CE 528 Project Progress Report

Solving Imbalanced Data Distribution Issues in Road Crash Severity Prediction via Machine Learning Algorithms

Lin Lyu

1. Introduction

According to report of World Health Organization, there are more than 1.35 million people die each year from causes related to road crashes [1]. The leading cause of death in children and young adults between 5 and 29 years old is a consequence of road traffic crashes [1]. Even though all road crashes are problematic, vulnerable road users (pedestrians, cyclists, and motorcyclists) represent about 54% of all road deaths worldwide and 43% in Europe [1]. Vehicle type has been identified as the most important variable associated to the injury severity with the vulnerable road users being the ones that have higher chances of suffering from a severe injury or fatality [2]. Motor cyclists have 30 times more of a chance of death when involved in a road crash when compared with car occupants [3]. These statements push us to improve processes aimed at enhancing the road safety level of infrastructures, moderating the number of crashes, and evaluating the key factors that are the cause or contributing factors to crashes.

Crash severity is one of the road safety-related aspects that requires thorough investigation. Crash severities are usually measured by several discrete categories which are fatal, incapacitating injury, non-incapacitating injury, possible injury, and property damage only. In recent years, there has been extensive investigation of the relationship linking crash severity and its associated risk factors, as well as several studies that have involved the study of crash severity modeling for prediction purposes. Statistical modeling has been applied in crash severity analysis for a long time as the models provide good indicators of crash severity likelihood, and the results are easy to interpret. Regression models such as linear regression are the most used techniques [4-6] to determine the risk factors associated to the severity of road crashes. Statistical modeling requires some assumptions about the underlying probability distribution of the data and pre-defined relationships between dependent and independent variables. If these assumptions are violated, incorrect

inferences can be produced. Machine learning algorithms (MLAs) appear as one of the prominent and most exciting tools for modeling crash severity due to their outstanding outcomes. These techniques do not contain pre-assumed relationships between variables and the prediction is possible without requiring understanding of the underlying mechanisms. Machine learning methods can provide accurate prediction models because of their ability to deal with more complex functions [7-9]. While sometimes it will be hard to explain and understand the underlying logic behind the machine learning algorithm.

Nonetheless, there is a significant number of studies [7-11] that omit the fact that, typically, crash datasets are acutely imbalanced. Indeed, the number of events that correspond to fatality or severe injury are generally far fewer than the number of events relating to property damage only or minor injury. Learning from datasets that include occasional events usually provides biased classifiers: they have higher predictive accuracy over the majority class, but weaker predictive capacities over the minority class [12-14].

The purposes of this project are manifold. First of all, recent studies on crashes severity modeling are reviewed, showing that most of them employ imbalanced datasets to train MLAs. Secondly, the project will do exploratory data analysis and recognize key factors that affect crash severity most. Next, in order to handle imbalanced crashes datasets and provide a better prediction for the minority class, the random under-sampling the majority class (RUMC) technique is used for minor classes. The project also implemented random over-sampling the majority classes (ROMC) and tried different sample distribution combinations to under sample minor classes while over sample the majority in order to improve the prediction accuracy for testing datasets. Four machine learning algorithms would be used in this project: K-Nearest Neighbor (KNN), Logistic Regression (LR), Decision Tree and Random Forest (RF). Furthermore, with a goal of showing that RUMC and ROMC are useful in defining non-weak classifiers in predicting the minority class, the project provides a comparison between two scenarios. One scenario is training the four machine learning algorithms on imbalanced training dataset. Another scenario is training four machine learning algorithms on balanced training set with the use of a RUMC-ROMC based method. The project will calculate and compare the performance of different classifiers on different metrics: accuracy,

true positive rate (recall), false positive rate, true negative rate, precision, F1-score, and the confusion matrix.

2. Literature Review

The literature review is focused on crash severity modeling and prediction and use of prediction and clustering methods in traffic safety research. A huge amount number of safety research is focused on crash severity modeling.

Table 1 summarized the literature review information including the data used, the methods implemented and the corresponding results of comparing each model in the literature. More literature review details are included after Tables 1.

Table 1. Summary of literature review

Literature	Data	Methods used	Results
Fan et al. [15]	Crashes at highway-rail grade crossings	Multinomial Logit and an Ordered Logit model	Multinomial Logit model performed better than the Ordered Logit model
Mohamed et al. [16]	Pedestrian-vehicle crashes in New York, US and Montreal, Canada	Latent Class Clustering (LCC) and K-means Clustering (KC),	LCC provided better results in a crash dataset compared to K-means Clustering (KC), while KC had better performance in another crash dataset, relative to LCC.
Al-Turaiki et al. [17]	85,605 crashes that occurred in Riyadh, Saudi Arabia, between October 2014 and October 2015,	Decision Tree and Naive Bayes.	Decision trees were slight better than the Naïve Bayes; A poor performance of the models was obtained for the fatalities class, which may be because the dataset is too unbalanced.
Gopinath et al. [18]	13,062 crashes that happened between 2010 and 2015 in Leeds UK	Support Vector Machine, Naïve Bayes, and decision trees for classification techniques; Self Organizing Map and K-modes clustering techniques for cluster analysis.	The best performance was obtained with Decision Tree (70.75%). Self Organizing Map obtained better results than K-modes techniques. The classification techniques improved their accuracy by about 5% in all techniques when clustering methods were applied.
Zhang et al. [19]	5,538 road crashes that occurred in Florida, United States, in a period of 3 years, in which 51 are fatalities	Ordered Probit model, Multinomial Logit model, K-Nearest Neighbor, Decision Tree, Random Forest, and Support Vector Machine	Random Forest presented the best overall prediction accuracy
Li et al. [20]	5,538 crashes that occurred in 326 freeway segments in the state of	Support Vector Machine and Ordered Probit	The Support Vector Machine model presented a correct prediction of 48.8%

	Florida (United States) from 2004 to 2006		against 44.0% for the Ordered Probit model.
Hosseinzadeh et al. [21]	8,390 at-fault truck-involved crashes, between 2011 and 2014	Support Vector Machine model and Random Parameter Logit Model	Support Vector Machine model achieved a more accurate prediction than the Random Parameter Logit model.
Wang and Kim [22]	201,581 road crashes, with 0.47% of fatalities that occurred in Maryland, Unites States, in 2017	Multinomial Logit model and a Random Forest model	Random Forest presented a higher prediction accuracy than the Multinomial model.
Ahmadi et al. [23]	Rear-end crashes that occurred in the state of California, United States, between 2007 and 2011	Multinomial logit, mixed multinomial logit, and support vector machine	Multinomial logit and mixed multinomial logit presented similar prediction accuracy as the variables had few mixed effects. Support vector machine presented slightly better prediction accuracy.

A common approach in these studies is using a statistical modeling tool with crash severity as the dependent variable and characteristics of crash, driver, roadway, weather, etc. as independent variables. Fan et al. [15] evaluated the injury severity factor in crashes at highway-rail grade crossings in the United States, considering a data set of crash from 2005 to 2012. A Multinomial Logit and an Ordered Logit model were developed for the purpose. It is concluded that the Multinomial Logit model performed better than the Ordered Logit model in predicting the vehicle crash severity levels based on the Akaike information criterion.

To account for heterogeneity in crash data (presence of unobserved factors that are correlated with observed variables), a number of crash severity studies utilized data clustering methods. Some papers preferred Latent Class Clustering (LCC) as a model-based clustering method for clustering crash data to obtain homogeneity prior to crash severity modeling. Mohamed et al. [16] showed that LCC provided better results in a crash dataset compared to K-means Clustering (KC), while KC had better performance in another crash dataset, relative to LCC.

Literature about using machine learning methods and statistical learning methods were also reviewed. Al-Turaiki et al. [17] considered 85,605 crashes that occurred in Riyadh, Saudi Arabia, between October 2014 and October 2015, with 421 fatalities to determine the factors leading to car crashes severity. Three different machine learning algorithms were used to model the crash

severity: Decision Tree and Naive Bayes. The performance of the obtained models was compared using three measures: accuracy, precision, and recall. All models achieved similar accuracy values, but the decision trees were slight better than the Naïve Bayes. It is worth to mention that a poor performance of the models was obtained for the fatalities class, which may be because the dataset is too unbalanced. Distraction while driving and the age of the car (older cars) were associated to more severe crashes.

In Gopinath et al. [18] analyzed 13,062 crashes that happened between 2010 and 2015 in Leeds UK to determine attributes that contribute to road crashes. Support Vector Machine, Naïve Bayes, and decision trees were used in this study as classification techniques and Self Organizing Map and K-modes clustering techniques for cluster analysis. The measurement of accuracy of the different techniques was made through the confusion matrix. Support Vector Machine presented the lower accuracy of 68.46%, and Naïve bays presented an accuracy of 68.53%. The best performance was obtained with Decision Tree (70.75%). Self Organizing Map obtained better results than K-modes techniques. The classification techniques improved their accuracy by about 5% in all techniques when clustering methods were applied. Zhang et al. [19] considered 5,538 road crashes that occurred in Florida, United States, in a period of 3 years, in which 51 are fatalities. Six different models were modeled in order to predict the injury severity of a road crash using the following methods: Ordered Probit model, Multinomial Logit model, K-Nearest Neighbor, Decision Tree, Random Forest, and Support Vector Machine. Random Forest presented the best overall prediction accuracy as well as the best prediction accuracy in the severe crashes followed by K-Nearest Neighbor. The Ordered Probit model and the Multinomial Logit model presented the lowest accuracy rates.

Li et al. [20] developed two models, Support Vector Machine and Ordered Probit, to predict crash injury severity considering 5,538 crashes that occurred in 326 freeway segments in the state of Florida (United States) from 2004 to 2006. The Support Vector Machine model presented a correct prediction of 48.8% against 44.0% for the Ordered Probit model. This way, the Support Vector Machine model performed better in predicting crash injury severity. 8,390 at-fault truck-involved crashes were analyzed by Hosseinzadeh et al. [21] to develop a Support Vector Machine model and Random Parameter Logit Model for injury severity pre- diction. Data between 2011 and 2014

on eight provinces of Iran were used: Isfahan, Qom, Qazvin, South Khorasan, Kerman, Mazandaran, Khuzestan, and the eastern district of Tehran. The Support Vector Machine model achieved a more accurate prediction than the Random Parameter Logit model in all modeled categories. Sensitivity, specificity, and Area Under the Curve were used to com- pare the models' performance. Fatigue and deviation to left were associated to fatal at-fault truck crashes.

Wang and Kim [22] considered 201,581 road crashes, with 0.47% of fatalities that occurred in Maryland, Unites States, in 2017 to identify the contributing factors to crash severity. Two different models were generated, a Multinomial Logit model and a Random Forest model, in which their prediction accuracies were compared (precision, recall, and F1 score). The test data were used for performance comparisons and sensitivity analysis. Random Forest presented a higher prediction accuracy than the Multinomial model. With the sensitivity analysis it was concluded that the Random Forest model was less sensitive than Multinomial Logit model. Collision type, occupant age, and speed limit were identified as the major contributing factors for crash severity. Ahmadi et al. [23] modeled the severity of rear-end crashes with three different algorithms: multinomial logit, mixed multinomial logit, and support vector machine, considering the rear-end crashes that occurred in the state of California, United States, between 2007 and 2011. Area under the curve (AUC) was applied to determine which variables best classify injury levels. Older and male drivers, driving in foggy and mountainous or rolling terrains, low light conditions, motorcycle riders and truck drivers, lower number of road lanes, improper turning, and driving under the influence were associated with more severe crashes. Multinomial logit and mixed multinomial logit presented similar prediction accuracy as the variables had few mixed effects. Support vector machine presented slightly better prediction accuracy.

In summary, the reviewed literature on crash injury severity showed that significant attention has been on crash severity modeling. The above literature reviews summarize different algorithms from two aspects: statistical learning and machine learning for crash severity prediction in different scenarios, such as, highway-rail grade crossings, analysis of rear-end crashes. One main problem for these literatures is that there is a significant trend towards neglecting the fact that crash datasets are acutely imbalanced. Overlooking this fact generally leads to weak classifiers for predicting the minority class (crashes with higher severity).

3. Methodology

This section will first introduce the unbalanced dataset problem, the potential solutions for solving the unbalanced dataset problem and then introduce four machine learning models: K-Nearest Neighbor (KNN), Logistic Regression (LR), Decision Tree (DT) and Random Forest (RF), with distinct classification logic.

3.1 Resampling Techniques

Many practical classification problems are imbalanced. The class imbalance problem happens if there are several more samples of some classes than others. In such cases, standard machine learning classifiers tend to be overwhelmed by the majority classes and overlook the minority one [24]. The performance of such classifiers in predicting the minority class decreases significantly. In order to overcome these issues, resampling approaches can be employed.

Mostly, there are two resampling approaches for handling imbalanced datasets: oversampling techniques and under-sampling techniques. Oversampling concerns techniques that increase the number of minority class samples until the dataset is balanced. A relatively recent and well-known oversampling technique is the Synthetic Minority Oversampling Technique (SMOTE) defined in the study by Chawla et al. [25]. SMOTE takes each minority class sample and creates new instances of the same class using k-nearest neighbors within a bootstrapping procedure. Moreover, SMOTE can be used for handling both continuous and categorical features. In the case of presence of categorical input factors, the technique is called SMOTE-NC, and it is introduced in [25]. Another random oversampling methods for minor classes (ROMC) is to sample data from the minor classes and then put the sampled data back to the minor classes, until the goal of oversampling data has been reached. The advantage of ROMC compared to SMOTE is that ROMC will only focus on the real data we have rather than the synthetic data generated and it could extract all the useful information from the limited dataset. Conversely, under-sampling techniques concern the methodologies to balance datasets by reducing the number of samples of the majority class. The RUMC approach is described in the study of Japkowicz [26] and Batista et al. [27] and consists of a random under-sampling of the majority class until the dataset is balanced.

Resampling techniques have the obvious advantage of being able to effectively handle imbalanced datasets by balancing them, thus defining training sets suitable for a satisfactory calibration of MLAs. There are also known drawbacks associated with the use of resampling techniques. The disadvantage with under-sampling such as RUMC is that it drops out potentially valuable data. Therefore, there is a possible information loss. The main disadvantage with oversampling such as SMOTE is that by creating very similar observations of existing samples, it makes overfitting likely. Indeed, with oversampling the minority class, the MLAs tend to learn too much from the specifics of the few examples, and they cannot generalize well. Moreover, factors appear to have a lower variance than they have. This project will implement oversampling for minor classes (ROMC) and under-sampling for major classes (RUMC). Through trying different oversampling size and under-sampling size distribution, this project will tend to optimize the data distribution and improve the accuracy for the test datasets, especially the prediction accuracy for the minor classes.

The RUMC technique involves randomly selecting examples from the majority class and removing them for training dataset. The majority class instances are discarded randomly until a balanced class distribution is reached (Figure 1). The ROMC technique involves randomly selecting examples from the minor classes and does not remove them from training dataset until the oversampling goal is reached.

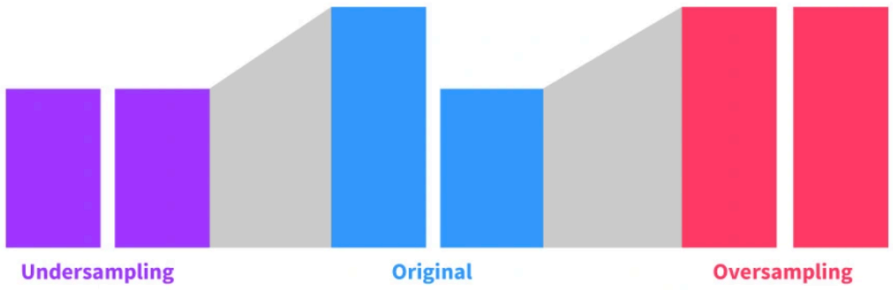


Figure 1. Random under sampling for majority class

3.2 Statistical Learning Algorithms

3.2.1 Multinomial Logistic Regression

Logistic Regression classifier was introduced by Berkson [29] and quickly became one of the most employed algorithms for classification purposes. LR firstly involves a linear multivariate

regression between the output (or dependent variable) and input factors (or independent variables). Subsequently, the output of the multivariate regression is passed by a logit function, leading to a numerical output within the range [0, 1]. Indeed, the logit function is a sigmoid function (i.e., S-shaped) that outputs a number between 0 and 1. Equation (2) below defines the logit function.

$$P(z) = \frac{1}{1+e^{-z}} \tag{2}$$

where: $P(z)$ is the probability of an occurrence (crash severity) that varies from 0 to 1; Equation (3) below defines z, which is the dependent variable of the linear multivariate regression.

$$z = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_m x_m \tag{3}$$

where: b_0 is a constant term; m is the number of independent variables; $x_i (i = 1,2,3 \ldots, n)$ represents the value of the i-th input factor; $b_i (i = 1,2,3 \ldots, n)$ is the regression coefficient assigned to the i-th input factor.

Once the probability $P(z)$ has been computed, the LR classifier makes its prediction $\hat{z}$ as follows:

$$\hat{z} = \begin{cases} class\ 0\ if\ P(z) < 0.5 \\ class\ 1\ if\ P(z) > 0.5 \end{cases} \tag{4}$$

Moreover, the regression coefficients determined in the logistic regression can be interpreted as a measure of the relative importance of the independent variables.

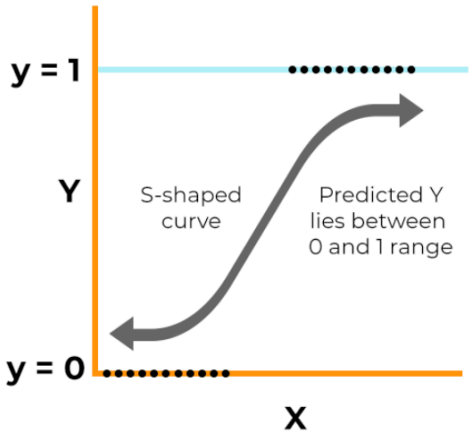


Figure 2. Logistic regression probability function

Multinomial logistic regression (often just called 'multinomial regression') is used to predict a nominal dependent variable given one or more independent variables. It is sometimes considered an extension of binomial logistic regression to allow for a dependent variable with more than two categories. Considering in this project, the severity level is not binary, so the multinomial logistic regression will be implemented here.

For the multiclass setting, we have severity level $y \in \{1, 2, \dots, K\}$, where K is the number of classes needed to be predicted. Instead of outputting a single probability value for the positive class, here the output is a vector of probabilities for each class:

$$f(x; w) = \begin{bmatrix} P(y = 1|x; w_1) \\ \vdots \\ P(y = K|x; w_K) \end{bmatrix} \quad (5)$$

where x is the feature considered to build the model; $P(y = K|x; w_K)$ is the probability of choosing one class.

The probability is calculated using a softmax format function:

$$P(y = K|x; w_K) = \frac{e^{w^T x}}{\sum_j^K e^{w_j^T x}} \quad (6)$$

where K is the number of classes needed to be predicted, x is the feature considered to build the model; w is the weight for each feature x. The final predicted class is the class with the highest probability. w can be estimated using the maximum likelihood estimation method.

3.3 Machine Leaning Algorithms

The used four models are supervised learning in nature which treat crash injury severity modeling as a classification problem: the crashes are classified into different categories according to their severity level. The machine learning models need to be trained by the labeled dataset, and then can predict the injury severity given the crash attributes. The model treats crash injury severity as a category variable. It could be interesting to see which model performs better in predicting crash severities.

3.3.1 K-Nearest Neighbor

KNN was first defined by Cover and Hart [28]. KNN is an algorithm that aims at classifying an observation by looking at the closest k observations contained in the feature space. The class that belongs to the majority of the k closest observations is taken as the class of the new observation. Therefore, KNN assigns to an unclassified sample point the class of the k nearest set of previously classified points. In the implementation of KNN, two key hyperparameters are important: the number of the k nearest neighbors and the distance function to employ to define a distance metric between observations into the feature space. The number of nearest neighbors should be tuned by

trying different values of k and finding the best accuracy of the classifier. The Euclidean distance is employed as a distance function to compute the distance between observations. The Euclidean distance d_{ij} between two points, i and j, is defined as:

$$d_{ij} = \sqrt[2]{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2} \quad (7)$$

where m is the dimension of the points (i.e., the number of independent variables); x_{ik} and x_{jk} are the values of the k-th independent variable for observations i and j, respectively.

3.3.2 Decision Tree (DT)

DT is a popular tool in machine learning which has been commonly used in classification tasks (which is noted as the classification tree approach). A flowchart-like structure is used in the DT for classification. Each internal node represents a test on a variable, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent the classification rules. DT with the closely related influence diagram is used as the visual and analytical decision support tool for the classification analysis.

For the crash severity modeling, each node in the DT represents a severity predictor and each branch represents one of the states of this variable. A tree leaf (or terminal node) denotes the expected injury severity depending on the information contained in the training dataset. When a new crash sample of the test dataset is obtained, a decision or prediction about the severity of the crash can be made by following the path in the tree from the root to a leaf, using the dividing variable values. The key to the decision tree is how to choose the best attributes. We hope that the samples in the branch nodes should belong to the same category as much as possible, that is, the purity of the nodes should be highest. In this project, the set maximum depth of the decision tree is 20.

3.3.3 Random Forest (RF)

RF is considered an ensemble learning method for solving classification, regression and other tasks. This method generates many classifiers and aggregates their results. Breiman [30] proposed this method as a prediction tool which consists of a collection of tree-structured classifiers with independent identically distributed random vectors. For a classification problem, the RF constructs a multitude of decision trees at the training time and outputs the class that is the mode of the classes,

see Figure 2. In RF, each node is split using the best among a subset of predictors randomly chosen at that node. Previous studies have reported that the RF performed well compared to many other classification models and suffer less overfitting issues. Two parameters need to be decided in the RF, which are the number of trees to grow and number of variables randomly sampled as candidates at each split. Here the number of tree to grow is set to be 50.

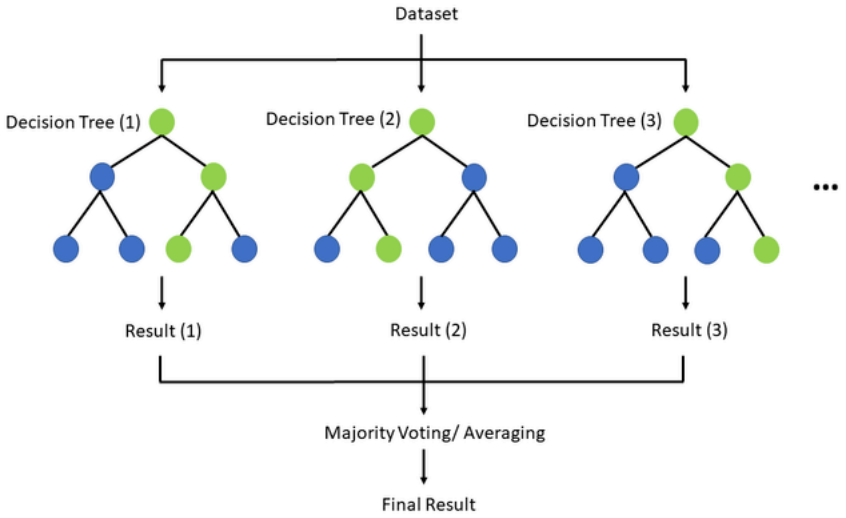


Figure 3. Random Forest structure

3.3 Performance Metrics

Several performance metrics are used for statistical learning methods and MLAs to be comprehensively evaluated and compared to each other. The used metrics are: accuracy, confusion matrix. The accuracy is calculated by the number of samples predicted correctly divided by the total number of samples in the testing dataset. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa.

4. Dataset

US-Crashes [32] currently contains data about 3 million instances of traffic crashes that took place within the contiguous United States. This is a countrywide car crashes dataset, which covers 49 states of the USA. The data is collected using two real-time data providers, namely “MapQuest

Traffic" and "Microsoft Bing Map Traffic", whose APIs broadcast traffic events (crashes, congestion, etc.) captured by a variety of entities - the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. Each crash record consists of a variety of intrinsic and contextual attributes such as location, time, weather, period-of-day, and relevant points-of-interest data (traffic signal, stop sign, etc.)

4.1 Attributes Introduction

The following Table 2, Table 3, Table 4, Table 5 and Table 6 summarize all the variables from five aspects traffic-related attributes, address-related attributes, weather-related attributes, point of interest attributes that related to the crash happening location and period of day attributes. Each table contains the variables name with their descriptions.

Table 2 Traffic-related attributes with description

Traffic-related attributes	**Description**
ID	This is a unique identifier of the crashes record.
Source	Indicates source of the crashes report (i.e. the API which reported the crashes.).
TMC	A traffic crashes may have a Traffic Message Channel (TMC) code which provides more detailed description of the event.
Severity	Shows the severity of the crashes, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the crashes) and 4 indicates a significant impact on traffic (i.e., long delay).
Start_Time	Shows start time of the crashes in local time zone.
End_Time	Shows end time of the crashes in local time zone.
Start_Lat	Shows latitude in GPS coordinate of the start point.
Start_Lng	Shows longitude in GPS coordinate of the start point.
End_Lat	Shows latitude in GPS coordinate of the end point.
End_Lng	Shows longitude in GPS coordinate of the end point.
Distance(mi)	The length of the road extent affected by the crashes.
Description	Shows natural language description of the crashes.

Table 3 Address-related attributes with description

Address attributes	**Description**
Number	Shows the street number in address field
Street	Shows the street name in address field.

Side	Shows the relative side of the street (Right/Left) in address field.
City	Shows the city in address field.
County	Shows the county in address field.
State	Shows the state in address field.
Zipcode	Shows the zipcode in address field.
Country	Shows the country in address field.
Timezone	Shows timezone based on the location of the crashes (eastern, central, etc.).

Table 4 Weather-related attributes with description

Weather attributes	**Description**
Airport_Code	Denotes an airport-based weather station which is the closest one to location of the crashes.
Weather_Timestamp	Shows the time-stamp of weather observation record (in local time).
Temperature(F)	Shows the temperature (in Fahrenheit).
Wind_Chill(F)	Shows the wind chill (in Fahrenheit).
Humidity(%)	Shows the humidity (in percentage).
Pressure(in)	Shows the air pressure (in inches).
Visibility(mi)	Shows visibility (in miles).
Wind_Direction	Shows wind direction.
Wind_Speed(mph)	Shows wind speed (in miles per hour).
Precipitation(in)	Shows precipitation amount in inches, if there is any.
Weather_Condition	Shows the weather condition (rain, snow, thunderstorm, fog, etc.).

Table 5 Point-of-interests attributes with description

Point-Of-Interest Attributes	**Description**
Amenity	A Point-Of-Interest (POI) annotation which indicates presence of amenity in a nearby location.
Bump	A POI annotation which indicates presence of speed bump or hump in a nearby location.
Crossing	A POI annotation which indicates presence of crossing in a nearby location.
Give_Way	A POI annotation which indicates presence of give_way sign in a nearby location.
Junction	A POI annotation which indicates presence of junction in a nearby location.
No_Exit	A POI annotation which indicates presence of no_exit sign in a nearby location.
Railway	A POI annotation which indicates presence of railway in a nearby location.
Roundabout	A POI annotation which indicates presence of roundabout in a nearby location.
Station	A POI annotation which indicates presence of station (bus, train, etc.) in a nearby location.

Stop	A POI annotation which indicates presence of stop sign in a nearby location.
Traffic_Calming	A POI annotation which indicates presence of traffic_calming means in a nearby location.
Traffic_Signal	A POI annotation which indicates presence of traffic_signal in a nearby location.
Turning_Loop	A POI annotation which indicates presence of turning_loop in a nearby location.

Table 6 Period of day attributes with description

Period of day attributes	**Description**
Sunrise_Sunset	Shows the period of day (i.e. day or night) based on sunrise/sunset.
Civil_Twilight	Shows the period of day (i.e. day or night) based on civil twilight.
Nautical_Twilight	Shows the period of day (i.e. day or night) based on nautical twilight.
Astronomical_Twilight	Shows the period of day (i.e. day or night) based on astronomical twilight.

4.2 Exploratory Data Analysis

This section will include how to do the data cleaning, deal with NAN data, select variables to build the models and implement several exploratory data analyses based on the US accidents dataset.

This following Table 7 summarized the details of columns of crash dataset like data type, unique count, percentage of NAN count and shows the action that needs to be done during EDA. We can see that the variables country and turning loop (which is A POI annotation which indicates presence of turning_loop in a nearby location) are deleted. Because throughout the whole dataset, these two variables just have one unique value. For other variables' NAN values, if the percentage of NAN values is smaller than 5%, NAN will be deleted. The variable like End_lat, End_lng, TMC, Number, wind_chill and precipitation that have too many NAN values are deleted and will not be considered into building the models. As for the variable wind_speed, it has 14.8% percentage of NAN values. The wind_speed NAN values are replaced by the closest data point that is defined by the Euclidian distance.

Figure 4 shows the total number of crashes in each state, where we can see CA and FL have the highest number of crashes in all states. Figure 5 shows the top ten cities with highest number of crashes, where Miami, Orlando and Los Angeles are the first three cities that have highest number of crashes.

Table 7 Summary of the variables with the data type, unique count and NAN count

	Data Type	Unique count	NAN Count	Percent(NAN)
ID	object	2974335	0	0.000000
Source	object	3	0	0.000000
TMC	float64	21	728071	24.478446
Severity	int64	4	0	0.000000
Start_Time	object	2743101	0	0.000000
End_Time	object	2761499	0	0.000000
Start_Lat	float64	1002359	0	0.000000
Start_Lng	float64	985099	0	0.000000
End_Lat	float64	298605	2246264	75.521554
End_Lng	float64	302906	2246264	75.521554
Distance(mi)	float64	12847	0	0.000000
Description	object	1597506	1	0.000034
Number	float64	37398	1917605	64.471722
Street	object	160715	0	0.000000
Side	object	3	0	0.000000
City	object	11685	83	0.002791
County	object	1713	0	0.000000
State	object	49	0	0.000000
Zipcode	object	377152	880	0.029586
Country	object	1	0	0.000000
Timezone	object	4	3163	0.106343
Airport_Code	object	1995	5691	0.191337
Weather_Timestamp	object	470781	36705	1.234057
Temperature(F)	float64	827	56063	1.884892
Wind_Chill(F)	float64	971	1852623	62.286965
Humidity(%)	float64	100	59173	1.989453
Pressure(in)	float64	994	48142	1.618580
Visibility(mi)	float64	81	65691	2.208595
Wind_Direction	object	24	45101	1.516339
Wind_Speed(mph)	float64	147	440840	14.821464
Precipitation(in)	float64	256	1998358	67.186716
Weather_Condition	object	120	65932	2.216697
Amenity	bool	2	0	0.000000
Bump	bool	2	0	0.000000
Crossing	bool	2	0	0.000000
Give_Way	bool	2	0	0.000000
Junction	bool	2	0	0.000000
No_Exit	bool	2	0	0.000000
Railway	bool	2	0	0.000000
Roundabout	bool	2	0	0.000000
Station	bool	2	0	0.000000
Stop	bool	2	0	0.000000
Traffic_Calming	bool	2	0	0.000000
Traffic_Signal	bool	2	0	0.000000
Turning_Loop	bool	1	0	0.000000
Sunrise_Sunset	object	2	93	0.003127
Civil_Twilight	object	2	93	0.003127
Nautical_Twilight	object	2	93	0.003127
Astronomical_Twilight	object	2	93	0.003127

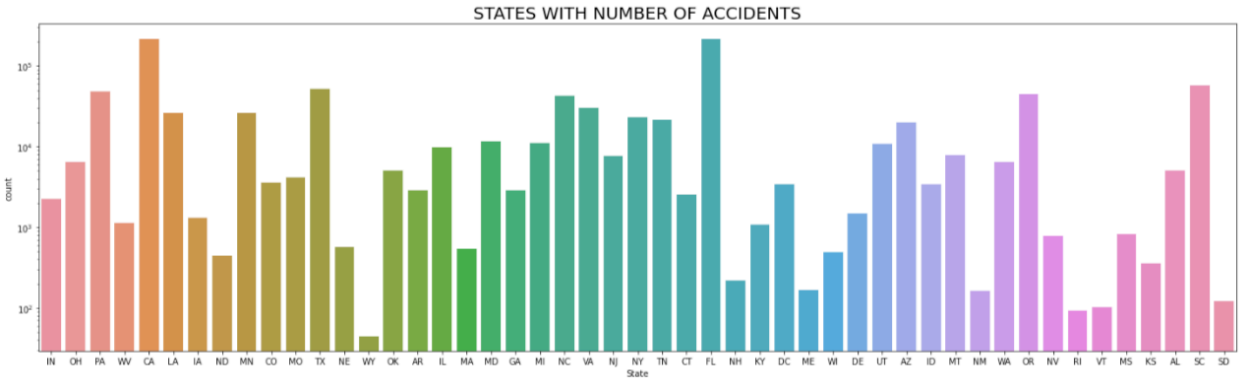


Figure 4. Number of crashes comparisons for all states

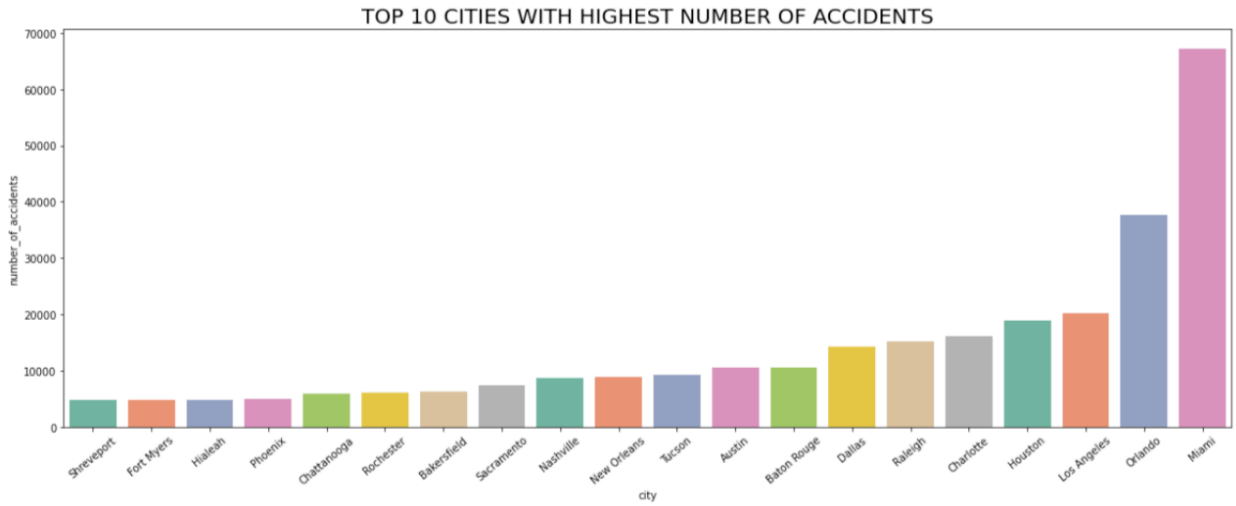


Figure 6. Top 10 cities with highest number of crashes

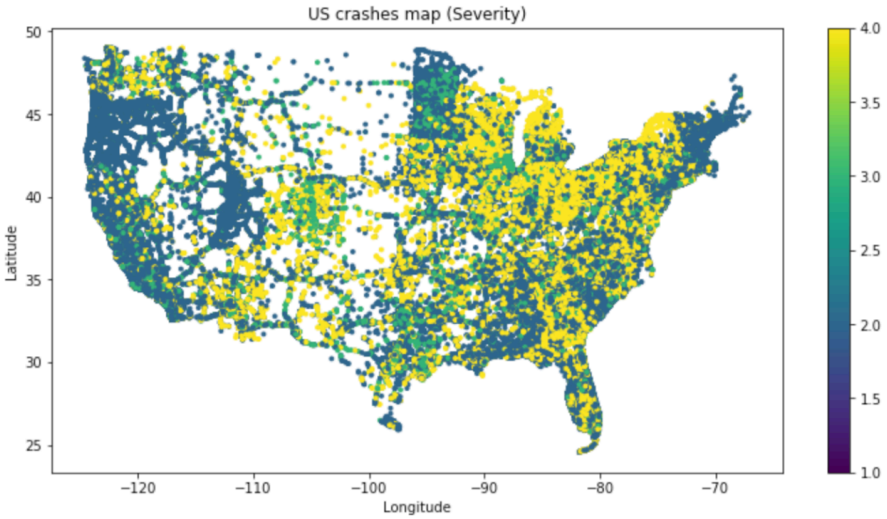


Figure 7. Scatter plot across the US for different severity level

The Figure 7. shows the crashes map and color represents the severity level. It can be observed that the east coast, west coast, and Michigan areas have a lot of crashes. This is because these areas are most industrialized and have large population density. Whereas central America has a few numbers of crashes. It is also observed that the crashes severity level is high in Michigan which is highlighted by yellow color. Figure 8 and Table 8 show the distribution of the different severity level, where we can see that class 1 and class 4 are the minor classes compared with class 2 and class 3. The data distribution for each type of severity is uneven.

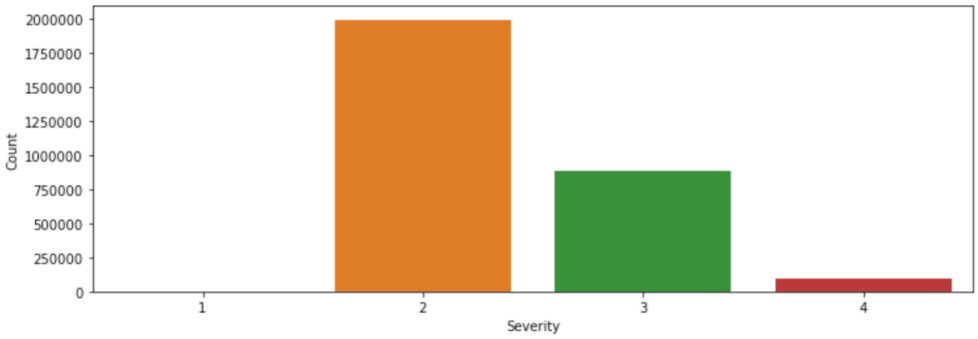


Figure 8. Scatter plot across the US for different severity level

Table 8. Count of each type of severity

Severity level	1	2	3	4
Count of observations	968	1993410	887620	92337

In Figure 9 and table 9, I have created a frame listing the total number of crashes reported by different sources and plotted a stacked bar chart and a pie chart. It can be observed that MapQUest reported the maximum number of crashes as represented by the blue color in the pie chart. The bar chart shows the total number of crashes reported by each source and also the number of crashes with different severity levels is highlighted in the bar chart. Orange shows the total number of crashes with severity level 2, which is maximum, green shows severity 3, red shows severity 4 and blue shows severity 1, which cannot be seen in the bar plot because of very less number of crashes with severity 1. In this entire report, the colors for the severity level are retained whenever a stacked bar chart is plotted.

Table 9. Severity distribution for three types of data sources

	Severity 1	Severity 2	Severity 3	Severity 4	Total
MapQuest	958	1444124	754247	4769	2204098
Bing	0	525341	115197	87533	728071
MapQuest-Bing	10	23945	18176	35	42166

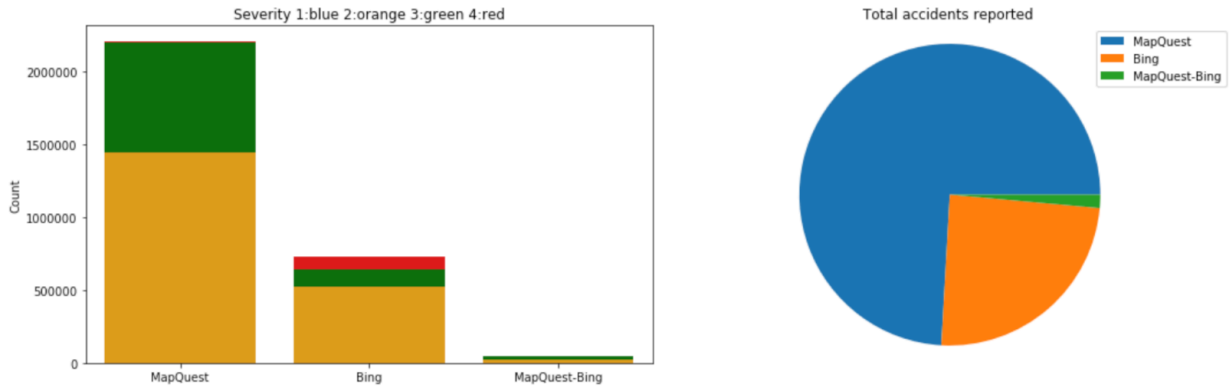


Figure 9. Severity distribution for three types of data sources

Figure 10. shows the crashes on right and left sides of the lane. The total number of crashes is more on the right side as shown by the blue region in the pie chart. From the stacked bar plot, it can be observed that the number of crashes with severity levels 2 and 3 is considerable on the right side but most of the crashes on the left side are with severity level 2. Overall, the crashes with severity level 2 is more compared to other severity levels on both right and left lanes.

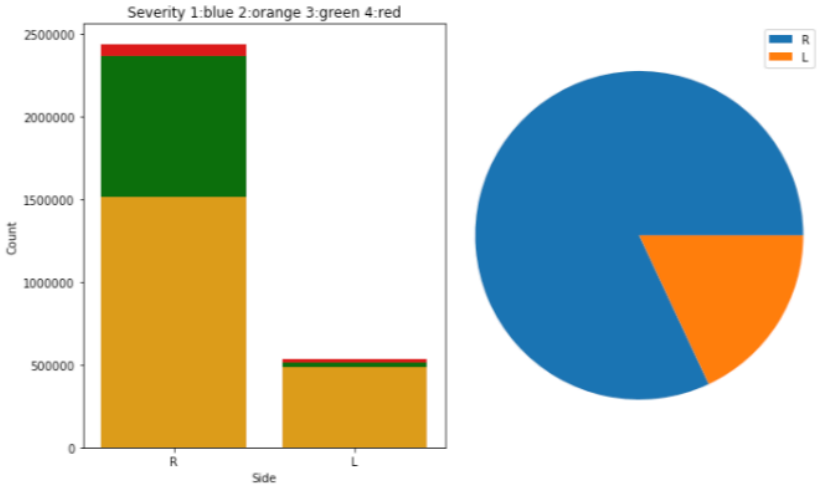


Figure 10. Severity distribution for three types of data sources

Figure 11. shows the box plot of the road extent such as distance and duration affected by crashes. It can be observed that there are so many outliers for the distance box plot for all level if severity. For the duration box plot, there are many outliers for severity 1. We should delete these outliers for further analysis.

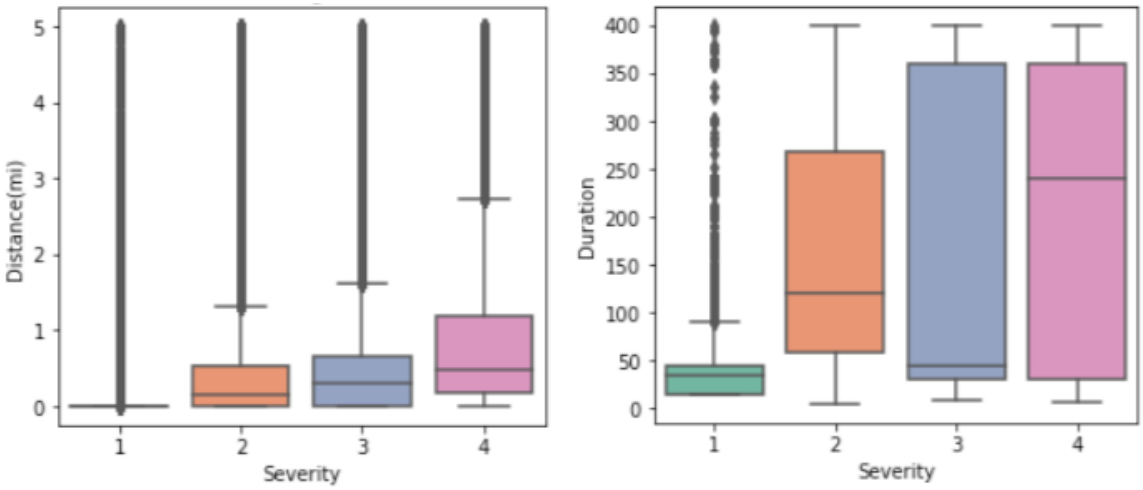


Figure 11. Severity box plot for distance and duration

Figure 11. shows the temperature effects on severity. Most of the crashes are between 10 F to 110 F and the violin plot shows that severity levels 2 and 3 have long tails because of outliers.

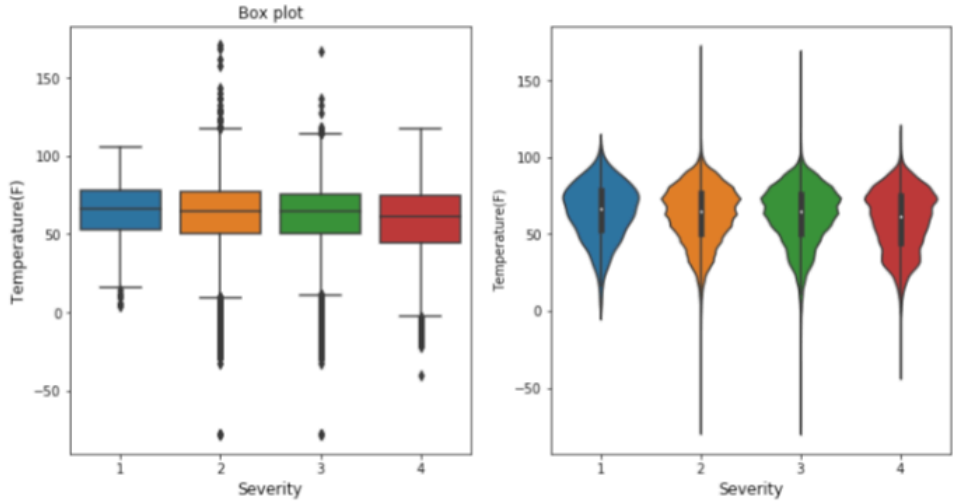


Figure 11. Severity box plot for temperature

Table 10 shows the statistics of humidity, pressure, wind speed, and precipitation. It is observed that 75% of the crashes have precipitation below 0.0 and occur around the pressure of 29.8 inches. The pressure may not have some intuitive significant effects on crashes. There are so many precipitation data missing. Based on the statistics, only humidity and windspeed are visualized and used for further analysis.

Table 10. Statistics analysis for humidity, pressure, wind speed and precipitation

	Humidity(%)	Pressure(in)	Wind_Speed(mph)	Precipitation(in)
count	2.915161e+06	2.926192e+06	2.533494e+06	975977.000000
mean	6.540542e+01	2.983189e+01	8.298063e+00	0.020495
std	2.255677e+01	7.213810e-01	5.138547e+00	0.235770
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000
25%	4.900000e+01	2.982000e+01	4.600000e+00	0.000000
50%	6.700000e+01	2.998000e+01	7.000000e+00	0.000000
75%	8.400000e+01	3.011000e+01	1.040000e+01	0.000000
max	1.000000e+02	3.304000e+01	8.228000e+02	25.000000

Figure 12 shows the box plot and leaf plot of humidity at different severity levels. There are no outliers in the humidity data.

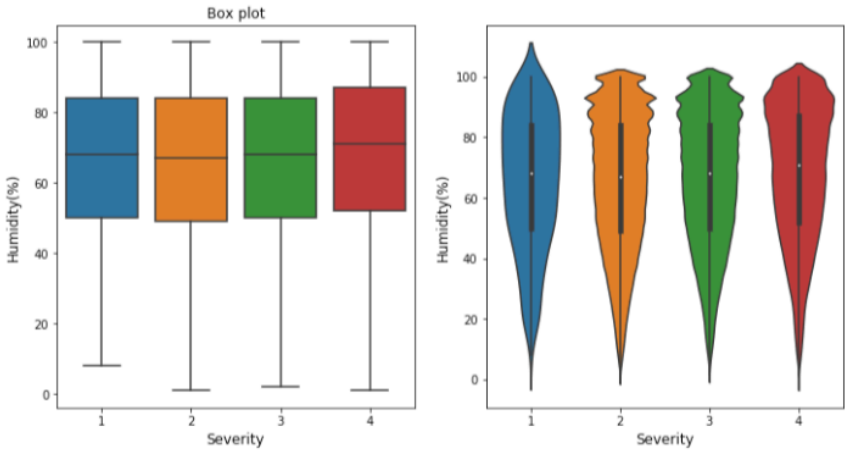


Figure 12. Severity box plot for humidity

Figure 13 shows that 99% of wind speed lie below 23 miles per hour and the maximum value is 828 miles per hour. It says that rest 1% of the data is varying a lot ranging from 23 to 822 and hence there are outliers in the box plot. Figure 14 shows the box and violin plot of wind speed after removing the data above 23 miles per hour. It can be observed that there are a few numbers of outliers.

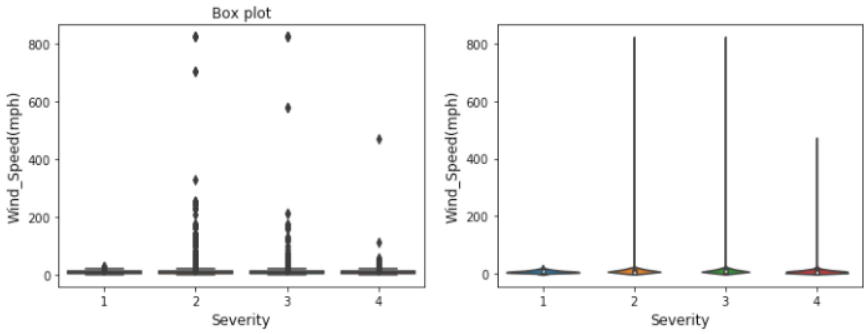


Figure 13. Severity box plot for wind speed

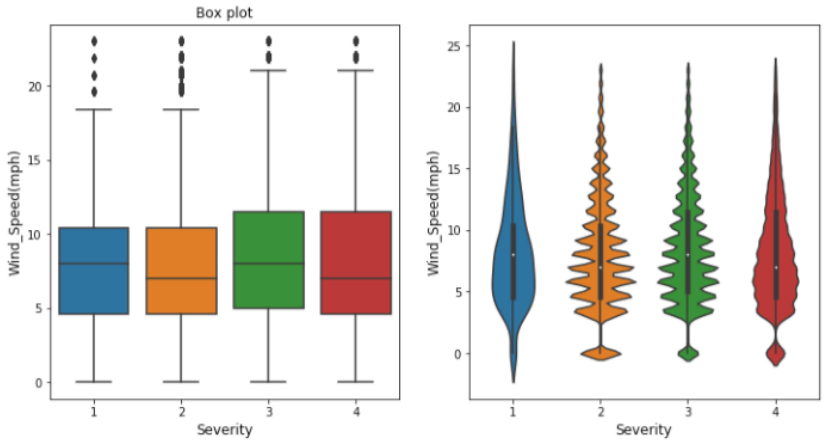
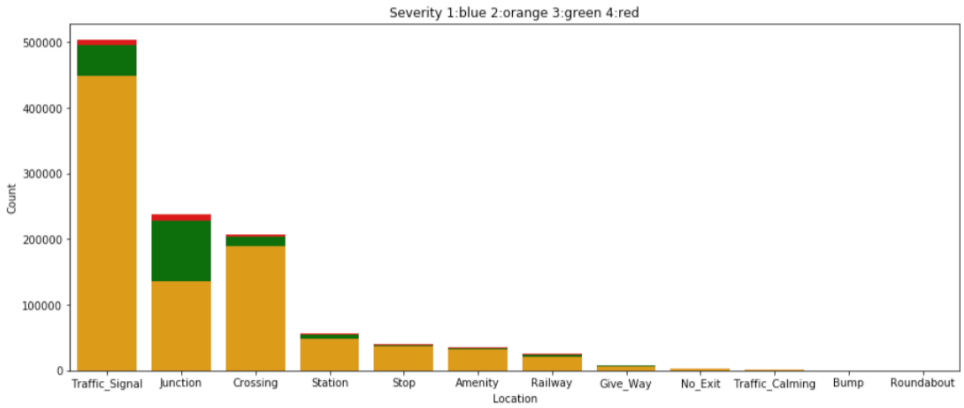


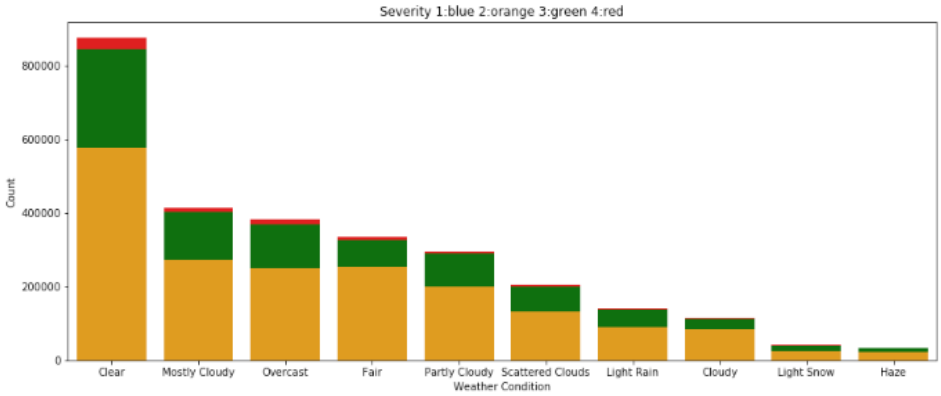
Figure 14. Severity box plot for wind speed after cleaning the outliers

Figure 15 shows some exploratory analysis about data distribution of each severity level based on different scenarios considering the location where a crash happed, the weather conditions, the years, the months, the weekdays, day or night and during a day (24h).

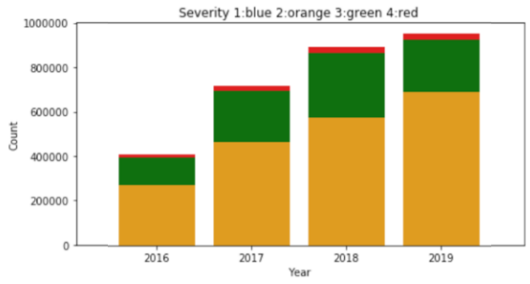
In figure 15(a) We can see that most of the crashes happened near traffic signal, junction and crossing, where the crashes near the junction have more severity 3 fraction compared with traffic signal and crossing. In Figure 15(b), we can see that most of the crashes happened during a clear whether, which makes sense because on average the weather should be clear at most of the time. Followed by weather condition is cloudy and overcast, which will have some impacts on the traffic conditions and crashes. In figure 15(c), we observe that the total number of crashes is increasing from year 2016 to year 2019 because more and more people tend to move to denser areas that may have a higher probability of leading to more crashes. In Figure 15(d), the data trend tells us that the total number of crashes is increasing from January to December. This maybe because there would be more extreme weather like snow during winter, but the exposure for winter should be also considered here. In Figure 15(e), it is pretty clear that weekdays will have more crashes compared with weekends, which may be due to more work-based trips generated during weekdays. In figure 15(f), it shows that there are more crashes happened during day compared with night and the crashes happened at night have a larger fraction of severity level 4 (most severe one). In figure 15(g), we can see that morning rush hour and evening rush hour will have more crashes compared to other period time of a day, which may be related to more work trips generated during that run hour period.



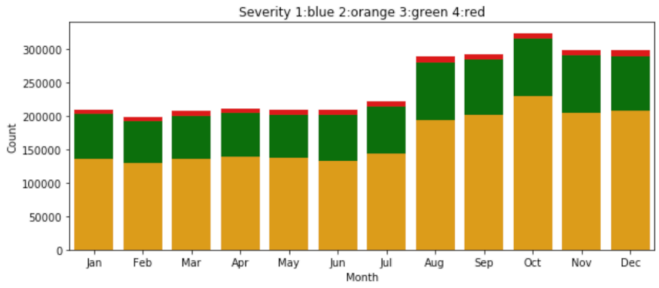
(a) Distribution over location



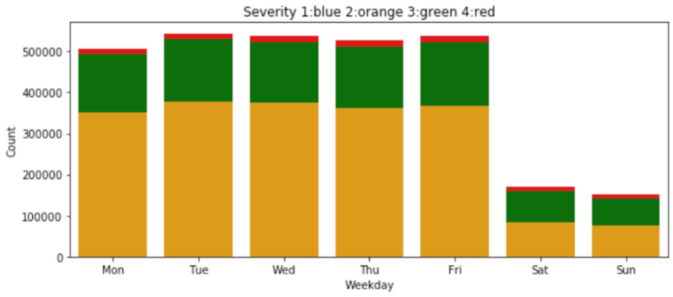
(b) Distribution over weather conditions



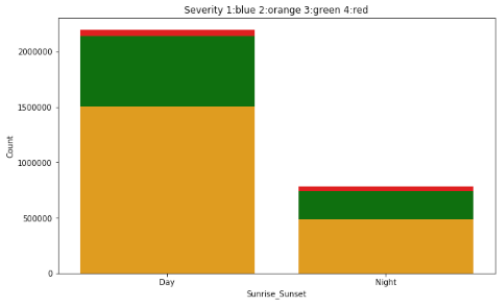
(c)Distribution over years



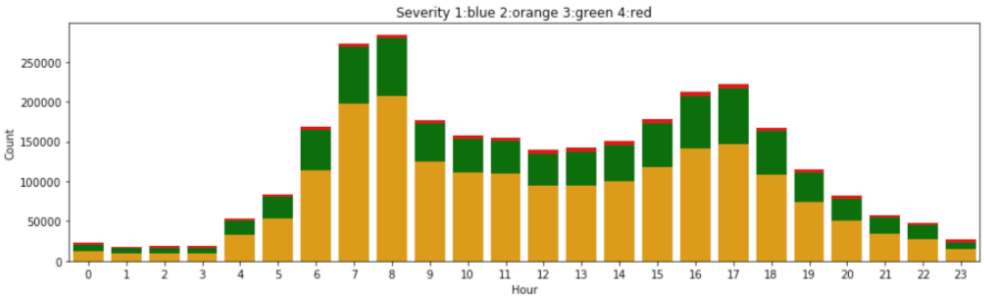
(d) distribution over months



(e) Distribution over weekdays



(f) Distribution over day or night



(g) distribution over a day (24h)

Figure 15. Exploratory analysis based on different conditions

After lots of EDA, this following Table 11 shows the summary of the data after cleaning.

Table 11 Summary of variable after cleaning

	Data Type	Unique count	NAN Count	Percent(NAN)
TMC	int64	21	0	0.000000
Severity	int64	4	0	0.000000
Start_Lat	float64	1002359	0	0.000000
Start_Lng	float64	985098	0	0.000000
Side	int64	2	0	0.000000
City	object	11685	83	0.002791
State	object	49	0	0.000000
Temperature(F)	float64	827	0	0.000000
Humidity(%)	float64	100	0	0.000000
Wind_Speed(mph)	float64	39	0	0.000000
Weather_Condition	int64	120	0	0.000000
Amenity	float64	2	0	0.000000
Crossing	float64	2	0	0.000000
Junction	float64	2	0	0.000000
Railway	float64	2	0	0.000000
Station	float64	2	0	0.000000
Stop	float64	2	0	0.000000
Traffic_Signal	float64	2	0	0.000000
Sunrise_Sunset	int64	2	0	0.000000
Day	int64	365	0	0.000000
Weekday	int64	7	0	0.000000
Hour	int64	24	0	0.000000
Minute	float64	1440	0	0.000000
Duration	float64	17900	0	0.000000

5 Experiments and Results

This section includes the final studied area, the data variables correlation analysis, the experiments setting for different scenarios and the final results.

5.1 Studied Area

The crashes dataset is around 3 million. From Exploratory Data Analysis (EDA), the state of California has more crashes compared to other states and the San Francisco bay area has around one-fifth of crashes. Considering the limited computational power, statistical learning and four machine learning models are built for the California bay area.

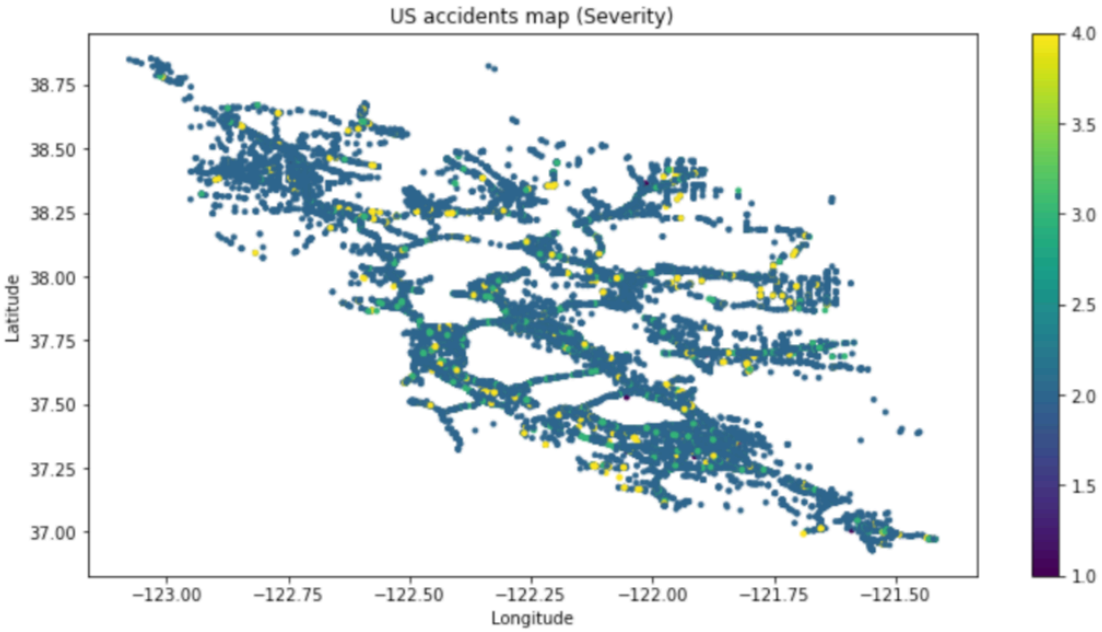


Figure 16. Scatter plot of bay area for different severity level

Figure 16 is a scatter plot of bay area for different severity level. Table 12 summarizes the count of each severity level in bay area, where we could see that the data distribution is very uneven. The severity 1 and severity 4 are the minor classes that we need to oversample and severity 2 and severity 3 are the majority classes that we need to do under-sampling.

Table 12. Count of different severity level in bay area

Severity level	1	2	3	4
Count of observations	72	90870	45704	1002

5.2 Correlation Analysis

Figure 17 shows correlation analysis of variables, where we can see for severity, weather conditions, start_lng, sunset_sunrise, side, duration and weekdays are the variables having higher correlation with severity, compared with other variables. In general, severity does not show strong correlation with any of the feature, so building some machine learning models might be helpful to capture some underlying relationship of data to predict the severity level.

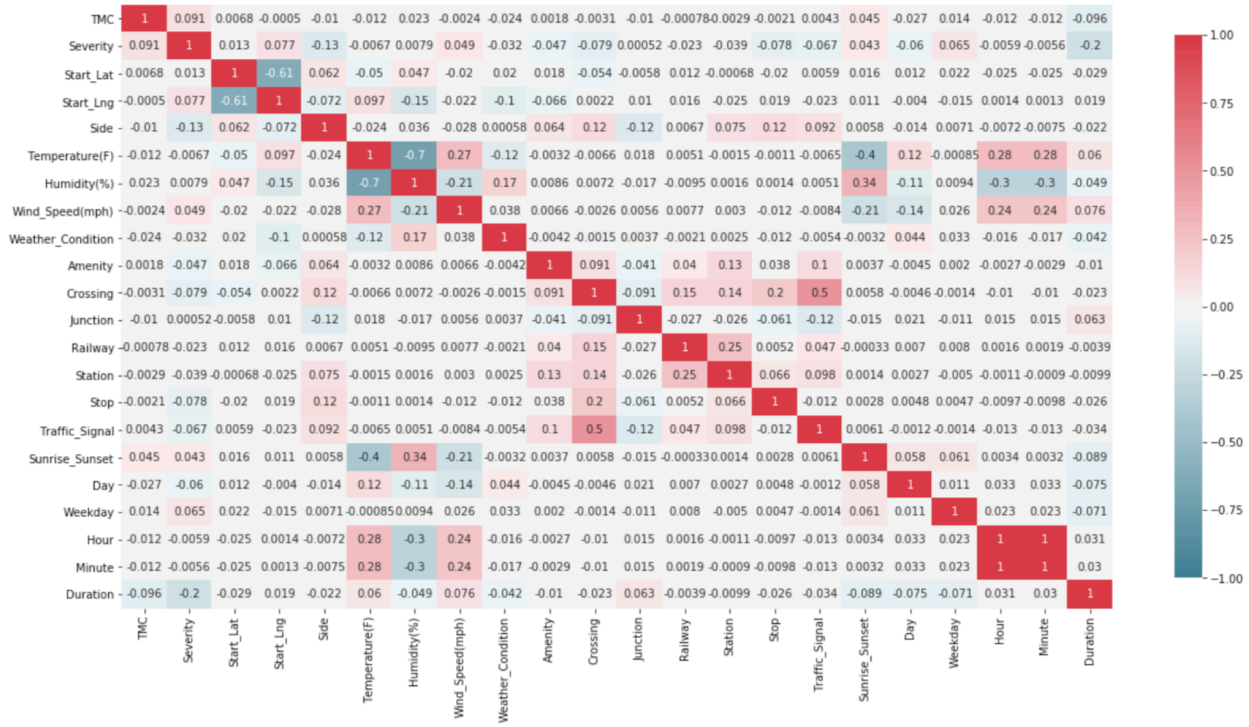


Figure 17. Heatmap of the correlation analysis of variables

5.3 Experiments Settings

Table 13 shows the experiments setting for four different scenarios. The testing data are all the same for the four different scenarios, where we have 9 samples for severity 1, 9021 samples for severity 2, 4620 samples for severity 3 and 115 samples for severity 4. The testing data distribution for the four scenarios is shown in Figure 18.

The bay area dataset is shown in Table 12 above. The training dataset size is 13000 for the four scenarios. The main difference between each scenario lies on the training data distribution. Setting 1 will randomly sample 13000 from all the bay area dataset in Table 12.

Setting 2 will first oversample 5000 samples for each of the four severity types from original bay area dataset, so there will be 20000 samples in total and four severity types are even distributed. Then setting two will sample 13000 data randomly from the 20000 samples with even distributed data.

From the original bay area data, setting 3 will oversample minor class: severity 1 and severity 4 to the size of 3000 and 3000 respectively, then under-sample majority class of severity 2 and severity 3 to a size of 7000 and 7000 respectively. Then from the 20000 datapoints, it will random sample 13000 data for training of setting 3.

From the original bay area data, setting 4 will oversample minor class: severity 1 and severity 4 to the size of 500 and 2000 respectively, then under-sample majority class of severity 2 and severity 3 to a size of 8250 and 8250 respectively. Then from the 20000 datapoints, it will random sample 13000 data for training of setting 4.

Table 13. Experiments setting for different scenarios

Settings	Test data	Training data size	Train data distribution
1	1 2 3 4 Severity count in test_data 9 9021 4620 115	13000	Randomly sample 13000 from bay area data for training: 1 2 3 4 Severity count in train_data 4 8616 4298 82
2	1 2 3 4 Severity count in test_data 9 9021 4620 115	13000	Oversample minor class and under-sample majority class all to 5000, then sample 13000 from 20000: 1 2 3 4 Severity count in train_data 3254 3283 3229 3234
3	1 2 3 4 Severity count in test_data 9 9021 4620 115	13000	Oversample minor class and under-sample majority class all to 3000, 7000, 7000, 3000 for severity of 1,2,3,4 respectively: then sample 13000 from 20000: 1 2 3 4 Severity count in train_data 1948 4589 4531 1932
4	1 2 3 4 Severity count in test_data 9 9021 4620 115	13000	Oversample minor class and under-sample majority class all to 500, 8250, 8250, 2000 for severity of 1,2,3,4 respectively: then sample 13000 from 20000: 1 2 3 4 Severity count in train_data 330 5706 5689 1275

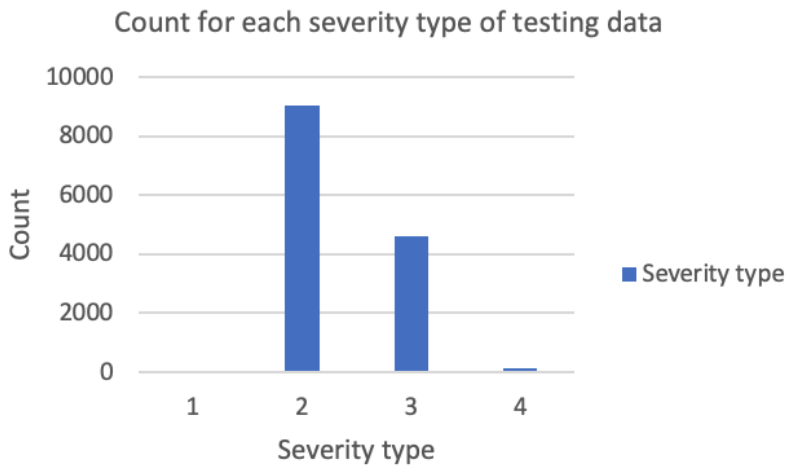


Figure 18 Testing data setting for four scenarios

The training data distribution for the four settings are shown in Figure 19, where we could see:
(1) data distribution in setting 1 is really unbalanced for the four severity types;

(2) data distribution in setting 2 is really pretty balanced for the four severity types;

(3) data distribution in setting 3 is really not very balanced for the four severity types but it is better than (1), the counts for minor classes have been reduced considering if we oversample the minor class and under-sample the major class to the same size, it will make the overall accuracy to be decreased and may encounter overfitting issues;

(4) data distribution in setting 4 is still not very balanced for the four severity types but it is better than (1) and more unbalanced compared with (3); the counts for minor classes severity 1 and minor class severity 4 are sampled differently because in original datasets, the count of severity 1 is much smaller than that of severity 4. If we oversample the severity 1 and severity 4 to the same size, it will encounter overfitting issues.

Figure 19 Training data setting for four scenarios

5.4 Results Analysis

Figure 20 summarizes the accuracy of KNN model in each setting. For each plot in Figure 20, the x-axis is the number of K-nearest neighbors, which is a hyper-parameters needed to be optimized. We can see for all of the four settings, the best hyper-parameters of K is all equal to 2 because when k=2, it has the highest testing accuracy. So for the following model performance comparisons, the KNN results shown in Figure 21 are using K =2.

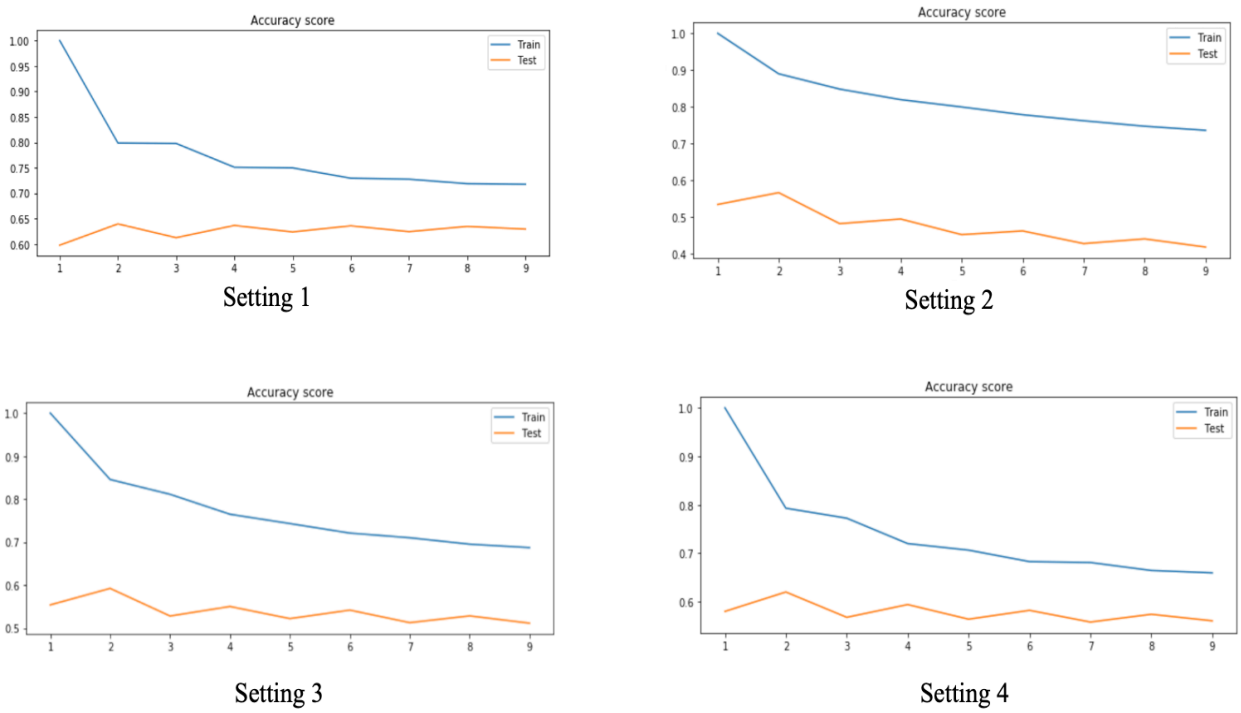


Figure 20 Summary of the accuracy of KNN model in each setting

Figure 21 is a summary of the prediction accuracy of each model in each setting. The higher the prediction accuracy, the better the model performance would be. We could see that the ranking of the model performance from higher to lower would be random forest, decision tree, KNN, multinomial logistic regression for all of the four settings. After applying the ROMC and RUMC methods to deal with the unbalanced issues in the dataset, we can observe from Figure 20 that the best accuracy in setting 2 and best accuracy in setting 3 is decreased. This is because when we do the RUMC and ROMC, we oversample too many minor classes while decreasing the effects of the majority classes in the whole datasets. This leads to a decrease in prediction accuracy. In setting 4, the overall accuracy for setting 4 is similar to overall accuracy of setting 1, while the minor classes perdition accuracy is higher than setting 1.

In the analysis later, the project will explain how the prediction accuracy for each severity is chaining in different settings.

	Train	Test
Logistic	0.669154	0.664657
KNN	0.750923	0.636469
Decision tree	0.976923	0.775663
Random forest	1.000000	0.816055

Setting 1

	Train	Test
Logistic	0.495308	0.312169
KNN	0.889923	0.566727
Decision tree	0.956154	0.683037
Random forest	1.000000	0.752488

Setting 2

	Train	Test
Logistic	0.510889	0.544642
KNN	0.781500	0.569197
Decision tree	0.935500	0.731638
Random forest	1.000000	0.790556

Setting 3

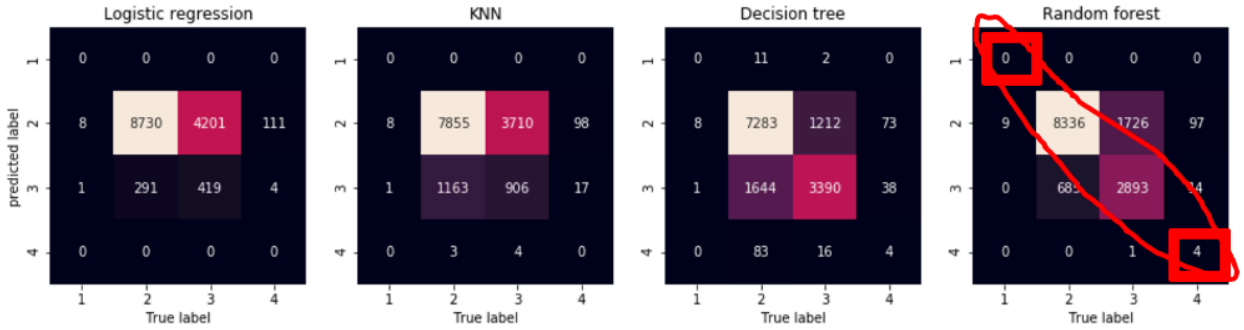
	Train	Test
Logistic	0.588605	0.603414
KNN	0.792846	0.619906
Decision tree	0.937210	0.777116
Random forest	1.000000	0.814457

Setting 4

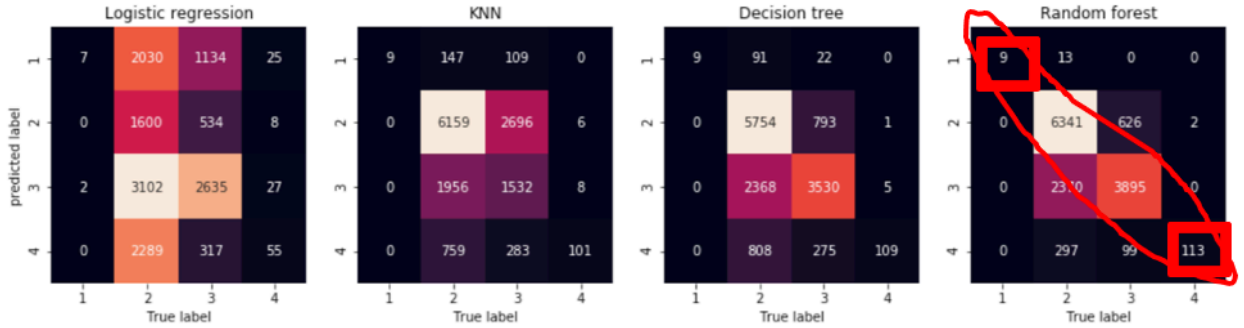
Figure 21 Summary of the accuracy of each model in each setting

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in a predicted class while each column represents the instances in a true class, or vice versa.

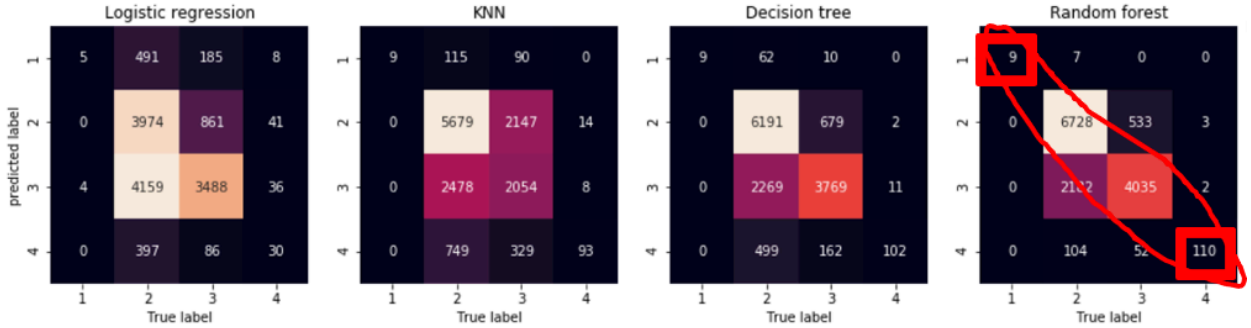
In the testing data, there are 9 samples for severity 1, 9021 samples for severity 2, 4620 samples for severity 3 and 115 samples for severity 4. In the setting 1, we could see that there is no prediction is correct for severity 1 and only 4 are predicted correctly for severity 4. After applying the RUMC and ROMC methods to change the data distribution, from setting 2 and setting 3 we could observe that the number of correct predictions for severity 1 and severity 4 has increased a lot while the number of correct predictions for severity 2 and severity 3 decrease a lot. In setting 4, when we try to apply the RUMC and ROMC methods, we need to find and test a proper oversampling size for minor class and a proper size under sampling size for minor classes, considering the original data distribution in Table 12 and avoiding making too large overfitting issue. The overall accuracy for setting 4 is similar to overall accuracy of setting 1, while the minor classes perdition accuracy is higher than setting 1.



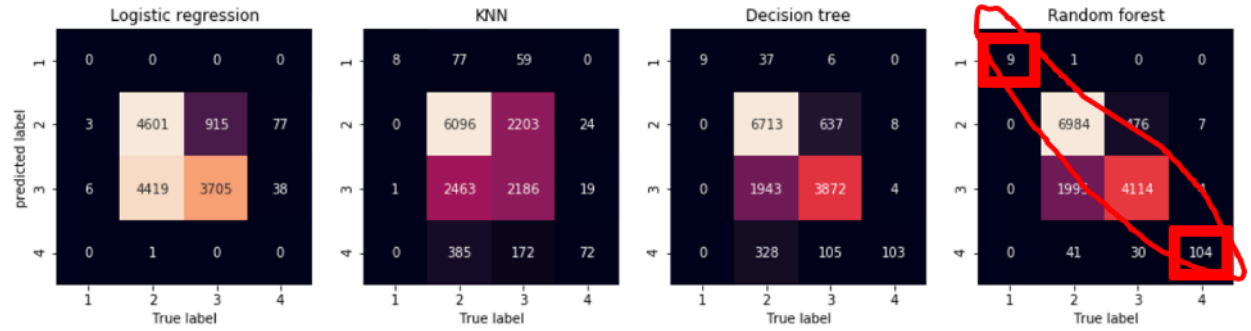
Setting 1



Setting 2



Setting 3



Setting 4

Figure 22 Summary of the confusion matrix of each model in each setting

6. Conclusions:

The country-wide crashes severity can be accurately predicted with limited data attributes (location, time, weather, and POI). Spatial patterns are also very important. For small areas like street and zipcode, severe crashes are more likely to happen at places having more crashes while for larger areas like city and airport region, at places having less crashes. If a crash happens on Interstate Highway, there is a 2% chance that it will be a serious one, which is about 2.3 times of average and higher than any other street type. A crash is much less likely to be severe if it happens near traffic signal while more likely if near junction.

To be specific, for a given crashes, without any detailed information about itself, like driver attributes or vehicle type, this model is supposed to be able to predict the likelihood of this crashes being a severe one. The crashes could be the one that just happened and still lack of detailed information, or a potential one predicted by other models. Therefore, with the sophisticated real-time traffic crashes prediction solution developed by the creators of the same dataset used in this project, this model might be able to further predict severe crashes in real-time.

The model performance ranking from low to high will be: logistic regression; KNN; Decision Tree; Random Forest. Under-sampling and over sampling methods could help the classifiers to better predict the minor classes, with the expense of most class accuracy estimations. If under-sampling the majority class and over sampling the minor class to the same size, it could lower the overall testing accuracy while improving the prediction accuracy of e minor classes. Adjusting the under-sampling size and over sampling size could help improve the overall testing accuracy while maintaining an accurate prediction for minor class.

It is worth noting that the severity in this project is "an indication of the effect the crashes have on traffic", rather than the injury severity that has already been thoroughly studied by many articles. Another thing is that the final model is dependent on only a small range of data attributes that are easily achievable for all regions in the United States.

As for future work, we can try to apply the models on the country-wide dataset if the computational resources are permitted. If we compare the testing accuracy with the training accuracy, there is a

high probability that the models are having overfitting issues. Therefore, it is worth to try some other advanced methods like using KNN generating some new data while avoiding the overfitting issues. It is also worth to try the model on other dataset considering some human factors as independent variables to see how these subjective factors will affect the prediction severity predictions.

References:

[1] World Health Organization. Global Status Report on Road Safety 2018: Summary 2018; World Health Organization: Geneva, Switzerland, 2018.

[2] Chang, L.-Y., & Wang, H.-W. (2006). Analysis of traffic injury severity: An application of non-parametric classification tree techniques. Crashes Analysis & Prevention, 38(5), 1019–1027.

[3] Lin, M.-R., & Kraus, J. F. (2008). Methodological issues in motorcycle injury epidemiology. Crashes Analysis & Prevention, 40(5), 1653–1660.

[4] Abrari Vajari, M., Aghabayk, K., Sadeghian, M., & Shiwakoti, N. (2020). A multinomial logit model of motorcycle crash severity at Australian intersections. Journal of Safety Research, 73, 17–24.

[5] Al-Ghamdi, A. S. (2002). Using logistic regression to estimate the influence of crashes factors on crashes severity. Crashes Analysis & Prevention, 34(6), 729–741.

[6] Kaplan, S., & Prato, C. G. (2012). Risk factors associated with bus crashes severity in the United States: A generalized ordered logit model. Journal of Safety Research, 43(3), 171–180.

[7] Mokhtarimousavi, S., Anderson, J. C., Azizinamini, A., & Hadi, M. (2020). Factors affecting injury severity in vehicle-pedestrian crashes: A day-of-week analysis using random parameter ordered response models and Artificial Neural Networks. International Journal of Transportation Science and Technology, 9(2), 100–115.

[8] Rezaie Moghaddam, F., Afandizadeh, S., & Ziyadi, M. (2011). Prediction of crashes severity using artificial neural networks. International Journal of Civil Engineering, 9(1), 41–48.

[9] Singh, G., Sachdeva, S. N., & Pal, M. (2018). Comparison of three parametric and machine learning approaches for modeling crashes severity on non-urban sections of Indian highways. Advances in Transportation Studies, 45, 123–140.

[10] Zong, F., Chen, X., Tang, J., Yu, P., & Wu, T. (2019). Analyzing Traffic Crash Severity with Combination of Information Entropy and Bayesian Network. IEEE Access, 7, 63288–63302.

[11] Zong, F., Xu, H., & Zhang, H. (2013). Prediction for Traffic Crashes Severity: Comparing the Bayesian Network and Regression Models. Mathematical Problems in Engineering, 2013, 9.

[12] Cateni, S.; Colla, V.; Vannucci, M. A method for resampling imbalanced datasets in binary classification tasks for real-world problems. Neurocomputing 2014, 135, 32–41.

[13] He, H.; Garcia, E.A. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 2009.

[14] Japkowicz, N.; Stephen, S. The class imbalance problem: A systematic study. Intell. Data Anal. 2002.

[15] Fan, W. (David), Gong, L., Washing, E. M., Yu, M., & Haile, E. (2016). Identifying and quantifying factors affecting vehicle crash severity at highway-rail grade crossings: models and their comparison. Transportation Research Board 95th Annual Meeting, No. 16-2085.

[16] Mohamed, M.G., Saunier, N., Miranda-Moreno, L.F., Ukkusuri, S.V., 2013. A clustering regression approach: a comprehensive injury severity analysis of pedestrian-vehicle crashes in New York, US and Montreal, Canada. Saf. Sci. 54, 27–37. http://dx.doi. org/10.1016/j.ssci.2012.11.001.

[17] Al-Turaiki, I., Aloumi, M., Aloumi, N., & Alghamdi, K. (2016). Modeling traffic crashes in Saudi Arabia using classification techniques. In 2016 4th Saudi International Conference on Information Technology (Big Data Analysis), KACSTIT 2016. Institute of Electrical and Electronics Engineers Inc.

[18] Gopinath, V., Purna Prakash, K., Yallamandha, C., Krishna Veni, G., & Krishna Rao, D. S. (2017). Traffic crashes analysis with respect to road users using data mining techniques. International Journal of Emerging Trends & Technology in Computer Science, 6(3), 15–20.

[19] Zhang, J., Li, Z., Pu, Z., & Xu, C. (2018). Comparing prediction performance for crash injury severity among various machine learning and statistical methods. IEEE Access, 6, 60079–60087.

[20] Li, Z., Liu, P., Wang, W., & Xu, C. (2012). Using support vector machine models for crash injury severity analysis. Crashes Analysis and Prevention, 45, 478–486.

[21] Hosseinzadeh, A., Moeinaddini, A., & Ghasemzadeh, A. (2021). Investigating factors affecting severity of large truck-involved crashes: Comparison of the SVM and random parameter logit model. Journal of Safety Research.

[22] Wang, X., & Kim, S. H. (2019). Prediction and factor identification for crash severity: comparison of discrete choice and tree-based models. Transportation Research Record.

[23] Ahmadi, A., Jahangiri, A., Berardi, V., & Machiani, S. G. (2018). Crash severity analysis of rear-end crashes in California using statistical and machine learning classification methods. Journal of Transportation Safety and Security, 12(4), 522–546.

[24] Zhao, Y.; Cen, Y. Data Mining Applications with R; Academic Press: Cambridge, MA, USA, 2013; ISBN 9780124115118.

[25] Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. 2002, 16, 321–357.

[26] Japkowicz, N. The Class Imbalance Problem: Significance and Strategies. In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI), Las Vegas, NV, USA, 12–15 July 2010.

[27] Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newsl. 2004.

[28] Cover, T.M.; Hart, P.E. Nearest Neighbor Pattern Classification. IEEE Trans. Inf. Theory 1967, 13, 21–27.

[29] Berkson, J. Application of the Logistic Function to Bio-Assay. J. Am. Stat. Assoc. 1944.

[30] Breiman, L. Random forests. Mach. Learn. 2001.

[31] Larson, S.C. The shrinkage of the coefficient of multiple correlation. J. Educ. Psychol. 1931, 22, 45–55.

[32] Moosavi, Sobhan, et al. "A countrywide traffic crashes dataset." arXiv preprint arXiv:1906.05409 (2019).