IRANIAN CHURN PREDICTION

PYTHON FOR DATA ANALYSIS – A5

CAZIER SIMON - CHASLE JEAN-LUC

SOMMAIRE

- Presentation du problème
- Problématique
- Compréhension et nettoyage des données
- Visualisation des données
- Modélisation des données
- API Flask

PRÉSENTATION DU PROBLÈME

- Face à la concurrence rude dans un domaine en perpétuel expansion, celui de la télécommunication, les opérateurs doivent savoir reconnaître et analyser les attitudes de leurs clients.

- Pour être compétitives, les organisations doivent connaître et prévoir les préférences et les comportements des clients afin de maximiser le nombre de fidélisations avant que leurs concurrents ne le fassent ou que ceux-ci résilient vers un autre opérateur.

- Dans cette étude, on tentera d'identifier les facteurs (paramètres) qui influent sur le taux de résiliation (colonne Churn) afin de permettre à l'opérateur de conserver ses clients et d'en acquérir de nouveaux, le plus précieux des actifs d'une organisation .

Problématique

Le but de notre projet est de prédire la résiliation potentielle d'un client ou/et d'étudier les paramètres qui caractérisent la résiliation des clients.

COMPRÉHENSION ET NETTOYAGE DES DONNÉES

- LES DONNÉES REGROUPENT 3150 CLIENTS SÉLECTIONNÉS DE FAÇON ALÉATOIRE DANS LA BASE DE DONNÉES D'UNE COMPAGNIE DE TÉLÉCOMMUNICATION IRANIENNE SUR UN AN.
- 14 VARIABLES DÉCRIVENT LE COMPORTEMENT ET LE STATUT DU CLIENT AVEC UN LABEL CHURN QUI SERA À PREDIRE SI UN CLIENT VA RÉSILIER OU NON

Nom de la variable	Description
Complains	Si le client s'est plaint au moins une fois auprès du réseau de télécommunications
Subscription Length	Le nombre total de mois d'abonnement du client
Charge Amount	Prix de l'abonnement allant du moins cher au plus cher sous forme de catégories
Seconds of use	Le nombre total de secondes de tous les appels d'un client
Frequency of use	Le nombre total d'appels d'un client
Frequency of SMS	Le nombre total de sms d'un client
Distinct Called Numbers	Le nombre total d'appels téléphoniques distincts d'un client
Age Group	Catégorie du client selon son âge
Tariff Plan	Si le forfait est contractuel ou si c'est une carte prépayée
Status	Si le client est actif ou inactif
Customer Value	La valeur du client auprès de l'entreprise (note)
Age	Age du clie
Call Failures	Nombre d'appels échoués
Churn	Variable à prédire (1 = résilier, 0 = Toujours abonné)

COMPRÉHENSION ET NETTOYAGE DES DONNÉES

- APRÈS AVOIR EXPLORER LES DONNÉES, ON CONSTATE QU'IL N'Y A AUCUNE VALEUR MANQUANTE

Vérification s'il existe des valeurs NULL présents dans le dataset

```
nb_null_colonne = df_iranian_churn.isnull().sum()
nb_null_colonne
```

```
Call Failure              0
Complains                 0
Subscription Length       0
Charge Amount             0
Seconds of Use            0
Frequency of use          0
Frequency of SMS          0
Distinct Called Numbers   0
Age Group                 0
Tariff Plan               0
Status                    0
Age                       0
Customer Value            0
Churn                     0
dtype: int64
```

Il n'y a pas de valeurs manquantes dans le dataset donc il n'y pas besoin de gérer les valeurs null

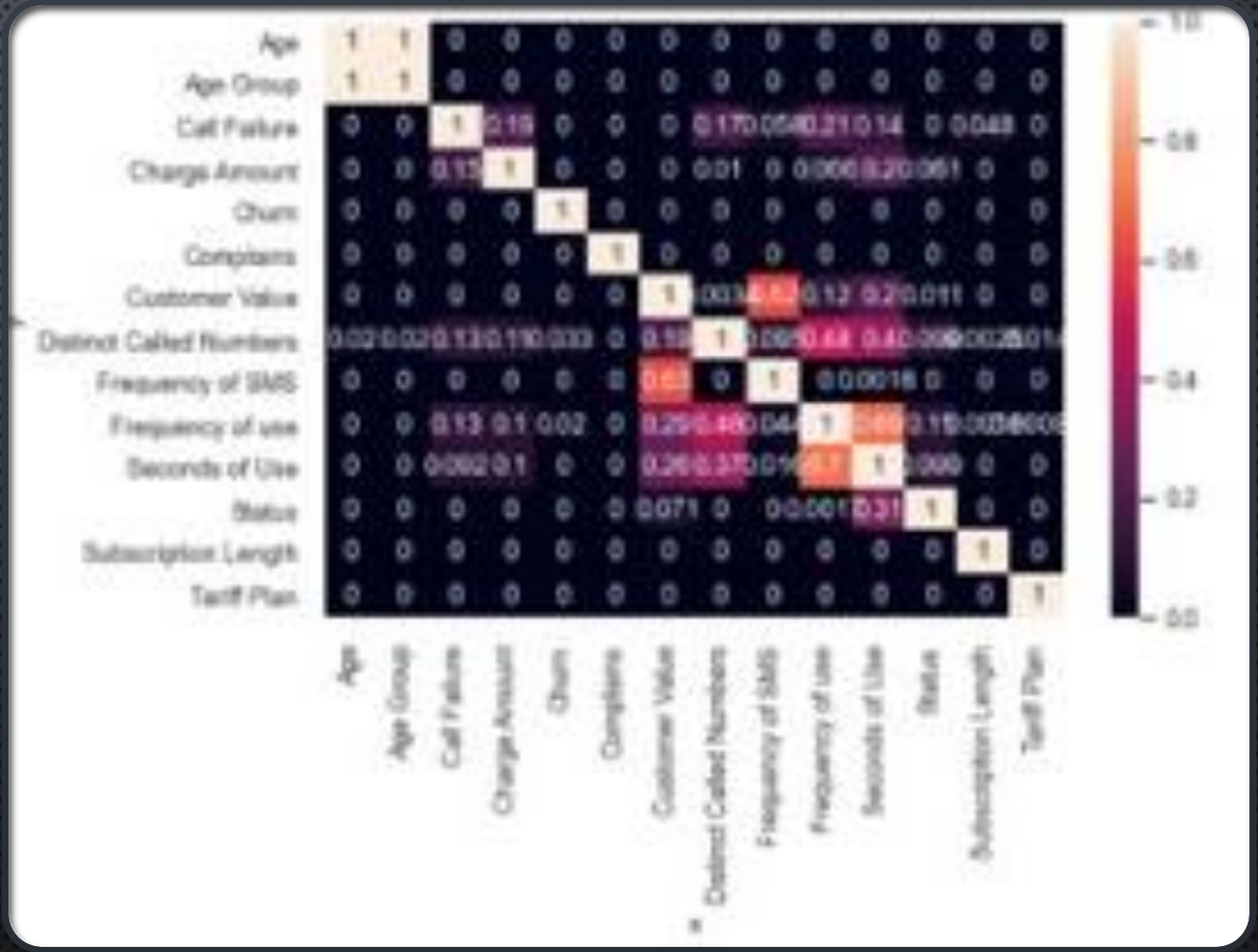
VISUALISATION DES DONNÉES

- La matrice de corrélation nous donne l'impact des différents paramètres sur la variable "Churn".

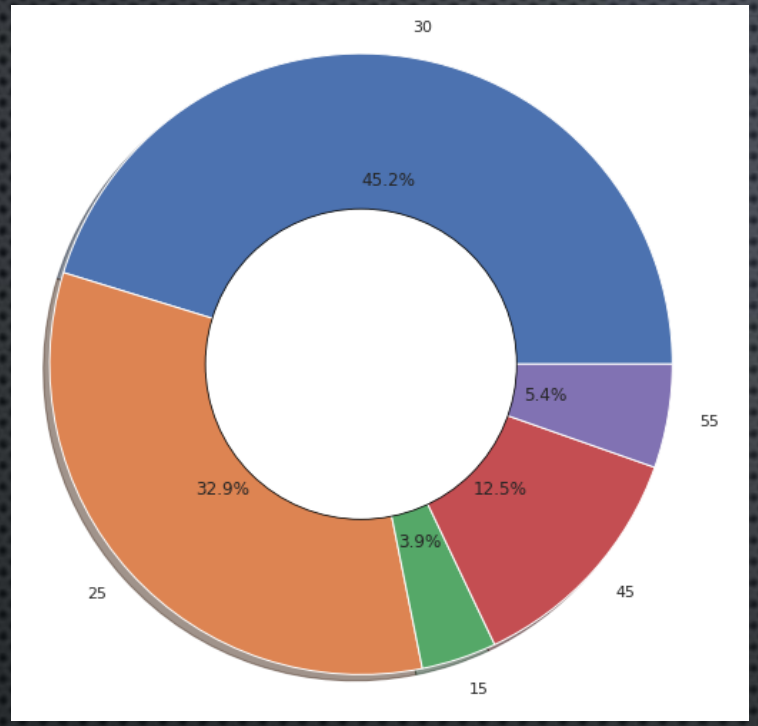
- On peut constater que les variables "Complains" et "Status" sont celles qui influent le plus sur le label de résiliation.

ALTERNATIVE À LA MATRICE DE CORRÉLATION : PPSCORE

- Le PPS est un score asymétrique, de type données-diagnostic, qui permet de détecter des relations linéaires ou non linéaires entre deux colonnes. Le score va de 0 (aucun pouvoir prédictif) à 1 (pouvoir prédictif parfait). Il peut être utilisé comme alternative à la corrélation (matrice).
- Un score de 0 signifie que la colonne x ne peut pas prédire la colonne y mieux qu'un modèle de base naïf.
- Un score de 1 signifie que la colonne x peut parfaitement prédire la colonne y étant donné le modèle.
- Un score entre 0 et 1 indique le rapport entre la puissance prédictive potentielle du modèle et celle du modèle de base.

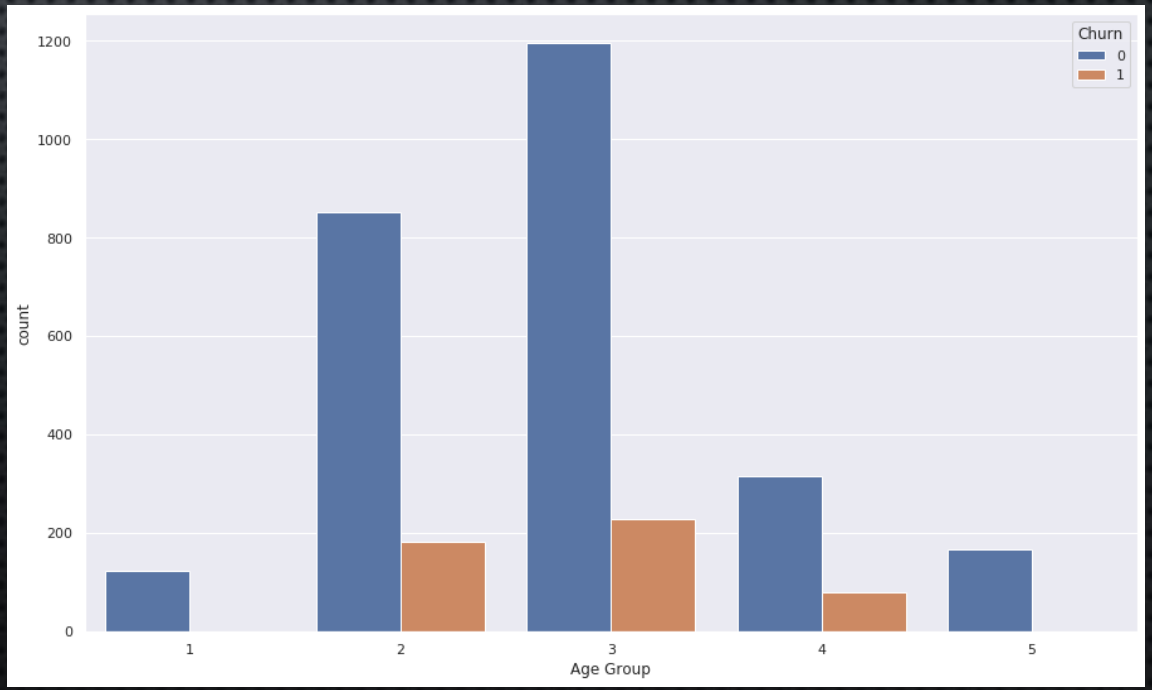


Distinct called number et frequence of use semblent potentiellement prédire la variable cible

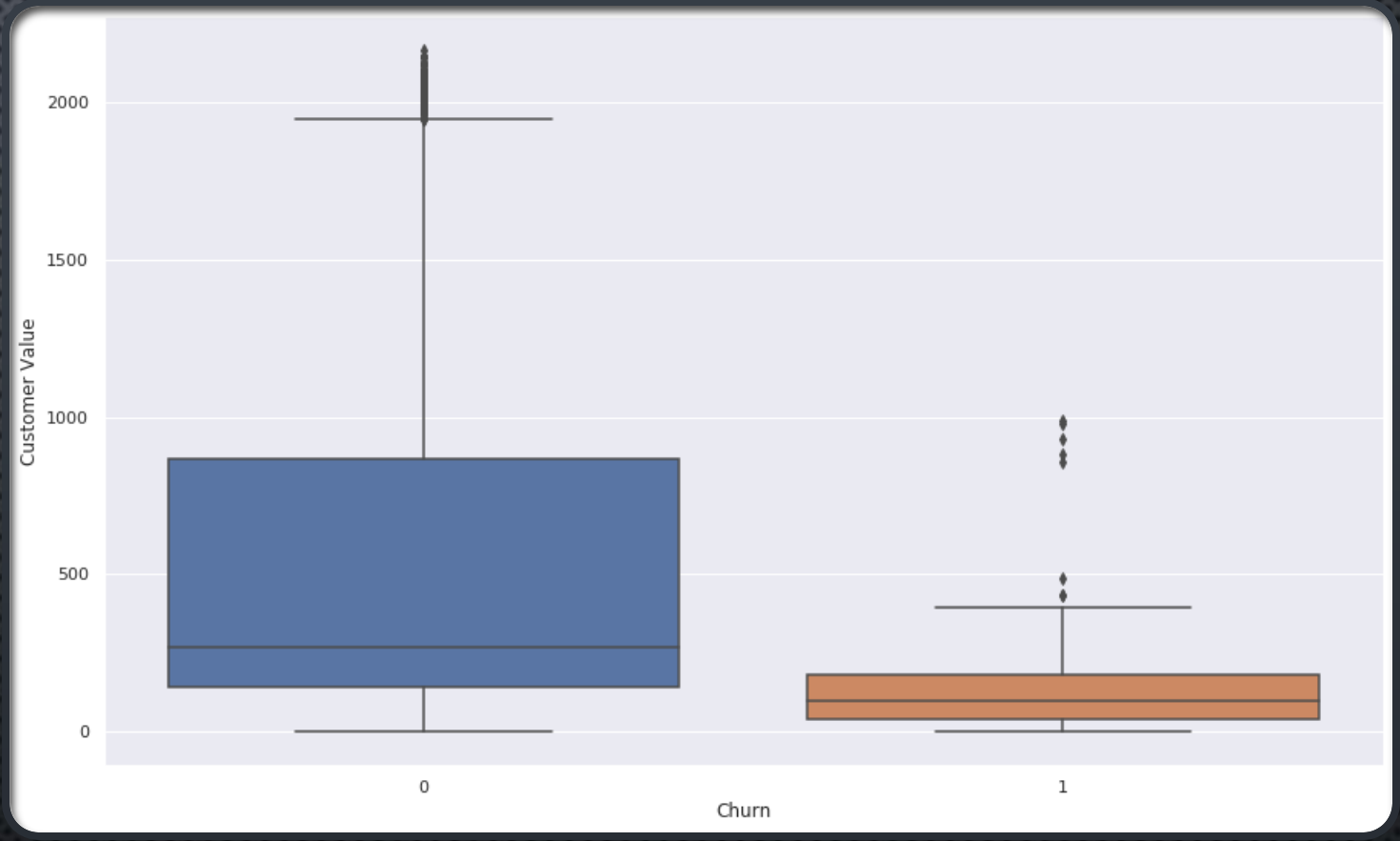


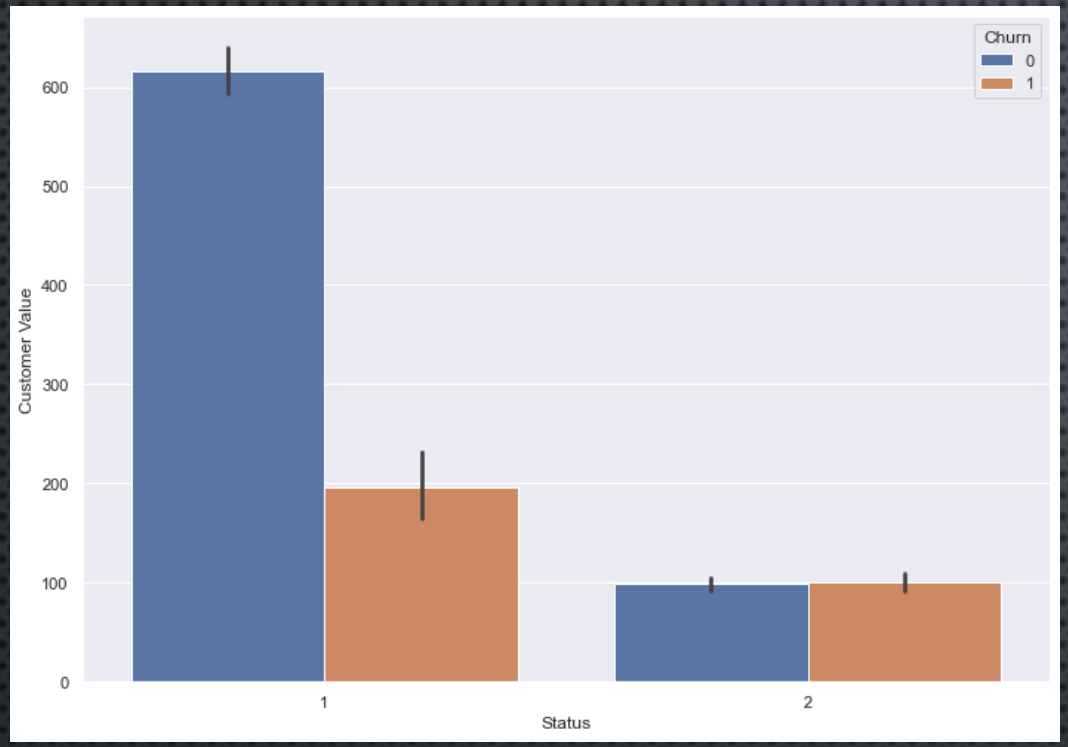
Ce graphe représente la répartition de la clientèle en fonction de leur âge au sein du dataset.
La majorité des utilisateurs sont des personnes entre 25 et 30 ans.

Ce second graphe représente le nombre de résiliations de clients par groupe d'âge.
On peut constater que les clients entre 25 et 45 ont plus tendance à résilier leur abonnement que le reste des autres clients.



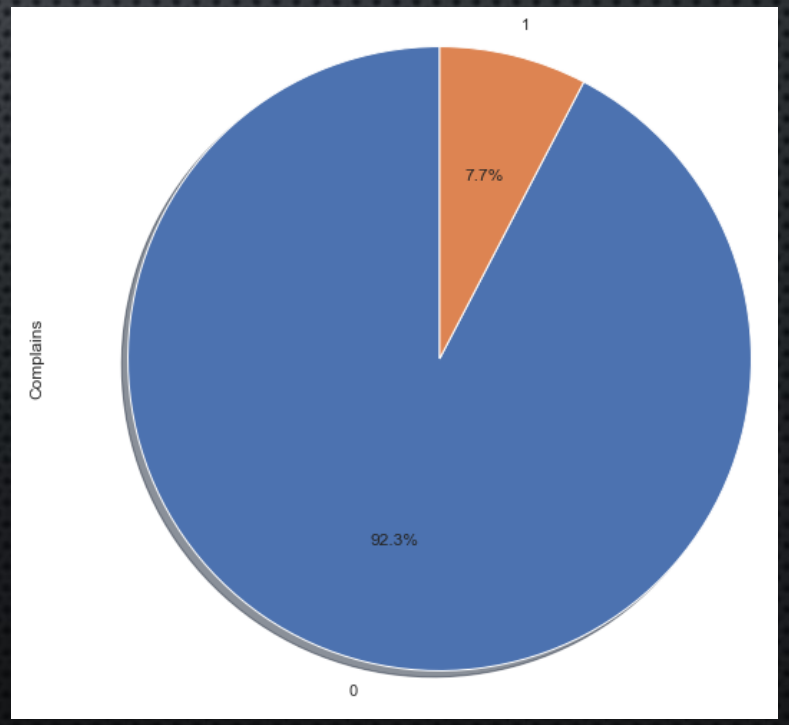
- Le graphique de droite nous indique que le "Customer Value" (valeur du client auprès de l'entreprise de télécommunication) a un impact sur la prise de décision d'un client lors de la résiliation de son abonnement.
- En dessous d'un seuil d'environ 200, la quasi-totalité des clients résilient.
- Ceci est donc un bon indicateur pour l'entreprise afin de connaitre les clients qui vont résilier.





Cette représentation du "Customer Value" en fonction des deux paramètres "Status" et "Churn", nous indique que presque tous les utilisateurs non-actifs (status: 2) résilient leurs abonnements

Ce graphe circulaire nous informe que 7.7% des clients qui se sont plaint vont jusqu'à résilier leur abonnement.



Modélisation des données : Machine Learning

- Afin de prédire la valeur churn d'un client, nous allons utiliser 5 algorithmes de classification puisque la valeur cible est catégorique (valeurs discrètes).
- Modèles utilisés :

1. Logistic regression
2. Logistic regression with smote
3. Random forest
4. Knn
5. Gradient boosting

On a découpé par la suite de notre jeu de données en un jeu d'entranement (80% = 2520 lignes) et un jeu de test (20% = 630 lignes)

LOGISTIC REGRESSION

Avec la régression logistique, nous avons une précision de 0.913.

Une précision de 0.92 et de 0.88 pour les valeurs 0 et 1 respectivement.

L'algorithme prédit de manière satisfaisante les labels mais on a un nombre assez conséquent de FP = 48, qui est un danger pour le besoin métier, en effet, on prédit qu'un client ne va pas résilier alors qu'il va résilier ce qui n'arrange pas l'opérateur

```
Accuracy Score:
0.913
Classification report:
              precision    recall  f1-score   support

           0       0.92      0.99      0.95       532
           1       0.88      0.51      0.65        98

    accuracy                           0.91       630
   macro avg       0.90      0.75      0.80       630
weighted avg       0.91      0.91      0.90       630

Confusion matrix:
 [[525   7]
 [ 48  50]]
```

LOGISTIC REGRESSION WITH SMOTE

- La classification déséquilibrée implique le développement de modèles prédictifs sur des ensembles de données de classification qui présentent un grave déséquilibre de classe.

- Le défi de travailler avec des ensembles de données déséquilibrés est que la plupart des techniques d'apprentissage automatique ignorent la classe minoritaire et ont à leur tour de mauvaises performances sur cette classe, bien que ce soit généralement la performance sur la classe minoritaire qui est la plus importante.

- Une approche pour remédier aux ensembles de données déséquilibrés consiste à suréchantillonner la classe minoritaire. L'approche la plus simple consiste à dupliquer des exemples dans la classe minoritaire, bien que ces exemples n'ajoutent aucune nouvelle information au modèle. Au lieu de cela, de nouveaux exemples peuvent être synthétisés à partir des exemples existants. Il s'agit d'un type d'augmentation des données pour la classe minoritaire, appelé "Synthetic Minority Oversampling Technique" (SMOTE).

LOGISTIC REGRESSION WITH SMOTE

- Avec le SMOTE, la regression logistique a une précision plus faible pour le nombre de clients ayant résiliés que celle sans. Le surechantillonage du nombre d'échantillons pour la classe 1 (le client résilie) a entrainé plus de FP qui est moins grave que d'avoir plus de FN.
- Par ailleurs, on constate qu'on a moins de FN ce qui est encourageant et on prédit mieux les personnes qui souhaitent résiliées avec cette méthode.

```
Accuracy Score:
0.837
Classifcation report:
              precision    recall  f1-score   support

           0       0.98      0.82      0.89       521
           1       0.52      0.91      0.66       109

    accuracy                           0.84       630
   macro avg       0.75      0.86      0.78       630
weighted avg       0.90      0.84      0.85       630

Confusion matrix:
 [[428  93]
 [ 10  99]]
```

RANDOM FOREST

Avec la régression logistique, nous avons une précision de 0.946.

Une précision de 0.96 et de 0.89 pour les valeurs 0 et 1 respectivement.

L'algorithme prédit de manière satisfaisante les labels et fait très peu d'erreur.

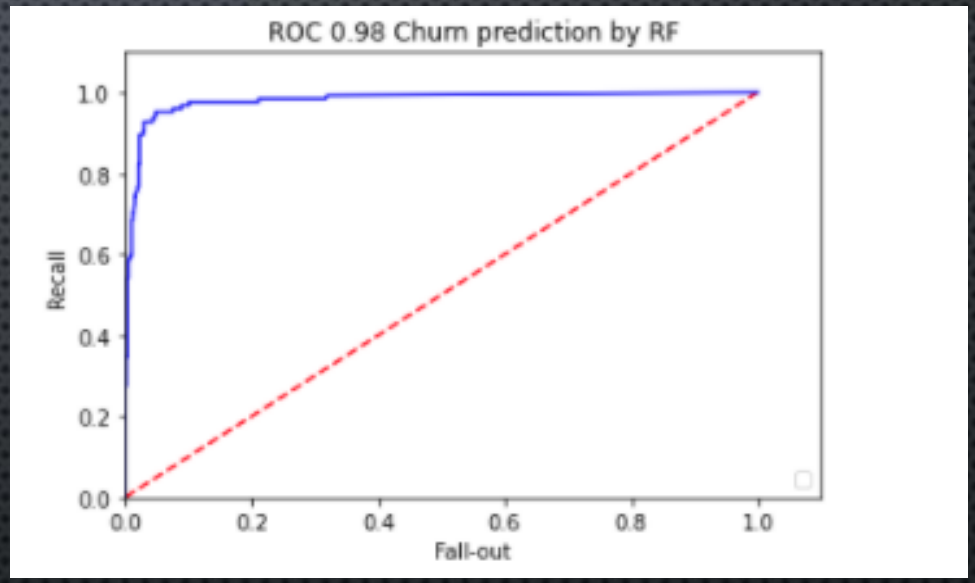
C'est un algorithme que l'on pourrait facilement retenir pour le besoin de l'opérateur

```
Accuracy Score:
0.948
ROC Score:
0.9828487911391292

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       521
           1       0.89      0.80      0.84       109

    accuracy                           0.95       630
   macro avg       0.92      0.89      0.90       630
weighted avg       0.95      0.95      0.95       630


Confusion Matrix:
[[510  11]
 [ 22  87]]
```

RANDOM FOREST AVEC GRILLE DE RECHERCHE

En faisant évoluer les hyperparamètres à l'aide du grid search, la meilleure configuration semble être la suivante : Min_samples_leaf =3, n_estimators = 10, n_jobs = -1 avec une précision de 0.967

On peut apercevoir avec la courbe ROC que le modèle prédit très bien les labels et est très précis.




```
19:41:27
------------------------------
 RandomForestClassifier
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
 best_score_ 0.9471870834085843 with =  RandomForestClassifier(min_samples_leaf=3, n_estimators=10, n_jobs=-1)
19:41:31
```

KNN

Avec le KNN, nous avons une précision de 0.846.

Tout comme la régression logistique, l'algorithme prédit de manière satisfaisante le label 0 qui est en majorité dans ce dataset mais on a un nombre assez conséquent de FP = 63, qui est un danger pour le besoin métier et on prédit mal les clients voulant résilier

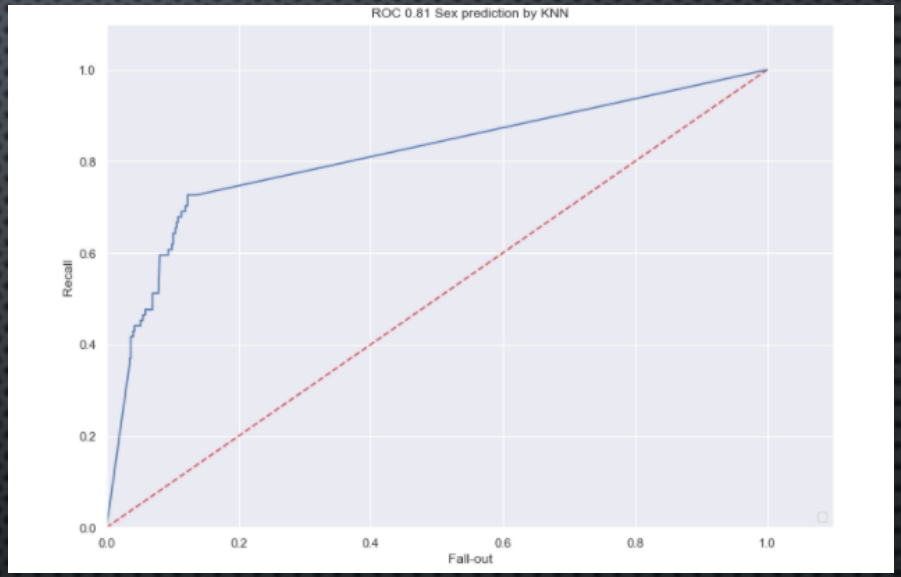
On a une bonne précision globale et un roc score correct mais la remarque ci-dessous pourrait être un argument pour ne pas choisir ce modèle dans le cadre de ce problème

```
Accuracy Score:
0.846
ROC Score:
0.8350208667171459

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       521
           1       0.57      0.42      0.49       109

    accuracy                           0.85       630
   macro avg       0.73      0.68      0.70       630
weighted avg       0.83      0.85      0.84       630


Confusion Matrix:
[[487  34]
 [ 63  46]]
```

KNN AVEC GRILLE DE RECHERCHE

En faisant évoluer les hyperparamètres à l'aide du grid search, la meilleure configuration semble être la suivante :
leaf_size =3, n_neighbors = 2, weights = distance avec une précision de 0.967

On peut apercevoir avec la courbe ROC que le modèle prédit correctement les labels et a une bonne performance dans sa manière de classifier.




```
16:43:10
------------------------------
 knn_grid_search
{'n_neighbors': [2, 3, 5, 8, 10], 'weights': ['uniform', 'distance'], 'algorithm': ['auto', 'ball_tree', 'kd_tree',
'brute'], 'leaf_size': [1, 3]}
 best_score_ = 0.86 with =  KNeighborsClassifier(leaf_size=1, n_neighbors=2, weights='distance')
16:43:25

Duration time : 15.205899953842163
```

GRADIENT BOOSTING

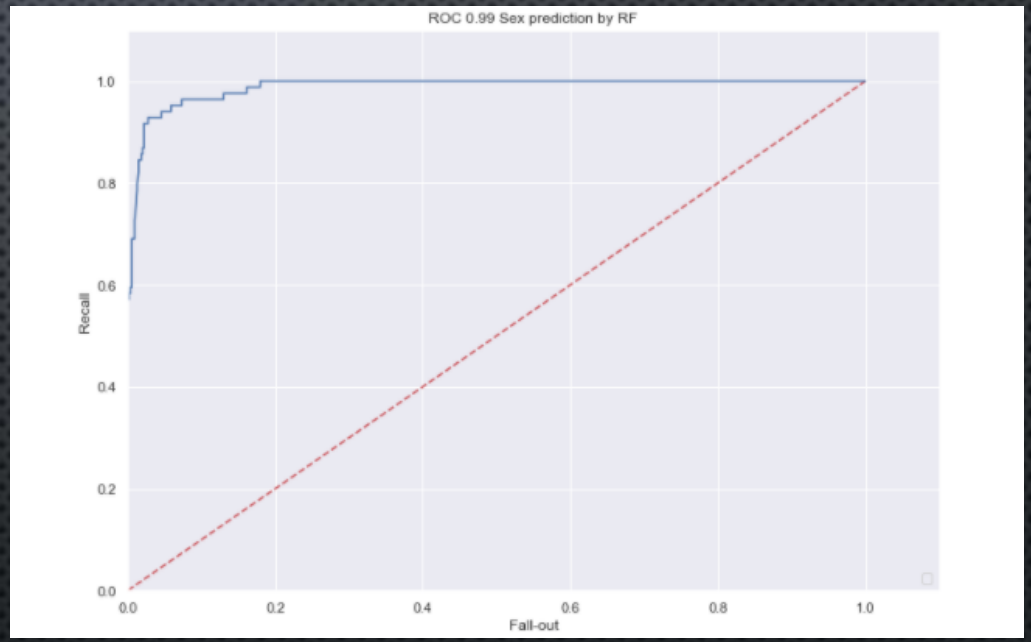
- Avec le gradient boosting, l'algorithme prédit de manière satisfaisante les labels et fait très peu d'erreur tout comme le random forest.
- C'est un algorithme que l'on pourrait facilement retenir pour le besoin de l'opérateur
- Mais elle présente un roc score plus faible que celui du random forest du à une plus faible précision de la prédiction des labels 1 qui concerne la résiliation d'un client mais cette différence est assez minime.

```
Accuracy Score:
0.940
ROC Score:
0.8350208667171459

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.97      0.96       521
           1       0.84      0.81      0.82       109

    accuracy                           0.94       630
   macro avg       0.90      0.89      0.89       630
weighted avg       0.94      0.94      0.94       630


Confusion Matrix:
[[504  17]
 [ 21  88]]
```

GRADIENT BOOSTING AVEC GRILLE DE RECHERCHE

En faisant évoluer les hyperparamètres à l'aide du grid search, la meilleure configuration semble être la suivante :
'n_estimators' = 10 , 'max_depth' = None , 'min_samples_split'=2, 'learning_rate'= 0.01 , 'loss':'ls' avec une précision de 0.94

On peut apercevoir avec la courbe ROC que le modèle prédit très bien les labels et est très précis.




```
16:45:06
------------------------------
 clf_grid_search
{'n_estimators': 10, 'max_depth': None, 'min_samples_split': 2, 'learning_rate': 0.01, 'loss': 'ls'}
 best_score_ = 0.94 with =  GradientBoostingClassifier()
16:45:09

Duration time : 3.7153279781341553
```

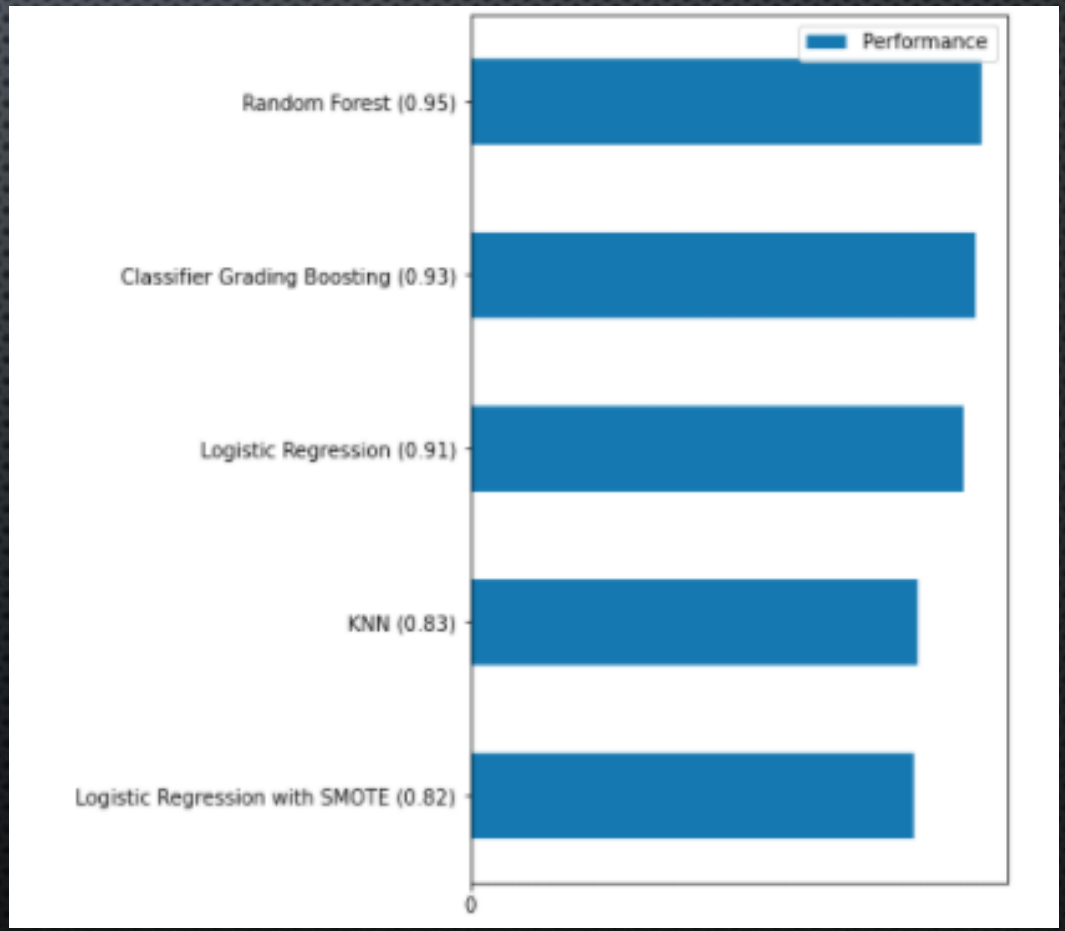
RÉSUMÉ : COMPARAISONS DES ALGORITHMES

En résumé des algorithmes utilisés dans le cadre de ce problème, tous les algorithmes ont une performance assez élevé.

Mais comme vu précédemment, certains algorithmes sont plus performants et plus prédisposés à prédire la variable cible. Le meilleur algorithme est le random forest suivi de près par le gradiant boosting.

Cependant malgré la dernière place de la régression logistique avec le SMOTE, elle semble plus adaptée et prend moins de temps d'éxécution que la régression logistique et le KNN pour le besoin de l'opérateur.

Le random forest serait le meilleur algorithme de par sa capacité à prédire correctement les labels pour l'opérateur et par sa rapidité d'éxécution



API FLASK

L'API permet de tester des données selon les différents modèles présentés plus haut. Pour sélectionner un modèle spécifique, il suffit de préciser le type de modèle en paramètre.

API Endpoint:

- /api/ Méthode: POST
  Prédiction de la résiliation ou non d'un client en fonction d'un modèle défini dans les paramètres.

Format JSON à envoyer:

```
{
    'values': [[8,0, 38, 0, 4370, 71, 5, 17, 3, 1, 1, 30, 69.764]],
    'model': 'RF'
}
```

Liste des modèles accessibles:

- LR : Logistic Regression
- LRS : Logistic Regression with SMOTE
- RF : Random Forest
- KNN : K-Nearest Neighbours
- CLF : Gradient Boosting