18, rue de l'Usine - ZI Aéroport
Charguia II - 2035 Ariana
Tél. : +216 71 941 541 (LG)
Fax. : +216 71 941 889
e-mail : contact@esprit.ens.tn
www.esprit.ens.tn

esprit
Se former autrement

Project Data Science

Predict Bonus Malus Score

Elaborated By: FODHA Ghassen

FENNI Hassen

BOUDEN Rim

TOUATI Imen

HARRATH Baha Eddine

Academic Year: 2019/2020

Table of Contents

Table of figures:

GENERAL INTRODUCTION

As part of the 4th year Data Science Project, professionals offer students different concrete subjects accompanied by problems, which they might face in their professional life. The study of these problems is the subject of our group work, supervised by one professional and multiple teachers. Its purpose is to test the ability of students to work as a team on a professional subject still "unknown" to their eyes, while using the knowledge acquired during their first four years of study in ESPRIT.

The subject on which our group has been working has been entrusted to us by the CGA (comité général des assurances), and consists in predicting a bonus malus class using MachineLearning models.

Chapter I: Business Understanding

1.CGA Description

The management of this fund is entrusted to the Ministry of the Interior and Local Development. A committee called "the General Insurance Committee" is established, endowed with legal personality and financial autonomy.

Figure 1 CGA Logo

Chapter II:
Business objective

II. Business objective

Our main objectives are:

- **Predict Bonus Malus Class :**

  Create an interface that allows the insured to fill in the fields and know their bonus bonus class

- **Fraud detection:**

  Create an interface that allows the insurer to fill out a form and know if the insured is fraudulent or not

- **Sentimental analysis on different car brands**

Chapter III:
Data Analytics

III. Data Analysis :

❖ Class Bonus Malus :

Overview Reproduction Warnings 26

Dataset statistics

Number of variables	26
Number of observations	93142
Missing cells	658479
Missing cells (%)	27.2%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	51.5 MiB
Average record size in memory	579.3 B

Variable types

NUM	12
CAT	8
BOOL	3
UNSUPPORTED	3

Figure 2 Class_Bonus_Malus_Statistics

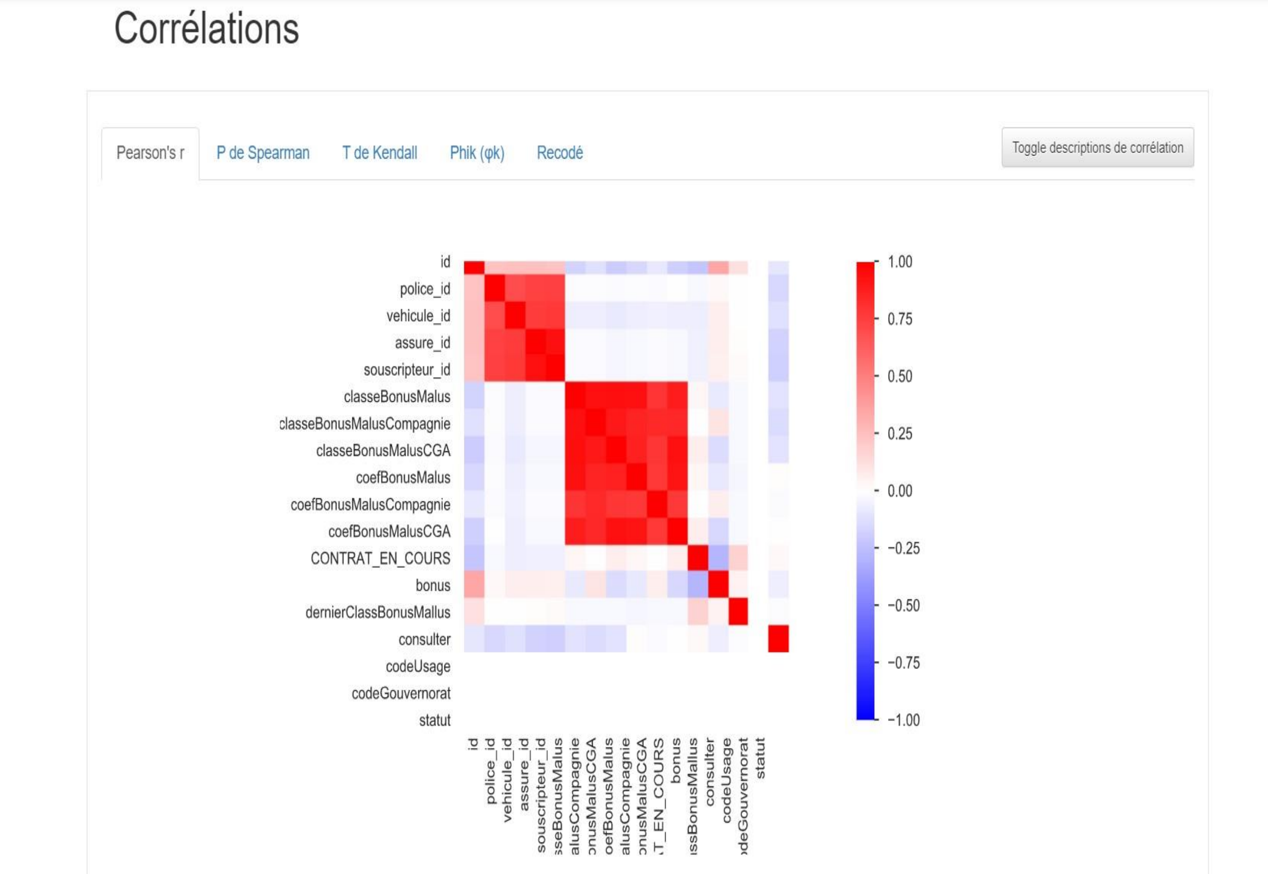

Figure 3 bonusMalus_correlation_matrix

Valeurs manquantes

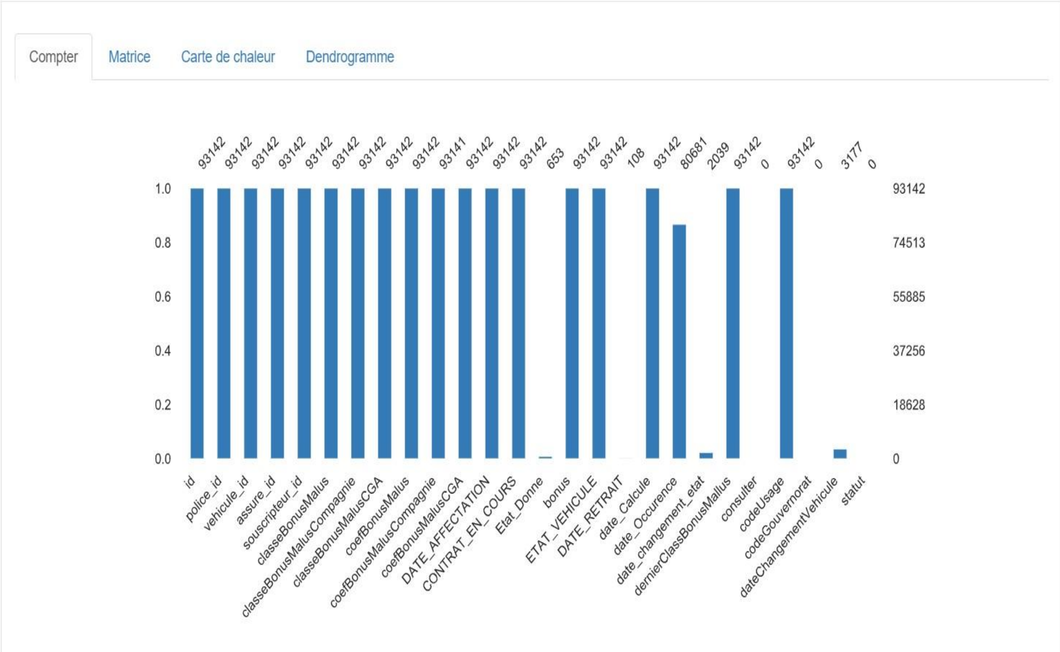

Figure 4 Valeurs manquantes

❖ Class Assure :

Overview

Overview | Reproduction | Warnings 3

Dataset statistics

Number of variables	4
Number of observations	34370
Missing cells	716
Missing cells (%)	0.5%
Duplicate rows	1
Duplicate rows (%)	< 0.1%
Total size in memory	2.9 MiB
Average record size in memory	89.0 B

Variable types

NUM	2
CAT	2

Figure 5 assure_statistics

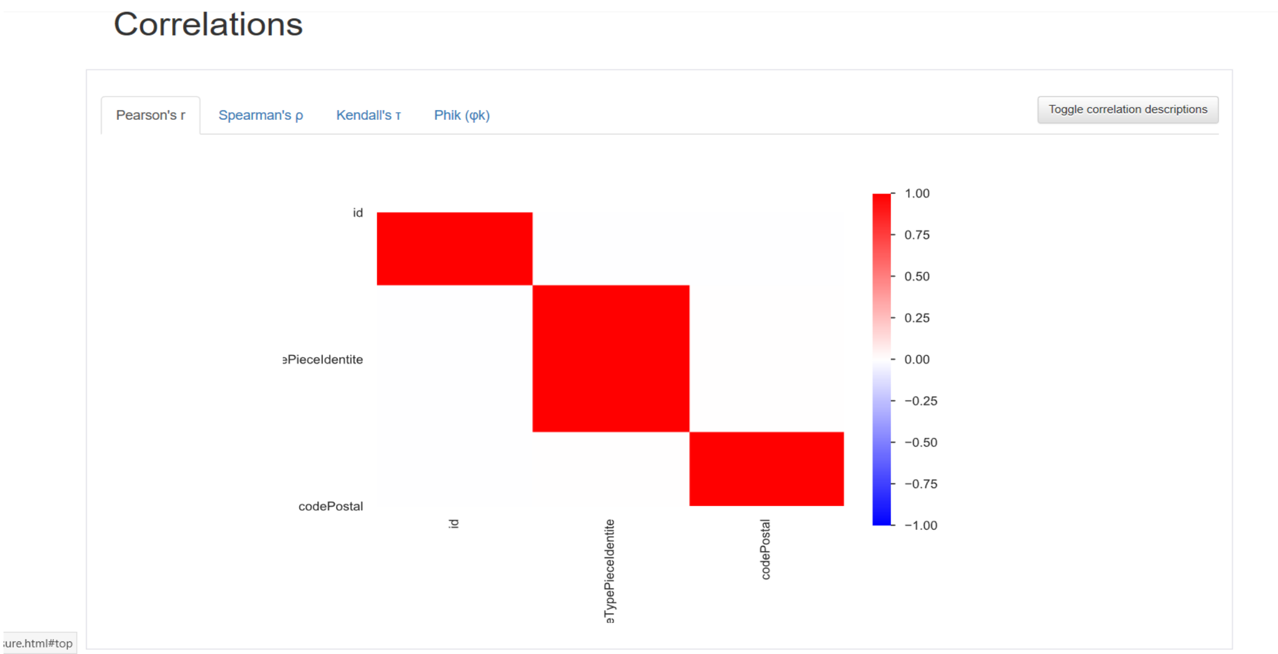

Figure 6 assure_correlation_matrix

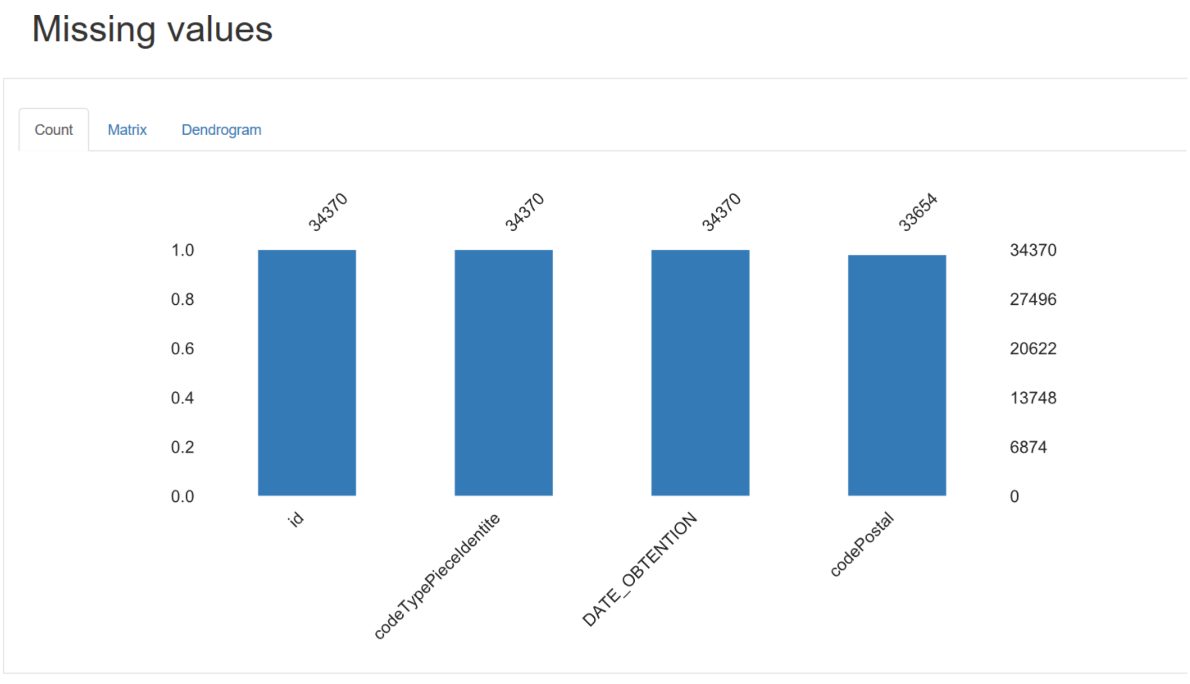

Figure 7 assure_missing_values

❖ **Class Vehicule :**

Overview

Overview | Reproduction | Warnings 15

Dataset statistics

Number of variables	15
Number of observations	92480
Missing cells	460676
Missing cells (%)	33.2%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	56.0 MiB
Average record size in memory	634.7 B

Variable types

CAT	11
NUM	3
UNSUPPORTED	1

Figure 8 vehicule_statistics

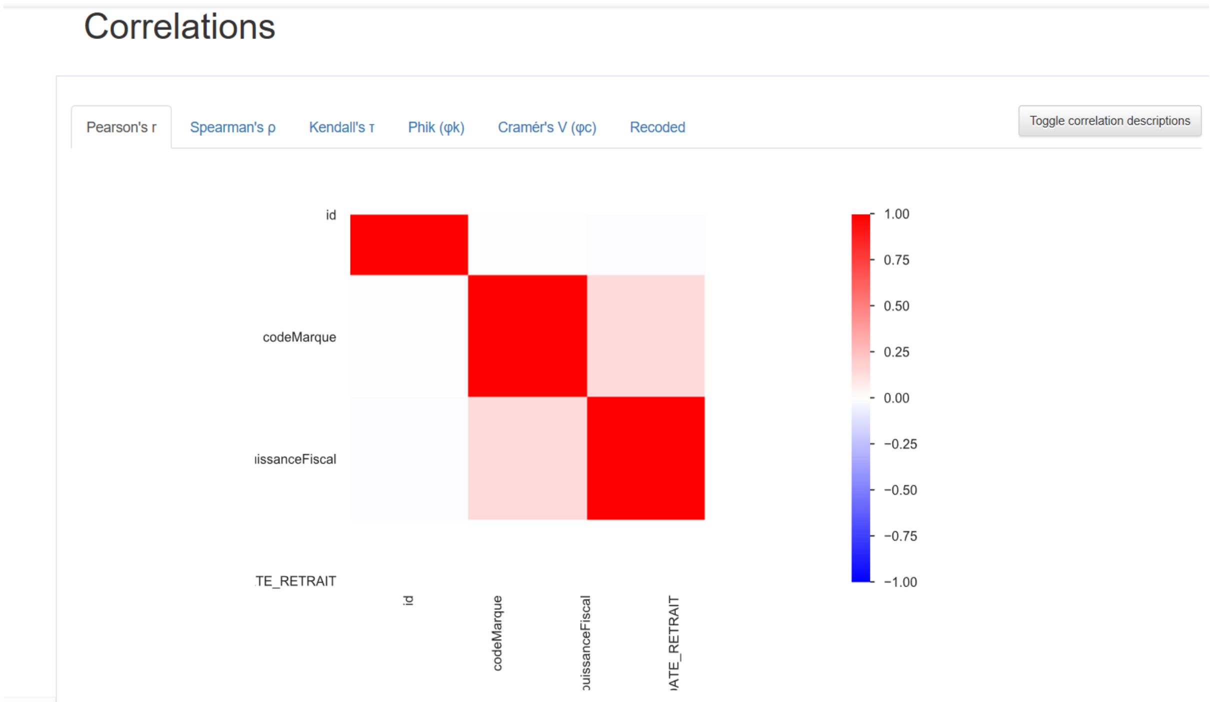

Figure 9 vehicule_correlation_matrix

Missing values

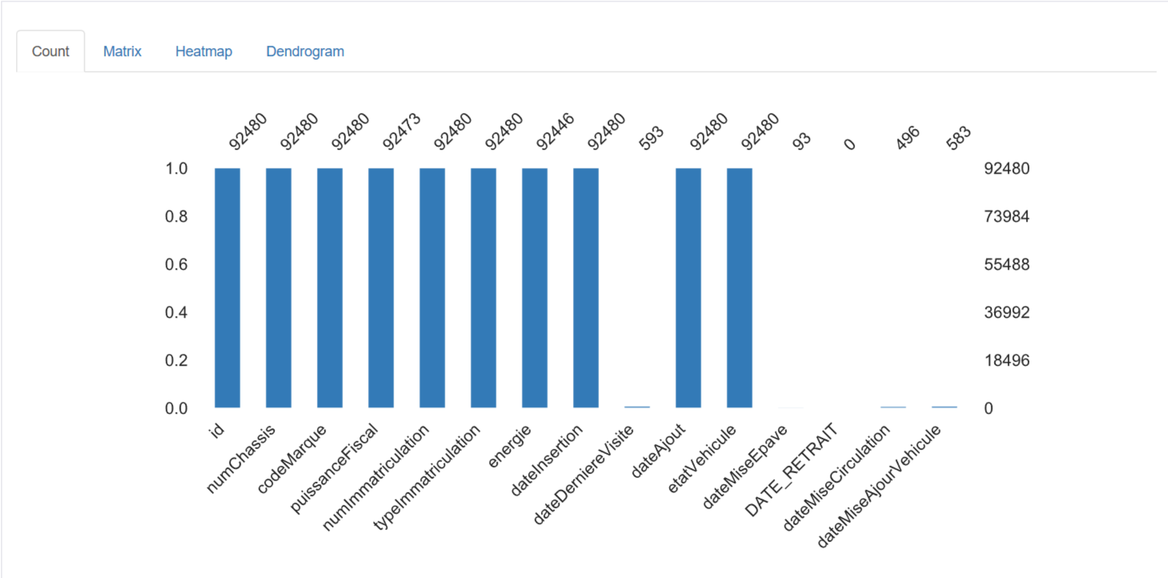

Figure 10 vehicule_missing_values

❖ Class sinister :

Overview

Overview | Reproduction | Warnings 22

Dataset statistics

Number of variables	21
Number of observations	30000
Missing cells	68037
Missing cells (%)	10.8%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	18.4 MiB
Average record size in memory	642.6 B

Variable types

CAT	10
NUM	9
UNSUPPORTED	1
BOOL	1

Figure 11 sinistre_statictics

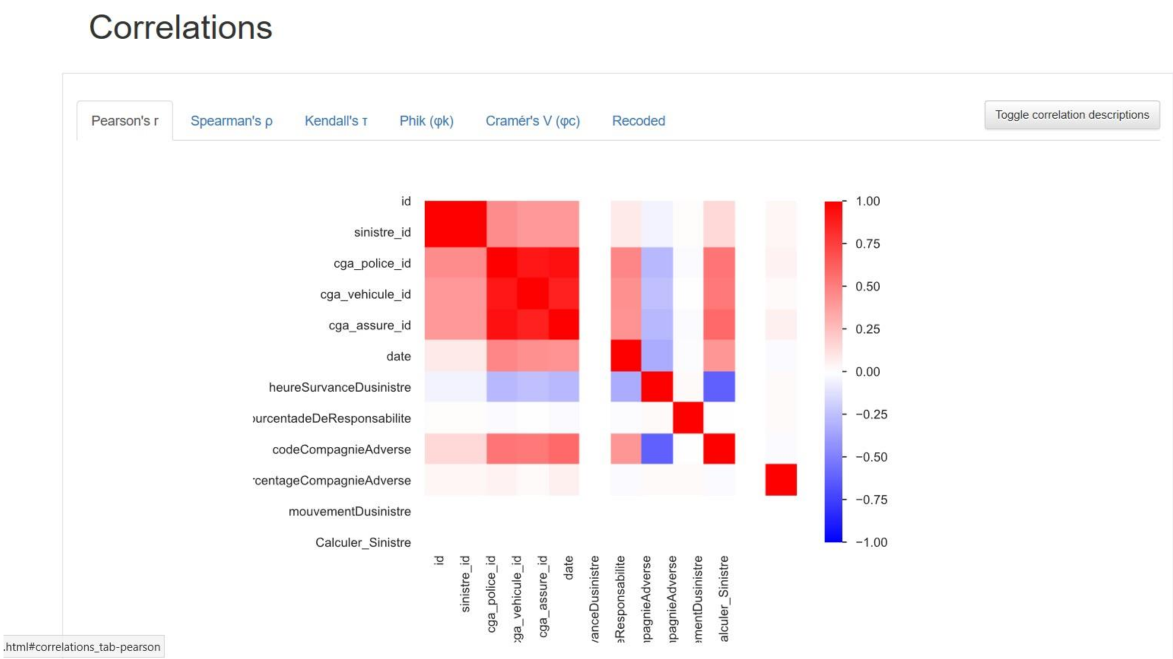

Figure 12 sinistre_correlation_matrix

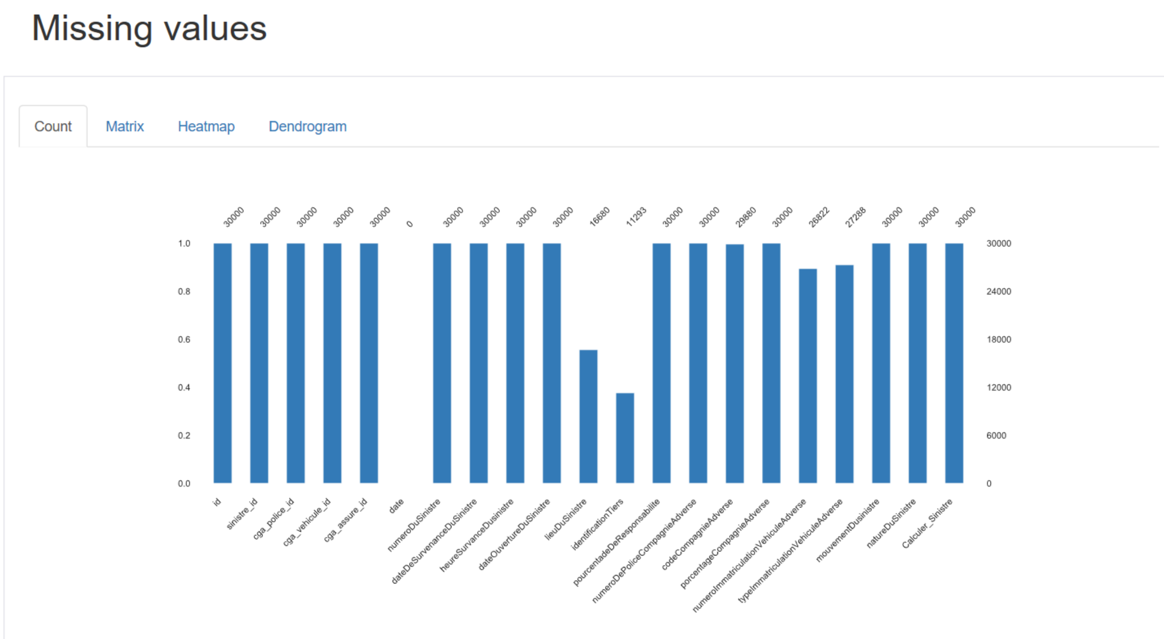

Figure 13 sinistre_missing_values

❖ **Class epave :**

Overview

Overview | Reproduction | Warnings 40

Dataset statistics

Number of variables	33
Number of observations	7209
Missing cells	81742
Missing cells (%)	34.4%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	7.6 MiB
Average record size in memory	1.1 KiB

Variable types

CAT	16
UNSUPPORTED	9
NUM	5
BOOL	3

Figure 14 epave_statistics

Correlations

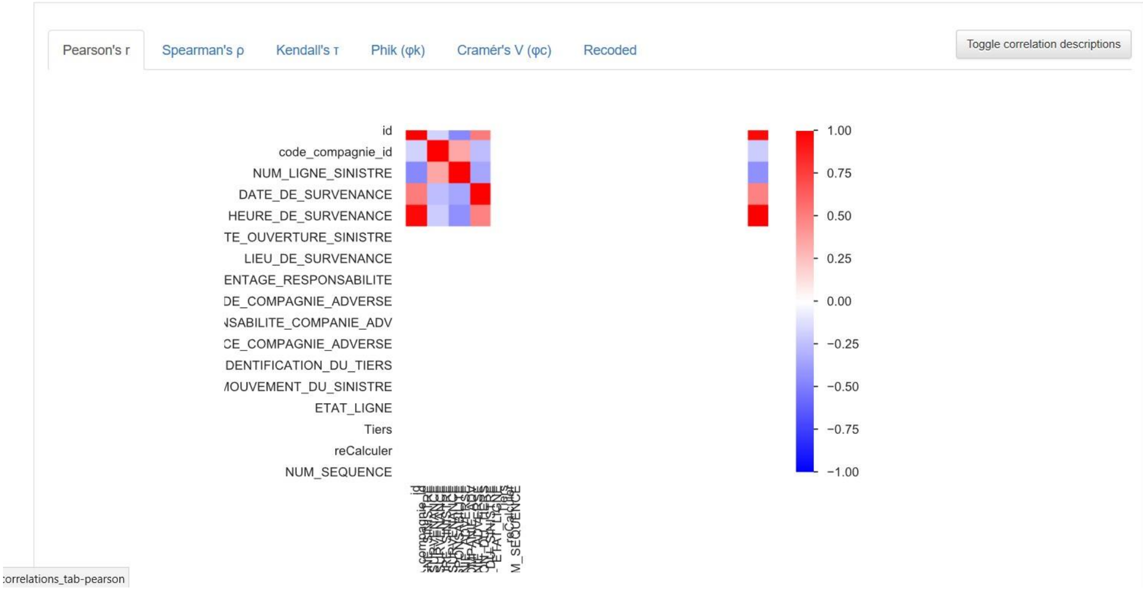

Figure 15 epave_ correlation_matrix

Missing values

Count | Matrix | Heatmap | Dendrogram

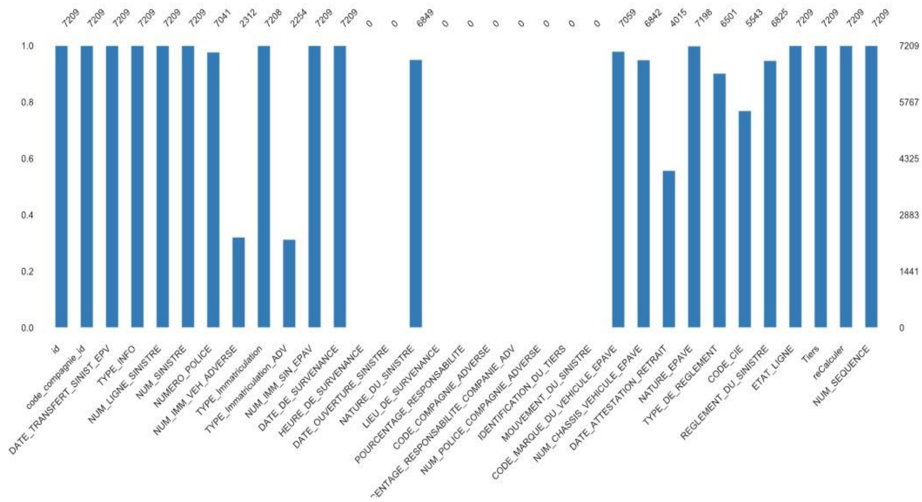

Figure 16 epave_Missing_values

❖ Class marque :

Overview

Overview | Reproduction | Warnings 10

Dataset statistics

Number of variables	8
Number of observations	1913
Missing cells	2571
Missing cells (%)	16.8%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	785.4 KiB
Average record size in memory	420.4 B

Variable types

CAT	6
NUM	2

Figure 17 marque_statistics

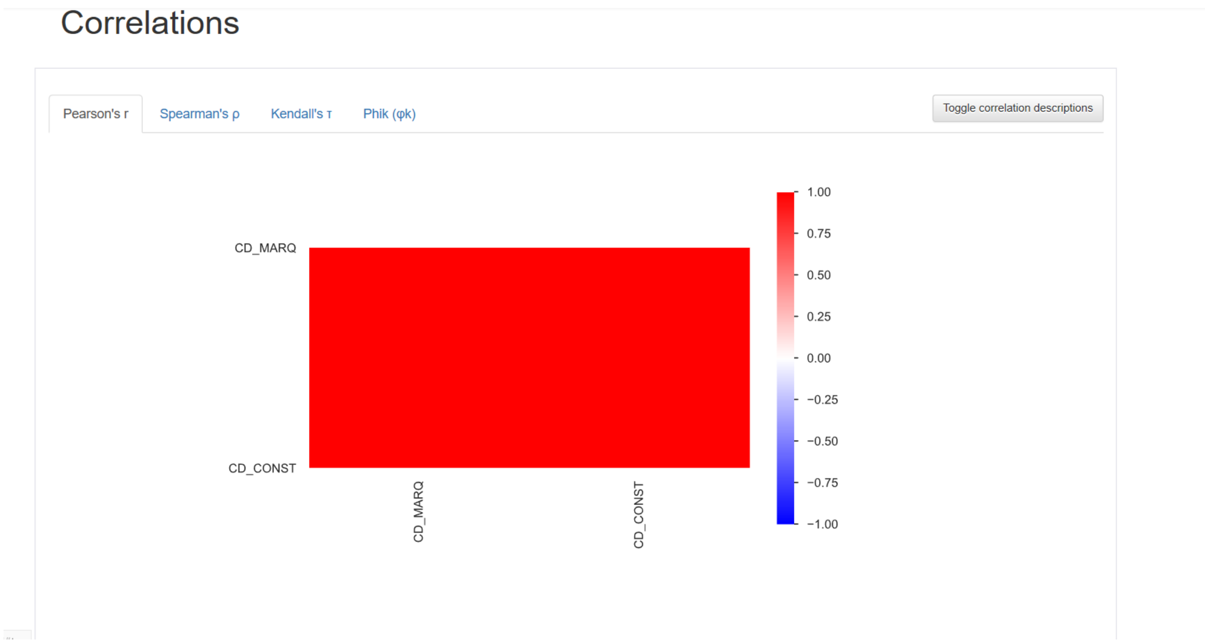

Figure 18 marque_ correlation_matrix

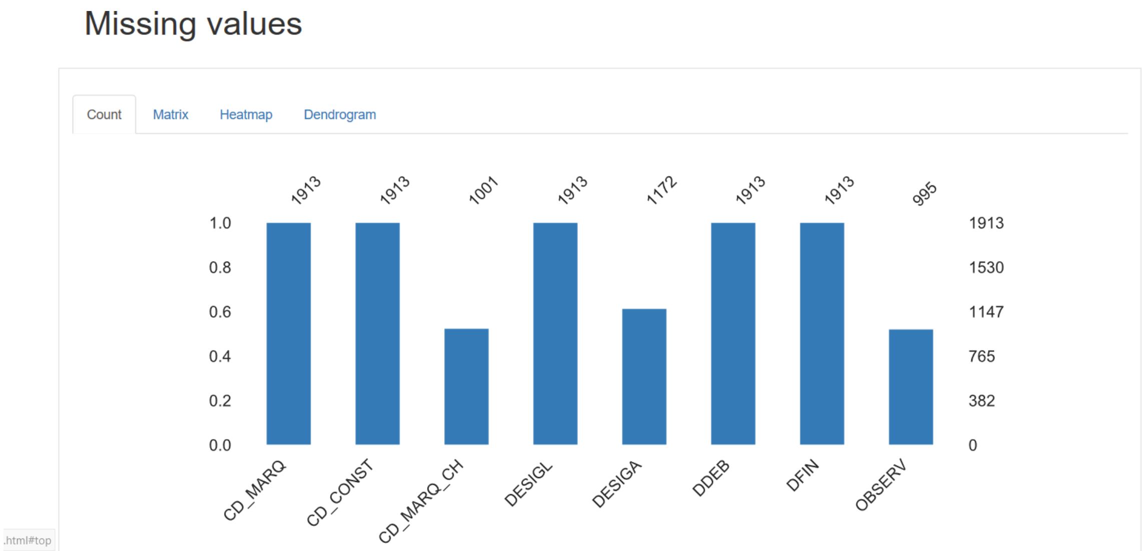

Figure 19 marque_Missing_values

❖ **Class Police :**

Overview

Overview | Reproduction | Warnings 17

Dataset statistics

Number of variables	17
Number of observations	94620
Missing cells	554408
Missing cells (%)	34.5%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	44.9 MiB
Average record size in memory	498.1 B

Variable types

CAT	9
NUM	4
UNSUPPORTED	3
BOOL	1

Figure 20 police_statistics

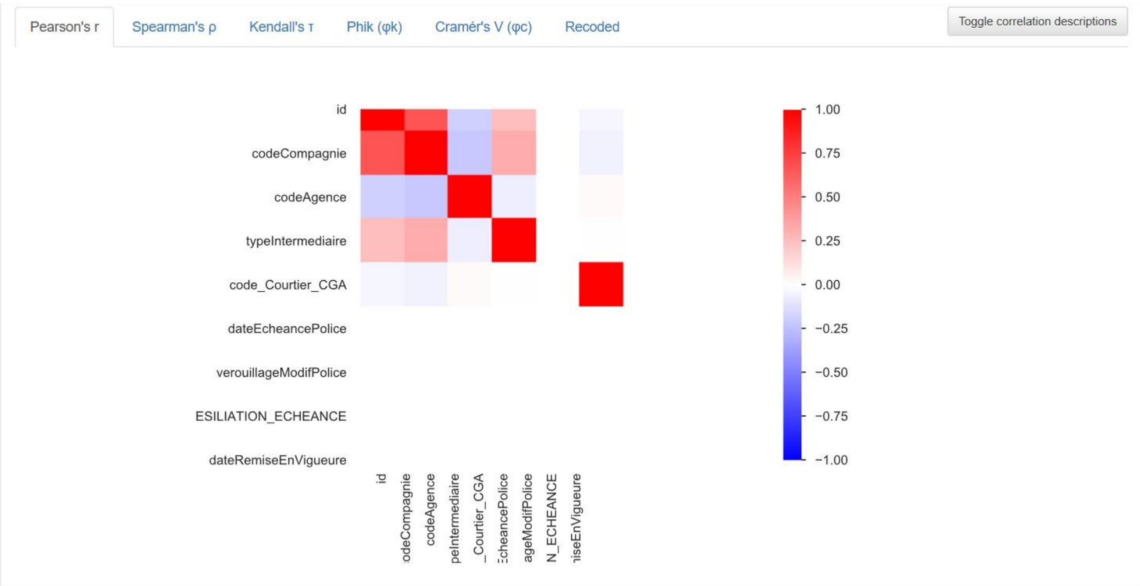

Figure 21 police_ correlation_matrix

Missing values

Count Matrix Heatmap Dendrogram

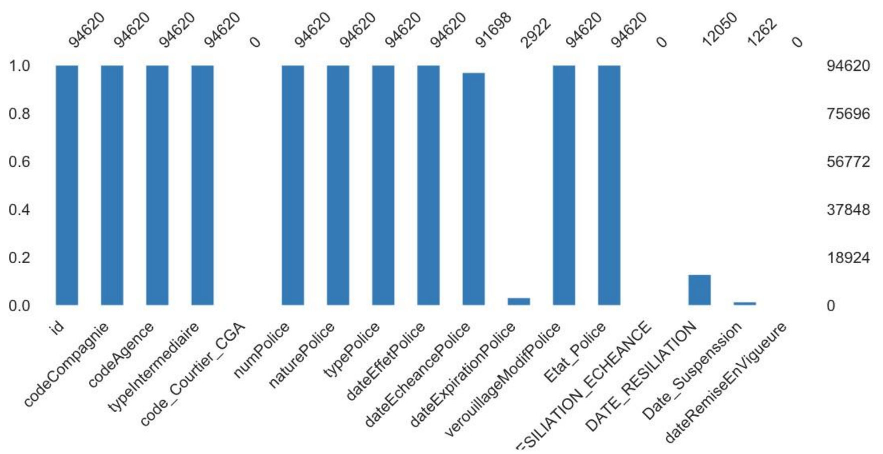

Figure 22 police_Missing_values

❖ **Class usage :**

Overview

Overview Reproduction Warnings 2

Dataset statistics

Number of variables	5
Number of observations	17
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	2.9 KiB
Average record size in memory	172.5 B

Variable types

CAT	2
NUM	2
BOOL	1

Figure 23 usage_statistics

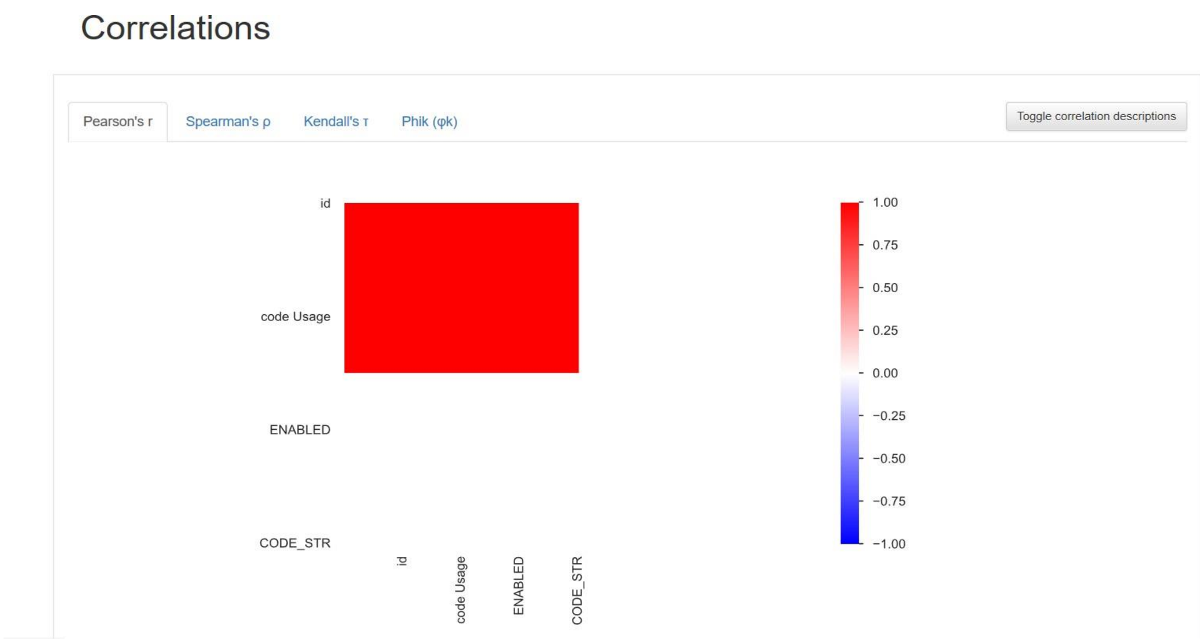

Figure 24 usage_ correlation_matrix

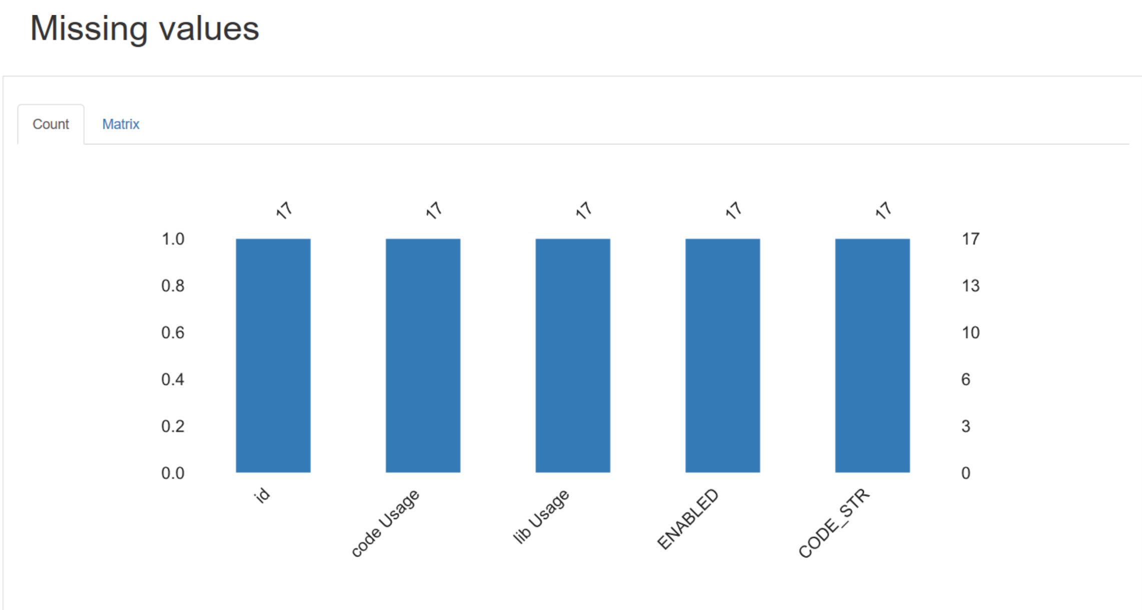

Figure 25 usage_missing_values

Chapter IV:
Data Preparation

A. Internal data :

1. Data Cleaning :

➢ **Data ClassBonusMalus :**

- The ClassBonusMalus table contains 93,142 rows and 26 columns. At the level of this table, the columns « Etat_Donne,DATE_RETRAIT,date_changement_etat,consulter,codeGourvernorat,dateChangementVehicule,statut » contain a lot of missing values (higher than 85000) so we deleted them.

- We have also dropped the columns: "classeBonusMalusCompagnie", "classeBonusMalusCGA", "coefBonusMalusCompagnie", "coefBonusMalusCGA" and "CoefBonusMalus" because we will only need the "classeBonusMalus" column to make our predictions. So the shape of this table became (93.142,14).

➢ **Data Sinistre :**

- The table « Sinistre » contains 30,000 rows and 21 columns. At the level of this table, some of this columns "id", "date", "dateDeSurvenanceDuSinistre", "heureSurvanceDusinistre", "numeroDuSinistre", "numeroDePoliceCompagnieAdverse", "codeCompagnieAdverse", "typeImmatriculationVehiculeAdverse","numeroImmatriculationVehiculeAdverse"," mouvementDusinistre" contain a lot of missing values and the others we don't need them for our prediction, so we deleted them.

- We replaced the missing values in the " lieuDuSinistre" column by "non mentionné" and "identificationTiers" by "autres".

- The shape of this table became (30000,11)

➢ **Data Vehicule :**

- The table « Vehicule » contains 92,480 rows and 15 columns. At the level of this table, the columns « dateDerniereVisite», « dateMiseEpave», « DATE_RETRAIT», « dateMiseCirculation», « dateMiseAjourVehicule» contain several missing values, so we deleted them, and then we did a dropna on the rest of the columns containing a small amount of missing values.We also dropped the « numChassis», « numImmatriculation»,« typeImmatriculation»,«dateInsertion», «dateAjout», « codeMarque» because we don't need them for our prediction .So the shape of this table became (92.446,4).

- **Data UsageCGA :**

  - The table « UsageCGA » contains 17 rows and 5 columns, it is already clean (nothing has been deleted).But on the other hand, the "CODE_STR" column contains only one value so we deleted it.

- **Data Police :**

  - The table « Police » contains 94,620 columns and 17 rows. The columns « code_Courtier_CGA», « dateExpirationPolice», « RESILIATION_ECHEANCE», « DATE_RESILIATION» ,« Date_Suspension» « dateRemiseEnVigueure» contain several missing values, so we deleted them and we replaced the missing values in the column « DateEcheancePolice» by 0. So the shape of this table became (94.620,11).

2. Merging Data :

- 1st step: we did a left merge between the “Police” and “classBonusMalus” tables into “df1”.
- 2nd step: we did a left merge between the “df1” and “Vehicule” tables into “df2”.
- 3rd step: we did a left merge between the “df2” and “usage” tables into “df3”.
- 4th step: we did a left merge between the “df3” and “sinistre” tables into “df4”.
- After merging all the tables in dfFinal, we deleted all the columns that contain ids: «numeroDuSinistre» , «codeUsage», «codeMarque», «codeCompagnie», «codeAgence», «numPolice», « id ».

3. Cleaning the merged data :

- We replaced the missing values caused by the left merge in « puissanceFiscal » column of «df2» by 0 and « energie », « etatVehicule » columns by « - ».
- We replaced the missing values caused by the left merge in « libUsage » column of «df3» by « - »
- We deleted “cga_vehicule_id” and “cga_assure_id” columns from df4 because we don’t need them for our prediction.

- We replaced the missing values caused by the left merge in "sinistre_id" column of df4 by 0 "dateOuvertureDuSinistre","lieuDuSinistre","identificationTiers","natureDuSinistre" columns of df4 by “-“and "Calculer_Sinistre", ","pourcentadeDeResponsabilite", "porcentageCompagnieAdverse"columns of df4 by “-1”

- We deleted 'DATE_AFFECTATION', 'dateEffetPolice', 'date_Calcule', 'dateOuvertureDuSinistre', 'dateEcheancePolice' from our database because we don’t need them for our prediction.

Fraudulent data:

We have noticed that our database contains a lot of fraudulent data, so we are going to eliminate it by creating an algorithm which is based on the following mechanism:

- The insured benefits from a transition to the lower class if he spends two years without having a responsible accident (the system only takes into account accidents in which the insured is responsible)

- The insured has a penalty of an increase of two classes if his liability is involved in a personal accident

- The insured has a penalty of one class increase if he is involved in a material accident.

After this fraud detection algorithm, an additional column has been created named Fraud which contains two methods 1: for fraudulent data and 0: for clean data.

For the rest of our project we will only the clean data (none fraudulent) so the shape of our new database became (4413,35).

4. Data encoding :

- DataFinal contains several dates and several qualitative variables,we encoded qualitative variables using TargetEncoder (Target encoding is the process of replacing a categorical value with the mean of the target variable. Any non-categorical columns are automatically dropped by the target encoder model)

- As for dates, we split each date into its individual parts: year,month,day.

Here’s what our final data looks like:

```
Data columns (total 44 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   police_id                     96671 non-null  int64
 1   typeIntermediaire             96671 non-null  int64
 2   NbrSinistre                   96671 non-null  float64
 3   NbrSinistreM                  96671 non-null  float64
 4   NbrSinistreC                  96671 non-null  float64
 5   verouillageModifPolice        96671 non-null  int64
 6   vehicule_id                   96671 non-null  int64
 7   assure_id                     96671 non-null  int64
 8   classeBonusMalus              96671 non-null  int64
 9   coefBonusMalus                96671 non-null  int64
 10  CONTRAT_EN_COURS              96671 non-null  int64
 11  bonus                         96671 non-null  int64
 12  dernierClassBonusMallus       96671 non-null  int64
 13  puissanceFiscal               96671 non-null  float64
 14  sinistre_id                   96671 non-null  float64
 15  pourcentadeDeResponsabilite   96671 non-null  float64
 16  porcentageCompagnieAdverse    96671 non-null  float64
 17  Calculer_Sinistre             96671 non-null  float64
 18  Duree                         96671 non-null  int64
 19  minE                          96671 non-null  int64
 20  Fraud                         96671 non-null  int64
 21  energie                       96671 non-null  float64
 22  etatVehicule                  96671 non-null  float64
 23  naturePolice                  96671 non-null  float64
 24  typePolice                    96671 non-null  float64
 25  Etat_Police                   96671 non-null  float64
 26  libUsage                      96671 non-null  float64
 27  lieuDuSinistre                96671 non-null  float64
 28  identificationTiers           96671 non-null  float64
 29  natureDuSinistre              96671 non-null  float64
 30  annee_effet_police            96671 non-null  int64
 31  mois_effet_police             96671 non-null  int64
 32  jour_effet_police             96671 non-null  int64
 33  annee_DATE_AFFECTATION        96671 non-null  int64
 34  mois_DATE_AFFECTATION         96671 non-null  int64
 35  jour_DATE_AFFECTATION         96671 non-null  int64
 36  annee_date_Calcule            96671 non-null  int64
 37  mois_date_Calcule             96671 non-null  int64
 38  jour_date_Calcule             96671 non-null  int64
 39  annee_dateOuvertureDuSinistre 96671 non-null  int64
 40  mois_dateOuvertureDuSinistre  96671 non-null  int64
 41  jour_dateOuvertureDuSinistre  96671 non-null  int64
 42  mois_dateEcheancePolice       96671 non-null  int64
 43  jour_dateEcheancePolice       96671 non-null  int64
dtypes: float64(17), int64(27)
memory usage: 33.2 MB
```

Figure 26 Final Data

B. External data :

This part is dedicated for detailing the collection process of the external data.

1 Web Scraping :

We can mention the source on the web which we applied on our scraping algorithms:

- https://www.twitter.com

2 Scraping Tools :

For extracting data from the Web, we used Python which is an interpreted high-level programming language for general-purpose programming.

Python has a large variety of frameworks for web scraping. We 'Tweepy' which is a Python library used for accessing the Twitter API.

Figure 27: Python Logo **Figure 28: Tweepy Logo**

Chapter V: Data Modeling and Evaluation

A. Data Modeling:

- Our targets «classBonusMalus” and “Fraud” are categoricals variables so for the predictions we will use a supervised classification models .In machine learning we have a lot of models like KNN, xgboost,random forrest,SVM…

- After testing all these models, we will limit ourselves to random forest and xgboost because they are more efficient than the others

Xgboost :

XGBoost (EXtreme Gradient Boosting) is an optimized open source implementation of the gradient boost trees algorithm which is a supervised learning algorithm whose principle is to combine the results of a simpler set of models in order to provide a better prediction.In other words xgboost combine decision trees,and start the combining process at the beginning, not at the end.

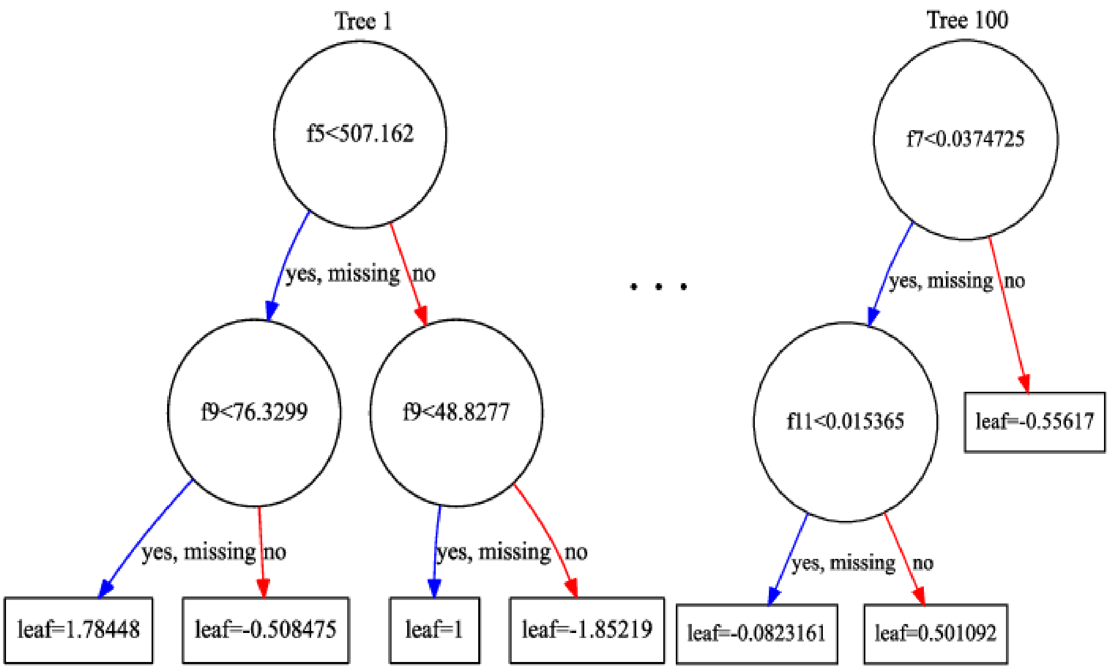

Figure 29: Xgboost Model Principle

Random Forrest:

The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction is more accurate than any other individual tree. In other words, **Random forests** are a large number of trees, combined (using averages or "majority rules") at the end of the process.

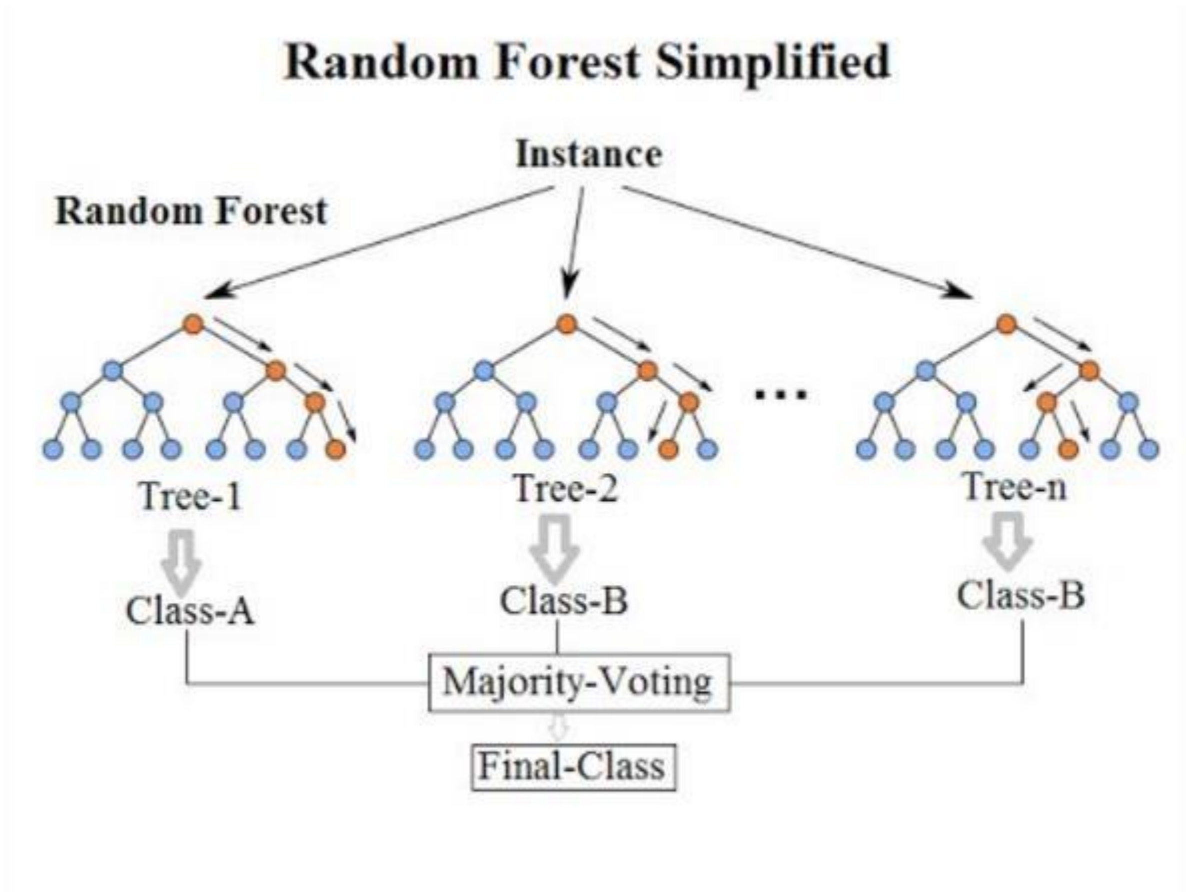

Figure 30: Random Forest Model Principle

B. Evaluation:

B.1 : Technichal :

- **Train/Test Set validation:** The principle of this method is to divide the sample of size n into training sample (> 60% of the sample) and test sample. The model is built on the learning sample and validated on the test sample.

- **Cross validation test:** In this method, we divide the sample k times, then select one of the k samples as validation set and the (k-1) other samples will constitute the learning package. Then repeat the operation by selecting another validation sample. The operation is repeated k times so that ultimately each subsample has been used exactly once as a set of validation.

➔ **While using these two methods, we noticed that the cross validation test gives better results than the test set validation with any chosen model.**

B.2: Indicators:

- **Accuracy:** The accuracy of a machine learning classification algorithm is one way to measure how often the algorithm classifies a data point correctly. Accuracy is the number of correctly predicted data points out of all the data points. More formally, it is defined as the number of true positives and true negatives divided by the number of true positives, true negatives, false positives, and false negatives.

- **Confusion matrix:** The confusion matrix is a summary of the prediction results on a classification problem.

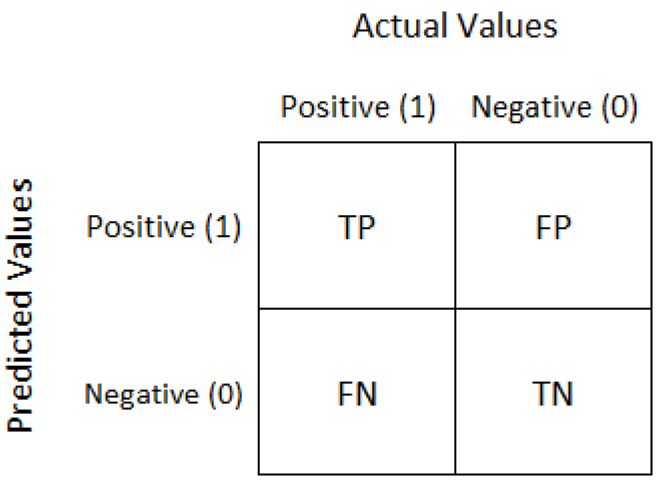

Figure 31: Confusion Matrix

A true positive or true negative is a data point that the algorithm correctly classified as true or false, respectively. A false positive or false negative, on the other hand, is a data point that the algorithm incorrectly classified.

- **Precision:** What percent of your predictions were correct? Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class, it is defined as the ratio of true positives to the sum of a true positive and false positive

- **Recall:** What percent of the positive cases did you catch? Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

- **F1 score:** What percent of positive predictions were correct? The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

- **Support:** is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn't change between models but instead diagnoses the evaluation process.

B.3: Detection Fraud Evaluation:

- **Random Forest:**

```
Accuracy test:  0.9167610419026048
Accuracy train:  1.0
```

Figure 32: Accuracy Random Forest 1

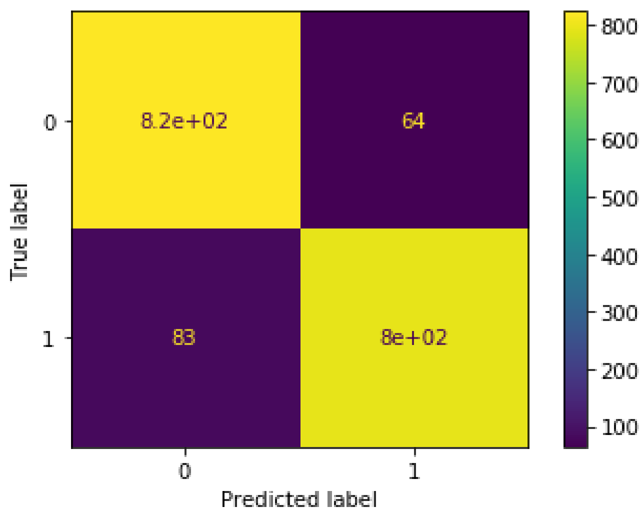

Figure 33: Confusion Matrix Random Forest 1

➔ This model gives higher true positive and true negative values than false positive and false negative values, which shows the performance of this model.

- **XGboost:**

```
Accuracy:  0.9597961494903737
Accuracy train:  0.9960339943342776
```

Figure 34: Accuracy XGboost 1

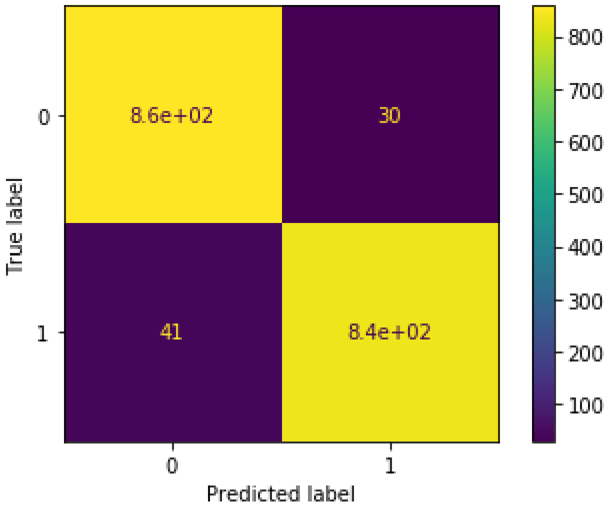

Figure 35: Confusion Matrix XGboost 1

➔ This model gives higher true positive and true negative values than false positive and false negative values, which shows the performance of this model.

⇨ **XGboost gives higher accuracy than Random Forest, so we chose it for fraud detection**

B.3: Class Bonus Malus classification Evaluation:

- **Random Forest:**

```
Accuracy:  0.9173272933182333
Accuracy train:  0.9994334277620397
```

Figure 36: Accuracy train/test set Random Forest 2

```
Mean Accuracy:  0.8905601651759231
```

Figure 37: Accuracy cross validation set Random Forest 2

➔ The train/test set technic gives higher accuracy than the cross validation that's why we kept the train/test set with the random forest in this classification .

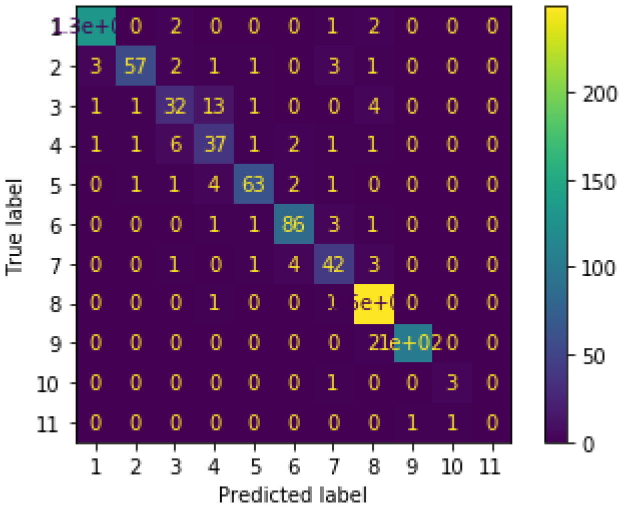

Figure 38: Confusion Matrix train/test set Random Forest 2

➔ This model gives higher true positive and true negative values than false positive and false negative values, which shows the performance of this model.

- **XGboost:**

```
Accuracy score (training): 0.999
Accuracy score (validation): 0.967
```

Figure 39: Accuracy train/test set XGboost 2

```
Mean Accuracy:  0.9551374796804339
```

Figure 40: Accuracy cross validation set XGboost 2

➔ The train/test set technic gives higher accuracy than the cross validation that's why we kept the train/test set with the xgboost in this classification .

Chapter VI: Deployment

- The deployment of machine learning models is the process for making your models available in production environments, where they can provide predictions to other software systems.To make our models available we made a Web Site using Dash

- **Dash:** a user interface library for creating analytical web applications. Those who use Python for data analysis, data exploration, visualization, modelling, instrument control, and reporting will find immediate use for Dash. It makes it dead-simple to build a GUI around your data analysis code.Dash app code is declarative and reactive, which makes it easy to build complex apps that contain many interactive elements.

Figure 41: Dash Logo

- In this project, we made two forms, one for the prediction of fraud and the other for the classification of the Bonus Malus class.

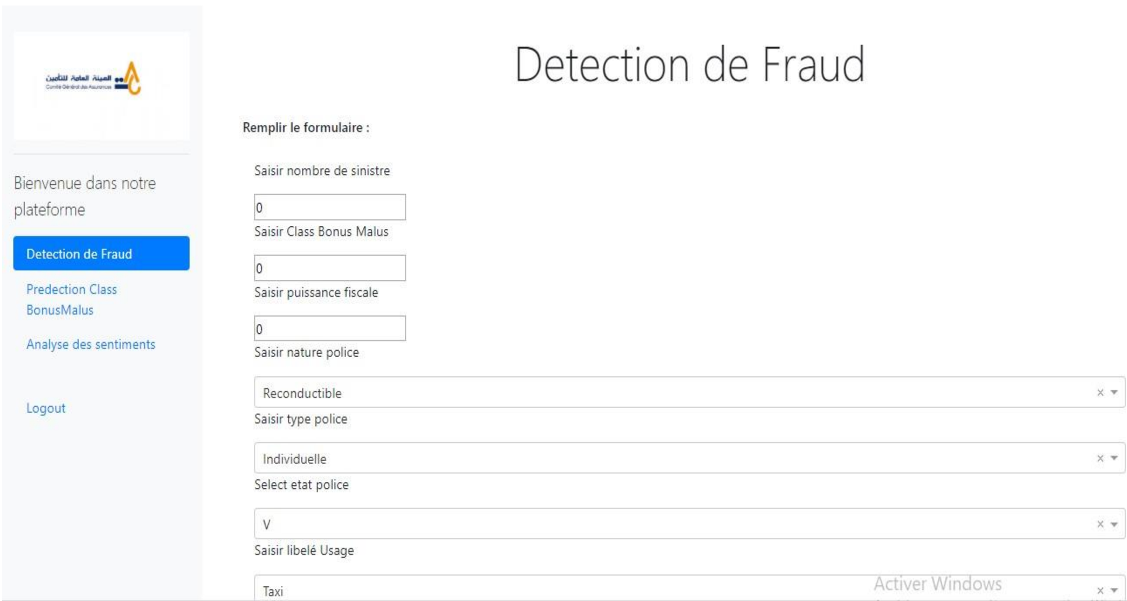

Figure 42: Detection Fraud Form

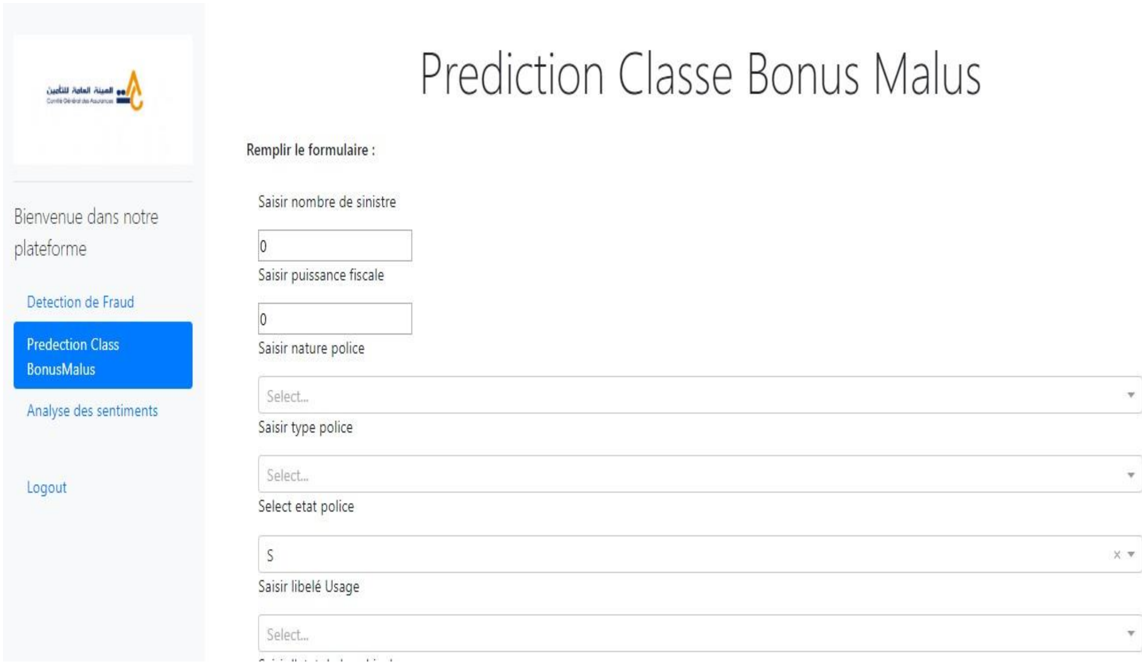

Figure 43: Prediction class Bonus Malus Form

Conclusion:

This project has been very enlightening to us as it gave us the chance to work with real world data and provided us with a clear scope over different tools and methodologies involved in a Data Science project.

We are grateful for this opportunity that allowed us work together towards a challenging set of goals and hope our work will have an impact in reducing insurance frauds and ultimately, in reducing the mortality rate of car accidents in Tunisia.

Perspectives:

- Additional data could help improve the way we detect fraudulent activity. Public records such as criminal records, judgments, foreclosures, address change frequency, bankruptcies, history of rejected claims are all data sources that can be integrated into our model.
- Technologies such as social network analysis can be integrated into our fraud identification process to detect if someone has a relationship to another individual with a prior case of fraud allowing us to have a "score" for different networks.