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

PHILIPPE Jean Mith & Soumana Hamadou Abdourahmane
PROF : MAI ANH BUI THI
Juin 2019

Introduction

Dans le cadre du cour 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.

Description et résumée des données

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

Download information

Datasource processing	
Computation time	234 ms
Allocated memory	1953 KB

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

Fig1 : Description des données

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 :** Etat 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**

Methode 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.

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 preparation des donnees indiques 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.

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.

	age	workclass	final_weight	education	education_num	marital_status	occupation	relationship
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife

Fig2 : Quelque ligne de notre ensemble de données d'entraînement

elationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income_class
Wife	White	Female	0	0	38	United-States	<=50K
Husband	White	Male	0	0	40	United-States	>50K
Unmarried	White	Female	0	0	40	United-States	<=50K
Own-child	White	Male	0	0	20	United-States	<=50K
Wife	White	Female	15024	0	40	United-States	>50K

Fig3 : Quelque ligne de notre ensemble de données d'entraînement (suite)

	age	workclass	final_weight	education	education_num	marital_status	occupation	relationship
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband
4	18	?	103497	Some-college	10	Never-married	?	Own-child

Fig4 : Quelque ligne de notre ensemble de données de test

ge	workclass	final_weight	education	education_num	marital_status	occupation	relationship	ra
39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	Wh
64	?	321403	HS-grad	9	Widowed	?	Other-relative	Bla
38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Wh
44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asia Pa Islanc
35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Wh

Fig5 : Quelque ligne de notre ensemble de données de test (suite)

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 déduplique 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.

	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0

Fig6 : Ensemble de données d'entraînement après suppression de la variable final_weight

	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	
32557	40	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	
32558	58	Private	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	
32559	22	Private	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	
32560	52	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	

Fig7 : Ensemble de données d'entraînement après suppression de la variable final_weight (suite)

Verification des donnees

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.

```
: 0        False
  1        False
  2        False
  3        False
  4        False
  5        False
  6        False
           ...
  48836    False
  48837    False
  48838    False
  48839    False
  48840    False
  48841    False
  48842    False
  Length: 48843, dtype: bool
```

```
: df.isnull().values.any()
```

```
: False
```

Fig8 : Verification de notre ensemble de données d'entraînement

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 problemes 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 predire.

ationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income_class
48842	48842	48842	48842.000000	48842.000000	48842.000000	48842	48842
6	5	2	NaN	NaN	NaN	42	4
Husband	White	Male	NaN	NaN	NaN	United-States	<=50K
19716	41762	32650	NaN	NaN	NaN	43832	24720
NaN	NaN	NaN	1079.067626	87.502314	40.422382	NaN	NaN
NaN	NaN	NaN	7452.019058	403.004552	12.391444	NaN	NaN
NaN	NaN	NaN	0.000000	0.000000	1.000000	NaN	NaN
NaN	NaN	NaN	0.000000	0.000000	40.000000	NaN	NaN
NaN	NaN	NaN	0.000000	0.000000	40.000000	NaN	NaN
NaN	NaN	NaN	0.000000	0.000000	45.000000	NaN	NaN
NaN	NaN	NaN	99999.000000	4356.000000	99.000000	NaN	NaN

fig9:***Ensemble de données d'entraînement et de test combine***
Pour continuer avec notre travaille nous avons combiné ensemble de données d'entraînement et l'ensemble de données de test afin de généraliser les problemes 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 predire.

La colonne income_class a quatre classes uniques, mais nous voulions avoir seulement deux classes.

```
Train shape: (32561, 14)
Test shape: (16281, 14)
```

```
train_data.shape[0] + test_data.shape[0]
```

```
48842
```

Fig10 : Ensemble de données d'entraînement après suppression de la variable final_weight

```
less     37155
less.    12435
more     11687
more.     3846
class        1
Name: income_class, dtype: int64
```

```
test_data.income_class.value_counts(dropna=False)
```

```
less.    12435
more.     3846
Name: income_class, dtype: int64
```

```
train_data.income_class.value_counts(dropna=False)
```

```
less     37155
more     11687
class        1
Name: income_class, dtype: int64
```

Fig11 : Ensemble de données d'entraînement après suppression de la variable final_weight

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.

Categorie supplementaire

```
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
```

Fig12 : Ensemble de données d'entraînement après suppression de la variable final_weight

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.

Espace dans les valeurs

```
Index(['?', 'Federal-gov', 'Local-gov', 'Never-worked', 'Private',
       'Self-emp-inc', 'Self-emp-not-inc', 'State-gov', 'Without-pay'],
      dtype='object')
```

Fig613 : Ensemble de données d'entraînement après suppression de la variable final_weight

Deux colonnes ont des valeur maximale avec tous les NaN (valeur non trouve) en elles.

Premier etape 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 (**.**) a 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%
```

Fig14 : Ensemble de données d'entraînement après suppression de la variable final_weight

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

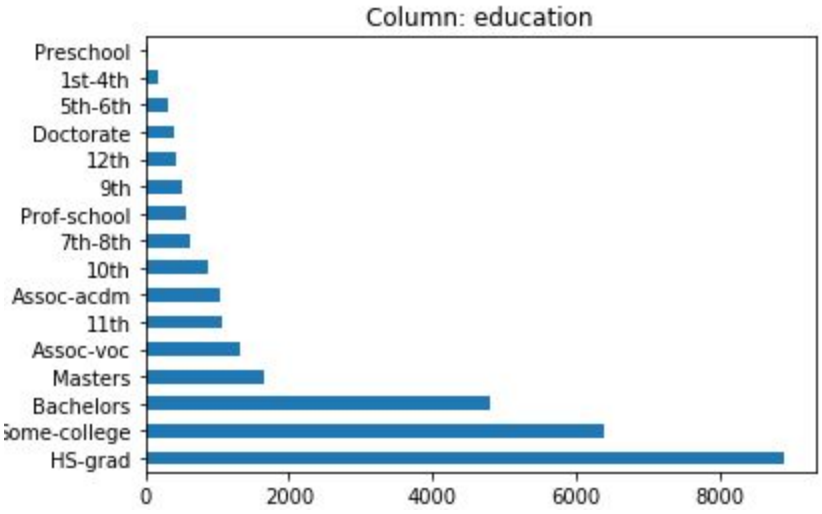
```
Pourcentage des classes positives dans les données d'entraînement: 24.78%
```

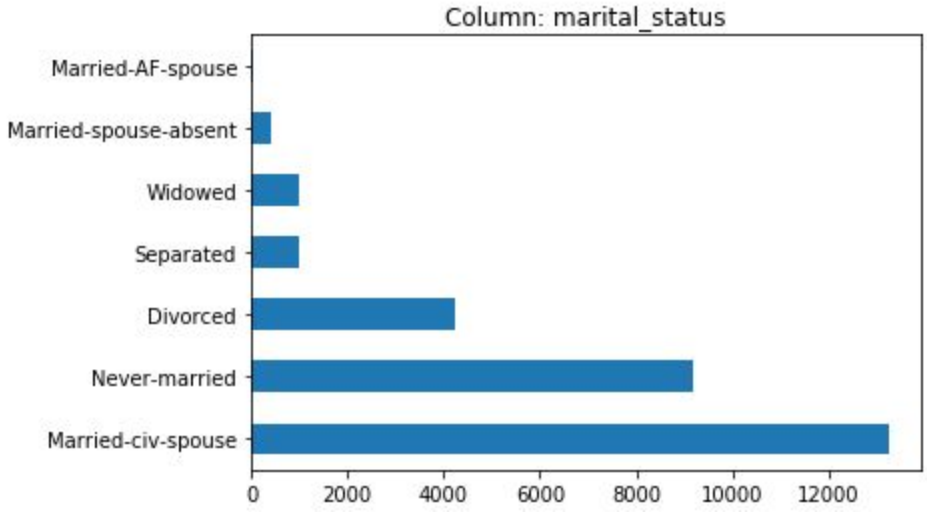
Nous avons faire la déduplication par ensemble de données, mais il y a des doublons entre l'ensemble de données de formation et l'ensemble de données de test. Avec des doublons entres les ensembles de données, nous pourrions obtenir des résultats trop confiants.

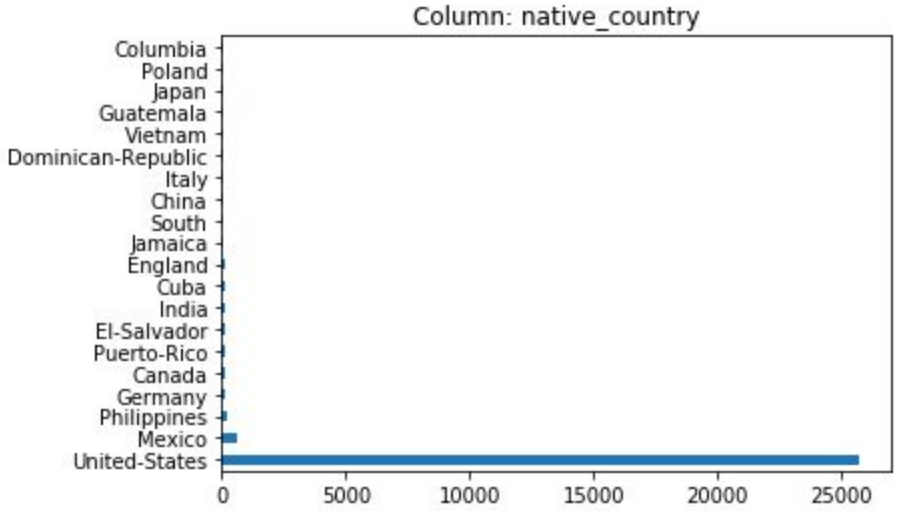
Valeur dominante dans la colonne de gain en capital

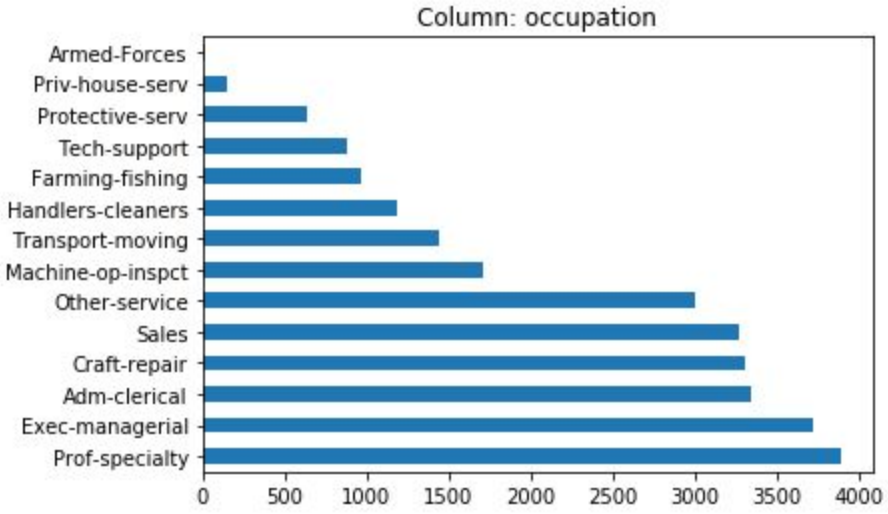
Quelques colonnes ont des valeurs assez dominates, ce qui signifie qu'elles occupent un pourcentage assez important des données (**0.9072724773164696**).

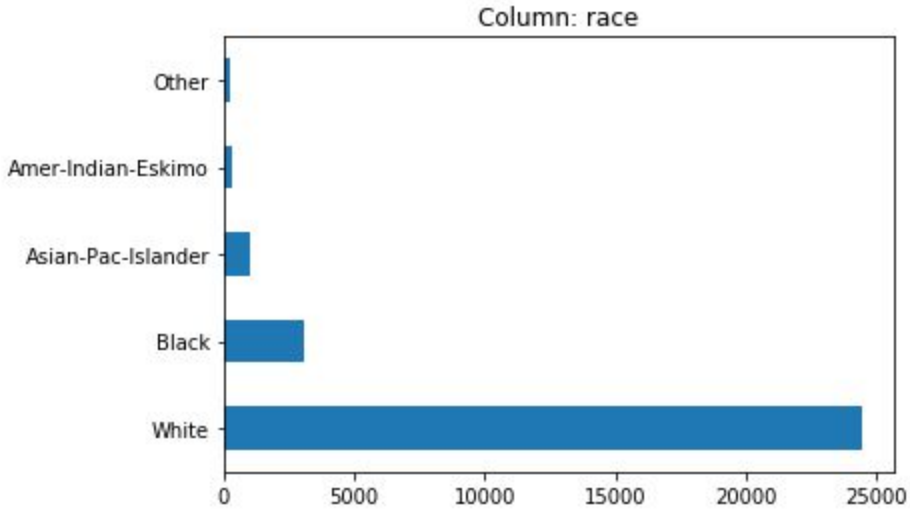
Verification des comptes par categories

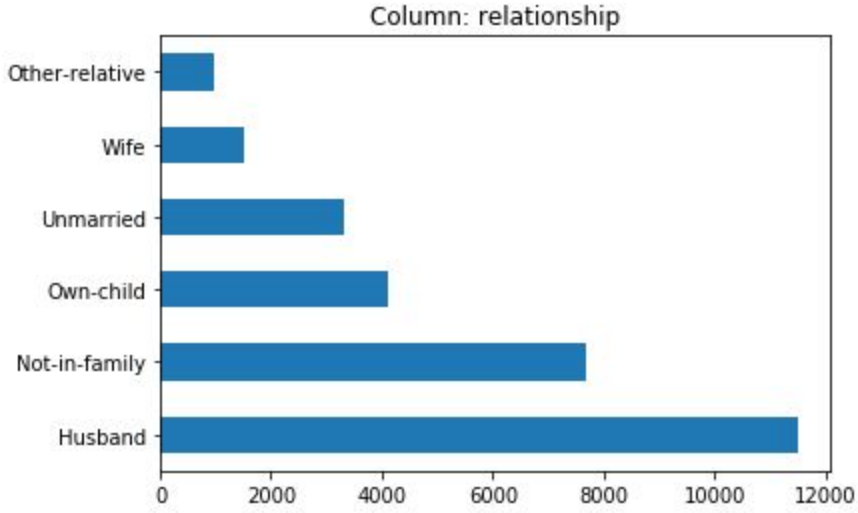


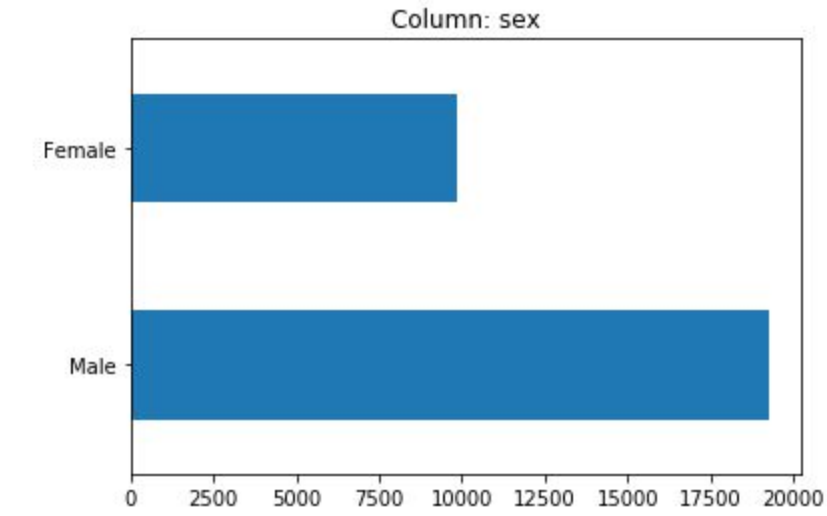


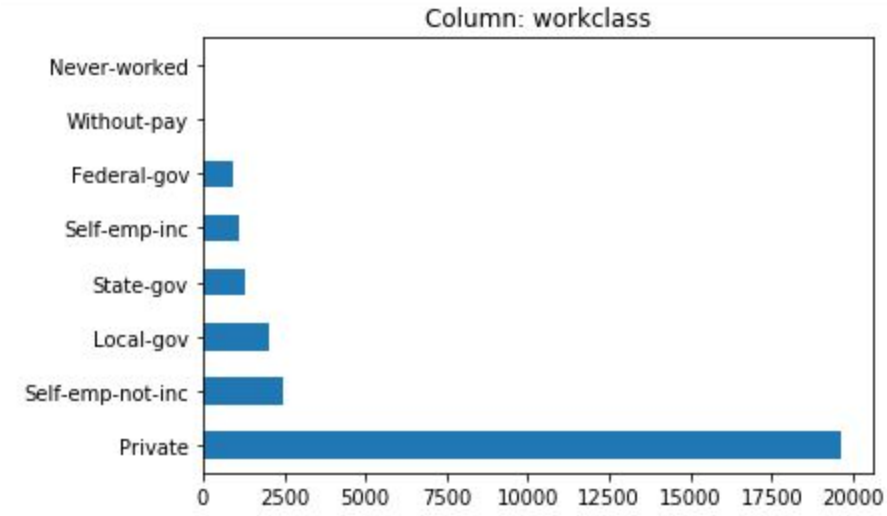



Column: occupation
Armed-Forces
Priv-house-serv
Protective-serv
Tech-support
Farming-fishing
Handlers-cleaners
Transport-moving
Machine-op-inspct
Other-service
Sales
Craft-repair
Adm-clerical
Exec-managerial
Prof-specialty
0
500
1000
1500
2000
2500
3000
3500
4000


Column: race
Other
Amer-Indian-Eskimo
Asian-Pac-Islander
Black
White
0
5000
10000
15000
20000
25000


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

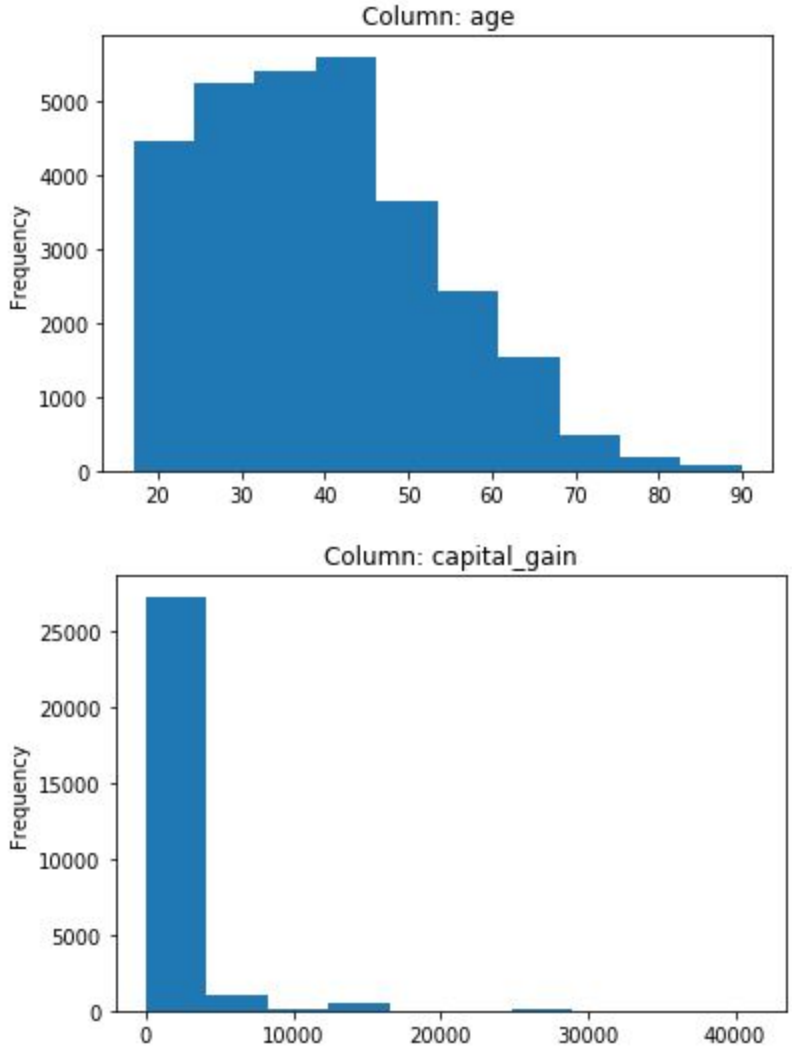


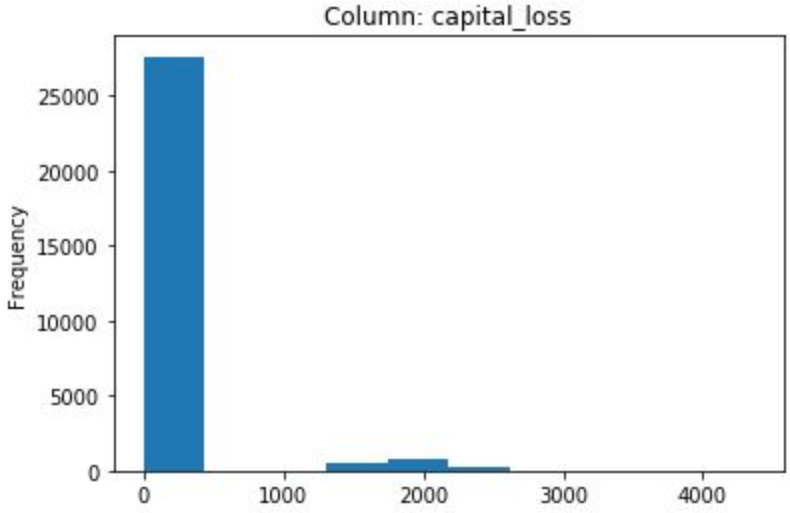


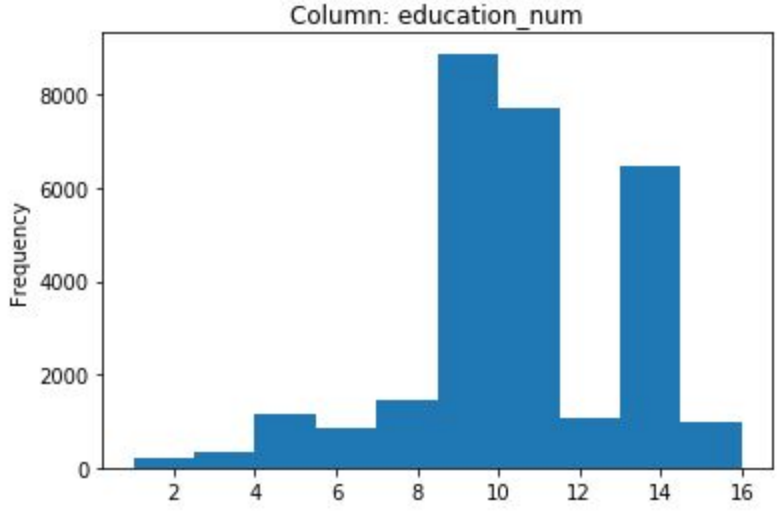
Observations:

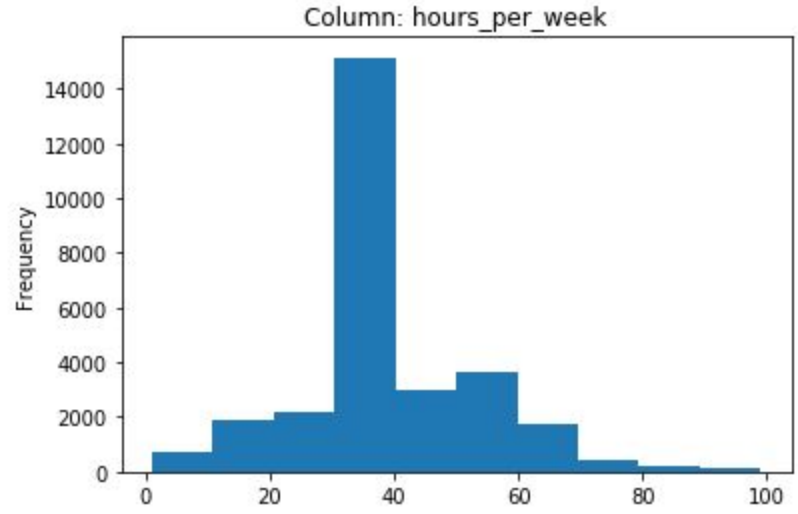
1. La variable ***race*** a un grand nombre de valeur ***white***
2. La variable ***workclass*** a un grand nombre de valeur ***Private***
3. La variable ***native_country*** a un grand nombre de valeur ***United-states***.
   Cette variable peut être ignorée ou remplacée par une variable binaire. Avec la valeur **Vrai** si la personne est des etats-unis sinon **Faux.**
4. La variable ***sex*** a un grand nombre de valeur **Male**
5. Pour la variable ***relationship*** la veleur Husband est dominant
6. Pour la variable ***education*** la valeur **Hs-grad** est dominant
7. Pour la variable ***occupation*** la valeur prof-speciality est dominant
8. Pour la variable ***marial_status*** la valeur **Married-civ-spouse** est dominant

Verification des histogrammes pour les colonnes numeriques







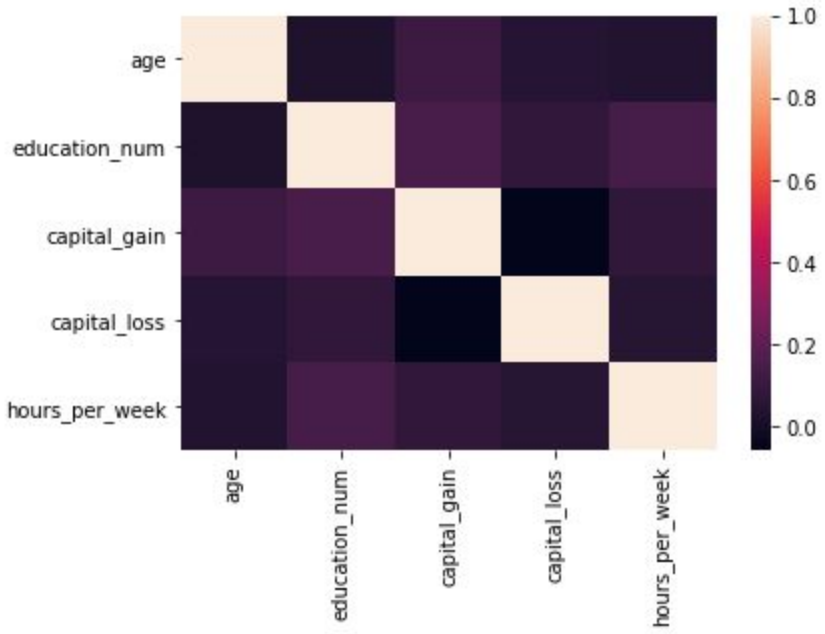


Observations:

1. La variable ***hours_per_week*** a un spike important ou la valeur est égale à 40, il s'agit du nombre d'heures de travail le plus courant de la semaine pour une personne moyenne.
2. Les variables ***capita_gain*** et ***capital_loss*** ont une grande spike ou la valeur est égale à zéro. Cela explique qu'une personne moyenne n'a pas de revenus supplémentaires. Il est logique de voir la distribution sans valeur zéro pour ces colonnes.
3. La variable ***education_num*** comporte deux spike. Le premier spike autour du nombre d'années qu'il faut pour terminer l'école pour la personne moyenne et le second pour l'université.

Correlation entre les colonnes numeriques

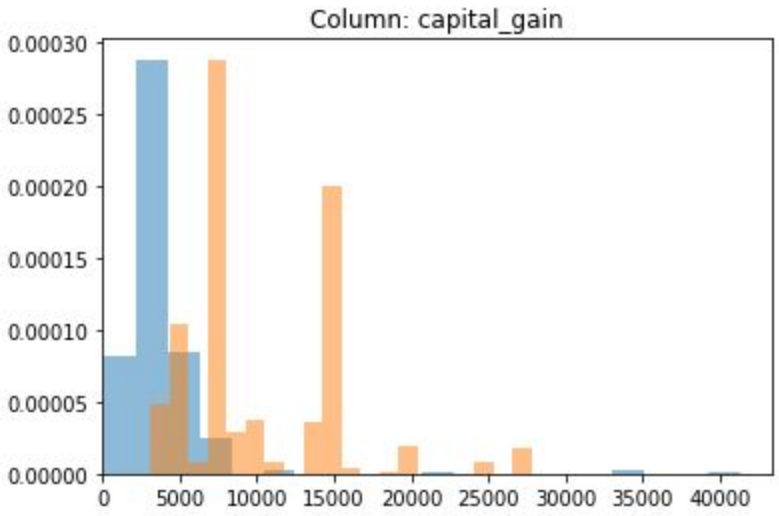
La base de données peut également avoir les attributs non pertinents.L'analyse de corrélation permet de savoir si deux attributs données sont liés.

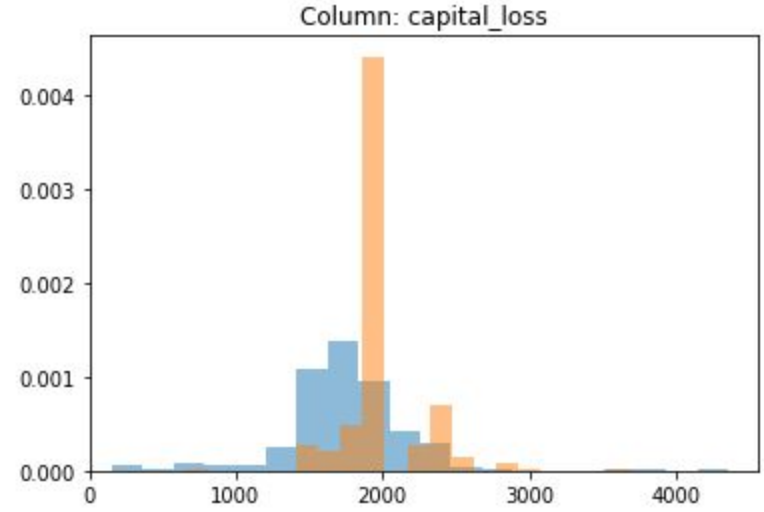


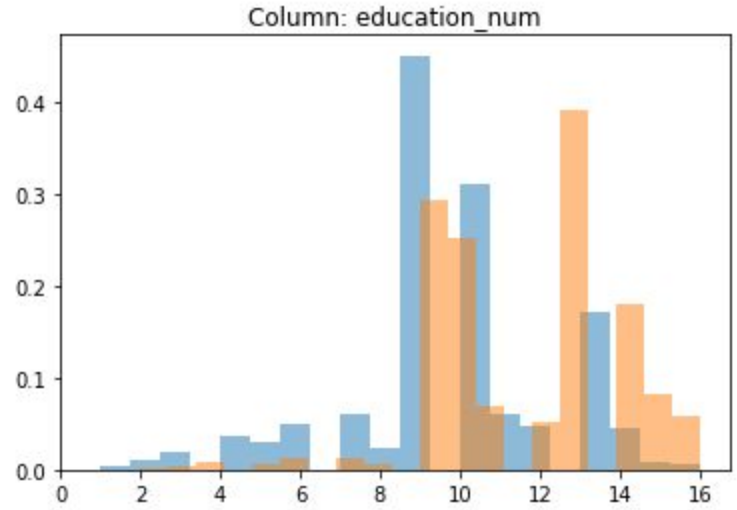
Les fonction numerique ont de tres faibles correlation.

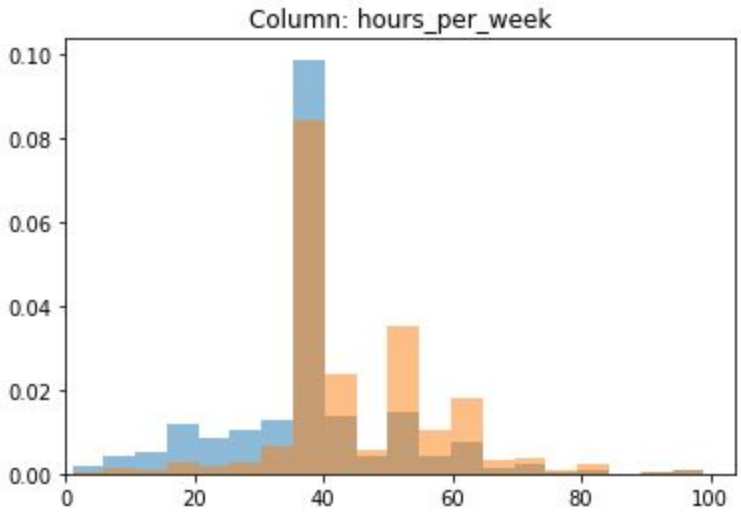
Separation des effets par categorie

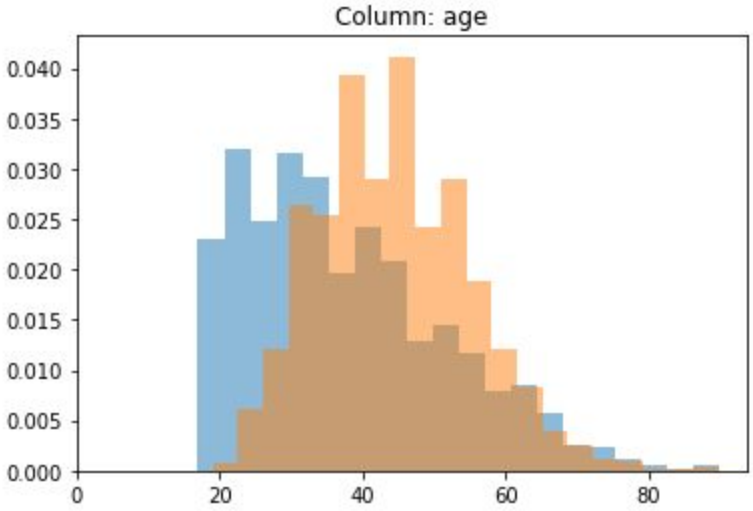
Variable continues


Column: capital_gain
0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0
5000
10000
15000
20000
25000
30000
35000
40000


Column: capital_loss
0.004
0.003
0.002
0.001
0.000
0
1000
2000
3000
4000


Column: education_num
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
10
12
14
16

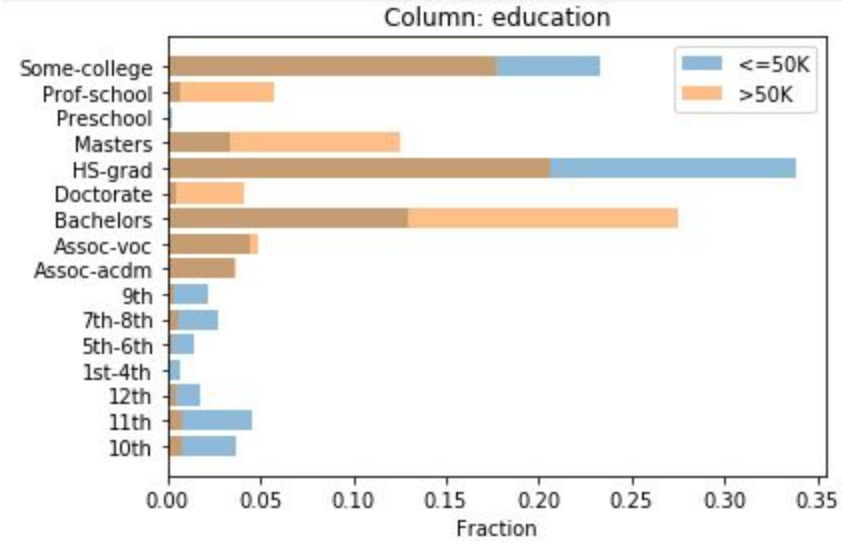


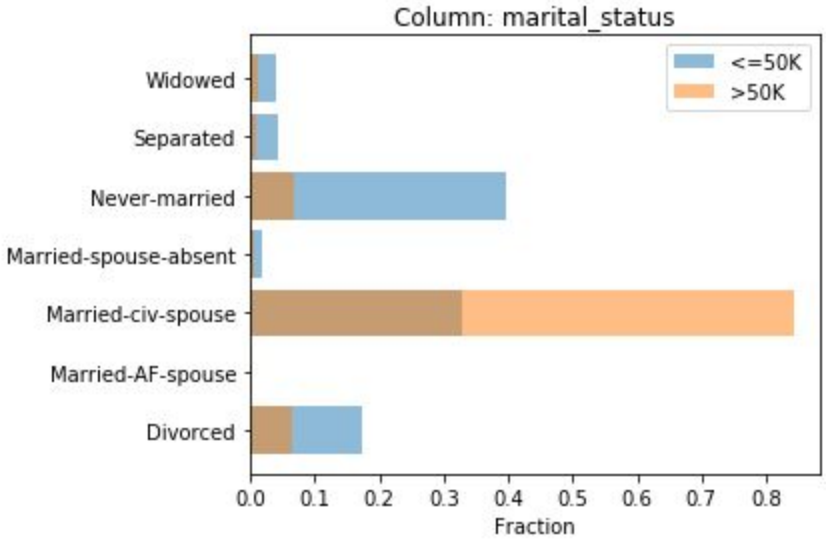


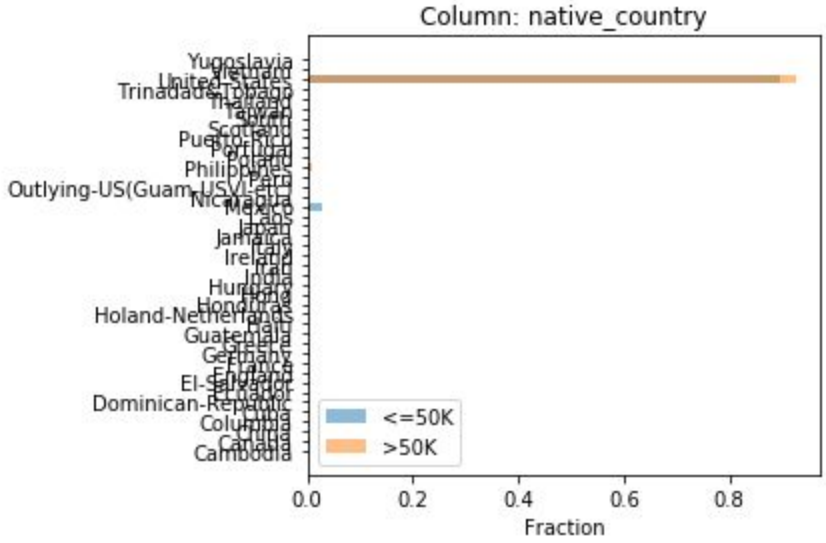
Observation

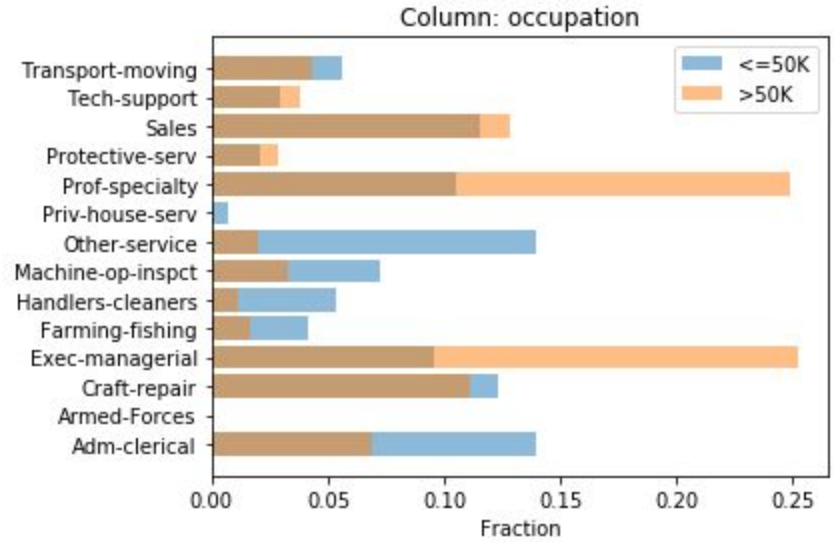
1. Plus la personne est âgée plus son revenu sera grande.
2. Plus le nombre d'années d'études est élevé, plus le revenu est élevé.
3. Plus une personne travaille, plus son revenu est élevé.

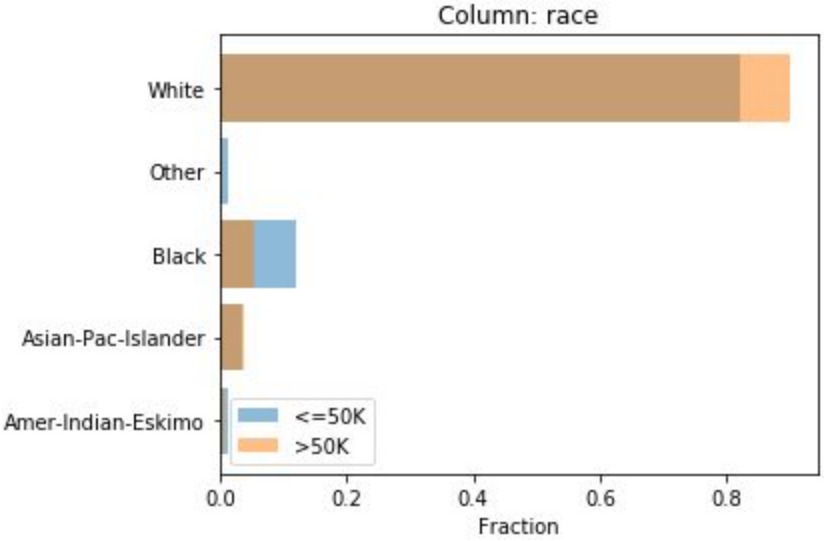
Variable discretes

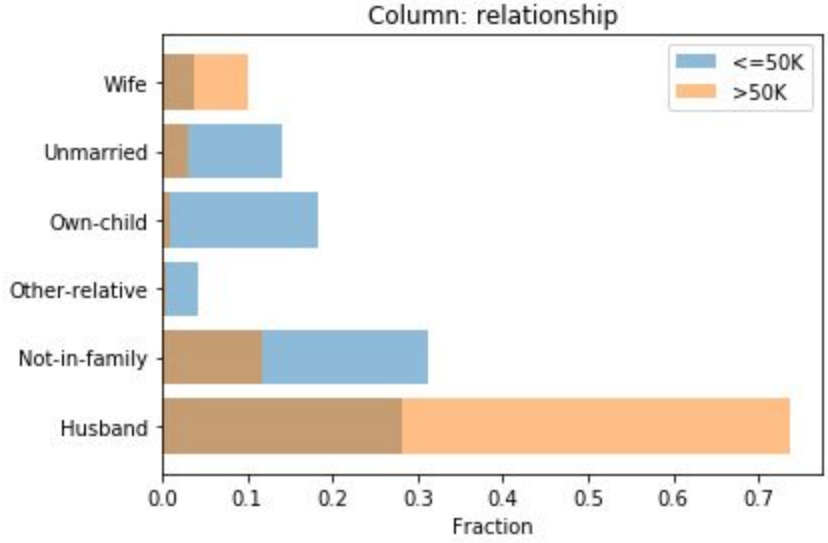

Column: education
Some-college
Prof-school
Preschool
Masters
HS-grad
Doctorate
Bachelors
Assoc-voc
Assoc-acdm
9th
7th-8th
5th-6th
1st-4th
12th
11th
10th
<=50K
>50K
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Fraction

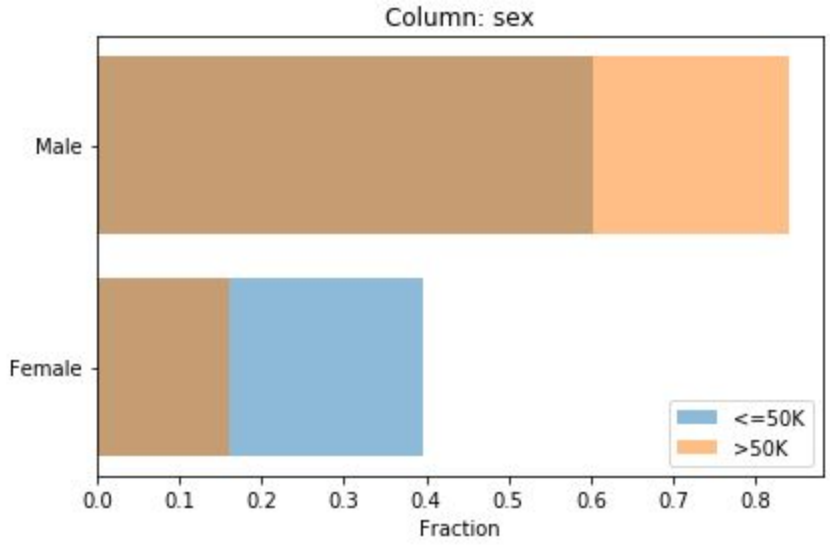

Column: marital_status
Widowed
Separated
Never-married
Married-spouse-absent
Married-civ-spouse
Married-AF-spouse
Divorced
<=50K
>50K
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Fraction

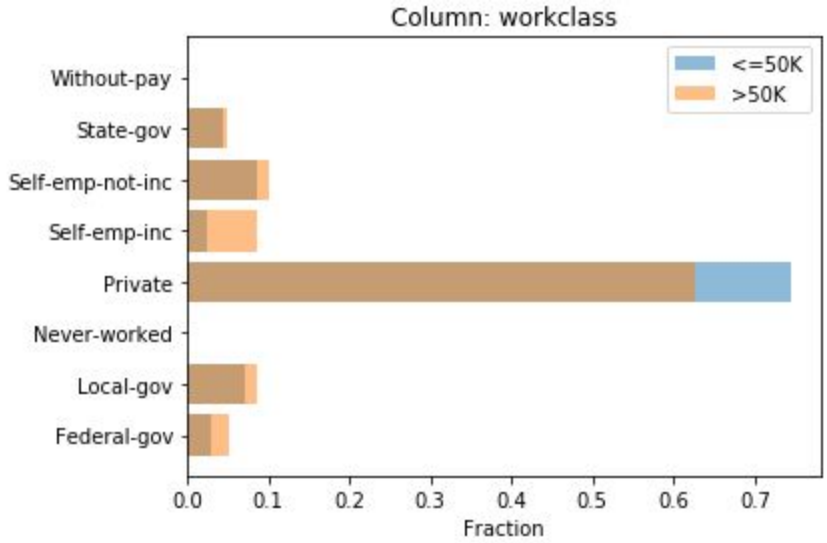

Column: native_country
Yugoslavia
Trinadad&Tobago
Outlying-US(Guam-USVI-etc)
Holand-Netherlands
Dominican-Republic
Cambodia
<=50K
>50K
0.0
0.2
0.4
0.6
0.8
Fraction


Column: occupation
Transport-moving
Tech-support
Sales
Protective-serv
Prof-specialty
Priv-house-serv
Other-service
Machine-op-inspct
Handlers-cleaners
Farming-fishing
Exec-managerial
Craft-repair
Armed-Forces
Adm-clerical
<=50K
>50K
0.00
0.05
0.10
0.15
0.20
0.25
Fraction


Column: race
White
Other
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
<=50K
>50K
0.0
0.2
0.4
0.6
0.8
Fraction


Column: relationship
Wife
Unmarried
Own-child
Other-relative
Not-in-family
Husband
<=50K
>50K
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Fraction





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.

Construction et evaluation des modeles

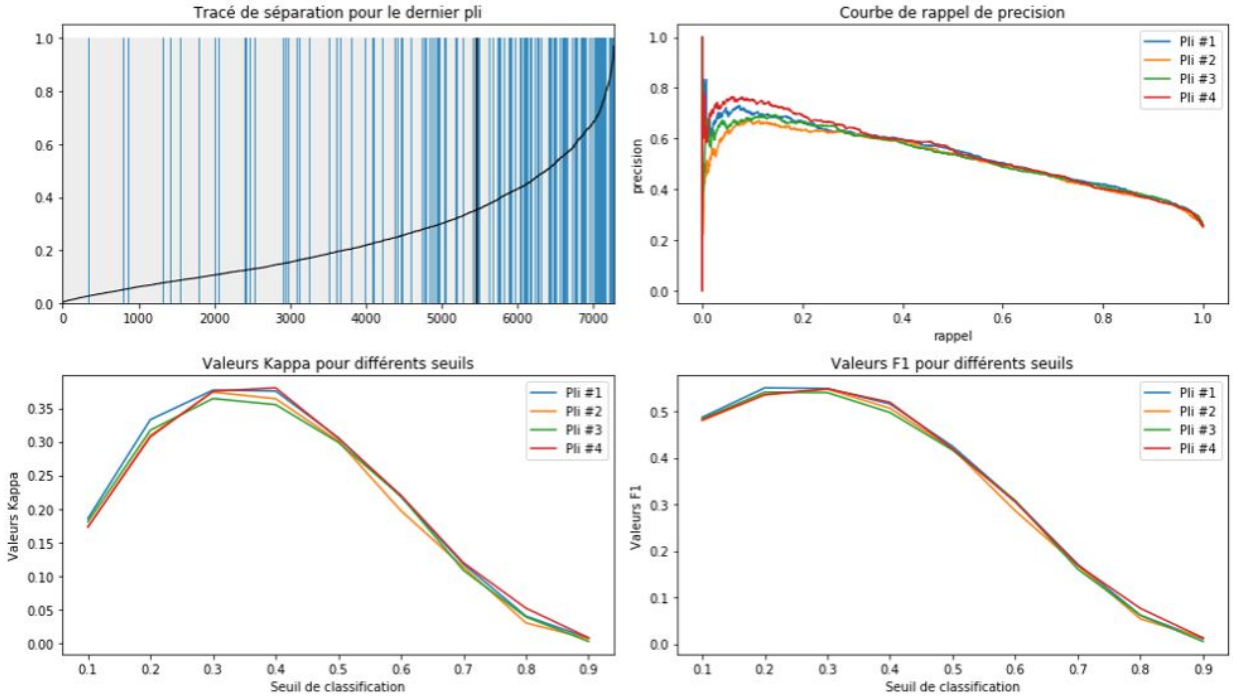
Modele lineaire

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

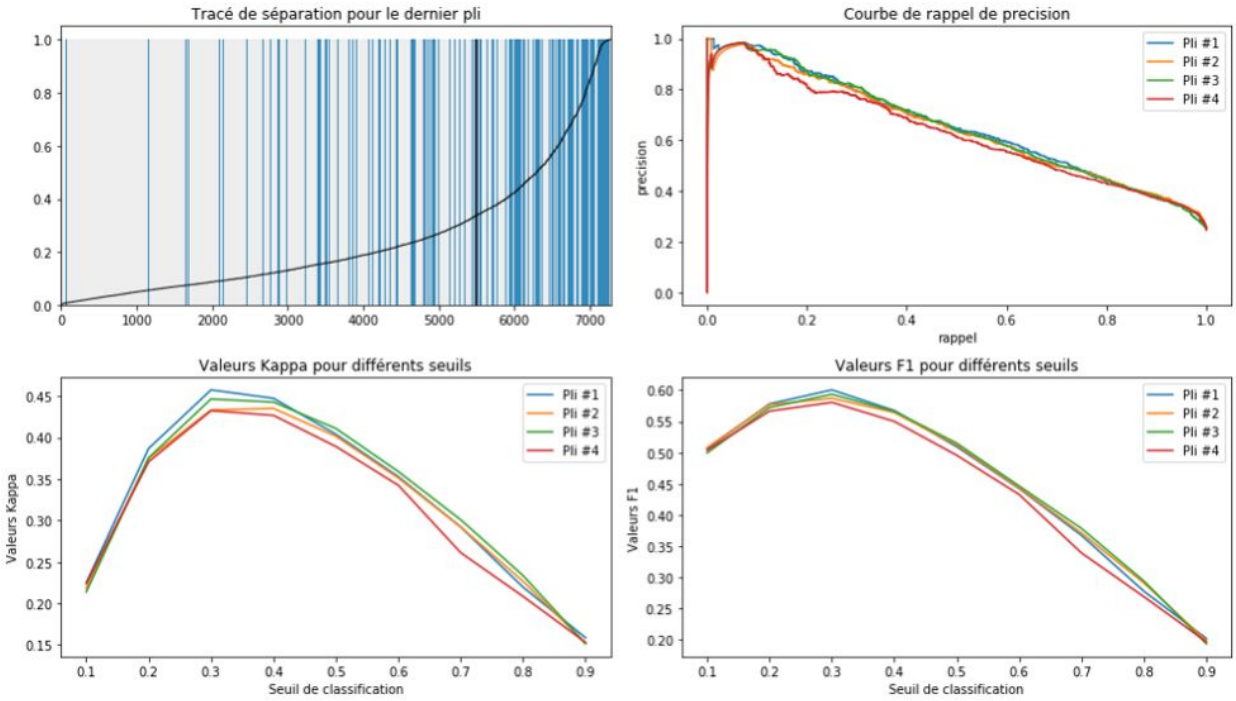
Seulement trois (3) variables donnent une prédiction avec un score ROC AUC presque égal a 0.8, C'est un bon début. Par contre, les scores kappa et f1 sont faibles.
Ce score dépend du seuil de 0.5. et c'est simplement un mauvais choix pour la séparation de classe.



Sur les deux graphiques inférieurs, nous pouvons voir que les scores f1 et kappa peuvent être améliorés si nous choisissons un seuil égal a 0.3.
Le résultat de l'analyse des correlation 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 ameliorer le resultat 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 a 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.

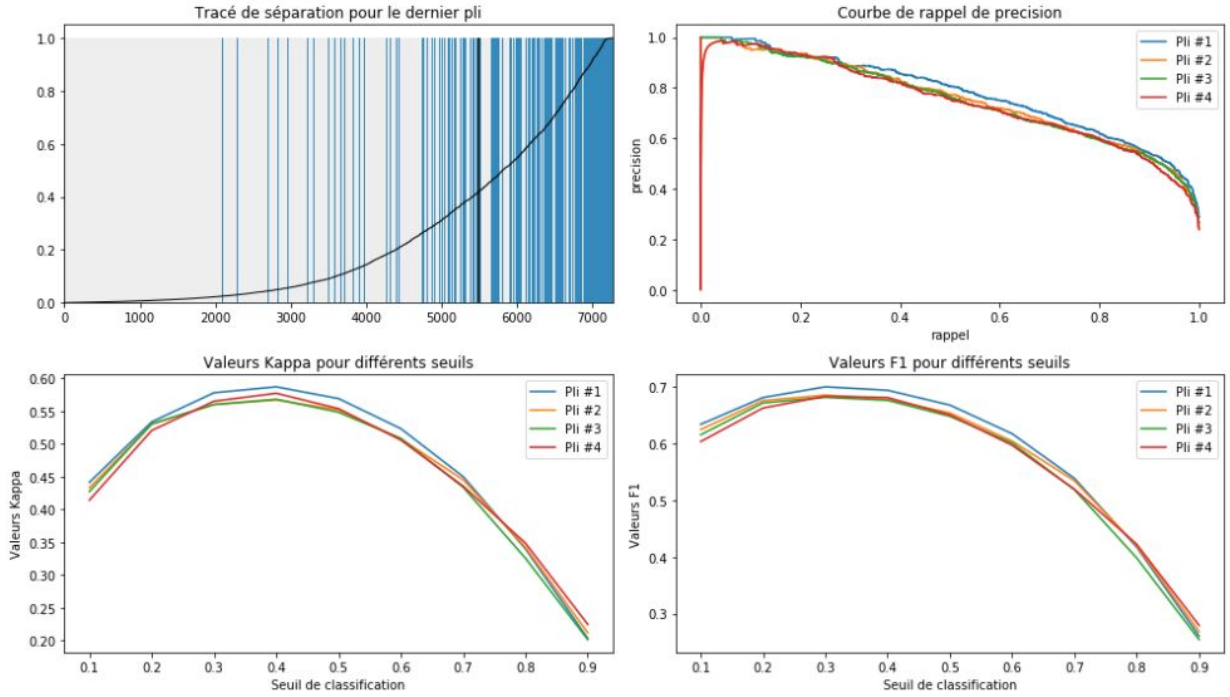
```
Fold #1
Score ROC AUC : 0.821
Score kappa : 0.416
Score F1 : 0.523
Precision : 0.814
------------------------------
Fold #2
Score ROC AUC : 0.826
Score kappa : 0.394
Score F1 : 0.503
Precision : 0.808
------------------------------
Fold #3
Score ROC AUC : 0.820
Score kappa : 0.387
Score F1 : 0.494
Precision : 0.807
------------------------------
Fold #4
Score ROC AUC : 0.832
Score kappa : 0.410
Score F1 : 0.515
Precision : 0.811
------------------------------

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 transformer en variables binaires les variables discretes et les ajouter au model.

```
Fold #1
Score ROC AUC : 0.922
Score kappa : 0.610
Score F1 : 0.697
Precision : 0.864
------------------------------
Fold #2
Score ROC AUC : 0.924
Score kappa : 0.637
Score F1 : 0.720
Precision : 0.871
------------------------------
Fold #3
Score ROC AUC : 0.927
Score kappa : 0.637
Score F1 : 0.720
Precision : 0.872
------------------------------
Fold #4
Score ROC AUC : 0.926
Score kappa : 0.614
Score F1 : 0.703
Precision : 0.862
------------------------------

Moyenne ROC AUC sur plusieurs plis : 0.925
Moyenne Kappa dans les plis : 0.625
Moyenne F1 dans les plis : 0.710
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 ete améliorées de 0,15.

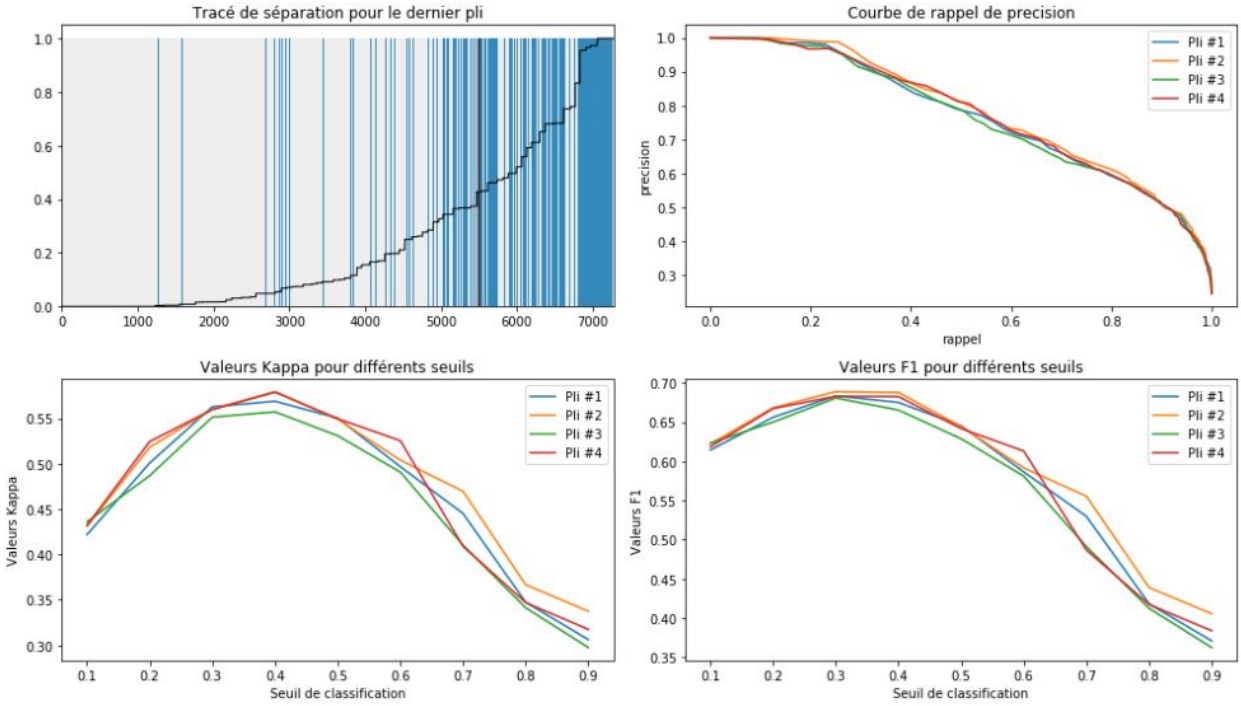
A partir du graphique de séparation nous pouvons voir que les classes positives forment maintenant des régions denses.
En outre, la courbes de seuil ne présente plus de point visible proche de 0,3. Au lieu de cela, il a une forme concave, ce qui signifie que la difference entre 0,3-0,4 est maintenant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

Arbre de decision

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 a partir des données.

Une des caracteristiques encodees a 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
```

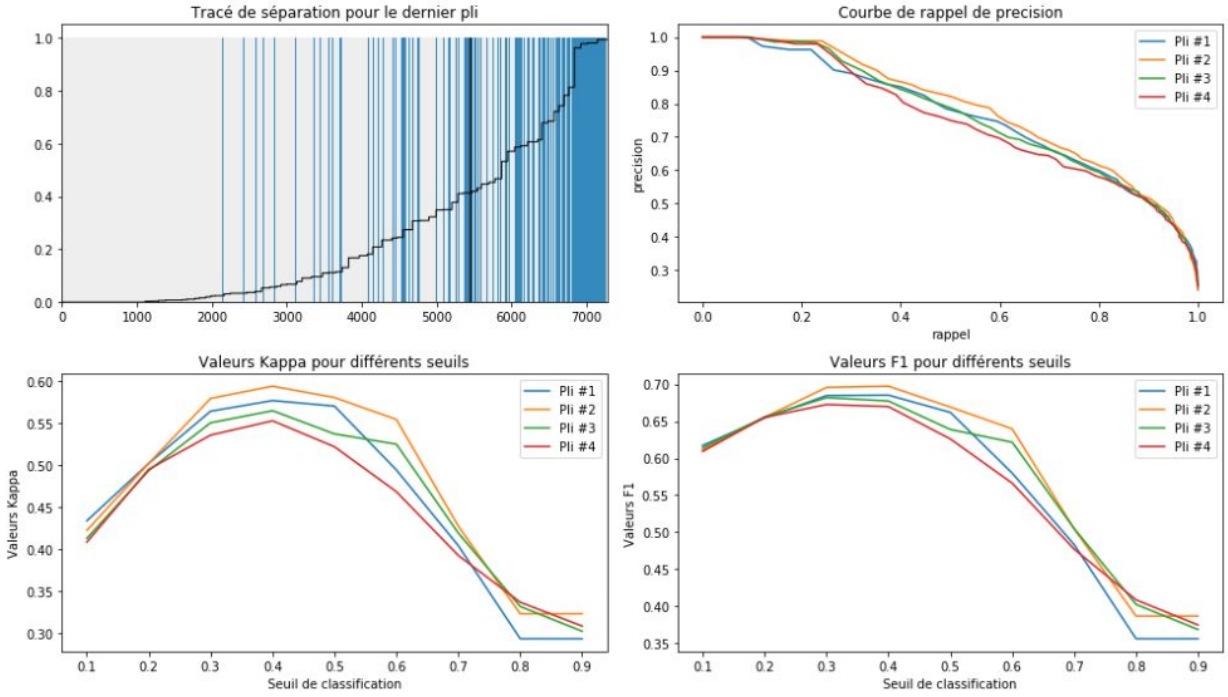


A partir de l'intrigue, nous pouvons constater qu'il existe presque maintenant une différence entre les scores par rapport au modèle linéaire.

Categories etiquetees 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
```



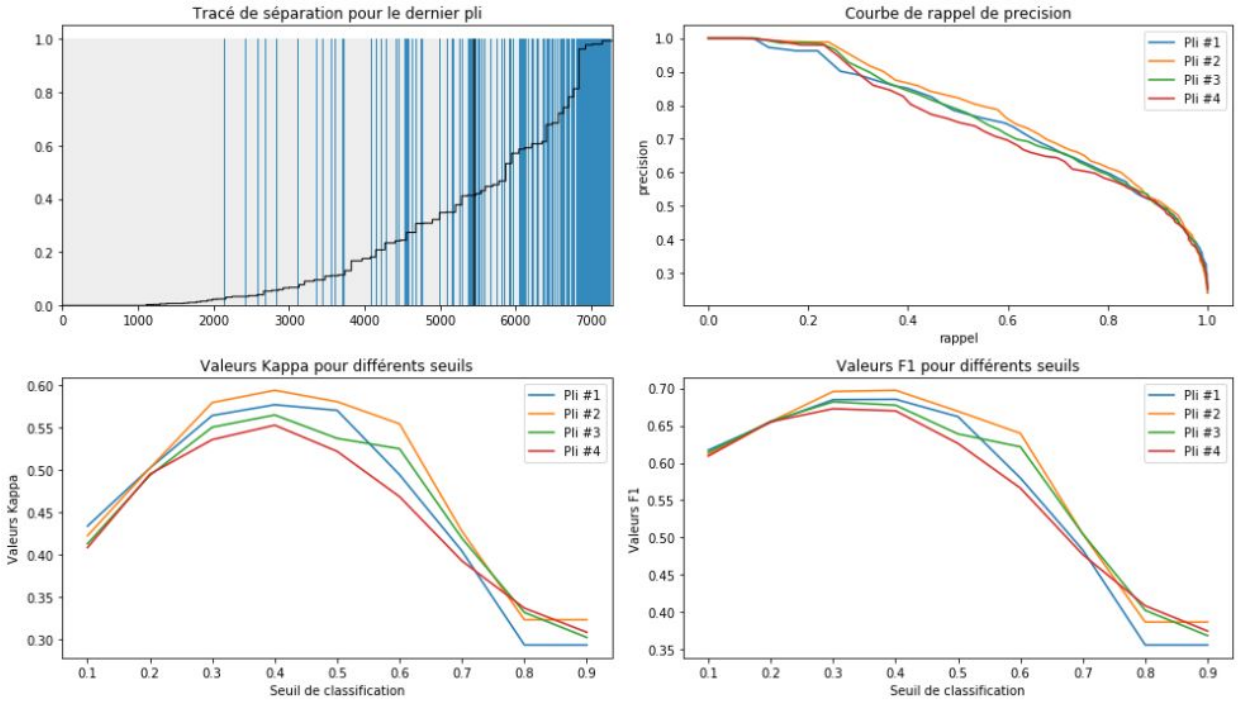
Le nombre de caractéristique est réduit d'un ordre de grandeur (de 100+ à 10+), on peut dire que notre modèle est encore amélioré. Cependant, le score n'a pas changé.

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 prediction 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.

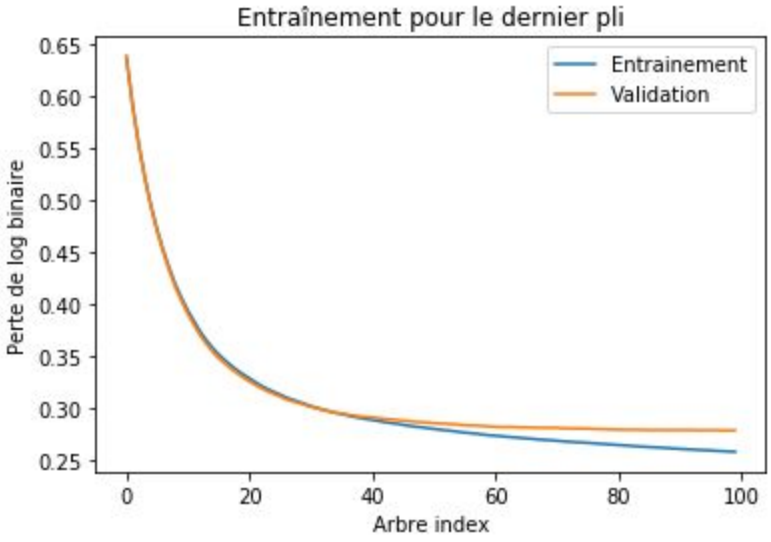
```
Fold #1
Score ROC AUC : 0.929
Score kappa : 0.631
Score F1 : 0.715
Precision : 0.869
------------------------------
Fold #2
Score ROC AUC : 0.923
Score kappa : 0.633
Score F1 : 0.719
Precision : 0.868
------------------------------
Fold #3
Score ROC AUC : 0.917
Score kappa : 0.606
Score F1 : 0.696
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 precision montre la qualite amelioree des prédictions par rapport aux methodes precedentes.

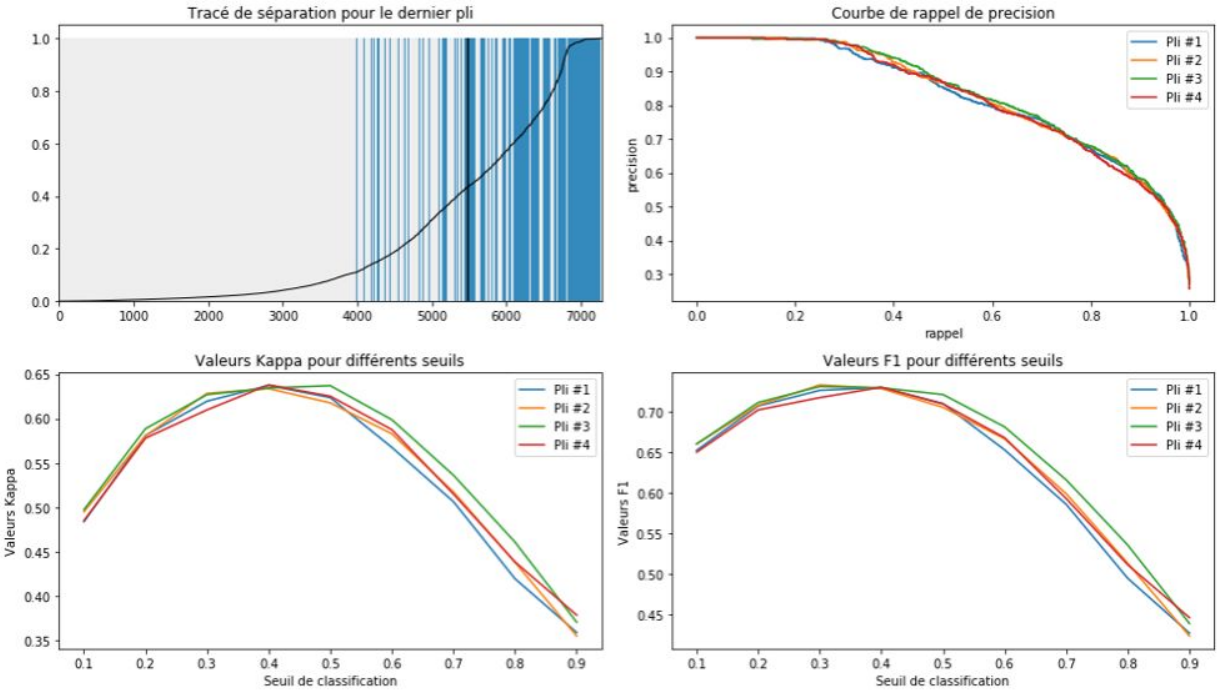
Lorsque nous avons fait une vue sur l'entraînement et la validation.



Il semble que nous n'avons pas fait de sur-ajustement pendant la formation.

Toujours dans l'objectif d'avoir de meilleur résultat, nous avons utilisé la méthode de scission avec l'algorithme SHAP.

```
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
```



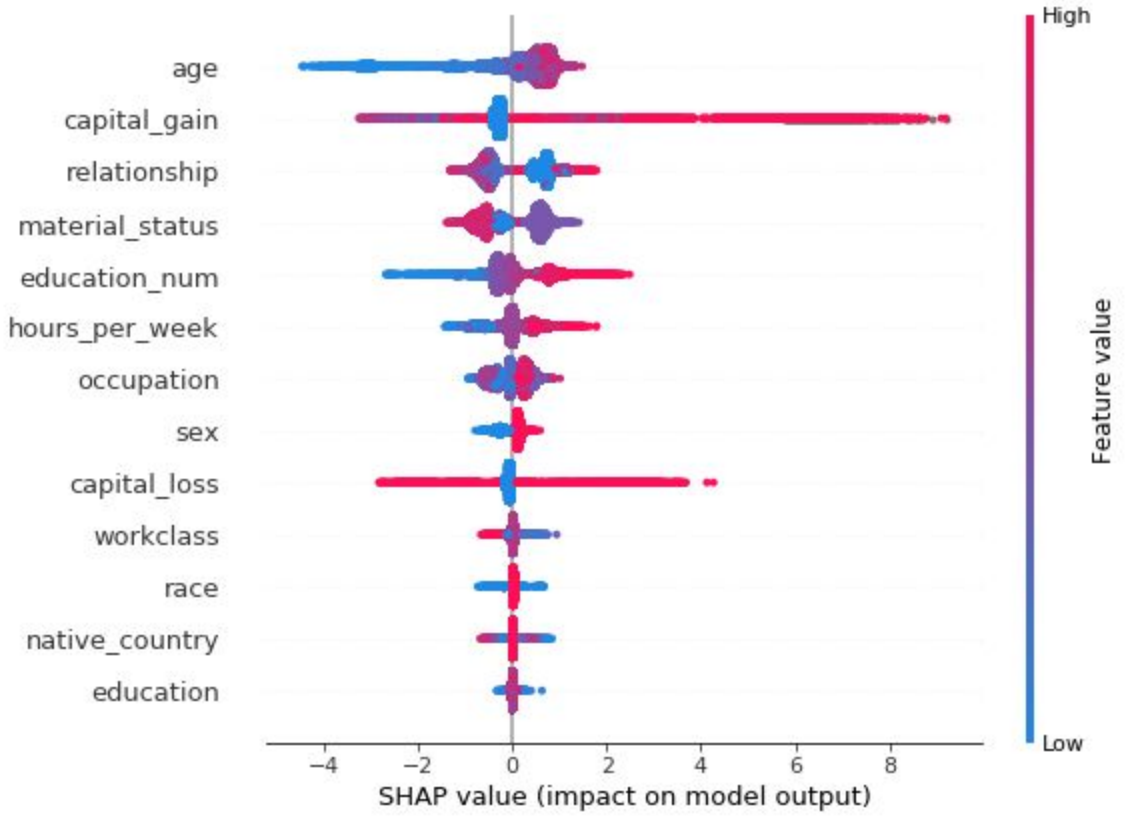
Avec cette méthode on n'a eu qu'une simple augmentation, donc pas de différence.

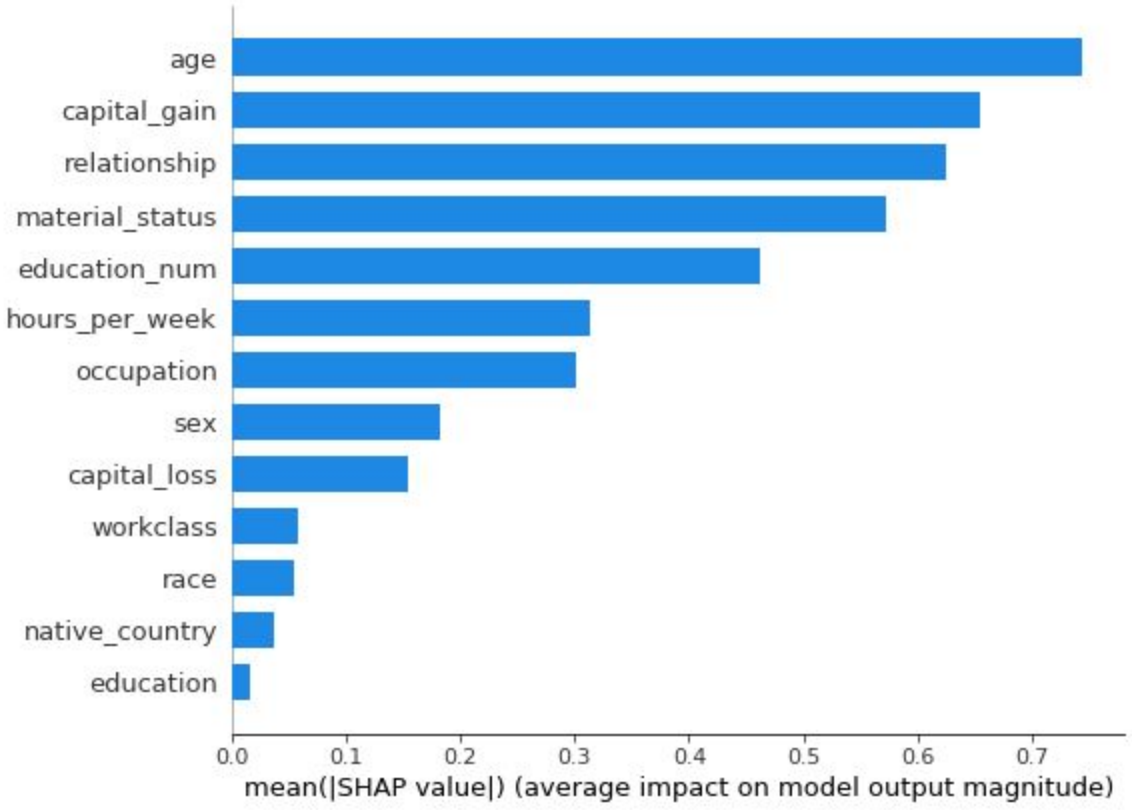
Evaluation de l'ensemble de donnees de test

```
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.

Analyse du modele de decision



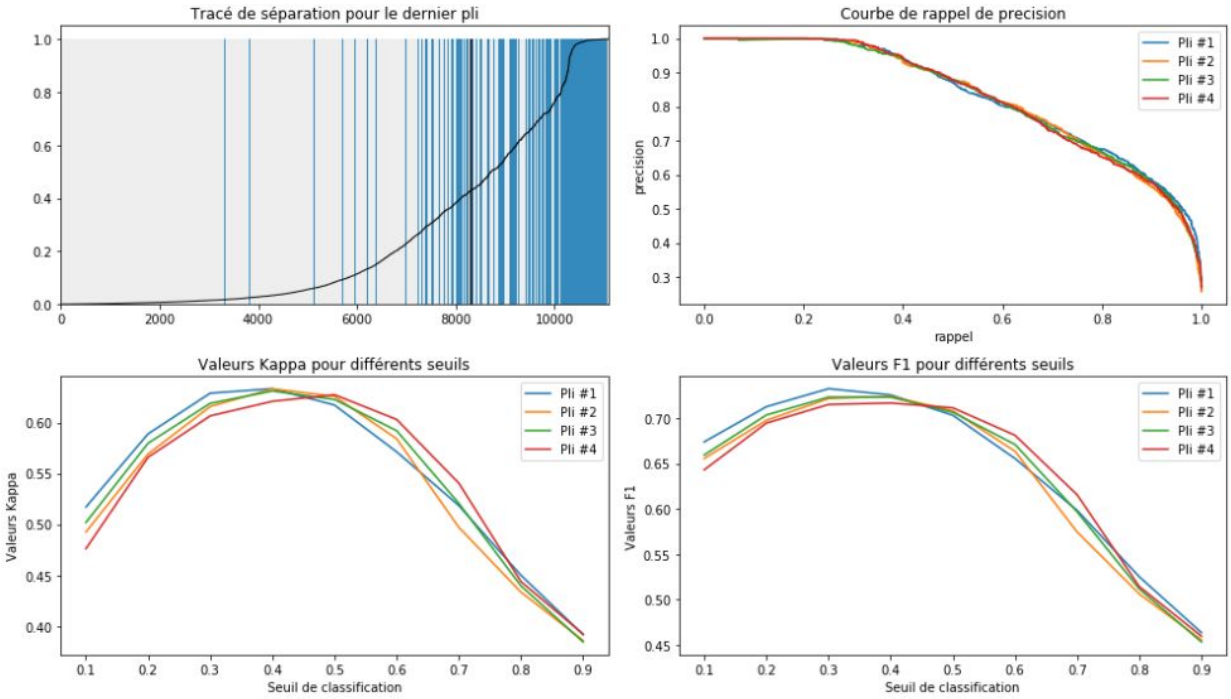


Observations:

1. La variable ***education*** duplique les informations obtenues à partir de la colonne ***education_num*** , c'est probablement pourquoi son impact est faible.
2. La variable ***native_country*** contient principalement des Etats-Unis et d'autres pays n'ont probablement pas d'impact considérable sur le résultat.
3. La variable ***race*** ressemble à une variable étrange a utiliser pour prédire les résultats.

Amelioration de la methode Arbre degrades

```
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 suppression sur les caracteristiques 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 fournie par les méthodes par défaut peut être peu fiable.

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.