# PROJET 3 : PRÉDICTION DE NIVEAU DE REVENUS

**Damien DIEUDONNE**
damien.dieudonne@etu.utc.fr

**Eloïse MOREIRA**
eloise.moreira@etu.utc.fr

**Morgan WESTMEYER**
morgan.westmeyer@etu.utc.fr

## RÉSUMÉ

Le but de ce projet est de prédire le niveau de revenus d'un individu en fonction de l'âge, de la situation personnelle et professionnelle, des études et de certaines informations personnelles de celui-ci. Nous détaillerons dans un premier temps notre analyse exploratoire des données afin de comprendre le dataset et de vérifier si des incohérences existent. Puis nous expliquerons les transformations effectuées sur les données dans la phase de pre-processing. Enfin, nous développerons la phase de modélisation et les choix que nous avons fait.

## CONTENTS

# 1 Analyse Exploratoire des Données (AED)

Ce dataset est disponible au public à partir de l'adresse suivante: https://archive.ics.uci.edu/dataset/2/adult.

Les données de ce dataset sont extraites de la base de données 1994 Census Bureau par Ronny Kohavi and Barry Becker. Le dataset est constitué de données extraites suivants les conditions suivantes :

- L'age est supérieur à 16
- Le revenu est supérieur à 100
- La valeur d'AFNLWGT est supérieure à 1 (avec une signification contextuelle spécifique)
- Le nombre d'heures travaillées par semaine est supérieur à 0

L'objectif de cette première étape est de connaître les données de notre dataset pour en déduire leur utilité pour le modèle prédictif.

## 1.1 Variables du Dataset

On commence par exécuter la commande df_revenus[:5] sur le dataframe afin de se rendre compte des différentes variables et de leurs variables rapidement.

```
df_revenus[:5]
```

| | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | income |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 1 | 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 2 | 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |
| 3 | 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| 4 | 28 | Private | 338409 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Wife | Black | Female | 0 | 0 | 40 | Cuba | <=50K |

Figure 1: Dataframe overview

Avec la commande df_revenus.shape et on constate que notre dataset contient 32561 observations avec 15 colonnes détaillées ci-dessous:

- **age** : Âge d'une personne - variable quantitative (int64)
- **workclass** : Catégorie d'emploi d'une personne - variable catégorielle (object)
- **fnlwgt** : Score en fonction de "comptes pondérés" de toutes les caractéristiques socio-économiques spécifiées de la population - variable quantitative (int64)
- **education** : Niveau d'éducation d'une personne - variable catégorielle (object)
- **education-num** : Nombre d'années d'études d'une personne - variable quantitative (int64)
- **marital-status** : Situation maritale d'une personne - variable catégorielle (object)
- **occupation** : Profession ou occupation d'une personne - variable catégorielle (object)
- **relationship** : Relation d'une personne avec sa famille ou son entourage - variable catégorielle (object)
- **race** : Race ou origine ethnique d'une personne - variable catégorielle (object)
- **sex** : Sexe d'une personne - variable catégorielle (object)
- **capital-gain** : Gains en capital d'une personne - variable quantitative (int64)
- **hours-per-week** : Nombre d'heures travaillées par semaine d'une personne - variable quantitative (int64)
- **native-country** : Pays d'origine d'une personne - variable catégorielle (object)
- **income** : Revenus soit supérieur à 50k/an ou inférieur ou égale à 50k/an - variable catégorielle (object)

**Précision** : Le terme fnlwgt (final weight) fait référence à un score en fonction de ”comptes pondérés” de toutes les caractéristiques socio-économiques spécifiées de la population. Les personnes ayant des caractéristiques démographiques similaires devraient avoir poids similaires. Le dataset a été réalisé auprès de la population états-unienne. La ”race” (terme accepté socialement aux EU), le sexe et l'age ont notamment été pris en compte pour établir ce score (par exemple, origine Hispanique).

On peut remarquer que la race est prise en compte dans la feature ”fnlwgt” alors qu'une feature race existe également, et que l'origine Hispanique est prise en compte dans fnlwgt alors qu'elle n'est pas indiquée dans race.

Les valeurs manquantes étant indiquées avec un ”?”, nous pouvons étudier où nous avons une absence de valeurs avec la commande : df_revenus.isin([” ?”]).mean()

```
Entrée [71]: df_revenus.isin([" ?"]).mean()

    Out[71]: age              0.000000
             workclass        0.056386
             fnlwgt           0.000000
             education        0.000000
             education-num    0.000000
             marital-status   0.000000
             occupation       0.056601
             relationship     0.000000
             race             0.000000
             sex              0.000000
             capital-gain     0.000000
             capital-loss     0.000000
             hours-per-week   0.000000
             native-country   0.017905
             income           0.000000
             dtype: float64
```

Figure 2: Dataframe missing value

Nous avons donc 5,63% de données manquantes pour la feature ”workclass”, 5,66% pour la feature ”occupation” et 1,79% pour la feature ”native-country”. Aucune ne dépasse 90% donc nous gardons toutes les features.

### 1.2 Variable Cible

La variable cible est une variable catégorielle qui prend uniquement deux valeurs : $\leq 50k$ ou $> 50k$ caractérisant si la personne gagne plus de 50k€ par an ou au maximum 50k€ par an.

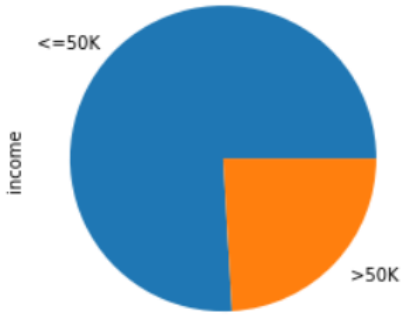


Figure 3: Répartition de la variable cible

Lors de l'analyse exploratoire des données, le constat à été fait que la distribution de la variable cible est très déséquilibrée, avec plus de 75% de salaire inférieur ou égale à 50k et moins de 25% de salaire supérieur à 50k. Cet important déséquilibre est un élément qui à été pris en compte lors de l'implémentation des différents modèles de machine learning car il a un fort impact sur leur performances. On pourrait procéder à des techniques propres aux problèmes de classes déséquilibres telles que le sous-échantillonnage.

### 1.3 Analyse Univariée

L'analyse univariée des données permet d'avoir une vue d'ensemble sur les types et les formes des données qui sont utilisées pour les modèles de machine learning. L'ensembles des features sont soit des int64 soit des object ce qui permet en théorie de les séparer facilement en variables catégorielles ou quantitatives.

On commence par analyser nos variables quantitatives :

- age
- fnlwgt
- education-num
- capital-gain
- capital-loss
- hours-per-week

Celles-ci sont toutes des variables "int", nous pouvons les étudier à l'aide d'histogrammes. Une fois les avoir générés, on se rend compte que la plupart de nos variables ont une répartition de valeurs cohérente sauf pour "capital-gain" et "capital-loss" où on retrouve des valeurs extrêmes.

Le cas des variables catégorielles à aussi du être étudié avec attention :

- workclass
- education
- marital-status
- occupation
- relationship
- race
- sex
- native-country
- income

Certaines features telles que "workclass", "occupation" ou "native-country" contiennent des valeurs "?" nécessitant un traitement particulier. Ces mêmes variables et "education" possèdent beaucoup de catégories menant à une réfléxion un peu plus approfondie pour savoir comment les traiter et éventuellement les modifier pour faciliter l'implémentation des algorithmes par la suite. De plus, nous remarquons que les variables "income" et "sex" sont les deux seules variables catégorielles binaires.

### 1.4 Analyse Bivariée

Etant donnée que le dataset contient des individus avec des revenus $\leq 50k$ et d'autres avec des revenus $> 50k$, nous allons procéder à la creation de deux sous ensembles (df_revenus_plus et df_revenus_moins) avec les commandes suivantes :

- df_revenus_plus = df_revenus[df_revenus['income'] == '$> 50k$']
- df_revenus_moins = df_revenus[df_revenus['income'] == '$\leq 50k$']

Après avoir étudier la part d'individu de chaque variable catégorielle dans la catégorie de revenus supérieur à 50k et inférieur ou égale à 50k, nous pouvons regrouper certaines variables ayant des pourcentages similaire, de plus seul income et sex auront des encodages binaires. Le reste aura un encodage ordinal en tenant compte des pourcentages présent dans l'analyse (Par exemple, dans

workclass : never worked aura un compteur ordinal plus petit que private). Pour les variables quantitatives, on observe des différences : Par exemple, plus une personne est agée, plus elle est sucéptible d'avoir un revenu $> 50k$. Nous observons de même que fnlwgt est très similaire pour les deux groupes. Sachant que cette variable est calculée en fonction de l'age, du sex et de la race, et que nous observons des différences dans chacune des variables individuellement, il serait judicieux de le retirer pour ne garder que les autres.
Enfin, la heatmap nous confirme qu'il n'y a aucune corrélation entre les données quantitatives

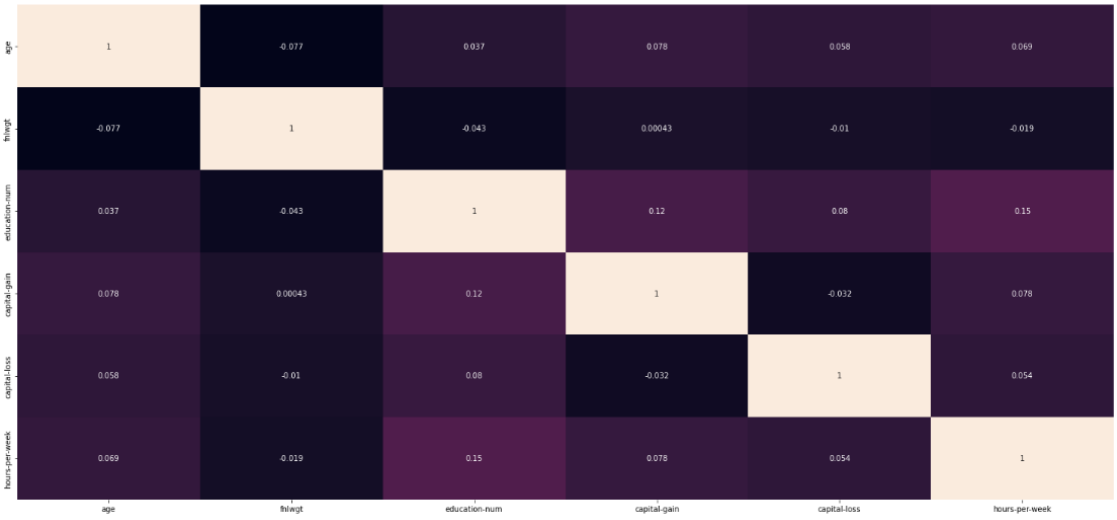


Figure 4: Heatmap variables intégrales

# 2 PRÉ-TRAITEMENT DES DONNÉES

À partir de maintenant, nous allons traiter les données afin de créer un modèle approprié. Pour cela, nous nous baserons sur les résultats de l'analyse exploratoire des données.

Dans un premier temps, nous devons séparer les données d'entraînement et de test. Heureusement, nous disposons de deux ensembles de données distincts : l'un pour les tests et l'autre pour l'entraînement. Ainsi, nous n'avons pas besoin de diviser un seul ensemble de données en deux. Ces deux ensembles de données ont respectivement une taille de 32 561 lignes et 16 281 lignes, chacun comportant 15 caractéristiques.

## 2.1 IMPUTATION DES DONNÉES

Étant donné que certaines valeurs sont manquantes, nous avons décidé de calculer le pourcentage de ces valeurs manquantes dans notre ensemble de données afin de déterminer la marche à suivre. Nous avons constaté qu'au pire des cas, il n'y a pas plus de 6% de valeurs manquantes par caractéristique. Par conséquent, nous avons choisi de supprimer les individus ayant des valeurs manquantes. Étant donné que notre ensemble de données est assez volumineux, nous pouvons nous le permettre.

De plus, il y a des caractéristiques que nous pouvons éliminer. Par exemple, "Education" peut être retirée car nous avons déjà la caractéristique "Education-Num" qui représente la même information encodée de manière ordonnée. De même, la variable "Fnlwgt" est un taux calculé à partir de l'âge, du sexe et de la race d'un individu. Cependant, lors de l'analyse exploratoire des données, nous avons remarqué qu'il n'y a pas de différences significatives entre les individus ayant un revenu supérieur ou inférieur à 50 000$ par an en fonction de cette variable. Par conséquent, nous conserverons les caractéristiques "Age", "Sexe" et "Race", qui présentent une plus grande disparité et contribueront à améliorer les performances du modèle.

## 2.2 ENCODAGE DES DONNÉES

Suite à l'analyse exploratoire des données, nous avons pu observer les taux de personnes gagnant plus de 50 000 $ par an et moins de 50 000 $ par an pour chaque variable caractéristique. Ainsi, nous avons décidé de regrouper ces variables en catégories. Les variables présentant des taux similaires ont été regroupées ensemble. Par exemple, pour la variable "Workclass", nous avons regroupé les variables "Never-Worked" et "Without-pay" en "Not-Working" (car aucun individu de ces catégories n'a un revenu supérieur à 50 000$ par an). Nous avons également regroupé les variables "Local-gov", "Self-emp-not-inc" et "State-gov" en "Selfemp-inc", car leur taux étaient similaires. Nous avons conservé les autres catégories avec leurs noms d'origine, car leur taux diffère significativement des autres. Après ce regroupement, nous avons procédé à un encodage ordinal, car nous avons observé des différences de taux entre ces catégories. Ainsi, nous avons classé ces catégories de la manière suivante :

- 0 : Not-Working
- 1 : Private
- 2 : Selfemp-inc
- 3 : Federal-gov
- 4 : Self-emp-inc

La catégorie la plus élevée correspondant à celle ayant le plus grand nombre de personnes avec un revenu supérieur à 50 000$ par an.

Nous avons effectué le même type de regroupement suivi d'un encodage ordinal pour les caractéristiques suivantes : "Marital-status", "Occupation", "Relationship", "Race" et "Native-country". De plus, nous avons encodé la caractéristique "Income" en binaire, avec 0 correspondant à inférieur ou égal à 50k et 1 correspondant à supérieur à 50k. La caractéristique "Sexe" a été encodée en one-hot.

### 2.3 NORMALISATION DES VARIABLES

Afin d'éviter qu'une certaine caractéristique ne soit surpondérée par rapport aux autres, nous avons effectué une normalisation des caractéristiques continues. Cela inclut notamment "Capital loss" et "Capital gain", qui présentent des valeurs importantes. Nous avons également normalisé "Hours-per-week" et "Age" pour les mêmes raisons.

# 3 MODÉLISATION ET ÉVALUATION

Pour la modélisation, nous avons d'abord sélectionné les jeux de données d'entraînement (Xtrain et ytrain) et de test (Xtest et ytest). Ensuite, nous avons créé une fonction appelée evaluation(model) pour évaluer les modèles.

La fonction evaluation(model) prend en entrée un modèle d'apprentissage automatique et évalue sa performance. Tout d'abord, le modèle est entraîné en utilisant les données d'entraînement. Ensuite, il est utilisé pour faire des prédictions sur l'ensemble de test (Xtest). Les prédictions obtenues sont ensuite comparées aux étiquettes réelles (ytest) à l'aide de la matrice de confusion et du rapport de classification.

La matrice de confusion permet de visualiser les prédictions correctes et incorrectes du modèle, tandis que le rapport de classification fournit des métriques telles que la précision, le rappel et le score F1 pour chaque classe.

En plus de cela, la fonction trace également une courbe d'apprentissage pour montrer comment les scores d'apprentissage et de validation évoluent en fonction de la taille de l'échantillon d'apprentissage. Cette courbe permet d'analyser les performances du modèle en termes de surapprentissage ou de sous-apprentissage.

La fonction evaluation(model) ne retourne pas explicitement de valeur, mais elle génère des sorties visuelles telles que la matrice de confusion et la courbe d'apprentissage pour aider à l'analyse de la performance du modèle.

Nous avons évalué plusieurs modèles en utilisant les paramètres par défaut :

## 3.1 MODÈLE 1 : ARBRE DE DÉCISION

Nous avons obtenu un score F1 de 58 %.

## 3.2 MODÈLE 2 : RANDOM FOREST

Nous avons obtenu un score F1 de 58 %.

## 3.3 MODÈLE 3 : ADABOOST

Nous avons obtenu un score F1 de 61%.

## 3.4 MODÈLE 4 : SVM

Nous avons obtenu un score F1 de 60%.

## 3.5 MODÈLE 5 : SVM

Nous avons obtenu un score F1 de 57%.

## 3.6 MODÈLE 6 : SVM

Nous avons obtenu un score F1 de 60%.

## 3.7 MODÈLE 7 : GRADIENTBOOSTINGCLASSIFIER

Nous avons obtenu un score F1 de 69%.

Ainsi, pour le moment, le meilleur modèle semble être le modèle Gradient Boosting Classifier avec un score F1 de 69%. Pour confirmer cela, nous avons utilisé la validation croisée (cross-validation) pour calculer la moyenne des scores F1.

Après les calculs, nous avons obtenu une moyenne de score F1 de 69%, qui reste la plus élevée parmi les modèles évalués.

Ensuite, nous avons optimisé les hyperparamètres du modèle Gradient Boosting Classifier en faisant varier les paramètres tels que le nombre d'estimateurs (n estimators), le taux d'apprentissage (learning rate) et la profondeur maximale de l'arbre (max depth). Cette optimisation a permis d'augmenter tous les scores, en particulier le score F1 du modèle Gradient Boosting Classifier qui est passé à 70%.

Finalement, nous avons sélectionné les meilleurs paramètres pour le modèle Gradient Boosting Classifier et avons obtenu un score F1 de 71%.

En examinant la matrice de confusion, nous observons que l'erreur la plus prédominante est celle de première espèce (faux positifs). Cela signifie que le modèle prédit souvent à tort que les individus ont un revenu supérieur à 50 000 $ par an.

## 4 CONCLUSION

En conclusion, en utilisant le modèle Gradient Boosting Classifier avec les meilleurs paramètres sélectionnés, nous avons obtenu un score F1 de 71%, ce qui indique une performance raisonnable dans la prédiction des revenus des individus. Cependant, il reste des possibilités d'amélioration, notamment en réduisant l'erreur de première espèce.

Pour améliorer davantage les performances du modèle et réduire cette erreur, des stratégies telles que l'ajustement des seuils de classification ou l'utilisation de techniques de pondération des classes pourraient être explorées. Cela pourrait aider à équilibrer la prédiction des classes positives et négatives.

# 5 ANNEXES

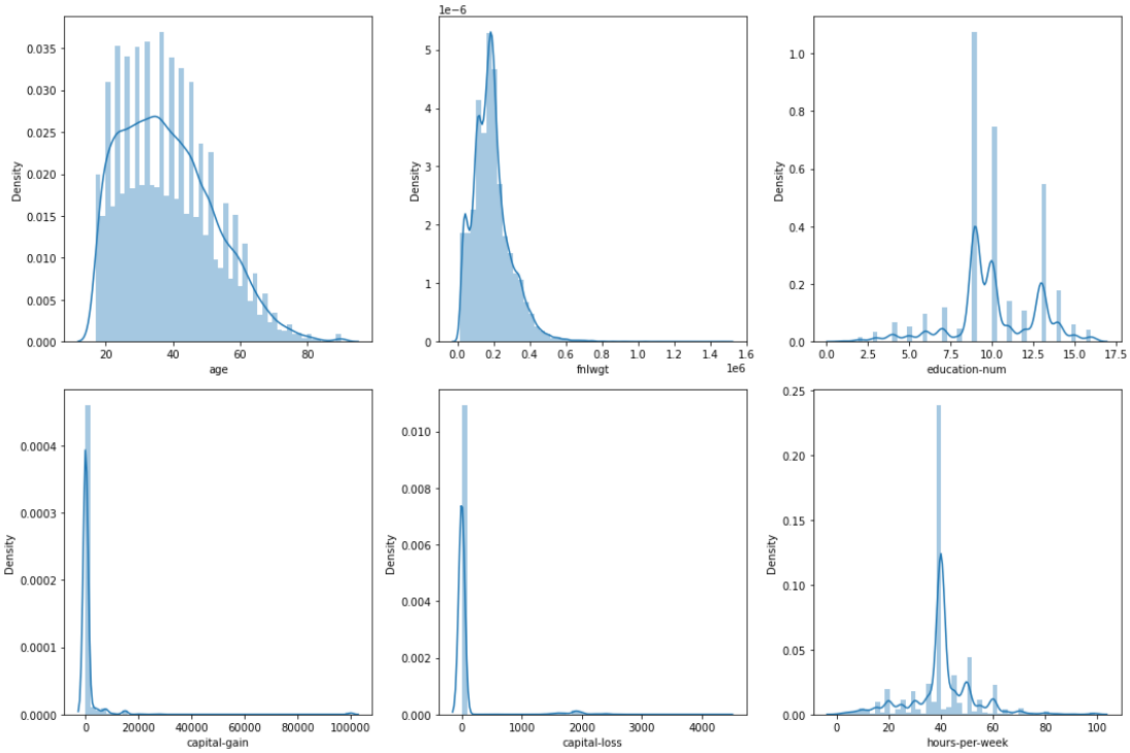


Figure 5: Histogrammes des variables quantitatives

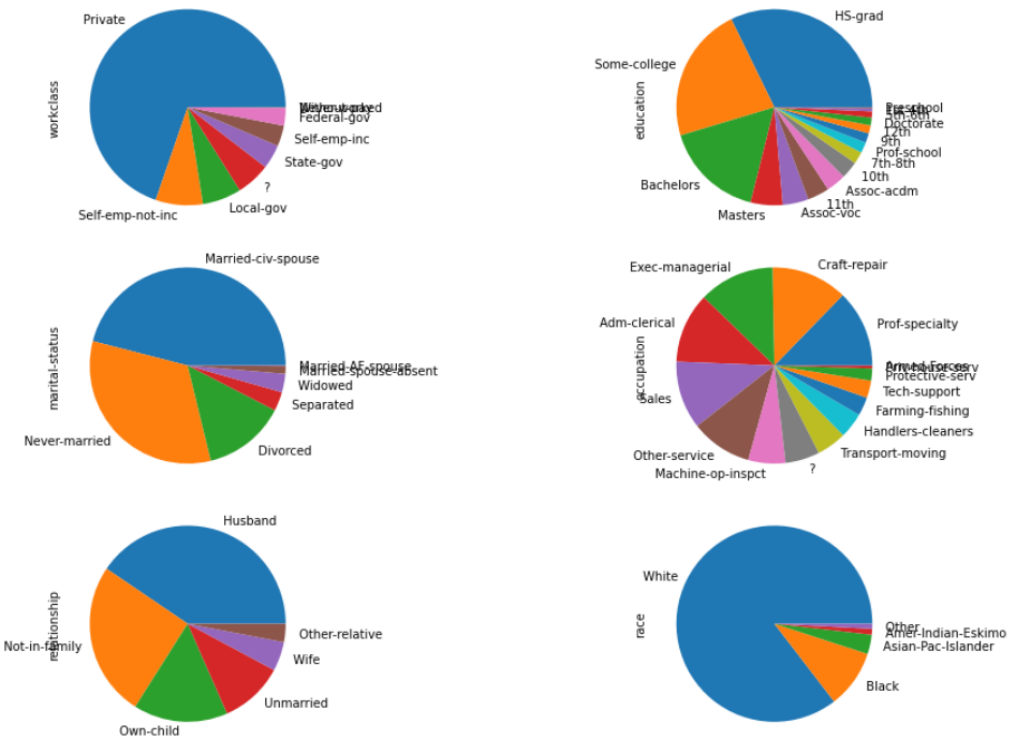


Figure 6: Pie chart des variables catégorielles part 1

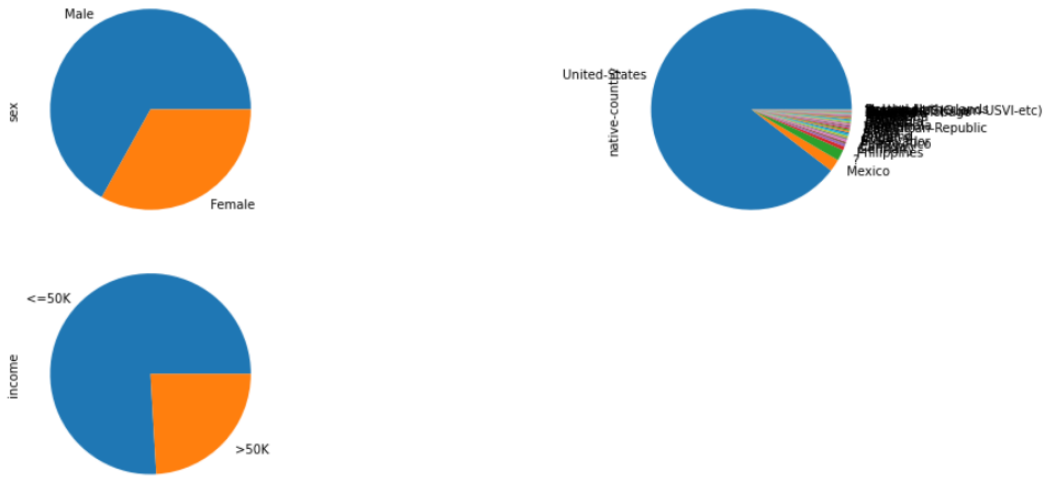


Figure 7: Pie chart des variables catégorielles part 2

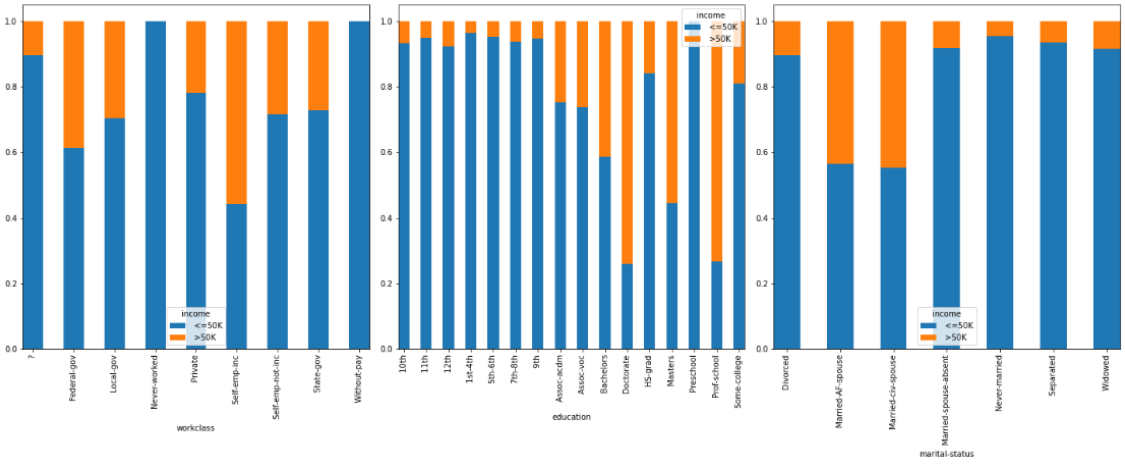


Figure 8: Etude bivariée des variables catégorielles part 1

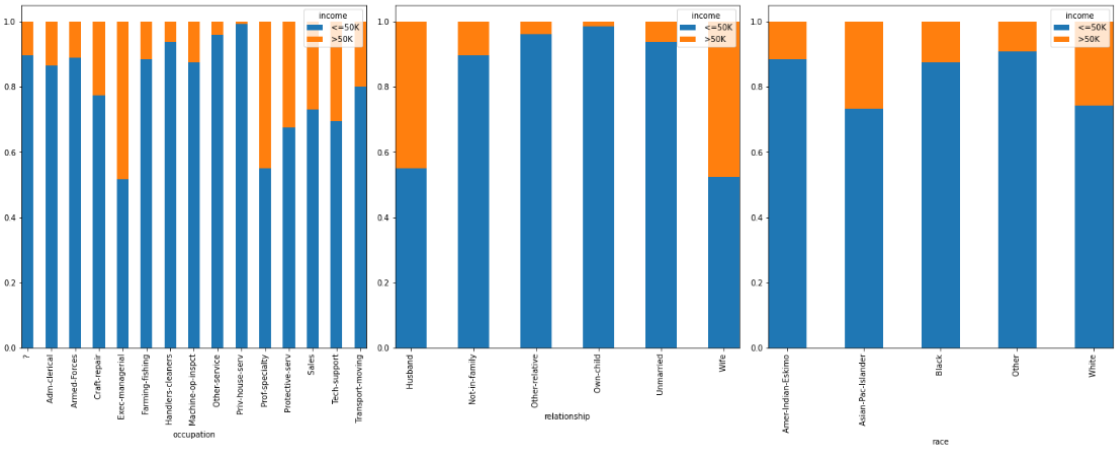


Figure 9: Etude bivariée des variables catégorielles part 2

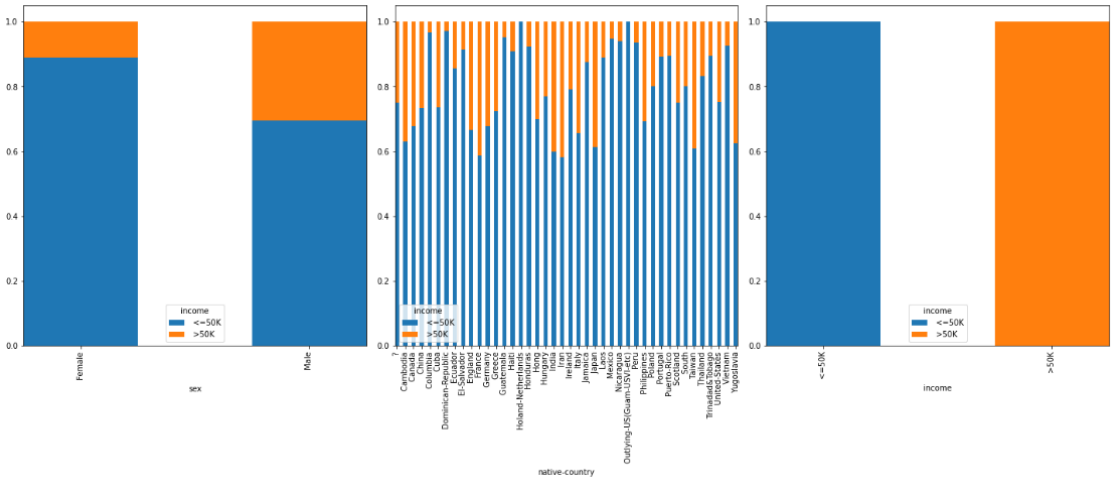


Figure 10: Etude bivariée des variables catégorielles part 3

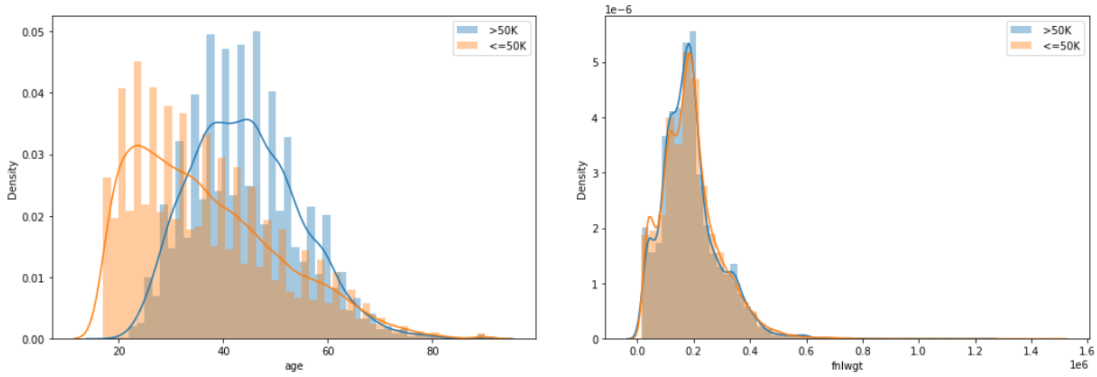


Figure 11: Etude bivariée des variables quantitatives part 1

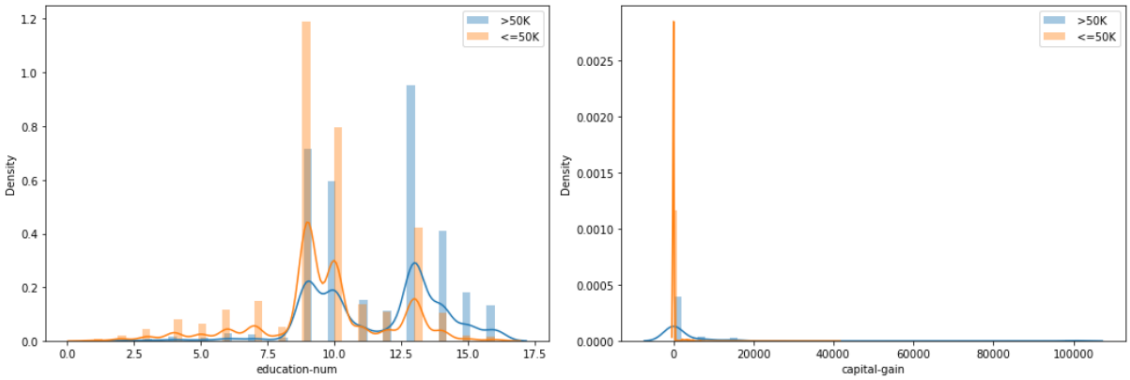


Figure 12: Etude bivariée des variables quantitatives part 2

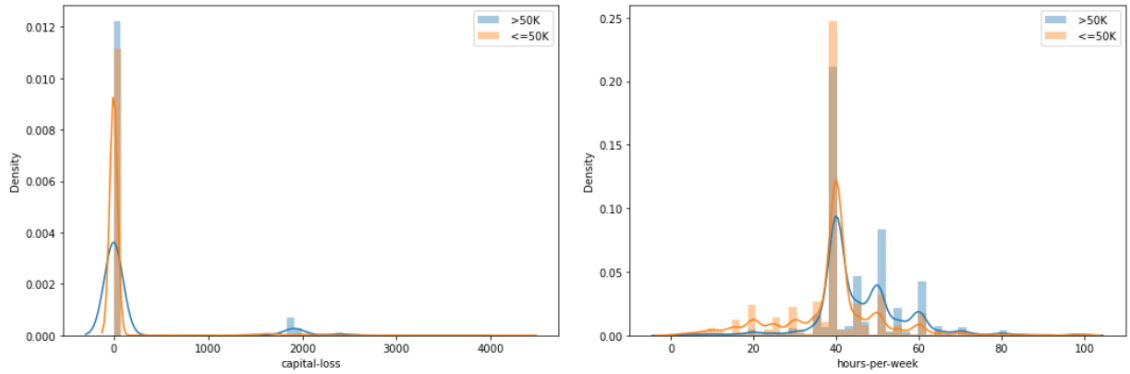


Figure 13: Etude bivariée des variables quantitatives part 3