

Rapport de stage de fin d'études

à retourner au plus tard **1 semaine avant la date de la soutenance**

Rapport de Stage
Ing5 - 3^{ème} année Cycle Ingénieur
Promotion 2021

Majeure :

Nom : SORY

- ENE
 SI/BDA/CYD

Prénom : Anas

- SE/VCA
 SAN
 FIN
 OCRES

Entreprise d'accueil

Nom : Acensi

Adresse : Tour Black Pearl, 14 rue du général Audran – 92400 Courbevoie

Adresse du lieu de stage si différent :

Engagement de Confidentialité : oui non

Reçu le 09 /08/2021

Rapport Confidentiel à remettre au tuteur de stage à l'issue de la soutenance : oui non

Pénalités Observées :

Description de la mission : Mise en place de plateforme de détection de fraude à la carte de crédit bancaire : Exploration des données, mise en place de modèles, benchmark de modèles et amélioration de la prédiction.

Remerciements

Au terme de cette expérience professionnelle, il m'apparaît opportun de commencer ce rapport en m'acquittant d'une dette de reconnaissance auprès de toutes les personnes dont l'intervention au cours de ce stage a favorisé son aboutissement.

Je voudrais tout d'abord adresser toute ma gratitude à mon maître de stage, Mr KOUTHON James, directeur technique au sein de l'entreprise ACENSI, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grace à la qualité de son encadrement en entreprise, j'ai pu accomplir les missions qui m'ont été attribuées. Il fut d'une aide précieuse dans les moments les plus délicats.

Je tiens aussi à remercier le corps professoral de l'école centrale d'électronique, qui m'a fourni les outils nécessaires au bon déroulement de mon stage spécialement Mr.BUSCA Jean Michel qui fut le premier à me soutenir dans ma démarche de stage. Mes remerciements sont aussi adressés à toutes les personnes qui m'ont aidé lors de la rédaction et relecture de mon rapport de stage notamment Tristan François. Leurs encouragements permanent et leurs aides m'ont énormément facilité la tache

Table des matières

Remerciements.....	2
Table des matières.....	3
Résumé	5
Introduction.....	5
Présentation de l'organisme d'accueil	6
Acensi en quelques mots	6
Domaine d'expertise.....	6
Acensi : Secteurs et clients	6
Etude de la maturité RSE	7
Mes missions : Cahier des charges	8
Solutions proposées	8
Machine learning dans la finance	8
Etat de l'art	8
Trading algorithmique	9
Détection de fraude.....	9
Les services de souscription de banques/assurances.....	9
Automatisation et chatbot	9
Prévision des prix.....	9
Opinion mining	10
Règlements des transactions.....	10
Blanchiment d'argent	10
Machine learning, Deep learning, Artificial intelligence et la Data science.....	10
Les types de machine learning	11
Apprentissage supervisé	12
Classification	12
Régression linéaire.....	12
Régression logistique.....	13
Support Vector Machine (SVM).....	13
K-nearest neighbors.....	13
Arbres décisionnels.....	14
Random forest	14
Apprentissage non-supervisé	15
Réduction de dimensionnalité.....	15
Clustering.....	15
Réseaux de neurones.....	16
Développer un modèle de machine learning en Python	17
Pourquoi Python.....	17
Les packages de python en machine Learning	17
Étapes du développement d'un modèle dans l'écosystème Python	18

Performance du modèle	18
Overfitting et Underfitting.....	18
Cross validation	19
Metrics d'évaluation.....	19
Hyper parameter tuning	21
Grid search.....	22
Diverses techniques utilisées.....	22
Nos dataset.....	23
Kaggle	23
Définition du problème	23
Exploratory data analysis.....	24
Dataset 2 : Archive credit card fraude detection.....	27
Exploratory data analysis.....	28
Optimisation	46
Bagging :	52
Théorie des jeux : Adversial learning.....	52
Free range attack.....	53
Restrained attack.....	53
Expériences : Resultats	53
Difficultés rencontrées	56
Suite du projet chez Acensi.....	56
Conclusion	56
Table des figures.....	56

Résumé

L'expansion du commerce électronique, ainsi que la confiance croissante des clients dans les paiements électroniques, font de la détection de la fraude un problème majeur dans le secteur de la finance. Il est difficile d'avoir des chiffres sur l'impact de la fraude, car les entreprises et les banques n'aiment pas divulguer le montant des pertes. Dans le même temps, les données publiques sont rarement disponibles pour des raisons de confidentialité, ce qui laisse sans réponse de nombreuses questions sur la meilleure stratégie à adopter. Un autre problème dans l'estimation des pertes dues à la fraude par carte de crédit est que nous ne pouvons mesurer les pertes que pour les fraudes qui ont été détectées, et qu'il n'est pas possible d'évaluer l'importance des fraudes non signalées/non détectées. Les schémas de fraude évoluent rapidement et la détection des fraudes doit être réévaluée pour passer d'une approche réactive à une approche proactive.

La détection des fraudes en temps réel exige la conception et la mise en œuvre de techniques d'apprentissage évolutives capables d'ingérer et d'analyser des quantités massives de données en continu.

Dans cet article, nous présentons le projet de détection de fraudes par carte bancaire qui intègre des outils Big Data et de Data science avec des approches d'apprentissage automatique et apprentissage profond qui traitent le déséquilibre des données, mise en place de modèles, benchmark de ces derniers pour améliorer les résultats de la prédiction.

Introduction

Dans un environnement dynamique et très concurrentiel tel que l'environnement des PMEs, les entreprises sont poussées à de plus en plus flexibles et à dépendre crucialement de leur capacité à innover, tant dans leur structure organisationnelle, leur mode de production que dans leur mode d'échange avec les clients. Cependant, dans leur recherche pour surmonter la concurrence, le principal obstacle auquel se heurtent les entreprises est la capacité à innover et créer quelque chose de nouveau qui permettrait de résoudre un problème donné. C'est dans cette optique que s'inscrit le sujet de ce stage de fin d'études qui vise à étudier et tester la capacité de la data science à résoudre la problématique : Création d'un système de détection de fraudes par carte bancaire.

La "fraude" dans les transactions par carte de crédit est l'utilisation non autorisée et non souhaitée d'un compte par une personne autre que le titulaire de la carte.

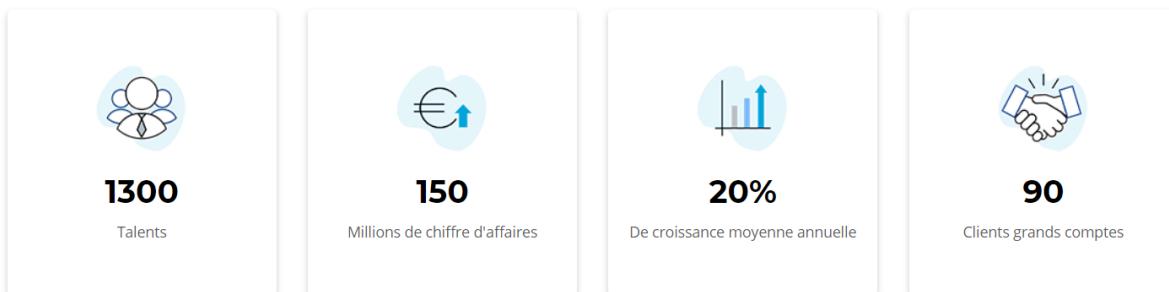
Les mesures de prévention nécessaires peuvent être adoptées pour mettre fin à cet abus et le comportement de ces pratiques frauduleuses peut être étudié afin de les minimiser et de se protéger contre les occurrences similaires à l'avenir. En d'autres termes, la fraude à la carte de crédit peut être définie à l'usage de la carte de crédit de quelqu'un d'autre à des fins personnelles alors que le propriétaire et les autorités émettrices de la carte ne savent pas que la carte est utilisée.

La détection de la fraude consiste à surveiller les activités des populations d'utilisateurs afin d'estimer, de percevoir ou d'éviter les comportements répréhensibles, qui consistent en des fraudes, des intrusions et des défaut de paiement. Il s'agit d'un problème très pertinent qui demande l'attention de communautés telles que l'apprentissage automatique et la science des données, où la solution à ce problème peut être automatisée. Ce problème est particulièrement difficile du point de vue de l'apprentissage, car il est caractérisé par divers facteurs tels que le déséquilibre des classes. Le nombre de transactions valides sont de loin beaucoup plus nombreuses que les transactions frauduleuses. En outre, les modèles de transaction changent souvent leurs propriétés statistiques au cours du temps

Présentation de l'organisme d'accueil

Acensi en quelques mots

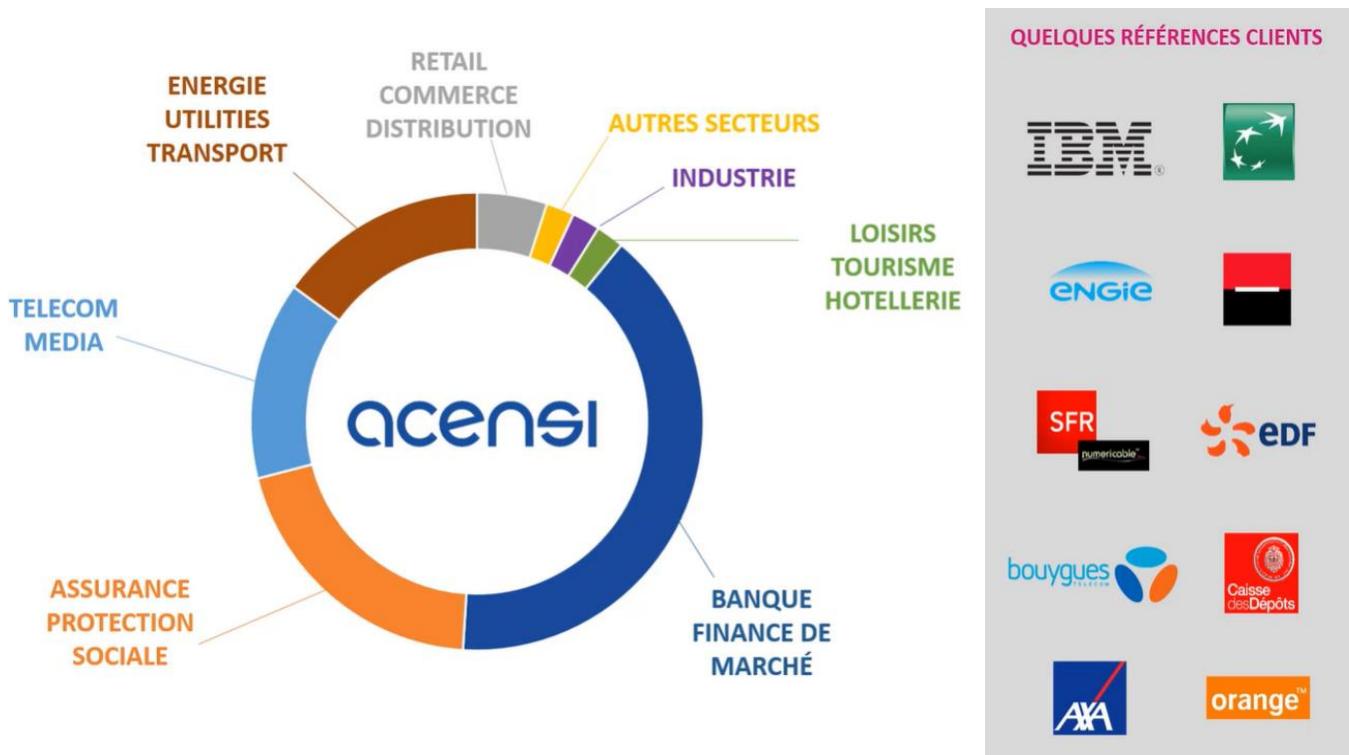
ACENSI SAS est une société par actions simplifiée. Créée en 2003, elle est située à COURBEVOIE (92400). Positionnée depuis sa création entre le cabinet de conseil généraliste et l'ESN spécialisée, le groupe ACENSI accompagne depuis 18 ans plus de 90 clients dans la gestion de leurs projets IT.



Domaine d'expertise

Avec l'émergence des nouvelles technologies de l'information, les entreprises des grands secteurs tels que la banque, l'automobile, l'énergie, l'assurance... voient leurs modèles dépassés. Certaines modifient leur modèle d'affaire ou changent leur marché, d'autres s'en réinventent un nouveau. L'objectif étant donc de s'adapter à une évolution rapide des réglementations et des technologies afin de permettre la création de nouvelles offres

Acensi : Secteurs et clients



Etude de la maturité RSE

Parce que la croissance du groupe doit se faire en osmose avec un environnement sain et durable, ACENSI participe à cette dynamique en mettant en place des solutions de développement durable : répondre aux besoins en énergie, optimiser l'utilisation des ressources, privilégier le dialogue, l'innovation sociétale et la diversité.

En adhérant au Pacte Mondial des Nations Unies, ACENSI s'est engagée sur quatre grands principes : la protection des droits de l'homme, le respect du droit du travail, la responsabilité en matière d'environnement et la lutte contre la corruption.

EcoVadis établit des évaluations RSE des entreprises en attribuant une note globale sur la base de scores qui reflètent la performance de l'entreprise sur 4 thèmes de responsabilité sociétale : environnement, social, éthique des affaires et achats responsables.

Le Groupe ACENSI a reconduit son évaluation en 2018 en obtenant une note de 67/100 pour ses actions RSE.

-Protection et respect de l'environnement :

ACENSI mène de nombreuses actions pour la protection de l'environnement dans le cadre de sa politique Opérateur d'Oxygène. Depuis 2013, ACENSI réalise également un Diagnostic Carbone chaque année, via le Programme Action Carbone de la Fondation GoodPlanet.

-Accompagnement des salariés :

ACENSI s'engage à promouvoir et à respecter la liberté d'association et la reconnaissance du droit de négociation collective. ACENSI veille également à lutter contre la discrimination de toute forme via une politique de recrutement équitable. Enfin, ACENSI s'efforce de créer un terrain de travail favorable au bien-être de ses collaborateurs, au développement de leurs compétences, et à la cohésion au sein du Groupe.

Depuis 2010, pour concrétiser ses valeurs de tolérance, ACENSI a fait le choix de s'engager plus particulièrement pour l'insertion des personnes en situation de handicap dans le monde du travail. ACENSI travaille avec des ESAT (Etablissements et Services d'Aide par le Travail) et contribue ainsi à l'insertion des personnes en situation de handicap dans l'entreprise. ACENSI noue des partenariats avec le secteur adapté dans le cadre de ses projets clients.

-Déontologie et éthique :

ACENSI favorise l'expression des personnalités des consultants, leur créativité et leur différence.

Dans le cadre de sa politique de responsabilité sociale, ACENSI a établi des codes de conduite à respecter par :

La société elle-même en tant que personne morale dans ses activités commerciales.

Tous les collaborateurs dans l'exercice de leurs fonctions.

Acensi détient aussi quelques certificats ISO :

ISO 9001 : met en œuvre et entretient un système de management de la qualité

ISO 27001 : met en œuvre et entretient un système de management de la sécurité de l'information

ISO 14001 : met en œuvre et entretient un système de management environnemental

Mes missions : Cahier des charges

Description de la problématique

Ces dernières années, l'utilisation de la carte de crédit est prédominante dans la société moderne et la fraude à la carte de crédit ne cesse de croître. Les pertes financières dues à la fraude affectent non seulement les commerçants et les banques (par exemple les remboursements) mais aussi les clients individuels. Si la banque perd de l'argent, les clients finissent par payer aussi, par le biais de taux d'intérêt plus élevés, de cotisations plus élevées etc. La fraude peut également affecter la réputation et l'image d'un commerçant, entraînant des pertes non financières qui, bien que difficiles à quantifier à court terme, peuvent devenir visibles à long terme.

Objectif

- Création d'un système de détection de fraude (FDS) en utilisant des modèles de machine Learning et Deep Learning.
- Dynamiser la processus en adaptant le modèle aux éventuels nouveaux comportements des fraudeur grâce à la théorie des jeux

Périmètre du projet (Client cible)

- Banques
- Pole finance

Solutions proposée (Taches à réaliser)

Sur deux sets de données :

- Exploratory data analysis.
- Data engineering.
- Mettre en évidence le problème du non balancement des données.
- Etudier quel indicateur de performance choisir
- Eprouver les algorithmes classiques (SVM, KNN, Régression logistique, gradient boost).
- Essayer les réseaux de neurones avec n couches. Benchmark sur la fonction d'activation.
- Essayer RNN. Benchmark avec plusieurs fonctions d'activation.
- Améliorer tous ces algorithmes avec le sampling pour corriger le problème du balancement.
- Proposer un moteur avec du bagging de tous ces moteurs ML.
- Modéliser le problème sous la forme d'un jeu entre le moteur de détection de fraude et le fraudeur.
- Modéliser le comportement du fraudeur.
- Conclure sur l'apport de la théorie des jeux.

Solutions proposées

Machine learning dans la finance

Etat de l'art

Actuellement, la plupart des entreprises financières, y compris les banques d'investissement, et les entreprises fintech, adoptent et investissent massivement dans l'apprentissage automatique. A l'avenir les institutions financières auront besoin d'un nombre croissant d'experts en machine learning et en datascience.

L'apprentissage automatique en finance est devenu plus important récemment en raison de la disponibilité de grandes quantités de données et d'une puissance de calcul plus abordable. L'utilisation de la datascience et machine learning connaissent une croissance exponentielle dans tous les domaines de la finance.

Le succès de l'apprentissage automatique en finance dépend de la mise en place d'une infrastructure efficace, de l'utilisation efficace, de l'utilisation du bon toolkit et de l'application des bons algorithmes.

Trading algorithmique

Le trading algorithmique est l'utilisation d'algorithmes pour effectuer des transactions de manière autonome. Ses origines remontent aux années 1970. Parfois appelé systèmes de négociation automatisés, ils permettent l'utilisation d'instructions de trading automatisées préprogrammées pour prendre des décisions de trading extrêmement rapides et objectives.

Non seulement des stratégies plus avancées peuvent être utilisées et adaptées en temps réel, mais les techniques basées sur l'apprentissage automatique peuvent offrir davantage de possibilités afin d'obtenir des informations particulières sur les mouvements du marché.

Détection de fraude

Il existe actuellement un risque important pour la sécurité des données en raison de la puissance de calcul élevée, de l'utilisation fréquente d'Internet et de la quantité croissante de données d'entreprise stockées en ligne. Alors que les systèmes antérieurs de détection des fraudes financières dépendaient fortement d'ensembles de règles complexes et robustes, les systèmes modernes de détection des fraudes va au-delà d'une liste de contrôle des facteurs de risque. Elle apprend activement et s'adapte aux nouvelles menaces potentielles (ou réelles) pour la sécurité.

Les services de souscription de banques/assurances

La souscription pourrait être décrite comme un emploi parfait pour l'apprentissage automatique en finance. En effet, le secteur craint que les machines remplacent une grande partie des postes de souscription qui existent aujourd'hui.

Surtout dans les grandes entreprises (grandes banques et compagnies d'assurance cotées en bourse), les algorithmes d'apprentissage automatique peuvent être entraînés sur des millions d'exemples de données de données de consommateurs et de résultats de prêts financiers ou d'assurance, comme le fait qu'une personne normale. Les tendances financières sous-jacentes peuvent être évaluées à l'aide d'algorithmes et analysées en continu afin de détecter les tendances qui pourraient influencer le risque de prêt et de souscription à l'avenir.

Automatisation et chatbot

L'automatisation est manifestement bien adaptée à la finance. Elle réduit la pression que les tâches répétitives et à faible valeur ajoutée sur les employés humains. Elle s'attaque à la routine, aux processus quotidiens, libérant ainsi les équipes pour qu'elles puissent terminer leur travail à forte valeur ajoutée. Ce faisant, elle permet de réaliser d'énormes économies de temps et des économies de coûts.

L'ajout de l'apprentissage automatique à l'automatisation ajoute un autre niveau de soutien aux employés. Avec l'accès aux données pertinentes, l'apprentissage automatique peut faire une analyse approfondie des données pour aider les équipes financières à prendre des décisions difficiles. Dans certains cas, peut même recommander le meilleur plan d'action à approuver et à mettre en œuvre par les employés.

Prévision des prix

La prédiction des prix des actifs est considérée comme le domaine le plus fréquemment discuté et le plus sophistiqué de la finance. Elle permet de comprendre les facteurs qui dirigent le marché et de spéculer sur la performance de ces derniers. Traditionnellement, la prédiction du prix d'actifs était réalisée en analysant les rapports financiers passés et les performances du marché afin de déterminer la position à adopter pour un titre ou une classe d'actifs spécifique. Cependant, avec l'augmentation considérable de la quantité de données financières, les approches traditionnelles pour l'analyse et les stratégies de sélection de titres sont complétées par des approches basées sur le ML.

Opinion mining

L'analyse des sentiments consiste à parcourir d'énormes volumes de données non structurées, telles que des vidéos, des transcriptions, des photos, des fichiers audio, des messages sur les médias sociaux, des articles et des documents commerciaux, afin de déterminer le sentiment du marché. L'analyse des sentiments est cruciale pour toutes les entreprises dans le monde du travail aujourd'hui et constitue un excellent exemple d'apprentissage automatique en finance.

L'utilisation la plus courante de l'analyse des sentiments dans le secteur financier est l'analyse des nouvelles financières - en particulier, la prévision des comportements et des tendances possibles des marchés. Le marché boursier évolue en fonction d'une myriade de facteurs humains, et l'on espère que l'apprentissage automatique sera capable de reproduire et d'améliorer l'intuition humaine sur l'activité financière en découvrant de nouvelles tendances et des signaux révélateurs.

Règlements des transactions

Le règlement des transactions est le processus de transfert des titres sur le compte de l'acheteur et des espèces sur le compte du vendeur à la suite d'une transaction d'un actif financier.

Bien que la majorité des transactions soient réglées automatiquement, avec peu ou pas d'interaction humaine, environ 30 % des transactions doivent être réglées manuellement.

L'utilisation de l'apprentissage automatique permet non seulement d'identifier la raison de l'échec d'une transaction, mais aussi d'analyser pourquoi les transactions ont été rejetées, de fournir une solution et de prédire quelles transactions pourraient échouer à l'avenir.

Blanchiment d'argent

Un rapport des Nations unies estime que le montant annuel du blanchiment d'argent dans le monde représente 2 % à 5 % du PIB mondial. Les techniques d'apprentissage automatique peuvent analyser les données internes, publiques et transactionnelles du réseau élargi d'un client pour tenter de déceler les signes de blanchiment d'argent.

Machine learning, Deep learning, Artificial intelligence et la Data science

Pour la majorité des gens, les termes apprentissage automatique, apprentissage profond, intelligence artificielle et science des données prêtent à confusion. En fait, beaucoup de gens utilisent un terme de manière interchangeable avec les autres.

Figure 1 montre la relation entre ces trois entités. L'apprentissage automatique est un sous-ensemble de l'IA qui consiste en des techniques permettant aux ordinateurs d'identifier des modèles dans les données et de fournir des applications d'IA. L'apprentissage profond, quant à lui, est un sous-ensemble de l'apprentissage automatique qui permet aux ordinateurs de résoudre des problèmes plus complexes.

La science des données n'est pas exactement un sous-ensemble de l'apprentissage automatique, mais elle utilise l'apprentissage automatique, l'apprentissage profond et l'IA pour analyser les données et parvenir à des conclusions exploitables. Elle combine l'apprentissage automatique, l'apprentissage profond et l'IA avec d'autres disciplines telles que l'analyse des big data et le cloud computing.

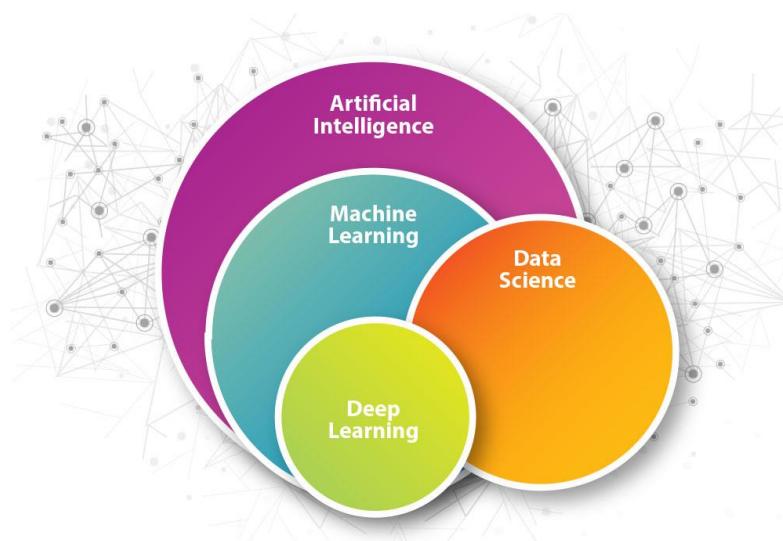


Figure 1: Machine learning deep learning et la data science

Artificial intelligence

L'intelligence artificielle est le domaine d'étude par lequel un ordinateur (et ses systèmes) développe la capacité d'accomplir avec succès des tâches complexes qui nécessitent habituellement l'intelligence humaine. Ces tâches comprennent, entre autres, la perception visuelle, la reconnaissance vocale, la prise de décision et la traduction entre langues. L'IA est généralement définie comme la science qui permet aux ordinateurs de faire des choses qui requièrent de l'intelligence lorsqu'elles sont faites par des humains.

Machine learning

L'apprentissage automatique est une application de l'intelligence artificielle qui donne au système d'IA la capacité d'apprendre automatiquement de l'environnement et d'appliquer ces leçons pour prendre de meilleures décisions. Il existe une variété d'algorithmes que l'apprentissage automatique utilise pour apprendre de manière itérative, décrire et améliorer les données, repérer les modèles, puis effectuer des actions sur ces modèles.

Deep learning

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique qui implique l'étude d'algorithmes liés aux réseaux neuronaux artificiels qui contiennent de nombreuses couches empilées les unes sur les autres. La conception des modèles d'apprentissage profond s'inspire du réseau neuronal biologique du cerveau humain. Il s'efforce d'analyser les données avec une structure logique similaire à la façon dont un humain tire des conclusions.

Data science

La science des données est un domaine interdisciplinaire similaire à l'exploration des données qui utilise des méthodes, des processus et des systèmes scientifiques pour extraire des connaissances ou des idées à partir de données sous diverses formes, structurées ou non. La science des données est différente de la ML et de l'AI car son objectif est d'obtenir des informations et de comprendre les données en utilisant différents outils et techniques scientifiques. Cependant, il existe plusieurs outils et techniques communs à la ML et à la science des données, dont certains sont présentés dans ce projet.

Les types de machine learning

Cette section présente tous les types d'apprentissage automatique. Les deux premiers types sont utilisés dans les différentes études de cas présentées dans ce projet. Comme le montre la figure 2, les trois types d'apprentissage automatique sont l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

Types of Machine Learning

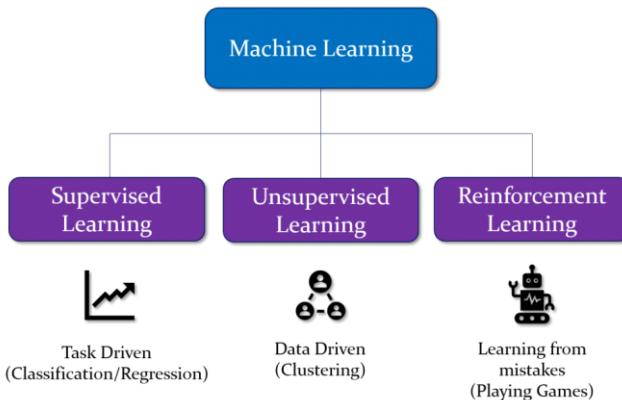


Figure 2: Les types de machine learning

Apprentissage supervisé

L'objectif principal de l'apprentissage supervisé est de former un modèle à partir de données étiquetées qui nous permet de faire des prédictions sur des données non vues ou futures. Ici, le terme supervisé fait référence à un ensemble d'échantillons dont les signaux de sortie souhaités (étiquettes) sont déjà connus. Il existe deux types d'algorithmes d'apprentissage supervisé : la classification et la régression.

Classification

La classification est une sous-catégorie de l'apprentissage supervisé dans laquelle l'objectif est de prédire les étiquettes de classe catégorielle des nouvelles instances sur la base des observations passées.

Régression linéaire

La régression est une autre sous-catégorie de l'apprentissage supervisé utilisée pour la prédiction de résultats continus. Dans la régression, nous disposons d'un certain nombre de variables prédictives (explicatives) et d'une variable de réponse continue (résultat ou cible), et nous essayons de trouver une relation entre ces variables qui nous va nous permettre de prédire un résultat.

La régression linéaire est un modèle linéaire, c'est-à-dire un modèle qui suppose une relation linéaire entre les variables d'entrée (x) et la variable de sortie unique (y). L'objectif de la régression linéaire est d'entraîner un modèle linéaire à prédire un nouveau y à partir d'un x non vu précédemment avec le moins d'erreur possible.

Notre modèle sera une fonction qui prédit y étant donné $x_1, x_2 \dots x_i$:

$y = \beta_0 + \beta_1 x_1 + \dots + \beta_i x_i$ où, β_0 est appelé intercept et $\beta_1 \dots \beta_i$ sont les coefficients de la régression.

La fonction de coût (ou fonction de perte) mesure le degré d'imprécision des prédictions du modèle. La somme des résidus au carré (RSS), telle que définie dans l'équation, mesure la somme au carré de la différence entre la valeur réelle et la valeur prédite et constitue la fonction de coût.

$$RSS = \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^n \beta_j x_{ij})^2$$

La figure 1-3 présente un exemple de régression par rapport à la classification. Le graphique de gauche montre un exemple de régression. La variable de réponse continue est le retour, et les valeurs observées sont tracées par rapport aux résultats prédis. Sur la droite, le résultat est une étiquette de classe catégorielle.

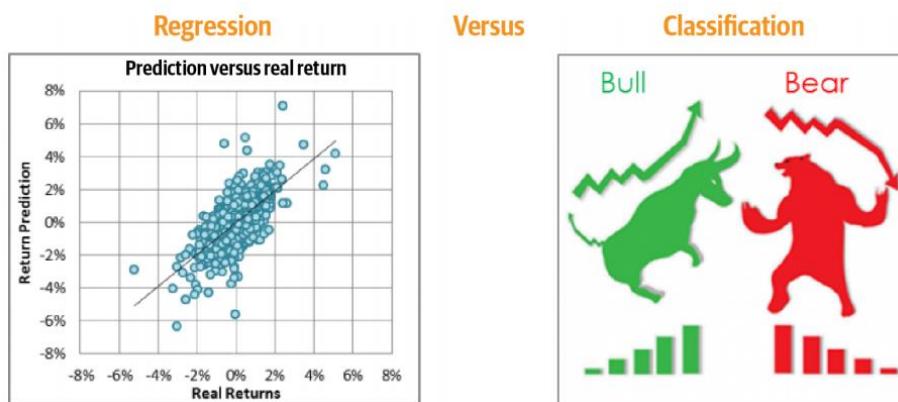


Figure 3: Régression linéaire vs Classification

Régression logistique

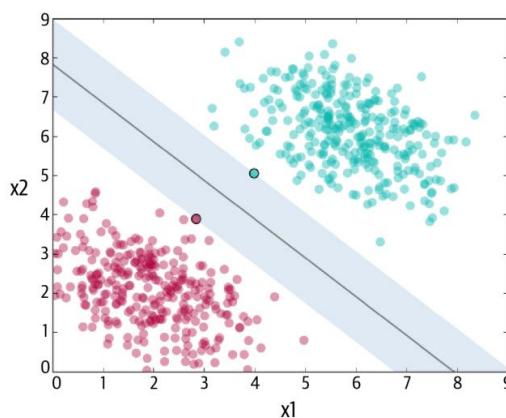
Si nous entraînons un modèle de régression linéaire sur plusieurs exemples où $Y=0$ ou 1 , nous pourrions finir par prédire certaines probabilités inférieures à zéro ou supérieures à un, ce qui n'a pas de sens. Au lieu de cela, nous utilisons un modèle de régression logistique, qui est une modification de la régression linéaire qui s'assure de produire une probabilité entre zéro et un en appliquant la fonction sigmoïde.

L'équation 4-2 montre l'équation d'un modèle de régression logistique. Comme dans le cas de la régression linéaire, les valeurs d'entrée (x) sont combinées linéairement à l'aide de poids ou de coefficients pour prédire une valeur de sortie (y). La sortie de l'équation 4-2 est une probabilité qui est transformée en une valeur binaire (0 ou 1) pour obtenir la prédiction du modèle.

$$y = \frac{\exp(\beta_0 + \beta_1 x_1 + \dots + \beta_i x_i)}{1 + \exp(\beta_0 + \beta_1 x_1 + \dots + \beta_i x_i)}$$

Support Vector Machine (SVM)

L'objectif de l'algorithme de la machine à vecteurs de support (SVM) est de maximiser la marge (représentée par la zone ombrée dans la figure 4-3), qui est définie comme la distance entre l'hyperplan de séparation (ou frontière de décision) et les échantillons d'apprentissage les plus proches de cet hyperplan, appelés vecteurs de support. La marge est calculée comme la distance perpendiculaire de la ligne aux seuls points les plus proches, comme le montre la figure 4-3. Par conséquent, le SVM calcule une frontière de marge maximale qui conduit à une partition homogène de tous les points de données.



K-nearest neighbors

K-nearest neighbors (KNN) est considéré comme un "apprenant paresseux", car le modèle ne nécessite aucun apprentissage. Pour un nouveau point de données, les prédictions sont effectuées en recherchant dans

l'ensemble de l'apprentissage les K instances les plus similaires (les voisins) et en résumant la variable de sortie pour ces K instances.

Pour déterminer lesquelles des K instances de l'ensemble de données d'apprentissage sont les plus similaires à une nouvelle entrée, une mesure de distance est utilisée. La mesure de distance la plus populaire est la distance euclidienne, qui est calculée comme la racine carrée de la somme des différences au carré entre un point a et un point b sur tous les attributs d'entrée i, et qui est représentée par $d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$. La distance euclidienne est une bonne mesure de distance à utiliser si les variables d'entrée sont de type similaire. Une autre mesure de distance est la distance de Manhattan, dans laquelle la distance entre un point a et un point b est représentée par $d(a, b) = \sum_{i=1}^n |a_i - b_i|$. La distance de Manhattan est une bonne mesure à utiliser si les variables d'entrée ne sont pas de type similaire.

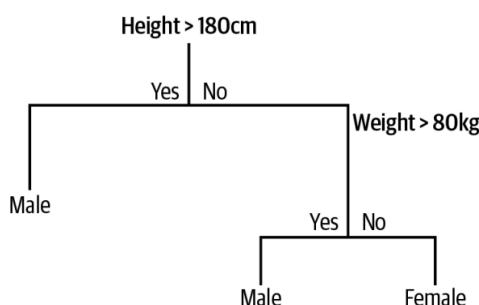
Les étapes de KNN peuvent être résumées comme suit :

1. Choisir le nombre de K et une métrique de distance.
2. Trouver les K plus proches voisins de l'échantillon que l'on veut classer.
3. Attribuer l'étiquette de classe par vote majoritaire.

Arbres décisionnels

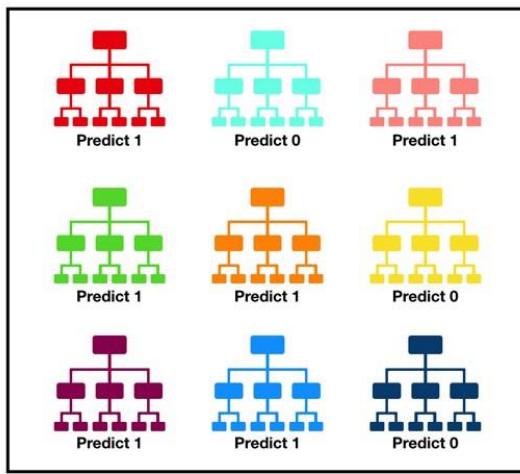
En termes les plus généraux, l'objectif d'une analyse via des algorithmes de construction d'arbres est de déterminer un ensemble de conditions logiques (de type "if-else") qui permettent une prédiction ou une classification précise des cas. Nous pouvons considérer que ce modèle décompose nos données et prend une décision en posant une série de questions. Cet algorithme est le fondement des méthodes d'ensemble telles que la forêt aléatoire et la méthode d'amplification par gradient.

Le modèle peut être représenté par un arbre binaire (ou arbre de décision), où chaque nœud est une variable d'entrée x avec un point de séparation et chaque feuille contient une variable de sortie y pour la prédiction.



Random forest

La foret aléatoire, comme son nom l'indique, se compose d'un grand nombre d'arbres décisionnels individuels qui fonctionnent comme un ensemble. Chaque arbre de la foret émet une prediction de la classe. La classe ayant reçu le plus de votes devient la prediction de notre modèle



Tally: Six 1s and Three 0s

Prediction: 1

Apprentissage non-supervisé

L'apprentissage non supervisé est un type d'apprentissage automatique utilisé pour tirer des conclusions à partir d'ensembles de données constitués de données d'entrée sans réponses étiquetées. Il existe deux types d'apprentissage non supervisé : la réduction de la dimensionnalité et le clustering.

Réduction de dimensionnalité

La réduction de la dimensionnalité est le processus qui consiste à réduire le nombre de caractéristiques, ou de variables, dans un ensemble de données tout en préservant l'information et la performance globale du modèle. Il s'agit d'un moyen courant et puissant de traiter les ensembles de données qui ont un grand nombre de dimensions.

La figure 4 illustre ce concept, où la dimension des données est convertie de deux dimensions (X_1 et X_2) en une seule dimension (Z_1). Z_1 convertit les informations similaires intégrées dans X_1 et X_2 et réduit la dimension des données.

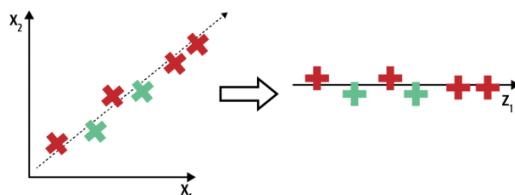


Figure 4: Réduction de dimensionnalité

Clustering

Le clustering est une sous-catégorie de techniques d'apprentissage non supervisées qui nous permet de découvrir des structures cachées dans les données. L'objectif du clustering est de trouver un regroupement naturel dans les données, de sorte que les éléments d'un même cluster soient plus similaires les uns aux autres que ceux de clusters différents.

Un exemple de clustering est illustré à la figure 5, où l'on peut voir que l'ensemble des données sont regroupées en deux groupes distincts par le biais de l'algorithme de clustering.

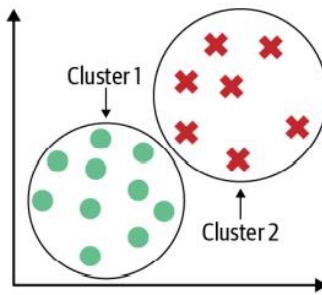
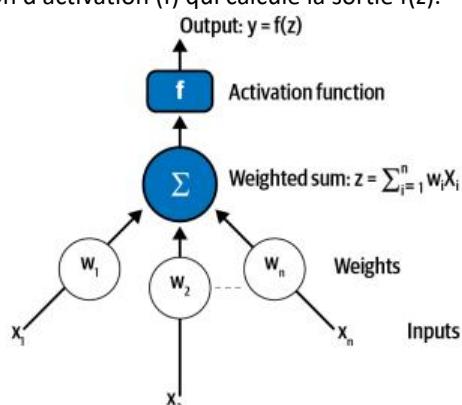


Figure 5: Clustering

Réseaux de neurones

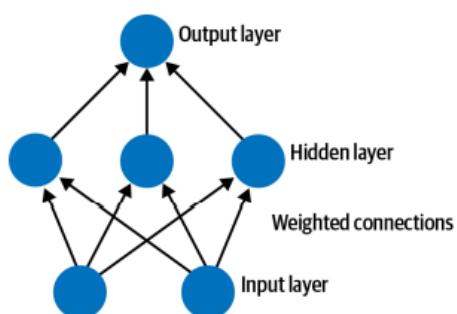
Les ANN contiennent plusieurs neurones disposés en couches. Un ANN passe par une phase d'apprentissage en comparant la sortie modélisée à la sortie souhaitée, où il apprend à reconnaître des modèles dans les données. Passons en revue les composants des ANN.

Les éléments constitutifs des ANN sont les neurones (également appelés neurones artificiels, nœuds ou perceptrons). Les neurones ont une ou plusieurs entrées et une sortie. Il est possible de construire un réseau de neurones pour calculer des propositions logiques complexes. Les fonctions d'activation de ces neurones créent des mappages fonctionnels complexes et non linéaires entre les entrées et les sorties. Un neurone prend une entrée ($x_1, x_2 \dots x_n$), applique les paramètres d'apprentissage pour générer une somme pondérée (z), puis transmet cette somme à une fonction d'activation (f) qui calcule la sortie $y = f(z)$.



Couches

La sortie $f(z)$ d'un seul neurone (comme le montre la figure 3-1) ne permettra pas de modéliser des tâches complexes. Ainsi, afin de traiter des structures plus complexes, nous avons plusieurs couches de ces neurones. À mesure que nous empilons les neurones horizontalement et verticalement, la classe de fonctions que nous pouvons obtenir devient de plus en plus complexe. La figure montre l'architecture d'un ANN avec une couche d'entrée, une couche de sortie et une couche cachée.



Développer un modèle de machine learning en Python

En ce qui concerne l'apprentissage automatique, il existe de nombreux algorithmes et langages de programmation. Cependant, l'écosystème Python est l'un des langages de programmation les plus dominants et dont la croissance est la plus rapide pour l'apprentissage automatique. Compte tenu de la popularité et du taux d'adoption élevé de Python, nous l'utiliserons comme principal langage de programmation tout au long du projet.

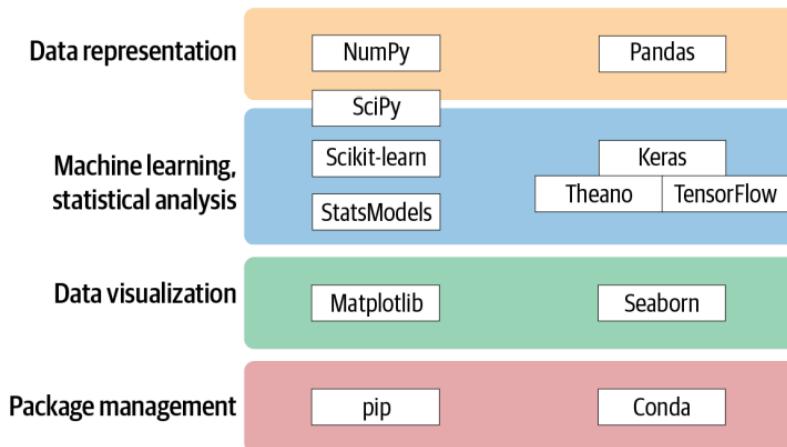
Pourquoi Python

Quelques raisons de la popularité de Python dans la data science :

- Syntaxe de haut niveau (par rapport aux langages de plus bas niveau que sont C, Java et C++)
- Les applications peuvent être développées en écrivant moins de lignes de code, ce qui rend Python est attrayant pour les débutants comme pour les programmeurs avancés.
- Cycle de vie de développement efficace.
- Large collection de bibliothèques open-source gérées par la communauté.
- Forte portabilité.

Les packages de python en machine Learning

Les packages principaux utilisés en machine Learning sont résumés dans la figure 6



Ci-dessous un petit résumé de chaque package :

Numpy

Prend en charge les grands tableaux multidimensionnels ainsi qu'une vaste collection de fonctions mathématiques.

Pandas

Une bibliothèque pour la manipulation et l'analyse des données. Elle offre, entre autres, des structures de données pour gérer les tableaux et les outils pour les manipuler.

Matplotlib

Une bibliothèque de traçage qui permet de créer des graphiques et des tracés en 2D.

SciPy

La combinaison de NumPy, Pandas et Matplotlib est généralement appelée SciPy. SciPy est un écosystème de bibliothèques Python pour les mathématiques, les sciences et l'ingénierie.

Scikit-learn

Une bibliothèque d'apprentissage automatique offrant un large éventail d'algorithmes et d'utilitaires.

Stats models

Un module Python qui fournit des classes et des fonctions pour l'estimation de nombreux modèles statistiques différents, ainsi que pour la réalisation de tests statistiques et l'exploration statistique des données.

Tensorflow et Theano

Bibliothèques de programmation de flux de données qui facilitent le travail avec les réseaux neuronaux.

Keras

Une bibliothèque de réseaux de neurones artificiels qui peut servir d'interface simplifiée aux paquets TensorFlow/Theano.

Seaborn

Une bibliothèque de visualisation de données basée sur Matplotlib. Elle fournit une interface de haut niveau pour dessiner des graphiques statistiques attrayants et informatifs

Pip et conda

Ce sont des gestionnaires de paquets. Python. pip est un gestionnaire de paquets qui facilite l'installation, la mise à niveau et la désinstallation de paquets Python. Conda est un gestionnaire de paquets qui gère les paquets Python ainsi que les dépendances des bibliothèques en dehors des paquets Python.

Étapes du développement d'un modèle dans l'écosystème Python

Il est essentiel de traiter les problèmes d'apprentissage automatique de bout en bout. L'apprentissage automatique appliquée ne prendra vie que si les étapes du début à la fin sont bien définies.

La figure 7 présente les grandes lignes d'un modèle de projet d'apprentissage automatique en sept étapes qui peut être utilisé pour lancer n'importe quel modèle d'apprentissage automatique en Python. Les premières étapes comprennent l'analyse exploratoire des données et la préparation des données, qui sont des étapes typiques de la science des données visant à extraire du sens et des informations des données. Ces étapes sont suivies de l'évaluation du modèle, du réglage fin et de la finalisation du modèle.

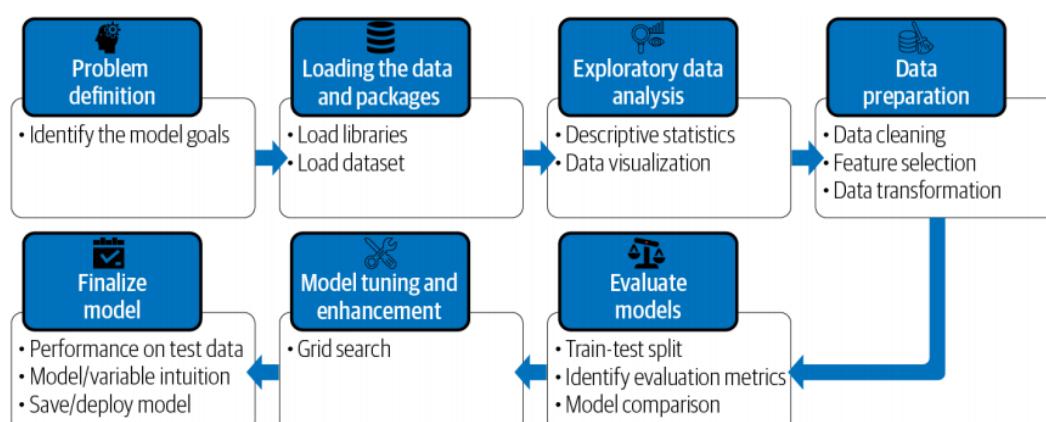


Figure 7: Etapes de développement d'un modèle de machine Learning

Performance du modèle

Dans cette section, nous allons développer ce processus en discutant des composants clés de l'évaluation de la performance du modèle, à savoir l'overfitting, la validation croisée et les métriques d'évaluation.

Overfitting et Underfitting

Un problème courant dans l'apprentissage automatique est l'overfitting, qui se définit par l'apprentissage d'une fonction qui explique parfaitement les données d'apprentissage à partir desquelles le modèle a été appris, mais qui ne se généralise pas bien aux données de test non vues. L'ajustement excessif se produit lorsqu'un modèle apprend trop des données d'apprentissage au point qu'il commence à capturer des idiosyncrasies qui ne sont pas représentatives des modèles du monde réel. Cela devient particulièrement problématique lorsque nous rendons nos modèles de plus en plus complexes. Le sous-ajustement est un problème connexe dans lequel le modèle n'est pas suffisamment complexe pour capturer la tendance sous-jacente des données. La figure 8 illustre l'overfitting et le underfitting. Le panneau de gauche de la figure 8 montre un modèle de régression linéaire ; une ligne droite est clairement sous-adaptée à la fonction réelle. Le panneau central montre qu'un polynôme de degré élevé s'approche raisonnablement bien de la relation réelle. En revanche, un polynôme de degré très élevé s'adapte presque parfaitement au petit échantillon et donne les meilleurs résultats sur les données

d'apprentissage, mais il n'est pas généralisable et ne permettrait pas d'expliquer correctement un nouveau point de données.

Les concepts d'overfitting et de underfitting sont étroitement liés au compromis biais-variance. Le biais désigne l'erreur due à des hypothèses trop simplistes ou à des hypothèses erronées dans l'algorithme d'apprentissage. Le biais entraîne une sous-adaptation des données, comme le montre le panneau gauche de la figure 8. Un biais élevé signifie que notre algorithme d'apprentissage manque des tendances importantes parmi les caractéristiques. La variance fait référence à l'erreur due à un modèle trop complexe qui tente d'ajuster les données d'apprentissage aussi précisément que possible. Dans les cas de variance élevée, les valeurs prédictives du modèle sont extrêmement proches des valeurs réelles de l'ensemble d'apprentissage. Une variance élevée donne lieu à un sur-ajustement, comme le montre le panneau de droite de la figure 8. En fin de compte, pour avoir un bon modèle, nous devons avoir un biais et une variance faibles.

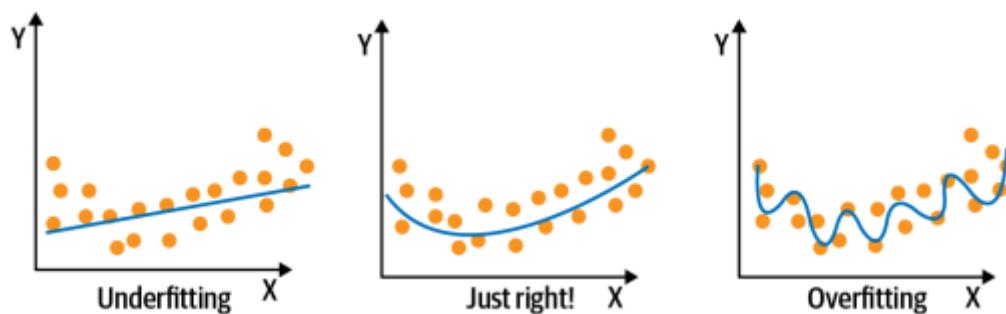


Figure 8: Overfitting vs Underfitting

Cross validation

L'un des défis de l'apprentissage automatique consiste à former des modèles capables de bien se généraliser à des données non vues. L'idée principale de la validation croisée est de diviser les données une ou plusieurs fois de manière à ce que chaque division soit utilisée une fois comme ensemble de validation et que le reste soit utilisé comme ensemble d'entraînement : une partie des données (l'échantillon d'entraînement) est utilisée pour entraîner l'algorithme, et l'autre partie (l'échantillon de validation) est utilisée pour estimer le risque de l'algorithme. La validation croisée nous permet d'obtenir des estimations fiables de l'erreur de généralisation du modèle. Il est plus facile de la comprendre à l'aide d'un exemple. Lors de la validation croisée à k plis, nous divisons aléatoirement les données d'entraînement en k plis. Ensuite, nous entraînons le modèle à l'aide de k-1 plis et évaluons la performance sur le kième pli. Nous répétons ce processus k fois et faisons la moyenne des scores obtenus.



Figure 9: Cross validation

Metrics d'évaluation

Le choix des métriques à utiliser influence la façon dont les performances des algorithmes d'apprentissage automatique sont mesurées et comparées. Les métriques influencent à la fois la façon dont vous pondérez l'importance des différentes caractéristiques dans les résultats et votre choix final d'algorithme.

Les principales métriques d'évaluation pour la régression et la classification sont illustrées dans la Figure 10.

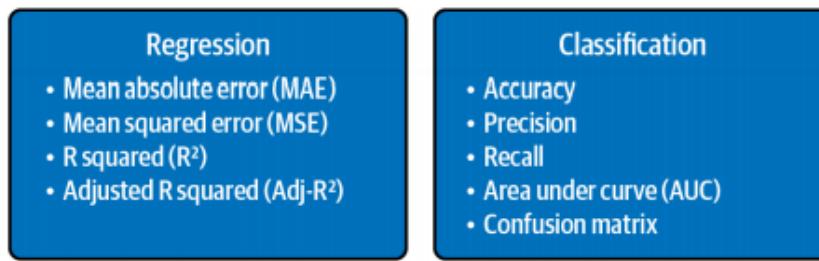


Figure 10: Métrics d'évaluation. Regression vs Classification

MAE

L'erreur absolue moyenne (MAE) est la somme des différences absolues entre les prédictions et les valeurs réelles. La MAE est un score linéaire, ce qui signifie que toutes les différences individuelles sont pondérées de manière égale dans la moyenne. Elle donne une idée de la mesure dans laquelle les prédictions étaient fausses. La mesure donne une idée de l'ampleur de l'erreur, mais aucune idée de la direction (par exemple, l'écart entre la valeur réelle et la valeur prédite).

MSE

L'erreur quadratique moyenne (EQM) représente l'écart type de l'échantillon des différences entre les valeurs prédites et les valeurs observées (appelées résidus). Elle ressemble beaucoup à l'erreur absolue moyenne en ce sens qu'elle donne une idée générale de l'ampleur de l'erreur. En prenant la racine carrée de l'erreur quadratique moyenne, les unités sont ramenées aux unités originales de la variable de sortie et peuvent être significatives pour la description et la présentation. C'est ce qu'on appelle l'erreur quadratique moyenne (RMSE).

R carré

La métrique R^2 fournit une indication de la "bonne adéquation" des prédictions à la valeur réelle. Dans la littérature statistique, cette mesure est appelée le coefficient de détermination. Il s'agit d'une valeur comprise entre zéro et un, pour une absence d'ajustement et un ajustement parfait, respectivement.

R carré ajustée.

Tout comme le R^2 , le R^2 ajusté montre également dans quelle mesure les termes s'adaptent à une courbe ou à une ligne, mais il est ajusté en fonction du nombre de termes dans un modèle. le nombre de termes dans un modèle. Il est donné par la formule suivante :

$$R_{adj}^2 = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1} \right]$$

Où n est le nombre total d'observations et k est le nombre de prédicteurs. Le R^2 ajusté sera toujours inférieur ou égal au R^2 .

Matrice de confusion

Une matrice de confusion présente les performances d'un algorithme d'apprentissage. La matrice de confusion est simplement une matrice carrée qui indique le nombre de vrais positifs (TP), de vrais négatifs (TN), de faux positifs (FP) et de faux négatifs (FN) prédits par un classificateur.

Vrais positifs

Prédictions positives qui sont réellement positives

Faux positifs

Prédictions positives qui sont réellement négatives

Vrais négatifs

Prédictions négatives qui sont réellement négatives

Faux négatifs

Prédictions négatives qui sont réellement positives

La différence entre trois métriques d'évaluation couramment utilisées pour la classification, l'exactitude, la précision et le rappel, est illustrée à la figure 11.

$$\text{Precision} = \frac{\text{True positive}}{\text{Actual results}} \quad \text{or} \quad \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

$$\text{Recall} = \frac{\text{True positive}}{\text{Predictive results}} \quad \text{or} \quad \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

$$\text{Accuracy} = \frac{\text{True positive} + \text{True negative}}{\text{Total}}$$

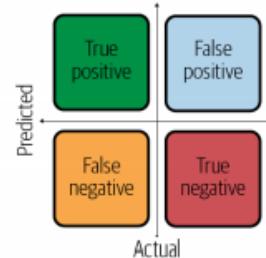


Figure 11: Matrice de confusion

Accuracy

Comme le montre la figure 8, la précision est le nombre de prédictions correctes effectuées par rapport à toutes les prédictions effectuées. Il s'agit de la métrique d'évaluation la plus courante pour les problèmes de classification, mais aussi la plus mal utilisée. Elle convient le mieux lorsqu'il y a un nombre égal d'observations dans chaque classe (ce qui est rarement le cas) et lorsque toutes les prédictions et les erreurs de prédiction associées sont d'importance égale, ce qui n'est souvent pas le cas.

Précision

La précision est le pourcentage d'instances positives sur le total des instances positives prédictes. Ici, le dénominateur est la prédiction du modèle effectuée comme positive à partir de l'ensemble des données. La précision est une bonne mesure à déterminer lorsque le coût des faux positifs est

Recall

Le rappel (ou sensibilité ou taux de vrais positifs) est le pourcentage d'instances positives par rapport au total des instances positives réelles. Par conséquent, le dénominateur (vrais positifs + faux négatifs) est le nombre réel d'instances positives présentes dans l'ensemble de données. Le rappel est une bonne mesure lorsqu'il y a un coût élevé associé aux faux négatifs.

ROC

L'aire sous la courbe ROC (AUC) est une mesure d'évaluation pour les problèmes de classification binaire. La courbe ROC est une courbe de probabilité, et l'AUC représente le degré ou la mesure de la séparabilité. Elle indique dans quelle mesure le modèle est capable de faire la distinction entre les classes. Plus l'AUC est élevée, plus le modèle est capable de prédire des zéros comme zéros et des uns comme uns. Une AUC de 0,5 signifie que le modèle n'a aucune capacité de séparation des classes. L'interprétation probabiliste du score AUC est la suivante : si vous choisissez au hasard un cas positif et un cas négatif, la probabilité que le cas positif surpassé le cas négatif selon le classificateur est de 0,5. négatif selon le classificateur est donnée par l'AUC.

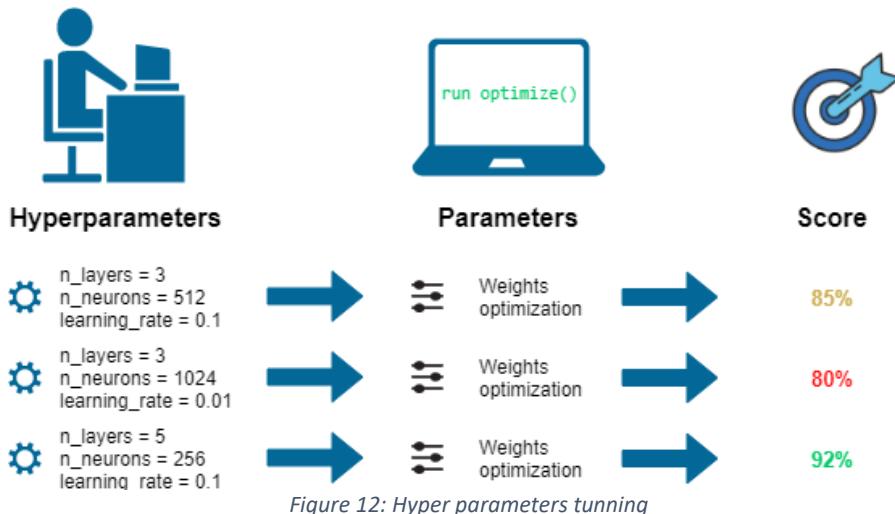
Hyper parameter tunning

Choisir les bons hyperparamètres pour votre algorithme d'apprentissage automatique est une tâche cruciale, car cela peut faire une grande différence sur les performances d'un modèle.

Les modèles d'apprentissage automatique sont composés de deux types de paramètres différents :

- Hyperparamètres = sont tous les paramètres qui peuvent être fixés arbitrairement par l'utilisateur avant de commencer l'apprentissage (par exemple, le taux d'apprentissage, le paramètre de régularisation, la taille du lot dans la descente de gradient mini-batch...).
- Paramètres du modèle = sont plutôt appris pendant l'apprentissage du modèle (par exemple, les poids dans la régression linéaire, les réseaux neuronaux...).

Les paramètres du modèle définissent comment utiliser les données d'entrée pour obtenir la sortie désirée et sont appris au moment de la formation. Au lieu de cela, les Hyperparamètres déterminent comment notre modèle est structuré en premier lieu. Le réglage des hyperparamètres est un type de problème d'optimisation. Nous disposons d'un ensemble d'hyperparamètres et nous cherchons à trouver la bonne combinaison de leurs valeurs, ce qui peut nous aider à trouver le minimum (par exemple, la perte) ou le maximum (par exemple, la précision) d'une fonction.



Grid search

La recherche par grille est une méthode par laquelle nous créons des ensembles de valeurs possibles pour chaque hyperparamètre, puis nous les testons les uns par rapport aux autres dans une "grille".

Voici le déroulement des opérations :

- Définir une grille sur n dimensions, où chacune d'entre elles correspond à un hyperparamètre. Par exemple n = (learning_rate, dropout_rate, batch_size).
- Pour chaque dimension, définir la gamme des valeurs possibles : par exemple, batch_size = [4, 8, 16, 32, 64, 128, 256].
- Rechercher toutes les configurations possibles et attendre les résultats pour établir la meilleure : par exemple C1 = (0.1, 0.3, 4) -> acc = 92%, C2 = (0.1, 0.35, 4) -> acc = 92.3%, et ainsi de suite.

L'image ci-dessous illustre une simple recherche en grille sur différentes valeurs de deux hyperparamètres. Pour chaque hyperparamètre, 10 valeurs différentes sont considérées, de sorte qu'un total de 100 combinaisons différentes sont évaluées et comparées.

Les contours bleus indiquent les régions ayant de bons résultats, tandis que les rouges indiquent les régions ayant de mauvais résultats.

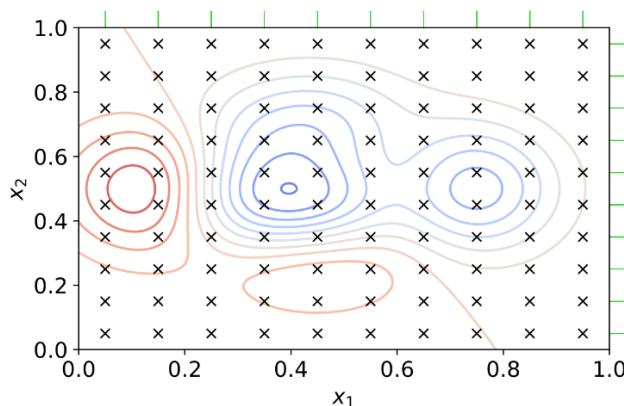


Figure 13: Illustration d'une grid search

Diverses techniques utilisées

One hot encoding :

L'encodage 1 parmi n porte sur le fait d'encoder une variable à n états sur n bits. Un seul bit prendra la valeur 1 qui est le numéro du bit dont l'état est pris par la variable.

L'avantage de ce type d'encodage est simple : pour passer d'un état à l'autre, il faudra faire deux transitions uniquement, le 1 devient 0 et le 0 devient 1. Par contre l'inconvénient est que l'espace de la mémoire augmente linéairement avec le nombre d'état au lieu d'une augmentation logarithmique.

Label encoding

Le label encoding remplace les valeurs catégoriques par une version numérique entre 0 et n-1 avec n le nombre de classes. L'avantage est qu'on garde la même dimension des attributs

Sampling

Le ré-échantillonnage peut améliorer les performances du modèle en cas de déséquilibre des ensembles de données :

- Sous-échantillonnage de la classe majoritaire (cas non frauduleux) Méthode simple pour ajuster des données déséquilibrées Effectuez des tirages aléatoires à partir des observations non frauduleuses, afin de faire correspondre les occurrences des observations de fraude.

- Sur-échantillonnage de la classe minoritaire (cas de fraude)

Prendre des tirages aléatoires parmi les cas de fraude et copier ces observations pour augmenter le nombre d'échantillons de fraude

Les deux méthodes permettent d'obtenir un équilibre entre les cas de fraude et les cas non frauduleux
 Inconvénients du sous-échantillonnage aléatoire : beaucoup d'informations sont perdues Avec le sur-échantillonnage, le modèle sera formé sur un grand nombre de doublons

Nos dataset

La fraude est l'un des problèmes les plus importants auxquels le secteur financier est confronté. Elle est incroyablement coûteuse. Selon une étude, on estime que l'organisation type perd chaque année 5 % de ses revenus annuels à cause de la fraude. Si l'on applique ce chiffre au produit mondial brut de 2017, estimé à 79 600 milliards de dollars, on obtient des pertes mondiales potentielles pouvant atteindre 4 000 milliards de dollars. La détection de la fraude est une tâche intrinsèquement adaptée à l'apprentissage automatique, car les modèles basés sur l'apprentissage automatique peuvent analyser d'énormes ensembles de données transactionnelles, détecter une activité inhabituelle et identifier tous les cas susceptibles d'être frauduleux. En outre, les calculs de ces modèles sont plus rapides que les approches traditionnelles basées sur des règles. En collectant des données provenant de diverses sources, puis en les mettant en correspondance avec des points de déclenchement, les solutions d'apprentissage automatique sont capables de découvrir le taux de défaillance ou la propension à la fraude pour chaque client potentiel et chaque transaction, fournissant ainsi des alertes et des informations clés aux institutions financières.

Kaggle

Définition du problème

Dans cette étude de cas, nous utiliserons divers modèles basés sur la classification pour détecter si une transaction est un paiement normal ou une fraude.

Les points forts de cette étude de cas sont :

- Le traitement de données déséquilibrées par le sous-échantillonnage et le suréchantillonnage des données.
- Sélection de la bonne métrique d'évaluation, étant donné que l'un des principaux objectifs est de réduire les faux négatifs (cas dans lesquels les transactions frauduleuses passent inaperçues).

Dans le cadre de classification défini pour cette étude de cas, la variable de réponse (ou cible) porte le nom de colonne "Classe". Cette colonne a une valeur de 1 en cas de fraude et une valeur de 0 dans le cas contraire. L'ensemble de données utilisé est obtenu auprès de Kaggle. Cet ensemble de données contient des transactions effectuées par des titulaires de cartes euro européennes qui ont eu lieu pendant deux jours en septembre 2013, avec 492 cas de fraude sur 284 807 transactions.

L'ensemble de données a été rendu anonyme pour des raisons de confidentialité. Étant donné que certains noms de caractéristiques ne sont pas fournis (c'est-à-dire qu'ils sont appelés V1, V2, V3, etc.)

Exploratory data analysis

Statistiques descriptives

La première chose que nous devons faire est de rassembler un sens de base de nos données. Rappelez-vous, à l'exception de la transaction et du montant, nous ne connaissons pas les noms des autres colonnes. La seule chose que nous savons est que les valeurs de ces colonnes ont été mises à l'échelle. Examinons la forme et les colonnes des données

Output

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.360	-0.073	2.536	1.378	-0.338	0.462	0.240	0.099	0.364	...	-0.018	0.278	-0.110	0.067	0.129	-0.189	0.134	-0.021	149.62	0
1	0.0	1.192	0.266	0.166	0.448	0.060	-0.082	-0.079	0.085	-0.255	...	-0.226	-0.639	0.101	-0.340	0.167	0.126	-0.009	0.015	2.69	0
2	1.0	-1.358	-1.340	1.773	0.380	-0.503	1.800	0.791	0.248	-1.515	...	0.248	0.772	0.909	-0.689	-0.328	-0.139	-0.055	-0.060	378.66	0
3	1.0	-0.966	-0.185	1.793	-0.863	-0.010	1.247	0.238	0.377	-1.387	...	-0.108	0.005	-0.190	-1.176	0.647	-0.222	0.063	0.061	123.50	0
4	2.0	-1.158	0.878	1.549	0.403	-0.407	0.096	0.593	-0.271	0.818	...	-0.009	0.798	-0.137	0.141	-0.206	0.502	0.219	0.215	69.99	0

Comme on peut le voir, les variables sont encodées (V1-V28).

```

1  Time      float64
2  V1       float64
3  V2       float64
4  V3       float64
5  V4       float64
6  V5       float64
7  V6       float64
8  V7       float64
9  V8       float64
10 V9      float64
11 V10     float64
12 V11     float64
13 V12     float64
14 V13     float64
15 V14     float64
16 V15     float64
17 V16     float64
18 V17     float64
19 V18     float64
20 V19     float64
21 V20     float64
22 V21     float64
23 V22     float64
24 V23     float64
25 V24     float64
26 V25     float64
27 V26     float64
28 V27     float64
29 V28     float64
30 Amount   float64
31 Class    int64

```

```

class_names = {0:'Not Fraud', 1:'Fraud'}
print(dataset.Class.value_counts().rename(index = class_names))

```

Output

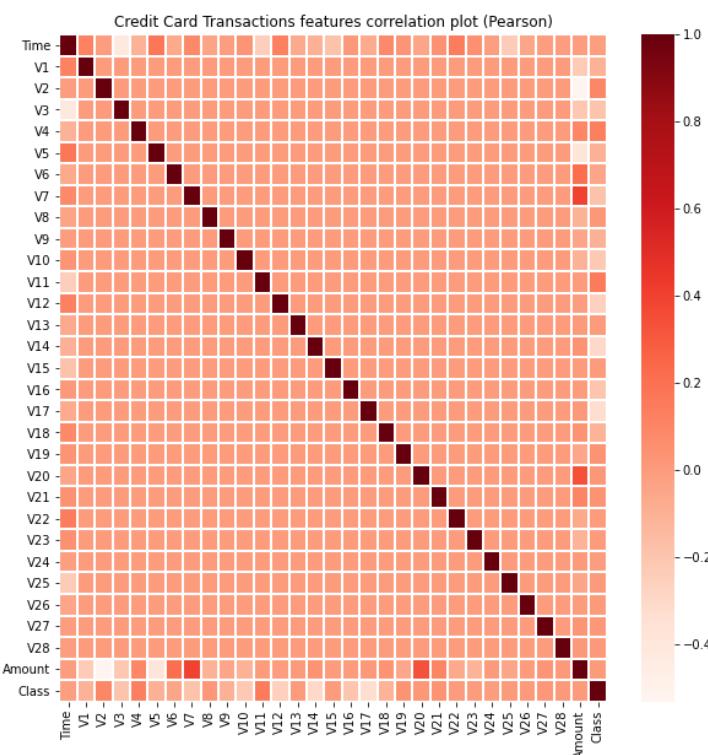
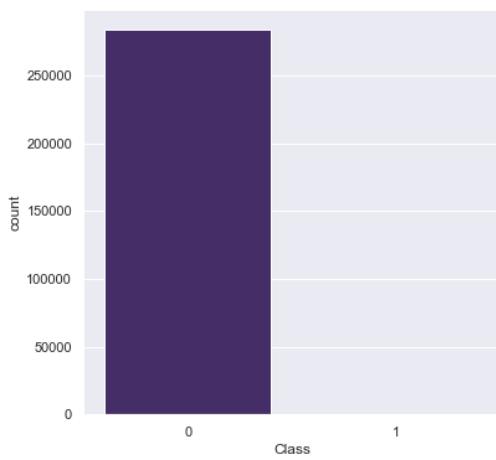
```

Not Fraud    284315
Fraud        492
Name: Class, dtype: int64

```

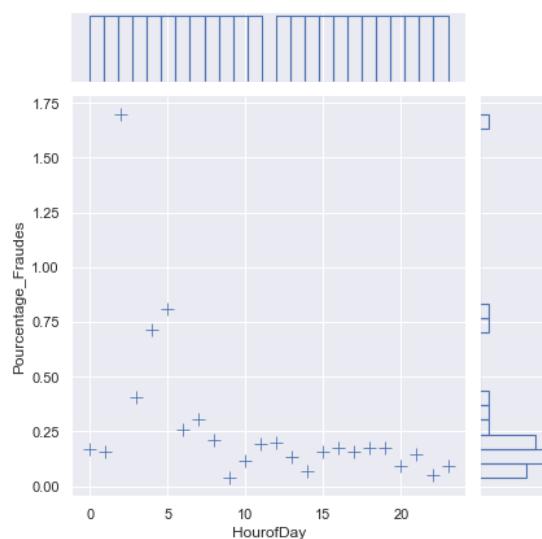
Remarquez le déséquilibre flagrant des étiquettes de données. La plupart des transactions ne sont pas frauduleuses. Si nous utilisons cet ensemble de données comme base pour notre modélisation, la plupart des modèles ne mettront pas suffisamment l'accent sur les signaux de fraude ; les points de données non frauduleux noieront tout le poids des signaux de fraude. En l'état, nous pourrions rencontrer des difficultés à modéliser la

prédition de la fraude, ce déséquilibre conduisant les modèles à simplement supposer que toutes les transactions ne sont pas frauduleuses. Ce serait un résultat inacceptable.



Ci-dessus la matrice de corrélation.

La matrice de corrélation est utilisée principalement pour évaluer la dépendance entre une ou plusieurs variables en même temps. Si la corrélation s'approche à 1 ou -1 cela veut dire que les deux variables ont le même comportement, on peut donc en supprimer une pour réduire la dimension de notre dataset.



Ici on a transformé le temps en heure et sur l'axe des abscisses le taux de fraudes. On peut voir que le taux de fraude a tendance à croître la nuit.

Préparation des données

Ces données proviennent de Kaggle et sont déjà dans un format nettoyé sans aucune ligne ou colonne vide. Le nettoyage ou la catégorisation des données est inutile.

Séparer les données

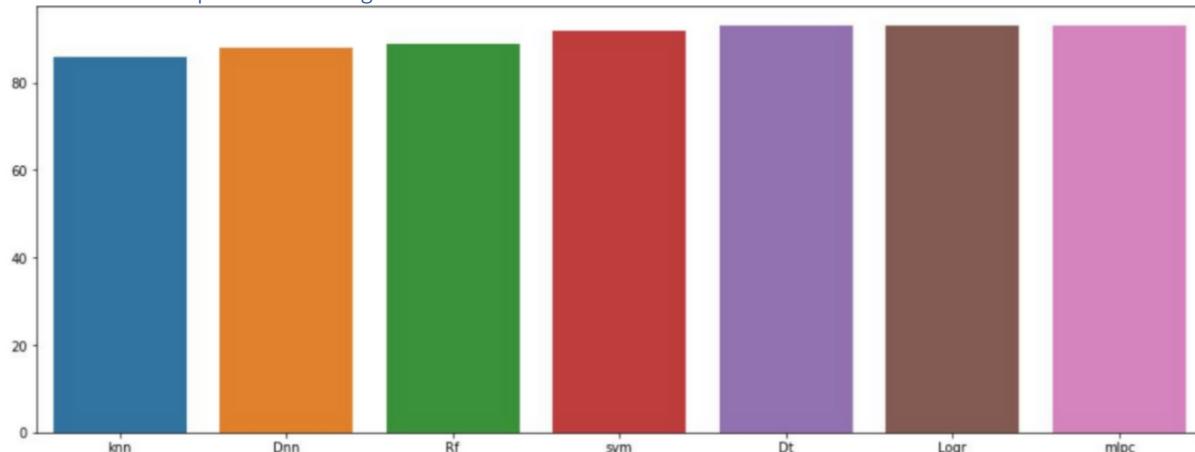
Comme décrit au chapitre 2, il est judicieux de diviser l'ensemble de données original en ensembles d'entraînement et de test. L'ensemble de test est un échantillon de données que nous réservons pour notre analyse et notre modélisation. Nous l'utilisons à la fin de notre projet pour confirmer la précision de notre modèle final. C'est le test final qui nous donne confiance dans nos estimations de précision sur des données non vues. Nous utiliserons 80 % de l'ensemble des données pour l'apprentissage du modèle et 20 % pour le test.

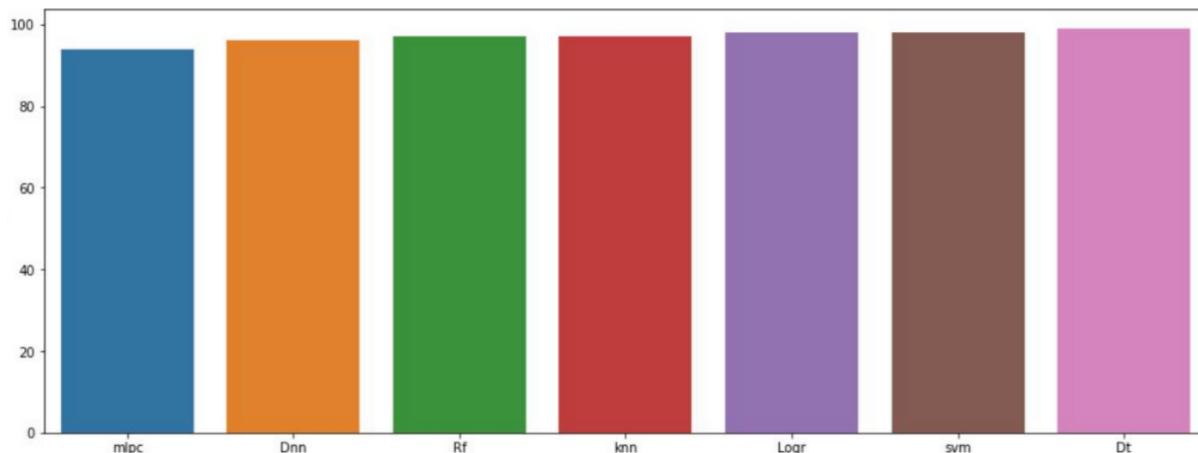
Entrainement du modèle et metric evaluation

Comme décrit dans la section théorique, on entraîne le modèle sur un set estimé à 80% du volume total et 20% pour le set de test.

Récapitulatif des résultats

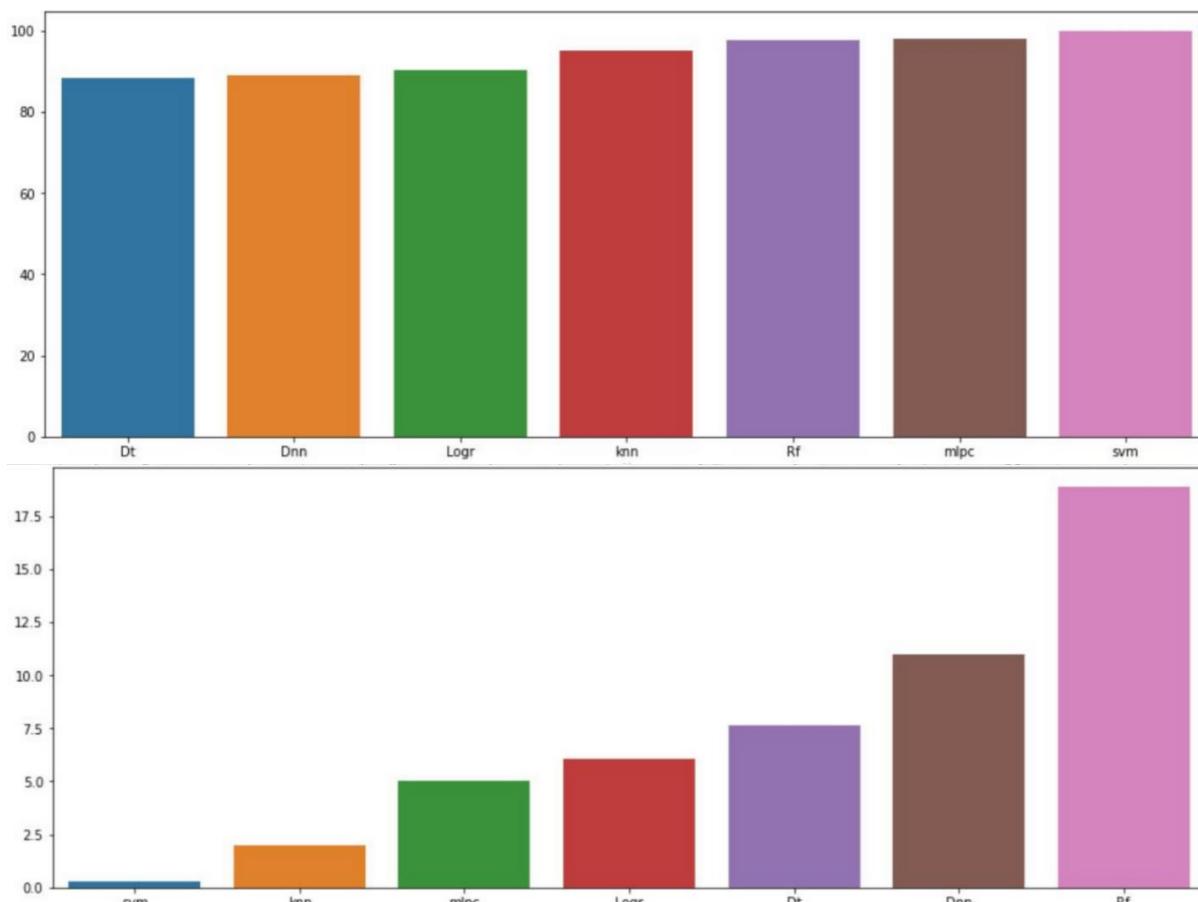
Recall : Undersampled data vs original data





En utilisant l'undersampling, le mpc est l'algorithme qui a donné le meilleur résultat en terme de recall.
Sur les données original, décision tree donne les meilleures performances

Précision : Undersampled data vs original data



Bien que sur les données avec l'undersampling la précision est élevée, cette dernière est estimée à 17.5% sur l'ensemble des données ce qui est médiocre et ne peut aucunement être pris en considération. En terme d'optimisation, on ne peut pas vraiment avancer en vue des données encodées.

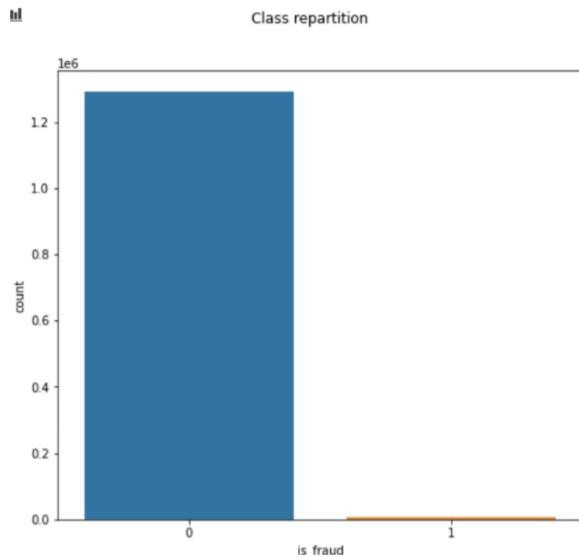
Dataset 2 : Archive credit card fraude detection

Il s'agit d'un ensemble de données de transaction de carte de crédit obtenus grâce à une simulation contenant des transactions légitimes et frauduleuses de la durée du 1er janvier 2019 au 31 décembre 2020. Il couvre les

cartes de crédit de 1000 clients effectuant des transactions avec un lac de 800 commerçants. La simulation est réalisée à l'aide du script Sparkov Data generation | Github Créé par Brandon Harris.

Exploratory data analysis

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	1296675	non-null int64
1	trans_date_trans_time	1296675	non-null object
2	cc_num	1296675	non-null int64
3	merchant	1296675	non-null object
4	category	1296675	non-null object
5	amt	1296675	non-null float64
6	first	1296675	non-null object
7	last	1296675	non-null object
8	gender	1296675	non-null object
9	street	1296675	non-null object
10	city	1296675	non-null object
11	state	1296675	non-null object
12	zip	1296675	non-null int64
13	lat	1296675	non-null float64
14	long	1296675	non-null float64
15	city_pop	1296675	non-null int64
16	job	1296675	non-null object
17	dob	1296675	non-null object
18	trans_num	1296675	non-null object
19	unix_time	1296675	non-null int64
20	merch_lat	1296675	non-null float64
21	merch_long	1296675	non-null float64
22	is_fraud	1296675	non-null int64

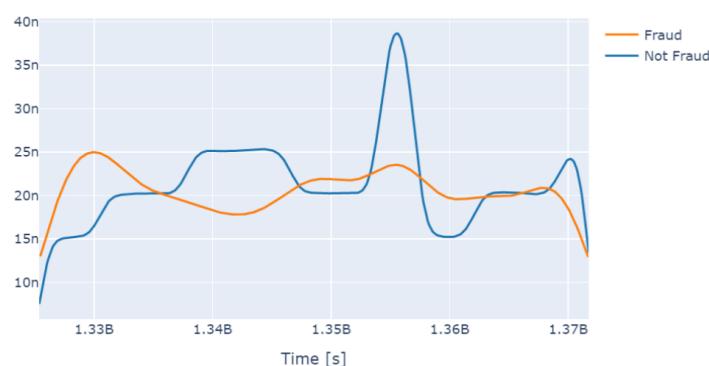


Credit Card Fraud Detection data - rows: 1296675 columns: 23

On remarque que la problématique reste la même, les données sont imbalancées, il faut donc procéder aux techniques d'échantillonnage pour ajuster le poids des classes.

Nombre de transaction / temps

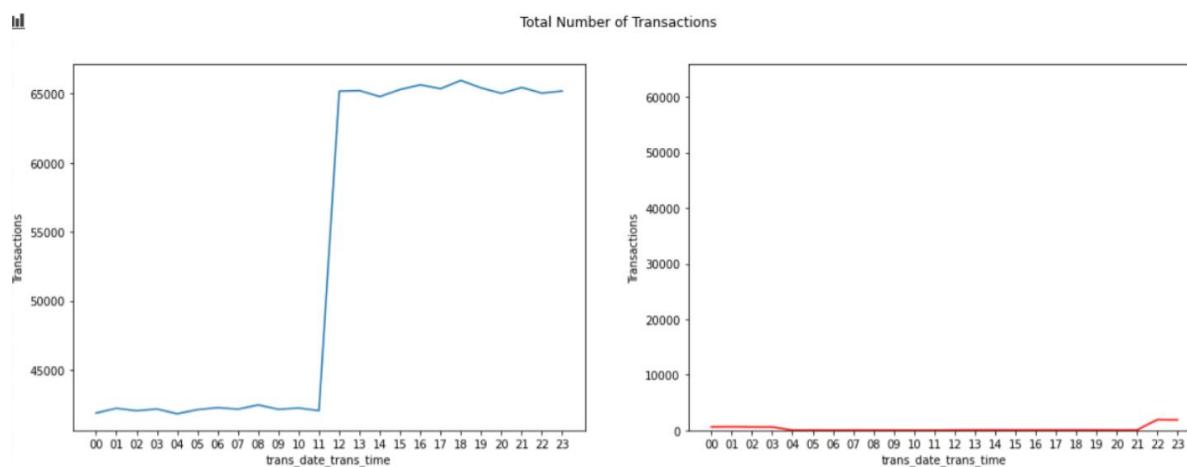
Credit Card Transactions Time Density Plot



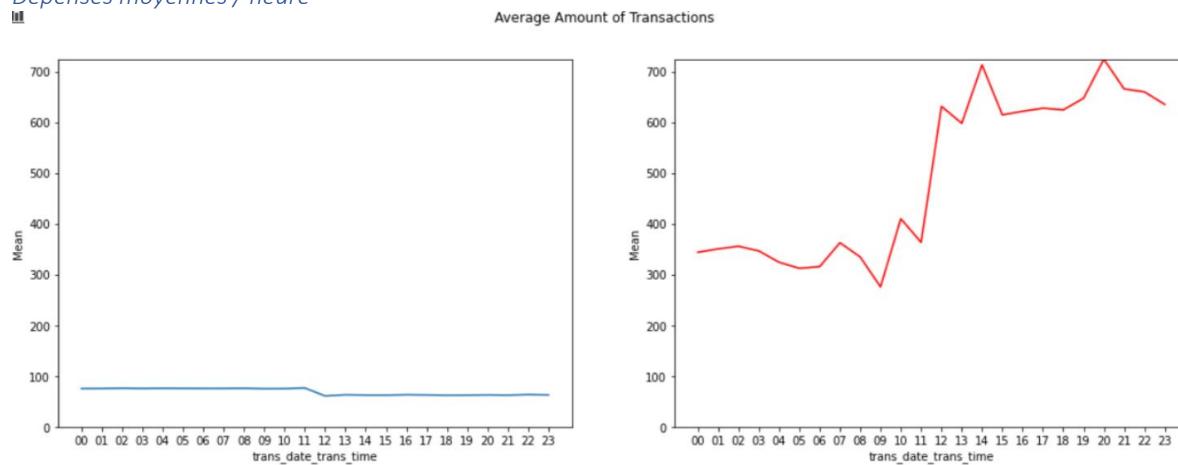
Ici on peut voir que la densité des données frauduleuses n'est pas la même on peut donc conclure qu'il y a quelques paramètres qui permettent cette différence. Notre but c'est de pouvoir les distinguer grâce à l'étude exploratoire.

Pour les prochaines densités, la courbe en rouge est dédiée aux fraude, celle en bleu sera pour les données non frauduleuses.

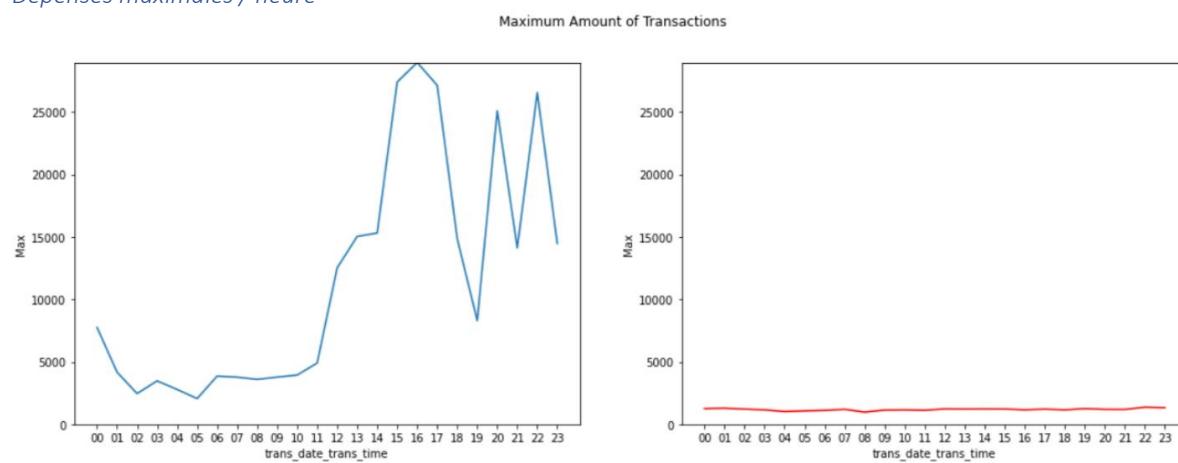
Nombre de transaction / heure



Dépenses moyennes / heure

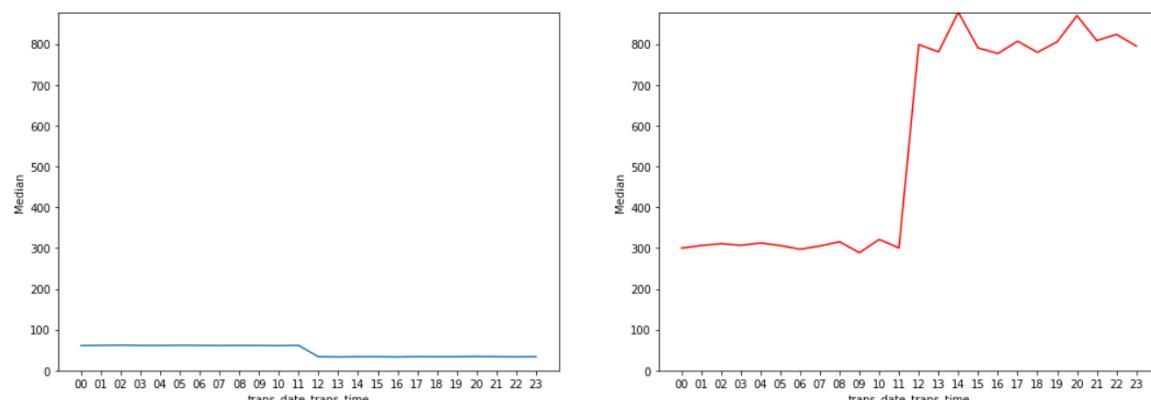


Dépenses maximales / heure



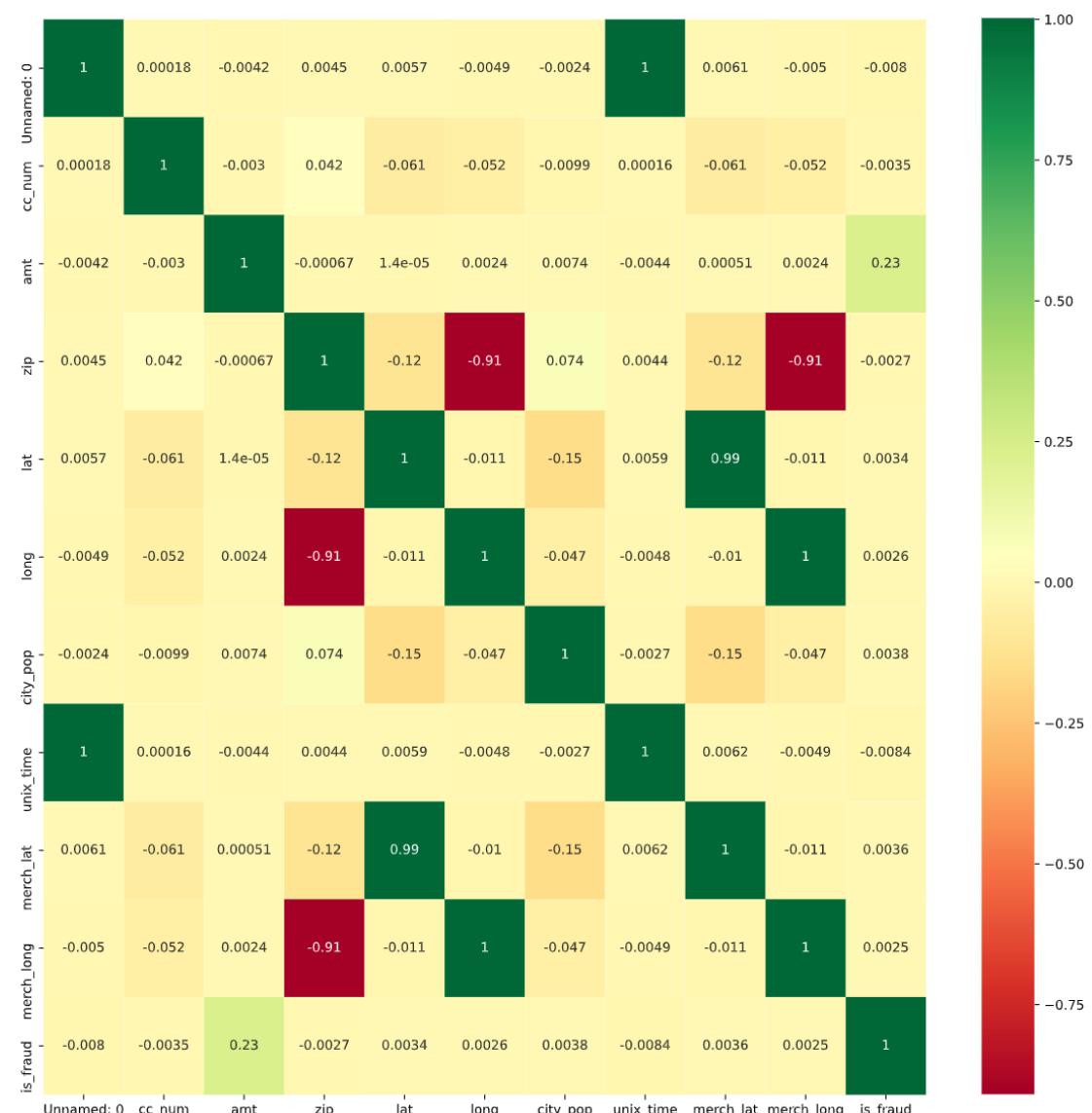
Montant médian

Median Amount of Transactions



On remarque que les fraudes se font principalement la nuit entre minuit et 4 heure du matin avec des dépenses moyennes plus au moins faibles car la banque bloque les achats avec des grands montants.

Matrice de corrélation



Densités

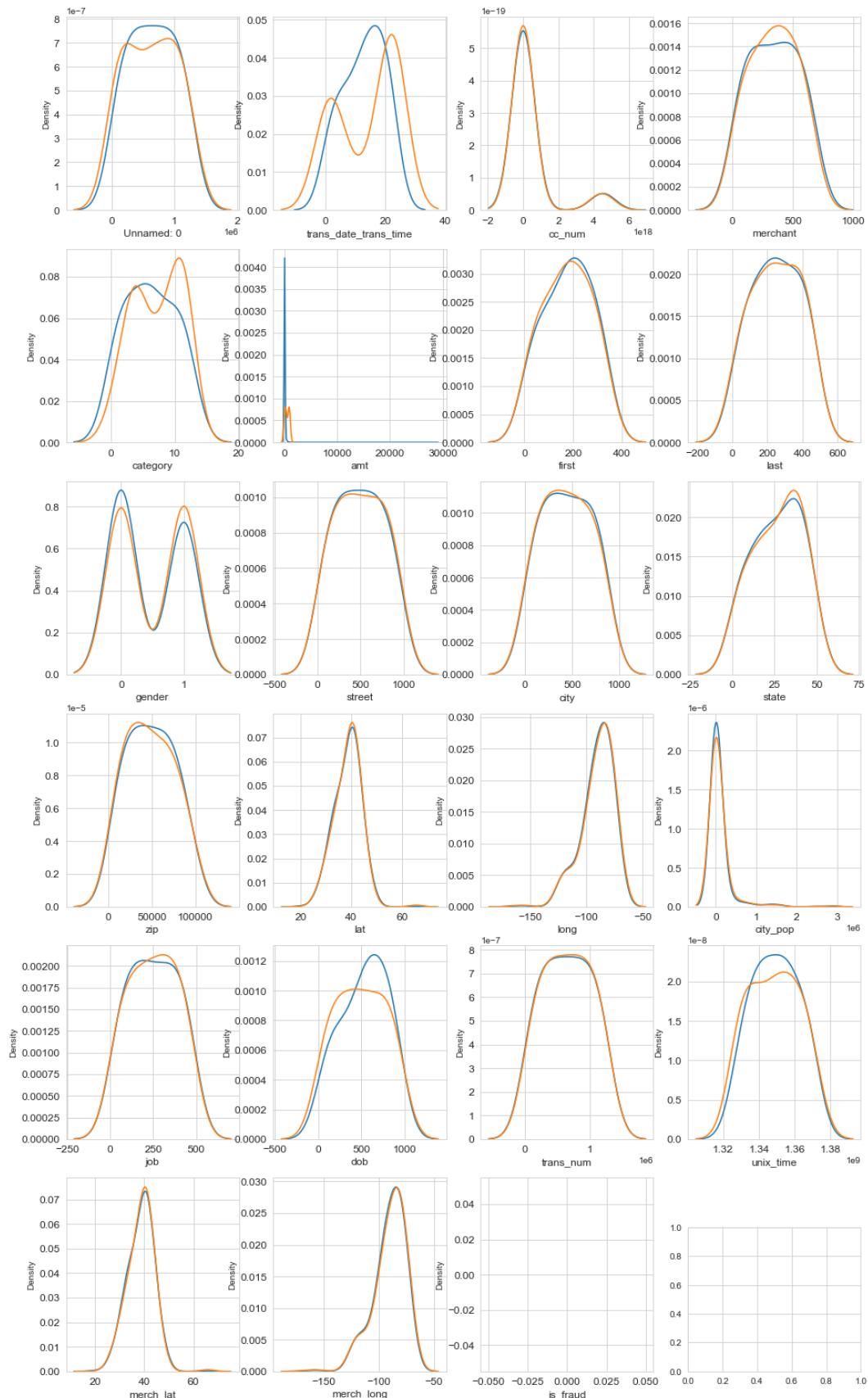


Figure 14 : Les densités entre les variables

L'interprétation des densités se fait comme suit : deux courbes superposées veulent dire qu'ils ont la même distribution, on ne peut donc utiliser ces features pour faire la décision du moment que le fraudeur se comporte de la même manière. Par contre, deux courbes différentes nous font sortir l'information que le fraudeur n'a pas eu le même comportement c'est-à-dire on peut l'utiliser pour faire notre décision.

Pourcentage de fraude / catégorie

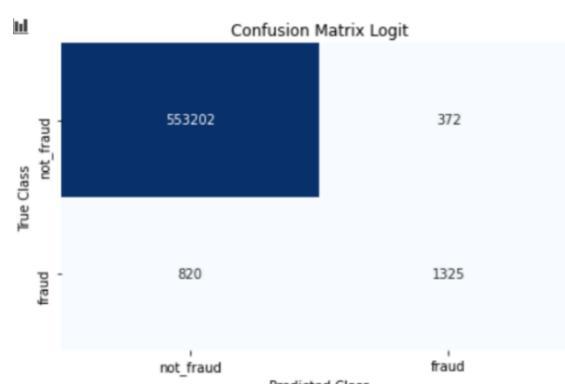
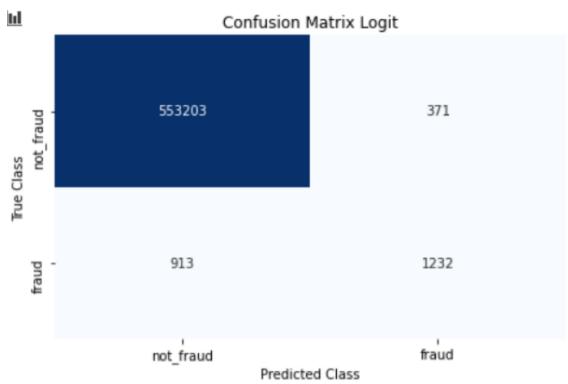
	category	is_fraud
11	shopping_net	1.756149
8	misc_net	1.445795
4	grocery_pos	1.409761
12	shopping_pos	0.722538
2	gas_transport	0.469394
9	misc_pos	0.313853
3	grocery_net	0.294817
13	travel	0.286370
0	entertainment	0.247835
10	personal_care	0.242403
7	kids_pets	0.211439
1	food_dining	0.165098
6	home	0.160825
5	health_fitness	0.154869

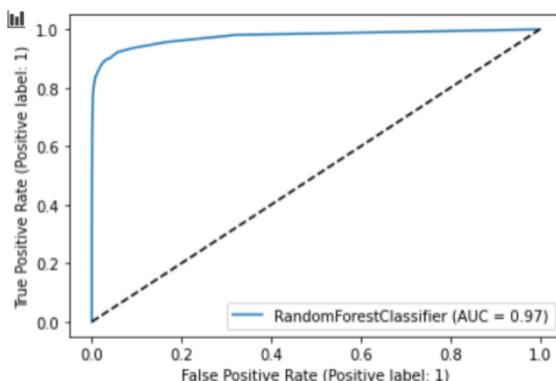
Random forest

Paramètres optimaux

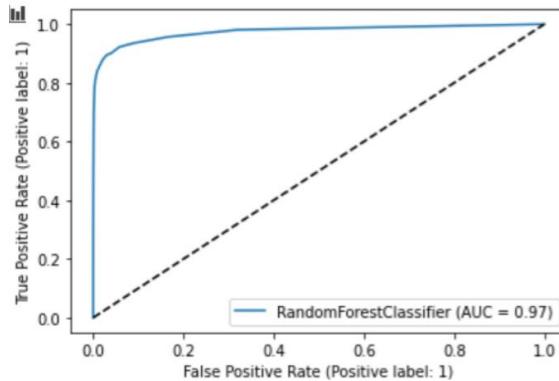
	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.77	0.57	0.66	2145
accuracy			1.00	555719
macro avg	0.88	0.79	0.83	555719
weighted avg	1.00	1.00	1.00	555719

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.78	0.62	0.69	2145
accuracy			1.00	555719
macro avg	0.89	0.81	0.84	555719
weighted avg	1.00	1.00	1.00	555719





Sans feature selection



Avec feature selection

On peut constater qu'avec la sélection des features les performances augmentent. Comment le choix de suppression est fait ? Tout simplement grâce aux densité qu'on avait traité ci haut. Tout les attributs qui n'ont pas d'impact sur la décision ont été supprimés

XGboost

Ici, plusieurs configurations ou hypothèses ont été adoptée. La modèle sera testé avec chacune de ces dernières pour voir lesquelles donnent les meilleures performances

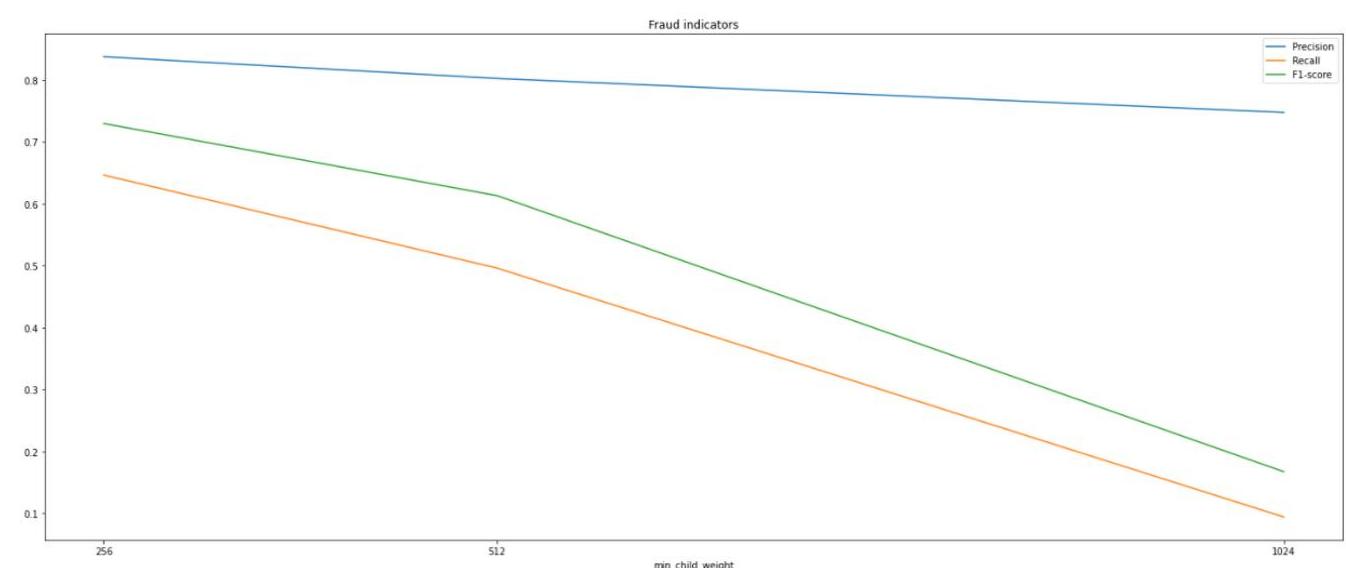
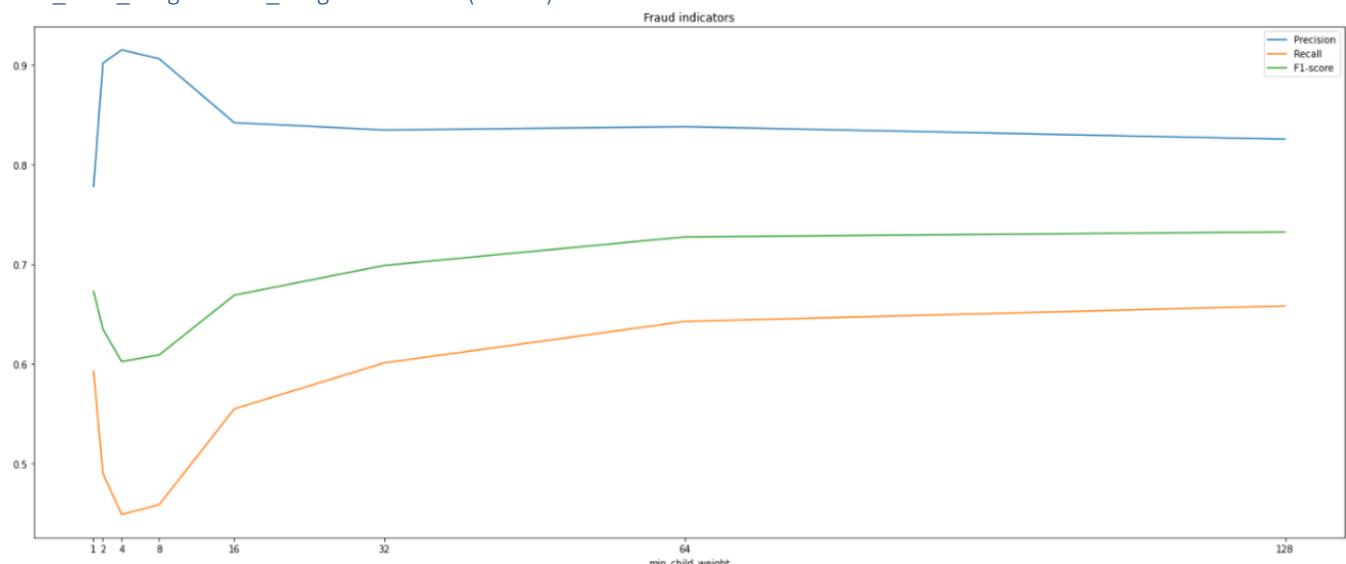
- Config1_1: One Hot Encoding 'category' and 'state'. 'merchant' is Label Encoded (high cardinality).
- Config1_2: LabelEncoding 'category', 'state' and 'merchant'.
- Config2_1: Dividing 'category', 'state' and 'merchant' values to smaller groups based on their fraud_risk. Similar values have the same One Hot Encoding.
- Config2_2: Dividing 'category', 'state' and 'merchant' values to smaller groups based on their fraud_risk. Similar values have the same Label Encoding.
- Config3_1: Dividing 'category' and 'state' values to smaller groups based on similarity (defined arbitrarily). Similar values have the same One Hot Encoding. 'merchant' is label Encoded.
- Config3_2: Dividing 'category' and 'state' values to smaller groups based on similarity (defined arbitrarily). Similar values have the same Label Encoding. 'merchant' is label Encoded.

	config1_1	config1_2	config2_1	config2_2	config3_1	config3_2
val_auc	0.993366	0.992621	0.989873	0.988629	0.975237	0.977445
val_precision_0	1.000000	1.000000	1.000000	1.000000	0.990000	0.990000
val_precision_1	0.780000	0.800000	0.880000	0.800000	0.720000	0.720000
val_recall_0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
val_recall_1	0.590000	0.560000	0.370000	0.460000	0.140000	0.130000
val_f1_0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
val_f1_1	0.670000	0.660000	0.520000	0.590000	0.240000	0.230000

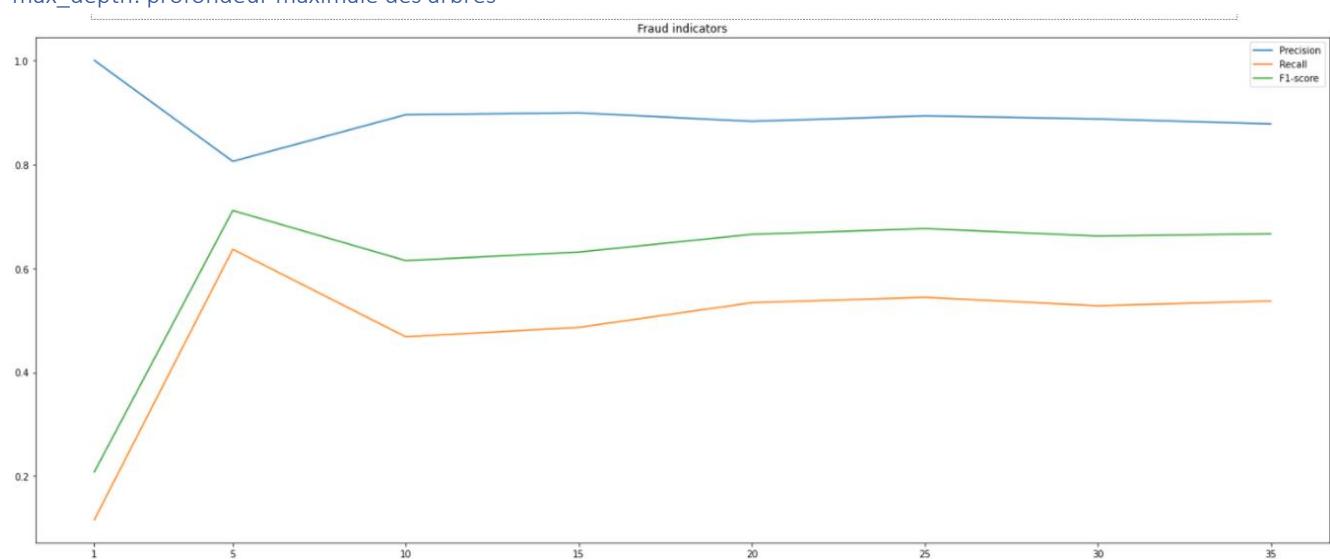
On remarque que la config1_1 donne en général des résultats supérieurs.

Cependant en terme d'optimisation, on ne peut pas utiliser une grid search à cause de la puissance machine limitée, on a donc opté par une adoption d'une étude de l'effet des différents paramètres du modèle

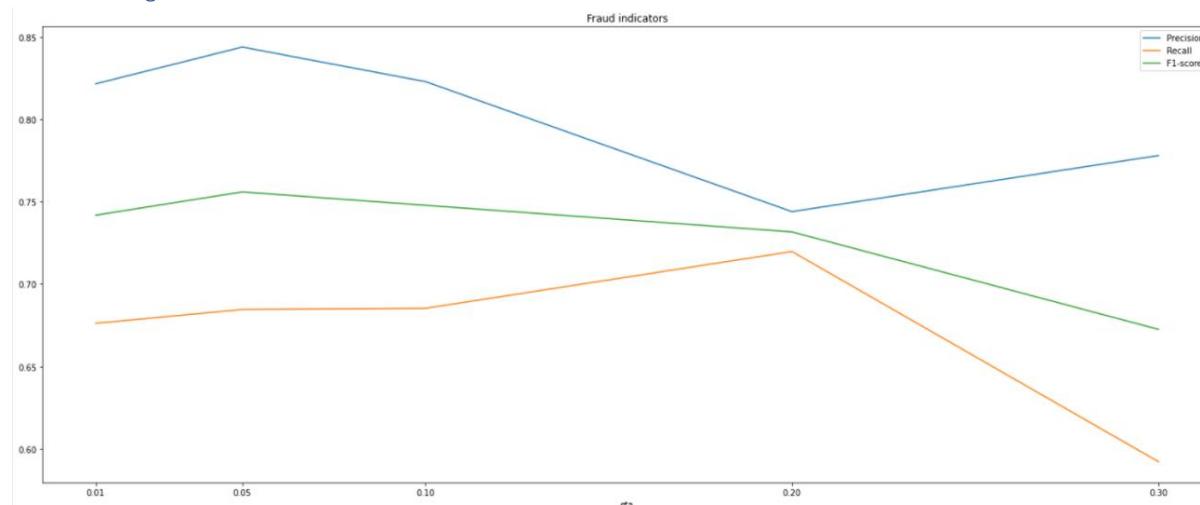
min_child_weight: class_weight de class 1 (fraude)



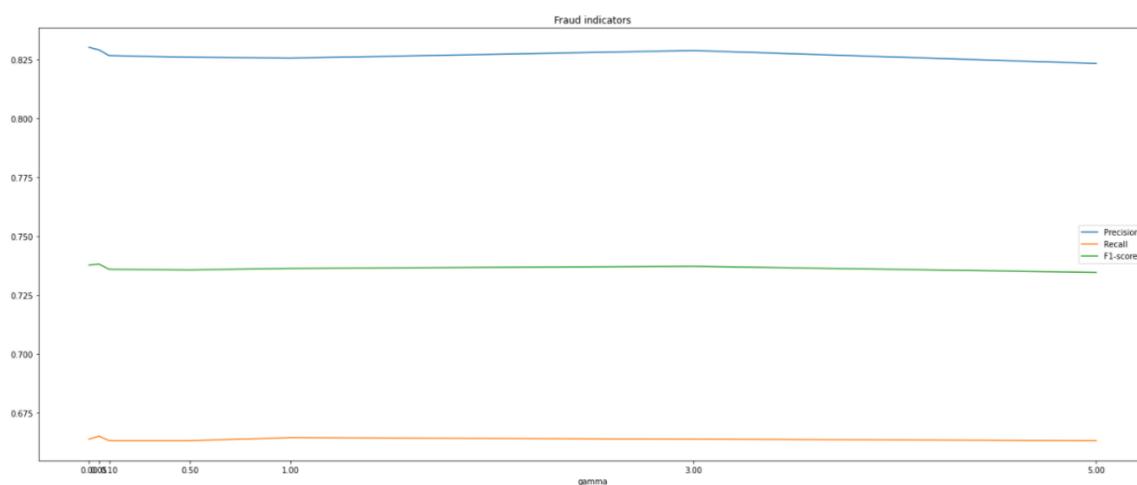
max_depth: profondeur maximale des arbres



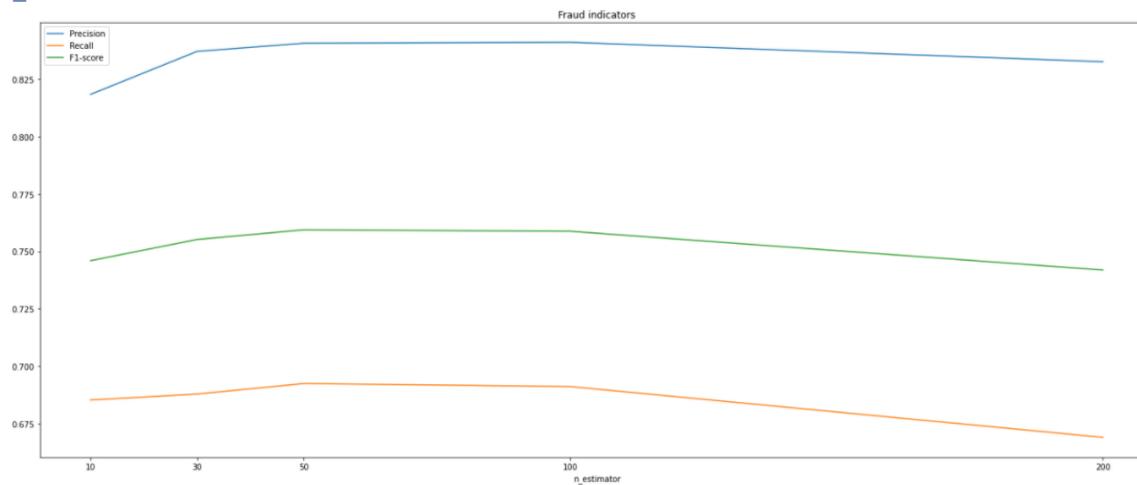
Eta : Learning rate



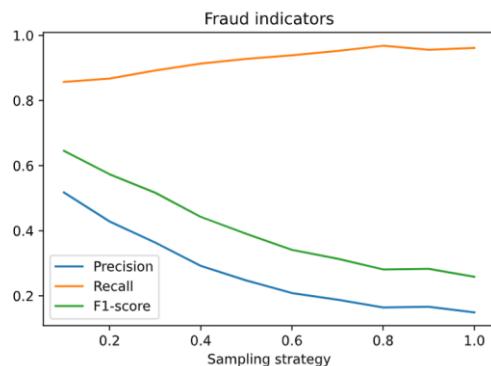
Gamme : Contrôle de profondeur



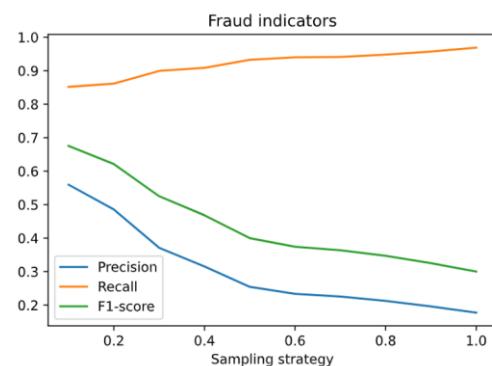
N_estimator : Nombre d'itérations



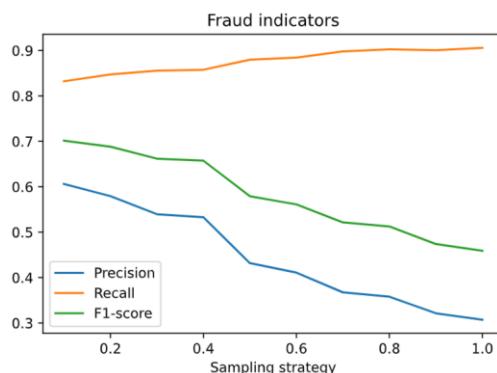
Undersampling avec les paramètres optimaux



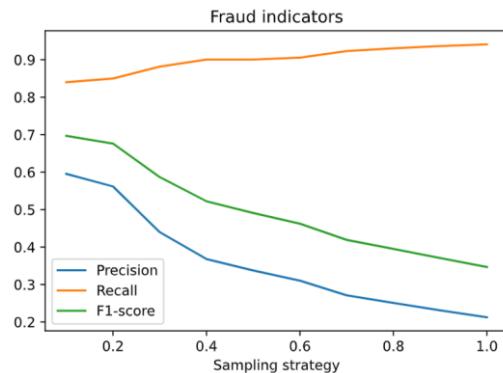
Oversampling avec les paramètres optimaux



SMOTE avec les paramètres optimaux



ADASYN avec les paramètres optimaux



Conclusion générale sur les paramètres

On a choisis les paramètres qui maximisent à la fois la précision, le recall et le f1-score :

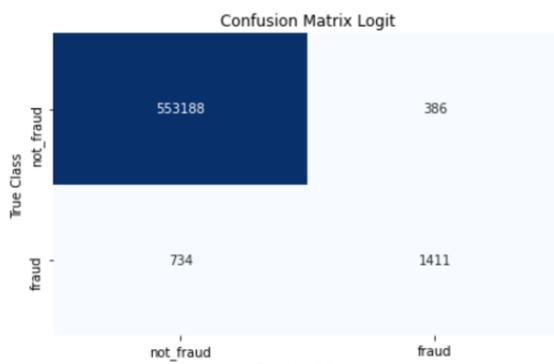
- Min_child : 2
- Max_depth : 5
- Eta : 0.04

Lorsque le coefficient de sampling augmente :

- Recall
- Précision diminue
- F1 score

XGboost meilleur performance

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.79	0.66	0.72	2145
accuracy			1.00	555719
macro avg	0.89	0.83	0.86	555719
weighted avg	1.00	1.00	1.00	555719



Neural network

Hyper parameters tuning

Le but est de trouver les meilleurs batch_size et learning rate

Average results for val_f1			
0.0001	0.0010	0.0100	
32	0.397891	0.556302	0.124167
128	0.251931	0.566706	0.502611
512	0.357037	0.579305	0.573378
1024	0.274436	0.576542	0.620776

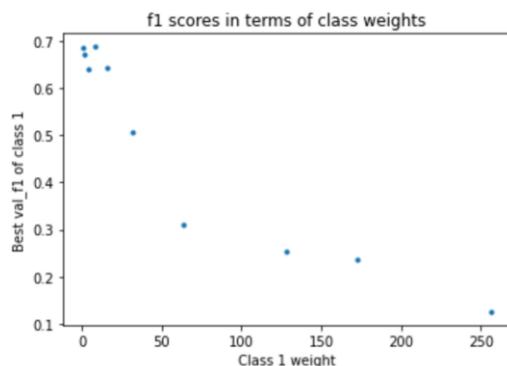
On a testé sur toutes les configurations

	config1_1	config1_2	config2_1	config2_2	config3_1	config3_2
loss	0.009847	0.013257	0.010842	0.011741	0.016096	0.015361
recall	0.683646	0.464812	0.627346	0.668063	0.459283	0.444370
precision	0.800000	0.650410	0.753775	0.681073	0.632004	0.675153
accuracy	0.997197	0.995484	0.996677	0.996291	0.995351	0.995573
auc	0.961765	0.939305	0.962513	0.959774	0.934670	0.933298
val_loss	0.020647	0.014968	0.020801	0.013752	0.017324	0.015914
val_recall	0.743823	0.518856	0.741873	0.579974	0.320546	0.524707
val_precision	0.669397	0.698774	0.611796	0.751474	0.733631	0.655032
val_accuracy	0.996302	0.995820	0.995677	0.996372	0.995280	0.995542
val_auc	0.982278	0.980221	0.983715	0.981171	0.962245	0.967386
f1	0.737261	0.542167	0.684774	0.674505	0.531975	0.535974
val_f1	0.704650	0.595522	0.670585	0.654679	0.446154	0.582671

Les effets de la classe weight

	1	2	4	8	16	32	64	128	173	256
loss	0.008793	0.016395	0.043329	0.038025	0.061033	0.095000	0.132109	0.185322	0.213966	0.278188
recall	0.684484	0.726542	0.694370	0.816689	0.839812	0.839477	0.912869	0.914209	0.959953	0.971682
precision	0.812935	0.703325	0.639210	0.588861	0.491373	0.449327	0.201517	0.184680	0.073845	0.066725
accuracy	0.997279	0.996664	0.995987	0.995665	0.994077	0.993158	0.978689	0.976286	0.930504	0.921647
auc	0.977067	0.973578	0.877195	0.990234	0.991052	0.990452	0.991833	0.992072	0.991844	0.990353
val_loss	0.010607	0.027113	0.018166	0.016377	0.026225	0.060725	0.067673	0.083219	0.110468	0.161661
val_recall	0.611834	0.786086	0.545514	0.776333	0.797139	0.847854	0.897269	0.918726	0.912874	0.962289
val_precision	0.778329	0.585756	0.779016	0.618012	0.540564	0.360819	0.187857	0.147664	0.134948	0.067126
val_accuracy	0.996665	0.995434	0.996387	0.995828	0.994779	0.990190	0.976386	0.968068	0.964779	0.920466
val_auc	0.983311	0.983828	0.880648	0.988265	0.990752	0.987942	0.991145	0.991237	0.990201	0.989374
f1	0.743200	0.714745	0.665649	0.684310	0.619990	0.585349	0.330152	0.307285	0.137141	0.124875
val_f1	0.685111	0.671294	0.641683	0.688184	0.644246	0.506211	0.310671	0.254434	0.235136	0.125498

F1 score en fonction de la classe weight



Meilleure performance sur X_test

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.75	0.60	0.66	2145
accuracy				1.00
macro avg				0.83
weighted avg				1.00

LSTM

On a utilisé les attributs suivants :

- Amt : montant de la transaction
- Time : Temps de la transaction
- Lat : Latitude
- Long : Longitude

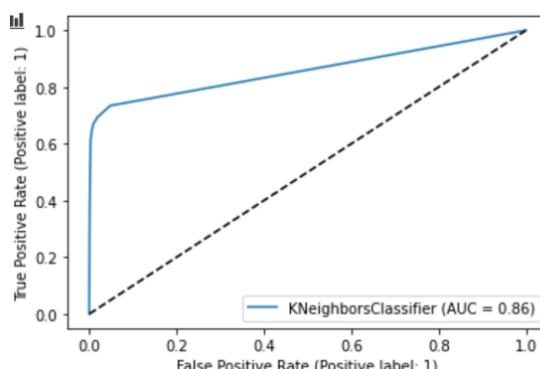
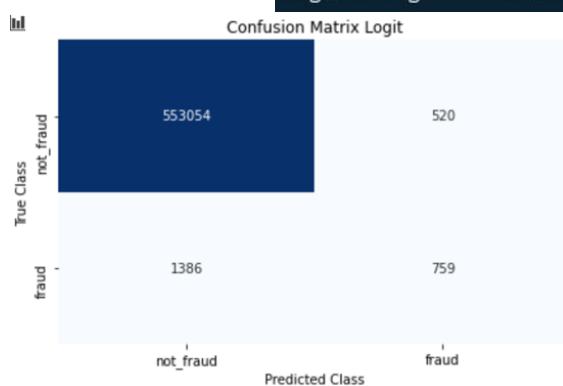
Meilleure performance sur le X_test

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	603555
1.0	0.84	0.68	0.75	2145
accuracy				1.00
macro avg				0.87
weighted avg				1.00

KNN

Meilleur modèle sur le X_test

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.59	0.35	0.44	2145
accuracy				1.00
macro avg				0.72
weighted avg				1.00



Effet de sampling : coef 0.1

	precision	recall	f1-score	support
0	1.00	0.99	1.00	553574
1	0.27	0.69	0.39	2145
accuracy			0.99	555719
macro avg	0.64	0.84	0.69	555719
weighted avg	1.00	0.99	0.99	555719

	precision	recall	f1-score	support
0	1.00	0.97	0.98	553574
1	0.08	0.70	0.15	2145
accuracy			0.97	555719
macro avg	0.54	0.83	0.57	555719
weighted avg	1.00	0.97	0.98	555719

UnderSampling

OverSampling

	precision	recall	f1-score	support
0	1.00	0.98	0.99	553574
1	0.14	0.68	0.23	2145
accuracy			0.98	555719
macro avg	0.57	0.83	0.61	555719
weighted avg	1.00	0.98	0.99	555719

	precision	recall	f1-score	support
0	1.00	0.98	0.99	553574
1	0.11	0.68	0.19	2145
accuracy			0.98	555719
macro avg	0.55	0.83	0.59	555719
weighted avg	1.00	0.98	0.99	555719

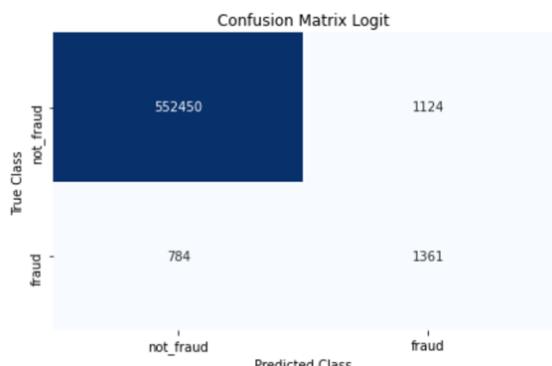
SMOTE

ADASYN

On remarque que l'undersampling donne les meilleures performances

Décision tree

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.55	0.63	0.59	2145
accuracy			1.00	555719
macro avg	0.77	0.82	0.79	555719
weighted avg	1.00	1.00	1.00	555719



Effet du sampling

	precision	recall	f1-score	support
0	1.00	0.99	1.00	553574
1	0.29	0.83	0.43	2145
accuracy			0.99	555719
macro avg	0.65	0.91	0.72	555719
weighted avg	1.00	0.99	0.99	555719

	precision	recall	f1-score	support
0	1.00	0.99	0.99	553574
1	0.26	0.85	0.39	2145
accuracy			0.99	555719
macro avg	0.63	0.92	0.69	555719
weighted avg	1.00	0.99	0.99	555719

UnderSampling

OverSampling

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.48	0.82	0.61	2145
accuracy			1.00	555719
macro avg	0.74	0.91	0.80	555719
weighted avg	1.00	1.00	1.00	555719

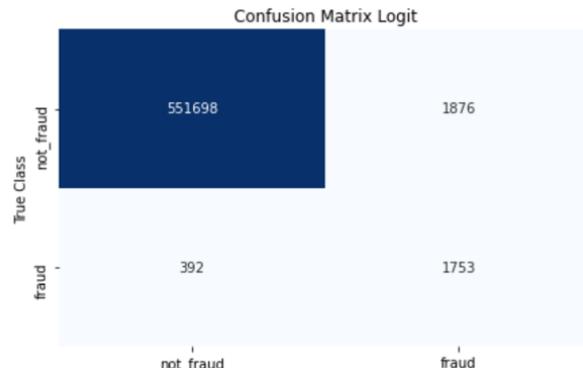
	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.48	0.80	0.60	2145
accuracy			1.00	555719
macro avg	0.74	0.90	0.80	555719
weighted avg	1.00	1.00	1.00	555719

SMOTE

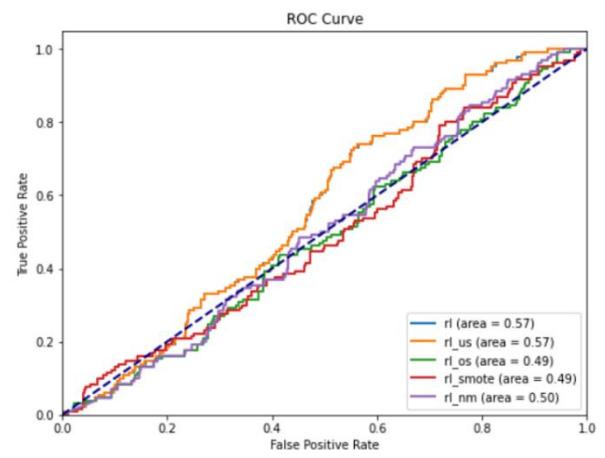
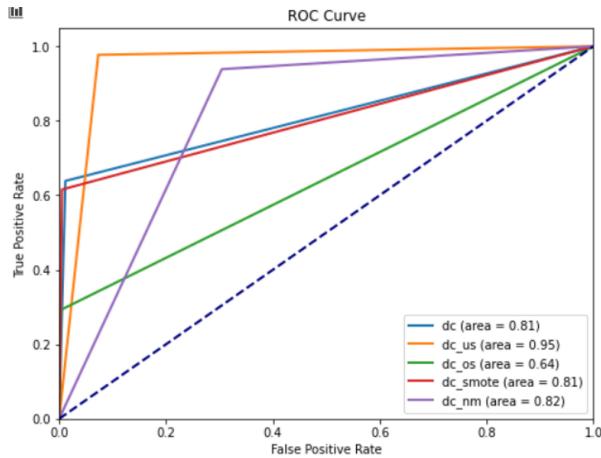
ADASYN

Meilleur modèle sur le X_test

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.48	0.82	0.61	2145
accuracy			1.00	555719
macro avg	0.74	0.91	0.80	555719
weighted avg	1.00	1.00	1.00	555719

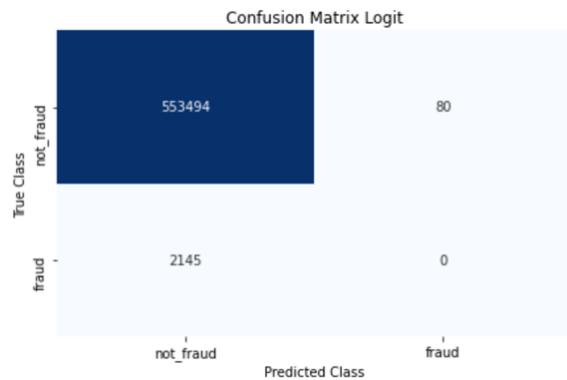


Ici le SMOTE est le mieux adapté



Régression logistique

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.00	0.00	0.00	2145
accuracy			1.00	555719
macro avg	0.50	0.50	0.50	555719
weighted avg	0.99	1.00	0.99	555719



Génération de nouvelles features

Suite à l'exploratory data analysis, on a conclus comme quoi le temps pesais son opoid quant à la décision de la classification. Pour améliorer les performances on a décidé de générer plus de features en fonction du temps notamment la date, heure, jour de la semaine.

```
Converting data type of trans_date_trans_time to datetime

[4] ▶ df['trans_date_trans_time'] = pd.to_datetime(df['trans_date_trans_time'])

[5] ▶ df['trans_hour'] = df['trans_date_trans_time'].dt.hour

[6] ▶ df['day_of_week'] = df['trans_date_trans_time'].dt.day_name()

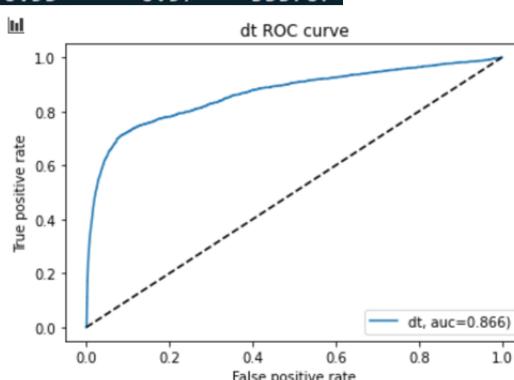
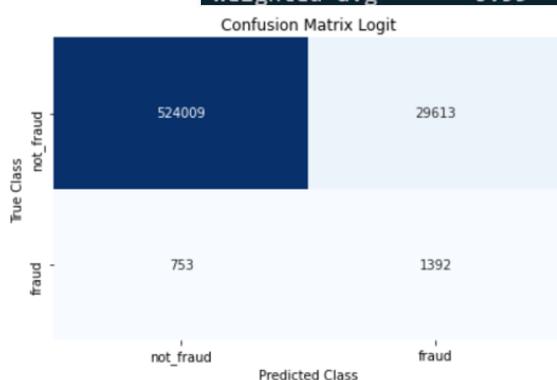
[7] ▶ df['year_month'] = df['trans_date_trans_time'].dt.to_period('M')

▶ df_hist_trans_60d = \
    df1 \
    .groupby(['cc_num'])['amt']\
    .rolling('60D')\
    .count()\ 
    .shift()\ 
    .reset_index()\ 
    .fillna(0)

df_hist_trans_avg_60d = \
    df1 \
    .groupby(['cc_num'])['amt']\
    .rolling('60D')\
    .mean()\ 
    .shift(1)\ 
    .reset_index()\ 
    .fillna(0)
```

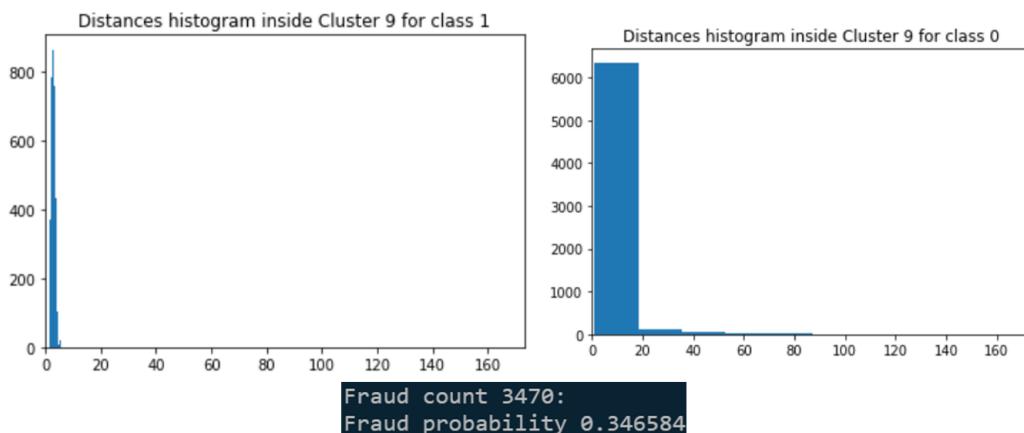
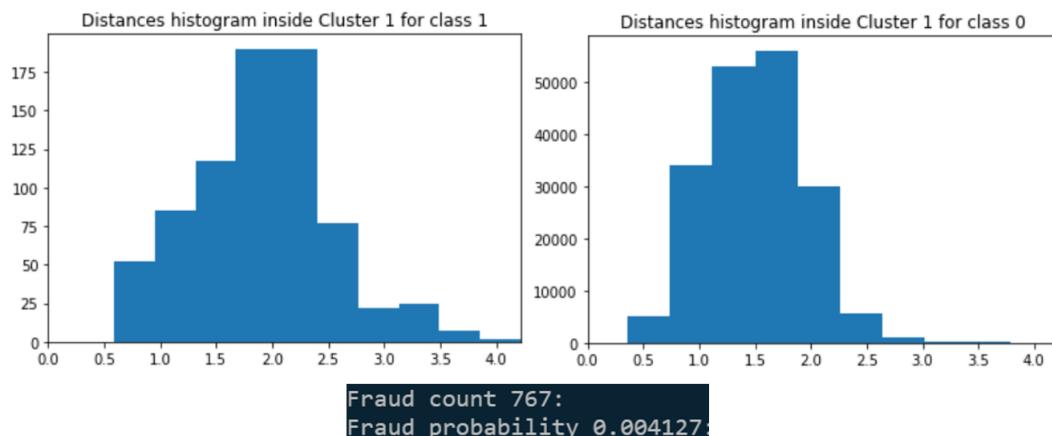
Meilleur modèle sur le X_test

	precision	recall	f1-score	support
0	1.00	0.95	0.97	553622
1	0.04	0.65	0.08	2145
accuracy			0.95	555767
macro avg	0.52	0.80	0.53	555767
weighted avg	0.99	0.95	0.97	555767

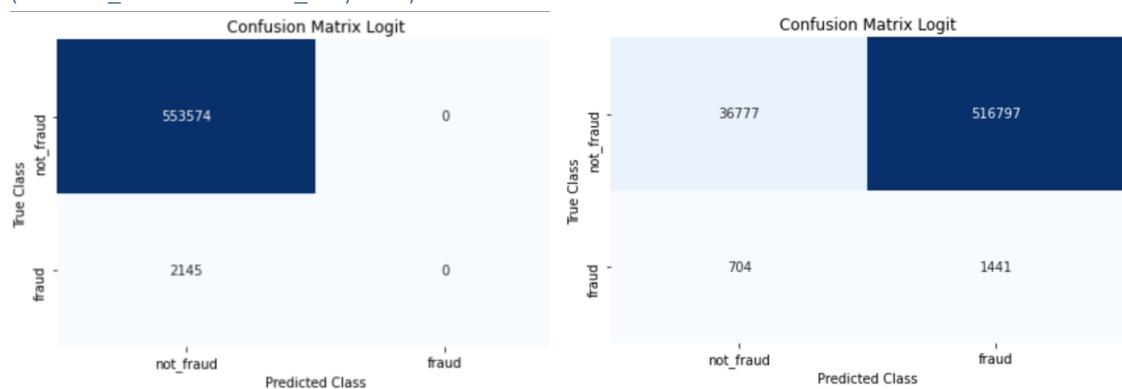


K-means

K=9



H1- Clusters de fraudes/non_fraudes: k-means+knn H2- Fraudes=anomalies
(distance_fraude>distance_moyenne)



Conclusion

On n'arrive pas à faire la classification avec k-means mais on parvient à définir des clusters à haut, moyen, bas risque de fraude.

Kohonen

precision recall f1-score support					precision recall f1-score support				
0 1.00 1.00 1.00 284315					0 1.00 1.00 1.00 284315				
1 0.79 0.34 0.47 492					1 0.23 0.65 0.34 492				
accuracy macro avg 0.89 0.67 0.74 284807					accuracy macro avg 0.61 0.82 0.67 284807				
weighted avg 1.00 1.00 1.00 284807					weighted avg 1.00 1.00 1.00 284807				
precision recall f1-score support					precision recall f1-score support				
0 1.00 0.95 0.97 284315					0 1.00 1.00 1.00 284315				
1 0.02 0.59 0.04 492					1 0.83 0.57 0.68 492				
accuracy macro avg 0.51 0.77 0.50 284807					accuracy macro avg 0.92 0.78 0.84 284807				
weighted avg 1.00 0.95 0.97 284807					weighted avg 1.00 1.00 1.00 284807				
precision recall f1-score support					precision recall f1-score support				
0 1.00 1.00 1.00 284315					0 1.00 0.99 1.00 284315				
1 0.83 0.48 0.60 492					1 0.16 0.66 0.25 492				
accuracy macro avg 0.91 0.74 0.80 284807					accuracy macro avg 0.58 0.83 0.62 284807				
weighted avg 1.00 1.00 1.00 284807					weighted avg 1.00 0.99 1.00 284807				

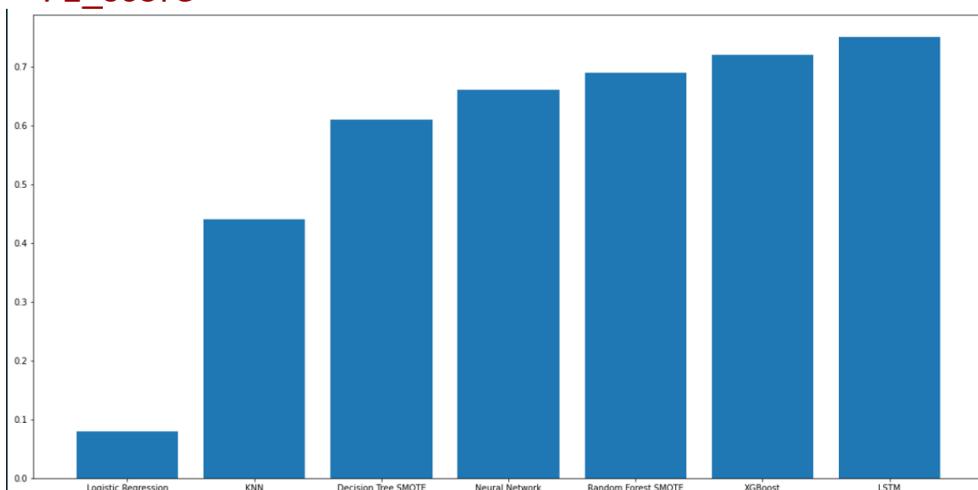
On a essayé d'implémenter l'algorithme kohonen, mais le résultat ne convergeait pas. Pour chaque exécution du modèle un nouveau résultat est obtenu.

Récapitulatif

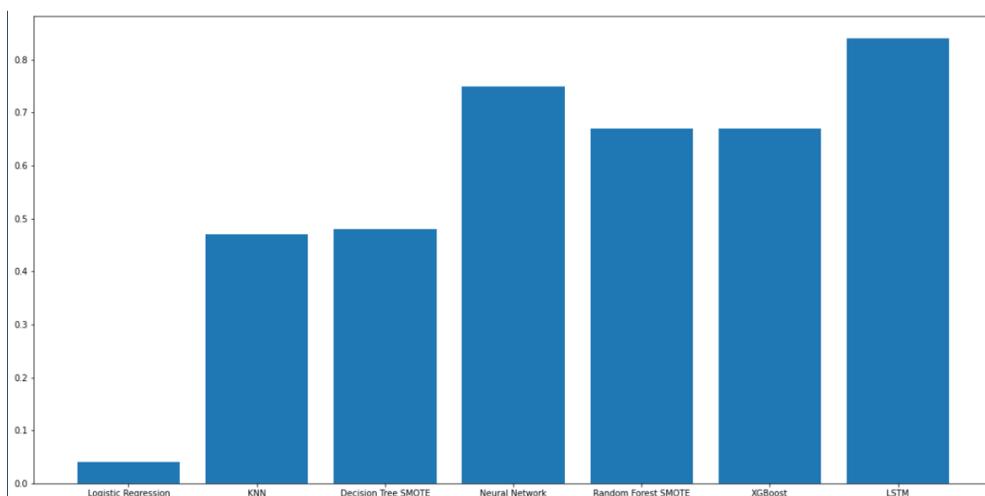
No sampling	Over	Under	SMOTE	ADASYN	NearMiss	f1_score_1	recall_1	precision_1	AUC
LSTM	*					0.75	0.68	0.84	-1.0
XGBoost	*					0.72	0.66	0.78	0.9812
Random Forest			*			0.69	0.71	0.67	0.9883
Random Forest	*					0.69	0.62	0.78	0.9828
Random Forest				*		0.68	0.69	0.67	0.9889
Random Forest		*				0.68	0.64	0.73	0.9852
Neural Network	*					0.66	0.6	0.75	0.9866
XGBoost			*			0.64	0.81	0.53	0.9922
XGBoost				*		0.64	0.82	0.52	0.9918
Decision Tree				*		0.61	0.82	0.48	0.9815
XGBoost	*					0.6	0.83	0.47	0.9937

	No sampling	Over	Under	SMOTE	ADASYN	NearMiss	f1_score_1	recall_1	precision_1	AUC
Decision Tree				*			0.6	0.8	0.48	0.9763
Decision Tree	*						0.59	0.63	0.55	0.973
XGBoost			*				0.56	0.83	0.43	0.994
Random Forest			*				0.56	0.82	0.42	0.9871
Neural Network			*				0.51	0.81	0.37	0.9843
Neural Network				*			0.48	0.76	0.35	0.9706
Neural Network				*			0.46	0.79	0.32	0.9788
KNN	*						0.44	0.41	0.47	0.81
Neural Network		*					0.43	0.81	0.29	0.9784
Decision Tree			*				0.43	0.83	0.29	0.957
Decision Tree		*					0.39	0.85	0.26	0.9642
	No sampling	Over	Under	SMOTE	ADASYN	NearMiss	f1_score_1	recall_1	precision_1	AUC
KNN			*				0.39	0.69	0.27	0.8891
KNN				*			0.23	0.68	0.14	0.8613
XGBoost					*		0.19	0.83	0.11	0.9817
KNN					*		0.19	0.68	0.11	0.8611
KNN		*					0.15	0.7	0.08	0.8562
Decision Tree					*		0.09	0.88	0.05	0.9482
Random Forest						*	0.08	0.83	0.04	0.956
Logistic Regression	*						0.08	0.65	0.04	0.8657
Neural Network						*	0.07	0.84	0.04	0.943
KNN						*	0.06	0.73	0.03	0.8367
	No sampling	Over	Under	SMOTE	ADASYN	NearMiss	f1_score_1	recall_1	precision_1	AUC
LSTM	*						0.75	0.68	0.84	-1.0
XGBoost	*						0.72	0.66	0.67	0.9812
Random Forest				*			0.69	0.71	0.67	0.9883
Neural Network	*						0.66	0.6	0.75	0.9866
Decision Tree				*			0.61	0.82	0.48	0.9482
KNN	*						0.44	0.41	0.47	0.81
Logistic Regression	*						0.08	0.65	0.04	0.8657

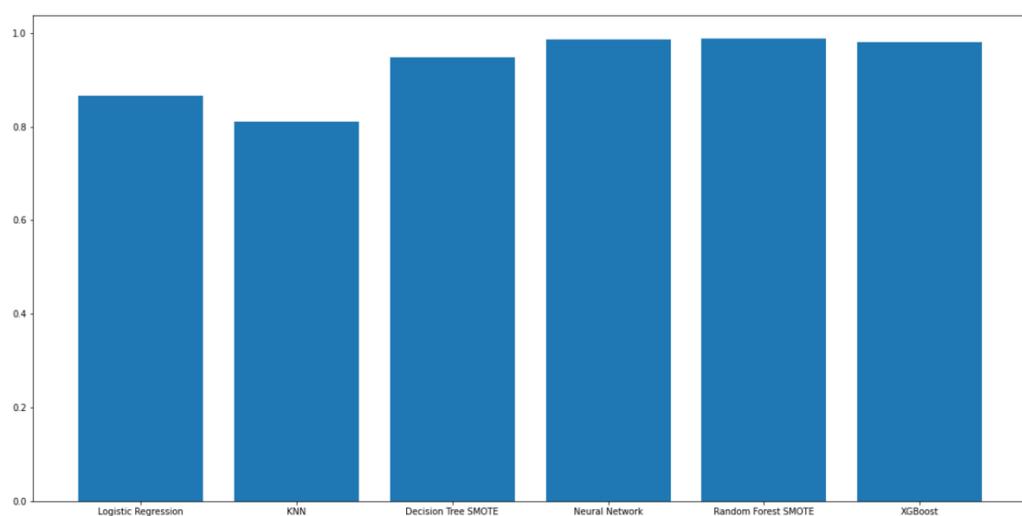
F1_score



Precision



AUC



Optimisation

Ajouter de nouvelles features

```
def time_for_last_purchase(df, criteria=None, time_col='unix_time', id_col='cc_num', inplace=False)

def delta_amt_purchase(df, criteria=None, amt_col='amt', id_col='cc_num', inplace=False)

def delta_dst_purchase(df, criteria=None, dst_cols=['merch_lat', 'merch_long'], id_col='cc_num', inplace=False)

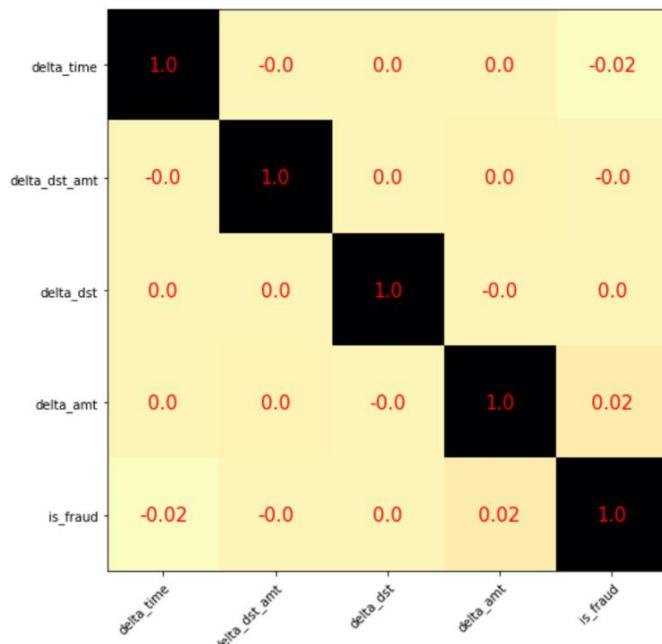
def delta_dst_amt(df, criteria=None, amt_col='amt', dst_cols=['merch_lat', 'merch_long'], id_col='cc_num', inplace=False)

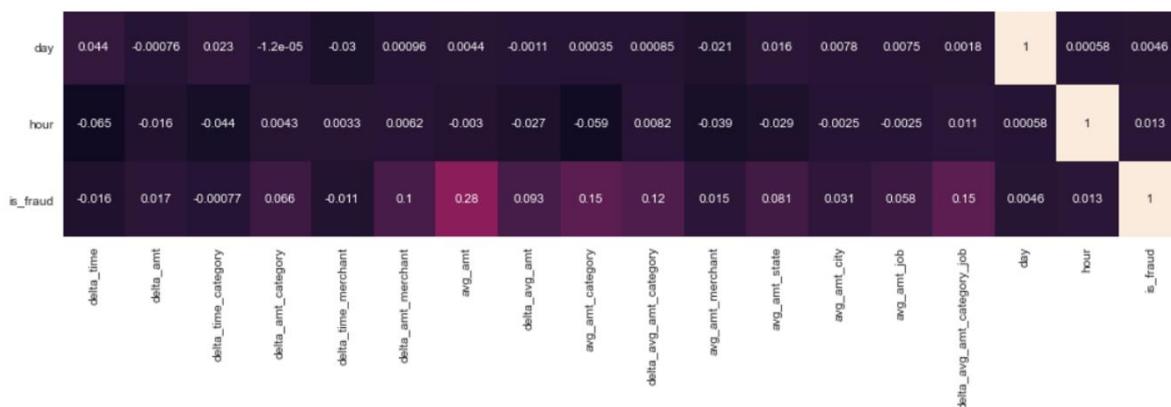
def avg_amt_purchase(df, window=60, criteria=None, amt_col='amt', id_col='cc_num', inplace=False)

def delta_avg_amt_purchase(df, window=60, criteria=None, amt_col='amt', id_col='cc_num', inplace=False)
```

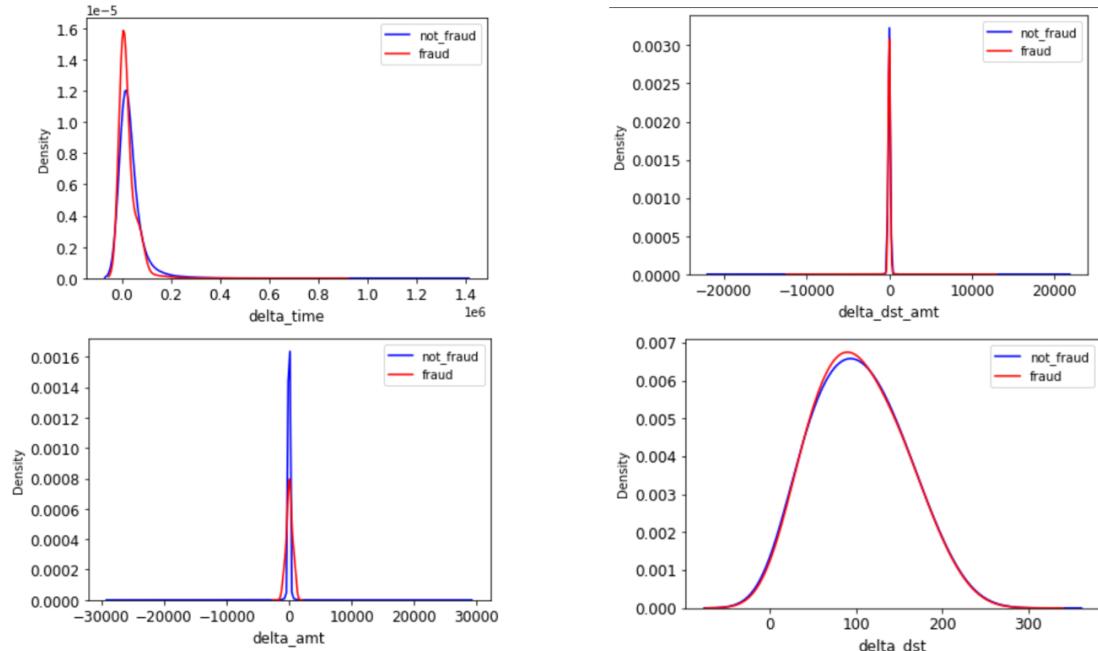
- Calcule la différence de temps/montant/distanc avec le dernier achat d'un utilisateur
- Criteria: un attribut (ou None), préciser une condition sur l'achat
- Criteria=category: on cherche le dernier achat pour cet utilisateur dans la même catégorie)
- Avg_amt_purchase: Le montant moyen des dépenses pour un utilisateur
- Delta_avg_amt_purchase: différence du montant dépensé avec la moyenne (amt - Avg_amt_purchase)
- Criteria: merchant ou category, pour spécifier les achats à considérer dans l'historique
- Window: nombre d'achats à considérer dans l'historique

Matrice de corrélation des nouvelles features

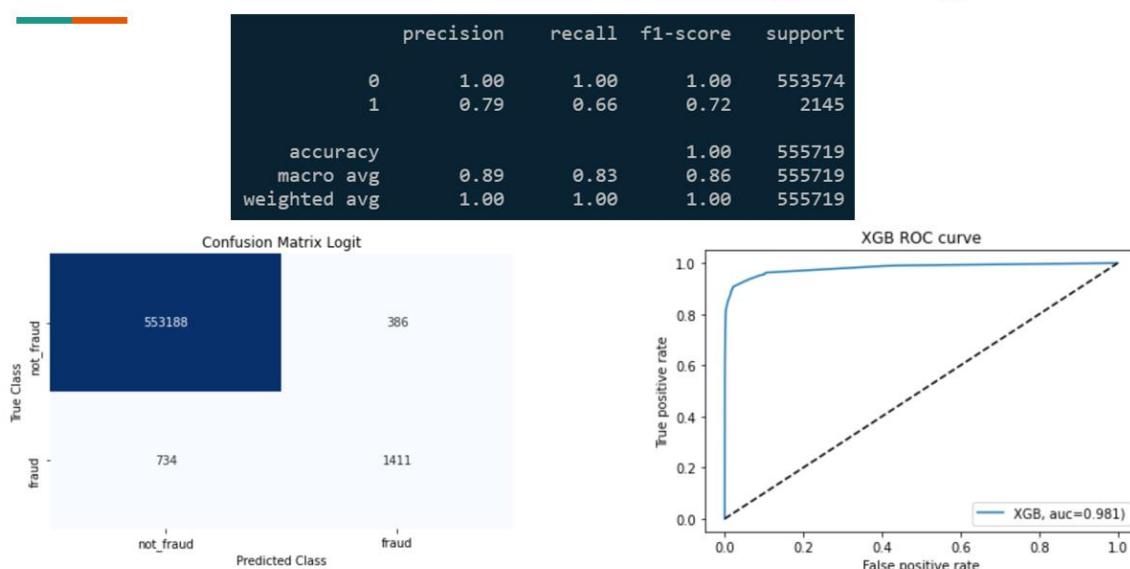




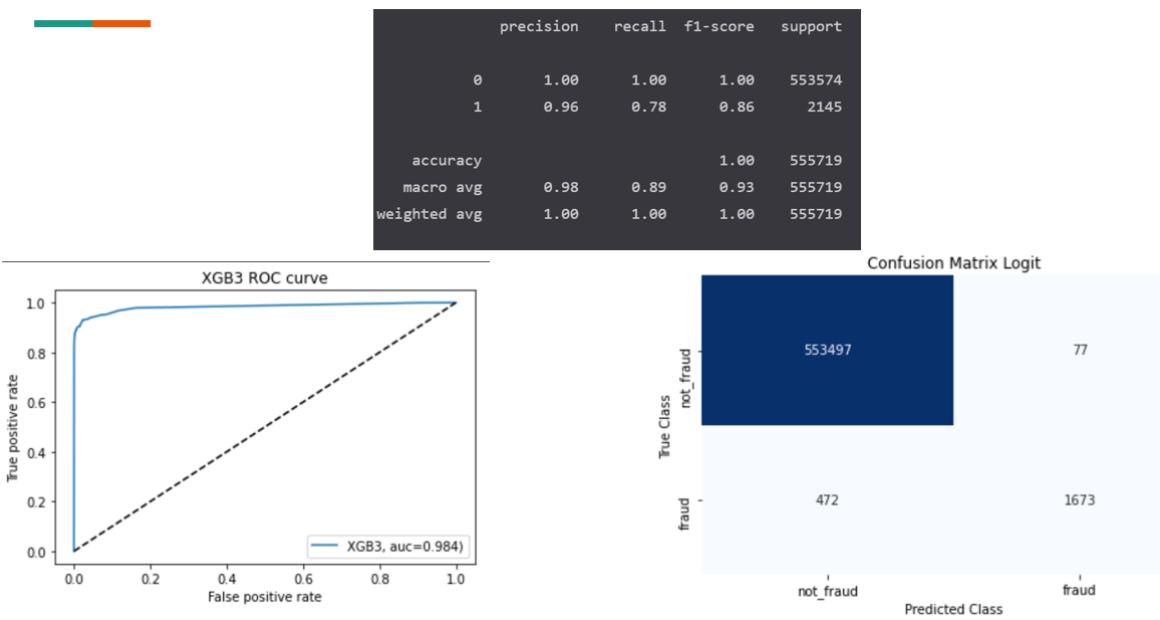
Densités



XGboost

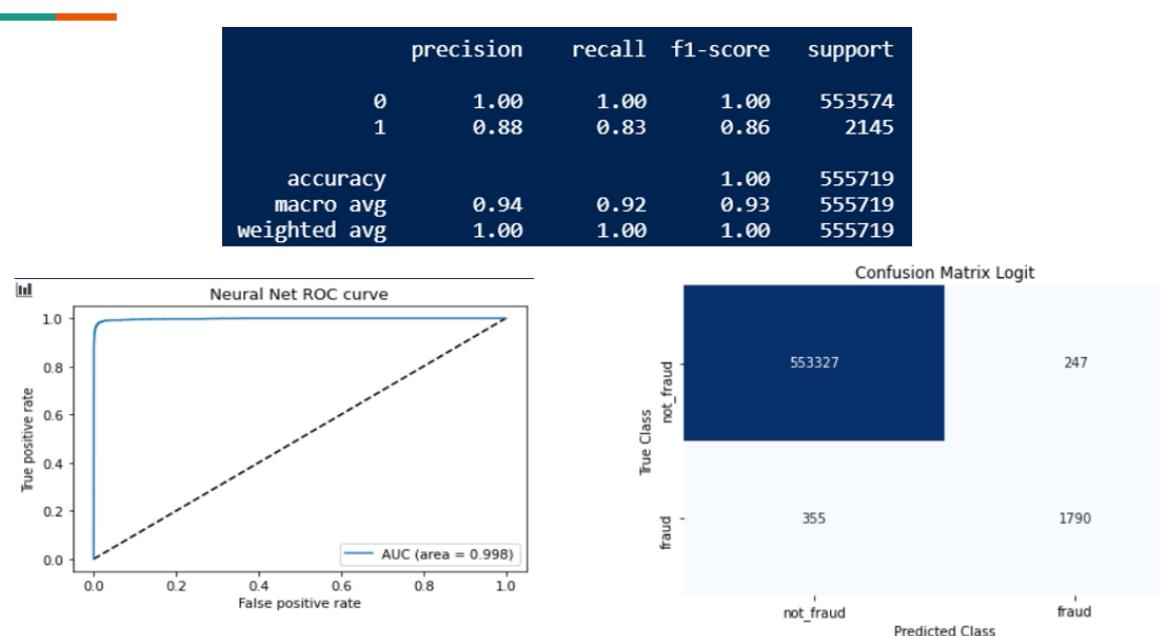


Sans les nouvelles features

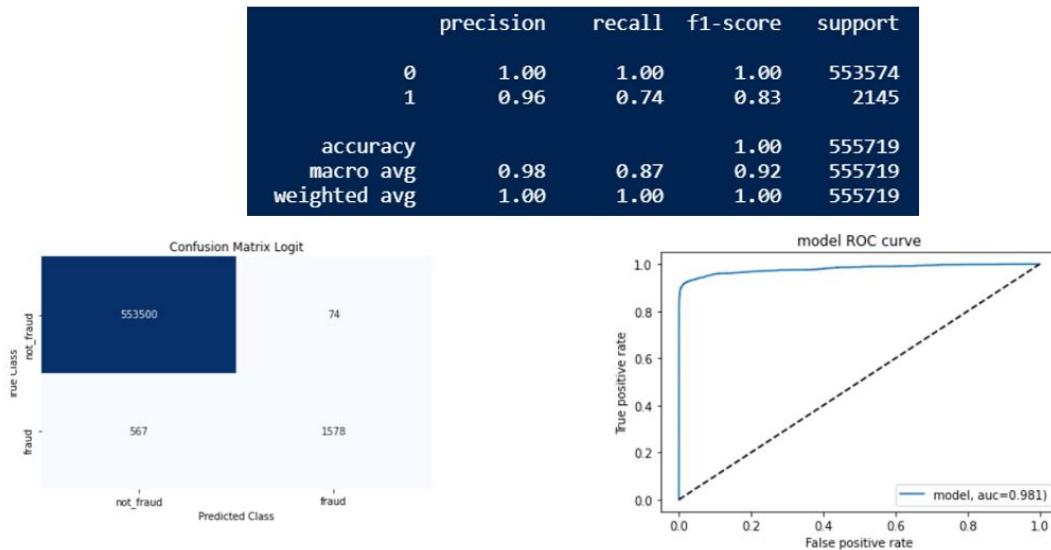


Avec les nouvelles features

Neural network



LSTM



Temps d'exécution

L'objectif est de pouvoir adopter nos meilleurs algorithmes et de les distribuer en temps réel. Il est donc important d'avoir un temps d'exécution le moindre possible.

On prendra en considération nos 3 meilleurs algorithmes tels sont :

- XGboost
- Random Forest
- Neural network

Principe

- Recevoir les transactions en forme brute
- les transactions arrivent d'une manière séquentielle
- On mesure le temps écoulé dès l'arrivée d'une transaction jusqu'à l'obtention de la prédiction.

Etapes

- Préparation : préparation des transactions passés nécessaire pour la classification (last_purchase par exemple)
- Feature Engineering: génération des nouvelles Features (Day, Hour, Delta_time ...)
- Preprocessing Time: transformation des Features (One Hot Encoding, Label Encoding, gender to binary ...)
- Prediction Time: prédiction de la nature de cette transaction (transformée)

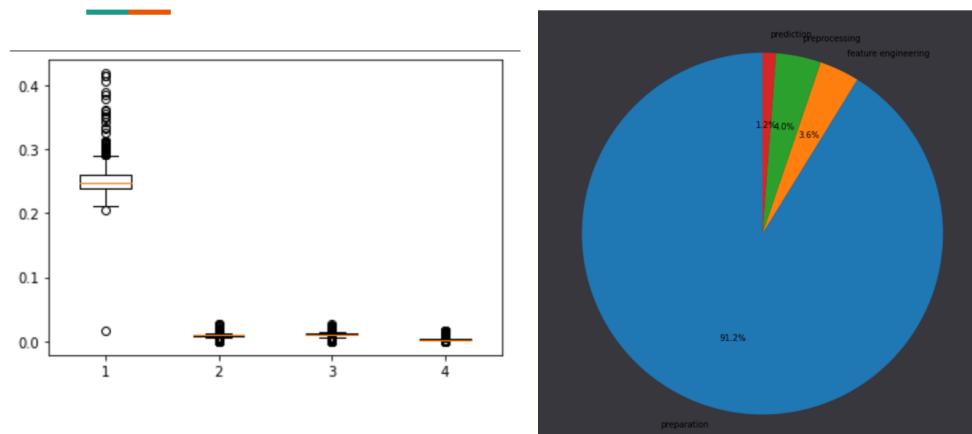
Pourquoi la division en plusieurs étapes

- Optimiser la partie qui prend le plus de temps
- En changeant le modèle, et en gardant les 3 premiers étapes, on refait que la Prediction Time
- En général: ne pas faire des expériences redondantes

Temps d'exécution d'XGboost

- 1000 échantillons
- Historique: 1 037 340 transactions
- Préparation: chercher à chaque fois les informations dans toute l'historique

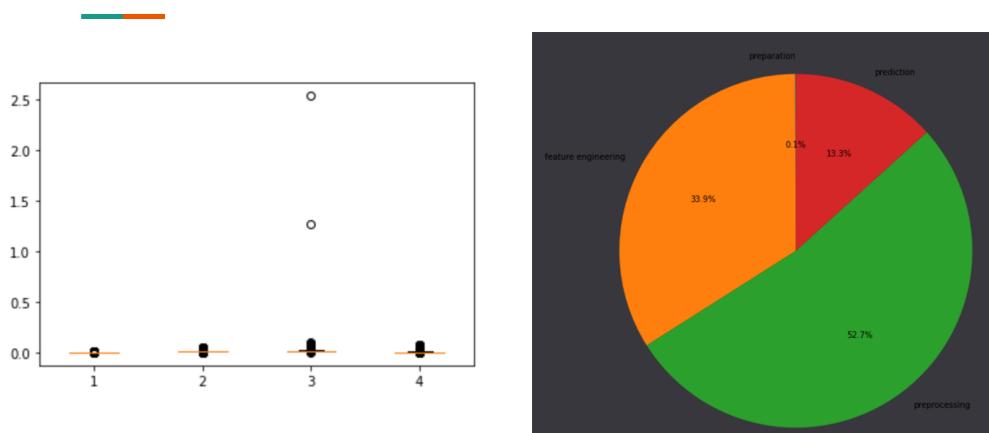
Average time for preparation: 251.30ms
Average time for feature engineering: 9.83ms
Average time for preprocessing: 11.10ms
Average time for prediction: 3.31ms
Average total time: 275.55ms



XGboost v2

- 259 335 échantillons
- Historique: 1 037 340 transactions
- Stocker que les informations potentiellement nécessaire dans un dictionnaire
- Préparation: chercher dans le dictionnaire

Average time for preparation: 0.02ms
Average time for feature engineering: 9.76ms
Average time for preprocessing: 15.17ms
Average time for prediction: 3.83ms
Average total time: 28.78ms



Récapitulatif

Models sorted by recall_1

	No sampling	Over	Under	SMOTE	ADASYN	NearMiss	f1_score_1	recall_1	precision_1	AUC
Decision Tree						*	0.09	0.88	0.05	0.9482
Random Forest opt				*			0.88	0.86	0.89	0.994
Decision Tree		*					0.39	0.85	0.26	0.9642
Neural Network						*	0.07	0.84	0.04	0.943
Neural Network opt	*						0.86	0.83	0.88	0.998
XGBoost		*					0.6	0.83	0.47	0.9937
XGBoost		*					0.56	0.83	0.43	0.994
Decision Tree		*					0.43	0.83	0.29	0.957
XGBoost			*				0.19	0.83	0.11	0.9817
Random Forest				*			0.08	0.83	0.04	0.956
XGBoost				*			0.64	0.82	0.52	0.9918
Decision Tree			*				0.61	0.82	0.48	0.9815
Random Forest		*					0.56	0.82	0.42	0.9871

Models sorted by precision_1

	No sampling	Over	Under	SMOTE	ADASYN	NearMiss	f1_score_1	recall_1	precision_1	AUC
XGBoost opt	*						0.86	0.78	0.96	0.984
LSTM opt	*						0.83	0.74	0.96	0.981
Random Forest opt				*			0.88	0.86	0.89	0.994
Neural Network opt	*						0.86	0.83	0.88	0.998
LSTM	*						0.75	0.68	0.84	-1.0
XGBoost	*						0.72	0.66	0.78	0.9812
Random Forest	*						0.69	0.62	0.78	0.9828
Neural Network	*						0.66	0.6	0.75	0.9866
Random Forest		*					0.68	0.64	0.73	0.9852
Random Forest			*				0.69	0.71	0.67	0.9883
Random Forest				*			0.68	0.69	0.67	0.9889
Decision Tree	*						0.59	0.63	0.55	0.973
XGBoost		*					0.64	0.81	0.53	0.9922

Models sorted by f1_score_1

	No sampling	Over	Under	SMOTE	ADASYN	NearMiss	f1_score_1	recall_1	precision_1	AUC
Random Forest opt				*			0.88	0.86	0.89	0.994
Neural Network opt	*						0.86	0.83	0.88	0.998
XGBoost opt	*						0.86	0.78	0.96	0.984
LSTM opt	*						0.83	0.74	0.96	0.981
LSTM	*						0.75	0.68	0.84	-1.0
XGBoost	*						0.72	0.66	0.78	0.9812
Random Forest			*				0.69	0.71	0.67	0.9883
Random Forest	*						0.69	0.62	0.78	0.9828
Random Forest				*			0.68	0.69	0.67	0.9889
Random Forest		*					0.68	0.64	0.73	0.9852
Neural Network	*						0.66	0.6	0.75	0.9866
XGBoost		*					0.64	0.81	0.53	0.9922
XGBoost			*				0.64	0.82	0.52	0.9918

Bagging :

- Utilisation des 3 modèles pour prédire
- Temps d'exécution: le temps d'exécution le plus élevés (Random Forest dans ce cas)
- 3 approches:
 - 1 vote fraude -> fraude: recall élevé

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.81	0.90	0.86	2145
accuracy			1.00	555719
macro avg	0.91	0.95	0.93	555719
weighted avg	1.00	1.00	1.00	555719

- 3 votes fraude -> fraude: précision élevée

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.99	0.73	0.84	2145
accuracy			1.00	555719
macro avg	1.00	0.86	0.92	555719
weighted avg	1.00	1.00	1.00	555719

- 2 votes fraude -> fraude: fscore élevé

	precision	recall	f1-score	support
0	1.00	1.00	1.00	553574
1	0.95	0.84	0.89	2145
accuracy			1.00	555719
macro avg	0.97	0.92	0.95	555719
weighted avg	1.00	1.00	1.00	555719

Théorie des jeux : Adversarial learning

On désigne un ensemble d'échantillons par $\{(x_i, y_i) \in (X, Y)\}$ $n i=1$, où x_i est le i ème échantillon et $y_i \in \{-1, 1\}$ est son étiquette, $X \subseteq \mathbb{R}^d$ est un espace de caractéristiques d -dimensionnel, n est le nombre total d'échantillons. Nous considérons un problème d'apprentissage adversatif où l'adversaire modifie les données malveillantes pour éviter la détection et donc atteindre ses objectifs planifiés. L'adversaire a la liberté de déplacer uniquement les données malveillantes ($y_i = 1$) dans n'importe quelle direction en ajoutant un vecteur de déplacement non nul δ_i à $x_i | y_i=1$.

Nous ne faisons aucune hypothèse spécifique sur la connaissance du système d'apprentissage par l'adversaire. Au lieu de cela, nous supposons simplement qu'il existe un compromis ou un coût pour modifier les données malveillantes. Par exemple, une stratégie pratique souvent employée par un adversaire consiste à déplacer les données malveillantes dans l'espace des caractéristiques aussi près que possible de l'endroit où les données inoffensives sont fréquemment observées. Cependant, l'adversaire ne peut modifier un point de données malveillantes qu'au point de ne pas perdre complètement son utilité malveillante. Si l'adversaire déplace un point de données trop loin de sa propre classe dans l'espace des caractéristiques, il peut avoir à sacrifier une

grande partie de l'utilité malveillante du point de données original. Par exemple, dans le problème de la détection de la fraude par carte de crédit, un attaquant peut choisir le "bon" montant à dépenser avec une carte de crédit volée pour imiter un achat légitime.

Free range attack

La seule connaissance dont l'adversaire a besoin est la portée de chaque caractéristique. Soit $x_{\max,j}$ et $x_{\min,j}$ la plus grande et la plus petite valeur que peut prendre la j ème caractéristique d'un point de données x_i . À toutes fins utiles, nous supposons que $x_{\max,j}$ et $x_{\min,j}$ sont tous deux bornés.

Une attaque est alors délimitée sous la forme suivante :

$$C_f(x_{\min,j} - x_{ij}) \leq \delta_{ij} \leq C_f(x_{\max,j} - x_{ij})$$

, $\forall j \in [1, d]$, où $C_f \in [0, 1]$ contrôle l'agressivité des attaques. $C_f = 0$ signifie l'absence d'attaques, tandis que $C_f = 1$ correspond aux attaques les plus agressives impliquant le plus large éventail de mouvements de données autorisés. Le grand avantage de ce modèle d'attaque est qu'il est suffisamment général pour couvrir tous les scénarios d'attaque possibles en ce qui concerne la modification des données. Associée à un modèle d'apprentissage, la combinaison produirait de bonnes performances contre les attaques les plus sévères. Cependant, lorsqu'il y a attaques légères, le modèle d'apprentissage devient trop "paranoïaque" et ses performances s'en ressentent.

Restrained attack

Soit x_i un point de données malveillant que l'adversaire vise à

Soit x_t , un vecteur à d dimensions, une cible potentielle vers laquelle l'adversaire voudrait pousser x_i . L'adversaire choisit x_t en fonction de son estimation de la distribution des données inoffensives. Idéalement, l'adversaire optimiserait x_t pour chaque x_i afin de minimiser le coût de sa modification et de maximiser le but qu'il peut atteindre. Le choix optimal de x_t est souhaité, mais nécessite souvent une grande connaissance de l'espace des caractéristiques et parfois le fonctionnement interne d'un algorithme d'apprentissage. De manière plus réaliste, l'adversaire peut définir x_t comme étant le centroïde estimé des données inoffensives, un point de données échantillonné à partir des données inoffensives observées, ou un point de données artificiel généré à partir de la distribution estimée des données inoffensives. Dans la plupart des cas, l'adversaire ne peut pas changer x_i en x_t comme il le souhaite car x_i pourrait perdre trop de son utilité malveillante. Par conséquent, pour chaque attribut j dans l'espace de caractéristiques à d dimensions, nous supposons que l'adversaire ajoute δ_{ij} à x_{ij} où

$$|\delta_{ij}| \leq |x_{ij}^t - x_{ij}|, \quad \forall j \in d.$$

En outre, nous plaçons une limite supérieure sur la quantité de déplacement pour l'attribut j comme suit :

$$0 \leq (x_{ij}^t - x_{ij})\delta_{ij} \leq \left(1 - C_\delta \frac{|x_{ij}^t - x_{ij}|}{|x_{ij}| + |x_{ij}^t|}\right) (x_{ij}^t - x_{ij})^2,$$

où $C_\delta \in [0, 1]$ est une constante modélisant la perte d'utilité malveillante suite au déplacement δ_{ij} .

Ce modèle d'attaque spécifie à quel point l'adversaire peut pousser x_{ij} vers x_t en fonction de la distance qui les sépare. Le terme $1 - C_\delta |x_t - x_{ij}| / (|x_{ij}| + |x_t|)$ est le pourcentage de $x_t - x_{ij}$ que δ_{ij} est autorisé à être au maximum. Lorsque C_δ est fixé, plus x_{ij} est proche de x_t , plus x_{ij} est autorisé à se rapprocher de x_t en pourcentage. L'inverse est également vrai. Plus x_{ij} et x_t sont éloignés, plus $|\delta_{ij}|$ sera petit. Par exemple, lorsque x_{ij} et x_t se trouvent de part et d'autre de l'origine, c'est-à-dire que l'un est positif et l'autre négatif, aucun mouvement n'est autorisé (c'est-à-dire $\delta_{ij} = 0$) lorsque $C_\delta = 1$.

Ce modèle établit un équilibre entre les besoins de déguiser la malveillance des données et de conserver leur utilité malveillante dans le temps. $(x_t - x_{ij})\delta_{ij} \geq 0$ garantit que δ_{ij} se déplace dans la même direction que $x_t - x_{ij}$. C_δ est lié à la perte d'utilité malveillante après la modification des données. C_δ définit la quantité d'utilité malveillante que l'adversaire est prêt à sacrifier pour franchir la frontière de décision. Un plus grand C_δ signifie une plus petite perte d'utilité malveillante, tandis qu'un plus petit C_δ modèle une plus grande perte d'utilité malveillante. Par conséquent, un C_δ plus grand conduit à des attaques moins agressives tandis qu'un C_δ plus petit conduit à des attaques plus agressives.

Expériences : Resultats

Après avoir appliqué la partie théorique, notre but était d'entrainer un modèle SVM qui essayera de couvrir à la fois la criticité d'une potentielle attaque et avoir une performance égale ou supérieure qu'un SVM normal dans

le cas où aucune attaque ne survient. Il est donc nécessaire de comparer les modèles free range,retrained avec des modèles sans attaque.

Commençons par le modèle free range :

	TowardCenterAttack_f=0	TowardCenterAttack_f=0.1	TowardCenterAttack_f=0.3	TowardCenterAttack_f=0.5
fr_naive_Cf=0.05	0.912389	0.920516	0.905270	0.858477
fr_naive_Cf=0.1	0.908589	0.918660	0.913835	0.902151
fr_naive_Cf=0.15	0.907836	0.916809	0.914580	0.904053
fr_naive_Cf=0.2	0.909655	0.912281	0.911157	0.900938
fr_naive_Cf=0.25	0.909154	0.911034	0.910283	0.898500
fr_naive_Cf=0.3	0.911096	0.910345	0.908463	0.897418
fr_naive_Cf=0.5	0.916351	0.907703	0.898916	0.896213
fr_naive_Cf=0.7	0.913734	0.896987	0.875714	0.879715
fr_naive_Cf=0.9	0.908526	0.892857	0.857659	0.840059
Normal SVM	0.912923	0.896528	0.824804	0.695147

	TowardCenterAttack_f=0.7	TowardCenterAttack_f=0.9
fr_naive_Cf=0.05	0.810629	0.778162
fr_naive_Cf=0.1	0.880000	0.860641
fr_naive_Cf=0.15	0.889825	0.885915
fr_naive_Cf=0.2	0.888967	0.880708
fr_naive_Cf=0.25	0.888420	0.882915
fr_naive_Cf=0.3	0.887716	0.883787
fr_naive_Cf=0.5	0.891939	0.883292
fr_naive_Cf=0.7	0.876918	0.873703
fr_naive_Cf=0.9	0.844298	0.850600
Normal SVM	0.550230	0.414315

Ci-dessous le modèle sans attaque :

	recall	precision	fscore	var_recall
Cf=0.05	0.837296	0.934443	0.883206	4.335499
Cf=0.1	0.821445	0.933757	0.874008	5.347330
Cf=0.15	0.817716	0.937467	0.873506	5.627848
Cf=0.2	0.820979	0.937201	0.875249	5.188577
Cf=0.25	0.820513	0.937666	0.875186	3.974642
Cf=0.3	0.823310	0.934886	0.875558	3.873731
Cf=0.5	0.829837	0.936842	0.880099	3.306787
Cf=0.7	0.820513	0.942689	0.877368	3.750406
Cf=0.9	0.820979	0.942719	0.877648	4.281028

Passons au modèle restreint :

	TowardCenterAttack_f=0.5	TowardCenterAttack_f=0.7	TowardCenterAttack_f=0.9
res_naive_Cd=0.95_Ce=1	0.857861	0.799245	0.742789
res_naive_Cd=0.9_Ce=1	0.892919	0.858273	0.821389
res_naive_Cd=0.85_Ce=1	0.909466	0.887640	0.862069
res_naive_Cd=0.8_Ce=1	0.917663	0.902287	0.886859
res_naive_Cd=0.75_Ce=1	0.921295	0.910155	0.898016
res_naive_Cd=0.7_Ce=1	0.924695	0.914384	0.905765
res_naive_Cd=0.65_Ce=1	0.928233	0.920191	0.913894
res_naive_Cd=0.6_Ce=1	0.931803	0.924541	0.919030
res_naive_Cd=0.5_Ce=1	0.934592	0.930988	0.928087
res_naive_Cd=0.3_Ce=1	0.940148	0.945114	0.950383
res_naive_Cd=0.1_Ce=1	0.939710	0.951715	0.958650
Normal SVM	0.695147	0.550230	0.414315
	TowardCenterAttack_f=0.5	TowardCenterAttack_f=0.7	TowardCenterAttack_f=0.9
res_naive_Cd=0.95_Ce=1	0.857861	0.799245	0.742789
res_naive_Cd=0.9_Ce=1	0.892919	0.858273	0.821389
res_naive_Cd=0.85_Ce=1	0.909466	0.887640	0.862069
res_naive_Cd=0.8_Ce=1	0.917663	0.902287	0.886859
res_naive_Cd=0.75_Ce=1	0.921295	0.910155	0.898016
res_naive_Cd=0.7_Ce=1	0.924695	0.914384	0.905765
res_naive_Cd=0.65_Ce=1	0.928233	0.920191	0.913894
res_naive_Cd=0.6_Ce=1	0.931803	0.924541	0.919030
res_naive_Cd=0.5_Ce=1	0.934592	0.930988	0.928087
res_naive_Cd=0.3_Ce=1	0.940148	0.945114	0.950383
res_naive_Cd=0.1_Ce=1	0.939710	0.951715	0.958650
Normal SVM	0.695147	0.550230	0.414315

Modèle sans attaque :

	recall	precision	fscore	var_recall
Cd=0.95,Ce=1	0.849417	0.929592	0.887698	3.349400
Cd=0.9,Ce=1	0.845688	0.933608	0.887476	3.570048
Cd=0.85,Ce=1	0.841958	0.936236	0.886598	3.900433
Cd=0.8,Ce=1	0.839161	0.937012	0.885391	3.739523
Cd=0.75,Ce=1	0.835897	0.934828	0.882599	3.180601
Cd=0.7,Ce=1	0.835431	0.936749	0.883194	3.564040
Cd=0.65,Ce=1	0.835431	0.936259	0.882976	3.612915
Cd=0.6,Ce=1	0.834499	0.938647	0.883514	3.582265
Cd=0.5,Ce=1	0.833566	0.940063	0.883617	3.571083
Cd=0.3,Ce=1	0.825175	0.939490	0.878630	2.931111
Cd=0.1,Ce=1	0.823310	0.938862	0.877298	3.390322

Difficultés rencontrées

Il est à noter que durant le stage que j'ai effectué au sein de la société Acensi, tout s'est déroulé sans soucis majeur à part quelques imprévus et désavantages qui n'ont nullement impacté le résultat et l'objectif du stage mais néanmoins ont contribué à rendre la tâche difficile.

Les difficultés étaient d'ordre organisationnel. Mon stage était principalement du travail à distance à cause de la crise sanitaire, cela a contribué à quelques préjudices mais ça n'a pas été d'un grand impact vu que j'étais encadré à distance en utilisant la plateforme Teams et j'avais de fréquentes réunion avec le directeur technique et les directeur adjoint pour leur montrer l'avancée de mon travail.

Suite du projet chez Acensi

Suite au travail que j'ai réalisé au sein de la société, j'ai pu prouver que la datascience malgré ses limites peut être utilisée pour la résolution d'un telle problématique. Grace au stage, la société a un POC (Proof of concept) sur laquelle elle pourra se baser pour la continuité du projet.

Conclusion

Pour conclure, j'ai effectué mon stage de fin d'étude au sein de la société Acensi. Lors de ce stage, j'ai pu mettre en pratique mes connaissances théoriques acquises tout au long de mon cursus universitaire. J'ai eu l'occasion de réaliser les taches qui m'ont été contribuée, ces dernières ont constitués ma mission de stage globale. Chacune de ses taches ont joué un rôle essentiel à ma formation et ont fait de la société une société qui priorise et pousse ses stagiaires vers l'avant. Je garde du stage un excellent souvenir, il constitue désormais une expérience professionnelle valorisante et encourageante pour mon avenir. Ce stage m'a permis également de comprendre les difficultés dont font face les data scientist surtout au niveau de la recherche de l'information et de la documentation. Telles difficultés sont principalement dues au nombre énorme de sources d'information disponible qui peuvent parfois leurs diriger vers une fausse ou route ou tout simplement vers l'erreur. Si je devrais donner une raison pour laquelle je voudrais travailler dans cette entreprise c'est tout d'abord parce j'ai reçus un encadrement de qualité ainsi qu'un accompagnement exemplaire. Une aide est toujours apportée à ceux qui la demandent. Finalement, cette expérience m'a offerte une bonne préparation à ce qui nous attend nous futur ingénieur en mettant en avant les différents domaines d'informatique qui fut pour moi un grand réconfort sur mon désir de vouloir être un ingénieur en big data spécialisé en intelligence artificielle.

Table des figures

Figure 1: Machine learning deep learning et la data science	11
Figure 2: Les types de machine learning.....	12
Figure 3: Régression linéaire vs Classification	13
Figure 4: Réduction de dimensionnalité.....	15
Figure 5: Clustering.....	16
Figure 6: Packages de python	17
Figure 7: Etapes de développement d'un modèle de machine Learning	18
Figure 8: Overfitting vs Underfitting.....	19
Figure 9: Cross validation.....	19
Figure 10: Métrics d'évaluation. Regression vs Classification	20
Figure 11: Matrice de confusion.....	21
Figure 12: Hyper parameters tuning	22
Figure 13: Illustration d'une grid search	22
Figure 14 : Les densités entre les variables	31

Annexe : Webographie

- SVM code avec SMO

[GitHub - nianlonggu/SupportVectorMachine](https://github.com/nianlonggu/SupportVectorMachine): simple implementation of the support vector machine algorithm, for detail see my blog <https://nianlonggu.com>

- SMO simplified

<https://chubek.github.io/svm/2020/12/27 smo-algorithm-simplified-copy.html>

- #### • SVM avec CVXOPT

SVM from scratch using Quadratic Programming | by Randeep ahlawat | Medium

- SVM visualization

[SVM Visualization](#)
[Visualization of SVM Kernels Linear, RBF, Poly and Sigmoid on Python \(Adapted from: http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html\)](#) .
[GitHub](#)

- CVX for non linear problems

Nonlinear Convex Optimization — CVXOPT User's Guide

- CVX pour SVM

Support Vector Machines — CVXOPT

SVM Implementation using CVXOPT - Python | Bhaskar's Blog (mbhaskar1.github.io)

[mbhaskar1.github.io/svm.py](https://github.com/mbhaskar1/svm-py) at master · mbhaskar1/mbhaskar1.github.io · GitHub

- Lagrange multiplier (forme dual de svm)

<https://ai-master.gitbooks.io/svm/content/lagrange-multiplier-and-dual-formulation.html>

- SVM (Kernel trick à la fin)

Understanding the mathematics behind Support Vector Machines (shuzhanfan.github.io)

- SMO (outil pour résoudre quadratique)

An Introduction to Support Vector Machines (SVM): Sequential Minimal Optimization (SMO)
- Gu Blog (nianlonggu.com)

- SMO

(57) 12 SMO and Stochastic SVM - YouTube

- Kernel trick

(57) SVM11 The Kernel Trick (Part2: Quadratic Polynomial Kernel) - YouTube

Understanding the Kernel Trick with fundamentals | by Chitta Ranjan | Towards Data Science

- CVXOPT (outil pour résoudre quadratique)

OpenCyberPDF (mit.edu)

EVALUATION DE L'ENTREPRISE
à compléter par le Maître de stage
à insérer dans le rapport de stage Obligatoirement

Fiche d'évaluation du stage
3^{ème} année Cycle Ingénieur
Promotion 2021

Etudiant : SORY ANAS
Entreprise : ACENSI FINANCE
Maître de stage : KOUTHON JAMES
Téléphone: 06 16 34 66 00 @ James.kouthon@acensi.fr

Majeure : SI

Maîtrise des domaines scientifiques et techniques :

- Capacité d'analyse/compréhension des problèmes
- Mise en œuvre de ses connaissances
- Aptitude à acquérir de nouvelles connaissances (formation ou autoformation)

10/10

Maîtrise des méthodes et des outils de l'ingénieur :

- Méthodologie/organisation du travail, gestion de projet : développement d'un outil ou d'une méthodologie
- Synthèse et communication des résultats, maîtrise des outils de communication

10/10

Conduite de l'action et prise de décision :

- Réalisation des objectifs, qualité du travail réalisé
- Autonomie/initiative/créativité/ouverture
- Respect des procédures (qualité, sécurité, santé...)

10/10

Intégration dans une organisation et capacité d'animation :

- Capacité à s'intégrer dans une équipe : exprimer ses attentes, donner son point de vue, sens de l'écoute, accepter la critique et se remettre en cause
- Communication sur ses activités et aptitude à rendre compte (réunion, relation client...)
- Prise en compte des enjeux métiers et économiques

5/15

Respect des valeurs sociétales, sociales et environnementales :

- Appropriation des valeurs, des codes, et de la culture de l'équipe et de l'organisation
- Comportement éthique

5/15

TOTAL

60/40

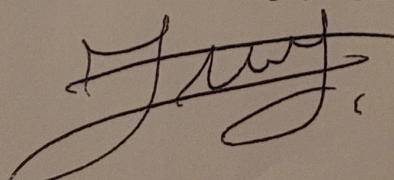
Appréciation globale sur le stage et observations

Stage très brillant. Tous les objectifs fixés ont été atteints. Très bonne capacité technique.

TOTAL

20/20

A Courbevoie 13/03/2021
 Signature du Maître de stage (obligatoire)



ACENSI FINANCE
 Rachet de l'entreprise
 SAS au capital de 140 000€
 TOUR BLACKPEARL
 14, Rue du Général Audran
 92400 Courbevoie
 Tel. 01 75 61 12 51 - Fax 01 41 43 25 79
 SIREN 488 984 014 APE 6202A

Fiche d'évaluation
Soutenance de Stage de fin d'études
Cycle Ingénieur 3^{ème} année
2020/2021

Etudiant : _____ Majeure : _____

Entreprise : _____

Composition du Jury

Représentant l'ECE : Président du Jury :

Représentant l'entreprise d'accueil :

Maître de Stage et invité(s) :.....

Soutenance effectuée le : /...../2021

Note

Présentation du contexte du stage	/5
- Présentation de l'entreprise et de sa politique RSE et description des activités - Définition et enjeux de la mission	
Présentation et valorisation du travail réalisé	/15
- Méthodologie et outils - Identification des connaissances utilisées et compétences acquises - Analyse critique de la réalisation des objectifs et du travail personnel réalisé	
Perspectives et bilan personnel	/10
- Perspectives de l'expérience en entreprise - Conséquences de ce stage sur votre avenir professionnel (secteur d'activité, type d'entreprise, fonctions, relation client, collaborations...) - Connaissance du métier de l'ingénieur : réflexion sur le métier d'ingénieur. La mission correspond-elle à ce que vous attendez du métier d'ingénieur ?	
Qualité générale de la soutenance	/5
- Gestion du temps - Qualité de l'expression orale (prestance, fluidité verbale, clarté, ...) - Efficacité des supports (visuels, graphiques, messages clés, ...)	
Sous total	/35
Qualité des réponses fournies au jury	/5
TOTAL	/40
Soit	/20

Observations :

Signatures des membres du Jury (Signature obligatoire du Président) :

Fiche d'évaluation
Rapport de stage de Fin d'Etudes
Cycle Ingénieur • 3^{ème} année
2020/2021

Etudiant :

Entreprise :

Majeure :

Correcteur :

Date de correction :.....

Présentation du contexte du stage	Note
<ul style="list-style-type: none"> - Présentation et histoire de l'entreprise, description des activités, organigramme - Politique de Responsabilité Sociétale de l'Entreprise (RSE) – - Définition et enjeux de la mission - Détails de la spécification des besoins, du CDC et du planning prévisionnel (Gestion de projet) – <u>Document Obligatoire</u> 	/5
Présentation et valorisation du travail réalisé	/20
<ul style="list-style-type: none"> - Méthodologie et outils - Connaissances préalables utilisées, et compétences acquises. - Qualité pédagogique du rapport - Qualité rédactionnelle 	
Bilan et perspectives	/15
<ul style="list-style-type: none"> - Analyse critique de la réalisation des objectifs et du travail personnel réalisé (prise d'initiative, créativité, recul, valeur ajoutée) - Difficultés rencontrées : réflexion sur les erreurs commises, les pertes de temps à postériori - Conséquences de ce stage sur votre avenir professionnel (secteur d'activité, type d'entreprise, fonctions, relation client, collaborations...) - Connaissance du métier de l'ingénieur : réflexion sur le métier d'ingénieur. La mission correspond-elle à ce que vous attendez du métier d'ingénieur ? - Perspectives 	
Total	/40
NOTE FINALE	/20

Observations :

A , le 2021

Signature du Correcteur (obligatoire)

Ingénieur en big data



SORY Anas

58 Rue Jean de la fontaine
75016 PARIS
+33 6 08 95 36 48
soryanas.stage@gmail.com

Né le : 11 / 07 / 1998
Célibataire

LANGUES

- > Anglais :Courant
- > Français :Bilingue
- > Arabe : Langue natale

LOGICIELS

- > Machine learning/Deep Learning
 - Python
 - Tensorflow
 - Scikit-learn
 - Pyspark
- > Développement web
 - Javascript / Html / CSS
 - Node js
 - Electron js
- > Data analytics
 - R

EXPÉRIENCE PROFESSIONNELLE

Présent

Avril.2021

Sept.2020

Mai.2020

Juillet.2019

Juillet -Sep.2018

Acensi Finance - PARIS

Stagiaire data scientist.

Développer un modèle d'intelligence artificielle pour la détection de fraude par carte bancaire en temps réel.
Théorie des jeux : Adapter le modèle aux éventuels nouveaux comportement des fraudeurs

FIVES - PARIS

Stagiaire développeur Electron Js / Node js.

évaluer la possibilité d'utiliser cette technologie dans un contexte très contraint de l'industriel pour des applications de type SCADA. Distribuer l'application dans le cadre de TWINCAT

OMPIC - CASABLANCA

Stagiaire développeur Javascript / Angular 8.

Elaboration à la création de la plateforme Web/mobile de création des entreprises en ligne.

Orange - CASABLANCA

Vendeur commercial et Chargé clientèle

Gagner en aisance face à un client pour appuyer la force commerciale de l'entreprise

Mes objectifs

Je souhaite monter en compétence en data science ainsi qu'en gestion de projet afin de viser un poste de directeur technique d'ici quelques années. Ce poste s'inscrit dans un plan de carrière

DEVELOPPEMENT PERSONNEL ET PROJETS

Certification Mooc Gestion de projet de la centrale de LILLE - PARIS

-Parcours Classique
-Spécialisation AS :Analyse stratégique dans les projets Spécialisation G2C :
Gestion de crise

PROJET DE FIN D'ETUDES : WISP - PARIS

-Prototype d'automatisation partiel de voitures visant à fluidifier le trafic.
-Utilisation du reinforcement learning pour des feux de croisement

FORMATION

2021-2019

2021-2019

2016

ECOLE CENTRALE D'ELECTRONIQUE - PARIS

Diplôme d'ingénieur informatique spécialisé en Big data

UNIVERSITE INTERNATIONALE DE CASABLANCA - CASABLANCA

Classes préparatoires intégrées
Tronc Commun -Système d'information

AL MOULTAKA - CASABLANCA

Baccalauréat- Physique Chimie

CENTRES D'INTÉRÊT



Natation(2006-2016)



Dames



Travail associatif



Psychologie