Predicting the severity of Car Accidents in Catalonia

Marc Bara

August 25th, 2020

1. Introduction / Problem

Road accidents are a major world economic and social problem of loss of lives and properties in many countries around the world. Key reports indicate the number of fatalities from road accidents per year of about 1.3 million and 50 million injuries [1] or an average of 3000 deaths/day and 30,000 injuries/day. Furthermore, its consequences have an impact on economic and social conditions in terms of health care costs of injuries and disabilities. The World Health Organization (WHO) [2] estimated the economic costs derived from road accidents reached 518 billion USD per year in high income countries and 65 billion USD per year in medium and low income countries.

It is true that the safety of vehicles has been increasing in the latest years/decades with the use of new materials and technology in vehicle manufacturing, and the awareness of this global problem is evident. Apart from this trend, there is an increasing need for including in this equation also road conditions. According to “Vehicle Safety 2018” by the European Commission, “Increasingly, vehicle systems which can integrate vehicle and road network interventions (integrated systems) are being pursued.”.

Indeed, there is a special interest in tackling this problem from all possible points of view. One of them is not only to try to reduce the number of total accidents, but trying to reduce the severity of the accidents that do occur as well. From this perspective, we may think of a possible machine learning system, operated by local/regional administration, that could enable decision making in near-real time. Depending on local road conditions, traffic data, weather conditions, type of roads, day of the week, etc., the system will try to predict the severity of the accident; severity meaning possible fatalities involved. Depending on the maturity of this system, it could be also embedded on board the vehicle dashboard to support driver decisions (as avoiding high-risk situations or conditions in route planning). Based on existing databases, we propose to create a machine learning algorithm to warn us about the probability of fatalities happening, in case of accident.

2. Data Source

Given the problem to tackle, we will be using relevant data with all possible descriptors that help our algorithm to succeed. In particular, we will focus on the area of Catalonia, in Spain, including Barcelona and its region.

The data set to be used for algorithm training and testing is here:

https://analisi.transparenciacatalunya.cat/Transport/Accidents-de-tr-nsit-amb-morts-o-ferits-greus-a-Ca/rmgc-ncpb

It consists of a structured data set with information on traffic accidents with deaths or serious injuries that have occurred in Catalonia since 2010. Among the data fields we find road, light and weather conditions when the accident occurred, number of fatalities or injured people, type of road, time/day of the week, etc. Quantitively speaking, it

records more than 16,000 accidents from 2010 to 2018, and for each of them we find 58 fields.

3. Data cleaning

The data was already pre-cleaned, as taken from the source. No “NaN” values were identified.

4. Data visualization

As explained in the data source reference, the data set consists of 58 features recorded from more than 1600 accidents in the area of Catalonia. It will be necessary for us to first visualize important contents of the data set, to get our first insights.

First we want to see the number of accidents per year:

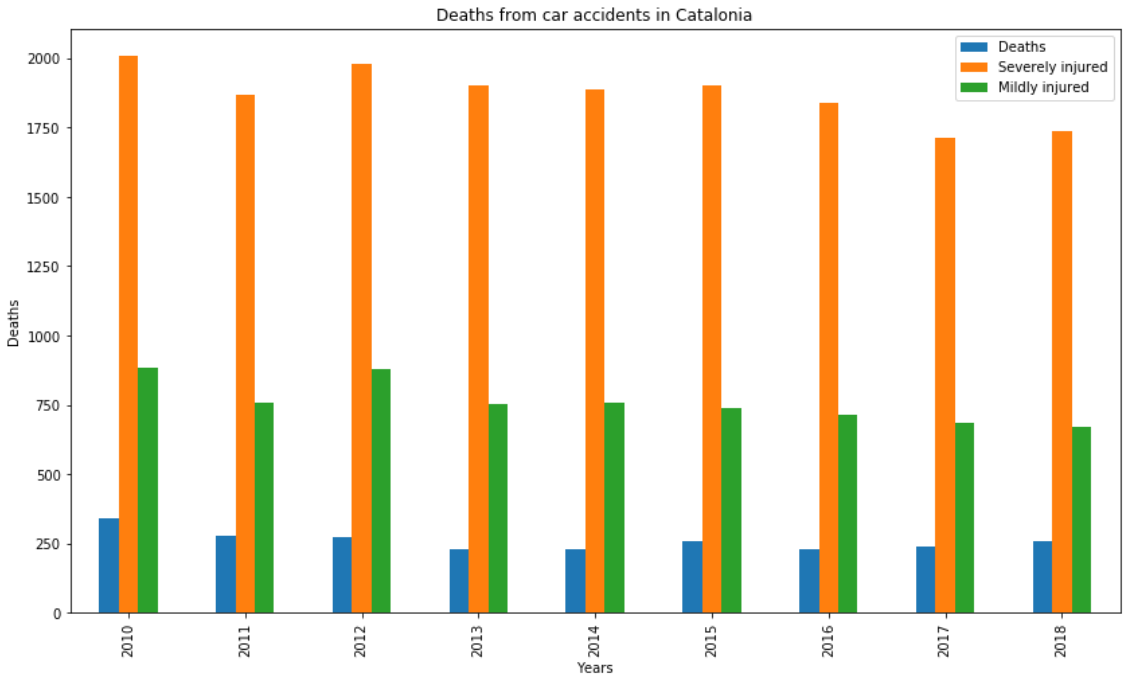

Figure 1. Accidents per year

Another interesting factor is, among all these accidents, which resulted in at least one fatality. This is shown in the next pie chart. As we can see, about 12% of the cases can be considered as a “1”, or high severity. Please notice that this is not close to 50%, so this means that our algorithm will not be trained with the same number of “1”s and “0”s, meaning this will have to be balanced during our model generation.

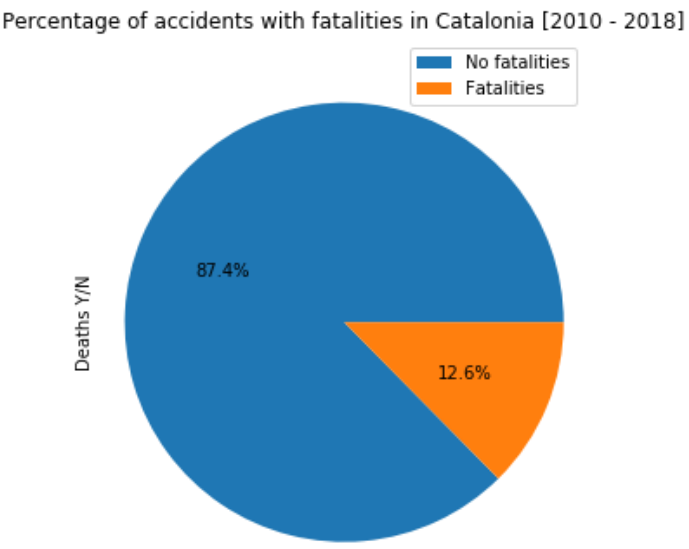

Figure 2. Percentage of accidents with fatalities in Catalonia

Next, it will be interesting to explore different features, like the percentage of accidents that occurred during weekends, or different parts of the day, or grouped according to province in Cataloina. This is shown in the following charts:

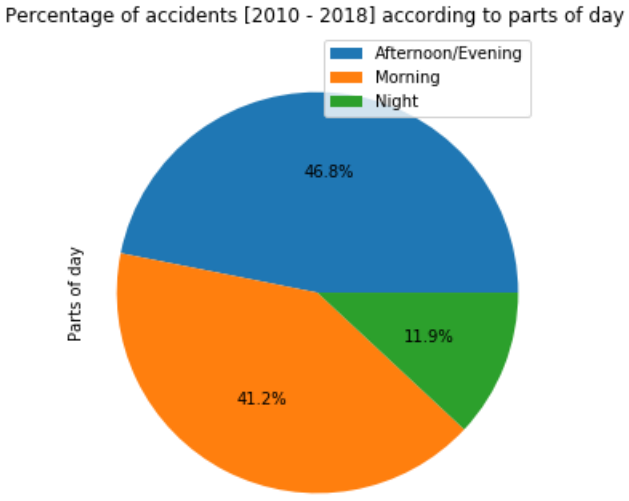

Figure 3. Percentage of accidents according to parts of day

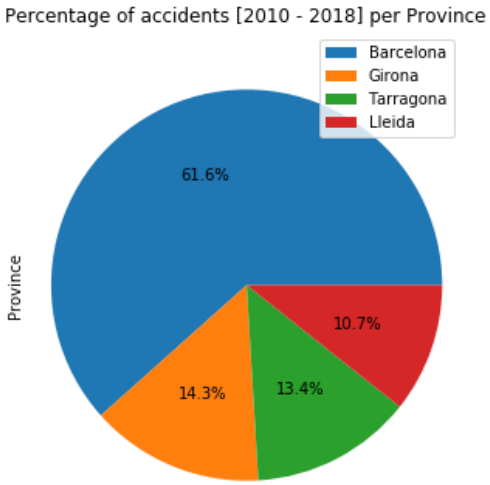

Figure 4. Percentage of accidents per Province

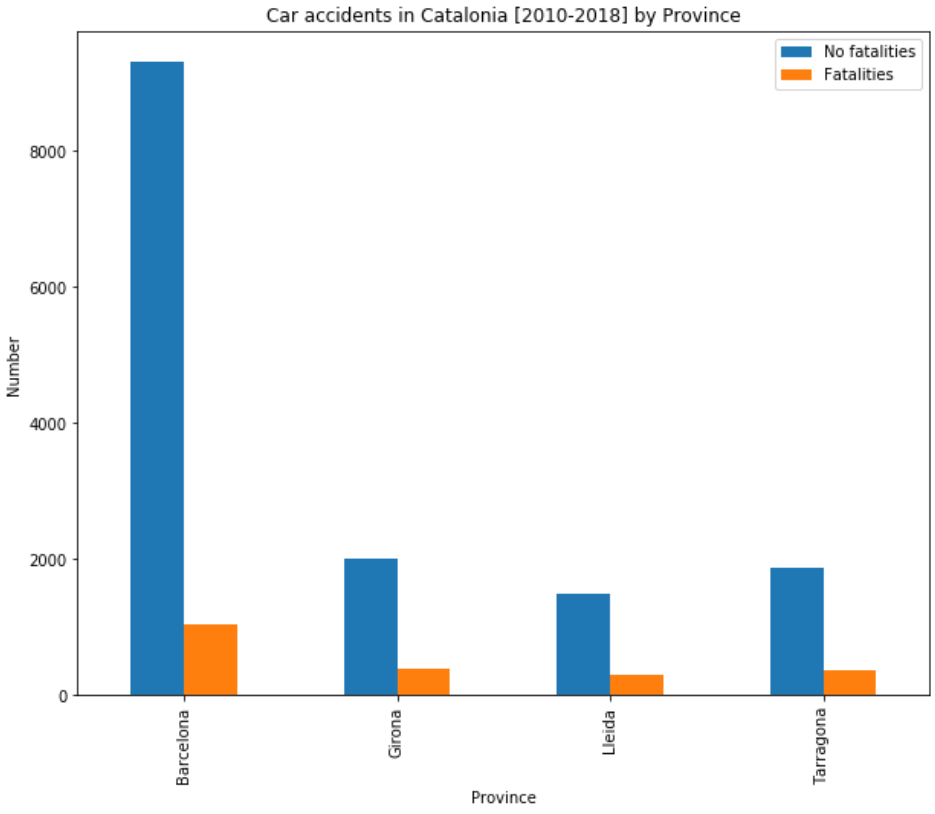

Figure 4. Number of accidents per Province and severity

5. Selected technique: Logistic Regression

Our decision has been to apply a Logistic regression model, which is very appropriate to determine a 0/1 type of prediction. We want to simplify the severity of the accident in terms of fatality occurred (1) or not (0).

We create target variable Y that it's 1 if there are any number of fatalities, 0 if none:

```
Y = (df_data_1['F_MORTS'] > 0).astype(int)
```

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model (a form of binary regression).

6. Feature selection

A common problem in applied machine learning is determining whether input features are relevant to the outcome to be predicted. This is the problem of feature selection.

In the case of classification problems where input variables are also categorical, we can use statistical tests to determine whether the output variable is dependent or independent of the input variables. If independent, then the input variable is a candidate for a feature that may be irrelevant to the problem and removed from the dataset.

So in this case we used the so-called “Contingency Table”. In statistics, a **contingency table** (also known as a cross tabulation or crosstab) is a type of **table** in a matrix format that displays the (multivariate) frequency distribution of the variables. They are heavily used in survey research, business intelligence, engineering and scientific research. Based on this table, we have determined the most dependent variables of the data set that may provide us with insights about the logistic regression problem. This contingency table, for each of pair variables, has been submitted to a **Pearson’s chi-squared statistical hypothesis test.**

The result is the following data set:

```
cat_df=df_data_1[[
        'tipDia', 'grupHor', 'nomDem', 'zona',
        'D_TIPUS_VIA','D_SENTITS_VIA','D_INTER_SECCIO','D_LLUMINOSITAT',
        'D_CARACT_ENTORN','C_VELOCITAT_VIA',
        F_VEH_PESANTS_IMPLICADES']]
```

With 11 dependent variables as features for the model.

7. Model creation and training

We have used a training and 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. Our code in Python selects 85% of the data set for training purposes, with the remaining 15% for testing.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.15)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

8. Model evaluation

To evaluate the model we have used, first, a confusion matrix, also known as an error matrix. This is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

In our case we find that the true positives are 69%, and the true negatives 62%. See the next figure with our results.

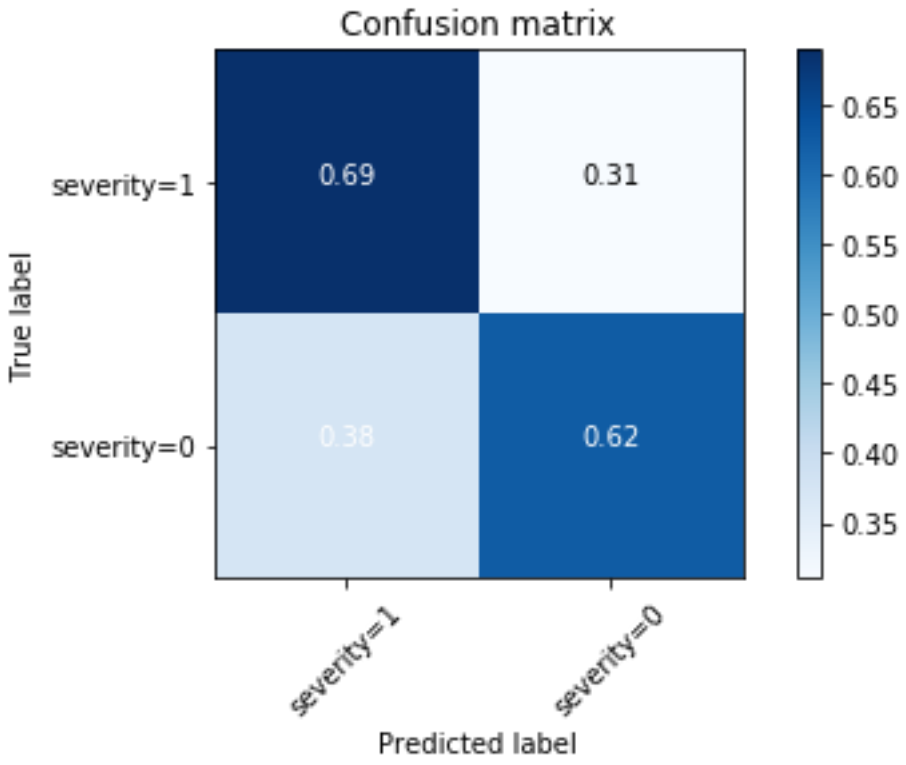

Figure 5. Confusion matrix with the results for the logistic regression model

Based on the count of each section, we can calculate precision and recall of each label:

- Precision is a measure of the accuracy provided that a class label has been predicted. It is defined by: precision = TP / (TP + FP)
- Recall is true positive rate. It is defined as: Recall = TP / (TP + FN)

So, we can calculate precision and recall of each class.

	precision	recall	f1-score	support
0	0.93	0.62	0.75	2205
1	0.21	0.69	0.32	312
micro avg	0.63	0.63	0.63	2517
macro avg	0.57	0.66	0.53	2517
weighted avg	0.84	0.63	0.70	2517

Now we are in the position to calculate the F1 scores for each label based on the precision and recall of that label. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. We can tell the average accuracy for this classifier is the average of the F1-score for both labels, which is 0.70 in our case.

Since our model not only predicts a “0” or a “1”, but a full vector with all the probabilities of being a “0” or a “1”, we may be losing information only with the confusion matrix. That is, in a real world application, we may use not a binary classifier, but a number in percentage that could tell us that, in case of an accident, what the probability of fatality is. That’s why it’s important to measure our logistic regression classifier with some other parameter, like the ROC curve explained in [3].

ROC curves can be more flexible to predict probabilities of an observation belonging to each class in a classification problem rather than predicting classes directly.

This flexibility comes from the way that probabilities may be interpreted using different thresholds that allow the operator of the model to trade-off concerns in the errors made by the model, such as the number of false positives compared to the number of false negatives. This is required when using models where the cost of one error outweighs the cost of other types of errors.

A diagnostic tool that helps in the interpretation of probabilistic forecast for binary (two-class) classification predictive modeling problems are ROC Curves. ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.

In our case, the computation of the ROC curve gives us the following figure.

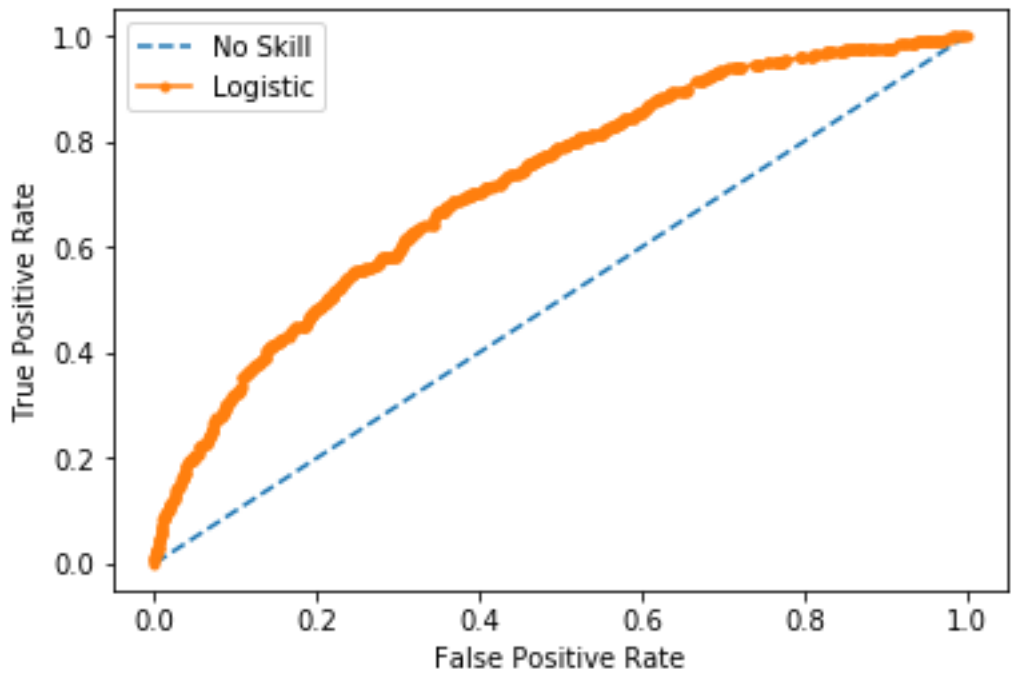

Figure 6. ROC Curve, comparing our algorithm with a “no skill” algorithm (random classification)

The ROC curve is a useful tool for a few reasons:

- The curves of different models can be compared directly in general or for different thresholds.
- The area under the curve (AUC) can be used as a summary of the model skill.

The shape of the curve contains a lot of information, including what we might care about most for a problem, the expected false positive rate, and the false negative rate. An operator may plot the ROC curve for the final model and choose a threshold that gives a desirable balance between the false positives and false negatives.

A skillful model will assign a higher probability to a randomly chosen real positive occurrence than a negative occurrence on average. This is what we mean when we say that the model has skill. Generally, skillful models are represented by curves that bow up to the top left of the plot.

A no-skill classifier is one that cannot discriminate between the classes and would predict a random class or a constant class in all cases. A model with no skill is represented at the point (0.5, 0.5). A model with no skill at each threshold is represented by a diagonal line from the bottom left of the plot to the top right and has an AUC of 0.5. In our case, our predictor is much better, and reaches a value AUC of 0.71.

```
No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.719
```

9. Conclusions and future directions

In this study, we have analyzed the relationship between fatalities in accidents and their associated conditions at the moment of occurrence, like time of day, day of week, weather/light/wind conditions, etc.

We have built a logistic regression model that, given certain conditions at every moment, predicts the probability of having fatalities in case of an accident, i.e., its severity as defined in this problem.. These models can be very useful in helping authorities in taking real-time decision to close roads, or give recommendations to drives via some panels or other technology. Also, it will be possible to integrate these algorithms in the vehicle for assisting drivers in route selection and overall decision making.

We believe that, given the available data, the algorithm gives notable results, and to improve it further we may need to incorporate other data sets which will enrich the features with more accurate weather conditions, etc. Multiple data sets (combined in a synchronous way) should be necessary to improve the algorithm performance.

10. References

[1] W.H. Organization
Global status report on road safety: time for action
World Health Organization (2009)

[2] M.M. Peden, W.H. Organization
World report on road traffic injury prevention
World Health Organization (2004)

[3] How to Use ROC Curves and Precision-Recall Curves for Classification in Python
https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/