Accident Severity Report

IBM Data Science Professional Certificate

Tien Hsin Lu

September 2020

Accident Severity Report

IBM Data Science Professional Certificate

Introduction

Imagining driving to a different city to work, or visit an old friend, it is rainy and windy, and on the way, there's a terrible traffic jam. You are waiting with long lines of cars, and the police cars start to appear. You realize that there might be an accident ahead. Later, you see a helicopter transporting the injured to the hospital, and you realize that this may be a severe accident causing possible fatality. What conditions on the road may be causing the accident? Could it be the weather, or the slippery road, or the dim light? This project is developed to predict the severity of an accident given the current weather, road and visibility conditions in order to reduce the frequency of car accidents/collisions in a community. This model will be able to predict the severity of accidents/collisions, and alert the drivers to be more careful if the conditions are critical.

Data

This project uses an updated dataset of collisions, provided by SPD and recorded by Traffic Records, to analyze how and what conditions would affect the severity of accidents. The dataset includes all types of collisions, display at the intersection or mid-block of a segment from 2004 to present, updated weekly.

Our target variable will be 'SEVERITYCODE', which is used to measure the severity of an accident. Attributes that will be used to measure the severity of an accident are 'WEATHER', 'ROADCOND', and 'LIGHTCOND', which represent the current weather, road condition, and light condition, respectively.

However, the raw dataset is not ideal for data analysis and the development of the desired algorithm. First of all, there are many columns that we will not use for this model. Also, most of the features are of type object, when they should be

numerical type. We will do some extracting and converting to the dataset for our convenience.

First we need to drop a few unnecessary columns due to the redundancy of those attributes:

	SEVERITYCODE	WEATHER	ROADCOND	LIGHTCOND
0	2	Overcast	Wet	Daylight
1	1	Raining	Wet	Dark - Street Lights On
2	1	Overcast	Dry	Daylight
3	1	Clear	Dry	Daylight
4	2	Raining	Wet	Daylight

Next, we check the data types of each attributes, and the balancing of the data.

```
: df.dtypes

8]: SEVERITYCODE      int64
    WEATHER          object
    ROADCOND         object
    LIGHTCOND        object
    dtype: object
```

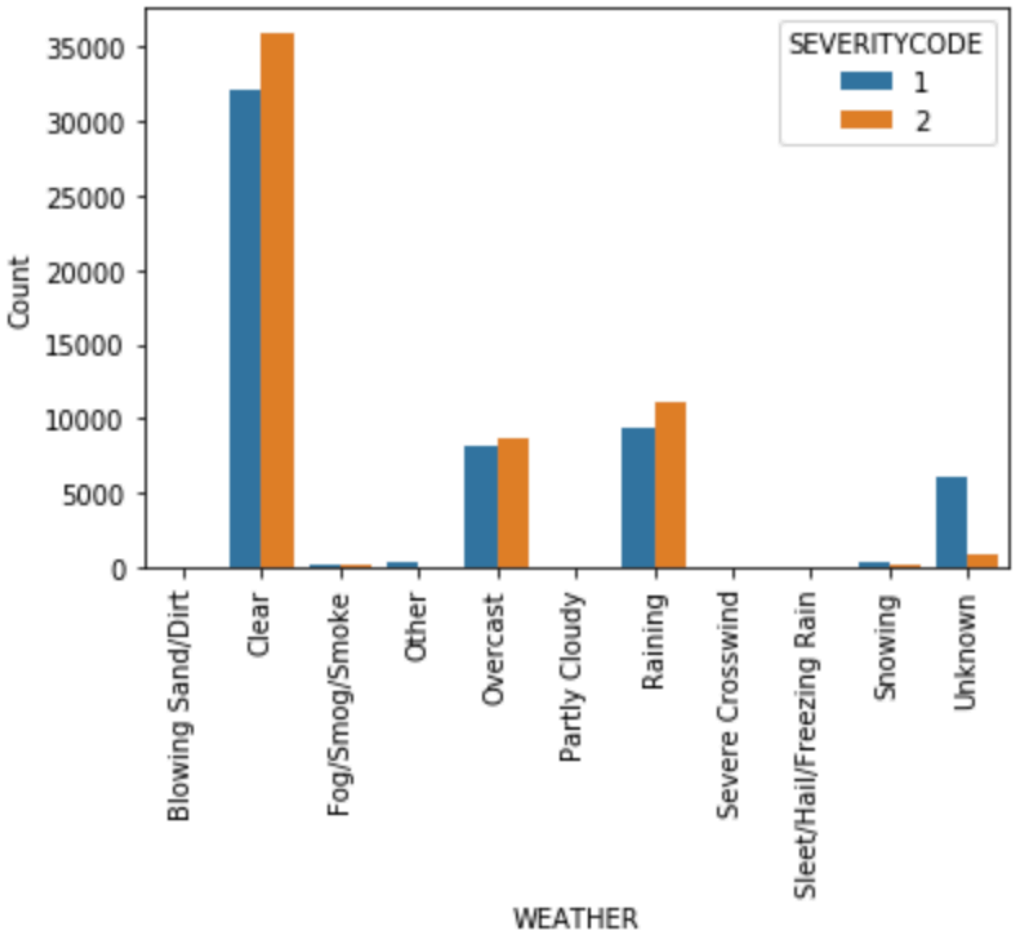
```
: df['SEVERITYCODE'].value_counts()

9]: 1    136485
    2     58188
    Name: SEVERITYCODE, dtype: int64
```

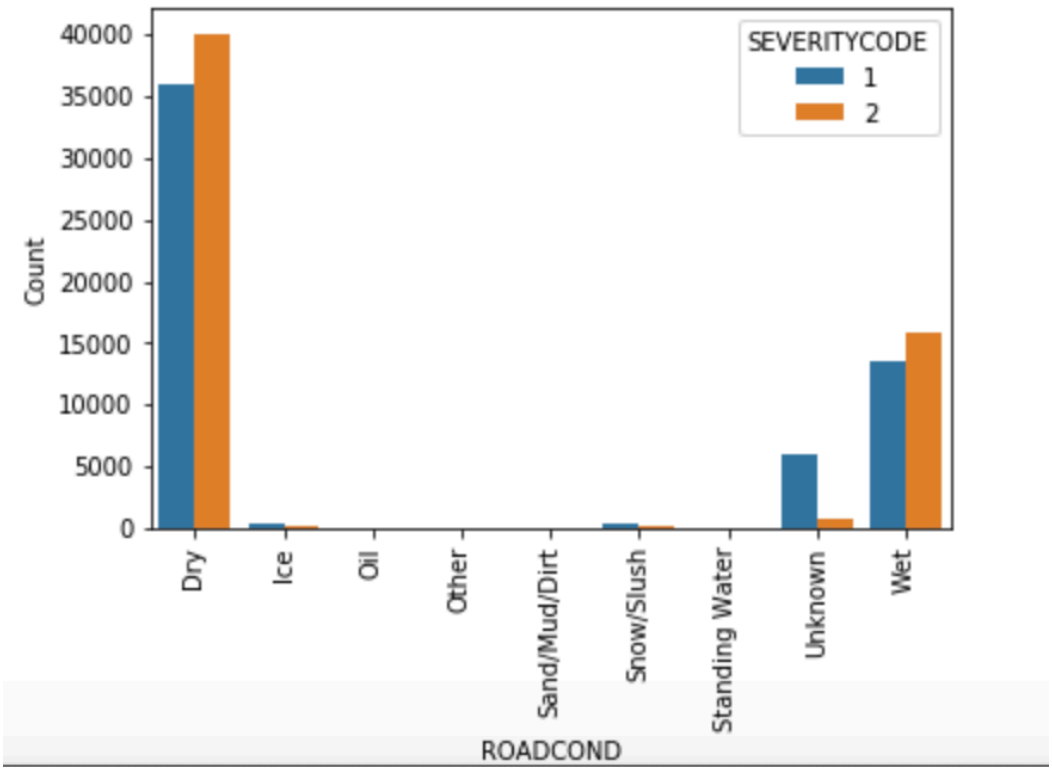
We observed that there is a lot more rows having “SEVERITYCODE” as “1”, meaning “no injury”, than “2”, indicating “injury”, hence, we will need to balance

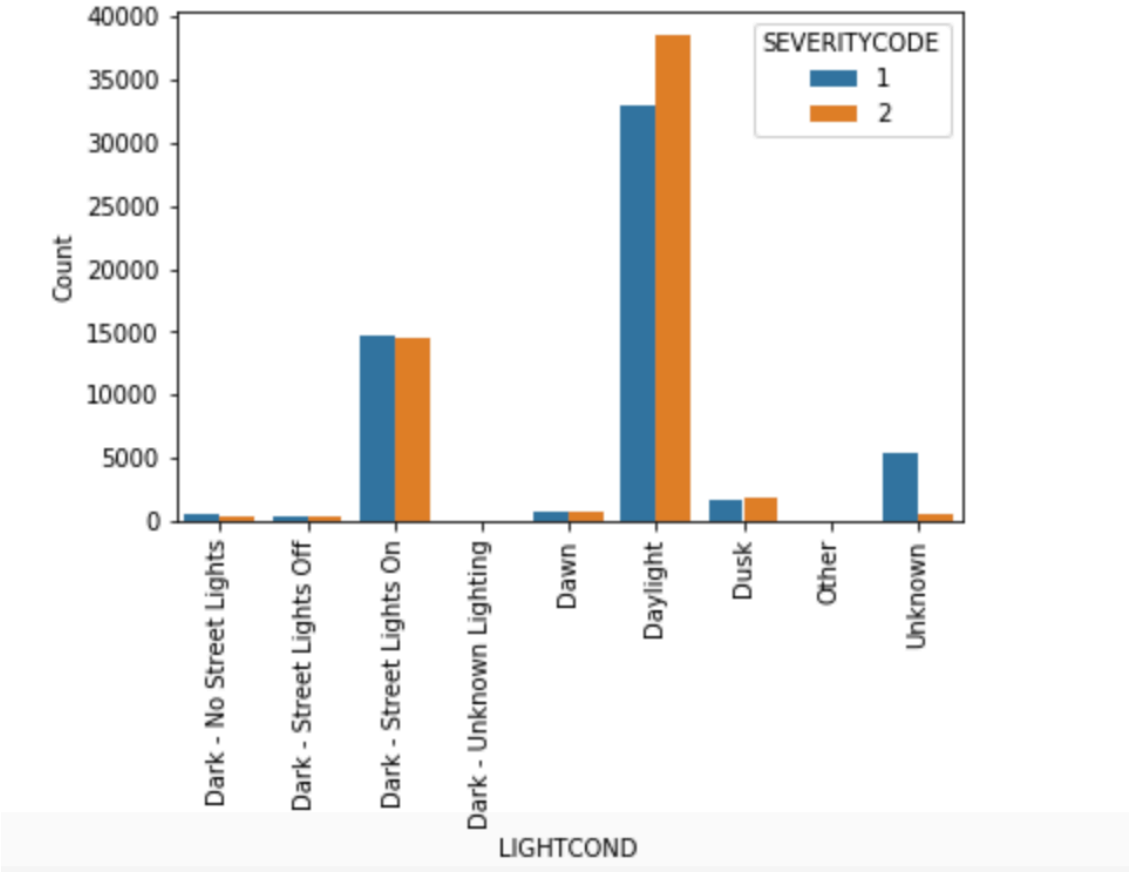
the data in order to have unbiased results. To achieve this, we downsampled the data entries having “SEVERITYCODE” as “1”, resulting in a balanced dataset.

The next step is to convert the categorical data to numerical value data. Let's first check out how each each attribute affects the severity code under each condition:



It appears that there’re more accidents when the weather is “clear”, “overcast”, and “raining”, and for each of those weather condition, there’s more accidents involving injury than those that do not. We also observed that there’re more accidents when the road condition is “dry” or “wet”, and when the light condition is “daylight” or “dark, street lights on”.


SEVERITYCODE
1
2
Count
40000
35000
30000
25000
20000
15000
10000
5000
0
Dry
Ice
Oil
Other
Sand/Mud/Dirt
Snow/Slush
Standing Water
Unknown
Wet
ROADCOND


SEVERITYCODE
1
2
Count
40000
35000
30000
25000
20000
15000
10000
5000
0
Dark - No Street Lights
Dark - Street Lights Off
Dark - Street Lights On
Dark - Unknown Lighting
Dawn
Daylight
Dusk
Other
Unknown
LIGHTCOND

Methodology

Once our data is now ready to be fed into machine learning models. We separate the dataset to a training set and a testing set with the test size equals to 20% of the entire dataset, with random state equals to 4. We will use the following models:

K-Nearest-Neighbors (KNN): A method for classifying cases based on their similarity to other cases.

A Decision Tree, which will provide a layout of all possible outcomes so we can fully analyze the consequences of a decision.

Logistic Regression: We can use logistic regression because the data set provides two severity code outcomes, resulted in a binary data set, which is perfect for logistic regression. We also would like to see a probabilistic result, and the impact of a feature. Logistic regression should be a good choice.

Result

We calculated the Jaccard index, F1-score, and the log loss for the algorithms we applied, and the following chart shows the result:

Algorithm	Jaccard	F1-score	LogLoss
KNN	0.5527152431689294	0.5518364129499082	nan
Decision Tree	0.5621670390101392	0.4808694306282786	nan
LogisticRegression	0.5621670390101392	0.4808694306282786	0.6844125103805109

We can see that among all of the machine learning algorithms we have tested, K-Nearest-Neighbors is the best algorithm to predict the severity of accidents given the conditions.

Conclusion

After evaluating all or the machine learning algorithms we have tested, and look at the previous charts and bar graphs, we can conclude that particular conditions have a some impact on whether or not travel could result in property damage or injury. However, it is possible that other attributes can affect the severity of accidents. For example, the time and date the accident occurs may have an impact on the severity of an accident, simply because the driver may either in a rush being late for work, or in a bad mood after an unsuccessful presentation or proposal. In the future, we can investigate the impact of other attributes, including date and time, on the severity of accidents. Furthermore, the severity of an accident may also depend on the type of the accident, but we will leave this for our future study.