface width), FC (road class), X (longitude of the mid-point of the road segment), Y (latitude of the mid-point of the road segment)

Step 3: Algorithm selection

In the third step, a nested cross validation method was performed to determine the best hyperparameter(s) for the candidate machine learning algorithms selected from the first step and to evaluate their performances. The results also serve as support analysis for the final algorithm selection.

The nested cross validation includes two layers of cross validation. The inner cross validation functions as a process for the hyperparameter selection/model optimization. A "grid search" of all desired hyperparameter settings were explored in this step and the settings corresponding to the best model performance (measured by r^2 score) were identified. The following outer cross validation evaluated the model with the optimized hyperparameter settings identified

from the inner cross validation. The resulted metrics of the performance (r^2) were treated as one of the major criteria for the final algorithm selection.

Such "inner-outer cross validation" process was performed in 20 iterations with different training/testing sets generated by shuffling the dataset and using random seeds in the train-split function to minimize the impact of outliers and noise. The "best" hyperparameters were selected based on a majority vote among the best parameters resulted from all the iterations.

Figure *1* shows the results from 20 iterations of the nested cross validation described above that was performed to the 2015 IDOT IRIS dataset.

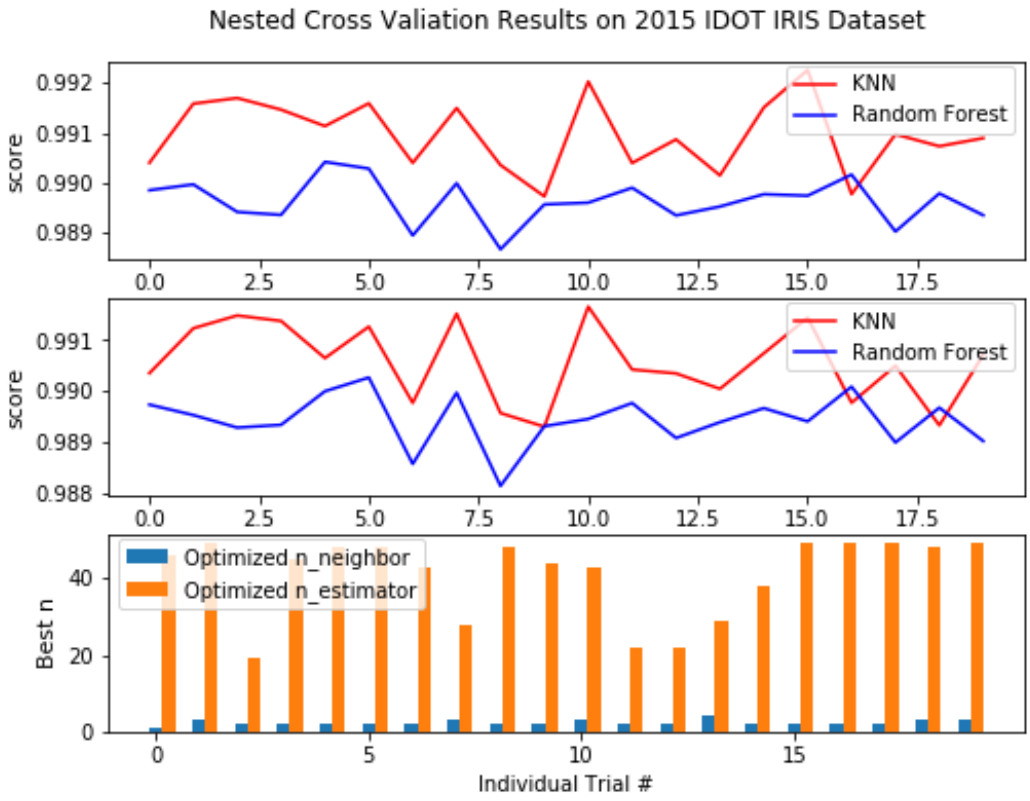

Figure 1. r^2 scores from the nested cross validation. Top: best r^2 from the inner cross validation (3 fold); Middle: mean r^2 scores from the outer cross validation (3 fold); Bottom: optimized hyperparameters in each iteration.

Although k-nearest neighbor regression show incrementally better performance (measured by r^2 score) than random forest regression algorithms, both algorithms showed r^2 scores that are well above 0.99 with regard to the randomly generated testing sets. **K-nearest neighbor (KNN) regression** algorithm with the number of neighbor as 2 was selected to be implemented in the final traffic model after evaluating its performances and characteristics. It yields highly stable performances (r^2 = 0.990 +/- 0.001) given randomly generated training and testing sets. More importantly, compared to random forest regression algorithm, the optimum values for its key

hyperparameter, number of neighbors, showed minimum variation in response to the perturbation of the training dataset.

Step 4: Model implementation

The IDOT IRIS datasets (in the format of shapefiles) are available on an annual basis. Therefore, the traffic model was implemented to each annual dataset (shapefile). The final outcomes of the modeling process were the predicted AADT values to each road segments by years of interest.

Crash model

Data Preprocessing

Illinois Highway Information System (IRIS)– Roadway Information

Please refer to the "Data preprocessing" section in the exposure model documentation for detailed descriptions.

Annual Crash Reports in the City of Chicago

Each crash record is supposed to have variables indicating the location of the occurrence of the crash. In the crash reports distributed by the IDOT, two sets of such variables were reported: X-Y coordinates and longitude-latitude. However, in multiple cases, one or both of these geospatial indicators were missing from the records (i.e. variables marked as 0). Data preprocessing practice aims to:

1. calculate any latitude and longitude pairs where X-Y coordinates are available with prior knowledge that the coordinates used in this dataset is Illinois West 1202
2. remove any crash records without any geospatial variables since the records need to be attached to each road segment for the modeling process

311 Maintenance Requests in the City of Chicago

The 311 requests datasets have time stamps associated with each request and were used to filter out records for the year of interest. The 311 request dataset for pothole repairs has a field "MOST RECENT ACTION" that could be used to indicate the actual validity of the record. Any records indicated as "NO PROBLEMS FOUND" or "NO SUCH ADDRESS FOUND" were thus removed from further analysis.

Model development

The exposure model was intended to predict the risk scores in the form of likelihood of occurrence of crashes on each road segment. Therefore, classification models were deemed appropriate for this purpose.

A complete training dataset for this model structure would require two subsets of samples: a positive sample set and a negative sample set. Each crash incident ingested in this model is

treated as a positive sample (i.e. 1 or true in the binary classification) and includes variables derived from the IRIS dataset and 311 requests, which is described above. Note that because there could be multiple crashes on one road segment during the time of interest, these positive samples (i.e. crash incidents) could be associated with the identical road-related information.

The construction of a negative sample subset requires some additional steps. For the current version of crash model, road segments without any crash incidents for two consecutive years [10]were treated as the "base sample" for the derivation of negative samples. Given that time related variables such as time of the day and day of the week will also be included into the final model[11], each road segment identified previously should be expanded into 24 * 7 (i.e. 24 hours and 7 days) samples to cover all possible time stamps.

However, it is unlikely that the features associated with the same road segment identified as "free of crashes" for two or more consecutive years would remain the same across these years. For example, the road conditions or number of 311 requests may very well vary from year to year. This may result in some confusion in constructing the negative sample subset. **The solution is:** in the current version of crash model, if a road segment is identified as free of crashes for two (or more) consecutive years but found having different feature values across the year, we still consider it as a qualified "base sample". However, the extra step is to treat each unique combination of feature values as an independent base sample and expand them to cover all possible time stamps. The possible justification/assumption would be that as long as the road is free of crashes for time period of interest, all the feature values during this time span can be associated to the "free-of-crash" status. To further strengthen the validity of this assumption, the model training-testing, model hyperparameter tuning and performance evaluation process can be performed for multiple iterations. In each iteration, the master sample dataset is constructed by randomly sampling from the both the positive and negative sample subsets. The statistics generated from these iterations for the performance metrics (e.g. accuracy) can be used to evaluate the methodology.

While following the similar procedures for the feature selection and hyperparameter tuning, due to the complexity and significantly increased computational cost, a simplified approach was adopted. The "mutual information" approach was used initially to provide a general guide to the feature selection and manually screening was performed to determine the variables to keep in the model. Instead of applying "GridSearchCV" find the best hyperparameter, which requires significant amount computational resources, the sensitivity of the performance was investigated against a few hyperparameter values, and a conservative one was selected for the final model implementation.

[10] Currently, only crash reports and road network information for 2015 and 2016 were considered.

[11] The fundamental requirement of this crash model is the ability to make predictions in fine temporal resolution (hourly or daily). Thus, time of day and day of week need to be kept as predictors.

Model deployment and results discussion

With the model successfully developed, a risk score was generated for each road segment for each combination of hour of day and day of week. The visualization of the results and a demo version of such application were developed in Tableau[12].

One interesting findings from this project is that higher crash risks are associated with higher number of 311 requests for street light maintenance and lower traffic volumes. More interestingly, 311 requests for street light maintenance is one of the most important predictors as indicated by the random forest classifier in the crash model. This may be counter-intuitive in that high-risk roads are not the ones with high traffic volume. In addition, higher number of 311 service requests may indicate neighborhoods that are generally poorly-managed. One of the possible outcomes from these observations might be that the roads in poorly managed neighborhoods may generally be at a higher risk of crash.

The work in this project demonstrates the good potential of this analysis tuning into a promising application. Future work to improve the models and fix the caveats may focus on:

- Random forest has been reported to generate probabilities that are biased. Other methods such logistic regression with lasso may be potentially helpful.
- Think about other metrics or modification of current metric to better represent the idea of crash risks
- Weather condition reported by the crash reports may result in bias estimation, and should be reconsidered and maybe removed

[12] Please see the following links:
https://public.tableau.com/profile/li7232#!/vizhome/CAP_Dashboard/Dashboard1
https://public.tableau.com/profile/li7232#!/vizhome/CAP_Dashboard2/Dashboard2