UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

SCORING

PROJET FINAL

Claire Gefflot
Romain Penichon
Rime Boumezaoued

M2 TIDE
Année 2022/2023

Table des matières

Le présent document à pour objectif de présenter l'ensemble de notre étude de scoring. Le projet porte sur le comportement de clients dans le domaine de la téléphonie et plus particulièrement sur l'identification des churners. Par définition, le churn est défini comme l'entrée en période d'invalidité à savoir l'absence d'appels entrants et sortants. Le but est de trouver le meilleur modèle de score permettant de les distinguer.

La base de données contient 49 071 contrats donc 17 506 churners sur six mois, les données étant figées à fin août. Les churners sont modélisés en novembre soit deux mois après la date d'arrêté. Le jeu de données comprend un total de 77 variables, réparties en 2 variables qualitatives et 75 variables quantitatives. Notre variable cible est "AFTERGRACE_FLAG" soit le passage en invalidité précédemment défini.

I. Analyse exploratoire

La première étape de notre projet consiste en une analyse exploratoire des données. Cette étape comprend plusieurs tâches importantes telles que l'identification du type de données utilisées, la détection et l'évaluation des valeurs manquantes, l'exploration de l'impact des variables explicatives sur la variable cible, ainsi que la construction d'une matrice de corrélation.

L'étude de l'ensemble des variables est réalisé à l'aide de la fonction *df_analyse* et retourne le résultat suivant :

```
Initial Analysis of df dataset
-------------------------------------------------------------------------------
- Dataset shape:                  49071 rows and 77 columns
- Total of NaN values:            13700
- Total of full duplicates rows:  0
- Total of empty rows:            0
- Total of empty columns:         0

- The key(s): ['CONTRACT_KEY'] is not present multiple times in the dataframe.
  It CAN be used as a primary key.

- Type object and records by columns         ( memory usage: 28.8+ MB )
-------------------------------------------------------------------------------
                       Name     Type  Records  % of NaN  Unique
0              CONTRACT_KEY  float64    49071     0.000   49071
17      ZERO_BALANCE_IND_M3  float64    49071     0.000       2
1              CUSTOMER_AGE  float64    49071     0.000     117
25  PASS_AFTERGRACE_IND_M1  float64    49071     0.000       2
24  PASS_AFTERGRACE_IND_M2  float64    49071     0.000       2
..                     ...      ...      ...       ...     ...
34   INC_PROP_SMS_CALLS_M1  float64    48760     0.634     213
63   OUT_PROP_SMS_CALLS_M3  float64    48733     0.689     781
35   INC_PROP_SMS_CALLS_M2  float64    48728     0.699     201
36   INC_PROP_SMS_CALLS_M3  float64    48720     0.715     221
29       CURR_HANDSET_MODE   object    42459    13.474     370

[77 rows x 5 columns]
```

L'ensemble du dataframe est étudié ce qui nous permet de savoir qu'il y a en tout 13 700 valeurs manquantes et que la variable *CONTRACT_KEY* est une clé primaire. Pour chaque variable nous avons d'une part son type et le nombre d'observations renseignées et d'autre part le pourcentage de valeur manquante et le nombre de valeur unique. Cela nous permet d'avoir une bonne vision globale de la base de données et notamment des valeurs manquantes qu'il faudra traiter plus tard dans le projet.

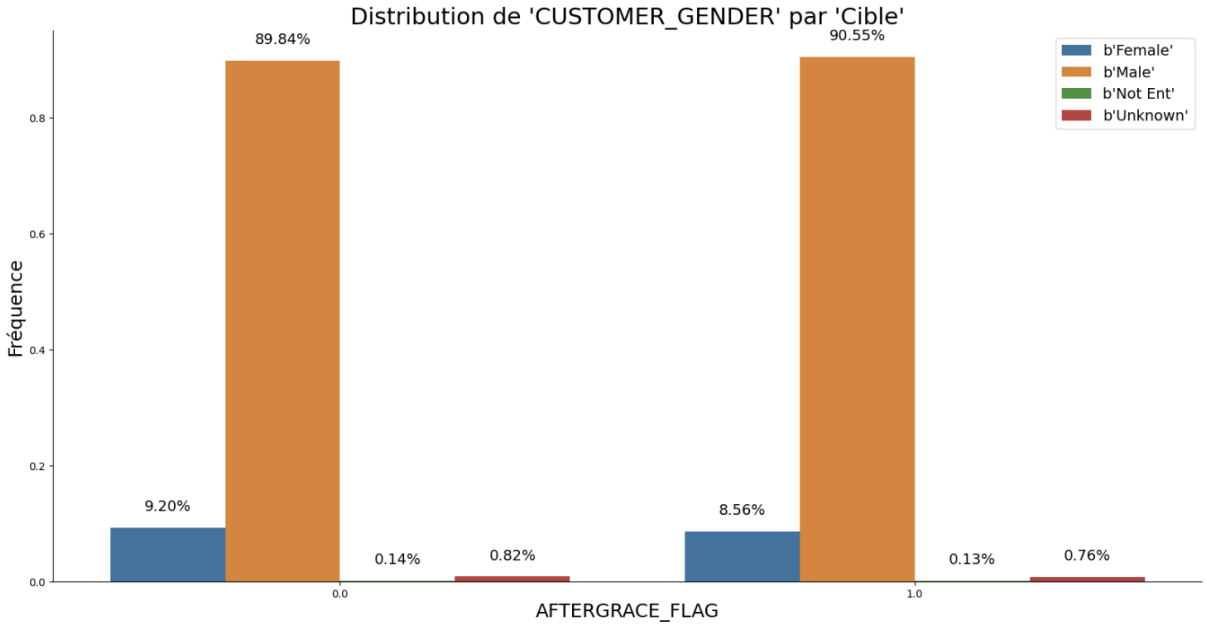
La visualisation de l'impact des variables explicatives sur la variable cible *AFTERGRACE_FLAG* donne des résultats intéressants. L'idée est de regarder la distribution de chaque variable en fonction

des deux modalités de la variable cible. Pour les variables qualitatives, il s'agira de diagramme en bâton tandis que pour les variables quantitatives, il s'agira d'un histogramme et d'une estimation de la densité de probabilité. Prenons un exemple pour chacun des types de variables : *CUSTOMER_GENDER* pour les qualitatives et *CONTRACT_TENURE_DAYS* pour les quantitatives.

CUSTOMER_GENDER possède 4 modalités à savoir :

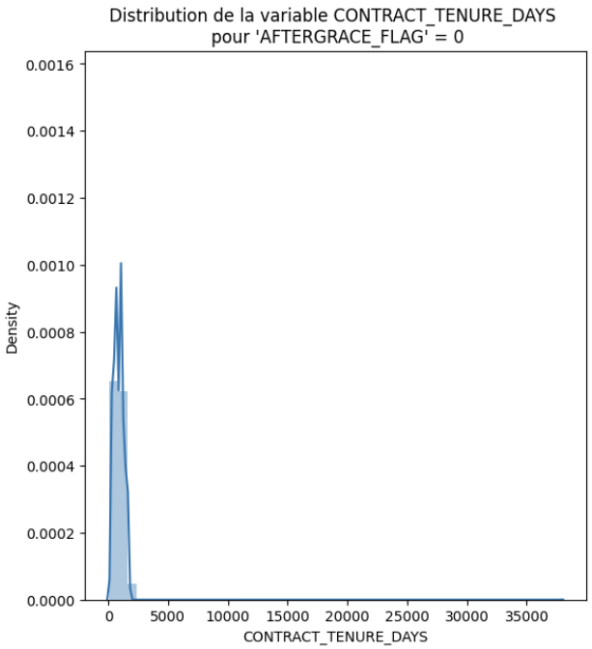
- *b'Female'*
- *b'Male'*
- *b'Not Ent'*
- *b'Unknown'*

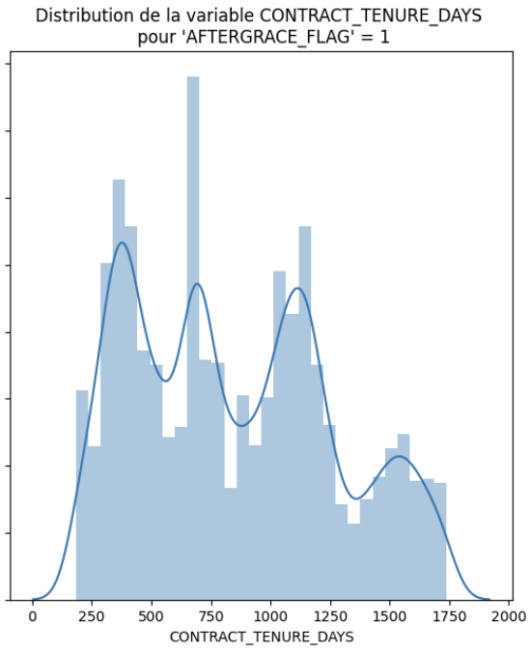
La représentation de leurs distributions en fonction de la variable cible est la suivante :



Que ce soit pour l'une ou l'autre des modalités d'*AFTERGRACE_FLAG,* les distributions sont presque identiques : les hommes représentent la plus grande proportion suivis par les femmes. Les modalités *b'Not Ent'* et *b'Unkown'* sont les plus faiblement présentes avec moins de 1% à chaque fois. En d'autres termes, il y a plus d'hommes dans les clients au total qui sont répartis de la même manière entre les churners et les non-churners. Le constat est le même pour les femmes : elles représentent la deuxième population de clients mais ne dépasse pas les 10% que ce soit dans l'un ou l'autre des deux groupes. On peut en déduire que le sexe est une variable qui n'a visiblement pas de réel impact sur le churn. Ce résultat doit cependant être vérifié avec la matrice de corrélation mais constitue une première intuition qu'il convient de garder en tête pour la suite de l'étude.

La variable *CONTRACT_TENURE_DAYS* dénombre l'ancienneté du contrat en jours. Sa distribution selon la variable cible est la suivante :



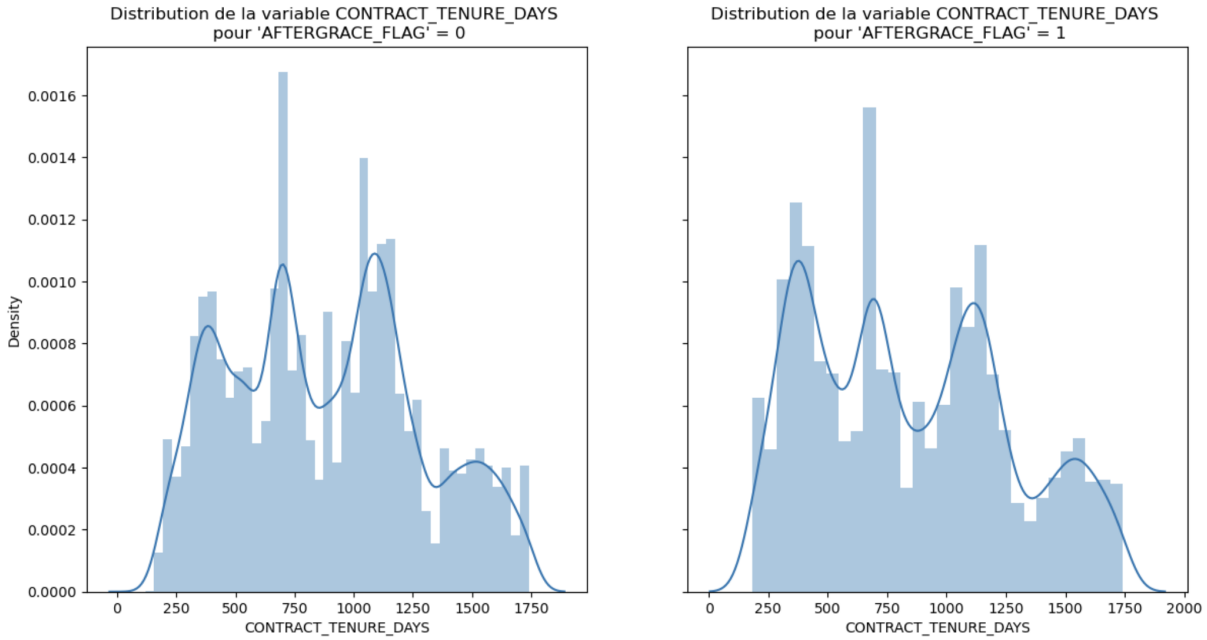


À contrario de *CUSTOMER_GENDER*, nous pouvons voir une réelle différence dans les distributions. Il semblerait que pour la modalité 0 d'*AFTERGRACE_FLAG*, la variable *CONTRACT_TENURE_DAYS* aient une queue de distribution plus étendue que pour la modalité 1. En effet, les valeurs sur l'axe des abscisses ne sont pas à la même échelle, pour la modalité 0 on va jusqu'à plus de 35 000 jours contre 2000 pour la modalité 1. Ainsi, on peut déduire sans problème que le plus vieux contrat de la base de données n'est pas celui d'un churners et qu'il s'agit sans doute d'une valeur extrême. Concernant les autres valeurs, elles semblent se concentrer sur l'intervalle [0;2500] comme pour la modalité 1. Si l'on regarde le nombre de contrat plus ancien de 2 500 jours on en compte 3, tous avec la même valeurs :

	CONTRACT_KEY	CUSTOMER_AGE	CUSTOMER_GENDER	CONTRACT_TENURE_DAYS	AVERAGE_CHARGE_6M	FAILED_RECHARGE_6M	AVERAGE_RECHARGI
109	9693.0	43.0	b'Male'	37862.0	NaN	0.0	
46013	2781414.0	33.0	b'Unknown'	37862.0	NaN	0.0	
46850	2863089.0	53.0	b'Male'	37862.0	NaN	0.0	

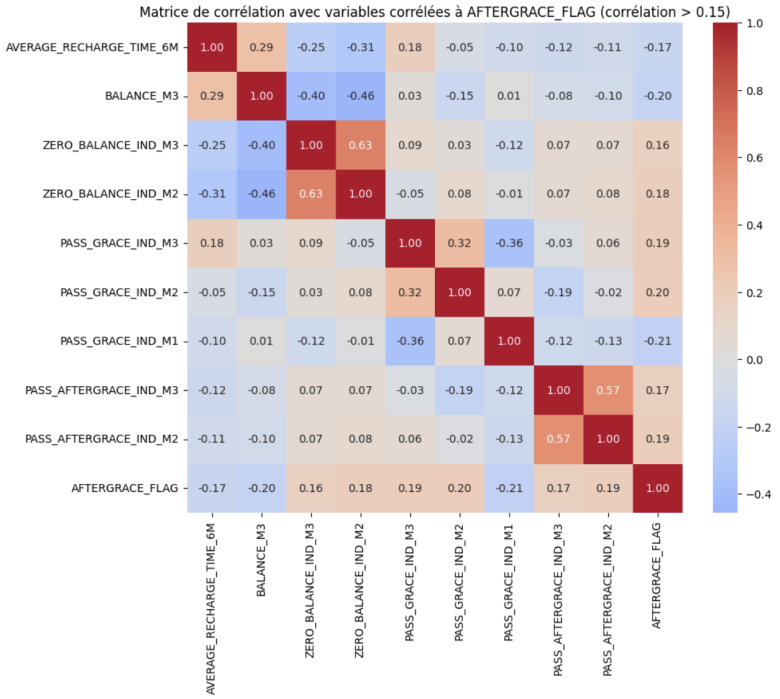
3 rows × 77 columns

En les supprimant et en effectuant de nouveau la distribution de *CONTRACT_TENURE_DAYS* en fonction d'*AFTERGRACE_FLAG,* on obtient le graphique suivant :



Les contrats les plus anciens ont plus grande importance pour la modalité 0 que pour la modalité 1 et inversement les contrats les plus récents semblent avoir plus de poids pour la modalité 1 que pour la modalité 0. Ainsi, les clients les plus à même d'entrer en période d'invalidité sont des nouveaux clients, des clients récents tandis que ceux qui restent sont des clients de longue date, des clients fidèles. Le même procédé est appliqué aux autres variables du dataset.

La matrice de corrélation permet de cibler les variables les plus à même d'influencer la variable cible. En fixant un seuil arbitraire à 0.15, on obtient la matrice suivante :



Les variables les plus corrélées en valeur absolue à *AFTERGRACE_FLAG* sont *PASS_AFTER_GRACE_IND_M2*, *PASS_AFTER_GRACE_IND_M3* et *BALANCE_M3* soient les indicateurs de passage en invalidité en M2 et en M3 et le reste du solde à la fin de M3. On peut noter que les variables retenues pour la matrice avec le seuil concerne pour la plupart M2 et M3 à savoir les mois de juillet et de juin. Les variables présentes dans la matrice devraient donc jouer un rôle important dans l'étape de modélisation. Il faudra comparer les résultats obtenus avec la matrice pour confirmer cette hypothèse.

Maintenant que nous avons une meilleure idée des variables qui influencent l'entrée en période d'invalidité, il faut traiter les valeurs manquantes que nous avons précédemment mis en lumière et, si besoin, procéder à certaines opérations de nettoyage de la donnée.

II. Traitement des valeurs anormales et manquantes avec création d'indicateurs

En procédant à une analyse variable par variable, il apparaît que certaines variables comportent un certain nombre de valeurs négatives alors que cela est impossible. C'est notamment le cas de la variable *CUSTOMER_AGE*. Pour les faire disparaître, nous adoptons la règle suivante : si la valeur absolue d'un âge est inférieur à 18, il est remplacé par *nan*, sinon on conserve sa valeur absolue. Le choix du seuil 18 tient dans le fait que l'on considère qu'une personne ne peut souscrire à un contrat que si elle a plus de 18 ans (loi française). Avec cette même règle de gestion nous retraitons la variable *CONTRACT_TENUE_DAYS* : en convertissant le nombre de jours en années, si le contrat a été contracté avant les 18 ans du client, on remplace *CONTRACT_TENUE_DAYS* par une valeur manquante. En considérant que la variable *NO_OF_RECHARGES_6M* représente le nombre total de recharges (celles effectivement effectuées et celles qui ont échouées), nous instaurons une autre règle. Si le nombre de recharges échouées est supérieur au nombre de recharge totale alors on remplace la valeur des recharges échouées FAILED_RECHARGE_6M par *nan*. Les variables de durée entrante sur durée sortante comportent des valeurs négatives ce qui est impossible. Nous remplaçons donc les valeurs concernées par leurs valeurs absolues. Enfin, la variable *CURR_HANDSET_MODE* est supprimée et remplacée par la variable *marque*, qui, comme son nom l'indique, contient les marques des téléphones plutôt que les modèles.

Les observations anormales étant traitées, nous passons ensuite à l'imputation des valeurs manquantes. Pour ce faire, nous allons commencer par séparer notre jeu de données en échantillon d'entraînement et de validation. La variable cible n'étant pas présente de manière égale dans le base de données (64,33% pour la modalité 0 contre 35,67% pour la modalité 1), il faut respecter cette répartition dans le partage des données. Pour cela, nous allons utiliser l'option *stratify* de la fonction *train_test_split.* Les données d'entraînement vont comporter 80% du dataset initial dont 64,33% de modalité 0 de la variable cible et 35,67% de la modalité. Pareillement les données de validation vont comporter 20% du dataset initial dont 64,33% de modalité 0 de la variable cible et 35,67% de la modalité 1.

Pour l'ensemble d'entraînement, les lignes où plus de 60% de l'information est manquante sont supprimées. Pour les autres observations, pour les variables quantitatives, nous remplaçons les valeurs manquantes à l'aide de *groupby*. L'idée est de regarder les variables les plus corrélées à la cible *AFTERGRACE_FLAG* et de choisir parmi celles-ci trois variables qui ont peu de modalités afin d'effectuer trois regroupements : un avec les trois variables les plus corrélées, un avec les deux variables les plus corrélées et un avec la variable la plus corrélée. Dans notre cas, ces variables sont respectivement *PASS_GRACE_IND_M1, PASS_GRACE_IND_M2* et *PASS_AFTERGRACE_IND_M2.* Nous commençons par imputer par la médiane du regroupement des trois variables. S'il reste encore des valeurs manquantes, nous imputons par la médiane du regroupement des deux variables. Le remplacement par le regroupement selon *PASS_GRACE_IND_M1* intervient en dernier, s'il reste encore des *nan* après les deux traitements précédents. Enfin, si des valeurs manquantes persistent même après les opérations précédentes, elles sont remplacées par la médiane de la colonne dans laquelle elles sont contenues. Pour l'ensemble de test, on procède de la même façon, si ce n'est que l'on utilise l'ensemble d'entraînement pour remplacer les valeurs manquantes et non pas les valeurs de

l’ensemble test. Ce choix s’explique par le fait que si l’on imputait les valeurs manquantes à partir des valeurs de cet ensemble, cela créerait du biais lors de la modélisation. Pour les variables qualitatives, peu importe l’ensemble considéré, les valeurs manquantes ne sont présentes que pour la variable marque. Aussi, elles sont remplacées par la valeur *unknown*.

La dernière étape du traitement de notre jeu de données est la création d'indicateurs. Nous les créons de la même façon pour les ensembles d'entraînement et de test. La date d’arrêté étant fixée au 31 août, il est possible de remonter le mois durant lequel la dernière recharge a été effectuée : soit à M1 (entre 0 et 31 jours), soit à M2 (entre 32 jours et 62 jours), soit à M3 (entre 63 et 92 jours) ou soit plus loin dans le temps (plus de 93 jours). De plus, il est également possible de savoir si plusieurs recharges ont été effectuées sur la période août-juin en considérant que les variables *BALANCE_M1*, *BALANCE_M2* et *BALANCE_M3* sont le reste des recharges effectuées à la fin du mois (équivalent du crédit téléphonique restant). Nous pouvons également créer des flag entrant et sortant qui enregistrent respectivement toutes les entrées sur le téléphone (appel et sms) et toutes les sorties du téléphone (appels, appels vers la messagerie, appels vers l’international, appels vers le service client et sms). Enfin, on peut catégoriser les contrats en nouveaux et en anciens : un nouveau contrat est un contrat de moins de deux ans, un ancien contrat est un contrat de plus de deux ans.

La base de données étant nettoyée et les indicateurs étant créés, nous pouvons passer à l’étape de modélisation. Avant cela, il convient de procéder à des tests statistiques de sélection de variables.

III. Sélection des variables et modélisation

Après la création d'indicateurs, les deux ensemble de données sont composés de 88 colonnes mais toutes ne sont pas significativement liées à la variable cible. Pour s'assurer de cela, nous effectuons deux tests statistiques : le test de Student pour les variables quantitatives et le test du chi-2 pour les variables qualitatives. Avec un seuil de 5%, les variables à retirer sont au nombre de 8 :

- PASS_AFTERGRACE_IND_M1
- INC_DURATION_MINS_M1
- INC_PROP_SMS_CALLS_M1
- INC_PROP_SMS_CALLS_M2
- INC_PROP_OPE1_MIN_M2
- INC_PROP_OPE2_MIN_M3
- OUT_DURATION_MINS_M1
- OUT_VMACC_NO_CALLS_M1

Avant de passer à la modélisation, une dernière étape de transformation des données est nécessaire. Les variables catégorielles doivent devenir exploitables pour le modèle. Pour cela, nous allons les encoder selon deux méthodes : le one hot encoding pour la variable *CUSTOMER_GENDER* et le target encoding pour la variable *marque*. Les variables quantitatives quant à elles sont normalisées.

Nous avons essayé trois modèles à savoir la régression logistique, XGBoost et LightGBM. Les résultats les plus probants sont ceux données par XGBoost avec pour métrique le f1 score micro. Nous avons choisi cette métrique car c'est la plus pertinente en cas de données déséquilibrées. Or c'est le cas de notre base de données. Les paramètres retenus sont les suivants :

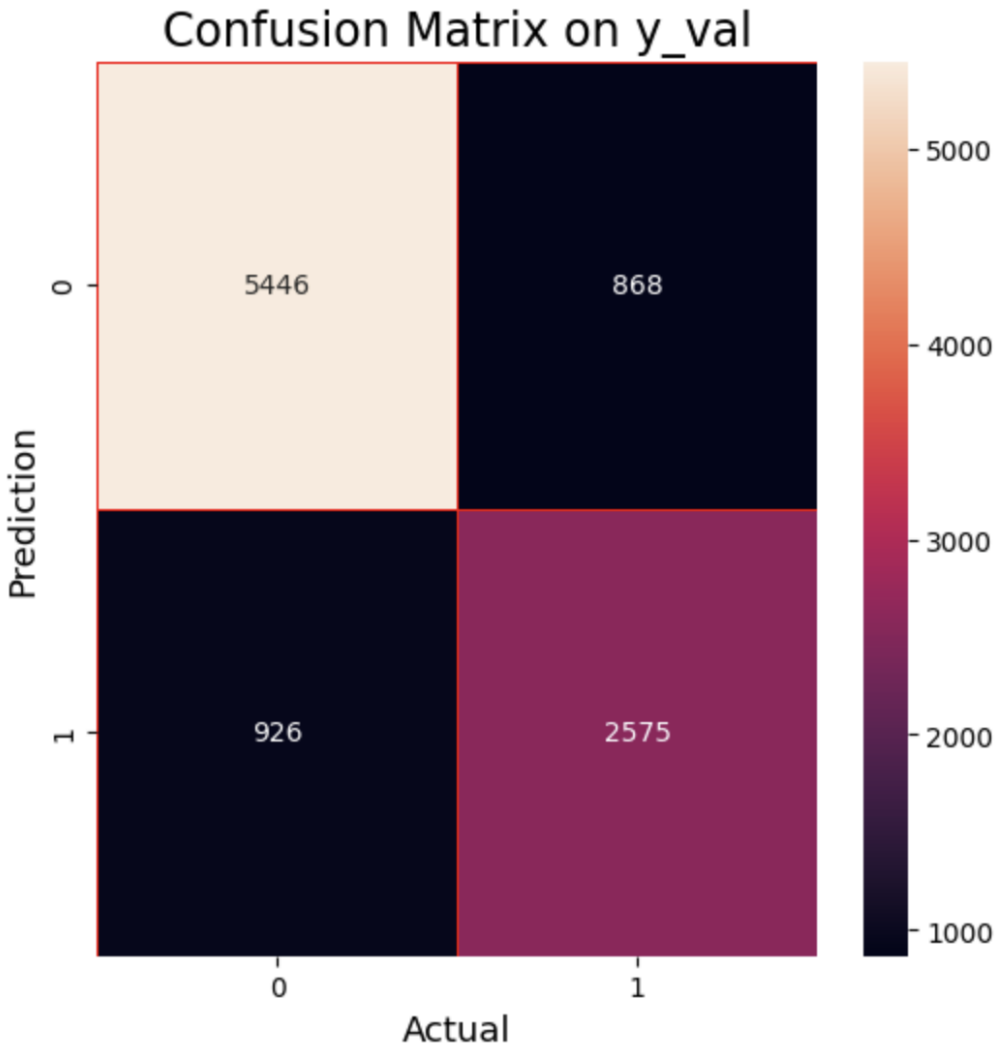
- n_estimatos = 1156
- max_depth = 10
- min_child_weight = 2
- learning_rate = 0.010535907159160793
- min_split_loss = 4.958509491523419
- colsample_bytree = 0.7224695121712467
- subsample = 0.8652722215243238

IV. Résultat et conclusion de l'étude

Les résultats de la modélisation sont récapitulés dans le tableau ci-dessous :

Métrique	Données d'entraînements (sans cross-validation)	Données d'entraînements (cross-validation)	Données test
Accuracy	0.8897	0.8049	0.8172
F1 score	0.8897	0.8049	0.8172
ROC_AUC	0.9575	0.8949	0.9009

La matrice de confusion sur les données de test est la suivante :

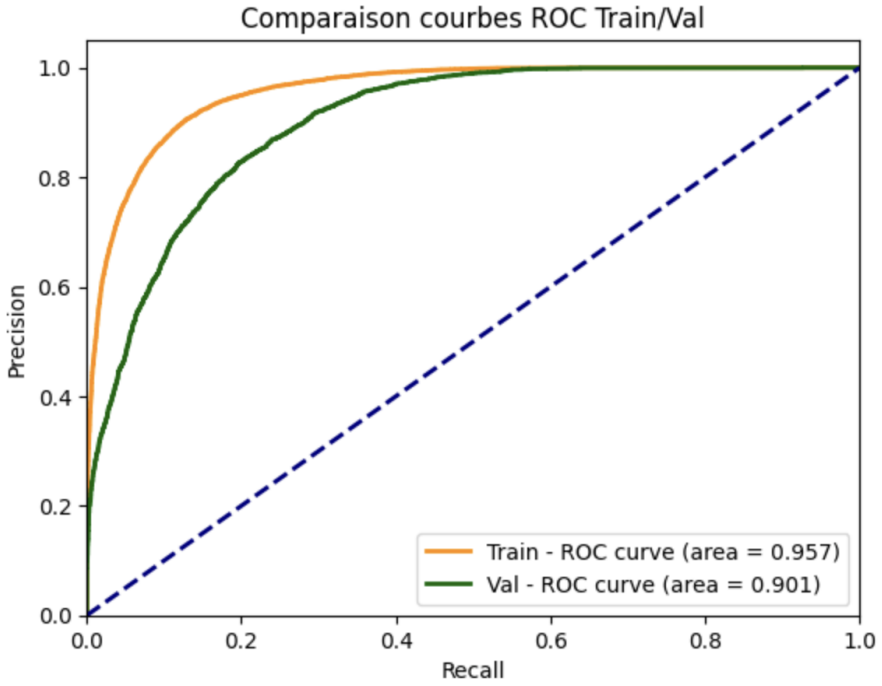


Les churners correctement détectés dans l'ensemble d'entraînement sont au nombre de 2 575. Les non-churners correctement détectés sont au nombre de 5 446. Les churners prédit par le modèle comme étant des non churners sont 868 tandis que les non churners prédit par le modèle comme étant des churners s'élèvent à 926. Les prédictions correctes sont plus importantes que les prédictions incorrectes, ce qui est rassurant. Pour s'assurer de la qualité des résultats, il faut regarder de plus près les scores associés à la prédiction du modèle :

	precision	recall	f1-score	support
0.0	0.85	0.86	0.86	6314
1.0	0.75	0.74	0.74	3501
accuracy			0.82	9815
macro avg	0.80	0.80	0.80	9815
weighted avg	0.82	0.82	0.82	9815

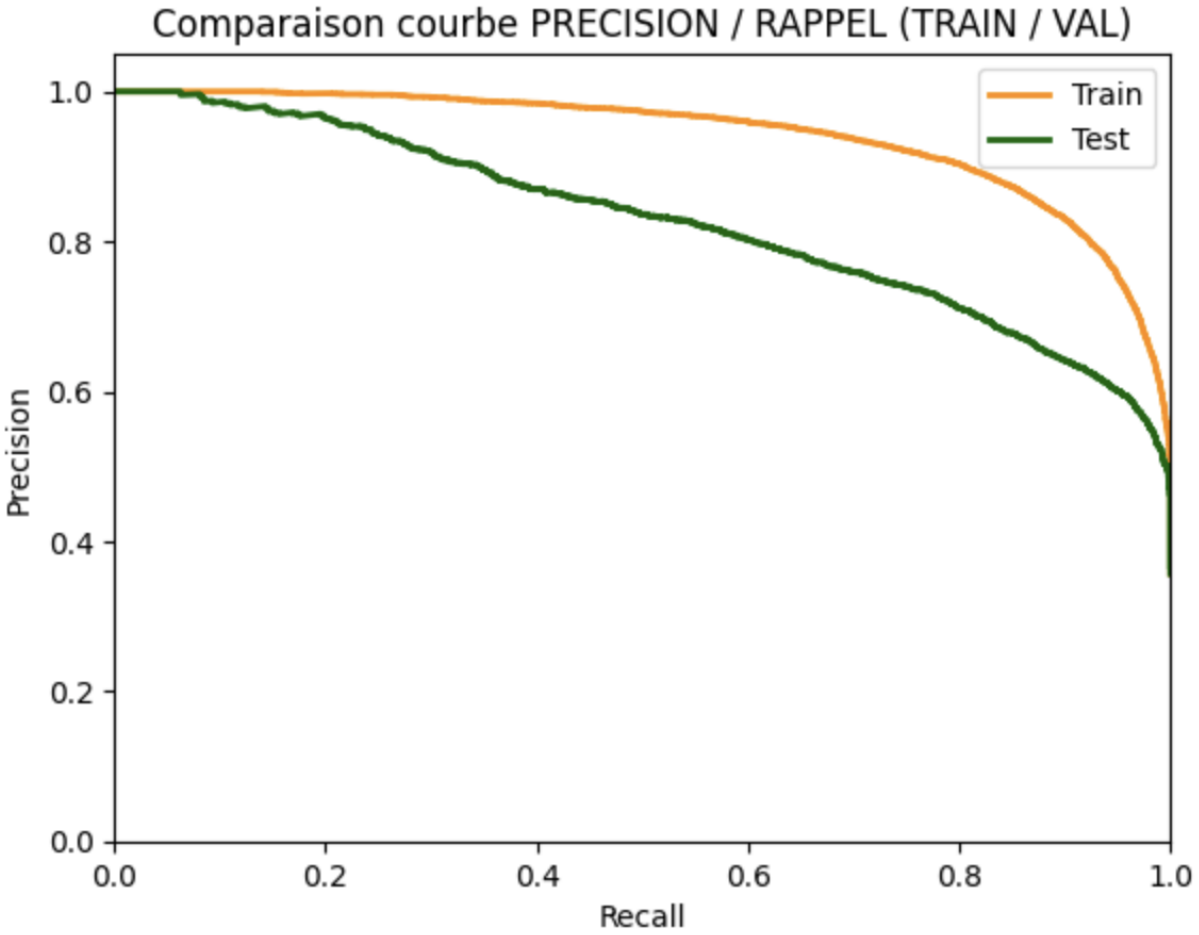
Le modèle prédit le mieux les clients non-churners avec une précision, un recall et un f1-score de respectivement 0.85, 0.86 et 0.86. Concernant les churners, les résultats sont plus bas avec une précision, un recall et un f1-score de respectivement 0.75, 0.73 et 0.74. Ainsi, le modèle peine plus à les détecter correctement.

Pour avoir une meilleure idée de la performance de notre modèle, regardons les courbes ROC associées :



Plus la courbe ROC est proche du coin supérieur gauche et meilleure est la performance du modèle. La courbe de l'ensemble de test en vert est relativement proche du coin supérieur gauche avec une aire sous la courbe de 0.901 ce qui atteste d'une bonne performance de notre modèle : il est capable d'identifier correctement les churners. Les courbes des données d'entraînement et de test sont peu éloignées mais assez pour suspecter un léger surapprentissage.

L'objectif de notre étude est d'optimiser à la fois le recall pour détecter le plus de churners et la précision pour les cibler de façon spécifique. En effet, si l'on adopte une logique marketing, une compagnie de télécom va vouloir identifier les churners (recall) pour leur adresser des offres qui les feront rester (précision). Si les individus ne sont pas cibler de façon précise, la compagnie s'expose à des coûts non négligeables : une offre trop avantageuse proposée à des clients qui n'allait pas partir représente une perte pour l'entreprise qu'elle aurait pu éviter en ciblant les individus dont elle est sûr qu'ils vont être des churners. L'optimisation des deux paramètres passe par le choix d'un seuil. L'arbitrage peut se faire à travers le graphique suivant :



Plutôt que d'essayer de trouver une valeur "un peu près" sur le graphique, nous avons opté pour une approche plus business. L'idée est de mettre en application notre modèle et de montrer qu'il permet de limiter la perte de bénéfice. En effet, pour essayer de garder les churners dans la compagnie, cette dernière va par exemple mettre en œuvre une offre promotionnelle. En sachant que la compagnie fait un profit mensuel sur chaque client qui a souscrit chez elle, la perte mensuelle de l'entreprise si elle ne conserve pas ses churners s'évalue comme suivant :

$$Total\ perte\ mensuelle\ sans\ modèle = montant\ recharge\ moyen \times total\ des\ churners \times profit\ en\ \%\ sur\ les\ recha$$

Notre modèle permettant de détecter les churners, la compagnie peut leur proposer une offre qui est susceptible de les faire rester. Aussi l'équation de la perte mensuelle devient :

$$Total\ perte\ mensuelle\ modèle = Total\ perte\ mensuelle\ sans\ modèle - (vrais\ positifs\ détectés\ par\ le\ modèle \times profit\ en\ \%\ sur\ les\ recharges \times montant\ recharge\ moyen - total\ churners\ prédit\ par\ modèle \times coût\ offre\ pour\ la\ compagnie)$$

Le gain sur la perte généré par les churners qui quittent la compagnie que permet le modèle s'évalue comme suivant annuellement :

$$Gain\ sur\ perte\ annuel = (Total\ perte\ mensuelle\ sans\ modèle - Total\ perte\ mensuelle\ modèle) \times 12$$

L'idée est d'obtenir le profit maximal. Les churners sont déterminés par le modèle selon le seuil choisi. Ainsi, le profit optimal est associé à un seuil qui permet de définir une précision et un rappel. En appliquant le raisonnement précédent sur nos données, nous fixons le profit à 40%, le montant moyen de recharge est obtenu à partir de la variable *AVERAGE_CHARGE_6M (*moyenne de la variable puis division par 6) et la remise faite par l'entreprise aux churners est de 20% du montant moyen de recharge. Appliquer 20% de remise au cas par cas ou sur la moyenne de la variable revient au même. Nous obtenons les résultats suivant respectivement pour les données d'entraînements et de test :

```
100%|          | 39190/39190 [28:53<00:00, 22.61it/s]
{'seuil': 0.48784318566322327,
 'recall': '0.8516534533247625',
 'precision': '0.8423283413393614',
 'tot_perte_sans_model': 403049.0186401651,
 'tot_perte_avec_model': 263546.4442290344,
 'profit_net_sauve_grace_au_model_sur_1an': 139502.57441113068}
```

```
100%|          | 9813/9813 [02:03<00:00, 79.53it/s]
{'seuil': 0.5277165770530701,
 'recall': '0.6983718937446444',
 'precision': '0.7688679245283019',
 'tot_perte_sans_model': 99900.84059908304,
 'tot_perte_avec_model': 75503.46307488537,
 'profit_net_sauve_grace_au_model_sur_1an': 24397.377524197662}
```

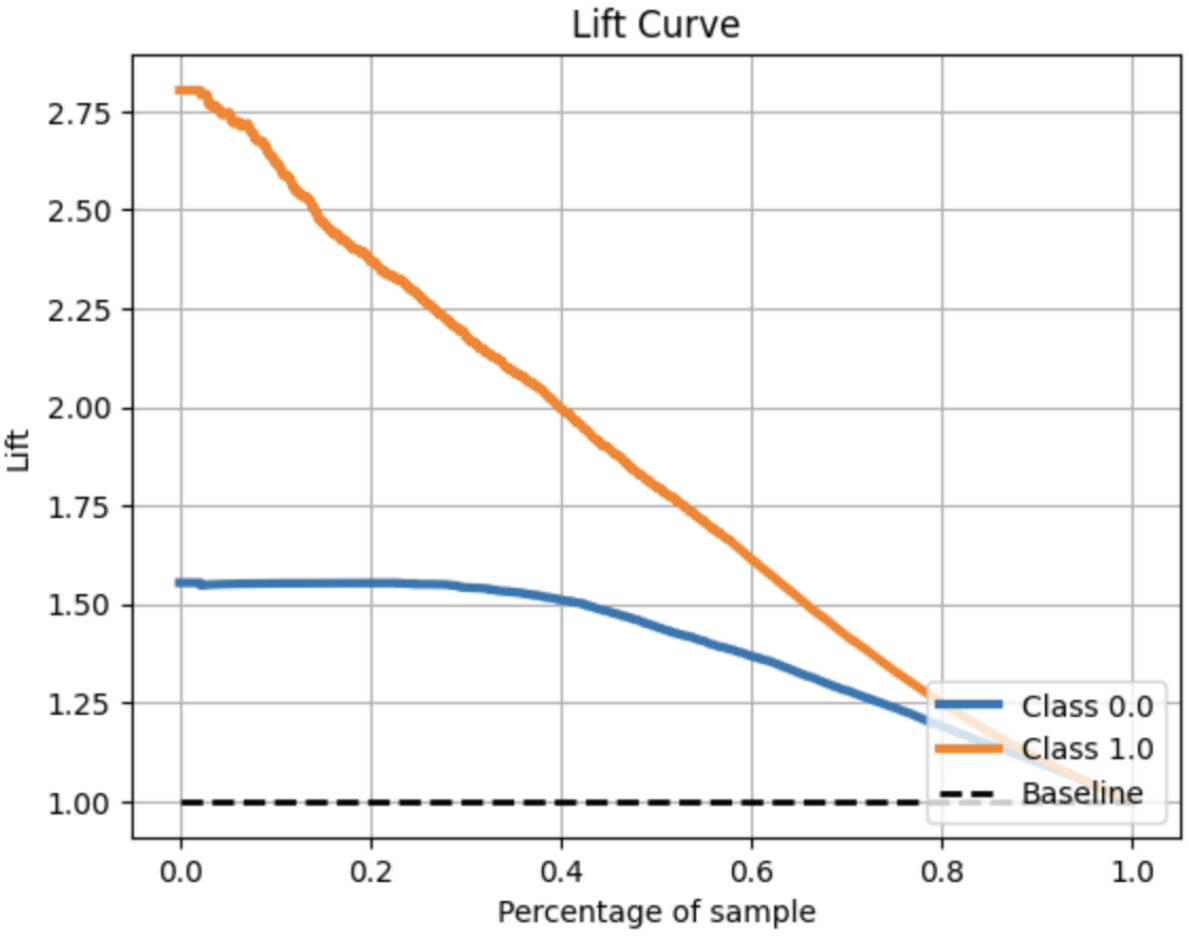
Dans l'ensemble d'entraînement, avec le seuil optimal retourné par le programme, le rappel et la précision sont très proche (moins de 0.1% d'écart) ce qui signifie qu'il est préférable pour la compagnie d'à la fois essayer de toper le plus de churners possible et de limiter les pertes (proposer à des non churners la promotion va entraîner des pertes non nécessaires, donc il faut bien les cibler et proposer à l'offre à des individus dont on est sûr qu'ils sont des churners). À contrario pour l'ensemble de test, la précision est supérieure au rappel de presque 7% ce qui signifie que la priorité

est donnée au fait de proposer la remise qu'aux clients dont on est sûr qu'ils sont des churners. En appliquant les seuils précédent on obtient les matrice de confusion suivante :

```
Metrique pour le jeu de données train :

 Recall : 0.8516534533247625

 Précision : 0.8423283413393614
       0      1
0  23010   2232
1   2077  11924
```

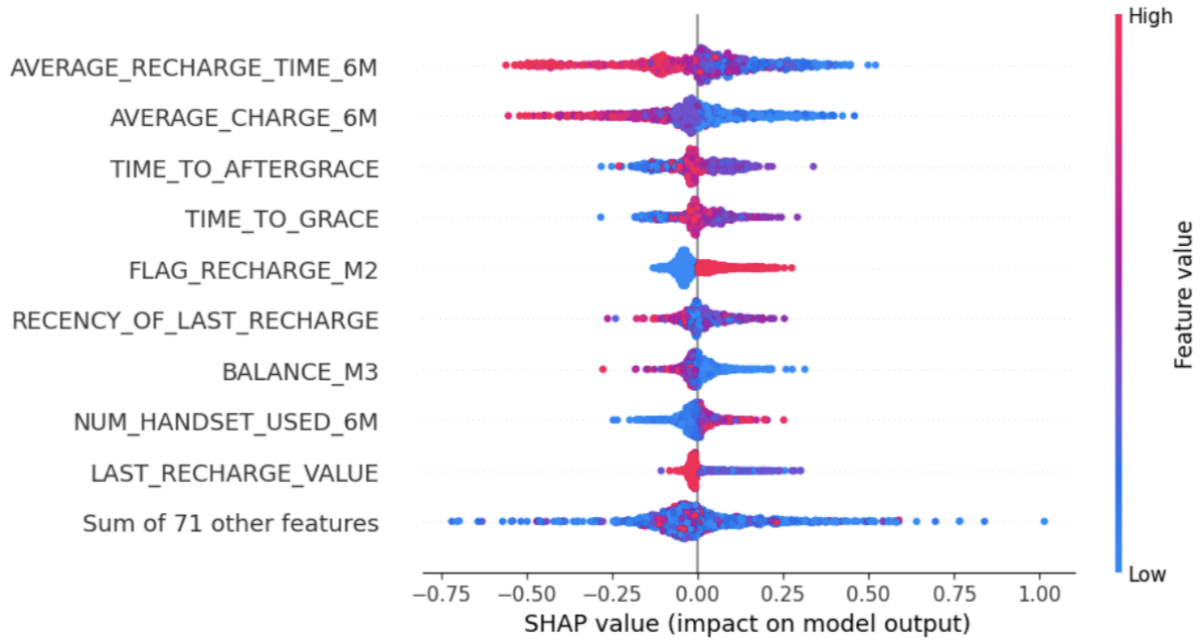
```
Metrique pour le jeu de données val avec le meilleur seuil :

 Recall : 0.6983718937446444

 Précision : 0.7688679245283019
      0     1
0  5579   735
1  1056  2445
```

Une autre mesure de la performance de notre modèle est la courbe Lift. En la traçant on obtient :



Si par exemple on prend les 20% de l'échantillon de test qui ont la plus grande probabilité d'être des churners d'après le modèle, on va arriver à capter environ 2,37 fois plus de vrais churners que si l'on sélectionné 20% de l'échantillon de manière aléatoire ce qui n'est pas négligeable.

La dernière étape de notre projet consiste en l'interprétation des résultats. Pour cela, nous utilisons le package SHAP. Les sorties sont les suivantes :

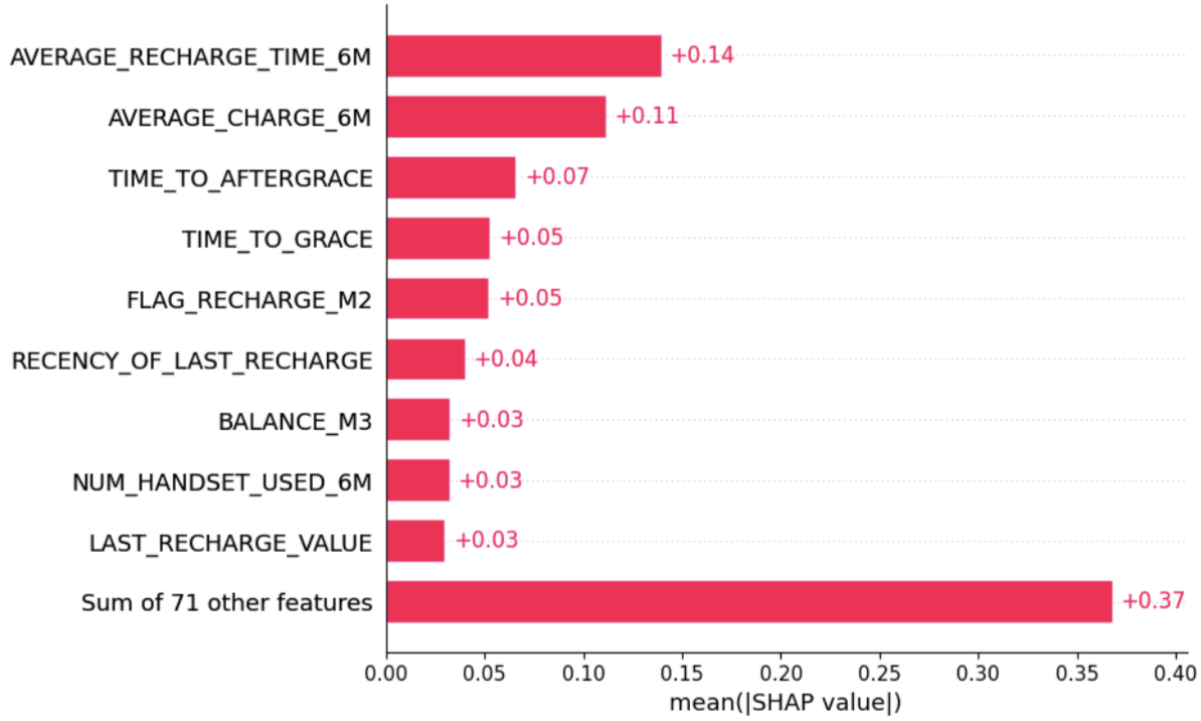


Dans le graphique ci-dessus, nous utilisons la méthode summary_plot pour visualiser l'impact de la directionnalité des entités. L'axe des x représente la valeur SHAP, tandis que l'axe des y répertorie toutes les variables. Chaque point du graphique correspond à une valeur SHAP associée à une prédiction et une caractéristique spécifique.

La couleur rouge indique une valeur plus élevée de la caractéristique, tandis que le bleu indique une valeur inférieure. En examinant la distribution des points rouges et bleus, nous pouvons obtenir une idée générale de l'impact de la directionnalité des fonctionnalités.

À partir de ce graphique, nous pouvons tirer les conclusions suivantes :

- Une valeur élevée de la variable AVERAGE_RECHARGE_TIME_6M (le délai inter-recharge) réduit les chances de churner. Une valeur plus faible de AVERAGE_RECHARGE_TIME_6M conduit à une plus grande chance de churner.
- Une valeur plus élevée de AVERAGE_CHARGE_6M réduit également les chances de churner. Une valeur inférieure de "AVERAGE_CHARGE_6M" conduit à une plus grande chance de churner.
- Une valeur plus élevée de TIME_TO_AFTERGRACE (délai avant période d'invalidité) augmente les chances de churner. À l'inverse, une faible valeur de TIME_TO_AFTERGRACE diminue les chances de churner.



Pour visualiser l'importance des caractéristiques, nous utilisons la méthode summary_plot avec le plot_type "bar". Les caractéristiques sont classées en fonction de leur influence sur les prédictions du modèle. L'axe des x représente la moyenne de la valeur SHAP absolue de chaque caractéristique. Dans notre cas, la variable *AVERAGE_RECHARGE_TIME_6M* est la caractéristique la plus importante, suivie par *AVERAGE_CHARGE_6M*, *TIME_TO_AFTERGRACE* et *TIME_TO_GRACE*.

En conclusion, le modèle XGBoost s'est avéré être le meilleur pour la détection des churners. Parmi les variables fortement corrélées à la variable cible, on observe notamment que la variable AVERAGE_RECHARGE_TIME_6M joue un rôle significatif dans le modèle. Après avoir optimisé le modèle, nous avons obtenu des performances très satisfaisantes sur les données de test, avec un f1-score de 81% et une aire sous la courbe ROC de 90%. Ces résultats démontrent l'efficacité du modèle et justifient sa mise en production.
Enfin, il est essentiel de souligner que le choix du seuil de décision, et donc de la stratégie adoptée par l'entreprise (favoriser le rappel au détriment de la précision, privilégier la précision au détriment du rappel, ou trouver un équilibre entre les deux), dépend des gains réalisés en évitant les pertes. Parfois, il peut être plus avantageux de laisser partir certains churners, même si l'on pourrait initialement penser qu'il est préférable de les retenir à tout prix. Cela nécessite une évaluation approfondie des coûts et des bénéfices potentiels pour prendre la décision la plus adaptée à l'entreprise.