

RAPPORT PROJET APPRENTISSAGE SUPERVISÉ

Etude comparative d'algorithmes d'apprentissage supervisé

Réalisé par :
Imed KERAGHEL
Oussama HEBROUNE

Encadré par :
Lazhar LABIOD

28 février 2021

TABLE DES MATIÈRES

Table des matières

1	Introduction	1
2	Données relationnelles	1
2.1	Description des données	1
2.2	Classification supervisée sur les données relationnelle	2
3	Carte visa	3
3.1	Description des données	3
3.2	Sélection de variables	3
3.3	Imputation de valeurs manquantes	4
3.4	Normalisation de données	4
3.5	Étude exploratoire de données	5
4	Fraude bancaire	8
4.1	Description des données	8
4.2	Étude exploratoire de données	9
5	Stratégies d'échantillonnage	10
5.1	SMOTE	11
5.2	Random UnderSampler	11
5.3	NearMiss	11
6	Résultats et discussion	12
6.1	Données relationnelles	12
6.2	Carte Visa	13
6.3	Fraude bancaire	15
7	Conclusion	17

1 Introduction

L'apprentissage supervisé est un domaine de recherche de l'intelligence artificielle. Il consiste à programmer des algorithmes permettant d'apprendre automatiquement de données et d'expériences passées, un algorithme cherchant à résoudre au mieux un problème considéré. Cette capacité de tirer des leçons de l'expérience et de l'observation analytique peut améliorer continuellement le système, augmentant ainsi son efficacité. L'apprentissage supervisé a un large éventail d'applications, nous prenons comme exemples : la reconnaissance vocale, le diagnostic médical, la détection de fraude, etc.

L'espace d'entrée est composé de données souvent divisées en un ensemble d'apprentissage et un autre de test. L'ensemble d'apprentissage consiste en un ensemble de points d'entrée dans un espace multidimensionnel. Le but consiste alors à trouver une correspondance entre les points d'entrée et certaines étiquettes correspondant à des catégories d'intérêt selon le domaine. L'ensemble de test est un ensemble d'exemples utilisé pour évaluer la performance de l'apprenant. La classification supervisée vise à définir des règles pour que les objets puissent être classés en fonction des variables qualitatives ou quantitatives qui les caractérisent.

Dans le cadre du Master 2 Machine Learning for Data Science (promotion 2020-2021) de l'université de Paris, il nous a été demandé, lors du cours apprentissage supervisé, d'étudier 5 jeux de données en utilisant les méthodes d'apprentissage supervisées vues en cours, afin de comparer les performances.

2 Données relationnelles

2.1 Description des données

La Table 1 résume les caractéristiques de chaque jeu de données.

Dataset	#individus	#liens	#variables	classes
Cora (fea, W, gnd)	2780	5429	1433	7
CitSeer (fea, W, gnd)	3327	4732	3703	6
Pubmed(fea, W, gnd)	19717	44338	500	3

TABLE 1 – Description des données relationnelles

Cora : Ce data set comporte 2780 publications scientifiques, chaque publication appartient à une des 7 classes. Le réseau de citations comporte 5429 liens. Chaque publication est décrite par un vecteur binaire de mot indiquant l'absence ou la présence du mot correspondant dans le dictionnaire (qui comporte 1433 mots uniques).

CitSeer : Ce data set comporte 3327 publications scientifiques, chaque publication appartient à une des 6 classes. Le réseau de citations comporte 4732 liens. Chaque publication est décrite

par un vecteur binaire de mot indiquant l'absence ou la présence du mot correspondant dans le dictionnaire (qui comporte 3703 mots uniques).

Pubmed : Ce data set comporte 19717 publications scientifiques, chaque publication appartient à une des trois classes. Le réseau de citations se compose de 44338 liens. Chaque publication de l'ensemble de données est décrite par un vecteur de mots pondéré.

2.2 Classification supervisée sur les données relationnelles

L'objectif de cette partie est de mener une étude comparative des différentes méthodes de classification sur ces données relationnelles en utilisant la matrice \mathbf{X} dans un premier temps, une combinaison des informations \mathbf{W} et \mathbf{X} dans un second temps, et finalement nous tentons d'autres méthodes permettant d'améliorer les performances.

Matrice \mathbf{X}

Cette approche consiste en l'utilisation de la matrice \mathbf{X} , i.e la matrice des features, tout en ignorant les relations qui existe entre les noeuds, i.e les individus, vu que nos datasets sont essentiellement des graphes, donc au lieu qu'un noeud dépendra que de son entourage (les noeuds avec lesquels il a des liens), il dépendra de toute la dataset.

Matrice \mathbf{M}

Pour y remédier au problème de la matrice \mathbf{X} , on la multiplie par la matrice d'adjacence \mathbf{W} , ce qui revient à respecter la propagation de l'information dans le réseau (graphe) et apprendre les représentations des nœuds en se basant sur leurs connectivités.

Cependant, la matrice \mathbf{W} n'est généralement pas normalisée et donc la multiplication avec cette matrice changera complètement l'échelle des vecteurs de \mathbf{X} . Normaliser $\mathbf{X} \cdot \mathbf{W}$ revient à normaliser \mathbf{W} en multipliant par l'inverse de la matrice \mathbf{D} , i.e la matrice des degrés de chaque nœud (cf. la formule 1).

$$M = D^{-1} \cdot W \cdot X \quad (1)$$

Matrice $\mathbf{W+I}$

La matrice \mathbf{M} représente la moyenne des features de l'entourage d'un nœud, et elle ignore les features du nœud lui même car : $\mathbf{w}_{ii} = \mathbf{0}$.

Pour résoudre ce problème, on ajoute une matrice identité \mathbf{I} à la matrice d'adjacence \mathbf{W} c.à.d ajouter des self loops, et puis on normalise avec l'inverse de la matrice \mathbf{D} (cf. la formule 2).

$$M = D^{-1} \cdot (W + I) \quad (2)$$

Normalisation symétrique

La normalisation avec l'inverse de la matrice \mathbf{D} normalise les lignes de la matrice \mathbf{W} . Or si on multiplie \mathbf{W} avec $\mathbf{D}^{-1/2}$ des deux cotés, on normalise les lignes et les colonnes de \mathbf{W} (cf. la formule 3).

$$M = D^{-1/2} \cdot (W + I) \cdot D^{-1/2} \quad (3)$$

3 Carte visa

3.1 Description des données

Elle s'agit d'une base de données très connue dans le domaine de l'apprentissage supervisé, cette dernière décrivant les clients d'une banque et leurs comportements (mouvements bancaires, soldes des différents comptes, etc). La variable à prédire est la variable binaire **possession de la carte Visa Premier**.

Dans un problème de classification, il arrive souvent d'avoir des data sets déséquilibrés. On parle d'un data set déséquilibré lorsque le ratio des observations d'une classe par rapport à l'ensemble des observations est faible. Nous remarquons, que ce jeu de données est assez déséquilibré, près de 2/3 des observations appartienne à la classe négative.

Table	#d'observations	#de variables	#de classes	Ratio
Visa Premier	1073	47	2	0.33

TABLE 2 – Description des données Carte Visa

3.2 Sélection de variables

C'est un processus qui permet de sélectionner un sous-ensemble de variables considérées comme pertinentes. La variable *matricul* représente l'identifiant client, ce qui la rend sans importance pour notre analyse. De plus, la variable *nbimpaye* a une valeur constante pour toutes les observations. C'est pour cette raison que nous avons décidé d'éradiquer ces variables, qui s'avèrent inutiles pour l'étude. Nous avons aussi retiré les variables qui ont une variance proche de zéro (near zero variance) comme : nbbon, mtbon, nbcb, nbpaiecb, etc. Ces variables ont pratiquement la même valeur pour toutes les observations.

La plupart des variables sont quantitatives, sauf 7 qui sont qualitatives :

- **departem** : département de résidence
- **ptvente** : point de vente
- **sexe** : sexe

- **sitfamil** : situation familiale
- **csp** : catégorie socio-professionnelle
- **sexer** : sexe (valeur binaire)
- **codeqlt** : code qualité client évalué par la banque

On remarque que la variable sexe apparaît deux fois, la première fois comme une chaîne de caractère et la deuxième comme une variable binaire. La même remarque est valable pour la variable cartevr. Nous pouvons donc supprimer ces doublons.

Notre ensemble qui initialement contenait 1073 observations et 47 variables, en contient 1072 lignes et 35 variables après cette étape de présélection.

3.3 Imputation de valeurs manquantes

Il arrive assez fréquemment que des observations soient incomplètes, i.e. les valeurs d'une ou plusieurs variables manquent. Une solution qui s'avère souvent meilleure consiste à imputer les données manquantes et à traiter les valeurs estimées comme des valeurs mesurées.

Dans notre jeu de données, plusieurs valeurs semblent manquer (valeur ".") , impliquant les variables suivantes : departem, codeqlt, agemvt et nbpaiecb. Il existe de nombreuses méthodes qui permettent de traiter ce problème. Dans le cadre de ce travail, nous avons choisi missMDA. Ce package permet de remplacer les données manquantes par des valeurs plausibles. Le principe est de prédire ces valeurs plausibles à partir d'un modèle qui prend en compte à la fois les similarités entre les individus et entre les variables, ce package prend en compte les données qualitatives et quantitatives(fonctions imputePCA et imputeMCA).

3.4 Normalisation de données

C'est une technique souvent appliquée dans le cadre de la préparation des données pour le Machine Learning. L'objectif de la normalisation est de modifier les valeurs des colonnes numériques du jeu de données pour utiliser une échelle commune, sans que les différences de plages de valeurs ne soient pas faussées et sans perte d'informations.

Par exemple, la variable age contient des valeurs allant de 18 à 65, tandis que les valeurs de moycred3 allant de 0 à 19579. La grande différence d'échelle des nombres peut donc poser un véritable problème lors de la combinaison des variables sous forme de fonctions. La normalisation permet d'échapper à ce problème en créant de nouvelles valeurs qui conservent la même distribution et les mêmes ratios que les données sources, tout en appliquant la même échelle aux valeurs des différentes colonnes numériques utilisées dans le modèle.

3.5 Étude exploratoire de données

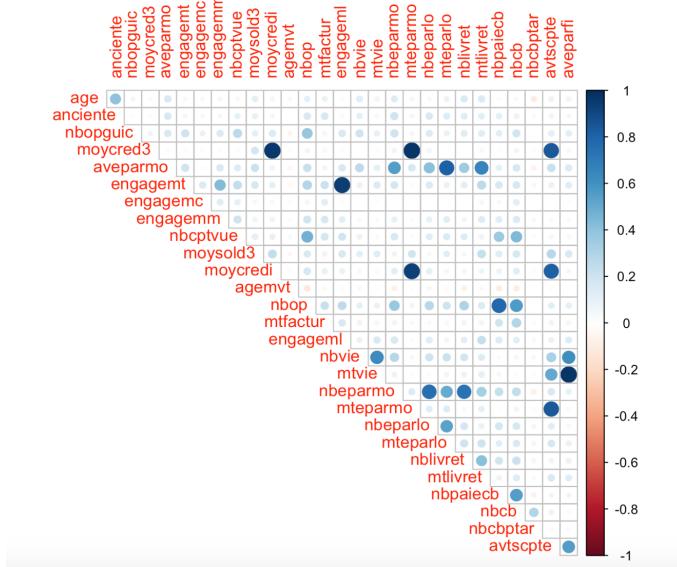


FIGURE 1 – Matrice de corrélation

Certaines variables semblent être fortement corrélées, voire complètement pour certaines d'entre elles. Il est intéressant de noter que le montant des produits contrats vie en francs (mtvie) est extrêmement corrélé avec le total des avoirs épargne financière en francs (aveparfi), cette dernière variable étant également fortement corrélée avec le total des avoirs sur tout les comptes (avtscpte), lui-même corrélé avec mtvie. Ces trois variables sont fortement corrélées entre elles. Le même commentaire peut être fait pour mteparmo, aveparmo et mteparlo.

aveparmo	mteparmo	0.99
mtvie	aveparfi	0.98
avtscpte	aveparfi	0.95
engagemt	engageml	0.95
mtvie	avtscpte	0.90
mteparmo	mteparlo	0.81
aveparmo	mteparlo	0.81
moycred3	moycredi	0.79
nbop	nbpaiecb	0.78
nbeparmo	nbeparlo	0.76

TABLE 3 – Les corrélations les plus élevées entre les variables

L'analyse en composantes principales (ACP) permet de transformer des variables quantitatives probablement liées entre elles, en nouvelles variables décorrélées les unes des autres. Cependant,

3 Carte visa

cette technique ne prend pas en compte les variables qualitatives. Nous pouvons effectuer une ACP sans ces variables, en les considérant comme des variables supplémentaires, et obtenir un bon résultat puisque nous n'avons pas beaucoup de variables qualitatives.

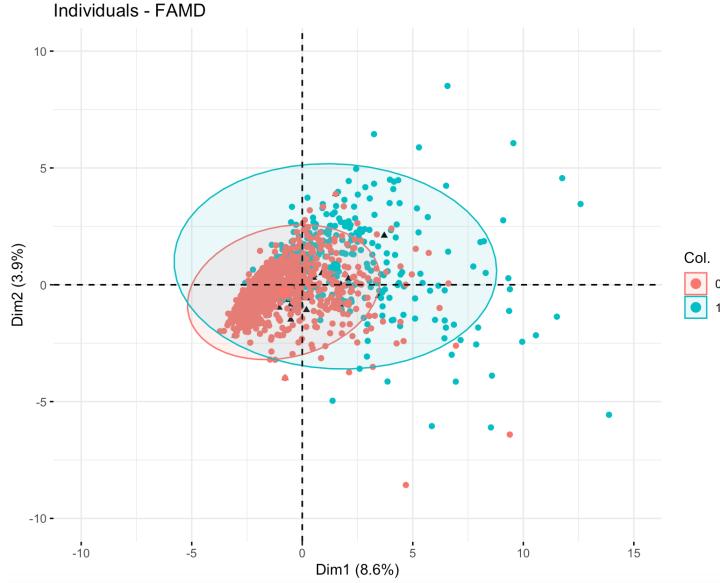


FIGURE 2 – Le premier plan factoriel de l'AFDM

Néanmoins, il serait beaucoup plus judicieux d'utiliser un algorithme capable de prendre en compte les deux types de variables, c'est le but de l'AFMD (analyse factorielle des données mixtes).

Sur le premier plan factoriel, nous remarquons le chevauchement des deux classes. Nous pouvons aussi dire que les variables fortement corrélées avec les deux composantes pourraient être importantes pour discriminer les deux classes, étant donné la forme des classes (ellipsoïde).

Nous constatons dans la figure 4, que les variables qualitatives ne semblent pas apporter beaucoup de variance aux données, puisque leur contribution est beaucoup plus faible par rapport aux variables quantitatives. Cependant, la modalité A de la variable qualitative **codeqlt** semble avoir une forte contribution et corrélation avec la première composante.

3 Carte visa

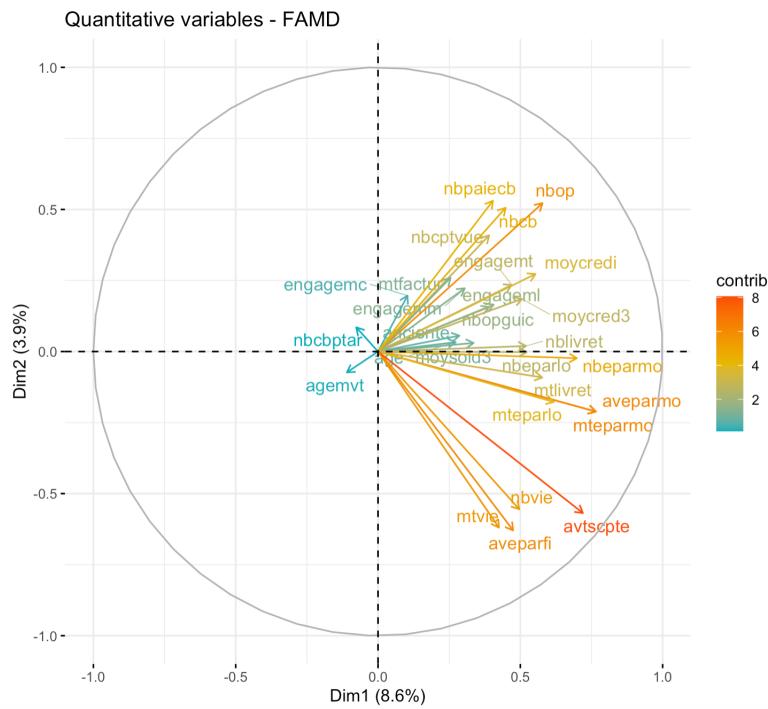


FIGURE 3 – AFDM des variables quantitatives

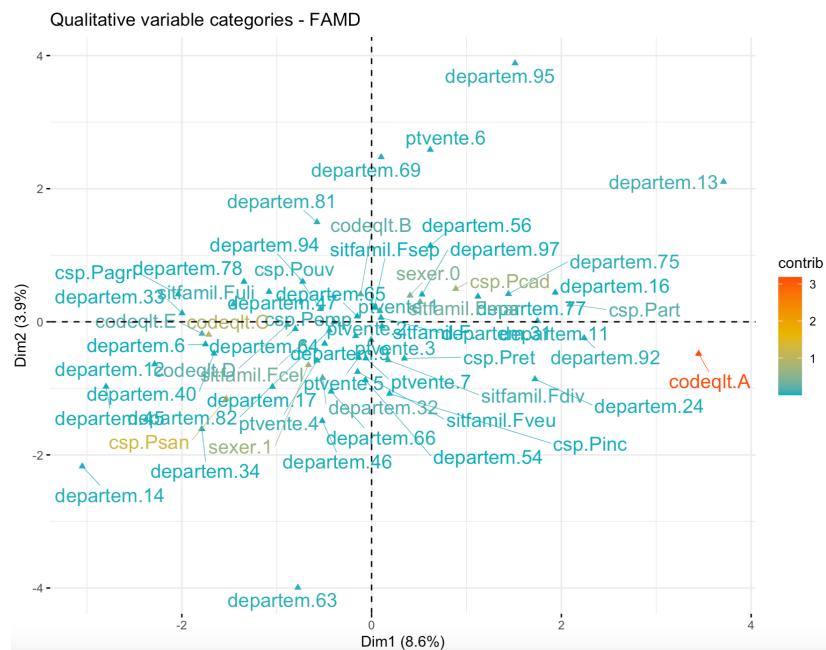


FIGURE 4 – AFDM des variables qualitatives

4 Fraude bancaire

4.1 Description des données

Ce data set contient les transactions effectuées par cartes de crédit en septembre 2013 par les titulaires de carte européennes. Cet ensemble de données présente les transactions qui se sont produites en deux jours, où nous avons 492 fraudes sur 284 807 transactions. L'ensemble de données est très déséquilibré, les classes positives (fraudes) représentent 0,172% de toutes les transactions. Il contient uniquement des variables d'entrée numériques résultant d'une transformation ACP. Les caractéristiques V1, V2, ... V28 sont les composantes principales obtenues avec l'ACP, les seules variables qui n'ont pas été transformées avec l'ACP sont Time et Amount. La variable Time contient le nombre de secondes écoulées entre chaque transaction et la première transaction de l'ensemble de données. La variable Amount représente le Montant de la transaction.

Table	#d'observations	#de variables	#de classes	Ratio
Fraud-carte-crédit	284 807	31	2	0.00172

TABLE 4 – Description des données Fraudes Bancaire

Ce data set ne contient aucune donnée manquante. De plus, les variables ont été déjà mises à l'échelle (avec une ACP normée). Cependant, il y a environ 1800 transactions dont le montant est égal à zéro. Ces transactions sont réparties de manière équiprobable entre les deux classes, nous avons donc décidé de les garder pour notre analyse, car aucune information n'a été donnée sur le fait qu'il s'agit d'une erreur ou non. Nous remarquons que ce jeu de données est fortement déséquilibré, seulement 0.172% des observations appartiennent à la classe positive.

4.2 Étude exploratoire de données



FIGURE 5 – Histogramme de la variable Amount

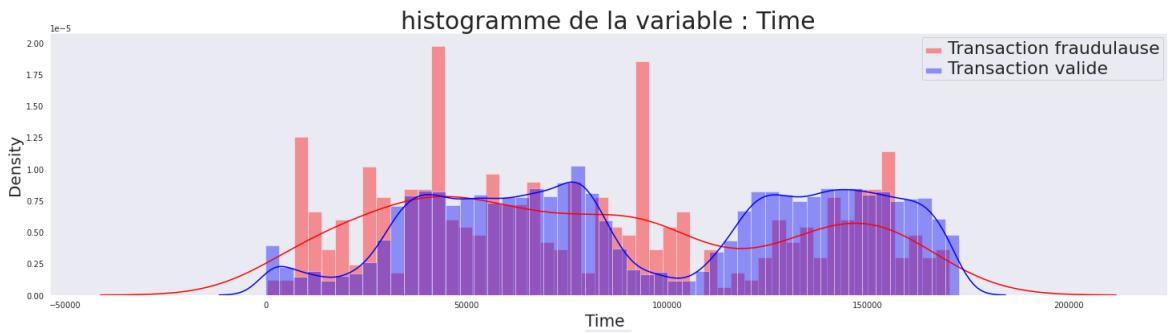


FIGURE 6 – Histogramme de la variable Time

La figure 5 montre que le montant des transactions des deux classes suit presque la même distribution. De même, aucune conclusion ne peut être tirée en regardant la distribution de la variable Time dans chaque classe (cf. figure 6).

La matrice de corrélation ne montre aucune corrélation entre les variables V1, ... V28 (les variables ont été décorrélées avec une ACP). Cependant, nous nous remarquons que les variables V1, V3, V5, V6, V7, V10, V12, V14, V16, V17 et V18 sont corrélées négativement avec la variable à expliquer class, tandis que V2, V4 et V11 sont corrélées positivement avec celle-ci.

On peut facilement remarquer, dans la figure 8, que le nombre de transactions valides augmente dans la journée, et atteint son maximum entre 8 h et 21 h, puis continue à baisser durant la nuit. Le nombre de transactions frauduleuses est quasi constant tout au long de la journée avec des pics vers 2 h, 11 h et 18 h.

La figure 9 montre que la plupart des transactions frauduleuses ont des montants compris entre 0 et 1000\$.

5 Stratégies d'échantillonnage

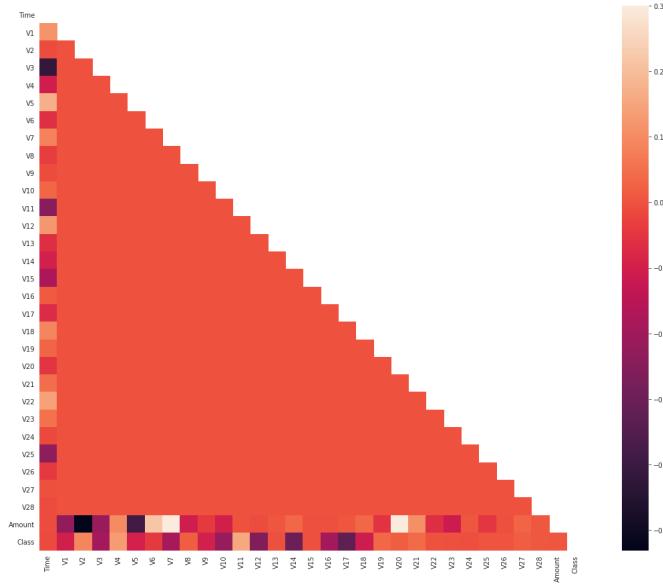


FIGURE 7 – Matrice de corrélation

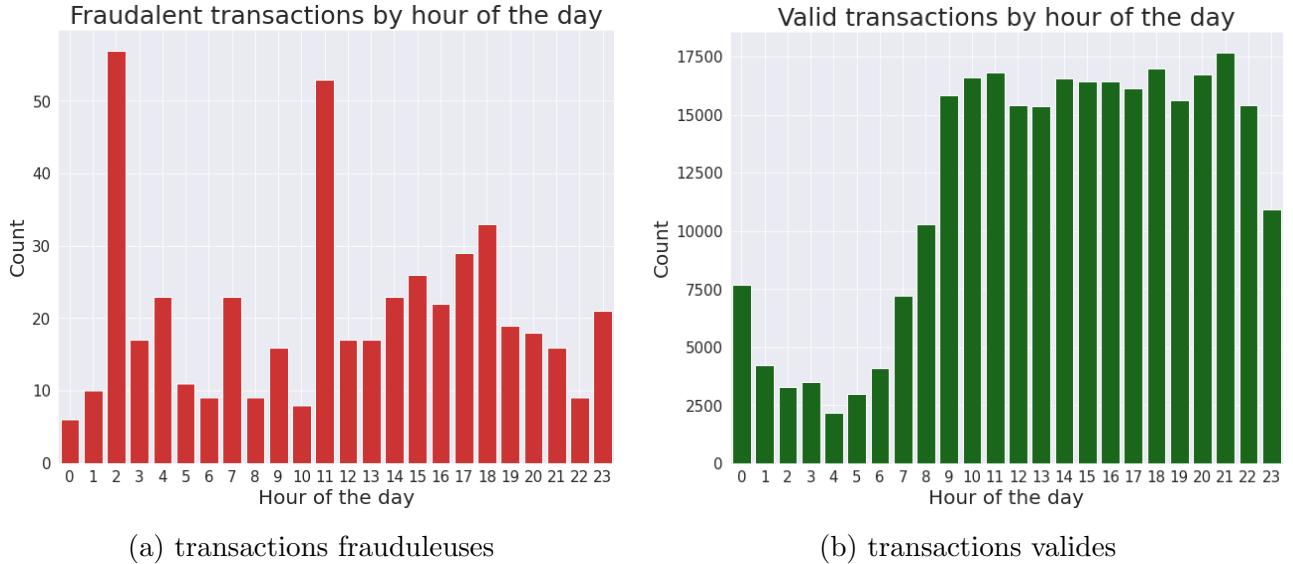


FIGURE 8 – Distribution des transactions au cours de la journée

5 Stratégies d'échantillonnage

L'échantillonnage est une technique qui consiste tout simplement à rééquilibrer le jeu de données. Soit en faisant de l'**undersampling**, en enlevant des données de la classe majoritaire, soit en faisant de l'**oversampling**, en rajoutant des nouvelles données dans la classe minoritaire.

Dans le cadre de ce travail, nous avons préconisé trois méthodes, l'undersampling aléatoire

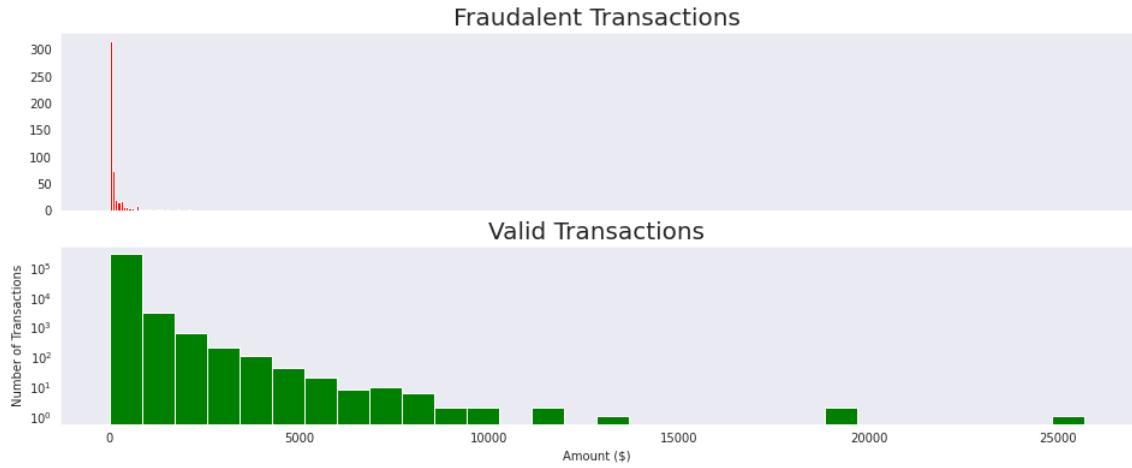


FIGURE 9 – Etude de corrélation

(RandomUnderSampler), NearMiss et SMOTE.

5.1 SMOTE

SMOTE est une méthode de sur-échantillonnage se concentre sur la classe minoritaire, qui est augmentée en créant des exemples artificiels. C'est l'un des algorithmes les plus utilisés pour améliorer la performance de classificateurs appliqués sur les données disproportionnées. L'algorithme fournit un ensemble de règles simples pour générer de nouvelles données. Bien que chaque nouvelle donnée synthétique soit construite à partir de ses parents (la donnée choisie et l'un de ses voisins les plus proches), la donnée générée n'est jamais un double exact de l'un de ses parents.

5.2 Random UnderSampler

Le sous-échantillonnage aléatoire (Random UnderSampler) consiste à tirer au hasard des échantillons de la classe majoritaire, avec ou sans remplacement. Toutefois, elle peut accroître la variance du classifieur et peut éventuellement éliminer des échantillons utiles ou importants.

5.3 NearMiss

Cette méthode d'undersampling, consiste à sélectionner des échantillons, tout en prenant en compte la distance entre les exemples de la classe majoritaire et ceux de la classe minoritaire.

6 Résultats et discussion

6.1 Données relationnelles

On remarque qu'en utilisant la matrice \mathbf{M} , $\mathbf{W} + \mathbf{I}$ on ait des résultats bien meilleurs que la matrice \mathbf{X} . Celà peut être expliqué par la perte de l'information (i.e la connectivité des noeuds entre eux), vu qu'en utilisant que la matrice \mathbf{X} des caractéristiques (features) on ignore la propagation de l'information qui est représentée par la matrice d'adjacence \mathbf{W} .

Par contre, on observe une petite différence entre les résultats obtenus avec la matrice \mathbf{M} et $\mathbf{W}+\mathbf{I}$, parce que dans la deuxième matrice $\mathbf{W}+\mathbf{I}$ on ajoute une information qui pourrait être pertinente, à savoir : l'information du noeud lui-même représenté par la matrice identité \mathbf{I} (self loops). Par conséquent, l'accuracy a augmenté de 3%.

Les résultats illustrés par le tableau 5 démontrent la performance de l'utilisation de la matrice $W+I$ dans la classification des données. Nous remarquons aussi, que les approches ensemblistes (Extra Trees, Gradient Boosting), ainsi que le SVM linéaire, Logistic Regression et le Perceptron Multicouche ont les meilleurs résultats. Il est aussi intéressant de noter que l'algorithme QDA est moins performant que les autres algorithmes.

Algorithmes	Matrice X	Matrice M	Matrice W+I	Normalisation symétrique
KNeighbors Classifier	0.42	0.82	0.84	0.80
Linear SVM	0.72	0.83	0.87	0.87
RBF SVM	0.34	0.37	0.38	0.38
Gaussian Process	0.40	0.84	0.87	0.86
Decision Tree	0.52	0.59	0.61	0.60
Random Forest	0.34	0.38	0.38	0.36
Neural Net	0.75	0.84	0.87	0.87
AdaBoost	0.60	0.67	0.56	0.60
Naive Bayes	0.52	0.72	0.81	0.80
QDA	0.12	0.25	0.27	0.27
Gradient Boosting	0.75	0.80	0.85	0.84
Logistic Regression	0.75	0.84	0.87	0.86
Extra Trees	0.78	0.83	0.88	0.88
LDA	0.52	0.68	0.70	0.64

TABLE 5 – l'accuracy moyenne pour les 4 méthodes sur les données Cora

Nous remarquant dans les résultats illustrés par le tableau 6 que les approches ensemblistes donnent une précision (accuracy) de 75% un peu prêt sauf Décision Trees qui à peine parvenue à 60% de précision.

Étant donné les résultats du QDA et RBF SVM, on peut dire que les données sont linéairement séparables et cela peut être vu dans les résultats du SVM Linéaire.

6 Résultats et discussion

Algorithmes	Matrice X	Matrice M	Matrice W+I	Normalisation symétrique
KNeighbors Classifier	0.10	0.66	0.72	0.66
Linear SVM	0.73	0.73	0.78	0.78
RBF SVM	0.21	0.30	0.29	0.29
Gaussian Process	0.59	0.34	0.21	0.21
Decision Tree	0.53	0.58	0.59	0.60
Random Forest	0.22	0.33	0.38	0.30
Neural Net	0.71	0.72	0.76	0.76
AdaBoost	0.62	0.63	0.66	0.67
Naive Bayes	0.62	0.67	0.73	0.72
QDA	0.19	0.44	0.31	0.43
Gradient Boosting	0.71	0.71	0.75	0.75
Logistic Regression	0.72	0.72	0.76	0.75
Extra Trees	0.75	0.73	0.77	0.76
LDA	0.39	0.53	0.54	0.51

TABLE 6 – l'accuracy moyenne pour les 4 méthodes sur les données CiteSeer

6.2 Carte Visa

Les modèles, que nous avons choisis, ont d'abord effectué leur tâche de classification sur le jeu de données. Ils ont été ensuite évalués. Nous avons choisi 2 métriques d'évaluation : l'accuracy moyenne avec une k-fold cross-validation ($k = 10$) et le score AUC.

Le tableau ci-dessous montre les résultats de notre expérimentation sur les données des fraudes bancaires (Fraud-carte-credit). Il illustre les résultats atteints par les 10 modèles, avec en gras les meilleurs résultats.

Méthode	SMOTE	Random	NearMiss
KNeighbors Classifier	0.88 ± 0.06	0.74 ± 0.06	0.85 ± 0.07
Gradient Boosting	0.93 ± 0.07	0.90 ± 0.13	0.90 ± 0.14
Logistic Regression	0.85 ± 0.05	0.83 ± 0.05	0.90 ± 0.06
Random Forest Classifier	0.93 ± 0.08	0.89 ± 0.05	0.90 ± 0.16
Extra Trees Classifier	0.93 ± 0.05	0.88 ± 0.05	0.92 ± 0.07
Linear Discriminant Analysis	0.83 ± 0.06	0.80 ± 0.08	0.87 ± 0.06
Quadratic Discriminant Analysis	0.82 ± 0.1	0.84 ± 0.08	0.84 ± 0.15
Decision Tree Classifier	0.86 ± 0.09	0.85 ± 0.10	0.84 ± 0.14
Support Vector Machine	0.87 ± 0.03	0.85 ± 0.04	0.92 ± 0.03
AdaBoost Classifier	0.92 ± 0.05	0.90 ± 0.09	0.92 ± 0.13

TABLE 7 – l'accuracy moyenne et l'écart-type pour les 3 méthodes (avec une k-fold cross-validation)

Les résultats illustrés par le tableau 7 démontrent la performance de la méthode d'échantillonnage

6 Résultats et discussion

SMOTE dans la classification de données, qui donne les meilleurs résultats ici. Nous remarquons aussi, que les approches ensemblistes (Random Forest, Extra Trees, Gradient Boosting, et AdaBoost) ont les meilleures performances.

Il est aussi intéressant de noter que l'algorithme QDA est moins performant que l'algorithme LDA si on applique les méthodes SMOTE ou NearMiss, et plus performant dans le cas de Random UnderSampler. Nous remarquons aussi, que l'application de la méthode NearMiss sur les modèles SVM et Logistic Regression a un impact remarquable sur les performances. L'utilisation de cette technique permet d'améliorer considérablement le taux d'accuracy.

Les méthodes SMOTE et NearMiss surpassent encore la méthode aléatoire d'undersampling, en prenant en compte le score AUC comme métrique d'évaluation (cf. table 8). Cependant, les meilleurs algorithmes ne sont pas les mêmes, ici, l' algorithme AdaBoost est le meilleur alors que c'était Extra-Trees lorsque nous comparons en utilisant l'accuracy moyenne.

Méthode	SMOTE	Random	NearMiss
KNeighbors Classifier	0.88	0.74	0.85
Gradient Boosting	0.97	0.93	0.96
Logistic Regression	0.91	0.90	0.97
Random Forest Classifier	0.96	0.94	0.94
Extra Trees Classifier	0.97	0.94	0.97
Linear Discriminant Analysis	0.89	0.88	0.94
Quadratic Discriminant Analysis	0.92	0.91	0.92
Decision Tree Classifier	0.86	0.84	0.82
Support Vector Machine	0.92	0.90	0.96
AdaBoost Classifier	0.97	0.95	0.98

TABLE 8 – Score AUC sur les données Carte Visa

6 Résultats et discussion

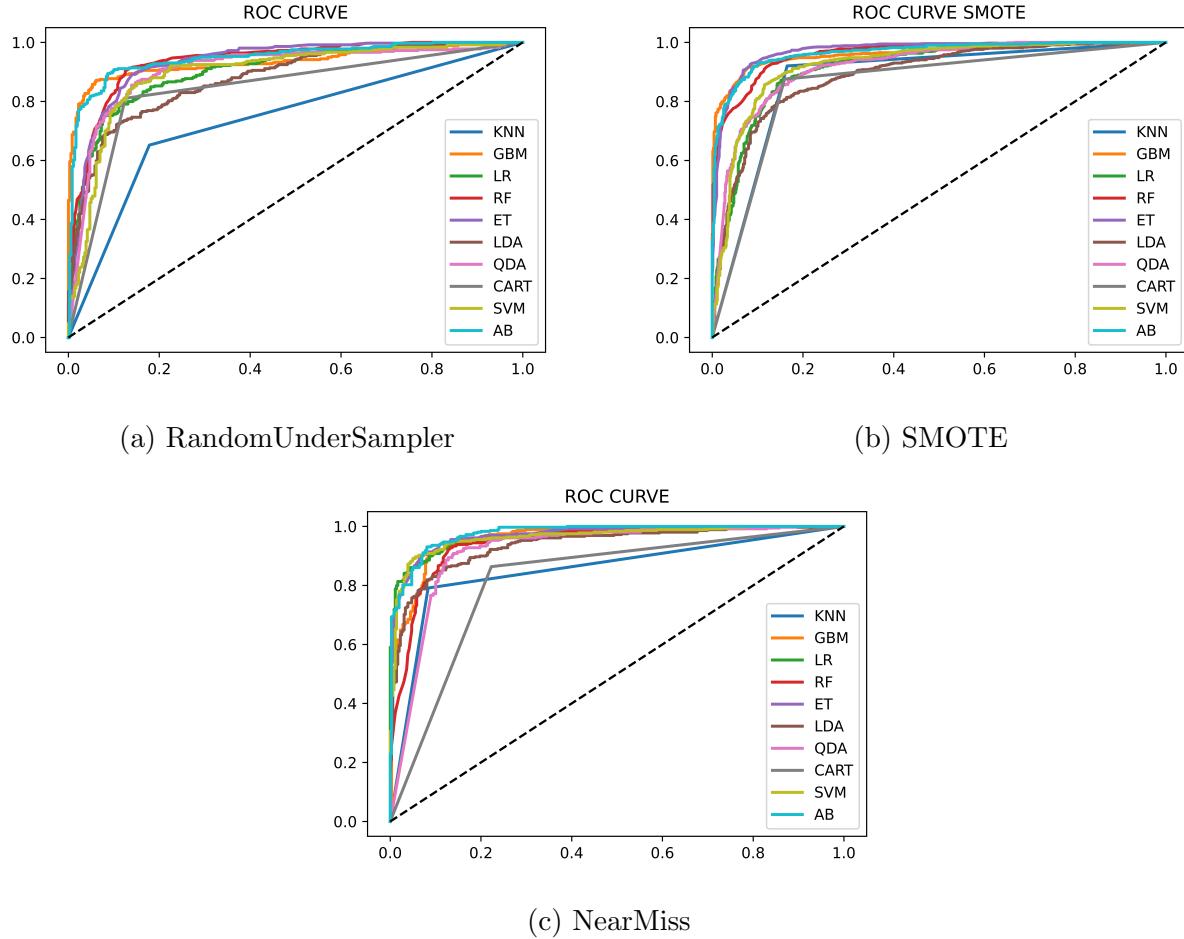


FIGURE 10 – Les courbes ROC

6.3 Fraude bancaire

Les modèles que nous avons choisis, ont d'abord effectué leur tâche de classification sur le jeu de données. Ils ont été ensuite évalués. Nous avons pris les mêmes métriques que dans la section précédente. Cependant, vu la taille des données, nous n'avons pas pu comparer ces algorithmes avec une technique d'oversampling comme SMOTE par exemple.

Les résultats illustrés par le tableau 9 démontrent la performance de la méthode NearMiss dans la classification de données, qui donne les meilleurs résultats ici. Nous remarquons, comme la section précédente, que les approches ensemblistes (Random Forest, Extra Trees, Gradient Boosting, et AdaBoost) ont les meilleurs résultats. Il est aussi intéressant de noter que l'algorithme QDA est plus performant que l'algorithme LDA avec les deux méthodes d'échantillonnage.

Nous remarquons aussi, que l'application de l'undersampling sur les modèles SVM et Logistic Regression a un impact remarquable sur les performances. L'utilisation de ces techniques permet à ces modèles d'atteindre les performances des modèles ensemblistes.

6 Résultats et discussion

Méthode	Random	NearMiss
KNeighbors Classifier	0.90 ± 0.11	0.90 ± 0.11
Gradient Boosting	0.94 ± 0.05	0.94 ± 0.04
Logistic Regression	0.94 ± 0.05	0.94 ± 0.04
Random Forest Classifier	0.94 ± 0.05	0.95 ± 0.04
Extra Trees Classifier	0.94 ± 0.05	0.95 ± 0.05
Linear Discriminant Analysis	0.88 ± 0.10	0.89 ± 0.10
Quadratic Discriminant Analysis	0.93 ± 0.03	0.93 ± 0.05
Decision Tree Classifier	0.92 ± 0.03	0.93 ± 0.03
Support Vector Machine	0.94 ± 0.05	0.94 ± 0.04
AdaBoost Classifier	0.94 ± 0.05	0.94 ± 0.06

TABLE 9 – Accuracy moyenne

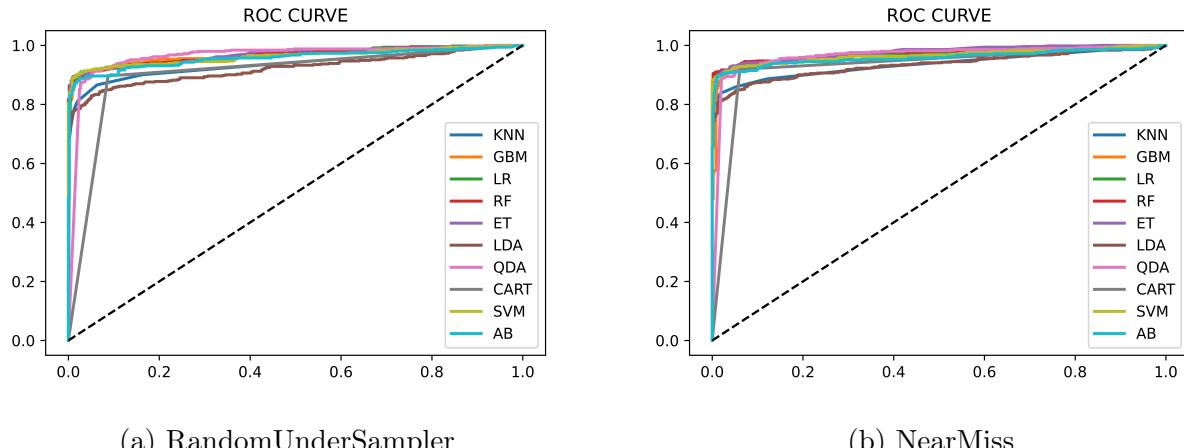


FIGURE 11 – Les courbes ROC

Les scores AUC donnent des résultats bien meilleurs pour NearMiss que lorsque l'on compare avec l'accuracy moyenne. Le même commentaire peut être fait avec Random, il parvient à avoir les mêmes performances que la méthode NearMiss (comme LDA, QDA ou AdaBoost).

Méthode	Random	NearMiss
KNeighbors Classifier	0.93	0.94
Gradient Boosting	0.97	0.97
Logistic Regression	0.96	0.97
Random Forest Classifier	0.97	0.98
Extra Trees Classifier	0.97	0.98
Linear Discriminant Analysis	0.93	0.93
Quadratic Discriminant Analysis	0.96	0.96
Decision Tree Classifier	0.91	0.93
Support Vector Machine	0.96	0.97
AdaBoost Classifier	0.97	0.96

TABLE 10 – Score AUC

7 Conclusion

Le projet consiste à mettre en pratique concrète d'un certain nombre de techniques d'apprentissage supervisé sur plusieurs datasets, à savoir relationnelles et non relationnelles et comparer entre elles.

Tout au long ce processus on a confronté plusieurs obstacles dont la disproportionnalité des données bancaires qui peut être critique dans le cas de détection de fraude qui est une classe minoritaire avec une grande importance. Pour y remédier, on a utilisé deux stratégies d'échantillonnage, soit en faisant de l'undersampling, en enlevant des données de la classe majoritaire, soit en faisant de l'oversampling, en rajoutant des nouvelles données dans la classe minoritaire.

Le deuxième obstacle, c'était au niveau des données relationnelles qui étaient caractérisés non seulement par une matrice de features \mathbf{X} mais aussi avec une matrice d'adjacence qui représente les liens et l'interaction entre les noeuds du graphe. Pour mieux exploiter ces données ainsi que leurs structures (i.e. graphe) on a proposé 4 combinaisons pour aligner les deux types d'informations cachées, afin de mieux mener notre tâche de classification.

On conclura que les approches ensemblistes (c.à.d AdaBoost, Random Forest, Extra Tree, Gradient Boosting) surpassent les autres méthodes d'apprentissage. Toutes fois SVM et le Perceptron Multicouche ont aussi montré leur efficacité dans les tâches de classification supervisée.

Réaliser ce travail nous a permis d'apprendre à créer et à optimiser des algorithmes d'apprentissage supervisé ayant pour dessein une prédiction catégorielle, et ce à partir de données déséquilibrées. Nous avons alors pu explorer différentes possibilités d'amélioration des performances par des stratégies d'échantillonnage, nous permettant ainsi de générer/retirer artificiellement des données.