# UNIVERSITE DE PARIS

## *SUJET :*

***Data science 1 :***
***COMPARAISON DES METHODES SUPERVISEES.***

---

***SPECIALITE :***

***Master 1 AMDS + CYBER SECURITE***

***FAIT PAR :***

***GHARBI MOHAMED 22018838***

***KWAN TEAU TIFFANY 51703508***

***ZERZOUR SOFIAN 51708707***

***Année Scolaire : 2020/2021***

# Table des matières



# Table des figures

# Introduction

La Data science est en fait un terme générique englobant toute une famille d'outils facilitant l'exploration et l'analyse des données pour des fins décisionnelles.

Les techniques mises en action lors de l'utilisation de ces instruments d'analyse et de prospection sont particulièrement efficaces pour extraire des informations significatives depuis de grandes quantités de données.

En dépit des méthodes classiques d'analyses statistiques, ces instruments d'analyse sont particulièrement adaptés au traitement de grands volumes de données et avec l'augmentation de la capacité de stockage des supports informatiques, un maximum de renseignements sera capté, ordonné et rangé « Comportement des acheteurs, caractéristiques des produits, historisation de la production », désormais plus rien n'échappe à la collecte.

Le travail présenté dans ce rapport rentre dans ce cadre et consiste en la réalisation d'une étude sur plusieurs sources d'informations afin de faire une comparaison entre ces différentes techniques d'analyse.

Le présent rapport est organisé de la manière suivante :

**Chapitre 1** : Etat de l'art

C'est la partie théorique où nous présentons les concepts de base relatifs au Machine Learning, en abordant ses différentes approches.

**Chapitre 2** : Réalisation de la solution

Ce chapitre est composé de deux parties à savoir :

- Partie 1 : Etude des données synthétiques
  - ✓ Exploratoire préliminaire : qui consiste en l'utilisation des méthodes classiques d'analyse statistique.
  - ✓ Classification supervisée : Dans cette étape nous appliquerons et comparerons les différentes approches de classification supervisée.
- Partie 2 : Etude des données réelles :

Dans cette partie nous allons faire une comparaison entre les approches de classification supervisée sur les données « visa premier, credit card fraud »

Enfin nous terminerons par une conclusion générale.

# Chapitre 1 : Etat de l'art

## Introduction

L'exploitation de l'information est l'un des problèmes majeurs que rencontre les entreprises, ce qui a fait naitre la data science.

Cette technologie fait appel à différentes approches à des fin d'exploitation et d'analyse de données. Ces approches se divisent en deux grandes familles de machine learning : « supervised learning » et « unsupervised learning ».

## Machine learning

Dans notre cas nous allons aborder celle du supervised learning.

**L'apprentissage automatique « machine learning »** : Est une discipline qui se base sur les statistiques, les probabilités, l'intelligence artificielle et l'optimisation tout en entrainant des algorithmes sur des données connues pour des fins décisionnelles et prédictives.

**L'apprentissage supervisé « supervised learning »** : Est le fait d'entrainer notre modèle afin qu'il puisse faire une liaison entre les inputs donnés et les output voulus.

L'apprentissage supervisé est constitué d'un ensemble d'approches de prédictions. Dans notre travail nous avons fait appel à certaines d'entre elles à savoir : « KNN, Logistic regression, SVC, LDA, QDA, Naive Bayes, Random Forest, Decision Tree …etc. ».

**KNN** (K-nearest neighbors) : l'algorithme du K plus proche voisin est une méthode pouvant être utilisée pour les cas de régression et de classification. L'idée est de pouvoir classer une donnée à partir d'un ensemble de données labélisés (classés). Les étapes de cet algorithme sont les suivantes :

- Choisir un nombre K de voisins.
- Calculer la distance (euclidienne, Manhattan…) de la donnée à classer aux autres données (labélisés).
- Prenez les K voisins ayant la plus petite distance par rapport à la donnée à classer.
- Déterminez à quelles catégories appartiennent les K données.
- Attribuez à la donnée à classer, la classe majoritaire parmi les K données.

**Bayésien-Naïf** : cette méthode de classification probabiliste repose, comme son nom l'indique, sur le théorème de Bayes (fondé sur les probabilités conditionnelles). Cette méthode devra attribuer une classe à une donnée en calculant la probabilité que cette donnée appartienne à telle ou telle classe sachant un certain nombre de caractéristiques. La classe correspondant à la probabilité la plus grande sera ensuite sélectionnée. Cette méthode est dite « naïve » puisqu'on suppose que les variables explicatives sont indépendantes, ce qui est en réalité faux.

**Régression Logistique :** cet algorithme d'apprentissage automatique est le plus couramment utilisé pour les problèmes de classification binaire, c'est-à-dire pour les classes pouvant prendre les valeurs « OUI/NON », « VRAI/FAUX », « A/B » … Cette méthode consiste à prédire la probabilité qu'un évènement arrive ou non et à déterminer une relation entre les variables explicatives (caractéristiques) et la variable à expliquer.

**SVM linéaire** (séparateur à vaste marge linéaire) : pour déterminer la classe d'un objet, il faut connaître la frontière séparant les classes (sur un plan) afin de déterminer à quelle catégorie appartient cet objet. Cette frontière est justement déterminée par le SVM, qui va faire en sorte de la placer le plus loin possible des points d'entrainement.

**SVM non linéaire** : ici, il est impossible de trouver une ligne droite qui permet de séparer les données. Il faut donc trouver une transformation qui va permettre de classer les données (on appelle cette méthode l'astuce du noyau).

**CART (Classification and Regression Tree) :** Les arbres de régression (Regression tree) permettent de prédire une valeur réelle (donnée quantitative) et les arbres de classification permettent de déterminer à quelle classe appartient une donnée. Ces arbres sont appelés arbres de décision et sont construits de manière itérative où sont appliquées des règles, test à chaque nœud. Chaque branche représente le résultat du test et les feuilles représentent les différentes valeurs ou classes possibles pour la variable à prédire.

**Random Forest** : cet algorithme utilise un ensemble d'arbres de décision indépendants. Il fonctionne selon le principe du bagging, c'est-à-dire qu'on divise les données en plusieurs sous-ensembles aléatoirement constitués, on entraîne un modèle sur chaque sous-ensemble puis avec les prédictions obtenues sur les différents arbres, on détermine le résultat en choisissant celui qui a la catégorie la plus fréquente ou bien en faisant la moyenne des valeurs prédites.

**LDA (Linear Discriminant Analysis)** : On modélise la distribution des variables explicatives par une loi de probabilité gaussienne puis on détermine les paramètres de la loi. Puis on applique la loi de Bayes pour déterminer la probabilité d'une classe sachant les variables explicatives.

**QDA (Quadratic Discriminant Analysis)** : elle constitue une généralisation de la LDA, sauf qu'ici on ne considère pas la matrice de covariance comme indépendante de la classe.

# Chapitre 2 : Réalisation de la solution

## Introduction

Après avoir terminé l'état de l'art, nous entamons la partie réalisation et mise en œuvre de la solution. Ce chapitre présente les étapes de réalisation de notre solution, qui sera divisé en deux parties de traitement de données à savoir : les données synthétiques et les données réelles.

### Partie 1 : Données synthétiques

#### *Descriptif des données*

Dans cette partie nous avons utilisé trois data sets « spiral, flame et aggregation » décris comme suit :

- Spiral : Ce dataset contient 312 observations et 3 variables d'études sachant que l'une d'elles est la variable à expliquer, qui contient 3 classes.
- Flame : Ce dataset est constitué de 240 observations, 2 variables et une seule caractéristique qui est composée de 2 classes.
- Aggregation : Ce dataset contient 788 observations et 3 variables tel que l'une d'elles et caractérisée par 7 classes.

#### *Etude exploratoire préliminaire*

Dans cette partie nous allons effectuer une étude statistique sur les données de chaque dataset afin de savoir leur comportement.

Données spiral

1. Normalisation des données :

Des variables peuvent ne pas être comparables du fait de l'incompatibilité des unités de mesure. Cette étape est donc nécessaire pour que les résultats ne soient pas affectés. Pour faire cette normalisation, on soustrait aux données leur moyenne et on les divise par leur écart-type.

Out[68]:

| | V1 | V2 |
|---|---|---|
| 0 | 1.858044 | -1.224394 |
| 1 | 1.748278 | -1.319198 |
| 2 | 1.652233 | -1.414003 |
| 3 | 1.549327 | -1.508807 |
| 4 | 1.439561 | -1.574441 |

*Figure 1: La normalisation du dataset spiral*

2. Visualisation du nuage de points

Qu'est-ce qu'un nuage de points :

L'interprétation d'un nuage de point sert à vérifier certains critères entre deux variables à savoir « intensité, affinité, corrélation, les points aberrants ».

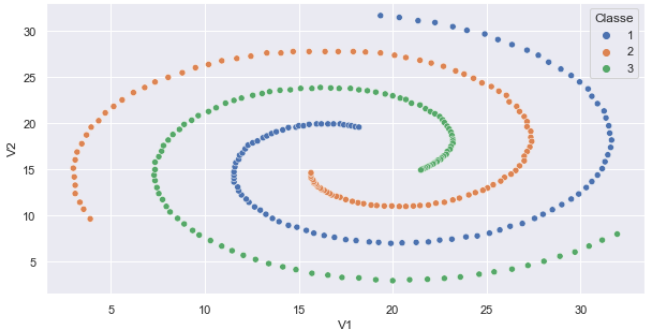



*Figure 2 : Nuage de points du dataset spiral*

D'après le graphe ci-dessus nous trouvons aucune corrélation, affinité ou points aberrants entre les deux variables.

3. Matrice de corrélation

Après avoir fait la description du nuage de points nous pouvons vérifier la corrélation avec la matrice de corrélation.

D'après la figure n° 3 nous trouvons que les 2 variables ne sont pas corrélées entre elles.

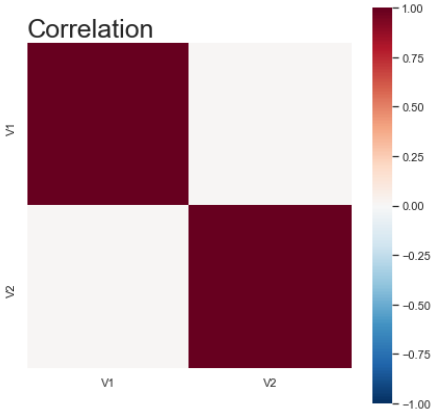



*Figure 3: Matrice de corrélation du dataset spiral*

4. La description statistique des variables et le boxplot

Spiral

| | V1 | V2 |
|---|---|---|
| count | 312.000000 | 312.000000 |
| mean | 18.408173 | 16.344712 |
| std | 7.299923 | 6.867232 |
| min | 3.000000 | 2.900000 |
| 25% | 12.912500 | 11.337500 |
| 50% | 18.325000 | 16.050000 |
| 75% | 23.400000 | 21.362500 |
| max | 31.950000 | 31.650000 |

*Figure 4: Summary du dataset spiral*

Dans le tableau ci-dessus nous trouvons la moyenne, l'écart-type, min, max, le premier et troisième quartile et la médiane des deux variables qui sont illustrés dans la figure 4.

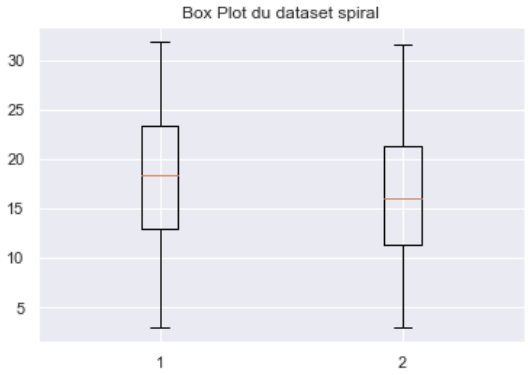



*Figure 5 : boxplot du dataset spiral*

Données flame

1. Normalisation des données :

Out[158]:

| | V1 | V2 |
|---|---|---|
| **0** | -1.712779 | 2.035183 |
| **1** | -1.869233 | 1.694577 |
| **2** | -1.853587 | 0.687568 |
| **3** | -2.025686 | 0.628332 |
| **4** | -2.135204 | 0.421006 |

*Figure 6: normalisation Des données flame*

2. Visualisation du nuage de points

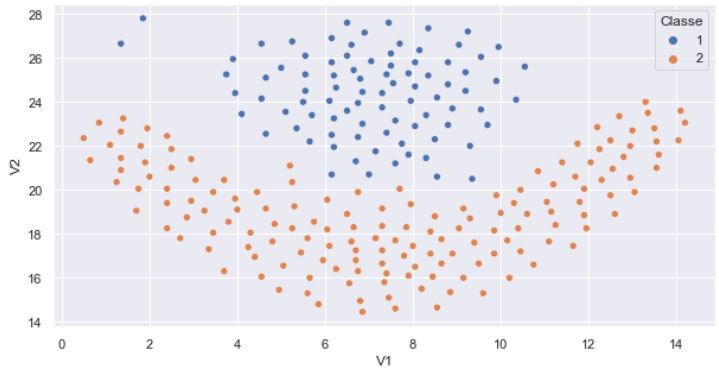



*Figure 7: nuage de points du dataset flame*

D'après le graphe ci-dessus nous trouvons aucune corrélation, affinité ou points aberrants entre les deux variables.

3. Matrice de corrélation

Après avoir fait la description du nuage de points nous pouvons vérifier la corrélation avec la matrice de corrélations.

D'après la figure 2 nous trouvons que les 2 variables ne sont pas corrélées entre elles.

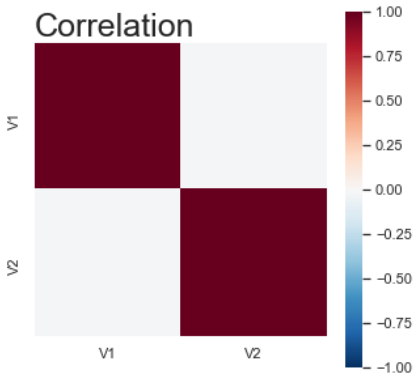



*Figure 8: matrice de corrélation du dataset flame*

4. La description statistique des variables et le boxplot

| | V1 | V2 |
|---|---|---|
| **count** | 240.000000 | 240.000000 |
| **mean** | 7.323750 | 20.928542 |
| **std** | 3.202509 | 3.383390 |
| **min** | 0.500000 | 14.450000 |
| **25%** | 5.250000 | 18.237500 |
| **50%** | 7.300000 | 20.775000 |
| **75%** | 9.312500 | 23.562500 |
| **max** | 14.200000 | 27.800000 |

*Figure 9: summary du dataset Flame*

Le tableau de la figure ci-dessus résume les différents comportements de chaque variable à savoir : « la moyenne, l'écart-type, min, max, le premier et troisième quartile et la médiane ».

Ces différentes valeurs sont illustrées ci-dessous.

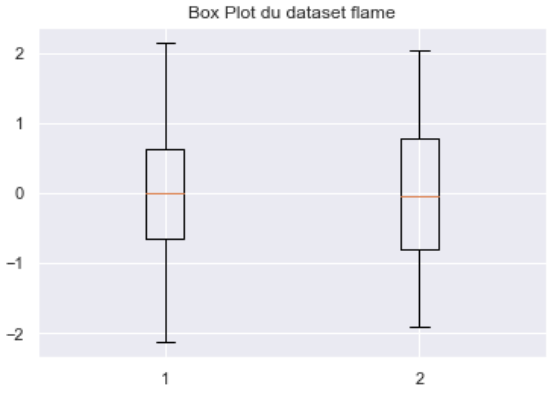



*Figure 10: Boxplot du dataset Flame*

Données Aggregation :

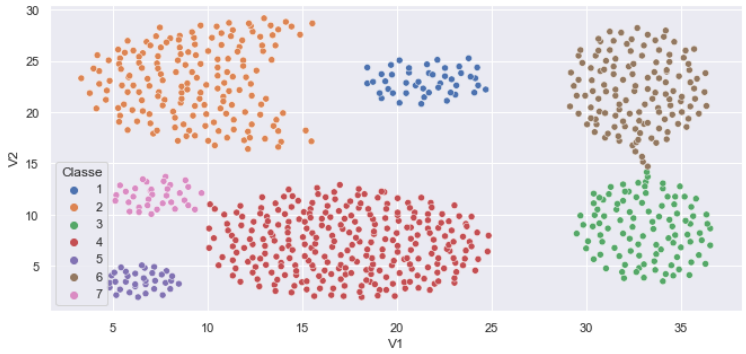
1. Normalisation des données :

Out[167]:

| | V1 | V2 |
|---|---|---|
| 0 | -0.405095 | 1.790853 |
| 1 | -0.470647 | 1.654791 |
| 2 | -0.516029 | 1.753745 |
| 3 | -0.546284 | 1.809407 |
| 4 | -0.586624 | 1.716637 |

*Figure 11: normalisation du dataset aggregation*

2. Visualisation du nuage de points




D'après le graphe ci-dessus nous trouvons aucune corrélation, affinité ou points aberrants entre les deux variables.

3. Matrice de corrélation

Après avoir fait la description du nuage de points nous pouvons vérifier la corrélation avec la matrice de corrélation.

D'après la figure 2 nous trouvons que les 2 variables ne sont pas corrélées entre elles.

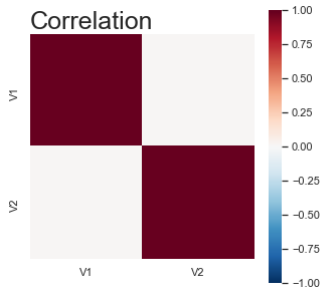



*Figure 12: matrice de corrélation du dataset aggregation*

4. La description statistique des variables et le boxplot

| | V1 | V2 |
|---|---|---|
| count | 788.000000 | 788.000000 |
| mean | 19.566815 | 14.171764 |
| std | 9.922042 | 8.089683 |
| min | 3.350000 | 1.950000 |
| 25% | 11.150000 | 7.037500 |
| 50% | 18.225000 | 11.725000 |
| 75% | 30.700000 | 21.962500 |
| max | 36.550000 | 29.150000 |

*Figure 13: Summary du dataset Aggregation*

Le tableau de la figure ci-dessus résume les différents comportements de chaque variable à savoir : « la moyenne, l'écart-type, min, max, le premier et troisième quartile et la médiane ».

Ces différentes valeurs sont illustrées ci-dessous.

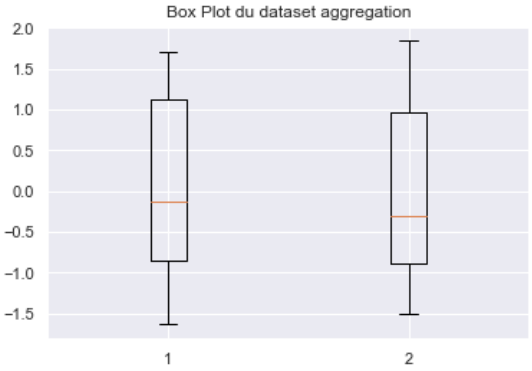



*Figure 14: Boxplot du dataset aggregation*

### *Comparaison des méthodes*

Dans cette partie nous avons comparé les différents modèles de classification cités dans le chapitre précédents afin de sortir avec le meilleur d'entre eux pour chaque dataset.

Pour faire la comparaison nous avons créé trois fonctions à savoir :

- Fonction retournant les meilleurs paramètres de chaque modèle.
- Fonction retournant les résultats de chaque modèle.
- Fonction retournant les meilleurs modèles avec un rapport de classification.

**Dataset spiral** :

Le résultat des différentes méthodes :

```
méthode / paramètre / score
('NB', {}, 0.36507936507936506)
('LR', {'penalty': 'l2', 'solver': 'newton-cg'}, 0.2857142857142857)
('LDA', {'solver': 'svd'}, 0.2857142857142857)
('KNN', {'metric': 'euclidean', 'n_neighbors': 2}, 1.0)
('CART', {'max_leaf_nodes': 9, 'min_samples_split': 2}, 0.7936507936507936)
('SVM', {'kernel': 'rbf'}, 1.0)
('QDA', {'reg_param': 0.2}, 0.36507936507936506)
('RF', {'n_estimators': 3}, 0.873015873015873)
```

*Figure 15: comparaison des méthodes " spiral"*

Choix de la meilleure méthode :

La meilleure méthode a été prise en faisant une comparaison entre les valeurs obtenues sur le test « nous avons pris le max » (sur les résultats des différentes méthodes).

```
les meilleurs méthodes sont : (Méthode , paramètre, score)
('KNN', {'metric': 'euclidean', 'n_neighbors': 2}, 1.0)
le rapport de classification est :
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        20
           2       1.00      1.00      1.00        21
           3       1.00      1.00      1.00        22

    accuracy                           1.00        63
   macro avg       1.00      1.00      1.00        63
weighted avg       1.00      1.00      1.00        63
```

*Figure 16: Le résultat du meilleur résultat " spiral"*

**<u>Dataset flame</u>** :

Le résultat des différentes méthodes :

```
méthode / paramètre / score
('NB', {}, 0.9791666666666666)
('LR', {'penalty': 'l2', 'solver': 'newton-cg'}, 0.8958333333333334)
('LDA', {'solver': 'svd'}, 0.8958333333333334)
('KNN', {'metric': 'euclidean', 'n_neighbors': 2}, 0.9791666666666666)
('CART', {'max_leaf_nodes': 6, 'min_samples_split': 2}, 0.9791666666666666)
('SVM', {'kernel': 'rbf'}, 1.0)
('QDA', {'reg_param': 0.1}, 0.9791666666666666)
('RF', {'n_estimators': 5}, 0.9791666666666666)
```

*Figure 17: comparaison des méthodes " flame"*

Choix de la meilleure méthode :

La meilleure méthode a été prise en faisant une comparaison entre les valeurs obtenues sur le test « nous avons pris le max »

```
les meilleurs méthodes sont : (Méthode , paramètre, score)
('SVM', {'kernel': 'rbf'}, 1.0)
le rapport de classification est :
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        18
           2       1.00      1.00      1.00        30

    accuracy                           1.00        48
   macro avg       1.00      1.00      1.00        48
weighted avg       1.00      1.00      1.00        48
```

*Figure 18: meilleur résultat " flame"*

**<u>Dataset Aggregation</u>**

Le résultat des différentes méthodes :

```
méthode / paramètre / score
('NB', {}, 0.9936708860759493)
('LR', {'penalty': 'l2', 'solver': 'newton-cg'}, 0.9936708860759493)
('LDA', {'solver': 'svd'}, 0.9873417721518988)
('KNN', {'metric': 'euclidean', 'n_neighbors': 3}, 0.9936708860759493)
('CART', {'max_leaf_nodes': 7, 'min_samples_split': 2}, 1.0)
('SVM', {'kernel': 'linear'}, 0.9873417721518988)
('QDA', {'reg_param': 0.1}, 0.9936708860759493)
('RF', {'n_estimators': 5}, 0.9936708860759493)
```

*Figure 19: comparaison des méthodes " aggregation"*

Choix de la meilleure méthode :

Les meilleures méthodes ont été prises en faisant une comparaison entre les valeurs obtenues sur le test « nous avons pris le max »

```
les meilleurs méthodes sont : (Méthode , paramètre, score)
('LR', {'penalty': 'none', 'solver': 'newton-cg'}, 1.0)
le rapport de classification est :
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        10
           2       1.00      1.00      1.00        32
           3       1.00      1.00      1.00        20
           4       1.00      1.00      1.00        59
           5       1.00      1.00      1.00         5
           6       1.00      1.00      1.00        28
           7       1.00      1.00      1.00         4

    accuracy                           1.00       158
   macro avg       1.00      1.00      1.00       158
weighted avg       1.00      1.00      1.00       158
```

*Figure 20: meilleur résultat " aggregation"*

## Partie 2 : Données réelles

Dans cette partie nous allons faire une étude comparative sur les données réelles « VisaPremier, Creditcard ».

### *VisaPremier*

Ce dataset est composé de 1073 individus, 47 variables tel qu'une est à expliquer et est binaire.

Traitement du dataset :

1) Encodage des variables qualitatives :

Données initiales

dataset initial

5]:

| | matricul | departem | ptvente | sexe | age | sitfamil | anciente | csp | codeqlt | nbimpaye | ... | mtbon | nbpaiecb | nbcb | nbcbptar | avtscpte | aveparfi | cartevp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 148009 | 31 | 1 | Shom | 51 | Fmar | 238 | Pcad | A | 0 | ... | 0 | 14 | 2 | 0 | 1303700 | 556967 | Cou |
| 1 | 442153 | 82 | 6 | Shom | 52 | Fmar | 270 | Pcad | A | 0 | ... | 19500000 | 5 | 2 | 0 | 19856243 | 133896 | Cou |
| 2 | 552427 | 97 | 1 | Shom | 58 | Fmar | 139 | Pcad | C | 0 | ... | 0 | 0 | 1 | 0 | 122745 | 0 | Cou |
| 3 | 556005 | 40 | 1 | Shom | 27 | Fcel | 99 | Psan | B | 0 | ... | 0 | 14 | 2 | 0 | 83224 | 0 | Cou |
| 4 | 556686 | 65 | 1 | Shom | 49 | Fsep | 89 | Pemp | A | 0 | ... | 0 | 11 | 3 | 1 | 494773 | 21423 | Cou |

5 rows × 48 columns

*Figure 21: dataset Creditcard*

Données après la transformation des variables qualitatives

dataset après la transformation des variables qualitatives

:

| | matricul | departem | ptvente | sexe | age | sitfamil | anciente | csp | codeqlt | nbimpaye | ... | mtbon | nbpaiecb | nbcb | nbcbptar | avtscpte | aveparfi | cartevp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 148009 | 8.0 | 1 | 1.0 | 51 | 3.0 | 238 | 2.0 | 1.0 | 0 | ... | 0 | 7.0 | 2 | 0 | 1303700 | 556967 | 1.0 |
| 1 | 442153 | 28.0 | 6 | 1.0 | 52 | 3.0 | 270 | 2.0 | 1.0 | 0 | ... | 19500000 | 43.0 | 2 | 0 | 19856243 | 133896 | 1.0 |
| 2 | 552427 | 33.0 | 1 | 1.0 | 58 | 3.0 | 139 | 2.0 | 3.0 | 0 | ... | 0 | 1.0 | 1 | 0 | 122745 | 0 | 1.0 |
| 3 | 556005 | 12.0 | 1 | 1.0 | 27 | 1.0 | 99 | 7.0 | 2.0 | 0 | ... | 0 | 7.0 | 2 | 0 | 83224 | 0 | 1.0 |
| 4 | 556686 | 21.0 | 1 | 1.0 | 49 | 4.0 | 89 | 3.0 | 1.0 | 0 | ... | 0 | 4.0 | 3 | 1 | 494773 | 21423 | 1.0 |

5 rows × 48 columns

*Figure 22: Dataset creditcard après transformation*

2) La suppression des variables insignifiantes :

En premier nous avons supprimé les variables :

- ✓ Variable à expliquer : Après avoir récupéré la variable « cartevpr » nous l'avons supprimé.
- ✓ Variable matricul : La variable « matricul » n'apporte aucune signification aux analyses car ce n'est qu'un identifiant de clients.

Puis, nous avons entamé l'étape de suppression des variables corrélées comme suit :

a) L'application de la matrice de corrélation :

Vu que les variables sont qualitatives, nous avons appliqué la corrélation afin de supprimer les doubles, tel que le seuil utilisé est « 0.95, -0.95 »

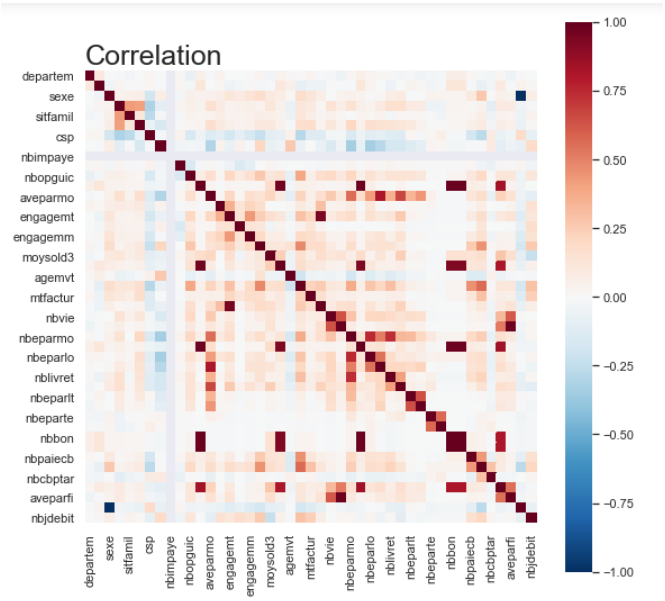



*Figure 23: Matrice de corrélation de Visapremier*

```
]: #recuperation de la variable à expliquer
   df = suppression(df_visa_tr)
   df.head()
```

]:

| | departem | ptvente | sexe | age | sitfamil | anciente | csp | codeqlt | mtrejet | nbopguic | ... | mtlivret | nbeparlt | mteparlt | nbeparte | mtepart |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 8.0 | 1 | 1.0 | 51 | 3.0 | 238 | 2.0 | 1.0 | 0 | 0 | ... | 181794 | 0 | 0 | 0 | |
| 1 | 28.0 | 6 | 1.0 | 52 | 3.0 | 270 | 2.0 | 1.0 | 0 | 4 | ... | 0 | 0 | 0 | 0 | |
| 2 | 33.0 | 1 | 1.0 | 58 | 3.0 | 139 | 2.0 | 3.0 | 0 | 0 | ... | 3402 | 0 | 0 | 0 | |
| 3 | 12.0 | 1 | 1.0 | 27 | 1.0 | 99 | 7.0 | 2.0 | 0 | 0 | ... | 30009 | 0 | 0 | 0 | |
| 4 | 21.0 | 1 | 1.0 | 49 | 4.0 | 89 | 3.0 | 1.0 | 0 | 0 | ... | 73351 | 0 | 0 | 0 | |

5 rows × 38 columns

*Figure 24: Le dataset après la suppression des variables*

Après avoir supprimé les variables, nous allons appliquer la matrice de corrélation afin de montrer que toutes les autres variables sont décorrélées.

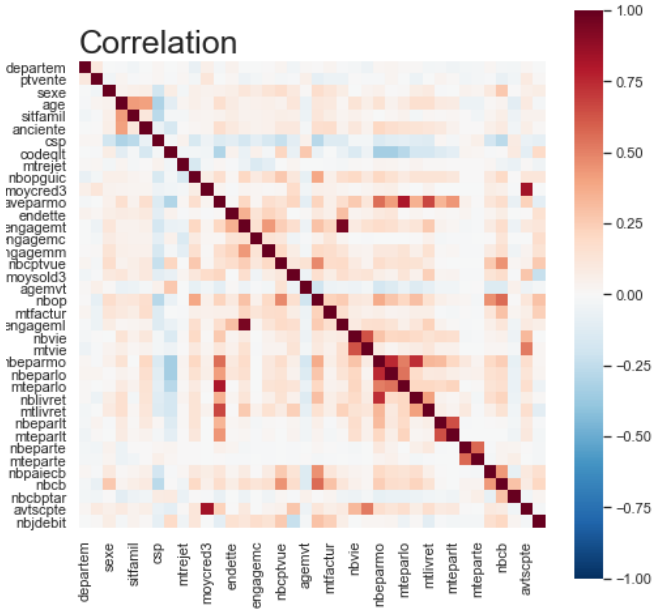



*Figure 25: Matrice de corrélation après la suppression*

3) La normalisation des données :

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -0.561090 | -0.611483 | 0.778645 | 0.711608 | 0.662635 | 0.518503 | -0.764923 | -0.881679 | 0.046063 | -0.538317 | ... | 2.650796 | -0.197012 | -0.130647 | -0.05295 |
| 1 | 2.173634 | 3.996113 | 0.778645 | 0.795645 | 0.662635 | 0.723642 | -0.764923 | -0.881679 | 0.046063 | 0.892305 | ... | -0.341355 | -0.197012 | -0.130647 | -0.05295 |
| 2 | 2.857316 | -0.611483 | 0.778645 | 1.299864 | 0.662635 | -0.116143 | -0.764923 | 0.589615 | 0.046063 | -0.538317 | ... | -0.285362 | -0.197012 | -0.130647 | -0.05295 |
| 3 | -0.014145 | -0.611483 | 0.778645 | -1.305268 | -1.090774 | -0.372566 | 1.811231 | -0.146032 | 0.046063 | -0.538317 | ... | 0.152564 | -0.197012 | -0.130647 | -0.05295 |
| 4 | 1.216481 | -0.611483 | 0.778645 | 0.543535 | 1.539339 | -0.436672 | -0.249693 | -0.881679 | 0.046063 | -0.538317 | ... | 0.865931 | -0.197012 | -0.130647 | -0.05295 |

5 rows × 38 columns

*Figure 26 : Le dataset après la normalisation*

4) Comparaison des modèles

Le résultat des différentes méthodes :

```
méthode / paramètre / score
('NB', {}, 0.8046511627906977)
('LR', {'penalty': 'l1', 'solver': 'liblinear'}, 0.8976744186046511)
('LDA', {'solver': 'svd'}, 0.8883720930232558)
('KNN', {'metric': 'manhattan', 'n_neighbors': 7}, 0.7906976744186046)
('CART', {'max_leaf_nodes': 19, 'min_samples_split': 2}, 0.8837209302325582)
('SVM', {'kernel': 'linear'}, 0.8976744186046511)
('QDA', {'reg_param': 0.1}, 0.8)
('RF', {'n_estimators': 8}, 0.9116279069767442)
```

*Figure 27: Les différents résultats de comparaison " Visapremier"*

Choix de la meilleure méthode :

La meilleure méthode a été prise en faisant une comparaison entre les valeurs obtenues sur le test « nous avons pris le max », comme sur les données synthétiques.

```
les meilleurs méthodes sont : (Méthode , paramètre, score)
('RF', {'n_estimators': 8}, 0.9116279069767442)
le rapport de classification est :
              precision    recall  f1-score   support

           0       0.91      0.96      0.93       140
           1       0.92      0.81      0.87        75

    accuracy                           0.91       215
   macro avg       0.92      0.89      0.90       215
weighted avg       0.91      0.91      0.91       215
```

*Figure 28: La meilleure méthode " Visapremier"*

## *Creditcard*

Ce dataset est composé de 284 807 transactions, 30 variables (V1 … V28) qui sont le résultat d'une ACP et deux autres (time et amount) et enfin une variable à expliquer composée de deux classes (fraude et non fraude).

### Traitement des données

En premier lieu nous avons vérifié s'il y a des transactions nulles dans chaque variable ainsi que leur type.

Puis nous allons faire l'étude préliminaire afin de compléter notre traitement en suivant les étapes suivantes :

1. La normalisation des données

| | Time | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | ... | V21 | V22 | V23 | V24 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -0.997065 | -1.359807 | -0.072781 | 2.536347 | 1.378155 | -0.338321 | 0.462388 | 0.239599 | 0.098698 | 0.363787 | ... | -0.018307 | 0.277838 | -0.110474 | 0.066928 | |
| 1 | -0.997065 | 1.191857 | 0.266151 | 0.166480 | 0.448154 | 0.060018 | -0.082361 | -0.078803 | 0.085102 | -0.255425 | ... | -0.225775 | -0.638672 | 0.101288 | -0.339846 | |
| 2 | -0.997053 | -1.358354 | -1.340163 | 1.773209 | 0.379780 | -0.503198 | 1.800499 | 0.791461 | 0.247676 | -1.514654 | ... | 0.247998 | 0.771679 | 0.909412 | -0.689281 | - |
| 3 | -0.997053 | -0.966272 | -0.185226 | 1.792993 | -0.863291 | -0.010309 | 1.247203 | 0.237609 | 0.377436 | -1.387024 | ... | -0.108300 | 0.005274 | -0.190321 | -1.175575 | |
| 4 | -0.997041 | -1.158233 | 0.877737 | 1.548718 | 0.403034 | -0.407193 | 0.095921 | 0.592941 | -0.270533 | 0.817739 | ... | -0.009431 | 0.798278 | -0.137458 | 0.141267 | - |

5 rows × 31 columns

*Figure 29: La normalisation Des données " Creditcard"*

2. L'identification des valeurs aberrantes et leur traitement :

| | Time | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 273482.000000 | 273482.000000 | 273482.000000 | 273482.000000 | 273482.000000 | 273482.000000 | 273482.000000 | 273482.000000 | 273482.000000 | 273482.00000 |
| mean | 0.118169 | 0.064800 | 0.130260 | 0.045680 | -0.018979 | 0.070783 | -0.036533 | -0.066520 | 0.018040 | 0.00533 |
| std | 0.558544 | 1.856582 | 1.370114 | 1.474985 | 1.399776 | 1.235051 | 1.272582 | 1.055922 | 1.167617 | 1.09053 |
| min | -0.997065 | -46.855047 | -47.429676 | -33.680984 | -5.683171 | -23.669726 | -23.496714 | -43.557242 | -50.943369 | -13.43406 |
| 25% | -0.358557 | -0.884097 | -0.513824 | -0.833243 | -0.856881 | -0.630053 | -0.779595 | -0.571644 | -0.199015 | -0.63019 |
| 50% | 0.000000 | 0.048895 | 0.104368 | 0.212868 | -0.030699 | -0.020194 | -0.295018 | 0.020150 | 0.028517 | -0.04719 |
| 75% | 0.641443 | 1.346344 | 0.832369 | 1.052662 | 0.722008 | 0.637256 | 0.350140 | 0.533126 | 0.334485 | 0.59402 |
| max | 1.033758 | 2.454930 | 22.057729 | 9.382558 | 13.129143 | 34.099309 | 11.607923 | 15.661716 | 20.007208 | 15.59499 |

8 rows × 31 columns

*Figure 30: Le summary du dataset " creditcard"*

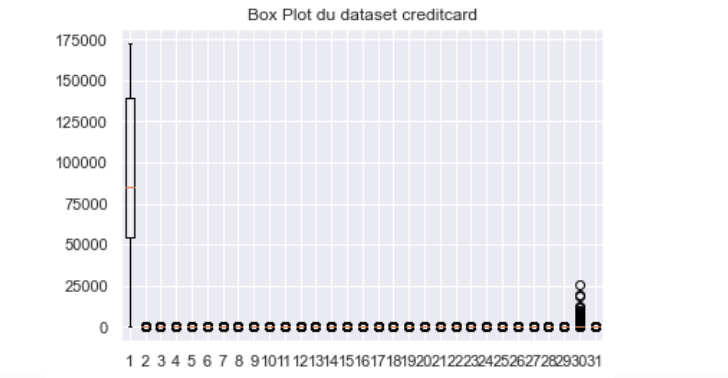



*Figure 31: Boxplot des variables " creditcard"*

D'après le graphe nous constatons que la variable 'amount' a des valeurs aberrantes et leur traitement a été fait tout en supprimant les individus appartenant à la classe 'No fraude' car celles de la classe 'Fraude' peuvent être très élevées de la norme.

La récupération des outliers a été faite en se basant sur le 1er et 3éme quantile récupérés du tableau de la figure précédente.

```
Nombre des outliers total : 11366
Nombre des outlies dans la classe Fraud: 41
Nombre des outliers dans la classe No fraude : 11325
```

*Figure 32: La récupération des outliers " creditcard"*

3. Vérifier l'équilibre des données

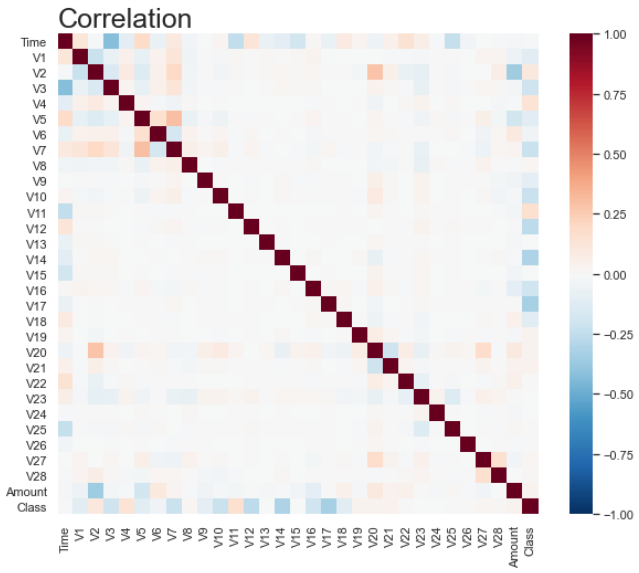



*Figure 33: Corrélation du dataset "creditcard"*

Depuis le graphe ci-dessus, nous constatons que les variables ne sont pas corrélées entres elles, ce qui nous amène à faire notre analyse dans toutes les variables du dataset.

***Partie comparaison***

En faisant la comparaison nous avons constaté qu'il y a un déséquilibre dans les données par rapport à la variable classe, ce qui nous a mené à utiliser des méthodes afin d'assurer l'équilibre à savoir : « Under-sampling et Over-sampling ».

```
No Fraud = 0 | Fraud = 1

0    99.820098
1     0.179902
Name: Class, dtype: float64
```

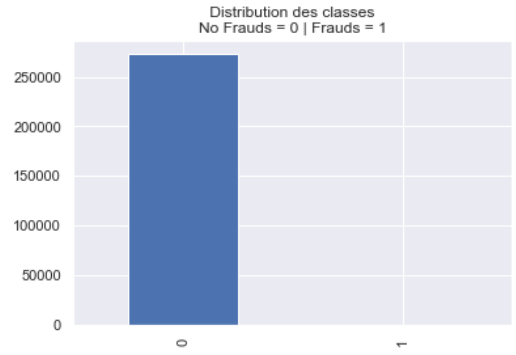



*Figure 34 : Le nombre d'individus des deux classes*

Après avoir effectué le découpage des données en tests et entrainements, nous allons appliquer les méthodes sur les données d'entrainement et les tester sur les données test.

4. Under-sampling :

Cette méthode se base sur un équilibre où nous devons prendre un taux de la classe supérieur qui doit être égal à celui de la classe inferieure.

```
1    290
0    290
Name: target, dtype: int64
AxesSubplot(0.125,0.125;0.775x0.755)
```

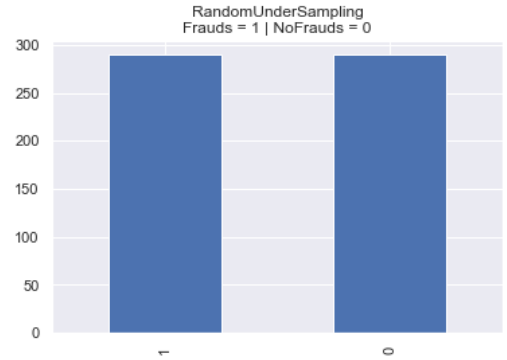



*Figure 35 : le nombre des individus après l'under-sampling*

L'un des inconvénients de cette méthode est de tomber dans le cas d'un under-fitting « avoir un mauvais résultat dans le train et test ».

Le résultat des méthodes :

```
méthode / paramètre / score
('NB', {}, 0.9747090011579012)
('LR', {'penalty': 'l2', 'solver': 'newton-cg'}, 0.9575842525443354)
('LDA', {'solver': 'svd'}, 0.9854591992199403)
('KNN', {'metric': 'manhattan', 'n_neighbors': 5}, 0.9827411786214882)
('CART', {'max_leaf_nodes': 5, 'min_samples_split': 2}, 0.9521725882137851)
('SVM', {'kernel': 'linear'}, 0.9496373941129868)
('QDA', {'reg_param': 0.1}, 0.9483576086294107)
('RF', {'n_estimators': 5}, 0.9453592540678896)
```

*Figure 36 : Résultat de comparaison des méthodes*

La meilleure méthode est :

```
les meilleurs méthodes sont : (Méthode , paramètre, score)
('LDA', {'solver': 'svd'}, 0.9854591992199403)
le rapport de classification est :
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     81910
           1       0.09      0.83      0.16       135

    accuracy                           0.99     82045
   macro avg       0.54      0.91      0.58     82045
weighted avg       1.00      0.99      0.99     82045
```

*Figure 37: Meilleure méthode de classification*

Après avoir choisi le bon modèle, nous devons vérifier son résultat en utilisant AUPRC (area under the precision-recall curve), tel que plus la surface est grande plus le modèle est bon.

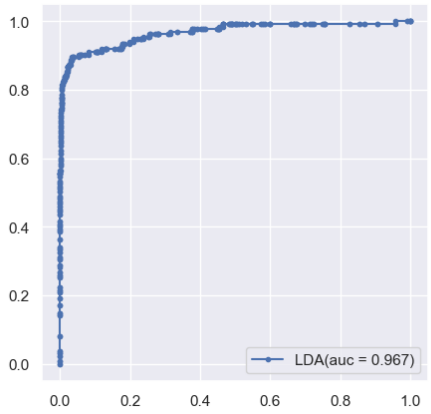



*Figure 38: l'auc de la meilleure méthode*

Conclusion nous trouvons qu'il n'y a pas d'under-fitting car le test et le train sont bons.

5. Over-sampling

Cette approche consiste à dupliquer les données de la classe inferieure afin qu'elles soient égales à celles de la classe supérieure.

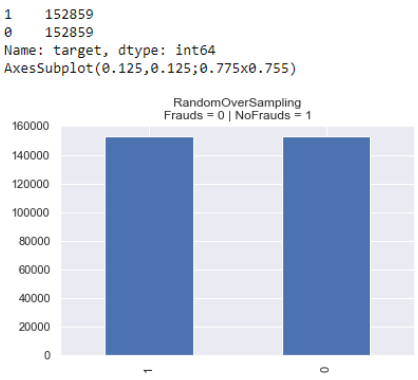



*Figure 39: Le nombre d'individus après l'over-sampling*

L'un des inconvénients de cette méthode est de tomber dans le cas d'un over-fitting « avoir un bon résultat dans le train et mauvais dans le test ».

Le résultat des méthodes

```
méthode / paramètre / score
('NB', {}, 0.977094651065608)
('LR', {'solver': 'newton-cg'}, 0.9758409945674885)
('LDA', {'solver': 'svd'}, 0.9900229837024656)
('CART', {'max_leaf_nodes': 9, 'min_samples_split': 2}, 0.9649498537400752)
('QDA', {'reg_param': 0.1}, 0.9657856247388216)
('RF', {'n_estimators': 4}, 0.9995037609694943)
('KNN', {'metric': 'euclidean', 'n_neighbors': 2}, 0.9993731717509402)
('SVM', {}, 0.9889521521103217)
```

*Figure 40: La comparaison des méthodes " over-sampling"*

La meilleure méthode

```
les meilleurs méthodes sont : (Méthode , paramètre, score)
('RF', {'n_estimators': 4}, 0.9995037609694943)
le rapport de classification est :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     38221
           1       0.93      0.78      0.85        67

    accuracy                           1.00     38288
   macro avg       0.96      0.89      0.92     38288
weighted avg       1.00      1.00      1.00     38288
```

*Figure 41: La meilleure méthode de classification " over-sampling"*

Après avoir choisi le bon modèle, nous devons vérifier son résultat en utilisant AUPRC, tel que plus la surface est grande plus le modèle est bon.

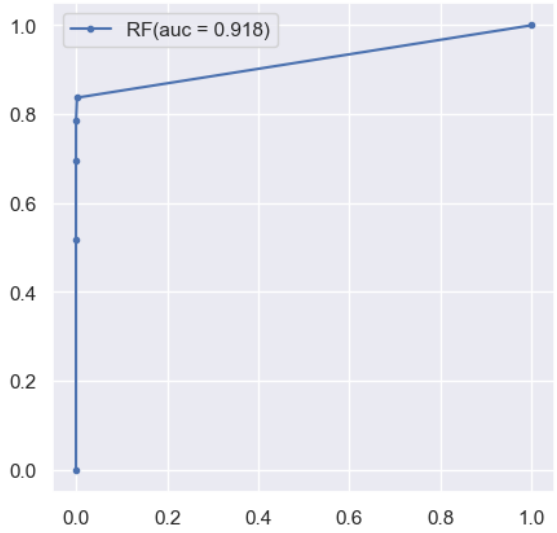



*Figure 42: l'auc de la meilleure méthode*

Comparaison

L'utilisation des deux approches d'équilibre de données a montré que les deux méthodes sont adaptées à notre data set mais avec des approches de classifications différentes. Par contre la méthode de sous-échantillonnage « under-sampling » a donné Le meilleur résultat avec la méthode LDA d'auc : 96.1%.

Remarque :

Vous trouverez le code sur le lien suivant :

https://github.com/Gharbimohamed/Projet_data_science

# Conclusion

Ce projet nous a permis, via le traitement de ces données, de se rendre compte et mieux comprendre les thèmes suivants :

- Le traitement de données : Avant d'appliquer des approches, nous devons traiter les données en regardant les valeurs nulles, les valeurs aberrantes ...etc. et leurs conséquences sur le résultat des approches.
- L'utilisation de la bonne approche : Chaque dataset a une ou plusieurs approches de classification.
- Il n'y a pas d'approche parfaite : Chaque approche a ses avantages et inconvénients, et chacune d'elles est adaptable à un dataset donné.