

Customer Churn



Réalisé par :

Benjeddou Wejdene

TABLE MATIERE

Introduction Générale	3
Introduction à CRISP-DM	4
Compréhension du problème métier	5
Compréhension des données	6
1- Aperçu sur la Base de données	6
2- Visualisation du pourcentage des clients	8
3 - Visualisation du nombres des Churn des clients	8
4-Visualisation du taux de désabonnement pour les hommes et les femmes	8
5 - Visualisation du nombre de churn pour le service Internet	8
6 - Visualisation du nombre de churn pour le “Monthly Charge” et “tenure”	9
7- Visualisation de la corrélation entre les variables	9
Préparation des données	10
1-Encodage	10
2-Normalisation	11
3-Imputation	12
4-Sélection	13
2- Arbre de Décisions (Decision Tree)	19
3- Régression Logistique (Logistic Regression)	21
4- XGBoost	23
5- SVM Classifier	25
Evaluation	27
Déploiement	31
1- Logiciel de déploiement	31
2- Réalisation	32
Conclusion et Perspectives	33
Références	34

I.

Introduction Générale



Le Machine Learning peut être défini comme étant une technologie d'intelligence artificielle permettant aux machines d'apprendre sans avoir été au préalablement programmées spécifiquement à cet effet. Le Machine Learning est explicitement lié au Big Data, étant donné que pour apprendre et se développer, les ordinateurs ont besoin de flux de données à analyser, sur lesquelles s'entraîner.

De ce fait, le Machine Learning, issu par essence du Big Data, a précisément besoin de ce dernier pour fonctionner. Le Machine Learning et le Big Data sont donc interdépendants.

Encore confus pour de nombreuses personnes, **le Machine Learning** est une science moderne permettant de **découvrir des répétitions (des patterns) dans un ou plusieurs flux de données et d'en tirer des prédictions en se basant sur des statistiques**. En clair, le Machine Learning se base sur le forage de données, permettant la reconnaissance de patterns pour fournir des analyses prédictives.

Pour **analyser de tels volumes de données**, le Machine Learning se révèle bien plus efficace en termes de vitesse et de précisions que les autres méthodologies traditionnelles. À titre d'exemple, le Machine Learning est capable de déceler une fraude en une milliseconde, rien qu'en se basant sur des données issues d'une transaction (montant, localisation...), ainsi que sur d'autres informations historiques et sociales qui lui sont rattachées. En ce qui concerne l'analyse de données transactionnelles, de données issues de plateformes CRM ou bien des réseaux sociaux, là encore le Machine Learning se révèle désormais indispensable.



Le Machine Learning est réellement la science idéale pour **tirer profit du Big Data et de ses opportunités**. Cette technologie est en effet capable d'extraire les données de valeur parmi d'immenses sources d'informations complexes, le Machine Learning révèle au contraire tout son potentiel lorsque les sources de données sont croissantes, lui permettant **d'apprendre et d'affiner des insights** avec une précision toujours améliorée.

II. Introduction à CRISP-DM

CRISP-DM, qui signifie Cross-Industry Standard Process for Data Mining, est **une méthode** mise à l'épreuve sur le terrain permettant d'orienter vos travaux d'exploration de données. v En tant que méthodologie, CRISP-DM comprend des descriptions des phases typiques d'un projet et des tâches comprises dans chaque phase, et une explication des relations entre ces tâches. v En tant que modèle de processus, CRISP-DM offre un aperçu du cycle de vie de l'exploration de données.

Le modèle de cycle de vie comporte six phases dotées de flèches indiquant les dépendances les plus importantes et les plus fréquentes entre les phases. La séquence des phases n'est pas strictement établie. De fait, les projets, pour la plupart, passent d'une phase à l'autre en fonction des besoins.

Adaptable, **le modèle CRISP-DM** peut être aisément personnalisé. Ainsi, si votre entreprise cherche à repérer un blanchiment d'argent, vous examinerez certainement une grande quantité de données sans objectif précis concernant la modélisation. Votre travail sera ciblé non sur la modélisation, mais sur l'exploration et la visualisation de données avec pour objectif de découvrir des configurations suspectes parmi les données financières. CRISP-DM vous permet de créer un modèle d'exploration de données adapté à vos besoins.

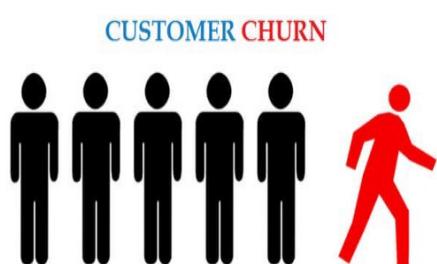
Dans une telle situation, **les phases de modélisation**, d'évaluation et de déploiement peuvent s'avérer d'un intérêt moindre que les phases de préparation et de compréhension des données. Toutefois, certaines des questions soulevées durant ces dernières phases sont tout de même à prendre en considération dans les futurs objectifs **d'exploration de données**.



III. Compréhension du problème métier

La perte de la clientèle ou d'abonnés est toujours un problème grave pour l'industrie des télécommunications, car les clients n'hésitent pas à désabonner ou de changer l'opérateur, s'ils ne trouvent pas ce qu'ils recherchent.

Les clients veulent certainement des prix compétitifs, et de la valeur ajoutée pour l'argent qu'ils payent et surtout, un service de haute qualité.



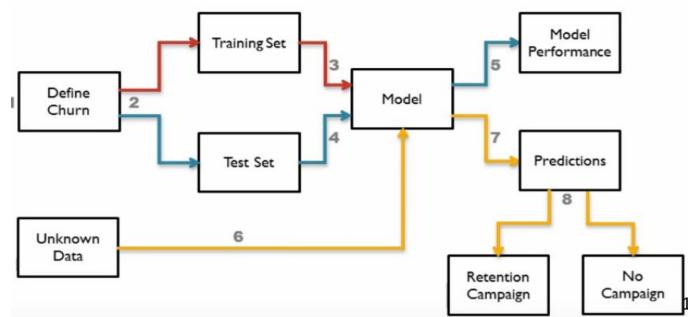
L'analyse Big Data avec le Machine Learning et **Data mining** s'est révélée être un moyen efficace d'identifier et **prédir le désabonnement des clients**.

Afin d'anticiper la rupture des clients avant qu'elle se produise.

How you calculate it?

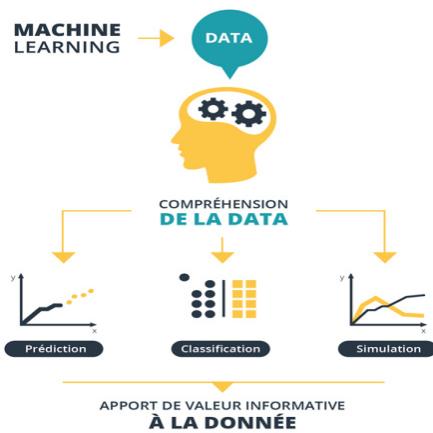
$$\frac{\# \text{ Lost customers}}{\# \text{ New customers}} = \% \text{ Churn}$$

Le churn ou bien la perte de clientèle est directement lié à la satisfaction du client. C'est un fait connu que le coût de l'acquisition d'un client est beaucoup plus élevé que le coût de la fidélisation d'un client, ce qui fait de la rétention des clients un prototype d'entreprise crucial. Il n'existe pas de modèle standard permettant de résoudre avec précision les problèmes de clientèle des fournisseurs de services télécoms mondiaux.



IV. Compréhension des données

Cette phase vise à déterminer précisément les données à analyser, à **identifier** la qualité des données disponibles et à faire le lien entre les données et leur signification d'un point **de vue métier**.

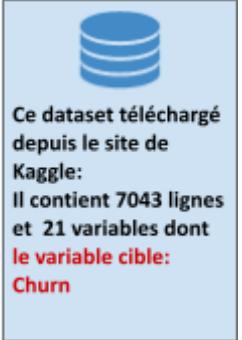


1- Aperçu sur la Base de données

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
7590-VHVEG	Female	0	Yes	No	1	N
5575-GNVDE	Male	0	No	No	34	Ye
3668-QPYBK	Male	0	No	No	2	Ye
7795-CFOCW	Male	0	No	No	45	N
9237-HQITU	Female	0	No	No	2	Ye
TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling		
No	No	No	Month-to-month			
No	No	No	One year			
No	No	No	Month-to-month			
Yes	No	No	One year			
No	No	No	Month-to-month			

7043 rows × 21 columns

Attributs	Type	
customerID	object	Nui
gender	object	sex
SeniorCitizen	int64	Stat
Partner	object	Stat
Dependents	object	Stat
tenure	int64	Pér
PhoneService	object	Éta
MultipleLines	object	Stat
InternetService	object	Stat
OnlineSecurity	object	Éta
OnlineBackup	object	Éta
DeviceProtection	object	Éta
TechSupport	object	Stat
StreamingTV	object	Éta
StreamingMovies	object	Éta
Contract	object	Stat
PaperlessBilling	object	Éta
PaymentMethod	object	Mo
MonthlyCharges	float64	Fra
TotalCharges	object	Fra
Churn	object	Stat



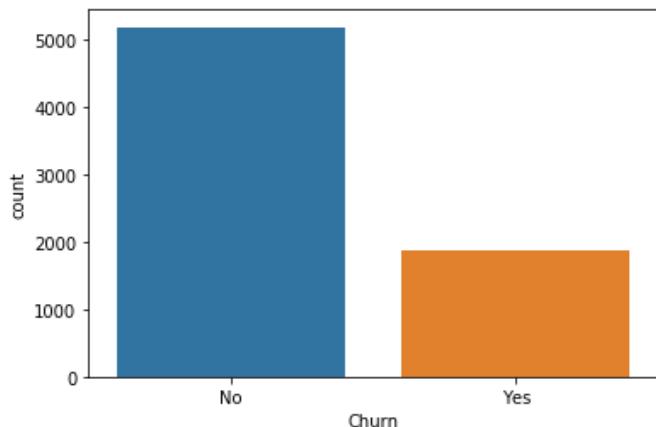
→ Il y a 18 attributs classés comme objet.
Un attribut de type d'objet consiste en des données pour chaque client dans un certain groupe.
Les autres types les données sont int64 et les données de l'attribut peuvent être calculées.

2- Visualisation du pourcentage des clients

73.4630129206304 % des clients restent dans la société.
26.536987079369588 % des clients quittent la société.

Environ **73,46%** des clients sont restés ou ont été retenus et environ **26,54%** des clients ont été désabonnés. C'est une information importante lorsque on essaie d'évaluer notre modèle pour prédire le taux de désabonnement des clients, car cela signifie qu'en supposant toujours qu'un client au hasard a été retenu dans l'ensemble de données, on a **73,46%** de chances de deviner correctement. Par conséquent, on souhaite que la précision du modèle permette de classer ou prédire si le taux de désabonnement d'un client sera supérieur à ce pourcentage.

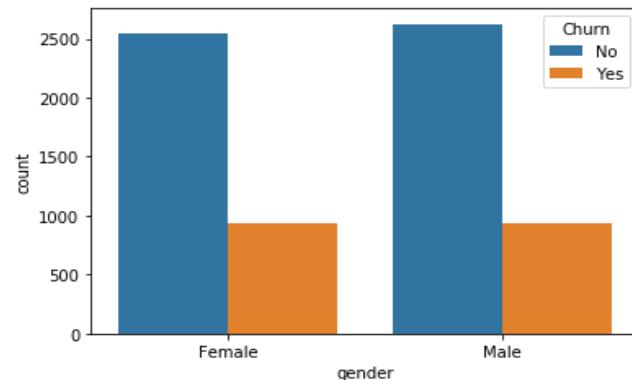
3 - Visualisation du nombres des Churn des clients



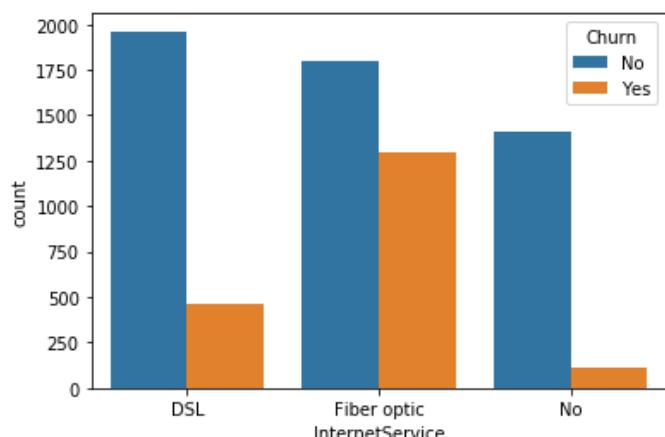
On constate que le nombre des gens qui ont quitté (churn) est inférieur au nombre des clients qui ont resté environ **5147 ont resté** et **1869 ont quitté**.

4-Visualisation du taux de désabonnement pour les hommes et les femmes

D'après l'intrigue ci-dessus, il semble que le **sexé** ne joue pas de rôle dans **le taux de désabonnement des clients**.



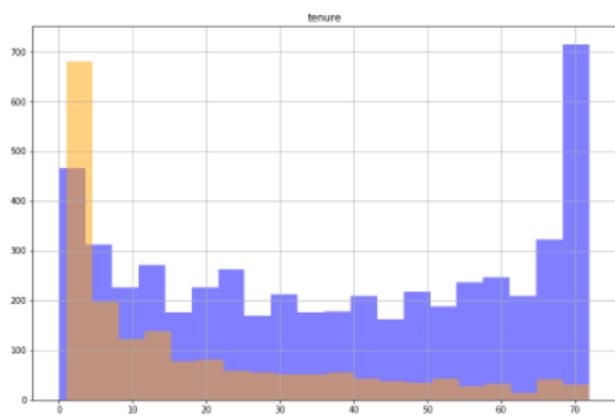
5 - Visualisation du nombre de churn pour le service Internet



Le graphique ci-dessus permet de distinguer les clients retenus et les clients désabonnés, il montre que la plupart des clients qui se sont désabonnés avaient le service Internet à fibre optique et que la plupart des clients qui ont été retenus avaient **un service Internet DSL**. L'entreprise devrait peut-être fournir uniquement le **DSL** comme service Internet ou arrêter de fournir des **fibres optiques** pour son service Internet.

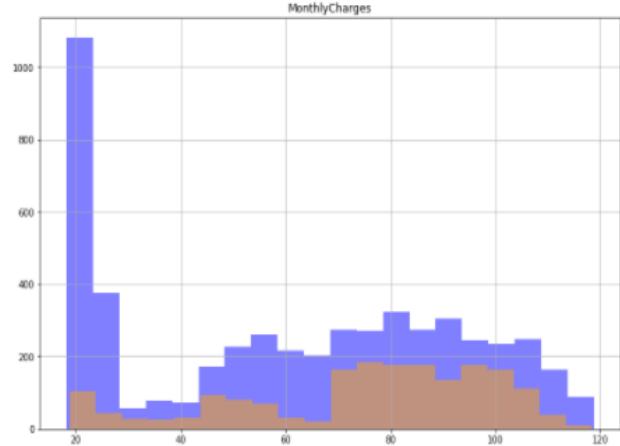
V. Préparation des données

6 - Visualisation du nombre de churn pour le “Monthly Charge” et “tenure”



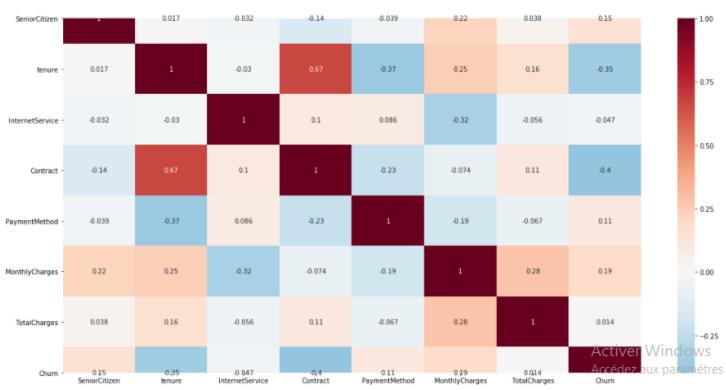
Le graphique ci-dessus de la tenure montre également une certaine discrimination. D'après le graphique, on peut voir que la plupart des clients qui se sont désabonnés ont passé entre 1 et 9 mois dans l'entreprise, tandis que la plupart des clients retenus avaient une ancienneté entre **24 et 72 mois**, soit **2 à 6 ans**.

Il peut donc être dans l'intérêt des entreprises de tout mettre en œuvre pour fidéliser leurs clients pendant **au moins 2 ans**.



D'après la deuxième graphique ci-dessus, on peut clairement voir qu'il y a une certaine discrimination dans les données. Le graphique des **frais mensuels** montre que la plupart des clients fidèles qui sont restés avec l'entreprise avaient **des frais mensuels compris entre 20 \$ et 30 \$**. La plupart des clients qui se sont désabonnés avaient des frais mensuels de **70 \$ à 100 \$**. L'entreprise devrait peut-être réduire les frais mensuels pour fidéliser ses clients.

7- Visualisation de la corrélation entre les variables



La **corrélation positive** la plus forte avec la variable cible “Churn” est les frais mensuels “**Monthly Charges**” et la Méthode de paiement “**Payment Method**”, tandis que la **corrélation négative** est avec le “**tenure**” , **contrat** et le **service internet** .

Le **data preprocessing** est l'une des étapes les plus importantes pour développer des modèles avec de bonnes performances.

1-Encodage

Les algorithmes d'apprentissage automatique fonctionnent mieux en termes de précision et d'autres mesures de performance lorsque les données sont représentées sous forme de nombre au lieu d'être catégoriques dans un modèle pour la formation et les tests.

Les techniques d'apprentissage automatique s'attendent à ce que les données soient numériques. Ainsi, les données catégorielles doivent être encodées en nombres avant de pouvoir les utiliser pour ajuster et évaluer un modèle.

original dataset			dataset with encoded labels		
X ₁	X ₂	y	X ₁	X ₂	y
5	8	calabar	5	8	0
9	3	uyo	9	3	2
8	6	owerri	8	6	1
0	5	uyo	0	5	2
2	3	calabar	2	3	0
0	8	calabar	0	8	0
1	8	owerri	1	8	1

LabelEncoder
→

```
{
    "calabar" --> 0
    "owerri" --> 1
    "uyo" --> 2
}
```

SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Partner_Yes	Dependents_Yes	PhoneService_Yes	MultipleLines_No phone service	MultipleLines_Yes
0	0	1	29.85	29.85	1	0	0	1
1	0	34	56.95	1889.50	0	0	1	0
2	0	2	53.85	108.15	0	0	1	0
3	0	45	42.30	1840.75	0	0	0	1
4	0	2	70.70	151.65	0	0	1	0
...
7038	0	24	84.80	1990.50	1	1	1	0
7039	0	72	103.20	7362.90	1	1	1	0
7040	0	11	29.60	346.45	1	1	0	1
7041	1	4	74.40	306.60	1	0	1	0
7042	0	66	105.65	6844.50	0	0	1	0

7043 rows × 29 columns

Activator Windows

La capture ci-dessus illustre la dataset après l'encodage .

En effet on a appliqué le **Label Encoder** sur la **variable target (Churn)** car (churn) est une variable catégorique à deux valeurs {0,1} . Ainsi on a appliqué **get_dummies** sur les **mesures** afin de rendre toute la dataset **numérique**.



2-Normalisation

La normalisation des notations signifie l'ajustement des valeurs mesurées à différentes échelles à **une échelle théoriquement** commune, souvent avant la moyenne. En effet c'est une méthode de prétraitement des données qui permet de **réduire la complexité** des modèles. C'est également un préalable à l'application de certains algorithmes. Elle **standardise la moyenne et l'écart-type** de tout type de distribution de données, ce qui permet de simplifier le problème d'apprentissage en s'affranchissant de ces deux paramètres.

MIN MAX SCALING

Rescales feature values to between 0 and 1

$$X'_i = \frac{X_i - \min(x)}{\max(x) - \min(x)}$$

Original value X_i

Minimum value in feature

Maximum value in feature

Chris Albon

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Partner_Yes	Dependents_Yes	PhoneService_Yes	MultipleLines_No phone service	MultipleLines_Yes
0	0.00	0.01	0.12	0.00	1.00	0.00	0.00	1.00	0.00
1	0.00	0.47	0.39	0.22	0.00	0.00	1.00	0.00	0.00
2	0.00	0.03	0.35	0.01	0.00	0.00	1.00	0.00	0.00
3	0.00	0.62	0.24	0.21	0.00	0.00	0.00	1.00	0.00
4	0.00	0.03	0.52	0.02	0.00	0.00	1.00	0.00	0.00
...
7038	0.00	0.33	0.66	0.23	1.00	1.00	1.00	0.00	1.00
7039	0.00	1.00	0.85	0.85	1.00	1.00	1.00	0.00	1.00
7040	0.00	0.15	0.11	0.04	1.00	1.00	0.00	1.00	0.00
7041	1.00	0.06	0.56	0.03	1.00	0.00	1.00	0.00	1.00
7042	0.00	0.92	0.87	0.79	0.00	0.00	1.00	0.00	0.00

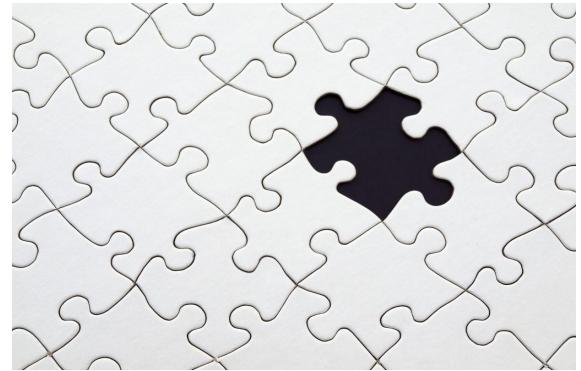
7043 rows × 29 columns

Activer Windows

→ La dataset ci-dessus représente une plage fixe , généralement les valeurs par la valeur maximale et la valeur minimale et en divisant par la différence maximale et minimale.

3-Imputation

L'**imputation des données** manquantes réfère au fait qu'on remplace les valeurs manquantes dans le jeu **de données** par des valeurs artificielles. Idéalement, ces remplacements ne doivent pas conduire à une altération sensible de la distribution et la composition du jeu **de données**.



```
SeniorCitizen  
tenure  
MonthlyCharges  
TotalCharges  
Partner_Yes  
Dependents_Yes  
PhoneService_Yes
```

→ Lors du prétraitement, la variable Total Charges est convertie en un type de données numérique.
On a détecté 11 valeurs de données sont manquantes à partir des lignes Total Charges.

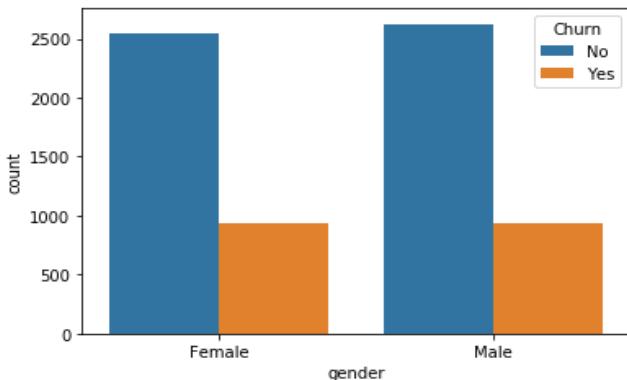


tenure	0
MonthlyCharges	0
TotalCharges	0
Partner_Yes	0

→ Pour résoudre ce problème on à appliquer la méthode "fillna" en effet on a remplacer les valeurs manquantes des lignes Total charges avec la valeur moyenne (mean)

4-Sélection

La sélection de feature est un **processus** utilisé en apprentissage automatique et en traitement de données.
Il consiste, étant donné des données dans un **espace de grande dimension**, à trouver un sous-ensemble de variables pertinentes.



→ Dans le graphe (à gauche) on constate que la variable gender présente une égalité entre les 2 pourcentages (homme et femme), donc la variable "gender" et "customerID" n'ont pas d'effet sur le phénomène Churn

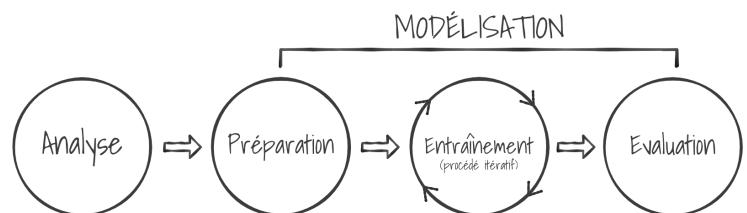
thumb Aperçu sur la base après avoir supprimer les variables

	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport
0	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	No
1	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No
2	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No
3	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes
4	0	No	No	2	Yes	No	Fiber optic	No	No	No	No
...
7038	0	Yes	Yes	24	Yes	Yes	DSL	Yes	No	Yes	Yes
7039	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes	Yes	No
7040	0	Yes	Yes	11	No	No phone service	DSL	Yes	No	No	No
7041	1	Yes	No	4	Yes	Yes	Fiber optic	No	No	No	No
7042	0	No	No	66	Yes	No	Fiber optic	Yes	No	Accédez aux paramètres pour activer Windows	Yes

thumb Taille du nouvelle Dataset

VI. Modélisation

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**. Ce processus est d'abord descriptif pour générer de la connaissance, en expliquant pourquoi les choses se sont passées. Il devient ensuite prédictif en expliquant ce qu'il va se passer, puis prescriptif en permettant d'optimiser une situation future.

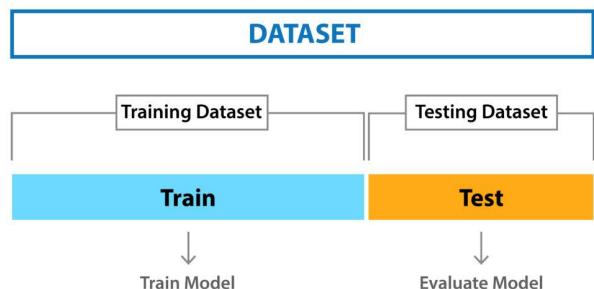


L'entraînement et le test de l'ensemble de données seront utilisés dans cette étude pour modéliser les données à l'aide de l'algorithme proposé.

Le rapport est de **80-20** pour **KNN** et **l'arbre de décision**, **70-30** pour la **régression logistique** et de **75-25** pour **SVM** et **XGboost**.

Basé sur la théorie, la technique évaluera les modèles prédictifs en divisant l'original échantillon dans un ensemble d'apprentissage pour **entraîner le modèle** et un **ensemble de test pour l'évaluer**.

La moyenne de tout le processus est produit comme résultat du modèle.



Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
NearestCentroid	0.71	0.75	0.75	0.73	0.04
BernoulliNB	0.74	0.75	0.75	0.75	0.06
GaussianNB	0.68	0.73	0.73	0.70	0.06
QuadraticDiscriminantAnalysis	0.72	0.71	0.71	0.74	0.06
AdaBoostClassifier	0.81	0.71	0.71	0.80	0.64
GradientBoostingClassifier	0.80	0.71	0.71	0.79	1.81
SGDClassifier	0.78	0.70	0.70	0.78	0.11
LogisticRegression	0.79	0.70	0.70	0.78	0.10
CalibratedClassifierCV	0.79	0.70	0.70	0.78	3.52
LinearDiscriminantAnalysis	0.78	0.70	0.70	0.78	0.51
LinearSVC	0.79	0.70	0.70	0.78	1.14
Perceptron	0.77	0.70	0.70	0.77	0.05
RidgeClassifierCV	0.79	0.70	0.70	0.78	0.08
RidgeClassifier	0.79	0.70	0.70	0.78	0.06
LogisticRegressionCV	0.79	0.70	0.70	0.78	1.12
RandomForestClassifier	0.79	0.69	0.69	0.78	1.04
ExtraTreesClassifier	0.78	0.68	0.68	0.77	1.19
SVC	0.79	0.68	0.68	0.78	1.97
BaggingClassifier	0.78	0.68	0.68	0.77	0.40

Afin de sélectionner les algorithmes avec les meilleures performances nous avons opté pour utiliser “Lazy Predict” qui va nous aider à comprendre quels modèles fonctionnent le mieux sans aucun réglage de paramètre.

On a choisi les algorithmes ci-dessous en se basant sur les accuracy observés dans la figure et en tenant compte aux fait qu'ils sont les plus utilisés :

- ★ **KNN** avec un accuracy **0.76**.
- ★ **Arbre de décisions** avec un accuracy **0.73**.
- ★ **Régression logistique** avec un accuracy **0.79**.
- ★ **XG boost** avec un accuracy **0.80**.
- ★ **SVM Classifier** avec un accuracy **0.79**.

❖ Choix des algorithmes :

👍 Afin de résoudre le problème de “Customer Churn” nous avons opté à utiliser cinq Algorithmes :



1- K-Nearest Neighbor(KNN)

L'algorithme **K-NN** (K-nearest neighbors) est une méthode d'apprentissage supervisé. Il peut être utilisé aussi bien pour la régression que pour la classification. Son fonctionnement peut être assimilé à l'analogie suivante "dis moi qui sont tes voisins, je te dirais qui tu es..." .

Pour effectuer une prédiction, l'algorithme **K-NN** ne va pas calculer un modèle prédictif à partir d'un Training Set comme c'est le cas pour la régression logistique ou la régression linéaire. En effet, K-NN n'a pas besoin **de construire un modèle prédictif**. Ainsi, pour K-NN il n'existe pas de phase d'apprentissage proprement dite. C'est pour cela qu'on le catégorise parfois dans le **Lazy Learning**. Pour pouvoir effectuer une prédiction, K-NN **se base sur le jeu de données** pour produire un résultat.

En effet nous avons appliqué l'algorithme **K Neighbors classifier** pour prédire les gens qui ont **churné** et les gens qui **n'ont pas churné** ensuite nous avons entraîné notre modèle et finalement nous avons calculé le score et on a obtenu **un score d'entraînement égale à 0.996** et **un score de test égale à 0.721** .

→ **On remarque que notre modèle arrive à bien classer les gens qui ont resté l'entreprise et les gens qui ont quitté l'entreprise avec un test score égal à 0.72**
On note un overfitting

Nous avons appliqué la méthode de validation croisée (**« cross-validation »**) afin d'optimiser les hyperparamètres de notre algorithme (KNN) , c'est une méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage .



```
Le train score est 0.9968051118210862  
Le test score est 0.7217885024840313
```

Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Puis nous avons fait recours à une méthode de sélection des variables les plus significatives d'abord nous avons vérifié la variance de chaque variable comme illustré dans la figure (à droite) , après nous avons appliqué la méthode («**Variance Threshold**») et finalement nous avons obtenu **un score d'entraînement égale à 0.83 et un test score égale à 0.75**

le score de train est 0.834043308484203
le score de test est 0.752306600425834

Afin d'améliorer le score de notre algorithme nous avons réglé les hyperparamètres de ce dernier avec la méthode "**Randomized Search CV**", nous avons opté à une combinaison adéquate avec **un type d'algorithme "ball_tree"** et **24 voisins** et **une métrique de Minkowski qui vaut 4.**

SeniorCitizen	0.14
tenure	0.12
MonthlyCharges	0.09
TotalCharges	0.07
Partner_Yes	0.25
Dependents_Yes	0.21
PhoneService_Yes	0.09
MultipleLines_No phone service	0.09
MultipleLines_Yes	0.24
InternetService_Fiber optic	0.25
InternetService_No	0.17
OnlineSecurity_No internet service	0.17
OnlineSecurity_Yes	0.20
OnlineBackup_No internet service	0.17
OnlineBackup_Yes	0.23
DeviceProtection_No internet service	0.17
DeviceProtection_Yes	0.23
TechSupport_No internet service	0.17
TechSupport_Yes	0.21
StreamingTV_No internet service	0.17
StreamingTV_Yes	0.24
StreamingMovies_No internet service	0.17
StreamingMovies_Yes	0.24
Contract_One year	0.17
Contract_Two year	0.18
PaperlessBilling_Yes	0.24
PaymentMethod_Credit card (automatic)	0.17
PaymentMethod_Electronic check	0.22
PaymentMethod_Mailed check	0.18
dtype: float64	

le train score est 0.807596734114306
le test score est 0.7792760823278921

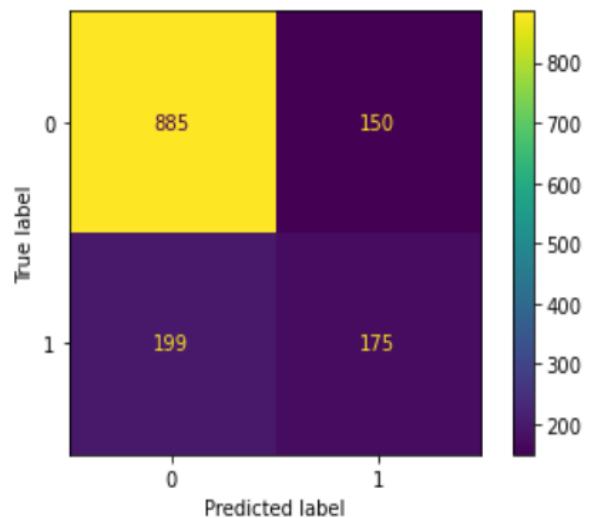
Afin de mesurer la qualité de notre modèle nous avons appliqué **La Matrice de confusion**, c'est un résumé des résultats de prédictions sur un problème de classification. Les prédictions correctes et incorrectes sont mises en lumière et réparties par classe. Les résultats sont ainsi comparés avec les valeurs réelles.

Cette matrice permet de comprendre de quelle façon le modèle de classification est confus lorsqu'il effectue des prédictions. Ceci permet non seulement de savoir quelles sont les erreurs commises, mais surtout le type d'erreurs commises.

Donc d'après cette matrice nous constatons que notre modèle arrive à bien prédire.

Parmi **1035 clients**, **885** n'ont pas quitté l'entreprise ce qui engendre une erreur de prédiction pour **150** tandis que **374 clients** ont quitté l'entreprise.

Notre modèle arrive à bien prédire **175 clients** donc on constate une erreur (False Negative) pour **199 clients** ce qui résulte des mauvaises prédictions.



Pour mesurer la qualité des prédictions de notre modèle nous avons utilisé

Le rapport de classification

Il présente les principales métriques de classification, précision, rappel et score f1 par classe. Les métriques sont calculées en utilisant **des vrais et faux positifs, des vrais et des faux négatifs**.

	precision	recall	f1-score	support
0	0.816421	0.855072	0.835300	1035
1	0.538462	0.467914	0.500715	374
accuracy			0.752307	1409
macro avg	0.677441	0.661493	0.668007	1409
weighted avg	0.742640	0.752307	0.746489	1409

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

D'après ce rapport, nous constatons que **l'accuracy** a une valeur de **0.752** et **le recall** a une valeur faible de **0.467**.

2- Arbre de Décisions (Decision Tree)

L'algorithme **Arbre de décision** ou (Decision Tree algorithm) permet de représenter un ensemble de choix sous la forme graphique **d'un arbre**.

C'est une des méthodes d'**apprentissage supervisé** les plus populaires pour les problèmes de **classification** de données.

Concrètement, un arbre de décision modélise une hiérarchie de tests **pour prédire** un résultat. Il existe deux principaux types d'arbre de décision : Les arbres **de régression** et **Les arbres de classification**.

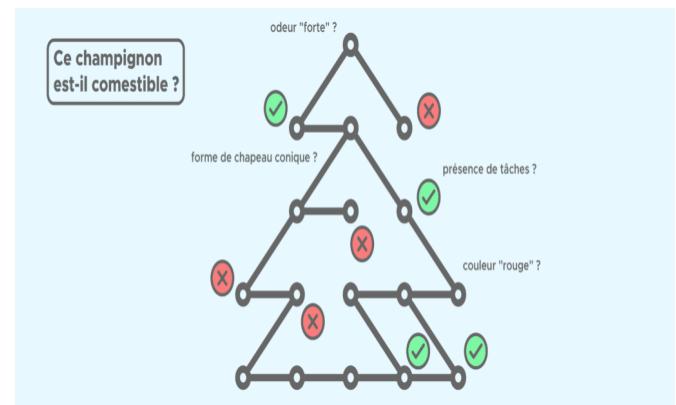
Comme notre projet s'agit d'un problème de classification, on va s'intéresser à ce type d'arbre

permettant de prédire à quelle classe la variable de sortie appartient.

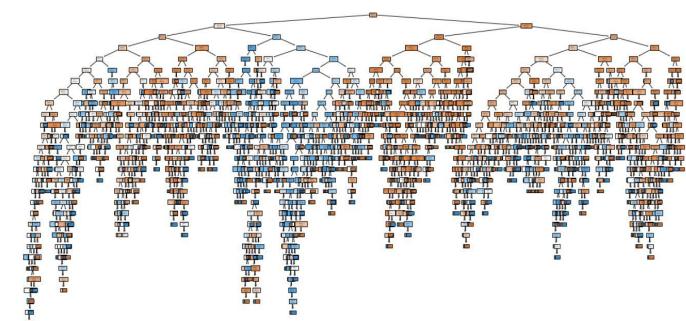
Les décisions possibles sont situées aux extrémités des branches (**les "feuilles" de l'arbre**) et sont atteintes en fonction de décisions prises à chaque étape.

Un arbre de décision fonctionne en appliquant de manière itérative des règles logiques très simples, chaque règle étant choisie en fonction du résultat de la règle précédente.

L'application de cet algorithme sur notre problème revient à prédire si le client va se désabonner des services ou pas, selon des tests réalisés de manière consécutive jusqu'à obtention du résultat.



```
Le train_score= 0.9971600993965212  
Le test_score= 0.7430801987224982
```



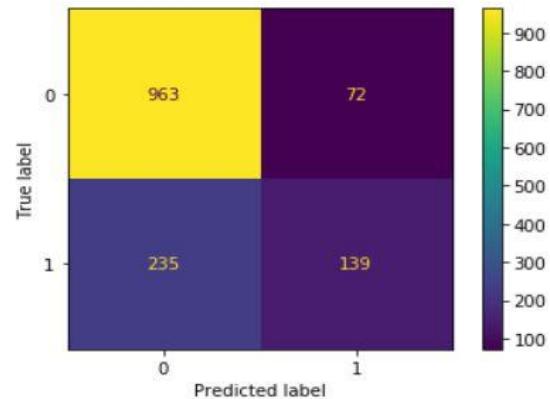
“L’arbre de la première application”

→ On remarque que notre modèle arrive à bien classer les gens qui ont resté l’entreprise et les gens qui ont quitté l’entreprise avec un test score égal à 0.74
On note un **overfitting** et un arbre très complexe

Nous avons procédé alors au réglage des hyperparamètres à l'aide de la méthode "**GridSearchCV**", nous avons obtenu les valeurs 'gini' pour le paramètre criterion et la valeur 2 pour le max_depth. On obtient alors un score d'entraînement de **0.79** et un score de test de **0.78**.

Afin de résumer les résultats de prédictions de notre algorithme, nous avons calculé la matrice de Confusion. Nous avons constaté que les **Faux-Positifs** et les **Faux-Négatifs** sont respectivement **235** et **72**. Notre but est de minimiser les **Faux-Négatifs** qui représentent les clients qui ont Churné et que notre algorithme a échoué de les détecter.

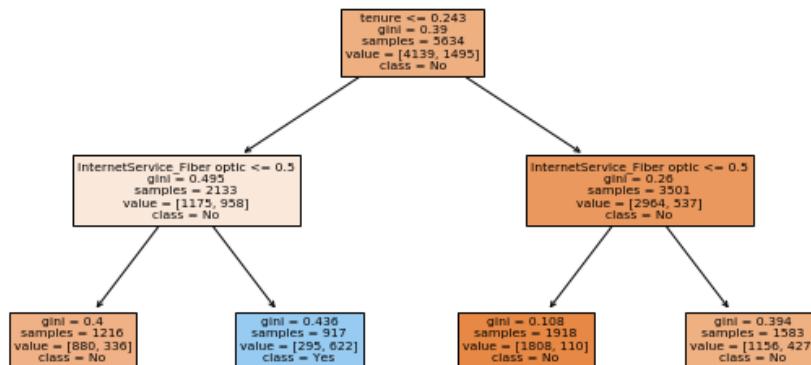
```
Le train_score= 0.7926872559460418
Le test_score= 0.7821149751596878
```



D'après le Rapport de classification, nous avons pu mieux visualiser les mesures de qualité (precision, recall, f1-score, support, accuracy) et il est évident que même si l'**Accuracy** du modèle est de **0.789**, le **recall** est très faible et de valeur **0.451** après l'étape d'optimisation des hyperparamètres avec **Grid Search CV**.

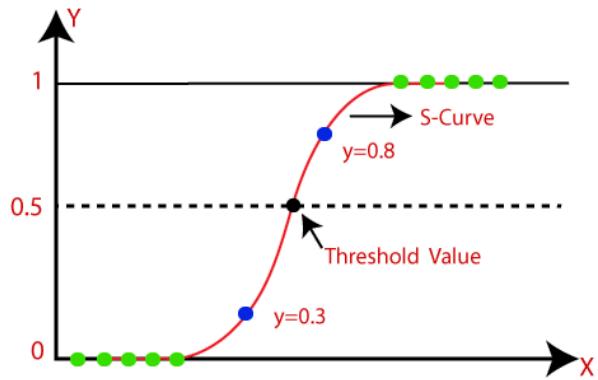
	precision	recall	f1-score	support
0	0.821584	0.912077	0.864469	1035
1	0.650000	0.451872	0.533123	374
accuracy				0.789922
macro avg				0.681974
weighted avg				0.698796
				1409
				1409
				1409

D'où l'obtention du nouvel arbre optimisé:



3- Régression Logistique (Logistic Regression)

La régression logistique est un algorithme supervisé d'apprentissage automatique qui est utilisé pour les **problèmes de classification**, c'est un algorithme d'analyse prédictive basé sur le concept de probabilité.



La fonction Sigmoid où Logistique:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

En effet nous avons appliqué l'algorithme **La régression logistique** pour prédire les gens qui ont **churné** et les gens qui **n'ont pas churné** ensuite nous avons entraîné notre modèle et finalement nous avons calculé le score et on a obtenu **un score d'entraînement égale à 0.804** et **un score de test égale à 0.7979**.

Une régression logistique un modèle de régression linéaire, utilise **une fonction de coût plus complexe**, cette fonction de coût définie par la fonction sigmoïde (fonction logistique) au lieu d'une fonction linéaire.

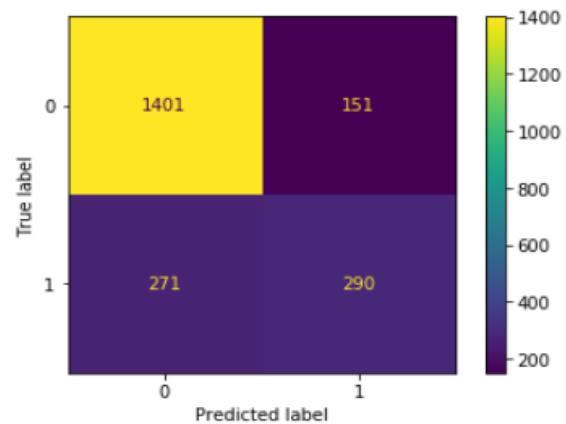
```
Le train_score= 0.8040567951318458  
Le test_score= 0.7979176526265973
```

→ On remarque que notre modèle arrive à bien classer les gens qui ont resté l'entreprise et les gens qui ont quitté l'entreprise avec un test score égal à **0.7979**

Afin d'améliorer les performances de notre modèle nous avons réglé les hyperparamètres de ce dernier avec la méthode "**Randomized Search CV**", nous avons opté à une combinaison adéquate avec **un nombre maximum d'itérations qui vaut 100 et penalty none**.

```
Le train_score= 0.8089249492900609  
Le test_score= 0.8002839564600095
```

Afin de résumer les résultats de prédictions de notre algorithme, nous avons calculé la matrice de Confusion. Nous avons constaté que les **Faux-Positifs** et les **Faux-Négatifs** sont respectivement **151 et 271**. Notre but est de minimiser les **Faux-Négatifs** qui représentent les clients qui ont Churné et que notre algorithme a échoué de les détecter.



D'après **le rapport de classification**, nous constatons que **l'accuracy** a une valeur de **0.800** et le **recall** a une valeur faible de **0.516**.

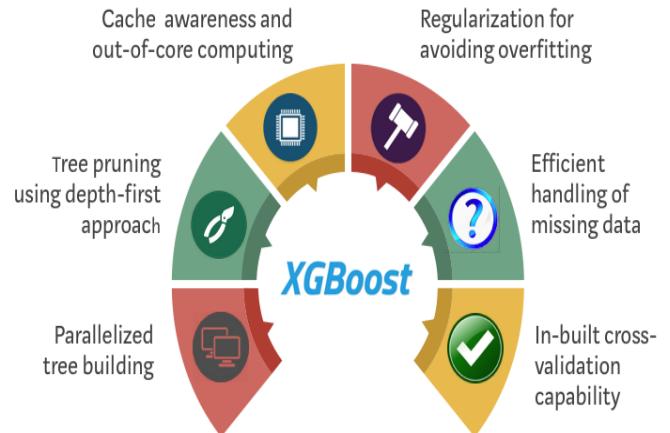
	precision	recall	f1-score	support
0	0.837919	0.902706	0.869107	1552
1	0.657596	0.516934	0.578842	561
accuracy			0.800284	2113
macro avg	0.747758	0.709820	0.723975	2113
weighted avg	0.790043	0.800284	0.792042	2113

4- XGBoost

Le Boosting de Gradient est un algorithme d'apprentissage supervisé dont le principe est de combiner les résultats d'un ensemble de modèles plus simples et plus faibles afin de fournir une meilleure prédiction. On parle d'ailleurs de méthode d'agrégation de modèles.

En fait, c'est un algorithme d'apprentissage automatique basé sur un arbre de décision qui utilise un cadre de renforcement de gradient. Dans les problèmes de prédiction impliquant des données non structurées (images, texte, etc.), les réseaux de neurones artificiels ont tendance à surpasser tous les autres algorithmes ou cadres. Il se base sur le principe d'auto-amélioration séquentielle.

Nous avons ainsi appliqué l'algorithme **XGBoost** pour prédire le Churn des Clients en traînant le modèle et en calculant son score. Nous avons obtenu en premier lieu les résultats ci-illustrés qui sont de valeur **0.935** pour le train score et **0.786** pour le test score.



```
Le train_score= 0.9354411207875805  
Le test_score= 0.7864849517319704
```

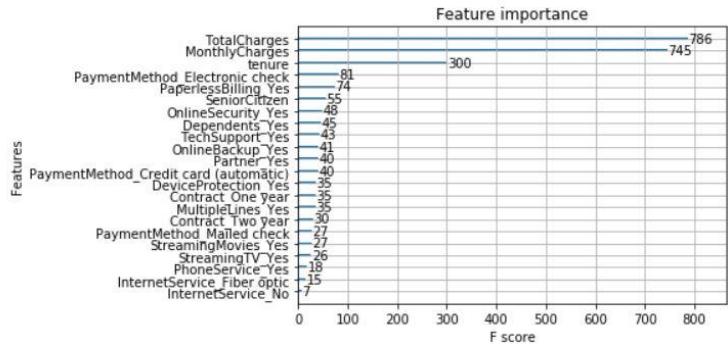
→ **On remarque que notre modèle arrive à bien classer les gens qui ont resté l'entreprise et les gens qui ont quitté l'entreprise avec un test score égale à 0.786**
On note un overfitting

Nous avons procédé alors au réglage des hyperparamètres à l'aide de la méthode "Randomized Search CV", nous avons obtenu les valeurs 0.8 pour "subsample", la valeur 0.01 pour **reg_alpha**, la valeur 10 pour **min_child_weight**, la valeur 4 comme **max_depth**, 0.25 pour le paramètre **gamma** et finalement 0.75 pour le paramètre **colsample_bytree**. On obtient alors un score d'entraînement de **0.826** et un score de test de **0.799**.

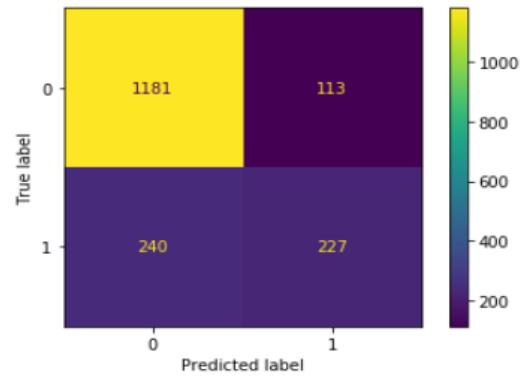
```
Le train_score= 0.8260128739113972  
Le test_score= 0.7995457126632595
```

Une application de feature_importance nous délivre la figure qui nous donne un idée sur l'importance des features dans l'application de notre algorithme.

On peut observer que **TotalCharges** et **MonthlyCharges** sont les deux features les plus importantes avec des valeurs respectivement de **786** et **745**.



Nous procémons maintenant à la matrice de Confusion. Nous avons constaté que les **Faux-Positifs** et les **Faux-Négatifs** sont respectivement **113 et 240**.



D'après **le rapport de classification**, nous constatons que **l'accuracy** a une valeur de **0.799** et le **recall** a une valeur faible de **0.486**.

	precision	recall	f1-score	support
0	0.831105	0.912674	0.869982	1294
1	0.667647	0.486081	0.562577	467
accuracy				0.799546
macro avg	0.749376	0.699378	0.716280	1761
weighted avg	0.787757	0.799546	0.788461	1761

5- SVM Classifier

Le **SVM Classifier** appartient à la catégorie des classificateurs linéaires (**qui utilisent une séparation linéaire des données**), et qui dispose de sa méthode à lui pour trouver **la frontière** entre les catégories.

Pour que le **SVM** puisse trouver cette frontière, il est nécessaire de lui donner des données d'entraînement.

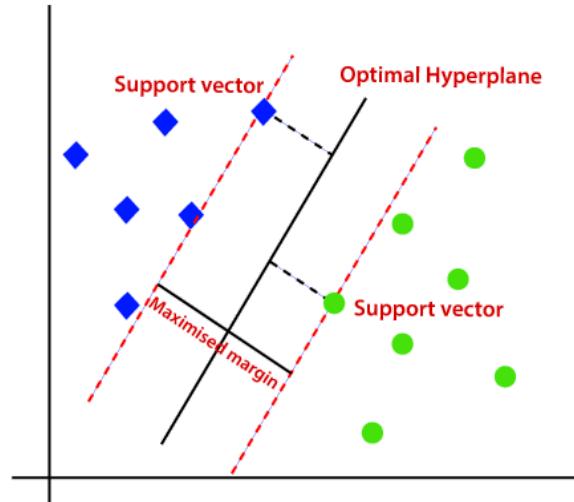
En l'occurrence, on donne au **SVM** un ensemble de points, le **SVM** va estimer l'emplacement le plus plausible de la frontière : c'est la période d'entraînement.

Une fois la phase d'entraînement terminée, le **SVM** a ainsi trouvé, à partir de données d'entraînement, l'emplacement supposé de la frontière.

En quelque sorte, il a « appris » l'emplacement de la frontière grâce aux **données d'entraînement**. Cette frontière est généralement une droite qui sépare les données. Cette dernière doit être distante des données représentées d'une façon équivalente et **maximale**.

Le **SVM** est maintenant capable de prédire à quelle catégorie appartient une entrée qu'il n'avait jamais vue avant, et sans intervention humaine.

En effet nous avons appliqué l'algorithme **SVM classifier** pour prédire les gens qui ont **churné** et les gens qui **n'ont pas churné** ensuite nous avons entraîné notre modèle et finalement nous avons calculé le score et on a obtenu **un score d'entraînement égale à 0.814** et **un score de test égale à 0.789**



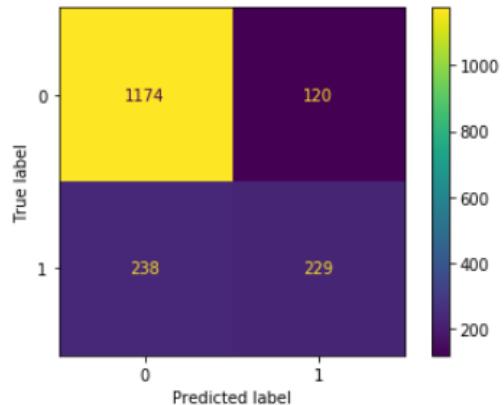
Le score d entraînement 0.812570995834911
Le test score 0.7932992617830777

→ On remarque que notre modèle arrive à bien classer les gens qui ont resté avec un score d'entraînement égale à **0.812** l'entreprise et les gens qui ont quitté l'entreprise avec un test score égale à **0.793**

Afin d'améliorer les performances de notre modèle nous avons réglé les hyperparamètres de ce dernier avec la méthode "**Randomized Search CV**", nous avons opté à une combinaison adéquate avec **un noyau kernel** de type "**rbf**" et un **Paramètre de régularisation C** qui vaut **1000** et **une coefficient du noyau kernel** égale à **0.001**. On obtient alors un score d'entraînement de **0.80** et un score de test de **0.796**.

Finalement pour mieux visualiser la performance du SVM Classifier , nous avons affiché la matrice de Confusion. Nous avons constaté que les **Faux-Positifs** et les **Faux-Négatifs** sont respectivement **120 et 238**. Notre but est de minimiser les **Faux-Négatifs** qui représentent les clients qui ont Churné et que notre algorithme a échoué de les détecter.

```
le score d entraînement 0.8032942067398713
le test score 0.7967064168086314
```



D'où **le rapport de classification** suivant, nous pouvons observer que **l'accuracy** a une valeur de **0.796** et le **recall** a une valeur faible de **0.490**.

	precision	recall	f1-score	support
0	0.831445	0.907264	0.867701	1294
1	0.656160	0.490364	0.561275	467
accuracy			0.796706	1761
macro avg	0.743803	0.698814	0.714488	1761
weighted avg	0.784961	0.796706	0.786440	1761

VII. Evaluation

L'évaluation vise à vérifier le(s) modèle(s) ou les connaissances obtenues afin de s'assurer qu'ils répondent aux objectifs formulés au début du processus. Elle contribue aussi à la décision de déploiement du modèle ou, si besoin est, à son amélioration. A ce stade, on teste notamment la robustesse et la précision des modèles obtenus.



Score de précision par modèle

Modèle	Score d'ent
KNN	0.8
Arbre de décisions	0.7
Régression logistique	0.8
XGBoost	0.8
SVM Classifier	0.8



Les modèles mis en œuvre ont déjà été décrits et analysés séparément, mais l'objectif principal de cette recherche était d'identifier quel modèle peut être considéré comme le meilleur pour résoudre ce problème.

Pour faire cette déclaration, nous devons vraiment comparer tous les modèles formés en ce qui concerne **la courbe ROC** et **la valeur AUC**

Courbe ROC

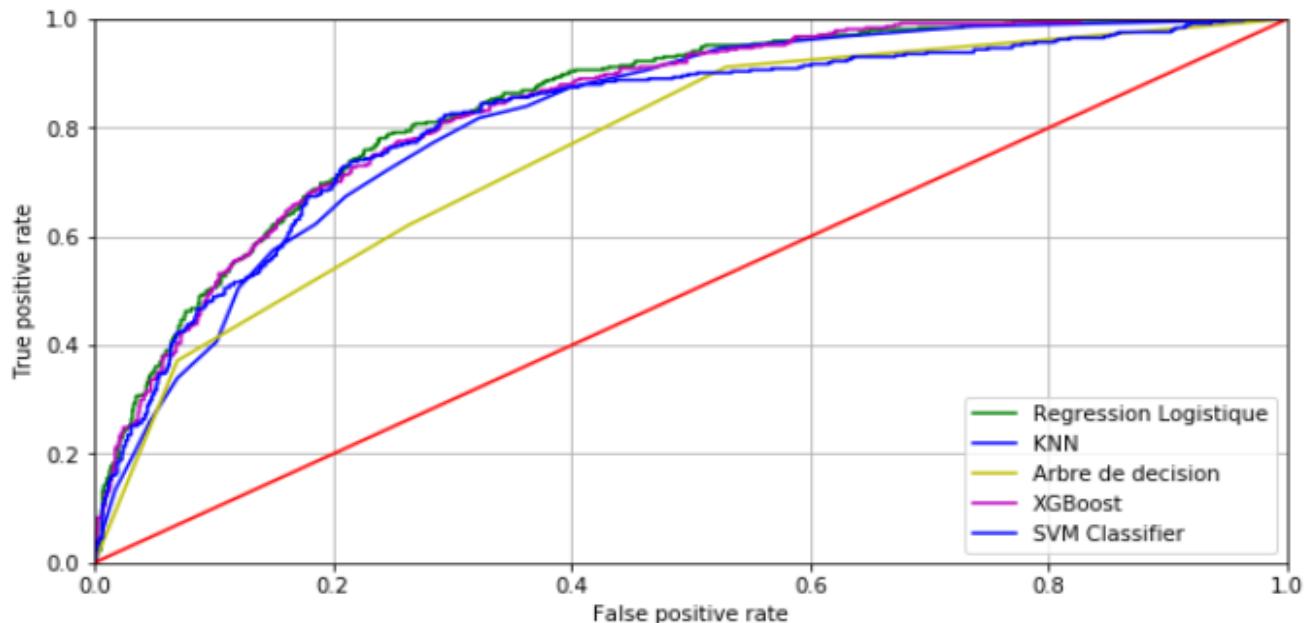
Une **courbe ROC** (receiver operating characteristic) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs :

Le **taux de vrais positifs (TVP)** est l'équivalent du rappel. Il est donc défini comme suit :

$$TVP = \frac{VP}{VP + FN}$$

Le **taux de faux positifs (TFP)** est défini comme suit :

$$TFP = \frac{FP}{FP + VN}$$



Nous constatons que les courbes des cinq modèles sont situés au dessus de la bissectrice ce qui résulte le bon fonctionnement de nos algorithmes mais nous avons remarqué une intersection entre les courbes des algorithmes donc nous ne pouvons pas conclure.

Nous pouvons en revanche calculer efficacement l'aire sous cette courbe, ou AUC, grâce à un algorithme de tri.

AUC : Aire sous la courbe

AUC signifie "**aire sous la courbe ROC**". Cette valeur mesure l'intégralité de l'aire à deux dimensions située sous l'ensemble de la courbe ROC.

L'AUC présente les avantages suivants :

- L'AUC est **invariante d'échelle**. Elle mesure la qualité du classement des prédictions, plutôt que leurs valeurs absolues.
- L'AUC est **indépendante des seuils de classification**. Elle mesure la qualité des précisions du modèle quel que soit le seuil de classification sélectionné.

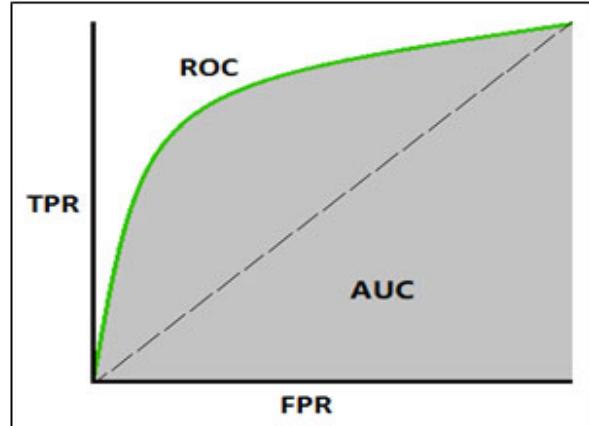


Figure 1: Indicating ROC Curve and AUC

Score de AUC : Aire sous la courbe par modèle

Modèle
KNN
Arbre de décision
Régression logistique
XGboost
SVM Classifier

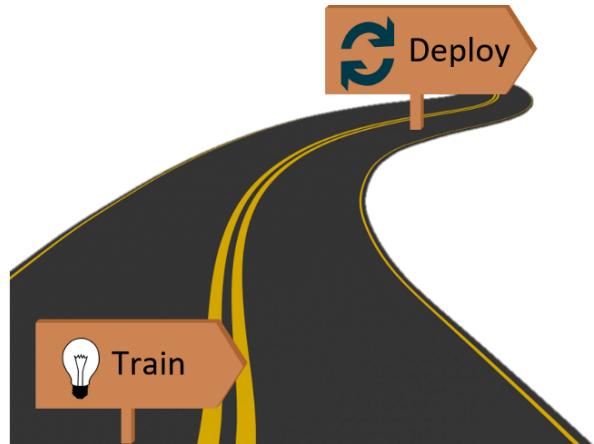


Selon les deux tableaux précédents , les deux algorithmes (**Regression Logistique et XGBoost**) montrent une exactitude relativement bonne et similaire pour les données d'entraînement et de test ainsi pour **l'AUC**

VIII. Déploiement

Il s'agit de l'étape finale du processus. Elle consiste en une mise en production pour les utilisateurs finaux des modèles obtenus. Son objectif : mettre la connaissance obtenue par la modélisation, dans une forme adaptée, et l'intégrer au processus de prise de décision.

Le déploiement peut ainsi aller, selon les objectifs, de la simple génération d'un rapport décrivant les connaissances obtenues jusqu'à la mise en place d'une application, permettant l'utilisation du modèle obtenu, pour la prédiction de valeurs inconnues d'un élément d'intérêt.



1- Logiciel de déploiement

Microsoft Azure est un logiciel de création et déploiement rapide des modèles Machine Learning à l'aide d'outils répondant aux besoins de l'utilisateur.

Il donne l'accès à des notebooks intégrés depuis le studio avec une expérience Jupyter.

Ce logiciel permet de lancer rapidement le calcul dans les notebooks et basculer entre calcul et noyaux en toute simplicité et bien évidemment de déployer vos projets Machine Learning en application.



2- Réalisation

Nous avons opté pour déployer un modèle de prédiction sur Azure ML Studio. La figure ci-joint présente l'interface de notre application. L'utilisateur doit remplir les champs des inputs **Senior Citizen**, **Tenure** et **Monthly Charges**.

Afin de réaliser un test, nous avons attribué la valeur 0 pour l'input "**Senior Citizen**", la valeur 2 pour l'input "**Tenure**" et finalement 70.70 pour "**Monthly Charges**" d'où les résultats suivants:

- ❖ L'application a prédit 1 signifie le client va chiner avec **un score de probabilité de 0.53**.

Test Experiment created on 1/1/2021 [Predictive Exp.] Service

Enter data to predict

SENIORCITIZEN
0

TENURE
2

MONTHLYCHARGES
70.70

← 'Experiment created on 1/1/2021 [Predictive Exp.]' test returned ["1","0.538864374160767"]...

✓ Result: {"Results":{"output1":{"type":"table","value":{"ColumnNames":["Scored Churn Value","Scored Probabilities"],"ColumnTypes":["Double","Double"],"Values":[[["1","0.538864374160767"]]]}}}}

NEW **DELETE**

Test Experiment created on 1/1/2021 [Predictive Exp.] Service

Enter data to predict

SENIORCITIZEN
0

TENURE
66

MONTHLYCHARGES
105

Pour le deuxième test, nous avons maintenant attribué la valeur 0 pour l'input "**Senior Citizen**", la valeur 66 pour l'input "**Tenure**" et finalement 105 pour "**Monthly Charges**" d'où les résultats suivants:

- ❖ L'application a prédit 0 signifie que le client ne va pas chiner avec **un score de probabilité de 0.09**.

← 'Experiment created on 1/1/2021 [Predictive Exp.]' test returned ["0","0.0911586284637451"]...

✓ Result: {"Results":{"output1":{"type":"table","value":{"ColumnNames":["Scored Churn Value","Scored Probabilities"],"ColumnTypes":["Double","Double"],"Values":[[["0","0.0911586284637451"]]]}}}}

NEW **DELETE**

IX. Conclusion et Perspectives

La prédition du churn est l'activité qui consiste à tenter de prévoir les phénomènes de perte de clientèle. Cette prédition et quantification du risque de perte de clientèle peut se faire de manière globale ou individuelle et est surtout utilisée dans les domaines où la commercialisation du produit ou service se fait par abonnement. Pour cela, l'importance de la prédition du churn ne cesse de croître. La collecte de données devient une tâche quotidienne pour toutes les entreprises, et la valeur de ces données peut provenir de sources multiples. La prédition du churn est en train de devenir l'une de ces sources qui génèrent des revenus pour l'entreprise et d'être capable de prévenir quand les clients vont cesser leur contrat avec la société ouvre la possibilité de renégocier ce contrat afin de fidéliser le client.

Par conséquent, cette recherche visait à construire un système qui prédit si un client va chiner ou pas. Ces modèles de prédition doivent atteindre des valeurs AUC élevées. Pour tester et former le modèle, l'échantillon de données est divisé en 2 une partie pour l'entraînement et l'autre les tests. Cinq algorithmes ont été choisis en raison de leur diversité et de leur applicabilité dans ce type de prédition. Ces algorithmes sont Arbre de décision, K-Nearest Neighbor, Régression Logistique XGBOOST et l'algorithme SVM Classifier. Nous avons choisi d'effectuer cross-validation pour la validation et l'optimisation des hyperparamètres.

