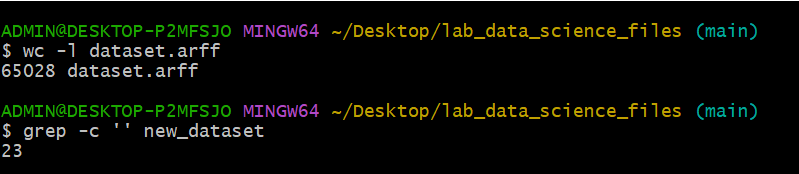
PREPROCESSING

1. grep '^%' dataset.arff > new\_dataset

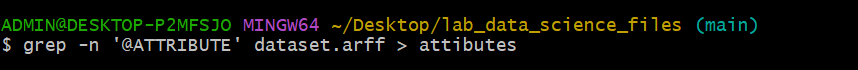


2-

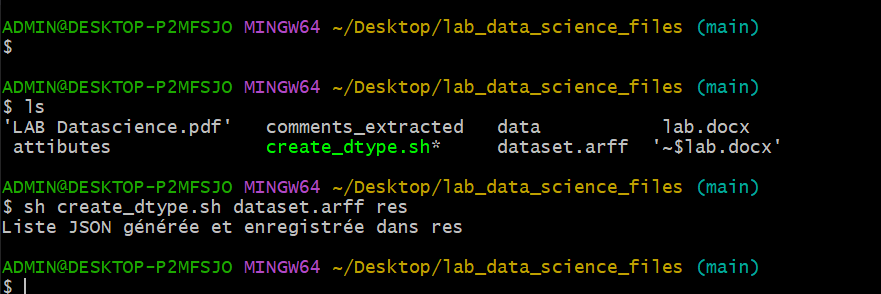


2eme methode :

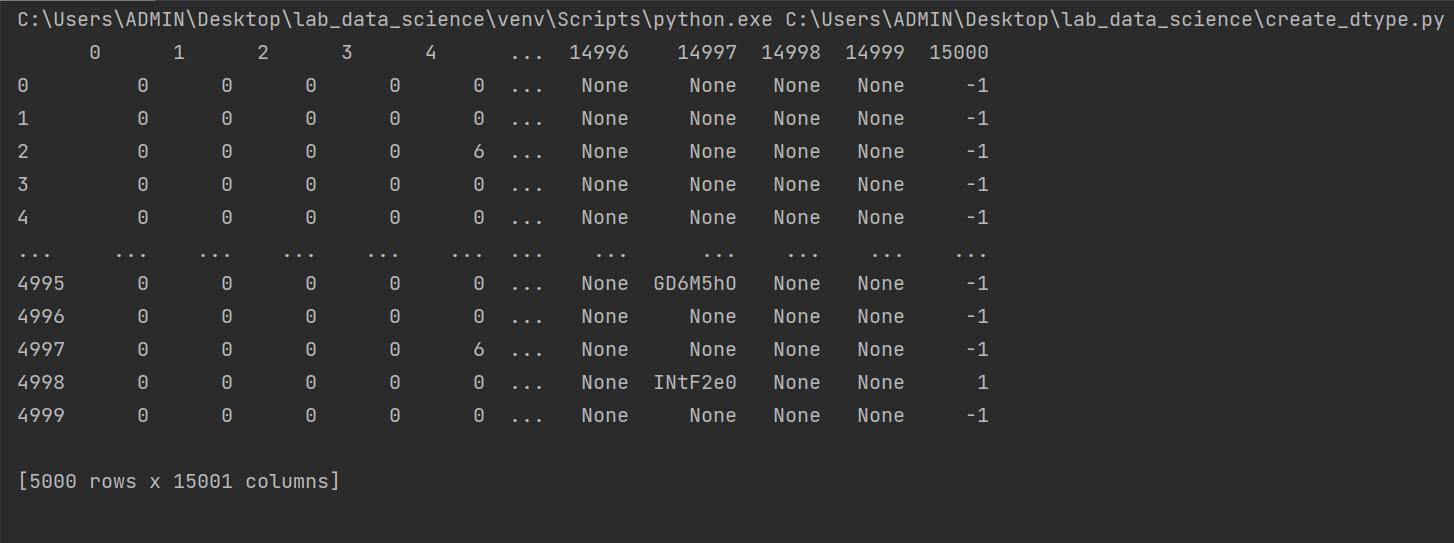
3 –



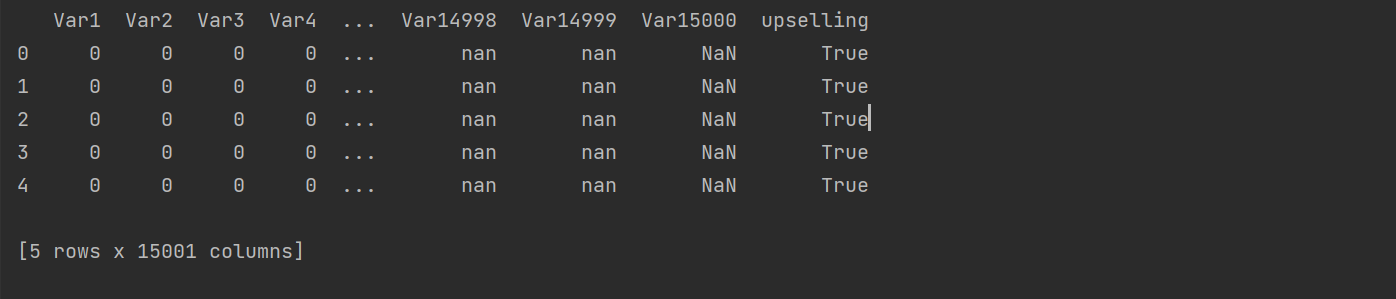
4- Le resultat est stocke dans le fichier res



6-

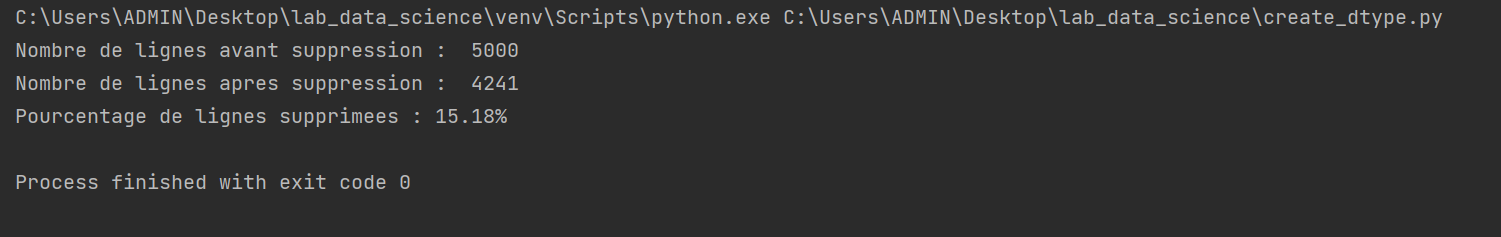


7-



8-

11 c –

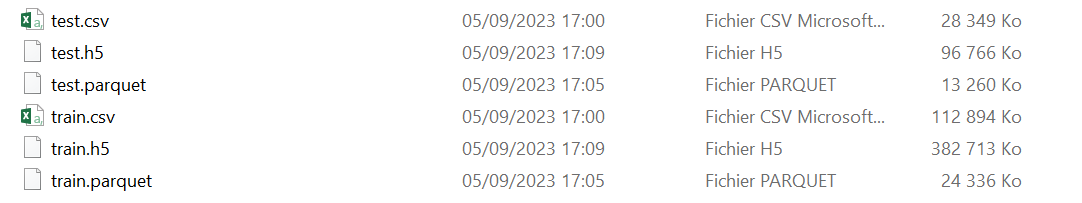


Etant donnee qu’on ne perd pas enormement de lignes, on peut utiliser l’option de supprimer les valeurs manquantes qui nous permettra d’avoir des donnees de meilleurs qualité.

12-

Remplacer les missings value par le min, le max, la moyenne

13-



Parquet ensuite csv puis h5

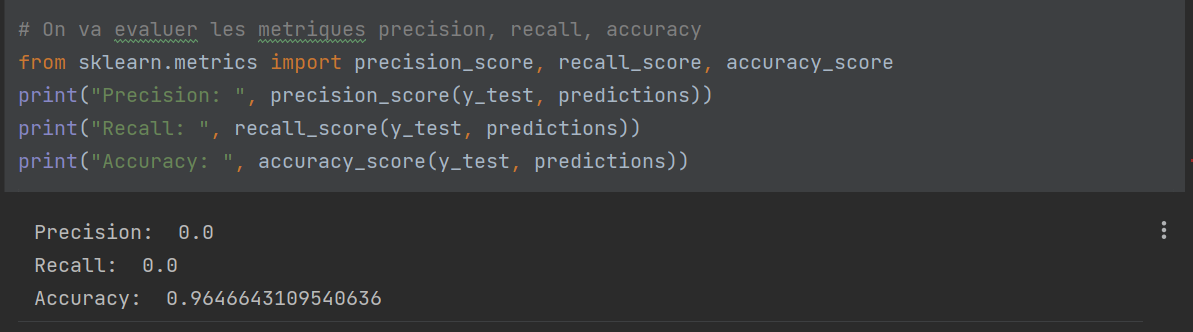
FEATURE ENGINEERING 1 – XGBOOST

1-

Encodage one-hot automatique lorsqu’on specifie le parametre enable\_categorical = true ; pas besoin de gerer manuellement les valeurs manquantes grace a l’encodage one-hot qui créer une categorie a part pour les valeurs manquantes

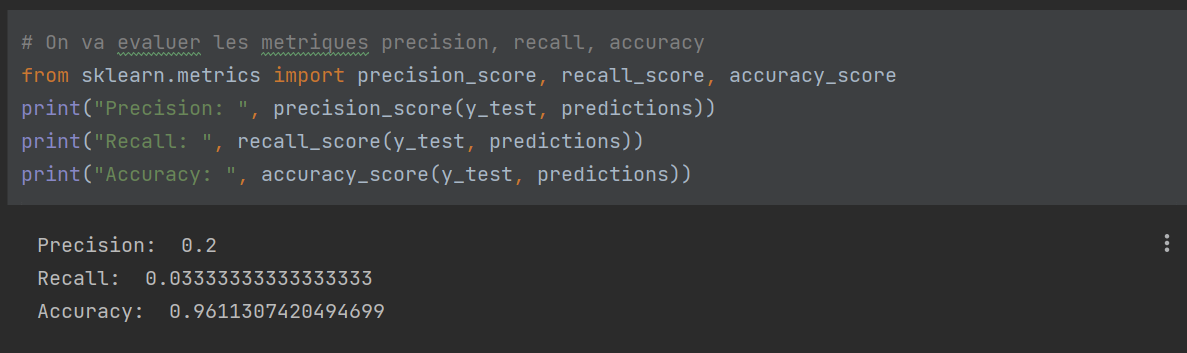
2 –

* avec treshold par default = 0.5 pour la classification binaire qui est notre cas

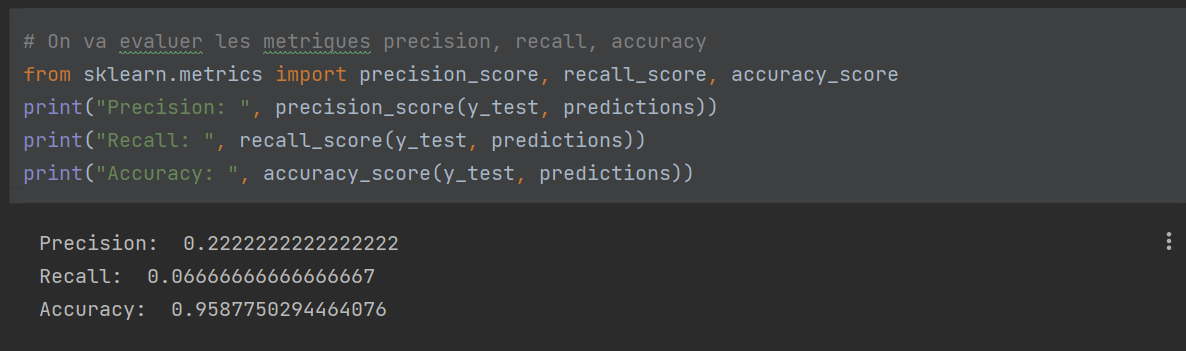


* Avec un treshold different

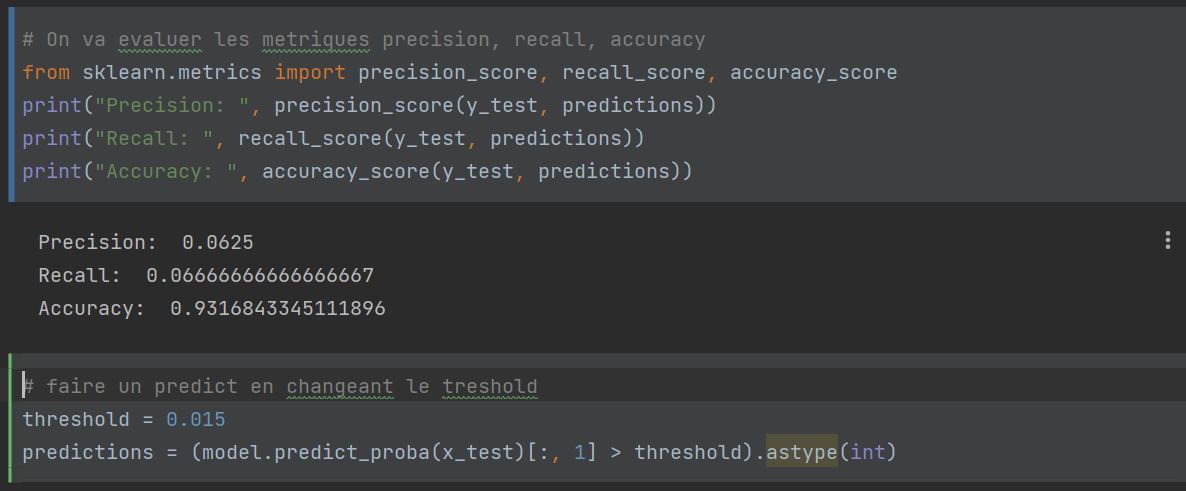
Threshold = 0.1



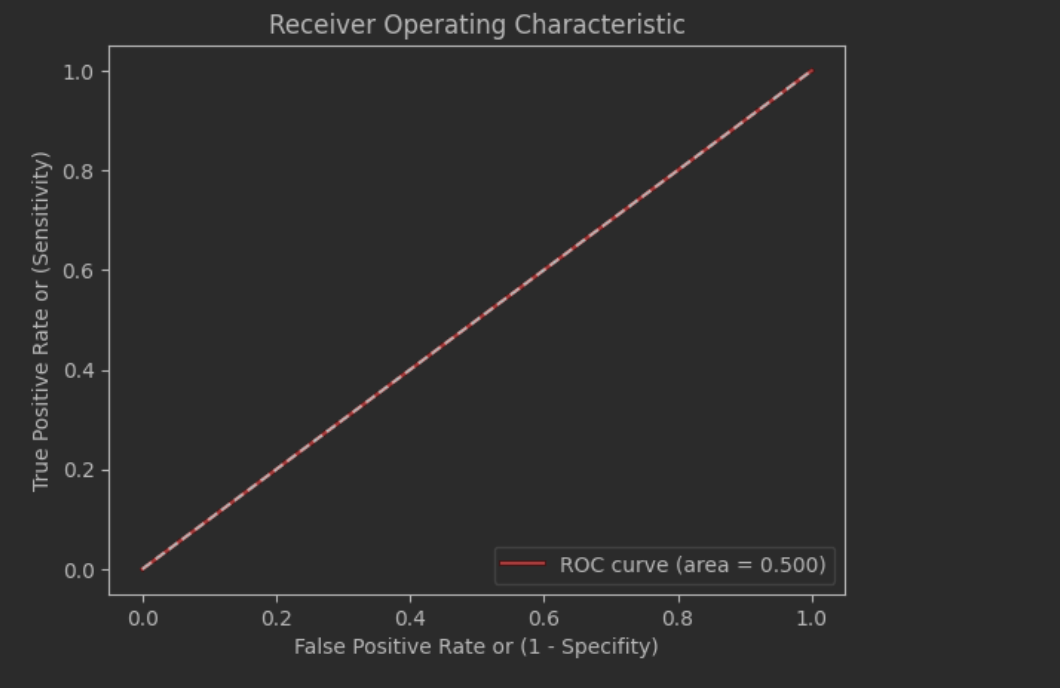
Threshold = 0.05



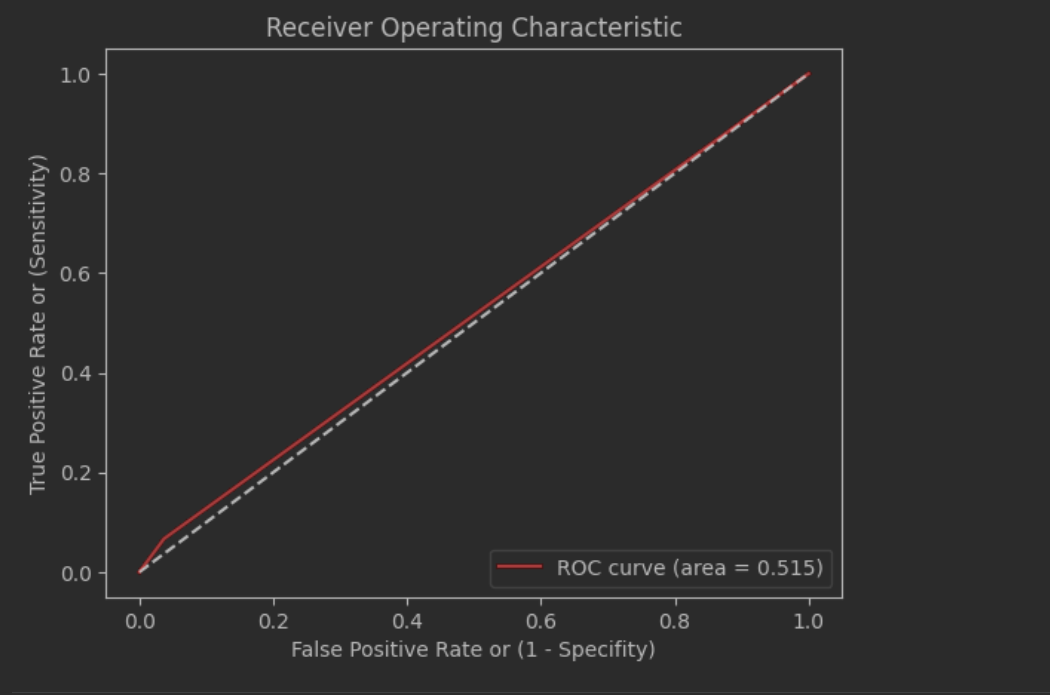
Threshold = 0.015



3-

Roc curve avec le sparametres par defaut :  


Roc curve avec le threshold à 0.015



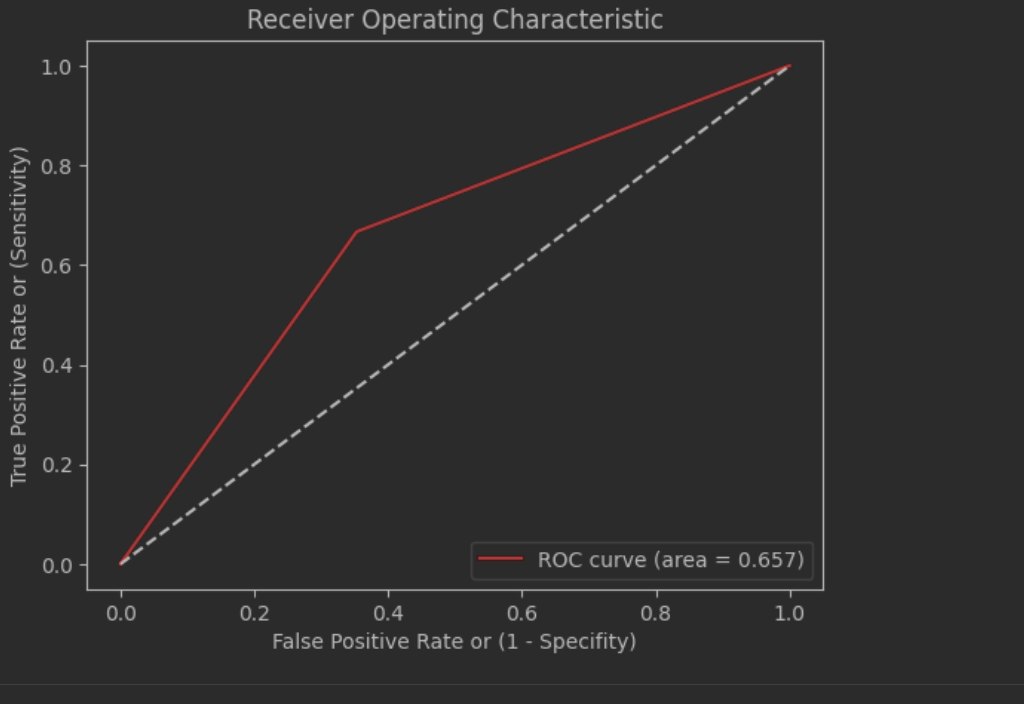
4- En modifiant les parametres du model

Avec le threshold par defaut, on a toujours le meme resultat 0.5, ce qui signifie qu’on a une prediction quasi aleatoire

On va donc travailler avec le treshold de prediction de 0.7



Resultats obtenus avec une approche de style grid\_search



5 – Pas d’amelioration de l’auc

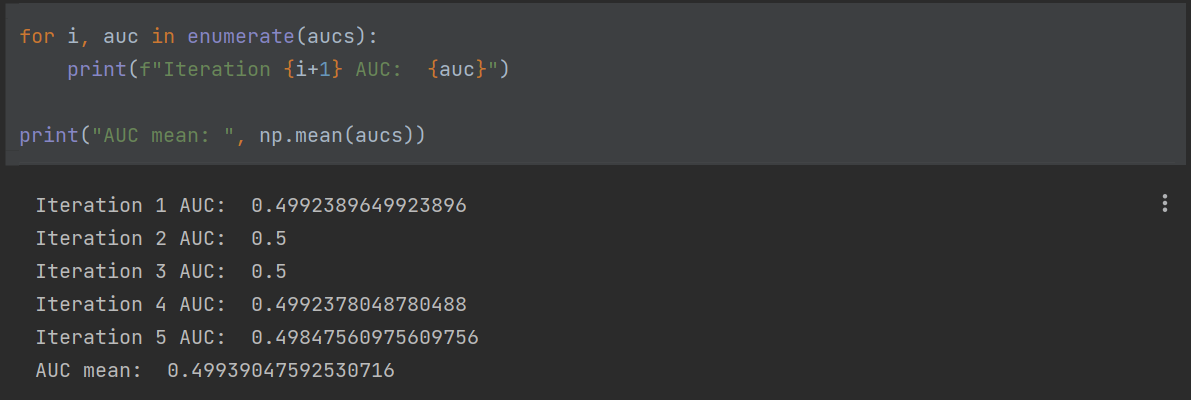
Avec un max\_depth trop grand : risque d’overfitting (du a l’augmantation de la vc-dim), plus de temps d’entrainement et plus de memoire a utiliser

6- Sans utilisation du stratified K-fold cross-validation, on aurait un risque d’overfitting et d’une mauvaise representation des donnees. Cela concerne plus les datasets dont les donnees sont imbalanced comme c’est le cas de nos donnees actuelles

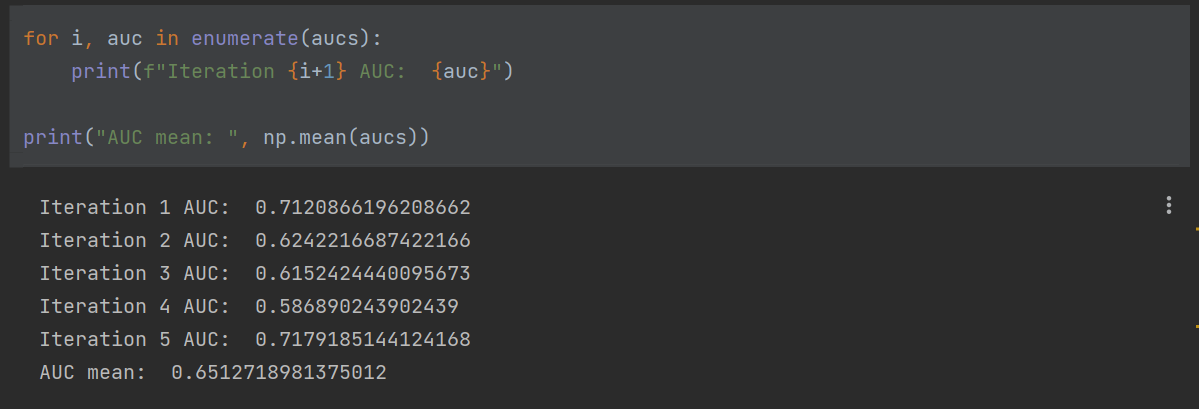
Utiliser cette methode permettrait d’avoir dans chaque folds utilises un equilibre des classes

On doit utiliser uniquement le train test, utiliser l’ensemble de la donnee rendrait la partie du test non fiable. On doit utiliser des donnees de test qui n’auront pas été vu par le model

* En utilisant la fonction predict normale



* Avec un threshold



7-

Contenu du model sous format pkl



FEATURE ENGINEERING 2 – XGBOOST

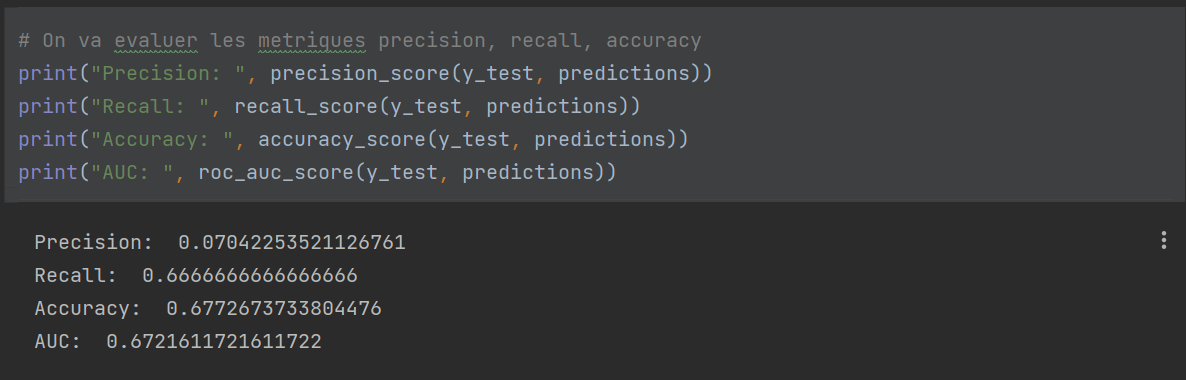
1 –

**max\_cat\_threshold : si le nombre total de categories distinct de toutes features confondues depasse cette valeur, xgboost peut decider de ne pas appliquer de transformation one-hot sur les categories**

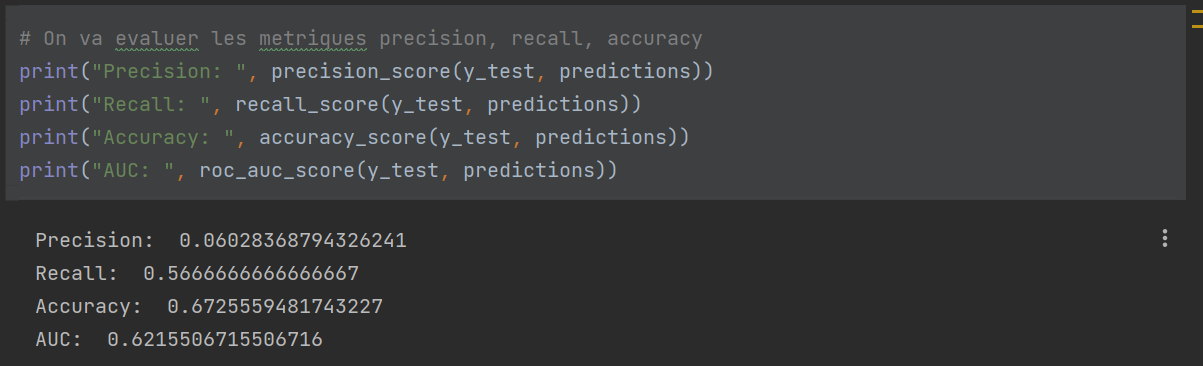
**max\_cat\_to\_onehot** : si le nombre total de categories pour une feature depasse cette valeur, alors aucune transformation n’est appliquee

L’utilisation du one-hot dependra des donnees qu’on a. etant donne que xgboost peut gerer les variables cateoricielles grace par exemple au CatBoost Encoding, cela peut ne pas etre utile surtout lorsque cela augmente enormenment la dimensionalite des donnees

* Avec max\_cat\_threshold = 20 et max\_cat\_to\_onehot = 4



* Avec max\_cat\_threshold = 50 et max\_cat\_to\_onehot = 7



2-

Oui c’est tout à fait possible

On peut utiiser la librairie Dask pour le calcul parallele et distribue

Le principe de l’apprentissage out-of-core est de diviser les donnees en sous ensemble dans le cas ou celles-ci ne peuvent pas toutes etre chargee en mémoire pour l’entrainement, et ainsi entrainer le model avec des differentes donnees a chaque iterations

3-

Il nest forcement necessaire de supprimer les variables correllees etant donné aue xgboost ou meme le random Forest peuvent gerer ces cas. Il serait tout meme benefique de proceder a leur suppression afin de reduire la dimensionalite des donnes, c a d les nombre de features, et ainsi gagner en temps et en performance du model

FEATURE ENGINEERING – Dummy