INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Université Nationale du Vietnam, Hanoï
(UNVH)

Rapport du TP Fouilles de données
Option : Système Intelligent et Multimédia
Niveau : Master I

PHILIPPE Jean Mith & Soumana Hamadou Abdourahmane
PROF :Nguyen Thi Minh Huyen
Juin 2019

Contents

I. Introduction

Dans le cadre du cours Fouille de Donnée, il nous a été demandé de choisir un jeu de donnée pour effectuer les différentes opérations afin d'expérimenter les méthodes d'apprentissage supervisées.
Pour cela, nous avons choisi de travailler sur un ensemble de donnée appelé ensemble de donnée « **Adulte** ». Cet ensemble de données contient 48842 exemples. L'extraction a été effectuée par Barry Becker de la base de données du recensement de 1994. Un ensemble d'enregistrements relativement propres a été extrait en utilisant les conditions suivantes: ((AGE> 16) && (AGI> 100) && (AFNLWGT> 1) && (HRSWK> 0)).
La tâche de prévision consiste à déterminer si une personne gagne plus de 50 000 $ par an en fonction des données du recensement.

II. Description et résumée des données

Pour faire la description de notre ensemble de donnée nous avons utilisé l'outil Tanagra. C'est un logiciel de Data mining gratuit pour l'enseignement et la recherche.

Dataset description

15 attribute(s)
48842 example(s)

Attribute	Category	Informations
age	Continue	-
workclass	Discrete	8 values
fnlwgt	Continue	-
education	Discrete	16 values
education_num	Continue	-
marital_status	Discrete	7 values
occupation	Discrete	14 values
relationship	Discrete	6 values
race	Discrete	5 values
sex	Discrete	2 values
capital_gain	Continue	-
capital_loss	Continue	-
hours_per_week	Continue	-
native_country	Discrete	41 values
class	Discrete	2 values

Comme on peut le voir dans l'image ci-dessus. Notre ensemble de donnée nous contient 15 variables dont 6 **continues** et **9 discrètes.**

Variables Continues

1. **Age**: ce variable indique l'âge de la personne
2. **Final_weight :** Le nombre de personnes que les recenseurs croient que l'observation représente.
3. **Education_num :** Plus haut niveau d'éducation sous forme numérique.
4. **Capital_gain :** Gains en capital enregistrés
5. **Capital_loss :** Indique les pertes de capitale de la personne
6. **Hours_per_week :** Indique les heures de travail par semaine de l'individu

Variables Discrètes

1. **Workclass :** L'attribut Workclass le plus haut niveau d'éducation atteint tel qu'un baccalauréat ou un doctorat
2. **Education :** L'attribut éducation contient le plus haut niveau d'éducation atteint tel qu'un baccalauréat ou un doctorat.
3. **Marital_status :** État civil de l'individu.
4. **Occupation :** indique ce que la personne fait dans sa vie.

5. **Relationship :** Contient les valeurs de relation familiale telles que mari, père, etc., mais n'en contient qu'une par observation. Nous ne savons pas ce que cela est supposé représenter
6. **Race :** indique de quelle race est la personne
7. **Sex:** indique le sexe de la personne
8. **Native_country :** Indique la zone de travail de la personne
9. **Class :** Si la personne gagne plus de 50 000 dollars par an de revenu.

Parmi les variables 6 **continues** et 9 **discrètes**

III. Méthode d'apprentissage

Étant donné que notre ensemble de donnée « **Adulte** » est destiné à la classification par classe binaire, nous avons utilisé la méthode de classification pour prédire si le revenu d'une personne dépasse 50,000$ par an.

III.1 Classification

La classification est une technique d'exploration de données qui assigne des catégories a une collection de données afin de permettre des prévisions et des analyses plus précises. Son objectif est de prédire avec précision la classe cible pour chaque cas dans les données.
Le principal problème est de préparer les données pour la classification pour la prévision. La préparation des données indique les activités suivantes:

- **Nettoyage des données**
- **Analyse de pertinence**
- **Transformation et réduction de données :** Les données peuvent être transformées par l'une des méthodes suivantes.
  - *Normalisation :* Les données sont transformées à l'aide de la normalisation. La normalisation implique la mise à l'échelle de toutes les valeurs pour un attribut donné afin qu'elles tombent dans une petite plage spécifiée. La normalisation est utilisée lorsque, dans la phase d'apprentissage, les réseaux de neurones ou les méthodes impliquant des mesures sont utilisées.
  - *Généralisation :* Les données peuvent également être transformées en les généralisant au concept supérieur. Pour cela, nous pouvons utiliser les hiérarchies de concepts.

IV. Lecture et Nettoyage des données

Le nettoyage des données implique l'élimination du bruit et le traitement des valeurs manquantes. Le bruit est résolu en appliquant des techniques de lissage et le problème des valeurs manquantes est résolu en remplaçant une valeur manquante par la valeur la plus courante pour cet attribut.

IV.1 Répartition du jeu de données

Afin de s'assurer que chaque côté contient une proportion raisonnable de chaque classe. On a partagé notre jeu de donné en 2/3 apprentissage, 1/3 test. On note cette proportion *t*. Deux classes 0 et 1, la proportion de la classe 1 est de *p* qu'on choisit petit. Cette répartition se fait de manière aléatoire les données afin de s'assurer que les jeux de test et d'apprentissage sont semblables. L'utilisation de données similaires pour l'apprentissage et les tests vous permet de minimiser les effets des différences de données et de mieux comprendre les caractéristiques du modèle.

	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_
32556	27	Private	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0.0	
32557	40	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0.0	
32558	58	Private	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0.0	
32559	22	Private	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0.0	
32560	52	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024.0	

Fig3 : Quelque ligne de notre ensemble de données d'entraînement

	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss
0	25	Private	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0.0	0
1	38	Private	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0.0	0
2	28	Local-gov	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0.0	0
3	44	Private	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688.0	0
4	18	?	Some-college	10	Never-married	?	Own-child	White	Female	0.0	0

Fig5 : Quelque ligne de notre ensemble de données de test

IV Colonnes à supprimer

Il semble que la colonne **final_weight** indique la proportion de la population qui possède le même ensemble de fonctionnalités. Fondamentalement, chaque ligne de la table d'origine était dé-dupliquer et final_weight stocke le nombre de lignes ayant exactement la même valeur. Nous n'avons certainement pas besoin d'utiliser cette colonne lors de la formation de modèle.

IV.3 Vérification des données

Pour nous assurer que notre ensemble de données d'entraînement a l'air correct et qu'il n'y a aucun problème nous allons faire une vérification s'il n'y a pas de NaN (Valeur manquante) et s'il n'y a pas de duplication dans notre ensemble de données.

```
test_data.duplicated()
```

```
0        False
1        False
2        False
3        False
4        False
5        False
6        False
         ...
16274    False
16275    False
16276    False
16277    False
16278    False
16279    False
16280    False
Length: 16281, dtype: bool
```

```
test_data.isnull().values.any()
```

```
False
```

Le résultat montre que nous n'avons pas de valeur null et pas de duplication dans notre de jeu de données pour l'entraînement.

Pour continuer avec notre travaille nous avons combiné ensemble de données d'entraînement et l'ensemble de données de test afin de généraliser les problèmes observes dans les données.
Chaque ensemble de données a 14 colonnes de prédicteur et l'ensemble de données d'apprentissage a une colonne supplémentaire avec une classe libellée que nous devons prédire.

	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_ga
count	48842.000000	48842	48842	48842.000000	48842	48842	48842	48842	48842	48842.0000
unique	NaN	9	16	NaN	7	15	6	5	2	Na
top	NaN	Private	HS-grad	NaN	Married-civ-spouse	Prof-specialty	Husband	White	Male	Na
freq	NaN	33906	15784	NaN	22379	6172	19716	41762	32650	Na
mean	38.643585	NaN	NaN	10.078089	NaN	NaN	NaN	NaN	NaN	1079.0676
std	13.710510	NaN	NaN	2.570973	NaN	NaN	NaN	NaN	NaN	7452.0190
min	17.000000	NaN	NaN	1.000000	NaN	NaN	NaN	NaN	NaN	0.0000
25%	28.000000	NaN	NaN	9.000000	NaN	NaN	NaN	NaN	NaN	0.0000
50%	37.000000	NaN	NaN	10.000000	NaN	NaN	NaN	NaN	NaN	0.0000
75%	48.000000	NaN	NaN	12.000000	NaN	NaN	NaN	NaN	NaN	0.0000
max	90.000000	NaN	NaN	16.000000	NaN	NaN	NaN	NaN	NaN	99999.0000

fig9: Ensemble de données d'entraînement et de test combine

Ici on remarque que les données de test ont un point supplémentaire à la fin du nom de classe, ce qui doit être corrigé dans la procédure finale de nettoyage des données.

IV.4 Catégorie supplémentaire

```
age                0
workclass       2799
education          0
education_num      0
marital_status     0
occupation      2809
relationship       0
race               0
sex                0
capital_gain       0
capital_loss       0
hours_per_week     0
native_country   857
income_class       0
dtype: int64
```

Si nous comparons le nombre de caractéristiques uniques pour d'autres variables, il est aisé de voir que **workclass**, **occupation** et **native_country** ont une valeur unique supplémentaire dans les données (**?**).
En outre, il est évident qu'il existe des espaces dans les colonnes qui peuvent être nettoyés pendant l'état d'analyse des données. Il peut être corrigé avec l'argument supplémentaire ***skipinitialspace*** dans la fonction read_csv.

IV.5 Premier étape du nettoyage

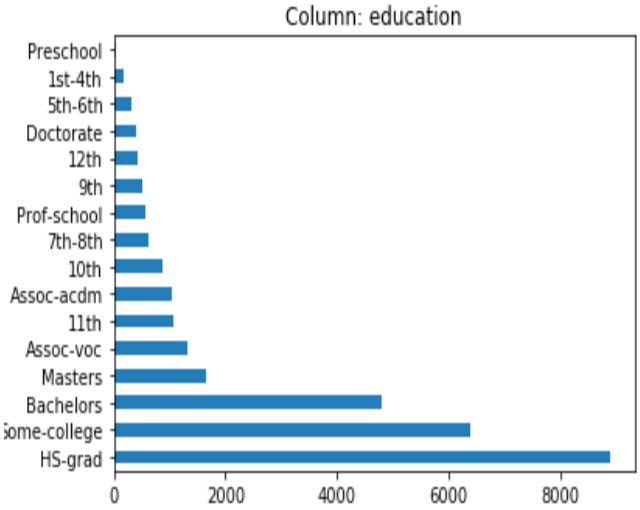
Dans cette étape nous allons résoudre les problèmes les plus importants rencontrés jusqu'à présent. Sans corrections, il sera plus difficile d'analyser les données.
Comme nous l'avons vu dans la ***fig11,*** l'ensemble de données de test a un point (**.**) à la fin, nous allons le supprimer afin d'unifier les noms entre les ensembles de données d'apprentissage et de test.
Les doublons peuvent créer des biais lors de l'analyse et de la prédiction stade, ils pourraient donner des résultats trop optimistes (ou pessimistes).

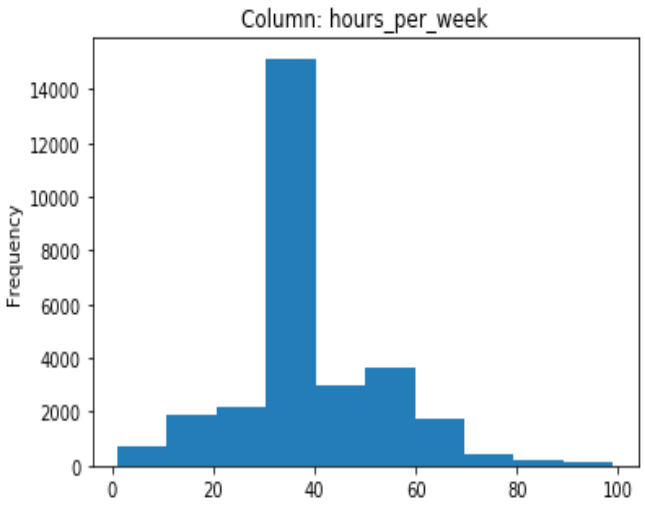
```
32561
29096
Doublons supprime: 10.64%
```

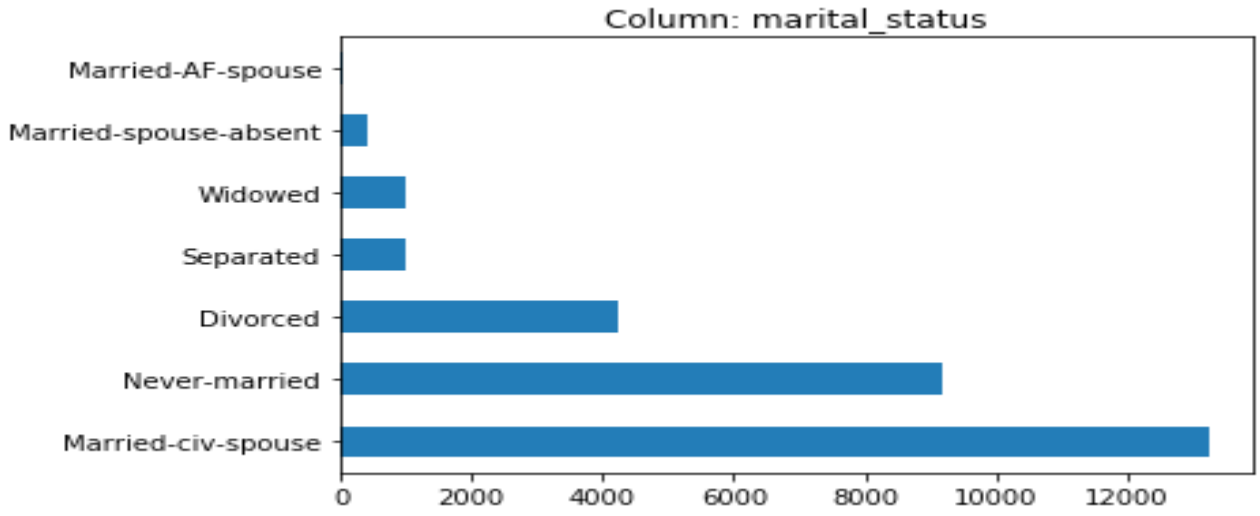
Après la suppression du variable final_weight, nous avons 10% de doublons dans l'ensemble de données pour l'entraînement.

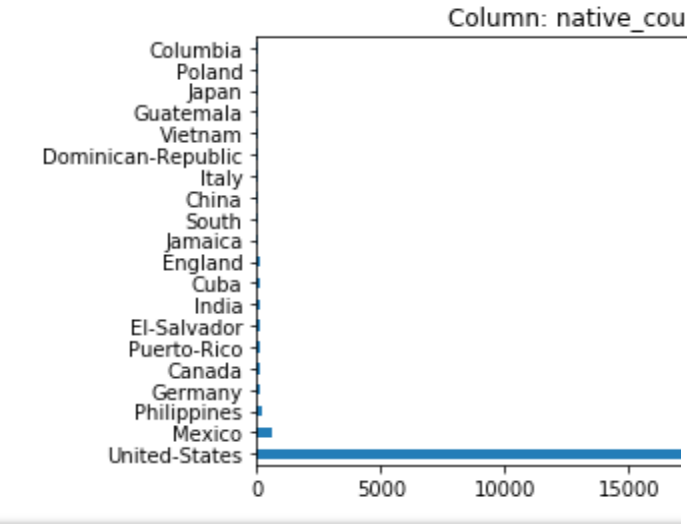
Nous avons faire la déduplication par ensemble de données, mais il y a des doublons entre l'ensemble de données de formation et l'ensemble de données de test. Avec des doublons entres les ensembles de données, nous pourrions obtenir des résultats trop confiants.

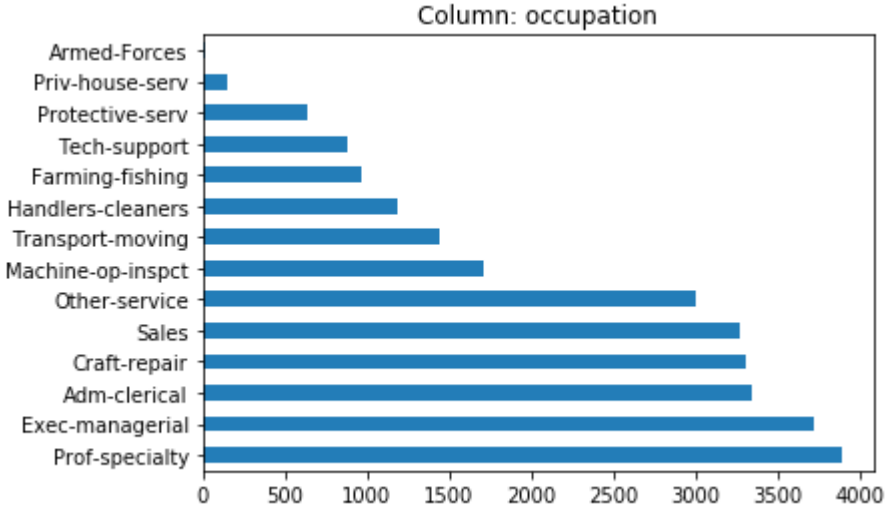
IV.6 Vérification des comptes par catégories

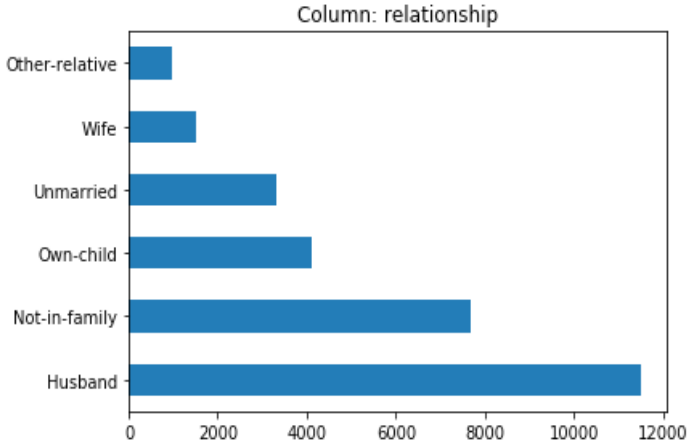


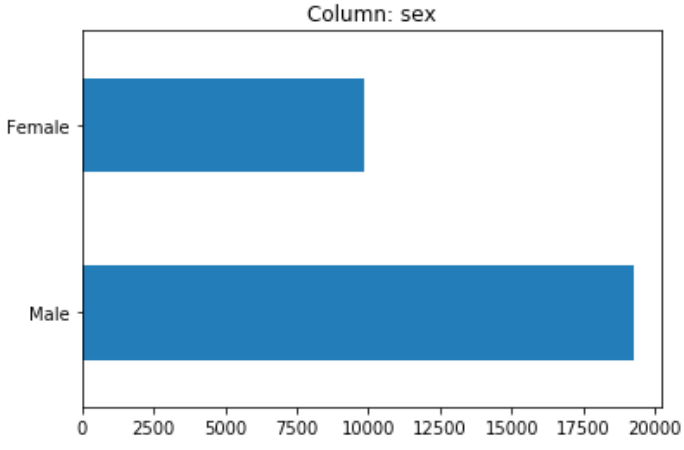


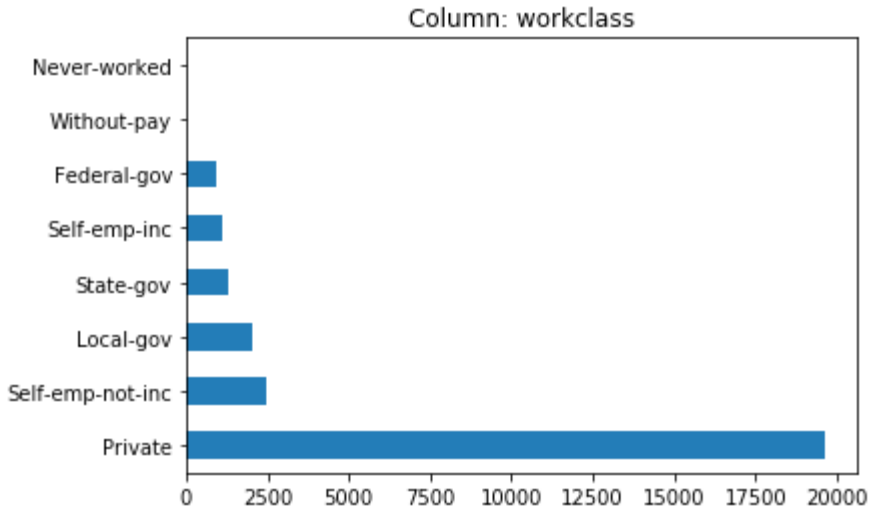



Column: native_country
Columbia
Poland
Japan
Guatemala
Vietnam
Dominican-Republic
Italy
China
South
Jamaica
England
Cuba
India
El-Salvador
Puerto-Rico
Canada
Germany
Philippines
Mexico
United-States
0
5000
10000
15000


Column: occupation
Armed-Forces
Priv-house-serv
Protective-serv
Tech-support
Farming-fishing
Handlers-cleaners
Transport-moving
Machine-op-inspct
Other-service
Sales
Craft-repair
Adm-clerical
Exec-managerial
Prof-specialty
0
500
1000
1500
2000
2500
3000
3500
4000


Column: relationship
Other-relative
Wife
Unmarried
Own-child
Not-in-family
Husband
0
2000
4000
6000
8000
10000
12000


Column: sex
Female
Male
0
2500
5000
7500
10000
12500
15000
17500
20000



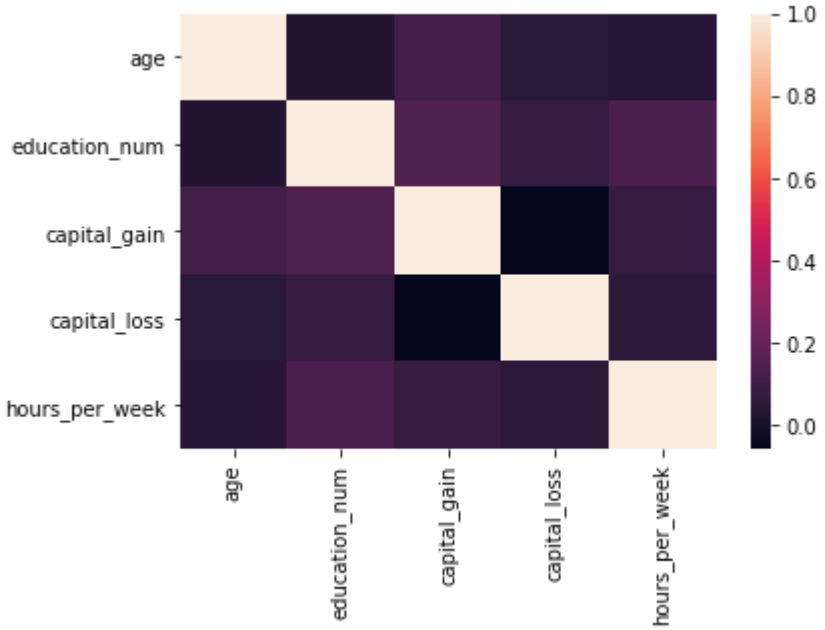
Observations:

1. La variable ***race*** a un grand nombre de valeur ***white***
2. La variable ***workclass*** a un grand nombre de valeur ***Private***
3. La variable ***native_country*** a un grand nombre de valeur ***United-states***.
   Cette variable peut être ignorée ou remplacée par une variable binaire. Avec la valeur **Vrai** si la personne est des États-Unis sinon **Faux.**
4. La variable ***sex*** a un grand nombre de valeur **Male**
5. Pour la variable ***relationship*** la valeur Husband est dominante
6. Pour la variable ***education*** la valeur **Hs-grad** est dominante
7. Pour la variable ***occupation*** la valeur prof-spécialité est dominante

Pour la variable ***marial_status*** la valeur **Married-civ-spouse** est dominante

V Corrélation

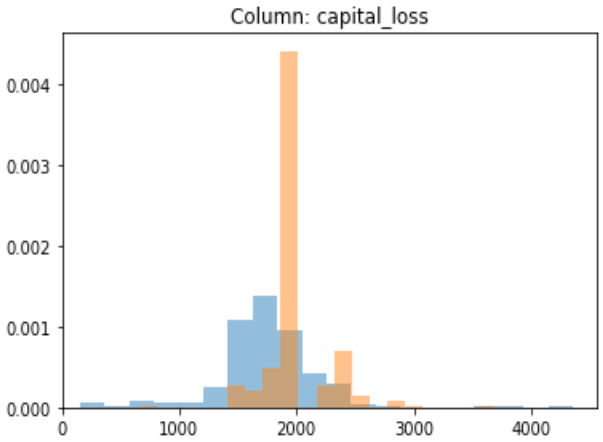
Notre ensemble de données peut également avoir les attributs non pertinents. L'analyse de corrélation permet de savoir si deux attributs donnés sont liés ou pas.

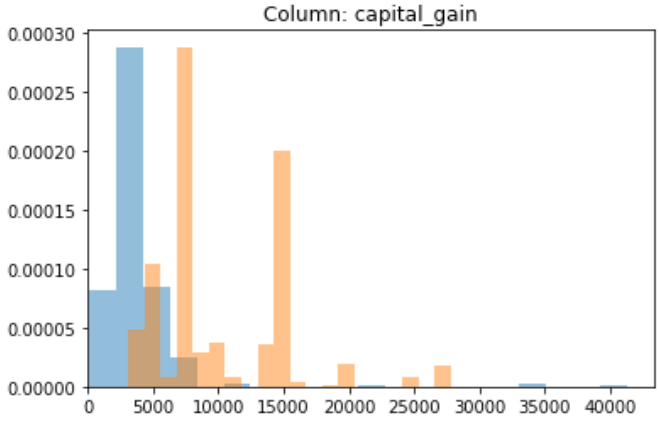


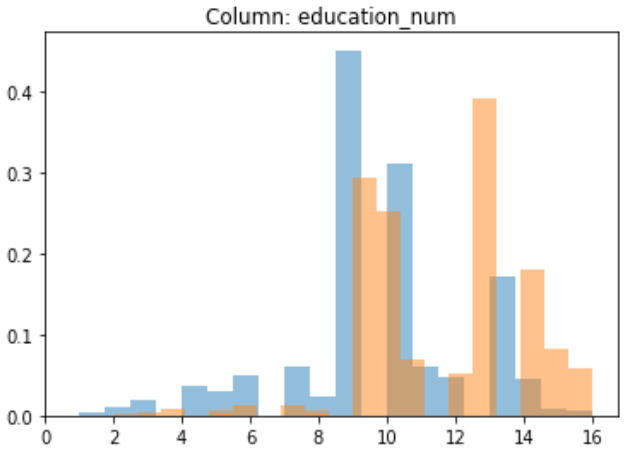
Les fonctions numériques ont de très faibles corrélations.

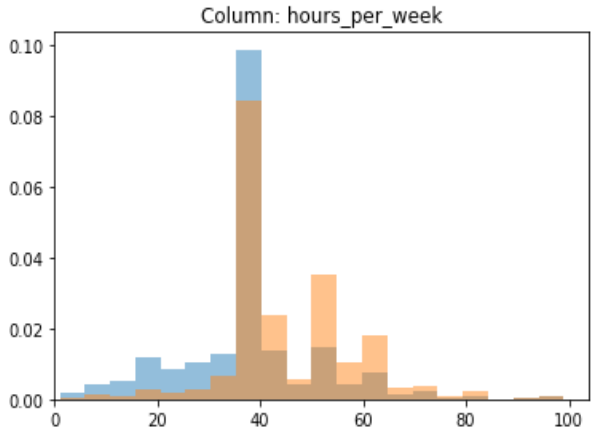
VI Séparation des effets par catégorie

VI Variable continues

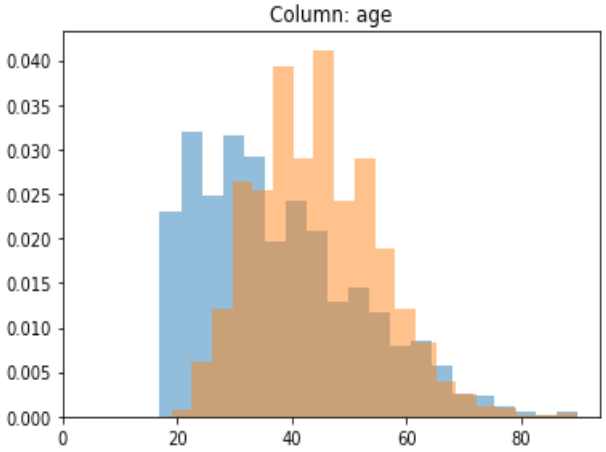








Observation



- Plus la personne est âgée plus son revenu sera grande.
- Plus le nombre d'années d'études est élevé, plus le revenu est élevé.
- Plus une personne travaille, plus son revenu est élevé.

Notons qu'il existe des catégories qui, avec exactement le même ensemble d'entrées s'attendent à prévoir différentes classes de revenus. Aucune caractéristique disponible ne permet de différencier ces classes de revenu.

VII Construction et évaluation des modeles

VII.2 Modèle linéaire

Nous allons utiliser une **régression logistique** pour notre premier prototype c'est un modèle de **régression binomiale**.

La régression logistique est l'un des modèles d'analyse multivariée les plus couramment utilisés en épidémiologie. Elle permet de mesurer l'association entre la survenue d'un évènement (variable expliquée qualitative) et les facteurs susceptibles de l'influencer (variables explicatives). Le choix des variables explicatives intégrées au modèle de régression logistique est basé sur une connaissance préalable de la physiopathologie de la maladie et sur l'association statistique entre la variable et l'évènement, mesurée par l'odds ratio.

Les principales étapes de sa réalisation, les conditions d'applications à vérifier, ainsi que les outils essentiels à son interprétation sont exposées de manière concise.

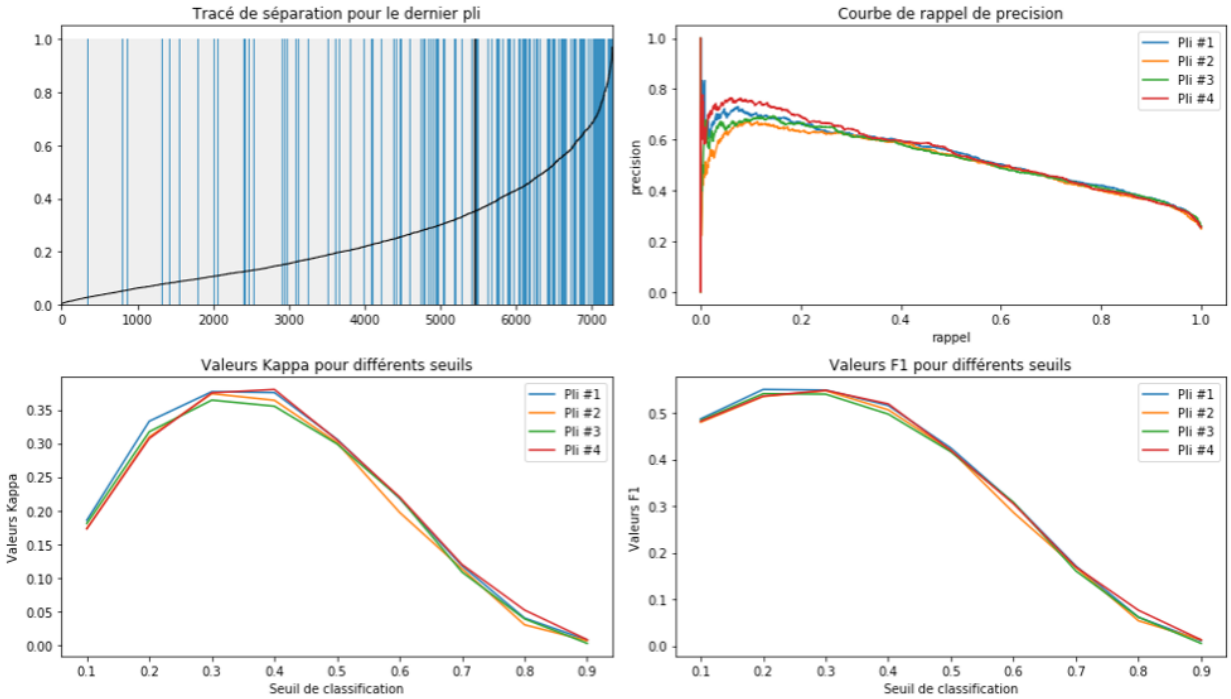
Comme pour tous les modèles de **régression binomiale**, il s'agit de modéliser au mieux un modèle mathématique simple à des observations réelles nombreuses.

Pour commencer nous allons utiliser trois variables numériques.

```
Fold #1
Score ROC AUC : 0.795
Score kappa : 0.306
Score F1 : 0.425
Precision : 0.782
------------------------------
Fold #2
Score ROC AUC : 0.785
Score kappa : 0.301
Score F1 : 0.420
Precision : 0.781
------------------------------
Fold #3
Score ROC AUC : 0.792
Score kappa : 0.299
Score F1 : 0.416
Precision : 0.782
------------------------------
Fold #4
Score ROC AUC : 0.790
Score kappa : 0.305
Score F1 : 0.419
Precision : 0.784
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.790
Moyenne Kappa dans les plis : 0.303
Moyenne F1 dans les plis : 0.420
Moyenne de precision dans les plis : 0.782
```

Seulement trois (3) variables donnent une prédiction avec un score ROC AUC presque égal à 0.8, C’est un bon début. Par contre, les scores kappa et f1 sont faibles.
Ce score dépend du seuil de 0.5. Et c’est simplement un mauvais choix pour la séparation de classe.

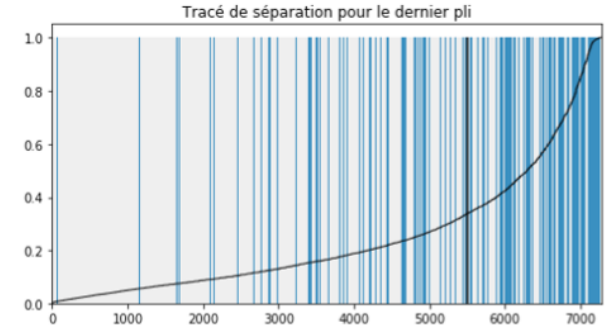


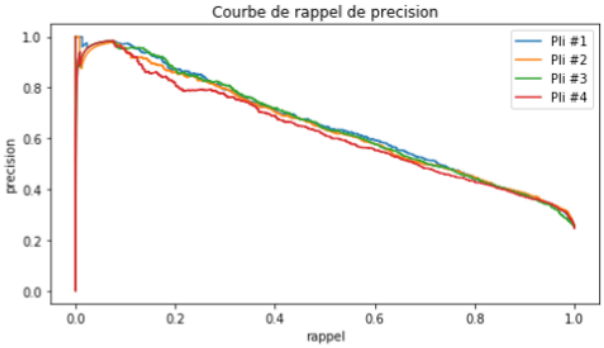
Sur les deux graphiques inférieurs, nous pouvons voir que les scores f1 et kappa peuvent être améliorés si nous choisissons un seuil égal à 0.3.
Le résultat de l'analyse des corrélations a montré que les entités ont une faible corrélation entre elle, comme nous avons normalisé toutes les entités à une seule échelle, nous pouvons utiliser les coefficients de la régression logistique afin d'interpréter les décisions de la classification.

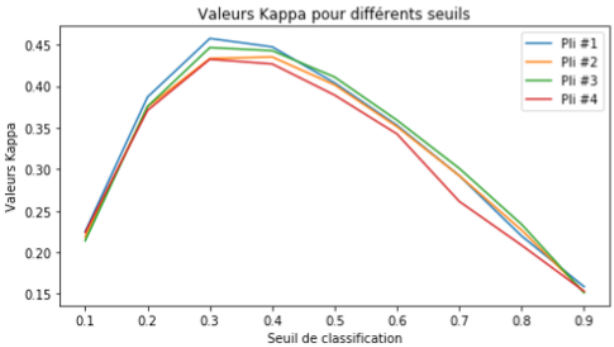
Pour améliorer le résultat nous avons ajouté deux autres variables numériques (***capital_gain*** et ***capital_loss***) et nous avons obtenu une amélioration importante des scores kappa et f1 (de 0,1%) et notable, mais une augmentation relativement faible du score de l'aire sous la courbe ROC (de 0,79 à 0,825).
En plus du graphique de séparation, on peut remarquer qu'il y a beaucoup de régions bleues sur le côté droit du graphique, ce qui signifie que le réseau est plus confiant dans la prédiction de certaines des classes positives.

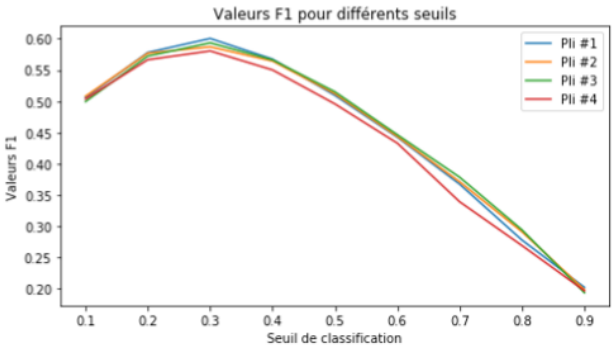
```
Fold #1
Score ROC AUC : 0.830
Score kappa : 0.404
Score F1 : 0.509
Precision : 0.811
------------------------------
Fold #2
Score ROC AUC : 0.822
Score kappa : 0.402
Score F1 : 0.512
Precision : 0.806
------------------------------
Fold #3
Score ROC AUC : 0.825
Score kappa : 0.411
Score F1 : 0.515
Precision : 0.814
------------------------------
Fold #4
Score ROC AUC : 0.821
Score kappa : 0.389
Score F1 : 0.496
Precision : 0.808
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.825
Moyenne Kappa dans les plis : 0.402
Moyenne F1 dans les plis : 0.508
Moyenne de precision dans les plis : 0.810
```





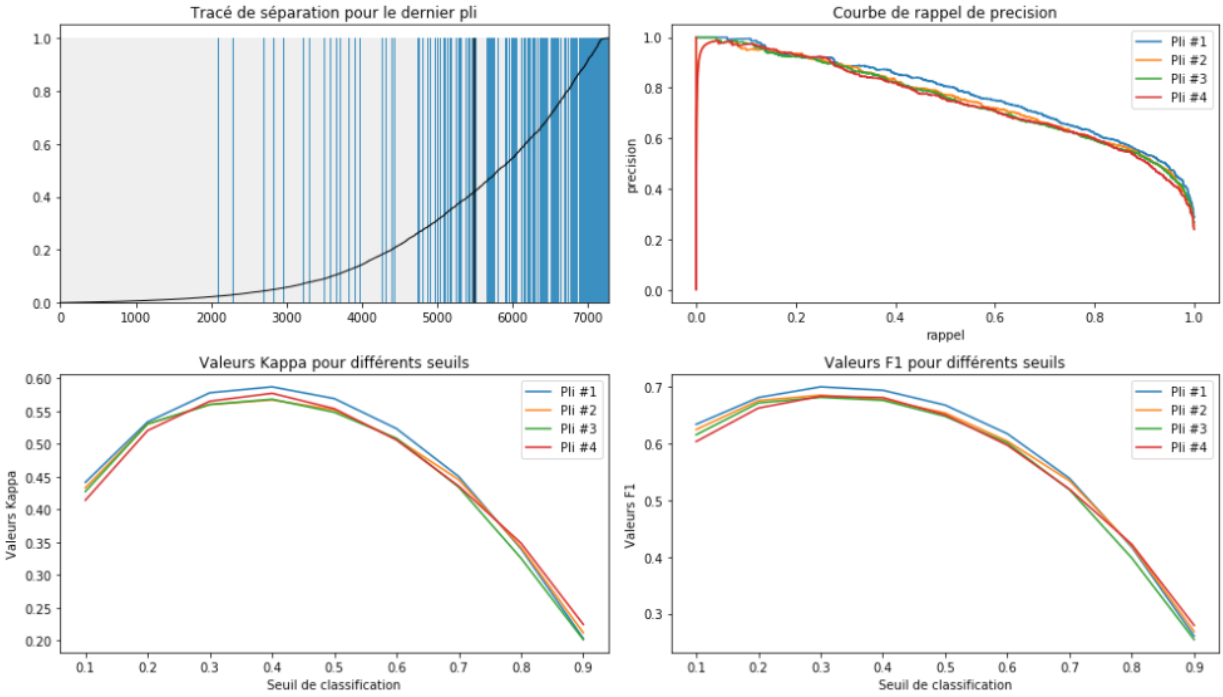




La partie gauche du graphique de séparation présente un nombre réduit de cas de faux négatifs (erreur de type 2).
Nous avons transformé en variables binaires les variables discrètes et les ajouter au model.

```
Fold #1
Score ROC AUC : 0.924
Score kappa : 0.624
Score F1 : 0.710
Precision : 0.866
------------------------------
Fold #2
Score ROC AUC : 0.923
Score kappa : 0.618
Score F1 : 0.705
Precision : 0.864
------------------------------
Fold #3
Score ROC AUC : 0.928
Score kappa : 0.637
Score F1 : 0.721
Precision : 0.870
------------------------------
Fold #4
Score ROC AUC : 0.926
Score kappa : 0.625
Score F1 : 0.709
Precision : 0.869
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.925
Moyenne Kappa dans les plis : 0.626
Moyenne F1 dans les plis : 0.711
Moyenne de precision dans les plis : 0.867
```



Lorsque nous avons ajouté des fonctionnalités catégoriques, nous avons eu une autre amélioration importante dans chaque métrique. Maintenant, le score moyen entre les différents plis a grimpé à 0,9. Les scores f1 et kappa ont également été améliorées de 0,15.

A partir du graphique de séparation nous pouvons voir que les classes positives forment maintenant des régions denses.
En outre, la courbe de seuil ne présente plus de point visible proche de 0,3. Au lieu de cela, il a une forme concave, ce qui signifie que la différence entre 0,3-0,4 est maintenant moins visible et que les probabilités de classe sont mieux séparées.

VII.2 Arbre de décision

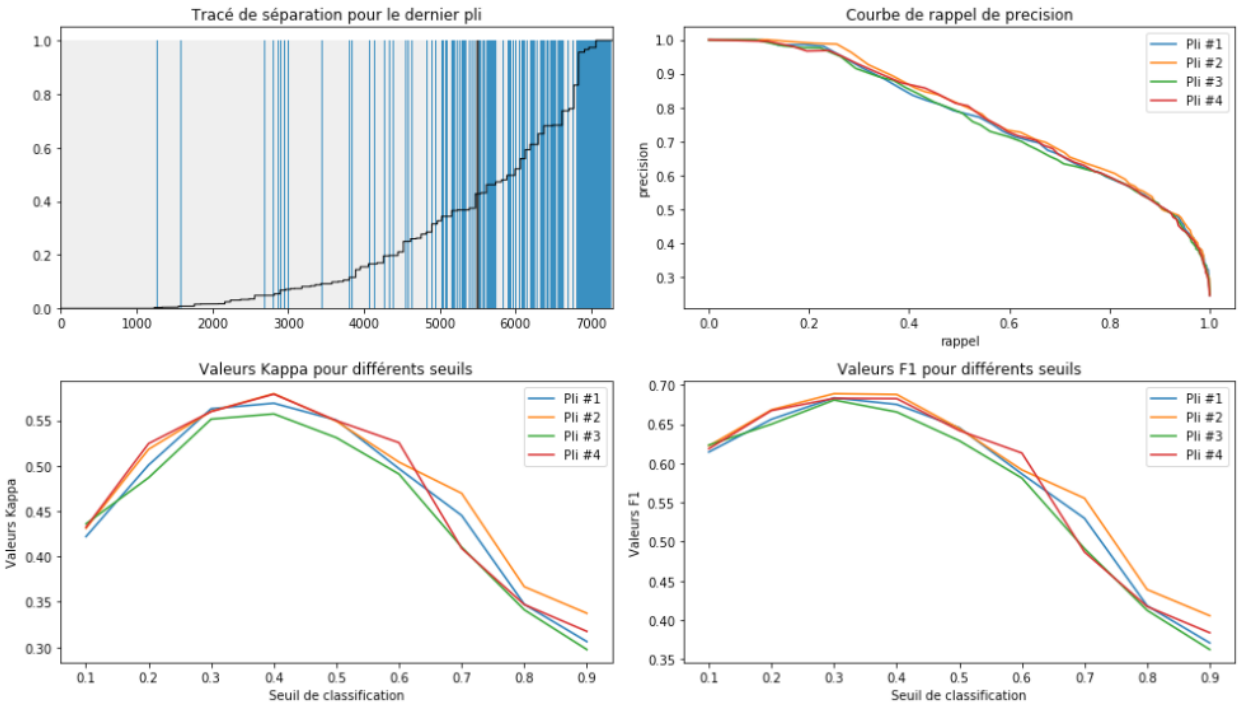
Un arbre de décision est une représentation visuelle d'un algorithme de classification de données suivant différents critères qu'on appellera décisions (ou noeuds).

Les modèles linéaires ne constituent pas souvent le meilleur choix car ils ne sont pas en mesure de capturer toute la complexité des données. L'arbre de décision est un autre algorithme simple capable de capturer des propriétés non linéaires à partir des données.

VII.2.1 Une des caractéristiques encodées à chaud

```
Fold #1
Score ROC AUC : 0.904
Score kappa : 0.564
Score F1 : 0.657
Precision : 0.851
------------------------------
Fold #2
Score ROC AUC : 0.894
Score kappa : 0.546
Score F1 : 0.644
Precision : 0.845
------------------------------
Fold #3
Score ROC AUC : 0.896
Score kappa : 0.537
Score F1 : 0.635
Precision : 0.842
------------------------------
Fold #4
Score ROC AUC : 0.904
Score kappa : 0.565
Score F1 : 0.660
Precision : 0.849
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.900
Moyenne Kappa dans les plis : 0.553
Moyenne F1 dans les plis : 0.649
Moyenne de precision dans les plis : 0.847
```



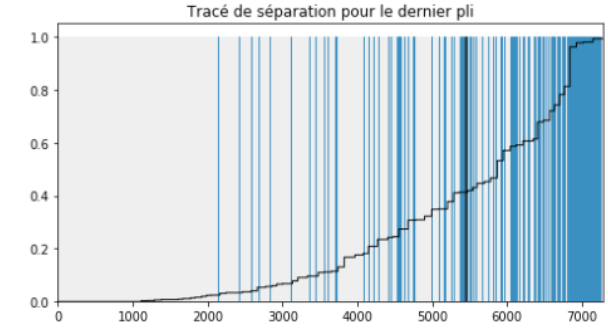
A partir de l'intrigue, nous pouvons constater qu'il existe presque maintenant une différence entre les scores par rapport au modèle linéaire.

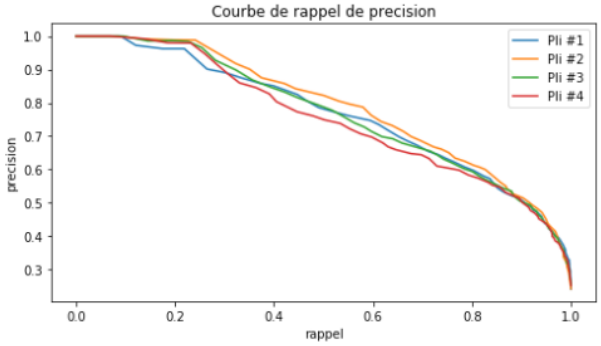
VII.2.4 Catégories étiquetées avec des entiers

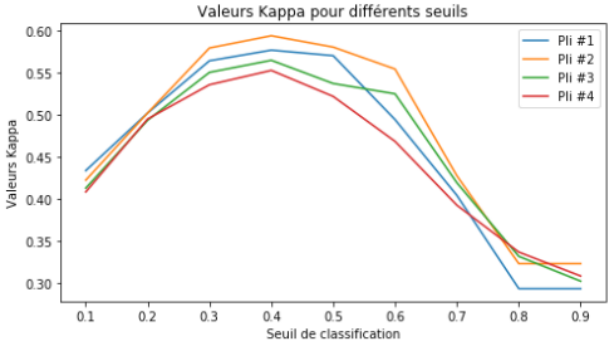
Avec les arbres de décision, nous pouvons essayer de réduire le nombre de fonctionnalités. Au lieu d'utiliser une fonctionnalité encodée à chaud, nous pouvons utiliser chaque colonne catégorique avec des valeurs catégorielles remplacées par des valeurs entières.

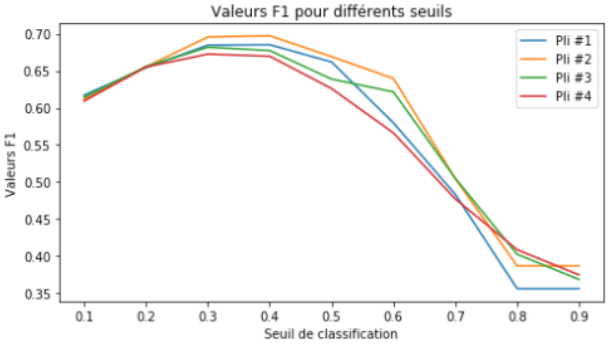
```
Fold #1
Score ROC AUC : 0.902
Score kappa : 0.570
Score F1 : 0.662
Precision : 0.853
------------------------------
Fold #2
Score ROC AUC : 0.907
Score kappa : 0.580
Score F1 : 0.669
Precision : 0.857
------------------------------
Fold #3
Score ROC AUC : 0.892
Score kappa : 0.537
Score F1 : 0.639
Precision : 0.838
------------------------------
Fold #4
Score ROC AUC : 0.888
Score kappa : 0.521
Score F1 : 0.626
Precision : 0.834
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.897
Moyenne Kappa dans les plis : 0.552
Moyenne F1 dans les plis : 0.649
Moyenne de precision dans les plis : 0.846
```









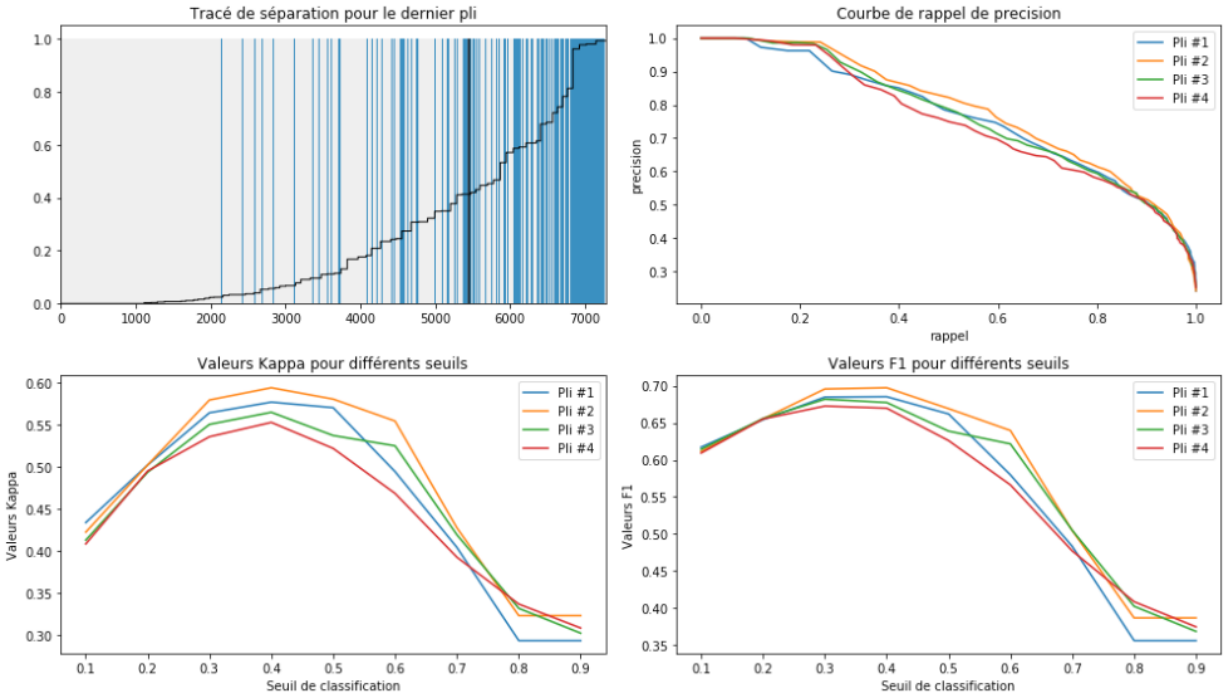
Le nombre de caractéristique est réduit d'un ordre de grandeur (de 100+ à 10+), on peut dire que notre modèle est encore amélioré. Cependant, le score n'a pas changé.
Notons que l'importance de la fonction ***numerical feature*** est assez similaire a celle du modèle linéaire. En outre, les fonctionnalités telles que ***race*** et ***native country*** n'ont pas d'importance, car elles n'ont logiquement aucun impact. La variable ***éducation*** était probablement en corrélation avec la variable ***eduction_num*** et son importance pourrait être réduite.
L'impact des variables ***occupation*** et ***martial_status*** sont faible sur la prédiction finale. Nous allons changer notre modèle pour quelque chose de mieux'

Nous avons essayé avec les **Arbres boostés a gradient** en utilisant des valeurs par défaut pour avoir de meilleur résultat.

```
Fold #1
Score ROC AUC : 0.929
Score kappa : 0.631
Score F1 : 0.715
Precision : 0.869
------------------------------
Fold #2
Score ROC AUC : 0.923
Score kappa : 0.633
Score F1 : 0.719
Precision : 0.868
------------------------------
Fold #3
Score ROC AUC : 0.917
Score kappa : 0.606
Score F1 : 0.696
Precision : 0.860
------------------------------
Fold #4
Score ROC AUC : 0.930
Score kappa : 0.620
Score F1 : 0.704
Precision : 0.868
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.925
Moyenne Kappa dans les plis : 0.622
Moyenne F1 dans les plis : 0.709
Moyenne de precision dans les plis : 0.866
```

Avec la configuration par défaut, GBT a réussi à améliorer les scores obtenus par l'arbre de décision et le modèle linéaire.



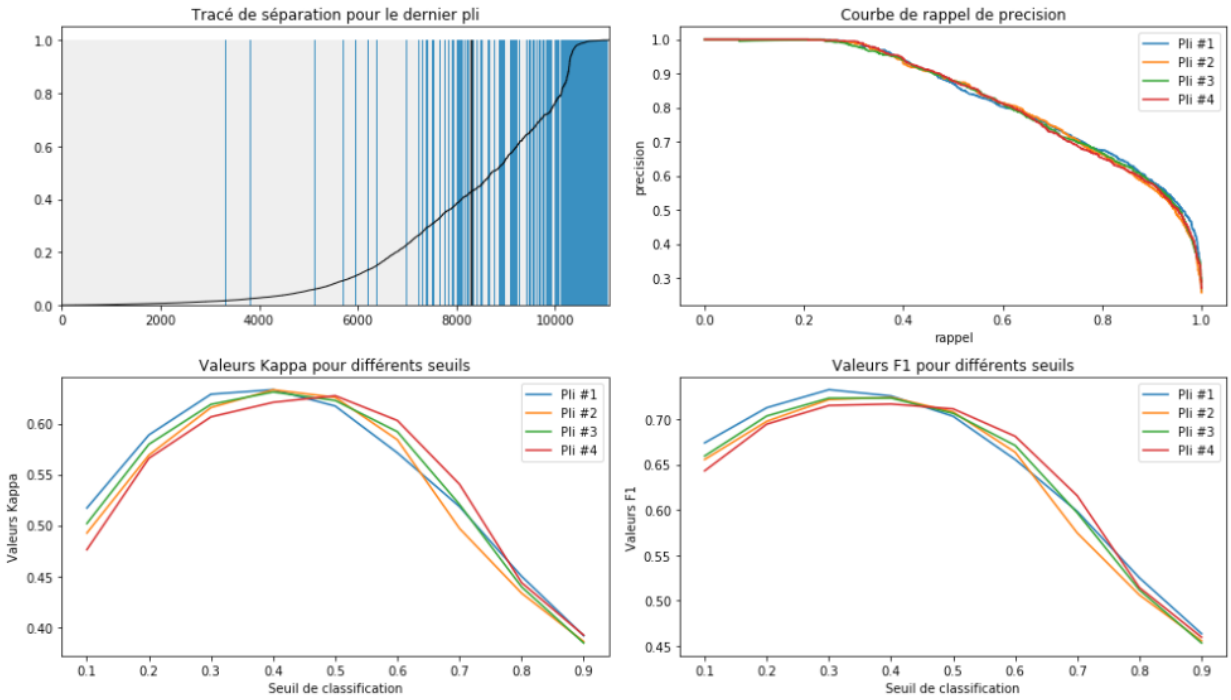
Le graphique sur le graphe de séparation a changé de forme, ce qui montre que le modèle peut redire des classes positives avec une confiance plus grande et une courbe de rappel de précision montre la qualité améliorée des prédictions par rapport aux méthodes précédentes.

Lorsque nous avons fait une vue sur l'entraînement et la validation.

VII.2.4 Amélioration de la méthode Arbre dégradés

```
Fold #1
Score ROC AUC : 0.928
Score kappa : 0.617
Score F1 : 0.703
Precision : 0.865
------------------------------
Fold #2
Score ROC AUC : 0.924
Score kappa : 0.626
Score F1 : 0.708
Precision : 0.870
------------------------------
Fold #3
Score ROC AUC : 0.928
Score kappa : 0.623
Score F1 : 0.707
Precision : 0.869
------------------------------
Fold #4
Score ROC AUC : 0.928
Score kappa : 0.627
Score F1 : 0.712
Precision : 0.869
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.927
Moyenne Kappa dans les plis : 0.623
Moyenne F1 dans les plis : 0.707
Moyenne de precision dans les plis : 0.868
```



En faisant des suppressions sur les caractéristiques avec des valeurs faibles, aucune différence n'était constatée dans la prédiction des scores métriques. Il montre également à quel point l’importance des fonctionnalités fournies par les méthodes par défaut peuvent être peu fiable.

VIII. Évaluation de l’ensemble de données de test

```
Score ROC AUC : 0.926
Score kappa : 0.627
Score F1 : 0.709
Precision : 0.871
```

La précision de la prévision sur l’ensemble de test est très proche de celle obtenue lors de la validation croisée moyenne, ce qui signifie que la conception précédente basée sur les données d’apprentissage est correct.

IX. Conclusion

En phase d’analyse on peut dire que les décisions relatives aux modèles que l’on peut voir à l’aide des valeurs simplistes sont intuitives et permettent d'établir de manière succincte les décisions prises par le modèle. Avec un pré-traitement assez simple est des configurations par défaut pour les arbres à gradient de densité, il était possible d’obtenir un score d’AUC ROC élevé sur les données de test ~0.926.

Références

1. http://cedric.cnam.fr/vertigo/Cours/ml2/coursArbresDecision.html
2. WikiStat : https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-rlogit.pdf
3. Régression logistique pour réponse binaires et multinomiales, https://www.xlstat.com/fr/solutions/fonctionnalites/regression-logistique-pour-reponse-binaires-et-multinomiales-logit-probit
4. Logistic Regression: A Simplified Approach Using Python, https://towardsdatascience.com/logistic-regression-a-simplified-approach-using-python-c4bc81a87c31
5. Machine learning with Python, https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_tutorial.pdf
6. Machine learning avec scikit-learn, http://eric.univ-lyon2.fr/~ricco/cours/slides/PJ%20-%20machine%20learning%20avec%20scikit-learn.pdf
7. Classification in Python with Scikit-Learn and Pandas, https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/
8. Statistics and Machine Learning in Python, Edouard Duchesnay, Tommy Löfstedt