Rapport de Projet

Détection d'évènements rares en deep learning

- Ayale HADDAD
- Hacene ISSELNANE

Sommaire

1.Introduction

L'apprentissage profond fait partie d'une famille plus large de méthodes d'apprentissage machine basées sur des réseaux neuronaux artificiels. L'apprentissage peut être supervisé, semi-supervisé ou non supervisé.

Les architectures d'apprentissage profond telles que les réseaux neuronaux profonds, ont été appliquées à des domaines tels que la vision par ordinateur, la vision artificielle, la reconnaissance vocale, le traitement du langage naturel, la reconnaissance audio, le filtrage des réseaux sociaux, la traduction automatique, la bio-informatique, la conception de médicaments, l'analyse d'images médicales, l'inspection des matériaux et les programmes de jeux de société, où elles ont produit des résultats comparables et, dans certains cas, supérieurs aux performances des experts humains.

La problématique est de pouvoir réaliser une classification sur des jeux de données déséquilibrés en utilisant différentes techniques de rééchantillonnage.

Dans cet exercice, étant face à des problématique de détection d'évènement rare, les modèles seront particulièrement évalués dans leur taux de bonne classification de ces évènements. En effet, il est plus dommageable de prédire un non-évènement (faux négatifs) que son inverse, exemple dans le cadre d'une détection de fraude, de maladie ou autre phénomène de même nature, une fausse prédiction aurait un impact déterminant pour l'utilisateur/client final.

2.Rééquilibrage de jeux d'apprentissage

Un ensemble de données de classification avec des proportions de classe asymétriques est appelé « déséquilibrer ». Les classes qui constituent une grande partie de l'ensemble de données sont appelées classes majoritaires. Les classes qui représentent une plus petite proportion sont les classes minoritaires.

Degré de déséquilibre	Proportion de la classe minoritaire
Léger	20-40%
Modéré	1 à 20 %
Extrême	<1%

Tableau 1 - Degré de déséquilibre d'un jeu d'apprentissage

Afin de remédier à ce problème, plusieurs techniques de rééchantillonnage sont proposées.

2.1. Pondération des classes

Cette méthode consiste à donner des poids différents aux classes majoritaires et minoritaires. La différence de poids influencera la classification des classes pendant la phase d'entraînement. L'objectif est de pénaliser la classification erronée de la classe minoritaire en fixant un poids de classe plus élevé et en réduisant en même temps le poids de la classe majoritaire.

2.2. Oversampling (sur-échantillonnage)

Le suréchantillonnage augmente le poids de la classe minoritaire en reproduisant les exemples de la classe minoritaire. Bien qu'il n'augmente pas l'information, il soulève le problème du surapprentissage, qui rend le modèle trop spécifique et moins précis face aux données non vues lors de l'apprentissage. Il existe différentes approches d'oversampling. Parmi elles, on utilisera :

2.2.1. Random oversampling

Le suréchantillonnage aléatoire ne fait que reproduire de façon aléatoire les exemples de la classe minoritaire. Cette méthode est connue pour augmenter la probabilité de sur-apprentissage.

2.2.2. Synthetic Minority Oversampling Technique (SMOTE)

Cette méthode génère des données synthétiques basées sur les similitudes d'espace de caractéristiques entre les instances minoritaires existantes. Afin de créer une instance synthétique, elle trouve les K voisins les plus proches de chaque instance minoritaire, sélectionne aléatoirement l'un d'entre eux, puis calcule des interpolations linéaires pour produire une nouvelle instance minoritaire dans le voisinage.

2.2.3. Adaptive Synthetic (ADASYN)

ADASYN génère des échantillons de la classe minoritaire en fonction de leur distribution de densité. Des données plus synthétiques sont générées pour les échantillons de la classe minoritaire qui sont plus difficiles à apprendre, par rapport aux échantillons de la classe minoritaire qui sont plus faciles à apprendre. Cet algorithme calcule les K plus proches voisins de chaque instance minoritaire, puis obtient le rapport de classe des instances minoritaires et majoritaires pour générer de nouveaux échantillons. En répétant ce processus, il déplace de manière adaptative la frontière de décision pour se concentrer sur les échantillons difficiles à apprendre.

2.3. Undersampling (sous-échantillonnage)

Contrairement au suréchantillonnage, cette technique permet d'équilibrer l'ensemble des données en réduisant la taille de la classe majoritaire. Son principal inconvenant est la perte d'information potentiellement contenu dans les échantillons mis à l'écart.

2.3.1. Random undersampling

Cette méthode consiste à prélever des échantillons au hasard dans la classe majoritaire, avec ou sans remplacement. C'est l'une des premières techniques utilisées pour réduire le déséquilibre de l'ensemble de données.

2.3.2. Tomek Links

Les liens Tomek suppriment les chevauchements indésirables entre les classes lorsque les liens de la classe majoritaire sont supprimés jusqu'à ce que toutes les paires de voisins les plus proches à distance minimale soient de la même classe. Un lien Tomek est défini comme

suit : étant donné une paire d'instances (x_i, x_j), où $x_i \in S_{min}, x_j \in S_{max}$ et $d(x_i, x_j)$ est la distance entre x_i et x_j, alors la pair (x_i, x_j) est appelé un lien Tomek s'il n'y a pas d'instance x_k tel que $d(x_i, x_k) < d(x_i, x_j)$ ou $d(x_j, x_k) < d(x_i, x_j)$.

Si deux instances forment un lien Tomek, l'une d'entre elle est soit un bruit ou les deux se trouvent près d'une frontière. Ainsi, on peut utiliser les liens Tomek pour nettoyer le chevauchement entre les classes.

En supprimant les exemples qui se chevauchent, on peut établir des groupes bien définis dans l'ensemble de formation et améliorer les performances de la classification.

Cependant, cette méthode s'avère très couteuse en temps de calcul, et est parfois inutilisable sur de très grands échantillons de données.

3.Modèles deep learning

Afin de répondre à la problématique de ce projet, nous avons proposés deux approches de modèle pour tester l'efficacité des méthodes de rééquilibrage des données présentées ci-dessus.

3.1. Modèle *baseline*

Ce modèle est un simple réseau séquentiel constitué d'une seule couche contenant 16 nœuds avec un drop-out fixé à 0.5 afin de réduire au mieux le sur-apprentissage, *Relu* est utilisé comme fonction d'activation.

Layer (type)	Output Shape
dense_2 (Dense)	(None, 16)
dropout_1 (Dropout)	(None, 16)
dense_3 (Dense)	(None, 1)

Figure 1- Modèle baseline

3.2. Modèle élaboré

Ce second modèle plus « complexe » est constitué de 5 couches contenant chacune un nombre de nœud compris entre 128 et 16 avec un drop-out fixé à 0.2, *Relu* comme fonction d'activation pour les *hidden layers*.

Layer (type)	Output Shape
dense_22 (Dense)	(None, 128)
dropout_13 (Dropout)	(None, 128)
dense_23 (Dense)	(None, 64)
dropout_14 (Dropout)	(None, 64)
dense_24 (Dense)	(None, 32)
dropout_15 (Dropout)	(None, 32)
dense_25 (Dense)	(None, 32)
dropout_16 (Dropout)	(None, 32)
dense_26 (Dense)	(None, 16)
dropout_17 (Dropout)	(None, 16)
dense_27 (Dense)	(None, 1)

Figure 2 - Modèle élaboré

4. Jeu de donnée « *Credit fraud* »

4.1. Présentation du jeu de données

Ces données contiennent les transactions effectuées par carte de crédit en septembre 2013 par les détenteurs de carte européennes.

Cet ensemble de données présente les transactions qui ont eu lieu en deux jours, où nous avons 492 fraudes sur 284 807 transactions. L'ensemble de données est donc très déséquilibré.

Il ne contient que des variables d'entrée numériques qui sont le résultat d'une transformation PCA.

4.2. Méthodologie

Pour chaque méthode de rééquilibrage utilisée, nous présenterons les principales métriques de performance ainsi que les courbe de ROC associées suivi d'une conclusion.

4.3. Analyse exploratoire

Les variables d'entrée sont : Vi (28 composantes principales de l'ACP), le temps et le montant. La classe est la sortie étiquetée.

La classe positive (les fraudes) ne représente que 0,172 % de toutes les transactions.

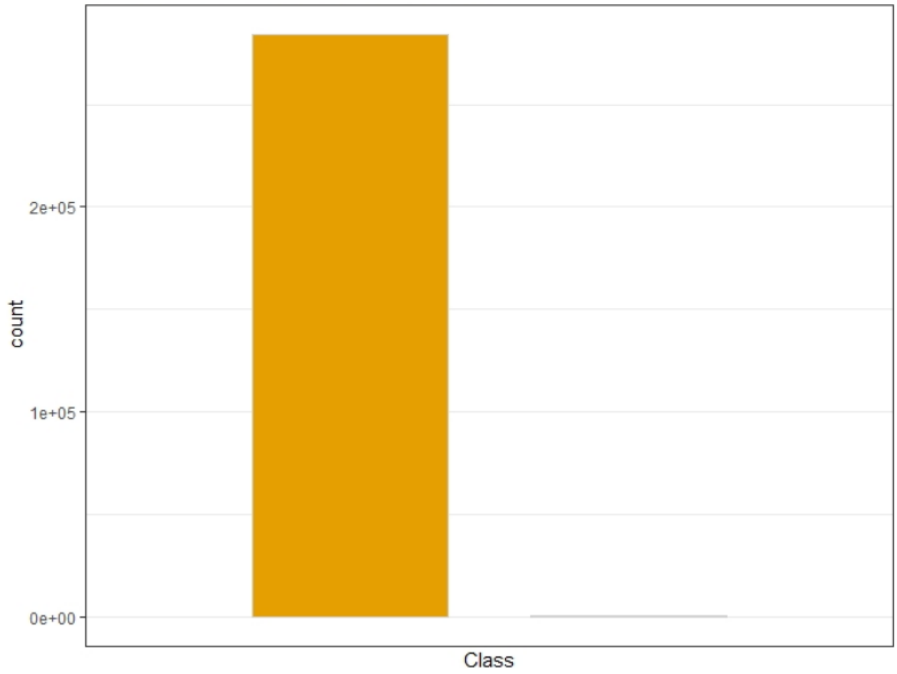


Figure 3 - distribution de la variable cible Classe

Selon l'aperçu de l'ensemble de données, la seule information donnée concernant la variable temporelle est le délai, car elle regroupe les transactions qui ont eu lieu en deux jours. Nous ne savons pas si la période de 48 heures est arbitraire ou non. En raison du manque de contexte, nous pouvons également supposer que l'enregistrement de donnée dont nous disposons commence à minuit.

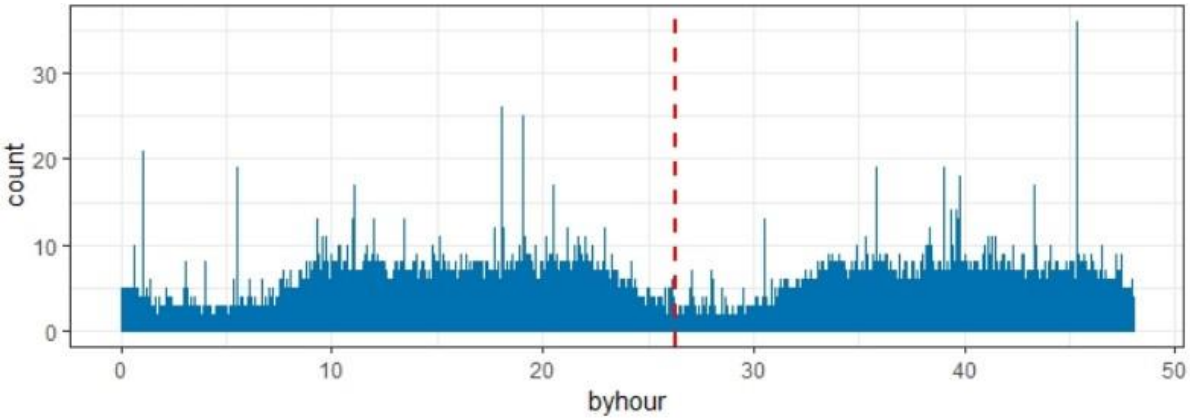


Figure 4 - répartition des transactions suivant le temps

La variable Montant n'est pas normalisée.

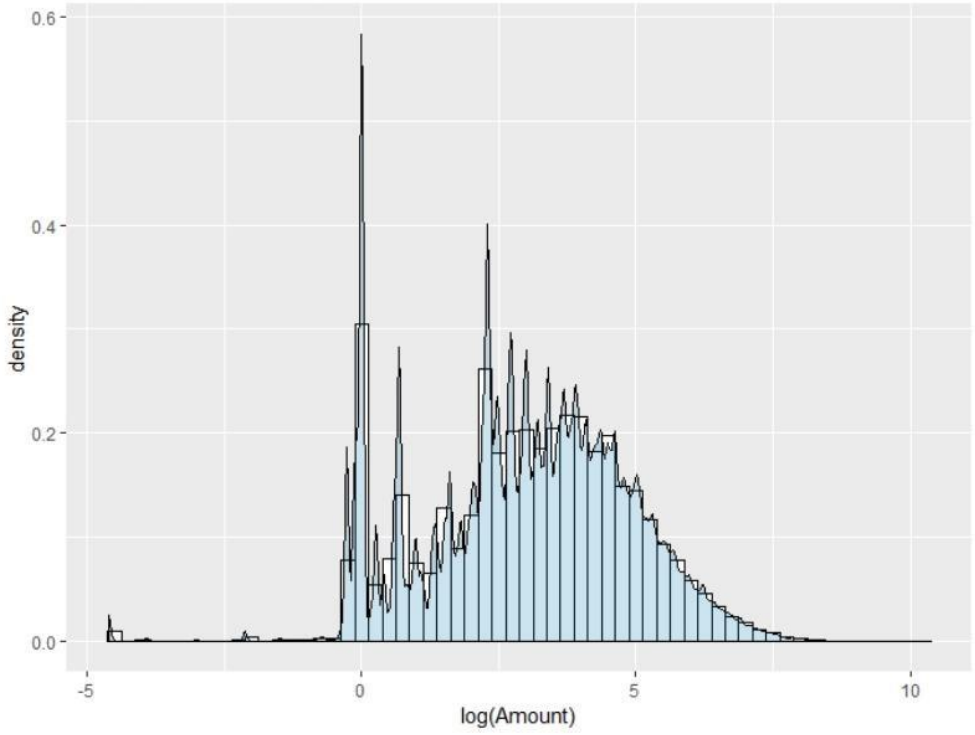


Figure 5 - densité de distribution de la variable Amount

En traçant la distribution des variables Vi, il semble qu'elles aient été normalisées.

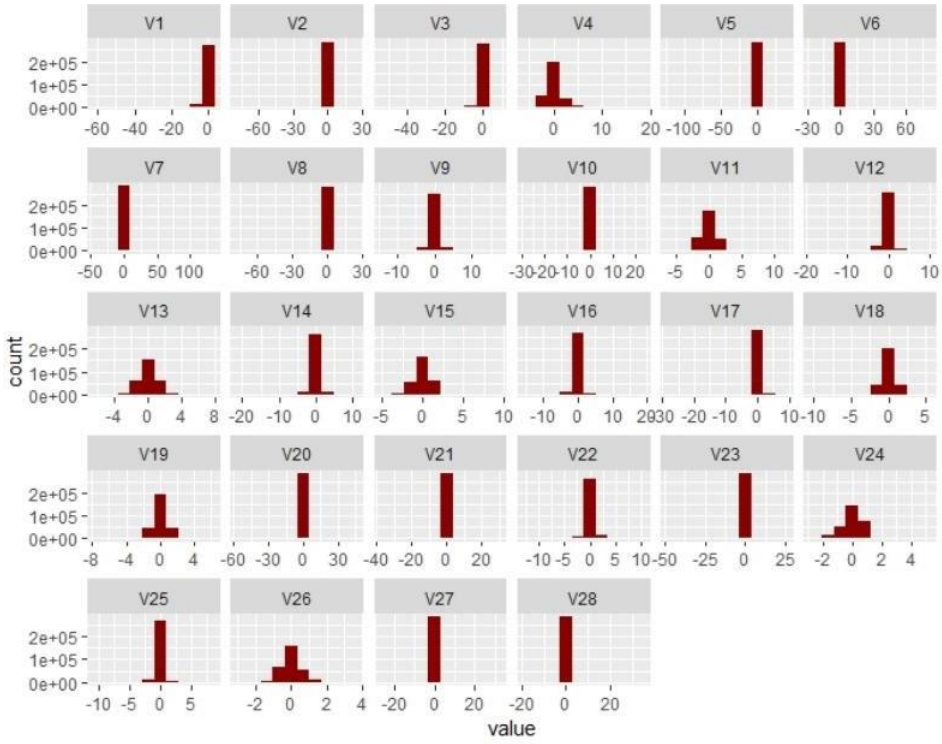


Figure 6 - Distribution des variables Vi

Comme les principales variables Vi sont les résultats d'une ACP, nous ne nous attendons pas à ce qu'elles soient corrélées et nous vérifions donc si cette hypothèse est confirmée.

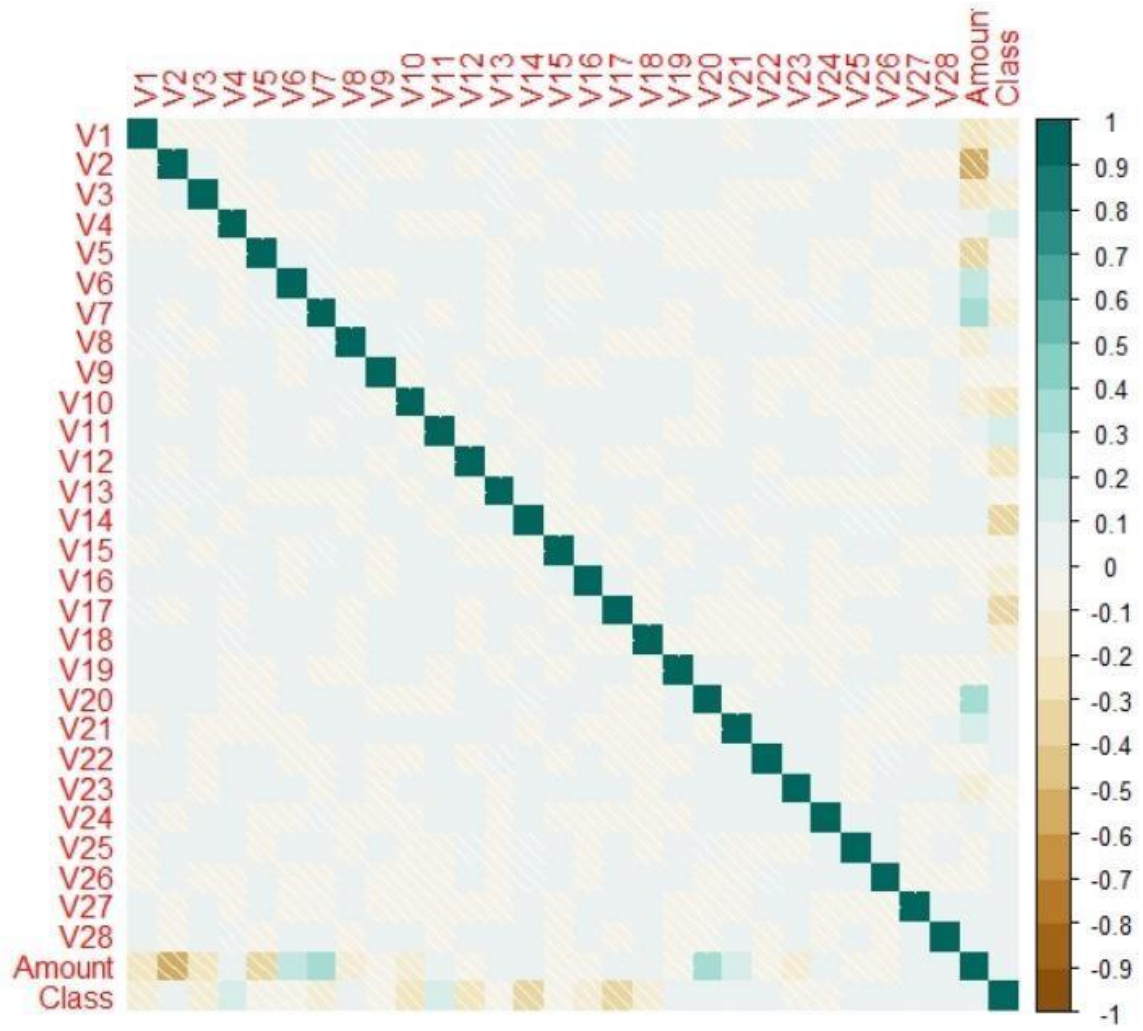


Figure 7 - Corrélogramme du jeu de données Credit Fraud

Ce corrélogramme montre effectivement que la plupart des données Vi ne sont pas corrélées.

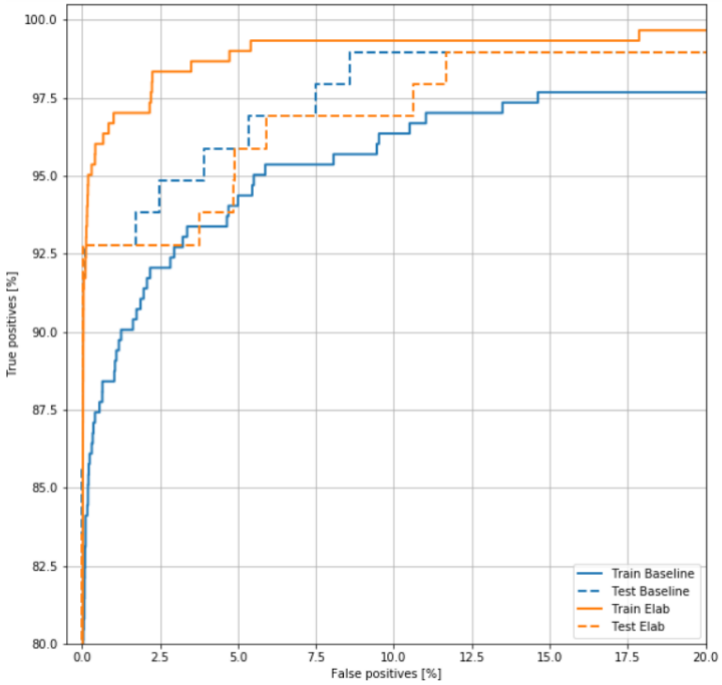
4.3.1. Bilan de l'analyse exploratoire

Cette analyse exploratoire des données a mis en évidence la nature déséquilibrée de la distribution de la classe cible et la nécessité de préparer davantage l'ensemble de données de formation, en le rééquilibrant pour la modélisation et en normalisant la variable *Amount*. Aucune information supplémentaire n'étant fournie pour la variable temporelle, celle-ci ne sera pas utilisée pour le reste du projet.

4.4. Entraînement simple

	Modèle Baseline	Modèle élaboré
Accuracy	0.99	0.99
Precision	0.92	0.86
Recall	0.87	0.90
AUC	0.96	0.96

4.4.1. Bilan



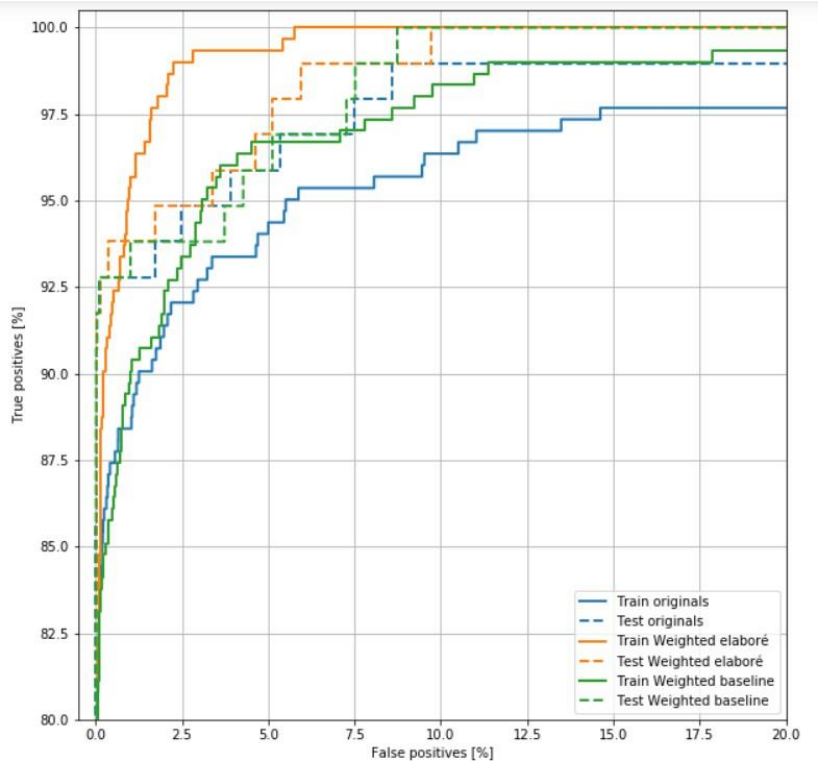
Sur ce deux premiers modèles, aucune méthode de rééquilibrage des données n'est utilisée. Ces modèles nous serviront de référence par la suite pour pouvoir établir des comparaisons.

A noter que le modèle élaboré performe légèrement mieux en termes de *recall*.

4.5. Entraînement avec pondération des classes

	Modèle Baseline	Modèle élaboré
Accuracy	0.97	0.96
Precision	0.07	0.04
Recall	0.93	0.94
AUC	0.99	0.99

4.5.1. Bilan



On constate que les modèles utilisant une pondération de classes obtiennent de meilleurs taux de *recall* comparativement au modèle original, cependant, leur précision chute grandement.

4.6. Entraînement avec oversampling

4.6.1. Random oversampling

	Modèle Baseline	Modèle élaboré
Accuracy	0.98	0.97
Precision	0.09	0.05
Recall	0.88	0.90
AUC	0.98	0.98

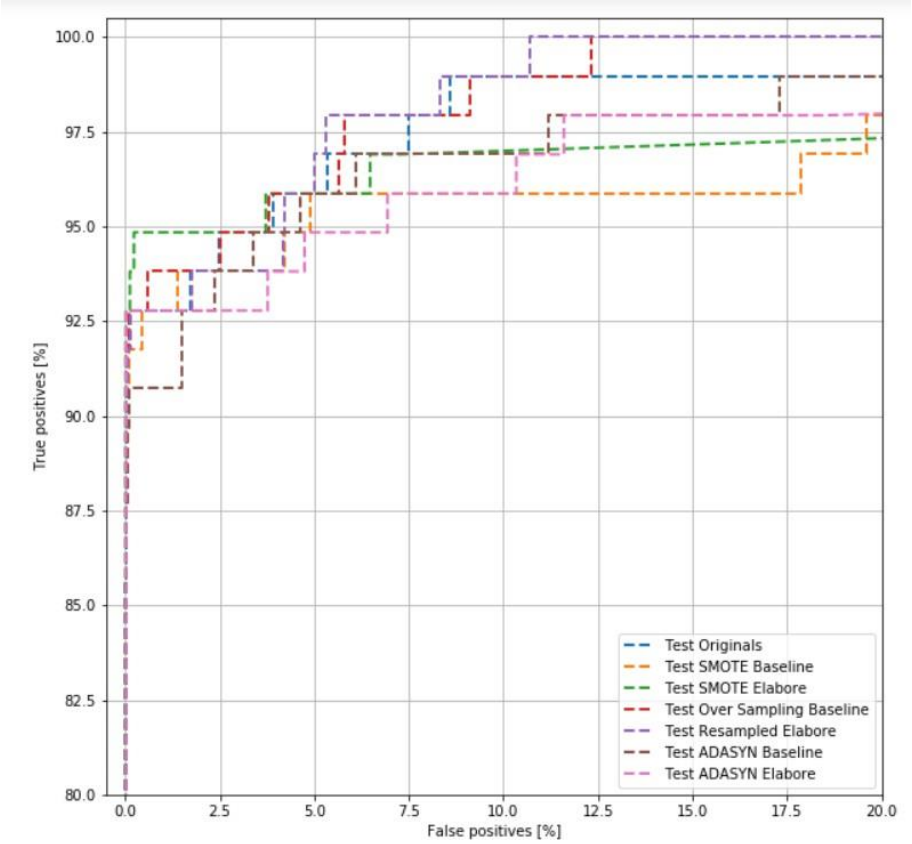
4.6.2. SMOTE

	Modèle Baseline	Modèle élaboré
Accuracy	0.99	0.99
Precision	0.16	0.82
Recall	0.92	0.92
AUC	0.96	0.96

4.6.3. ADASYN

	Modèle Baseline	Modèle élaboré
Accuracy	0.99	0.99
Precision	0.14	0.82
Recall	0.90	0.92
AUC	0.95	0.96

4.6.4. Bilan



On constate que les méthodes ADASYN et SMOTE (avec modele élaboré) obtiennent les meilleures performances comparativement avec le modèle original en termes de *recall* et *précision*.

4.7. Entraînement avec undersampling

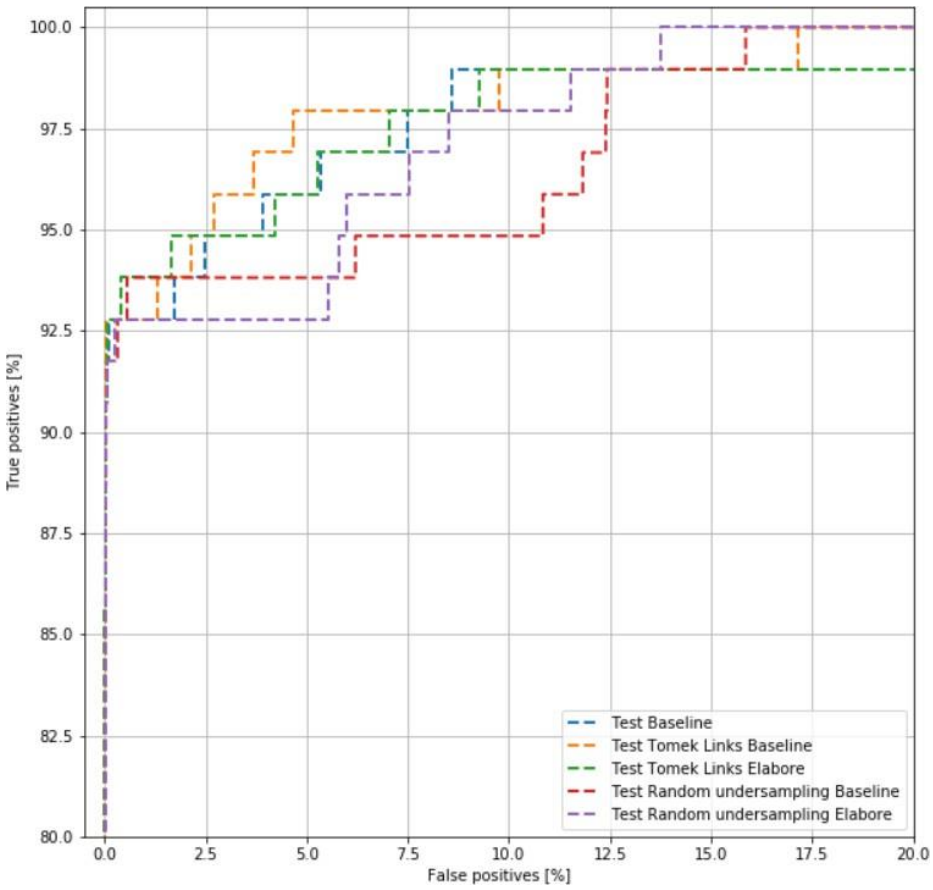
4.7.1. Random undersampling

	Modèle Baseline	Modèle élaboré
Accuracy	0.99	0.95
Precision	0.14	0.03
Recall	0.90	0.92
AUC	0.92	0.99

4.7.2. Tomek Links

	Modèle Baseline	Modèle élaboré
Accuracy	0.99	0.99
Precision	0.97	0.90
Recall	0.79	0.88
AUC	0.96	0.96

4.7.3. Bilan



On constate que Random undersampling et Tomek Link (avec modèle élaboré) obtiennent relativement (dans cette famille de méthode) les meilleures performances en termes de recall.

Encore une fois, les *précisions* chutent énormément avec les méthodes randomisées.

4.8. Entraînement avec une approche hybride de resampling

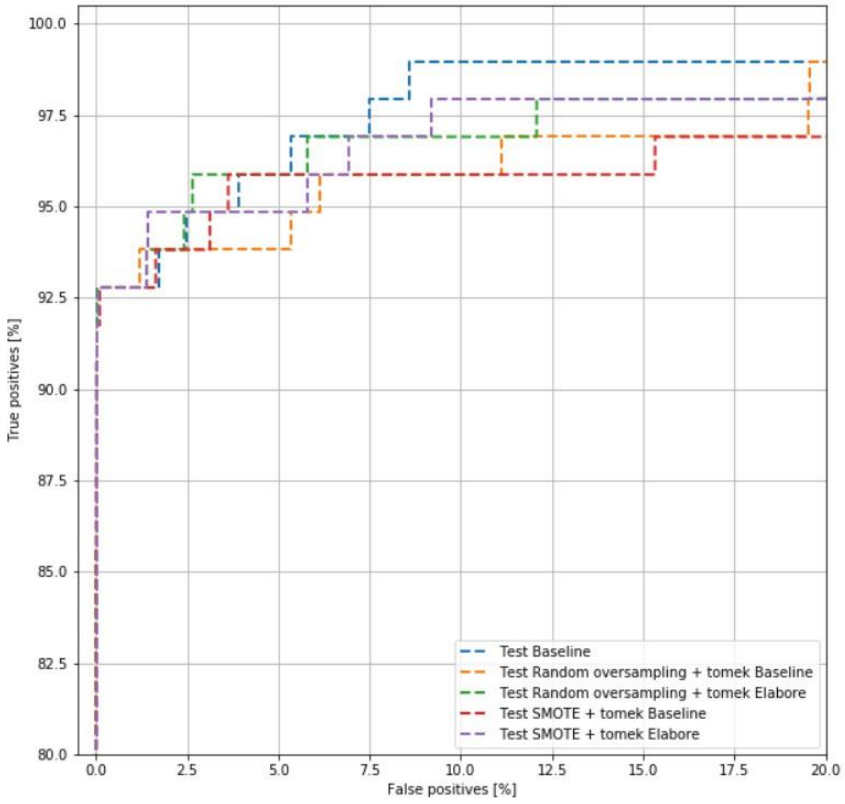
4.8.1. Random oversampling + Tomek links

	Modèle Baseline	Modèle élaboré
Accuracy	0.99	0.99
Precision	0.21	0.79
Recall	0.92	0.92
AUC	0.96	0.96

4.8.2. SMOTE + Tomek Links

	Modèle Baseline	Modèle élaboré
Accuracy	0.99	0.99
Precision	0.22	0.76
Recall	0.92	0.91
AUC	0.96	0.96

4.8.3. Bilan



Les deux méthodes obtiennent des résultats très similaires et sont en deçà du modèle original.

4.9. Conclusion

Sur ce jeu de données on constate que les méthodes de type SMOTE et ADASYN permettent d'obtenir les meilleures performances sur les modele élaborés.

Toutefois, l'approche *Baseline* avec entrainement simple s'avère obtenir de meilleures performances (rapport *précision/recall*) en comparaison avec les autres méthodes de rééquilibrage utilisées.

5. Jeu de données « *Bank marketing* »

5.1. Présentation du jeu de données

Les données sont liées aux campagnes de marketing directes d'une institution bancaire portugaise. Les campagnes de marketing ont été basées sur des appels téléphoniques. Souvent, plusieurs contacts avec le même client ont été nécessaires pour savoir si le produit (dépôt à terme bancaire) serait "oui" ou "non" souscrit.

5.2. Analyse exploratoire

Selon les définitions fournies, l'ensemble de données contient 17 variables avec les informations suivantes :

- Âge : âge du client
- Emploi : type d'emploi
- Marital : état civil
- Éducation : "inconnue", "primaire", "secondaire", "tertiaire"
- Défaut de paiement : "non", "oui", "inconnu"
- Solde : solde annuel moyen, en euros
- Loan : prêt hypothécaire ?
- Prêt : prêt personnel ?

Informations sur le dernier contact pendant la campagne de marketing :

- Contact : type de contact ("inconnu", "téléphone", "cellulaire")
- Jour : jour du dernier contact
- Mois : mois du dernier contact
- Durée : durée du dernier contact, en secondes

Autres attributs :

- Campagne : nombre de contacts avec le client pendant la campagne
- *pjours* : nombre de jours écoulés depuis que le client a été contacté pour une campagne de marketing précédente
- *Précédent* : nombre de contacts avec le client avant la campagne de marketing en cours
- *poutcome* : résultat de la campagne de marketing précédente

Variable cible :

- y : le client a-t-il souscrit à un dépôt à terme ? (binaire : "oui", "non")

Un premier aperçu des données montre ce qui suit :

- 4521 observations et 17 variables
- 7 variables numériques : âge, solde, jour, durée, campagne, pdays, précédent
- 10 variables catégorielles :
- 6 variables catégorielles à valeurs multiples : emploi, situation de famille, éducation, contact, mois, revenu
- 3 variables binaires oui/non : défaut, logement, prêt
- 1 variable cible : y

Il n'y a pas de valeurs manquantes.

Seuls 11 % environ des personnes interrogées dans le cadre de la campagne actuelle ont finalement accepté de verser un acompte. Cela rend l'ensemble de données très déséquilibré et nécessitera l'application de méthodes pour compenser ce déséquilibre.

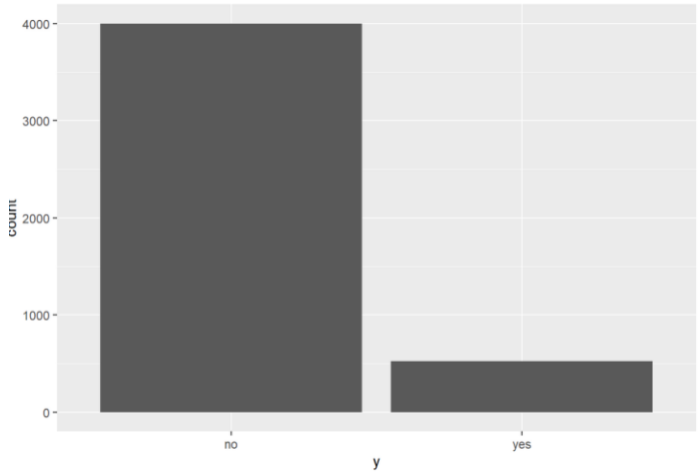


Figure 8 - Répartition de la variable cible Y

Il n'y a pas de corrélations particulièrement significatives entre les variables numériques, sauf, comme prévu, entre les variables previous et pdays qui ont une forte corrélation positive, puisque pdays=-1 correspond à previous=0.

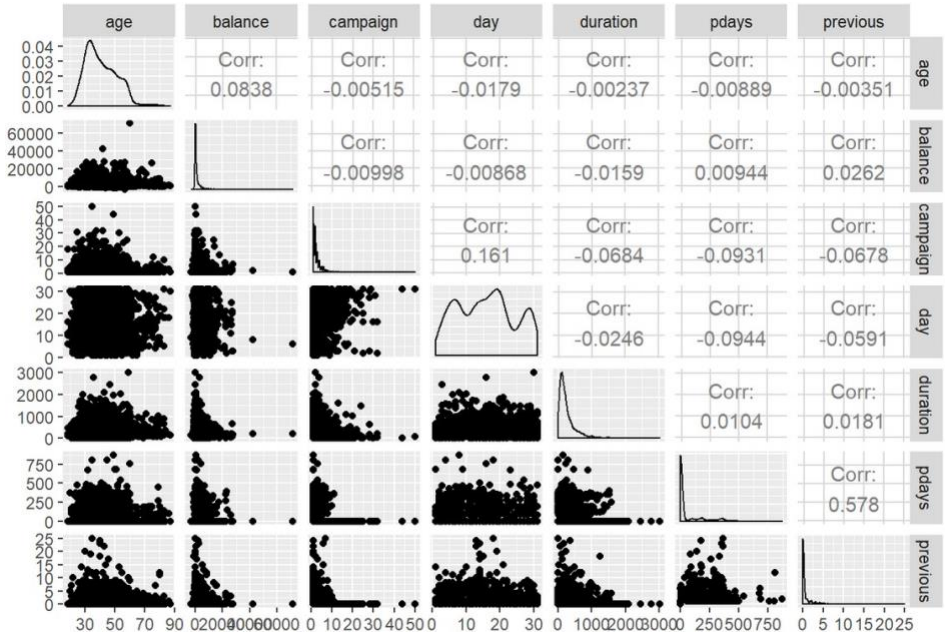


Figure 9 - Corrélogramme des variables numériques

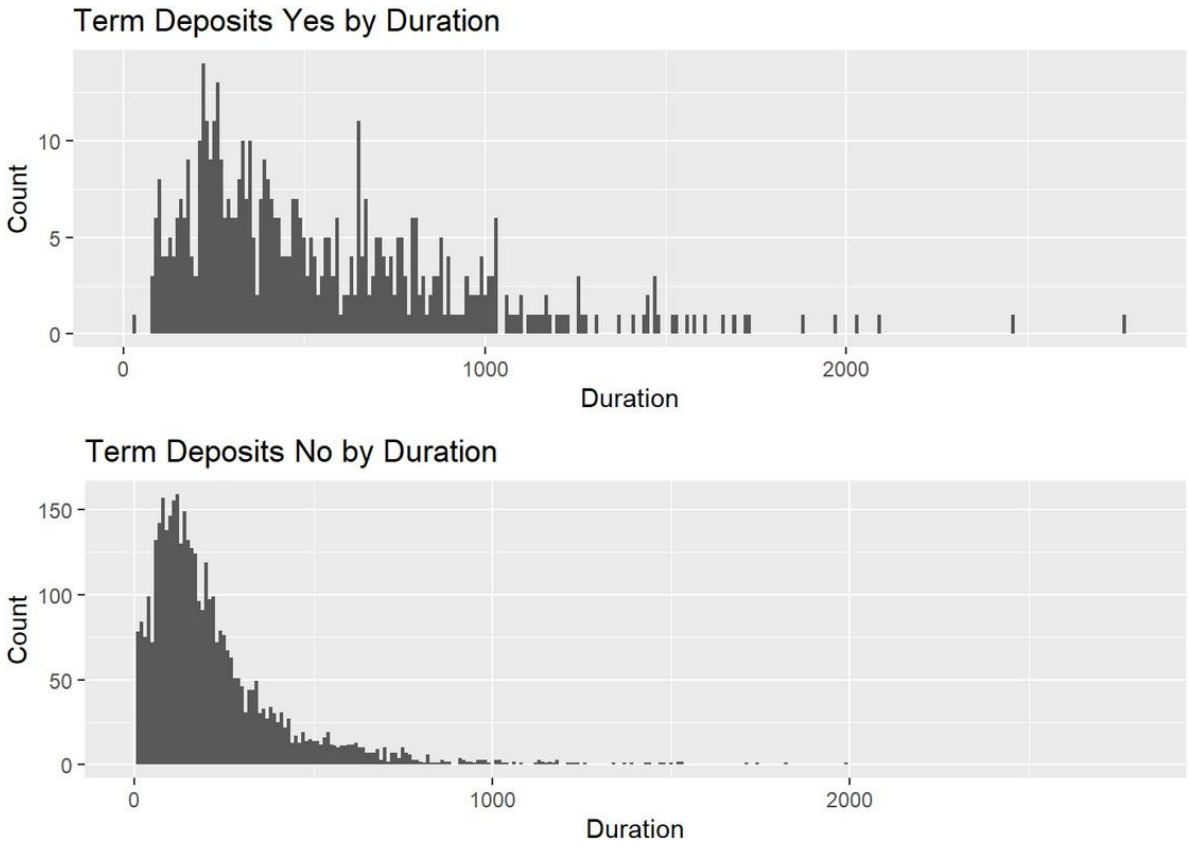


Figure 10 - Distribution de la variable Duration en fonction de la variable cible

La durée a un impact important sur le résultat visé, en ce sens que lorsque la durée est de 0, le résultat de la souscription au dépôt à terme est toujours non. L'autre constatation importante est que presque toutes les personnes qui ne souhaitent pas souscrire à un dépôt à terme décident dans les 5 premières minutes de l'appel, tandis que les personnes qui souhaitent souscrire prennent parfois un peu plus de temps pour se convaincre et décider.

A noter qu'une transformation des données catégorielles en données numérique est effectuée par la suite.

5.3. Bilan de l'analyse exploratoire

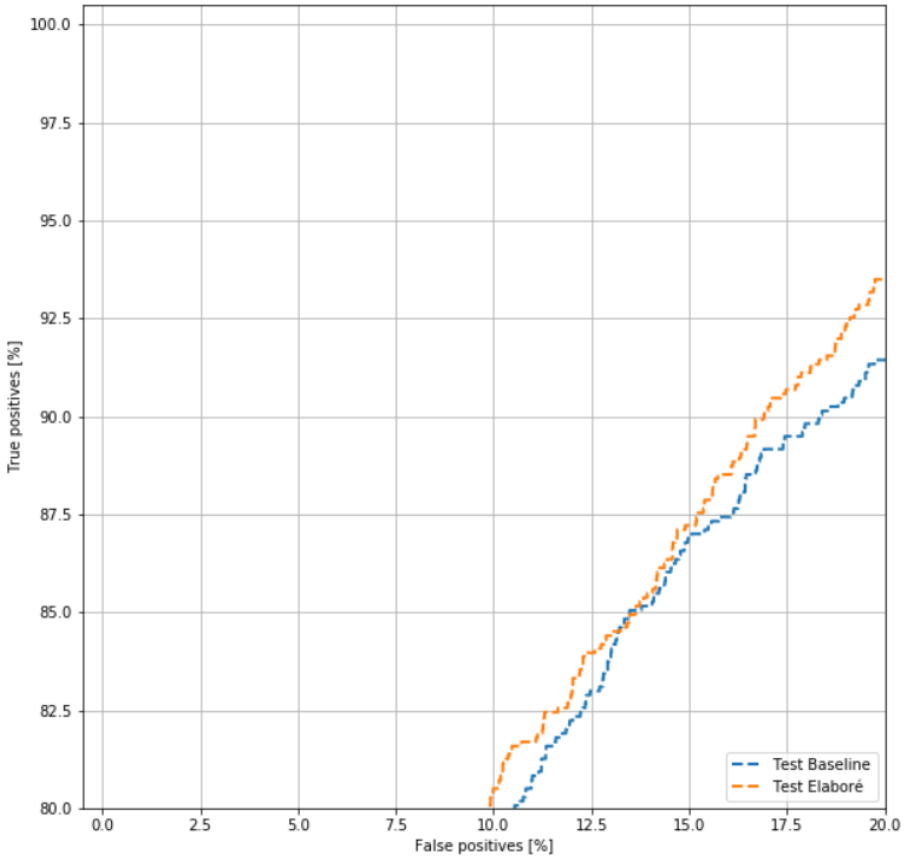
Après avoir analysé le jeu de données, il nous a paru naturel de garder toutes les variables présentes. En effet, toutes sont à notre avis pertinentes pour la prédiction de la variable cible.

La méthodologie est la même que pour le jeu de données précédent.

5.4. Entraînement simple

	Modèle Baseline	Modèle élaboré
Accuracy	0.90	0.90
Precision	0.71	0.56
Recall	0.28	0.57
AUC	0.93	0.92

5.4.1. Bilan

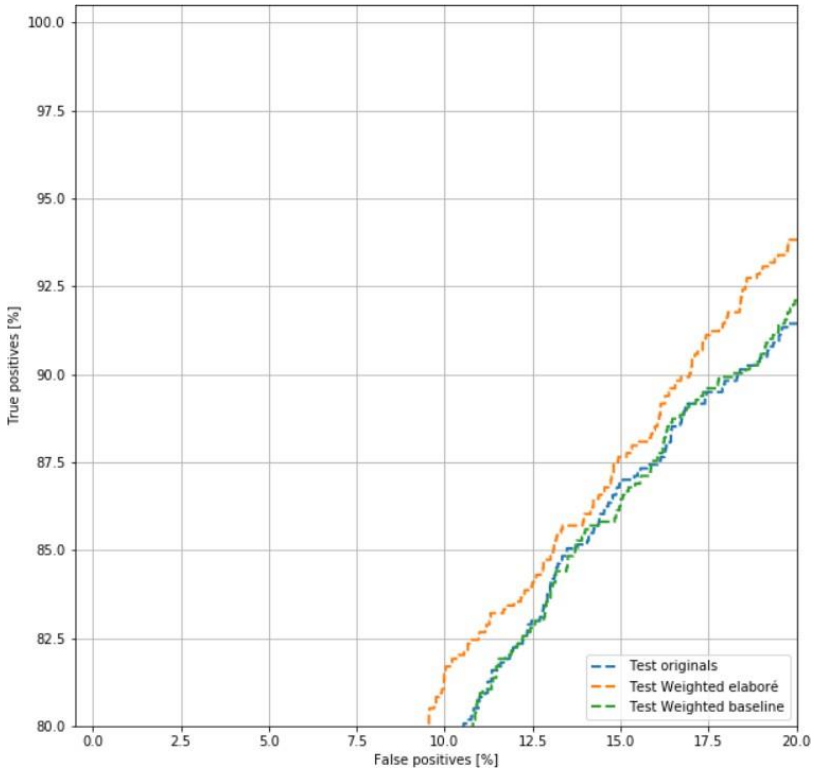


Les deux modèles obtiennent des performances très similaires, voir identique. Le taux de *recall* est très bas, ce qui indique une mauvaise détection des évènements cible.

5.5. Entraînement avec pondération des classes

	Modèle Baseline	Modèle élaboré
Accuracy	0.84	0.81
Precision	0.41	0.36
Recall	0.85	0.92
AUC	0.92	0.93

5.5.1. Bilan



Les taux de *recall* obtenus par les deux modèles *Baseline* et élaboré sont nettement au-dessus de ceux résultants d'un entrainement simple. La pondération des classes agit positivement dans le sens voulu. Cependant, le taux de précision est lui, à l'inverse réduit de près de 30%. Ce qui suggérerai un effet de surapprentissage.

5.6. Entraînement avec oversampling

5.6.1. Random oversampling

	Modèle Baseline	Modèle élaboré
Accuracy	0.84	0.84
Precision	0.40	0.41
Recall	0.87	0.88
AUC	0.92	0.93

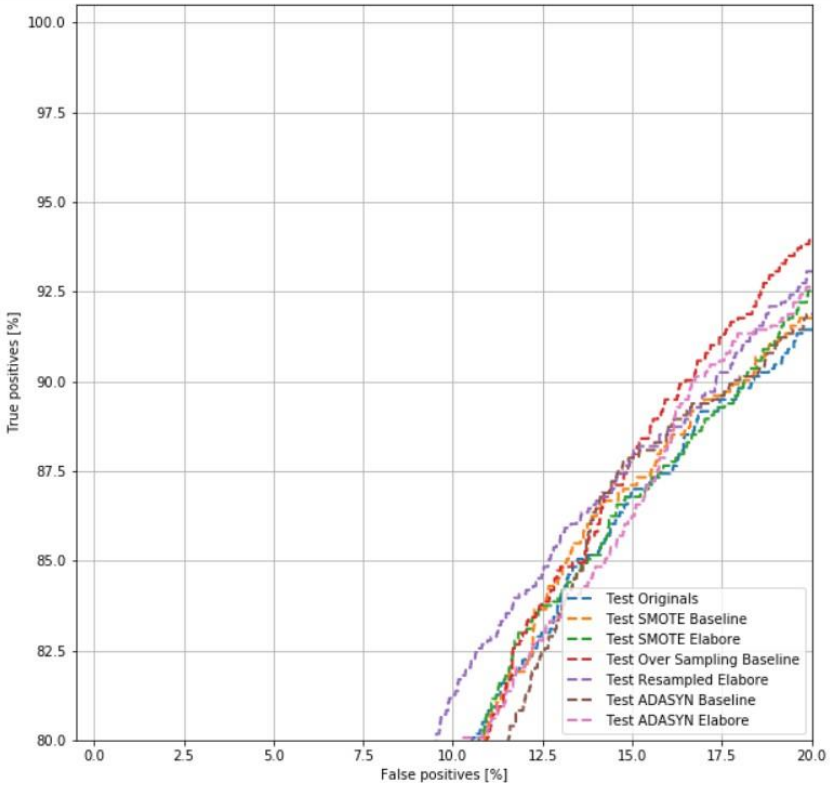
5.6.2. SMOTE

	Modèle Baseline	Modèle élaboré
Accuracy	0.83	0.84
Precision	0.39	0.40
Recall	0.88	0.89
AUC	0.93	0.93

5.6.3. ADASYN

	Modèle Baseline	Modèle élaboré
Accuracy	0.81	0.84
Precision	0.37	0.40
Recall	0.91	0.89
AUC	0.93	0.93

5.6.4. Bilan



Les résultats de ces trois méthodes sont proches, et ressemblent à ceux obtenus par la méthode de pondération des classes, à savoir un taux de *recall* relativement élevé et une précision qui elle est en revanche très inférieure à un entrainement simple.

5.7. Entraînement avec undersampling

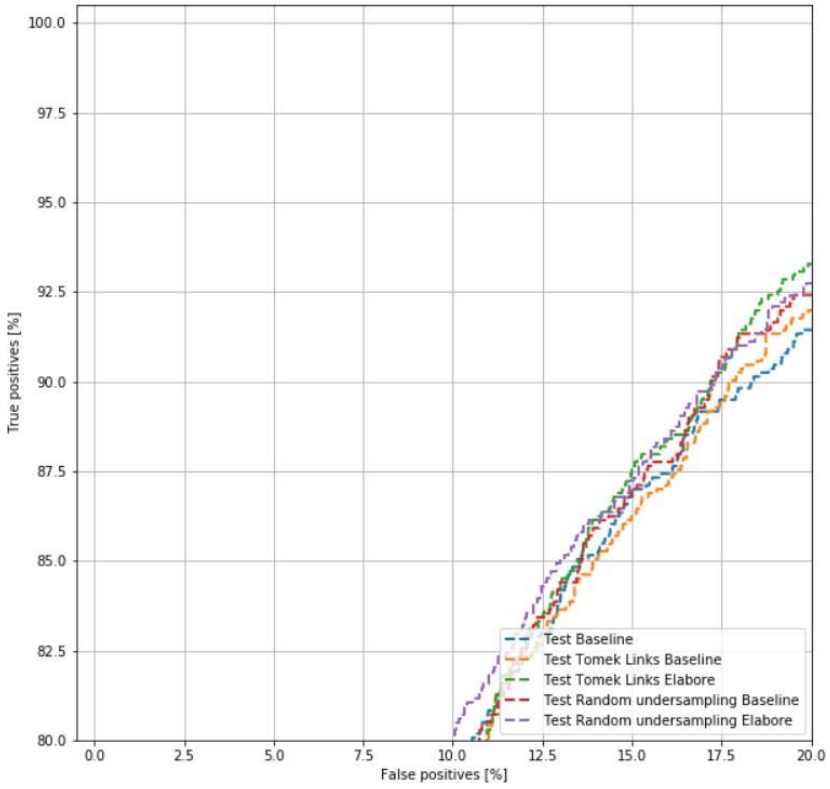
5.7.1. Random undersampling

	Modèle Baseline	Modèle élaboré
Accuracy	0.81	0.81
Precision	0.37	0.36
Recall	0.91	0.90
AUC	0.93	0.92

5.7.2. Tomek Links

	Modèle Baseline	Modèle élaboré
Accuracy	0.90	0.90
Precision	0.64	0.57
Recall	0.40	0.65
AUC	0.92	0.93

5.7.3. Bilan



Le bilan est similaire aux méthodes utilisées précédemment. A savoir, un compromis entre *recall* et précision. Tomek Links et random under sampling obtiennent des résultats presque identiques.

5.8. Entraînement avec une approche hybride de resampling

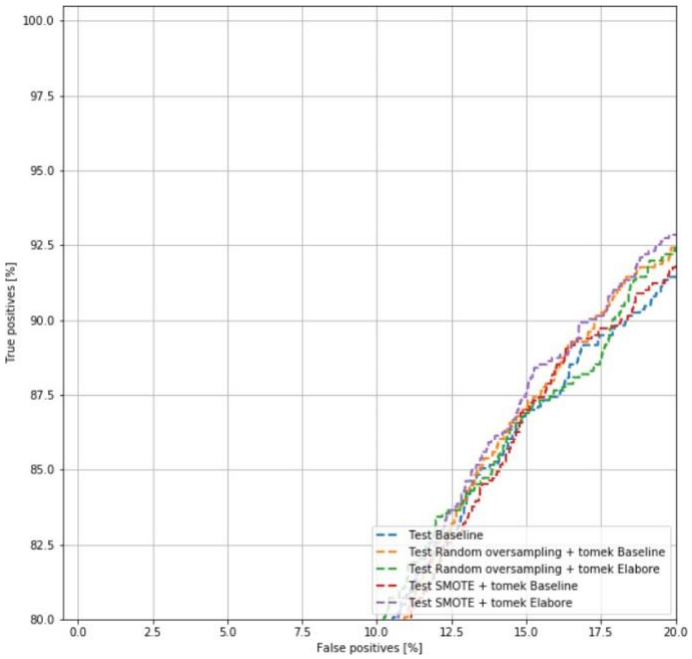
5.8.1. Random oversampling + Tomek links

	Modèle Baseline	Modèle élaboré
Accuracy	0.81	0.83
Precision	0.37	0.40
Recall	0.91	0.89
AUC	0.93	0.93

5.8.2. SMOTE + Tomek Links

	Modèle Baseline	Modèle élaboré
Accuracy	0.83	0.86
Precision	0.38	0.44
Recall	0.87	0.85
AUC	0.92	0.93

5.8.3. Bilan



Dans cette famille de méthodes hybrides, on obtient le meilleur compromis *recall/precision* en combinant *Tomek links* avec *rand. oversampling.*

5.9. Conclusion

Sur ce jeu de données, il n'apparait pas de grande différence de résultats entre les différentes méthodes de rééquilibrage utilisées. Toutes arrivent à avoir de meilleures performances comparativement à un entrainement simple.

6. Jeu de données « *Employee attrition* »

6.1. Présentation du jeu de données

L'échantillon de données compte 1 470 lignes et 35 colonnes (soit 1 470 instances et 35 variables). Les variables comprennent l'âge de chaque salarié, la distance par rapport au domicile, le nombre de voyages d'affaire, le niveau d'éducation, le fait que le salarié ait ou non quitté l'entreprise et plusieurs autres caractéristiques du salarié.

6.2. Analyse exploratoire

L'ensemble de données est bien organisé et ne comporte aucune valeur manquante. La classe cible est déséquilibrée, avec un taux d'attrition de 16 %.

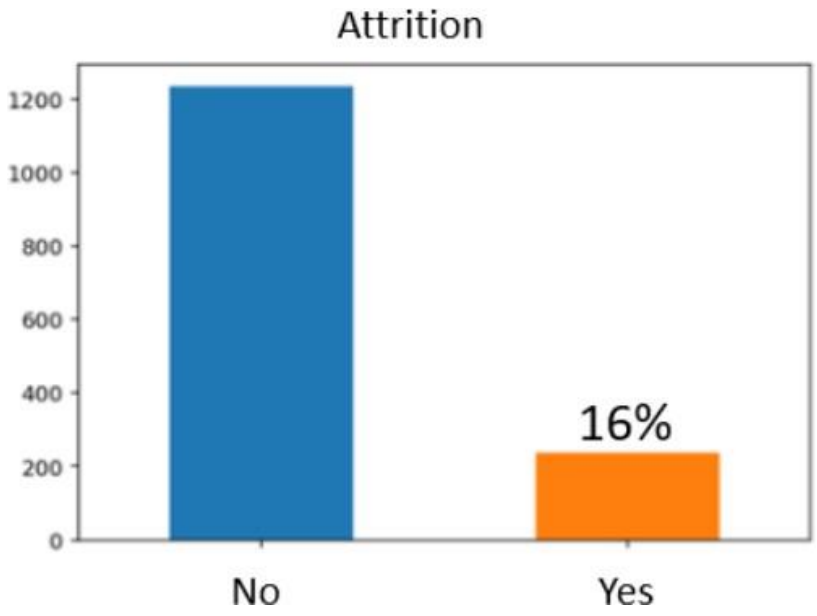


Figure 11 - taux d'attrition

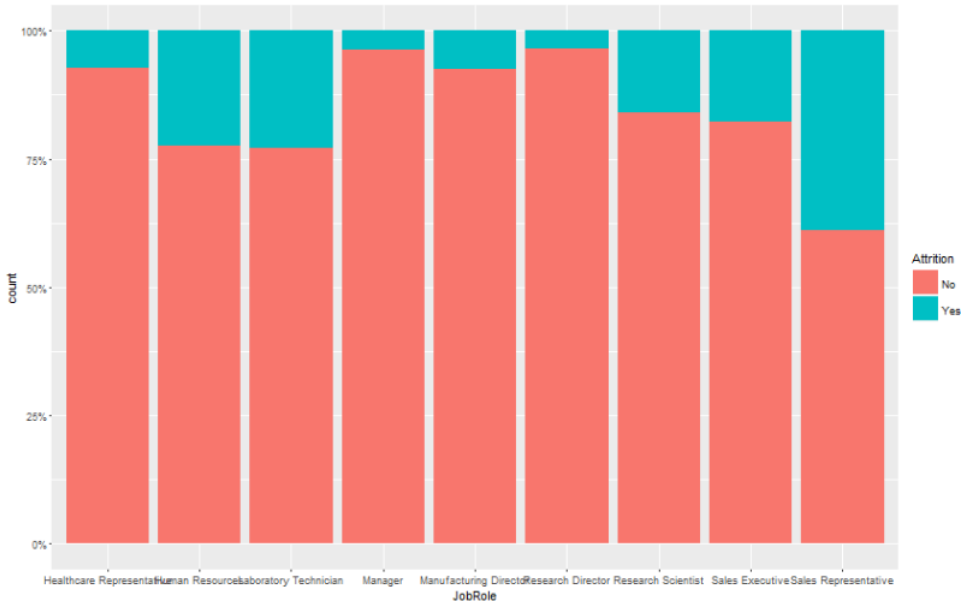


Figure 12 - Attrition des employés en fonction des lignes métiers

Comme on peut le voir ci-dessus le poste de représentant commercial connaît un taux d'attrition nettement supérieur à celui des autres postes.

Les employés sont payés à un taux horaire de 30 à 100 dollars, et l'attrition semble se produire à tous les niveaux, quel que soit le taux horaire de l'employé. Cela peut être confirmé plus tard à la rubrique "Actualité".

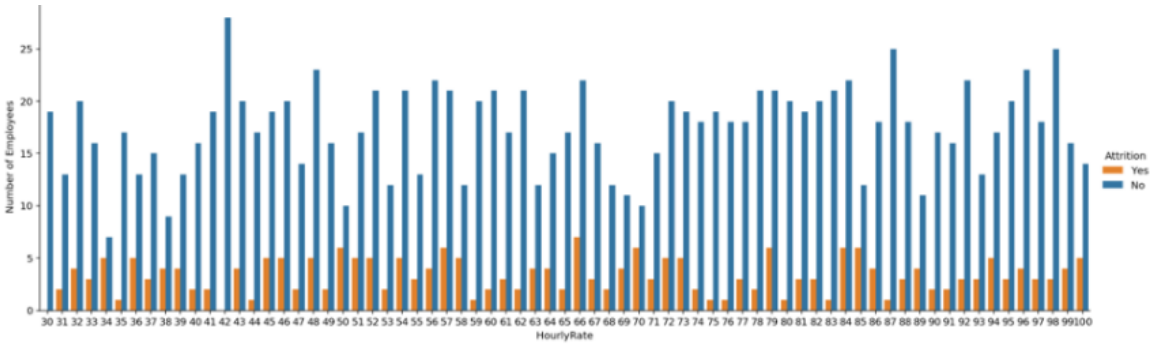


Figure 13 - Attrition suivant le taux horaire

Les heures supplémentaires semblent être l'un des facteurs clés de l'attrition, car une plus grande proportion des employés ayant effectué des heures supplémentaires sont partis.

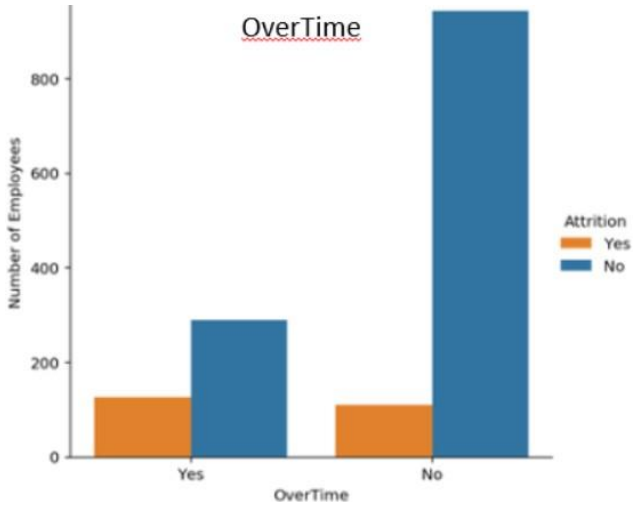


Figure 14 - Attrition en fonction des heures supplémentaires

Des variables telles que le nombre total d'années de travail, le nombre d'années au sein de l'entreprise, le nombre d'années en cours, le nombre d'années depuis la dernière promotion et le nombre d'années avec le gestionnaire actuel sont étroitement liées les unes aux autres. Cela peut éventuellement conduire à une multi-colinéarité dans le contexte de la modélisation de la régression. L'interprétation dépend de la modélisation que l'on souhaite réaliser.

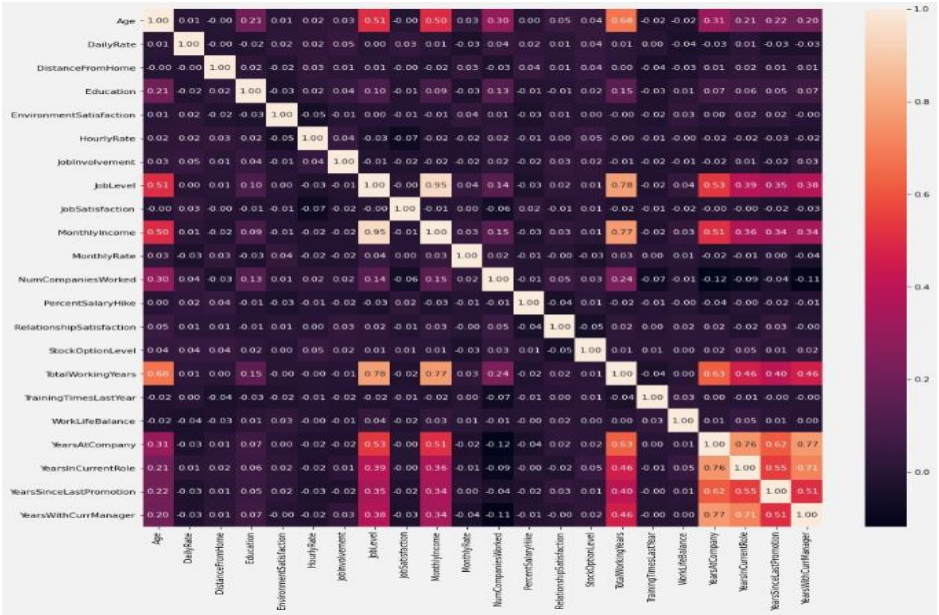


Figure 15 - Corrélations entres les variables du jeu de données

Ce graphique montre les valeurs de corrélation de Pearson, et il y a une présence de valeurs de corrélation élevées entre différents ensembles de variables telles que le niveau de poste et le revenu mensuel, le

niveau de poste et le nombre total d'heures de travail et bien d'autres encore. Cependant, le niveau de l'emploi semble être une variable ordinale.

De même, l'âge semble avoir une relation linéaire modérée avec le revenu mensuel, le nombre total d'années de travail et le revenu mensuel, ce qui est intuitivement logique.

6.3. Conclusion de l'analyse exploratoire

Selon la question commerciale en jeu, il existe de nombreuses façons de manipuler davantage l'ensemble de données. Les résultats ci-dessus sont vraiment intéressants et auraient pu être améliorés par des tests d'hypothèses et des modélisations plus poussés, en tenant compte des hypothèses appropriées chaque fois que cela était nécessaire.

Toutes les variables sont pertinentes dans le cadre d'une modélisation prédictive.

6.4. Entraînement simple

	Modèle Baseline	Modèle élaboré
Accuracy	0.85	0.85
Precision	1	0.57
Recall	0.007	0.03
AUC	0.71	0.69

6.4.1. Bilan

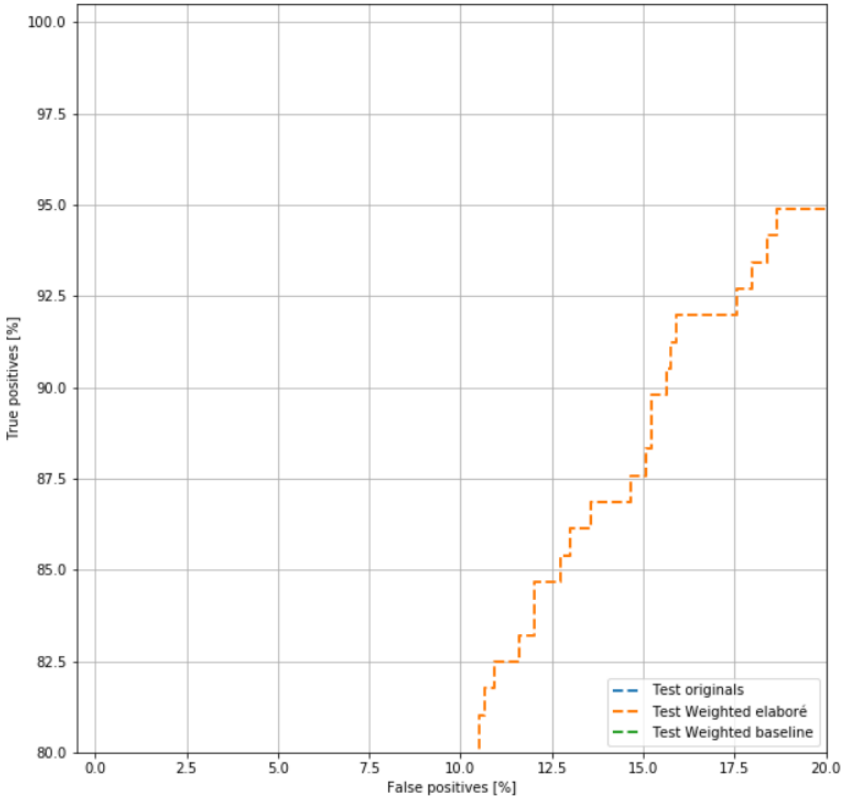
Les deux modèles obtiennent des résultats proches de zéro en *recall*. Un classifieur aléatoire obtiendrai de meilleures performances.

Cela indique qu'ils ne sont pas adaptés aux données.

6.5. Entraînement avec pondération des classes

	Modèle Baseline	Modèle élaboré
Accuracy	0.76	0.72
Precision	0.32	0.29
Recall	0.54	0.59
AUC	0.73	0.74

6.5.1. Bilan



Les deux modèles (baseline et élaboré) sont nettement au-dessus de des précédents à entrainement simple. Les gains en *recall* sont significativement plus élevés pour le modèle élaboré. A noter que sur ces courbes de ROC, les modèles Baseline ne sont pas représenté car sont sensiblement en dessous des modèles élaborés.

6.6. Entraînement avec oversampling

6.6.1. Random oversampling

	Modèle Baseline	Modèle élaboré
Accuracy	0.80	0.98
Precision	0.40	0.98
Recall	0.77	0.88
AUC	0.84	0.94

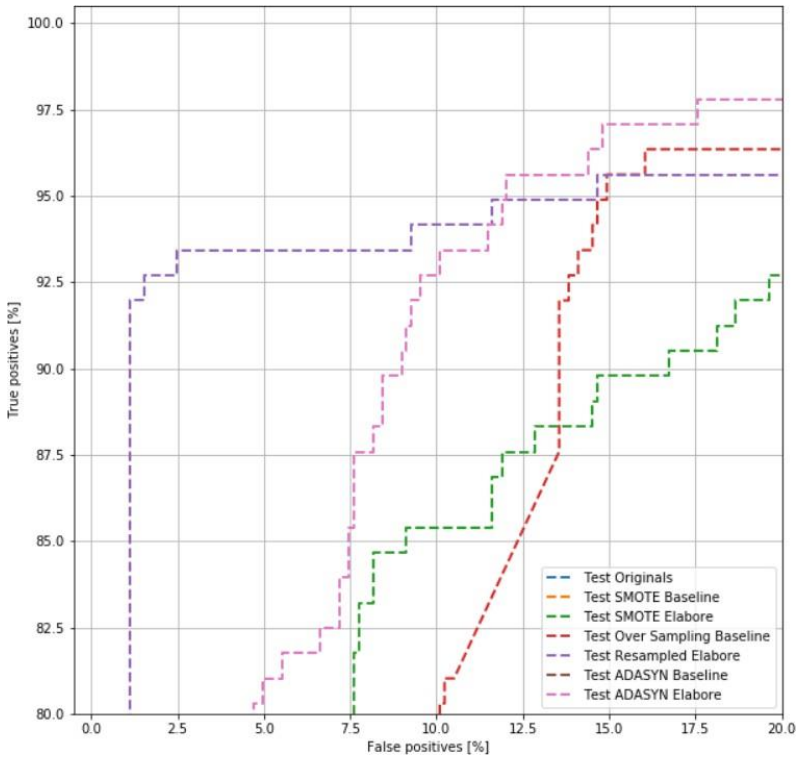
6.6.2. SMOTE

	Modèle Baseline	Modèle élaboré
Accuracy	0.78	0.97
Precision	0.37	0.95
Recall	0.67	0.88
AUC	0.81	0.94

6.6.3. ADASYN

	Modèle Baseline	Modèle élaboré
Accuracy	0.79	0.97
Precision	0.38	0.95
Recall	0.70	0.88
AUC	0.82	0.94

6.6.4. Bilan



La méthode de sur-échantillonnage semble particulièrement bien adaptées pour ce jeu de données, elles obtiennent des performances élevées à la fois en *recall* et *précision*.

6.7. Entraînement avec undersampling

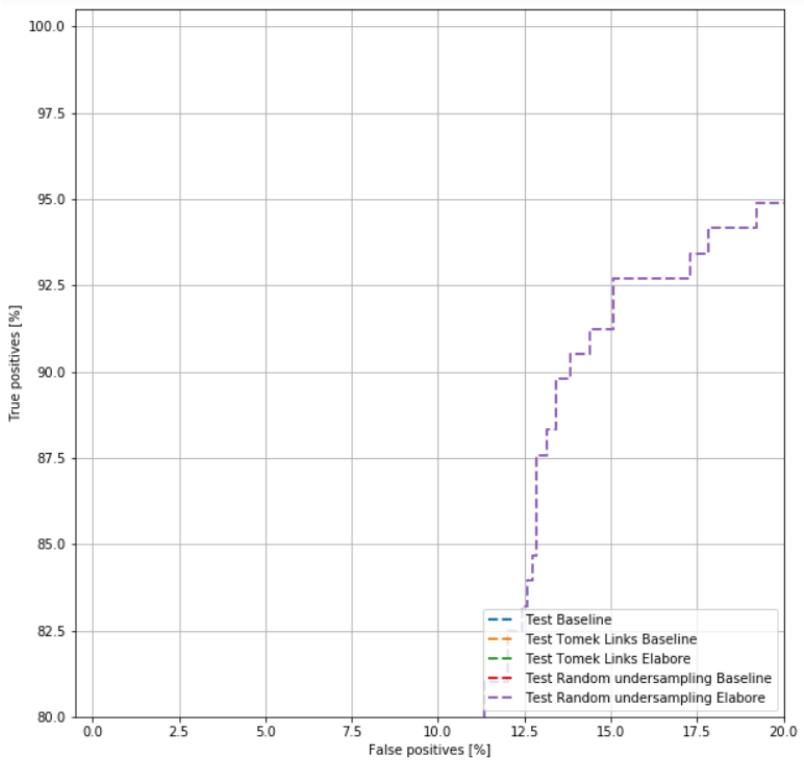
6.7.1. Random undersampling

	Modèle Baseline	Modèle élaboré
Accuracy	0.79	0.81
Precision	0.38	0.44
Recall	0.70	0.89
AUC	0.82	0.91

6.7.2. Tomek Links

	Modèle Baseline	Modèle élaboré
Accuracy	0.86	0.95
Precision	0.69	0.95
Recall	0.14	0.74
AUC	0.80	0.93

6.7.3. Bilan



Les méthodes de sous échantillonnage obtiennent de moins bons résultats comparativement à celles utilisant le sur-échantiollange.

6.8. Entraînement avec une approche hybride de resampling

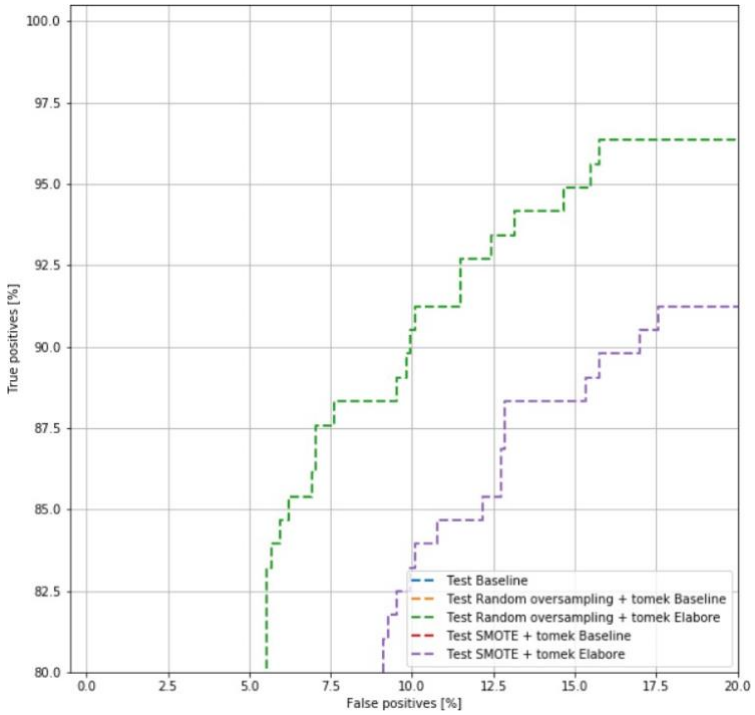
6.8.1. Random oversampling + Tomek links

	Modèle Baseline	Modèle élaboré
Accuracy	0.78	0.96
Precision	0.36	0.88
Recall	0.66	0.86
AUC	0.82	0.93

6.8.2. SMOTE + Tomek Links

	Modèle Baseline	Modèle élaboré
Accuracy	0.78	0.97
Precision	0.37	0.95
Recall	0.67	0.85
AUC	0.82	0.94

6.8.3. Bilan



Les deux méthodes hybrides profitants des effets de l'utilisation d'approches de sur-échantillonnage obtient des résultats très probants. Cela confirme et va dans le sens des résultats précédents.

6.9. Conclusion

Sur ce jeu de données, les méthodes d'over sampling obtiennent les meilleurs résultats. Celles-ci sont donc les plus adaptées aux données.

7. Conclusion générale

Ce projet nous aura permis de mettre en pratique de multiples méthodes de rééquilibrage de données en les appliquant à des dataset issus de problématiques réelle et par la suite d'entrainer des classifieurs *deeplearning*.

Les résultats obtenus sur chaque jeu de données ont montré des disparités très importantes.

Il n'existe de pas de méthode universelle permettant de résoudre le problème de manière efficace, chaque méthode convient à un type de jeu de données.

Il donc est important de pouvoir essayer toutes les approches avant de prendre une décision quant à celle la plus adaptée.