Université Lumière Lyon 2

Apprentissage Déséquilibré
Advanced Supervised Learning

Auteur :
Noé LEBRETON
Vishwa ELANKUMARAN

23 octobre 2021

Table des matières

1 Introduction

De nos jours, nous avons accès à une multitude de donnée. Cependant, cette quantité de données s'accompagne de données déséquilibrées. On appelle un ensemble de données déséquilibrées si les classes ne sont pas représentées de manière approximativement égale (Exemple : "A" : 50 observations, "B" : 5 observations). On observe la présence de données déséquilibrées dans des domaines tels que la télé-communication, les maladies rares, l'apprentissage de la prononciation des mots, les appels télephoniques frauduleux... Il existe deux types de déséquilibres dans un ensemble de données. L'un est le déséquilibre entre classes, dans lequel certaines classes ont beaucoup plus d'observations que d'autres. L'autre est le déséquilibre intra-classe, les individus d'une classe sont regroupés dans des sous-parties de l'espace. Par convention, dans les ensembles de données déséquilibrées, nous appelons les classes ayant plus d'observations les classes majoritaires et celles ayant moins d'observations les classes minoritaires. Pourquoi s'intéresser aux données déséquilibrées ? Les algorithmes traditionels ne fonctionnent pas très bien pour un ensemble de données déséquilibrées car la distribution des données n'est pas réellement prise en compte. En effet, si on ignore cette asymétrie, toutes nos méthodes dites classiques seront biaisées et bien sûr la détection de ces classes minoritaires seraient difficile à détecter. Nous allons introduire des techniques qui vont permettre de rééquilibrer nos classes. De plus, nous chercherons à améliorer la performance des algorithmes d'apprentissage automatique autrement dit nos mesures d'erreurs par le biais de ces techniques de ré-échantillonnage. Dans un premier temps, nous verrons des techniques d'échantillonnage, et dans un second temps, nous utiliserons plusieurs techniques d'échantillonnage synthétique, qui sont SMOTE (Synthetic Minority Over-sampling Technique), ensuite une autre technique de sur-échantillonnage quelque peu similiaire, qui se nomme Borderline-SMOTE, et puis, nous terminerons par une approche d'échantillonnage synthétique adaptatif appelée ADASYN. Dans cette étude, nous utiliserons un jeu de données publié le 6 avril 2021, qui traite de l'analyse du risque de crédit [4]. Le principe de ce dernier est d'étudier ou rendre un avis sur les demandes de crédits et évaluer les risques liés à leur octroi selon plusieurs facteurs tels que la stratégie commerciale d'une entreprise. Notre jeu de données contient des données plutôt complètes sur les prêts émis entre 2007 et 2015. Nous avons accès à une palette d'informations comme le statut du prêt (en cours, en retard, entièrement payé ...) ou encore sur les informations de paiement. Notre jeu de données contient un total de 855969 observations avec 73 variables. De plus, l'ensemble de données est très deséquilibré, avec environ 6% des prêts considérés comme impayés. Ce jeu de données comporte différents types d'éléments tels que des catégories (variable qualitative), des chiffres et des dates. Dans cette étude, pour des raisons computationnelles, nous avons quelque peu simplifié notre jeu de données en considérant uniquement les variables d'importances (caractéristiques importantes) proposées par notre "fournisseur" [4] de notre jeu de données qui sont les suivantes :

- **loan_amnt** : Le prêt demandé par le client.
- **int_rate** : L'intérêt du prêt.
- **grade** : Grade de prêt noté par des catégories : A, B, C, D, E, F, G.
- **annual_inc** : Revenu annuel du client.
- **purpose** : Le but principal du prêt (de l'emprunt).
- **installments** : Montant mensuel des paiements pour le prêt.
- **term** : Durée du prêt jusqu'à son remboursement.

De plus, nous allons sélectionner notre variable cible qui se nomme **default_ind**, qui nous indique quelles sont les personnes qui n'ont pas payé leurs prêts. Ainsi, un défaut de paiement peut se produire lorsqu'un emprunteur est incapable d'effectuer des paiements en temps voulu, par un manque de paiement, ou encore lorsqu'il évite ou arrête de faire des paiements. Penchons-nous un peu plus en détail sur nos données. Nous remarquons qu'il y a des variables qualitatives, or les traiter/représenter par des méthodes dites classiques est assez compliqué. C'est pour cela que nous allons utiliser une analyse factorielle sur nos données explicatives c'est-à-dire sur nos 7 axes.

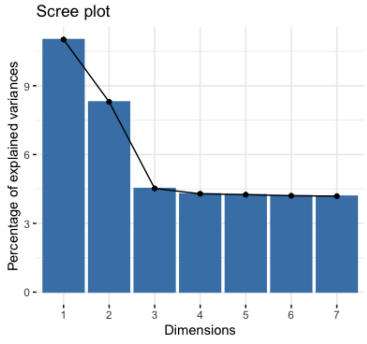


FIGURE 1 – Proportions de variances expliquées par les différents axes

Bien sûr, nous obtenons une inertie qui n'est pas énorme environ 41%, ce qui veut dire que nos axes n'expliquent pas "correctement" nos données car nous avons déjà subdivisé notre échantillon en 8 variables sur 73 variables initialement (à cause d'un problème computationnel). Par la suite, dans cette étude, nous utiliserons ce modèle.

2 Mesure d'Erreur

2.1 Matrice de Confusion

Regardons comment nos données sont déséquilibrées. D'après nos données, on remarque qu'il y a environ 5.43% de personnes qui n'ont pas payé leurs prêts. Ce qui représente un gros déséquilibre. À présent, regardons notre mesure d'erreur par le biais d'une matrice de confusion. Mais avant, définissons ce qu'est une matrice de confusion. Pour faire simple, celle-ci appelée aussi tableau de contingence est une matrice qui mesure la qualité des prédictions dans un système de classification.

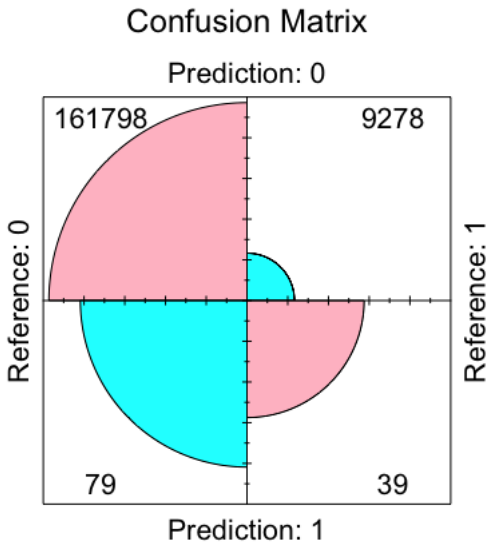


FIGURE 2 – Matrice de Confusion sur nos données initiales en utilisant un modèle d'analyse discriminante linéaire

On remarque que lorsqu'il s'agit de prédire que les prêts d'une personne ont été payés, notre modèle s'en sort plutôt bien. Il prédit la classe 0 161798 fois (True Positive) sur 171076, et ne se trompe que rarement $1 - \frac{TP}{TP+FN} = 5.4\%$ (1 - recall/rappel). Par contre, lorsqu'il s'agit d'estimer les prêts d'une personne n'ayant pas payé celui-ci, il se trompe beaucoup plus souvent, soit $\frac{FP}{TN+FP} = 66.94\%$ du temps. Cela peut s'expliquer par le fait que la répartition des classes qui n'est pas très équilibrée. Intéressons nous à d'autres métriques. A partir de toutes les classes positives (ici 0), nous avons prédit correctement environ $\frac{TP}{TP+FP} = 99.95\%$ (précision) ce qui n'est pas si mauvais. De plus, parmi toutes les classes, nous avons prédit correctement $\frac{TP+TN}{TP+FN+FP+TN} = 94.53\%$ (accuracy). Par conséquent, on en déduit que le taux d'erreur de notre modèle est de 5.46%. Ici, l'accuracy n'est pas vraiment un bon indicateur lorsque les données sont déséquilibrées. Regardons une autre métrique. F1-Score est une moyenne harmonique c'est-à-dire l'inverse de la moyenne arithmétique, de précision et de rappel. Elle est intéressante pour rechercher un équilibre entre rappel et précision car elle traite des faux positifs et des faux négatifs. Ici, la distribution des faux positifs et faux négatifs est de $\frac{2*Prcision*Rappel}{Prcision+Rappel} = 97.19$ pour la classe 0. Ce qui implique que la précision et le rappel est très bonne. Or, en présence d'échantillons déséquilibrées, il est préférable d'utiliser d'autres mesures tels que la courbe ROC ou d'autres techniques similaires. Cependant, nous nous sommes heurtés à quelques difficultés concernant la courbe ROC. En effet, cette dernière nous renvoyait des courbes aux allures étrangères (incohérences). C'est pour cela, que nous préférons faire l'impasse sur cette potentiel partie.

Dans cette étude, nous cherchons à rééquilibrer les classes par des méthodes d'échantillonnages afin d'en tirer un meilleur taux d'erreur.

3 Technique d'échantillonnage classique

Les méthodes d'échantillonnages correspondent à la sélection d'un échantillon de la population étudiée. Bien sûr, ce dernier doit être constitué de manière aléatoire pour éviter d'éventuels biais. Etant donné qu'on cherche à ré-équilibrer les classes, on va utiliser deux techniques d'échantillonnages.

3.1 Sur-Échantillonnage

Le sur-échantillonnage consiste à rééchantillonner/dupliquer les données en faisant des tirages avec remise de la classe minoritaire. Bien sûr, nous allons faire un sur-échantillonnage aléatoire. Ainsi, les tirages seront fait de manière aléatoire avec remise. On peut sur-échantillonner plusieurs fois (2x, 3x, 5x, 10x, ...). Cependant, un "sur sur-échantillonnage" peut augmenter le risque qu'il y ait des sur-ajustements car il y aura beaucoup de copies de la classe minoritaire. Dans un premier temps, en supposant qu'on veuille rééquilibrer notre échantillon à 50|50 via un sur-échantillonnage, notre nouvel échantillon contient 1619004 observation. Maintenant que nous avons rééchantillonné notre jeu de données et rééquilibré en sur-échantillonnant ce dernier, appliquons-le à un modèle afin de savoir si nous avons de meilleures performances liées à la mesure d'erreur. Nous emploierons les mêmes modèles qu'utilisés au modèle/échantillon initial. Regardons ce que nous renvoie la matrice de confusion.

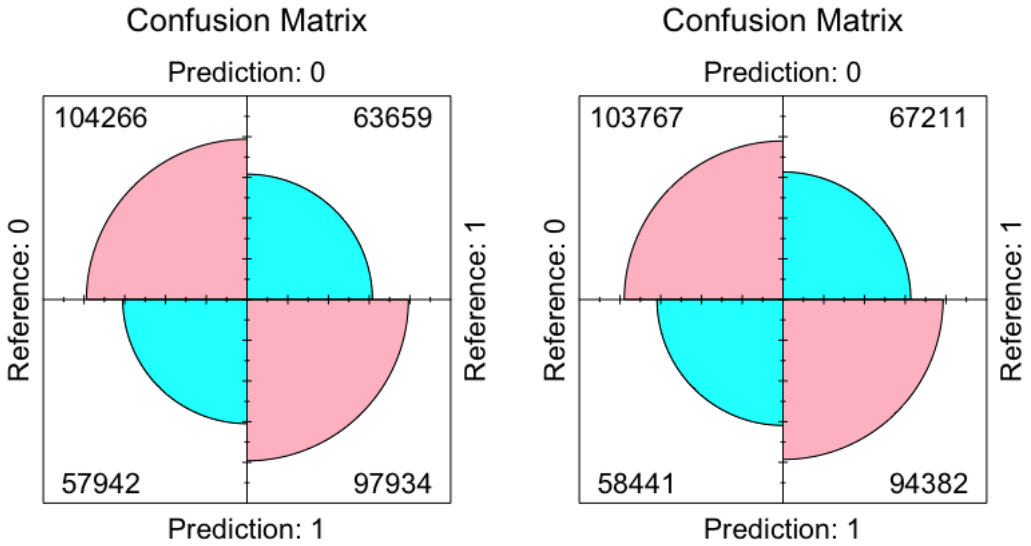


FIGURE 3 – Matrice de Confusion sur nos données sur-échantillonnées sur nos deux modèles

Cette fois-ci, on remarque qu'il y a "plus de valeurs" prédictes dans la classe 1, tout simplement car nous avons plus d'observations dans notre échantillon (comme nous avons équilibré ce dernier). On remarque qu'on effectue plus d'erreur car le taux d'erreur généralisé de nos modèles est d'environ 37.55% et 38.8% (1 - accuracy) respectivement ce qui est énorme, mais attention, ce n'est pas comparable à celui de nos données initiales car la taille et la répartition de nos données ne sont plus les mêmes. Cependant, on peut expliquer cette erreur par le fait du sur-ajustement des données car nous avons potentiellement en moyenne 17 fois copiés (tirages avec remise) la classe minoritaire pour pouvoir rééquilibrer notre échantillon. De plus, nos précisions sur nos classes ne sont plus les mêmes. En effet, la précision sur la classe positive est de 63.97%. On pourrait dans ce cas utiliser le modèle d'analyse discriminante car cette dernière procure de meilleures performances.

3.2 Sous-Échantillonnage

Le sous-échantillonnage est le contraire du sur-échantillonnage c'est-à-dire qu'on retire de manière aléatoire des échantillons issus de la classe majoritaire, afin de réduire le nombre d'observations dans la classe majoritaire. Le principal inconvénient du sous-échantillonnage est que cette méthode peut rejeter des données potentiellement utiles qui pourraient être importantes. De plus, elle peut augmenter la variance du classificateur. Appliquons cette méthode à nos modèles et une regardons une nouvelle fois les performances de ce dernier. Cette fois-ci, on a 92934 observations.

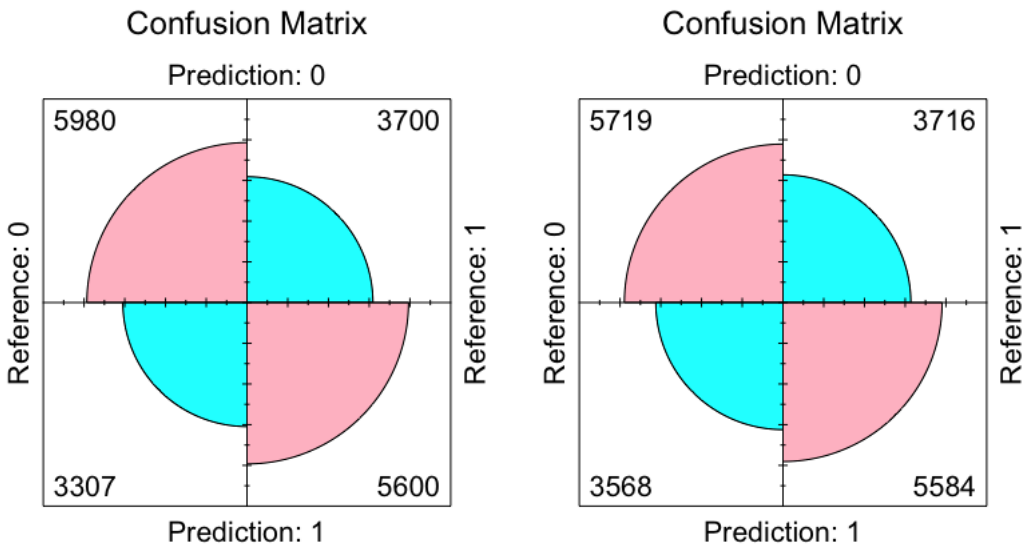


FIGURE 4 – Matrice de Confusion sur nos données sous-échantillonnées sur nos deux modèles

Ici, le taux d'erreur de généralisation est d'environ 39% sur nos deux modèles. Cette fois-ci, il est possible d'utiliser les deux méthodes (analyse discriminante et arbre de décision) car les taux d'erreurs sont similaires.

4 Technique d'échantillonage synthétique

Il existe plusieurs méthodes pour sur-échantillonner un ensemble de données pour un problème de classification.

4.1 SMOTE : Synthetic Minority Over-sampling Technique

Nous proposons une approche de sur-échantillonnage dans laquelle la classe minoritaire est sur-échantillonnée en générant des échantillons synthétiques appelés SMOTE [5]. Cette technique est basée sur les plus proches voisins avec la notion de distance euclidienne. En effet, la classe minoritaire est sur-échantillonnée en prenant chaque échantillon de la classe minoritaire et en introduisant des points synthétiques sur les segments de ligne joignant tout les k plus proche voisins de la classe minoritaire. Les k plus proches voisins sont choisis aléatoirement. Ainsi, la donnée générée n'est jamais un double exact de l'un de ses "parents". Notre implémentation utilise les cinq plus proches voisins. Pour faire plus simple, le SMOTE se décompose en plusieurs étapes :

— On choisit notre vecteur caractéristique de notre classe minoritaire

— On sélectionne ses k plus proches voisins et on choisit de manière aléatoire l'un des k voisins.

— On effectue la différence entre notre vecteur caractéristique ($Sample$) et son voisin le plus proche choisi précédemment ($kNNSample$).

$$dif = Sample - kNNSample$$

— On multiplie cette différence par un nombre aléatoire entre 0 et 1, et on l'additionne à notre vecteur caractéristique, qui va nous permettre de générer nos données synthétiques.

$$synthetic = Sample + dif \cdot U([0, 1])$$

Dans notre cas, représenter graphiquement ceux que nous donne le SMOTE n'est pas possible car nous sommes dans un espace de dimension supérieure R^8, mais pour qu'on ait une idée de comment ce dernier est représenté, utilisons un jeu de données simple qui est **iris**. Nous avons fait en sorte que ce jeu de données soit désequilibré, et avons utilisé cette méthode de sur-échantillonnage.

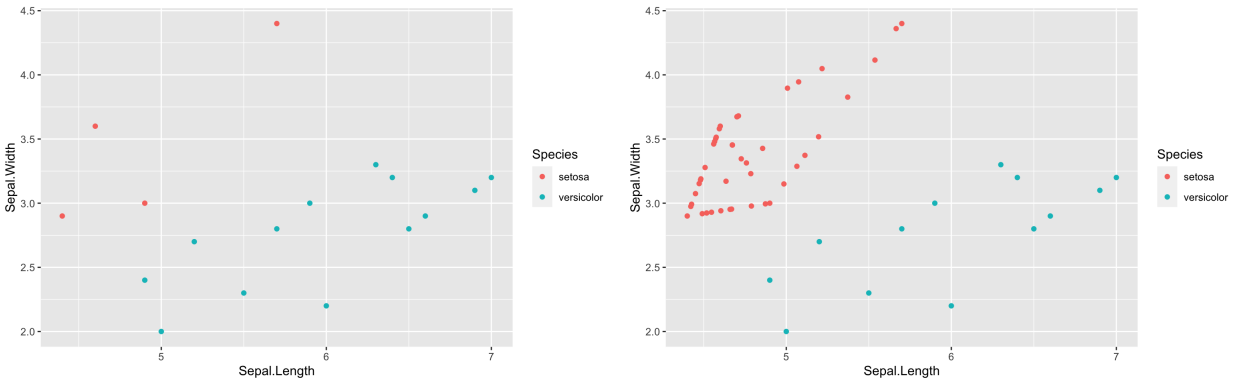


FIGURE 5 – Données sur Iris avant et après SMOTE

On remarque que les données synthétiques sont entre les segments des classes minoritaires et sont tous différents les uns des autres. Ici, la quantité de sur-échantillonnage est de 1700% et seul 3 de ses plus proches voisins ont été considérés. À présent, regardons ce que le SMOTE nous donne pour nos données de crédit. Bien sûr, comme la représentation graphique n'est pas possible nous nous contenterons d'analyser les performances liées à ce nouvel échantillon.

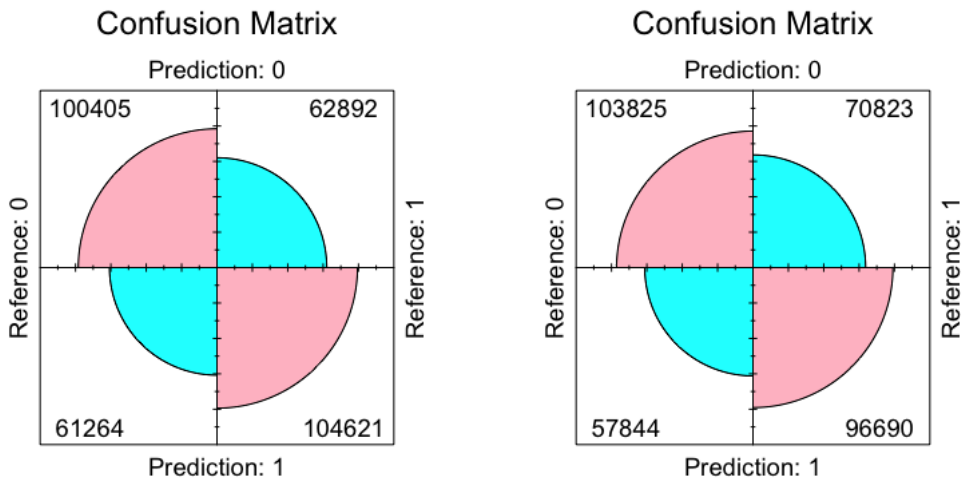


FIGURE 6 – Matrice de Confusion sur nos données utilisant du SMOTE sur nos deux modèles

D'un coup d'oeil, on remarque que les deux modèles nous donnent à peu près les mêmes résultats en terme de métriques. En effet, le taux d'erreur généralisé est de environ 39%, et on remarque qu'on obtient à quelques pourcentages près les mêmes taux d'erreur en généralisation que les techniques d'échantillonnage classiques. De plus, on remarque que nos modèles se trompent environ 40% du temps pour qu'il s'agit de prédire si une personne va payer son prêt. Ces résultats sont semblables aux résultats obtenus en faisant rééchantillonnant de manière classique. Mais on constate une légère amélioration de SMOTE. Néanmoins, cette technique ne prend pas en compte les voisins qui peuvent provenir de la classe majoritaire. Cela a pour conséquence d'augmenter les probabilités qu'il y ait des zones dans lesquelles il y a un chevauchement entre la classe majoritaire et minoritaire, faisant ainsi diminuer les performances de cet algorithme car elle rajouterait du bruit. Il existe toute de même d'autres variantes du SMOTE qui sur-échantillonne en fonction de positionnement des classes minoritaire parmi les classes majoritaires.

4.2 Borderline-SMOTE

Nous cherchons à obtenir de meilleures prédictions avec un taux d'erreur le plus minimal possible. Nous avons vu que les algorithmes de classification tentent de comprendre la limite de chaque classe. Or les points situés à la limite des classes sont plus susceptibles d'être mal classés que ceux qui en sont éloignés. En se basant sur ces dires, on peut sur-échantillonner des minorités dans lesquelles seuls les points situés à la limite de la classe minoritaire/majoritaire seront sur-échantillonnés. Cette méthode se nomme borderline-SMOTE [3], et bien sûr elle se base sur la méthode du SMOTE. On va supposer que l'ensemble de notre échantillon est noté E, notre classe minoritaire sera noté m et bien sûr notre classe majoritaire sera lui aussi noté M. Donc borderline-SMOTE se construit de la manière suivante :

— Pour chaque élément de la classe minoritaire m, nous calculons ses k plus proches voisins sur l'ensemble de notre échantillon E. Et on notera M' le nombre de majoritaires parmi les k plus proches voisins.

— Si $\frac{k}{2} \leq M' \leq k$ c'est-à-dire que le nombre de majoritaire voisins est plus grand que le nombre de ses minoritaires voisins alors m est mis dans un ensemble que l'on nomme DANGER. Ces derniers sont les points situés à la bordure des classes.

— On calcule pour chaque point dans DANGER son k plus proche voisins de la classe minoritaire.

— A partir de là, nous générons α points synthétique à partir du sous-échantillon DANGER où α est le nombre de points synthétiques que nous souhaitons générer. Pour chaque points dans DANGER, nous choisissons de manière aléatoire son plus voisin parmi la classe minoritaire.

— Puis nous générons notre point synthétique comme avec SMOTE.

$$synthetic = Sample + U([0,1]) * (Sample - kNNSample)$$

Comme précédemment, reprenons les données iris pour avoir une représentation graphique de ce que notre algorithme fait.

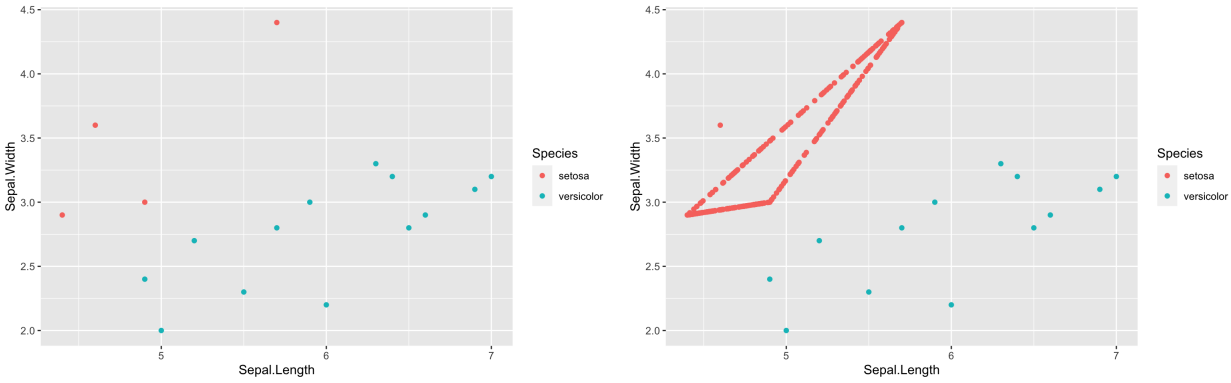


FIGURE 7 – Données sur Iris avant et après Borderline-SMOTE

Il est un peu plus dur de voir ici que les points synthétiques sont générés parmi les classes minoritaires qui sont au bord de la classe majoritaire. Nous tenons à préciser que dans cet exemple, nous avons pris seulement 2 plus proches voisins. Et bien sûr, nous a généré beaucoup de points synthétiques pour connaître en quelque sorte l'allure de ce dernière. A présent, regardons si cette nouvelle méthode va améliorer nos performances. Bien sûr, nous conserverons les mêmes paramètres que lors des méthodes précédentes pour pouvoir avoir un ordre de comparation, c'est-à-dire que notre échantillon sera rééquilibrée à 50|50, et nous entraînerons nos modèles sur 80% des données.

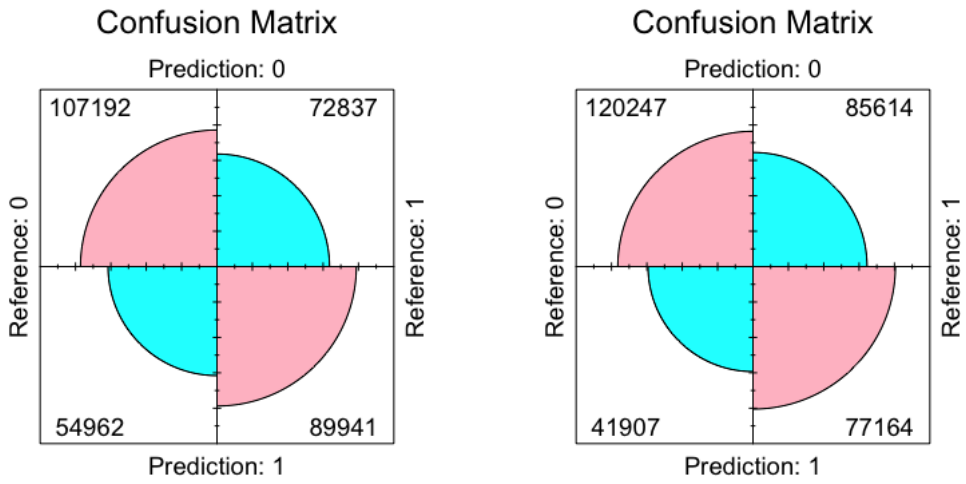


FIGURE 8 – Matrice de Confusion sur nos données utilisant du borderline SMOTE sur nos deux modèles

Encore une fois, nous obtenons des taux d'erreurs similiaires 39%. Le rappel c'est-à-dire le taux de positif dans la classe 0 est de 74.16%. On pourrait privilégier le premier modèle dans ce cas. Il existe d'autres méthodes plus récente de rééchantillonnage qui sont prometteurs. C'est ce que nous allons voir.

4.3 ADASYN : Adaptive Synthetic Sampling Approach for Imbalanced Learning

La dernière méthode approchée se nomme ADASYN [2], celle-ci a également pour but de rééchantillonner les données en sur échantillonnant la classe minoritaire à travers la création de données synthétiques. Cette approche reprend sur beaucoup de points la méthode SMOTE [5], comme la manière de créer une nouvelle donnée synthétique :

$$s_i = x_i + (x_{zi} - x_i) * \lambda$$

où x_i est une donnée minoritaire, x_{zi} est une donnée minoritaire faisant partie des k plus proches voisins de x_i et λ une valeur aléatoire : $\lambda \in [0, 1]$.

Dans cette formule calculant la donnée synthétique un premier élément diffère entre les 2 méthodes, pour SMOTE x_{zi} est choisis aléatoirement dans les k plus proches voisins de x_i. Cela traduit qu'il est possible de créer une valeur synthétique entre une donnée provenant de la classe minoritaire (x_i) et une de la classe majoritaire (x_{zi}). Le choix de laisser la possibilité de créer la valeur synthétique à partir d'une donnée provenant de la

classe majoritaire reste discutable car l'idée principale est de créer de nouvelles données représentant la classe minoritaire. Au contraire avec ADASYN, l'algorithme permet d'obtenir uniquement les données synthétiques à partir des données provenant de la classe minoritaire.

Pour revenir sur la sélection des données x_{zi} permettant de créer les variables synthétiques, il est possible que dans les plus proches voisins de x_i les données proviennent uniquement de la classe majoritaire. Cela peut potentiellement provenir de 2 différents aspects :

— L'utilisation d'une méthode de *fast KNN* pour trouver les plus proches voisins de x_i.
— Le chevauchement des différentes classes, c'est-à-dire que certains points ne se différencient pas avec les classes.

L'algorithme ADASYN ne permet pas de traiter ce cas, une condition a donc été appliqué pour faire un choix aléatoire de x_{zi} lorsque les k plus proches voisins de x_i font uniquement partie de la classe majoritaire.

ADASYN ne se résume pas uniquement à cette différence, l'idée principale de cet algorithme est d'utiliser une distribution de densité permettant de décider le nombre de données synthétiques qu'il faut générer pour chaque donnée minoritaire x_i. Les auteurs [2] donne :

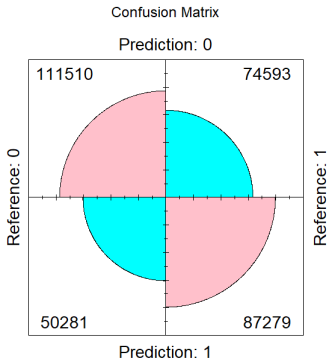
$$r_i = \Delta_i / K, \quad i = 1, ..., m_s$$

où Δ_i est le nombre de données de la classe majoritaire dans les k plus proches voisins de x_i et m_s le nombre de données de la classe minoritaire.

Pour obtenir obtenir une distribution de probabilité il est nécessaire de normaliser r_i, pour cela il introduise $\hat{r_i} = r_i / \sum_{i=0}^{m_s} r_i$ ou $\hat{r_i}$ est une distribution de densité ($\sum_{i}^{m_s} r_i = 1$). L'idée suivante est d'obtenir le nombre de données synthétiques à créer pour chaque donnée minoritaire x_i : $g_i = \hat{r_i} * G$ avec G le nombre total de données synthétiques voulues.

La mise en place de la distribution de densité permet d'obtenir pour chaque x_i un nombre de données synthétiques à créer en fonction de la nature de ses plus proches voisins. En d'autres termes, lorsque le nombre de voisins de x_i provenant de la classe minoritaire est fort il n'est pas nécessaire d'obtenir beaucoup de données synthétiques ($\hat{r_i}$ faible et donc g_i faible). Au contraire, quand beaucoup de données de la classe majoritaire sont dans le voisinage de x_i, il est intéressant de générer plus de données synthétiques ($\hat{r_i}$ grand et donc g_i grand). En pratique, cette distribution de densité permet de créer plus de données là où l'apprentissage est davantage compliqué.

Pour revenir à l'objectif de ce projet, nous avons rééchantillonné nos données avec cette méthode. La première étape était d'obtenir un jeu de données quasiment équilibré où 50% des données appartient à la classe 0 et 50% à la classe 1. Pour cela, le paramètre β a été fixé à 1 ce qui représente approximativement l'équilibre des classes.



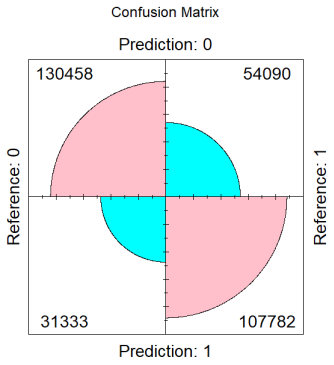


FIGURE 9 – Matrice de confusion sur nos données utilisant du ADASYN ($\beta = 1$) sur nos deux modèles (ADL et arbre de décision)

En comparaison avec les données déséquilibrées, on constate que les modèles arrivent mieux à prédire la classe cible (1). Pour l'ADL, les résultats deviennent équilibrés même si la classe qui à l'origine est majoritaire compte plus de données ; on observe que les prédictions de 0 sachant que la vraie classe est 1 sont plus nombreuses que les prédictions de la classe 1 sachant que la vraie classe est 0. En utilisant l'arbre de décision pour prédire nos données, les observations sont les mêmes mais ici les pourcentages d'individus mal prédits baissent pour les deux classes.

Le tableau ci-dessous nous permet d'affirmer que l'utilisation d'un arbre de décision pour notre problème de classification est mieux que l'utilisation de l'ADL. La totalité des métriques présentes sont plus hautes pour l'arbre de décision.

TABLE 1 – Résultats en fonction de la méthode utilisée

	ADL	Arbre de décision
Accuracy	0.61	0.74
Sensitivity	0.69	0.81
Specificity	0.54	0.67

Dans les premiers résultats explosés ci-dessus nous avons choisi de rééquilibrer totalement notre jeu de données en fixant $\beta = 1$. Plusieurs articles parlent de la variation des valeurs de β ($[0, 1]$) et expliquent qu'il n'est pas obligatoire de rééquilibrer totalement le jeu de données pour obtenir de bons résultats. L'idée est de faire varier la valeur prise par β ce qui va directement jouer sur la taille de l'échantillon final. Pour étudier cet aspect, nous avons pris des valeurs de β allant de 0 à 1 par pas de 0.1. Au total, 10 nouveaux échantillons ont été créer.

Pour commencer, deux graphiques (Figures 10 et 11) sont présentés permettant d'observer l'évolution des valeurs des métriques en fonction de β qui représente le taux de rééchantillonnage. Ici, les métriques ont été calculées en fonction de la modalité 1 (originellement la classe minoritaire) de la variable cible. En utilisant l'ADL (Figure 10), on constate que les valeurs des métriques augmentent lorsque l'échantillon se rééquilibre et cela de manière plutôt linéaire. De manière générale les résultats obtenus ne sont pas jugés bons dans le cadre de la classification, mais dans l'idée du rééchantillonnage les valeurs augmentent ce qui traduit l'amélioration du modèle lorsque les classes deviennent équilibrées. Avec l'arbre de décision (Figure 11), on observe certaines valeurs manquantes car ce modèle n'arrive pas à prédire d'individus appartenant à la classe minoritaire lorsque l'échantillon est extrêmement déséquilibré. De plus, l'évolution des métriques est forte pour les premières valeurs de β puis on observe une évolution plus lente (un type de plateau) à partir de $\beta = 0.3$. Avec ces premiers résultats, on peut conclure que l'ADL nécessite des échantillons fortement équilibrés pour obtenir de bons résultats alors que l'arbre de décision n'a pas besoin d'un rééchantillonnage aussi conséquent.

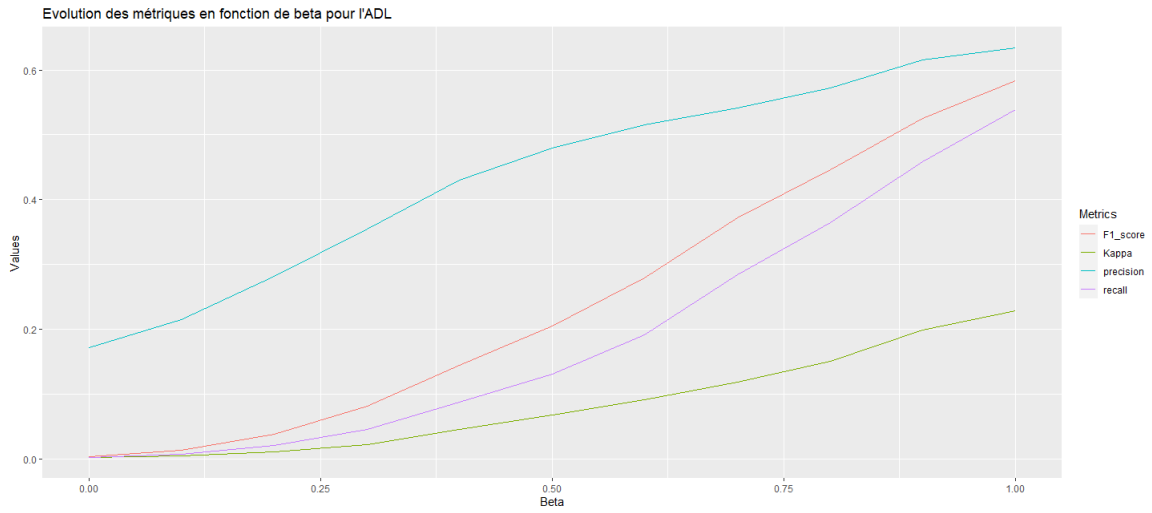


FIGURE 10 – Evolution des métriques obtenues avec l'ADL en fonction de β

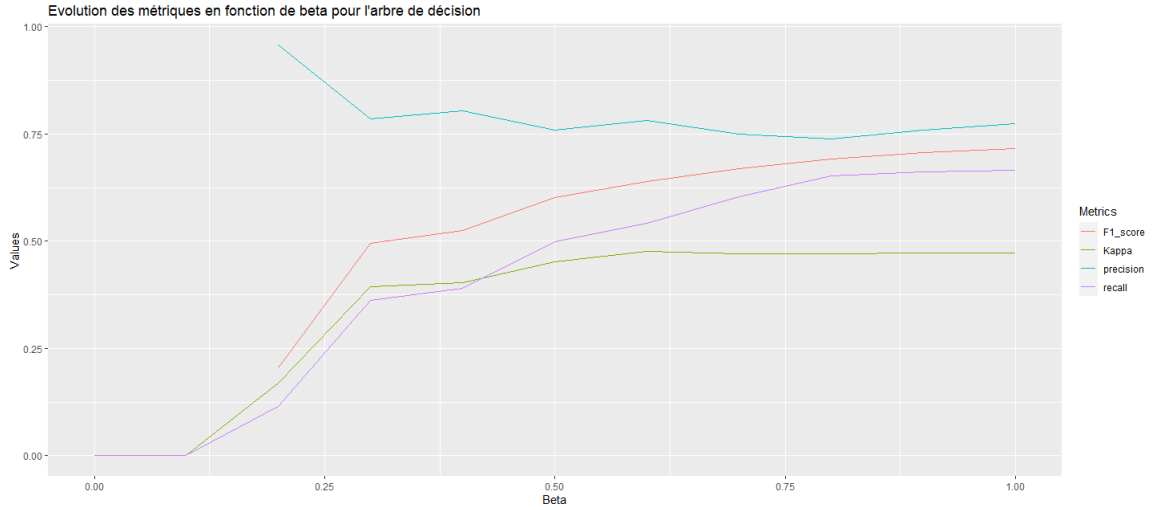


FIGURE 11 – Evolution des métriques obtenues avec l'arbre de décision en fonction de β

Pour continuer notre étude sur le rééchantillonnage avec la méthode ADASYN, nous nous sommes intéressé à une autre vision des résultats. Pour cela, nous avons fait le choix de transformer les matrices de confusion en pourcentage pour chaque méthode et chaque valeur de β. L'idée générale de cette transformation est de supprimer l'effet d'échelle lié au rééchantillonnage qui augmente (non proportionnellement) pour chaque β le nombre d'individus présents dans la matrice de confusion. La deuxième raison de cette modification est de pouvoir affirmer que les proportions d'individus augmentent pour la classe originellement minoritaire et baisse pour la classe originellement majoritaire. Les figures 12 et 13 peuvent être vues comme des matrices de confusion ($\hat{y}$ en ordonnée et y en abscisse) avec comme premier graphique en haut à gauche les résultats (en pourcentage) des individus de la classe 0 bien classés. Pour les deux figures on observe que les graphiques des individus bien classés sont cohérents avec le rééchantillonnage, la classe majoritaire diminue et au contraire, la classe minoritaire augmente. La deuxième diagonale présente également des résultats intérressants, le nombre d'individus de la classe majoritaire prédit dans la classe minoritaire augmente de manière constante lorsque β augmente. Concernant les individus de la classe minoritaire qui sont prédits dans la classe majoritaire, on observe une augmentation plus nette puis une stagnation voir une diminution de la proportion. Cette observation traduit qu'à une certaine valeur de β le modèle fait moins d'erreurs de prédiction pour la classe minoritaire.

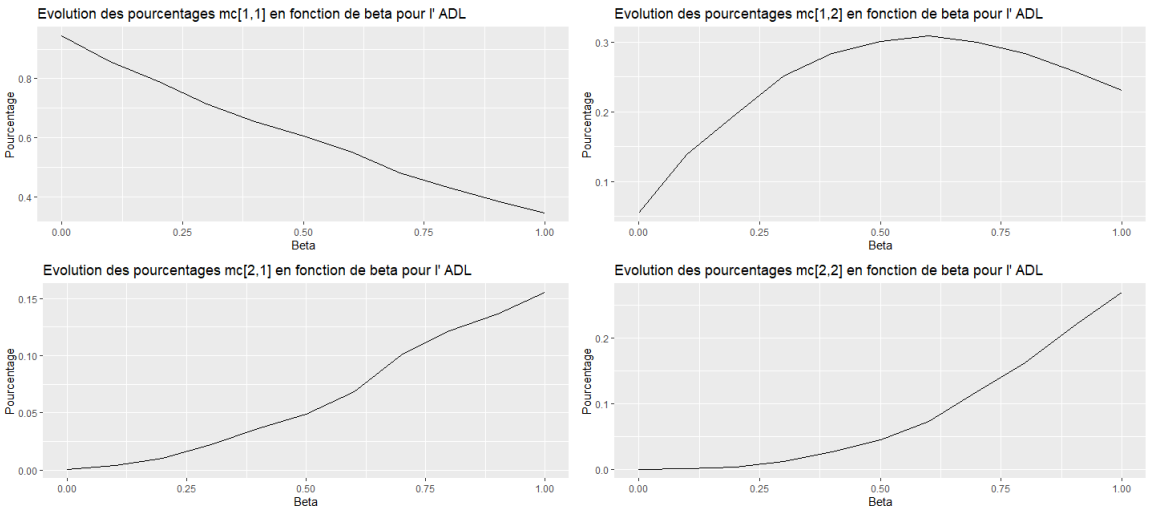


FIGURE 12 – Evolution des pourcentages des matrices de confusion obtenues avec l'ADL en fonction de β

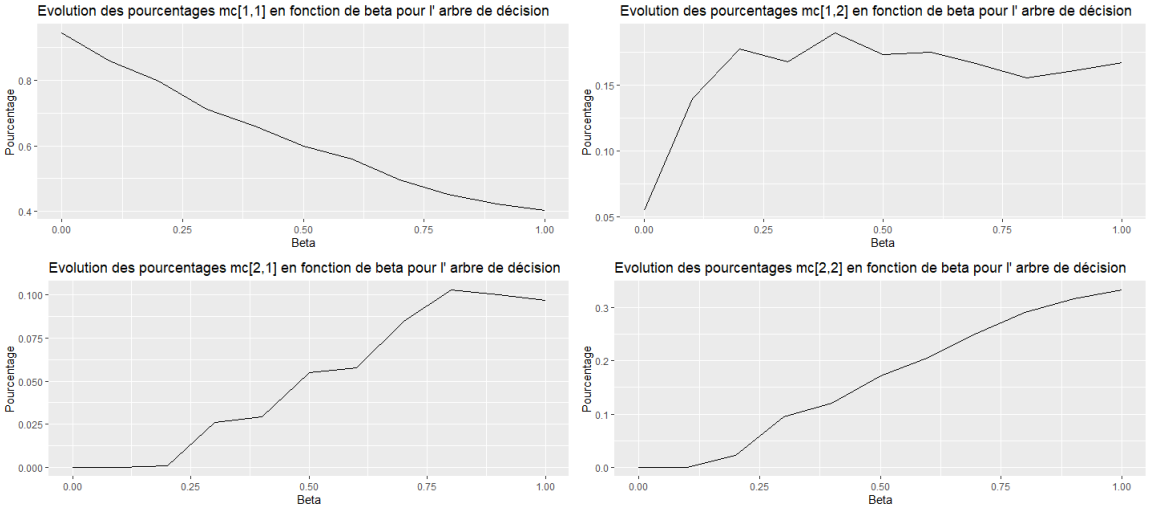


FIGURE 13 – Evolution des pourcentages des matrices de confusion obtenues avec l'arbre de décision en fonction de β

Pour conclure sur cette méthode, nous avons pu constater que la création de données synthétiques aboutit à une évolution des résultats, et plus particulièrement pour la classe minoritaire qui est en pratique souvent la classe d'intérêt.

5 Conclusion

Les données déséquilibrées sont devenues fréquentes dans la plupart des projets d'apprentissage et nécessitent une attention particulière. Comme vous avez pu le voir, la non-prise en compte de ce déséquilibre peut amener à des résultats qui sembleraient bons comme avec le taux d'erreur qui est souvent très petit mais qui ne reflète pas la réalité. Dans ces cas, les modèles et toutes les métriques associées n'ont plus de valeurs d'un point de vue statistique car elles subissent l'effet de grandeur provenant de la classe majoritaire.

Dans ce projet nous avons pu étudier plusieurs méthodes permettant de prendre en compte et modifier ce déséquilibre. Dans un premier temps, des approches simples de sur et sous échantillonnage qui en réalité permet simplement d'améliorer les résultats des modèles. La rapidité ainsi que la simplicité de ces méthodes sont un avantage mais, au contraire, la perte d'information causée par le sous échantillonnage et la redondance de données en utilisant le sur échantillonnage sont des inconvénients. La deuxième approche était la création de données synthétiques, ici l'idée était de se placer dans le cadre du sur échantillonnage mais d'une manière différente que la duplication de données existantes. Cela permet d'avoir davantage de données appartenant à la classe initialement minoritaire en créant des données proches de celles existantes. Pour cela, trois méthodes ont été étudiées : SMOTE, Borderline-SMOTE et ADASYN. L'idée dernière ces approches sont similaires avec quelques différences comme la manière de créer les données synthétiques ou encore le nombre de données synthétiques créées à partir d'une donnée de la classe minoritaire. Les résultats obtenus en utilisant l'ADL et l'arbre de décision sont discutables, lorsqu'on observe uniquement les métriques avec une vision statistique, celles-ci ne peuvent pas être jugées bonnes. Cette observation provient potentiellement de la capacité des variables explicatives à prédire Y. Mais nous avons pu voir que pour chaque méthode mise en place les résultats liés au rééchantillonnage augmentent lorsque le nombre d'individus de la classe initialement minoritaire devient plus important.

Ces différents constats nous permettent de montrer que ces méthodes de rééchantillonnage à partir de données synthétiques sont performantes. Le travail effectué est une certaine vision de traiter les données déséquilibrées, il est important de savoir que de nombreuses approches existent pour ce type de problème. Pour citer un exemple, certaines méthodes comme Adaboost [1] qui a pour but d'avoir une étape de sélection d'individus pour constituer l'échantillon d'apprentissage. Cette étape est réalisée avec des poids donnés aux individus qui évoluent au fil des itérations en fonction de l'erreur de classification. Cela permet donc de constituer des échantillons d'apprentissage comportant majoritairement des données difficiles à prédire.

Références

[1] Yoav Freund and Robert E. Schapire. A short introduction to boosting, 1999.

[2] Edwardo A. Garcia Haibo He, Yang Bai and Shutao Li. Adasyn : Adaptive synthetic sampling approach for imbalanced learning, 2008.

[3] Wen-Yuan Wang Hui Han and Bing-Huan Mao. Borderline-smote : A new over-sampling method in imbalanced data sets learning, 2005.

[4] Ramesh Mehta. Credit risk analysis. <https://www.kaggle.com/rameshmehta/credit-risk-analysis>, 2021. [Dataset].

[5] Lawrence O. Hall Nitesh V. Chawla, Kevin W. Bowyer and W. Philip Kegelmeyer. Smote : Synthetic minority over-sampling technique, 2002.