

DATA SCIENCE IN CAR INSURANCE

Supervised by

Mr Barkeoui Ahmed

Mme Mosbah Sonia

Mme Trabelsi Dorra

Elaborated by

Barkallah Ahmed

Jebari Aziz

Mbarki Mohamed Amine

Mlaouhi Yahya

Mnif Noura

Omri Marwen

2020-2021

TABLE OF CONTENTS

INTRODUCTION.....	6
1. CRISP DM.....	6
Business understanding	8
1. CGA	8
2. Business objectives.....	8
3. Data Science objectives.....	9
Data analysis	10
1. Internal data	10
2. external data	11
a) web scrapping	11
b) Forms	13
Data preparation.....	14
1. Internal data	14
a) DATA WAREHOUSE.....	14
b) FUSION.....	14
c) Data cleaning.....	17
d) Encoding	21
2. External data.....	22
Data modelling.....	23
l. Internal data	23
1. Naïve Bayes	23
a) Building the model.....	23
b) Evaluation.....	23
2. KNN.....	24
a) Building the model.....	24
b) Evaluation.....	24
3. Multi Layer Perceptron (MLP)	25
a) Building the model :	26
b) Evaluation.....	26
4. Decision Tree	26
a) Building the model :	27
b) Evaluation.....	27
5. Random Forest	27
a) Building the model.....	27

b)	Evaluation.....	28
6.	SVM.....	28
a)	Building the model.....	28
b)	Evaluation.....	29
7.	Logistic regression	29
a)	Building the model.....	29
b)	Evaluation.....	29
8.	Multi class classification.....	30
a)	Building the model.....	30
b)	Evaluation.....	31
II.	External data.....	31
1.	K-means	31
a)	Building the model.....	32
b)	Evaluation.....	32
	Deployment	33
	Conclusion.....	36

TABLE OF FIGURES

FIGURE 1 MACHIE LEARNING	6
FIGURE 2 CRISP DM	7
FIGURE 3 LOGO CGA.....	8
FIGURE 4 CGA TABLES.....	10
FIGURE 5 SELENIUM LOGO.....	11
FIGURE 6 REVIEWS ABOUT THE BEST INSURANCE IN TUNISIA	11
FIGURE 7 PARSE HUB LOGO	12
FIGURE 8 RATING THE CAR INSURANCE	12
FIGURE 9 CAR INSURANCE COVERAGE	13
FIGURE 10 VEHICLE USAGE.....	13
FIGURE 11 REASON OF SWITCHING COMPANY	13
FIGURE 12 SWITCH INSURER	13
FIGURE 13 DATA WAREHOUSE.....	14
FIGURE 14 MERGE POLICE.CSV AND POLICE_CGA INTO PP	15
FIGURE 15 MERGE SINISTRE.CSV AND SINISTRE_CGA INTO SS	15
FIGURE 16 MERGE VEHICLE.CSV AND VEHICLE_CGA INTO VV	15
FIGURE 17 CREATING BMV	16
FIGURE 18 CREATING FAIT_BM	16
FIGURE 19 CREATING SSVV	16
FIGURE 20 CREATING FAIT_SINISTRE.....	17
FIGURE 21 DROP ALL THE UNNECESSARY DATA FROM BM_FAIT	17
FIGURE 22 DROP ALL THE UNNECESSARY DATA FROM SINISTRE_FAIT	17
FIGURE 23 FILL MISSING VALUES IN ENERGY BY UNKNOWN	18
FIGURE 24 FILL MISSING VALUES IN TYPE INTERMEDIARE BY 1 OR 2 OR 3.....	18
FIGURE 25 DELETE ALL CLIENTS WHO HAVE A POLICY EQUAL TO NULL.....	18
FIGURE 26 ADDING A NEW COLUMN NAMED NUM_ACCIDENTS.....	18
FIGURE 27 CLEANED DATA	20
FIGURE 28 EXTRACT THE CLIENT WHO HAD MADE ACCIDENTS	20
FIGURE 29 CORRELATION OF PEARSON.....	21
FIGURE 30 ENCODING USING LABEL ENCODER	21
FIGURE 31 ENCODING USING GET DUMMIES	21
FIGURE 32 TRANSFORM THE COLUMN OF RATING TO FLOAT:	21
FIGURE 33 INSURANCE VOTE	22

FIGURE 34 EXAMPLE OF NAIVE BAYES	23
FIGURE 35 MODEL OF NAIVE BAYAS.....	23
FIGURE 36 ACCURACY OF THE MODEL.....	23
FIGURE 37 EXAMPLE OF KNN	24
FIGURE 38 MODEL OF KNN	24
FIGURE 39 ACCURACY OF THE MODEL.....	24
FIGURE 40 ACCURACY FOR EACH CLIENT	25
FIGURE 41 EXAMPLE OF MLP.....	25
FIGURE 42 MODEL OF MLP	25
FIGURE 43 ACCURACY OF MLP.....	26
FIGURE 44 EXAMPLE OF DECISION TREE	26
FIGURE 45MODEL OF DECISION TREE.....	27
FIGURE 46ACCURACY OF DECISION TREE.....	27
FIGURE 47EXAMPLE OF RANDOM FOREST	27
FIGURE 48MODEL OF RANDOM FOREST	28
FIGURE 49ACCURACY OF RANDOM FOREST	28
FIGURE 50 EXAMPLE OF SVM	28
FIGURE 51 MODEL OF SVM	28
FIGURE 52 ACCURACY OF SVM	29
FIGURE 53 EXAMPLE OF LOGISTIC REGRESSION.....	29
FIGURE 54 MODEL OF LOGISTIC REGRESSION	29
FIGURE 55 ACCURACY OF LOGISTIC REGRESSION.....	30
FIGURE 56 EXAMPLE OF MULTI CLASS.....	30
FIGURE 57 MODEL OF MULTI CLASS.....	30
FIGURE 58 DROPOUT LAYERS AND A REGULARIZES.....	31
FIGURE 59 ACCURACY OF MULTI CLASS.....	31
FIGURE 60 K-MEANS EXAMPLE	31
FIGURE 61 MODEL OF K-MEANS	32
FIGURE 62 2CLUSTERS	32
FIGURE 63 ACCURACY OF K-MEANS	32
FIGURE 64 LOGIN INTERFACE.....	33
FIGURE 65 DARK MODE	33
FIGURE 66 DASHBOARD CLIENT PER USAGE.....	34
FIGURE 67 DASHBOARD SATISFACTION OF INSURANCE COMPANIES	34
FIGURE 68 BONUS-MALUS PREDICTION INTERFACE	35
FIGURE 69 FRAUD DETECTION	35

INTRODUCTION

“A baby learns to crawl, walk and then run. We are in the crawling stage when it comes to applying machine learning.” ~Dave Waters

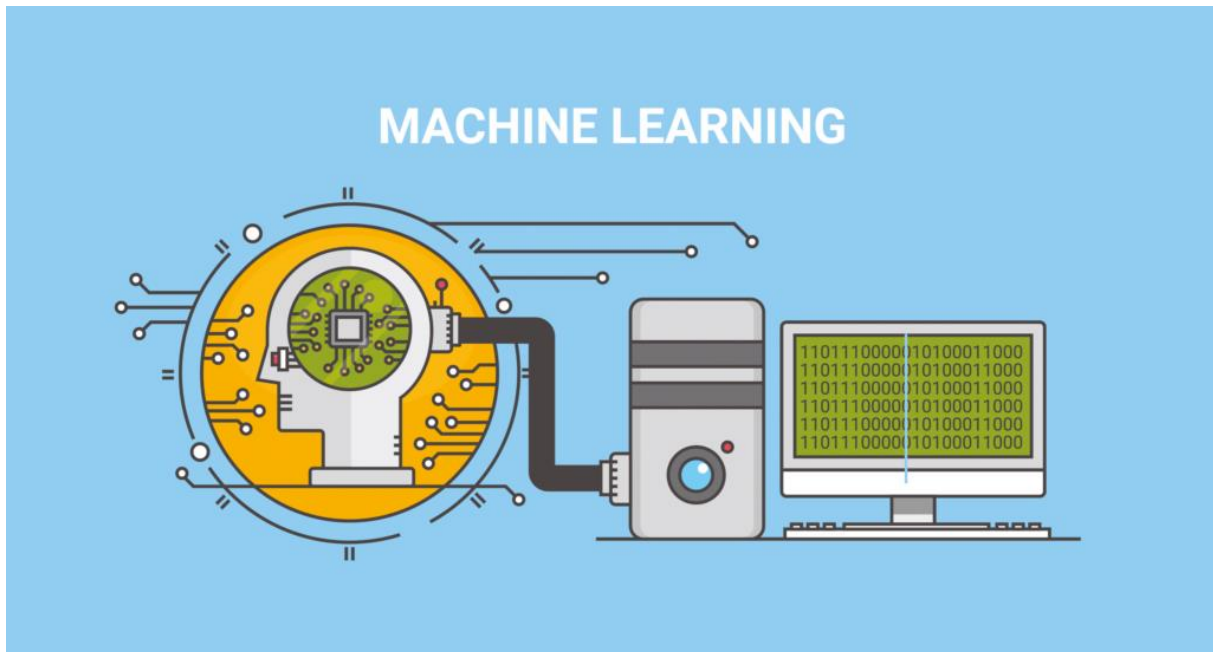


FIGURE 1 MACHIE LEARNING

Machine learning is becoming more and more essential in various industries, given its many applications such as predicting values and classifying data

For this end of year project, the CGA (Comité Général des assurances) tasked fourth year data science students with programming an online interface to assist its agents in decision making and providing them with easy to understand dashboards

1. CRISP DM

CRISP-DM stands for cross-industry process for data mining. The CRISP-DM methodology provides a structured approach to planning a data science project. It is a robust and well-proven methodology. It has also proven to be practically powerful, flexible as well as being useful when using analytics to solve thorny business issues. It is the golden thread than runs through almost every client engagement.

CRISP-DM breaks the process of data science into six major phases:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment



FIGURE 2 CRISP DM

The sequence of the phases is not strict and moving back and forth between different phases as it is always required. The arrows in the process diagram indicate the most important and frequent dependencies between phases. The outer circle symbolizes the cyclic nature of a data science project itself. A data science process continues after a solution has been deployed. The lessons learned during the process can trigger new, often more focused business questions.

Business understanding



FIGURE 3 LOGO CGA

1. CGA

The "Comité Général des Assurances" ensures the protection of the rights of policyholders and beneficiaries of insurance contracts and the soundness of the financial base of insurance companies and their ability to honor their commitments.

Within the missions assigned to it, the committee is in charge of:

- The supervision of insurance companies and professions related to the insurance sector and the monitoring of their activities.
- The study of legislative, regulatory and organizational issues relating to insurance and reinsurance operations and insurance companies submitted to it by the ministry of Finance.
- The study of technical and economic inquiries relating to the development of the insurance sector and its organization and the presentation of proposals to this effect to the ministry of Finance,
- And in general, to study and give its opinion on any other question falling within its attributions.

2. Business objectives

Understanding the end goal of the business is the first step to create a reliable prediction model, for the insurance field, these goals are:

- **Profit maximization**

From the insurer's point of view, the goal of an insurance company is to maximize profit.

- **Bonus Malus prediction**

One of the important aspect is the bonus-malus index (BM) which is given to a client when they start an insurance policy, most of the company's financial loss comes from attributing the default Index to a client that is prone to making accidents

- **Fraud detection**

Another cause of financial loss are fraudulent claims, which take considerable time to be reviewed.

3. Data Science objectives

After translating the previously mentioned goals to fit the technical aspect of the project, these are the data science goals:

- **Reducing Human Error**

Our application will manage to execute human tasks automatically with no errors through machine learning algorithms.

- **Data Collection**

Centralize the data from various insurance companies into one global database: Before this was implemented, data was scattered across various insurance companies, this made it easier for clients to transfer from one agency to another without their record carrying over, thus resetting their bonus malus index.

- **Data Update**

Update client information using our stored data simultaneously with every occurring change.

- **Dashboarding**

Create a user interface providing the agents and clients with various statistics concerning insurance policies and claims.

Data analysis

1. Internal data

After identifying the goals of the project, the next step is to acquire the necessary data. This task has already been done by the CGA, which has collected and separated the data into various tables, the database provided consists of

- **ClassesBonusMalus**

The main table for predicting the BM index, it contains said index for each client, along with various info such as the car id and policy type

- **Police**

Contains every policy that is registered by the CGA, its type, nature, the owner of said policy, etc...

- **Sinistres**

Accidents processed by the CGA are stored here, the tables plays an important role in identifying certain types of fraud

- **Véhicule**

Insured automobiles are stored here, along with their corresponding usage, brand, fuel type and date of registration

- **Assure & Souscripteur**

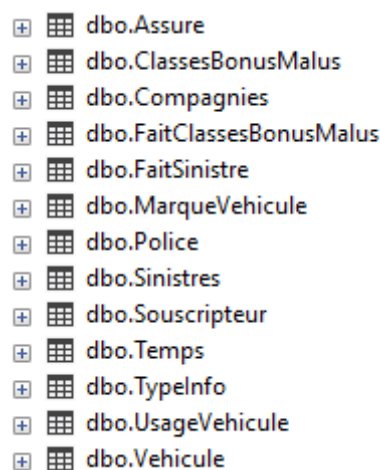
Id of the insured and ID of the underwriter exist in these tables respectively

- **Compagnies & MarqueVehicule & UsageVehicule**

These tables contain a literal description of the corresponding ID value

- **UsageVehicule**

Literal description for abbreviations used in the other tables





























		dbo.Assure
		dbo.ClassesBonusMalus
		dbo.Compagnies
		dbo.FaitClassesBonusMalus
		dbo.FaitSinistre
		dbo.MarqueVehicule
		dbo.Police
		dbo.Sinistres
		dbo.Souscripteur
		dbo.Temps
		dbo.TypeInfo
		dbo.UsageVehicule
		dbo.Vehicule

FIGURE 4 CGA TABLES

2. external data

a) web scrapping

➤ Selenium



FIGURE 5 SELENIUM LOGO

As for our external data, we managed to gather some information through web scrapping using selenium, from a website called “lemeilleur.tn”. The information gathered reflects people’s reviews about the best automobile insurance company in Tunisia.

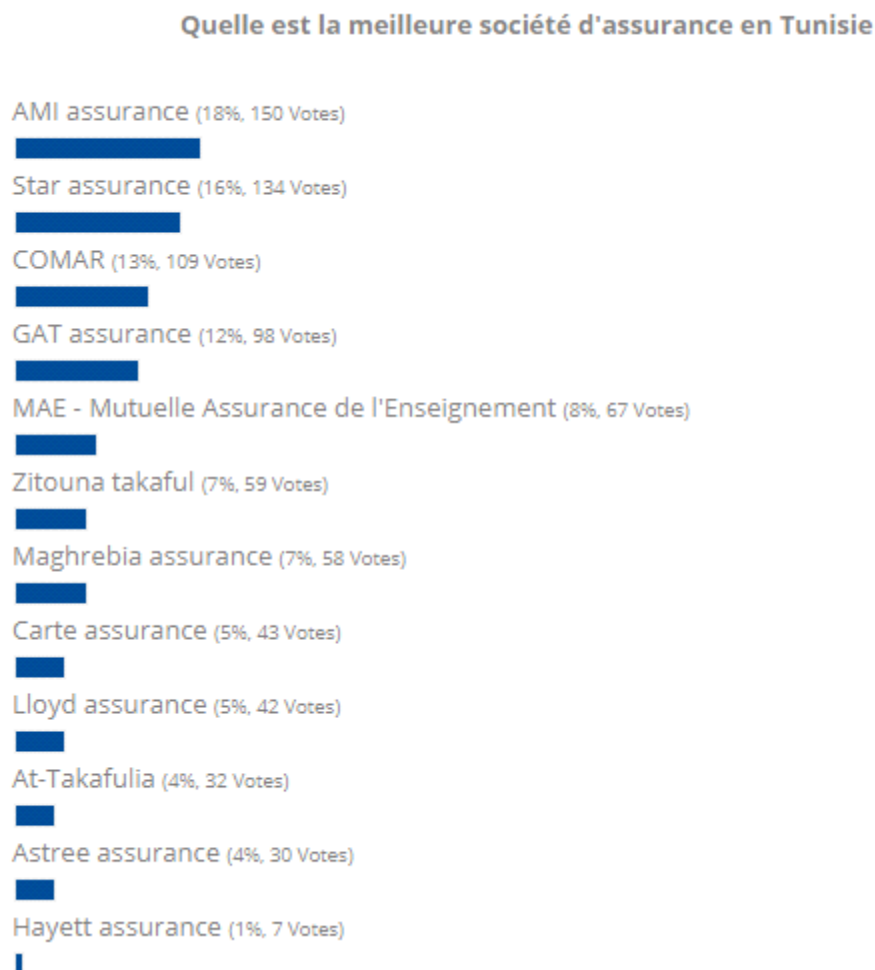


FIGURE 6 REVIEWS ABOUT THE BEST INSURANCE IN TUNISIA

➤ ParseHub



FIGURE 7 PARSE HUB LOGO

We succeed to gather some information through web scrapping using parsehub, from a website called “ghorbel-opinion.tn” to extract data from multiple pages as shown here

Les avis les plus récents		
ASSUREUR	NOTE SUR 5	AVIS
	4/5	Avis rédigé le 2020-09-17 12:08:00 par opinion : bon expérience (...) Lire la suite
	4/5	Avis rédigé le 2017-04-25 11:56:00 par yasser : TRÈS PROFESSIONEL (...) Lire la suite
	4/5	Avis rédigé le 2017-04-25 11:54:00 par yasser : bon service (...) Lire la suite
	3/5	Avis rédigé le 2017-04-25 11:53:00 par foufou : pas mal (...) Lire la suite
	3/5	Avis rédigé le 2017-04-25 11:44:00 par foufou : service moyenne (...) Lire la suite
	3/5	Avis rédigé le 2017-04-24 14:03:00 par foufou : meilleur prix sur le marché (...) Lire la suite

FIGURE 8 RATING THE CAR INSURANCE

b) Forms

A public survey has been launched to collect extra information from citizens and their opinions about their insurance companies.

The data collected, however, doesn't have the same format or size as the internal data which why we couldn't merge it with the internal databases and won't interfere in building our predictive models but it will serve as a mean of dashboarding and providing useful statistics as shown in the pie charts for the decision making

Vehicule usage
31 réponses

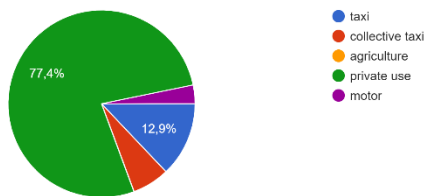


FIGURE 10 VEHICLE USAGE

Do you currently have Car insurance coverage?
31 réponses

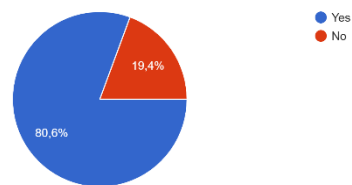


FIGURE 9 CAR INSURANCE COVERAGE

Have you switched your insurer in the past 12 months?
31 réponses

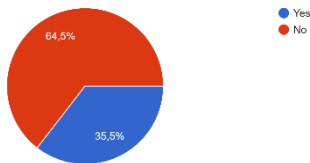


FIGURE 11 SWITCH INSURER

What was the main reason for switching your car insurance company?
31 réponses

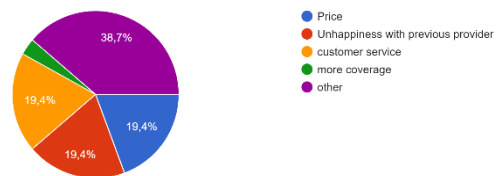


FIGURE 12 REASON OF SWITCHING COMPANY

Data preparation

1. Internal data

Now that the data is clearly understood, the next step is identifying which tables are to be used, in correspondence with the two main goals (BM prediction and fraud detection), a Data warehouse chart is created

a) DATA WAREHOUSE

The Data warehouse consists of two data marts

- **FaitClasseBonusMalus**

Will be used in the prediction models for the BM index, it is composed of three tables: ClasseBonusMalus, Police and Vehicule.

- **FaitSinistre**

Plays a major role in detecting fraudulent activities. Sinistres, Police and Vehicule are fused to create this fact table

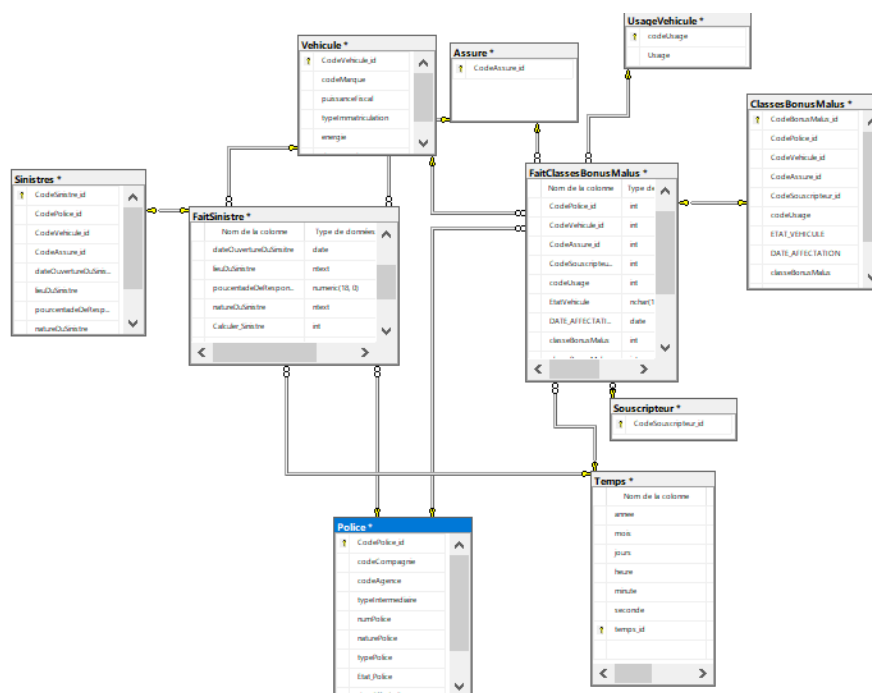


FIGURE 13 DATA WAREHOUSE

b) FUSION

Along with the CGA provided database, last year's database was provided as complimentary data, fusing these two guarantees a bigger data pool which can increase the potency of the machine learning models.

For this task, SQL Server Integration Services (SSIS) is used to join the police, vehicle and Sinister tables saved in SQL Server Management Studio (SSMS) and in CSV files.

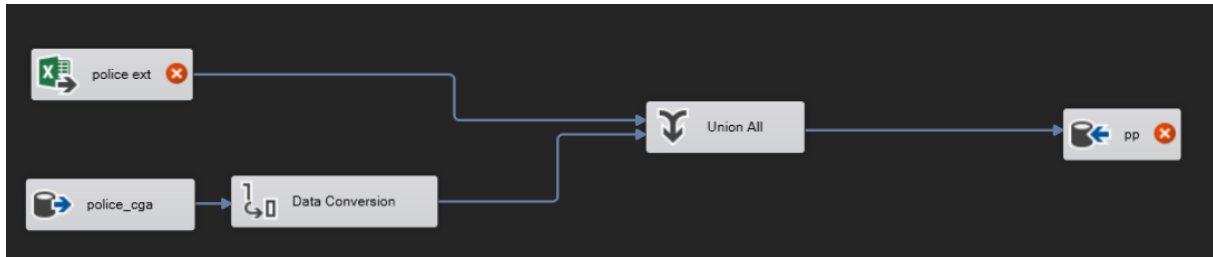


FIGURE 14 MERGE POLICE.CSV AND POLICE_CGA INTO PP

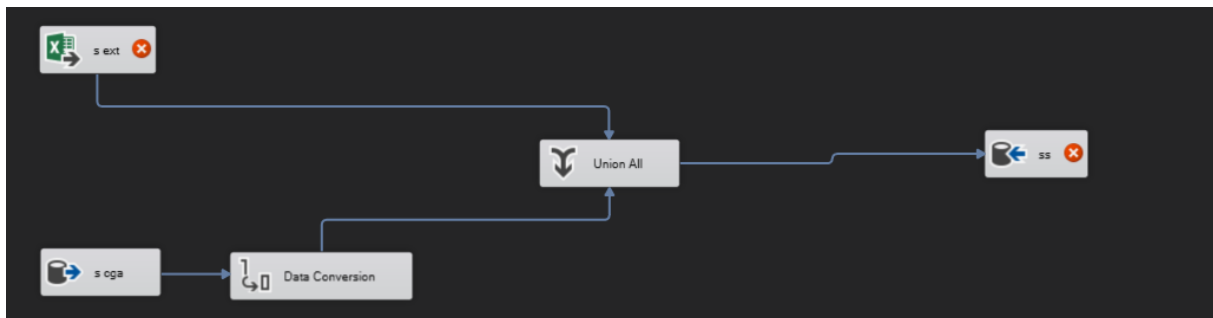


FIGURE 15 MERGE SINISTRE.CSV AND SINISTRE_CGA INTO SS

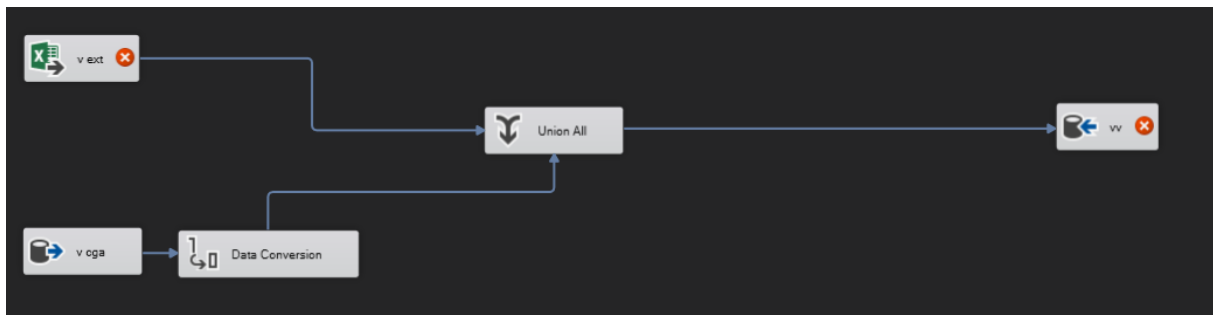


FIGURE 16 MERGE VEHICLE.CSV AND VEHICLE_CGA INTO VV

- Note that the “union all” feature is also used to get rid of all but the columns that exist in both joined tables

The common column that is used to join the tables is the ID column (CodePolice_id, CodeSinistre_id, CodeVehicule_id). “ss”, “vv”, “pp” are the resulting tables from this joint.

To create “**FaitClasseBonusMalus**”, the tables are merged in this order:

- ✚ “ClassesBonusMalus” + “Vehicule” using the column ‘CodeVehicule_id’ as the common column creating “BMV”

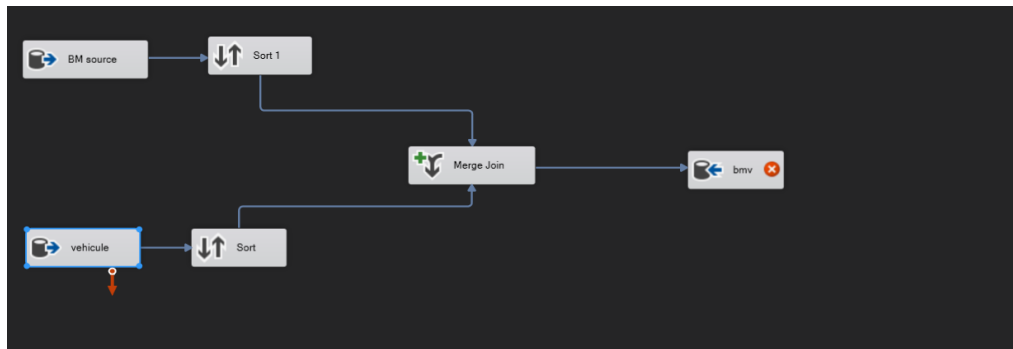


FIGURE 17 CREATING BMV

- ✚ “BMV” + “Police” using ‘CodePolice_id’ as the common column creating “fait_BM”

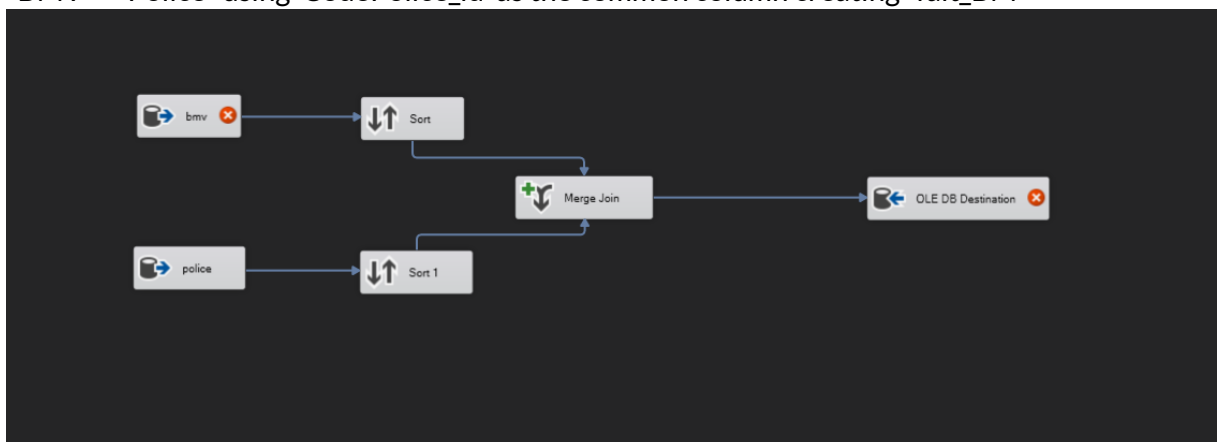


FIGURE 18 CREATING FAIT_BM

To create “FaitSinistre”,the tables are merged in this order:

- ✚ “ss”+”vv” ” using the column ‘CodeVehicule_id’ as the common column, creating “ssvv”

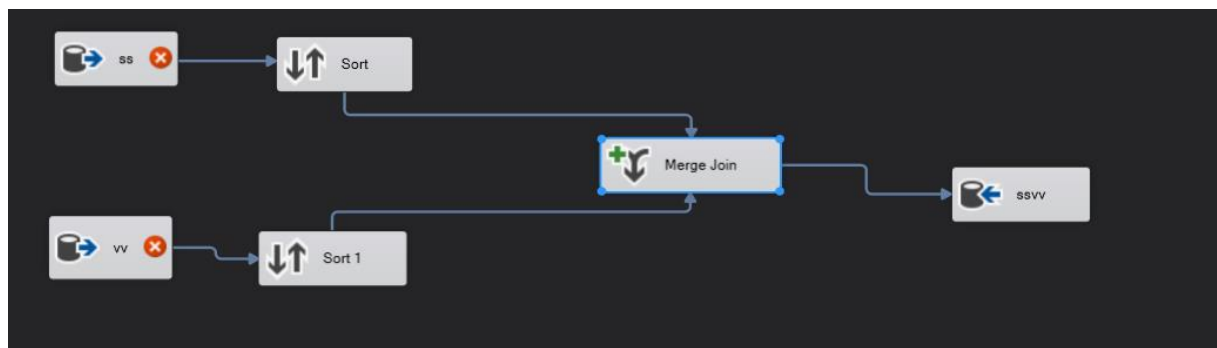


FIGURE 19 CREATING SSVV

- ✚ “ssvv”+”pp” using ‘CodePolice_id’ as the common column, Creating “fait_sinistre”

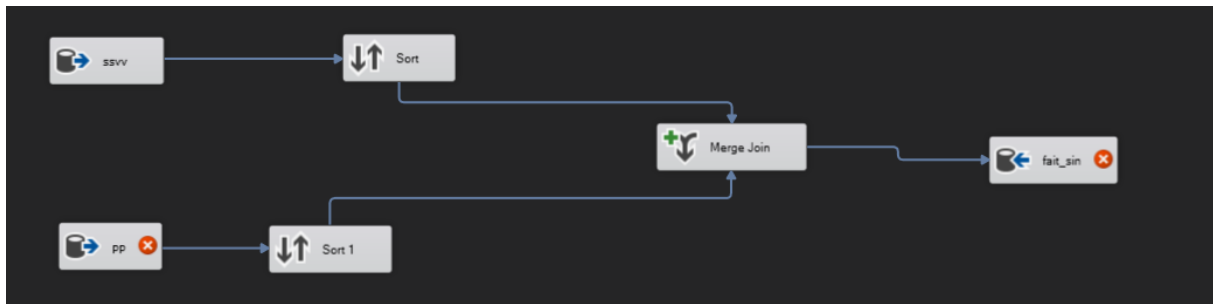


FIGURE 20 CREATING FAIT_SINISTRE

- these are the dimensions for the resulting fact tables:

fait_BM : 1307361 rows × 44 columns

fait_sinistre: 562802 rows × 29 columns

c) Data cleaning

During data imputation, the most prominent action is deleting empty and quasi-empty columns.

- For “fait_BM” these are

```
#F1_X : 9415
#numChassis : 9415
#numImmatriculation : 9415
#dateDerniereVisite : 4
#dateAjout : 9415
#etatVehicule : 9415
#dateMiseEpave : 0
#DATE_RETRAIT : 0
#dateMiseCirculation : 0
#dateMiseAJourVehicule : 4
#code_Courtier_CGA : 0
#dateExpirationPolice : 0
#verouillageModifPolice : 0
#Etat_Police : 1307361
#RESILIATION_ECHEANCE : 0
#DATE_RESILIATION : 0
#dateSuspension : 0
#dateRemiseEnVigueure : 0
#F1_Y : 0
fait_BM = fait_BM.drop(['index', 'F1_Y', 'F1_X', 'CodeSouscripteur_id', 'classeBonusMalusCompagnie', 'numChassis',
                        'numImmatriculation', 'dateDerniereVisite', 'dateAjout', 'etatVehicule',
                        'dateMiseEpave', 'DATE_RETRAIT', 'dateMiseCirculation', 'dateMiseAJourVehicule',
                        'code_Courtier_CGA', 'numPolice', 'dateEffetPolice', 'dateEcheancePolice',
                        'dateExpirationPolice', 'verouillageModifPolice', 'RESILIATION_ECHEANCE', 'DATE_RESILIATION',
                        'DateSuspension', 'dateRemiseEnVigueure'], axis=1)
```

FIGURE 21 DROP ALL THE UNNECESSARY DATA FROM BM_FAIT

- For “fait_sinistre”:

```
#etatVehicule is almost null
#"date ajout" we already have "data dinsertion"
#"numeroDuSinistre" inutile on a sinistre ID
#"typeImmatriculationVehiculeAdverse" we have "numeroImmatriculationVehiculeAdverse ""
#"numPolice" we "id police"

fait_Sinistre = fait_Sinistre.drop(['index', 'identificationTiers',
                                    'numeroDePoliceCompagnieAdverse', 'typeImmatriculationVehiculeAdverse',
                                    'dateAjout', 'numPolice'], axis=1)
```

FIGURE 22 DROP ALL THE UNNECESSARY DATA FROM SINISTRE_FAIT

Next step, filling missing values.

Most missing values are replaced proportionally with the most common data in its respective columns.

```
fait_BM['energie']=fait_BM['energie'].replace(['0'], 'unknown')
```

FIGURE 23 FILL MISSING VALUES IN ENERGY BY UNKNOWN

```
fait_Sinistre["typeIntermediaire"]=fait_Sinistre["typeIntermediaire"].fillna(1,limit=250000)
```

```
fait_Sinistre["typeIntermediaire"]=fait_Sinistre["typeIntermediaire"].fillna(2,limit=65000)
```

```
fait_Sinistre["typeIntermediaire"]=fait_Sinistre["typeIntermediaire"].fillna(3)
```

FIGURE 24 FILL MISSING VALUES IN TYPE INTERMEDIAIRE BY 1 OR 2 OR 3

```
query = """SELECT count(CodeVehicule_id) as count_CodeVehicule_id,CodeAssure_id
FROM fait_Sinistre where typePolice is NULL
group by CodeAssure_id having count(CodeVehicule_id)<4 """
results = cursor.execute(query).fetchall()
```

```
dataframe = pd.read_sql(query, con = conn)
dataframe
```

```
condition=(fait_Sinistre["CodeAssure_id"].isin(dataframe["CodeAssure_id"]))
fait_Sinistre.loc[condition,'typePolice']='I'
```

FIGURE 25 DELETE ALL CLIENTS WHO HAVE A POLICY EQUAL TO NULL

Adding a new column to “fait_BM” named “num_accidents” , which contains the sum of accidents. For each insured, given that these accidents are either totally or partially their fault.

```
query = """SELECT CodeAssure_id,count(Codesinistre_id) as num_accidents FROM fait_Sinistre_fraud where
pourcentageDeResponsabilite>50 And Fraud=0 group by CodeAssure_id"""
results = cursor.execute(query).fetchall()
num_accidents = pd.read_sql(query, con = conn)
num_accidents.CodeAssure_id = num_accidents.CodeAssure_id.astype(int)
num_accidents
```

```
fait_BM = pd.merge(fait_BM,
                  num_accidents,
                  on='CodeAssure_id',
                  how='left')
```

```
fait_BM['num_accidents'] = fait_BM['num_accidents'].fillna(0)
```

FIGURE 26 ADDING A NEW COLUMN NAMED NUM ACCIDENT

To ensure an accurate prediction of the BM index, we must determine the values which correspond to non-fraudulent clients, for this task, we added a new column in both fact tables, named “fraud”.

These types of frauds are detected as follows:

 In “fait_BM”:

1) policy type is "individual" but the client has more than 3 vehicles

```
query = """SELECT count(CodeVehicule_id) as count, CodeAssure_id
FROM fait_BM where typePolice='I'
group by CodeAssure_id having count(CodeVehicule_id) >=3 """
results = cursor.execute(query).fetchall()
```

2) policy type is "fleet " but the client has less than 3 cars

```
query = """SELECT count(CodeVehicule_id) as count, CodeAssure_id
FROM fait_BM where typePolice='F'
group by CodeAssure_id having count(CodeVehicule_id) <3 """
results = cursor.execute(query).fetchall()
```

3) usage code=1 (private use) so policy type must not be "fleet"

```
query = "SELECT CodeVehicule_id FROM fait_BM where codeUsage=1 and typePolice='F' "
results = cursor.execute(query).fetchall()
```

4) usage code=14 or 15 (rent or travel agency) so policy type must not be "individual"

```
query = "SELECT CodeVehicule_id FROM fait_BM where codeUsage in (15,14) and typePolice='I' "
results = cursor.execute(query).fetchall()
```

5) client has more than 4 vehicles so policy type must not be "individual"

```
query = """SELECT count (CodeVehicule_id) as count_CodeVehicule_id, CodeAssure_id
FROM fait_BM where typePolice='I'
group by CodeAssure_id having (count (CodeVehicule_id)) >=3 """
results = cursor.execute(query).fetchall()
```

6) multi insurance fraud

```
query = """SELECT CodeVehicule_id ,count (codeCompagnie) as count_codeCompagnie
FROM fait_BM where typePolice='I' group by CodeVehicule_id having (count (codeCompagnie)) >=3 """
results = cursor.execute(query).fetchall()
```

7) wrecked vehicle still has an operational policy

```
query = "SELECT CodeVehicule_id FROM fait_BM where ETAT_VEHICULE='R' "
```

8) unusual date

```
query = "SELECT DATE_AFFECTATION FROM [fait_BM] where DATE_AFFECTATION>'2021-03-16 00:00:00'"
results = cursor.execute(query).fetchall()
```

In "fait_sinistre":

1) abnormal number of accidents

```
condition = fait_Sinistre.groupby('CodeAssure_id').CodeAssure_id.transform('size') >=15
```

```
fait_Sinistre.loc[condition, 'Fraud'] = 1
```

2) planned accidents

```
query = """SELECT count (numeroImmatriculationVehiculeAdverse) as count, CodeVehicule_id
from fait_sinistre group by CodeVehicule_id having count (numeroImmatriculationVehiculeAdverse) >3"""
results = cursor.execute(query).fetchall()
```

After identifying the Bonus-malus indexes that are affected by fraudulent activities, we isolate them, thus creating a “clean” fact table.

```
data=fait_BM[fait_BM["Fraud"]!=0]
```

FIGURE 27 CLEANED DATA

- Its dimension being 1082005 rows × 23 columns

Another look at the data shows that certain BM indexes are illogical (drivers who made no accidents having an index above 8 and vice versa)

people who made	12 accidents and have BM index of 1 ,their total is	1
people who made	12 accidents and have BM index of 2 ,their total is	10
people who made	12 accidents and have BM index of 3 ,their total is	4
people who made	12 accidents and have BM index of 4 ,their total is	3
people who made	12 accidents and have BM index of 5 ,their total is	2
people who made	12 accidents and have BM index of 6 ,their total is	1
people who made	12 accidents and have BM index of 7 ,their total is	1
people who made	12 accidents and have BM index of 8 ,their total is	0
people who made	12 accidents and have BM index of 9 ,their total is	0
people who made	12 accidents and have BM index of 10 ,their total is	0
people who made	12 accidents and have BM index of 11 ,their total is	0

FIGURE 28 EXTRACTING THE CLIENTS WHO HAD MADE ACCIDENTS

These values were deleted:

- ✚ BM=1 and num_accidents > 0
- ✚ BM=11 and num_accidents < 2
- ✚ 8<BM<11 and num_accident < 1
- ✚ 0<BM<8 and num_accident > 2
- ✚ BM=2 and num_accident > 1

- New dimension 1025544 rows × 23 columns

The cleaning phase is reaching its final steps , for predictions , IDs serve no value , so they are dropped.

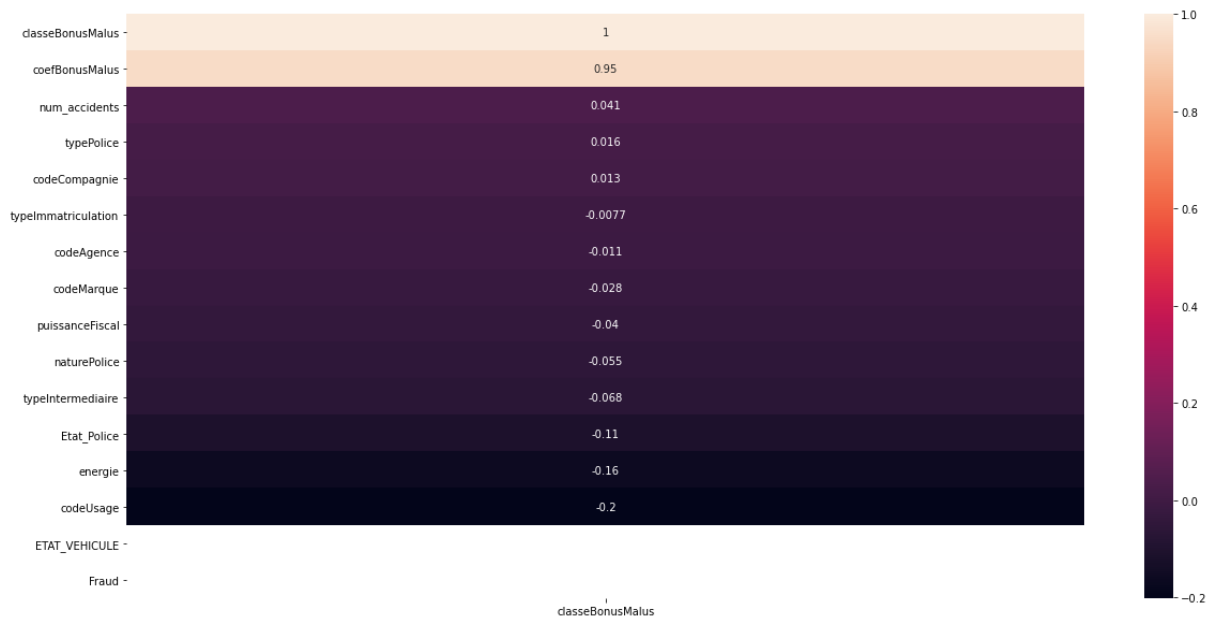


FIGURE 29 CORRELATION OF PEARSON

This correlation map shows that the columns “typeImmatriculation”, “Fraud” and “ETAT_VEHICULE” have little to no effect on the target data, these columns are dropped.

The final BM fact table contains these columns:

-codeUsage	-energie	-Etat_Police
-naturePolice	-puissanceFiscal	-codeMarque
-codeAgence	-codeCompagnie	-typePolice
-num_accidents	-coefBonusMalus	-classeBonusMalus

d) Encoding

In order to apply our models, most of which work only on numerical data, we needed to encode our columns using different encoding methods.

```
data=MultiColumnLabelEncoder(columns = ['energie', 'ETAT_VEHICULE', 'typeImmatriculation', 'naturePolice', 'typePolice', 'Etat_Police']).fit_transform(data)
```

FIGURE 30 ENCODING USING LABEL ENCODER

```
pd.get_dummies(data, columns=["naturePolice"])
```

FIGURE 31 ENCODING USING GET DUMMIES

2. External data

Opinions concerning insurance companies were scrapped from this web page

<https://www.ghorbel-opinion.tn/frontendAssurance.php>

We convert the data from string to number in this phase and we delete the Nan value as shown below

```
[ ] #transform the column of rating to float
for i in range(34):
    if d.iloc[i]['Assureur_rating_name'] == '1/5':
        d['Assureur_rating_name'][i]=1.0/5
    elif d.iloc[i]['Assureur_rating_name']=="2/5":
        d['Assureur_rating_name'][i]=2.0/5
    elif d.iloc[i]['Assureur_rating_name']=="3/5":
        d['Assureur_rating_name'][i]=3.0/5
    elif d.iloc[i]['Assureur_rating_name']=="4/5":
        d['Assureur_rating_name'][i]=4.0/5
    elif d.iloc[i]['Assureur_rating_name']=="5/5":
        d['Assureur_rating_name'][i]=5.0/5
```

FIGURE 32 TRANSFORM THE COLUMN OF RATING TO FLOAT

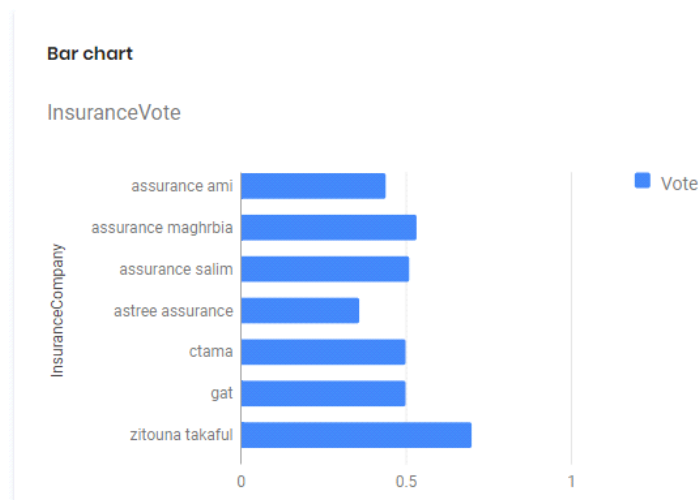


FIGURE 33 INSURANCE VOTE

Data modelling

I. Internal data

Once our internal data is cleaned and prepared. It is time for the modelling phase. We noticed that our data is a labeled discrete with the target class being the Bonus Malus index, ranging from 1 to 11, this necessitates classification supervised learning models, these are :

- ✚ K nearest neighbor (KNN)
- ✚ Naïve Bayes
- ✚ Multi-Layer Perceptron (MLP)
- ✚ Support vector machine (SVM)
- ✚ Decision tree + random forest algorithm
- ✚ Logistic regression

1. Naïve Bayes

Naive Bayes is a probabilistic technique for constructing classifiers. The characteristic assumption of the naive Bayes classifier is to consider that the value of a particular feature is independent of the value of any other feature, given the class variable.

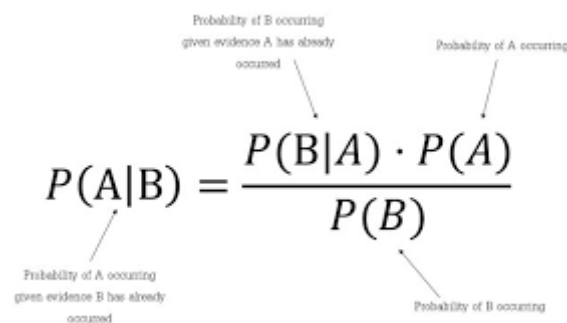

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

FIGURE 34 EXEMPLE OF NAIVE BAYES

a) Building the model

```
from sklearn.naive_bayes import MultinomialNB, GaussianNB, BernoulliNB
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score

nb = {'gaussian': GaussianNB(),
      'bernoulli': BernoulliNB(),
      'multinomial': MultinomialNB()}
```

FIGURE 35 MODEL NB

b) Evaluation

```
'gaussian': 0.6439376672131247,
'bernoulli': 0.29104316612528724,
'multinomial': 0.17779348760108965}
```

FIGURE 36 ACCURACY OF THE MODEL

- This Naïve Bayes model is 69% accurate.

2. KNN

K Nearest Neighbor is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classify a data point based on how its neighbors are classified.

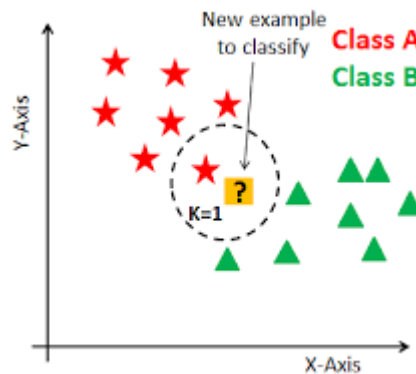


FIGURE 37 EXAMPLE OF KNN

a) Building the model

```
Entrée [592]: from sklearn.neighbors import KNeighborsClassifier  
classifier = KNeighborsClassifier(n_neighbors=3)  
classifier.fit(X_train, y_train)
```

```
Out[592]: KNeighborsClassifier(n_neighbors=3)
```

```
Entrée [593]: y_pred = classifier.predict(X_test)
```

FIGURE 38 MODEL KNN

b) Evaluation

```
Entrée [595]: from sklearn import metrics  
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.941358104945655
```

FIGURE 39 ACCURACY OF THE MODEL


```
Entrée [594]: from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[78195 3025  0  0  0  0  0  0  0  0  0]
 [ 2093 38475 2374  0  0  0  0  0  0  0  0]
 [  4  2281 54814 3095  0  0  0  0  0  0  0]
 [  0  0  3076 46755 991  0  0  0  0  0  0]
 [  0  0  4  663 24481 2  0  0  0  0  0]
 [  0  0  0  0  15 13364 1  1  0  0  0]
 [  0  0  0  0  6  4 2891 287  0  0  0]
 [  0  0  0  0  2  0 111 29725  0  0  0]
 [  0  0  0  0  0  0  0  4 744  0  0]
 [  0  0  0  0  0  0  0  1 1 165  0]
 [  0  0  0  0  0  0  0  0  0  1 13]]
```

	precision	recall	f1-score	support
1	0.97	0.96	0.97	81220
2	0.88	0.90	0.89	42942
3	0.91	0.91	0.91	60194
4	0.93	0.92	0.92	50822
5	0.96	0.97	0.97	25150
6	1.00	1.00	1.00	13381
7	0.96	0.91	0.93	3188
8	0.99	1.00	0.99	29838
9	1.00	0.99	1.00	748
10	0.99	0.99	0.99	167
11	1.00	0.93	0.96	14
accuracy			0.94	307664
macro avg	0.96	0.95	0.96	307664
weighted avg	0.94	0.94	0.94	307664

FIGURE 40 ACCURACY FOR EACH CLASS

➤ This KNN model is 94% accurate.

3. Multi Layer Perceptron (MLP)

The multilayer perceptron is a deep learning algorithm that is used mainly with tabular dataset. It also can be used on image data, text data as well as time series data due to its great flexibility.

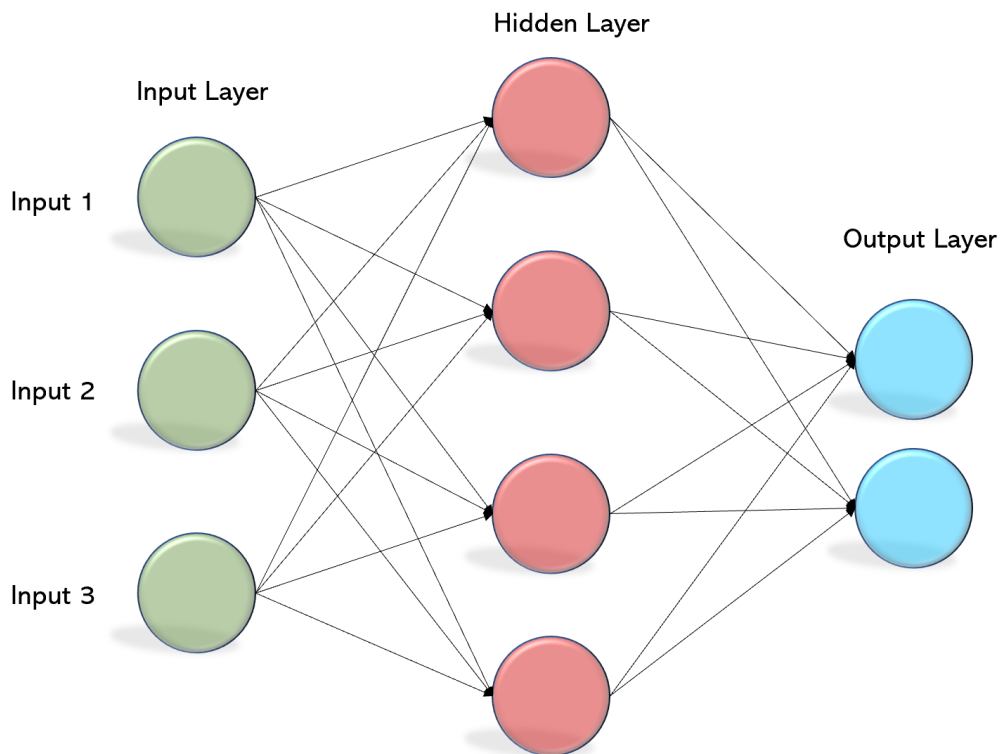


FIGURE 41 EXAMPLE OF MLP

a) Building the model :

```
from sklearn.neural_network import MLPClassifier  
  
clf = MLPClassifier(random_state=1, max_iter=300).fit(X_train,y_train)
```

FIGURE 42 MODEL OF MLP

b) Evaluation

```
clf.score(X_test, y_test)  
: 0.9265317465347693
```

FIGURE 43 ACCURACY OF MLP

- This MLP model is 93% accurate.

4. Decision Tree

Decision Trees is a rule-based model, it used for classification and generally used for decision making and in forecasting future outcomes and assigning probabilities to those outcomes.

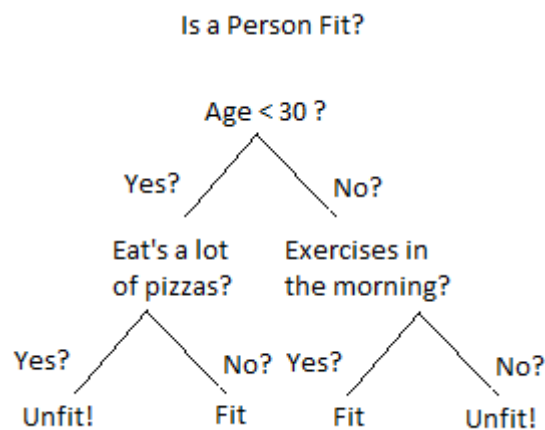


FIGURE 44 EXAMPLE OF DECISION TREE

a) Building the model

```
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=3)  
from sklearn.tree import DecisionTreeClassifier  
dt = DecisionTreeClassifier(random_state=0)  
  
dt.fit(X_train, y_train)
```

```
DecisionTreeClassifier(random_state=0)
```

FIGURE 45 MODEL OF DECISION TREE

b) Evaluation

```
print('Le test score est :', dt.score(X_test, y_test))
```

```
Le test score est : 0.9495293567008165
```

FIGURE 46 ACCURACY OF THE MODEL

5. Random Forest

Random forests are an improvement to decision trees, they operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees.

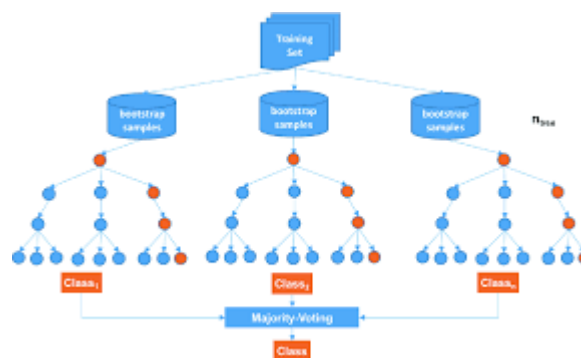


FIGURE 47 EXAMPLE OF RANDOM FOREST

a) Building the model

```

: from sklearn.ensemble import RandomForestClassifier
  rfc = RandomForestClassifier(random_state=0)
  rfc.fit(X_train, y_train)

```

```
RandomForestClassifier(random_state=0)
```

FIGURE 48 MODEL OF RANDOM FOREST

b) Evaluation

```

: # Predict the Test set results
  y_pred = rfc.predict(X_test)
  # Check accuracy score
  from sklearn.metrics import accuracy_score
  print('Model accuracy score with 10 decision-trees : {0:0.4f}'.format(accuracy_score(y_test, y_pred)))

```

```
Model accuracy score with 10 decision-trees : 0.9492
```

FIGURE 49 ACCURACY OF RANDOM FOREST

6. SVM

The SVM is essentially used for all kinds of recognition such as face detection or handwriting recognition and it can be used for both classification and regression. In our case we used the SVM for classification. The difficult part of SVM is choosing the right Kernel.

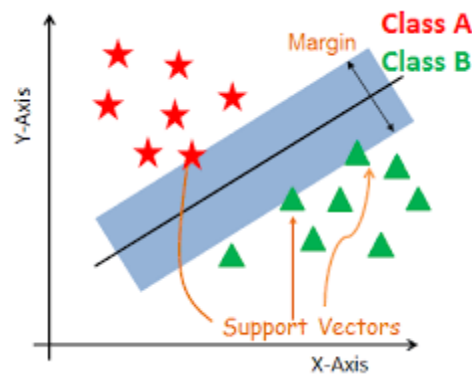


FIGURE 50 EXAMPLE OF SVM

a) Building the model

```
from sklearn.svm import SVC
```

```
svm = SVC()
svm.fit(X_train, y_train)
```

```
SVC()
```

FIGURE 51 MODEL OF SVM

b) Evaluation

```
from sklearn.metrics import classification_report
y_pred_svc_1 = svm.predict(X_test)
print(classification_report(y_pred_svc_1, y_test))
```

	precision	recall	f1-score	support
1	0.97	0.93	0.95	84913
2	0.62	0.90	0.73	29503
3	0.90	0.76	0.82	71583
4	0.80	0.88	0.84	45802
5	0.91	0.73	0.81	31298
6	0.60	0.80	0.69	9956
7	0.71	0.71	0.71	3187
8	1.00	0.98	0.99	30498
9	0.99	1.00	1.00	745
10	0.99	0.99	0.99	167
11	0.86	1.00	0.92	12
accuracy			0.86	307664
macro avg	0.85	0.88	0.86	307664
weighted avg	0.88	0.86	0.86	307664

FIGURE 52 ACCURACY OF SVM

- We tested the model with different kernels and the one with the best results was the polynomial kernel with 0.94 accuracy.

7. Logistic regression

The logistic regression as opposed to linear regression is a classification algorithm used to assign observations to a discrete set of classes.

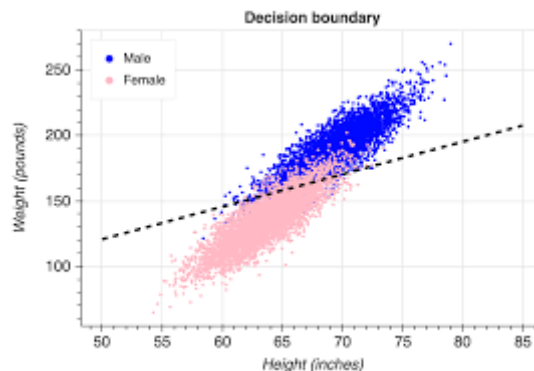


FIGURE 53 EXAMPLE OF LOGISTIC REGRESSION

a) Building the model

```
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
dlf = LogisticRegression(random_state=0).fit(X_train, y_train)
```

FIGURE 54 MODEL OF LOGISITIC REGRESSION

b) Evaluation

```
: dlf.score(X_test, y_test)
```

```
0.6616341203390712
```

FIGURE 55 ACCURACY OF THE MODEL

- The evaluation of the model gave us an accuracy of 66%.

8. Multi class classification

Keras Multi Class Classification is a deep learning algorithm that provide a classification model for more than two categories like we have in the bonus malus index using neural network to learn from the train sample and predict using activation function like relu and softmax.

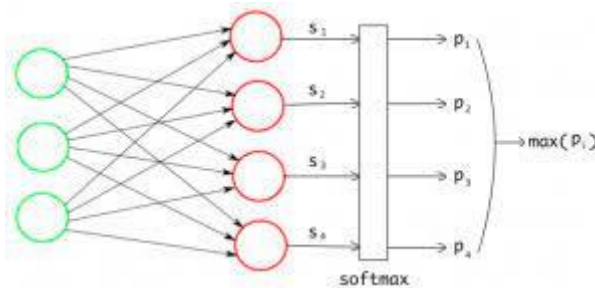


FIGURE 56 EXAMPLE OF MULTI CLASS

a) Building the model

```
# create model
def classification_small_model():
    model = Sequential()
    model.add(Dense(15, activation='relu', input_shape=(20,),
        kernel_regularizer=regularizers.l1_l2(l1=1e-5, l2=1e-4),
        bias_regularizer=regularizers.l2(1e-4),
        activity_regularizer=regularizers.l2(1e-5)))

    model.add(Dropout(0.5))

    model.add(Dense(15, activation='relu',
        kernel_regularizer=regularizers.l1_l2(l1=1e-5, l2=1e-4),
        bias_regularizer=regularizers.l2(1e-4),
        activity_regularizer=regularizers.l2(1e-5)))

    model.add(Dropout(0.5))

    model.add(Dense(output, activation='softmax'))
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

FIGURE 57 MODEL OF MULTI CLASS

b) Evaluation

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 15)	315
dropout_3 (Dropout)	(None, 15)	0
dense_6 (Dense)	(None, 15)	240
dropout_4 (Dropout)	(None, 15)	0
dense_7 (Dense)	(None, 12)	192
Total params: 747		
Trainable params: 747		
Non-trainable params: 0		

FIGURE 58 DROPOUT LAYERS AND A REGULARIZES

In this graph we summarize the accuracy and the loss function of our model

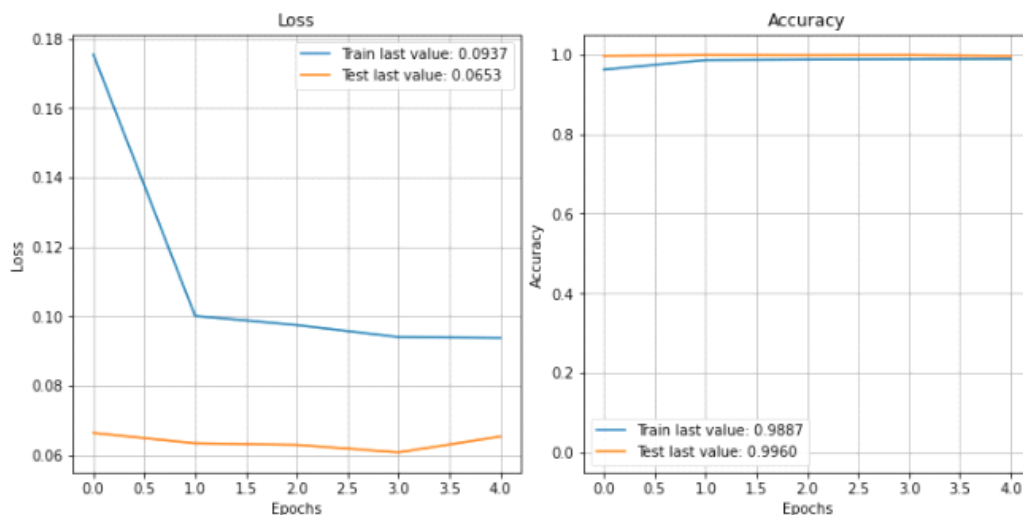


FIGURE 59 ACCURACY OF THE MODEL

II. External data

1. K-means

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. *k*-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the

more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances.

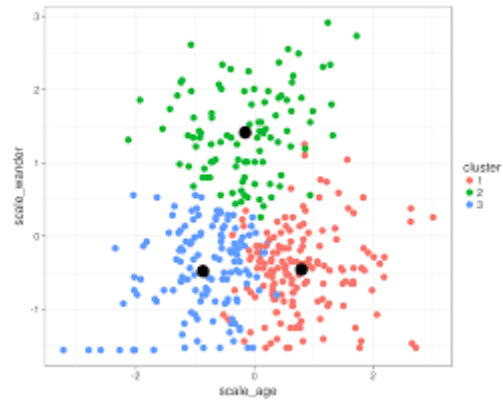


FIGURE 60 EXAMPLE OF K-MEANS

a) Building the model

We used an unsupervised model `k_means` to make the data into two clusters:

```
[24] from sklearn.cluster import KMeans
      k_means = KMeans(init = "k-means++", n_clusters = 2, n_init = 100)
```

FIGURE 61 2 CLUSTERS

```
[28] k_means.fit(X)
```

FIGURE 62 MODEL OF K_MEANS

b) Evaluation

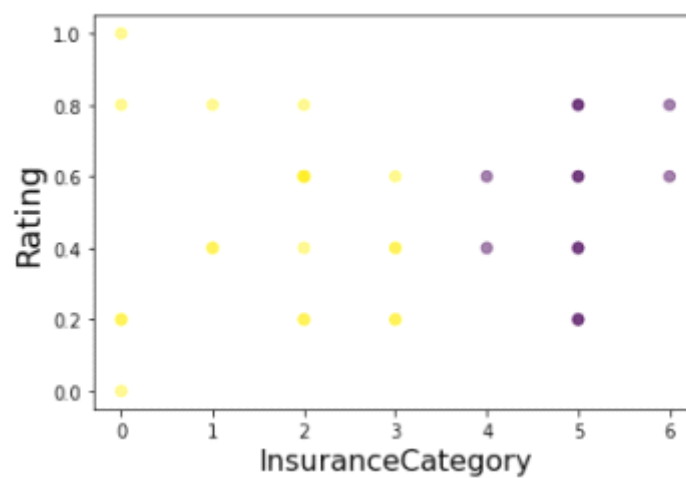


FIGURE 63 EVALUATION

Deployment

Deploying the models is the last step in the project, and arguably the most important one, it's about providing the user with an easy to use interface through which they can access the various features, such as viewing statistics, investigate frauds and predicting the bonus malus index, the interface is created with FLASK and runs python script.

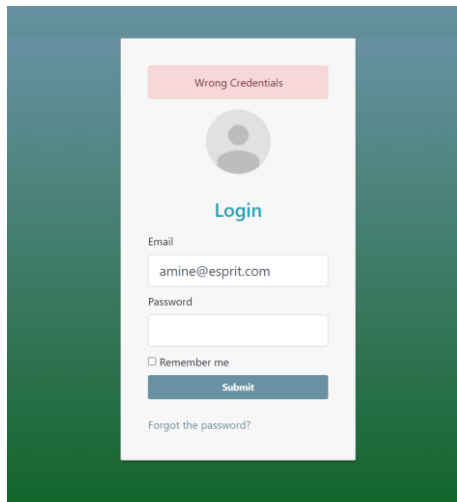


FIGURE 64 LOGIN INTERFACE

As shown here, we offer customizable user interfaces with the option of using dark or light mode.

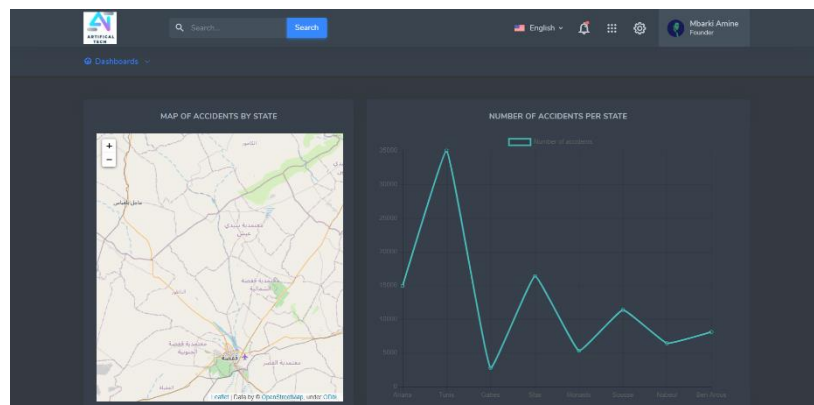


FIGURE 65 DARK MODE

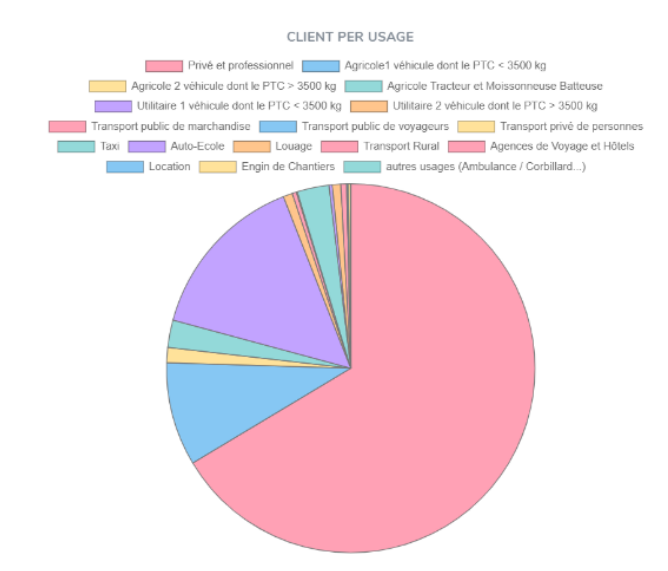


FIGURE 66 DASHBOARDS : CLIENT PER USAGE

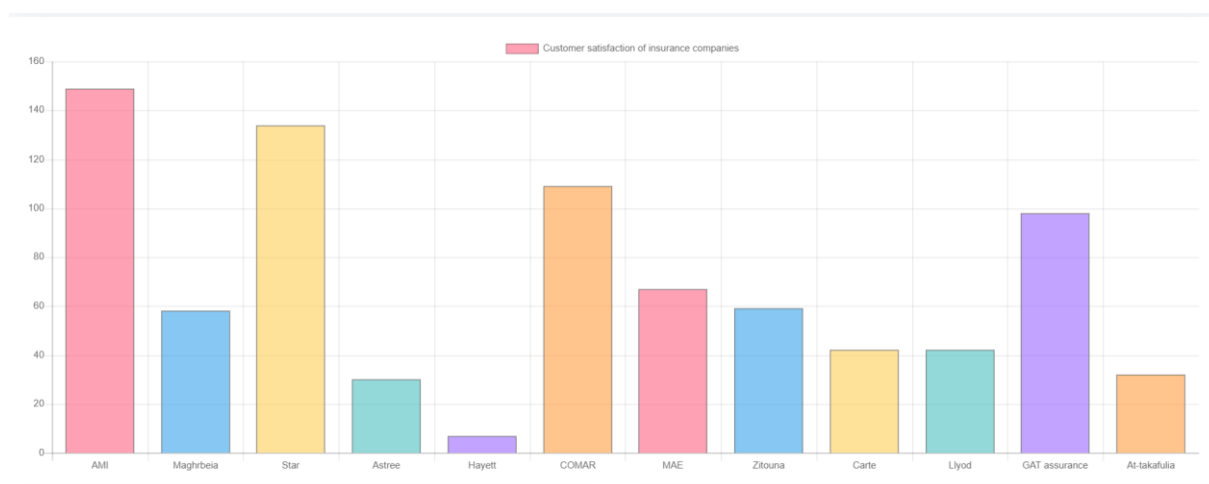


FIGURE 67 DASHBOARDS : CUSTOMER SATISFACTION OF INSURANCE COMPANIES

Detect BM class

Client Informations

Usage Code:

Number of accidents:

Vehicule Informations

Car Brand:

Horse power:

Fuel:

Policy Informations

Intermediate type:

Policy Nature:

Policy Type:

Policy State:

FIGURE 68 BONUS-MALUS PREDICTION INTERFACE

Detect fraudulent client

Companies Informations

Company:

Agence Code:

Client Informations

Usage Code:

BonusMalus Class:

Number of accidents:

Car Informations

Car Brand:

Horse Power:

Fuel:

Policy Informations

Intermediate type:

Policy Nature:

Policy Type:

Policy State:

FIGURE 69 FRAUD DETECTION

Conclusion

Insurance is a large investment for the insured as well as for the insurance companies. The companies' investment resides in its data provided by their customers. The availability of this data via connected devices has skyrocketed, and insurers will have to make progress in advanced analytics and artificial intelligence.

Our decision making tool is only but a first step in revolutionizing the insurance industry in Tunisia, yet it will still be important for insurers to explore additional levers in conjunction with reducing claims expenditure via optimized risk selection. More effectively combating fraud, increasing use of allied repair workshops, and offering assistance and service add-ons are all initiatives that could potentially more than compensate for decreasing premiums.

Insurance companies could, for example, provide services for avoiding risk, point out necessary maintenance work to drivers, and identify intelligent parking solutions. Insurers also can sell their data and analysis solutions to third parties, such as media agencies focusing on location-based advertising.