SEP23\_ALLIANZ\_DATA\_SCIENTIST

**Road accidents in France**

Final Report

horizontal line

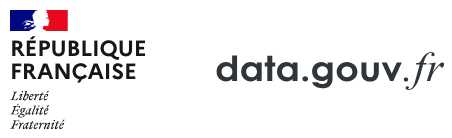
# 

|  |  |
| --- | --- |
| Submitted by | Project Mentor |
| Jennifer Pousada Alvarez  Alessandro Perani  Ahed Abdelky  Falk Kegler | Antoine Tardivon |

# Problem Definition

The objective of this project is to try to predict the severity of road accidents in France. Predictions will be based on historical data.

# Data Collection

The data is provided on the platform [data.gouv.fr](https://www.data.gouv.fr/en/), the official open data platform of the French government. It is managed by Etalab, a mission under the authority of the Prime Minister. The platform aims to make public data freely available and reusable by anyone, including citizens, businesses and researchers.

From this site we used the dataset "Bases de données annuelles des accidents corporels de la circulation routière - Années de 2005 à 2022"



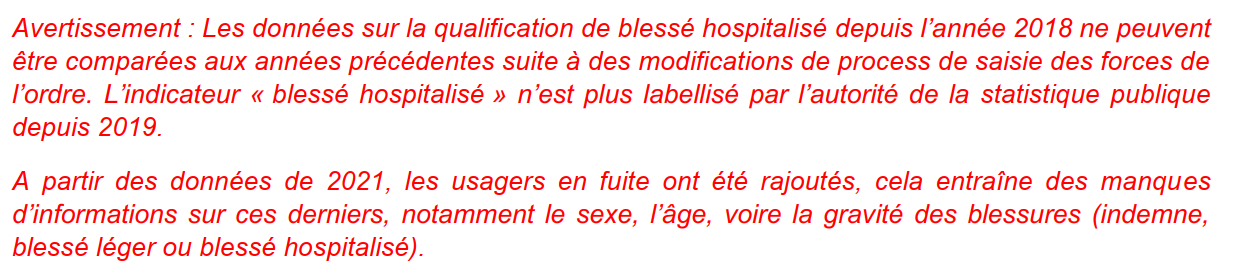
# Data Preprocessing

The yearly released datasets contain information in csv-files for each accident on

- Characteristics: describing the general circumstances of the accident  
- Locations: providing information about the main location of the accident  
- Vehicles: delivering descriptions on the vehicles involved in the accident  
- Users: informing about people involved in the accident and the severity of their injuries

We first had a look at all the data, scanned for missing values, duplicates as well as changes in the data-structure.

We have noticed that there has been a significant change in the data structure from 2019, particularly in terms of user data. This is confirmed in the document “description-des-bases-de-donnees-annuelles-2022.pdf” where it states:



Google translated: "Warning: Data on the qualification of hospitalized injured since 2018 cannot be compared to previous years following changes in the data entry process by law enforcement. The "hospitalized injured" indicator has not been labelled by the public statistics authority since 2019.

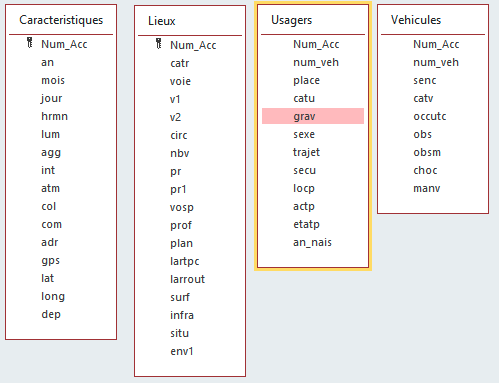
From the 2021 data, users who fled were added, which results in a lack of information on the latter, in particular gender, age, and even the severity of injuries (uninjured, slightly injured or hospitalized injured)."

# We therefore decided to limit ourselves to accidents that occurred between 2005 and 2018.

# Exploratory Data Analysis

## Keys

Since there are 4 CSV files per year, we needed to find an attribute to merge the files. The attribute we found is Num\_Acc, the ID number of an accident. For each user and vehicle involved in the accident, there are 1:n relationships to features and locations. Therefore, the data in the user and vehicle tables needed to be aggregated to the accident level before merging with the feature and location tables.



## Target

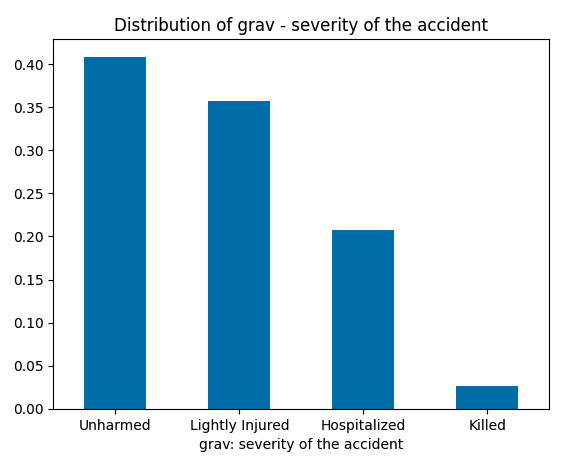
To achieve the goal of the project, we next had to look for attributes that provide information about the severity of the accidents. We found an attribute in the users table called grav, for French gravité, in which the severity of the accident is classified into one of the following categories for each user involved:

unharmed (1),   
killed (2),   
injured or hospitalized (3) and   
lightly injured (4).   
  
In terms of a machine learning project, the *grav*-attribute is related to the quality to predict – the target.

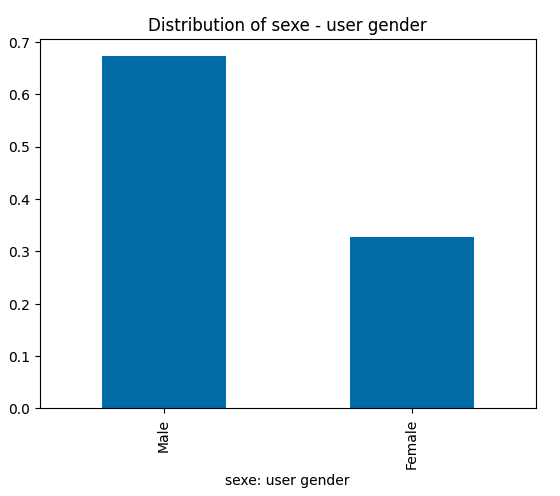
## Distributions

### Users

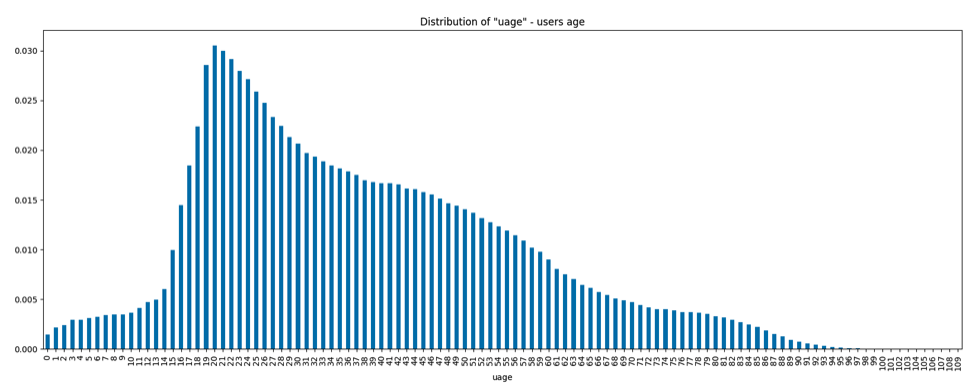
Most individuals involved in road accidents are either unharmed or lightly injured, indicating that many accidents result in minor or no injuries. However, a notable portion is hospitalized, and a small yet significant percentage are killed.



Males are involved in road accidents at a significantly higher rate than females, representing a majority of the cases. This disparity may reflect differences in driving habits, exposure, or risk-taking behaviors between genders.

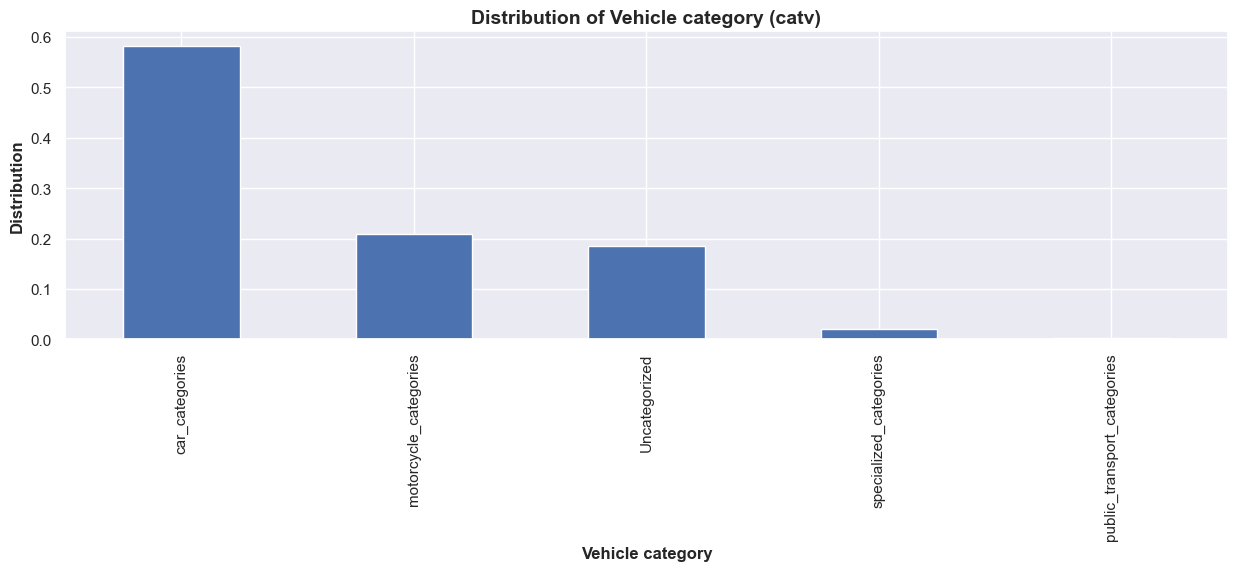


The analysis of the age distribution of car accident occupants shows that people between 19 and 29 years of age are particularly at risk. The age used here is a recoded attribute from the available year of birth an\_nais in the users table.

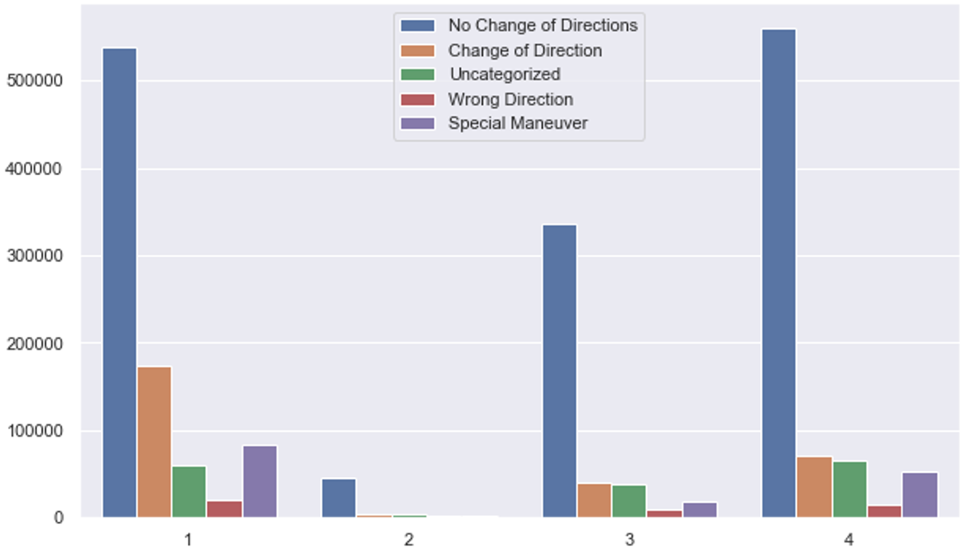


### Vehicles

Based on the provided categories for the variable *catv*, we created groups. This grouping is based on the similarity of vehicle types and their usage. The results depict the distribution of transportation modes: cars dominate at approximately 71%, followed by motorcycles at 25%, with specialized modes making up a smaller proportion at 2%. Public transport represents a minimal fraction, accounting for only about 0.3% of the total.

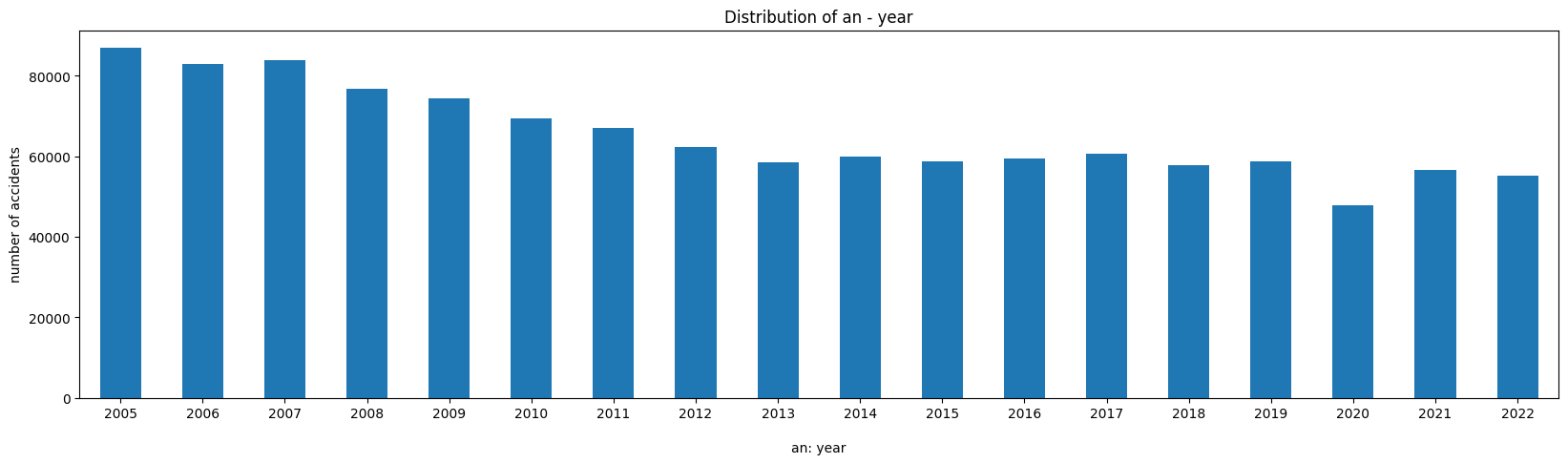


The characteristics of the variable main maneuver have been grouped into 5 categories. The highest level of severity accounts for most of the accidents without change of direction.

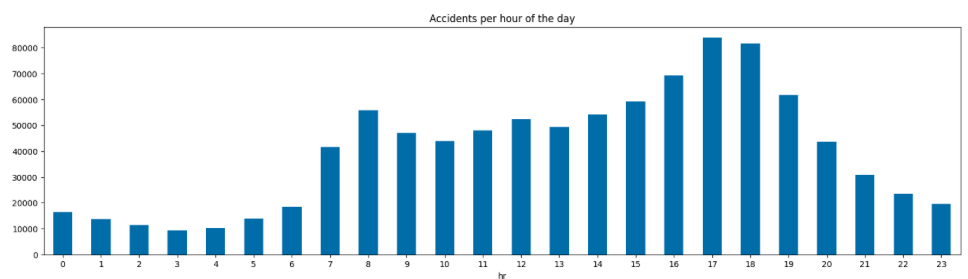


### Characteristics

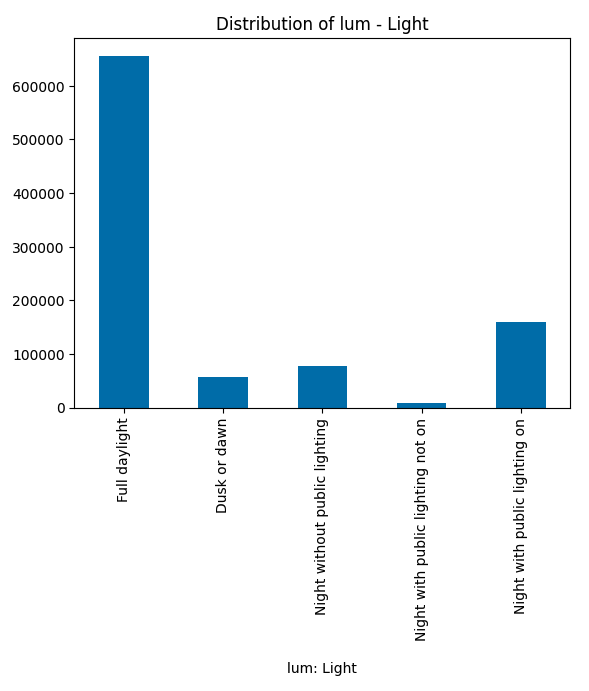
Starting 2005 we observe a decrease in the number of accidents per year until the year 2012. From there on the number of accidents per year stagnated at a similar level. We also see the effect of lockdowns during the COVID pandemic in 2020 with a considerably lower number of accidents that can be attributed to the reduction of car usage during that period.



On a daily level the number of accidents has local maxima at the rush-hours in the mornings around 8 o'clock and in the evening around 17 o'clock.



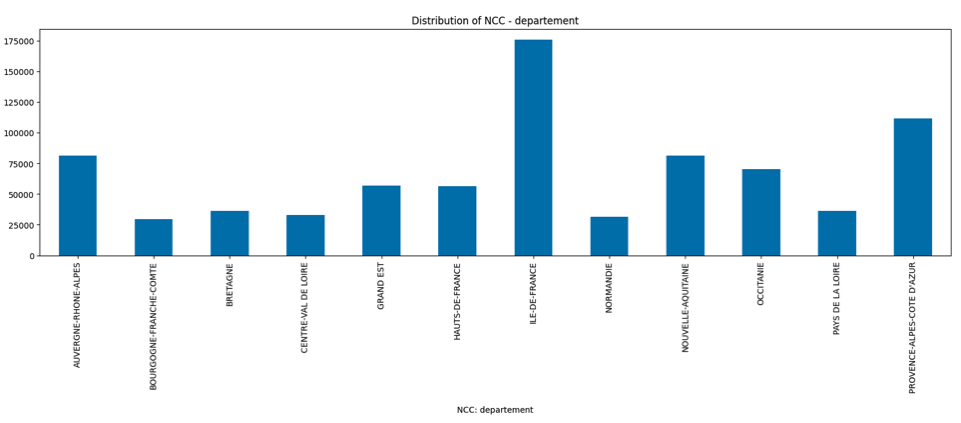
Most accidents happen in full daylight and in urban areas.

Ein Bild, das Text, Screenshot, Diagramm, Rechteck enthält.

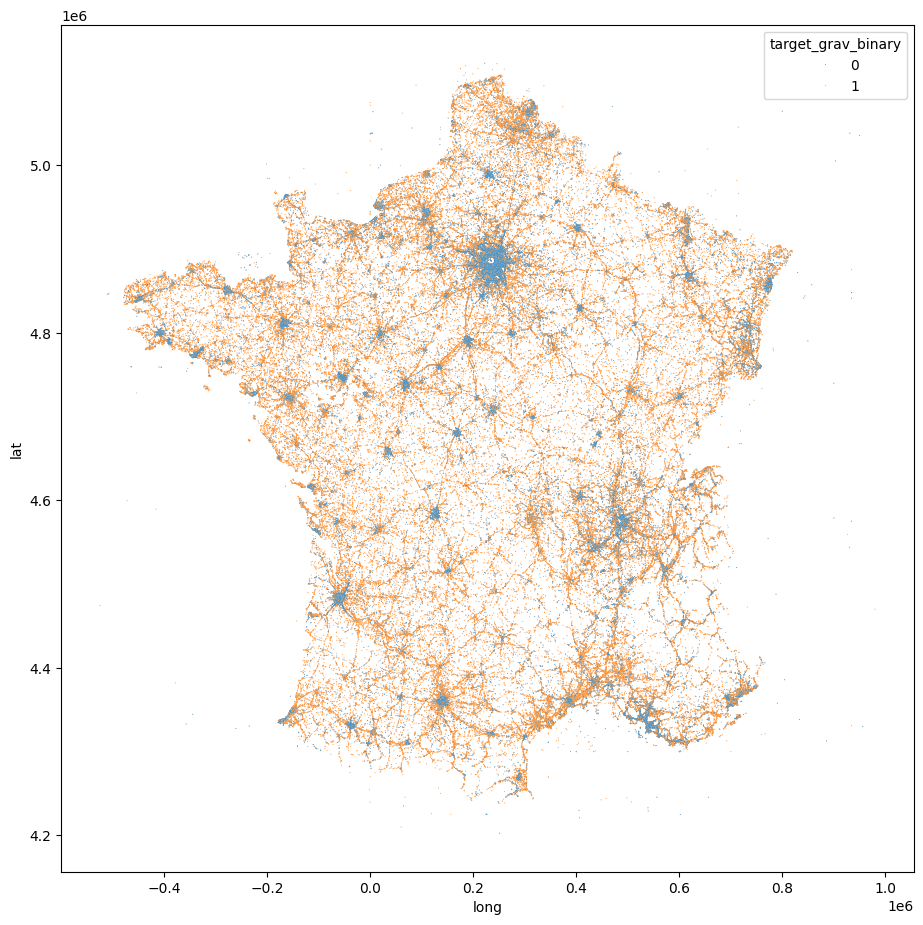
Automatisch generierte Beschreibung

### Places

Looking at the following graph, it can be seen that the highest number of accidents relates to the region Ile-de-France including Paris. This has been expected as population density in this region is the highest in France. However, taking the accident severity into consideration, it can be observed that it is quite low for this region.



Latitude and longitude coordinates enable us to observe how accidents occurred in in-built areas are more often with lower severity. High severity claims often occur outside the cities. The map confirms the results coming from the analysis per region. Even though most accidents are concentrated in the Paris area, most of them are low-severity accidents, marked with blue colour.



## Feature engineering

### Dummy encoding

As nearly all features are categorical, we dummy encoded them after cleaning them up. This procedure is demonstrated here for the gender column sex in the user table:

# Number of Users per gender

col='sexe'

col\_short='sex'

actual\_col\_map = {

'sex\_1':'usr\_Male'

,'sex\_2':'usr\_Female'}

col\_sex = df[['Num\_Acc', col]].join(pd.get\_dummies(df[col], prefix=col\_short, dtype="int64")).drop(col, axis=1).groupby('Num\_Acc').sum()

col\_sex=col\_sex.rename(columns=actual\_col\_map)

col\_sex

First the values are mapped to a unique column name with a prefix (sex\_) resulting in a 1 or 0 per user. After that, the sum per dummy column and accident is calculated, resulting in feature columns with the prefix usr\_ like this:

| **usr\_Male** | **usr\_Female** |
| --- | --- |
| **Num\_Acc** |  |  |
| **200500000001** | 4 | 2 |
| **200500000002** | 2 | 0 |
| **200500000003** | 2 | 0 |
| **200500000004** | 2 | 2 |
| **200500000005** | 2 | 0 |
| **...** | ... | ... |
| **201800057779** | 2 | 0 |
| **201800057780** | 1 | 0 |
| **201800057781** | 2 | 0 |
| **201800057782** | 1 | 1 |
| **201800057783** | 1 | 1 |

958469 rows × 2 columns

So in the accident with ID **200500000001** 4 men and 2 women have been involved.

The prefixes of the feature names for our final DataFrame have been chosen as follows:

• crc for characteristic-features  
• plc for place related features  
• usr for user related features  
• veh for vehicle related features

### Target(s)

As for the target we decided to create 2 variants: a ternary target and a binary target.

#### Ternary Target

# There are no accidents with all persons unharmed, hence the 3 bins instead of 4

dfr['target\_grav\_ternary'] = np.where((dfr['grav\_4\_killed'] > 0), 2, 0)

dfr['target\_grav\_ternary'] = np.where((dfr['grav\_3\_hospitalized'] > 0) & (dfr['grav\_4\_killed'] == 0), 1, dfr['target\_grav\_ternary'])

dfr.target\_grav\_ternary.value\_counts()

This leads to an unbalanced distribution of the ternary target:

Out [85]:

target\_grav\_ternary

0 418,694

1 332,272

2 48,803

Name: count, dtype: int64

#### Binary target

# This binary target could serve as a hint, whether an ambulance is needed or not

dfr['target\_grav\_binary'] = np.where((dfr['grav\_3\_hospitalized'] > 0) | (dfr['grav\_4\_killed'] > 0), 1, 0)

dfr.target\_grav\_binary.value\_counts()

This binary target is better balanced and could be practical to use for planning, how many ambulances are needed.

Out [86]:

target\_grav\_binary

0 418,694

1 381,075

Name: count, dtype: int64

### Cyclic variables

Cyclic variables such as the hour of the day, the day of the month, the month of the year, and even latitude and longitude should be converted using a sine-cosine approach to make the machine learning model aware that, for example, Monday is closer to Sunday than to Wednesday.

The time related features “minute of the day” and “month” in the characteristics-Tables for example have been replaced using a sin/cos decoding to have nearby times and dates better represented for the models to learn from them.

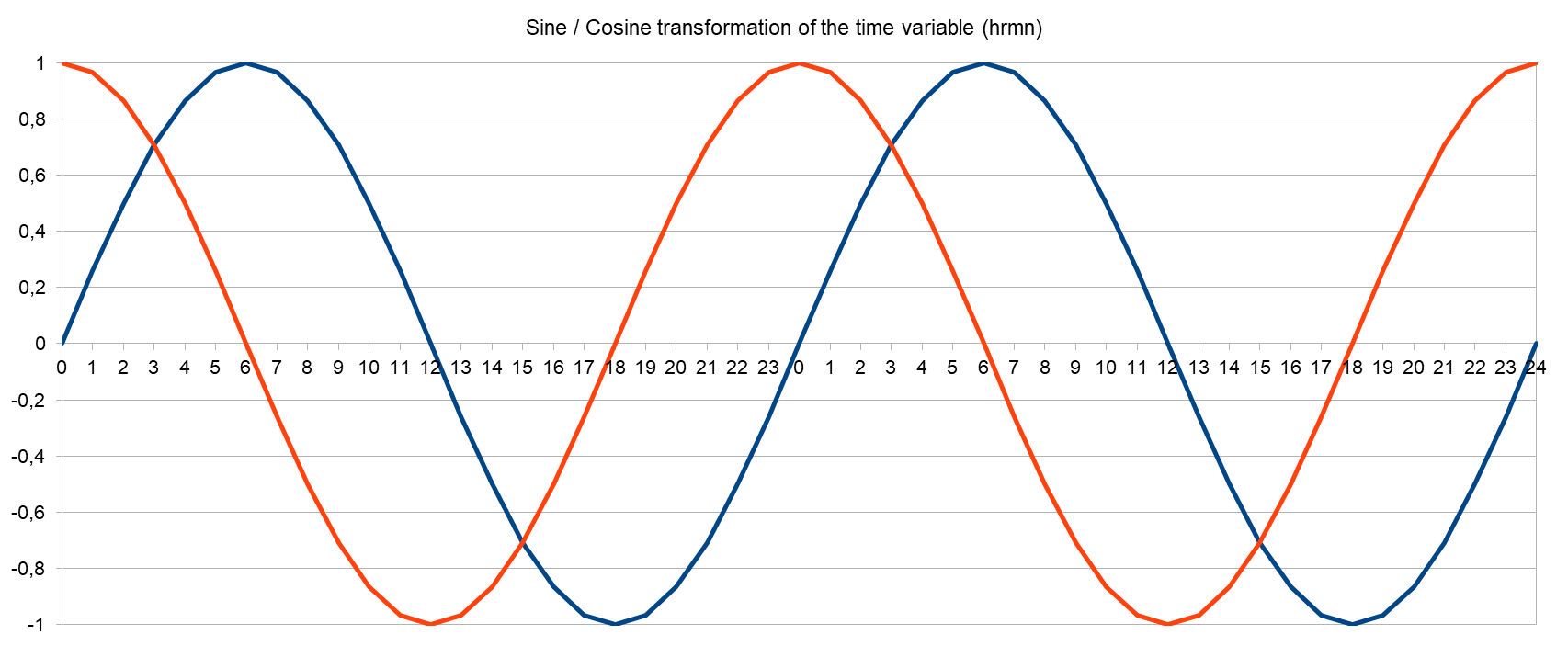
The hrmn (hour minute) variable is transformed as follows:

n = 1440

factor = 2 \* np.pi

df["sin\_hrmn"] = np.sin(df['hrmn']\*factor / n)

df["cos\_hrmn"] = np.cos(df['hrmn']\*factor / n)



The month variable is transformed like so:

n = 12

factor = 2 \* np.pi

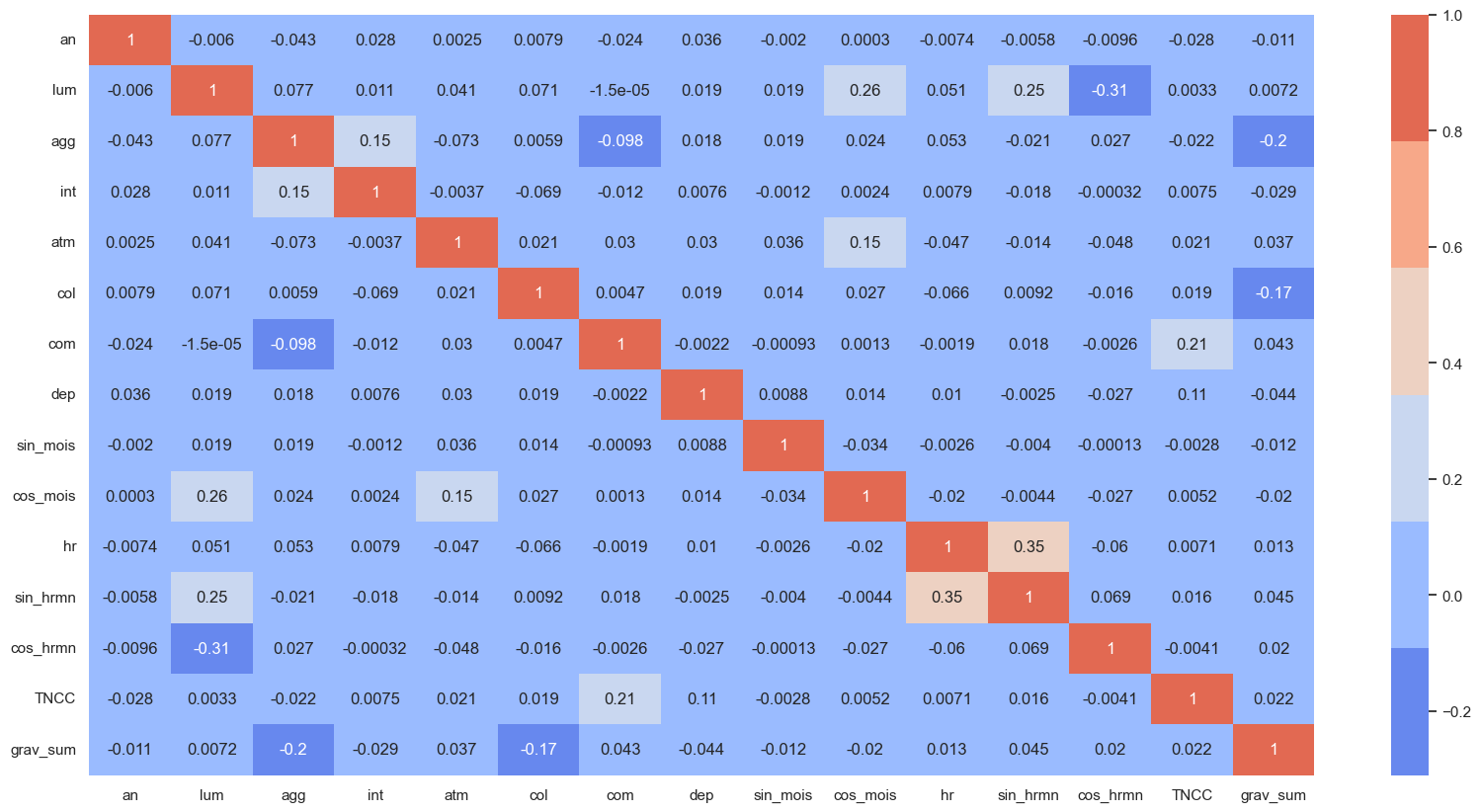
df["sin\_mois"] = np.sin(df['mois']\*factor / n)

df["cos\_mois"] = np.cos(df['mois']\*factor / n)

### Statistical Analysis and Cleaning

After dummy encoding and aggregating on an accident basis, we did statistical tests regarding the correlation of the features because some machine learning models suffer from correlated features, because they unnecessarily lead to more complex models.

The correlation in a DataFrame in Python can be shown using the corr-function in conjunction with the Seaborn heatmap-plot, resulting, for the user table in a plot like follows:



The values are showing the Pearson standard correlation coefficient, measuring the strength and direction of the linear relationship between two variables.

# Model Selection

## Classification of the problem

The aim of our study is to design a classification model for predicting the severity of road accidents in France. Predicting the severity of road accidents in France is a task related to predictive analytics and risk assessment. This type of task involves analysing historical data and other relevant factors to predict future outcomes or the severity of incidents.

We evaluated the performance of each model by calculating performance metrics such as accuracy (Accuracy) and F1-score (F1-score) on the training set and test set.

Accuracy was used to measure the overall precision of the model. We used Accuracy as it is straightforward to understand and interpret. It measures the proportion of correctly predicted instances (both true positives and true negatives) out of the total instances. If the classes (severity levels) are balanced as it is the case in our data, accuracy can give a good overall measure of the model's performance.

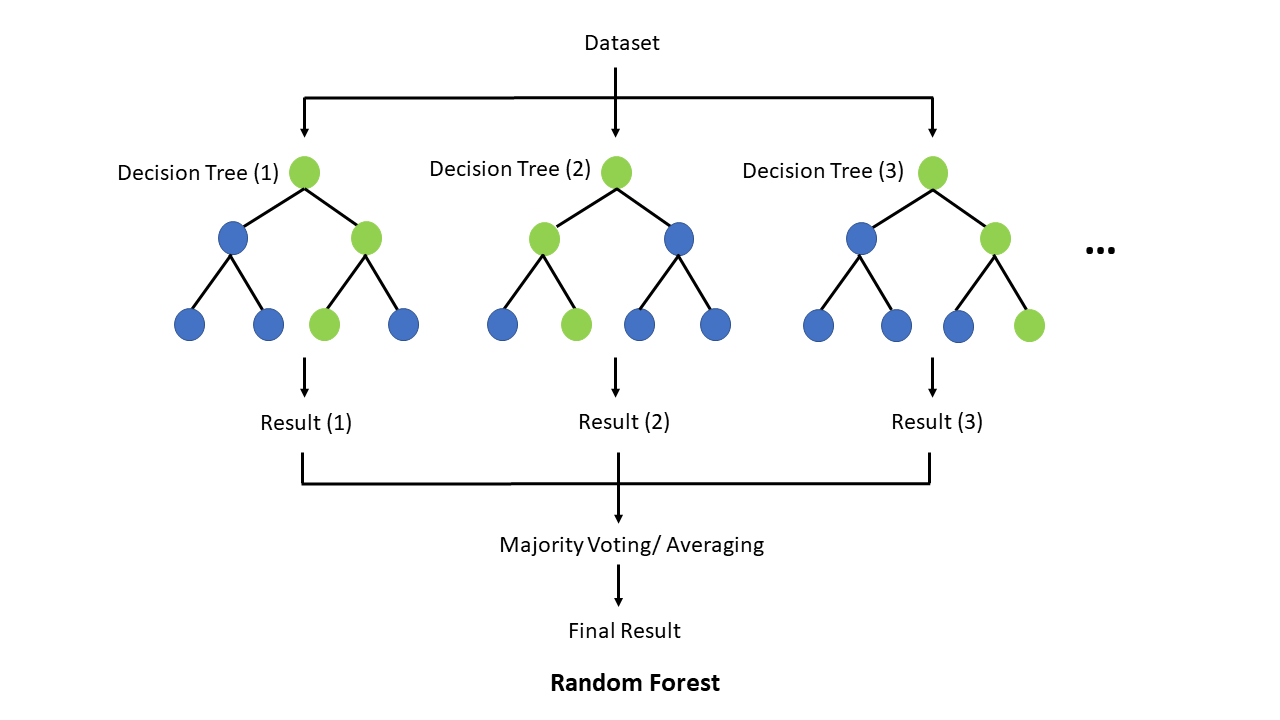
Furthermore, the F1-score was taken into account. This metric is as well useful for binary classification problems, as it takes into account both precision and recall to calculate an overall score.

For the model’s performance, we also considered the Area under the ROC curve (Roc-AuC).

We tried 5 models to predict the severity of the accidents.

* Random Forest Classifier for a ternary target
* Random Forest Classifier for a binary target
* KNN Classifier (k = 3) for a binary target
* XGBoost with GridSearchCV for a binary target
* Dense Neural Network for a binary target

## RANDOM FOREST



Our first models were RandomForest Classifiers. We modelled with a ternary and a binary target for the severity of the accident. We also examined the feature importance and visualized it with the help of the SHAP-package for the binary-target model on an instance for target 0 and target 1 respectively.

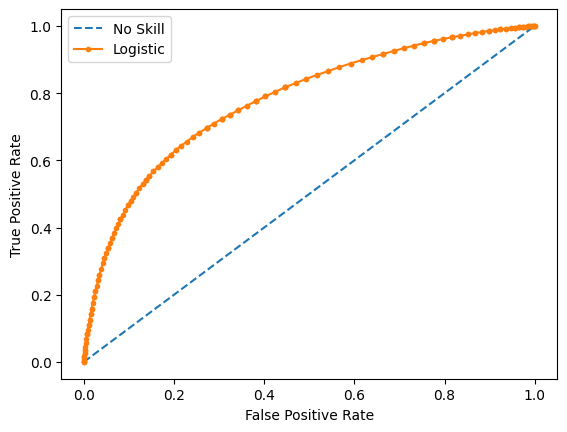
### Random Forest Classifier with ternary target

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RandomForest | precision | recall | f1-score | support |
| 0 | 0.69 | 0.82 | 0.75 | 83.718 |
| 1 | 0.62 | 0.56 | 0.59 | 66.542 |
| 2 | 0.54 | 0.05 | 0.09 | 9.694 |
| accuracy |  |  | **0.66** | 159.954 |
| macro avg | 0.62 | 0.48 | 0.48 | 159.954 |
| weighted avg | 0.65 | 0.66 | 0.64 | 159.954 |

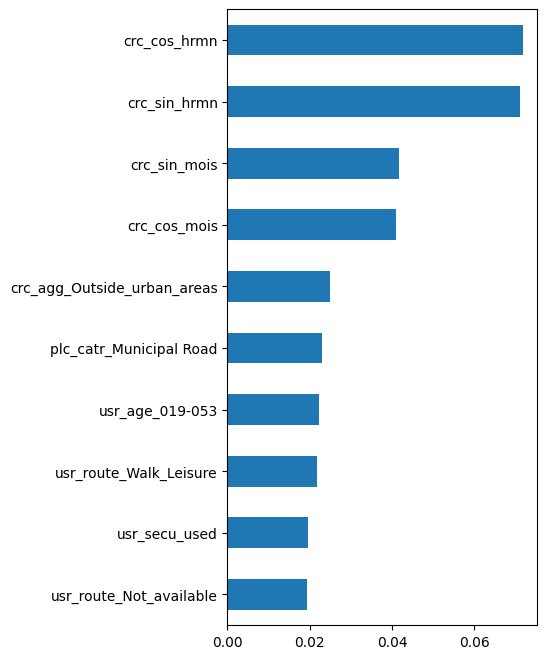
### Random Forest Classifier with binary target

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RandomForest | precision | recall | f1-score | support |
| 0 | 0.70 | 0.80 | 0.75 | 83.718 |
| 1 | 0.74 | 0.63 | 0.68 | 76.236 |
| accuracy |  |  | **0.72** | 159.954 |
| macro avg | 0.72 | 0.71 | 0.71 | 159.954 |
| weighted avg | 0.72 | 0.72 | 0.71 | 159.954 |

### ROC AUC=0.780



### Feature importance from Random Forest Classifier (Top 10):



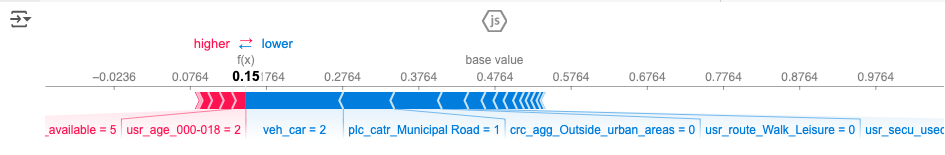
The prefixes of the feature names are:

* crc for characteristic-features
* plc for place related features
* usr for user related features
* veh for vehicle related features

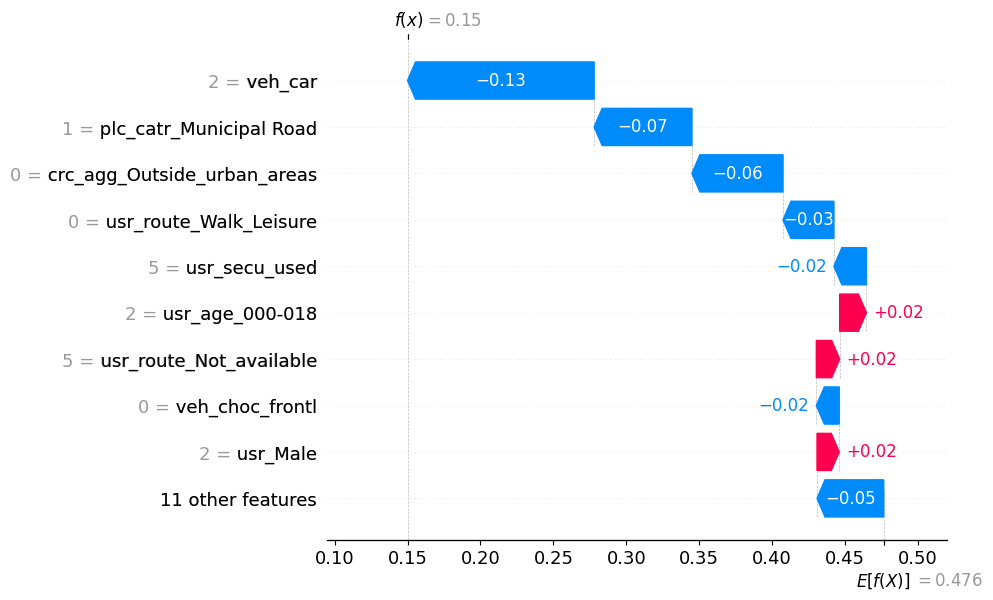
### SHAP

For the Random Forest Classifier we used the SHAPly values to visualize the feature importance and to show how and to what extent each feature contributed to the final prediction result. 2 instances were chosen, one with prediction result 0 and another with prediction result 1.

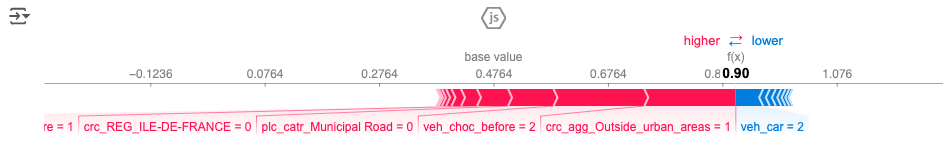
Shap-explainer (force) for instance with prediction result 0



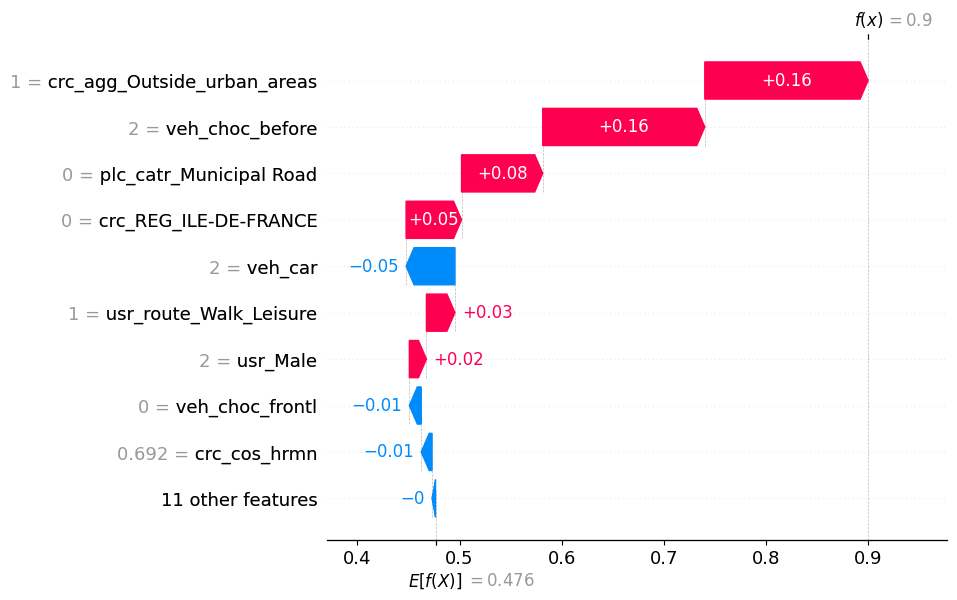
Shap-explainer (waterfall) for instance with prediction result 0:



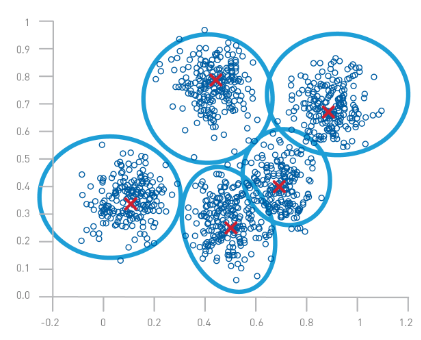
Shap-explainer (force) for instance with prediction result 1:



Shap-explainer (waterfall) for instance with prediction result 1:



## KMeans Cluster



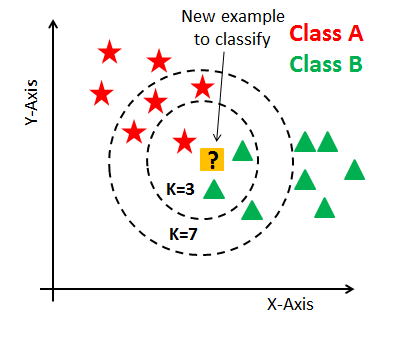
We also tried a KMeans Clustering model to see, if there is a natural cluster-mean for the binary target representing the severity of the accident. It turned out, that the correlation of the clusters found to the target was 0.25, so there is no indication of a natural severity-based clustering in the dataset.

### KMeans Cluster (n\_clusters = 2)

np.abs(dfc.corr()['cluster'][:]).sort\_values(ascending=False)[:12]

|  |  |
| --- | --- |
| **correlations** | **cluster** |
| cluster | 1.00 |
| crc\_agg\_Outside\_urban\_areas | 0.91 |
| plc\_catr\_Municipal Road | 0.61 |
| plc\_catr\_Highway | 0.40 |
| crc\_lum\_Night\_without\_public\_lighting | 0.34 |
| crc\_int\_Outside\_intersection | 0.29 |
| plc\_plan\_Curvature | 0.29 |
| veh\_obsm\_pedestrian | 0.27 |
| veh\_obsm\_unknown | 0.25 |
| plc\_situ\_Roadside\_verge | 0.25 |
| **target** | **0.25** |
| crc\_lum\_Night\_with\_public\_lighting\_on | 0.22 |

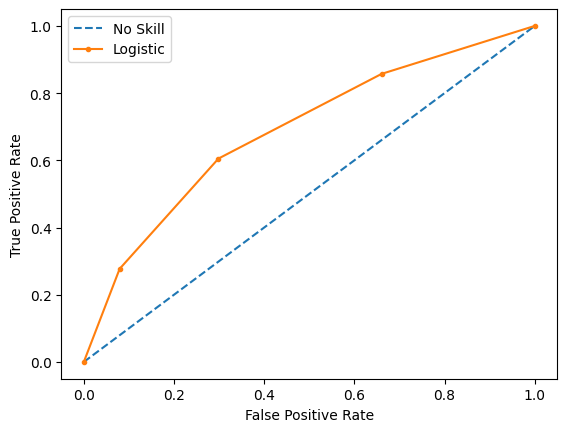
## K-Nearest Neighbour (k=3)



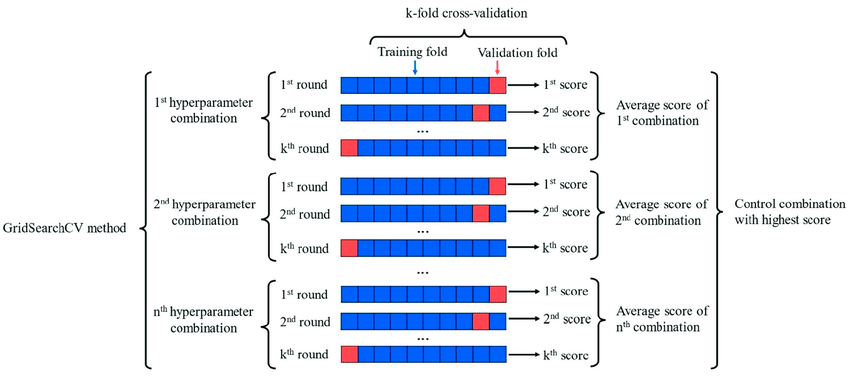
The k-nearest-neighbours algorithm classifies an unclassified datapoint like the majority of it's k-nearest neighbours is classified. As the algorithm is computationally expensive, we had to limit ourselves to the 3 nearest neighbours and a subsample of 10.000 records, resulting an accuracy of 0.66.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| kNN (k=3) | precision | recall | f1-score | support |
| 0 | 0.66 | 0.70 | 0.68 | 5.250 |
| 1 | 0.65 | 0.61 | 0.63 | 4.750 |
| accuracy |  |  | **0.66** | 10.000 |
| macro avg | 0.65 | 0.65 | 0.65 | 10.000 |
| weighted avg | 0.66 | 0.66 | 0.65 | 10.000 |

resulting in an AuC of 0.668.



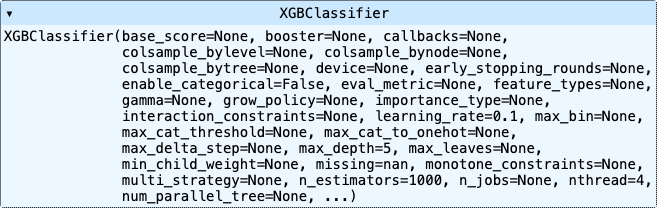
## XGBoost with GridSearch CrossValidation



The XGBoost-Algorithm is famous for its high accuracy, especially on Kaggle. We tried to find the best model with the help of a grid search cross-validation. Indeed, the model with the best estimators achieved the highest accuracy of all our models (73%) and the highest value for the ROC-AuC (0.793).

### XGBoost Classifier with Hyperparameter Tuning (grid search)

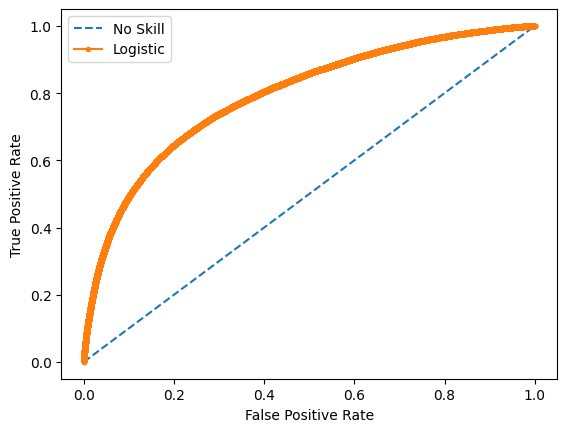
grid\_search.best\_estimator\_:



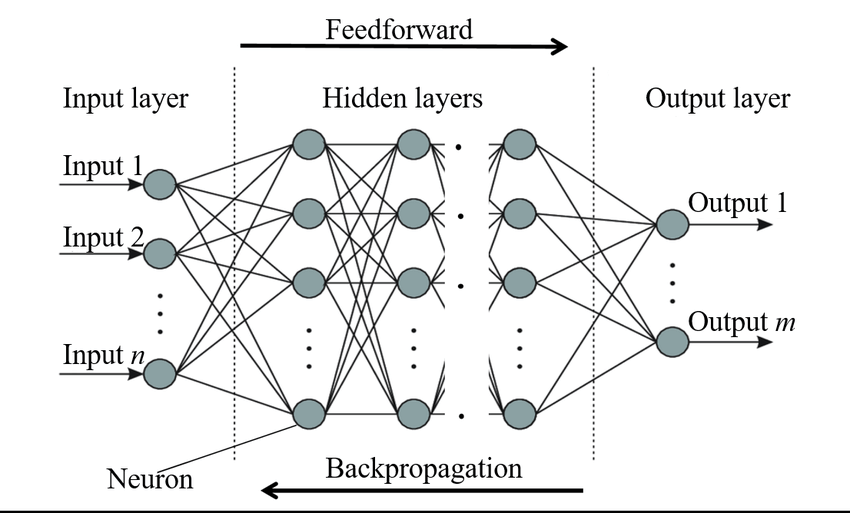
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| XGBoost | precision | recall | f1-score | support |
| 0 | 0.71 | 0.80 | 0.75 | 83.718 |
| 1 | 0.74 | 0.65 | 0.69 | 76.236 |
| accuracy |  |  | **0.73** | 159.954 |
| macro avg | 0.73 | 0.72 | 0.72 | 159.954 |
| weighted avg | 0.73 | 0.73 | 0.72 | 159.954 |

The best parameters across ALL searched params: grid\_search.best\_params\_

{'learning\_rate': 0.1, 'max\_depth': 5, 'n\_estimators': 1000}  
  
ROC AUC=0.793



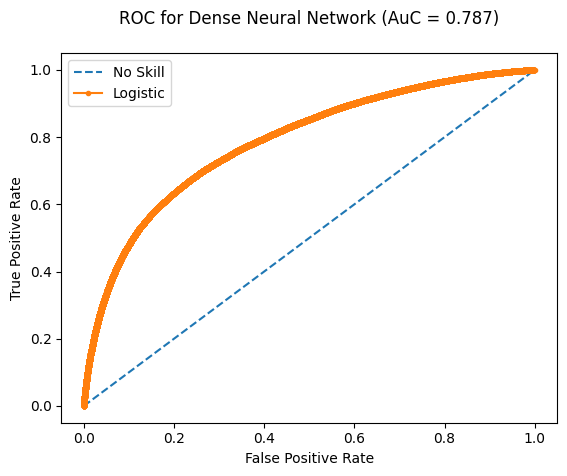
## Deep Learning



To practice what we have learned, we decided to try a deep learning model as well. With two hidden layers in a dense neural network we were able to achieve an accuracy of 71%.

### Dense Neural Network with 2 hidden layers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DNN | precision | recall | f1-score | support |
| 0 | 0.70 | 0.80 | 0.75 | 83.718 |
| 1 | 0.74 | 0.62 | 0.68 | 76.236 |
| accuracy |  |  | **0.71** | 159.954 |
| macro avg | 0.72 | 0.71 | 0.71 | 159.954 |
| weighted avg | 0.72 | 0.71 | 0.71 | 159.954 |



As we received the best accuracy with the XGBoost estimator, we will use the following **model** for all future tasks:

## Best MODEL

**from xgboost import XGBClassifier**

**model = XGBClassifier(**

**objective = 'binary:logistic',**

**learning\_rate = 0.1,**

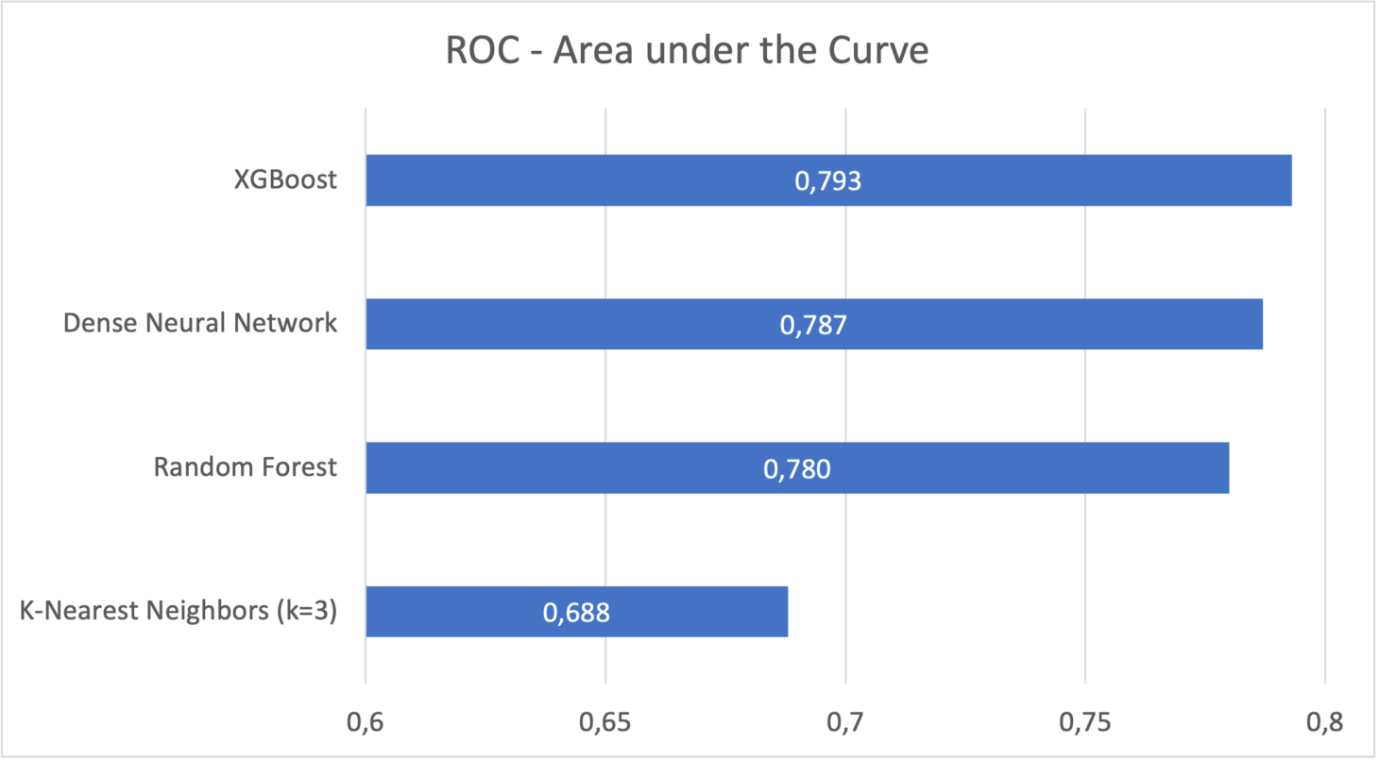
**n\_estimators = 1000,**

**max\_depth = 5,**

**seed = 42)**

## Interpretation of results

The different models all resulted in an accuracy of about 72%. The best model we found was an XGBoost Classifier with an accuracy of 73% and an Area under the curve of 0.787. The addition of external data, such as holiday dates in France, did not improve the forecast accuracy.



# Model Deployment and Application

## Risk Zones

After calculating the results with the model, we now need to split the map of France into different risk zones based on the expected severity of potential accidents within each zone.

1. To achieve this we first create the **risk zones**:

* the coordinates (latitude and longitude) of each accident are used to identify its location
* the data was filtered to keep accidents that happened in mainland France and adjacent area only. This was done by keeping accidents with latitude value between 42.0 and 51.0 and longitude value between -5.0 and 10.0)
* then for each accident 4 new coordinates (2 latitudes and 2 longitudes) are calculated by rounding up each of the accident’s coordinates to the previous and next decimals.  
  **Example:** an accident with the coordinates (lat **50.51**, long **2.94**) will give the following 4 additional coordinates: lat. **50.5** and **50.6** and long. **2.9** and **3.0**
* those 4 new coordinates are used to identify a rectangular zone around each accident on the France map. Each resulting risk zone measures roughly **11.11 km** x **8.29 km** and the complete map of France is then divided into **8219** risk zones.

1. Then we calculate the **predicted accident severity per risk zone per weather condition**:

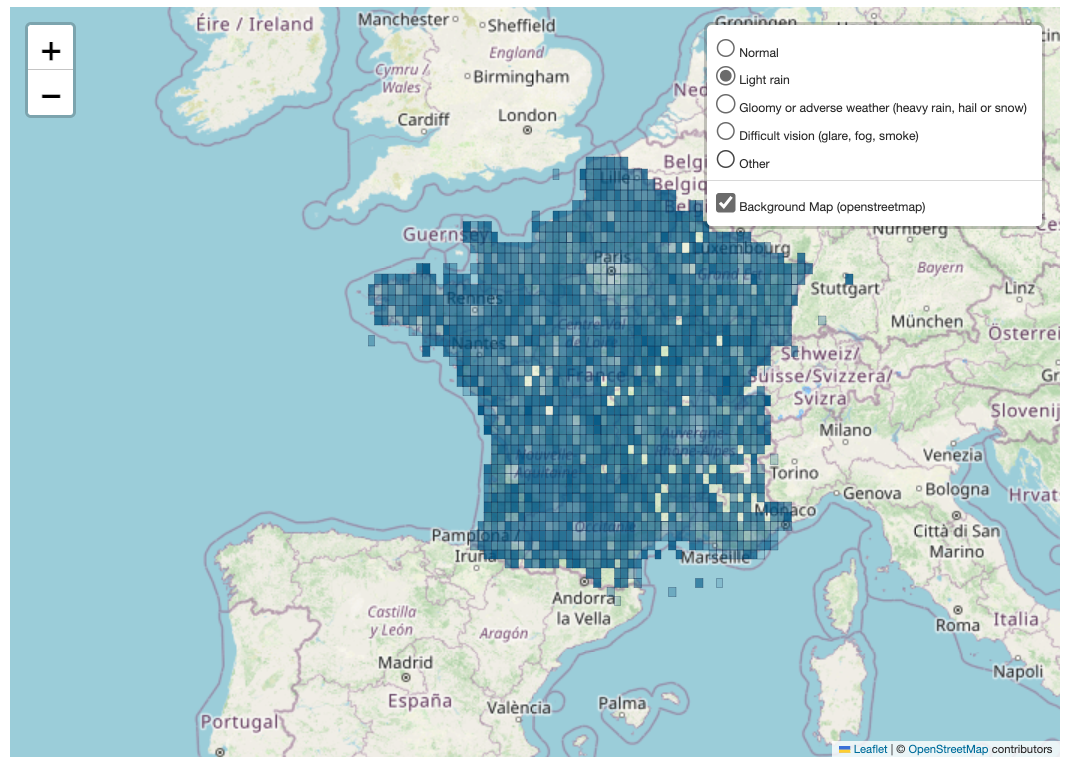
* For each risk zone, we take accidents severity predicted by our XGBoost model for all accidents that occurred within the zone
* We further split the assigned predicted severity based on the weather / vision conditions provided in the data for the accidents.
* Ten different conditions are available in the data. Similar conditions with similar average severity are then grouped together in order to keep a sufficient number of observations per risk zone per condition.
* This results in 5 different conditions: “**Normal**”, “**Light rain**”, “**Gloomy or adverse weather** (heavy rain, hail or Snow)”, “**Difficult vision** (glare, fog, smoke)”, and “**Other**”
* By splitting and aggregating our data, we ended up with an average model-predicted accident severity per risk zone per condition (weather / vision)

1. Finally we **display on an interactive map** the different risk zones with the predicted severity:

* After testing multiple options, **Folium** maps are selected to display our results
* In order to do this, we generate the risk zones on the map using two pairs of coordinates (North East and South West) for each rectangle. The opacity of the risk zones is decided by the predicted accident severity. The darker the colour, the higher the predicted severity.
* Folium allows a functionality to add layers to the map which was leveraged to show the results for the 5 different weather conditions. Selecting a weather condition on the legend would then show the corresponding layer with the correct predicted accident severity per risk zone.

## Streamlit App

The interactive map is then added to a Streamlit application allowing users to select specific weather and visibility scenarios. The map dynamically highlights the corresponding risk zones in different colors, indicating the predicted severity of road accidents. This visualization tool provides an intuitive way to assess and understand the impact of environmental conditions on road safety, aiding in better decision-making and preventive measures.



Additionally, different chapters are added to the app to summarize the work done for data exploration, visualization and pre-processing, feature engineering and modelling, as well as an introduction and a conclusion.

# Monitoring and Maintenance

To maintain model accuracy, retraining would be required annually due to the observed yearly decline in both the number of accidents and their severity in France. Additionally, before deploying the model, further plausibility checks would be necessary; for instance, we excluded overseas territories and would need a separate model for accidents occurring in regions like the Atlantic or Mediterranean.

# Conclusion

In this project, we utilized data from 2005 to 2018, as variables were redefined, and new variables were introduced from 2019 onwards, for which no prior data existed. Furthermore, we opted to create a dataset with one instance per accident. The original data was at the "person" level, so it was necessary to aggregate this information into an accident-level dataset.

The different models used all resulted in an accuracy of about 72%. The XGBoost model achieved the highest performance with a ROC-AUC score of 0.793, indicating its superior ability to balance sensitivity and specificity. The Dense Neural Network (DNN) followed closely with a score of 0.787, demonstrating its effectiveness in capturing complex patterns within the data. The Random Forest classifier also performed well, achieving a score of 0.780. However, the KNN algorithm, with a ROC-AUC score of 0.688, was less effective for this classification task. Based on these results, XGBoost is identified as the most promising model for accurately predicting accident severity, and further tuning of its hyperparameters is recommended to enhance its performance. The addition of external data, such as holiday dates in France, did not improve the forecast accuracy. Interestingly, the results indicated that accident severity was lowest during light rain conditions.

One challenge we encountered was the large number of categories in the data, particularly for unsupervised learning methods like K-Means, DBSCAN, and Optics, for which we lacked sufficient computing capacity. Specifically, for the slack package, we did not have enough computational power to assess the influence of features on the target at the model level; calculations were only feasible for individual instances.

For practical implementation, such as deployment as an API, it may be necessary to select a model that uses only the most significant features or employs a Principal Component Analysis (PCA) to ensure acceptable performance.

This solution can potentially help reduce amount of high severity road accidents. Generally speaking, such analysis can be useful for different agencies and companies; e.g.:

* Roads and transportation authorities in towns/cities can benefit from such analysis to design and optimize traffic infrastructure for safety, e.g., improving road safety (speed limits, signage, number of lanes, safety lanes, emergency refuge areas, etc.), planning of new roads, scheduling of road maintenance, etc…
* Insurers can use such analysis for processes like risk assessments, pricing, claims handling for Motor insurance
* Car manufacturers can use such analysis to identify areas of improvement for car safety
* Industries and retail that heavily rely on effective supply chain can use such analysis to avoid substantial disruptions in supply chain
* Navigation system providers and autonomous vehicles manufacturers can use such analysis to deliver functionalities like planning trip routes and suggesting safer roads This all could potentially help save a lot of lives -millions of people sustain severe or fatal injuries each year- and ease heavy financial burdens on people and institutions.

At Allianz, understanding and analyzing frequency and severity of insurable risks are a big competitive advantage. Creating classes or segments with different risk scores is a vital step in pricing insurance products and reserving sufficient funds for future claims. Creating risk zones based on geography for example is a vital part of pricing and reserving for NatCat insurance or insurance of similar risks (hazards). At the same time, predicting severity of road accidents is one of the two main actuarial tasks that actuaries try to model for Motor insurance. Predicting severity of road accidents using ML models and data shared about the accident (claims data) is becoming more widely used, this could help to triage claims, automate some claims handling steps, predict large claims common in the case of bodily injury or predict a total loss (i.e., vehicle needs to be replaced).

# Links

GitHub repository: [GitHub - Road Accidents in France (Main)](https://github.com/aperani/Sep23_RoadAccidents_repository/tree/main)

GitHub repository (for the Streamlit app): [GitHub - Road Accidents in France (Streamlit App)](https://github.com/jenpous/streamlit_app)

The Streamlit application: [Road Accident in France Streamlit](https://road-accidents-0924.streamlit.app/)

# Credits

We would like to thank the DataScientest team for the excellent training to become a data scientist. The courses were challenging and we didn't pass all exams on the first attempt. But we learned a lot with the help of the lessons, master classes and reviews. A very well thought out and well implemented curriculum! Our special thanks go to Oumaima, Sarah, Gaspard and last but not least to our project mentor Antoine, who was always there for us and because of us had to reschedule more appointments than he would have liked.

Munich and Milano

August 2024