

Predicting Car Accident in France

Laurent Cesari

October 2020

1. Introduction

1.1. Background

Although decreasing over the past 20 years, road / car accidents still represent a very significant number of deaths and injuries in most countries, impacting not only human lives but also national economies (health care, infrastructure, productivity...).

Worldwide, approximately 1.35 million people die each year as a result of road traffic crashes. Road traffic injuries are the leading cause of death for children and young adults aged 5-29 years. And Road traffic crashes cost most countries 3% of their gross domestic product (source: [World Health Organization](#)).

1.2. Problem

In 2019, nearly 3,500 people died on the roads of mainland France or overseas. The cost linked to personal accidents represent €39.7 billions, and overall cost of road accidents is estimated at €50.9 Billions representing 2.2% of the country GDP (source: [French road safety observatory](#)).

Thanks to the accident data provided by the French Government, our goal is to **build a model that will allow to predict the severity of an accident in France**.

This information can be useful:

1. For government or local institutions to prevent such accidents through a variety of measures: prevention, communication, renovation or enhancements of infrastructure...
2. For hospitals or healthcare services to anticipate the needs for medical services for certain periods of time and locations. Or when they are about to handle accident injuries and receive the accident detail from the emergency services.

2. Data description

2.1. Data Source

For this project, we will use a **dataset provided by the French Government on personal accident in France from 2005 to 2018**. This dataset can be downloaded on the portal dedicated to French public data sharing: data.gouv.fr. It actually includes accidents including any kind of vehicles (cars, trucks, motocybless, bus, bike, scooter, train...) and any kind of user involved (driver, passenger, pedestrian)

2.2. Data Description

This dataset provided for a period of 13 years (2005-2018) details regarding the accident, the location and environment, the user(s) and vehicule. It is actually shared in 4 files provided for each year:

- A file for the accident details: date and time, address / geolocation, lighting condition, type of intersection, weather conditions, type of accident,..
- A file for the location details: road type, traffic type, number of traffic lane, reserved lane, road inclination, surface condition...
- A file for the user details: type, location in the vehicle, severity of accident, sex, birth year, safety equipment, reason for traveling,...
- A file for the vehicle details: type, number of person in the vehicle, fixed of mobile obstacle collision, collision area, manoeuvre type,...

The detail of these columns and information is provided in the [appendix](#).

We will use for our prediction model the **severity of the accident** that is provided at the user level. This severity is provided on a scale of 4, that we may review and simplify.

These data was of course removed of any detail that could allow us to link them to the actual person involved in these accidents:

- There was no personal information, no “human factor” such as alcohol / dugs consumption, vehicle speed, use of mobile...The speed limit at the location of the accident was not mentioned neither.
- There was neither detail of the vehicle (such as engine power) nor identification of the vehicle (brand, model...) except for a generic ID number which just allowed to differentiate the different vehicles included in an accident)

It is to be noted that there hase been a change of qualification regarding the classification of accident severity in 2018, which makes the corresponding data not comparable with previous years. Out of precaution, I have preferred to remove the data for 2018.

In order to try to improve the model, I have also decided to add the information of holiday as well as the distinction between weekdays and weekends.

3. Methodology

3.1. Data Collection

Since the data is split in 4 files per year, I had to go through several steps to aggregate the data split in these 52 files into one single dataset:

- First, I set a function that concatenates for each kind of file (accident, vehicle, location and user) the different years (from 2005 to 2017).
- Then, I have modified the column name to make them more understandable,
- And finally, I merged the 4 files while ensuring we could keep as many accident details as possible. Since several people and vehicles could be involved in an accident, I decided to keep all the information related to each person linked to an accident provided that the severity of the accident was mentioned.

At the end of that process, I ended up with a dataset of 51 columns (including the severity) and 2,012,026 rows. Each row representing the detail of one person / victim linked to a vehicle, an accident and a location.

3.2. Data Exploration and Preparation

3.2.1. Check and remove duplicate rows

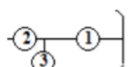
When checking for duplicate, I realized that a significant part of them (58% of the 2,505 rows identified as duplicates) might not be considered as real ones.

Indeed, when we look at the detail of the place in vehicle within these duplicate rows, we can see that more than 58% of them correspond to people involved in an accident while they were in a public transport (bus, coach, train, tramway). If we look at the data description, we see that we can have people being assigned to the same place in vehicle for public transport ('Transport en commun'), while this is not possible for car ('voiture') or moto / side-car.

			Transport en commun								
			4	7	7	7			7	7	1
			5	8	8	8			8	8	6
			5	8	8	8			8	8	6
			5	8	8	8			8	8	6
			3	9	9	9			9	9	2

Voiture		
4	7	1
5	8	6
3	9	2

Moto / Side-car



I have thus decided to keep them.

When looking at the rest of the duplicate rows, I realized that we also had around 500 users classified as pedestrians or rollerblade / scooter users, that could as well be considered as different victims included in the same accident.

As for the users with the same place in vehicle in public transport, I decided to keep them and to remove the remaining duplicate rows from the dataset.

3.2.2. Check and update missing values

Regarding the missing values, I decided to focus on attributes for which we miss more than 2% of the data, considering that we should be able to remove the missing rows without impacting significantly our model for the other ones (having less than 2% of missing information).

	na_perc
road_det2	95.74
road_det1	62.73
long	52.78
lat	52.78
gps	51.97
rm_dist	49.24
road_marker	49.08
address	16.85
road_id	6.99
place_in_veh	5.58
security	2.61

This filter listed 11 attributes. The 7 first attributes of this list had missing information for almost 50% or more of the data set (road details, gps code, longitude, latitude, road marker information...). Then, came the address (almost 17% of missing information), which presented a format (not formalized) that would be difficult to use for our model.

I wrote a function to find out if the missing information would be available for other persons / vehicles linked to the same accident number, but unfortunately it was not available for all these attributes. Hence, I decided to drop these columns considering I could use other attributes such as the county, city or road ID to manage the location.

For the 4 remaining attributes having more than 2% of missing data, I managed to replace the missing data by some alternative information:

- For the road id, I replaced the 7% of missing data by a combination of county code and city code. I also removed from a limited number of road IDs some characters corresponding to the road category that should have been removed.
- For the place in vehicle, the 5.6% of missing data actually corresponded to pedestrians and rollerblade / scooter users. So, I replaced the na values by the value 0 used for this category of users.
- For the security feature, the 2.6% of data missing was also corresponding to pedestrians and rollerblade / scooter users. So, I replaced as well the missing values by 0.

3.2.3. Review and simplify the data

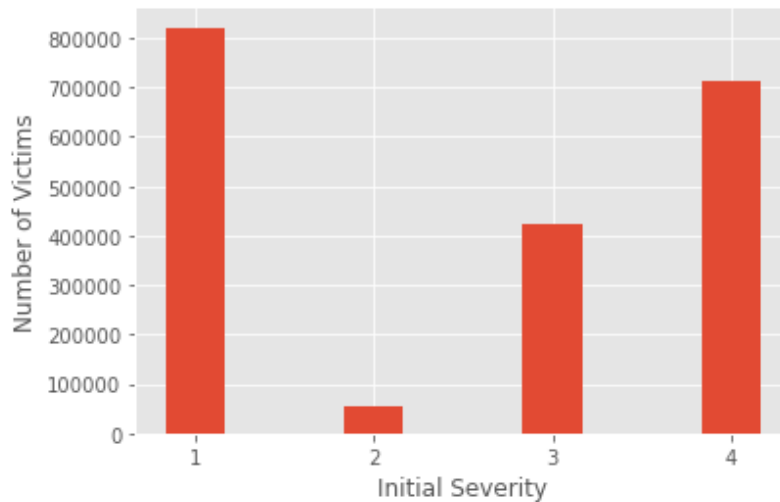
I had first a look at the distribution of the different attributes to identify which one(s) we could try to simplify without potentially impacting the accuracy of our model.



I focused first on the classification variable: the severity of the accident. As a reminder, there were 3 categories of injuries and 1 for the unarmed users:

1. Unarmed
2. Killed
3. Wounded - hospitalized
4. Light injury

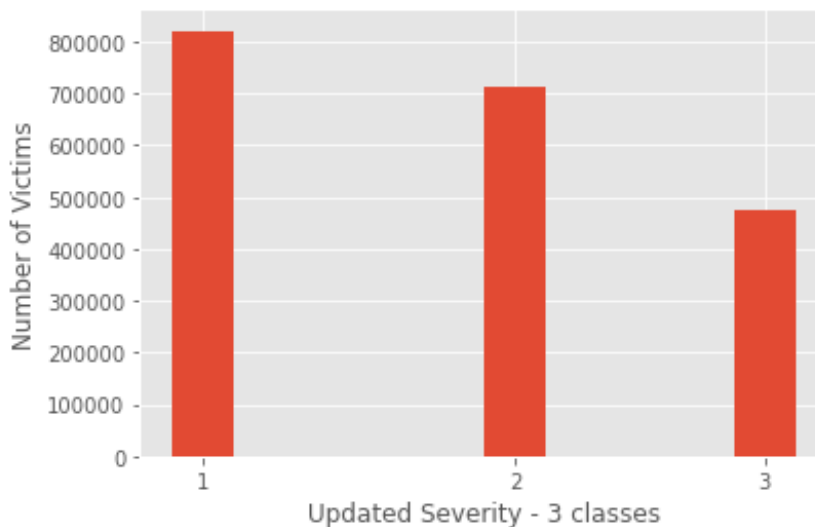
The distribution shows unbalanced labels especially for the higher severity "Killed", which might impact the performance of our model.



In order to get a more balanced distribution, I have decided to create another severity classification by combining the severity and 2 and 3 in a single category "Severly Injured or Killed", which leaves us with 3 levels:

1. Unarmed
2. Light Injury
3. Severly Injured or Killed

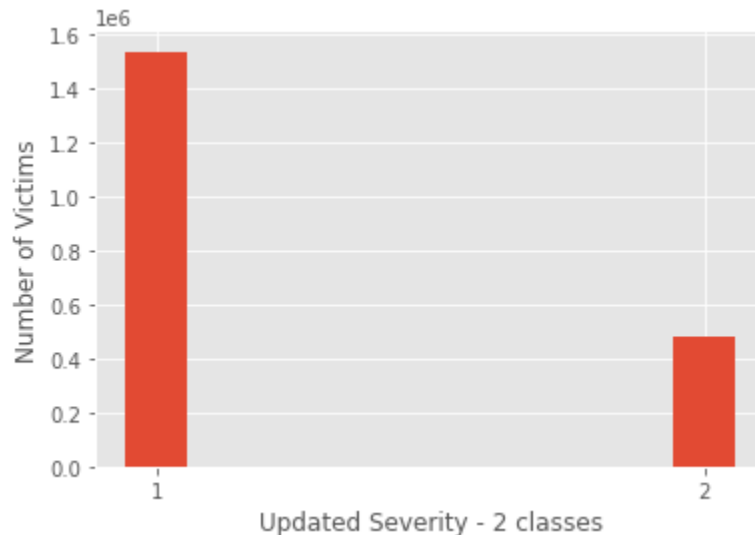
This new distribution appears indeed more balanced.



However, this classification shows 2 categories that could also be combined together so that we could differentiate the lowest severity accidents from the most severe ones with:

1. Unarmed or light injury
2. Severly Injured or Killed

Although it is less balanced than the second one, this distinction should allow to be more pro-active towards the situations where we have the most severe accidents.



To ensure our model(s) will have the best performance possible, I had decided to test them on these different classifications and to pick up the one showing the highest accuracy.

Then, I reviewed all the features and decided:

1. To simplify and / or update the following ones:

- Vehicle category: This attribute had several categories that were not used anymore and needed to be reclassified with the new corresponding category. In order to limit the number of categories, I also reclassified the categories which are linked to less than 1,000 users when they had shown similar breakdown of severity.
- Security: there were several values that were not 2-digit codes as they should be. I corrected them by adding the appropriate second digit.
- Number of lanes: I have reclassified the category representing missing information (0) with the most common number of lanes (2) which was also showing a similar distribution of severity. And to simplify this attribute, I have put the very limited number of accidents having occurred on a road of more than 6 lanes in a single category.
- Purpose: same as for the number of lanes, I have reclassified the category representing missing information (0) as well as the few missing values with the category "other purposes".

2. To add some additional features:

- Based on the year of birth provided with the data, I have added the age at the time of the accident, which might be more appropriate / better performing.
- Based on the feature day, I added the following attributes to see if they could help us to improve the performance of our model:
 - One attribute to identify the holidays, when we might have more traffic or different traffic condition depending on the location,
 - Two to differentiate weekdays and weekends where the traffic conditions can also be different.
- Based on the time of the accident provided with the format hhmm, I added these attributes hoping as well that they could improve the prediction accuracy:
 - Hour: to have the accidents time categorized by hour of the day,

- Time of the day: to differentiate morning, from afternoon, evening and night time.

3. To remove some features:

- Flow direction: the information was 0, which corresponds to missing or not mentioned for almost 90% of the data. Thus, I have removed this attribute from our dataset.
- Number of persons in the vehicle: the situation was similar for this attribute with almost 99% of the data not properly updated. I have removed this attribute as well.
- Median strip width: there was no median strip in almost 82% of the user accidents. Besides, the distribution is then spread on a high number of values (420). Last, the correlation and covariance with the severity were very low. So not only, will it be difficult to simplify this attribute but its impact on the accuracy of the model will also be very limited. Thus, I have dropped this attribute as well.

3.2.4. Select Features

First, and following our analysis in the previous section, we can select the attributes to drop because we consider they will not be significant or relevant to predict the severity of a future accident:

- The accident and vehicle identification numbers should obviously not being relevant to predict an accident in the future,
- Same for the full date of the accident and the year, knowing we already have attributes like the day, month, holiday status, period of the day...
- The features we identified in our previous section that we can remove: flow direct, number of persons in the vehicle, median strip width.

Then, I did a correlation matrix of the complete dataset.

The most significant correlation I could notice were:

- Corresponding to the attributes added like the different severity attributes and birth_year - age, which was expected,
- The features lighting and day_period, which was also understandable,
- To a lesser extend between area (urban / rural) and road_category with is also understandable.

When we focus on the correlation between the updated severity and the other features, none except maybe the area was standing out from the others.

	upd_sev_2
upd_sev_2	1.000000
upd_sev_3	0.831718
area	-0.192041
fixed_obst	0.159615
severity	0.156667
user_type	0.126557
loc	0.124147
mobile_obst	-0.115105
road_shape	0.109346
pedestrian_situation	0.108913
pedestrian_action	0.106718
pedestrian_loc	0.098013
veh_cat	0.092049
lane_numb	-0.081494
county	-0.070168

Thus, I decided to keep all these columns and to use the modeling and validating phase to select the most appropriate features.

Once we have finalized this step, we end up with a dataset of:

- 1,972,656 rows or observations,
- 3 potential classifications to train and select from,
- 42 potential features to train and select from.

3.3. Modeling

After having reviewed some articles and documentations, I decided to try 2 models based on Decision Tree which are known to perform quite well for similar classification of structured categorical data while limiting the impact of overfitting that the Decision Tree model has: the Random Forest Model and the XGboost Classifier.

3.3.1. Split the dataset

In order to be able to perform training, validation and finally testing of the model, I have split the dataset:

- Between the training and test sets,
- And for the training set, in one set that was used to train the model, and another one that was used as validation test to fine-tune the model hyperparameters.

Both splits were made with the 80/20 ratio.

3.3.2. Apply the Random Forest Model

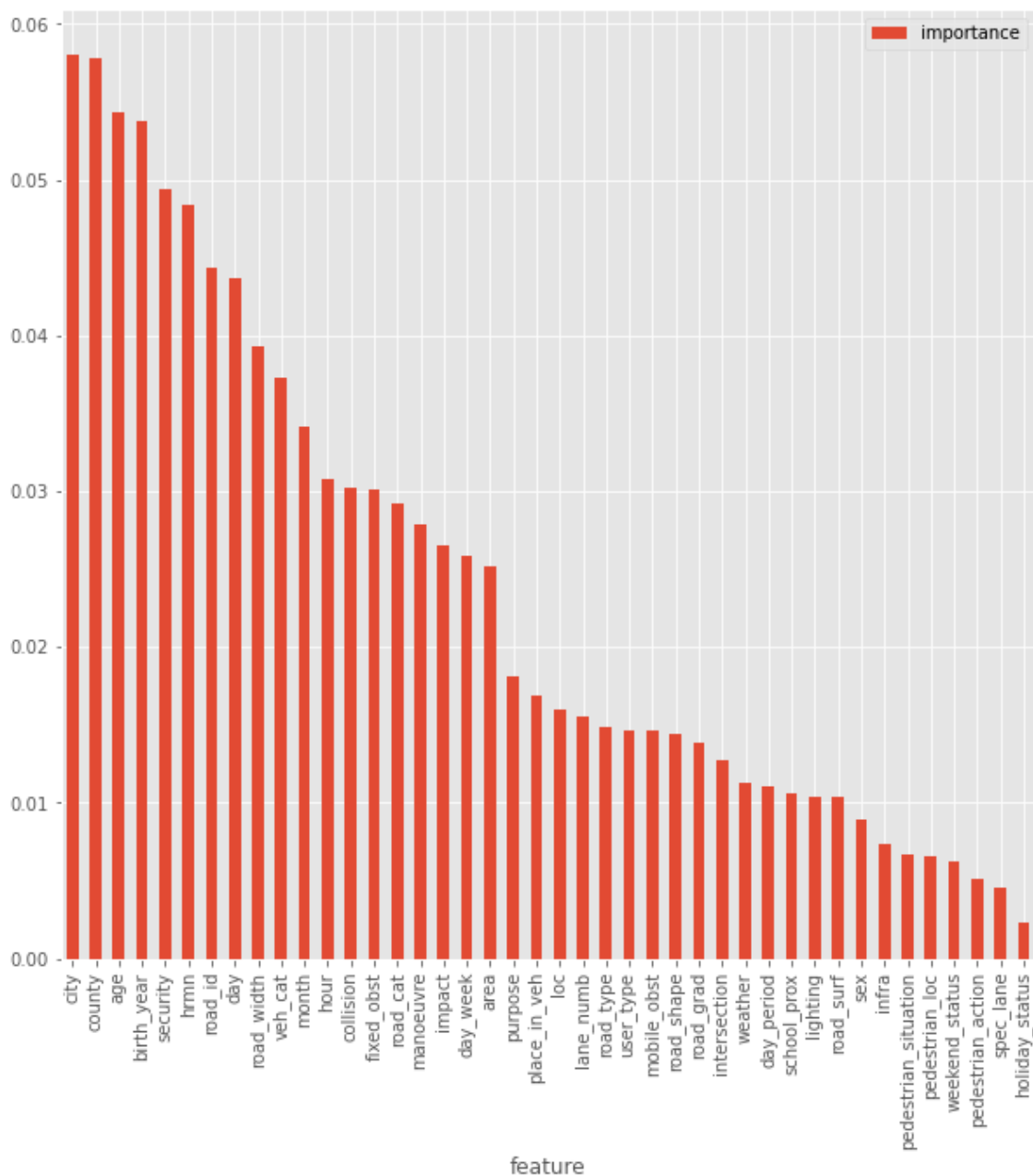
I trained the model with just a number of trees of 100 for the 3 severity classifications and got the following results:

Indicators	Initial Severity	3 Classes Severity	2 Classes Severity
Accuracy	0.6694	0.6912	0.8256
F1 Score	0.655	0.6855	0.8109

From these first modeling, we could see that the severity with only 2 classes is the best performing by a very significant margin.

Since having these 2 levels also makes sense regarding our goal to predict the severity for our potential "customers / stakeholders" (government, institutions, hospitals, healthcare systems...), I have selected this classification.

Then, I reviewed the most important features for the model and decide which ones we could keep.



Based on the review of the full list, I noticed that:

- The age we have added has slightly more weight than the initial birth year, although being quite close. However, since it did not bring anything more in terms of information, I removed it,
- The initial time of the accident seems more appropriate than the 2 features we had added (hour and period of the day), so I decided to drop them as well,
- The initial day looked as well to have more impact than the other attributes I had added. However, after several tests showing the holiday status and weekend status had some impact (although small), I decided to remove only the day of the week.

Once I had removed these features, I managed to improve the scores both from an accuracy and false positives / negatives standpoint:

Indicators	2 Classes Severity 42 features	2 Classes Severity 38 features
Accuracy	0.8256	0.827
F1 Score	0.8109	0.813
Wall Time	4 min 35	4 min

On the Kaggle cloud environment I used, the wall time for this modeling was around 5 min.

However, we were still using a significant number of features. So, I tried to limit the number of features by selecting the x first ones by order of importance for our model to see how it impacted our prediction scores:

Indicators	2 Classes Severity				
	5 features	10 features	15 features	20 features	25 features
Accuracy	0.746	0.8062	0.815	0.8236	0.8253
F1 Score	0.7305	0.7883	0.7993	0.8105	0.8117
Wall Time	2 min 45	3 min 57	3 min 49	4 min 27	5 min 11

So, depending on the prediction goals as well as the number of information we have available to predict the severity of an accident, we can adapt the selection of our features to maximize the accuracy of our model. Considering the gain in performance from 5 to 10 features, we might recommend to start, if possible, with a model having at least 10 or 15 features.

However, if we can get the maximum of our information, we can use the model with all the features since the processing time is not significantly different than the 20 or 25 features models.

I also made other tests by tuning some of the hyperparameters (like the number of trees, the maximum features, the maximum depth...) and managed to get some incremental improvement of the scores (especially with a higher number of trees):

- Max features = None did not improve the performance while significantly increasing the processing time:

Indicators	2 Classes Severity	
	Default max features	Max features None
Accuracy	0.827	0.826
F1 Score	0.813	0.816
Wall Time	4 min	21 min 04

- While the increase of the number of trees to 200 and 300 allowed a slight increase of the scores requiring a very significant increase of the processing time:

Indicators	2 Classes Severity		
	100 Trees	200 trees	300 trees
Accuracy	0.827	0.8278	0.8281
F1 Score	0.813	0.814	0.8143
Wall Time	4 min	10 min 3	14 min 7

When looking at the significant difference in processing time vs limited gain in accuracy, I decided to stay with 200 trees.

Unfortunately, considering the size of the dataset and the limit of my (old) computer and (free) cloud platforms (IBM, Google Colab and Kaggle) I was using, I did not manage to implement a proper fine tuning of the model hyperparameters with the grid search function or validation curves (memory error message).

3.3.3. The XGBoost Model

Since the random forest model had shown some quite good results, I decided try another model also based on decision tree and which has shown great and sometimes better performance than the Random Forrest on similar problem.

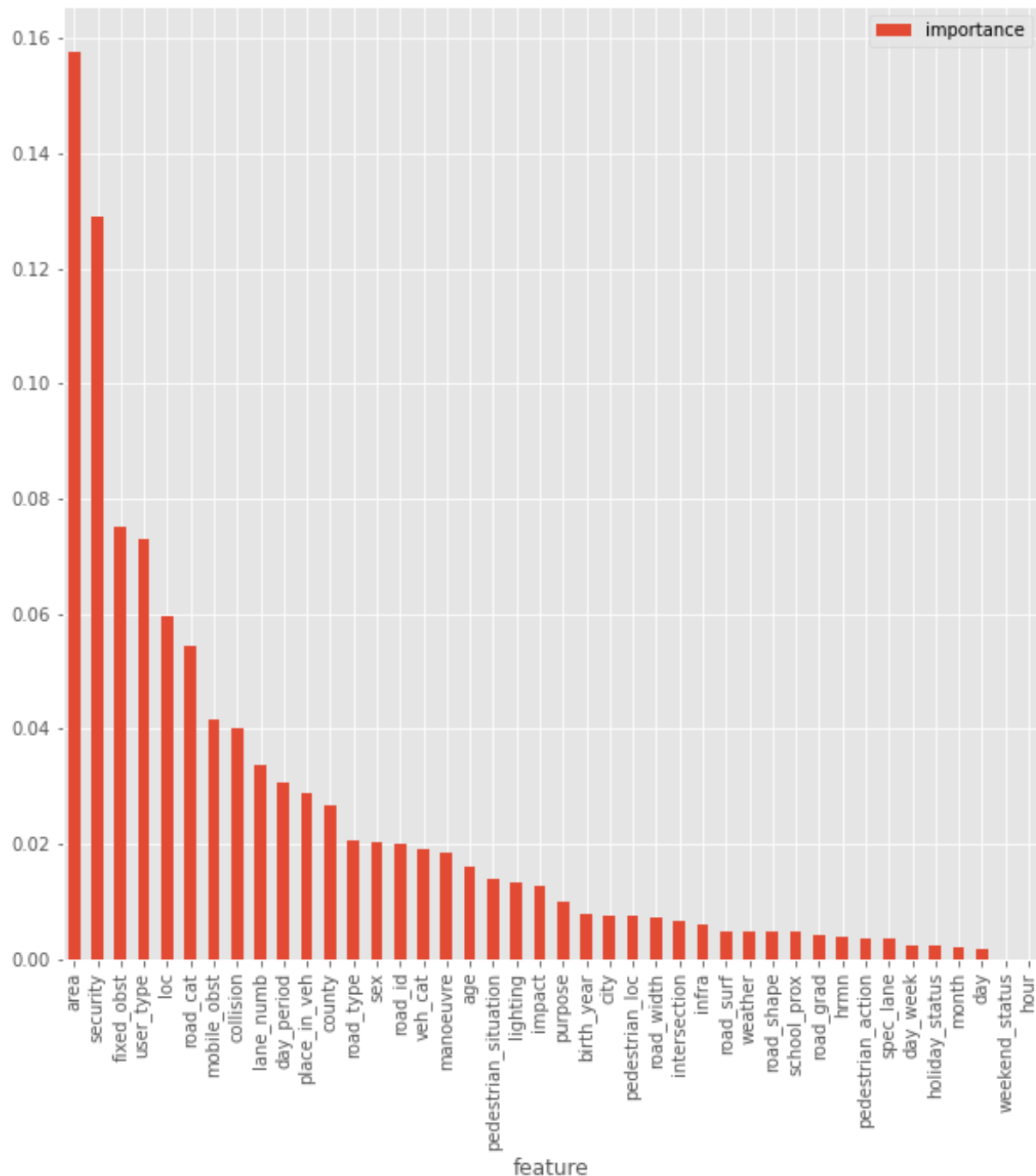
With the same number of trees (100) and all the features for the first modeling, the severity with 2 classes was performing the best:

Indicators	Initial Severity	3 Classes Severity	2 Classes Severity
Accuracy	0.6733	0.6952	0.8276
F1 Score	0.6619	0.6905	0.8163

The scores at that stage were actually slightly better than the Random Forest's with better processing time.

Indicators	2 Classes Severity	
	Random Forest	XGBoost
Accuracy	0.8256	0.8276
F1 Score	0.8109	0.8163
Wall Time	5 min 9	2 min 56

The review of the performances for this model revealed a different ranking than with the Random Forest:



If we consider the trade-offs we did for the Random Forrest:

- The age has significantly more weight than the initial birth year, so we can remove the birth year as well.
- The day period seems to have much more importance than the initial hrmn and the hour, so we will drop them.
- Of the different features linked to the day, the weekday seems to have more weight although all of them are at the end of the list. However, after several tests, the model that kept the day and holiday status performed slightly better, thus I decided to remove only the weekend status.

Once I had removed these features, we got the following results:

Indicators	2 Classes Severity 42 features	2 Classes Severity 38 features
Accuracy	0.8276	0.8278
F1 Score	0.8163	0.8166
Wall Time	2 min 49	2 min 27

The test on the number of features gave us the following scores and processing times:

Indicators	2 Classes Severity				
	5 features	10 features	15 features	20 features	25 features
Accuracy	0.7919	0.8038	0.8161	0.8241	0.827
F1 Score	0.7574	0.7776	0.801	0.8122	0.816
Wall Time	39s	57s	1 min 14	1 min 31	1 min 56

As for the Random Forest model, we can choose the model that fits the amount of information we have to predict the severity of the accident.

Considering the gain in performance from 5 to 15 features, we might recommend to start, if possible, with a model having at least 15 features.

Depending on the goals in terms of accuracy, we can use whether the model with 25 features or the model with all the features if we can have all the information.

As well as for the Random Forest Model, I also made other tests by tuning some of the hyperparameters and managed as well to get some incremental improvement of the scores, but remained limited by my computer or cloud environments to implement a proper fine tuning of the model hyperparameters.

Those slight improvements with the number of trees are as follow:

Indicators	2 Classes Severity		
	100 Trees	200 trees	300 trees
Accuracy	0.8278	0.831	0.8319
F1 Score	0.8166	0.8209	0.8223
Wall Time	2 min 27	4 min 52	6 min 58

Considering the processing time with 300 trees is even lower than the Random Forest with a lower parameter of trees, we can keep it for our model.

Last, I also tried this parameter with a lower number of features to see if we get a similar level of accuracy (as we noticed with 200 trees and 25 most important features).

Indicators	2 Classes Severity 38 features	2 Classes Severity 25 features
Accuracy	0.8319	0.8305
F1 Score	0.8223	0.82080
Wall Time	6 min 58	5 min 12

With these lower scores we got with the 25 features, I decided to keep the model with 38 features for the validation.

4. Results

For the evaluation, considering the trade-off performance / processing time, I set the following parameters:

- Features selection: all the features (38) for both models,
- 200 trees for the Random Forest and 300 for the XGBoost.

Using the test sets, I got similar results if not slightly better scores:

1. With the Random Forest Model:

Indicators	Validation Set	Test Set
Accuracy	0.8278	0.8282
F1 Score	0.814	0.8145
Wall Time	8 min 50	8 min 55

2. With the XGBoost Classifier:

Indicators	Validation Set	Test Set
Accuracy	0.8319	0.8315
F1 Score	0.8223	0.822
Wall Time	6 min 58	6 m 44

As we could expect, the XGBoost Classifier offers slightly better performance both in terms of score and processing time.

5. Discussion

The goal of this study was to build a model to predict the severity of an accident based on several features / criteria.

With the reduction of the number of classes for our severity classification from 4 to 2, we managed to get a model that performed fairly well at predicting this severity while keeping some room for improvement.

Although some of features appeared to have more importance than others for our models, the best performing model(s) were the one(s) including a significant number of features. As we already knew road accidents are generally the combination of several factors, we could realize that the importance of these factors was not always the same for the different algorithms.

Depending on the goal and use of this model, we know that we will not always have all the information available which might impact the accuracy of the prediction. For instance, an emergency service receiving the information of an accident to handle the potential victims will certainly not have the detail of the 38 features we selected after the preparation of the data. Fortunately, we saw that a more limited number of features (from 5 or 10) was already providing an accuracy or around 80%.

For anticipation purpose (government, local institutions willing to prevent such accidents, healthcare services trying to forecast the needs for medical services for certain periods of time and locations), we should be able to provide the model with more information and to reach a higher accuracy.

As mentioned above, we could have improved the performance of these models with a proper fine-tuning of the hyperparameters in a more appropriate technical environment. Another alternative would have been to limit the dataset to maybe the last 10 or even 5 years, which may have may represent more relevant data (because focusing on the most recent years) while allowing to perform this tuning without compromising significantly the prediction accuracy.

Among the other improvements of our model, we could also consider to implement some clustering techniques on some features that had an important number of unique values such as the location, days, time of the day...

Although the models selected provided some pretty good scores in terms of performance, we could also try other classification models not relying on Decision Tree (such as logistic regression, support vector machine, neural networks....)

Beyond the choices I made for the feature selection and model tuning, it could have also been interesting to adapt the severity classification to include the number of victims involved in one single accident. This would have allowed us to give more weight on the most critical accidents involving several people.

Last but not least, our model was not including some factors which are often considered as quite if not very important in terms of impact on road accidents, including:

- Human factors such as alcohol / drugs consumption, vehicle speed vs maximum speed limit, use of mobile...
- Vehicle characteristics such as car type, power engine, age of the vehicle...

6. Conclusion

Based on a dataset of road accidents in France from 2005 to 2017, we have built a model to predict the severity of a road accident. This model could be used by government or local institutions to prevent such accidents or for hospitals or healthcare services to anticipate the needs for medical services.

Using models based on Tree Decision models, we managed to reach 83.15% of accuracy while reducing the percentage of false positives / negatives to less than 18% (precision and recall scores being both quite close to the F1 score). The best of the 2 models we tested was the XGBoost Classifier.

These results leave some room for improvements:

- When it comes to the data:
 - First, by adding human factors which are often considered as very important in terms of impact on road accidents, or vehicle characteristics,

- Then, by including in the severity the weight of the number of victims per accident,
- Regarding the model:
 - By performing a proper fine-tuning of the hyperparameters, which might require to limit the number of years of our dataset,
 - By implementing some clustering techniques to handle some features with an important number of unique values,
 - And maybe by testing other types of algorithms to ensure they do not provide better results.

Appendix

Accident Data¶

- acc_ID: Accident identification number
- day: Day of the accident
- month: Month of the accident
- year: Year of the accident
- hrnm: Time of the accident in hour and minutes (hhmm)
- lighting : Lighting conditions
 1. Daylight
 2. Dusk or dawn
 3. Darkness without public lighting
 4. Darkness with public lighting not lit on
 5. Darkness with public lighting on
- county: French "department" code as defined by INSEE (National Institute of Statistics and Economic Studies)
- city: City code as defined by INSEE
- area:
 1. Rural area
 2. Urban area
- intersection:
 1. No intersection
 2. Intersection in X shape
 3. Intersection in T shape
 4. Intersection in Y shape
 5. Intersection with more than 4 roads
 6. Roundabout
 7. Square
 8. Railroad crossing
 9. Other intersection
- weather:
 1. Normal
 2. Light rain
 3. Heavy rain
 4. Snow - hail
 5. Fog - smoke
 6. Strong wind - storm
 7. Blinding sunny weather
 8. Cloudy weather
 9. Other
- collision:
 1. Two vehicles - frontal collision
 2. Two vehicles - collision from the rear
 3. Two vehicles - side collision
 4. Three vehicles and more - chain collision
 5. Three or more vehicles - multiple collisions
 6. Other collision
 7. Without collision
- adress: Postal address in City area

- gps: GPS code:
 - M = Mainland France
 - A = French West Indies (Martinique or Guadeloupe)
 - G = French Guiana
 - R = Réunion island
 - Y = Mayotte
- lat: Latitude in decimal degrees
- long: Longitude in decimal degrees

User Data

- acc_ID: Accident identification number.
- place_in_veh: place occupied by the user in the vehicle at the time of the accident
- user_type:
 1. Driver
 2. Passenger
 3. Pedestrian
 4. Pedestrian - rollerblade or scooter users
- severity: There are 3 categories of injuries + 1 for the unarmed users.
 1. Unarmed
 2. Killed
 3. Hospitalized wounded
 4. Light injury
- sex:
 1. Male
 2. Female
- purpose: Reason for traveling at the time of the accident:
 1. Home - work
 2. Home - school
 3. Shopping
 4. Professional use
 5. Leisure
 9. Other
- security: there are 2 characters:
 - the first character details the existence of a safety equipment
 1. Safety Belt
 2. Helmet
 3. Children car seat / booster
 4. Reflective gear
 9. Other
 - the second one details the use of the safety equipment
 1. Yes
 2. No
 3. Not determinable
- pedestrian_loc: Location of the pedestrian at the time of the accident
 - On roadway:
 1. More than 50 m from the pedestrian crossing
 2. Less than 50 m from the pedestrian crossing
 - On pedestrian crossing:
 3. Without crossing lights / signals
 4. With crossing lights / signals

- Other:
 5. On the sidewalk
 6. On the road shoulder
 7. On the emergency lane
 8. On a side road
- pedestrian_action: Action of the pedestrian at the time of the accident:
 0. Not specified or not applicable
 1. Moving towards the vehicle
 2. Moving on the opposite direction from the vehicle
 3. Crossing the road
 4. Hidden from the vehicle
 5. Playing - running
 6. With a domestic animal
 9. Other
- pedestrian_situation: whether the injured pedestrian was alone or not
 1. Alone
 2. With someone
 3. In a group
- birth_year: Year of birth of the user
- veh_ID: Vehicle identification number (including pedestrians who are linked to the vehicles that hit them)

Vehicle Data

- acc_ID: Accident identification number.
- flow_dir: Flow direction :
 1. Increasing kilometre marker or post code
 2. Decreasing kilometre marker or post code
- veh_cat: Category of the vehicle.
 01. Bicycle
 02. Motorbike / moped (<50cm³)
 03. Small Car (<125cm³)
 04. Registered scooter - Not used since 2006
 05. Motorcycle - Not used since 2006
 06. Side-car - Not used since 2006
 07. Car - Light Vehicle only
 08. Car - Light Vehicle with caravan / mobile home - Not used used anymore
 09. Car - Light Vehicle with trailer - Not used used anymore
 10. Utility vehicle with or without trailer and with a 1,5T <= Total Weight <= 3,5T
(formerly Utility Vehicle only with 1,5T <= Total Weight <= 3,5T)
 11. Utility vehicle with caravan / mobile home - Not used since 2006
 12. Utility vehicle with trailer - Not used since 2006
 13. Heavy truck alone with 3,5T <= Total Weight <= 7,5T
 14. Heavy truck alone with 7,5T < Total Weight
 16. Road Tractor alone
 17. Road Tractor with semi-trailer
 18. Public transport - not used since 2006
 19. Tramway - not used since 2006
 20. Heavy plant machinery

- 21. Tractor
- 30. Scooter (<50cm³)
- 31. Motorbike (50cm³ < ≤125cm³)
- 32. Scooter (50cm³< <125cm³)
- 33. Motorbike (<125cm³)
- 34. Scooter (<125cm³)
- 35. Four wheeler bike / ATV light (50cm³≤)
- 36. Four wheeler bike / ATV (<50cm³)
- 37. Bus
- 38. Coach
- 39. Train
- 40. Tramway
- 99. Other vehicle
- user_num: number of user(s) in the vehicle
- fixed_obst: Fixed obstacle hit by the vehicle.
 - 1. Other parked vehicle
 - 2. Tree
 - 3. Metal safety barrier
 - 4. Concrete safety barrier
 - 5. Other safety barrier
 - 6. Building, wall, bridge pillar
 - 7. Road sign or traffic signal
 - 8. Pole
 - 9. Street furniture
 - 10. Railing / parapet
 - 11. Shelter / border marker / bollard
 - 12. Sidewalk edge
 - 13. Ditch / embankment / rock wall
 - 14. Other fixed obstacle on the roadway
 - 15. Other fixed obstacle on the sidewalk or road shoulder
 - 16. End of roadway without obstacle
- mobile_obst: Mobile obstacle.
 - 1. Pedestrian
 - 2. Vehicle
 - 3. Railcar
 - 4. Domestic animal
 - 5. Wild animal
 - 9. Other
- impact: first point of impact.
 - 1. Front
 - 2. Front right
 - 3. Front left
 - 4. Back
 - 5. Back right
 - 6. Back left
 - 7. Right side
 - 8. Left side
 - 9. Multiple impacts (rollover)
- manoeuvre:
 - Main manoeuvre before the accident:
 - 1. No change of direction

- 2. Same flow direction, same lane
- 3. Between 2 lanes
- 4. Reversing
- 5. Going the wrong way
- 6. Going through the median strip
- 7. In the bus lane, right way
- 8. In the bus lane, wrong way
- 9. Entering the road
- 10. U-turn
- Changing Lane:
 - 11. To left
 - 12. To right
- Drifting:
 - 13. Left
 - 14. Right
- Turning:
 - 15. Left
 - 16. Right
- Overtaking:
 - 17. Left
 - 18. Right
- Other:
 - 19. Crossing the roadway
 - 20. Parking
 - 21. Dodging
 - 22. Opening the door of the vehicle
 - 23. Stopped (not parked)
 - 24. Parked (with user(s) in the vehicle)
- veh_ID: Vehicle identification number

Location Data

- acc_ID: Accident identification number.
- road_cat : Category of road.
 - 1. Highway / motorway
 - 2. Main Road / A road
 - 3. County road / B road
 - 4. Local road / C road
 - 5. Private road
 - 6. Parking lot open to public traffic
 - 9. Other
- road_id: Road identification number
- road_det1: Additional information on the road (example: 2 bis, 3 ter etc.)
- road_det2: Additional information on the road (alpha-numerical ID)
- road_type: Traffic type.
 - 1 - One way
 - 2 - 2-Lane single carriageway
 - 3 - Dual carriageway
 - 4 - Multi-lane single carriageway

- lane_num: total number of traffic lanes.
- road_marker: Road marker ID.
- rm_dist: Road marker distance.
- spec_lane: Special lane.
 1. Stand alone Bike lane
 2. Bike lane (on roadway)
 3. Other special lane
- road_grad:
 1. Flat
 2. Slope
 3. Top of the slope / hilltop
 4. End of slope
- road_shape:
 1. Straight
 2. Left curve
 3. Right curve
 4. S curve
- med_strip: width of the median strip (if any).
- road_width: width of the roadway only (excluding emergency lanes, parking slots...)
- road_surf: surface of the road.
 1. Normal
 2. Wet
 3. Water puddle
 4. Flooded
 5. Snow
 6. Muddy
 7. Icy
 8. Oily
 9. Other
- infra: infrastucture.
 1. Tunnel
 2. Bridge
 3. Interchange ramp
 4. Railway
 5. Special crossroad
 6. Pedestrian zone
 7. Road toll zone
- loc: location of the accident.
 1. On the roadway
 2. On the emergency lane
 3. On the road shoulder
 4. On the sidewalk
 5. On the bike lane
- school_prox: proximity with a school.