Projet Machine Learning

Telco

short line4 DataScience 2

Réalisé par:

BELAIDI Nada

BEN HMIDA Nizar

CHARAABI Khaled

KHECHANA Fares

LAKHAL Abir

MAGHERBI Racha

[**Compréhension du problème métier**](#_wj249xa64hfo) **2**

[problématique:](#_61q0w4ufj2dz) 2

[Objectif:](#_gy5mfzex9e0w) 2

[**Compréhension des données**](#_rorudpeomw2q) **2**

[**Préparation des données**](#_uqtd0acr3ecl) **4**

[**Modélisation**](#_deiyhf4w4gjr) **7**

[1-K Nearest Neighbors](#_73ylibnktg0f) 7

[2-Arbre de décision :CART](#_j702524h5vty) 8

[3-Random Forest:](#_l9okl99wxz8r) 11

[4- Machine à vecteurs de support SVM:](#_wxiikyf3ola5) 13

[5- Naive Bayes:](#_vi4hhxtbmn6s) 13

[**Conclusion**](#_wmmmxdpt6bpk) **14**

# Compréhension du problème métier

## problématique:

Il va sans dire que l’un des problèmes majeurs auquel et souvent confronter n’importe quelle entreprise est celui de satisfaire sa clientèle pour pouvoir la maintenir. pour se faire, elle doit prévoir une stratégie commerciale efficace et rentable

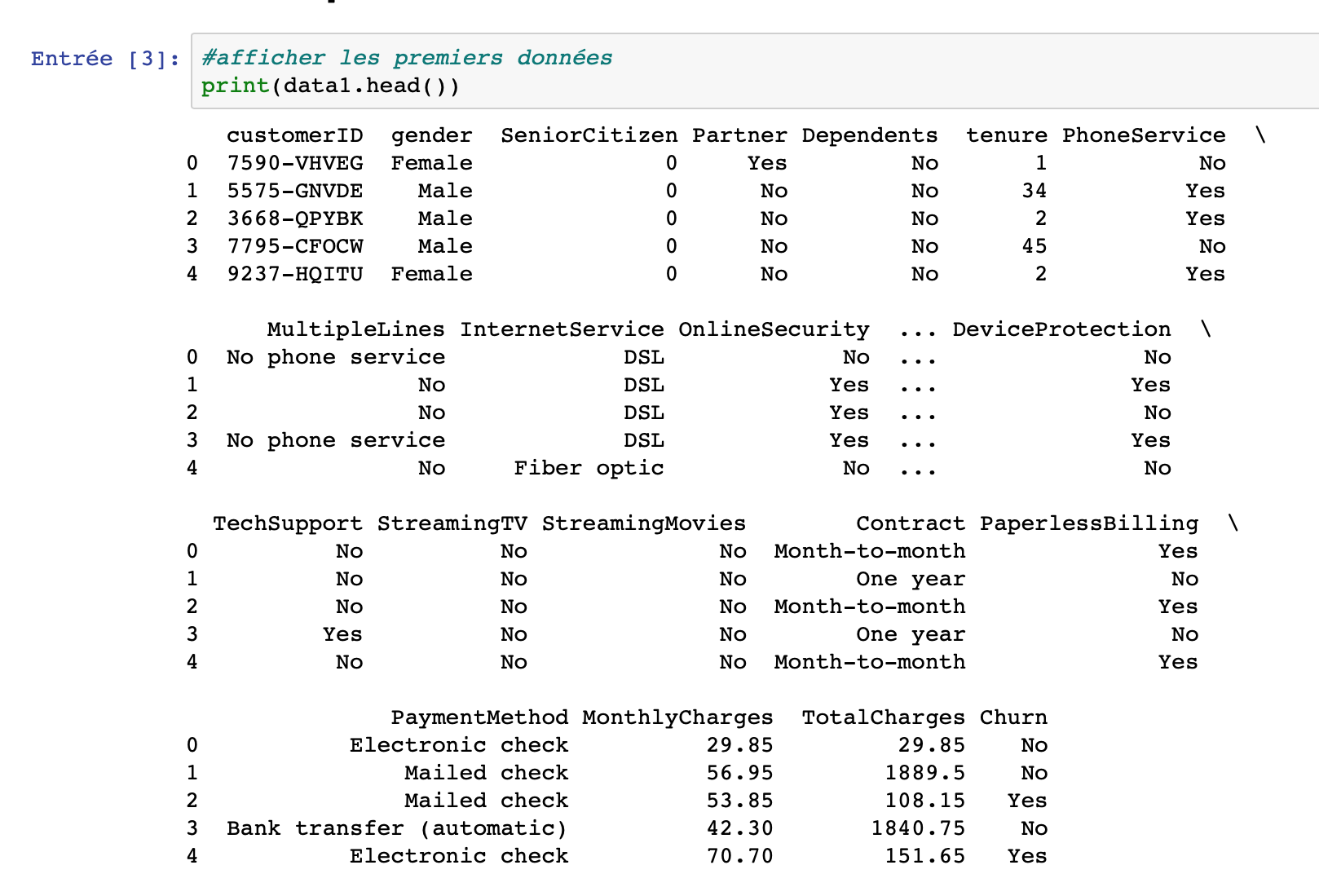
Une telle stratégie doit tenir compte des prévisions des taux de désabonnement et parer à une telle éventualité, du coup la facilité de changer d’opérateur est l’un des défis majeurs auxquels elle doit faire face. Pour fidéliser les clients existants, les organisations doivent améliorer le service-client, hausser la qualité des produits et être en mesure de savoir à l’avance quels clients susceptibles de se désabonner. En prévoyant le taux de désabonnement des clients, les entreprises peuvent immédiatement prendre des mesures pour les fidéliser.

La prédiction peut être effectuée en analysant les données des clients à l'aide de techniques d'exploration.

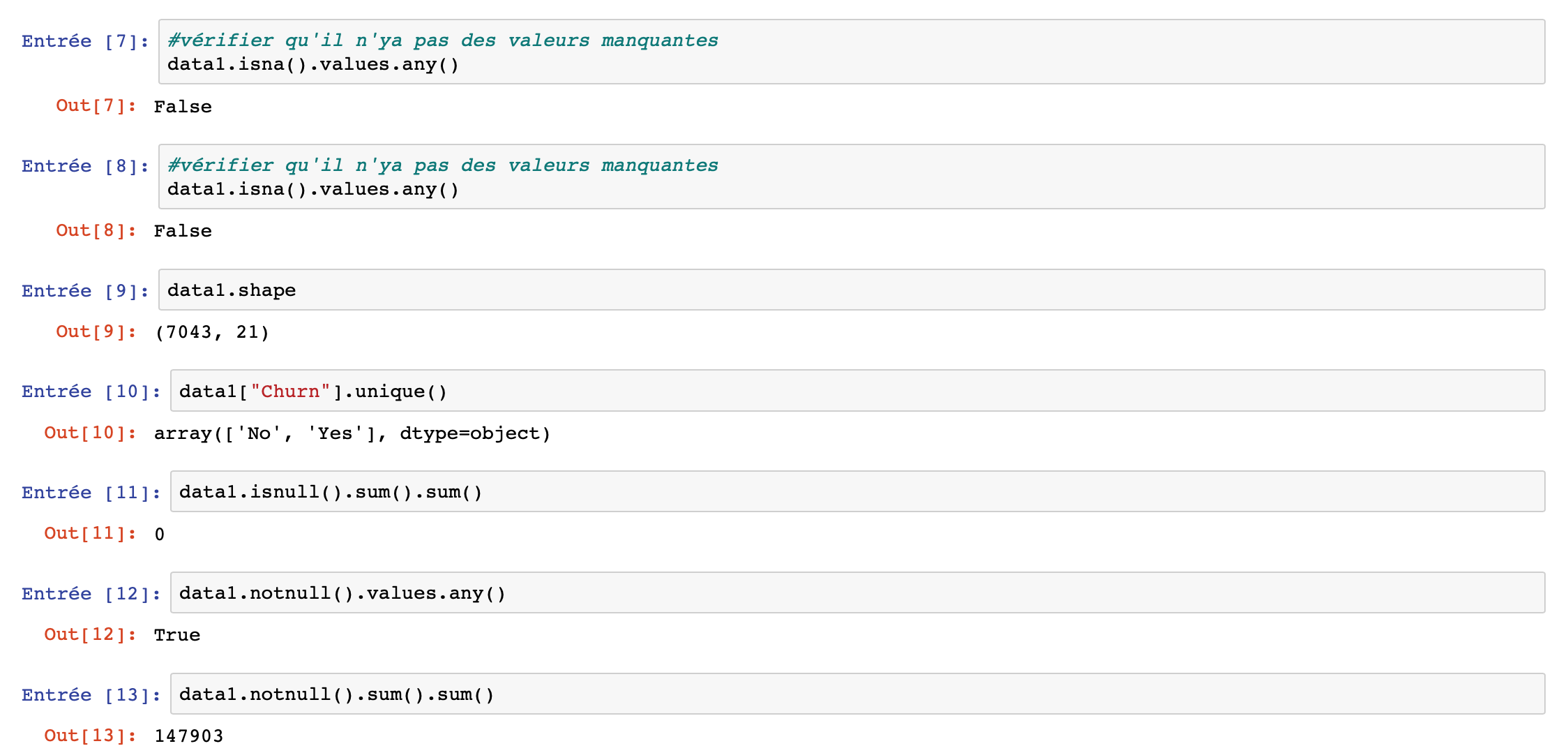
## Objectif:

Au cours de notre projet, notre objectif est de tester et trouver l’algorithme d’apprentissage adéquat pour résoudre un problème de machine learning qui consiste à vérifier si le client peut se désabonner de son opérateur Telco ou non.

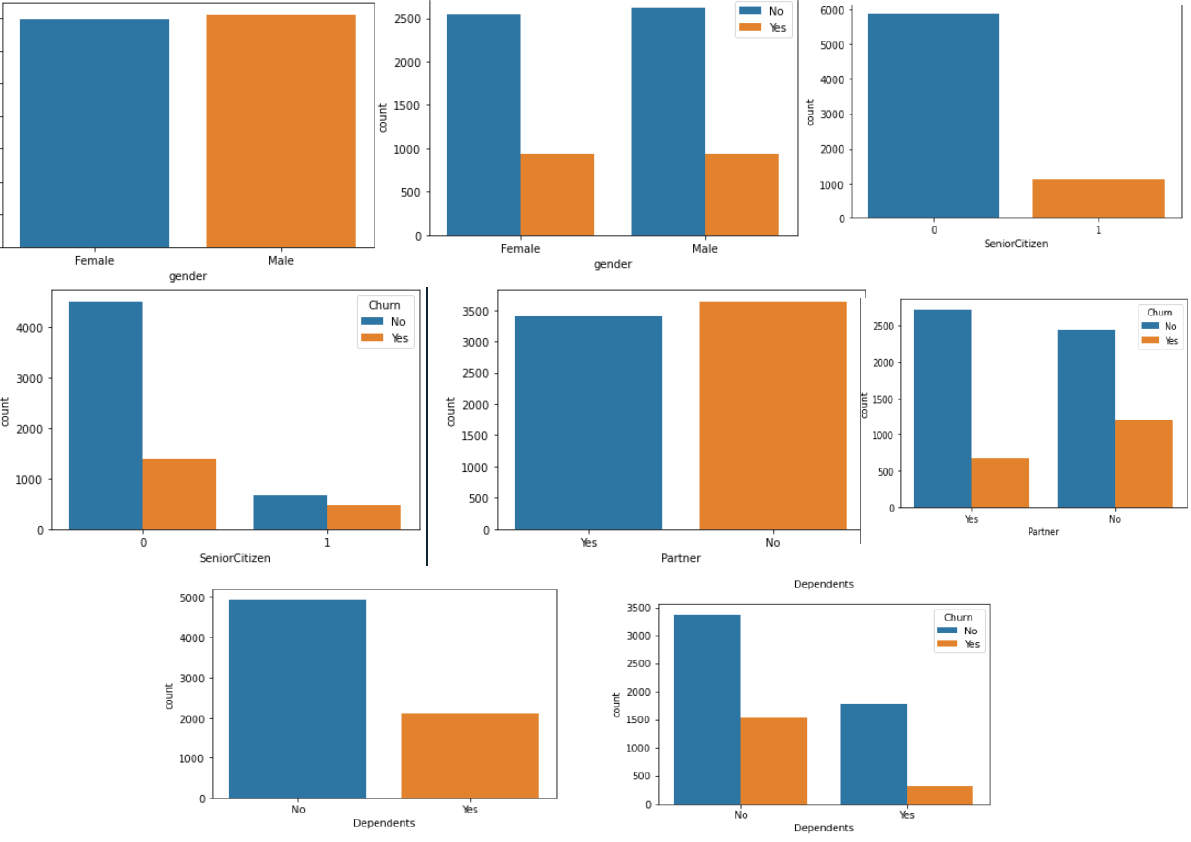
# Compréhension des données



On peut visualiser ici les 5 premières lignes de notre DataFrame.



Ici ,on a testé l’existence des valeurs Nan .Pas de valeurs manquantes ou bien nulles.



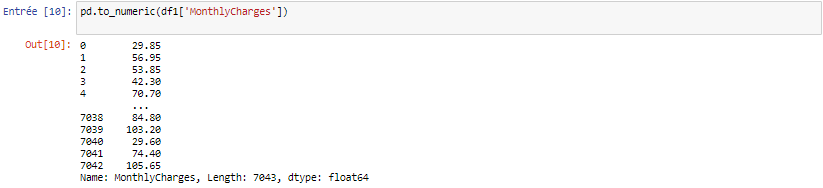
Ces représentations graphiques montrent les relations entre le “Churn” et le reste des fonctionnalités .

# Préparation des données



Pas de valeurs manquantes ou bien nulles, on a pas besoin d’effectuer l’imputation.







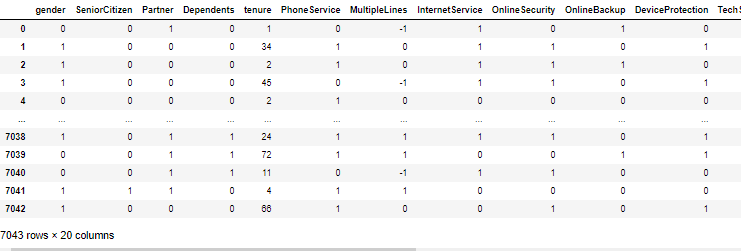
Ici, on a changé les colonnes non numériques en d'autres à valeurs permettant d’effectuer l’ACP.

La valeur 489 ème de la colonne ‘TotalCharges” et un espace qui ne peut pas être transformé en une valeur numérique, on le remplace par la 0.

******

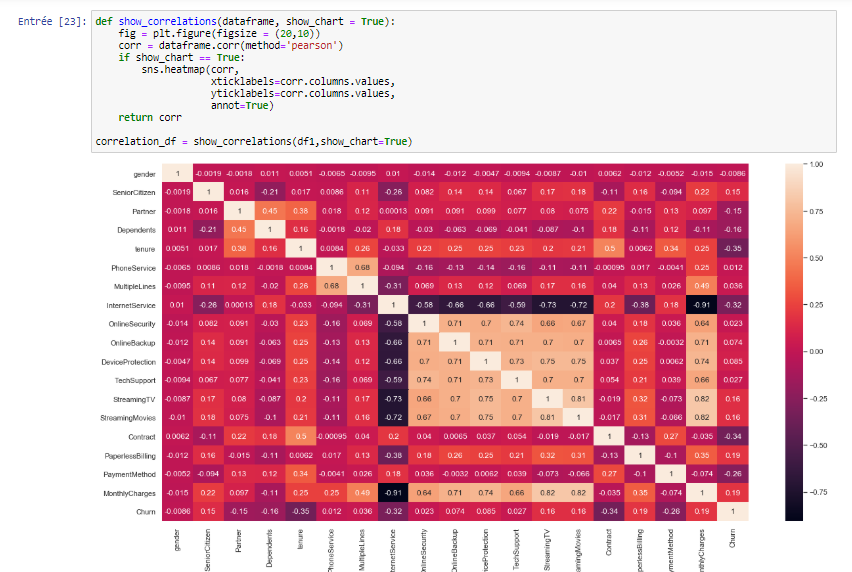
******

On se retrouve avec cette dataframe:



***Feature Selection:***

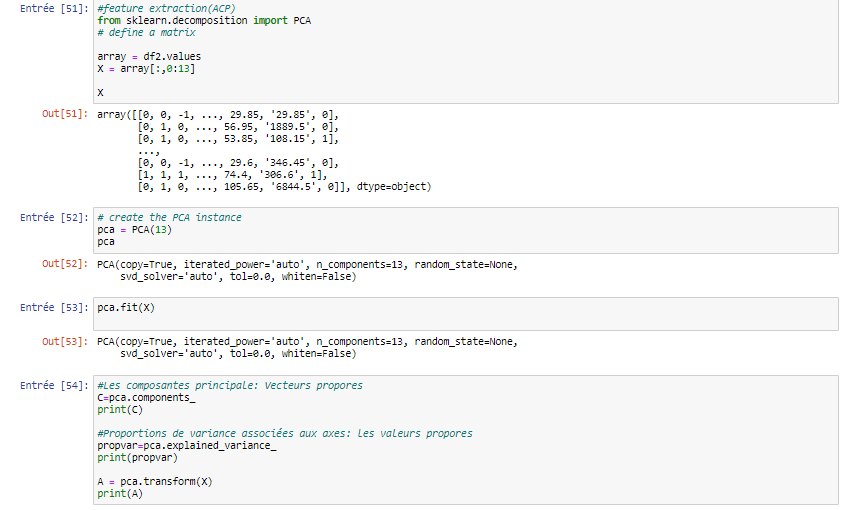
On a opté pour une méthode de sélection basée sur la corrélation comme on nous l'a demandé dans le document du projet et on a affiché notre heat map pour bien voir nos résultats.



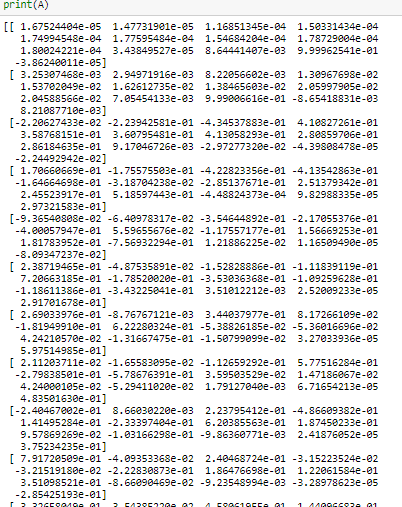
On remarque que la corrélation maximale qu’on a obtenue est 0.19.Donc, on a choisi d'éliminer les corrélations négatives.



On passe maintenant à l'application de l’ACP:



Et voici une partie de ce qu’on a obtenu:

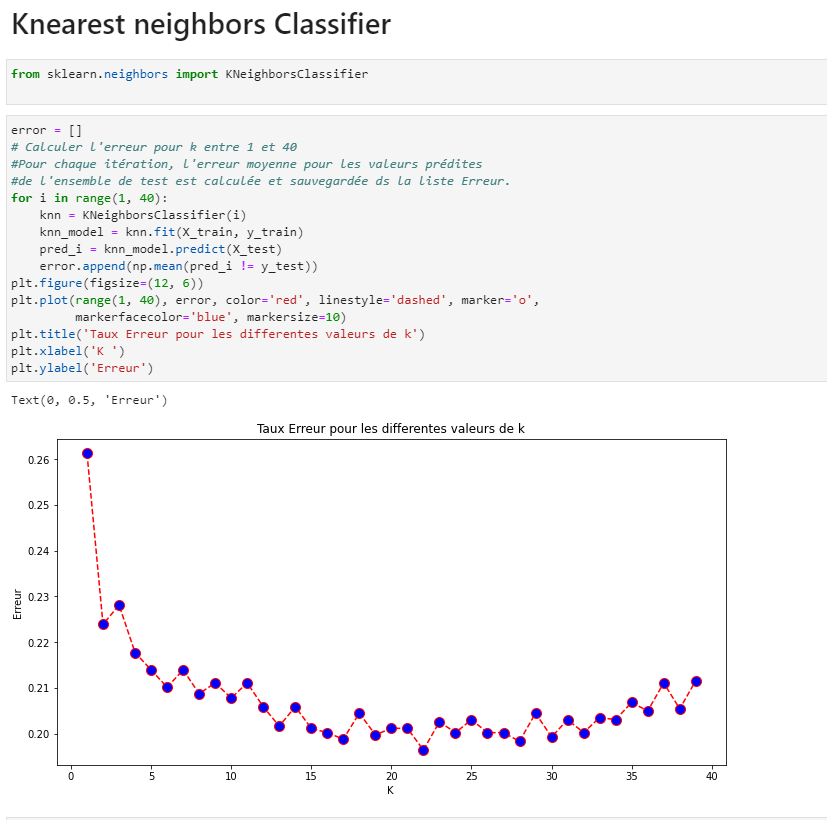


# Modélisation

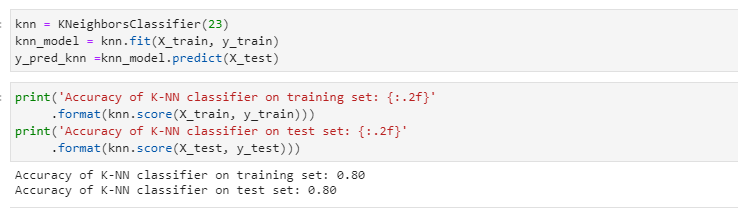
### 1-K Nearest Neighbors

En appliquant le KNN avec une plage allant de 1 vers 40 On a obtenu comme suit le graph donnant le taux d’erreur pour les différentes valeurs de K.

On a choisi la valeur de K la plus petite, c.à.d ayant le taux d’erreur le plus petit d'où on a choisit **K=23.**



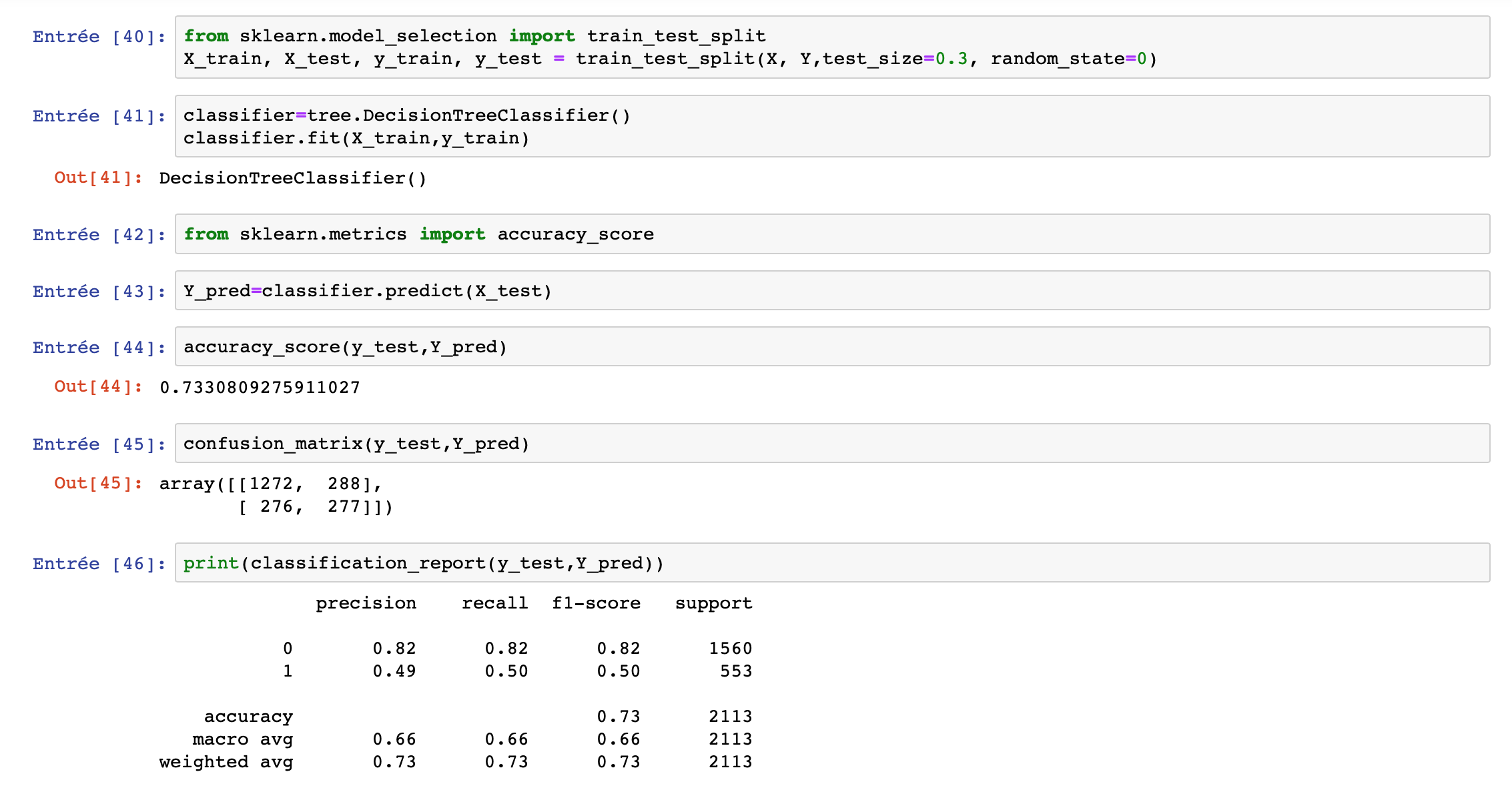
On passant maintenant notre train et test data frames on a obtenu comme suit des résultats de **80% accuracy** pour le **training set** et **80% accuracy** pour le t**est set**



### 2-Arbre de décision :CART

On a divisé l’ensemble des observations X et l’ensemble de classe Y, chacun en deux sous-ensemble :

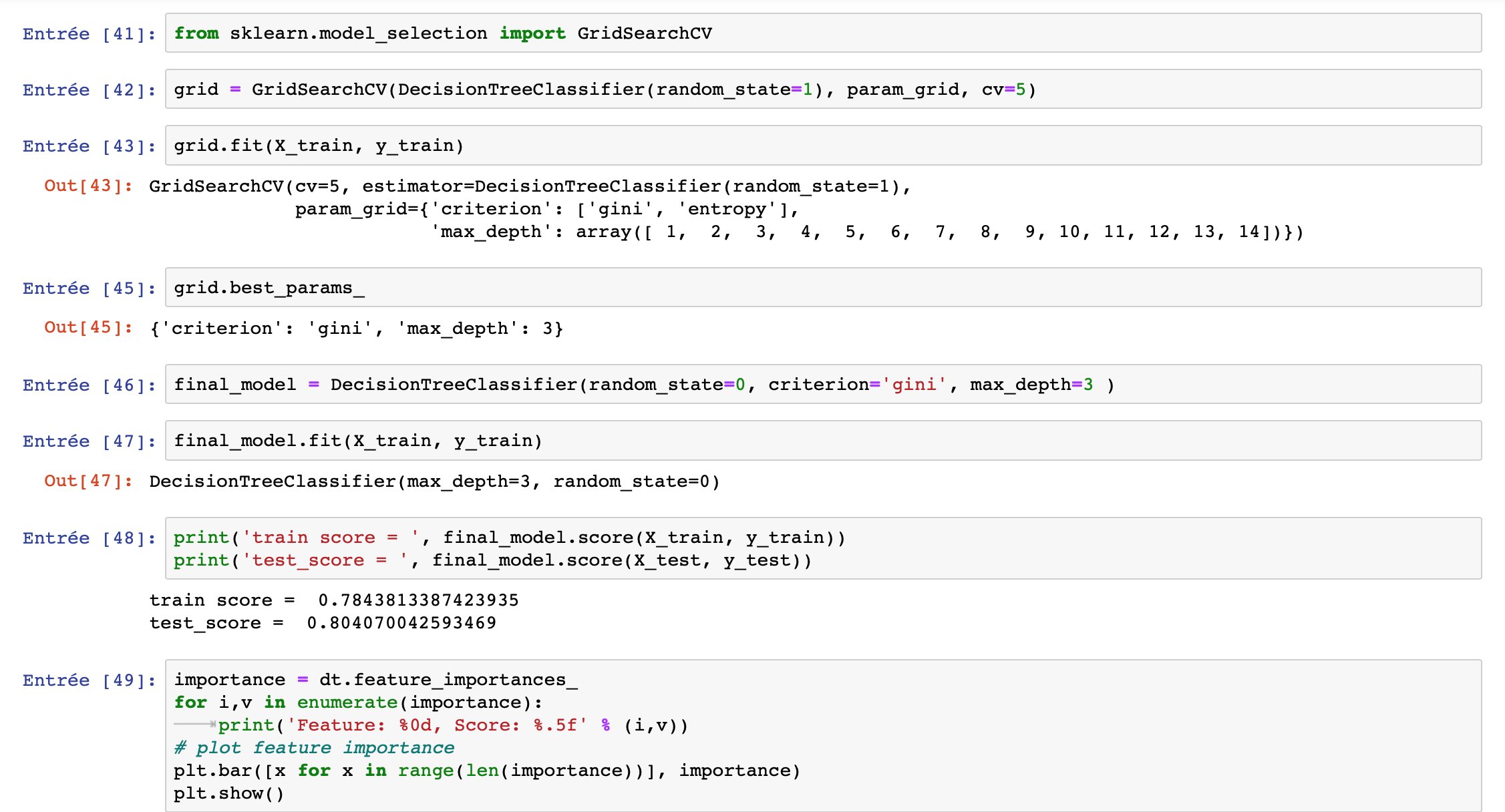
* un sous-ensemble d'apprentissage : 70% de l'ensemble initial
* un sous-ensemble de test : 30% de l'ensemble initial

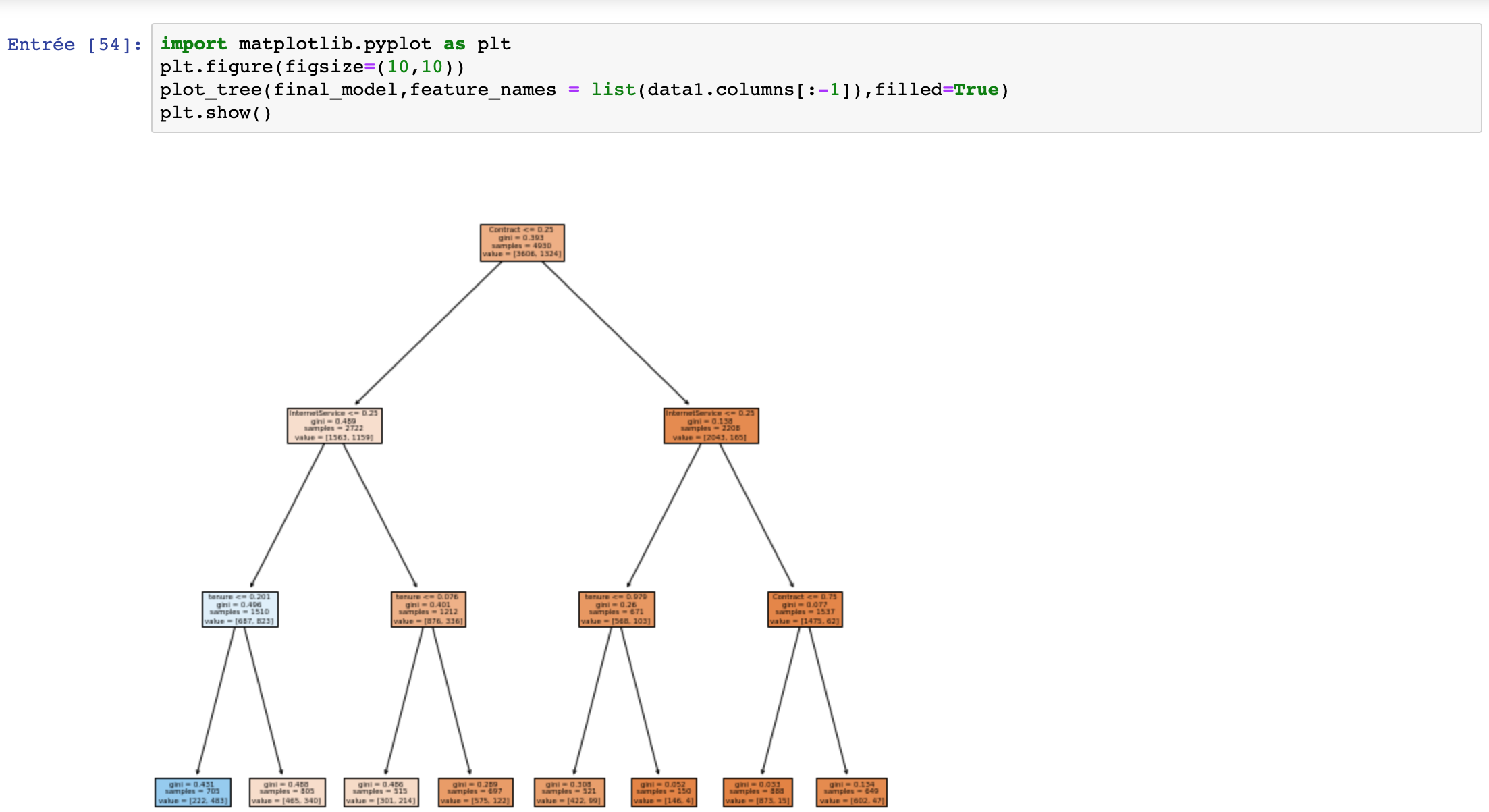


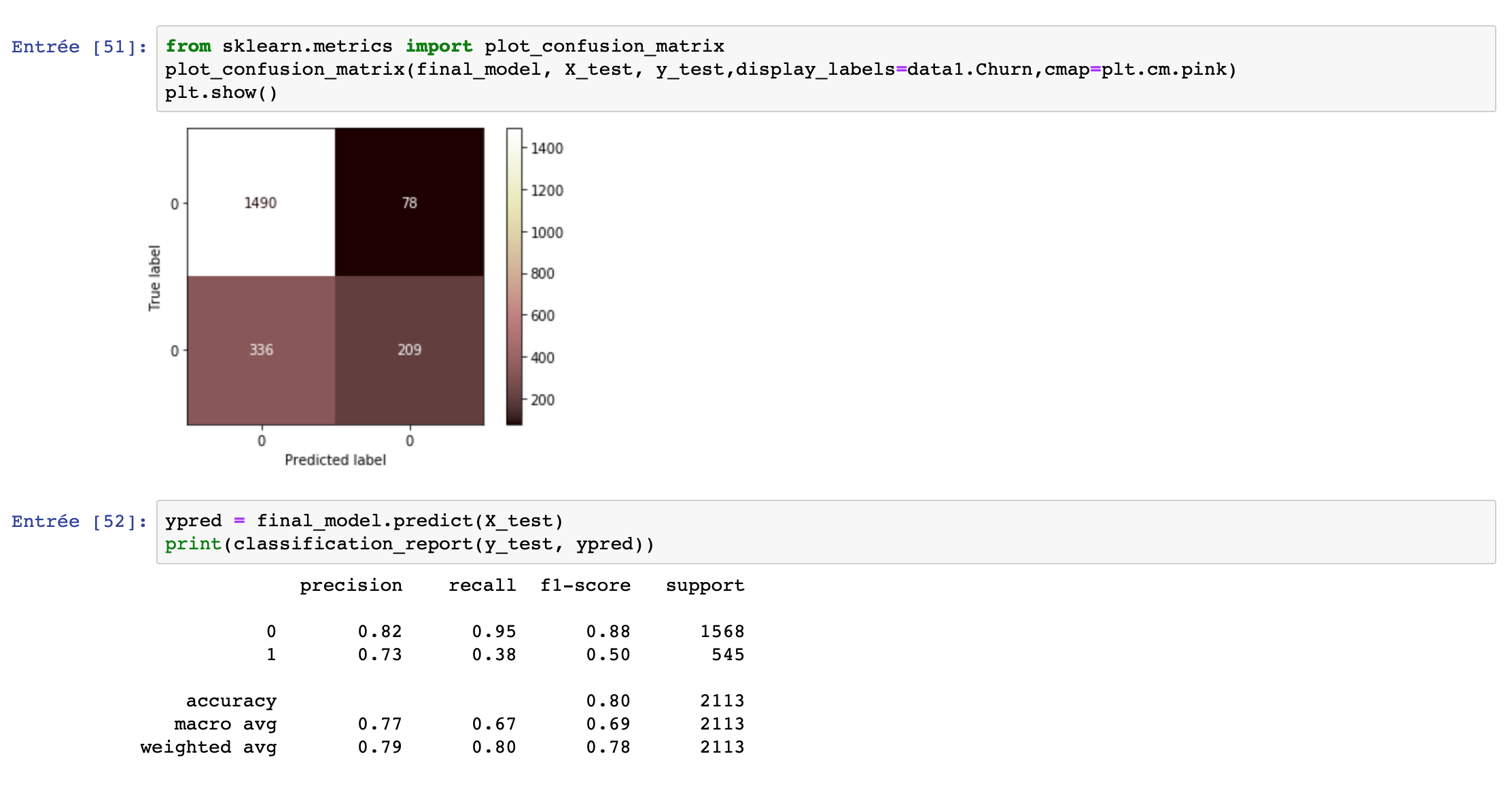
l'application de l'arbre de décision a donnée une performance plus ou moins bonne 0.73 , concernant les précisions on remarque une très bonne prédiction pour les personnes qui ne vont se désabonner (0.82) , par contre il y a une baisse de précision pour les personnes qui vont se désabonner 0.49 (à améliorer )

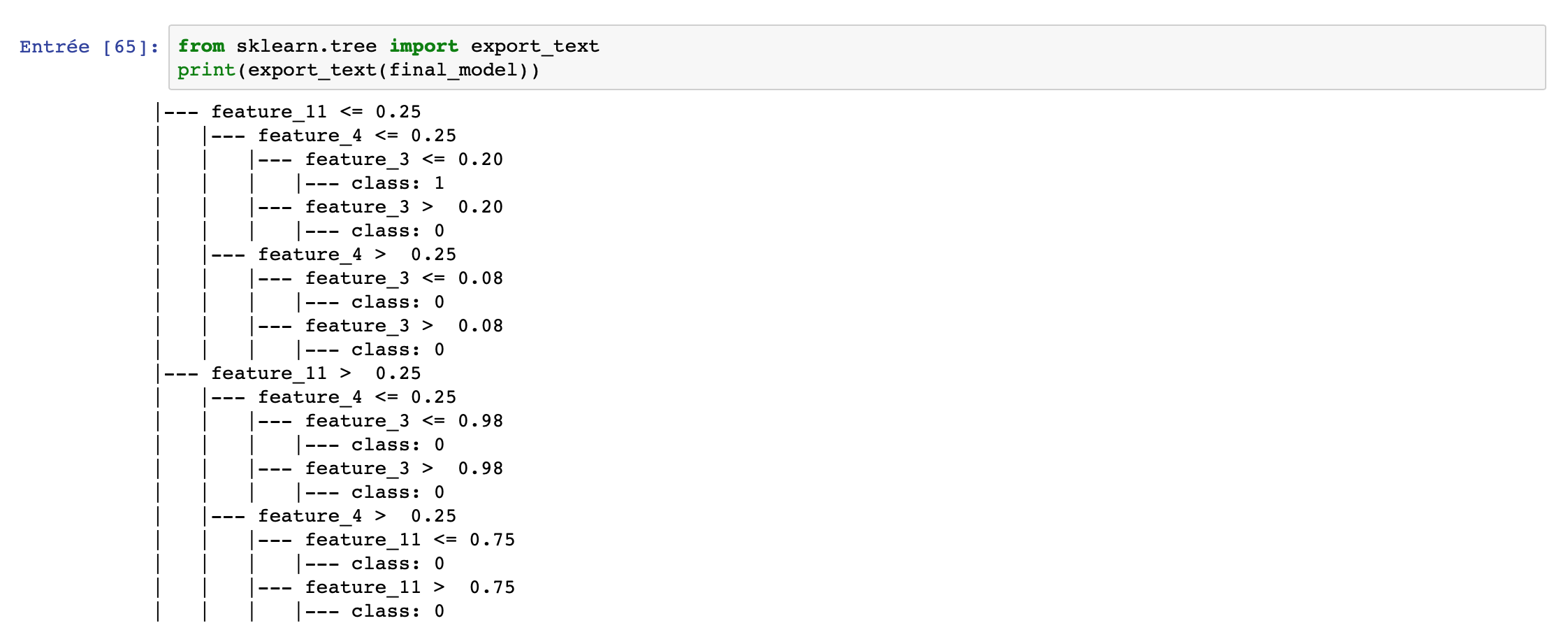
L'arbre est trop long et difficile à interpréter. Pour faire face à ce problème on va essayer de régler les hyperparamètre de notre modèle

#### 







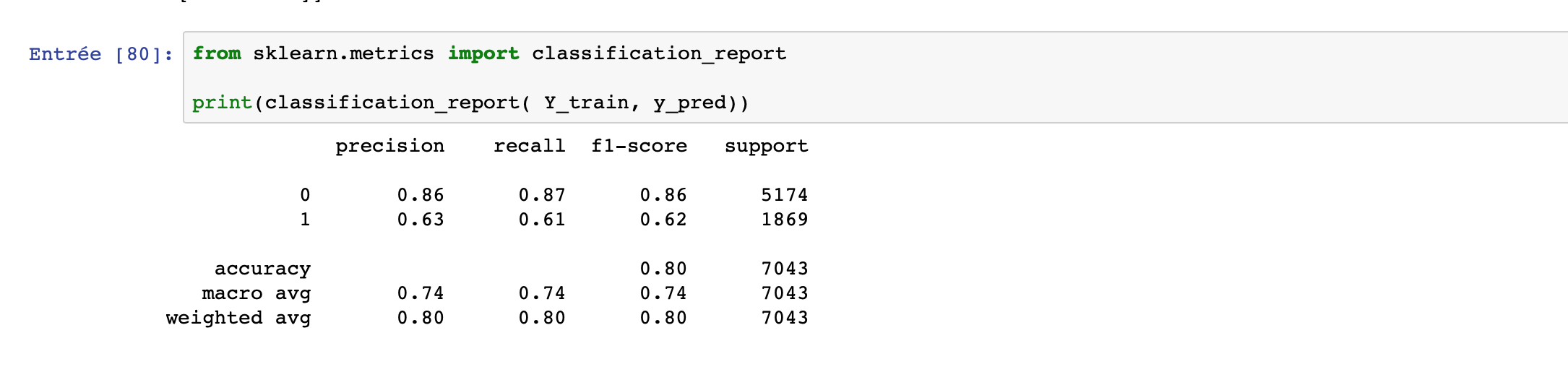


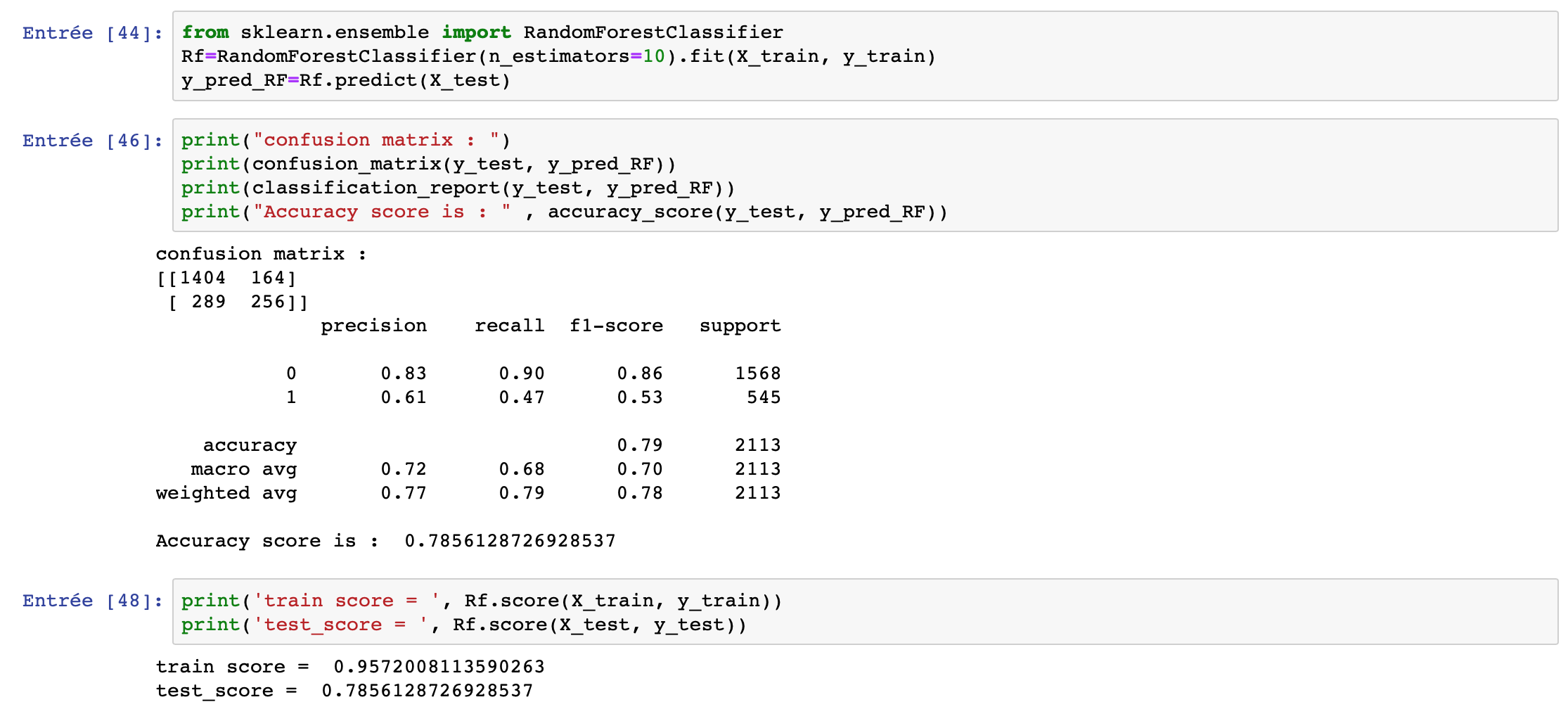
Après application de SearchGridCV qui nous a aidé à déterminer les meilleures valeurs des hyperparamètres réglés et après évaluation de la performance de final\_model sur le sous-ensemble de données approprié on a trouvé 0.82 pour les personnes qui ne vont se désabonner et 0.73 pour les personnes qui vont se désabonner avec une performance de 0.80

|  |  |
| --- | --- |
| train\_score | test\_score |
| 78% | 80% |

### 3-Random Forest:



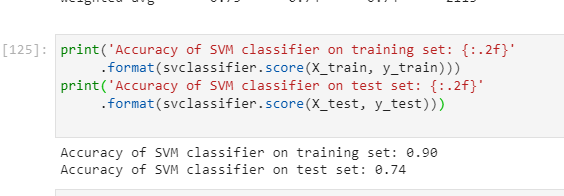




Après traitement de l’algorithme Random Forest, il nous donné une performance de 0.78 , concernant les précisions on remarque une très bonne prédiction pour les personnes qui ne vont se désabonner (0.83) , pour les personnes qui vont se désabonner 0.61

|  |  |
| --- | --- |
| train\_score | test\_score |
| 95% | 78% |

### 4- Machine à vecteurs de support SVM:



l’algorithme Random SVM, il nous donné une performance de 0.80 , concernant les précisions on remarque une très bonne prédiction pour les personnes qui ne vont se désabonner (0.84) , pour les personnes qui vont se désabonner 0.65

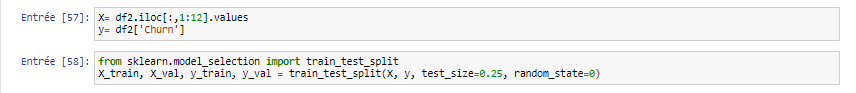
|  |  |
| --- | --- |
| train\_score | test\_score |
| 90% | 74% |

### 5- Naive Bayes:

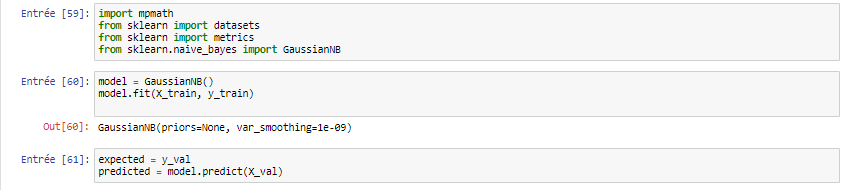
On a commencé par diviser notre dataset en deux parties:

-Partie entrainement (75%) .

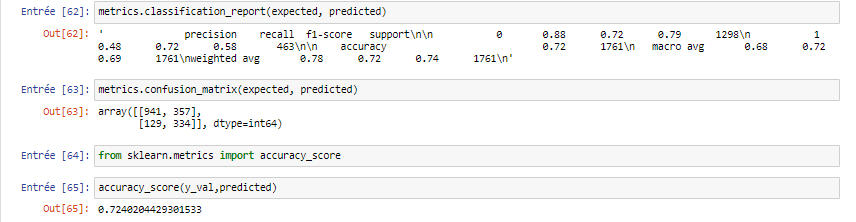
-Partie test (25%) .



Puis on a appliqué notre modèle sur ces deux derniers.



Notre modèle nous a donné les résultats suivants:



# Conclusion

La finalité et l’objectif de cet ensemble des processus qu’on a effectué dans ce projet est d’avoir les plus grands pourcentages possible. En appliquant plusieurs algorithmes diffèrent après avoir passé par les étapes de la méthodologie CRISP on a trouvé que le Classifier **K Nearest Neighbors ( KNN)** nous a donné la plus haute accuracy avec **80% pour le Test set est aussi pour le train set.** Ce résultat n'est pas définitif et peut encore être amélioré.