# Predicting severity of road accidents in Belgium

## Introduction to the business problem

Car accidents are one of the major causes of death nowadays. Being able to better understand what causes them could greatly help reduce the number of accident and save lives. In this project we try to build predictors of what are the main causes of road accidents and more specifically predict the severity of said accident in term injuries or death of people involved. In this project we'll mainly focused of accidents occurring in Belgium between 2005 and 2019. The results could help various stakeholders:

* Help roads users (drivers, cyclist, pedestrians,) better understand factors that increase the risk of accident and adapt their behaviour accordingly.
* Help road safety agencies design awareness campaigns focusing on the most appropriate message
* Help emergency services better anticipate when sever accident are likely to occur to size response teams accordingly

Other more long term potentials could be considered like guiding the development of new or improved safety features in vehicles or on roads. Such algorithms could also help autonomous vehicles identify risky conditions and adapt the driving behaviour accordingly.

## Presentation of the data

To build our model we'll use historical car accident casualties’ data available on the Belgian statistical website [www.data.gov.be](http://www.data.gov.be). This data are available on an annual basis for every year between 2005 and 2019. and includes only the accidents with casualties expressed as "slightly injured", "severely injured" and/or "death" as well as the number of victims of each type for each accident. The dataset also contains various information about the accidents like:

* Date, day of the week and time
* Lighting conditions (day, sunrise/sunset, night with public lighting and night without public lighting)
* Location of the accident (address), type of neighbourhood (inside or outside city) and type of road (local, regional or highway)
* Identity of the victim (age category and sex) and type (driver, passenger, pedestrian, cyclist)
* Type of vehicle involved

Unfortunately some potentially intersecting features are missing. More details about weather conditions could be useful (rain, frost, snow,) but are not directly available. This information could however be reconstructed using historical weather data publicly available from weather data provider using date, time and location of each accident (e.g. from <https://dev.meteostat.net/>). Other missing information like was speeding, alcohol or drugs involved will probably be difficult to reconstruct. Age, time and day of the week of the accident could give us proxies of this but we probably need new dataset including these features to properly account for them.

The work was divided in 2 main parts.

* The analysis of the data were the data are imported and cleaned. And some statistical analysis are done to g better understand the data at hand
* The modelling of the data were we tried to design algorithm to predict the severity of an accident.

## Data description

### Preparation

#### Import

The raw data is available from Belgian statistical website as a set of files in csv or xlsx files. Each file was manually downloaded and the automatically read and combined into a single csv file using Python. The resulting file included 843 620 rows, each describing an accident was casualties that occurred in Belgium between 2005 and 2019. For each accidents 42 columns were available, however several columns included the same information, expressed first as an index, then in plain text in French and Dutch (the 2 national languages in Belgium). In this case, both text columns were discarded to keep only the index one.

Exact date of the accident was also replaced by month as it is quite unlikely that the severity of an accident would be linked to a specific date. However the month (and the associated season) could be an influencing factor. In total we kept the 10 following features:

* Month : 1 (January) - 12 (December)
* Hour of the year : 0 - … - 23
* Day of the week : 1 (Monday) - … - 7 (Sunday)
* Area: 1 (inside city) - 2 (outside city)
* Victim: 1 (Driver) - 2 (Passenger) - 3 (Pedestrian) - 4 (Other) - 5 (Cyclist) - 6 (Motorcyclist) - 7 (Moped rider)
* Age category of the victim: 1 (0 to 4 years) - … - 16 (75 year and more)
* Sex of the victim: 1 (Male) - 2 (Female)
* Road type: 1 (Highway) - 2 (Main road) - 3 (Secondary road)
* Light conditions: 1 (Daylight) - 2 (Sunrise or sunset) - 3 (Night time with light on) - 4 (Night time with light off)
* City: code between 0 and 595

#### Additional data

We also tried adding information from other sources, and more specifically information related to weather conditions at the time and place of the accident. Unfortunately after several attempts, we couldn’t find a reliable (and free) data sources that had enough accuracy and allowed us to go back in time all the way to 2005 with relevant weather information (e.g. rain, forts, fog,…). Therefore at this stage we decided not include that information even if we do believe it would still be relevant to do it.

#### Cleaning

Several rows were also incomplete in the sense that one or several pieces of information were missing. The missing information seemed to be distributed randomly over the data and given the size of the available dataset, we simply discarded them. In total 13% of the entries had missing data. This meant that after cleaning we still had 734 301 entries in the dataset.

The outcome of each accident was expressed using 3 columns:

* The number of people slightly injured in the accident
* The number of people severely injured in the accident
* The number of people killed in the accident

For the sake of the analysis, this information was converted into a single columns with 3 possible label:

1. If the accident only lead to slightly injured people
2. If the accident included at least 1 severely injured person but no death
3. If the accident lead to the death of at least 1 person

This columns would then serve as label for the machine learning algorithms that were developed in the following step.

### Statistical indicators

Before starting the modelling part, we had a look at different statistical indicators about our dataset.

#### Data description

The table below gives some basic description about the data at our disposal after cleaning. It was among other thing used to make sure that all remaining entries were complete i.e. max and min values were within the expected range.

Looking at the mean severity of 0.12 we notice that we’ve a largely unbalanced data were most of our cases are labelled 0, i.e. accident leading to slightly injured persons. This is good news for the road users, but something to watch out for in our modelling.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **Month** | 6.63 | 3.36 | 1 | 4 | 7 | 10 |  |
| **Hour** | 13.27 | 5.52 | 0 | 9 | 14 | 17 | 23 |
| **DOW** | 4.01 | 1.96 | 1 | 2 | 4 | 6 | 7 |
| **Area** | 1.46 | 0.50 | 1 | 1 | 1 | 2 | 2 |
| **Victim** | 2.70 | 2.04 | 1 | 1 | 2 | 5 | 7 |
| **Vehicle** | 7.65 | 8.24 | -8 | 1 | 2 | 16 | 28 |
| **Age** | 7.88 | 3.64 | 1 | 5 | 7 | 10 | 16 |
| **Sex** | 1.41 | 0.49 | 1 | 1 | 1 | 2 | 2 |
| **Road** | 2.31 | 0.63 | 1 | 2 | 2 | 3 | 3 |
| **Light** | 1.61 | 0.95 | 1 | 1 | 1 | 3 | 4 |
| **City** | 272.17 | 179.03 | 0 | 103 | 263 | 423 | 595 |
| **Severity** | 0.12 | 0.37 | 0 | 0 | 0 | 0 | 2 |

Existing correlation between features was also tested and the results are presented in the table below

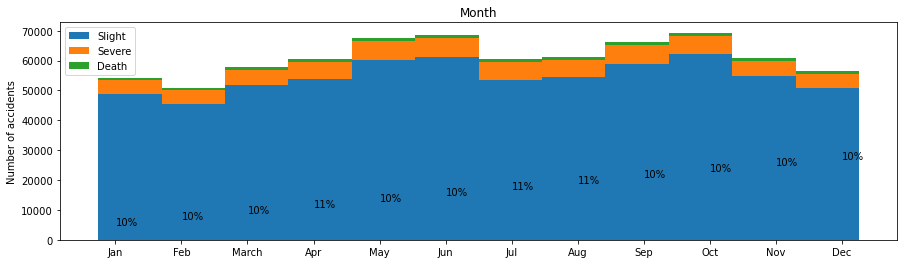
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Month | Hour | DOW | Area | Victim | Vehicle | Age | Sex | Road | Light | City | Severity |
| Month | 1.00 | 0.01 | 0.00 | -0.01 | 0.02 | 0.02 | 0.00 | 0.00 | 0.00 | 0.06 | 0.00 | 0.00 |
| Hour | 0.01 | 1.00 | -0.05 | -0.03 | 0.06 | 0.05 | 0.01 | 0.03 | 0.04 | -0.03 | -0.01 | -0.02 |
| DOW | 0.00 | -0.05 | 1.00 | 0.04 | -0.05 | -0.08 | -0.02 | -0.04 | -0.01 | 0.12 | 0.01 | 0.03 |
| Area | -0.01 | -0.03 | 0.04 | 1.00 | -0.24 | -0.31 | 0.02 | -0.03 | -0.39 | 0.06 | 0.09 | 0.09 |
| Victim | 0.02 | 0.06 | -0.05 | -0.24 | 1.00 | 0.76 | -0.04 | -0.13 | 0.21 | -0.16 | -0.02 | 0.05 |
| Vehicle | 0.02 | 0.05 | -0.08 | -0.31 | 0.76 | 1.00 | 0.02 | -0.09 | 0.25 | -0.17 | -0.03 | 0.06 |
| Age | 0.00 | 0.01 | -0.02 | 0.02 | -0.04 | 0.02 | 1.00 | 0.04 | 0.00 | -0.10 | 0.01 | 0.08 |
| Sex | 0.00 | 0.03 | -0.04 | -0.03 | -0.13 | -0.09 | 0.04 | 1.00 | 0.00 | -0.10 | 0.00 | -0.07 |
| Road | 0.00 | 0.04 | -0.01 | -0.39 | 0.21 | 0.25 | 0.00 | 0.00 | 1.00 | -0.06 | -0.02 | -0.03 |
| Light | 0.06 | -0.03 | 0.12 | 0.06 | -0.16 | -0.17 | -0.10 | -0.10 | -0.06 | 1.00 | 0.00 | 0.07 |
| City | 0.00 | -0.01 | 0.01 | 0.09 | -0.02 | -0.03 | 0.01 | 0.00 | -0.02 | 0.00 | 1.00 | 0.01 |
| Severity | 0.00 | -0.02 | 0.03 | 0.09 | 0.05 | 0.06 | 0.08 | -0.07 | -0.03 | 0.07 | 0.01 | 1.00 |

The most significant correlation the appear is a 0.755 correlation between vehicle type (car, bike, motorbike,...) and victim type (driver, passenger, pedestrian,...). This is quite understandable. In order to reduce the complexity of the modelling we decided to drop the vehicle feature as it doesn't bring much additional information. We decided to keep the victim feature as it allows us to make important distinction between driver and passenger for example.

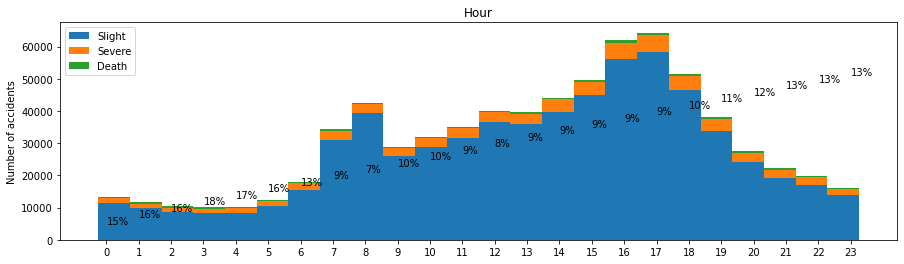
Other low correlations can be seen between Area and Vehicle and Area and Road. Again understandable as highways and motored vehicle like car and trucks are more frequently found highways and main roads outside cities. Whereas pedestrian and bikes typically remain inside city on secondary roads. However in this case the correlation is small and we deemed it worth keeping all three attributes. Other features seems very poorly correlated as well.

#### Data distribution

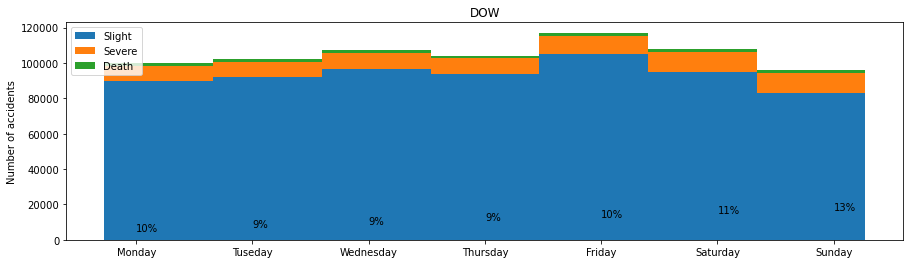
To investigate further how the different features explain accident severity we plotted the distribution of accident severity in function of the different features. In each plot we showed the distribution of the 3 types of accident severity in function of the feature. We also displayed the fraction of accident that were either sever or deathly. This percentage gives a 1st estimation of the influence the feature as on the severity of the accident.



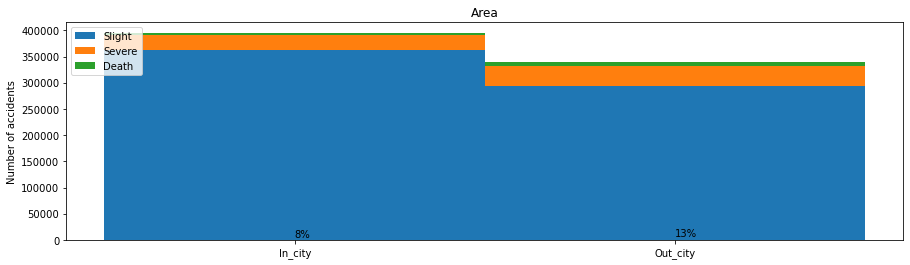
Surprisingly it seems that there is less accident in December, January and February than the rest of the year. We could expect more accident in the winter due to bad weather, but it could also be that there is less people on the road (especially pedestrians, bikers and motorbikes) around this time of the year, hence less accident globally). We also see that the proportion of sever and deathly accidents remain more or less constant at 10%. This was also visible in the very low correlation observed previously.



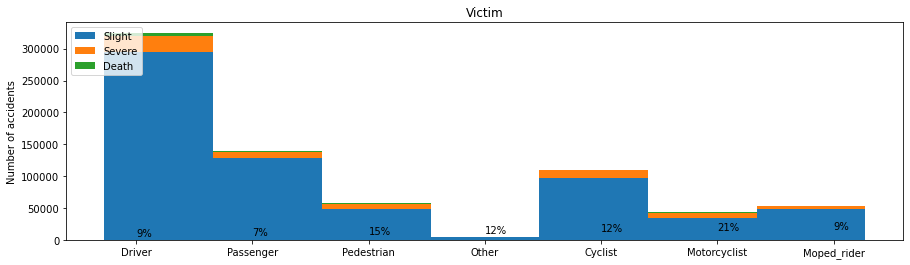
Most of the accident happens between 7am and 8am or 4pm and 6pm, probably during rush hour when most people are on the road. However Accidents seems to be more sever at night with 15% to 18% of sever and deathly accident between midnight and 5am than the rest of the day (8% - 13%).



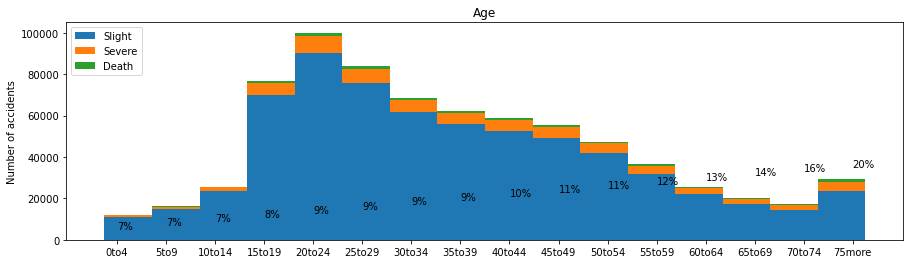
Friday is the day with the most accidents, however on Sunday accidents seems, on average, slightly worse.



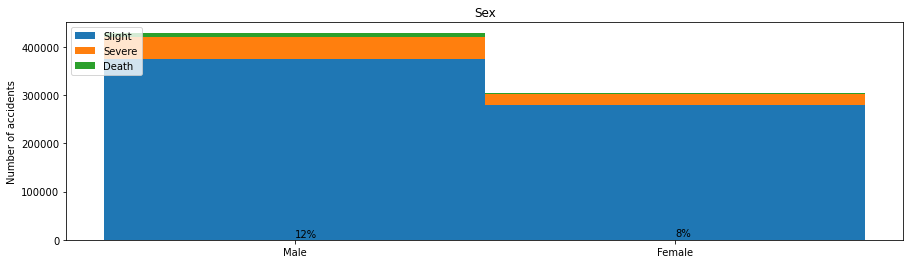
Most accidents happens inside city area, but the ones occurring outside tends to be worse.

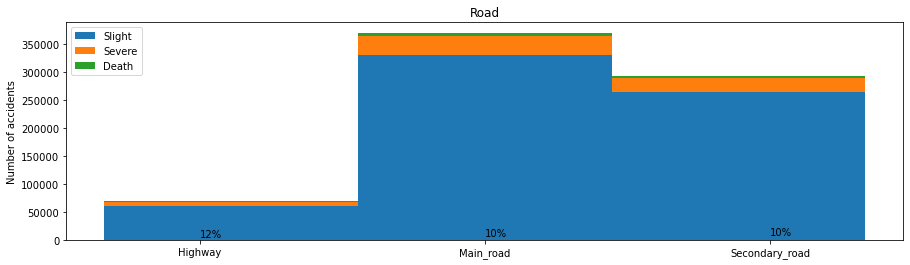


The type of user seems to have a significant impact on the severity of accidents. Motorcyclist are the most at risk during an accident, followed by pedestrian. Bikers are also slight worse off than car drivers, passengers or moped riders.

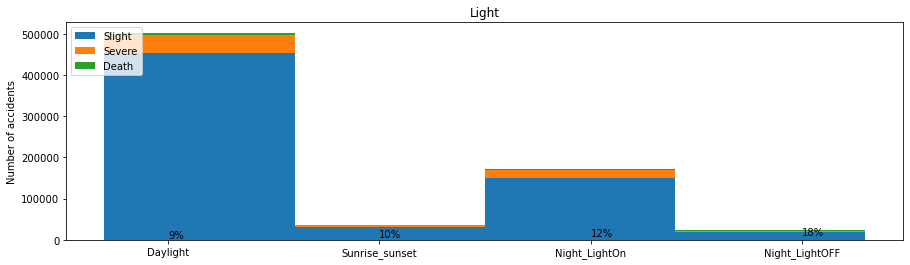


Young people between 15 and 30 years old are the most frequent victims of car accidents. However it seems that the consequence of said accident are worse when you get older. 20% of the people older than 75 years involved in an accidents suffered severe injuries or death whereas less than 10% for people younger than 40 years old.

Men seems to be both the most frequent and worst victims of accidents.



Only a very small number of accident happens on the highway. The vast majority take place on main roads or secondary roads. However highway accidents seems slightly worse that the rest.



Most accident occur during the day, but night accident tends to be more sever, especially if there was no street lighting.

## Modelling methodology

### Preparation

#### Scaling

Before training machine learning algorithm, each feature in the dataset was scaled in order to speed the optimisation processes of the different algorithms. This was done using the standard scaler where each feature value X is transformed into a new value X’ using the following formula:

Where µ is the mean value of the feature and σ the standard deviation.

#### Under sampling

As presented earlier, 89.3% of the accident are of type 0 (slight injury), 9.2% are of type 1 (sever injury) and only 1.5% are of type 2 (death). To avoid issue related to imbalanced data, we decided to under sample the dataset. This consists in removing some data of type 0 and 1 in order to end up with more or less the same number of data of each class. A basic strategy could consist in randomly removing entries with label 0 and 1. But this entail the risk of losing interesting information. For example we could end-up discarding all the data points within a specific region of our feature space. Making it impossible for our algorithm to properly predict accidents in this region.

Another option is to use the Cluster Centroids algorithm. This algorithm uses the K-mean clustering technic to replace entries in the 2 dominant classes by the centroid of a cluster around nearby entries of the class. This minimizes the risk of losing interesting information while rebalancing and reducing the size of the dataset. However applying this algorithm is still computationally expensive and wasn’t possible to apply on the present dataset given its size and the computational power available. Therefor a combination of random under sampling and cluster centroid under sampling was used:

1. We randomly reduced the “slightly injured” entries by a factor 50 and the severely injured by a factor 10 and the deathly by a factor 5. This meant a smaller dataset with 22140 examples (13108 slightly injured, 6745 severely injured and 2287 deathly)
2. We used cluster centroid under sample to balance the dataset with 2287 examples of each type. Hence a total of 6861 examples

This step ensured both a balanced dataset to ease the model training as well as a smaller dataset to speed computation and allow faster model training. Given the computational power available, it wouldn’t have been possible to train our models on the entire dataset anyway.

#### Splitting

After scaling and under sampling, the dataset was divided in 3 sub-dataset:

1. Training set (80% of the examples): this set will be used train the different machine learning algorithm and derive the model parameter that best predict the data
2. Test set (20% of the examples): this set will be used to evaluate the accuracy of the final model after training of the model and selection of its hyperparameters.

It is worth noting here that the test set will be kept to evaluate the final accuracy of the algorithm, after training of the parameters and selection of the hyperparameters. Selection of the hyperparameters will be done using grid search and K-Fold cross validation. This prevents the overfitting of both the model and its hyperparameters as well as a coherent comparison of the different algorithms on data that have been used neither for the training of the model nor for the selection of the hyperparameters.

#### Modelling

For this assignment, we tested 4 different machine learning algorithms:

1. Decision tree
2. K nearest neighbours
3. Support vector machine (SVM)
4. Logistic regression

For each of them we tested several hyperparameters and used macro averaged f1-score to select the best one. The hyper parameter selection is done using a grid search with 5-fold cross-validation.

First the dataset is used as is with its 3 classes (slight, severe and death). In a second step the problem was reduced to a 2-class problem where slight and sever accident are grouped together. This was done to see whether our data could be used to accurately predict deathly accident from the 2 other types.

## Results

### Modelling: 3-class classifiers

#### Decision tree

The 1st investigated algorithm is the decision tree where we tested tree depth between 1 and 20 layers. Both “Gini” and “Entropy” criteria were considered to evaluate to quality of a split in term of information gain. This was done using a grid search and a 5-fold cross validation. The graph below plots the f1-score for both criteria with different tree depths.

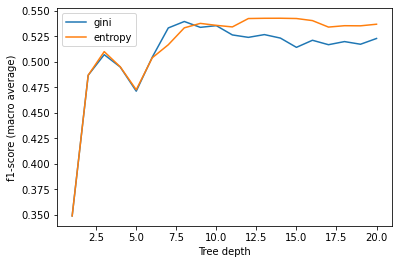


Figure 1:Evolution of decision tree accuracy

As we can see both criteria exhibit similar behaviour with first a rapid gain in f1-score to peak at a depth around 10-10 for Entropy and 10-15 for Gini. After the peak, the f1-score remains fairly constant when the depth keeps increasing. The best score is obtained with a Tree of 14 layers based on Gini impurity. To validate the model, we also checked its accuracy on the test set. The table and figure bellow give the corresponding precision, recall, f1-scores and confusion matrix. We see that the f1-score on the test set is 0.56 which is very similar to the accuracy we got based on the training set. This is a good indicator that we’re not overfitting the model and that the K-fold cross validation used in the selection the model hyperparameters worked properly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decision tree classification report | | | | |
|  | Precision | Recall | f1-score | Support |
|  |  |  |  |  |
| Slight | 0.59 | 0.54 | 0.57 | 454 |
| Sever | 0.43 | 0.38 | 0.40 | 449 |
| Death | 0.63 | 0.75 | 0.68 | 470 |
|  |  |  |  |  |
| Accuracy | |  | 0.56 | 1373 |
| Macro avg | 0.55 | 0.56 | 0.55 | 1373 |
| Weighted avg | 0.55 | 0.56 | 0.55 | 1373 |

It also appears that sever accidents are the most difficult to predict. This is understandable as they are likely to exhibit feature close to either slight or deathly accident and therefore more difficult to classify. On the other hand deathly accident are predicted the best. 75% of the deathly accident are correctly predicted and 63% of the accidents that are predicted as deathly are actually deathly. If or goal was to separate deathly from non-deathly, we could probably improve the accuracy by join the slight and sever classes together.

However we still see that overall our algorithm is not doing a particularly good job at classifying accidents and that roughly 44% of entries are misclassified. This could either be because our decision tree is not a good choice for this problem or because we’re missing some crucial information in the dataset. The performances of other algorithm should help us decide what is the most likely cause.

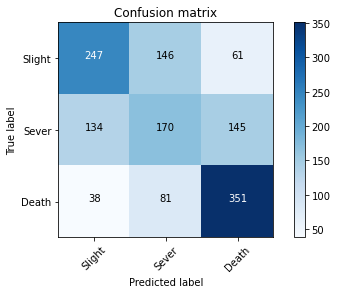


Figure 2: Decision tree confusion matrix

#### K Nearest Neighbour

For this algorithm we also performed a grid search to find the best hyperparameters with a 5-fold cross validation. The following parameters were considered:

* Number of neighbours: between 1 and 5
* Weight function:
  + Uniform weight for all neighbour
  + Weight neighbour based on the invers of their distance to give more influence to nearby neighbours
* Power for the distance metric: 1 (Manhattan distance), 2 (Euclidean distance)

The best model was found when using 4 neighbours to classify new data, using Manhattan distance (p=1) and a uniform weighting for the contribution of each point. Using these settings to predict the label of the test gave an macro averaged f1-score of 33%. The classification report and confusion matrix are given below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 nearest neighbours classification report | | | | |
|  | Precision | Recall | f1-score | Support |
|  |  |  |  |  |
| Slight | 0.31 | 0.36 | 0.33 | 454 |
| Sever | 0.21 | 0.18 | 0.2 | 449 |
| Death | 0.45 | 0.44 | 0.45 | 470 |
|  |  |  |  |  |
| Accuracy | |  | 0.33 | 1373 |
| Macro avg | 0.32 | 0.33 | 0.33 | 1373 |
| Weighted avg | 0.33 | 0.33 | 0.33 | 1373 |
|  |  |  |  |  |

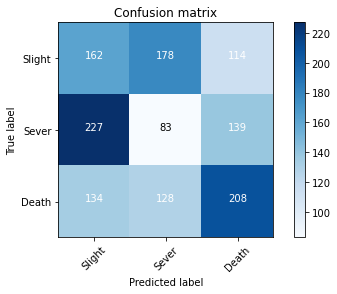


Figure 3: 3 nearest neighbours confusion matrix

The K nearest neighbours algorithm seems to perform very poorly, close to a random classifier with a 33% change of guessing right given the rebalanced nature of the dataset. It looks like K nearest neighbour isn’t a particularly good quantitate for our problem, especially when it comes to predict sever accidents.

#### Support vector machine

To train the model and select the best parameters we again used a grid search to select both the regularization parameter and the kernel type. The following options were considered:

* Regularization: 0.001, 0.01, 0.1, 1, 10, 100
* Kernel: Linear, Polynomial (degree 3), Radial basis function (RBF) and Sigmoid functions

The gird search identified the RBF kernel with a regularization coefficient of 0.1 as the best hyperparameters. With such a model, we got a f1-score of 45% on our test set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SVM classification report | | | | |
|  | Precision | Recall | f1-score | Support |
|  |  |  |  |  |
| Slight | 0.47 | 0.43 | 0.45 | 454 |
| Sever | 0.38 | 0.34 | 0.36 | 449 |
| Death | 0.47 | 0.56 | 0.51 | 470 |
|  |  |  |  |  |
| Accuracy | |  | 0.45 | 1373 |
| Macro avg | 0.44 | 0.44 | 0.44 | 1373 |
| Weighted avg | 0.44 | 0.45 | 0.44 | 1373 |

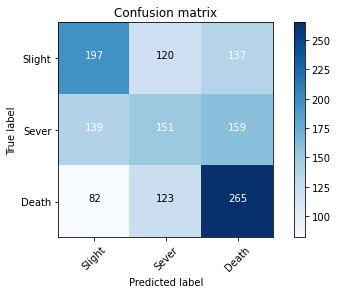


Figure 4: SVM confusion matrix

The SVM algorithm exhibit a behaviour somewhat similar to the decision tree. Sever accidents are the hardest to predict correctly. Deathly ones are predicted a bit better, but not as good as with the decision tree. And overall the accuracy remains disappointing with less than 50% of correct predictions.

#### Logistic regression

This was the last model tested. Grid search was again used to select the best macro-parameters for the model within the following space:

* Regularization: 0.001, 0.01, 0.1, 1, 10
* Optimisation solver: Newton-cg, LBFGS, sag and saga
* Multi class was handled using both one-versus rest and multinomial approaches

The best fitting model was obtained with a regularization equal to 0.1 using the newton-cg solver and the multinomial methodology for the multi-class classification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Logistic regression classification report | | | | |
|  | Precision | Recall | f1-score | Support |
|  |  |  |  |  |
| Slight | 0.45 | 0.49 | 0.47 | 454 |
| Sever | 0.40 | 0.30 | 0.34 | 449 |
| Death | 0.47 | 0.54 | 0.50 | 470 |
|  |  |  |  |  |
| Accuracy | |  | 0.45 | 1373 |
| Macro avg | 0.44 | 0.45 | 0.44 | 1373 |
| Weighted avg | 0.44 | 0.45 | 0.44 | 1373 |

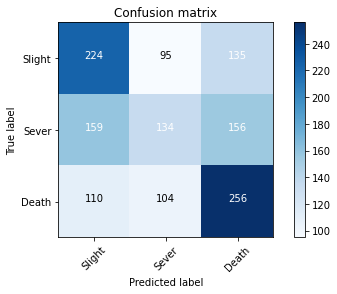


Figure 5: Logistic regression confusion matrix

Again this algorithm exhibit a similar behaviour as the decision tree and the SVM. But is even poorer than both of them to predict deathly accident.

### Modelling: 2-class classifiers

As we saw, decision tree appears to be relatively accurate in predicting deathly accident. Therefore a final attempts was made to make a classifier able to accurately predict whether an accident would be deathly or not. In this step, slight and sever accident are bundled together into the same category. The same methodology as the one described above is used to train and select the hyperparameters of the different algorithms. The table below summarizes the f1-score for each of them:

|  |  |  |
| --- | --- | --- |
| F1-score for the 2-classes classification problem | | |
|  | F1-score (training set) | F1-score (test set) |
| Decision tree | 87% | 87% |
| K nearest neighbours | 60% | 60% |
| Support vector machine | 66% | 67% |
| Logistic regression | 65% | 66% |

As we already anticipated from the 3-class classification problem, the decision tree is quite efficient is predicting deathly accident. Especially if It doesn’t need to distinguish slight from sever accidents. Interesting to know that the tree depth returned by the optimisation is significantly larger in this case (18 layers) than for the 3-class problem (14 layers). The classification report and confusion matrix of the model are given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decision tree classification report (2-class classifier) | | | | |
|  | Precision | Recall | f1-score | Support |
|  |  |  |  |  |
| Slight or Sever | 0.89 | 0.84 | 0.86 | 454 |
| Death | 0.85 | 0.89 | 0.87 | 461 |
|  |  |  |  |  |
| Accuracy | |  | 0.87 | 915 |
| Macro avg | 0.87 | 0.87 | 0.87 | 915 |
| Weighted avg | 0.87 | 0.87 | 0.87 | 915 |

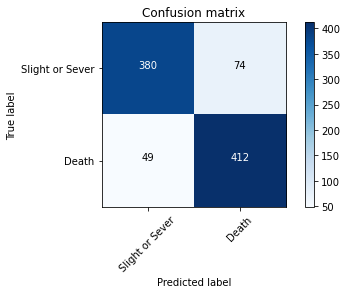


Figure 6: Decision tree confusion matrix (2-class classifier)

## Discussion

Out of the four algorithms presented in the previous, it seems that the decision is the one performing the best. The table bellow compare the f1-scores obtained for each of them on both the training and test sets respectively.

|  |  |  |
| --- | --- | --- |
| F1-score for the 3-classes classification problem | | |
|  | F1-score (training set) | F1-score (test set) |
| Decision tree | 54% | 55% |
| K nearest neighbours | 34% | 33% |
| Support vector machine | 44% | 44% |
| Logistic regression | 44% | 44% |

The good agreement between the train and test scores indicates that none of the model has been overfitted. We also see that the decision tree appears to be the best performing model while K nearest neighbours is clearly the worst. However none of the model appears to perform particularly good. We could try other algorithm. But it seems very unlikely that we will get better results. The most likely cause for this low accuracy is that we don’t have enough information to properly predict accident severity. Given the size of the dataset it is unlikely that gathering addition examples will significantly improving accuracy if at all. The best option would be the add additional features. As already mentioned in the introduction good quantitates could be:

* Detailed information about weather conditions (rain, fog, snow, frost, storm,…). We tried to reconstruct this information using weather services however none of the services we tried were accurate enough. And most of them were not free.
* Whether or not drugs, alcohol or other substances were involved
* Whether or not speeding or reckless behaviour were involved

Before going further in term of data modelling the recommendation would be to discuss with road safety expert (traffic police, emergency services,…) to see what would be the best features to add and how could we efficiently gather them (if not already available somewhere).

In the meantime, the decision tree remains the best performing algorithm. And we also saw that decision tree could do quite a good job in predicting whether an accident is deathly or not. In fact if we group slight and severe accidents into a single class and we retrain a decision tree and this new dataset we get much better results. In such a case, the accuracy increase to 87% which could already be quite useful in practice. The accuracy of the other algorithm also improve as we could expect but we never got accuracy better than 67% which is only marginal better than a random guess.

## Conclusion

In this work we tried to evaluate the possibility to predict accidents severity using machine learning. This was done using data of road accident in Belgium between 2005 and 2019. The data was first cleaned and analysed using statistical tools. This highlighted that the type of road user and age appeared to be significant factors in dictating the outcome of accident. Motorcyclist, pedestrian and old people tends to suffer, on average more severe consequences of accidents than the rest of the road users. Accidents occurring at nigh (0:00 to 6am) and outside cities (on highways) also appear worse. Men also seems slightly more impact by accidents than women. Other variables like presence of light, day of week and month seemed to have less impact on the accident severity. Overall it didn’t appear like there was a specific predominate factor dictating the accident severity.

In a second step 4 different classification algorithms were tested. To avoid issues related to imbalanced dataset we performed under sampling to have the same number of entries for each class (slight, severe and deathly). Standard scaling was used to accelerate the different optimisations. The resulting dataset was then split into a train set (80% of the data remaining after under sampling) and a test set (20%).

Out of the 4 algorithms, the decision tree appear to be the best choice. However none of them was doing a particularly good job at predicting accident severity on the 3-class scale (slight, severe, death). By reducing the problem to a 2-class problem (slight or sever vs death) we were able to get a much better model using a decision tree that had a final accuracy of 87%.

In order to improve the model further and eventually get a model able to accurately distinguish slight from severe accident. We most likely need additional feature in our dataset. Probable quantitates are weather information, or involvement of alcohol, drugs, or speeding. Discussion with domain expert would be most welcome here.