Predicting the severity of a collision

Laura Dohle, October 15, 2020

1. Introduction

1.1. Background

In 2018, the United States had roughly 276 million vehicles in operation.[1] Out of those vehicles, 12 million were involved in crashes.[2] The United States is also among the countries with the highest rate of traffic-related fatalities per one million population.[3]

1.2. Problem

A lot of these accidents could be prevented if the drivers were more informed about the possible severity of car accident based on different factors. The focus of this project therefore is to predict the severity of an accident, using a number of features. For performance reasons, we will focus our efforts on accidents in the Seattle area, but this will allow us to draw conclusions which are relevant for other areas in the US as well.

1.3. Interest

Reliable severity prediction of possible car accidents can help commuters to drive with more foresight or change their travel plans accordingly. In addition, an accident severity prediction system can be interesting for insurance companies in order to prevent possible costs for accidents suffered by their members.

2. Data

2.1. Data source

The dataset was obtained from the Seattle Department of Transportation website.[4] The data was collected from 2004 to present for Seattle. The number of observations in the data is about 220,000 and each observation represents a traffic collision. The original data contains our

[1] https://www.statista.com/statistics/859950/vehicles-in-operation-by-quarter-united-states/, accessed on September 25, 2020.

[2] https://www.statista.com/statistics/192097/number-of-vehicles-involved-in-traffic-crashes-in-the-us/, accessed on September 25, 2020.

[3] https://www.statista.com/statistics/485456/road-fatalities-by-country/, accessed on September 25, 2020.

[4] Data obtained from https://data.seattle.gov/Land-Base/Collisions/9kas-rb8d, accessed October 12, 2020.

chosen label, i.e a severity code based on the severity of a collision, where 1 indicates just property damage and 4 indicates a fatality as highest impact of the collision. It also consists of 39 attributes.

2.2. Data cleaning

There were several steps necessary to clean the dataset and make it usable for the problem statement:

First, not all of the attributes in the dataset are relevant for the problem statement covered in this report. Therefore 11 features were removed from the report entirely:

- OBJECTID, INCKEY, SDOTCOLNUM and COLDETKEY were just different variations of unique identifiers, not important to the problem statement at hand.
- SDOT_COLDESC was a description of the collision, that directly duplicated the information given via SDOT_COLCODE.
- ST_COLDESC was a description of the collision, that directly duplicated the information given via ST_COLCODE.
- EXCEPTRSNCODE and EXCEPTRSNDESC was not described in the metadata and also only consisted of different forms of missing data.
- INCDATE was a duplicate of information also present in INCDTTM.
- STATUS and REPORTNO we removed since it they were not described in the metadata.

Second, we had to check and make sure that all data was in the correct format (int, float, text or other). It turned out that some columns were not in the correct data type:

- The multiclass categorical data of 'SDOT_COLCODE' needs to have type 'str' (object).
- The Boolean values 'INATTENTIONIND', 'UNDERINFL', 'PEDROWNOTGRNT', 'SPEEDING', 'HITPARKEDCAR' need to be changed to numeric values 0/1 instead of True/False and therefore be of type 'float' or 'int' to make them more usable for the machine learning section later on.
- Finally, 'INCDTTM' needs to be converted to the datatype 'datetime'.

Third, the attributes mentioned in the list above were also cleaned of any duplicate or missing data values in the same step:

- 'SDOT_COLCODE' had one missing value, so we dropped the relevant row
- For the Boolean attributes INATTENTIONIND, SPEEDING and PEDROWNOTGRNT people only entered 'Y' or did not enter anything at all, instead of entering 'N'. For those three attributes we replaced the 'Y' values with 1 and the NaN values with 0
- For the Boolean attribute 'UNDERINFL' there was a mix between numeric and string boolean values, as well as some missing values. We converted the 'Y' values into 1 and the 'N' values into 0 before having a closer look at the NaN values. Due to the fact that, if we look at the other boolean values that indicate a contributing factor to the accident, the number of TRUE(1) values is roughly between 2-15% and we already have 5% TRUE values for the 'UNDERINFL' attribute, we count the NaN values as FALSE(0).

Fourth, the date and time attribute INCDTTM was split into smaller date and time increments. Date and time attributes might be very relevant when trying to predict accident severity. For example, there might be larger severity accidents during rush hour when people are commuting than during general working hours. That is why it made sense to split the Datetime attribute into smaller increments: YEAR, MONTH, DAY, HOUR and WEEKDAY. After splitting INCDTTM we removed to original attribute from the data set.

Fifth, we had a look at the data that was not handled as part of before mentioned data cleaning to see if there were any outstanding measures that needed to be taken:

- ADDRTYPE had missing values that needed to be replaced. We can actually tell if a ADDRTYPE is a Block, Intersection or Alley by looking at the LOCATION attribute, as they are very related:
  - For “Blocks” the phrasing of the LOCATION is: "X BETWEEN Y AND Z"
  - For “Intersections” the phrasing of LOCATION is: "X AND Y"
  - For Alleys the LOCATION always has a NaN value, while LOCATION never has a NaN value for ADDRTYPE “Block” or “Intersection”.

  If we looked at the rows with ADDRTYPE NaN, all of them also had LOCATION NaN. Because LOCATION NaN always meant “Alley” we will assigned them as “Alleys”.

- Missing values for JUNCTIONTYPE, WEATHER and ROADCOND were replaced with the most frequent values of each attribute
- 9413 rows of ST_COLCODE were dropped because they were missing data
- The missing values for COLLISIONTYPE were replaced by taking them from the descriptions of ST_COLCODE for each row, as the information is very related.
- We derived the missing values for LIGHTCOND by having a look at the MONTH and HOUR data, which gave us an idea about sunrise and sunset for each collision event.
- SEGLANEKEY, CROSSWALKKEY, INTKEY, X, Y and LOCATION were dropped entirely because the majority of rows had missing data.
- The missing data for SEVERITYDESC was replaced by combining it with data from the columns FATALITY, SERIOUSINJURIES and INJURIES
- Missing values for the label SEVERITYCODE could then be replaced by having a look at SEVERITYDESC

2.3. Initial feature selection

After data cleaning, there were 212,112 samples and 26 attributes in the data. Upon examining each attribute, it was clear that there was some further redundancy.

- ST_COLCODE indicates the type of collision and is therefore almost identical with COLLISIONTYPE, so I decided to drop ST_COLCODE.

- SEVERITYDESC is the more detailed description version of SEVERITYCODE. I therefore decided to drop SEVERITYDESC to decrease redundancy.

The remaining 24 attributes were as follows:

#	Attribute	Description	Example value
1	ADDRTYPE	Collision address type: Alley, Block, Intersection	Block
2	COLLISIONTYPE	Type of collision	Cycles
3	PERSONCOUNT	The total number of people involved in collision	1

4	PEDCOUNT	The number of pedestrians involved in collision	1
5	PEDCYLCOUNT	The number of bicycles involved in collision	1
6	VEHCOUNT	The number of vehicles involved in collision	1
7	INJURIES	The number of total injuries in collision	1
8	SERIOUSINJURIES	The number of serious injuries in collision	1
9	FATALITIES	The number of fatalities in collision	1
10	JUNCTIONTYPE	Category of junction at which collision took place	Driveway Junction
11	SDOT_COLCODE	A code given to collision by SDOT (Seattle Department of Transportation)	26
12	INATTENTIONIND	Whether or not collision was due to inattention	0/1
13	UNDERINFL	Whether or not a driver involved was under the influence of drugs or alcohol	0/1
14	WEATHER	A description of the weather conditions during the time of the collision	Clear
15	ROADCOND	The condition of the road during the collision	Dry
16	LIGHTCOND	The light conditions during the collision	Daylight
17	PEDROWNOTGRNT	Whether or not the pedestrian right of way was granted	0/1
18	SPEEDING	Whether or not speeding was a factor in the collision	0/1
19	HITPARKEDCAR	Whether or not the collision involved hitting a parked car	0/1

20	YEAR	The year in which the collision took place	2011
21	MONTH	The month in which the collision took place	11
22	DAY	The day in which the collision took place	14
23	HOUR	The hour in which the collision took place	12
24	WEEKDAY	The weekday in which the collision took place	Monday

Table 1: Remaining attributes

3. Methodology

In this project we tried to predict the severity of an accident in the Seattle area based on different predictor features.

In our first step we have collected a suitable database that includes accident severity as a label and a number of additional attributes that could serve as predictor features potentially.

Our second step was to clean and wrangle the dataset:

- Obviously irrelevant features have been dismissed
- Categorical data has been cleaned
- Features containing any missing data points have been cleaned and dropped if necessary

The next step in our analysis will be the exploration of the relationship between SEVERITYCODE and our potential predictor features to get a better understanding of possible good candidates for features in our ML models. After finishing the data exploration we will drop features which do not serve as good predictor attributes for SEVERITYCODE.

We will then go ahead and preprocess the remaining data for our ML models by turning categorical data into numerical data and splitting our data set into training and testing data. Lastly, we will try ML different models and decide which one is the most suitable for the task.

3.1. Exploratory Data Analysis

3.1.1. Relationship between SEVERITYCODE and numerical attributes

For the numerical attributes in our data set I decided to use a scatterplot visualization and the Kendall's rank coefficient combined with the p value for quantitative analysis.

The Kendall's rank coefficient allows the comparison of columns of ranked data. A value close to 0 means no relationship and values close to 1 mean a perfect relationship. The test can also produce negative values, but they can just be treated like positive values.

In total, there were seven numerical attributes for which the relationship to SEVERITYCODE was explored further: PERSONCOUNT, PEDCOUNT, PEDCYLCOUNT, VEHCOUNT, INJURIES, SERIOUSINJURIES and FATALITIES.

It turned out that the only attribute with a strong correlation to SEVERITYCODE and high significance was the INJURIES attribute (Figure 1). For the rest of the attributes, to visualizations as well as the correlation coefficient and p value pointed to no significant relationship, and they are therefore not considered as suitable predictors for SEVERITYCODE.

Metric	**PERSONCOUNT**	**PEDCOUNT**	**PEDCYLCOUNT**	**VEHCOUNT**	**INJURIES**	**SERIOUSINJURIES**	**FATALITIES**
Kendall's rank coefficient	~0.176	~0.278	~0.222	~0.021	~0.949	~0.260	~0.088
P value	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

Table 2: Correlation metrics for numerical attributes

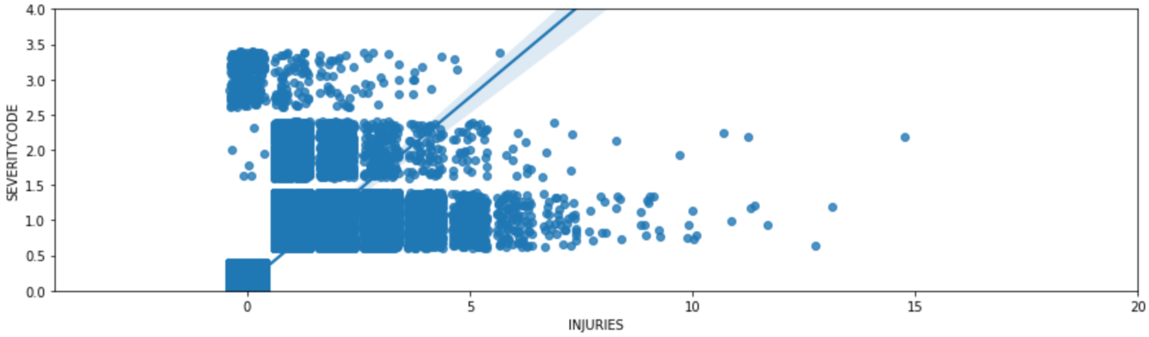

Figure 1: Relationship between INJURY and SEVERITYCODE

3.1.2. Relationship between SEVERITYCODE and categorical attributes

A good way to visualize categorical variables and identify their relationship with the label variable is by using boxplots. I decided to visualize all categorical attributes and also look at the frequency of each category of an attribute compared to the categories of the dependent variable. This will give me a good idea whether or not an attribute would be a helpful predictor variable.

In total, there are 17 categorical attributes for which we need to explore the relationship to SEVERITYCODE further: ADDRTYPE, COLLISIONTYPE, JUNCTIONTYPE, SDOT_COLCODE, INATTENTIONIND, UNDERINFL, WEATHER, ROADCOND, LIGHTCOND, PEDROWNOTGRNT, SPEEDING, HITPARKEDCAR, YEAR, MONTH, DAY, HOUR and WEEKDAY.

The only attributes with a significant enough distribution of SEVERITYCODE as well as a somewhat evenly distributed frequency of their categories were ADDRTYPE, PEDROWNOTGRNT and HITPARKEDCAR (Figures 2-4). The other categorical attributes did not show a significant difference in distribution between their categories and some of them were also very skewed.

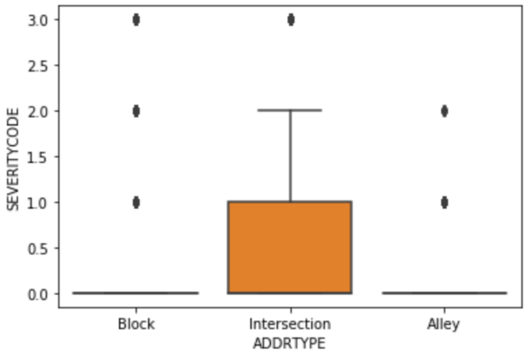

Figure 2: Relationship between ADDRTYPE and SEVERITYCODE

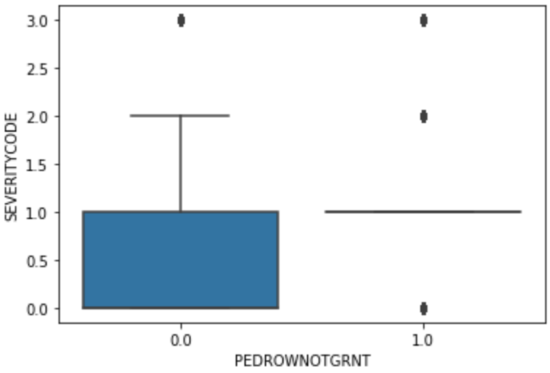

Figure 3: Relationship between PEDROWNOTGRNT and SEVERITYCODE

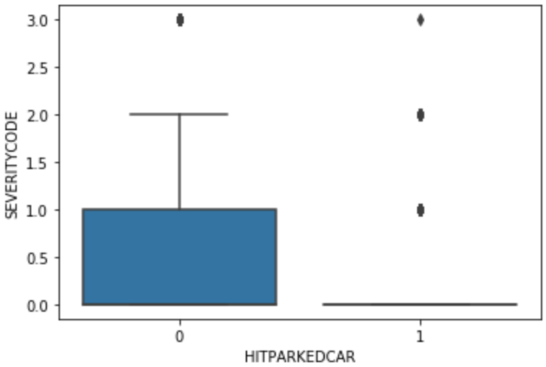

Figure 4: Relationship between HITPARKEDCAR and SEVERITYCODE

3.2. Data preparation

3.2.1. Format categorical data

Most ML models do not understand string values, only numbers. That is why I translated categorical values into numbers.

I did so by using the get_dummies() method. This method assigns a numerical variable to a category of an attribute. They are called 'dummies' because the numbers themselves don't have inherent meaning.

3.2.2. Split data

An important step in testing a model is to split data into training and testing data. We will put the target data SEVERITYCODE in a separate dataframe and drop price data in another separate x data frame for our predictor variables.

Next, I randomly split the data into training and testing data using the function train_test_split. The testing set was set to 15% of the total dataset.

3.3. Predictive Modelling

The scope of this Capstone was to create predictions for accident severity using Supervised Machine Learning techniques.

In general, this gives us two algorithm choices: Regression and classification algorithms. So which one would be suitable for our problem at hand?

3.3.1. Regression algorithms

In supervised machine learning, regression algorithms attempt to estimate the mapping function from the input variables to numerical or continuous output variables. With regression algorithms, y is a real value, which can be an integer or a float.

For example, when provided with a dataset about cars, and you are asked to predict their prices, that is a regression task because price will be a continuous output.

An example of a common regression algorithm would be linear regression.

3.3.2. Classification algorithms

Classification algorithms estimate the mapping function from the input variables to discrete or categorical output variables. With classification algorithms, y is a category that the mapping function predicts.

For example, when provided with a dataset about cars, a classification algorithm can try to predict whether the prices for the cars sell for a low, high or medium price compared to the recommended retail price. So the price will be classified into three discrete categories.

Examples of common classification algorithms include Logistic Regression, Naïve Bayes, decision trees, and K Nearest Neighbors.

3.3.3. Building classification models

Even though we are measuring severity using numbers, the problem we are facing can only be identified as a classification problem, because the numerical values are discrete and not continuous.

In order to classify as continuous, data needs to be able to have almost any numeric value and can be meaningfully subdivided into finer and finer increments, depending upon the precision of the measurement system. This is not the case for Severity.

That is why we will focus on the most common classification algorithms for our problem: Naive Bayes, Logistic Regression, K-nearest neighbors and decision trees.

I chose logarithmic loss and accuracy as the metrics of performance. Logarithmic loss was chosen because it works well with multiclass categorical output data and is useful to compare models not only on their output, but on their probabilistic outcome. In short, a lower logarithmic loss means better predictions.

3.3.4. Results

Between the models I built, the Decision Trees model performed the best (~98.3% accuracy, ~0.071 logarithmic loss), and Logistic Regression model performed almost equally good (Table 3). In general the performance differences between models were minor.

	Naive Bayes	**Logistic Regression**	**K-nearest Neighbors**	**Decision Trees**
Accuracy	0.707	**0.983**	**0.983**	**0.983**
Logarithmic loss	0.730	0.074	0.367	**0.072**

Table 3: Performance of classification models. Best performance labeled in red and bold.

4. Discussion

Even though I was able to achieve a high accuracy score for the models, I think they still leave a lot of room for improvement.

For example, the chosen dataset was very feature rich, most features did not have a good correlation with the value we wanted to predict. So one option would be to combine the dataset with additional collision data for Seattle, to get more correlated features that can be used in the prediction models.

In terms of usefulness of the predicted metric, it would be good to not only consider injuries and fatalities when looking at severity, but also look at the impact the collision has on the traffic, e.g. traffic delays for other commuters.

Finally, it would be nice to improve the model by not only predicting the severity of an accident, but to also predict whether or not a person might be involved in an accident when driving a certain route at a certain time and what severity that accident might have.

5. Conclusion

In this report, I analyzed the relationship between a collision severity and other characteristic of the collision. I identified injuries, address type, pedestrian right of way granted and hit parked car as the most important features that allow us to predict the severity. I built classification models to predict the severity of an accident. These models can be very useful in commuters to drive with more foresight or change their travel plans accordingly when a collision occurs on their way to work. In addition, the prediction model can help insurance companies predict possible costs for accidents suffered by their members.