telecom saint-étienne
école d'ingénieurs nouvelles technologies

UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Projet Big Data

Manal MOUAYANI

Qi LI

Lucas MARIE

SOMMAIRE

Contexte

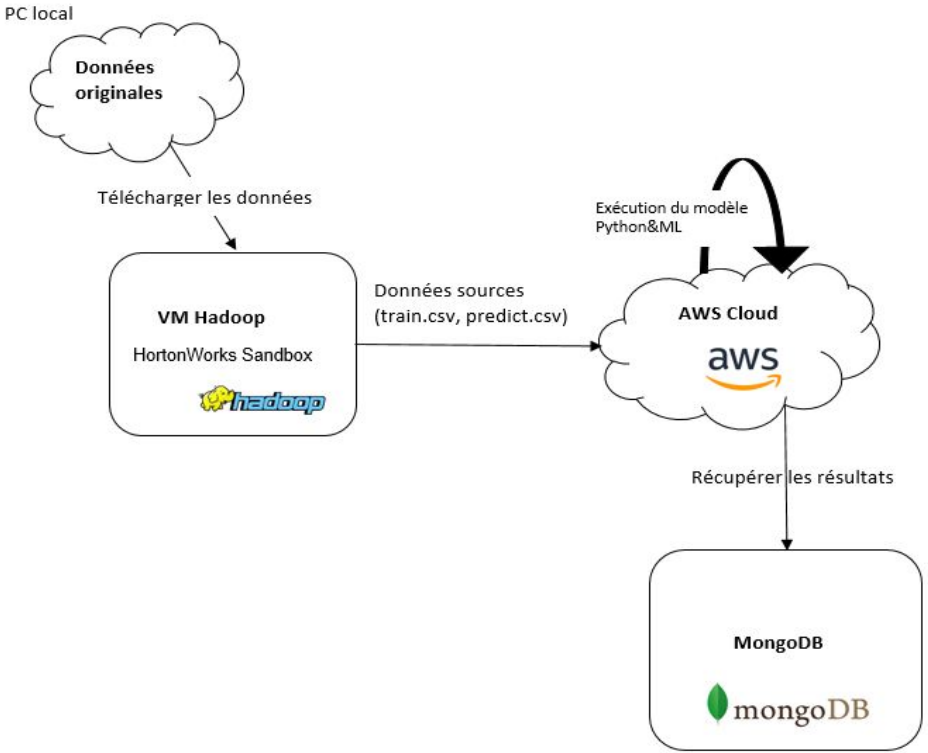
But de l’entreprise d’assurance :

Présentation des données

- 2 set de données :
- un set de prédiction
- un set de train

- 129 caractéristiques par individu

Présentation et interprétation du sujet



Créer des dossiers et modifier les autorisations des utilisateurs

```
[root@sandbox-hdp ~]# sudo -u hdfs hadoop dfs -chown admin:hdfs /input
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

[root@sandbox-hdp ~]# sudo -u hdfs hadoop fs -chmod 777 /input
[root@sandbox-hdp ~]# hadoop fs -ls /
Found 15 items
drwxrwxrwt   - yarn   hadoop          0 2018-11-29 17:56 /app-logs
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 19:01 /apps
drwxr-xr-x   - yarn   hadoop          0 2018-11-29 17:25 /ats
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 17:26 /atsv2
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 17:26 /hdp
drwxrwxrwx   - admin  hdfs            0 2020-01-27 08:17 /input
drwx------   - livy   hdfs            0 2018-11-29 17:55 /livy2-recovery
drwxr-xr-x   - mapred hdfs            0 2018-11-29 17:26 /mapred
drwxrwxrwx   - mapred hadoop          0 2018-11-29 17:26 /mr-history
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 18:54 /ranger
drwxrwxrwx   - spark  hadoop          0 2020-01-27 13:47 /spark2-history
drwxrwxrwx   - hdfs   hdfs            0 2020-01-30 10:21 /test
drwxrwxrwx   - hdfs   hdfs            0 2018-11-29 19:01 /tmp
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 19:21 /user
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 17:51 /warehouse
```

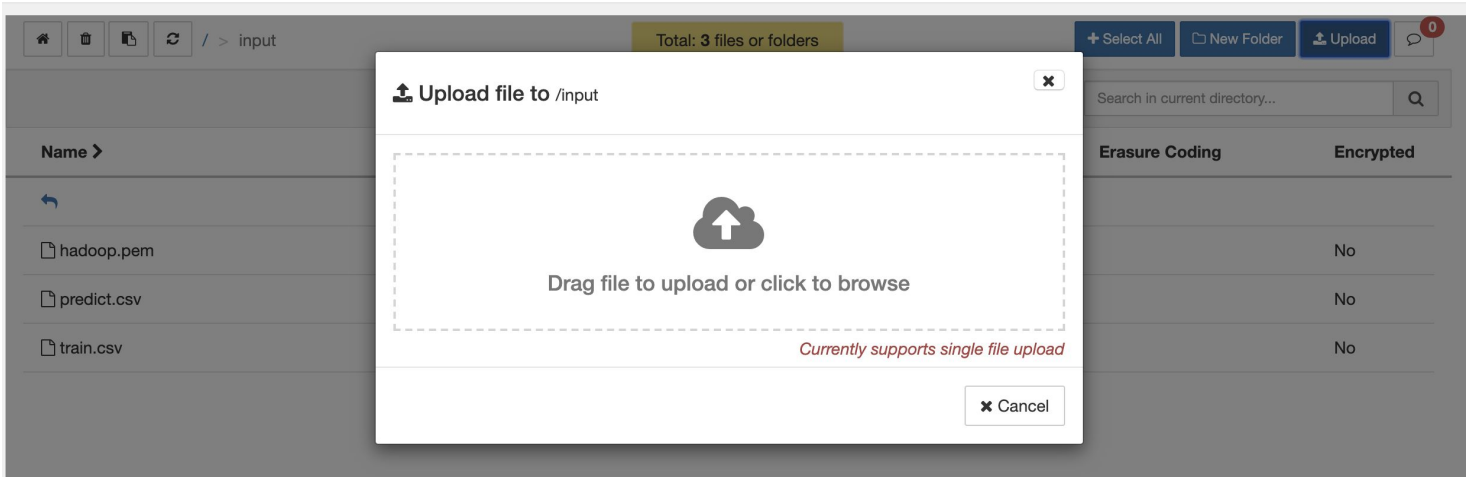
PC Local -> Hadoop VM -> HDFS

```
liqi@wifiroam041176 ~/Desktop/BIG DATA scp -P 2222 predict.csv root@localhost:
root@localhost's password:
predict.csv                                   100%  176KB  41.6MB/s   00:00
liqi@wifiroam041176 ~/Desktop/BIG DATA scp -P 2222 train.csv root@localhost:
train.csv                                     100%   20MB  91.0MB/s   00:00
```

```
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  dead.letter  hadoop.pem  ubuntu-AWS.pem
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  dead.letter  hadoop.pem  predict.csv  ubuntu-AWS.pem
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  hadoop.pem   train.csv
dead.letter      predict.csv  ubuntu-AWS.pem
```

```
[root@sandbox-hdp ~]# hadoop fs -ls /input
Found 1 items
-rwxrwxrwx   1 admin hdfs       1696 2020-01-27 08:17 /input/hadoop.pem
[root@sandbox-hdp ~]# hadoop fs -put predict.csv /input
[root@sandbox-hdp ~]# hadoop fs -put train.csv /input
[root@sandbox-hdp ~]# hadoop fs -ls /input
Found 3 items
-rwxrwxrwx   1 admin hdfs       1696 2020-01-27 08:17 /input/hadoop.pem
-rw-r--r--   1 root  hdfs     179969 2020-02-14 10:23 /input/predict.csv
-rw-r--r--   1 root  hdfs   21063261 2020-02-14 10:24 /input/train.csv
```

PC Local -> HDFS -> Hadoop VM




```
[root@sandbox-hdp ~]# hadoop fs -get /input/train.csv
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  hadoop.pem   train.csv
dead.letter      predict.csv  ubuntu-AWS.pem
```

Hadoop VM <-> AWS

```
[root@sandbox-hdp ~]# chmod 400 hadoop.pem
[root@sandbox-hdp ~]# scp -i hadoop.pem train.csv ec2-user@ec2-18-234-56-202.compute-1.amazonaws.com:
train.csv                                                                   100%   20MB 787.0KB/s   00:26
[root@sandbox-hdp ~]# scp -i hadoop.pem predict.csv ec2-user@ec2-18-234-56-202.compute-1.amazonaws.com:
predict.csv                                                                 100%  176KB 270.2KB/s   00:00
[root@sandbox-hdp ~]#
```

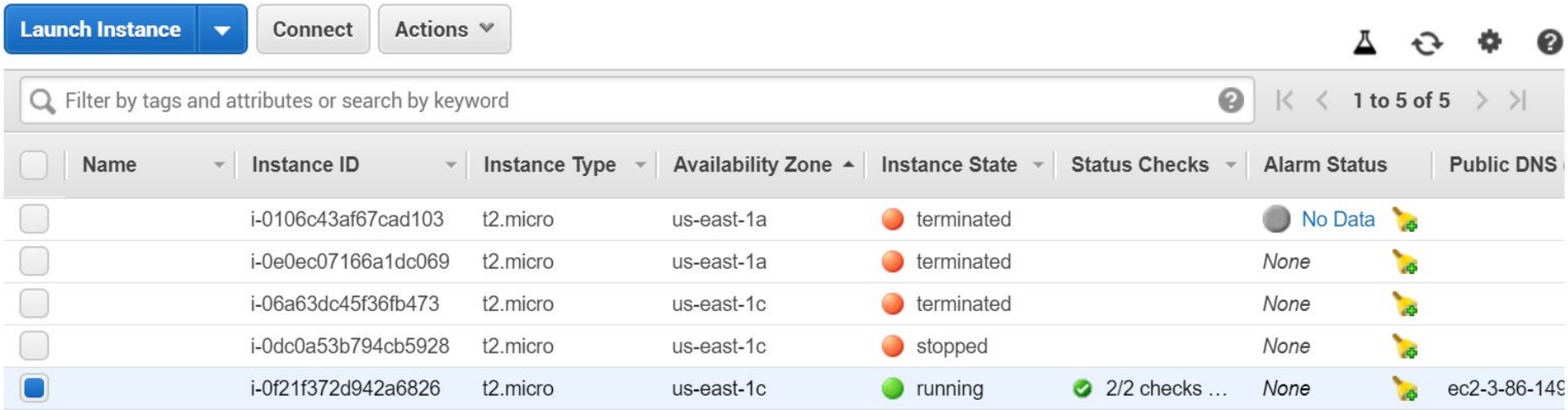
```
[ec2-user@ip-172-31-35-61 ~]$ ls
[ec2-user@ip-172-31-35-61 ~]$ ls
predict.csv  train.csv
```

2ème partie:

Cloud

AWS

Création de l’instance de EC2



Téléchargement du fichier resultats.csv sur le FileSystem de la VM AWS

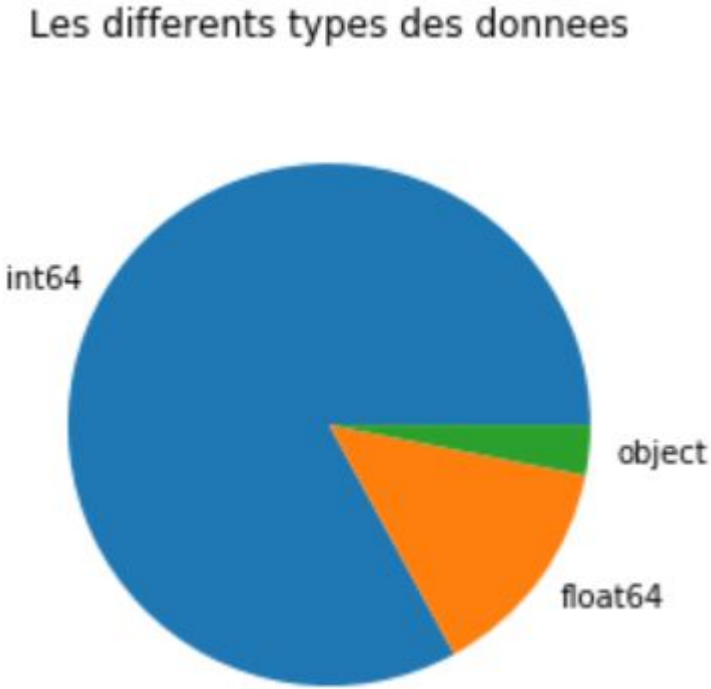
```
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  dead.letter  hadoop.pem  predict.csv  train.csv  ubuntu-AWS.pem
[root@sandbox-hdp ~]# scp -i hadoop.pem ec2-user@ec2-18-234-56-202.compute-1.amazonaws.com:results.csv ./
results.csv                                                                                   100%    0     0.0KB/s   00:00
[root@sandbox-hdp ~]# ls
anaconda-ks.cfg  dead.letter  hadoop.pem  predict.csv  results.csv  train.csv  ubuntu-AWS.pem
[root@sandbox-hdp ~]#
```

3eme partie:

Analyse des données

Les différents types de données:

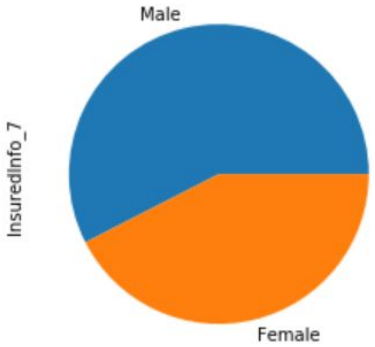
Nombre des lignes: 58881
Nombre des colonnes:129



➡ **Pour les caractéristiques de type objet, nous avons besoin d'utiliser la numérisation.**

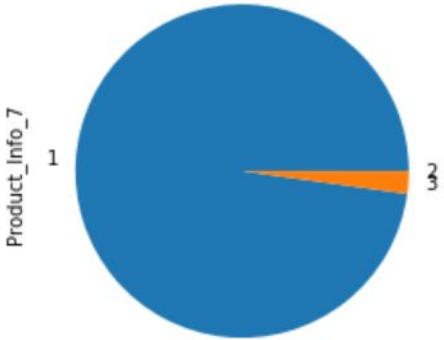
Analyse des variables

Les 2 sexes:



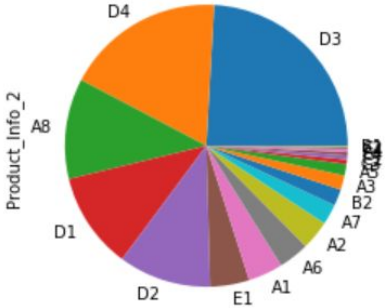
Male	0.575279
Female	0.424721

Le types de produits_7:

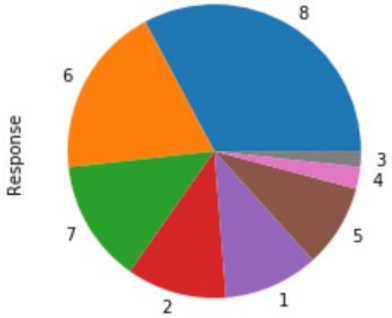


Type de prosuit_7	Le taux
1	**0.978142**
2	**0.021824**
3	**0.000034**

Les types de produit_2:

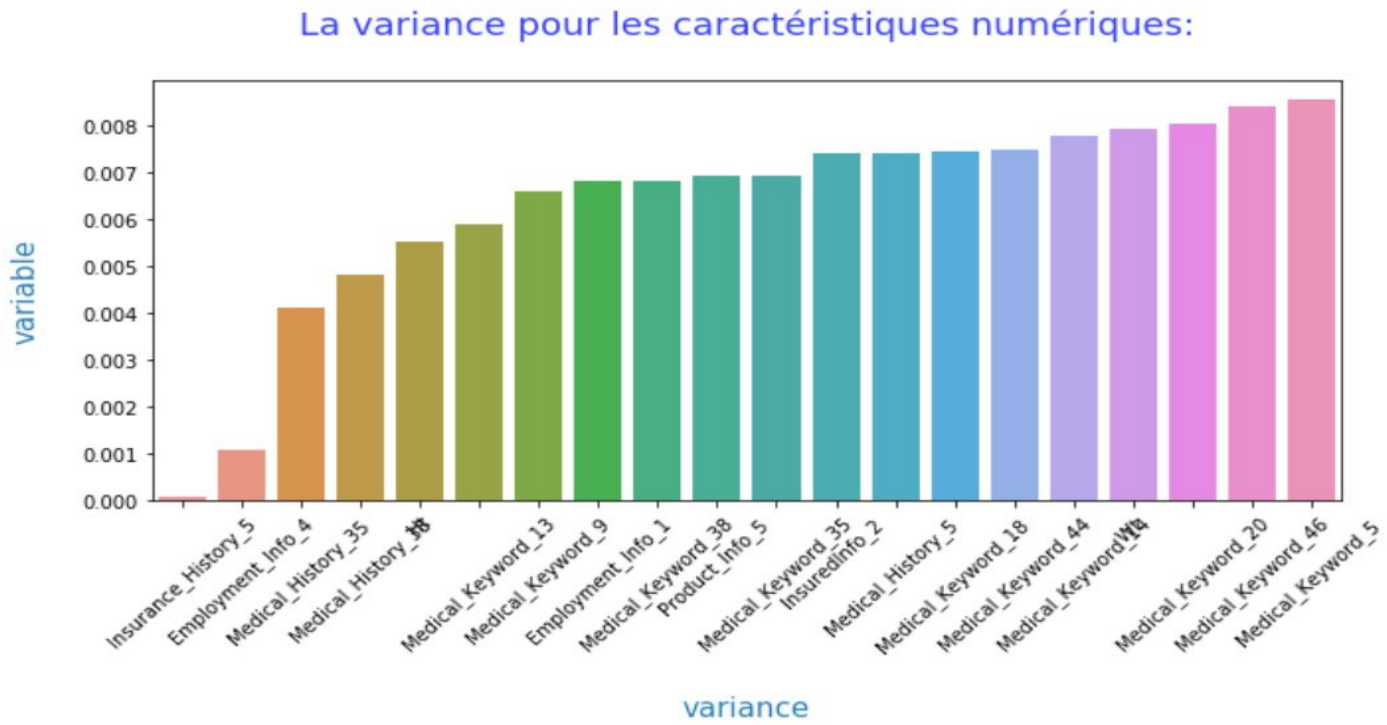


Le niveau de risque:

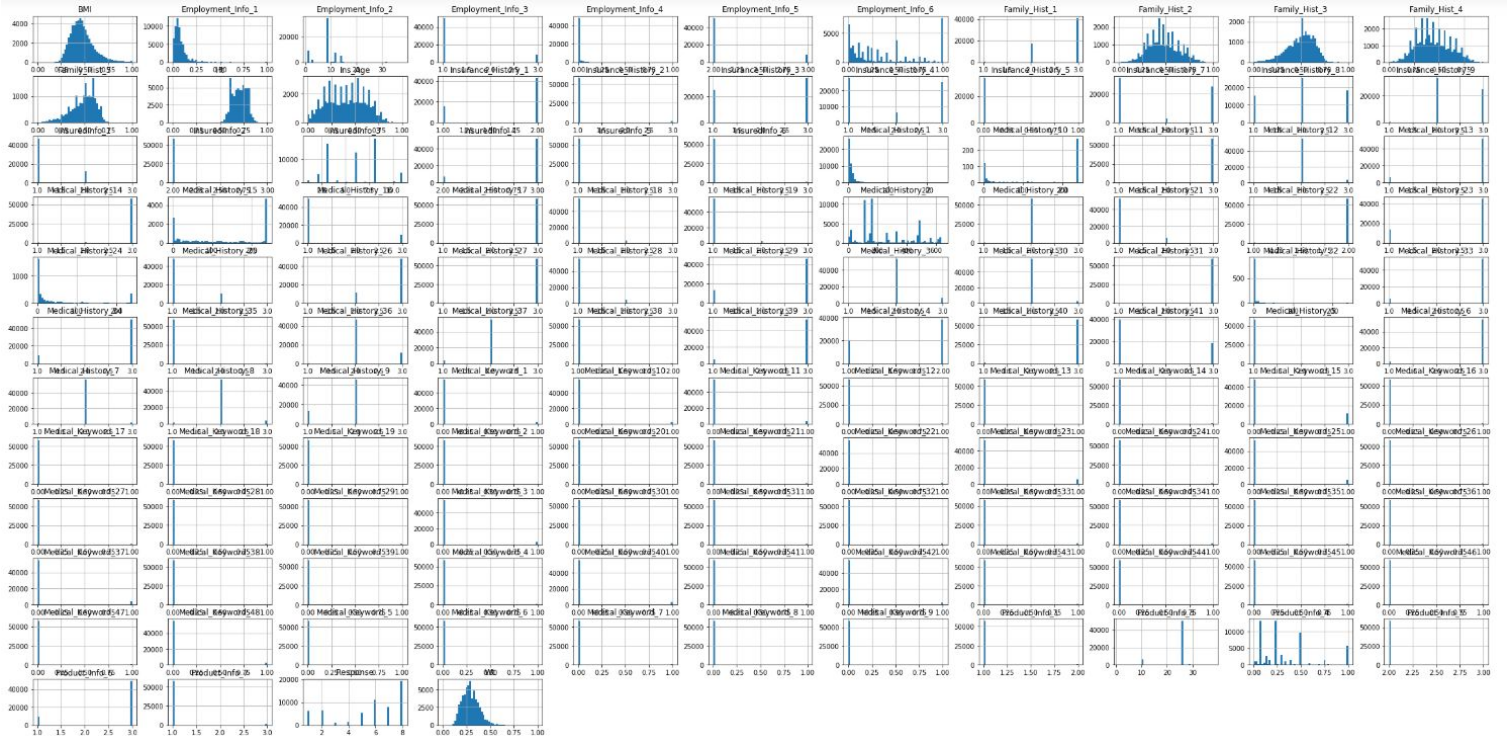


Le niveau de risque	Le taux
8	0.328238
6	0.189127
7	0.134950
2	0.110307
1	0.104584
5	0.091541
4	0.024116
3	0.017136

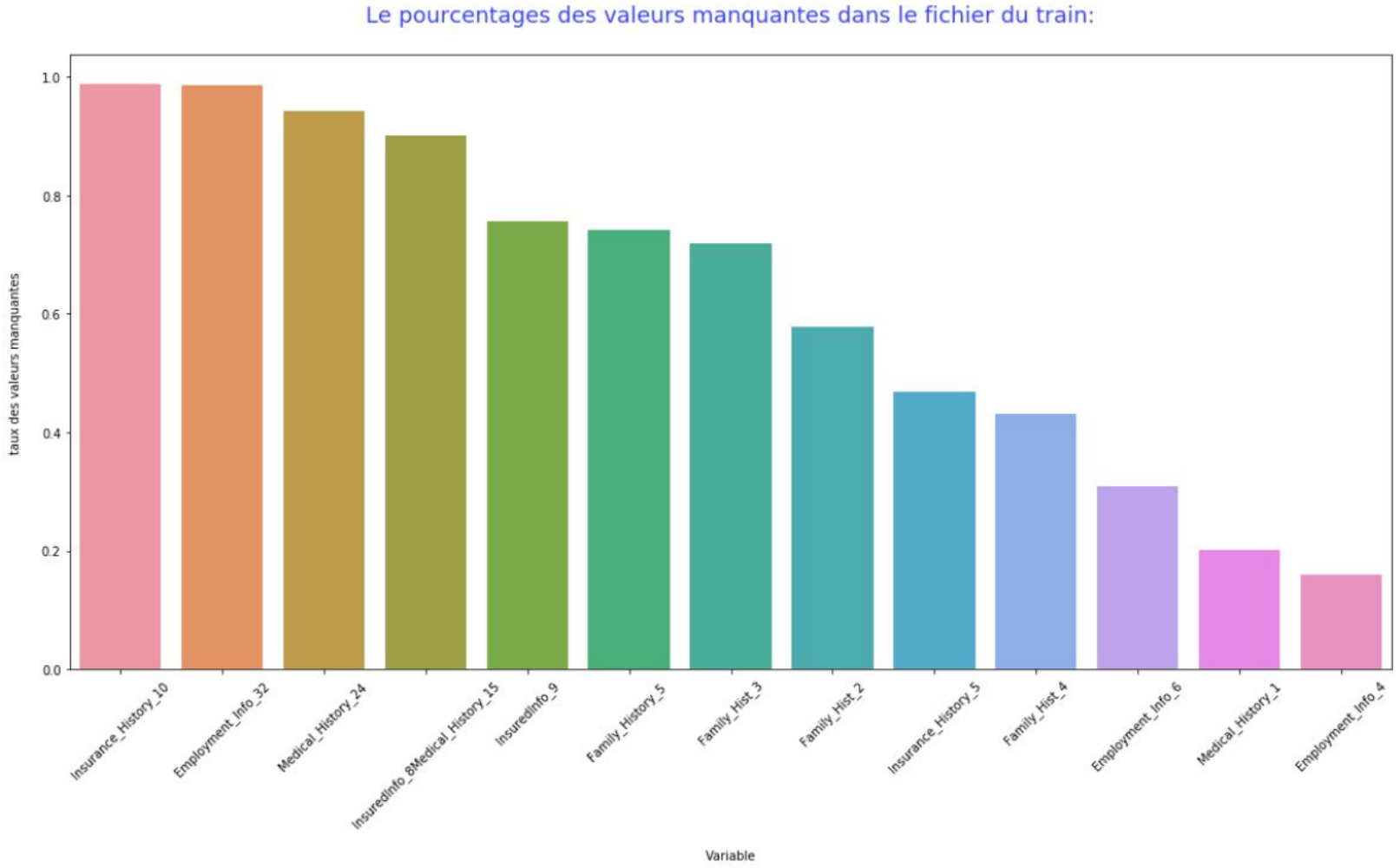
La variance pour les caractéristiques numériques :

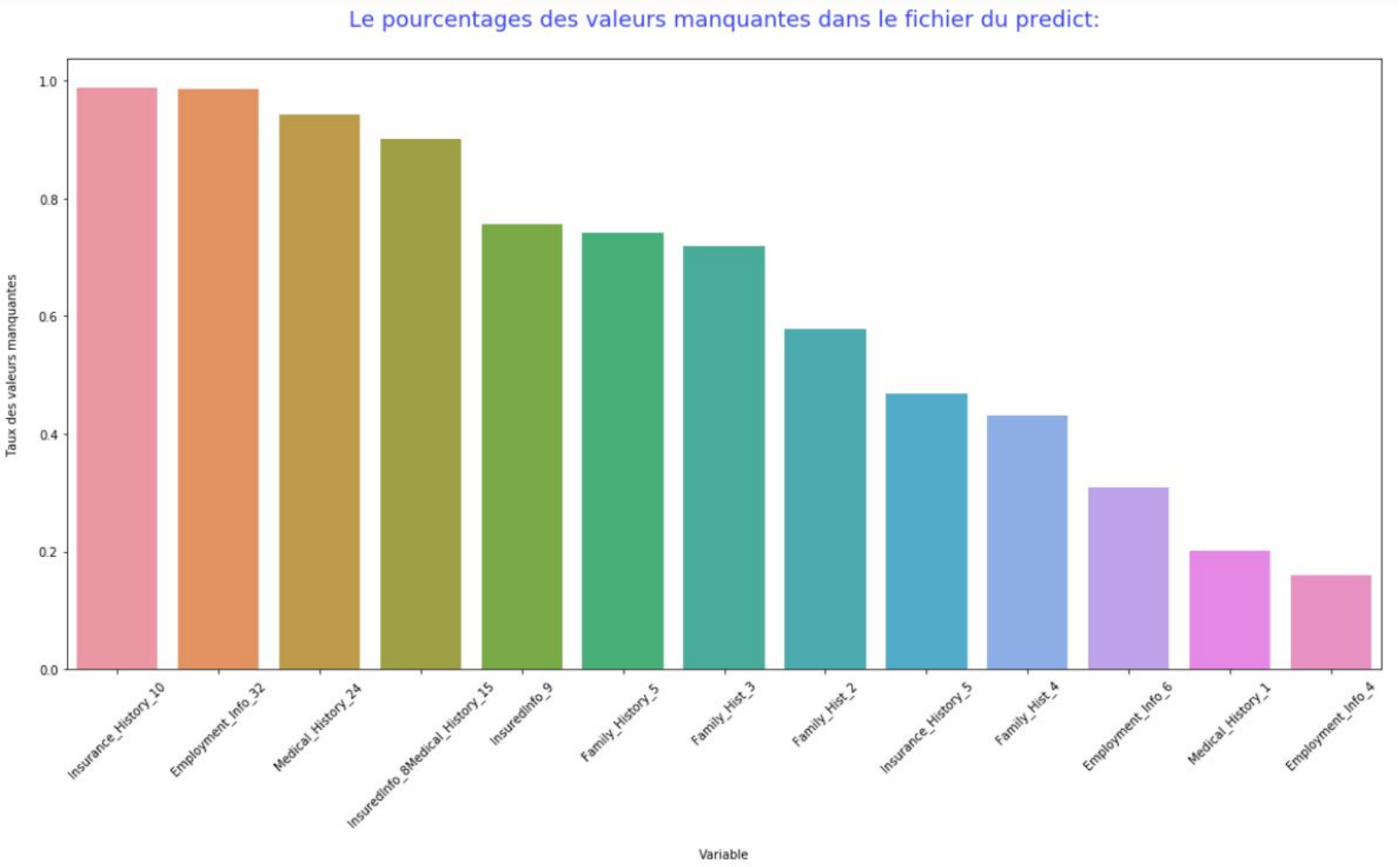


La distribution des données :

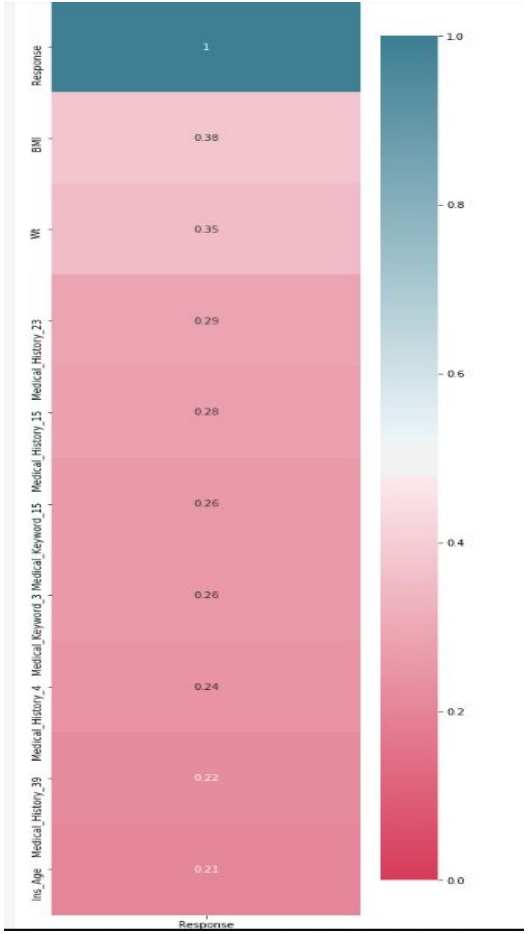


D'après la table de variance et le diagramme de distribution, nous pouvons voir qu'il existe de nombreuses caractères avec une faible variance, les valeurs de ces caractères sont presque fixes, ce qui n'a aucun sens, nous pouvons supprimer certaines parmi eux.

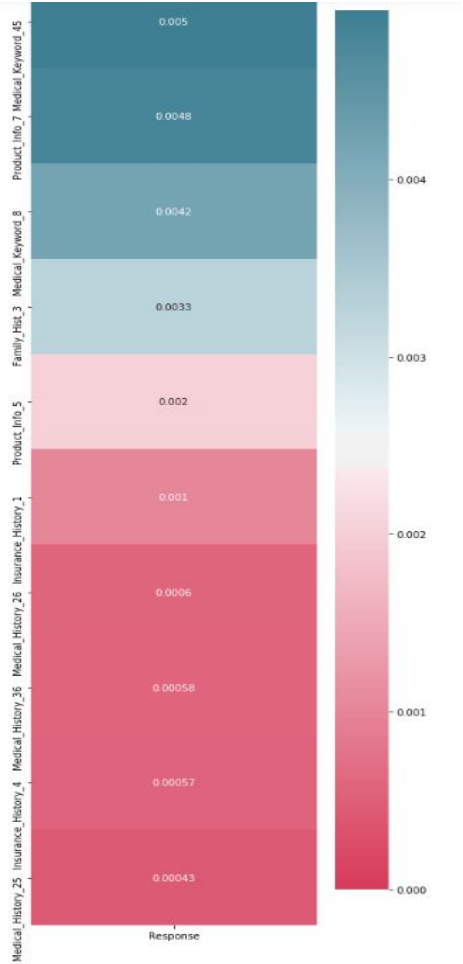

Le pourcentages des valeurs manquantes dans le fichier du train:
taux des valeurs manquantes
1.0
0.8
0.6
0.4
0.2
0.0
Insurance_History_10
Employment_Info_32
Medical_History_24
InsuredInfo_8Medical_History_15
InsuredInfo_9
Family_History_5
Family_Hist_3
Family_Hist_2
Insurance_History_5
Family_Hist_4
Employment_Info_6
Medical_History_1
Employment_Info_4
Variable


Le pourcentages des valeurs manquantes dans le fichier du predict:
Taux des valeurs manquantes
1.0
0.8
0.6
0.4
0.2
0.0
Insurance_History_10
Employment_Info_32
Medical_History_24
InsuredInfo_8Medical_History_15
InsuredInfo_9
Family_History_5
Family_Hist_3
Family_Hist_2
Insurance_History_5
Family_Hist_4
Employment_Info_6
Medical_History_1
Employment_Info_4
Variable

La corrélation entre la variable cible et les 10 valeurs les plus corrélés avec elle:



La corrélation entre la variable cible et les 10 valeurs les moins corrélés avec elle:



Traitement des données & Formation aux modèles

Traitement des données nécessaire:

1、Supprimer les lignes en double
2、Traitement de la valeur manquante
3、Numérisation des données
4、L'échantillonnage stratifié

Modèle de base : LinearSVC

Méthode utilisée :

1、Variable de contrôle
2、Rasoir d'Ockham
3、Validation croisée

Traitement de la valeur manquante

Qui remplacer ? et Comment ?

Taux de valeur manquant > 0.5 ou > 0.7 ?
Médiane ? Moyenne

Taux de valeur manquant > 0.7
Médiane

```
precision in train set: 0.3915239648455196
precision in test set: 0.3914786626800761
              precision    recall  f1-score   support

           1       0.40      0.11      0.18      1539
           2       0.20      0.49      0.28      1624
           3       0.00      0.00      0.00       252
           4       0.07      0.07      0.07       355
           5       0.35      0.12      0.18      1347
           6       0.28      0.02      0.04      2783
           7       0.37      0.13      0.19      1986
           8       0.50      0.89      0.64      4830

    accuracy                           0.39     14716
   macro avg       0.27      0.23      0.20     14716
weighted avg       0.37      0.39      0.31     14716
```

```
Average prediction score: 0.29346375188867685
```

Taux de valeur manquant > 0.5
Médiane

```
precision in train set: 0.395080184832835
precision in test set: 0.3925659146507203
              precision    recall  f1-score   support

           1       0.50      0.03      0.05      1539
           2       0.32      0.04      0.08      1624
           3       0.00      0.00      0.00       252
           4       0.00      0.00      0.00       355
           5       0.83      0.00      0.01      1347
           6       0.24      0.51      0.33      2783
           7       0.54      0.00      0.01      1986
           8       0.50      0.88      0.64      4830

    accuracy                           0.39     14716
   macro avg       0.37      0.18      0.14     14716
weighted avg       0.45      0.39      0.29     14716
```

```
Average prediction score: 0.32408257588891964
```

Traitement de la valeur manquante

Qui remplacer ? et Comment ?

Taux de valeur manquant > 0.5 ou > 0.7 ?
Médiane ? Moyenne

Taux de valeur manquant > 0.7
Moyenne

```
precision in train set: 0.16897707710428558
precision in test set: 0.1679804294645284
              precision    recall  f1-score   support

           1       0.67      0.01      0.03      1539
           2       0.34      0.08      0.12      1624
           3       0.00      0.00      0.00       252
           4       0.21      0.01      0.02       355
           5       0.10      0.92      0.18      1347
           6       0.34      0.07      0.12      2783
           7       0.27      0.07      0.11      1986
           8       0.76      0.15      0.25      4830

    accuracy                           0.17     14716
   macro avg       0.34      0.16      0.10     14716
weighted avg       0.47      0.17      0.15     14716
```

```
Average prediction score: 0.31712280064812987
```

Taux de valeur manquant > 0.5
Moyenne

```
precision in train set: 0.12510192987224789
/usr/local/lib/python3.7/site-packages/sklearn/svm/base
precision in test set: 0.12700462082087524
              precision    recall  f1-score   support
  "the number of iterations.", ConvergenceWarning)

/usr/local/lib/python3.7/site-packages/sklearn/metrics/
  predicted samples.
           1       0.42      0.01      0.01      1539
  'precision', 'predicted', average, warn_for)
           2       0.16      0.10      0.12      1624
           3       0.00      0.00      0.00       252
           4       0.00      0.00      0.00       355
           5       0.09      0.92      0.17      1347
           6       0.00      0.00      0.00      2783
           7       1.00      0.00      0.00      1986
           8       0.78      0.09      0.17      4830

    accuracy                           0.13     14716
   macro avg       0.31      0.14      0.06     14716
weighted avg       0.46      0.13      0.09     14716
```

```
Average prediction score: 0.27721585818942307
```

Créer nouvelle caractères

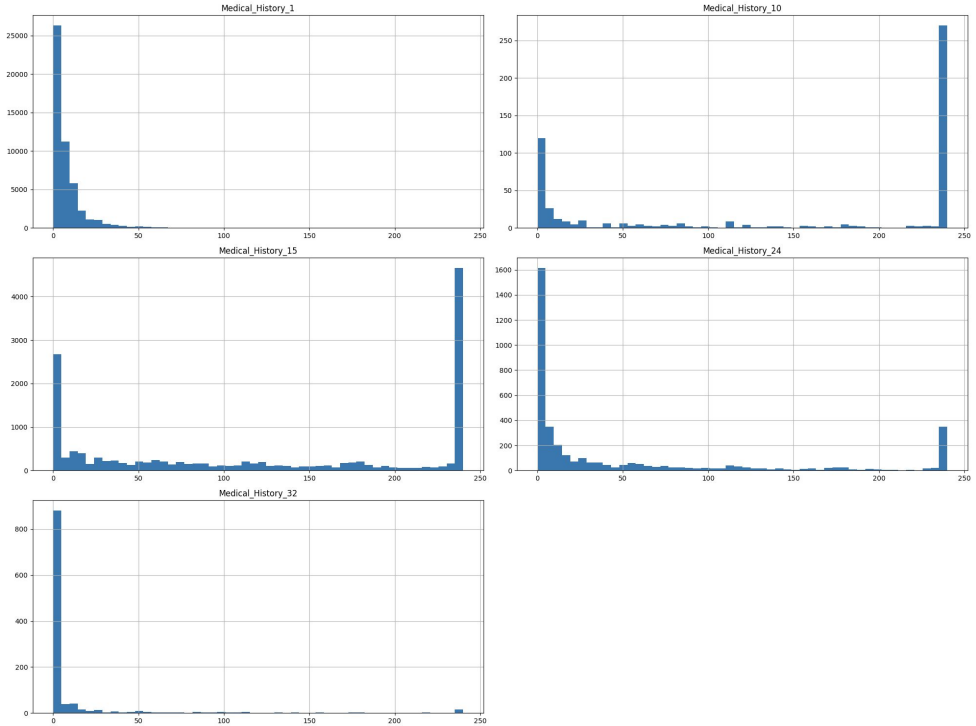
Matrice creuse

```
D3    14200
D4    10699
A8     6775
D1     6510
D2     6234
E1     2632
A1     2335
A6     2080
A2     1959
A7     1376
B2     1114
A3      971
A5      769
C3      303
C1      282
C4      218
A4      210
C2      160
B1       54
Name: Product_Info_2, dtype: int64
```

CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS
Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key	Medical_Key
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
						0	0	0	0	0	0	0	0	1	0	0
						0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
Average prediction score: 0.3315402717711335
```

Standardization des données



variables_discrete =
['Medical_History_1',
'Medical_History_10',
'Medical_History_15',

'Medical_History_24',
'Medical_History_32']

PCA dimensionality reduction

caractères 128 - > 119

```
precision in train set: 0.29949261574703273
precision in test set: 0.29532481652622994
              precision    recall  f1-score   support

           1       0.29      0.09      0.13      1539
           2       0.19      0.25      0.22      1624
           3       0.03      0.53      0.05       252
           4       0.37      0.02      0.04       355
           5       0.38      0.06      0.10      1347
           6       0.34      0.10      0.15      2783
           7       0.25      0.17      0.20      1986
           8       0.65      0.62      0.63      4830

    accuracy                           0.30     14716
   macro avg       0.31      0.23      0.19     14716
weighted avg       0.41      0.30      0.31     14716
```

```
Average prediction score: 0.240338138574792 25
```

Sélection des caractères

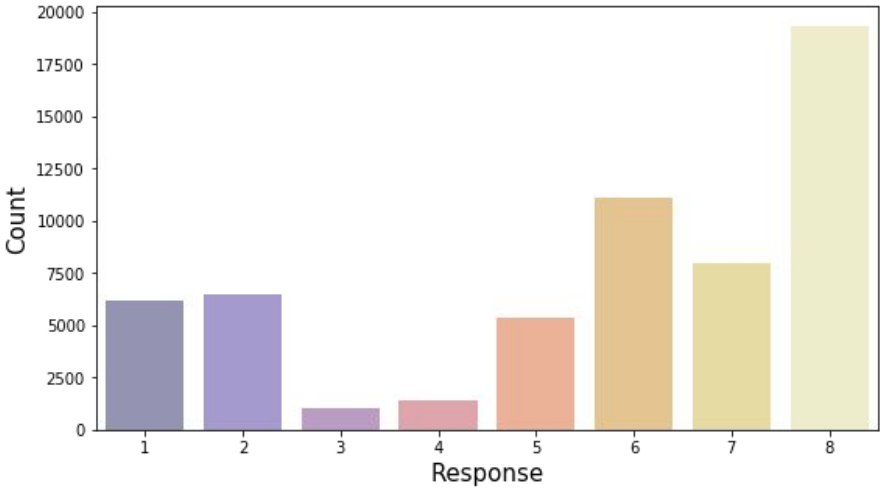
```
Insurance_History_5     0.000054
Employment_Info_4       0.001083
Medical_History_35      0.004106
Medical_History_38      0.004817
Ht                      0.005513
Medical_Keyword_13      0.005892
Medical_Keyword_9       0.006580
Employment_Info_1       0.006803
Medical_Keyword_38      0.006814
Product_Info_5          0.006898
Medical_Keyword_35      0.006898
InsuredInfo_2           0.007400
Medical_History_5       0.007401
Medical_Keyword_18      0.007417
Medical_Keyword_44      0.007467
Medical_Keyword_14      0.007785
Wt                      0.007923
Medical_Keyword_20      0.008019
Medical_Keyword_46      0.008403
Medical_Keyword_5       0.008537
dtype: float64
```

```
precision in train set: 0.33693485548609226
precision in test set: 0.33765969013318836
```

	precision	recall	f1-score	support
1	0.52	0.02	0.04	1539
2	0.19	0.62	0.29	1624
3	0.00	0.00	0.00	252
4	0.00	0.00	0.00	355
5	0.32	0.02	0.04	1347
6	0.26	0.48	0.33	2783
7	0.49	0.02	0.04	1986
8	0.63	0.53	0.57	4830
accuracy			0.34	14716
macro avg	0.30	0.21	0.16	14716
weighted avg	0.43	0.34	0.30	14716

```
Average prediction score: 0.2967050024909657
```

Resampling



8	19318
6	11134
7	7944
2	6494
1	6157
4	5906
5	5389
3	4094

8	19318
6	11134
7	7944
2	6494
1	6157
5	5389
4	1419
3	1009

```
precision in train set: 0.5516286350773677
precision in test set: 0.5289300981395629
Average prediction score: 0.526862880190074
              precision    recall  f1-score   support

           1       0.50      0.22      0.31      1539
           2       0.46      0.26      0.33      1624
           3       0.59      0.40      0.48      1023
           4       0.47      0.64      0.54      1477
           5       0.55      0.47      0.51      1347
           6       0.38      0.44      0.41      2784
           7       0.45      0.36      0.40      1986
           8       0.65      0.85      0.74      4829

    accuracy                           0.53     16609
   macro avg       0.51      0.46      0.46     16609
weighted avg       0.52      0.53      0.51     16609
```

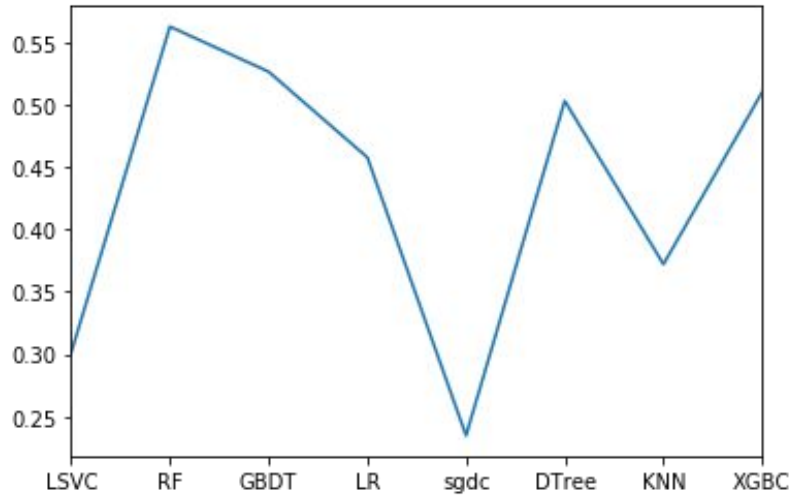
Sélection du modèle

RandomForestClassifier : Overfitting

```
precision in train set: 0.9917514600517792
precision in test set: 0.5518092600397375
Average prediction score: 0.5469323930128527
```

GradientBoostingClassifier

```
precision in train set: 0.55160856563710 44
precision in test set: 0.5298332229514119
Average prediction score: 0.5283284303475009
```



XGBoost

```
precision in train set: 0.5268830152327052
precision in test set: 0.5109278102233729
Average prediction score: 0.5144804817310304
```

Grid Search & Prévision

```
Parameters
----------
loss : {'deviance', 'exponential'}, optional (default='deviance')
    loss function to be optimized. 'deviance' refers to
    deviance (= logistic regression) for classification
    with probabilistic outputs. For loss 'exponential' gradient
    boosting recovers the AdaBoost algorithm.

learning_rate : float, optional (default=0.1)
    learning rate shrinks the contribution of each tree by `learning_rate`.
    There is a trade-off between learning_rate and n_estimators.

n_estimators : int (default=100)
    The number of boosting stages to perform. Gradient boosting
    is fairly robust to over-fitting so a large number usually
    results in better performance.

subsample : float, optional (default=1.0)
    The fraction of samples to be used for fitting the individual base
    learners. If smaller than 1.0 this results in Stochastic Gradient
    Boosting. `subsample` interacts with the parameter `n_estimators`.
    Choosing `subsample < 1.0` leads to a reduction of variance
    and an increase in bias.

criterion : string, optional (default="friedman_mse")
    The function to measure the quality of a split. Supported criteria
    are "friedman_mse" for the mean squared error with improvement
    score by Friedman, "mse" for mean squared error, and "mae" for
    the mean absolute error. The default value of "friedman_mse" is
    generally the best as it can provide a better approximation in
    some cases.

    .. versionadded:: 0.18

min_samples_split : int, float, optional (default=2)
    The minimum number of samples required to split an internal node:

    - If int, then consider `min_samples_split` as the minimum number.
    - If float, then `min_samples_split` is a fraction and
      `ceil(min_samples_split * n_samples)` are the minimum
      number of samples for each split.

    .. versionchanged:: 0.18
       Added float values for fractions.

min_samples_leaf : int, float, optional (default=1)
```

learning_rate=0.01,
n_estimators=1200,
max_depth=7,
min_samples_leaf=60,
min_samples_split=1200

```
Average prediction score: 0.6264816458570729
```

1、Former le modèle sur toutes les données
2、Générer des résultats de prédiction

MongoDB

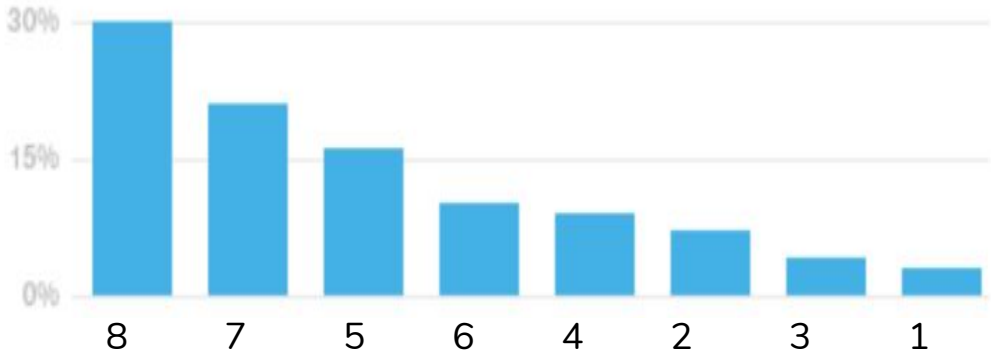
Importation des données dans MongoDB

- predict.csv
- résultats du modèle

Visualisation des résultats

Response

string



Affichage de statistique

statistique pour le cas très peu risqué (classe 1):

statistique pour le cas très risqué (classe 8):

Détermination de critères importants

statistique pour le cas 1 :

Insurance_History_4

string

50 % 41 % 9 %

statistique pour le cas 2 :

Insurance_History_4

string

53 % 33 % 13 %

Merci de votre attention