Projet machine learning : Détection des fraudes liées aux cartes de credit.

KASSI Omar

2 mars 2021

Table des matières

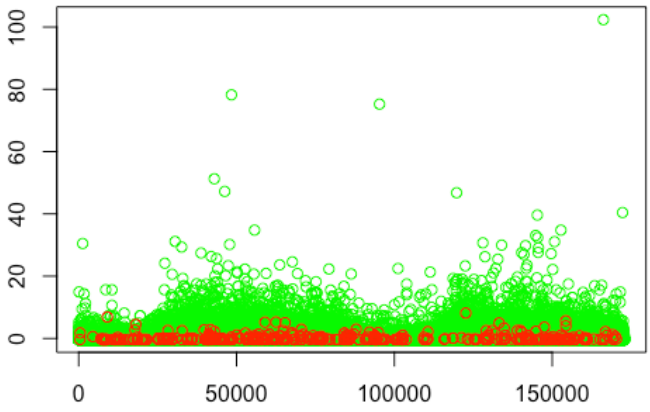
1 Introduction :

Dans le cadre du cours Projet Machine Learning pour la prévision, je me suis intéressé à l'application des méthodes du Machine Learning à la prévision de fraude de carte de crédit.
J'ai à cet effet récupéré la base de donnée de site Kaggle des transactions effectuer par des cartes européenne en Septembre 2013.
La data contient les transactions qui ont été effectuer en deux jours avec 492 fraudes qui est détécté d'un totale de 284807 trasanctions et donc les fraudes représente 0.172% de totale des transactions.
Nous avons mis en place des méthodes usuelles de classification : régression logistic, arbres de décision, boosting, et des réseaux de neurones. Certain modèle peuvent de plus prévoir les probabilité d'appartenance à une classe spécifique tel que la régression logistique tandis que d'autre revoit directement la classe associée.

NOus avons essayé également d'ajuster les données afin de valorisé la classe minoritaire vu que les données ont un carachtaire déséquilibré

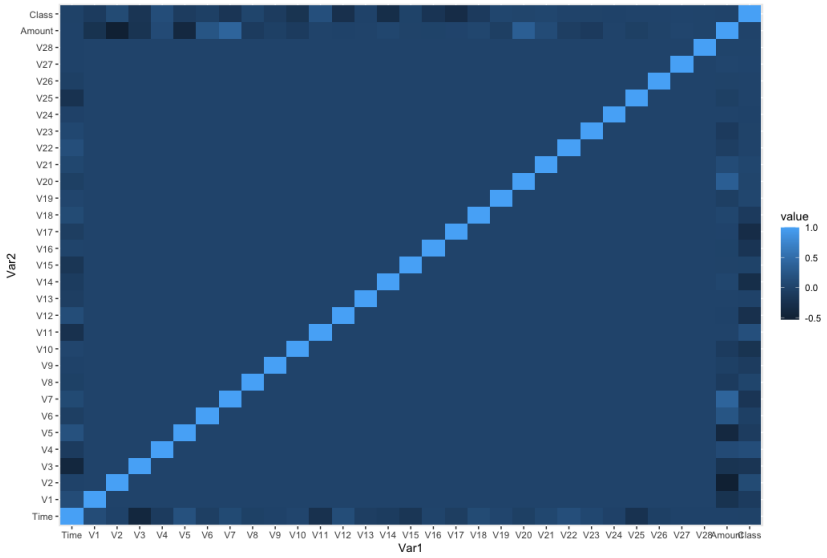
2 Les données :

le jeu de données ne contient que des variables d'entrée numériques qui sont le résultat d'une transformation PCA. Malheureusement, en raison de problèmes de confidentialité, nous ne pouvons pas avoir accées aux fonctionnalités d'origine et plus d'informations générales sur les données. Les fonctionnalités V1, V2,... V28 sont les principaux composants obtenus avec PCA, les seules fonctionnalités qui n'ont pas été transformées avec PCA sont «Time» et «Amount». La fonction «Time» contient les secondes écoulées entre chaque transaction et la première transaction de l'ensemble de données. La fonction «Amount» est le montant de la transaction. La caractéristique «Classe» est la variable de réponse et prend la valeur 1 en cas de fraude et 0 sinon.
L'objet qu'on cherche à prédire est la caractéristique «Class» qui est une variable booléenne, donc c'est un problème de classification.



Montant en fonction de temps

On remarque ici que la plus part des fraudes visent des montants bas.



les variables v1, v2, ..., v28 ne sont pas fortement corrélées les unes aux autres. il y a également une bonne corrélation entre la classe et certains covariables.

3 Choix de la métrique adaptée :

Les données que nous manipulons ici sont déséquilibrées c'est à dire que la classe 0 est majoritairement dominante, elle représente 99.82 % de l'enssemble des données. Ainsi l'utilisation des métriques standard pour évaluer des modèle deviennent peu fiables ou même trompeuses, par exemple si nous considérons un modèle naif qui nous rend toujours 0 quoi que ce soient les covariables et si nous utilisons la métrique de *accuracy rate* :

$$accuracy\ rate\ = \frac{\text{Prédictions correctes}}{\text{Prédictions totales}}$$

.
Ce modèle alors aura une pérformance de 99.82% ce qui est bien sûr trompeur. Donc le choix de la bonne métrique est ici très important.

3.1 Caractéristique de fonctionnement du récepteur (ROC) :

Le ROC est basé sur l'analyse des quantités suivantes pour dériver des caractéristiques qui permettent de classer les modèles :

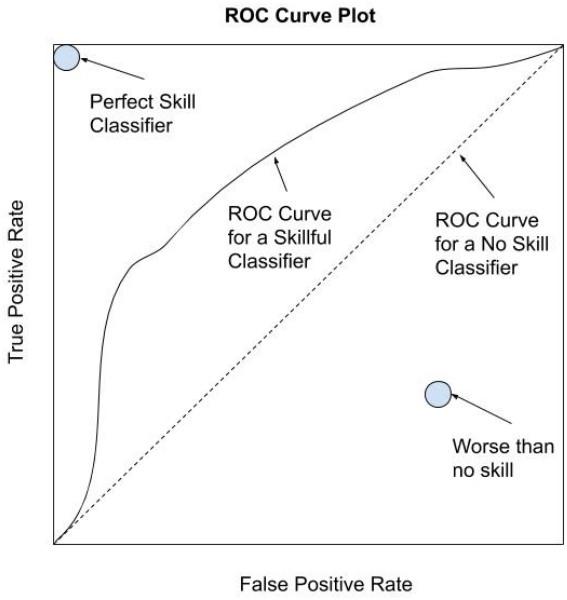
- TP = nombres des prédictions positives vrais.
- FP = nombres des prédictions positives fausses.
- TN = nombres des prédictions négatives vrais.
- FN = nombres des prédictions négatives fausses.

Nous définissons en suite :

- Sensibilité = TPR = $\frac{TP}{TP+FN}$

— Spécificité = TNR = $\frac{TN}{TN+FP}$
— Précision = PPV = $\frac{TP}{TP+FP}$
— 1 - Specificité = FPR = 1-TNR

La courbe ROC est créée en traçant le taux de vrais positifs (TPR) par rapport au taux de faux positifs (FPR) à divers réglages de seuil de classifieur utilisé.



Exemple d'une courbe ROC

un modèle est de plus en plus bon de plus en plus il est en dessus de la courbe diagonale.
une caractéristique qui permet d'évaluer la qualité d'un classifieur est l'AUC "aire sous la courbe ROC" que nous pouvons peut interpréter comme une mesure de la probabilité pour que le modèle classe un exemple positif aléatoire au-dessus d'un exemple négatif aléatoire. UN classifieur parfait aura une AUC égale à 1.

3.2 Métriques de seuil pour la classification déséquilibrée :

Un premier élément qui nous permet d'évaluer un certain modèle est la matrice de confusion :

TP	FP
FN	TN

Un modèle est bon si les valeurs diagonales sont plus grandes que les autres et il est parfait si cette matrice est diagonal. La matrice de confusion permet de mieux comprendre non seulement les performances d'un modèle prédictif, mais également les classes correctement prédites.

On peut aussi calculer la moyenne géométrique :

$$G - mean = \sqrt{\text{sensibilité} \times \text{spécificité}} = \sqrt{\text{TPR} \times \text{TNR}}$$

Ou encore la moyenne harminique :

$$F_\beta = \frac{(2+\beta^2) \times PPV \times TPR}{\beta^2 \times PPV + TPR}$$

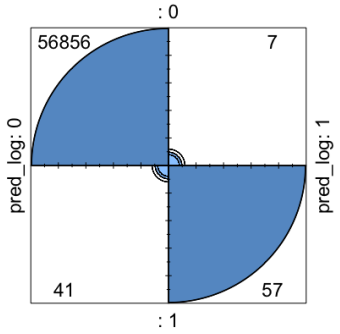
Toutes ces métriques se base sur la prédiction d'une classe en particulier la chose qui nous permet d'éviter les éffets de la classe dominante.

4 modèles et prévision :

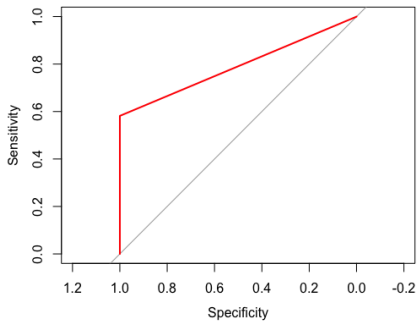
On présente dans cette partie differents modèles que nous avons entrainés sur 80% de nous données aprés tester sur les 20% restant.

4.1 Regression logistique :

La regression linéaire est un modèle que nous utilisons lorsque le résultat attendu est continue. Lorsque on traite un modèle de classification on utilise plutôt la regression logistique. Les valeurs ajustées dans un modèle logistique ne sont pas binaires mais sont plutôt des probabilités représentant la probabilité que le résultat appartienne à l'une des deux catégories. On attribue aprés au résultat la classe qui a la probabilité la plus grande.



La matrice de confusion de regression logistique.



La courbe ROC de modèle regression logistique.

D'aprés la matrice de confusion on remarque que ce modèle prédit les fraude à moitié, 57 fraude détecté et 41 fraud non détecté.

4.2 Arbes de Décision :

Un modèle de prévision intéressant est celui d'arbre de décision. On entrâine notre arbre sur notre ensemble d'apprentissage, et on fait notre prédiction en passant les nouvelles données dans notre modèle.

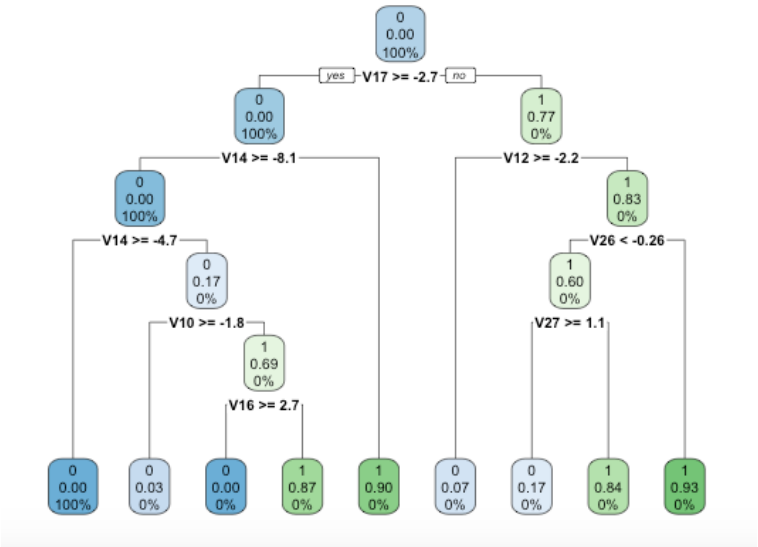
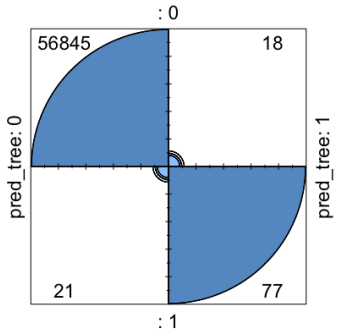
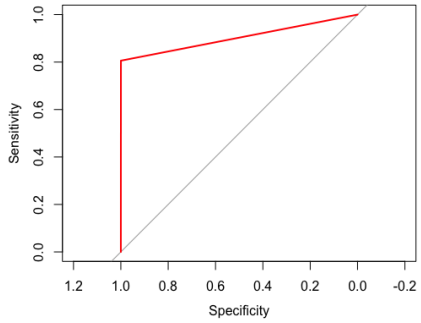


Figure 1 : Arbre de décision sur les données d'entrainement.

On applique ce modéle à la partie test on trouve le résultat suivant :



La matrice de confusion de modèle d'arbre de décision.

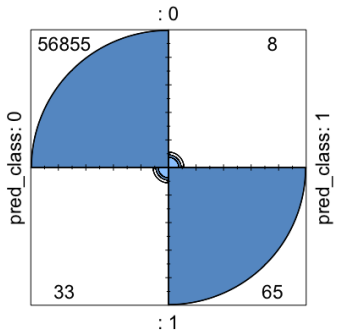


La courbe ROC de modèle d'arbre de décision.

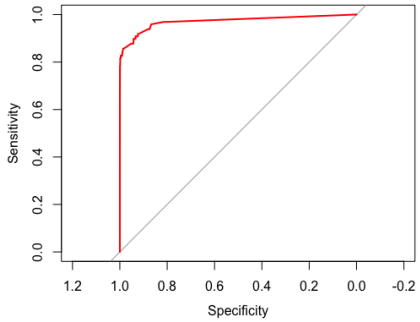
Notre modèle arrive donc à prédire 77 transactions frauduleuses vrais et 58845 transactions non frauduleuses vrais par contre il a classé 21 transations frauduleuses comme non frauduleuses et 18 transactions non frauduleuses comme frauduleuses.
L'air au dessous de la courbe est 0.9028

4.3 Le gradient de boosting :

Nous allons maintenant nous intéresser aux méthodes de boosting. Les algorithmes de boosting sont des méthodes d'ensemble, qui proviennent d'une agrégation séquentielle d'arbres simples.
Nous commençons par utiliser un modèle "Generalized boosted regression".

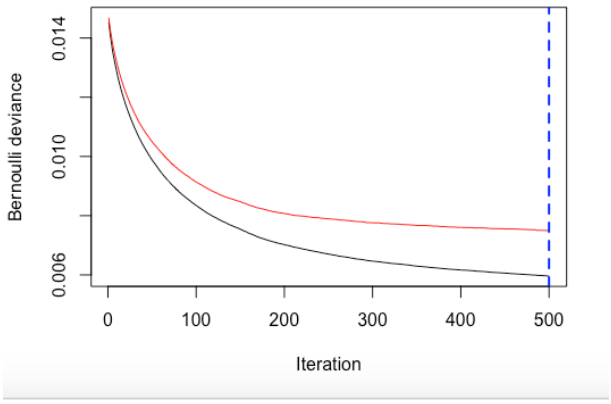


Matrice de confusion de modèle gbm

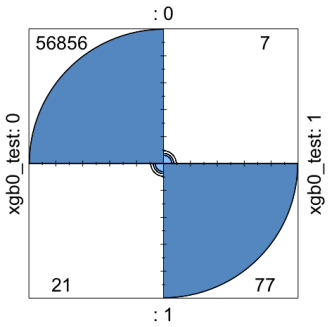


Courbe ROC de modèle gbm

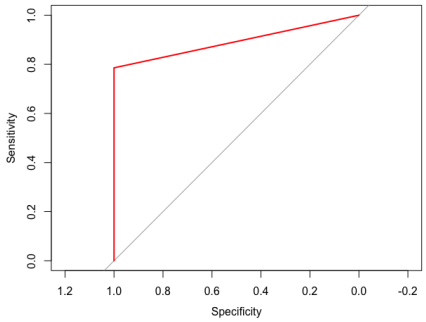
Notre modèle arrive donc à prédire 65 transactions frauduleuses vrais et 56855 transactions non-frauduleuses vrais par contre il a classé 33 transations frauduleuses comme non frauduleuses et 8 transactions non frauduleuses comme frauduleuses.L'air au dessous de la courbe est 0.9717.
L'arbe de décision dépendent de plusieurs paramètresque l'on peut chercher à optimiser dans notre modèle : Ntree ....
On optimise Ntree par la méthode de test pour chercher à réduire notre erreur de prédiction.
On trouve ainsi que Ntree=500



En suite on enchaîne avec le modèle de Boosting xgb.



Matrice de confusion de modèle xgb

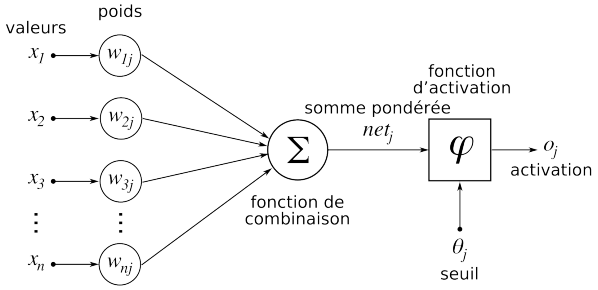


Courbe ROC de modèle xgb

Le modèle xgb arrive donc à prédire 77 transactions frauduleuses vrais et 56856 transactions non-frauduleuses vrais par contre il a classé 21 transations frauduleuses comme non frauduleuses et 7 transactions non frauduleuses comme frauduleuses.L'air au dessous de la courbe est 0.8928.
On constate donc que le modèle xgb est jusqu'a maintenant le mieilleurs des modèles que nous avons testés.

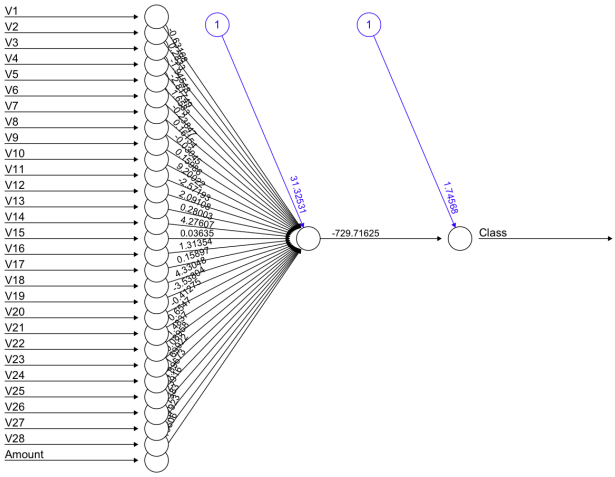
4.4 Réseaux de neurones artificiels :

Les réseaux de neurones artificiels est la méthode la plus populaire en classification et généralement la plus efficace.
Il exists plien de structure des réseaux de neurones les plus basique sont les neurones formels :

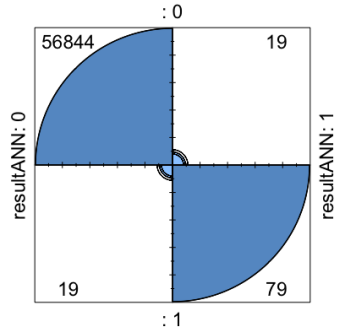


Structure d'un neurone artificiel ou neurone formel.

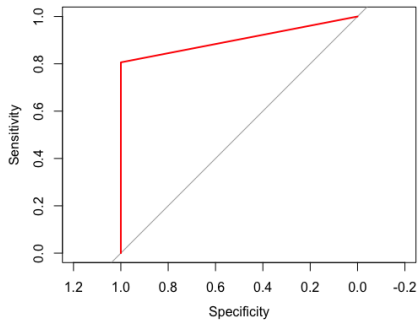
Le neurone calcule la somme de ses entrées x_i, pondérée par les poids synaptiques $w_{i,j}$, puis cette valeur passe à travers la fonction d'activation φ pour produire sa sortie o_j.
La fonction d'activation φ compare la somme de ses entrées x_i, pondérée par les poids synaptiques $w_{i,j}$ avec un seuil θ_j et revoie 0 si cette somme est inférieur au seuil, et 1 au cas contraire.
On applique donc ce modèle à nos données :



réseau de neurones obtenu.

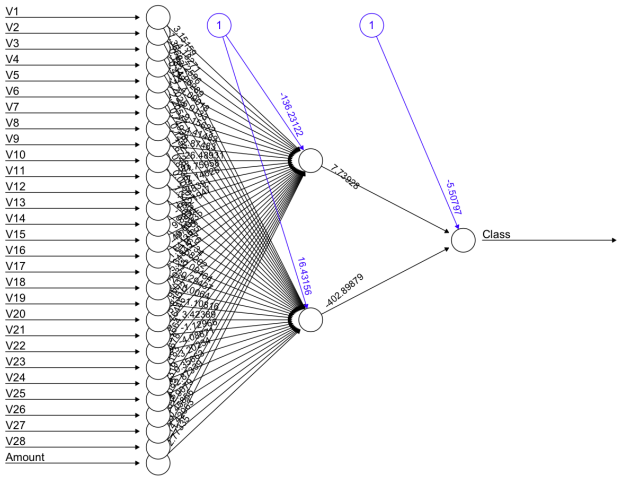


Matrice de confusion de modèle ANN

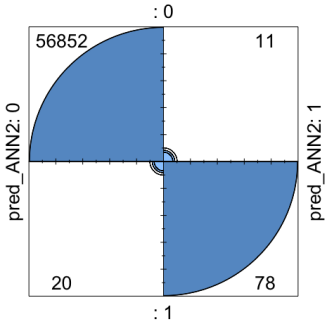


Courbe ROC de modèle ANN

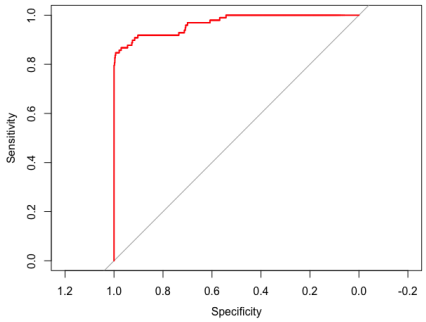
Parmis nos modèles, le modèle ANN est celui qui détécte plus de fraude. on peut également evaluer un réseau de neurones de deux couches cachées, on trouve ansi le résultat :



Réseau de neurones avec deux couches cachées.



Matrice de confusion de modèle ANN2



Courbe ROC de modèle ANN2

Le modèle de réseau de neurones à deux couches cachées améliore plutôt la partie des transactions non-frauduleuses en revanche il commis plus d'érreur sur les transactions frauduleuses.

4.5 Comparaison :

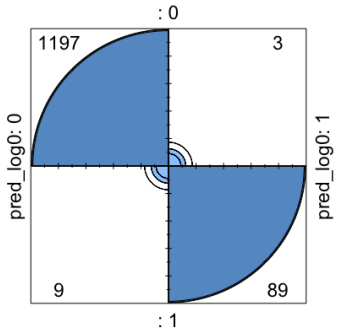
On présente dans ce tableau les differentes mesures de nos modèles.

	TPR	TNR	PPV	G-mean	F_1	AUC
Régression logistique	0.5816	0.99987	0.8906	0.7626	0.7037	0.7908
Decision Tree	0.7897	0.9996	0.8105	0.8862	0.7979	0.9028
GBM	0.6632	0.99985	0.8904	0.8143	0.7602	0.9717
XGB	0.7857	0.99987	0.9166	0.8863	0.8461	0.8920
ANN	0.8061	0.99966	0.8061	0.8976	0.8061	0.9029
ANN2	0.7959	0.999806	0.8764	0.8920	0.8342	0.9676

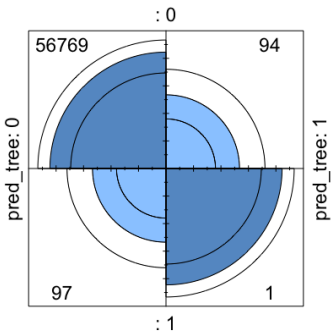
Le choix de modèle lié à nos objectif, par exemple
Si on s'intéresse à la détection des probabilités d'appartenance à une classe on utilise AUC comme crétaire de comparaison. Donc le mielleur modèle est celui de GBM.
Si on s'intéresse à l'existance de fraude ou pas et si bloquer une transaction non-frauduleuse vaut mieux que de laisser passer une transaction frauduleuse, dans ce cas on utilise TPR comme crétaire de comparaison, et donc le modèle de réseau de neurones ANN est le mielleur.
Si laisser passer une transaction frauduleuse vaut mieux que de bloquer une transaction non frauduleuses (ce qui est rare) dans ce cas on utilise TNR comme crétaire de comparaison, et donc on peut choisir soit le modèle de regression logistique ou celui de XGB.
Si on veut bloquer le maximum des fraudes tout en laissant passer le maximum des transactions non-frauduleuses, on utilise dans ce cas G-mean comme crétaire de comparaison, et donc le modèle de réseau de neurones ANN est le plus adapté.

5 ajustement des données

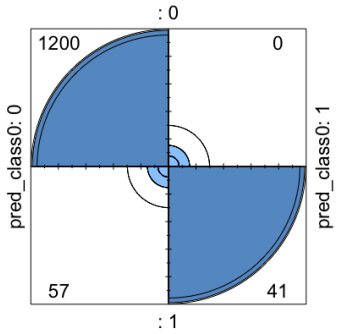
Dans cette section on ajuste nos données en prenant toutes les fraudes qui sont en nombre de 492 ainsi que 6000 des cas non frauduleux et on entraine nos modèles sur cette nouvelle base des données on trouve les matrices de confusion suivantes :



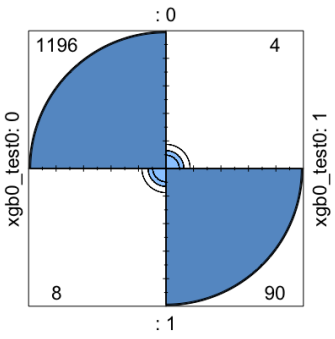
Matrice de confusion de la régression logistique.



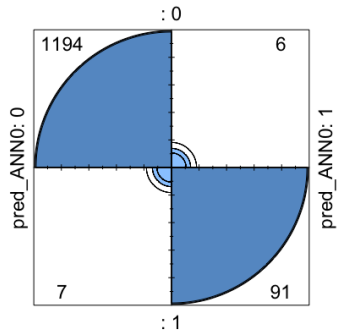
Matrice de confusion d'arbres de décision.



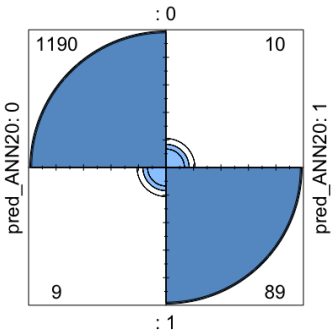
Matrice de confusion de modèle gbm



Matrice de confusion de modèle XGB

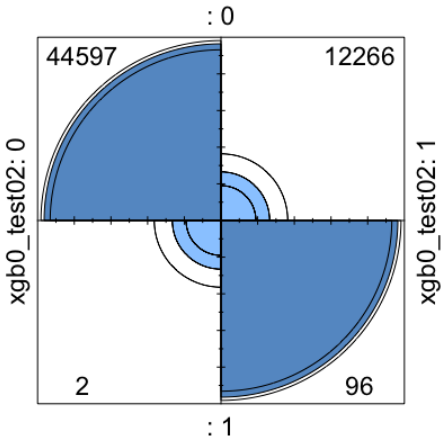


Matrice de confusion de modèle ANN



Matrice de confusion de modèle ANN2

Ces matrices peuvent nous induire en erreur parce que même s'il y a le même nombre de cas fraduleux, l'erreur de modèle peut grandir par rapport à la prévision des transactions non frauduleuses. Par exemple si on test notre modèle de XGB obtenu avec les nouvelle données sur les données de test d'avant on obtient la matrice suivante :



On remarque donc que l'erreur sur les transactions non frauduleuses augmente considérablement. On présente dans ce tableau les differents mesures de nos nouveaux modèles.

	TPR	TNR	PPV	G-mean	F_1	AUC
Régression logistique	0.9081	0.9975	0.9673	0.9517	0.9368	0.9528
Decision Tree	0.0102	0.9983	0.01052	0.1009	0.01036	0.5037
GBM	0.4183	1	1	0.6468	0.58991	0.9702
XGB	0.9183	0.9966	0.9574	0.9567	0.9375	0.9575
ANN	0.9285	0.995	0.9381	0.961212	0.9333	0.9738
ANN2	0.9081	0.9916	0.8989	0.9489	0.9035	0.9841

6 Conclusion.

Notre meilleur modèle n'arrive pas à predire toutes les fraudes cela est dûe la perte d'information lors de la transformation PCA que les données ont subit et également à l'incertitude de modèle. Malgré cela nous avons arrivé à predire 80% des fraude avec le modèle de réseau de norones. Nous avons également essayé de favorisé les cas frauduleux pour mieux prédire les fraude et on a réussie à avoir 92,85% de pérformanc pour le réseau de neurones mais cela augment l'erreur lié au transactions non frauduleuses.