

LA DETECTION DE LA
FRAUDE A LA CARTE
BANCAIRE

Par
M. CIFTCI
Thèse entrant dans le cadre de
l'obtention du

Master of Science Machine Learning
for Business Intelligence

Aivancity Paris-Cachan

2022

Approuvé par

DOREID AMAR
Président du jury

Sujet : La détection de la fraude à la carte bancaire homologué
pour l'obtention du diplôme Master of Science Machine Learning for
Business Intelligence

Promotion : Octobre 2021-2022

AIVANCITY PARIS-CACHAN
EXTRAIT
LA DETECTION DE LA FRAUDE A LA CARTE BANCAIRE

Par M. CIFTCI

Président du jury

Professeur Doreid AMMAR
Professor of data science and computer science

La présente thèse traite la détection de la fraude à la carte bancaire ainsi que des solutions envisageables grâce au Machine Learning. Les difficultés du domaine se placent au sein même des données à exploiter, avec pour difficulté principale le déséquilibre des classes. Par ailleurs, la fraude ne représente qu'une infime partie des transactions bancaires et les modèles de prédiction classiques ne permettent pas de la prédire convenablement. Pour cela, il existe des solutions au problème de déséquilibre des classes. Dans ce sens, cette thèse a pour objectif d'exploiter les différentes méthodes de Machine Learning dans le cadre de la détection de fraude bancaire, en passant par le traitement du déséquilibre afin de permettre aux banques de mieux le combattre.

TABLE DES MATIERES

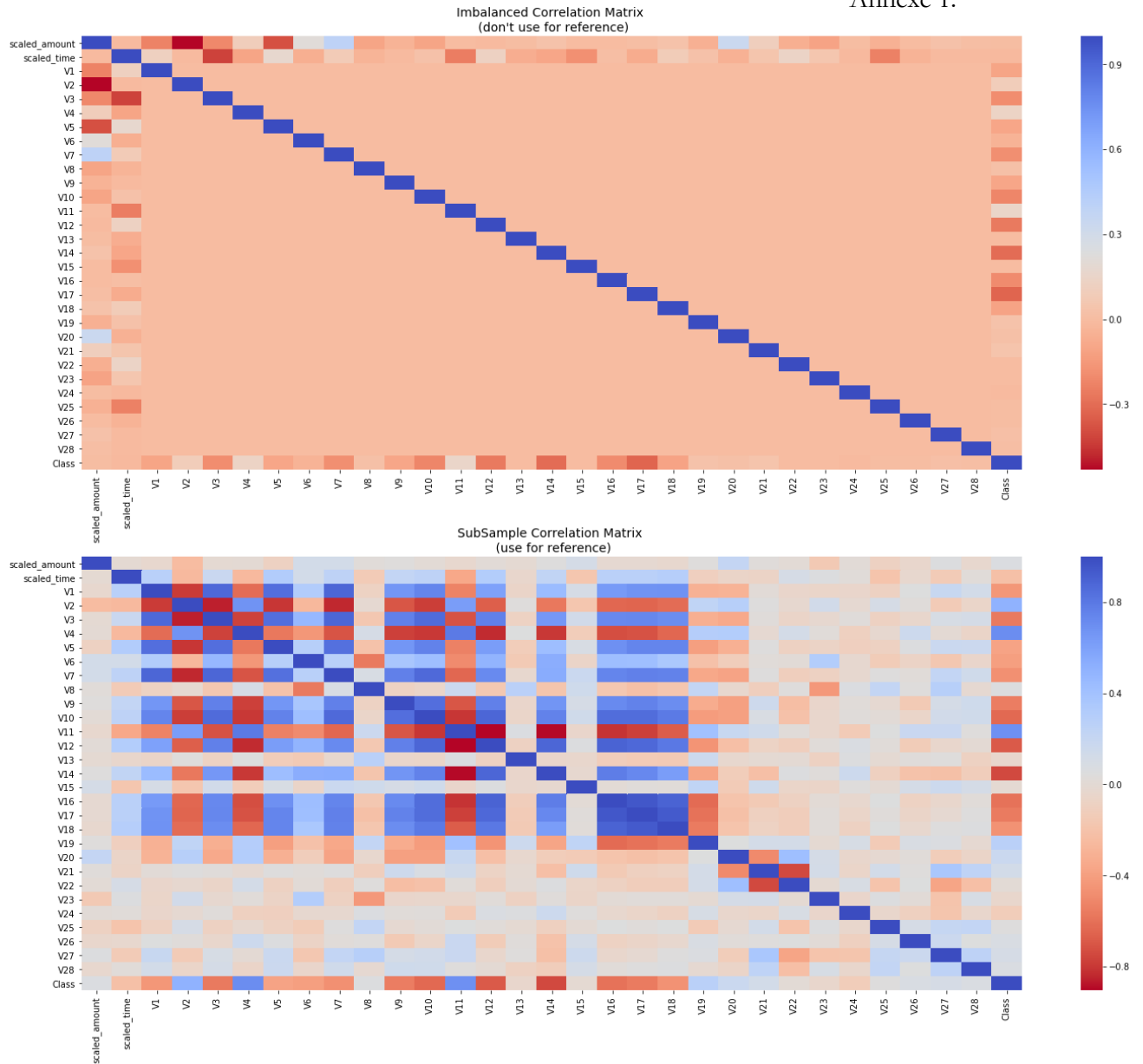
Remerciements	2
Liste des tableaux	3
Introduction	10
a. Identifier les usages et les applications de l'IA pour le projet	10
b. Identification de la problématique	11
c. Les objectifs de notre projet.....	12
Chapitre I : Comprendre nos données.....	14
a. La méthodologie adoptée.....	14
b. La compréhension de nos données	15
c. Le Skewness et le Kurtosis	17
Chapitre II : Mise à l'échelle et distribution	22
L'Undersampling et l'Oversampling.....	22
Chapitre III : Random Undersampling et Oversampling pour les ensembles de données déséquilibrés – Partie 1	25
a. La distribution et la corrélation.....	25
b. Analyser leurs implications éthiques.....	29
c. La détection d'anomalies	30
Chapitre IV : Random Undersampling et Oversampling pour les ensembles de données déséquilibrés – Partie 2	35
a. La réduction de la dimensionnalité et regroupement (t-SNE).....	35
b. Le « Classifiers ».....	36
c. Un regard approfondi sur la régression logistique	39
d. L'Oversampling avec la méthode « SMOTE ».....	42
Chapitre V : Phase de test.....	46
a. Matrice de confusion.....	49
Conclusion.....	52
Bibliographie	54

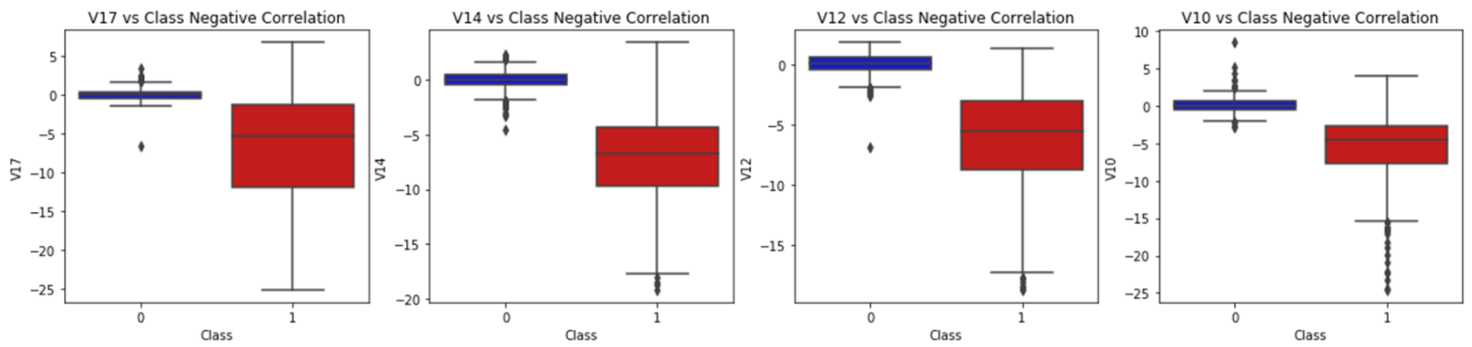
REMERCIEMENTS

Je tiens à exprimer sa sincère reconnaissance à M. Le Professeur Ammar pour son aide lors de la préparation de ce manuscrit. Je souhaite également remercier les enseignants dont la maîtrise des cours, tant sur le plan des besoins que des idées exposées, a été d'une grande aide pour la mise en route de ce projet. Enfin, merci également aux membres du comité étudiant pour leur soutien.

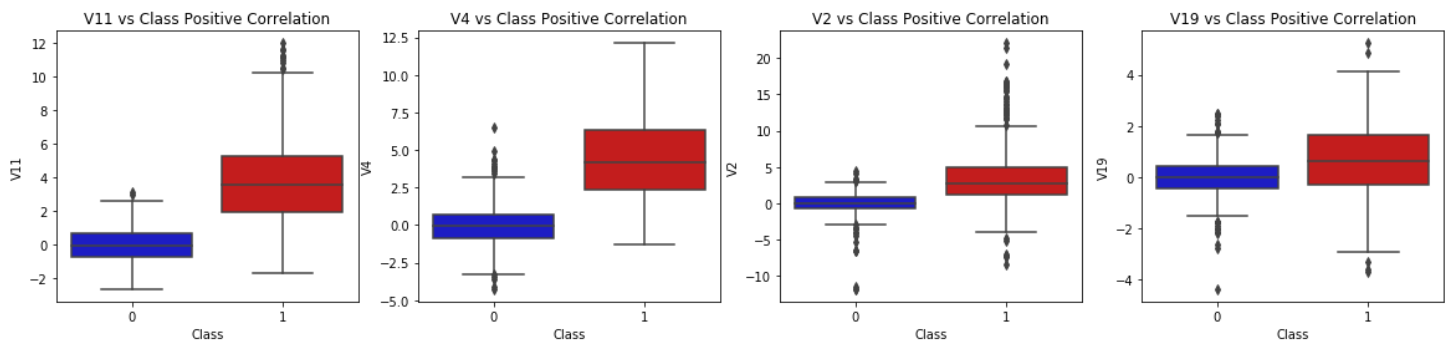
LISTE DES TABLEAUX

Annexe 1.

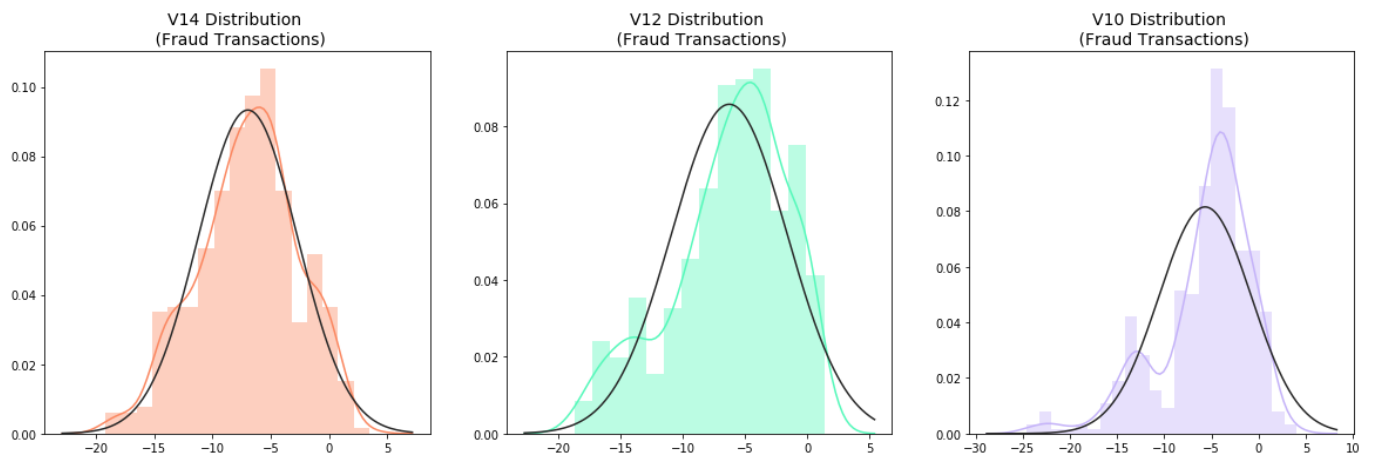




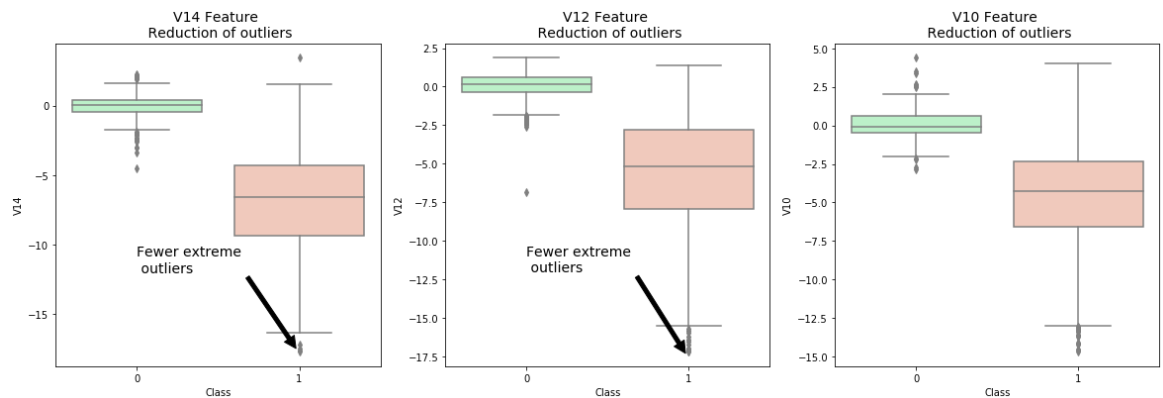
Annexe 2.



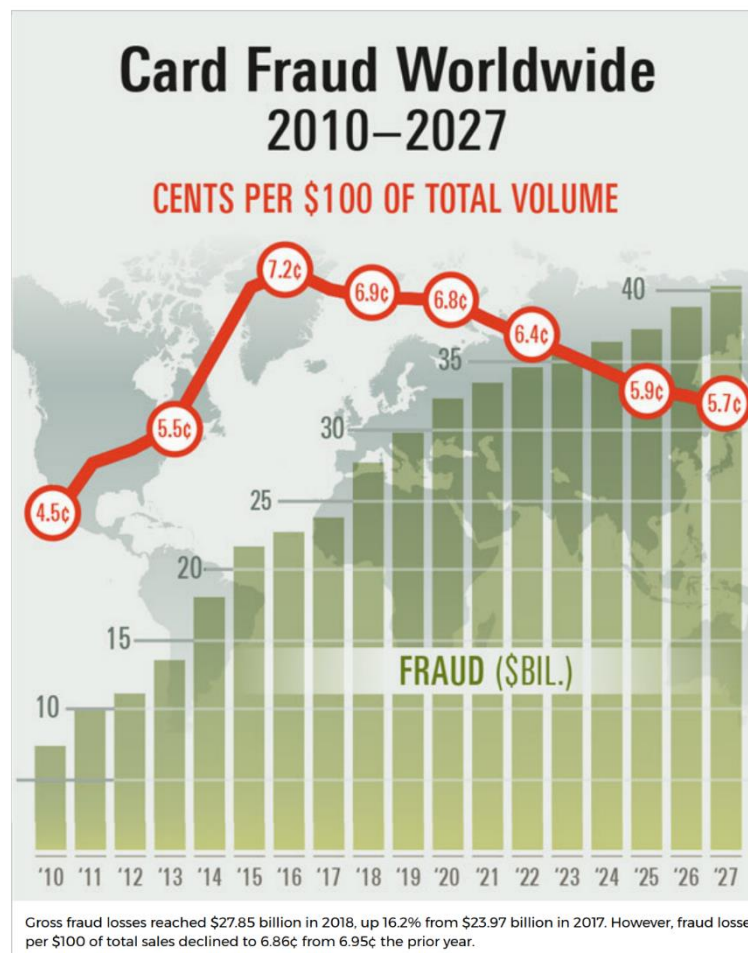
Annexe 3.



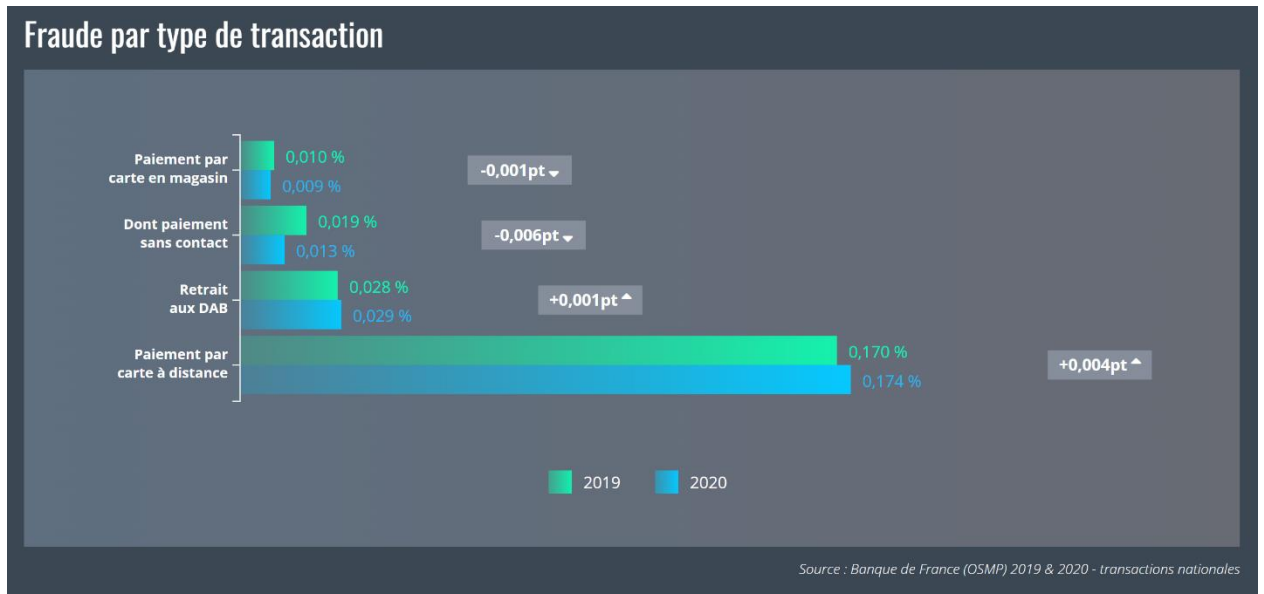
Annexe 4.



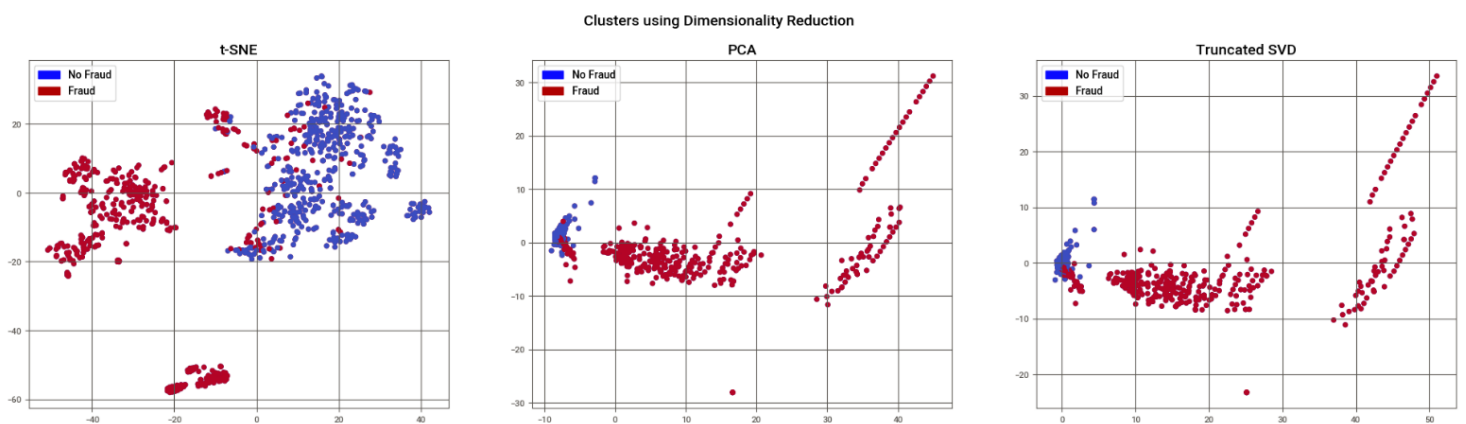
Annexe 5.



Annexe 6.

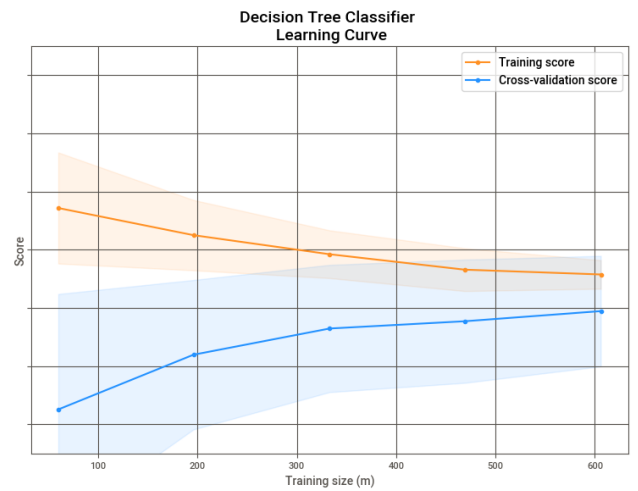
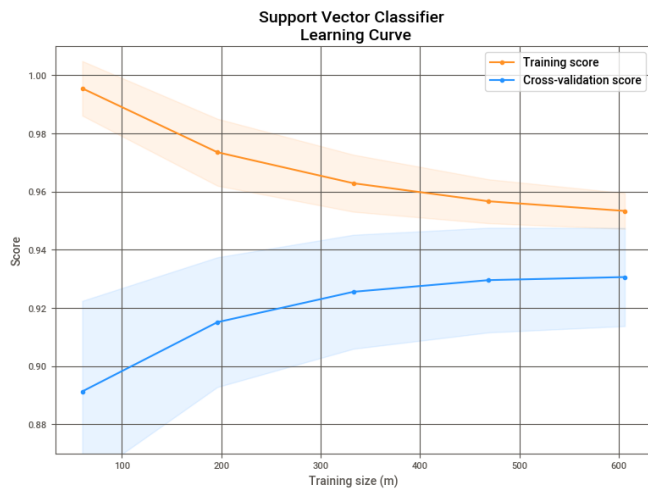
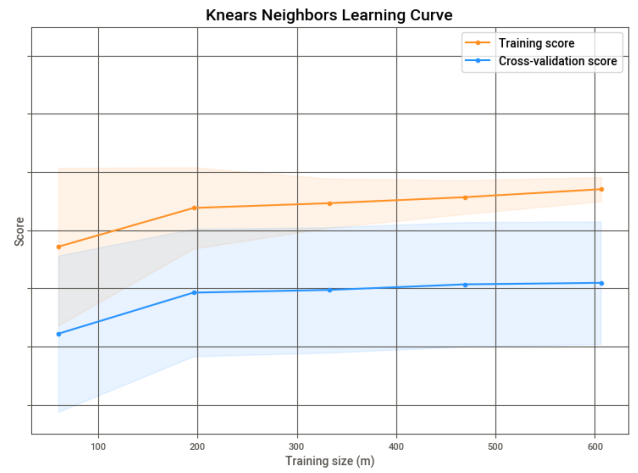
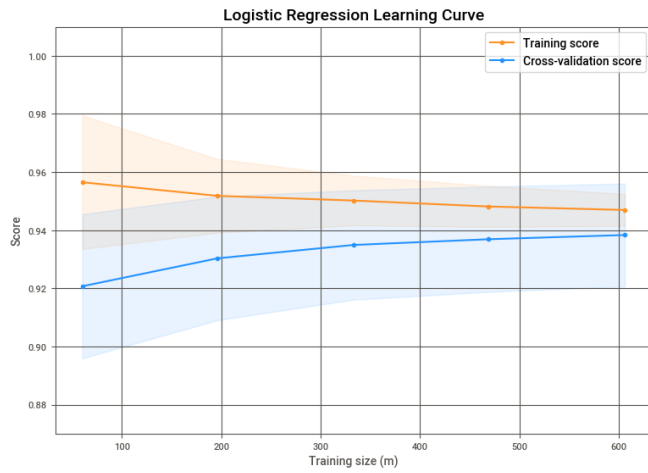


Annexe 7.

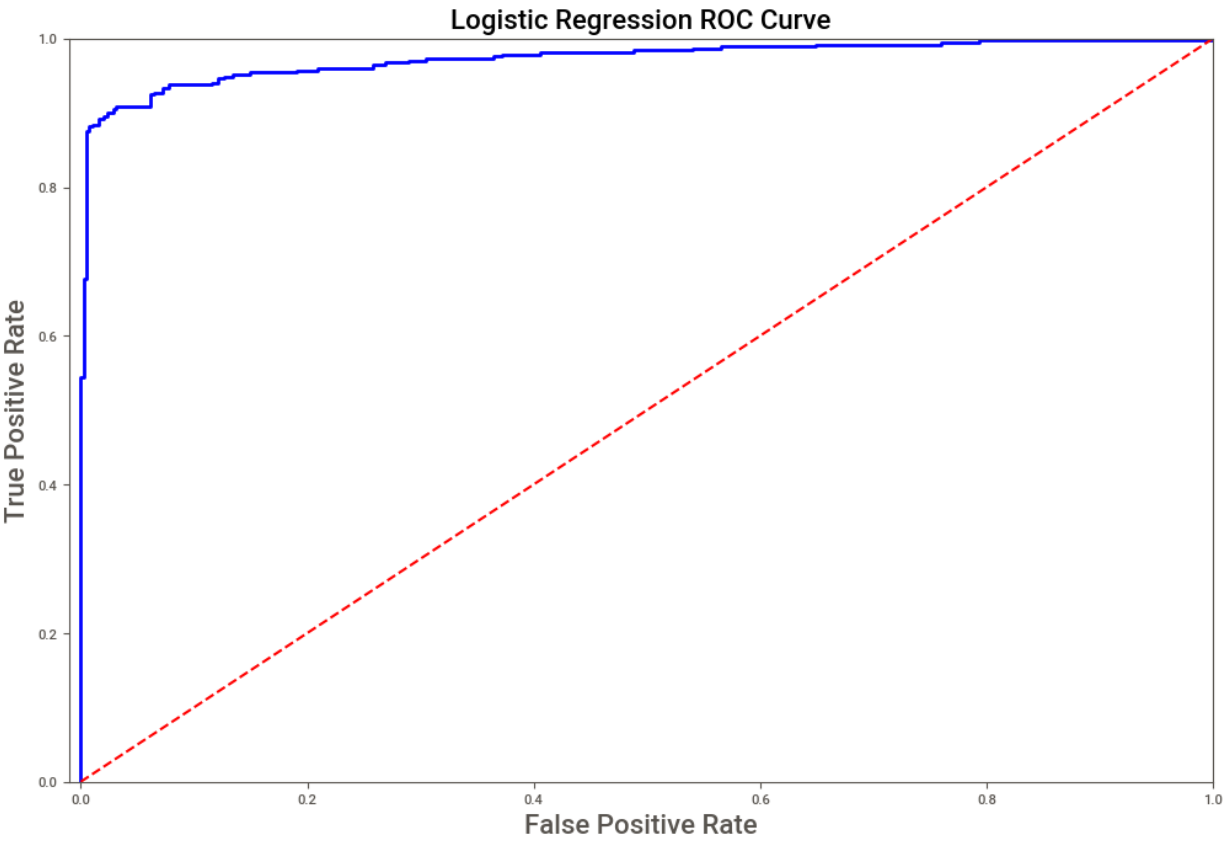
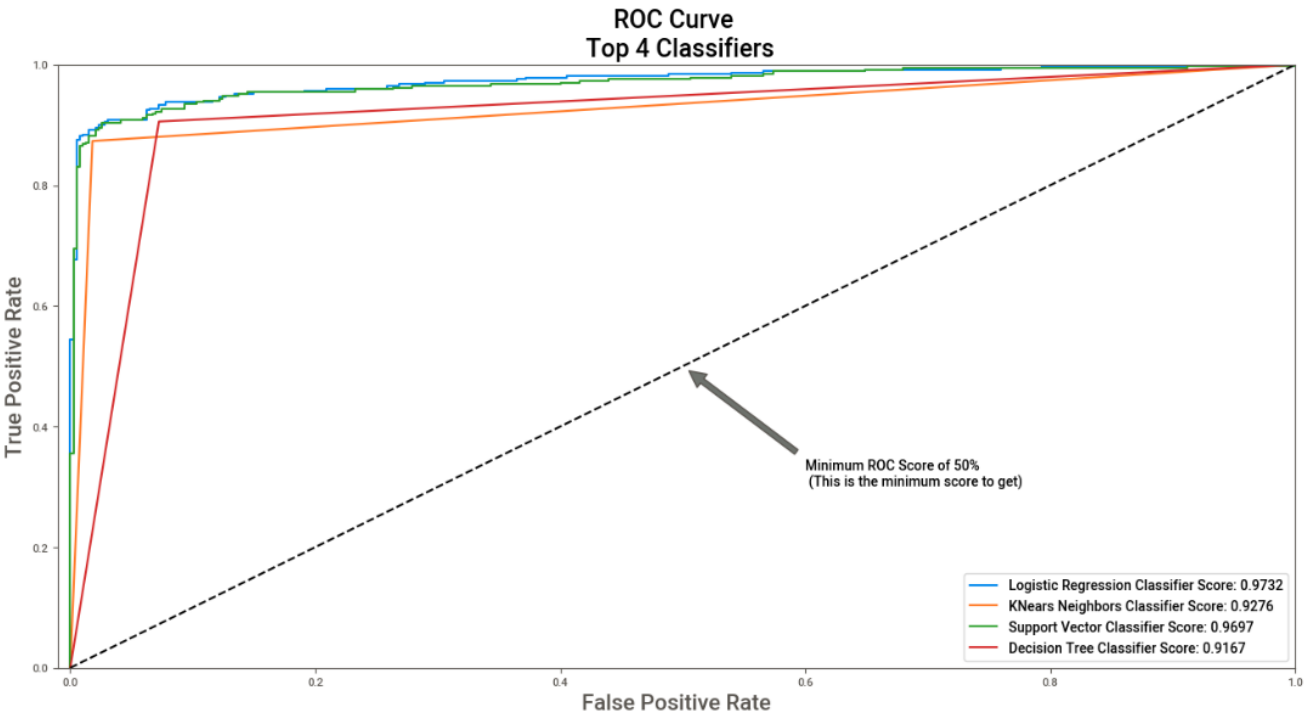


Annexe 8.

Annexe 9.

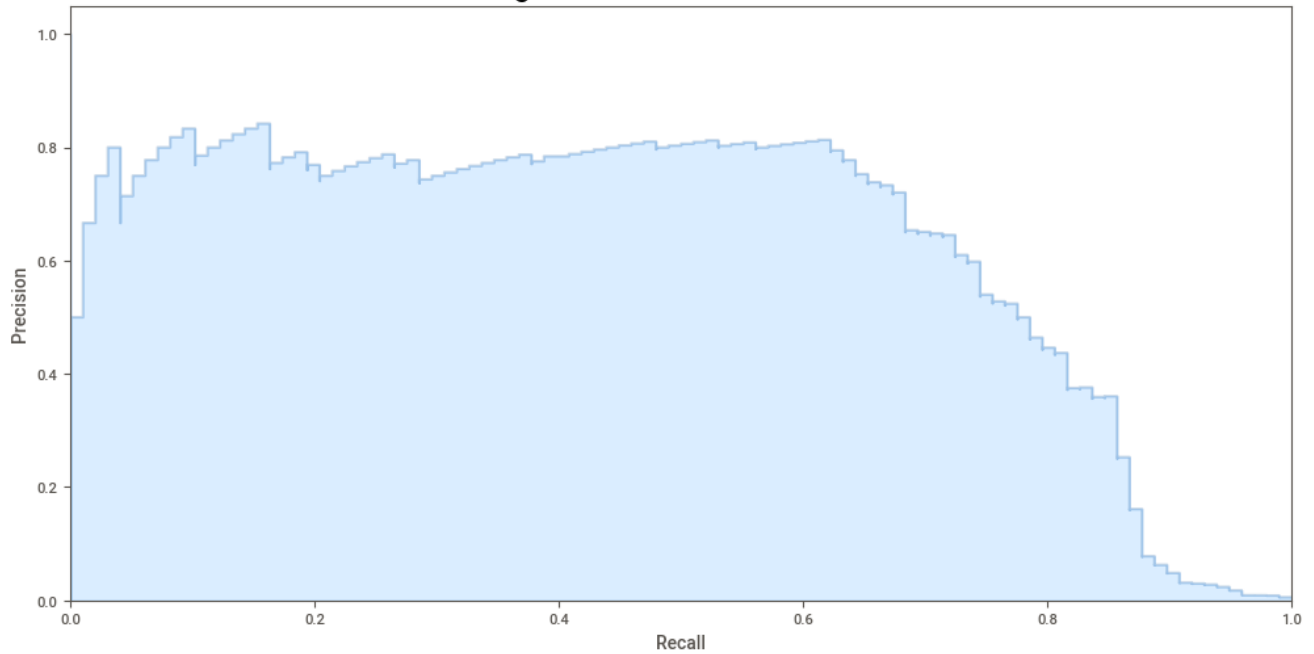


Annexe 10.



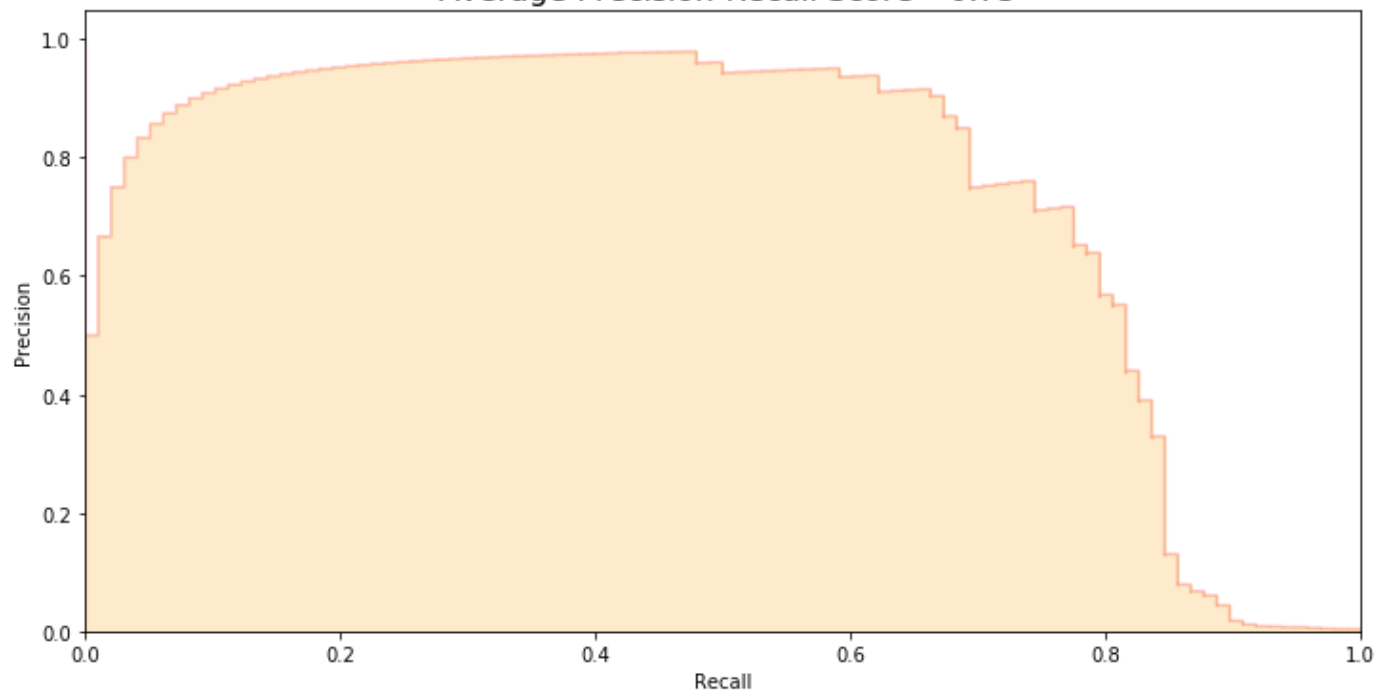
Annexe 11.

**UnderSampling Precision-Recall curve:
Average Precision-Recall Score =0.63**



Annexe 12.

**OverSampling Precision-Recall curve:
Average Precision-Recall Score =0.75**



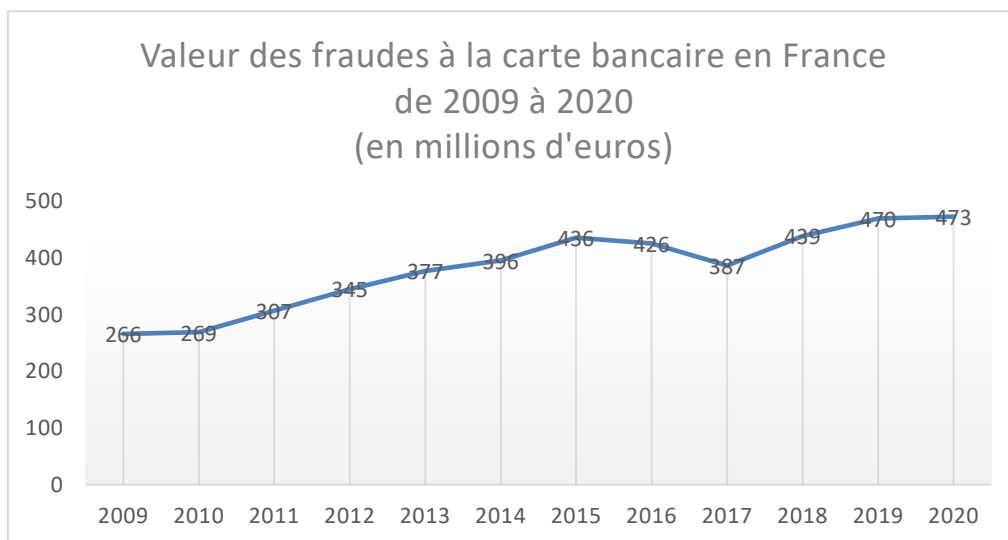
INTRODUCTION

LA FRAUDE A LA CARTE BANCAIRE

a. Identifier les usages et les applications de l'IA pour le projet

De nos jours, les techniques de fraude évoluent sans cesse, et la fraude sur internet coûte cher aux entreprises. Le comportement des techniques de fraudes change régulièrement. Les systèmes de détection de fraudes par carte de crédit doivent être proactifs pour créer une solution efficace contre le caractère changeant de ces types de fraudes. De plus, ces systèmes doivent être efficaces et en temps réel pour faciliter la prise de décision. En effet, les récents systèmes de détection de fraudes utilisent les techniques d'apprentissages supervisés et non supervisés.

Dans le domaine des paiements, la détection des transactions frauduleuses est également un champ d'application notable de l'analyse de données en temps réel permise par l'IA. La maîtrise de ces techniques pourrait déboucher, dans un deuxième temps, sur d'autres usages, aux objectifs plus commerciaux, liés à la meilleure compréhension des habitudes de consommation des clients¹.



1. Valeur des fraudes à la carte bancaire en France de 2009 à 2020 (en millions d'euros)².

En 2020, le taux de fraude des cartes bancaires pour les transactions réalisées en France est de 0,044% selon le rapport annuel de l'OSMP publié en juillet

¹ Intelligence artificielle : enjeux pour le secteur financier, Décembre 2018. ACPR

² Publié par Statista Research Department, 14 juillet 2021

2021³. Le graphique ci-dessous présente la variation de la valeur des fraudes à la carte bancaire en France entre 2009 et 2020, en millions d'euros. On constate ainsi qu'en 2020, le montant des fraudes à la carte bancaire en France s'élevait à environ 473 millions d'euros.

Les escroqueries et infractions économiques et financières ont fortement augmenté ces dernières années avec le développement des nouveaux moyens de paiement et l'utilisation frauduleuse des informations bancaires, grâce au détournement des réseaux de communication où circulent ces données.

b. Identification de la problématique

Pour cette institution financière, la mise en lumière des cas frauduleux constituait un véritable défi. Elle devait identifier les opérations malveillantes, mais aussi maintenir un service client de qualité en veillant à ce que la vigilance en matière de détection de la fraude n'altère le service client en signalant - et en bloquant - les transactions licites.

Cette institution financière souhaitait moderniser son système de détection de la fraude basé sur des règles et trouver un équilibre entre la surveillance et l'expérience client. Pour ce faire, elle a collaboré avec notre organisation pour mettre en œuvre une solution de détection de la fraude basée sur le Machine Learning qui tire parti d'un ensemble de réseaux neuronaux pour créer deux scores de fraude différents :

1. Un premier score de fraude est une évaluation de la probabilité qu'un compte soit en situation de fraude.
2. Un second score transactionnel est une évaluation de la probabilité que le paiement est licite et donc non frauduleux.

Cette approche a aidé l'institution financière à identifier correctement près d'un million de dollars de transactions mensuelles recensées à tort comme frauduleuses. Elle a également permis d'identifier 1,5 million de dollars par mois de fraude supplémentaire qui n'avait jamais été détectée. En plus d'améliorer considérablement la capacité de l'entreprise à détecter les fraudes, la solution analytique a augmenté de manière significative la satisfaction des clients. Et cela, grâce en améliorant le processus d'approbation des transactions tout en augmentant l'efficacité de la détection de la fraude, les frictions entre l'entreprise et ses clients ont été considérablement réduites.

³ Source : Banque de France (OSMP) 2019 & 2020 - transactions nationales

Aujourd'hui, la fraude par carte de crédit est commise sous deux formes : la fraude hors ligne ou la fraude via internet. La fraude hors ligne est commise en utilisant une carte physique volée. Dans ce cas, l'institution émettrice de la carte peut la verrouiller avant qu'elle ne soit utilisée de manière frauduleuse. En ce qui concerne la fraude en ligne, elle est commise via le Web, les achats par téléphone. Seules les informations de la carte sont nécessaires pour dérober de l'argent dans ce cas.

c. Les objectifs de notre projet

La difficulté dans la détection des fraudes est le manque de données réelles en raison de la sensibilité des données et du problème de confidentialité. Le deuxième problème est le traitement des données déséquilibrées (pour la classification) ou de distribution biaisée des données car le nombre de transactions frauduleuses est très inférieur à celui des transactions légitimes. Pour surmonter ce problème, des méthodes de suréchantillonnage sont utilisées pour augmenter le nombre de données à faible incidence dans les jeux de données.

Dans l'analyse de notre projet, nous allons utiliser différents modèles prédictifs pour voir dans quelle mesure ils sont précis pour détecter si une transaction est un paiement normal ou une fraude. Comme décrit dans le jeu de données, les caractéristiques sont **mises à l'échelle et les noms des caractéristiques ne sont pas indiqués pour des raisons de confidentialité**. Néanmoins, nous pouvons quand même analyser certains aspects importants de l'ensemble de données.



- ✚ Dans l'analyse de notre jeu de données, nous disposons de 284807 entrées et 31 colonnes
- ✚ Aucune donnée manquante dans le jeu de données
- ✚ Les données sont anonymisées (29 colonnes sur 31) pour des raisons de confidentialité (RGPD) ceux qui ne nous permettront pas de mettre

en lumière l'explicabilité des données, ni de comprendre l'importance des 29 colonnes anonymisées

✚ Nous avons 30 colonnes numériques et 1 colonne catégorique

Nos objectifs sont multiples :

- Comprendre la distribution des données qui nous ont été fournies.
- Créer un ratio de 50/50 de transactions "frauduleuses" et "non frauduleuses" dans le Dataframe via l'algorithme de Near-Miss ou un autre algorithme pour rééquilibrer le jeu de données.
- Déterminez les classificateurs que nous allons utiliser et choisir celui qui a la plus grande précision.
- Créez un réseau neuronal et comparez sa précision à celle de notre meilleur classificateur.
- Comprendre les erreurs courantes commises avec des ensembles de données déséquilibrés.

Dans ce projet, je vais vous présenter de manière constructive la démarche que j'ai adoptée pour mener à bien mon projet dans le cadre de « la détection de la fraude au niveau des cartes bancaires ».

Ce projet sera la phase finale d'une année riche en apprentissage où il m'a fallu investir beaucoup de temps et d'énergies afin de pouvoir être totalement autonome dans un contexte data.

Le code pourra m'être demandé par la suite lors de ma présentation orale. Un QR code sera mis à disposition pour y accéder directement.

- +Sélectionner un algorithme adapté au projet
- +Concevoir une plateforme(s) adaptée(s) au projet
- +Construire des indicateurs de performance et des alertes d'anomalies
- +Anticiper les risques liés à la sécurisation des données et aux comportements des modèles
- +Favoriser la maintenabilité à long terme du code et la pérennité des solutions déployées

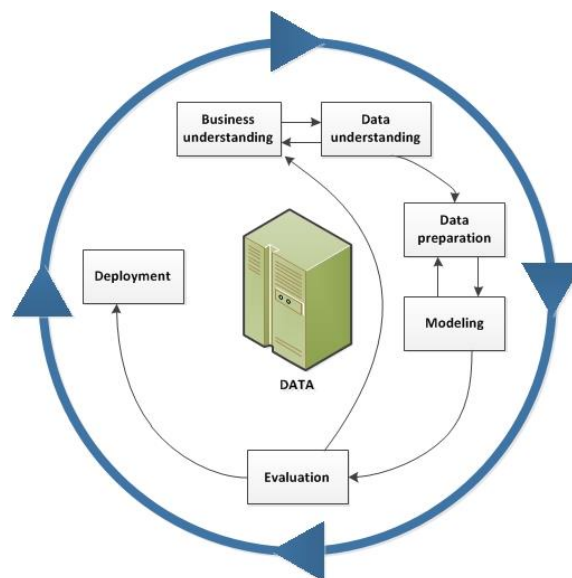
CHAPITRE 1

COMPRENDRE NOS DONNEES

a. La méthodologie adoptée

Avant de commencer par l'analyse des données, nous devons au préalable adopter une démarche analytique propre à la data science. La méthode CRISP (initialement connue comme CRISP-DM, qui signifie Cross-Industry Standard Process for Data Mining) a été au départ développée par IBM dans les années 60 pour réaliser les projets Datamining. Elle reste aujourd'hui la seule méthode utilisable efficacement pour tous les projets Data Science. C'est une méthode mise à l'épreuve sur le terrain permettant d'orienter nos travaux d'exploration de données.

- En tant que **méthodologie**, CRISP-DM comprend des descriptions des phases typiques d'un projet et des tâches comprises dans chaque phase, et une explication des relations entre ces tâches.
- En tant que **modèle de processus**, CRISP-DM offre un aperçu du cycle de vie de l'exploration de données.



2. Illustration CRISP-DM⁴

⁴ <https://www.ibm.com/docs/fr/spss-modeler/saas?topic=dm-crisp-help-overview>

Il comporte six phases séquentielles⁵:

- 1) Compréhension de l'entreprise - De quoi l'entreprise a-t-elle besoin ?
- 2) Compréhension des données – De quelles données disposons-nous / avons-nous besoin ? Est-ce propre ?
- 3) Préparation des données – Comment organisons-nous les données pour la modélisation ?
- 4) Modélisation – Quelles techniques de modélisation devrions-nous appliquer ?
- 5) Évaluation – Quel modèle répond le mieux aux objectifs commerciaux ?
- 6) Déploiement – Comment les parties prenantes accèdent-elles aux résultats ?

b. La compréhension de nos données

La première chose à faire est d'avoir une idée de notre base de nos données. Notons toutefois qu'à l'exception de la transaction et du montant, nous ne savons pas ce que sont les autres colonnes (pour des raisons de confidentialité). La seule chose que nous savons, c'est que les colonnes inconnues ont déjà été mises à l'échelle.

Résumé :

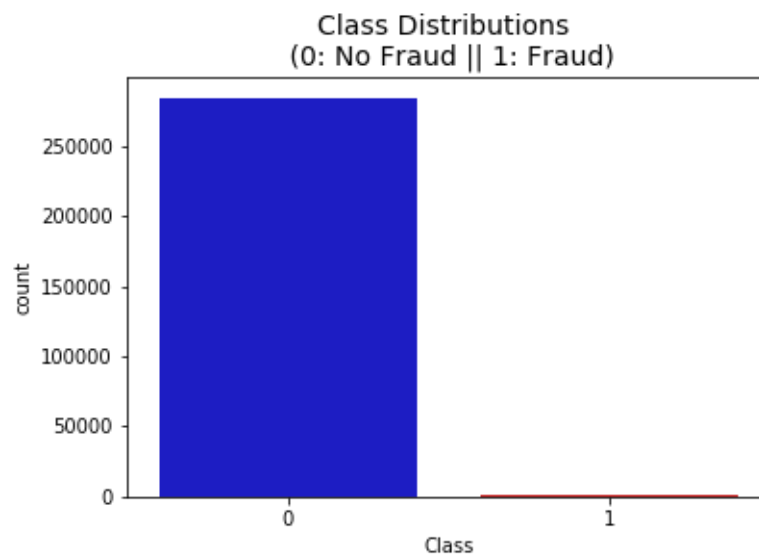
- Le montant de la transaction est relativement faible. La moyenne des transactions effectuées est d'environ 88\$ USD.
- Il n'y a pas de valeurs "Null", donc nous n'avons pas à travailler sur les moyens de remplacer les valeurs.
- La plupart des transactions étaient **non frauduleuses (99,83%)** du temps, tandis que les **transactions frauduleuses se produisent (0,17%)** exceptionnellement dans le jeu de données.

Caractéristiques techniques :

⁵ <https://www.datascience-pm.com/crisp-dm-2/>

Transformation PCA : La description des données indique que toutes les caractéristiques ont subi une transformation PCA (technique de réduction de la dimensionnalité) (à l'exception du temps et du montant)⁶. D'ailleurs, il s'agit des deux seules features n'ayant pas été anonymisées.

Mise à l'échelle : pour mettre en œuvre une transformation PCA, les caractéristiques doivent être préalablement mises à l'échelle. (Dans ce cas, toutes les caractéristiques V ont été mises à l'échelle ou du moins c'est ce que nous supposons que les personnes qui ont développé l'ensemble de données ont fait).

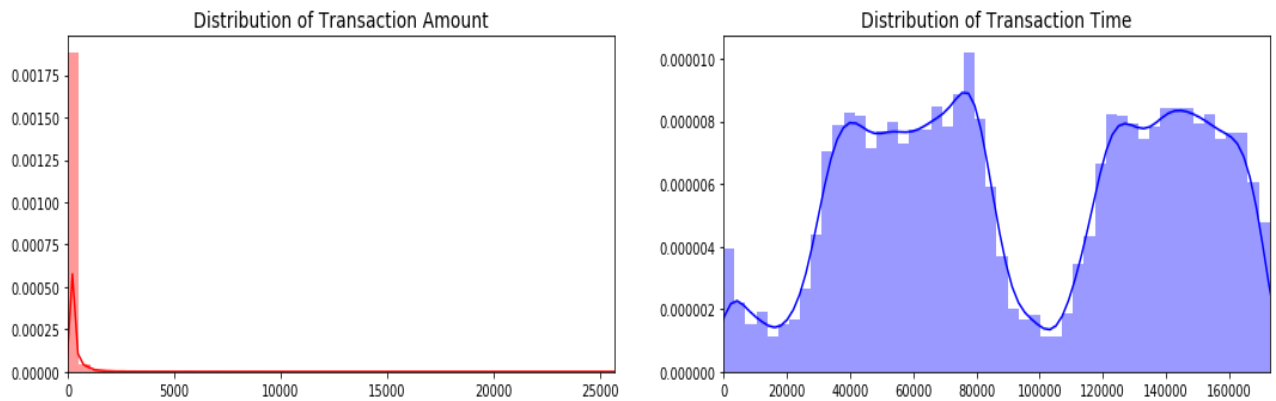


3. Illustration de la distribution de la Classe « No Fraud » et « Fraud ».

Analyse 1 : Nous remarquons à quel point notre jeu de données original est déséquilibré ! La plupart des transactions ne sont pas frauduleuses. Si nous utilisons cet ensemble de données comme base pour nos modèles prédictifs et nos analyses, nous risquons d'obtenir un grand nombre d'erreurs et nos algorithmes seront probablement sur-ajustés (overfitting) puisqu'ils "supposeront" que la plupart des transactions ne sont pas frauduleuses.

Mais nous ne voulons pas que notre modèle suppose, nous voulons que notre modèle détecte des modèles qui donnent des signes de fraude réels.

⁶ <https://towardsdatascience.com/using-principal-component-analysis-pca-for-machine-learning-b6e803f5bf1e>



4. Illustration de la distribution des transactions en montants et en temps.

De ce fait, en voyant les distributions, nous pouvons avoir une idée de l'asymétrie de ces caractéristiques, nous pouvons également voir d'autres distributions des autres caractéristiques. Il existe des techniques qui peuvent aider les distributions à être moins asymétriques et qui seront mises en œuvre par la suite.

c. Le Skewness et le Kurtosis

Le Skewness (coefficient d'asymétrie) et le Kurtosis (coefficient d'aplatissement) sont deux mesures de la Data Science pour analyser la disparité dans un dataset. Quand on traite des données, il est important de savoir analyser la disparité d'un dataset. La disparité, c'est quand les données d'un dataset sont déséquilibrées.

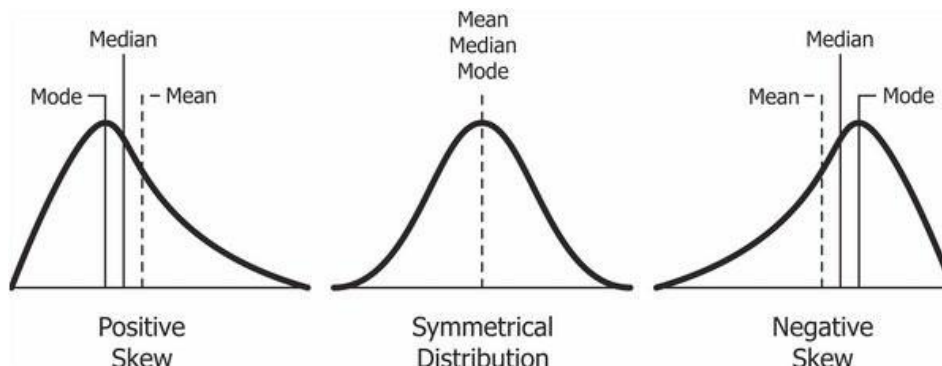
Le coefficient d'asymétrie ou le Skewness : c'est le degré de distorsion de la courbe en cloche symétrique ou de la distribution normale. Il mesure le manque de symétrie dans la distribution des données. Il différencie les valeurs extrêmes dans l'une par rapport à l'autre queue. Une distribution symétrique aura une asymétrie de 0.

Il existe deux types d'asymétrie : **positif** et **négatif**

1. La Skewness ou **le coefficient d'asymétrie correspond à une mesure de l'asymétrie de la distribution d'une variable aléatoire réelle.** Elle permet de calculer la symétrie de notre dataset. Un dataset est symétrique quand les données sont également réparties de part et d'autre de la moyenne. Lorsque la Skewness est égale 0, le dataset est symétrique. Mais cette mesure nous renseigne aussi sur le type d'asymétrie⁷.

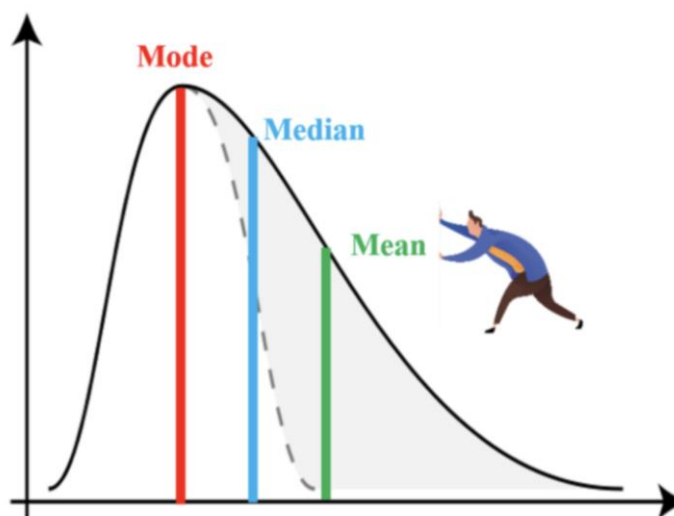
⁷ <https://ashington.medium.com/interpretation-of-measures-of-shape-skewness-kurtosis-b8b87c72c65>

Si la Skewness est supérieur à 0, alors le dataset est *skewed* sur la droite. C'est à dire que la majorité des données se trouvent sur la gauche et les outliers se trouvent sur la droite.



4. Illustration des Types d'asymétrie

Si la Skewness est inférieur à 0, alors le dataset est skewed sur la gauche. C'est à dire que la majorité des données se trouvent sur la droite et les outliers se trouvent sur la gauche.



5. Illustration asymétrie positive

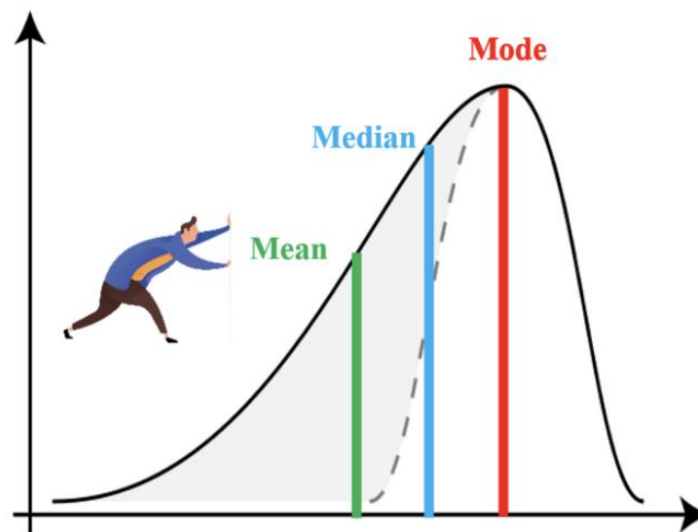
Cela signifie que la majorité de la distribution des données se situera sur le côté gauche de la moyenne, tandis que les valeurs de plage inférieures se trouveront sur le côté droit de la courbe.

La valeur de l'asymétrie pour une distribution positivement asymétrique est supérieure à zéro. Cela indique également la direction des valeurs aberrantes, qui se trouve du côté droit de la courbe dans la queue.

$$Q_3 - Q_2 > Q_2 - Q_1$$

6. Illustration asymétrie positive en termes de quartiles

Lorsque la distribution est symétrique, le coefficient de Skewness est nul. Lorsque la distribution possède une forte queue vers la droite, le coefficient de Skewness est positif (les + l'emportent). Lorsque la distribution possède une forte queue vers la gauche, le coefficient de Skewness est négatif (les - l'emportent)⁸.



7. Illustration asymétrie négative

Cela signifie que la majorité de la distribution des données se situera sur le côté droit de la moyenne, tandis que les valeurs de plage inférieures se trouveront sur le côté gauche de la courbe.

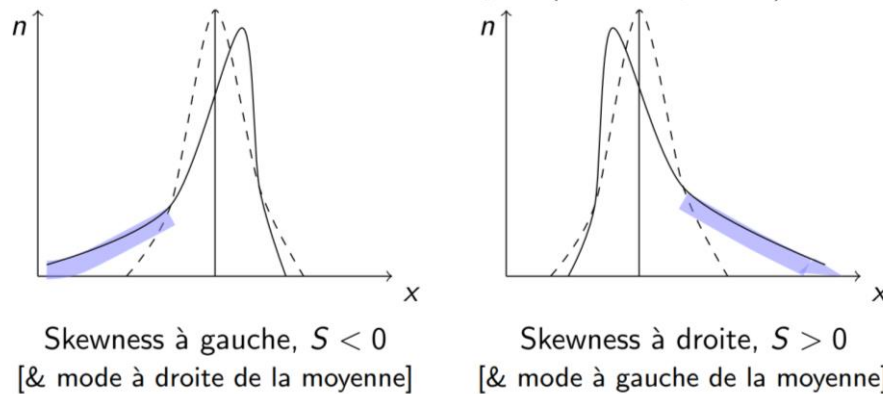
La valeur de l'asymétrie pour une distribution asymétriquement négative est inférieure à zéro. Cela m'indique également la direction des valeurs aberrantes, qui se trouve sur le côté gauche de la courbe dans la queue.

✚ Lorsque la distribution est symétrique, le coefficient de Skewness est nul.

⁸ Learning from Imbalanced Data Sets Hardcover – 1 Nov. 2018, by Alberto Fernández (Author), Salvador García (Author), Mikel Galar (Author), Ronaldo C. Prati (Author), Bartosz Krawczyk (Author), Francisco Herrera (Author), Springer; 1st ed. 2018 edition (1 Nov. 2018)

✚ Lorsque la distribution possède une forte queue vers la droite, le coefficient de Skewness est positif (les + l'emportent).

✚ Lorsque la distribution possède une forte queue vers la gauche, le coefficient de Skewness est négatif (les - l'emportent).



2. Le **Kurtosis** ou le **coefficient d'aplatissement** permet lui aussi de calculer la disparité d'un dataset mais différemment. Il permet de calculer l'aplatissement de notre courbe. Un dataset est plat quand les données sont également réparties. Il est bossu, lorsqu'on distingue un regroupement des données à un endroit. Attention à ne pas confondre avec la symétrie⁹.

Pour un dataset symétrique :

- Si le Kurtosis est supérieur à 0, alors le dataset est leptokurtique. C'est à dire que la majorité des données se situe à la moyenne, et que la bosse est accentuée.
- Si le Kurtosis est inférieur à 0, alors le dataset est platikurtique. C'est à dire que les données ont tendances à s'éloigner de la moyenne, et que la bosse est aplatie.

L'aplatissement concerne les queues de la distribution, et non les pics ou la planéité. Il est utilisé pour décrire les valeurs extrêmes dans l'une par rapport à l'autre queue. C'est en fait le mesure des valeurs aberrantes présent dans la distribution.

Un aplatissement élevé ou **un faible kurtosis** dans un ensemble de données est un indicateur que les données ont des queues lourdes ou des valeurs

⁹ Introduction à l'analyse statistique - Paramètres de dispersion d'une distribution, Université Paris-Dauphine, Arnold Chassagnon, 2010

aberrantes. S'il y a un kurtosis élevé, alors, nous devons rechercher pourquoi avons-nous tant de valeurs aberrantes. Cela indique beaucoup de choses, peut-être une mauvaise saisie de données ou d'autres choses. Si nous obtenons un faible aplatissement, nous devons également enquêter et réduire l'ensemble de données des résultats indésirables.

Analyse 2 : En voyant les distributions, nous pouvons avoir une idée de l'asymétrie de ces caractéristiques, nous pouvons également voir d'autres distributions des autres caractéristiques. Il existe des techniques qui peuvent aider les distributions à être moins asymétriques et qui seront implémentées par la suite.

Par conséquent, il existe une différence significative entre les tailles d'échantillon des deux classes dans un ensemble de données déséquilibré.

Problème avec l'ensemble de données déséquilibré :

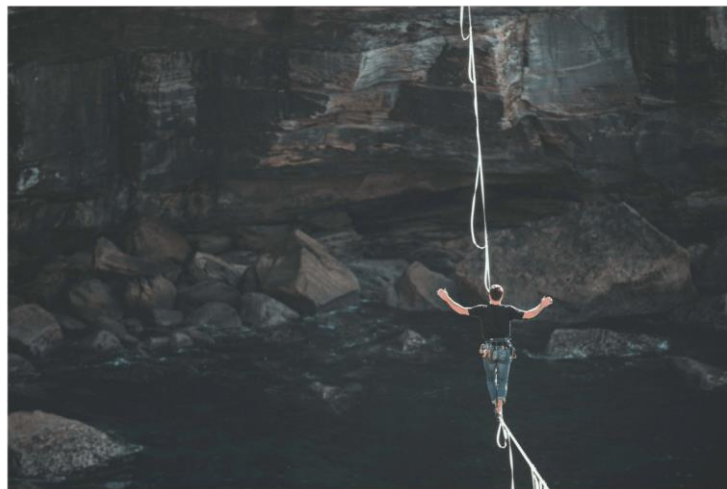
- Les algorithmes peuvent être biaisés en faveur de la classe majoritaire et ont donc tendance à prédire la sortie en tant que classe majoritaire.
- Les observations des classes minoritaires ressemblent à du bruit pour le modèle et sont ignorées par le modèle.
- Un ensemble de données déséquilibré donne un score de précision trompeur.

Nous allons étudier dans le chapitre 2, la mise à l'échelle des données et étudier ensemble la méthode de « sous-échantillon » de jeu de données afin d'avoir un nombre égal de cas frauduleux et non frauduleux.

CHAPITRE 2

MISE A L'ECHELLE ET DISTRIBUTION

Dans cette phase de notre projet, nous allons d'abord mettre à l'échelle les colonnes Time et Amount. Le temps et le montant doivent être mis à l'échelle comme les autres colonnes. D'autre part, nous devons également créer un sous-échantillon de la trame de données afin d'avoir un nombre égal de cas frauduleux et non frauduleux, ce qui aidera nos algorithmes à mieux comprendre les modèles qui déterminent si une transaction est frauduleuse ou non.



Qu'est-ce que l'Undersampling et l'Oversampling ?

Le **sous-échantillonnage** ou **Undersampling** consiste à rééquilibrer le jeu de données en diminuant le nombre d'instances de la classe majoritaire. Dans ce scénario, notre sous-échantillon sera un cadre de données avec un ratio 50/50 de transactions frauduleuses et non frauduleuses. Cela signifie que notre sous-échantillon aura le même nombre de transactions frauduleuses et non frauduleuses.

Ce type d'approches – appelées data-level solutions – se décline sous 2 formes principales : **Le sous-échantillonnage (undersampling)** et **Le sur-échantillonnage (oversampling)**



8. Illustration de l'Oversampling vs Undersampling¹⁰

Le **suréchantillonnage ou Oversampling** est approprié lorsque l'on ne dispose pas de suffisamment d'informations. Une classe est abondante, ou est en majorité, et l'autre est rare, ou elle est en minorité. Donc, cette technique consiste à rééquilibrer le jeu de données en augmentant artificiellement le nombre d'instances de la classe minoritaire (SMOTE). Pour définir la méthode SMOTE (*Synthetic Minority Oversampling Technique*), elle consiste à synthétiser des éléments pour la classe minoritaire, à partir de ceux qui existent déjà. Elle fonctionne en choisissant au hasard un point de la classe minoritaire et en calculant les k-voisins les plus proches pour ce point. Les points synthétiques sont ajoutés entre le point choisi et ses voisins. Mais on n'y viendra plus tard dans le cas de son utilisation.

Pourquoi faire de l'Undersampling ?

Au début de ce projet, nous avons vu que le cadre de données original était fortement déséquilibré ! L'utilisation du cadre de données original entraînera les problèmes suivants :

- **Surajustement ou overfitting** : Nos modèles de classification supposeront que dans la plupart des cas, il n'y a pas de fraude ! Ce que nous voulons pour notre modèle, c'est être certain qu'il y a fraude.
- **Corrélations erronées** : Bien que nous ne sachions pas ce que représentent les caractéristiques "V", il sera utile de comprendre comment chacune de ces caractéristiques influence le résultat (fraude ou non). En ayant un cadre de données déséquilibrées, nous ne sommes pas en mesure de voir les véritables corrélations entre la classe et les caractéristiques.

¹⁰ <https://www.mastersindatascience.org/learning/statistics-data-science/undersampling/>

Pour résumé : le « montant » et le « temps » sont les colonnes avec des valeurs mises à l'échelle.

Il y a 492 cas de fraude dans notre ensemble de données, nous pouvons donc obtenir de manière aléatoire 492 cas de non-fraude pour créer notre nouveau sous-ensemble de données. Nous concilions les 492 cas de fraude et de non fraude, créant ainsi un nouveau sous-échantillon (après la mise en place de la méthode du Random Undersampling).

Séparation des données (jeu de données original)

Avant de procéder à la technique de sous-échantillonnage aléatoire, nous devons séparer le cadre de données d'origine. Effectivement, même si nous divisons les données lorsque nous mettons en œuvre les techniques de sous-échantillonnage ou de sur-échantillonnage aléatoire, nous voulons tester nos modèles sur l'ensemble de **test original** et **non sur l'ensemble de test créé par l'une de ces techniques**. L'objectif principal est d'ajuster le modèle avec les données qui ont été sous-échantillonnées et suréchantillonnées (afin que nos modèles puissent détecter les modèles), et de le tester sur l'ensemble de test original.

Voici un exemple des étiquettes de formation et de test qui sont distribuées de manière identique par rapport à la technique de l'Undersampling.

No Frauds 99.83 % of the dataset
Frauds 0.17 % of the dataset

```
Train: [ 30473 30496 31002 ... 284804 284805 284806] Test: [ 0 1 2 .
.. 57017 57018 57019]
Train: [ 0 1 2 ... 284804 284805 284806] Test: [ 30473 30496 3100
2 ... 113964 113965 113966]
Train: [ 0 1 2 ... 284804 284805 284806] Test: [ 81609 82400 8305
3 ... 170946 170947 170948]
Train: [ 0 1 2 ... 284804 284805 284806] Test: [150654 150660 1506
61 ... 227866 227867 227868]
Train: [ 0 1 2 ... 227866 227867 227868] Test: [212516 212644 2130
92 ... 284804 284805 284806]
```


Label Distributions :

```
[0.99827076 - 0.00172924]
[0.99827952 - 0.00172048]
```

CHAPITRE 3

RANDOM UNDERSAMPLING ET OVERSAMPLING POUR LES ENSEMBLES DE DONNEES DESEQUILIBRES – PARTIE 1

a. La distribution et la corrélation

Dans cette phase du projet, nous allons mettre en œuvre un "sous-échantillonnage aléatoire ou le Random Undersampling" qui **consiste à rééquilibrer le jeu de données tout en diminuant le nombre d'instances de la classe majoritaire** afin d'obtenir un ensemble de données plus équilibré et d'éviter ainsi que nos modèles ne soient en situation d'overfitting.



Rappelons tout de même que lorsque le modèle s'entraîne trop longtemps sur des exemples de données ou lorsque le modèle est trop complexe, il peut commencer à **apprendre le « bruit »**, ou sur des informations non pertinentes, dans l'ensemble de données. Et donc, **le modèle mémorise le bruit et s'adapte trop étroitement à l'ensemble d'apprentissage**, le modèle va « **sur-ajusté** » et il est incapable de bien généraliser aux nouvelles données. Si un modèle ne peut pas bien généraliser à de nouvelles données, **il ne pourra pas effectuer les tâches de classification ou de prédiction pour lesquelles il était destiné.**

- a) La première chose à faire est de déterminer à quel point notre classe est déséquilibrée (en utilisant "value_counts()" sur la colonne classe pour déterminer le montant de chaque étiquette).
- b) Une fois que nous avons déterminé combien d'instances sont considérées comme des transactions frauduleuses (Fraude = "1"), nous devons amener les transactions non frauduleuses au même niveau que les transactions frauduleuses (en supposant que nous voulons un rapport 50/50), ce qui équivaut à **492 cas de fraude et 492 cas de**

transactions non frauduleuses. Après avoir mis en œuvre cette technique, nous avons un sous-échantillon de notre cadre de données avec un ratio de 50/50 en ce qui concerne nos classes.

- c) L'étape suivante consiste à mélanger les données pour voir si nos modèles peuvent maintenir une certaine précision à chaque fois que nous exécutons le code, ce qui n'est pas certains.

Le principal problème du "**sous-échantillonnage aléatoire**" est que nous courons le risque que nos modèles de classification ne soient pas aussi précis que nous le souhaiterions, car il y a une grande perte d'informations (**492 transactions non frauduleuses parmi 284 315 transactions non frauduleuses**).

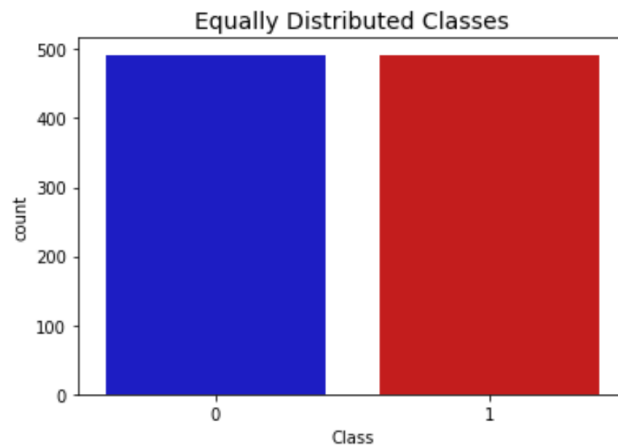
Autre limite que nous pourrions rencontrer, bien que ce soit une technique simple et efficace, les données sont supprimées sans se soucier de leur utilité ou de leur importance dans la détermination de la frontière de décision entre les classes. Cela signifie qu'il est possible, voire probable, **que des informations utiles soient supprimées**, ce qui est problématique dès lors que l'on souhaite étudier l'importance de nos données¹¹.

	scaled_amount	scaled_time
35719	0.368616	-0.545789
154670	1.145812	0.209084
59856	-0.125900	-0.418872
14170	1.089779	-0.698951
8842	-0.307413	-0.852912

9. Illustration du Dataframe équilibré

Maintenant que notre Dataframe est correctement équilibré, nous allons pouvoir continuer dans l'analyse et le data preprocessing. Dans l'illustration ci-dessous, nous voyons concrètement l'équilibre des deux classes que nous avons pu réaliser grâce au « Random Undersampling ».

¹¹ Learning from Imbalanced Data Sets 1st ed. 2018 Edition



L'étape d'après, sera l'étude de la matrice de corrélation qui sera essentielle à la compréhension même de notre jeu de données. Pour comprendre ce qu'est une matrice de corrélation, voici une définition succincte¹² : « Une **matrice de corrélation** est utilisée pour évaluer la **dépendance** entre plusieurs variables en même temps. Le résultat est une table contenant les **coefficients de corrélation** entre chaque variable et les autres ». Ce tableau va tout simplement indiquer la corrélation entre toutes les paires de valeurs possibles dans un format matriciel.

Il existe différentes méthodes de **tests de corrélation** : Le **test de corrélation de Pearson**, la **corrélation de Kendall** et celle de **Spearman** qui sont des tests basés sur le rang.

Coefficient de corrélation

Avant de commencer la matrice de corrélation, on se doit de comprendre ce qu'est le coefficient de corrélation et ses caractéristiques. Les coefficients de corrélation sont utilisés pour vérifier la force d'une relation entre deux variables. Le coefficient de corrélation linéaire r donne une mesure de l'intensité et du sens de la relation linéaire entre deux variables.

- Plus le coefficient est proche de 1, plus la relation linéaire positive entre les variables est forte.
- Plus le coefficient est proche de -1 , plus la relation linéaire négative entre les variables est forte.

¹² Statistical Tools For High-Throughput Data Analysis

- Plus le coefficient est proche de 0, plus la relation linéaire entre les variables est faible.

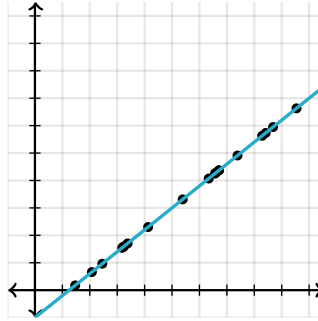


Illustration ici d'une corrélation positive parfaite entre les deux variables, $r=1$

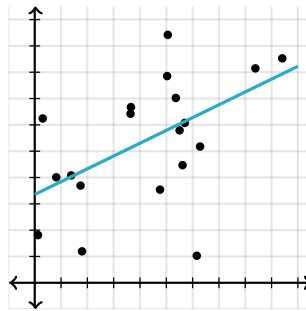


Illustration ici d'une corrélation positive faible entre les deux variables, $r=0,5$

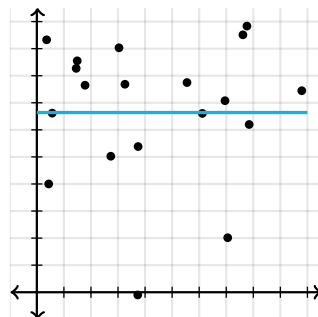


Illustration ici d'une absence totale de corrélation entre les deux variables, $r=0$

Résumons ce que nous voyons dans la matrice de corrélation en annexe 1 :

Corrélations négatives : V17, V14, V12 et V10 sont négativement corrélés. On voit bien que plus ces valeurs sont faibles, plus le résultat final sera probablement une transaction frauduleuse.

Corrélations positives : V2, V4, V11 et V19 sont positivement corrélés. Notez que plus ces valeurs sont élevées, plus il est probable que le résultat final soit une transaction frauduleuse.

b. Analyser leurs implications éthiques

L'intelligibilité des algorithmes en général et particulièrement ceux de l'intelligence artificielle (IA) est devenue un critère prépondérant en raison du rapport Villani en France et à la mise en place du RGPD en Europe¹³.

Le problème que nous rencontrons dans ce jeu de données et dans la plupart des données bancaires, c'est qu'ils sont anonymisés et qu'on ne peut pas donc donner d'explicabilité pour les différentes features présentes dans le jeu de données. Par ailleurs, le processus d'anonymisation vise à éliminer toute possibilité de réidentification : **il implique donc une nécessaire perte de qualité des données**¹⁴.

Comme vérifier l'efficacité de l'anonymisation ?

Dans leur avis de 2014¹⁵, les autorités de protection des données européennes définissent trois critères qui permettent de s'assurer qu'un jeu de données est véritablement anonyme :

1. **La non-individualisation** : il ne doit pas être possible d'isoler un individu dans le jeu de données
2. **La non-corrélation** : il ne doit pas être possible de relier entre eux des ensembles de données distincts concernant un même individu
3. **La non-inférence** : il ne doit pas être possible de déduire de façon quasi certaine de nouvelles informations sur un individu.

Ce questionnaire aide à déterminer le procédé d'anonymisation le plus pertinent, c'est-à-dire l'enchaînement des techniques d'anonymisation à mettre en place qui peuvent être regroupées en deux familles : la randomisation et la généralisation.

- La **randomisation** consiste à modifier les attributs dans un jeu de données de telle sorte qu'elles soient moins précises, tout en conservant

¹³ Z. Lipton, The Mythos of Model Interpretability, 2017

¹⁴ <https://www.cnil.fr/fr/lanonymisation-des-donnees-un-traitement-cle-pour-lopen-data>

¹⁵ <https://www.cnil.fr/fr/le-g29-publie-un-avis-sur-les-techniques-danonymisation>

la répartition globale. Cette technique permet de protéger le jeu de données du risque *d'inférence*.

- La **généralisation** permet de généraliser les attributs du jeu de données en modifiant leur échelle ou leur ordre de grandeur afin de s'assurer qu'ils soient communs à un ensemble de personnes. Cette technique permet d'éviter l'individualisation d'un jeu de données. Elle limite également les possibles corrélations du jeu de données avec d'autres.

L'importance de l'explicabilité



Transparence
des modèles



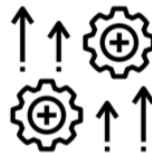
Générer de la
confiance



Respecter la
réglementation



Comprendre
la prise des
décisions



Améliorer la
performance
des modèles



Réduire les
biais éthiques
et moraux

Illustration sur la phase de l'explicabilité des données¹⁶

c. La détection d'anomalies

Notre objectif principal dans cette section est d'éliminer les "**valeurs aberrantes ou outliers**" des caractéristiques qui ont une forte corrélation avec nos classes. Cela aura un impact positif sur la précision de nos modèles et c'est ce que nous souhaitons. Mais d'abord, essayons de comprendre ce qu'est une valeur aberrante. Une valeur aberrante est une valeur qui s'écarte fortement des valeurs des autres observations, anormalement faible ou

¹⁶ H. Dam, T. Tran, A. Ghose, Explainable Software Analytics, Feb 2018

élevée¹⁷. Nous définirons généralement les valeurs aberrantes comme des échantillons exceptionnellement éloignés du courant dominant des données¹⁸.

Méthode de l'intervalle Interquartile Range (IQR)

Mais justement comment traiter ses données aberrantes. Il faut comprendre une chose, c'est que toutes les données ne sont pas normales ou suffisamment normales pour être traitées comme étant tirées d'une distribution gaussienne. Une bonne statistique pour résumer un échantillon de données de distribution non gaussienne est l'intervalle interquartile, ou *IQR*.

Une valeur aberrante dans une distribution est un nombre qui est plus d'une fois et demie la longueur de la boîte à l'écart du quartile inférieur ou supérieur. Globalement, si un nombre est inférieur à $Q1 - 1,5 \times IQR$ ou supérieur à $Q3 + 1,5 \times IQR$, alors c'est une valeur aberrante¹⁹.

→ $Q1 - 1,5 \times \text{Ecart interquartile ou } IQR$

→ $Q3 + 1,5 \times \text{Ecart interquartile ou } IQR$

L'IQR est calculé comme la différence entre le 75e et le 25e centile des données et définit la boîte dans un tracé en boîte et à moustaches.

Dans les statistiques descriptives, l'écart interquartile (IQR) est une mesure de dispersion statistique, étant égal à la différence entre le troisième quartile ($Q3$) et le premier quartile ($Q1$), c'est-à-dire, $IQR = Q3 - Q1$.

Compromis pour la suppression des valeurs aberrantes

Nous devons également faire attention à la distance à laquelle nous voulons que le seuil de suppression des valeurs aberrantes soit atteint. Nous déterminons le seuil en multipliant un nombre (ex : 1,5) par (l'écart interquartile). **Plus ce seuil est élevé, moins les valeurs aberrantes seront détectées** (en multipliant par un nombre plus élevé, par exemple 3), et **plus ce seuil est bas, plus les valeurs aberrantes seront détectées**.

¹⁷ <https://fr.khanacademy.org/math/be-4eme-seconaire2/x213a6fc6f6c9e122:statistiques-1/x213a6fc6f6c9e122:diagramme-en-boite-et-ecart-interquartil/a/identifying-outliers-iqr-rule>

¹⁸ Applied Predictive Modeling 1st ed. 2013, Corr. 2nd printing 2018 Edition

¹⁹ <https://miniwebtool.com/fr/outlier-calculator/#:~:text=Une%20valeur%20aberrante%20dans%20une,c'est%20une%20valeur%20aberrante.>

Par ailleurs, **plus ce seuil est bas, plus il éliminera de valeurs aberrantes** ; cependant, nous voulons nous concentrer davantage sur les "**valeurs aberrantes extrêmes**" plutôt que sur les simples valeurs aberrantes. Parce que nous risquons de perdre de l'information, ce qui réduira la précision de nos modèles.

- ✓ On commence tout d'abord par la visualisation des distributions.
Nous commençons par visualiser la distribution de la caractéristique que nous allons utiliser pour éliminer certaines des valeurs aberrantes. V14 est la seule caractéristique qui a une distribution gaussienne par rapport aux caractéristiques V12 et V10.
- ✓ Détermination du seuil : Après avoir décidé du nombre que nous utiliserons pour multiplier l'IQR (plus le nombre d'aberrations éliminées est faible), nous déterminerons les seuils supérieurs et inférieurs en soustrayant $q25 - \text{seuil}$ (seuil extrême inférieur) et en ajoutant $q75 + \text{seuil}$ (seuil extrême supérieur).
- ✓ Suppression conditionnelle : nous créons un tableau conditionnel indiquant que si le "seuil" est dépassé dans les deux extrêmes, les instances seront supprimées.

Représentons ses visualisations à travers le boxplot où le nombre de "*valeurs extrêmes aberrantes*" a été réduit de façon considérable.

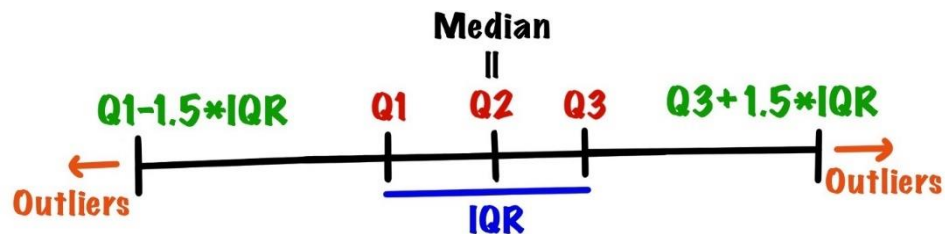


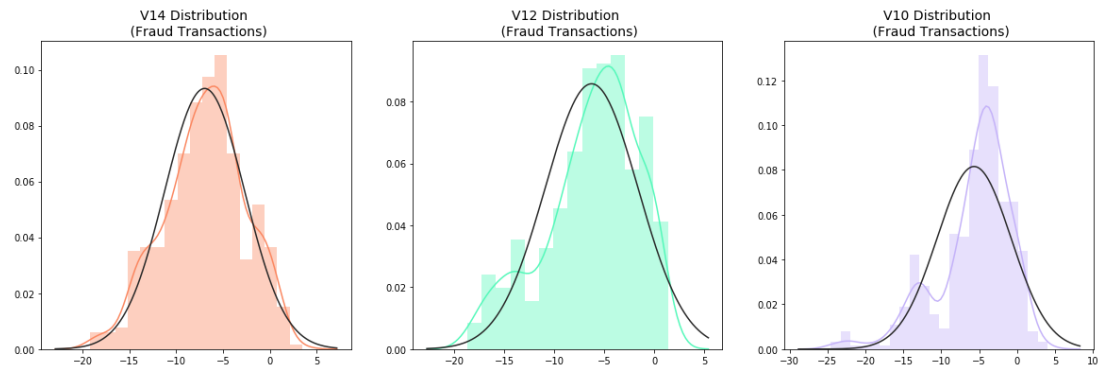
Illustration de la méthode IQR pour la détection des outliers²⁰

La méthode de l'intervalle interquartile définit les valeurs aberrantes comme des valeurs **supérieures à $Q3 + 1,5 * IQR$** ou des valeurs **inférieures à $Q1 - 1,5 * IQR$** ²¹.

²⁰ <https://towardsdev.com/outlier-detection-using-iqr-method-and-box-plot-in-python-82e1e15232bd>

²¹ <https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/identifying-outliers-iqr-rule>

Après avoir mis en œuvre la réduction des outliers, notre précision a été améliorée de plus de 3%. Certaines valeurs aberrantes peuvent fausser la précision de nos modèles, mais n'oublions pas que nous devons éviter une perte d'information extrême, sinon notre modèle risque d'être en « Underfitting »²².



Quartile 25: -9.692722964972385 | Quartile 75: -4.282820849486866
iqr : 5.409902115485519
Cut Off: 8.114853173228278
V14 Lower: -17.807576138200663
V14 Upper: 3.8320323237414122
Feature V14 Outliers for Fraud Cases: 4
V10 outliers : [-19.2143254902614, -18.8220867423816, -18.4937733551053, -18.049997689859396]

V12 Lower: -17.3430371579634
V12 Upper: 5.776973384895937
V12 outliers: [-18.683714633344298, -18.047596570821604, -18.4311310279993, -18.553697009645802]
Feature V12 Outliers for Fraud Cases: 4
Number of Instances after outliers removal: 976

V10 Lower: -14.89885463232024
V10 Upper: 4.920334958342141
V10 outliers: [-24.403184969972802, -18.9132433348732, -15.124162814494698, -16.3035376590131, -15.2399619587112, -15.1237521803455, -14.9246547735487, -16.6496281595399, -18.2711681738888, -24.5882624372475, -15.346098846877501, -20.949191554361104, -15.2399619587112, -23.228254

²² How to Use Statistics to Identify Outliers in Data by Jason Brownless (Machine Learning Mastery blog)

8357516, -15.2318333653018, -22.1870885620007, -17.141513641289198, -19.836148851696, -22.1870885620007, -16.6011969664137, -16.7460441053944, -15.563791338730098, -14.9246547735487, -16.2556117491401, -22.1870885620007, -15.563791338730098, -22.1870885620007]

Feature V10 Outliers for Fraud Cases: 27

Number of Instances after outliers removal: 947

CHAPITRE 4



RANDOM UNDERSAMPLING ET OVERSAMPLING POUR LES ENSEMBLES DE DONNEES DESEQUILIBRES – PARTIE 2

a. L'algorithme de la réduction de la dimensionnalité (t-SNE)

Elle permet d'éliminer les variables corrélées qui ne contribuent aucune prise de décision. Il existe plusieurs approches pour ce faire, dont t-SNE (***t-distributed Stochastic Neighbor Embedding***), qui est un algorithme d'apprentissage non supervisé connu notamment pour sa capacité à faciliter la visualisation des données non linéaires ayant beaucoup de descripteurs²³.

Il consiste à prendre des données dans un espace de grande dimension, et à les remplacer par d'autres **dans un espace de dimension inférieure**, mais qui contiennent encore ici, la plupart des informations contenues dans le grand ensemble. C'est-à-dire, qu'on cherche à construire moins de variables tout en **conservant le maximum d'informations** possible.

En Machine Learning, ce processus de traitement de données est crucial dans certains cas, parce que les jeux de données plus petits sont plus faciles à explorer, exploiter et à visualiser, et rendent l'analyse des données beaucoup plus facile et plus rapide. Cette étape est importante aussi dans les cas du sur-apprentissage et des données très éparpillées (**fléau de la dimensionnalité**), qui nécessitent beaucoup de temps et de puissance de calcul pour les étudier. En utilisant un espace de plus petite dimension, on obtient des algorithmes plus efficaces, ainsi qu'un panel de solutions plus réduit. Pour ce faire, on peut agir de deux manières différentes²⁴.

-  En ce qui nous concerne, l'algorithme t-SNE peut regrouper assez précisément les cas de fraude et de non-fraude dans notre ensemble de données.
-  Bien que le sous-échantillon soit assez petit, l'algorithme t-SNE est capable de détecter les clusters de manière assez précise dans chaque scénario.

²³ <https://blent.ai/algorithme-tsne/>

²⁴ StatQuest: t-SNE, Clearly Explained by Joshua Starmer

- ✚ Cela nous donne une indication que les modèles prédictifs futurs seront assez performants pour séparer les cas de fraude des cas non frauduleux

T-SNE Implementation → T-SNE took 6.8 s

PCA Implementation → PCA took 0.02 s

TruncatedSVD → Truncated SVD took 0.0093 s

b. Le « Classifiers »

Classificateurs (sous-échantillonnage) :

Dans cette section, nous allons former quatre types de classificateurs et décider lequel sera le plus efficace pour détecter les transactions frauduleuses. Avant cela, nous devons diviser nos données en ensembles de formation et de test et séparer les caractéristiques des étiquettes.

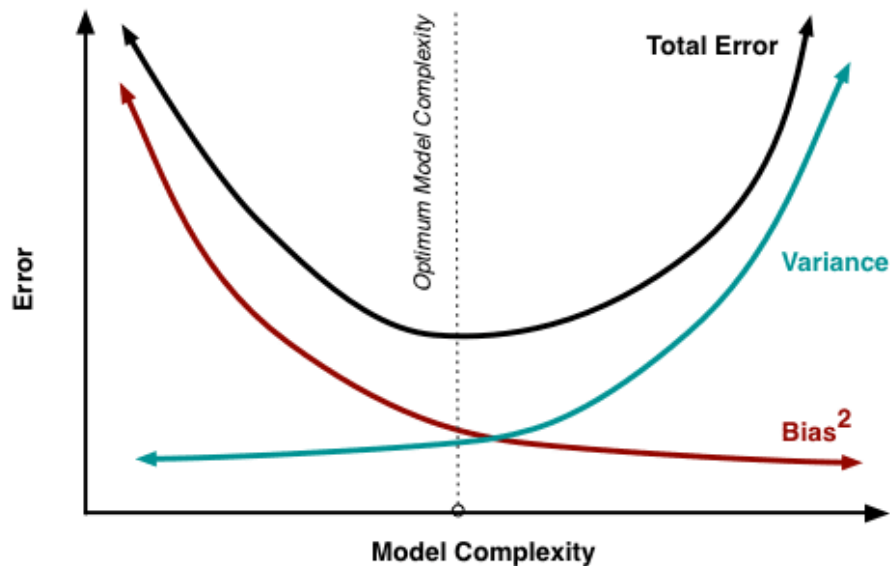
- Le classificateur de régression logistique est plus précis que les trois autres classificateurs dans la plupart des cas. (Nous analyserons également plus en détail la régression logistique).
- GridSearchCV est utilisé pour déterminer les paramètres qui donnent le meilleur score prédictif pour les classifieurs.
- La régression logistique a le meilleur score ROC (*Receiving Operating Characteristic*), ce qui signifie que la régression logistique sépare assez précisément les transactions frauduleuses et non frauduleuses.

Lors de la création de modèles d'apprentissage automatique, nous voulons maintenir l'erreur aussi faible que possible. Les deux principales sources d'erreur sont le biais et la variance. Si nous parvenions à réduire ces deux facteurs, nous pourrions construire des modèles plus précis et remplir ainsi notre objectif qui est de parvenir à construire fiable et robuste.

De ce fait, le biais et la variance ne peuvent qu'augmenter l'erreur d'un modèle. D'un point de vue plus intuitif, nous voulons **un faible biais** pour éviter de construire un modèle trop simple. Dans la plupart des cas, un modèle simple fonctionne mal sur les données de formation, et il est extrêmement probable qu'il répète les mauvaises performances sur les données de test.

De même, nous voulons **une faible variance** pour éviter de construire un modèle trop complexe. Un tel modèle correspond presque parfaitement à tous les points de données de l'ensemble d'apprentissage. Les données de formation,

cependant, contiennent généralement du bruit et ne sont qu'un échantillon d'une population beaucoup plus large. Un modèle trop complexe capture ce bruit. Et lorsqu'elles sont testées sur des données *hors échantillon*, les performances sont généralement médiocres. C'est parce que le modèle apprend trop bien les *exemples de données d'apprentissage*²⁵.



Courbes d'apprentissage ou Learning Curves

- ⇒ Plus l'écart entre le score de formation et le score de validation croisée est important, plus il est probable que votre modèle soit sur ajusté (**variance élevée**).
- ⇒ Si le score est faible à la fois dans les ensembles de formation et de validation croisée, cela indique que notre modèle est sous-adapté (**biais élevé**).
- ⇒ **Le classificateur de régression logistique** montre le meilleur score dans les deux ensembles de formation et de validation croisée.
- ⇒ En bref, une courbe d'apprentissage montre comment l'erreur change à mesure que la taille de l'ensemble d'apprentissage augmente

²⁵ <https://www.dataquest.io/blog/learning-curves-machine-learning/>

⇒ Nous avons donc deux scores d'erreur à surveiller : **un pour l'ensemble de validation et un pour les ensembles d'apprentissage**

- LogisticRegression a un score d'entraînement de 93.0 % de précision
- KNeighborsClassifier a obtenu un score d'entraînement de 93,0 % de précision.
- SVC a obtenu un score d'apprentissage de 93,0 % du score d'entraînement
- DecisionTreeClassifier a un score d'apprentissage de 90,0 % de précision

Logistic Regression Cross Validation Score	94.05%
Knears Neighbors Cross Validation Score	92.73%
Support Vector Classifier Cross Validation Score	93.79%
DecisionTree Classifier Cross Validation Score	91.41%

Train: [56959 56960 56961 ... 284804 284805 284806] Test: [0 1 2 .
.. 57174 58268 58463]

Train: [0 1 2 ... 284804 284805 284806] Test: [56959 56960 5696
1 ... 115109 116514 116648]

Train: [0 1 2 ... 284804 284805 284806] Test: [113919 113920 1139
21 ... 170890 170891 170892]

Train: [0 1 2 ... 284804 284805 284806] Test: [168136 168614 1688
17 ... 228955 229310 229751]

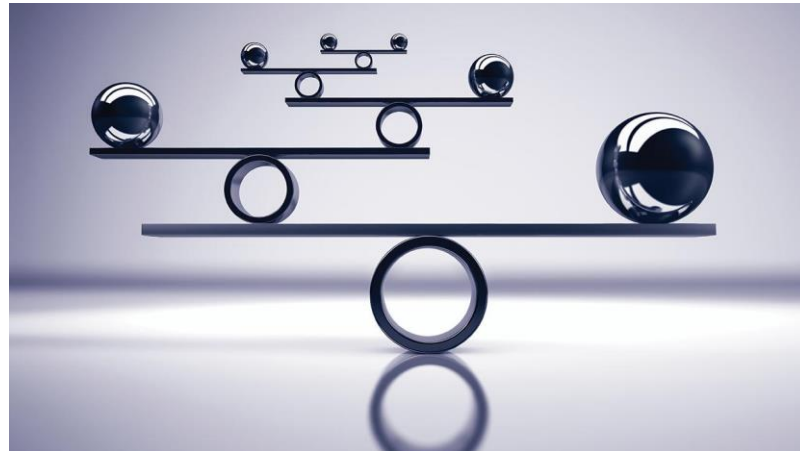
Train: [0 1 2 ... 228955 229310 229751] Test: [227842 227843 2278
44 ... 284804 284805 284806]

NearMiss Label Distribution: Counter({0: 492, 1: 492})

Nous devons également définir l'**algorithme Near-Miss**²⁶ pour les ensembles de données déséquilibrés. Il peut être regroupé sous des algorithmes de sous-échantillonnage et constitue un moyen efficace d'équilibrer les données. L'algorithme le fait en examinant la distribution des classes et en éliminant au hasard des échantillons de la classe la plus large. Lorsque deux points appart

²⁶ <https://analyticsindiamag.com/using-near-miss-algorithm-for-imbalanced-datasets/>

enant à des classes différentes sont très proches l'un de l'autre dans la distribution, cet algorithme élimine le point de données de la classe la plus grande en essayant ainsi d'équilibrer la distribution.



1. L'algorithme calcule d'abord la distance entre tous les points de la plus grande classe avec les points de la plus petite classe. Cela peut faciliter le processus de sous-échantillonnage.
2. On sélectionne ensuite les instances de la plus grande classe qui ont la distance la plus courte avec la plus petite classe. Ces instances doivent être stockées pour être éliminées.
3. S'il y a m instances de la plus petite classe, l'algorithme renverra $m \times n$ instances de la plus grande classe.

Il est essentiel de s'assurer que les données ne sont pas biaisées avant que le modèle puisse être formé. L'algorithme de Near-Miss est celui qui garantit que les données sont réparties plus uniformément et ne causent pas de biais.

Logistic Regression	97.31%
KNears Neighbors	92.76%
Support Vector Classifier	96.96%
Decision Tree Classifier	91.66%

c. Un regard approfondi sur la régression logistique

Dans cette section, nous allons examiner de plus près le classificateur de la régression logistique.

- Les vrais positifs : Transactions frauduleuses correctement classées
- Faux positifs : Transactions frauduleuses incorrectement classifiées
- Vrais négatifs : Transactions non frauduleuses correctement classées
- Faux négatifs : Transactions non frauduleuses incorrectement classées
- Précision : $\text{Vrais positifs} / (\text{Vrais positifs} + \text{Faux positifs})$
- Rappel : $\text{Vrais positifs} / (\text{Vrais positifs} + \text{Faux négatifs})$

La **précision**, comme son nom l'indique, indique le degré de précision (de certitude) de notre modèle dans la détection des transactions frauduleuses, tandis que le **rappel** représente le nombre de cas de fraude que notre modèle est capable de détecter.

Compromis précision/rappel : Plus notre modèle est précis (sélectif), moins il détectera de cas. Exemple : En supposant que notre modèle ait une précision de 95%, disons qu'il n'y a que 5 cas de fraude pour lesquels le modèle est précis à 95% ou plus qu'il s'agit de cas de fraude. Ensuite, disons qu'il y a 5 autres cas que notre modèle considère à 90% comme des cas de fraude, si nous réduisons la précision, il y a plus de cas que notre modèle sera en mesure de détecter.

Pour résumé, la précision commence à diminuer entre 0,90 et 0,92 ; néanmoins, notre score de précision est encore assez élevé et nous avons toujours un score de rappel en baisse.

Overfitting:

Recall Score: 0.90
Precision Score: 0.76
F1 Score: 0.82
Accuracy Score: 0.81

Ce que ça aurait dû être :

Accuracy Score: 0.65

Precision Score: 0.00

Recall Score: 0.29

F1 Score : 0.00

Average precision-recall score : 0.63

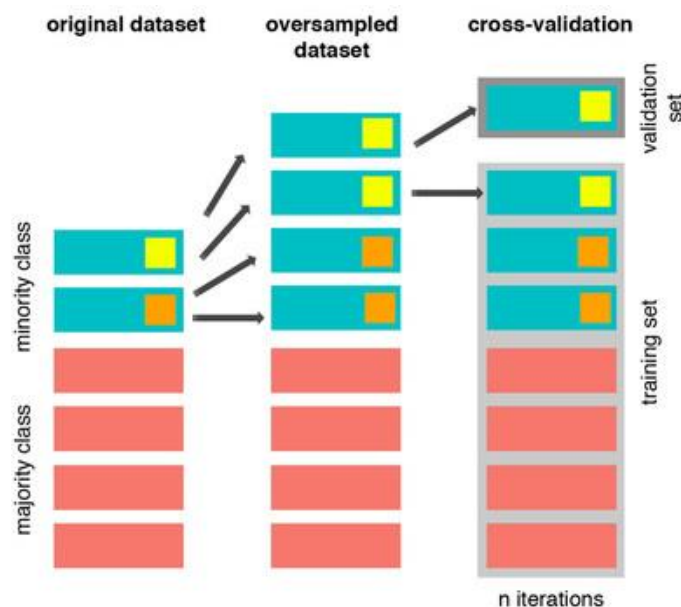
La méthode **SMOTE** est l'abréviation de *Synthetic Minority Over-sampling Technique*. La technique de suréchantillonnage synthétique minoritaire, ou SMOTE en abrégé, est une technique de prétraitement utilisée pour remédier à un déséquilibre de classe dans un ensemble de données

Contrairement au sous-échantillonnage aléatoire, SMOTE crée de nouveaux points synthétiques afin d'obtenir un équilibre entre les classes. Il s'agit d'une autre alternative pour résoudre les "problèmes de déséquilibre des classes".

- Faire la différence entre un échantillon et son voisin le plus proche
- Multipliez la différence par un nombre aléatoire entre 0 et 1
- Ajoutez cette différence à l'échantillon pour générer un nouvel exemple synthétique dans l'espace des fonctionnalités
- Continuer avec le prochain voisin le plus proche jusqu'au numéro défini par l'utilisateur
- Emplacement des points synthétiques : SMOTE choisit la distance entre les plus proches voisins de la classe minoritaire, entre ces distances il crée des points synthétiques.
- Et au final, nous conservons plus d'informations puisque nous n'avons pas eu à supprimer de lignes contrairement au sous-échantillonnage aléatoire.
- Compromis précision || temps : Bien qu'il soit probable que SMOTE soit plus précis que le sous-échantillonnage aléatoire, il faudra plus de temps pour l'entraîner puisqu'aucune ligne n'est éliminée comme indiqué précédemment.

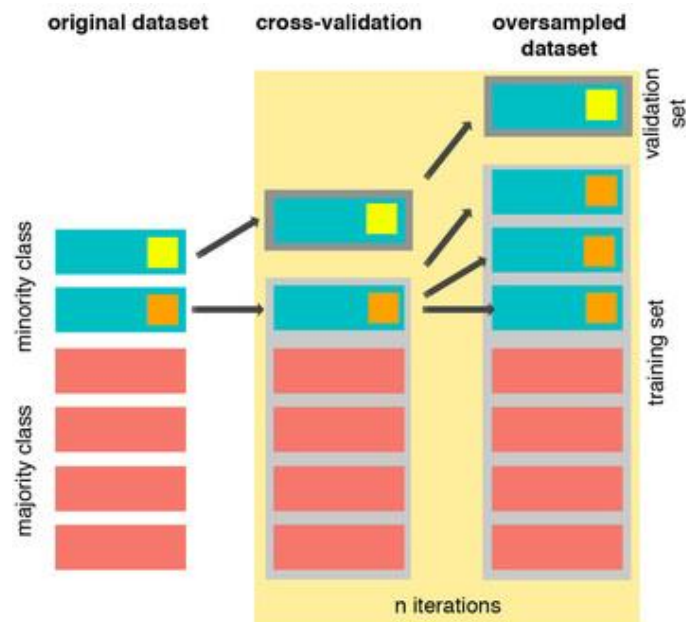
d. L'Oversampling avec la méthode « SMOTE »

Dans notre analyse du sous-échantillonnage, je veux vous montrer une erreur courante commise par la plupart des data scientist. Par exemple, si vous voulez sous-échantillonner ou suréchantillonner vos données, vous ne devez pas le faire avant la validation croisée. En effet, vous influencerez directement l'ensemble de validation avant de mettre en œuvre la validation croisée, ce qui entraînera un problème de "fuite de données". Dans la section suivante, on verra des scores de précision et de rappel étonnants, mais en réalité, nos données sont en overfitting.



Comme mentionné précédemment, si nous obtenons la classe minoritaire ("Fraude") dans notre cas, et que nous créons les points synthétiques avant la validation croisée, nous avons une certaine influence sur « l'ensemble de validation » du processus de validation croisée. Rappelez-vous comment fonctionne la validation croisée, supposons que nous divisons les données en 5 lots, 4/5 de l'ensemble de données seront l'ensemble d'entraînement et 1/5 sera l'ensemble de validation. L'ensemble de test ne doit pas être touché !

Pour cette raison, nous devons créer des points de données synthétiques "pendant" la validation croisée et non avant, comme ci-dessous ²⁷:



Comme on peut le voir dans l'illustration ci-dessus, SMOTE²⁸ se produit "pendant" la validation croisée et non "avant" le processus de validation croisée. Les données synthétiques sont créées uniquement pour l'ensemble de formation sans affecter l'ensemble de validation²⁹.

Length of X (train): 227846 | Length of y (train): 227846
Length of X (test): 56961 | Length of y (test): 56961

Accuracy: 0.9411012499098698
Precision: 0.06129812709958554
Recall: 0.9137293086660175
F1: 0.1130377886634673

precision	recall	f1-score	support
-----------	--------	----------	---------

²⁷ <https://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation>

²⁸ Japkowicz, N. (2000). The Class Imbalance Problem: Significance and Strategies. In Proceedings of the 200 International Conference on Artificial Intelligence (IC-AI2000): Special Track on Inductive Learning Las Vegas, Nevada.

²⁹ SMOTE explained for noobs

No Fraud	1.00	0.99	0.99	56863
Fraud	0.11	0.86	0.20	98
accuracy				0.99
macro avg	0.56	0.92	0.60	56961
weighted avg	1.00	0.99	0.99	56961

Average precision-recall score: 0.76

Dans le cas d'une classification multi-classes, nous adoptons des méthodes de calcul de **moyenne** pour le calcul du score F1, ce qui se traduit par un **ensemble de scores moyens différents** (macro, pondéré, micro). Pour comprendre la matrice ci-dessus, nous devons en comprendre les fondements expliqués ci-après.

Précision : il se concentre uniquement sur les clients pour lesquels le modèle a prédit une fraude et donne une indication sur les faux positifs. Les faux positifs ce sont les clients pour lesquels le score a prédit une fraude mais où il n'y en a pas. Il faut limiter les faux positifs pour réduire le coût des campagnes.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Le recall : il se concentre uniquement sur les clients qui ont subi une fraude et donne une indication sur la part de faux négatifs. Les faux négatifs ce sont les fraudes mais qui ne sont pas détectés par le score. Concrètement ce sont des fraudes que vous ne détectez pas et pour lesquels vous ne pourrez pas agir pour éviter des cas de fraude.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

L'indicateur le plus simple est l'**Accuracy** : il indique le pourcentage de bonnes prédictions.

Pour évaluer les performances du modèle de manière exhaustive, nous devons examiner à la **fois** la précision et le rappel. Le score F1 est une mesure utile qui tient compte des deux scores précédents.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{F1 Score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2} (\text{FP} + \text{FN})}$$

Moyenne macroéconomique ou macro AVG : le score F1 macro-moyen (ou score macro F1) est calculé à l'aide de la moyenne arithmétique (ou moyenne **non pondérée**) de tous les scores F1 par classe.

Cette méthode traite toutes les classes de la même manière, quelles que soient leurs valeurs de **support**. Calcul : $(0.99 + 0.20) / 2 = 0.60$

Moyenne pondérée ou weighted AVG : le score F1 moyen **pondéré** est calculé en prenant la moyenne de tous les scores F1 par classe tout en tenant compte du soutien de chaque classe. La valeur calculée de **0,64** correspond à la moyenne pondérée du score F1 de notre modèle.

CHAPITRE 5

PHASE DE TEST

Dans cette partie, nous allons analyser les résultats de la matrice de confusion qui sont classés en quatre grandes catégories : **les vrais positifs, les vrais négatifs, les faux positifs et les faux négatifs.**

Positif/Négatif : Type de classe (étiquette) ["Non", "Oui"] Vrai/Faux : Correctement ou incorrectement classé par le modèle.

Vrais négatifs (true negative) (carré en haut à gauche) : Il s'agit du nombre de classifications correctes de la classe "Non" (Aucune fraude détectée). Ce sont les cas où les prédictions et les valeurs réelles sont toutes les deux négatives. Avec le même exemple, le test indique alors qu'une opération est frauduleuse, ce qui est bel et bien le cas.

Faux négatifs (carré en haut à droite) : Il s'agit du nombre de classifications incorrectes de la classe "Non" (Pas de fraude détectée). Ce sont les cas où les prédictions sont négatives alors que les valeurs réelles sont positives. Ils sont également considérés comme des erreurs de type 2.

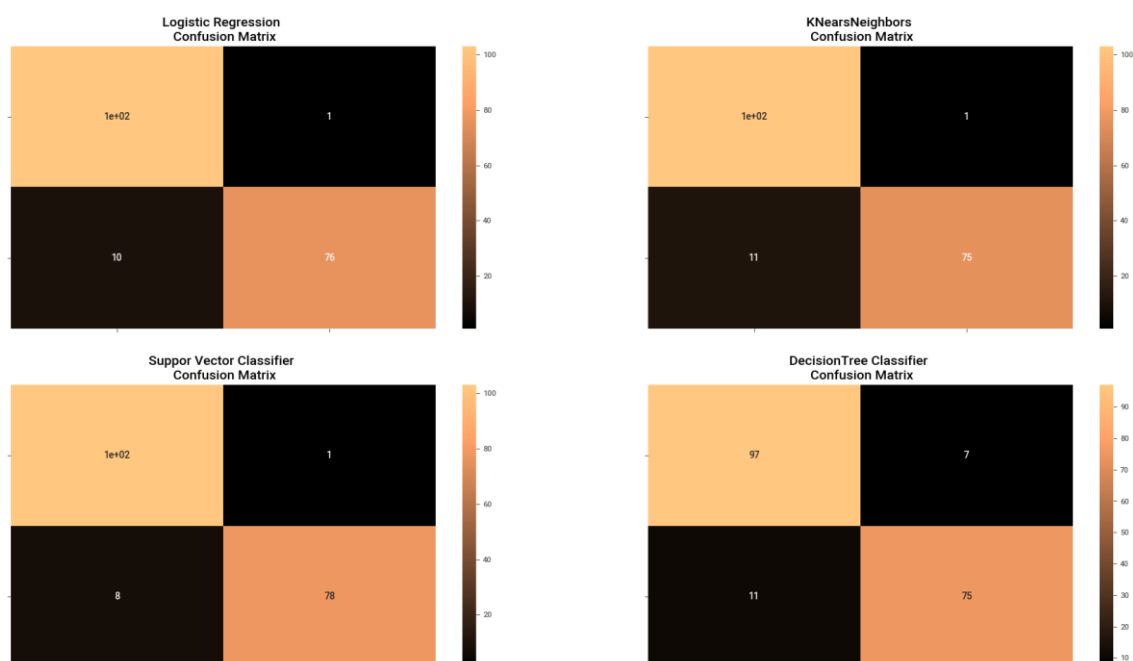
Faux positifs (false positive) (carré en bas à gauche) : Il s'agit du nombre de classifications incorrectes de la classe "Oui" (Fraude détectée). Les faux positifs ou FP indiquent quant à eux une prédiction positive contraire à la valeur réelle qui est négative. Ils sont également considérés comme des erreurs de type 1.

Les vrais positifs ou TP (true positive) (carré en bas à droite) : indiquent les cas où les prédictions et les valeurs réelles sont effectivement positives. Cela signifie que le système reconnaît comme frauduleux une fraude qui l'est

réellement. Il s'agit du nombre de classifications correctes de la classe "Oui" (Fraude détectée).

Sous-échantillonnage aléatoire : nous allons évaluer la performance finale des modèles de classification dans le sous-ensemble de sous-échantillonnage aléatoire. On fait attention à ne pas confondre un point important car il ne s'agit pas des données de la trame de données originale.

Modèles de classification : on voit que les modèles les plus performants sont la régression logistique et le classificateur à vecteur de support (SVM).



Logistic Regression:

	precision	recall	f1-score	support
0	0.91	0.99	0.95	104
1	0.99	0.88	0.93	86
accuracy			0.94	190
macro avg	0.95	0.94	0.94	190
weighted avg	0.95	0.94	0.94	190

KNeighbors Neighbors:

	precision	recall	f1-score	support
0	0.90	0.99	0.94	104

1	0.99	0.87	0.93	86
accuracy			0.94	190
macro avg	0.95	0.93	0.94	190
weighted avg	0.94	0.94	0.94	190

Support Vector Classifier:

	precision	recall	f1-score	support
0	0.93	0.99	0.96	104
1	0.99	0.91	0.95	86
accuracy			0.95	190
macro avg	0.96	0.95	0.95	190
weighted avg	0.95	0.95	0.95	190

Support Vector Classifier:

	precision	recall	f1-score	support
0	0.90	0.93	0.92	104
1	0.91	0.87	0.89	86
accuracy			0.91	190
macro avg	0.91	0.90	0.90	190
weighted avg	0.91	0.91	0.91	190

	Technique	Score
0	Random UnderSampling	0.942105
1	Oversampling (SMOTE)	0.987974

Nous passons aux réseaux neuronaux testant le sous-échantillonnage aléatoire des données par rapport au sur-échantillonnage (SMOTE)

Dans cette section, nous implémentons un réseau de neurones simple (*avec une couche cachée*) afin de déterminer lequel des deux modèles de régression logistique implémentés dans la section (sous-échantillonnage ou sur-échantillonnage (SMOTE)) est le plus précis pour détecter les transactions frauduleuses et non frauduleuses.

La matrice de confusion : voici encore une fois, comment fonctionne la matrice de confusion

	Pas de fraude	Fraude
Positif	Faux positif La quantité de transactions correctement classées par notre modèle d'absence de fraude	Vrai positif Le nombre de transactions incorrectement classées comme des cas de fraude, mais l'étiquette réelle n'est pas une fraude.
Négatif	Vrai négatif Le nombre de transactions incorrectement classées comme des cas d'absence de fraude, mais l'étiquette réelle est la fraude	Faux négatif Le nombre de transactions correctement classées par notre modèle comme des cas de fraude

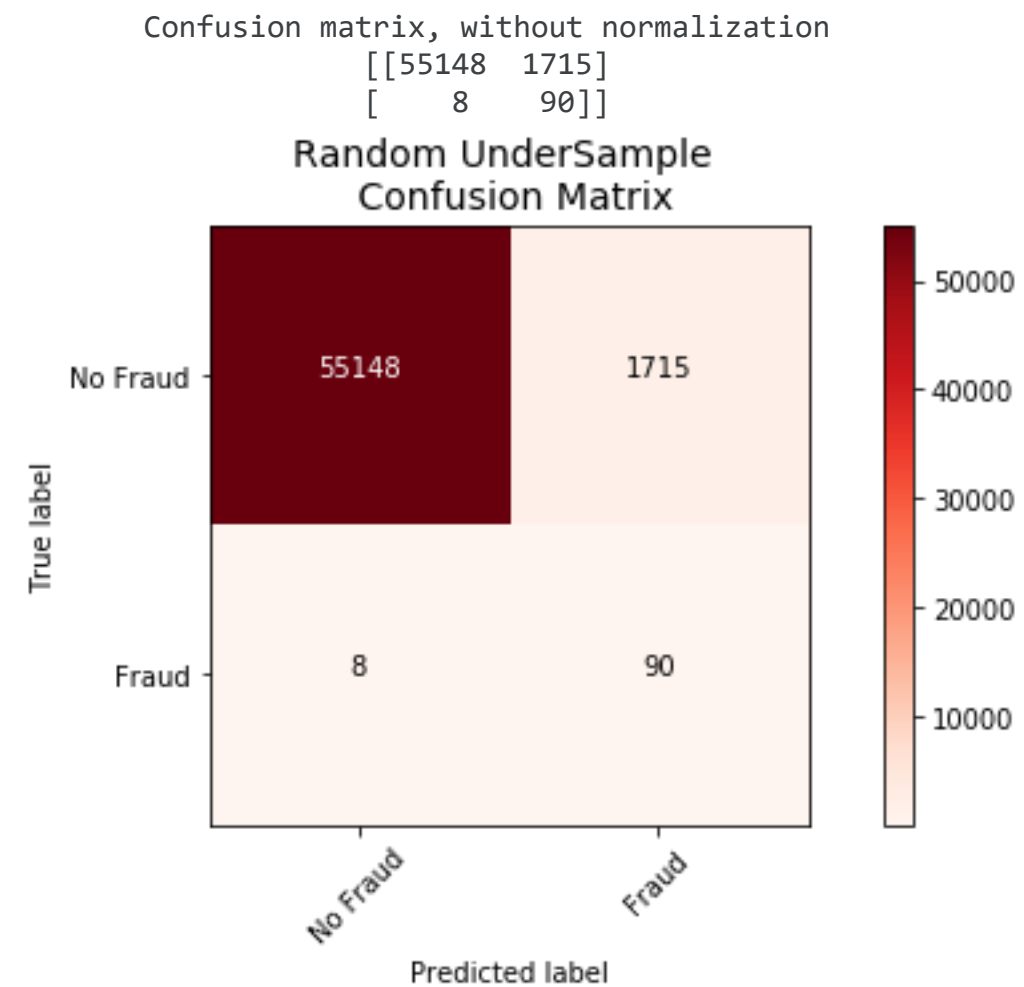
Résumé (Keras || Random Undersampling) :

Ensemble de données : Dans cette phase finale de test, nous ajusterons ce modèle à la fois dans le sous-ensemble sous-échantillonné aléatoire et dans l'ensemble de données suréchantillonné (SMOTE) afin de prédire le résultat final en utilisant les données de test de la trame de données originale.

Structure du réseau neuronal : Comme indiqué précédemment, il s'agira d'un modèle simple composé d'une couche d'entrée (où le nombre de nœuds est égal au nombre de caractéristiques) plus un nœud de biais, d'une couche cachée de 32 nœuds et d'un nœud de sortie composé de deux résultats possibles 0 ou 1 (pas de fraude ou fraude).

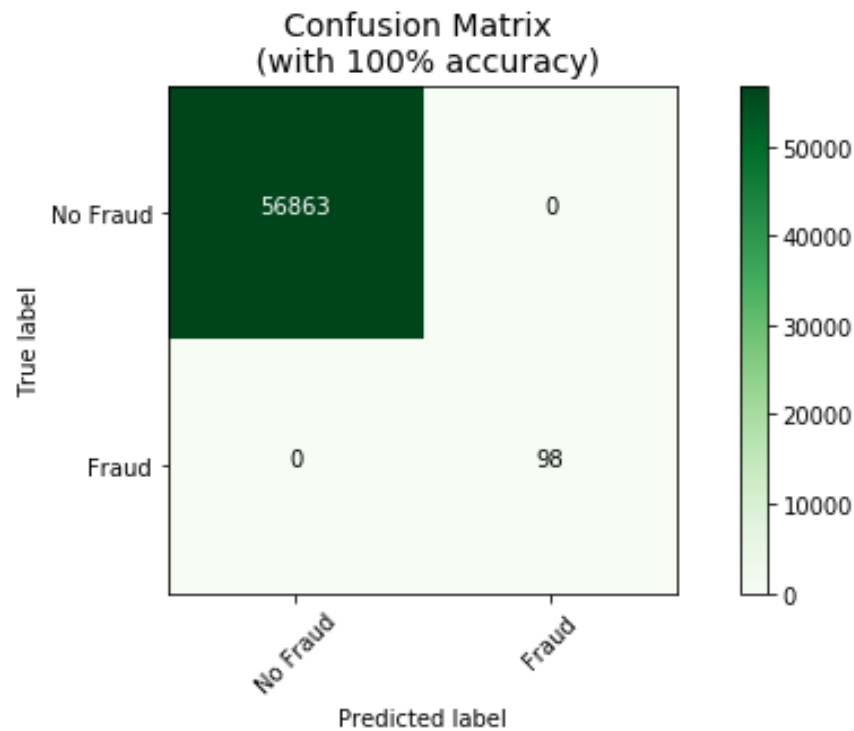
Autres caractéristiques : Le taux d'apprentissage (*learning rate*) sera de 0,001, l'optimiseur que nous utiliserons est l'AdamOptimizer, la fonction d'activation utilisée dans ce scénario est "Relu" et pour les sorties finales, nous utiliserons l'entropie croisée catégorielle clairsemée, qui donne la probabilité qu'une

instance soit frauduleuse ou non (la prédiction choisira la probabilité la plus élevée entre les deux).



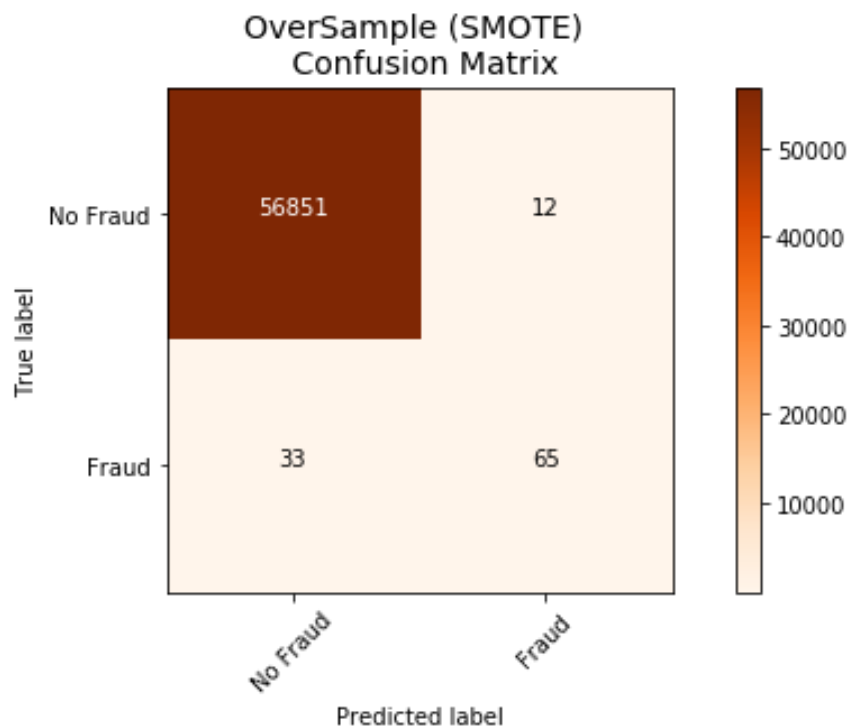
Confusion matrix, without normalization

```
[[56863    0]
 [     0   98]]
```



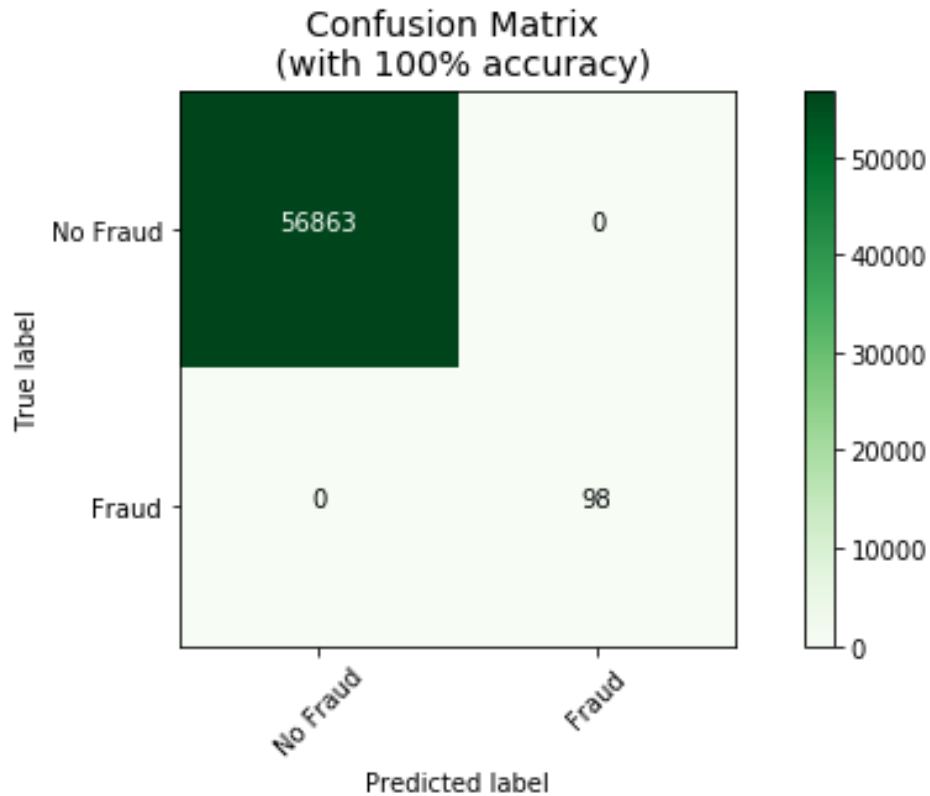
Confusion matrix, without normalization

```
[[56851  12]
 [  33  65]]
```



Confusion matrix, without normalization

$$\begin{bmatrix} 56863 & 0 \\ 0 & 98 \end{bmatrix}$$



Conclusion :

L'implémentation de SMOTE sur notre ensemble de données déséquilibré nous a aidé à résoudre le déséquilibre de nos étiquettes (plus de transactions sans fraude que de transactions avec fraude). Néanmoins, je dois dire que parfois le réseau neuronal sur l'ensemble de données suréchantillonné prédit moins de transactions frauduleuses correctes que notre modèle utilisant l'ensemble de données sous-échantillonné. Cependant, il ne faut pas oublier que la suppression des valeurs aberrantes n'a été mise en œuvre que sur l'ensemble de données aléatoires du sous-échantillon et non sur celui du sur-échantillon.

De plus, dans nos données de sous-échantillon, notre modèle est incapable de détecter correctement un grand nombre de transactions non frauduleuses et,

au lieu de cela, classe à tort ces transactions non frauduleuses comme des cas de fraude. On peut par exemple imaginez que des personnes qui effectuaient des achats réguliers voient leur carte bloquée parce que notre modèle a classé cette transaction comme une transaction frauduleuse, ce serait un énorme désavantage pour l'institution financière et on pourrait rapidement avoir des retours négatifs sur ce type de cas s'ils s'avèrent que les clients se plaignent de plus en plus. Le nombre de plaintes et le mécontentement des clients pourraient rapidement augmenter créant ainsi un mécontentement général des clients auprès de leur banque. L'objectif a été rempli pour une partie même s'il reste des points perfectibles dans le modèle et dans pas mal de domaines, notamment l'exploitation des données qui sont pour la plupart anonymisées et qui ne sont pas exploitables

BIBLIOGRAPHIE

- Aurélien Géron.
Hands on Machine Learning with Scikit-Learn & TensorFlow.
O'Reilly, 2017.
- Bruce Ratner.
Statistical and Machine Learning Data Mining. Third Edition, 2017.
- Japkowicz, N.
(2000). The Class Imbalance Problem: Significance and Strategies. In Proceedings of the 200 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning Las Vegas, Nevada.
- Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* (2002): 321-357.
- Fele-Žorž, Gašper, et al. "A comparison of various linear and non-linear signal processing techniques to separate uterine EMG records of term and pre-term delivery groups." *Medical & biological engineering & computing* 46.9 (2008): 911-922.
- Ren, Peng, et al. "Improved Prediction of Preterm Delivery Using Empirical Mode Decomposition Analysis of Uterine Electromyography Signals." *PloS one*. 10.7 (2015): e0132116.
- Fergus, Paul, et al. "Prediction of preterm deliveries from EHG signals using machine learning." (2013): e77154. *PloS one*