Rapport Détaillé: Détection de Fraudes sur les Cartes de Crédit

Gaye Alioune : https://www.linkedin.com/in/alioune-gaye-1a5161172

Introduction

Le présent rapport documente une analyse approfondie de la détection des fraudes sur les cartes de crédit, basée sur un jeu de données contenant 284 807 transactions enregistrées par des détenteurs de cartes européens en septembre 2013. Parmi ces transactions, seulement 492 (environ 0,17 %) sont signalées comme frauduleuses. Cette répartition extrêmement asymétrique pose des défis uniques pour l'entraînement et l'évaluation des modèles.

Le montant des transactions est relativement faible. Le montant moyen de toutes les transactions est d'environ 88 USD. Il n'y a pas de valeurs "Null", donc nous n'avons pas besoin de travailler sur des méthodes de remplacement de valeurs. La valeur minimale de Time est 0, ce qui représente le début des transactions enregistrées et sa valeur maximale est de 172792 secondes (environ 48 heures), ce qui signifie que les données couvrent environ deux jours.

Amount	Class
284807.000000	284807.000000
88.349619	0.001727
250.120109	0.041527
0.000000	0.000000
5.600000	0.000000
22.000000	0.000000
77.165000	0.000000
25691.160000	1.000000

	Time
count	284807.000000
mean	94813.859575
std	47488.145955
min	0.000000
25%	54201.500000
50%	84692.000000
75%	139320.500000
max	172792.000000

8 rows × 31 columns

Les données contiennent des variables transformées via une analyse en composantes principales (ACP) pour préserver la confidentialité, et deux variables principales : “Time” (temps écoulé depuis la première transaction) et “Amount” (montant de la transaction). L'objectif de ce projet est de construire un modèle prédictif pour classifier les transactions

comme frauduleuses ou non, tout en gérant les caractéristiques des données et les défis des classes déséquilibrées.

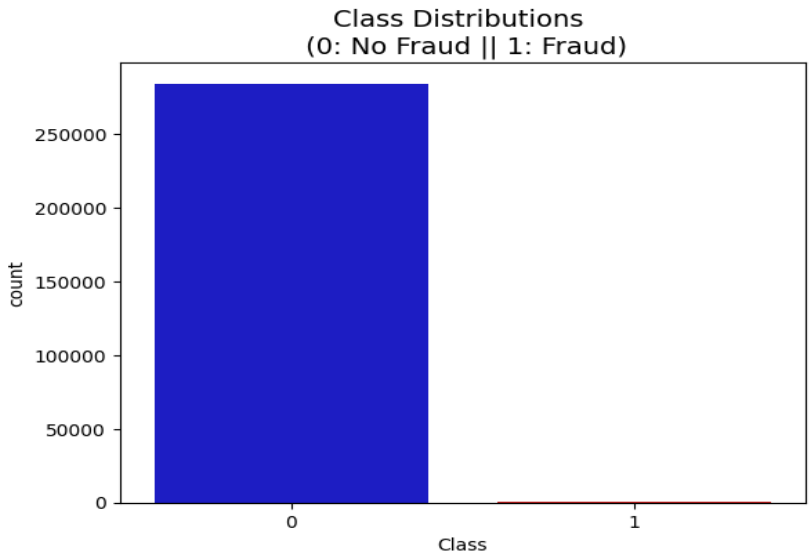
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739

5 rows × 31 columns

V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

Exploration des Données

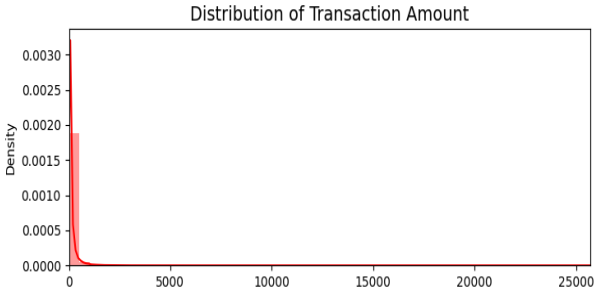
****Répartition des classes****

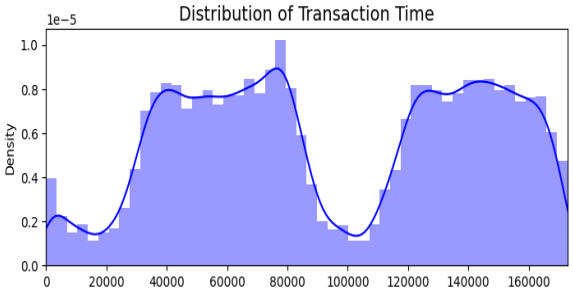


- **Transactions non frauduleuses**: 99,83 %
- **Transactions frauduleuses**: 0,17 %

Cette répartition met en évidence un fort déséquilibre, avec beaucoup plus de cas Non-Fraude (0) que de cas Fraude (1). Ce déséquilibre peut amener les modèles d'apprentissage automatique à être biaisés vers la prédiction de la classe majoritaire (Non-Fraude), ce qui entraîne une mauvaise performance en matière de détection de fraude. Pour aider les algorithmes à apprendre les motifs qui distinguent les transactions Frauduleuses des transactions Non-Frauduleuses, il est utile de créer un ensemble de données équilibré. Cependant, après l'entraînement, vous souhaitez tester le modèle sur les données originales, déséquilibrées, pour évaluer sa capacité de généralisation. Et le jeu de données original permet une évaluation réaliste des performances du modèle dans le monde réel, où les cas de fraude sont rares.

Caractéristiques principales:
- **Time**: Représente le temps écoulé depuis la première transaction enregistrée. Permet de capturer des motifs temporels dans les transactions.
- **Amount** : Montant de la transaction, essentiel pour détecter des anomalies.
- **V1 à V28** : Variables transformées par ACP, avec une variance conservée pour optimiser les performances des modèles.





****Mise à l'échelle** :**

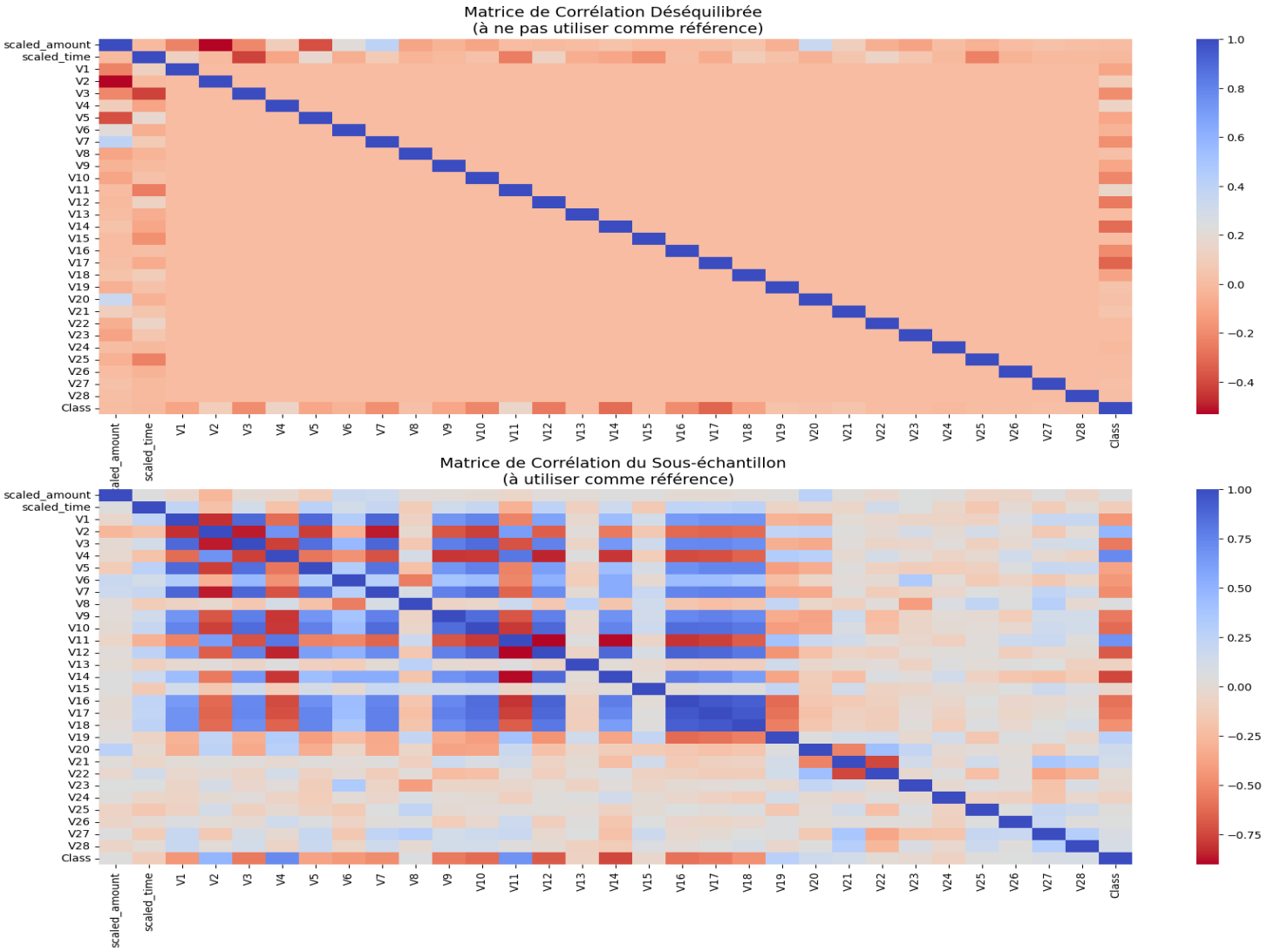
Ce que cela signifie : La mise à l'échelle est le processus de normalisation ou de standardisation des valeurs des caractéristiques pour s'assurer qu'elles sont dans la même plage. De nombreux algorithmes d'apprentissage automatique (comme ceux qui reposent sur des métriques basées sur la distance, tels que la régression logistique, les SVM ou les réseaux de neurones) fonctionnent mieux lorsque les caractéristiques sont sur des échelles similaires. Cela garantit qu'aucune caractéristique (comme un montant en dollars) ne domine les autres (comme les caractéristiques transformées par PCA). La caractéristique **Time** représente le temps en secondes depuis la première transaction. Ses valeurs sont assez grandes par rapport aux autres caractéristiques (qui sont déjà mises à l'échelle en raison de la PCA). Pour éviter de biaiser les modèles sensibles à l'échelle, cette caractéristique doit être mise à l'échelle. La caractéristique **Acount** représente le montant de la transaction en dollars (ou dans la devise locale), qui peut varier considérablement. Pour garantir la cohérence, cette caractéristique doit également être mise à l'échelle, surtout parce qu'elle est dans une plage numérique différente par rapport aux caractéristiques transformées par PCA (V1 à V28).

Aucune valeur manquante n'a été identifiée dans les données, indiquant une préparation soignée.

Analyse des Corrélations

Fraudes sur les cartes de credit

Matrice de Corrélation
Une matrice de corrélation a été générée pour identifier les caractéristiques ayant une relation étroite avec les transactions frauduleuses.



Remarque : Il est important d'utiliser le sous-échantillon dans notre matrice de corrélation, sinon cette dernière sera affectée par le fort déséquilibre entre nos classes. Ce phénomène est dû au déséquilibre élevé des classes dans le dataframe original.

- **Corrélations Positives**: V2, V4, V11
- **Corrélations Négatives**: V17, V14, V12

Interprétation
Ces résultats indiquent que certaines variables présentent des motifs qui peuvent être exploités pour améliorer la détection des fraudes. Par exemple, une valeur élevée de V4 peut être un indicateur de fraude.

Gaye Alioune

Visualisation des Données

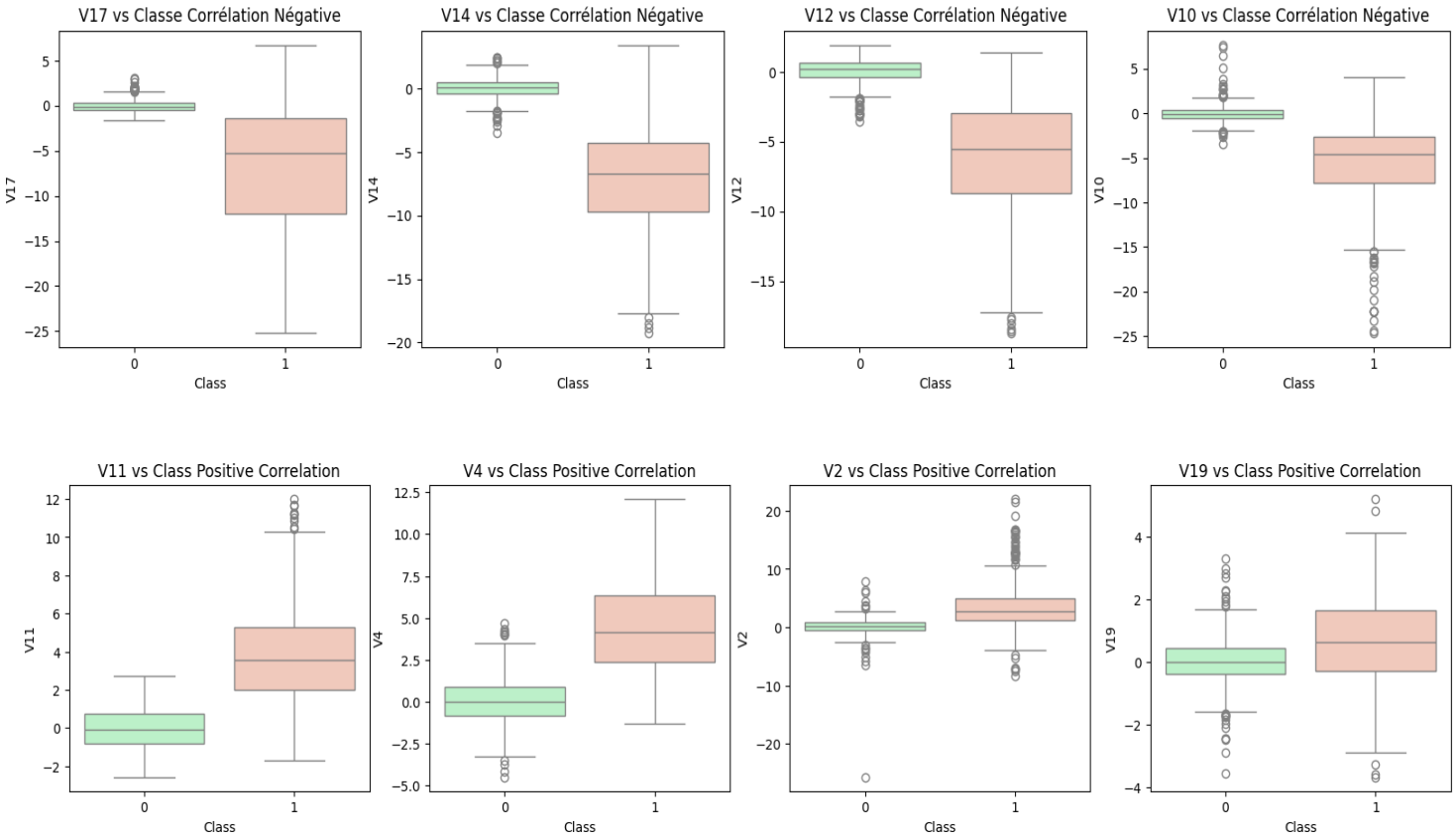
****Diagrammes en Boîte (Boxplots)****

Les boxplots (ou diagrammes en boîte) sont des outils visuels très utiles pour analyser la distribution des données. Ils permettent de visualiser facilement les percentiles, notamment :

- Q1 (25ème percentile) : la limite inférieure de la boîte.
- Q3 (75ème percentile) : la limite supérieure de la boîte.
- Valeurs aberrantes : les points qui se situent au-delà des limites extrêmes, c'est-à-dire en dehors des "whiskers" (les lignes qui s'étendent de la boîte).

Les boxplots ont révélé des valeurs aberrantes importantes dans des caractéristiques comme V14, V10 et V12, … associées aux fraudes.

Les variables V11, V4, V2, et V19 montrent toutes une corrélation positive avec la classe 1. V14, V10 et V12 montrent toutes une corrélation négative avec la classe 1.



Exemple:

- **V14**: Présente une forte dispersion pour les transactions frauduleuses. Les anomalies

dans cette variable sont des indicateurs forts de fraude. Les données de la classe 0 sont également regroupées autour de 0, avec peu de variation et quelques outliers visibles. Pour la classe 1, la médiane est négative, et les valeurs s'étendent jusqu'à environ −20. Ceci indique que V14 a des valeurs typiquement plus faibles pour la classe 1, confirmant une corrélation négative.

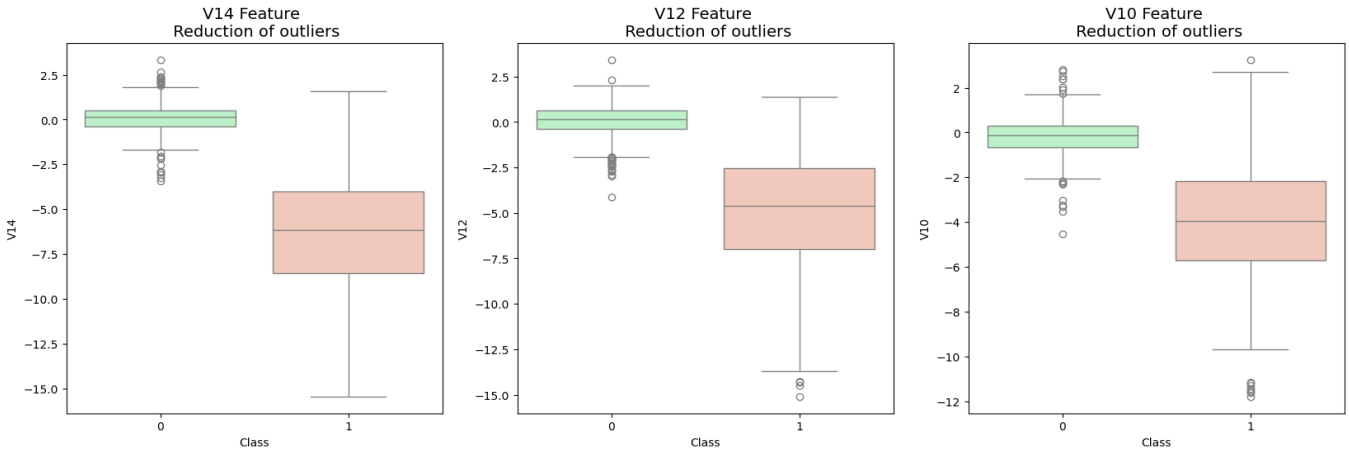
Méthode de l'Écart Interquartile (IQR) :

L'écart interquartile (IQR) est une mesure statistique qui permet d'évaluer la dispersion d'un ensemble de données. Il est calculé comme la différence entre le 75ème percentile (Q3) et le 25ème percentile (Q1).

Objectif : Mon but est de créer des seuils au-delà des percentiles 75 et 25. Si une instance dépasse ces seuils, elle sera considérée comme une valeur aberrante et sera supprimée. Lors de la suppression des valeurs aberrantes, il est crucial de trouver un équilibre entre la détection des valeurs aberrantes et la préservation des données pertinentes.

- ****Impact**:** La suppression de ces valeurs aberrantes a permis une amélioration de 3 % de la précision des modèles.

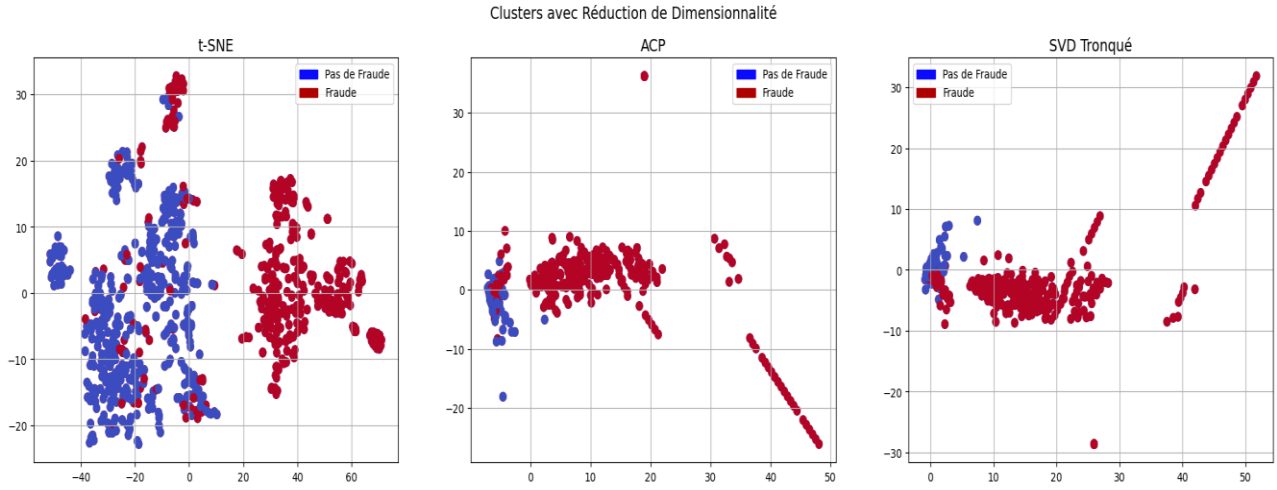
Vérification de la réduction des valeurs aberrantes dans nos étiquettes :



Techniques de Réduction de Dimensionnalité

Les techniques de réduction de dimensionnalité, comme l'ACP, t-SNE et SVD Tronqué, ont été appliquées pour mieux comprendre la structure des données.

Technique	Description
t-SNE	Méthode non-linéaire, efficace pour visualiser des clusters et des séparations dans une dimension fortement réduite.
PCA	Méthode linéaire qui conserve le plus de variance possible, utile pour identifier les principaux motifs dans les données.
SVD Tronqué	Autre technique linéaire, optimisée pour les données creuses, offrant une perspective différente sur l'apparence des clusters.



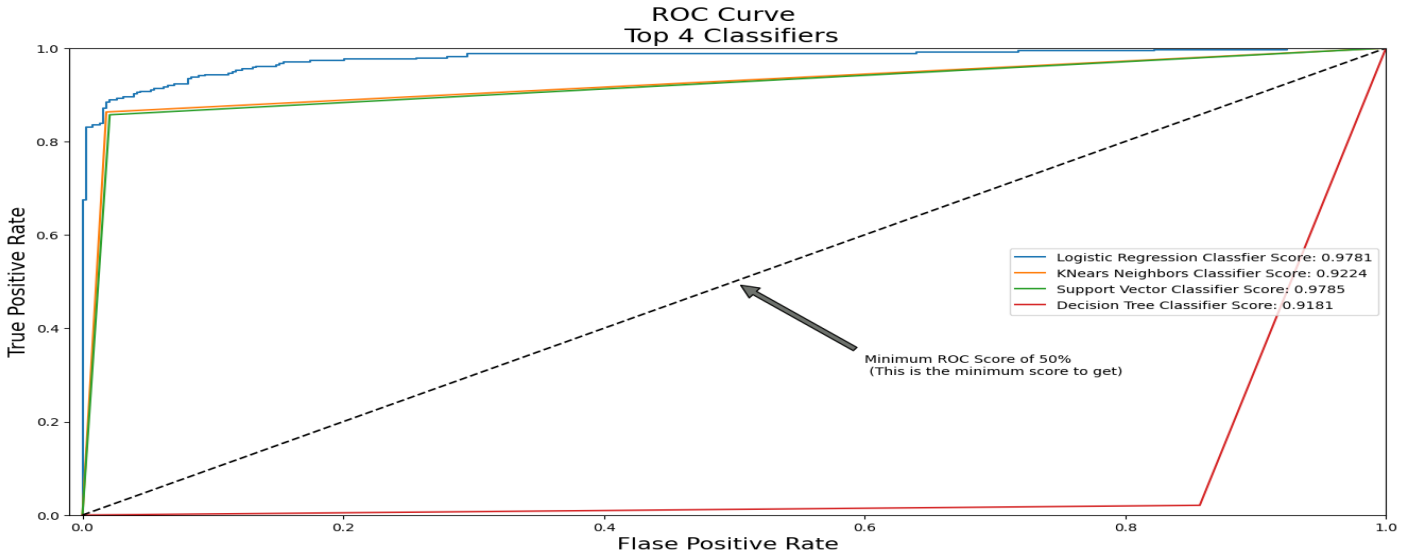
Bien que l'ACP ait été utilisée pour normaliser les données, cela concernait l'ensemble des données en haute dimension. Ici, nous appliquons t-SNE, ACP, et SVD Tronqué sur le sous-échantillon pour réduire les données à deux dimensions (x et y). L'objectif est d'observer si les cas de fraude et de non-fraude forment des clusters distincts ou des groupes séparés dans l'espace 2D.

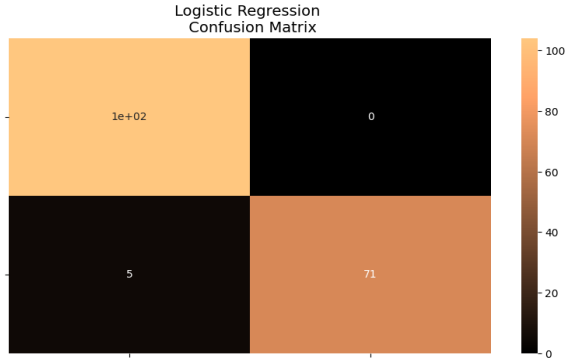
Modélisation et Évaluation

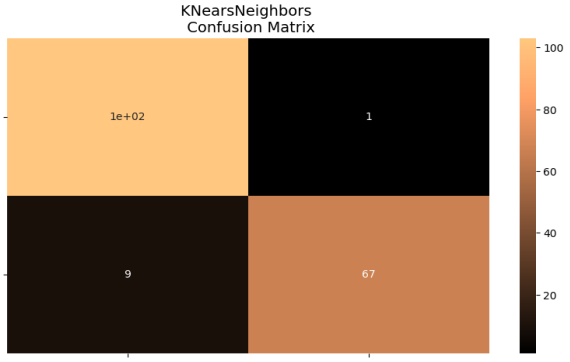
****Modèles Testés****

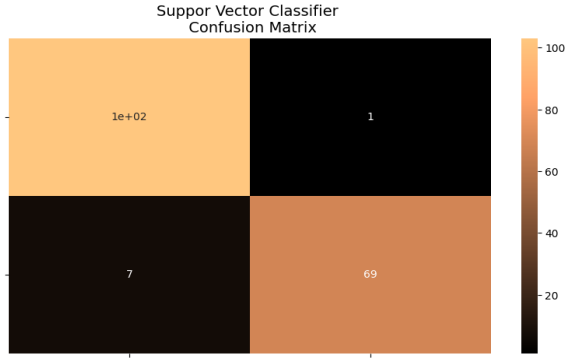
- Régression Logistique
- SVM (Support Vector Machine)
- K-Neighbors Classifier
- Arbre de Décision

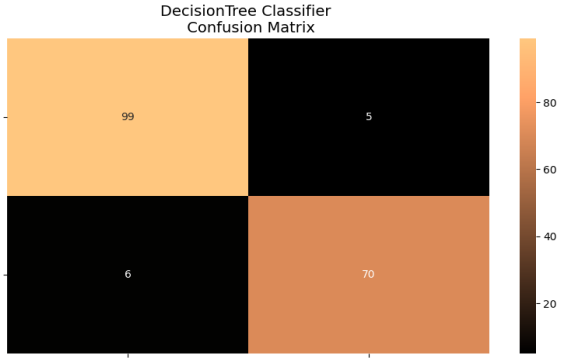
****Performances des Modèles****


ROC Curve
Top 4 Classifiers
True Positive Rate
Flase Positive Rate
Logistic Regression Classfier Score: 0.9781
KNears Neighbors Classifier Score: 0.9224
Support Vector Classifier Score: 0.9785
Decision Tree Classifier Score: 0.9181
Minimum ROC Score of 50%
(This is the minimum score to get)


Logistic Regression
Confusion Matrix
1e+02
0
5
71


KNearsNeighbors
Confusion Matrix
1e+02
1
9
67


Suppor Vector Classifier
Confusion Matrix
1e+02
1
7
69

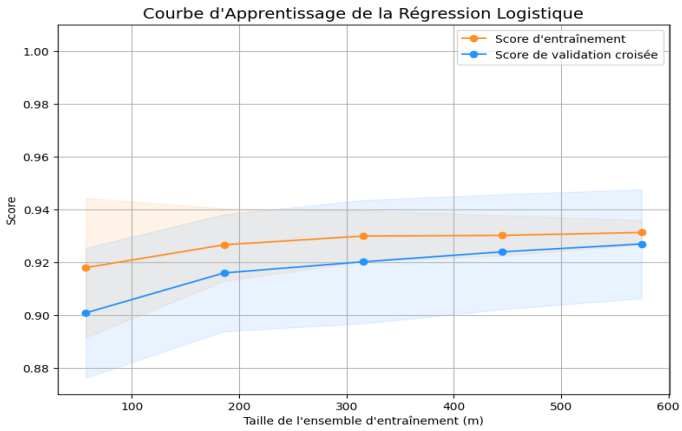

DecisionTree Classifier
Confusion Matrix
99
5
6
70

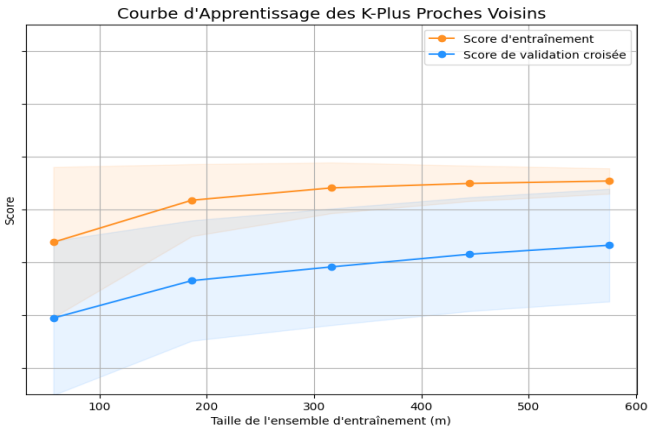
```
···   Logistic Regression:
                  precision    recall  f1-score   support

               0       0.95      1.00      0.98       104
               1       1.00      0.93      0.97        76

        accuracy                           0.97       180
       macro avg       0.98      0.97      0.97       180
    weighted avg       0.97      0.97      0.97       180

    KNears Neighbors:
                  precision    recall  f1-score   support

               0       0.92      0.99      0.95       104
               1       0.99      0.88      0.93        76

        accuracy                           0.94       180
       macro avg       0.95      0.94      0.94       180
    weighted avg       0.95      0.94      0.94       180

    Support Vector Classifier:
                  precision    recall  f1-score   support

               0       0.94      0.99      0.96       104
               1       0.99      0.91      0.95        76
    ...
        accuracy                           0.94       180
       macro avg       0.94      0.94      0.94       180
    weighted avg       0.94      0.94      0.94       180
```

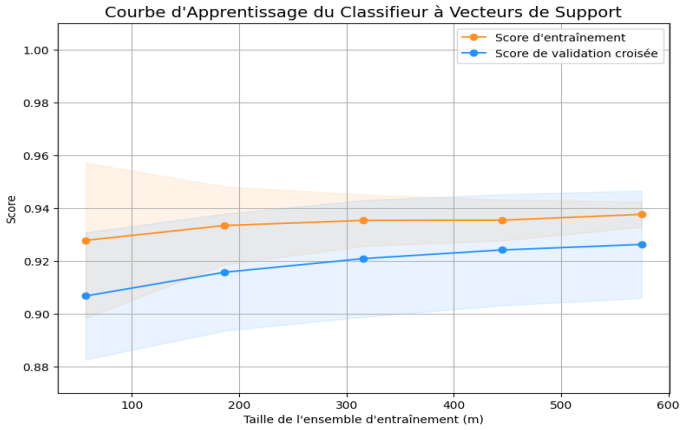
****Analyse des Résultats**:**
- **Régression Logistique**: Montre la meilleure combinaison entre précision et rappel, indiquant un modèle bien adapté à ce problème.
- **SVM**: Comparable à la régression logistique, mais avec un temps d'exécution plus élevé.

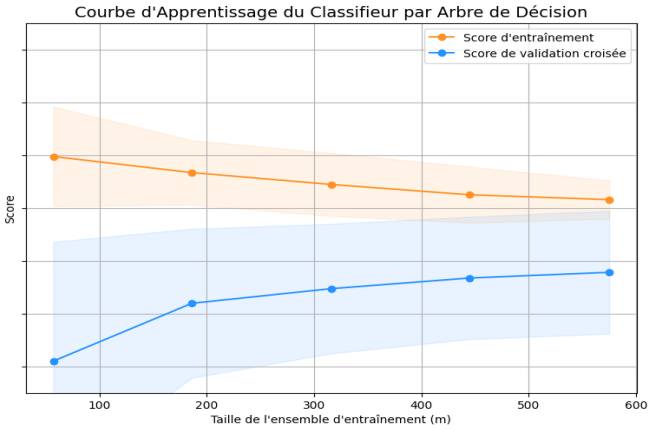
****Courbes d'Apprentissage****

Les courbes confirment que la régression logistique généralise bien sans surapprentissage, même avec un échantillonnage stratifié.









****Score final dans l'ensemble de test pour la régression logistique** :**

	Technique	Score
0	Sous-échantillonnage aléatoire	0.972222
1	Sur-échantillonnage (SMOTE)	0.980759

Conclusion et Recommandations

Ce projet a mis en lumière des approches efficaces pour gérer des données déséquilibrées et détecter les fraudes à l'aide de modèles robustes comme la régression logistique et le SVM.

Gaye Alioune

****Recommandations**:**
1. Intégrer SMOTE pour mieux gérer les données déséquilibrées.
2. Explorer des architectures avancées comme les réseaux neuronaux.
3. Tester davantage d'hyperparamètres pour les modèles actuels.

Avec ces approches, il est possible d'améliorer encore la précision et de réduire les faux positifs pour une détection optimale des fraudes.