Capstone - Car Collision Severity

Introduction

The car is the most-used vehicle of the last century. In 2017 there were almost 1,200,000,000 cars all over the world. Of course, with such a high number of people relaying on cars in their everyday lives, car collisions and accidents happen continously; they have become a regular part of our modern world.

In so many instances, car accidents are fatal for everyone involved. In fact, according to an OMS study in 2015, there were a calculated 1.25 million annual accident-related deaths on the road.

Any car collision is inherently related to other factors, such as a long line of cars on the highway, a problem with the road itself or excessive traffic due to maintenance after another collision. The company that manages the roads views these collisions as a high expense. To better prepare themselves and avoid these high costs, companies should better predict these collisions and their severity in order to save money and to invest them in innovative infrastructures.

This can be achieved by studying the data that has been collected in the past years and using a prediction model in order to estimate the severity of future collisions and find solutions to avoid them entirely, thereby saving lives and money in the process.

For example, let's take a look at a database provided by the “SDOT Traffic Management Division” in Seattle, Washington. We will study the Seattle cases since 2004 using our data science powers to give the best severity prediction of hypothetical future accidents.

Data

In this specific case, our purpose is find a way to predict the severity level of a collision based on our knowledge of other attributes such as weather conditions, road conditions, light conditions, types of collision and even the description of the collision itself.

For this exercise, we will use a common algorithm, the K-Nearest Neighbors method, which will help us predict the severity level of future accidents by using a portion of the attributes that are stored in the dataset.

For our purposes, let's use the factors that we have just mentioned to create our model: light, road and weather conditions. We will train the model with a large part of the dataset, and then use the remaining part as the test portion.

In this case, according to those attributes, we can predict the severity level and prepare ourselves for how we are going to face an eventual collision. This will help the stakeholders to make important decisions about how to invest their money in the company for the purpose of preventing collisions (alternative roads, traffic limits, new traffic laws, etc.).

Let's start with the dataset

We can begin by importing the dataset and all of the libraries we will use during this project, including numpy and pandas libraries.

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGHTCOND	PEDROWNOTGRNT	SDOTCOLNUM	SPEEDING	ST_COLCODE	ST_COLDESC
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Daylight	NaN	NaN	NaN	10	Entering at angle
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark - Street Lights On	NaN	6354039.0	NaN	11	From same direction - both going straight - bo...
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Daylight	NaN	4323031.0	NaN	32	One parked--one moving
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Daylight	NaN	NaN	NaN	23	From same direction - all others
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Daylight	NaN	4028032.0	NaN	10	Entering at angle
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
194668	2	-122.290826	47.565408	219543	309534	310814	E871089	Matched	Block	NaN	...	Dry	Daylight	NaN	NaN	NaN	24	From opposite direction - both moving - head-on
194669	1	-122.344526	47.690924	219544	309085	310365	E876731	Matched	Block	NaN	...	Wet	Daylight	NaN	NaN	NaN	13	From same direction - both going straight - bo...
194670	2	-122.306689	47.683047	219545	311280	312640	3809984	Matched	Intersection	24760.0	...	Dry	Daylight	NaN	NaN	NaN	28	From opposite direction - one left turn - one ...
194671	2	-122.355317	47.678734	219546	309514	310794	3810083	Matched	Intersection	24349.0	...	Dry	Dusk	NaN	NaN	NaN	5	Vehicle Strikes Pedalcyclist
194672	1	-122.289360	47.611017	219547	308220	309500	E868008	Matched	Block	NaN	...	Wet	Daylight	NaN	NaN	NaN	14	From same direction - both going straight - on...

Let's count the different values that our target, the "**SEVERITYCODE**" attribute, has in this dataset:

```
1    136485
2     58188
```

Basically we have 194,673 rows of data, and each row represents a collision. Each column represents a different attribute of that collision (like weather, road condition, latitude, longitude, a description of the accident.

It is important to determine which attributes we would need to use in order to predict our target value, the "SEVERITYCODE" attribute. In this case, we will try to build our model basing it on the weather conditions, the light conditions and the road conditions, as I previously stated.

Let's count the different values of the weather that are available in the dataset:

```
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
```

Now about the light conditions:

```
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
```

And finally about the road conditions:

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

Let's have a look at these features in relation to the first 5 collisions in the dataset, including the related target value:

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

Data Cleaning

Unfortunately, for several of the accidents listed, the data collectors failed to record some of the data that we are hoping to study.

For this reason, the best possible way to proceed is to clean the dataset, saving the given data in a different dataset. For example, we can eliminate all the rows that contain at least one "NaN" value in the weather column, the light conditions column or the road conditions column.

	SEVERITYCODE	WEATHER	LIGHTCOND	ROADCOND
0	2	Overcast	Daylight	Wet
1	1	Raining	Dark - Street Lights On	Wet
2	1	Overcast	Daylight	Dry
3	1	Clear	Daylight	Dry
4	2	Raining	Daylight	Wet
...	...	...	...	...
194668	2	Clear	Daylight	Dry
194669	1	Raining	Daylight	Wet
194670	2	Clear	Daylight	Dry
194671	2	Clear	Dusk	Dry
194672	1	Clear	Daylight	Wet

As you can see, we have reduced the number of rows to 189,337 instead of 194,673 so that we can better focus on the critical data.

Let's have a look at the new counts for the weather conditions:

```
Clear                     111008
Raining                    33117
Overcast                   27681
Unknown                    15039
Snowing                      901
Other                        824
Fog/Smog/Smoke               569
Sleet/Hail/Freezing Rain     113
Blowing Sand/Dirt             55
Severe Crosswind              25
Partly Cloudy                  5
```

The light conditions:

```
Daylight                  116077
Dark - Street Lights On    48440
Unknown                    13456
Dusk                        5889
Dawn                        2502
Dark - No Street Lights     1535
Dark - Street Lights Off    1192
Other                        235
Dark - Unknown Lighting       11
```

And the road conditions:

```
Dry              124300
Wet               47417
Unknown           15031
Ice                1206
Snow/Slush          999
Other               131
Standing Water      115
Sand/Mud/Dirt        74
Oil                  64
```

As I stated earlier, we have decided to use the K-Nearest Neighbors method to study these cases, with 3 different independent values (**WEATHER CONDITIONS**, **LIGHT CONDITIONS** and **ROAD CONDITIONS**) that will help us to predict the target value, which is the **SEVERITYCODE**.

Talking about the independent values, we now need to normalize the data.

In fact, **Data Standardization** give data zero mean and unit variance, it is good practice, especially for algorithms such as KNN which is based on distance of cases.

Out of Sample Accuracy is the percentage of correct predictions that the model makes on data that that the model has not been trained on. Doing a train and test on the same dataset will most likely have low out-of-sample accuracy, due to the likelihood of being over-fit.

It is important that our models have a high, out-of-sample accuracy, because the purpose of any model, of course, is to make correct predictions on unknown data. So how can we improve out-of-sample accuracy? One way is to use an evaluation approach called **Train/Test Split**. Train/Test Split involves splitting the dataset into training and testing sets respectively, which are mutually exclusive. After which, you train with the training set and test with the testing set.

This will provide a more accurate evaluation on out-of-sample accuracy because the testing dataset is not part of the dataset that have been used to train the data. It is more realistic for real world problems.

```
Train set: (170403, 3) (170403,)
Test set: (18934, 3) (18934,)
```

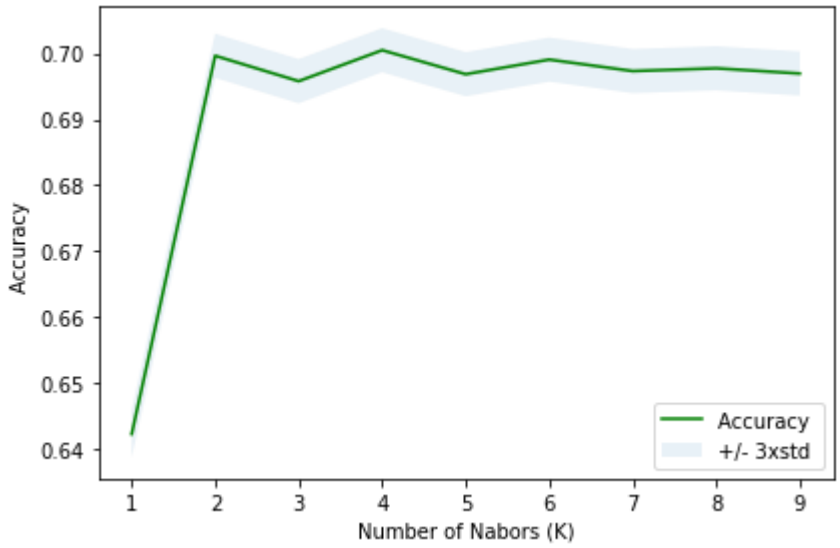
In multilabel classification, **accuracy classification score** is a function that computes subset accuracy. This function is equal to the jaccard_similarity_score function. Essentially, it calculates how closely the actual labels and predicted labels are matched in the test set.

With a k=5 we found out that:

```
Train set Accuracy:  0.6926051771389001
Test set Accuracy:  0.6968416605049118
```

We chose a number for the "**k**" parameter but it would be ideal to find the best "**k**" that can give us the most accuracy and that fits the model in the best way. How can we do this? The general solution is to reserve a part of your data for testing the accuracy of the model. Then chose k =1, use the training part for modeling, and calculate the accuracy of prediction using all samples in your test set. Repeat this process, increasing the k, and see which k is the best for your model.

Searching which one is the best “k” in between 1 and 10, we found out that k=4 is the best parameter, as we can see in the plot below:




```
The best accuracy was with 0.7004858983838598 with k= 4
```

It appears that with k=4 we have the best accuracy, with a score of 0.70 out of 1.

Let's have a look at the train set and the test set accuracy scores:

```
Train set Accuracy:  0.6958034776383045
Test set Accuracy:  0.7004858983838598
```

As we can see, both sets have a high level of accuracy.

This determines that with 90% of the data used as a training set, and the remaining 10% used as a test set, we can predict the severity of a collision according to the 3 most visible parameters (weather, light and road conditions at the moment of the accident), with an accuracy of 70%.

Using this model and only these three parameters has provided us with a favorable lens through which we can observe the probability of collision. In fact, if we would have used more attrbiutes as independent variables, we would have had a lower accuracy. For example, let's say that we want to add the latitude and the longitude of the collisions to the independent parameters.

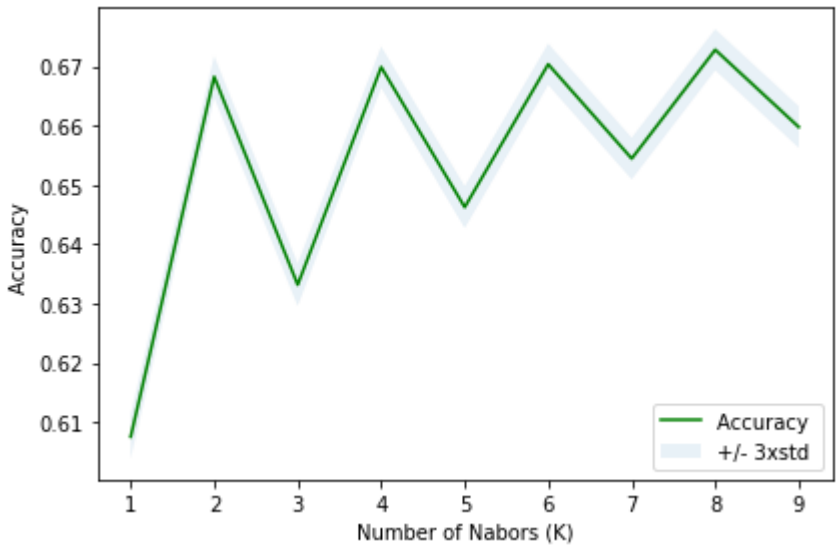
Let's see what happens and comment on the results.

	SEVERITYCODE	WEATHER	LIGHTCOND	ROADCOND	X	Y
0	2	Overcast	Daylight	Wet	-122.323148	47.703140
1	1	Raining	Dark - Street Lights On	Wet	-122.347294	47.647172
2	1	Overcast	Daylight	Dry	-122.334540	47.607871
3	1	Clear	Daylight	Dry	-122.334803	47.604803
4	2	Raining	Daylight	Wet	-122.306426	47.545739
...	...	...	...	...	...	...
194668	2	Clear	Daylight	Dry	-122.290826	47.565408
194669	1	Raining	Daylight	Wet	-122.344526	47.690924
194670	2	Clear	Daylight	Dry	-122.306689	47.683047
194671	2	Clear	Dusk	Dry	-122.355317	47.678734
194672	1	Clear	Daylight	Wet	-122.289360	47.611017

The accuracy scores:

```
Train set Accuracy:  0.7237345399698341
Test set Accuracy:  0.6728023022207743
```

And for the Ks:




```
The best accuracy was with 0.6728023022207743 with k= 8
```

As we can see, the model with more variables is less accurate than the first one. In fact, the accuracy in this case is 0.6728 instead of a 0.7004. This means that sometimes “less is more”, meaning that using more parameters is less productive for the scope than using few but important factors.

Conclusion

In conclusion, the K-Nearest Neighbors algorithm provided us with a perspective of what could happen in a situation with varying conditions.

These three parameters were chosen because they represent the top three danger factors for a collision, therefore the study of these three factors combined gives us the background of every possible accident. It is clear that some characteristics of the single collision are difficult to predict, such as the culpable drivers' behaviours or the reactions of the victims right before they are hit.

It is important to understand that, having this information, it is possible to predict and hopefully prevent car collisions, by investing in new infrastructures following these parameters to reduce accidents, consequentially avoiding emergency maintenance and allowing drivers to have a more comfortable and safe driving experience with their families.