Université Gustave Eiffel

École des Ponts ParisTech

UFR des Mathématiques
M2-Probabilités Et Statistiques Des Nouvelles Données

Apprentissage statistiques et Applications

Djamila AZZOUZ
Alpha Oumar DIALLO
Dorcas BOYO KOUVASI

Gestion des sinistres de BNP Paribas Cardif.

Professeur référent :

Romualad ELIE

Année académique : 2022 - 2023

Table des matières

I Introduction Générale

Le machine learning est une branche de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données sans être explicitement programmés. Au lieu de suivre des instructions strictes, les algorithmes de machine learning utilisent des modèles mathématiques et statistiques pour analyser des données et trouver des schémas.

Il existe plusieurs types de machine learning, y compris l'apprentissage supervisé, non supervisé et par renforcement. Dans l'apprentissage supervisé, un algorithme apprend à partir d'un ensemble de données d'entraînement étiquetées, où les bonnes réponses sont fournies. Dans l'apprentissage non supervisé, les données ne sont pas étiquetées et l'algorithme doit trouver des schémas et des structures par lui-même. L'apprentissage par renforcement est un processus d'essais et d'erreurs, où un algorithme apprend à prendre des décisions en obtenant des récompenses ou des punitions pour ses actions.

Le machine learning est largement utilisé dans de nombreux domaines, y compris la reconnaissance de la parole, la vision par ordinateur, la recommandation de produits, la détection de fraudes et la prédiction des résultats financiers. Les applications de machine learning incluent des systèmes de reconnaissance de la parole tels que Siri et Alexa, des algorithmes de recommandation sur des sites comme Netflix et Amazon, et des outils d'analyse prédictive utilisés dans les entreprises pour améliorer les décisions commerciales.

Le machine learning a le potentiel de résoudre des problèmes complexes et de transformer de nombreux domaines d'activité. Cependant, il est important de comprendre que le machine learning n'est pas une solution miracle et que les algorithmes peuvent être biaisés ou produire des résultats incorrects si les données d'entrée sont biaisées ou si les modèles ne sont pas correctement paramétrés. Il est donc essentiel de mettre en place des processus de vérification et de validation rigoureux pour garantir que les résultats du machine learning soient fiables et pertinents.

II Présentation du sujet

Le projet Kaggle sur la gestion des sinistres BNP Paribas Cardif est une compétition de data science qui vise à améliorer la précision des modèles de prévision des sinistres pour l'entreprise d'assurance BNP Paribas Cardif.

L'objectif de ce projet est d'aider BNP Paribas Cardif à améliorer l'expérience client et à réduire les coûts en prévoyant avec précision les demandes de remboursement d'assurance. Les participants sont invités à construire des modèles de machine learning qui peuvent prédire si une demande de remboursement est susceptible d'être approuvée ou refusée, en utilisant un ensemble de données fourni par BNP Paribas Cardif.

Les jeu de données fournis contiennent des informations sur les sinistres passés, les caractéristiques des assurés, les caractéristiques des contrats d'assurance, ainsi que des variables anonymisées supplémentaires. Les participants devront nettoyer les données, effectuer des analyses exploratoires et développer des modèles de machine learning qui peuvent prédire avec précision si une demande de remboursement sera approuvée ou non.

III Le traitement des données

Le traitement des données est une étape importante dans la mise en place de modèles de machine learning efficaces. Il s'agit d'un processus qui vise à nettoyer et à transformer les données brutes pour les rendre exploitables par les algorithmes de machine learning. Voici quelques explications sur les principales étapes du pré-traitement des données :

- **Nettoyage des données :** Cette étape consiste à détecter et à supprimer les données manquantes, erronées, ou dupliquées dans l'ensemble de données. Il est également important de traiter les valeurs aberrantes (outliers) qui peuvent fausser les résultats.
- **Traitement des données catégorielles :** Les données catégorielles sont des variables qui prennent des valeurs dans un ensemble fini de catégories. Les algorithmes de machine learning ne peuvent pas directement utiliser ces variables, donc il est nécessaire de les encoder en variables numériques. Cela peut être fait en utilisant des techniques telles que l'encodage one-hot ou l'encodage ordinal.
- **Normalisation ou mise à l'échelle des données :** Les données peuvent être mesurées dans des unités différentes, ce qui peut affecter la performance des algorithmes de machine learning. Pour remédier à cela, il est souvent nécessaire de normaliser ou de mettre à l'échelle les données. Par exemple, les données numériques peuvent être transformées en des valeurs normalisées allant de 0 à 1 ou en des valeurs standardisées avec une moyenne de 0 et une variance de 1.
- **Sélection des variables :** Lorsque l'ensemble de données contient de nombreuses variables, il peut être difficile de déterminer celles qui sont les plus importantes pour la prédiction. La sélection des variables peut être réalisée en utilisant des techniques telles que l'analyse en composantes principales (PCA) ou la méthode du Lasso, qui permettent de réduire la dimensionnalité de l'ensemble de données tout en préservant les informations les plus importantes.
- **Division des données :** Enfin, l'ensemble de données est généralement divisé en deux parties : une partie pour l'entraînement du modèle et une partie pour l'évaluation de sa performance. Cette division permet de mesurer la capacité du modèle à généraliser à de nouvelles données qu'il n'a pas vu pendant l'entraînement.

Dans le cadre de notre projet, nous avons décidé d'employer deux méthodes différentes de traitement de données dans le but de mésurer l'impact et l'importance de cette étape lorsque l'on se prépare à utiliser des algorithmes de machine learning.

III.1 Première approche du traitement des données

Dans le cadre de notre projet, nous avons choisi de traiter premièrement les valeurs manquantes sur l'ensemble des valeurs numériques de notre jeu de données avant d'entrer dans la prédiction proprement dite.

III.1.1 Traitement des valeurs manquantes et des variables catégorielles

Le remplacement des valeurs **NaN (Not a Number)** est une étape courante dans le traitement des données manquantes dans les jeux de données. Les valeurs NaN peuvent se produire pour diverses raisons, telles que des données manquantes ou des erreurs dans les mesures, et peuvent affecter les performances des modèles d'apprentissage automatique.
L'une des méthodes courantes pour remplacer les valeurs NaN est de les remplacer par la moyenne des valeurs existantes pour cette variable. Cette méthode est souvent appelée "imputation par la moyenne" ou "imputation par la valeur moyenne". L'idée derrière cette méthode est que la moyenne est une mesure centrale de la distribution des données, et qu'elle peut être utilisée comme une approximation raisonnable des valeurs manquantes.
Voici deux tableaux qui présentent les résultats de ce traitement :

	ID	target	v1	v2	v3	v4
0	3	1	1.335739	8.727474	C	3.921026
1	4	1	NaN	NaN	C	NaN
2	5	1	0.943877	5.310079	C	4.410969
3	6	1	0.797415	8.304757	C	4.225930
4	8	1	NaN	NaN	C	NaN

	ID	target	v1	v2	v3	v4
0	3	1	1.335739	8.727474	110584.0	3.921026
1	4	1	1.630686	7.464411	110584.0	4.145098
2	5	1	0.943877	5.310079	110584.0	4.410969
3	6	1	0.797415	8.304757	110584.0	4.225930
4	8	1	1.630686	7.464411	110584.0	4.145098

FIGURE 1 – TAB1 : Avant le traitement et TAB2 : Après le traitement

Après avoir traité le cas des valeurs numériques, il est nécessaire de porter un regard sur les variables catégorielles. Pour ce faire, nous avons choisi de remplacer les variables catégorielles par leur fréquence d'apparition.
Le remplacement des variables **catégorielles** par la fréquence est une technique courante utilisée pour la prédiction dans le domaine de l'apprentissage automatique. Les variables catégorielles sont des variables qui prennent des valeurs discrètes et non numériques, comme le sexe, la race, la nationalité, le niveau d'éducation, etc. Les modèles d'apprentissage automatique ne peuvent pas traiter directement ces variables catégorielles, ils ont donc besoin d'être convertis en variables numériques avant d'être utilisés. Une méthode courante pour convertir les variables catégorielles en variables numériques consiste à remplacer chaque valeur catégorielle par sa fréquence dans l'ensemble des données. Par exemple, si nous avons une variable catégorielle "couleur de voiture" avec trois valeurs possibles (rouge, vert, bleu) et que nous avons un ensemble de données de 100 voitures, dont 30 sont rouges, 40 sont vertes et 30 sont bleues, nous pouvons remplacer chaque valeur catégorielle par sa fréquence : rouge = 0,3, vert = 0,4, bleu = 0,3.

Cette méthode a l'avantage de conserver l'information sur la fréquence de chaque valeur catégorielle dans l'ensemble des données, ce qui peut être utile pour les modèles qui cherchent à apprendre des relations entre les différentes variables. Deux tableaux explicitant le traitement effectué :

	ID	target	v1	v2	v3	v4
0	3	1	1.335739	8.727474	C	3.921026
1	4	1	NaN	NaN	C	NaN
2	5	1	0.943877	5.310079	C	4.410969
3	6	1	0.797415	8.304757	C	4.225930
4	8	1	NaN	NaN	C	NaN

	ID	target	v1	v2	v3	v4
0	3	1	1.335739	8.727474	110584.0	3.921026
1	4	1	NaN	NaN	110584.0	NaN
2	5	1	0.943877	5.310079	110584.0	4.410969
3	6	1	0.797415	8.304757	110584.0	4.225930
4	8	1	NaN	NaN	110584.0	NaN

FIGURE 2 – TAB1 : Avant le traitement et TAB2 : Après le traitement

Voici de manière brève, un ensemble de graphiques nous permettant de faire une certaine visualisation de données :

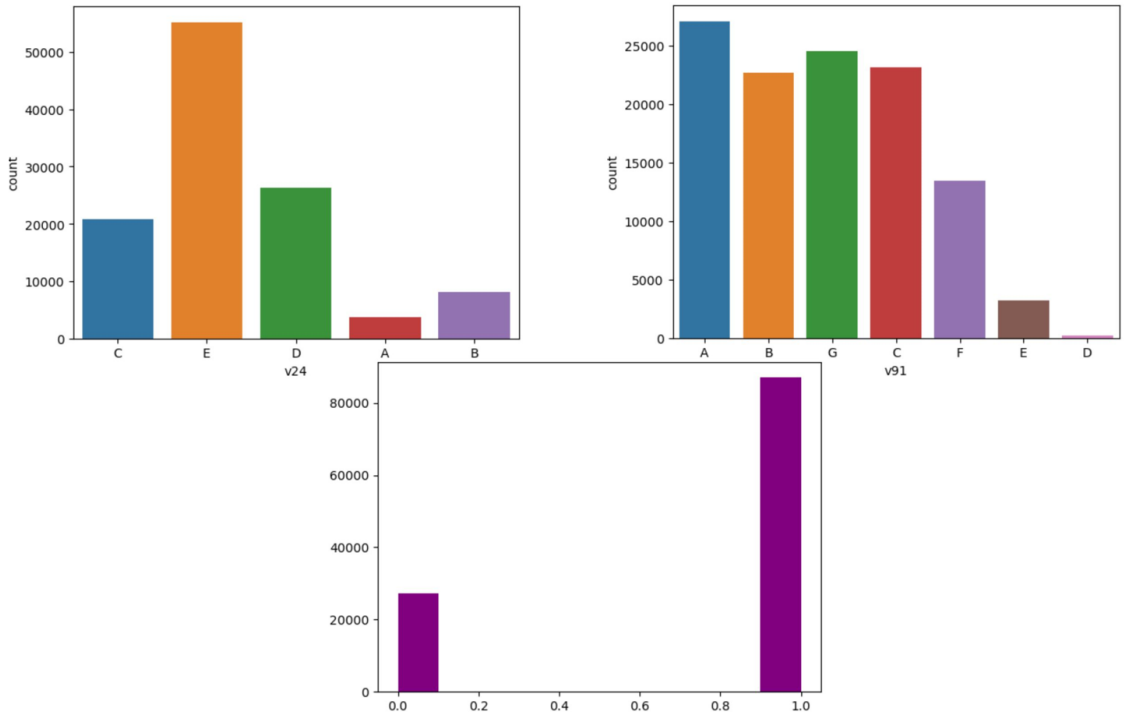


FIGURE 3 – Histogrammes des fréquences des variables catégorielles pour la classe V3 pour chaque client de BNP Paribas cardif et histogramme représentant la proportion des clients nécissitant une intervention

III.1.2 La prédiction avec la première approche

Dans le cadre de ce projet, nous avons choisi d'utiliser deux algorithmes de machine learning : le **Random Forest** et la **Régression logistique**.

Le Random Forest est un algorithme d'apprentissage automatique supervisé utilisé pour la classification, la régression et d'autres tâches de prédiction. Il s'agit d'un ensemble de modèles d'arbres de décision, où chaque arbre est construit sur un échantillon aléatoire de données et utilise un sous-ensemble aléatoire des caractéristiques (variables d'entrée). Le résultat final est obtenu en agrégeant les prédictions de tous les arbres, ce qui permet de réduire le surapprentissage et d'augmenter la précision de la prédiction. Le Random Forest est largement utilisé dans de nombreuses applications telles que la bioinformatique, la reconnaissance de la parole, la vision par ordinateur, la finance, etc.

L'algorithme de régression logistique quant à lui, est une technique d'apprentissage automatique supervisé utilisée pour la classification binaire. Il est utilisé pour prédire la probabilité d'appartenance à une classe binaire (1 ou 0) en utilisant une combinaison linéaire de variables d'entrée. La régression logistique utilise une fonction appelée fonction logistique pour transformer la sortie linéaire en une valeur entre 0 et 1, qui représente la probabilité d'appartenance à la classe positive. La fonction d'erreur utilisée pour entraîner l'algorithme est la fonction de coût de la log-vraisemblance négative.Il est largement utilisé dans de nombreuses applications telles que la détection de spam, la classification de patients malades ou non, etc. Il est nécessaire de préciser donc que le but poursuivi est celui de comparer les performances de ces deux modèles.

Observations :

— Les résultats obtenus sont tels que le modèle de Random Forest a un score de 0.78 contre un score de 0.76 pour la régression logistique. Ceci nous permet de conclure que le Random Forest est de 2 pourcent plus précis dans la prédiction que la régression logistique et donc un peu plus performant que ce dernier. Voici un tableau récapitulatif des résultats obtenus en termes de probabilités de prédictions et de précision dans la prédiction pour chaque modèle :

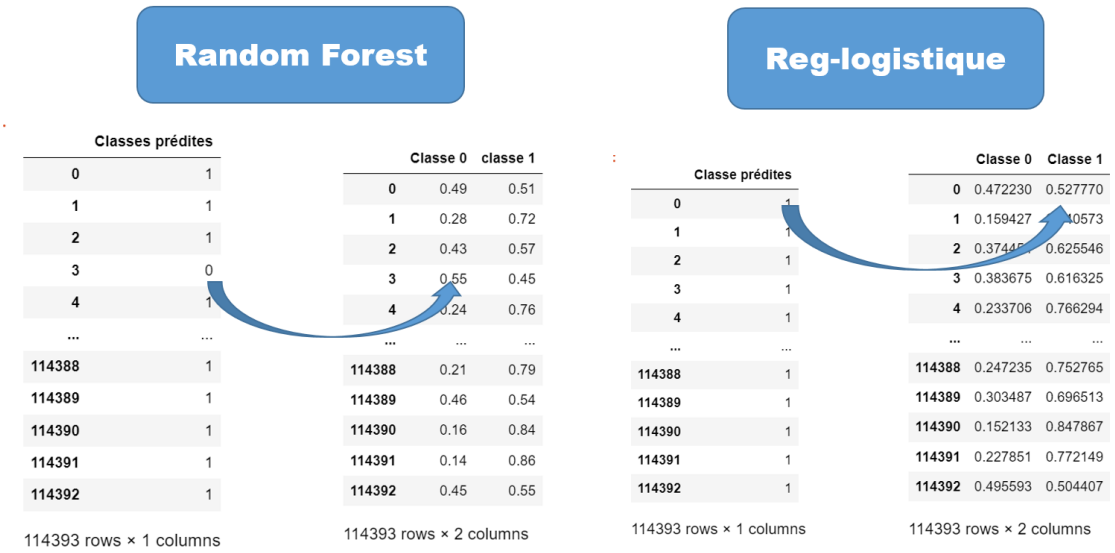


FIGURE 4 – Probabilités et classes prédites pour chaque client

— **Une manière de comprendre la prédiction :** Le client ID = 3 n'aura malheureusement pas droit à une intervention accélérée de la part de BNP car sa probabilité d'appartenir à la classe 0 (i.e classe de ceux qui rentrent dans le critère d'accélération) est supérieur à celle d'appartenir à la classe 1 dans le cadre de l'algorithme de Random Forest. Le même principe de lecture s'applique aussi pour le cas de la régression logistique .

— L'image ci-dessous représente les densités de probabilités prédites pour chaque modèle :

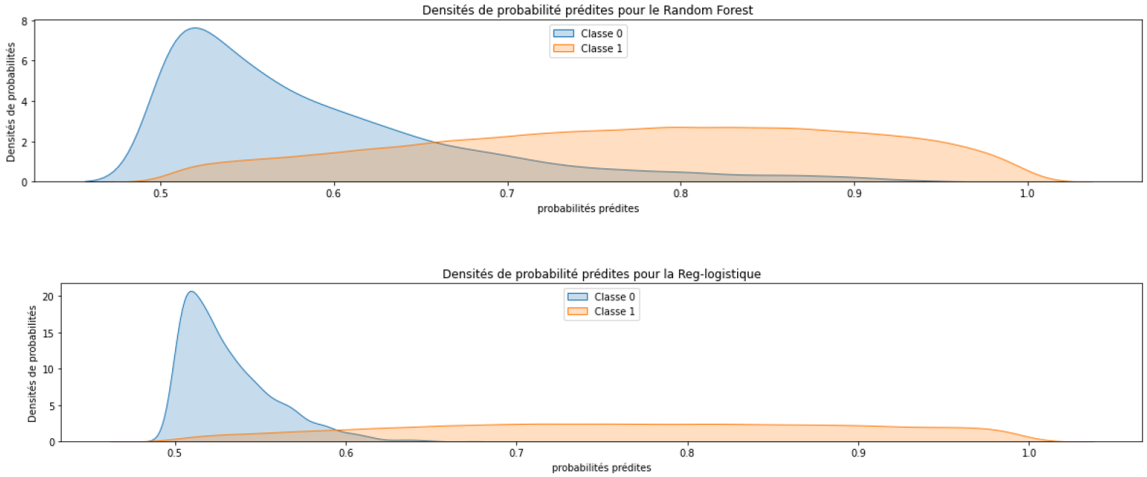


FIGURE 5 – Probabilités et classes prédites pour chaque client

Notre analyse dans le cadre de cette première approche nous a permise d'une part de choisir un premier modèle de traitement de données et ensuite de tester les performances des 2 modèles que nous avons décidé d'utiliser. Il nous a semblé quand même assez utile de chercher à voir si un autre mode de traitement de donées auraient un impact sur la prédiction de nos 2 modèles. Cette question fera l'objet de la prochaine section.

III.2 Deuxième approche du traitement des données :

Dans cette deuxième partie de notre étude, nous allons chercher à améliorer les performances de nos modèles en adoptant une approche différente pour traiter les données. Concrètement, nous allons prendre en compte les valeurs manquantes et les variables catégorielles d'une manière différente de ce que nous avons fait dans la première partie. Nous espérons ainsi pouvoir évaluer l'impact de ces changements sur la qualité de nos prédictions. Par ailleurs, nous allons mettre en œuvre la méthode de sélection de variables afin de voir si nous pouvons améliorer les performances de nos modèles en éliminant les variables les moins pertinentes. Enfin, nous allons comparer les résultats obtenus à l'issue de cette deuxième partie avec ceux obtenus dans la première partie pour en tirer des conclusions pertinentes sur les approches à privilégier pour obtenir des prédictions précises et fiables.

III.2.1 Traitement des valeurs manquantes et catégorielles :

Dans le cadre de cette deuxième approche du traitement de données, nous avons décidé d'utiliser la technique d'encodage de label pour traiter les variables **"catégorielles"**. Cette technique consiste à remplacer chaque valeur unique d'une variable catégorielle par une étiquette numérique unique. Ces étiquettes sont attribuées en fonction de l'ordre d'apparition des valeurs uniques dans la variable catégorielle, sans tenir compte de leur signification ou de leur ordre intrinsèque. Ensuite, nous allons procéder au traitement des valeurs manquantes en remplaçant les **"NAN"** de chaque colonne par la médiane de la colonne concernée. Cette méthode est couramment utilisée pour remplacer les valeurs manquantes dans les ensembles de données numériques, car elle permet de minimiser l'impact des valeurs aberrantes sur la médiane. Nous espérons que ces deux techniques nous permettront d'obtenir des données plus fiables et plus cohérentes pour notre étude, ce qui devrait se traduire par une amélioration des performances de nos modèles de machine learning.
Les résultats de ce traitement est présenté sous la figure suivante :

	ID	target	v1	v2	v3	v4	v5	v6	v7	v8	...	v122	v123	v124	v125
0	3	1	1.335739	8.727474	C	3.921026	7.915266	2.599278	3.176895	0.012941	...	8.000000	1.989780	0.035754	AU
1	4	1	NaN	NaN	C	NaN	9.191265	NaN	NaN	2.301630	...	NaN	NaN	0.598896	AF
2	5	1	0.943877	5.310079	C	4.410969	5.326159	3.979592	3.928571	0.019645	...	9.333333	2.477596	0.013452	AE
3	6	1	0.797415	8.304757	C	4.225930	11.627438	2.097700	1.987549	0.171947	...	7.018256	1.812795	0.002267	
4	8		NaN	NaN	C	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	
5	9	0	NaN	NaN	C	NaN	8.856791	NaN	NaN	0.359993	...	NaN	NaN	0.049861	
6	12	0	0.899806	7.312995	C	3.494148	9.946200	1.926070	1.770427	0.066251	...	3.476299	1.992594	0.083758	
7	21	1	NaN	NaN	C	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	BY
8	22	0	2.078651	8.462619	NaN	3.739030	5.265636	1.573033	2.303371	0.015869	...	8.148148	1.875560	0.018659	S
9	23	1	1.144802	5.880606	C	3.244469	9.538384	2.500001	1.559405	0.412610	...	7.325843	4.896617	0.008944	E

	ID	target	v1	v2	v3	v4	v5	v6	v7	v8	...	v122	v123	v124	v125
0	3	1	1.335739	8.727474	2	3.921026	7.915266	2.599278	3.176895	0.012941	...	8.000000	1.989780	0.035754	21
1	4	1	1.469550	7.023803	2	4.205991	9.191265	2.412790	2.452166	2.301630	...	6.749117	2.739239	0.598896	6
2	5	1	0.943877	5.310079	2	4.410969	5.326159	3.979592	3.928571	0.019645	...	9.333333	2.477596	0.013452	5
3	6	1	0.797415	8.304757	2	4.225930	11.627438	2.097700	1.987549	0.171947	...	7.018256	1.812795	0.002267	
4	8		1.469550	7.023803	2	4.205991	8.670867	2.412790	2.452166	0.386032	...	6.749117	2.739239	0.139864	
5	9	0	1.469550	7.023803	2	4.205991	8.856791	2.412790	2.452166	0.359993	...	6.749117	2.739239	0.049861	
6	12	0	0.899806	7.312995	2	3.494148	9.946200	1.926070	1.770427	0.066251	...	3.476299	1.992594	0.083758	37
7	21	1	1.469550	7.023803	2	4.205991	8.670867	2.412790	2.452166	0.386032	...	6.749117	2.739239	0.139864	52
8	22	0	2.078651	8.462619	3	3.739030	5.265636	1.573033	2.303371	0.015869	...	8.148148	1.875560	0.018659	82
9	23	1	1.144802	5.880606	2	3.244469	9.538384	2.500001	1.559405	0.412610	...	7.325843	4.896617	0.008944	68

FIGURE 6 – Traitement des valeurs manquantes et catégorielles

— Après avoir appliqué les techniques de traitement des données mentionnées précédemment, nous allons utiliser les mêmes algorithmes que dans la première partie, à savoir le **Random Forest** et la **Régression logistique**. Nous allons examiner les probabilités prédites pour chaque client de BNP Paribas Cardif, ainsi que leur classe. Enfin, nous allons observer les performances de ces modèles, afin de déterminer si les modifications apportées à la manière dont nous avons traité les données ont un impact significatif sur la qualité de nos prédictions.

Observation :

En comparant les deux scores obtenus, on remarque que le modèle Random Forest a une précision de 78%, tandis que la précision du modèle de Régression Logistique est de 77%. Cela signifie que le modèle Random Forest a prédit correctement la classe cible de 78% des observations de test, alors que le modèle de Régression Logistique a eu une précision de 77%. Ainsi, le modèle Random Forest est légèrement plus performant que le modèle de Régression Logistique dans la prédiction des nouvelles données dans notre cas d'étude.

— Pour confirmer cela, nous pouvons afficher les matrices de confusion qui sont utilisées pour évaluer les performances des modèles de classification avec deux classes ou plus. Dans le cas binaire, tel que celui que nous avons, la matrice de confusion est un tableau à quatre valeurs qui représente les différentes combinaisons de valeurs réelles et prédites, comme illustré dans le tableau ci-dessous.

	Réel 0	Réel 1
Prédit 0	1120	4327
Prédit 1	675	16743

TABLE 1 – Matrice de confusion pour le modèle Random Forest

Observations :

Les deux tableaux de confusion présentent les résultats de deux modèles de classification binaire utilisés pour prédire si un client de BNP Paribas Cardif pourrait bénéficier d'une approbation pour accélérer la gestion de ses sinistres. Les colonnes des tableaux représentent les classes réelles et les lignes représentent les prédictions du modèle. Les cases vertes indiquent des prédictions correctes (True Positives et True Negatives), tandis que les cases rouges indiquent

	Réel 0	Réel 1
Prédit 0	1165	4282
Prédit 1	832	16580

TABLE 2 – Matrice de confusion pour le modèle Régression logistique

des prédictions incorrectes (False Positives et False Negatives). Le modèle Random Forest a correctement prédit 1120 clients sans approbation (True Negatives) mais a eu 675 fausses prédictions, tandis que le modèle Régression logistique a correctement prédit 1165 True Negatives mais a eu 832 fausses prédictions. Le modèle Random Forest a également correctement prédit 16743 clients avec approbation (True Positives), mais a eu 4327 fausses prédictions, tandis que le modèle Régression logistique a correctement prédit 16580 True Positives mais a eu 4282 fausses prédictions.

— Nous pourrons aussi observer les classes et les probabilités prédites pour chaque client de BNP Paribas cardif présentés dans la figure suivante :

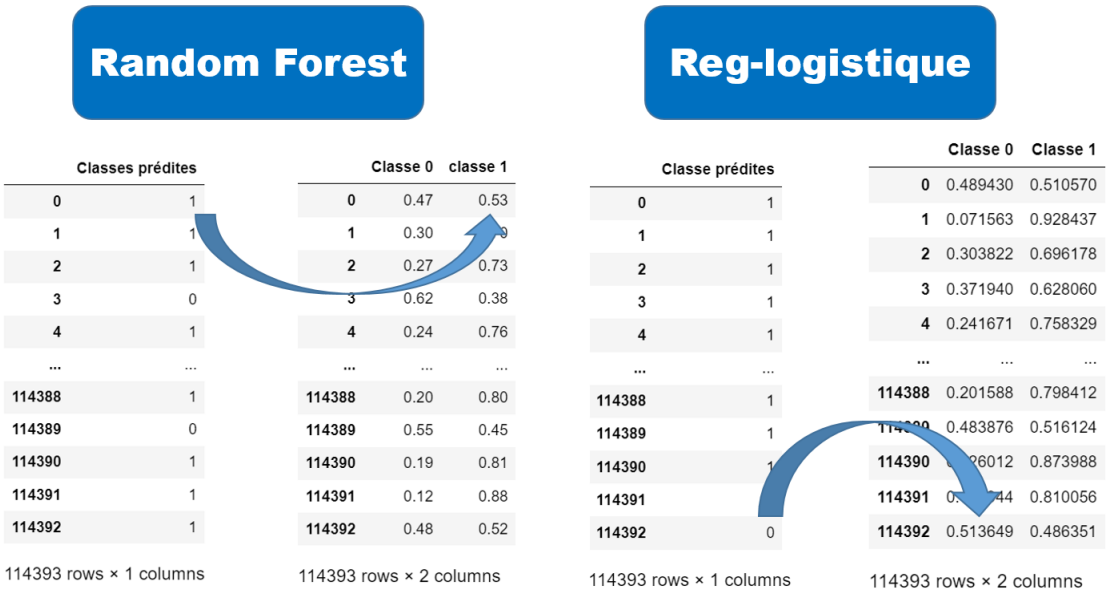


FIGURE 7 – Probabilités et classes prédites pour chaque client de BNP Paribas cardif

Remarque :

— En observant, les résultats, nous comprenons bien comment nos deux algorithmes ont pu classés chaque client. En effet, prenant par exemple le client de $ID = 0$, Le Random Forest a estimé que nous pourrons accélérer la gestion de ses sinistres car la probabilité de lui accorder **ie classe 1** est supérieure à celle de ne pas lui accorder c'est pourquoi l'algorithme nous a retourné **1**. Pareil pour le client de $ID = 0$, la régression logistique a estimée que nous pourrons pas lui accorder l'accélération de la gestion de ses sinistres car la probabilité de ne pas lui accorder est supérieure à celle de lui accorder.

III.3 Sélection de variables avec Lasso :

La régression avec Lasso (Least Absolute Shrinkage and Selection Operator) est une technique de régression linéaire qui permet de sélectionner les variables les plus importantes pour prédire une variable cible. L'objectif de la méthode Lasso est de minimiser la somme des carrés des résidus, tout en limitant la somme des valeurs absolues des coefficients à une valeur donnée. Cela signifie que certains coefficients de régression peuvent être réduits à zéro, ce qui permet de sélectionner les variables les plus pertinentes pour la prédiction.

La méthode Lasso est particulièrement utile lorsque le nombre de variables explicatives est important et que certaines de ces variables sont peu corrélées avec la variable cible. Elle permet de réduire la dimensionnalité des données en éliminant les variables moins importantes, ce qui peut améliorer la précision des prévisions et faciliter l'interprétation des résultats.

Lorsque les variables explicatives les plus importantes sont sélectionnées, nous utiliserons le Random Forest uniquement sur ces variables pour avoir les prédictions et la matrice de confusion représentées sous les figures ci dessus :

	Réel 0	Réel 1
Prédit 0	1165	4282
Prédit 1	838	16580

TABLE 3 – Matrice de confusion pour le modèle Random Forest

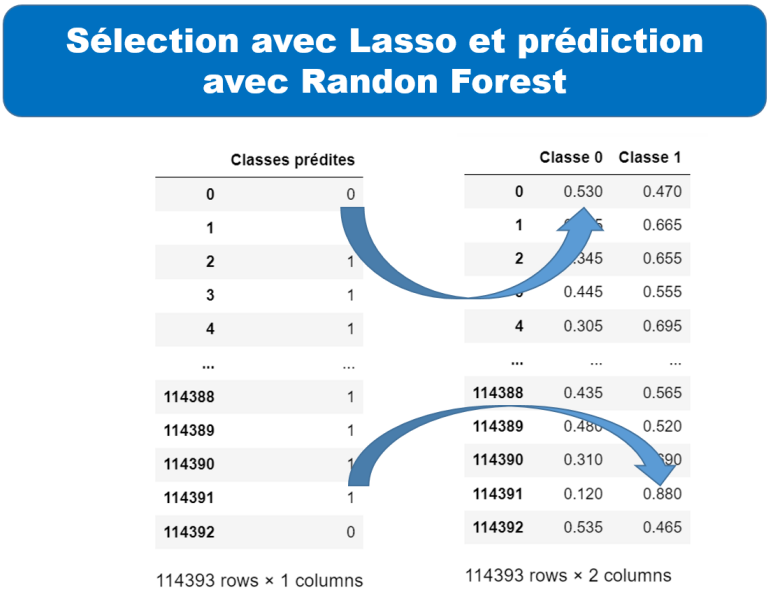


FIGURE 8 – Résultats de la prédiction après la sélection des variables explicatives les plus importantes avec Lasso

— Avec cette méthode que nous avons utilisé, nous obtenons un score de 0.78 similaire au score du Random Forest sans sélection de variables et une matrice de confusion similaire à celle de la régression logistique, nous constatons qu'ici dans notre cas nous n'avons pas pue améliorer la performance de notre modèle.

— Afin d'évaluer plus précisément les performances des trois algorithmes, nous utilisons la courbe ROC, qui représente une méthode d'évaluation pour les modèles de classification binaire. Cette courbe permet de visualiser les performances du modèle en termes de sensibilité et de spécificité pour différentes valeurs de seuils de décision.

— l'air sous la courbe ROC des trois modèles sont présentés dans le tableau suivant :

	Air sous la courbe ROC
Random Forest	0.78
Reg-logistique	0.73
Random Forest avec lasso	0.73

Les courbes ROC des trois modèles sont représentées dans la figure suivante :

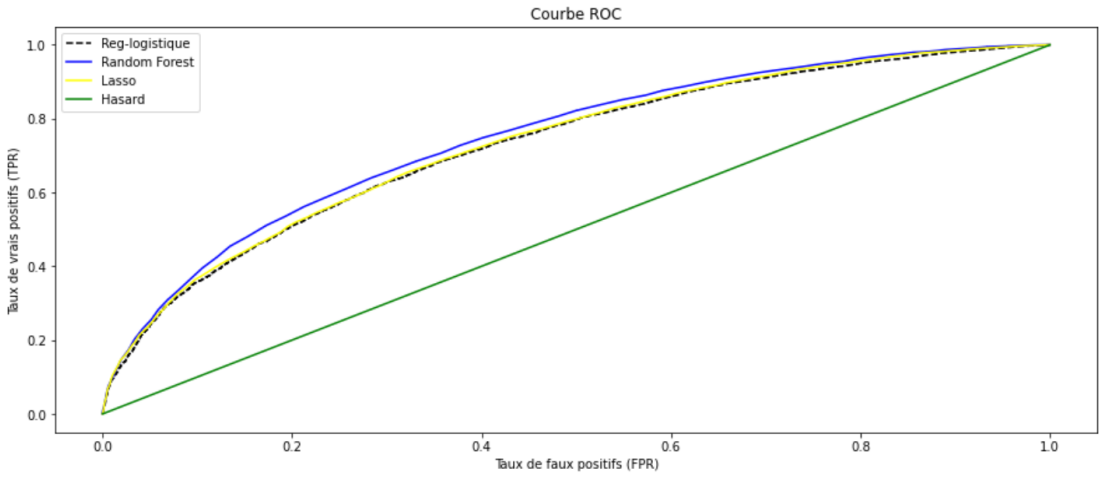


FIGURE 9 – Courbes ROC pour chaque modèle

Observation :

En utilisant l'aire sous la courbe, nous pouvons évaluer la qualité du modèle sélectionné pour la classification. Nous constatons que l'aire sous la courbe de la fonction Random Forest est supérieure à celles obtenues avec la régression logistique et le Random Forest avec Lasso. Cette différence indique que le Random Forest est un meilleur classificateur que les deux autres modèles. La courbe présentée en vert représente la classification aléatoire. Ainsi, plus la courbe est éloignée de cette ligne, meilleure est la classification et donc les performances du modèle.

— Nous avons également les densités de probabilités prédites représentées dans l'image ci dessus :

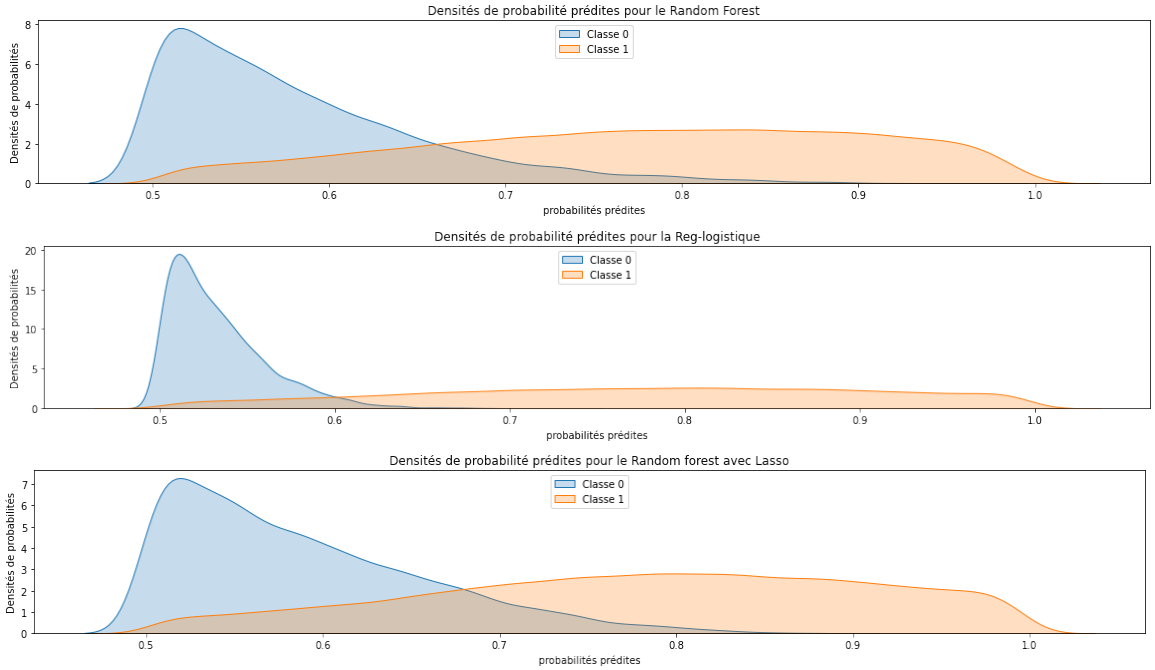


FIGURE 10 – Densités de probabilités prédites de chaque modèle

— Le Random Forest a réussi à prédire beaucoup plus de classe 1 avec moins d'erreurs que la classe 0, où il a commis de nombreuses erreurs à partir de la probabilité de 0,5. En revanche, la régression logistique a prédit beaucoup plus la classe 0 avec moins d'erreurs que la classe 1. Après avoir utilisé Lasso pour sélectionner les variables, le Random Forest s'est amélioré légèrement par rapport au Random Forest sans sélection de variables. Ces résultats sont cohérents avec les tables de confusion. Étant donné que la classe positive "1" est associée à l'approbation accélérée du processus de gestion des sinistres des clients de BNP Paribas, nous avons choisi le Random Forest comme modèle adapté à notre problème.

Observation générale :

Dans cette partie, nous avons effectué plusieurs tests pour évaluer l'impact de différentes approches de prétraitement de données sur les performances de nos modèles. Nous avons constaté que malgré la mise en œuvre d'une nouvelle méthode de prétraitement de données, il n'y avait pas d'amélioration significative dans les performances de nos modèles.

De plus, nous avons également testé la sélection de variables avec la régression Lasso. Cependant, même avec cette approche, nous n'avons pas réussi à améliorer les performances du modèle Random Forest.

Malgré cela, nous avons constaté que dans notre cas d'étude, l'algorithme Random Forest a obtenu les meilleures performances parmi les modèles testés. Cet algorithme a permis une meilleure classification des clients de BNP Paribas Cardif.

Enfin, malgré nos efforts pour améliorer les performances de nos modèles, nous n'avons pas réussi à surpasser les performances du modèle Random Forest dans notre cas d'étude.

IV Le Deep learning dans tout ça :

Aux limites des algorithmes classiques de machine learning , a vu le jour les techniques de deep learning. Une méthode révolutionnaire qui est aujourd'hui utilisée dans l'apprentissage automatique. Partant de la reconnaissance visuelle ou vocale jusqu'au pilotage automatique d'un véhicule , sans oublier la prédiction ou détection d'une maladie, le deep learning touche à tout.

Les techniques d'apprentissage utilisées reposent sur les réseaux de neurones. A l'image des neurones biologiques, l'analyse se fait par entrée et sortie. Un neurone (biologique) reçoit plusieurs informations (neurotransmetteurs) , au niveau de ses dendrites. Lorsque la quantité d'information dépasse un certain seuil,le neurone s'active envoi un courant électrique dans son axone ce qui lui permet d'émettre à son tour des neurotransmetteurs via les synapses.

Techniquement un neurone artificiel est une application non linéaire qui en ses paramètres , reçoit un vecteur d'entrée x et renvoie une sortie $f(x)$. Le i-ième neurone artificiel f_i est définit par :

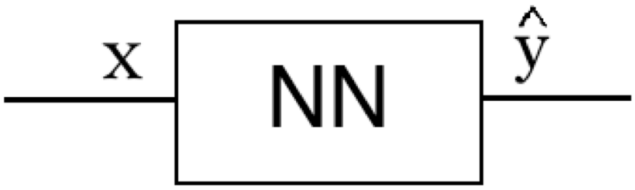
$$f_i(x) = \phi(< w_i, x > +b_i)$$

Avec :

$$< w_i, x >= \sum_{r=1}^{p} w_i^r x^r$$

- Les w_i pondèrent les variables d'entrées $(x^1, x^2, ..., x^p)$
- b_i définit le biais du neurone i
- ϕ est appelée **fonction d'activation**

IV.1 Exemple d'un neurone simple

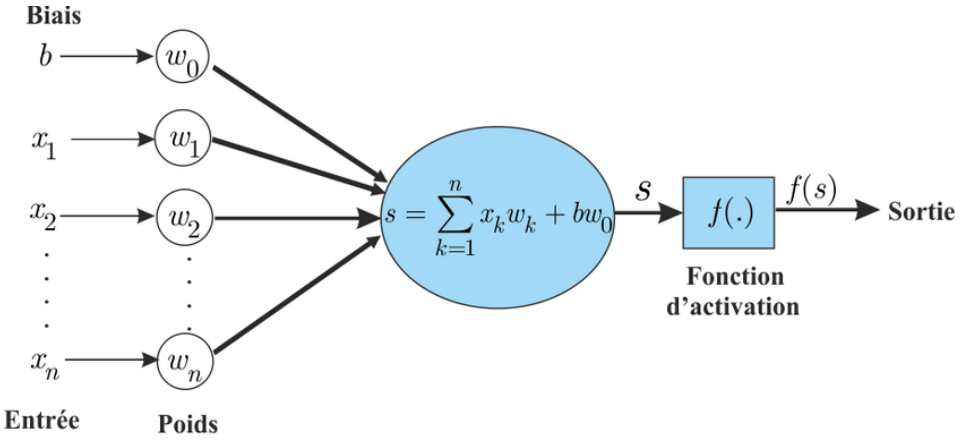


Ici nous avons l'exemple d'un neurone simple qui reçoit une entrée x et qui une fois entraînée , nous sort la prédiction $\hat{y}$

IV.2 Réseaux de neurones

Définition 4.1. Un réseau de neurones est composé de nombreuses unités de traitement appelées "neurones" interconnectées et organisées en couches. Chaque neurone reçoit des entrées, effectue des calculs à l'aide de poids et de fonctions d'activation, puis transmet sa sortie à d'autres neurones.

L'apprentissage se produit en ajustant les poids du réseau pour minimiser une fonction de coût, qui mesure l'écart entre les sorties du réseau et les valeurs désirées. Cela se fait généralement à l'aide de techniques d'optimisation telles que la rétro-propagation du gradient.



Commentaires : Içi typiquement, nous avons un exemple de réseau de neurones , avec ses entrées, le biais , le poids , la fonction d'activation et la sortie.

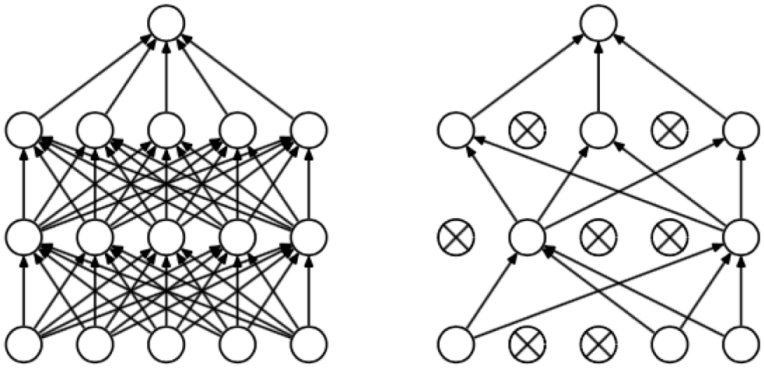
Toutes les neurones artificiels fonctionnent de la sorte , donc il est pratiquement simple d'implanter un algorithme de réseaux de neurones. Mais un détail important vient sur les données d'entrées et leurs pré-traitements.

Nous n'allons pas rentrer trop dans les détails théoriques du deep learning. Nous savons qu'il est capable de faire de la prédiction , de la classification , de la reconnaissance d'images et vocale ... . En ce qui nous concerne le problème de **Gestion de sinistres chez BNP** est un problème de **classification**. Nous allons donc voir comment mettre en place un algorithme de réseau de neurones pour un problème de **classification**.

De façon générale , pour mettre en place un algorithme de Deep learning , nous devons suivre quelques étapes importantes (nous pouvons appeler cela la recette à suivre) :

- **Pré-traitement des données :** cela nous permets d'éviter les erreurs lors de l'entraînement ou alors faire un mauvais apprentissage, nous pensons que c'est l'une des étapes les plus importantes de machine learning en générale.
- **l'initialisation du modèle :** une autre étape cruciale , celle çi nous permets de bien démarrer et s'assurer que notre alogo converge , en effet si les **poids** sont très grands ou trop petits alors les neurones risquent de saturer et leurs gradients ne seront pas dans la zone linéaire.
- **La régularisation :** Pour éviter qu'il y est des disproportions de poids , il est utile de rajouter un critère de régularisation , ceci permet d'éviter **l'overfitting** i.e empêcher que le modèle soit efficace sur des nouvelles données non connues. La régularisation de type L2 (norme) est souvent celle qui est utilisée avec le coefficient $\lambda = 10^{-2}$.

  Il existe une autre technique pour éviter le **l'overfitting** , il s'agit de la technique du **dropout** , en effet cette méthode permet d'isoler ou d'omettre aléatoirement certains neurones. Pour cela il suffit de créer un vecteur avec la même taille que la couche de neurones ayant des variables issues de la loi de **Bernouilli** (avec un coefficient à choisir) et le multiplier élément par élément.



Commentaires : Ici nous avons un exemple d'utilisation de la technique du dropout, au début nous avons (à gauche) une couche de départ intacte puis après le **dropout**, 7 neurones ont été omis avec un coefficient de **Bernouilli** de 0.5.

- **Utilisation de la techniques des auto-encodeurs :** Les autoencodeurs sont une technique courante utilisée pour apprendre des représentations comprimées de données brutes, ce qui permet de réduire la dimensionnalité des données tout en conservant les informations importantes. Les autoencodeurs sont composés de deux parties principales : l'encodeur et le décodeur. L'encodeur prend en entrée les données brutes et produit une représentation comprimée de ces données, tandis que le décodeur prend cette représentation comprimée en entrée et produit une reconstruction des données brutes. L'objectif de l'autoencodeur est de minimiser la différence entre les données d'entrée et la sortie du décodeur.
- **Utilisation du PCA :** le PCA (Principal Component Analysis) peut être utilisé pour extraire les caractéristiques les plus importantes des données en entrée, ce qui peut aider à mieux comprendre les données et à améliorer les performances du modèle. En effet, en identifiant les caractéristiques les plus importantes, on peut mieux comprendre les facteurs qui influencent les prédictions du modèle et ainsi améliorer la qualité des résultats. Elle permet aussi de réduire fortement le temps de training. Il est important de noter que cela ne devrait être fait que si les données ont une corrélation linéaire.

Remarque : Ici nous avons établis quelques étapes non exhaustives à suivre pour mettre en place un algorithme de deep learning , il existe d'autres techniques qui peuvent être plus ou moins performantes selon la problématique.

IV.3 Application sur notre problématique de classification :

L'utilisation d'un réseau de neurones pour ce problème de classification dans cette gestion de sinistres chez BNP est relativement un cas simple. Nous allons donc appliquer les étapes précédentes à fin d'arriver au résultat escompté.

Mise en place et entraînement du modèle :

- Nous utiliserons le pré-traitement des données fait dans la première partie de ce rapport.
- Ici nous allons utiliser le **modèle séquentiel** car il est adapté pour les problèmes de classification binaire, où la sortie du modèle est une probabilité comprise entre 0 et 1, indiquant la classe à laquelle chaque exemple appartient. La dernière couche du modèle est une couche de sortie avec une fonction d'activation **sigmoïde** qui transforme la sortie en une probabilité. Enfin, la fonction de perte utilisée est la perte binaire de **cross-entropy** (binary cross-entropy loss), qui est une fonction de perte commune pour les problèmes de classification binaire.
- Nous avons ajouté une couche Dense au modèle avec 10 neurones, et une fonction d'activation ReLU. Cette couche est utilisée pour extraire des caractéristiques à partir des données d'entrée. Puis nous avons ajouté une couche Dense avec 1 neurone (sortie binaire). (**Nous n'avons pas utiliser le PCA car nos données ne sont pas fortement linéairement corrélées**) dans notre cas. Enfin, nous avons ajouté une couche de sortie avec 1 neurone et une fonction d'activation sigmoïde pour prédire la probabilité d'appartenance à la classe positive.
- Ainsi nous entraînons le modèle en ajustant les poids des neurones à partir des données d'apprentissage normalisées en utilisant un nombre fixe d'epochs (50) et une taille de lot (batchsize) de 50, ce qui signifie que le modèle sera ajusté après avoir traité chaque lot de 50 observations.

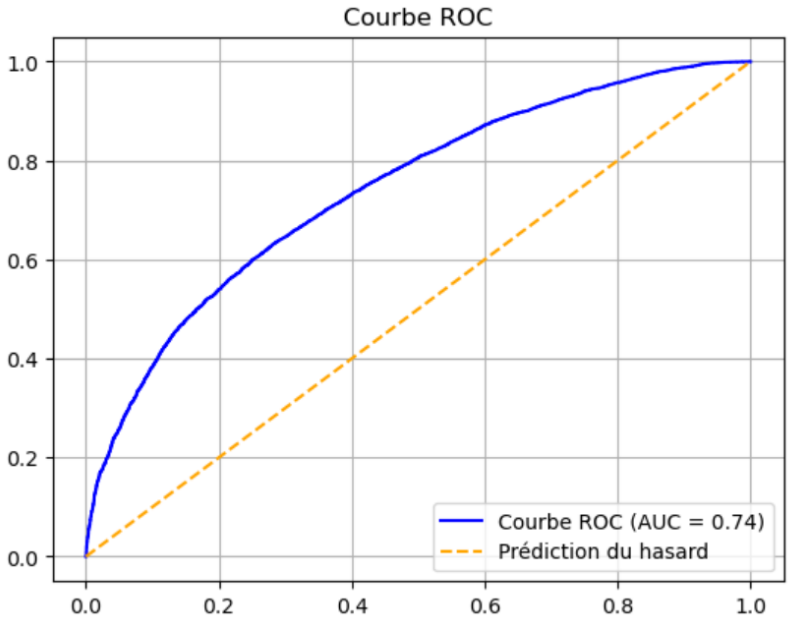
Observons la performance du modèle :

Score du modèle :

```
715/715 [==============================] - 1s 1ms/step - loss: 0.4780 - accuracy: 0.7770
score du modèle : 77.70%
```

Une précision (accuracy) de 78% signifie que le modèle de classification binaire a prédit correctement la classe de 78% des observations dans l'ensemble de test. Cela signifie que le modèle a correctement classé 78% des échantillons et a mal classé les 22% restants.

Courbe ROC :



Observation : Cette courbe indique que le modèle a une performance modérée en termes de séparation des classes positives et négatives. Cela signifie que le modèle est capable de distinguer les deux classes, mais il peut y avoir une certaine confusion entre les deux, ce qui conduit à un certain nombre de faux positifs et de faux négatifs. En général, un score de courbe ROC supérieur à 0,5 est considéré comme un indicateur de performance supérieure à celle d'un classificateur aléatoire.

Synthèse : Nous avons réussi à mettre en place l'algorithme de réseau de neurones qui nous permettra de classifier correctement (avec une certaine erreur) les clients de BNP selon s'il nécessite une intervention accélérée (classe 1) ou non (classe 0). Sa performance s'aligne avec celle du **Random forest**.

Notons que nous avons mis en place un réseau de neurones pas très complexe, nous pouvons sûrement ajuster le modèle en jouant sur les hyperparamètres à fin d'optimiser le résultat , et nous sommes convaincus que la performance peut nettement être amélioré avec d'autres techniques de prétraitement des données et/ou l'ajout de techniques de modélisation.

Observons la prédiction sur les données :

	Result
0	0
1	0
2	0
3	0
4	0
...	...
114388	0
114389	0
114390	0
114391	1
114392	0

114393 rows × 1 columns

Commentaires : Sur des données que l'algorithme n'avait jamais vu , il nous prédit donc les clients pour lesquelles une intervention accélérée est nécessaire et dans ce cas il appartient à la classe **1** et l'inverse dans la classe **0**. Ici Nous voyons que les clients comme 1 et 2 ne nécessitent pas une intervention accélérée contrairement au client 114391. Évidement que P(client 1 $\in$ calsse 0) $>$ P(client 1 $\in$ calsse 1) raison pour laquelle l'algorithme a fait cette prédiction et ceci est valabe pour les autres.

V Conclusion

Le machine learning et le deep learning sont devenus de nos jours des outils indispensables dans tous les métiers. En ce qui concerne ce projet de BNP Paribas, ça nous a permis de faire une prédiction pour savoir quels clients appartiennent à quelle classe et avec quelle probabilité. Le choix des algorithmes a été totalement personnel et cela découle de l'analyse statistique que nous avons fait sur les données. A travers deux modélisations différentes les meilleures modèles sont le Random forest et le réseau de neurones. C'était un peu ce qui est escompté , mais le score reste quand même améliorable et aussi limiter les fautes de classement. Nous avons utilisé un modèle de réseau de neurones plutôt simple sachant qu'il existe d'énormes possibles d'améliorations.Quant aux algorithmes nous avions également d'autres choix comme les **arbres de décision** qui sont des modèles qui utilisent une série de questions pour classer les données en fonction de leurs caractéristiques. Mais aussi un puissant modèle peu populaire mais très efficace , il s'agit des **SVM** qui sont des modèles qui trouvent une frontière de décision optimale entre les données en utilisant des techniques mathématiques avancées. Ils sont particulièrement utiles pour les tâches de classification.Les SVM cherchent à trouver l'hyperplan qui sépare les classes avec la marge maximale. Si les classes ne sont pas linéairement séparables, les SVM peuvent utiliser une technique appelée "kernel trick" pour projeter les données dans un espace de plus grande dimension où les classes peuvent être séparées linéairement.

Références

— Cours d'apprentissage statistique de Monsieur Roumuald Elie.
— Cours de statistique en grande dimension de Monsieur Hebiri et Denis.
— Cours d'architecture de Big Data de Monsieur Roland TROSIC.

[1] Code python du projet.
[2] Machine Learnia