Predicting Car Accident Severity

Victoria Quinn | Coursera Capstone | August 25, 2020

Introduction

People do not enjoy car accidents. In minor cases, the car is usually damaged, and possibly there is property damage. In the worst cases, people can be injured and killed in a severe crash. The factors that go into the severity of a crash are numerous. Road conditions, lighting, weather, all play a part. So, this project will look to see if we can use data to predict what conditions lead to the most severe crashes.

We can use data this data in a multitude of ways. In a small way, this could be used in driving courses, emphasizing to new drivers what conditions to be most careful of. We could use this in cars and map applications in order to warn drivers of conditions and to slow down. This could also possibly be used in self-driving cars as a way to reduce accidents.

Data

The data I used was the Data Collision information for the city of Seattle, from 2004-2020. This data is pulled from the police reports of the incidents, and has many variables, such as the weather, location, how the car hit something, and if people such as pedestrians or cyclists were involved. The target label, severity, is split into two results “1” for property-only collisions and “2” for collisions with injuries.

Overall, the data had 38 features, and 194,674 rows. This means it will be crucial to pick the best variables to test the data on.

Methodology

While the dataset is large, it is clear from the outset that many columns can be dropped. Columns like “INATTENTIONIND” and “EXCEPTRSNCODE” are full of null values. Columns like “STATUS” and “REPORTNO” are filled, but irrelevant to the severity of the crash. Therefore, there are relatively few variables that can directly correlate to crash severity. A column like “INATTENTIONIND” could be useful, as it indicates if the driver was distracted at the time of the accident. However, that is rarely recorded, presumably because it is difficult to prove distraction unless the driver admits it or there is a witness to prove the driver was not paying attention to the road.

However, a few of the columns jump out immediately as relevant, such as the road conditions at the time of the accident. This makes sense. Conditions such as rain or snow makes roads slippery, meaning a car could unexpectedly veer, and that braking will require more room because of the loss of friction against the road surface.

The second one was light conditions, such as daylight, dawn, dark – with streetlights on, and so on and so forth. This is also immediately makes sense. When it’s dark or dim out, it is harder to see both details and the larger area, meaning a driver might not see an obstacle until it’s too late.

However, as we will see later, it will seem contradictory that most accidents are happing during optimal driving conditions, such as when it’s clear, during the day, and when the roads are dry. But this does make sense. Most people have day jobs, requiring them to drive to and from work each day.

ROAD CONDITION

As we can see, most accidents happen when the roads are dry. Now, the unknown and other variables are not helpful in assessing severity, so they will be disregarded.

```
Dry                   124510
Wet                    47474
Unknown                15078
Ice                     1209
Snow/Slush              1004
Other                    132
Standing Water           115
Sand/Mud/Dirt             75
Oil                       64
Name: ROADCOND, dtype: int64
```

Along with that, the other variables: Standing water, Sand/Mud/Dirt, and Oil are also such a small percentage of accidents that they can be discarded as well. This leaves us with: Dry, Wet, Ice, and Snow/Slush. In the end, I also kept Standing Water, as it was quite similar to Wet conditions.

LIGHT CONDITION

The second variable I wanted to look at was the light conditions at the time of the accident. These conditions are much more evenly distributed, and as such I will only discard the unknown variables, and Other. That leaves us with 6 conditions to use later.

```
Daylight                      116137
Dark - Street Lights On        48507
Unknown                        13473
Dusk                            5902
Dawn                            2502
Dark - No Street Lights         1537
Dark - Street Lights Off        1199
Other                            235
Dark - Unknown Lighting           11
Name: LIGHTCOND, dtype: int64
```

WEATHER

This third variable was the weather at the time of the incident. Much like the road conditions, it has a sharp drop-off point after the most common weather events.

```
Clear                          111135
Raining                         33145
Overcast                        27714
Unknown                         15091
Snowing                           907
Other                             832
Fog/Smog/Smoke                    569
Sleet/Hail/Freezing Rain          113
Blowing Sand/Dirt                  56
Severe Crosswind                   25
Partly Cloudy                       5
Name: WEATHER, dtype: int64
```

Unfortunately, looking at this, the variables overlap too much with road condition. While not a direct one-to-one, it's too much for it to be one of the few variables chosen for the model. After all, if it's *raining*, the road is going to be wet. And if it's snowing, there's going to be snow on the ground. In the end, it's not going to be a particularly useful variable.

CORRELATION

Next, I looked to correlation to see if there were any numbers-based columns that could provide another variable. Unfortunately, even the strongest correlation was relatively weak.

	SEVERITYCODE
SEVERITYCODE	1.000000
X	0.010309
Y	0.017737
INCKEY	0.022065
COLDETKEY	0.022079
SEVERITYCODE.1	1.000000
PERSONCOUNT	0.130949
PEDCOUNT	0.246338
PEDCYLCOUNT	0.214218
VEHCOUNT	-0.054686
SDOT_COLCODE	0.188905
SDOTCOLNUM	0.004226
SEGLANEKEY	0.104276
CROSSWALKKEY	0.175093

As we can see in the image, the strongest correlation accounts for just shy of 25 percent of the data. We must therefore keep looking for another variable.

First I looked at Junction Type, which was where the incident took place, such as at an intersection, alley, or street. However, this did not feel as relevant as other variables.

COLLISION TYPE

The collision type of the accident was a much better variable. After all, hitting a parked car and sideswiping a car were very different things, and could have a much greater impact on severity than where in the street you were hit.

```
Parked Car    47987
Angles        34674
Rear Ended    34090
Other         23703
Sideswipe     18609
Left Turn     13703
Pedestrian     6608
Cycles         5415
Right Turn     2956
Head On        2024
Name: COLLISIONTYPE, dtype: int64
```

Like Light Condition, we can see that there are many valuable variables within this column. However, for the purposes of the model, I only used those variables with 10,000+ incidents (except Other, of course) giving us: Parked Car, Angles, Rear Ended, Sideswipe, and Left Turn.

Results and Discussion

I used the KNN method of evaluating these variables and their relation to the severity code target label. First, I had to drop all other columns, and then drop the unwanted labels from each variable as well.

Once that was done, I think had to re-index the variables so they were all variables instead of objects. As so:

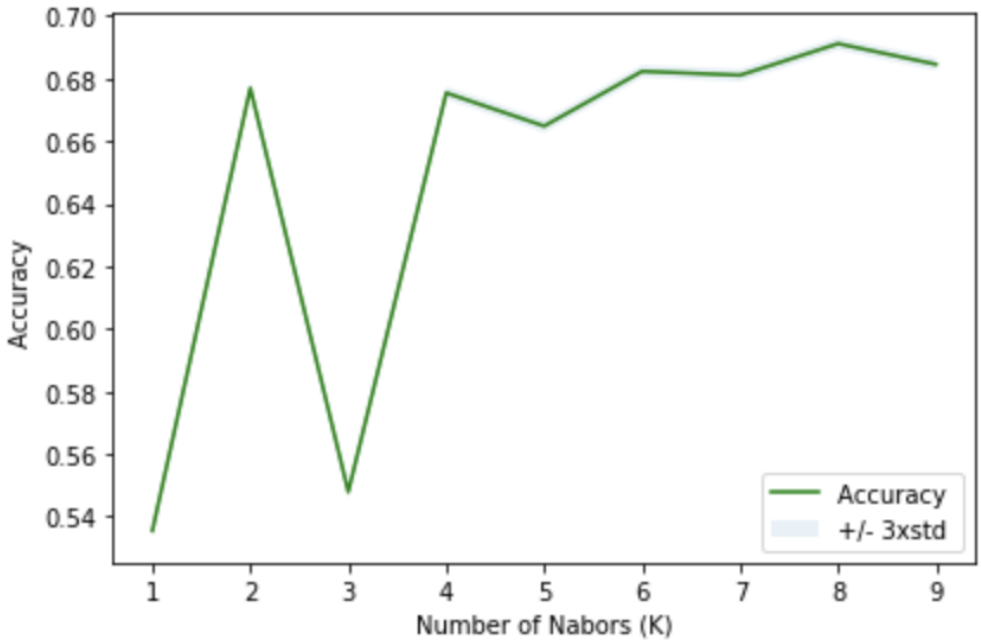
	index	COLLISIONTYPE
1	Angles	33758
2	Parked Car	33755
3	Rear Ended	32670
4	Sideswipe	17650
5	Left Turn	13363

	index	ROADCOND
1	Dry	95215
2	Wet	34717
3	Ice	631
4	Snow/Slush	585
5	Standing Water	48

	index	LIGHTCOND
1	Daylight	89676
2	Dark - Street Lights On	34361
3	Dusk	4458
4	Dawn	1709
5	Dark - No Street Lights	992

I then converted the index strings to numerical values (matching those on the left-hand side) to complete the conversion.

For the algorithm I chose a test-train split of 40-60, with a +/- 3 std deviation. From the graph, we can see that the std deviation is small, given that it's almost completely invisible on the graph. Along with the results of that algorithm.




```
The best accuracy was with 0.6911526515368052 with k= 8
```

Conclusion

So, from the KNN we can see that we are most accurate when k = 8 with an R-value of 0.69. Meaning that car crash severity can be predicted with 69% accuracy. Which is pretty good considering that this is a fairly basic model.

In general, we can say that encouraging people to slow down and pay more attention to these variables can indeed help reduce car crash severity and the number of overall car crashes.

The model itself can fairly easily be adapted to add new variables or change the ones already included.