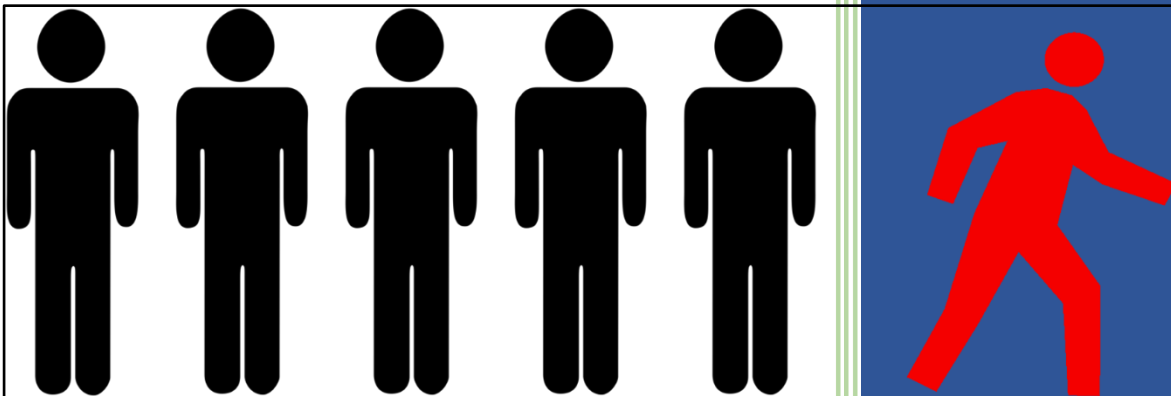


# Projet Machine Learning : Customer Churn



Sous la supervision du :  
Dr Lotfi NCIB

Présenté par :

**AMORRI Houssein  
BENNEJI Mohamed  
DRISS Mohammed Ahmed  
GAYAP Hadrien  
LONTCHI TABOUA Freddy**

# **Rapport du Projet Machine Learning Customer Churn**

Présenté par

**AMORRI Houssein  
BENNEJI Mohamed  
DRISS MohammedAhmed  
GAYAP Hadrien  
LONTCHI TABOUA Freddy**

**Elèves Ingénieurs en 4<sup>ème</sup> année Data Science à ESPRIT**

Sous la supervision du

**Dr NCBI Lotfi**

# SOMMAIRE

RESUME .....	IV
INTRODUCTION .....	1
I.    COMPRÉHENSION DES DONNÉES .....	2
A.    LES COLONNES .....	3
B.    EXPLORATION DES DONNÉES .....	3
II.   LA CONSTRUCTION DU DATA HUB.....	8
A.    LE TYPE DE VARIABLES.....	8
B.    AJOUT DE VARIABLES .....	8
C.    SUPPRESSION DE VARIABLES.....	9
D.    NORMALISATION DES DONNEES.....	9
III.   MODELISATION ET OPTIMISATION .....	10
A.    DÉFINITIONS DES TERMES CLÉS.....	10
B.    MODÉLISATION DE QUELQUES ALGORITHMES .....	12
IV.    EVALUATION DES MODÈLES .....	25
V.    MISE EN PRODUCTION .....	26
CONCLUSION .....	27
REFERENCES.....	28

# TABLEAUX

TABLEAU 1 : EXEMPLE DE MATRICE DE CONFUSION .....	10
TABLEAU 2 : COMPARAISON DES ALGORITHMES.....	25

## **FIGURES**

FIGURE 1 : RÉSUMÉ DE LA BASE DE DONNÉES .....	2
FIGURE 2 : RÉPARTITION DES SEXES .....	3
FIGURE 3 : RÉPARTITION EN ÂGE.....	4
FIGURE 4 : CLIENTS AVEC/SOUS PERSONNES À CHARGE SELON QU'ILS ONT OU NON UN PARTENAIRE.....	4
FIGURE 5 : NOMBRE DE CLIENTS PAR DURÉE D'ABONNEMENTS .....	5
FIGURE 6 : NOMBRE DE CLIENTS PAR TYPE DE CONTRAT .....	5
FIGURE 7 : RÉPARTITION DES DIFFÉRENTS SERVICES .....	6
FIGURE 8 : DÉSABONNEMENT PAR CATÉGORIE D'ÂGE.....	7
FIGURE 9 : DÉSABONNEMENT ET CHARGE MENSUELLE.....	7
FIGURE 10 : EXEMPLE DE COURBE ROC .....	11
FIGURE 11 : FORMULE DE L'ALGORITHME NAÏVES BAYES.....	12
FIGURE 12 : SCORES DES ALGORITHMES DE NAÏVE BAYES .....	12
FIGURE 13 : CLASSIFICATION REPORT NAÏVE BAYES .....	13
FIGURE 14 : MATRICE DE CONFUSION NAÏVE BAYES .....	13
FIGURE 15 : FONCTIONNEMENT DU RANDOM FOREST CLASSIFIER .....	14
FIGURE 16 : CLASSIFICATION REPORT RANDOM FOREST CLASSIFIER .....	14
FIGURE 17 : MATRICE DE CONFUSION RANDOM FOREST CLASSIFIER .....	15
FIGURE 18 : RÉSULTATS DE LA RÉGRESSION LINÉAIRE .....	15
FIGURE 19 : MATRICE DE CONFUSION RÉGRESSION LOGISTIQUE .....	16
FIGURE 20 : CLASSIFICATION REPORT RÉGRESSION LOGISTIQUE .....	16
FIGURE 21 : PRINCIPE DU GRADIENT BOOSTING.....	17
FIGURE 22 : COURBE ROC XGB ET GRADIENT BOOSTING .....	18
FIGURE 23 : MATRICE DE CONFUSION GRADIENTBOOST .....	19
FIGURE 24 : CLASSIFICATION REPORT GRADIENT BOOST.....	20
FIGURE 25 : LEARNING CURVE GRADIENT BOOST.....	20
FIGURE 26 : CLASSIFICATION REPORT AVEC LA FONCTION DE DÉCISION .....	21
FIGURE 27 : MEILLEURE VARIABLES POUR GRADIENT BOOST .....	22
FIGURE 28 : CLASSIFICATION REPORT XGBOOST.....	22
FIGURE 29 : LEARNING CURVE XGBOOST .....	23
FIGURE 30 : MATRICE DE CONFUSION DE LGBCLASSIFIER.....	24
FIGURE 31 : RAPPORT DE CLASSIFICATION LGBM .....	24
FIGURE 32 : MATRICE DE CONFUSION DES MEILLEURES ALGORITHMES (LGB ET GB) .....	25
FIGURE 33 : FORMULAIRE DE PRÉDICTION L'APPLICATION DJANGO.....	26
FIGURE 34 : PRÉDICTION DE L'APPLICATION .....	26

# RESUME

L'ère du numérique a vu naître les réseaux de télécommunications dont les entreprises du secteur se comptent aujourd'hui en milliers. Un des soucis majeurs pour ces entreprises est le problème de désabonnement des clients plus connus sur le nom de « **Customer Churn** ». Ces entreprises, sans avoir été prévenues du départ de leur client, constatent leur désabonnement dû au fait de la grande concurrence, ce qui implique une diminution considérable de bénéfice pour cette entreprise. Le problème se précise dans la mesure où le coût pour ces opérateurs d'avoir de nouveaux utilisateurs est plus élevé que le coût de préserver leurs clients actuels et c'est pour cela qu'ils ont pour objectif de garder leurs clientèles. Il a été question pour nous au cours de ce projet, à l'aide d'anciennes données fournies par une de ces entreprises, de déployer un algorithme pouvant prédire quel client est susceptible de quitter l'entreprise à l'aide des techniques de Machine Learning. Plusieurs algorithmes ont été pour cela déployés et les algorithmes qui ont donnés les meilleures prédictions appartiennent à la famille de Gradient Boosting. Le meilleur (LGBMClassifier) nous a donné un pourcentage de 98% de prédiction. L'algorithme a donc été déployé dans une application pour servir et faire valoir ce que de droit.

**Mots Clés :** Customer Churn, Prédiction, Algorithme , Machine Learning

# Introduction

La science des données depuis sa naissance ne cesse de démontrer sa place dans notre quotidien, que ce soit envers de individus ou des personnes morales, comme les entreprises. Il est d'un très grand apport particulièrement dans le domaine de télécommunications pour résoudre le problème de désabonnement de client (de l'anglais « **Customer Churn** »).

Afin d'apporter une aide significative à ce problème, plusieurs méthodes d'approche de résolutions d'un problème Data Science nous sont proposés. Nous avons fait du choix de la méthode « **CRISP-DM** » mise au point par IBM et qui « *reste aujourd'hui la seule méthode utilisable efficacement pour tous les projets Data Science...* Cette méthode est agile et itérative, c'est-à-dire que chaque itération apporte de la connaissance métier supplémentaire qui permet de mieux aborder l'itération suivante »<sup>1</sup>

Cette méthode a donc été scrupuleusement respectée : **de la compréhension métiers au déploiement en passant par la compréhension, la préparation, la modélisation et l'évaluation des données.**

Durant ce projet nous utiliserons plusieurs classificateurs de tous types pour trouver la meilleure façon de prédiction de « churn » possible. De ces types on peut citer les méthodes statistiques, d'arbres de décisions ainsi que d'autres ensembles de méthodes notamment celle qui étudie les gradients.

---

<sup>1</sup> <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/> 09/01/21  
01 :38

# I. Compréhension des données

Dataset Shape: (7043, 33)

	Nom	dtypes	Manquants	Uniques	Premiere Valeur	Deuxieme Valeur	Avant derniere Valeur	Derniere Valeur	Entropy
0	CustomerId	object	0	7043	3668-QPYBK	9305-CDSKC	4801-JZAZL	3186-AJIEK	3.85
1	Count	int64	0	1	1	1	1	1	0.00
2	Country	object	0	1	United States	United States	United States	United States	0.00
3	State	object	0	1	California	California	California	California	0.00
4	City	object	0	1129	Los Angeles	Los Angeles	Angelus Oaks	Apple Valley	2.86
5	Zip Code	int64	0	1652	90003	90006	92305	92308	3.22
6	Lat Long	object	0	1652	33.964131, -118.272783	34.048013, -118.293953	34.1678, -116.86433	34.424926, -117.184503	3.22
7	Latitude	float64	0	1652	33.9641	34.048	34.1678	34.4249	3.22
8	Longitude	float64	0	1651	-118.273	-118.294	-116.864	-117.185	3.22
9	Gender	object	0	2	Male	Female	Female	Male	0.30
10	Senior Citizen	object	0	2	No	No	No	No	0.19
11	Partner	object	0	2	No	No	Yes	No	0.30
12	Dependents	object	0	2	No	Yes	Yes	No	0.23
13	Tenure Months	int64	0	73	2	8	11	66	1.78
14	Phone Service	object	0	2	Yes	Yes	No	Yes	0.14
15	Multiple Lines	object	0	3	No	Yes	No phone service	No	0.41
16	Internet Service	object	0	3	DSL	Fiber optic	DSL	Fiber optic	0.46
17	Online Security	object	0	3	Yes	No	Yes	Yes	0.45
18	Online Backup	object	0	3	Yes	No	No	No	0.46
19	Device Protection	object	0	3	No	Yes	No	Yes	0.46
20	Tech Support	object	0	3	No	No	No	Yes	0.45
21	Streaming TV	object	0	3	No	Yes	No	Yes	0.46
22	Streaming Movies	object	0	3	No	Yes	No	Yes	0.46
23	Contract	object	0	3	Month-to-month	Month-to-month	Month-to-month	Two year	0.43
24	Paperless Billing	object	0	2	Yes	Yes	Yes	Yes	0.29
25	Payment Method	object	0	4	Mailed check	Electronic check	Electronic check	Bank transfer (automatic)	0.59
26	Monthly Charges	float64	0	1585	53.85	99.65	29.6	105.65	3.02
27	Total Charges	object	0	6531	108.15	820.5	346.45	6844.5	3.80
28	Churn Label	object	0	2	Yes	Yes	No	No	0.25
29	Churn Value	int64	0	2	1	1	0	0	0.25
30	Churn Score	int64	0	85	86	86	59	38	1.89
31	CLTV	int64	0	3438	3239	5372	2793	5097	3.47
32	Churn Reason	object	5174	20	Competitor made better offer	Moved	NaN	NaN	1.22

Figure 1 : Résumé de la Base de Données

La figure ci-dessus nous permet d'avoir un aperçu rapide de la base de données.



## A. Les colonnes

Il serait bon pour nous afin de mieux travailler, de comprendre l'apport de chaque colonne de notre dataset :

Les clients qui sont partis au cours du dernier mois - la colonne s'appelle **Churn**.

- Services auxquels chaque client a souscrit :
  - Phone
  - multiple lines
  - internet
  - online security
  - online backup
  - device protection
  - tech support
  - streaming TV et movies
- Informations sur le compte du client :
  - Depuis combien de temps il est client
  - contract
  - payment method
  - paperless billing
  - monthly charges et total charges :
- Informations démographiques sur les clients
  - Gender
  - age range
  - Dependent
  - Partner

## B. Exploration des données

Toujours dans l'optique de bien comprendre notre dataset, une exploration sur les données est faite : en fonction de la démographie, le compte des clients, les services et sur les raisons de départ.

### 1. Démographie

L'analyse démographique a laissé voir que :

- Environ la moitié de notre Dataset est constituée d'hommes et l'autre moitié de femmes.

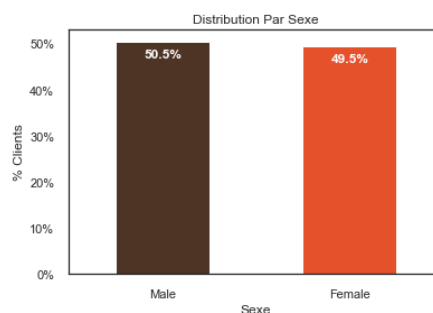


Figure 2 : Répartition des sexes

- 16,2% des clients est très âgée, donc la plupart des personnes de notre dataset est jeune

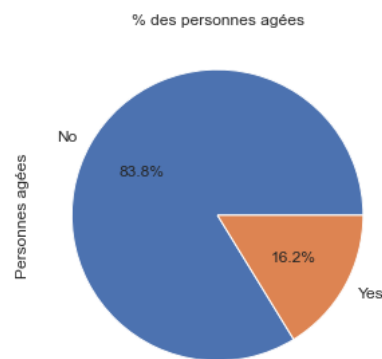


Figure 3 : Répartition en âge

- La grande majorité (92%) des personnes n'ayant pas de partenaires n'ont pas de personnes à charge
- 40% des personnes ayant un partenaire a une personne à charge

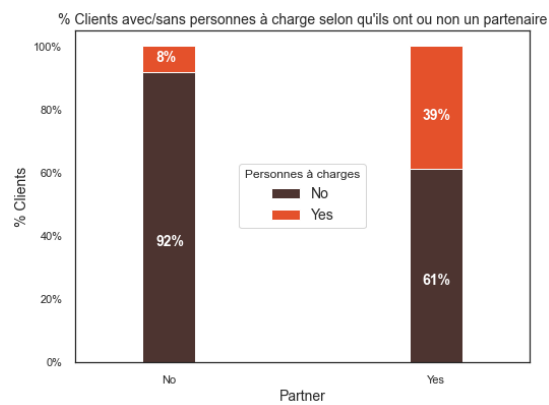


Figure 4 : Clients avec/sous personnes à charge selon qu'ils ont ou non un partenaire

## 2. Comptes clients

Il est question ici de voir le compte général des clients en fonction de leur type d'abonnement (en temps) :

- Un grand nombre (>1200) de clients vient de s'abonner à la société
- Un grand nombre (entre 800 et 100) de clients s'est abonné à la société il ya environ 70 mois

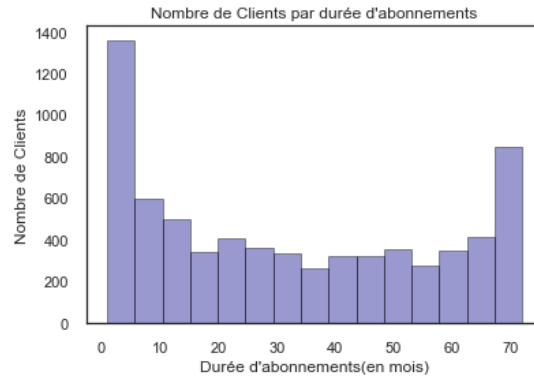


Figure 5 : Nombre de clients par durée d'abonnements

- La grande majorité des clients a un contrat de paie mensuel
- Il y a un nombre presque égal entre les abonnés à contrat annuel et biennuel  
Cela peut s'expliquer à cause du grand nombre de service. Et donc il y a des services intéressants qui retiennent les clients et d'autres non

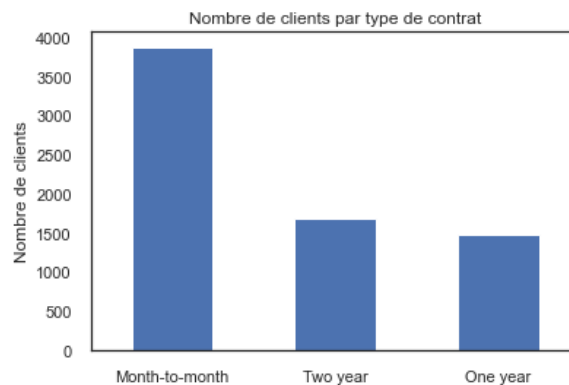


Figure 6 : Nombre de clients par Type de contrat

### 3. Les services

La figure suivante donne un récapitulatif des différents services :

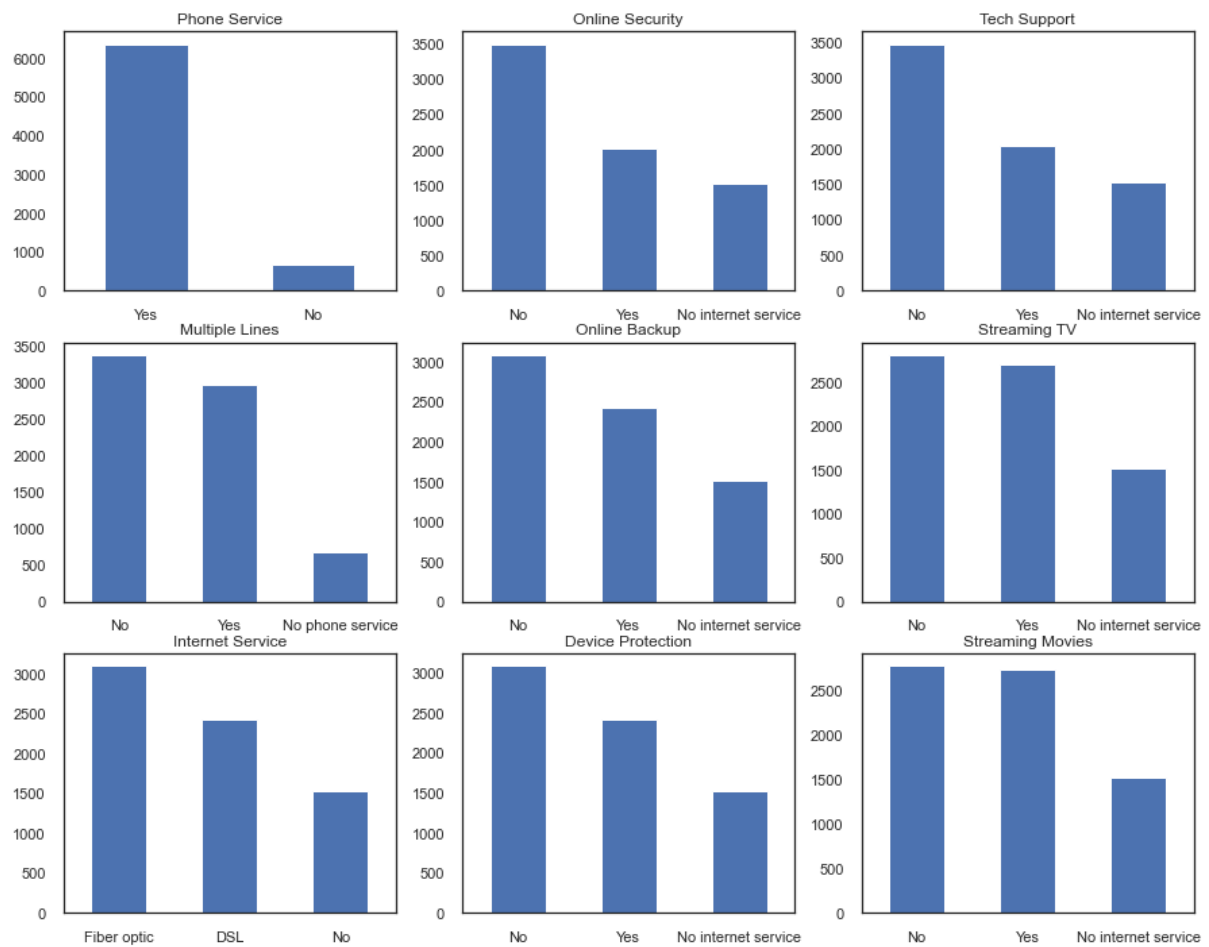


Figure 7 : Répartition des différents services

On peut ainsi faire une liste des services les plus utilisées :

- Phones services
- Streaming Service
- Streaming Movies
- Internet (Fibre optique)

#### 4. L'étiquette (Churn)

Il est question de voir dans cette partie la relation qui lit les désabonnements a l'ensemble des données.

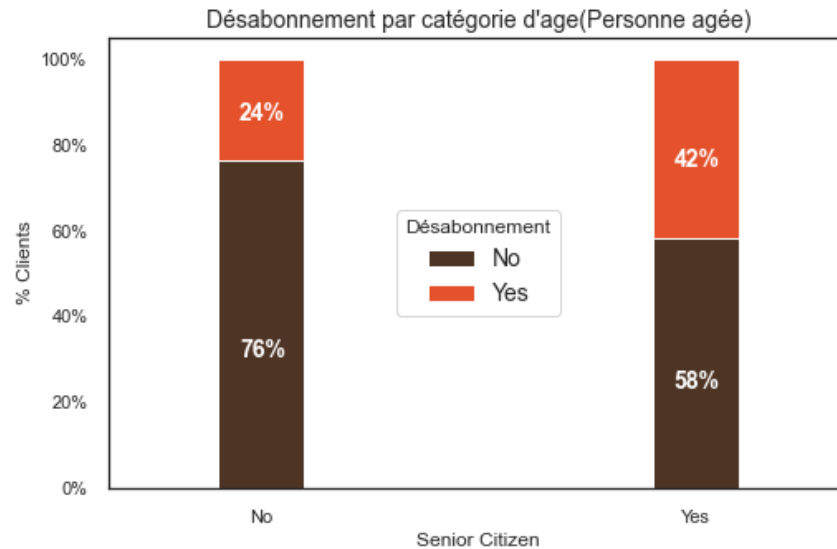


Figure 8 : Désabonnement par catégorie d'âge

On peut conclure de la que les clients qui résilient plus leur contrat sont les personnes les âgées ( 2 fois plus élevées que les jeunes).

Qu'en est il du taux de désabonnement et des charges mensuelles ?

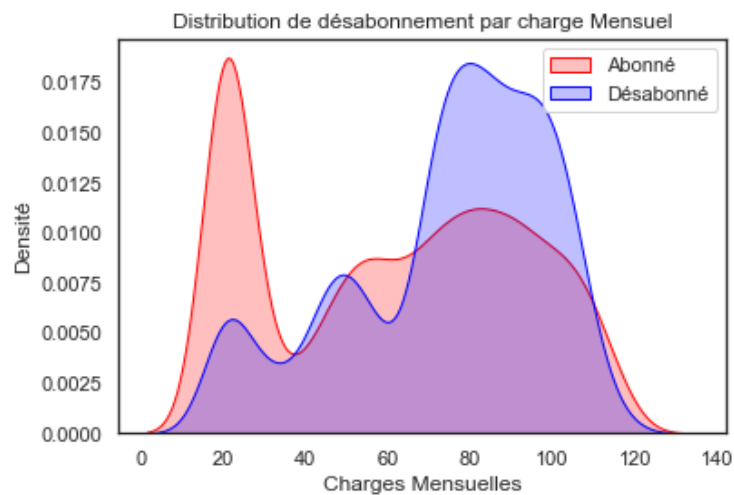


Figure 9 : Désabonnement et charge mensuelle

On peut voir que **plus la charge mensuelle augmente, plus les clients se désabonnent**

## II. La construction du Data Hub

Cette phase de **préparation des données** regroupe les activités liées à la construction de l'ensemble précis des données à analyser, faite à partir des données brutes. Elle inclut ainsi le classement des données en fonction de critères choisis, le nettoyage des données, et surtout leur recodage pour les rendre compatibles avec les algorithmes qui seront utilisés.<sup>2</sup>

### A. Le type de variables

La totalité des algorithmes que nous utilisons pour ce problème de Customer churn utilise des variables quantitatives.

Il a été donc question

- de supprimer des variables qualitatives :
  - ID Customer
  - Zip Code
  - Lat Long
  - Latitude
  - Longitude
- De transformer les variables catégoriques (Yes et No) en variables binaires (0 et 1)
- De transformer les variables multi catégoriques en variables en sous variables binaires

### B. Ajout de Variables

Dans le but de rendre plus précis certains algorithmes que nous avons choisis ( de la famille Gradient Boosting ), suite à la compréhension de données des variables ont été ajoutées :

- Engaged : variable booléenne qui prend la valeur 0 lorsque le contrat est « month-to-month »
- YandNotE : variable booléenne qui précise les jeunes qui ont un contrat différent de « month-to-month »
- ElectCheck : variable booléenne qui précise tous ceux qui ont contrat différent de « month-to-month » et payent par chèque électronique
- Fiberopt : variable booléenne qui précise ceux qui ont souscrit au services fibre optique
- StreamNoInt : variable aléatoire qui précise si un client a souscrit ou non au service streaming TV
- NoProt : variable booléenne qui précise si un client a souscrit à des options de sécurité de sa ligne

---

<sup>2</sup> <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/>

### C. Suppression de variables

Bien qu'étant quantitatives, certaines variables se sont vues en fonction de l'algorithme être supprimées en raison de leur forte corrélation avec d'autres, diminuant ainsi la précision de la prédiction de ces algorithmes. Ça a été le cas de 6 variables par exemple pour Random Forest.

### D. Normalisation des données

Dans le but d'homogénéiser la base de données et pour rendre plus précis nos algorithmes, le dataset a été normalisé (de façon entière pour certains algorithmes (naïves bayes, Random Forest) et de façons partielles pour d'autres (gradient boosting))

Le Data Hub ainsi conçu, nous passons à l'itération suivante qui est celle de la modélisation.

### III. Modélisation et optimisation

C'est la phase de Data Science proprement dite. La **modélisation** comprend le choix, le paramétrage et le test de différents algorithmes ainsi que leur enchaînement, qui constitue un modèle.<sup>3</sup>

Les algorithmes qui ont été utilisés sont les suivants :

- Naive Bayes
- Random Forest Classifier
- De Régression Linéaire et Logistique
- Gradient Boosting Classifier
- XGBoost Classifier
- LGBM Classifier

Tout au long de cette partie nous ferons allusion à des termes qu'il serait nécessaire de définir au préalable

#### A. Définitions des termes clés

##### 1. Matrice de confusion

En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. La cellule ligne L, colonne C contient le nombre d'éléments de la classe réelle L qui ont été estimés comme appartenant à la classe C.<sup>4</sup>

		Classe réelle	
		-	+
Classe prédite	-	<b>True Negatives</b> <i>(vrais négatifs)</i>	<b>False Negatives</b> <i>(faux négatifs)</i>
	+	<b>False Positives</b> <i>(faux positifs)</i>	<b>True Positives</b> <i>(vrais positifs)</i>

Tableau 1 : Exemple de Matrice de confusion

<sup>3</sup> <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/>

<sup>4</sup>

[https://fr.wikipedia.org/wiki/Matrice\\_de\\_confusion#:~:text=En%20apprentissage%20automatique%20supervisé%20la,correspond%20%C3%A0%20une%20classe%20estim%C3%A9e.](https://fr.wikipedia.org/wiki/Matrice_de_confusion#:~:text=En%20apprentissage%20automatique%20supervisé%20la,correspond%20%C3%A0%20une%20classe%20estim%C3%A9e.)



## 2. Score

Des rapports de classification seront faits pour pouvoir déterminer les scores suivants:

- **Precision**: il détermine la capacité du classificateur à ne pas étiqueter comme positif un échantillon négatif.

$$\text{precision} = \text{tp} / (\text{tp} + \text{fp})$$

- **recall** : il détermine la capacité du classificateur à trouver tous les échantillons positifs.

$$\text{recall} = \text{tp} / (\text{tp} + \text{fn})$$

- **f1-score** : c'est une moyenne pondérée de la precision et du recall , où un score F1 atteint sa meilleure valeur à 1 et son pire score à 0

$$\text{f1-score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

### a. Courbe ROC

une courbe pour montrer comment la sensibilité évolue en fonction de la spécificité du modele binaire<sup>5</sup>

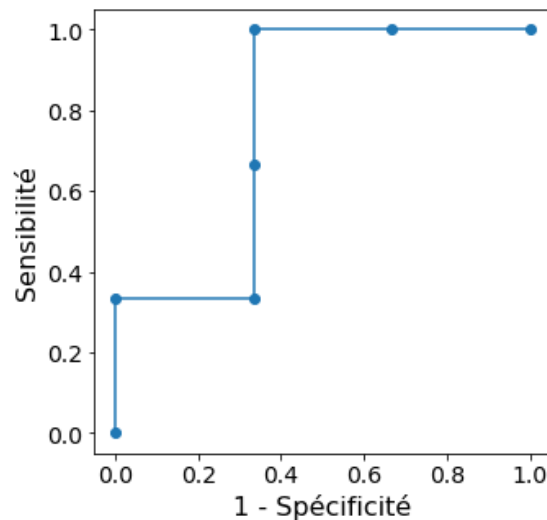


Figure 10 : Exemple de courbe ROC

<sup>5</sup> <https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308261-evaluez-un-algorithme-de-classification-qui-retourne-des-scores>

## B. Modélisation de quelques algorithmes

### 1. Naive Bayes

La méthode naïve bayésienne comme son nom l'indique est une méthode naïve qui regroupe plusieurs modèles qui sont extrêmement rapides et simples qui peuvent des fois bien fonctionner sur des dataset avec de grandes dimensions. Ci-joint une figure expliquant la logique de l'algorithme de Naïve Bayes classifier.

### Naive Bayes Classifier

The diagram shows the formula for Naive Bayes classification: 
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
 Arrows point from the following labels to the corresponding parts of the formula: 'Likelihood' points to  $P(x|c)$ ; 'Class Prior Probability' points to  $P(c)$ ; 'Posterior Probability' points to  $P(c|x)$ ; and 'Predictor Prior Probability' points to  $P(x)$ .

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Figure 11 : Formule de l'algorithme Naives Bayes

Cette méthode admet trois lois qu'elle suit : Gaussien, Multinomial et Bernoulli. On doit calculer les scores de ces 3 lois puis choisir celle avec le meilleur score pour pouvoir continuer.

```
{'gaussian': 0.8796444444444445,  
'bernoulli': 0.7848888888888889,  
'multinomial': 0.7409777777777777}
```

Figure 12 : Scores des algorithmes de Naïve Bayes

Nous avons opté pour Gaussien. L'optimisation des hyperparamètres nous permettra d'améliorer ce score. Mais malgré ces changements on ne peut atteindre un score acceptable et nous ne trouvons qu'une amélioration de quelques pourcentages pour atteindre les 87 et 89%. Cela peut être dû au fait que ces algorithmes sont simplistes pour ce problème et donc ne peuvent pas être fiables ou du fait que cet algorithme donne de mauvais résultats s'il y a une grande corrélation entre les caractéristiques.

Les figures suivantes présentent un rapport de classification et la matrice de confusion après optimisation des hyperparamètres :

modele NaiveBayes				
	precision	recall	f1-score	support
No churn	0.941468	0.918683	0.929936	1033
Churn	0.789474	0.842246	0.815006	374
accuracy			0.898365	1407
macro avg	0.865471	0.880465	0.872471	1407
weighted avg	0.901066	0.898365	0.899386	1407

Figure 13 : Classification report Naïve bayes

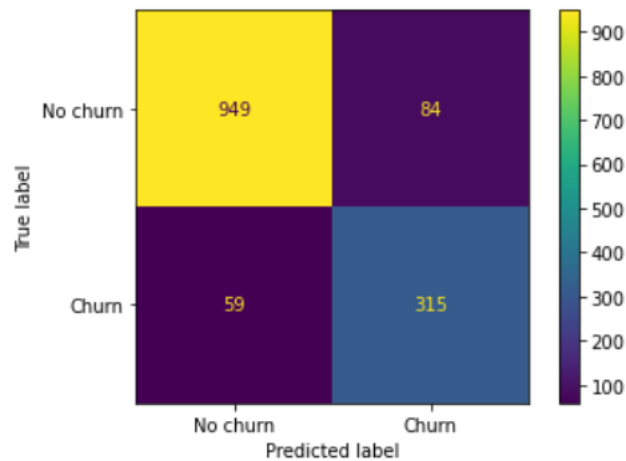


Figure 14 : Matrice de confusion Naïve Bayes

## 2. Random Forest classification

Il se compose d'un grand nombre d'arbres de décision individuels qui fonctionnent comme un ensemble . Chaque arbre individuel dans la forêt aléatoire crache une prédiction de classe et la classe avec le plus de votes devient la prédiction de notre modèle (voir figure ci-dessous).

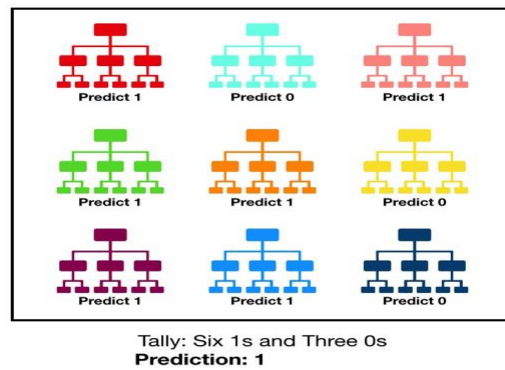


Figure 15 : Fonctionnement du Random forest Classifier

Dans le but d'améliorer la prédiction de l'algorithme, nous avons modifié notre dataset par la suppression des variables inutiles (6) en se basant sur la matrice de corrélation. Par suite on a entraîné notre algorithme sur consacrer 80% des données et nous avons consacré 20% pour la partie test avec l'utilisation de l'hyper paramètre (stratify=y) pour assurer un fractionnement proportionnel aux valeurs fournies.

De plus, nous avons fait quelques modifications au niveau des hyper-paramètres de l'algorithme afin d'améliorer le taux de prédiction :

- **n\_estimators = 100 (le max)** : représente le nombre d'arbres à utiliser dans notre algo, plus le nombre augmente, plus nous avons de chance d'avoir une meilleure prédiction.
- **Random\_state= 0** : il est souvent difficile de dupliquer exactement les résultats. Ce paramètre permet de répliquer facilement les résultats si les mêmes données et paramètres d'entraînement sont fournis.

Ainsi la figure suivante nous présente les différents scores pour Random Forest Classifier appliqués à nos données :

modele Random Forest	precision	recall	f1-score	support
No churn	0.965193	0.993224	0.979008	1033
Churn	0.979651	0.901070	0.938719	374
accuracy			0.968728	1407
macro avg	0.972422	0.947147	0.958863	1407
weighted avg	0.969036	0.968728	0.968298	1407

Figure 16 : Classification report random forest classifier

La matrice de confusion finale est donnée dans la figure qui suite :

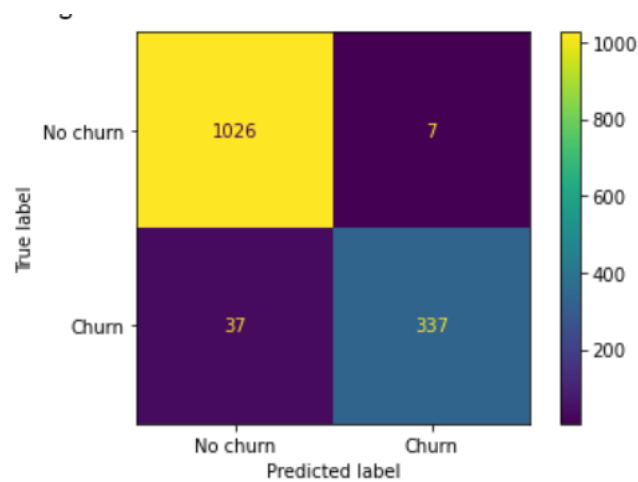


Figure 17 : Matrice de confusion random forest classifier

### 3. Régression linéaire et régression logistique

#### a. Régression Linéaire

$$f(X_1, X_2, \dots) = a_0 + a_1 * X_1 + a_2 * X_2 + \dots + a_{12} * X_{12}$$

La régression linéaire multiple est une méthode de régression mathématique étendant la régression linéaire simple pour décrire les variations d'une variable endogène associée aux variations de plusieurs variables exogènes.

Dans notre cas, nous allons essayer d'écrire le Churn en fonction des autres données que nous avons.

Nous utilisons pour cela différentes méthodes (comme SelectKbest ou encore supprimer les variables qui ont la variance la plus faible pour essayer d'améliorer le score à chaque fois.)

```
test_score = 0.5367340284230433
R2 = 0.5367340284230433
MAE = 0.09109296735550114
RMSE = 0.3018161151355261
MAE = 0.2516103950677144
Median = 0.23038957191468945
```

Figure 18 : Résultats de la régression linéaire

## b. Régression Logistique

La régression logistique est utilisée pour le classement et pas la régression linéaire. Mais elle est considérée comme une méthode de classification puisqu'elle sert à estimer la probabilité d'appartenir à une classe. Il y a trois types de régression logistique :

- Régression logistique binaire : ici, le but de la classification est d'identifier si un échantillon appartient à une classe ou non.
- Régression logistique multinomiale : ici, le but de la classification est d'identifier à quelle classe appartient un échantillon parmi plusieurs classes.
- Régression logistique ordinale : ici, le but de la classification est de chercher la classe d'un échantillon parmi des classes ordonnées. Un exemple de classes : non satisfait, satisfait, très satisfait

Pour combiner entre les différentes caractéristiques, on utilise une fonction linéaire (exactement comme la régression linéaire):

$$z(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_m x_m$$

Cette valeur est transformée à une probabilité en utilisant la fonction logistique. Donc, la probabilité qu'un échantillon avec les caractéristiques  $x_1, \dots, x_n$  appartienne à une classe  $y$  est calculée comme suit:

$$h_{\theta}(x) = p(y = 1|x) = \frac{1}{1 + e^{-z(x)}}$$

Les figures suivantes nous présentent sa matrice de confusion et son rapport de classification :

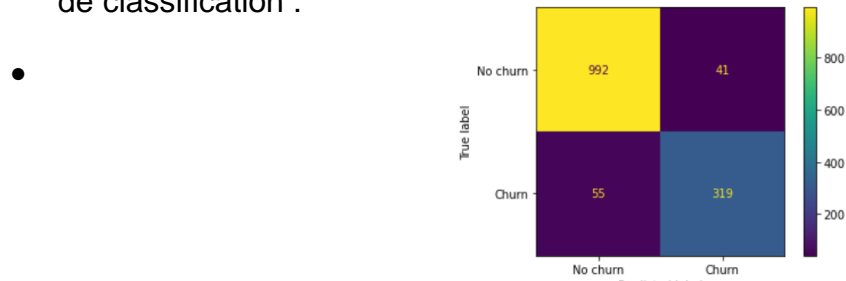


Figure 19 : Matrice de confusion Régression Logistique

modele Regression Logistique				
	precision	recall	f1-score	support
No churn	0.947469	0.960310	0.953846	1033
Churn	0.886111	0.852941	0.869210	374
accuracy			0.931770	1407
macro avg	0.916790	0.906625	0.911528	1407
weighted avg	0.931159	0.931770	0.931349	1407

Figure 20 : Classification report Régression Logistique

## 4. Gradient Boosting Classifier

### a. Principe

Gradient Boosting classifier est un algorithme de machine Learning basé sur le boosting. Le boosting étant une technique séquentielle qui fonctionne sur le principe de l'ensemble. Il combine un ensemble d'algorithmes ayant des résultats dit faibles et améliore leur performance combinée pour avoir un meilleur résultat. Les résultats d'une prédiction  $T$  se font en fonction des prédictions des résultats à l'instant  $T-1$ . Pour être plus claire, la faiblesse des uns rend fort les autres. Pour que ce principe soit efficace, il est important que les algorithmes qui participent au Boosting réunissent deux conditions à savoir la diversification des résultats et un score individuel supérieur à 50%. Il utilise des modèles d'arbres de décision comme modèle faible. A chaque séquence, la nouvelle arborescence est ajoutée à l'arborescence précédente et ce principe est répété jusqu'à atteindre un nombre d'arbre bien prédéfini ou un certain seuil.

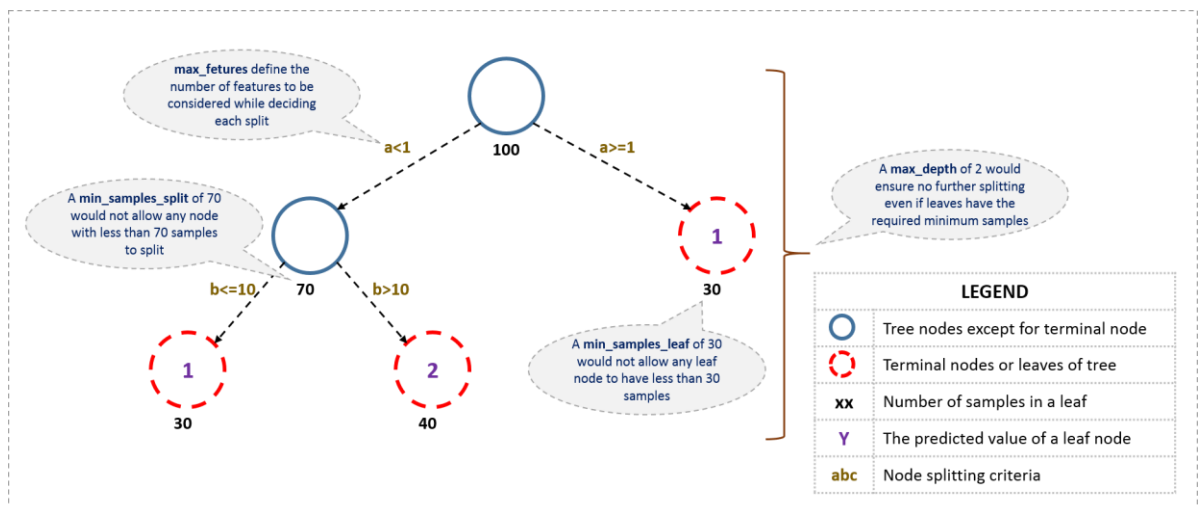
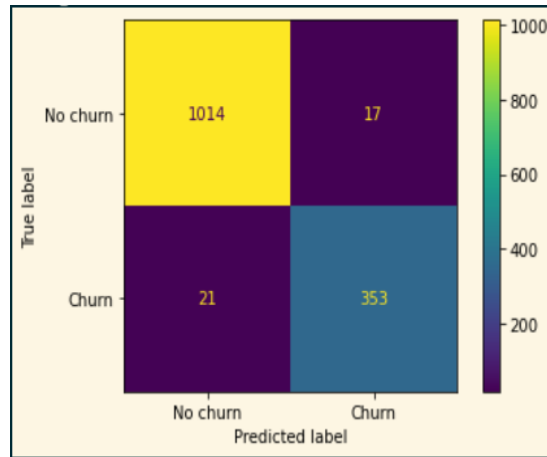


Figure 21 : Principe du Gradient boosting

## b. Application Gradient Boosting Classifier modèle de base

Gradient Boosting est un algorithme d'ensemble très puissant. Un entraînement avec les paramètres de base nous on permet d'avoir la matrice de confusion suivante.



De plus la courbe roc classe ces modèles par ordre de performance comme suite Xgboost > Gradient Boosting > RandomFoorest. Bien que Xgboost soit légèrement plus performant que Grandient Boosting, il existe un danger dû au fait que Xgboost et randomForest sont en sur-entraînement (overfitting) et ont donc du mal à généraliser l'apprentissage, pourtant après un entraînement sur 80% des données contrairement à Gradient Boosting qui n'est pas en overfitting et est capable de généraliser.

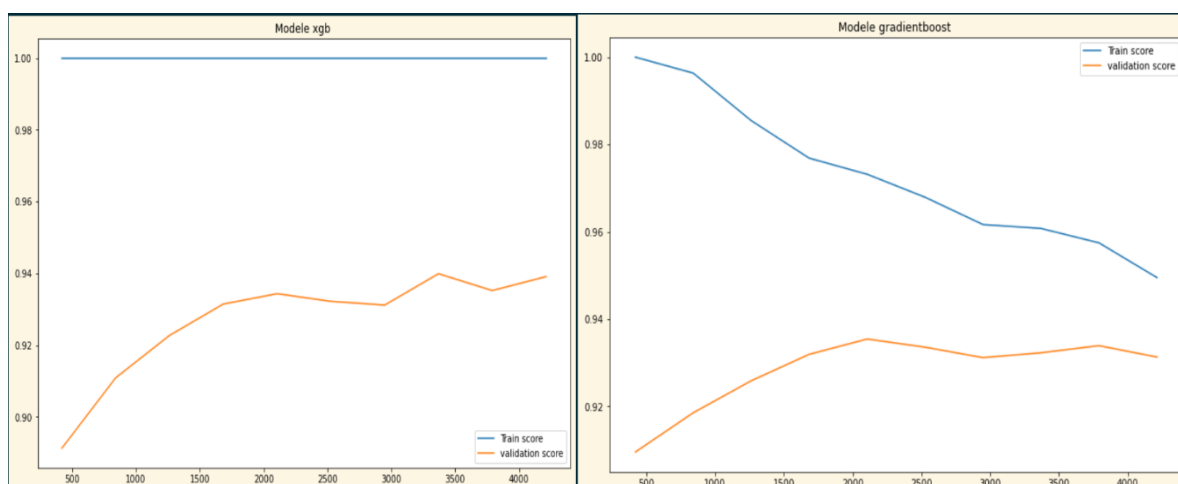


Figure 22 : Courbe ROC XGB et Gradient Boosting



Notre objectif sera donc de régler les hyper-paramètres de ce modèle pour qu'il soit plus performant, généralise plus rapidement et consomme moins de ressource.

### c. Réglage des Hyperparamètres de Gradient Boosting Classifier

RandomizeSearchCV est souvent utilisé pour régler tous les hypers paramètres en même temps. Ce n'est souvent pas très efficace et nous n'avons pas souvent les meilleurs paramètres, C'est pour cette raison que nous allons régler chaque paramètre un à un à un GridSearchCV.

Les principaux paramètres sont les suivants et vont être ajoutés dans cet ordre

- **Learning\_rate** : Cela détermine l'impact de chaque arbre sur le résultat final. Plus ce nombre est petit, plus le modèle est robuste et plus nous aurons besoin d'un grand nombre d'arbre
- **N\_estimators** : nombre d'arbres séquentiels à modéliser. Un très grand nombre rend l'algorithme plus performant mais peut entraîner l'overfitting.
- **Max\_depth** : la profondeur maximale de l'arbre.
- **Min\_samples\_split** : Définit le nombre minimum d'échantillons (ou d'observations) qui sont requis dans un nœud pour être pris en compte pour le fractionnement.
- **Min\_sample\_leaf** : Définit les échantillons (ou observations) minimum requis dans un nœud terminal ou une feuille.
- **Max\_features** : Le nombre de features à considérer lors de la recherche de la meilleure répartition.
- **Subsample** : La fraction des observations à sélectionner pour chaque arbre.

Après le réglage des hyperparamètres, on obtient les résultats suivants.

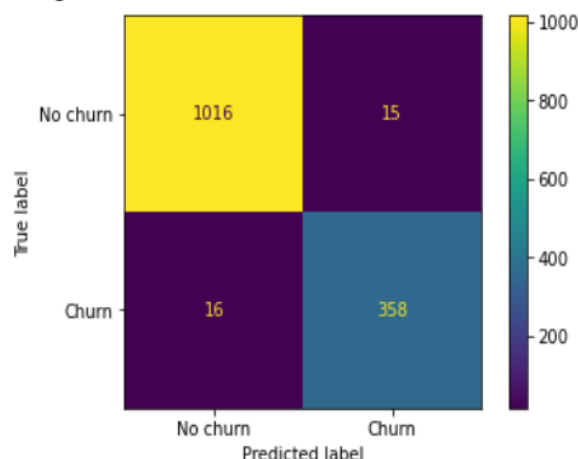


Figure 23 : Matrice de confusion GradientBoost

modele Gradientboost				
	precision	recall	f1-score	support
No churn	0.984496	0.985451	0.984973	1031
Churn	0.959786	0.957219	0.958501	374
accuracy			0.977936	1405
macro avg	0.972141	0.971335	0.971737	1405
weighted avg	0.977918	0.977936	0.977927	1405

Figure 24 : Classification report Gradient Boost

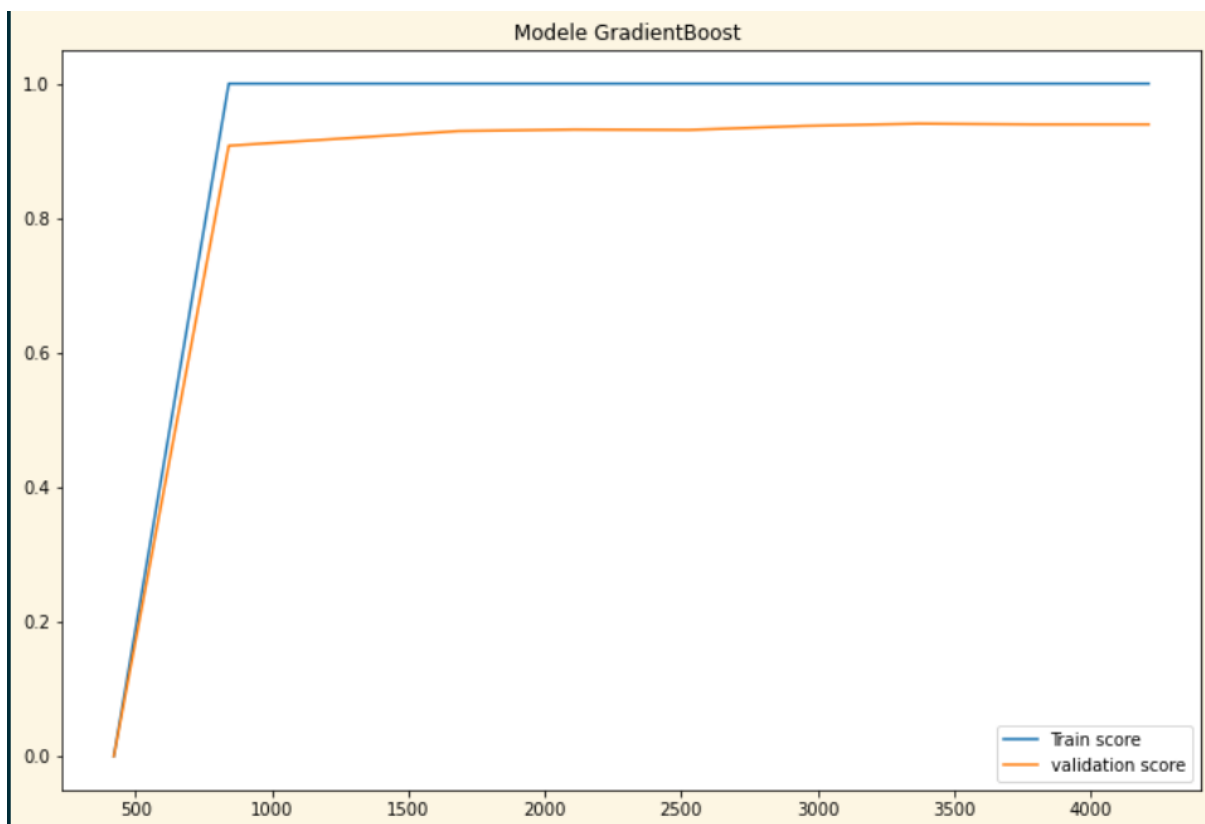


Figure 25 : Learning curve Gradient Boost

Nous remarquons qu'après réglage des hyper-paramètres le modèle a pu détecter quatre nouveaux clients susceptibles de se désabonner et un susceptible de rester. L'ajout de quelques données pourrait rendre le modèle parfait.

#### d. Fonction de décision

Nous rappelons que l'objectif majeur est de déterminer avec précision les clients susceptibles de se désabonner pour éviter des pertes à l'entreprise. Nous pouvons donc nous permettre de perdre un peu de précision pour améliorer le recall. D'où l'utilité de la fonction de décision.

Un croisement entre la courbe de précision et de recall nous permet de définir un seuil pour améliorer le recall. Ainsi donc nous quittons d'un recall churn de 95.7% à 98.66% et nous avons un modèle final qui équilibré et stable dans ses prédictions.

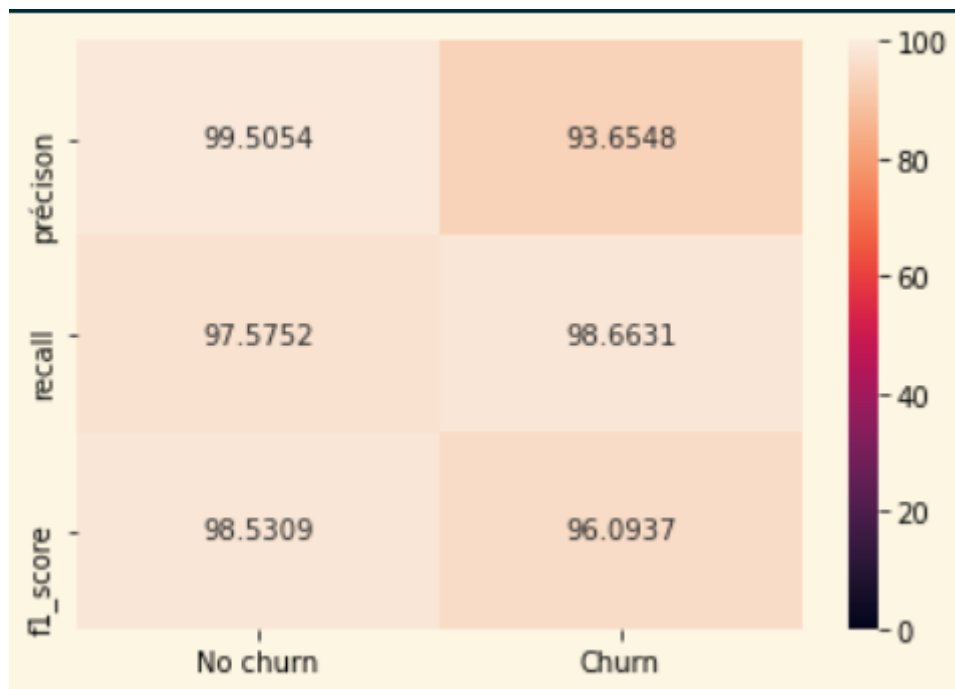


Figure 26 : Classification report avec la fonction de décision

#### e. Best features

Le modèle final après l'utilisation de l'attribut `feature_importances_` retourne les features par ordre décroissant d'importance dans le processus de prédiction.

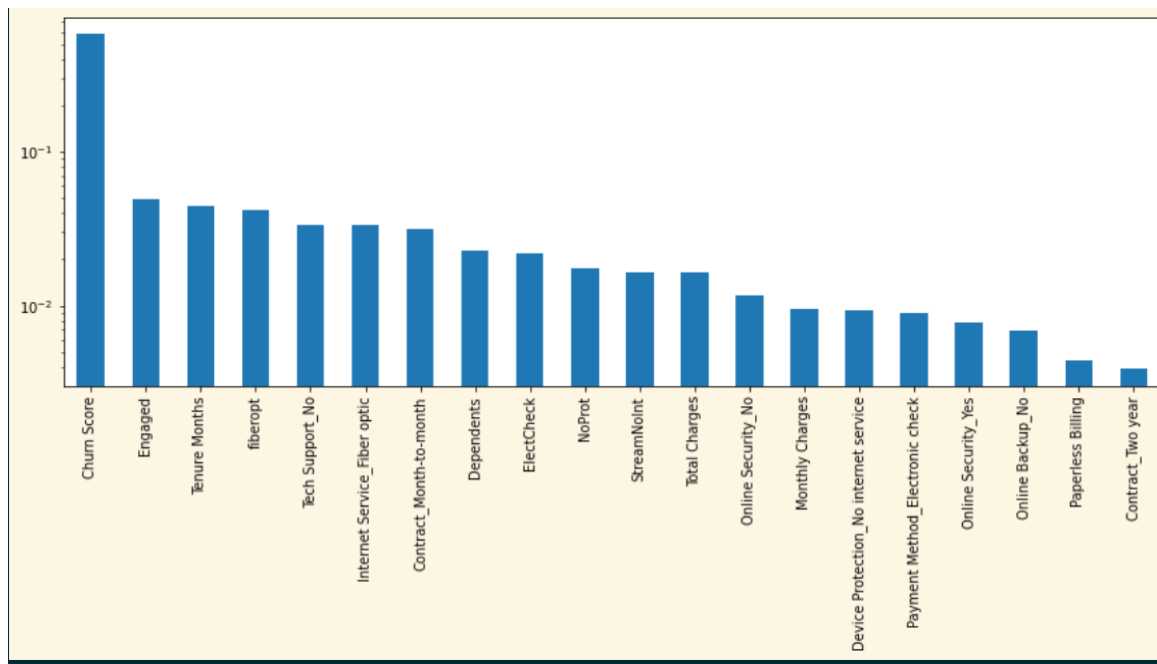


Figure 27 : Meilleure variables pour Gradient boost

## 5. XGBoost Classifier

XGBoost pour eXtreme Gradient Boosting est une implémentation de l'algorithme d'arbres de boosting de gradient (algorithme d'apprentissage supervisé dont le principe est de combiner les résultats d'un ensemble de modèles plus simple et plus faibles afin de fournir une meilleure prédiction.).

Un traitement de données particulier a encore été fait, afin de maximiser le taux de prédiction de cet algorithme. Il a fallu pour cela, supprimer des variables inutiles pour l'algorithme à l'aide de la matrice de corrélation. Ce travail a permis d'enlever 6 variables.

Par la suite un processus de réglage de hyperparamètres a été fait, dans le but d'obtenir la prédiction le meilleur possible.

Ainsi la figure suivante nous présente ces différents scores pour XGBoost appliqué à nos données :

modele XGBClassifier				
	precision	recall	f1-score	support
No churn	0.983495	0.982541	0.983018	1031
Churn	0.952000	0.954545	0.953271	374
accuracy			0.975089	1405
macro avg	0.967748	0.968543	0.968144	1405
weighted avg	0.975111	0.975089	0.975100	1405

Figure 28: Classification report XGBoost

La matrice de confusion finale est donnée dans la figure qui suit :

### a. Courbe d'apprentissage

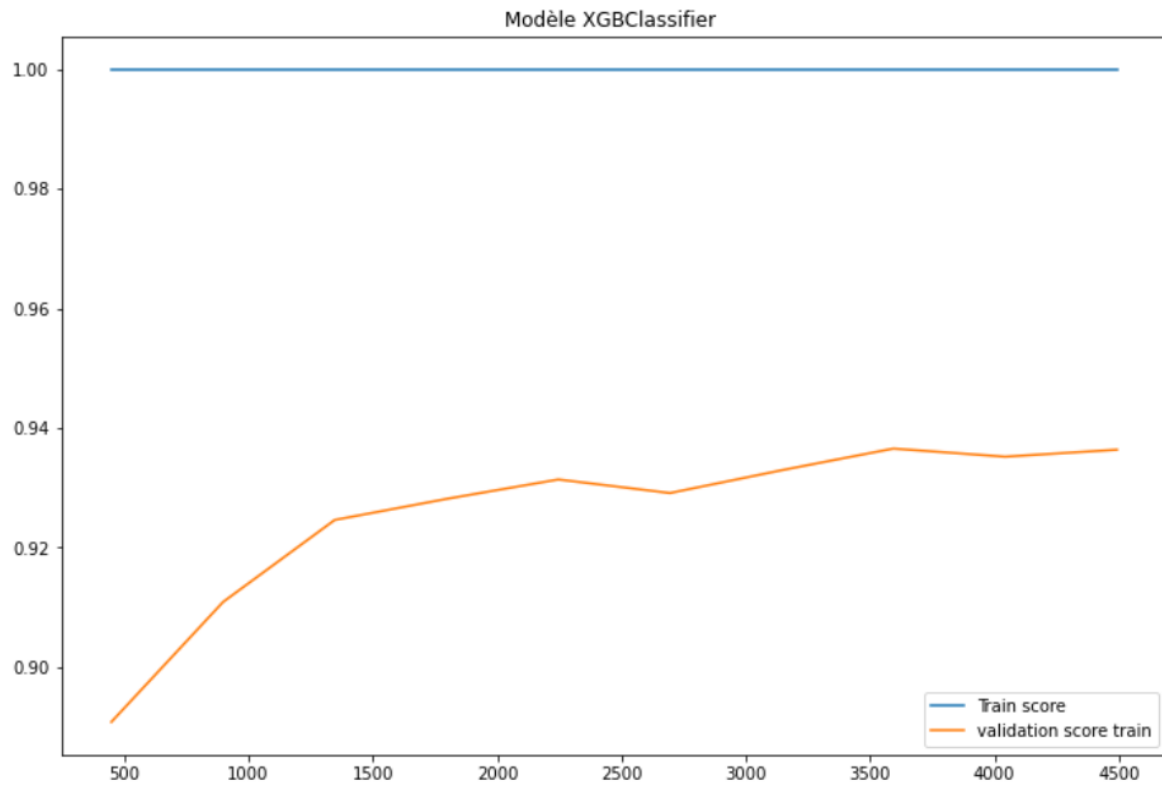


Figure 29 : Learning curve XGBoost

Il est clair qu'il y a un problème qui se pose, celui du surapprentissage (Overfitting). Le réglage des hyperparamètres afin de le faire sortir de là baisse considérablement les résultats, raison pour laquelle nous nous penchons sur son concurrent (LGBClassifier)

## 5. Light Gradient Boosting

LightGBM est un Framework de gradient boosting qui utilise des algorithmes d'apprentissage basés sur l'arbre<sup>6</sup>. Il est conçu pour être distribué et efficace avec les avantages suivants :

- Une vitesse d'apprentissage plus rapide et une efficacité accrue.
- Une utilisation de la mémoire plus faible.
- Meilleure précision.
- Prise en charge de l'apprentissage parallèle et GPU.
- Capable de traiter des données à grande échelle.

Il a été développé par Microsoft et concurrence bien XGBoost.

avec une précision moyenne de 0.970644 , il a été le meilleur algorithme de prédiction pour ce problème de prédictions.

Sa matrice de confusion et son rapport de classification après réglages des hyperparamètres sont les suivants :

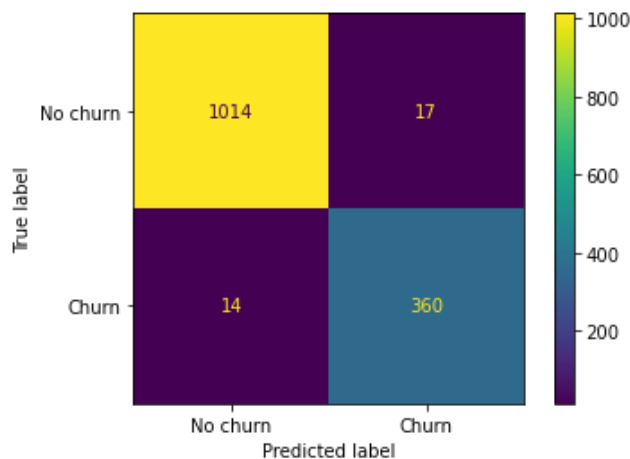


Figure 30 : Matrice de confusion de LGBClassifier

modele LGBMClassifier				
	precision	recall	f1-score	support
No churn	0.986381	0.983511	0.984944	1031
Churn	0.954907	0.962567	0.958722	374
accuracy			0.977936	1405
macro avg	0.970644	0.973039	0.971833	1405
weighted avg	0.978003	0.977936	0.977964	1405

Figure 31 : Rapport de Classification LGBM

<sup>6</sup> <https://www.kaggle.com/prashant111/lightgbm-classifier-in-python>

## IV. Evaluation des modèles

Après avoir effectué une modélisation et évaluation des différents algorithmes, nous pouvons constater que les meilleurs algorithmes ont été LGBClassifier et Gradient Boosting dont les matrices de confusion sont les suivantes :

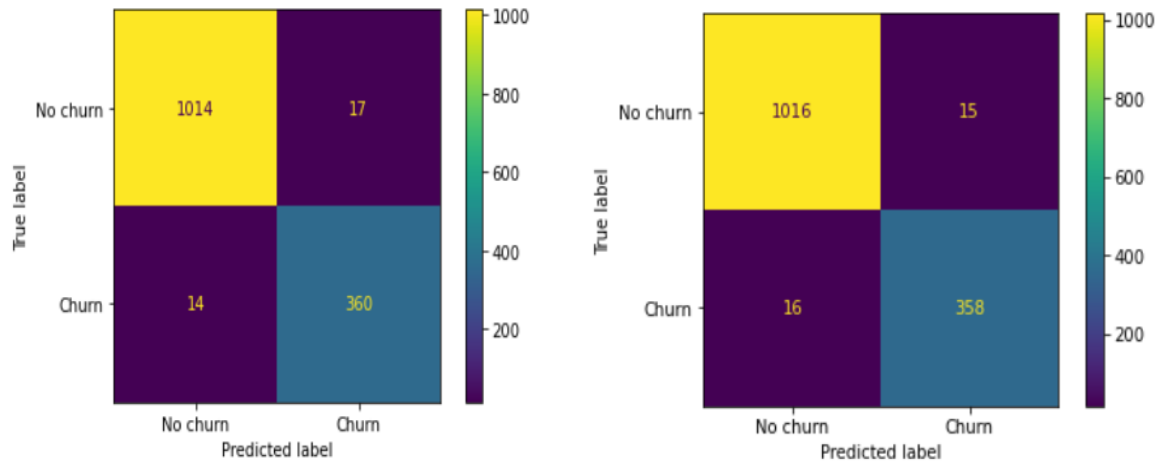


Figure 32 : Matrice de Confusion des meilleures Algorithmes (LGB et GB)

Un résumé général permettant de mieux comparer les algorithmes utilisés nous est d'une grande utilité.

Tableau 2 : Comparaison des algorithmes

Modèles	précision	recall	f1_score	accuracy	test_size	Etat
XGBClassifier	99.7994	96.5082	98.1262	97.2954	1031	No churn
XGBClassifier	91.1765	99.4652	95.1407	97.2954	374	Churn
Gradientboost	98.4496	98.5451	98.4973	97.7936	1031	No churn
Gradientboost	95.9786	95.7219	95.8501	97.7936	374	Churn
LGBMClassifier	98.6381	98.3511	98.4944	97.7936	1031	No churn
LGBMClassifier	95.4907	96.2567	95.8722	97.7936	374	Churn
Regression Logist	94.7469	96.031	95.3846	93.177	1033	No churn
Regression Logist	88.6111	85.2941	86.921	93.177	374	Churn
RamdomForest	96.5193	99.3224	97.9008	96.8728	1033	No churn
RamdomForest	97.9651	90.107	93.8719	96.8728	374	Churn
NaiveBayes	94.1468	91.8683	92.9936	89.8365	1033	No churn
NaiveBayes	78.9474	84.2246	81.5006	89.8365	374	Churn

Nous remarquons que les algorithmes d'ensembles sont meilleurs que les autres pour le problème de prédiction de désabonnement. Notamment LightGradientBoost, GradientBoost et XGBoost qui sont tous de la famille des algorithmes de boosting ont les meilleurs scores en accuracy avec respectivement 97.79%, 97.79%, 97.29%.

## V. Mise en Production

Afin de rendre profitable notre travail au grand public, un déploiement a été fait sur une application développée à l'aide du Framework python Django.

Le modèle qui a été choisis est bien évidemment le meilleur modèle qui a été entraîné, à savoir LGB Classifier qui fait une prédiction à 98% :

La page d'accueil de l'application permet de remplir un formulaire correspond aux informations sur le client.

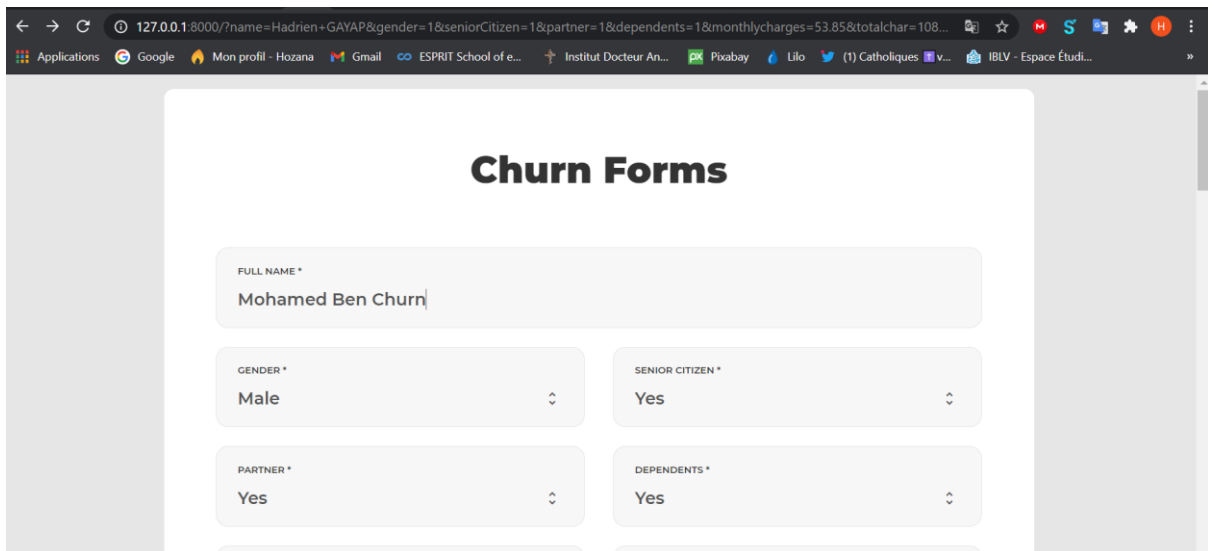


Figure 33 : Formulaire de Prédiction à l'aide de l'application Django

Après le remplissage du formulaire, une soumission doit être faite pour avoir le résultat de prédiction.

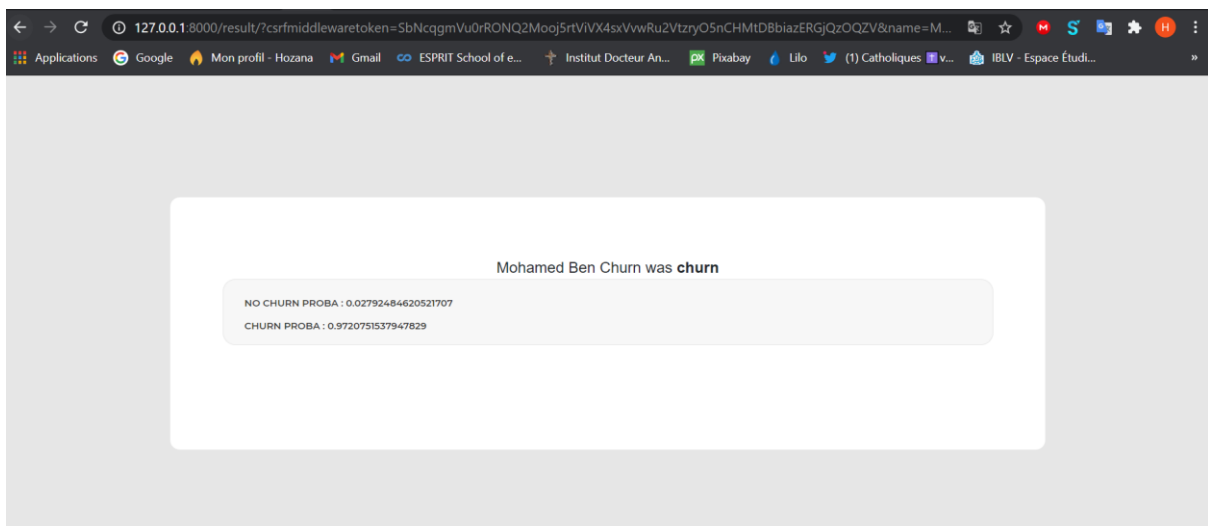


Figure 34 : Prédiction de l'application



## CONCLUSION

Il était question pour nous durant ce projet d'apporter notre pierre à la résolution du problème de « Customer Churn » dans les entreprises de télécommunications à travers les techniques de Data Science. Tout compte fait, le travail a été effectué avec succès à l'aide de la célèbre méthode CRISP-DM mise sur pieds par IBM. Aussi il a été constaté que les meilleurs algorithmes de prédictions pour le problème de Churn Customer proviennent de la famille des Gradient Boosting et le meilleur algorithme pour notre cas est LGB Classifier développé par Microsoft dont le modèle entraîné donne une prédiction à 98%. Ce modèle a été déployé sur une application. Que faire des faire des clients dont la probabilité de départ est grande ?

## REFERENCES

- 1 <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/>  
09/01/21 01 :38
- 2 <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/>
- 3 <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/>
- 4 [https://fr.wikipedia.org/wiki/Matrice\\_de\\_confusion#:~:text=En%20apprentissage%20automatique%20supervis%C3%A9%2C%20la,correspond%20%C3%A0%20une%20classe%20estim%C3%A9e.](https://fr.wikipedia.org/wiki/Matrice_de_confusion#:~:text=En%20apprentissage%20automatique%20supervis%C3%A9%2C%20la,correspond%20%C3%A0%20une%20classe%20estim%C3%A9e.)
- 5 <https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308261-evaluez-un-algorithme-de-classification-qui-retourne-des-scores>
- 6 <https://www.kaggle.com/prashant111/lightgbm-classifier-in-python>

# TABLE DE MATIERE

<b>RESUME .....</b>	<b>IV</b>
<b>INTRODUCTION.....</b>	<b>1</b>
<b>I.    <u>COMPRÉHENSION DES DONNÉES</u>.....</b>	<b>2</b>
A. <u>LES COLONNES</u> .....	3
B. <u>EXPLORATION DES DONNEES</u> .....	3
1. <u>Démographie</u> .....	3
2. <u>Comptes clients</u> .....	5
3. <u>Les services</u> .....	6
4. <u>L'étiquette (Churn)</u> .....	7
<b>II.   <u>LA CONSTRUCTION DU DATA HUB</u> .....</b>	<b>8</b>
A. <u>LE TYPE DE VARIABLES</u> .....	8
B. <u>AJOUT DE VARIABLES</u> .....	8
C. <u>SUPPRESSION DE VARIABLES</u> .....	9
D. <u>NORMALISATION DES DONNEES</u> .....	9
<b>III. <u>MODELISATION ET OPTIMISATION</u> .....</b>	<b>10</b>
A. <u>DÉFINITIONS DES TERMES CLÉS</u> .....	10
1. <u>Matrice de confusion</u> .....	10
2. <u>Score</u> .....	11
a. <u>Courbe ROC</u> .....	11
B. <u>MODÉLISATION DE QUELQUES ALGORITHMES</u> .....	12
1. <u>Naive Bayes</u> .....	12
2. <u>Random Forest classification</u> .....	13
3. <u>Régression linéaire et régression logistique</u> .....	15
a. <u>Régression Linéaire</u> .....	15
b. <u>Régression Logistique</u> .....	16
4. <u>Gradient Boosting Classifier</u> .....	17
a. <u>Principe</u> .....	17
b. <u>Application Gradient Boosting Classifier modèle de base</u> .....	18
c. <u>Réglage des Hyperparamètres de Gradient Boosting Classifier</u> .....	19
d. <u>Fonction de décision</u> .....	21
e. <u>Best features</u> .....	21
5. <u>XGBoost Classifier</u> .....	22
a. <u>Courbe d'apprentissage</u> .....	23
6. <u>LIGHT GRADIENT BOOSTING</u> .....	24
<b>IV.    <u>EVALUATION DES MODELES</u> .....</b>	<b>25</b>
<b>V.    <u>MISE EN PRODUCTION</u>.....</b>	<b>26</b>
<b>CONCLUSION.....</b>	<b>27</b>
<b>REFERENCES.....</b>	<b>28</b>