Predicting the Severity of Car Accidents

By Javier Lago, November 2020

1. Introduction

1.1. Background

Let's suppose that we are driving and, in a certain moment, we observe some traffic congestion. When we approach the area, we realise that it has been caused by a car accident. The accident might have different degrees of severity, from relatively small consequences (e.g. small harm to third parties or to third party property) to fatal consequences (deaths). If the accident is more severe, usually more resources (police, medical, firemen, cleaning) and more time will be needed to solve the situation.

The severity degree, among others, could depend on certain environmental factors (light condition, weather condition, road condition, etc.), in such a way that if those factors are known, it might be possible to have a hint on what could be the potential (most probable) consequences of an accident under these circumstances. It might also depend on the timeframe we are considering (e.g. Monday versus Friday or Sunday, beginning of the working day versus lunch time, July versus October, etc.), that will be also an "external" factor.

1.2. Business Problem

We can summarize our problem in the following way: we would like to know if it is possible to predict the severity degree of a car accident depending on different parameters of the environment, and build a model that is able to perform this prediction and with which degree of accuracy. This will also allow us to know which is the "weight" of these "external" factors regarding the potential consequences of an accident.

1.3. Target audience

The results of this model might be used to implement different preventive measures to reduce the risk of an accident with high severity. For example, city authorities could reduce speed limits, launch warnings to the drivers encouraging them to increase precaution, increase the number of emergency resources (firemen, ambulances, police) on alert condition, or even apply a different traffic lights pattern or establish traffic controls at specific times.

So, the main audience in this case will be composed by:

- City Authorities
- Traffic Department
- Emergency Services (police department, fire department, emergency department, hospitals)

JUNCTIONTYPE	SDOT_COLCODE	SDOT_COLDESC
At Intersection (intersection related)	0	NOT ENOUGH INFORMATION / NOT APPLICABLE

INATTENTIONIND	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND	PEDROWNOTGRNT	SDOTCOLNUM
Y	N	Clear	Dry	Daylight	Y	10168010

SPEEDING	ST_COLCODE	ST_COLDESC	SEGLANEKEY	CROSSWALKKEY	HITPARKEDCAR
	1	Vehicle turning right hits pedestrian	0	525345	N

2.3. Dataset transformation and feature selection

Original dataset contains 194673 records, with 38 data fields as we have shown in the previous point.

The first transformation will be extracting a copy of the dataset including only a subset of the data fields, that will contain the features that we will use for the analysis. In this case the selected columns will be: 'SEVERITYCODE', 'INCDTTM', 'WEATHER', 'ROADCOND' and 'LIGHTCOND'.

The second step will be transforming the type of the 'INCDTTM' column from the type assigned on the import operation (object) in a suitable type to perform date and time operations (datetime).

Once this has been done, we will add three new columns to the dataset calculated from the 'INCDTTM' value: 'WEEK' (week number), 'WEEKDAY' (day of the week) and 'HOUR' (hourly interval during the day). Using a datetime value with such granularity level as 'INCDTTM' will not be practical for our purpose, so we will group the accidents using these three timescales. Once these variables have been calculated and added to the dataset, the original 'INCDTTM' column will be dropped. After this operation we will still have 194673 data records, but only 9 data columns.

The last step before performing an exploratory data analysis will be to clean records that contain at least one empty value, as they do not provide the required information. Afterwards, the dataset contains 189337 records.

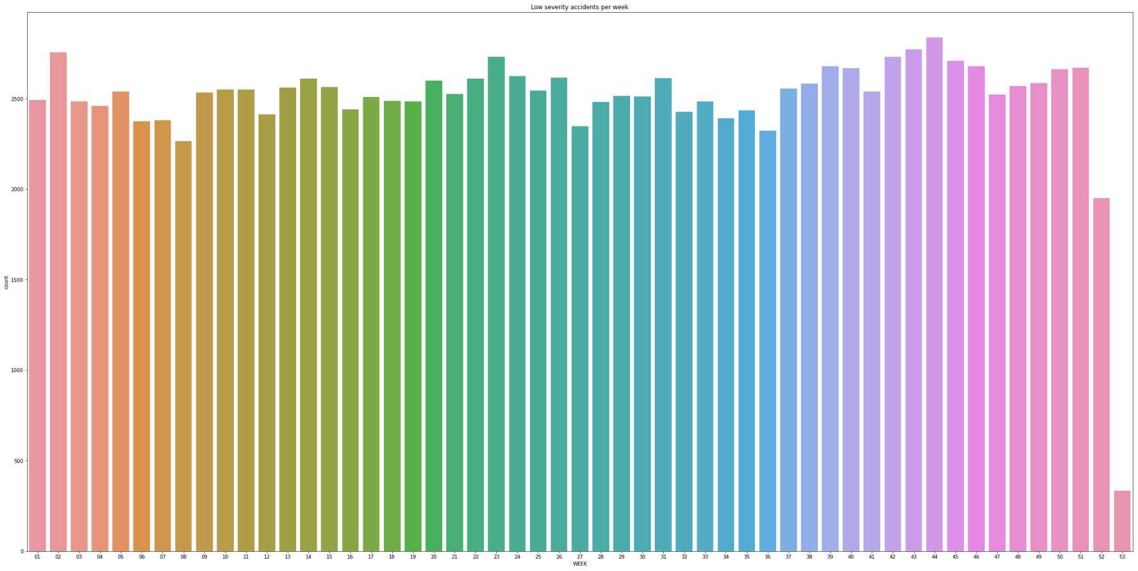
3. Methodology

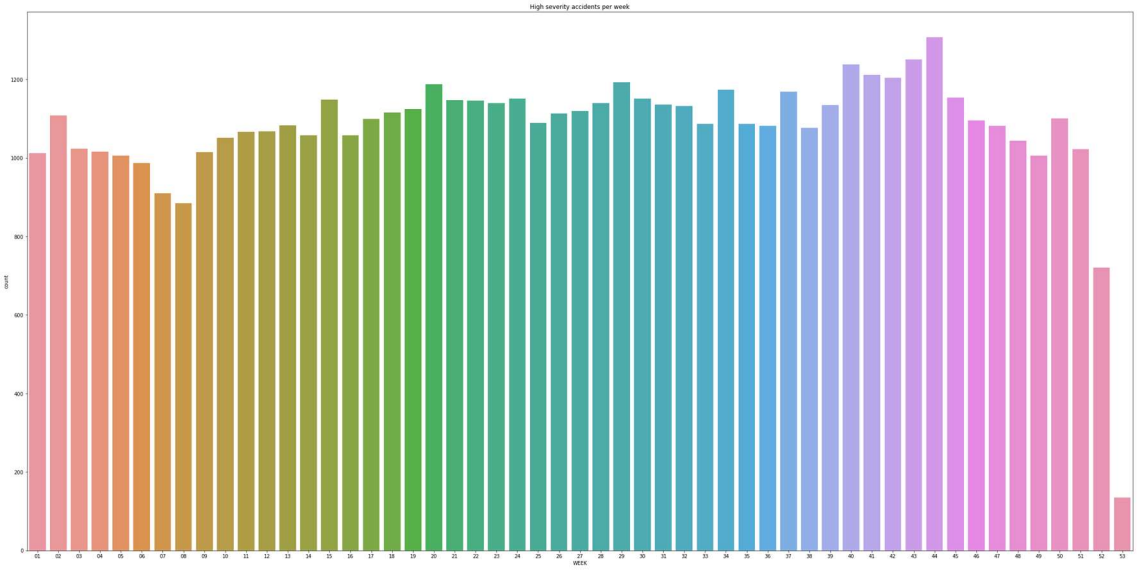
As exposed in the introduction section, the objective is to predict the severity degree of an accident. Given that severity is a discreet value, what we have, in terms of Data Science, is a classification problem. This will determine the methods (algorithms) that we might use to perform the analysis.

3.1. Target value analysis

First thing to analyse is the target value (also called "label"), in our case the 'SECURITYCODE' value. We can see that there are two values, corresponding to different severity degrees:

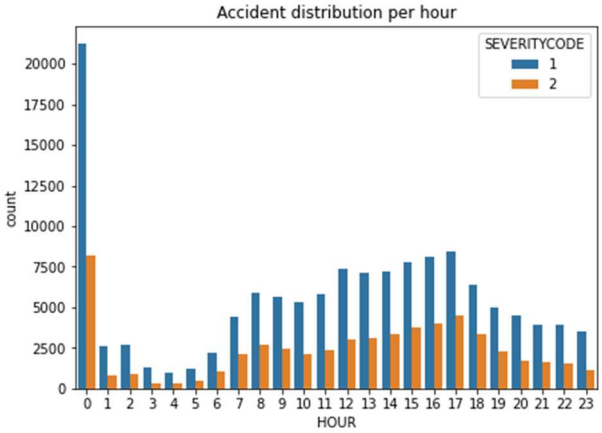
- '**1**' (Property Damage Only Collision) → low severity level
- '**2**' (Injury Collision) → high severity level





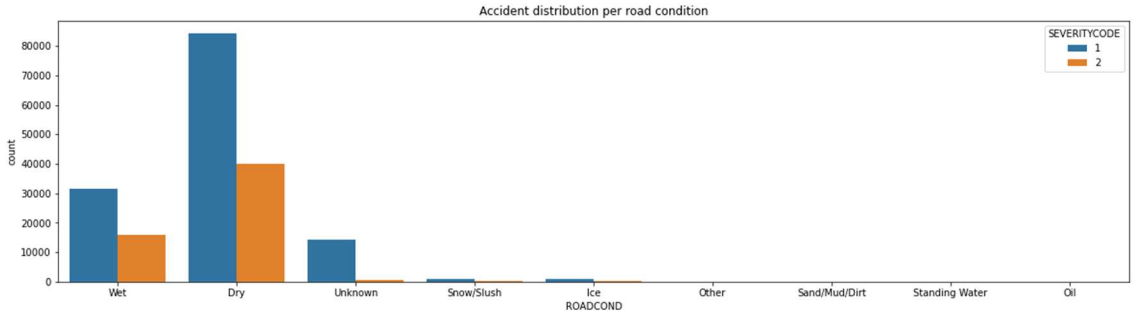
We see that both graphics have the same shape, and thus we visually confirm that the distribution is similar in both cases.

We will repeat the analysis with the 'HOUR' value, obtaining the following results:



HOUR	Severity = 1	Severity = 2
Average(%)	70.88 %	29.12 %
StdDev(%)	3.71 %	

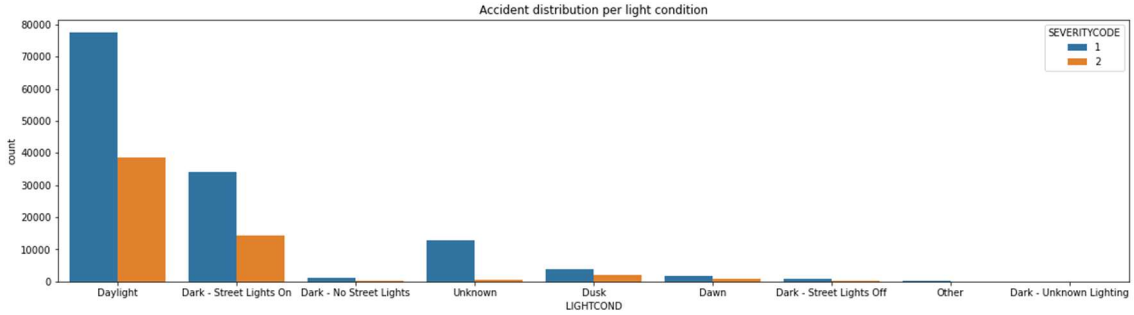
If we repeat this analysis with the road condition information ('ROADCOND') we obtain:



ROADCOND	Severity = 1	Severity = 2	# records	% records
Dry	67.82 %	32.18 %	124300	65,65%
Ice	77.36 %	22.64 %	1206	0,64%
Oil	62.50 %	37.50 %	64	0,03%
Other	67.17 %	32.82 %	131	0,07%
Sand / Mud / Dirt	68.92 %	31.08 %	74	0,04%
Snow / Slush	83.38 %	16.62 %	999	0,53%
Standing Water	73.91 %	26.09 %	115	0,06%
Unknown	95.03 %	4.97 %	15031	7,94%
Wet	66.80 %	33.20 %	47417	25,04%
Average(%)	73.66 %	26.34 %		
StdDev(%)	1.02 %			

In this case, we will keep only the values 'Dry' and 'Wet' and the combination of 'Ice' and 'Snow / Slush' in a single value (as they reflect a similar condition). Rest of values will be discarded.

Finally, if we do the same with the light condition ('LIGHTCOND') feature, we obtain:



LIGHTCOND	Severity = 1	Severity = 2	# records	% records
Dark – No Street Lights	78.24 %	21.76 %	1535	0,81%
Dark – Street Lights Off	73.49 %	26.51 %	1192	0,63%
Dark – Street Lights On	70.17 %	29.83 %	48440	25,58%
Dark – Unknown Lightning	63.64 %	36.36 %	11	0,01%
Dawn	67.07 %	32.93 %	2502	1,32%
Daylight	66.81 %	33.19 %	116077	61,31%
Dusk	67.09 %	32.91 %	5889	3,11%
Other	77.87 %	22.13 %	235	0,12%
Unknown	95.50 %	4.50 %	13456	7,11%
Average(%)	73.32 %	26.68 %		
StdDev(%)	0.97 %			

called "averaging methods", based on averaging the result of several independent estimators reducing variance, and the one called "boosting methods", where base estimators are built sequentially, each one trying to reduce the bias of the combined estimator.

The ensemble algorithms that we will consider will be:

- Random Forest Tree
- Gradient Tree Boost
- Voting Classifier

We will use the train set for fitting (training) the different models, and then we will predict the outcomes of the test set, comparing them with the expected (real) ones, in order to measure the performance of each of the algorithms.

3.5. Performance metrics

In order to measure the performance of the different algorithms, we will use a series of metrics. If we take as starting point the following classification of the outcome of any of these methods (what we call the "confusion matrix"):

		Predicted severity values	
		1	2
Actual severity values	1	TN (True Negative)	FP (False Positive)
	2	FN (False Negative)	TP (True Positive)

In our case, the higher severity case (SEVERITYCODE = 2) will be the case we want to detect. More severe accidents will be the ones requiring a faster response and more resources, and also the ones that will have the worst consequences (injured people).

From this definition, it is possible to define different metrics, that we will use for evaluating the performance of the different models:

- **Precision**: that provides the relationship between the "true positives" (those events belonging to the category of interest that have been properly classified) and the total number of cases that have been classified into this category. This measures the "quality" of the model. The formula to calculate this parameter is the following:

$$precision = \frac{TP}{TP + FP}$$

- **Recall**: that provides the relationship between the "true positives" and the total number of cases that really should have been classified into this category. This provides a measure of the "quantity" of cases than the algorithm can really detect from the total. The formula to calculate this parameter is the following:

$$recall = \frac{TP}{TP + FN}$$

	KNN	Decision Tree	SVM	Logistic Regression
Parameter 1	K = 5	Criteria = entropy	Kernel = rbf	Solver = newton-cg
Parameter 2	-	Max. depth = 6	-	-
Max. Recall	51.45%	72.19%	73.60%	54.12%
Recall	50.77%	58.27%	62.84%	54.12%
Min. Recall	26.12%	47.53%	49.39%	54.07%
Max. Precision	49.66%	51.31%	51.14%	51.30%
Precision	49.66%	51.26%	51.14%	51.30%
Min. Precision	49.08%	50.18%	49.41%	51.29%
Max. Accuracy	50.71%	52.35%	52.33%	52.30%
Accuracy	50.61%	52.35%	52.33%	52.30%
Min. Accuracy	50.01%	51.03%	50.38%	52.29%
F1	50.21%	54.54%	56.39%	52.67%
F2	50.54%	56.72%	60.09%	53.53%

	Random Forest	Gradient Tree	Voting (hard)	Voting (soft)
Parameter 1	-	Estimators = 2	-	-
Parameter 2	-	Learning rate = 0.2	-	-
Max. Recall	53.97%	68.50%	-	-
Recall	53.70%	56.52%	54.66%	54.77%
Min. Recall	49.76%	47.98%	-	-
Max. Precision	50.32%	51.37%	-	-
Precision	50.27%	51.37%	53.27%	51.96%
Min. Precision	49.79%	49.86%	-	-
Max. Accuracy	51.29%	52.43%	-	-
Accuracy	51.24%	52.43%	52.46%	51.15%
Min. Accuracy	50.74%	50.81%	-	-
F1	51.93%	53.82%	53.96%	53.33%
F2	52.97%	55.41%	50.45%	47.93%

The first thing we observe is that the maximum accuracy level obtained after testing all of the different methods (both individual and ensemble) is below 52.44%. This means that our models are roughly able to properly classify half of the events (the other half is classified in a wrong category). Although it is not a good result, in this case we have to consider other metrics. Accuracy is a strong decision metric in the case that all of the categories are equally important, as it provides an overall quality score for the model. In our case, we are more interested in a specific category (high severity accidents) and thus we might still have a good model with this accuracy level (we might be properly classifying most of the wanted elements and unproperly few of the unwanted ones). To verify this, we have to take into account two other metrics: recall and precision.

Recall will provide a measure of the ability of the model to properly detect events belonging to a specific category, in our case high severity accidents. We can see that some methods provide recall levels up to 73.60%, meaning that we could detect three out of four high severity accidents as such. Unfortunately, this metric alone can provide a wrong perception of the quality of the algorithm (in the limit, if we considered all of the events to be “high severity” we will have a 100% recall score, but at the same time we will be classifying wrongly around a 70% of the total

(between 66% and 67%), while the recall values are below 3,11%. This indicates that the model is not detecting the "high severity" accidents as such, but as "low severity" accidents. This is an example of a model providing a good accuracy value (almost 70%) but being not useful for our objective. Given the proportion between low and high severity accidents (70% - 30%, approximately), classifying all the accidents as being "low severity" will provide a 70% accuracy, but a poor performance as we will not detect the events belonging to the other category, that is the most interesting for us.

Model Selection and final validation

Once we have selected the three best performing algorithms and made the tuning of its parameters to obtain the best result, we will train the model using the whole debiased dataset and, once done, we will use the biased dataset for testing purpose, trying to get a better indication of the results we will have on a real environment. The results can be seen in the following table:

	SVM	Decision Tree	Gradient Tree Boost
Recall	61.08%	59.56%	55.41%
Precision	34.67%	34.88%	34.98%
Accuracy	49.43%	50.22%	51.54%
F1	44.23%	43.99%	42.88%
F2	53.01%	52.17%	49.61%
Fitting Time	623.49 s	0.18 s	4.14 s
Execution Time	396.10 s	0.25 s	0.15 s
TP	33723	32880	30591
FN	21485	22328	24617
FP	63550	61383	56867
TN	49407	51574	56090

We can observe that we obtain a reduction on the performance of the algorithm once we consider the whole input dataset (biased), especially regarding the precision score. This means that we are classifying a big number of "low severity" accidents (approximately a 65%) as being "high severity", a similar proportion to the one between both categories. Accuracy is also a little bit lower, but the difference is small, and the same effect can be observed with the recall score.

In summary, with the best performing methods, with fine tuned parameters, we can hardly detect between 55.41% and 61.08% of the "high severity" accidents properly, and on the other side we wrongly consider between 50.34% and 56.26% of the "low severity" accidents as being "high severity".

These results show that we can't properly identify the severity degree of an accident taking into account only features that are related to "external conditions" (road, light and weather) or the time period in which the accident happens.

- When applying the model to the whole dataset (biased) we observe that accuracy and recall scores remain quite similar to the ones we obtained during the previous testing phase, but precision drops dramatically. This is caused by a great number of events belonging to the most present category, and it is proportionally worse with higher recall values (the more events we properly detect belonging to the category of interest, the increase in the wrongly classified events from the unwanted category grows more than proportionally). This might make us not choose the model that classifies accurately more events on the interest category but that keeps a better balance between both (as wrong detection of unwanted events might cause an increase on the number of resources needed that might be unaffordable).
- It is not possible to accurately predict the severity of a car accident based on the feature set chosen, so it is necessary to use either more features or a completely different feature set.

7. Appendices

7.1. Dataset Information

Attribute information about the dataset used can be found in the following document, published by the owner of the data (Seattle Police Department):

https://www.seattle.gov/Documents/Departments/SDOT/GIS/Collisions_OD.pdf