Faculté des sciences économiques, sociales et des territoires

Université de Lille

SIAD

2025-2026

INTRODUCTION MLOPS

Projet MLOps – Analyse prédictive du salaire

De l'exploration des données à la modélisation

M. Vasseur

Djamila BENBAHLOULI
Mamadou DIALLO
Moussa ADAM
Sara ABDI
Master 2 SIAD - DS Grp 1

Tables des matières

Introduction

Dans le cadre du cours d'introduction au **MLOps**, ce projet vise à mettre en œuvre un **pipeline complet de data science orienté vers un usage opérationnel**, couvrant l'ensemble des étapes allant de l'analyse exploratoire des données à l'automatisation de la prédiction. L'approche adoptée s'appuie sur les bonnes pratiques en matière de structuration du code, de reproductibilité des traitements et de séparation des responsabilités entre les différentes phases du workflow.

Le projet repose sur l'exploitation d'un jeu de données socio-démographiques dont l'objectif est de **prédire la probabilité qu'un individu perçoive un revenu annuel supérieur à 50 000 dollars (>50K)**. Cette tâche de classification binaire constitue un cas d'usage classique en analyse prédictive, permettant d'évaluer à la fois la performance des modèles et leur capacité à être intégrés dans un processus automatisé.

Au-delà de la recherche de performance prédictive, l'enjeu central du projet réside dans la **construction d'un code modulaire et réutilisable**, capable de traiter de nouvelles données sans modification de la logique du programme. Cette exigence s'inscrit dans une démarche MLOps, où la robustesse, la maintenabilité et l'industrialisation des modèles occupent une place centrale.

Le workflow suivi comprend les étapes suivantes : **exploration des données, pré-traitement, modélisation, évaluation, prédiction et automatisation**. Plusieurs modèles de classification sont comparés, notamment la régression logistique et le Random Forest, afin d'identifier une solution adaptée aux problèmes rencontrés.

Les données utilisées proviennent d'un jeu de données revenus.csv, ils sont anonymisé et issu du cours de **Scoring**. Chaque observation correspond à un individu décrit par des caractéristiques socio-démographiques telles que l'âge, le niveau d'éducation, la situation matrimoniale, la profession, le temps de travail hebdomadaire ou encore les gains et pertes de capital. La variable cible est binaire et distingue les individus percevant un revenu inférieur ou égal à 50 000 dollars de ceux percevant un revenu supérieur à ce seuil. Un jeu de données distinct est ensuite mobilisé afin d'illustrer la phase de prédiction sur des données inédites.

Problématique

Quelles sont les caractéristiques des individus percevant un revenu élevé (supérieur à 50 000 dollars par an) et dans quelle mesure ces caractéristiques permettent-elles de prédire l'appartenance à cette catégorie de revenus ?

Afin de répondre à cette problématique, le rapport s'organise en plusieurs parties complémentaires.

Dans un premier temps, la **méthodologie générale et la structure du projet** sont présentées. Cette partie décrit les choix d'organisation des fichiers, la séparation

des différentes étapes du workflow de data science et la logique MLOps adoptée, dans une optique de lisibilité, de reproductibilité et de réutilisation du code.

Dans un second temps, le rapport s'attache à **l'exploration des données**, à travers une analyse descriptive du jeu de données. Cette étape vise à comprendre la nature des variables et de leurs modalités, ainsi qu'à étudier la distribution de la variable cible et le déséquilibre existant entre les classes de revenus.

La troisième partie est consacrée au **pré-traitement des données**. Elle détaille les opérations de nettoyage, de transformation et de regroupement des modalités mises en œuvre afin de rendre les données exploitables pour la phase de modélisation.

La quatrième partie porte sur la **modélisation, l'évaluation des performances et la prédiction**. Elle présente l'entraînement et la comparaison de plusieurs modèles de classification à l'aide d'indicateurs de performance adaptés (accuracy, précision, recall, F1-score et ROC-AUC), ainsi que le processus de sélection du modèle final. Une attention particulière est accordée à la détection de la classe minoritaire (>50K) et à l'utilisation opérationnelle du modèle retenu sur un jeu de données inédit.

Enfin, la cinquième partie est dédiée à **l'automatisation**. Elle illustre la mise en place d'un pipeline automatisé permettant de générer des prédictions sur de nouvelles données sans modification du code, conformément aux principes fondamentaux du MLOps.

I. Méthodologie

La méthodologie adoptée dans ce projet repose sur une **décomposition explicite du pipeline de data science en étapes distinctes**, chacune correspondant à une phase clé du cycle de vie d'un modèle de machine learning. Cette organisation vise à garantir la lisibilité du code, la reproductibilité des traitements et la cohérence entre les différentes phases, tout en s'inscrivant dans une logique conforme aux bonnes pratiques du MLOps. Chaque étape du workflow est implémentée dans un notebook dédié, permettant une séparation claire des responsabilités et facilitant la maintenance et l'évolution du projet.

A) Les étapes

Exploration des données

La première étape du projet est consacrée à l'**exploration des données**, réalisée dans le notebook *01_exploration*. Cette phase vise à acquérir une compréhension globale du jeu de données avant toute transformation ou modélisation. Elle comprend l'analyse de la structure du jeu de données, l'identification des types de variables (numériques et catégorielles), l'étude des valeurs manquantes ainsi que l'examen de la distribution de la variable cible. Une attention particulière est portée au déséquilibre entre les classes de revenus, élément déterminant pour la suite du projet. Cette analyse descriptive permet

également de formuler de premières hypothèses métier et d'orienter les choix méthodologiques retenus pour le pré-traitement et la modélisation.

Pré-traitement des données

La seconde étape correspond au **pré-traitement des données**, implémenté dans le notebook *02_preprocessing*. L'objectif principal de cette phase est de transformer les données brutes en un jeu de données cohérent, exploitable et compatible avec les modèles de machine learning. Les opérations réalisées incluent le nettoyage des données (gestion des valeurs manquantes, harmonisation des noms de variables), le regroupement raisonné des modalités des variables catégorielles afin de réduire leur cardinalité, ainsi que la transformation de certaines variables numériques en classes interprétables.
Une sélection des variables pertinentes est également effectuée afin de limiter la complexité des modèles et d'améliorer leur stabilité. Cette étape vise à garantir une **cohérence stricte entre les données utilisées lors de l'entraînement et celles utilisées ultérieurement pour la prédiction**, condition indispensable à toute démarche d'automatisation.
Un point central du pré-traitement réside dans la garantie d'une cohérence stricte entre les données utilisées pour l'entraînement et celles destinées à la prédiction. L'ensemble des transformations appliquées aux variables (nettoyage, regroupements, discrétisations et harmonisation des modalités) a été encapsulé dans une fonction dédiée. Ainsi il est possible de reproduire exactement les mêmes étapes de transformation sur tout nouveau jeu de données, quel que soit le notebook dans lequel il est utilisé. Elle assure donc la reproductibilité des traitements et prépare l'automatisation ultérieure du pipeline sans modification de la logique de transformation.
À l'issue de cette phase, les variables sont homogènes, interprétables et structurées de manière compatible avec les algorithmes de classification. Les données sont ainsi prêtes à être exploitées dans la phase de modélisation, au cours de laquelle plusieurs modèles sont entraînés et comparés.

Modélisation, évaluation et prédiction

La troisième étape du projet, développée dans le notebook *03_modelisation*, est dédiée à la **modélisation, l'évaluation des performances et prédiction**. Deux familles de modèles de classification sont testées : la régression logistique, utilisée comme modèle de référence en raison de son interprétabilité, et le Random Forest, choisi pour sa capacité à capturer des relations non linéaires plus complexes. Pour chaque modèle, une phase d'entraînement est suivie d'une évaluation reposant sur plusieurs indicateurs complémentaires : accuracy, précision, recall, F1-score et aire sous la courbe ROC (ROC-AUC). L'analyse des matrices de confusion permet d'approfondir l'étude des erreurs de classification, notamment en ce qui concerne la classe minoritaire (>50K). Des modeles optimisés integrent une pondération des classes et un contrôle de la complexité du modèle, le Modele random Forest optimisé (M4) est finalement retenu

comme modèle final en raison de sa meilleure capacité à détecter les individus à revenu élevé. Ce modèle est ensuite sauvegardé sous forme de fichier afin de pouvoir être réutilisé sans réentraînement.

Automatisation

La dernière étape, présentée dans le notebook *04_automation*, illustre la **dimension MLOps et opérationnelle du projet**. Ce notebook permet de charger un nouveau jeu de données, d'appliquer exactement le même pré-traitement que celui utilisé lors de l'entraînement, de charger le modèle final sauvegardé et de générer automatiquement des prédictions accompagnées de leurs probabilités associées. Les résultats sont ensuite exportés sous forme de fichiers exploitables. L'ensemble du processus repose sur des paramètres de chemins d'accès, ce qui permet de modifier le jeu de données d'entrée ou le modèle utilisé sans changer la logique du code. Cette approche permettra de réaliser une automatisation future via un script Python, une tâche planifiée (cron) ou un job MLOps dédié.

B) Structure du projet et logique MLOps

Le projet est structuré selon une organisation hiérarchique claire dans un Github, distinguant les données, les notebooks, les résultats produits et les éléments de documentation. Les dossiers dédiés aux données séparent les données brutes, les données pré-traitées et les données destinées à la prédiction. Les notebooks sont organisés de manière séquentielle, chacun correspondant à une étape précise du pipeline. Les sorties du projet regroupent les figures, les tableaux de résultats, les modèles sauvegardés ainsi que les versions exportées des notebooks. Cette organisation facilite ainsi la lisibilité du projet, la réutilisation du code et l'industrialisation progressive du pipeline

Structure du projet :

```
projet/
│
├── data/
│   ├── raw/
│   ├── processed/
│   └── prediction/
│
├── notebooks/
│   ├── 01_exploration.ipynb
│   ├── 02_preprocessing.ipynb
│   ├── 03_modelisation.ipynb
│   └── 04_automation.ipynb
```

```
│
├── outputs/
│   ├── figures/       # graphiques, courbes ROC, distributions
│   ├── tables/        # tableaux récapitulatifs, métriques
│   ├── jobs/          # modèles sauvegardés (.joblib)
│   └── notebooks_pdf/  # versions exportées des notebooks
│
├── rapport/
│
└── README.md
```

L'exécution du projet repose sur une logique séquentielle. Les notebooks doivent être exécutés dans l'ordre suivant : exploration, pré-traitement, modélisation puis automatisation. Chaque notebook dépend des sorties du précédent, ce qui garantit la cohérence globale du pipeline et la reproductibilité complète des résultats.

II. Exploration des données

L'exploration des données constitue la première étape du projet et vise à comprendre la structure, la qualité et les principales caractéristiques du jeu de données avant toute transformation ou modélisation. Cette phase permet d'identifier les spécificités des variables, d'analyser la distribution de la variable cible et de détecter d'éventuels déséquilibres susceptibles d'influencer les performances des modèles.

A) Structure générale du jeu de données

Initialement, le jeu de données utilisé contient 48842 observations individuelles décrites par 15 variables sociodémographiques et professionnelles telles que l'âge (age), le niveau d'éducation (education et educational_num), la situation matrimoniale (marital_status, relation_ship), le genre (gender), l'ethnie (race), l'origine (native_country), la profession (worclass et occupation), le nombre d'heures travaillées par semaine(hours_per_weeks), le poids des variables (fnlwgt), ainsi que les gains (capital-gain) et pertes en capital (capital_loss). La variable cible (income) est binaire et indique si le revenu annuel d'un individu est inférieur ou égal à 50 000 dollars (≤50K) ou supérieur à 50 000 dollars (>50K).

B) Qualité des données

L'étude des valeurs manquantes a révélé la présence de 6465 valeurs codées sous forme de « ? » dans certaines variables catégorielles, notamment occupation, workclass et native_country.

Figure 1 - Tableau données non renseigné en effectif et en pourcentage

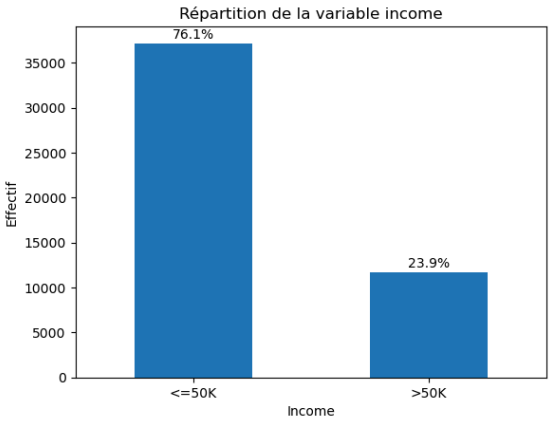
	Variable	Effectif	Pourcentage (%)
0	workclass	2799	5.73
1	occupation	2809	5.75
2	native-country	857	1.75
3	Total	6465	13.23

Ces valeurs ont été identifiées comme des informations non renseignées plutôt que comme de véritables données manquantes. Leur traitement a été planifié dans la phase de pré-traitement afin d'éviter toute incohérence ou perte d'information lors de la modélisation.
Par ailleurs, aucune anomalie majeure de structure (colonnes vides, doublons massifs ou incohérences de types) n'a été observée à ce stade.

C) Analyse de la variable cible

L'analyse de la distribution de la variable cible met en évidence un déséquilibre entre les classes.
Figure 2 - Distribution de la variable cible (income) en effectif et en pourcentage

	Modalité income	Effectif	Pourcentage (%)
0	<=50K	37155	76.07
1	>50K	11687	23.93
2	Total	48842	100.00



La proportion d'individus appartenant à la catégorie ≤50K est significativement plus élevée (76,07%) que celle des individus >50K (23,93%). Ce déséquilibre constitue un enjeu important pour la modélisation.

D) Analyse descriptive des variables explicatives

Statistiques descriptives - df

	count	mean	std	min	25%	median	50%	75%	max
age	48842.00	38.64	13.71	17.00	28.00	37.00	37.00	48.00	90.00
fnlwgt	48842.00	189664.13	105604.03	12285.00	117550.50	178144.50	178144.50	237642.00	1490400.00
educational-num	48842.00	10.08	2.57	1.00	9.00	10.00	10.00	12.00	16.00
capital-gain	48842.00	1079.07	7452.02	0.00	0.00	0.00	0.00	0.00	99999.00
capital-loss	48842.00	87.50	403.00	0.00	0.00	0.00	0.00	0.00	4356.00
hours-per-week	48842.00	40.42	12.39	1.00	40.00	40.00	40.00	45.00	99.00

Les statistiques decriptives générale et l'analyse univariée des variables a permis de dégager plusieurs tendances générales :

- Une concentration des individus dans certaines classes d'âge (entre 30 et 50 ans) avec une hétérogénéité des profils perçu dans les valeurs minimales et maximales (17 et 90 ans)
- Une population davantage masculine (66,85%) que féminine (33,15%)
- Des individus provenant des United-stated (89,74%) et appartient principalement à la catégorie raciale *White* (85,5 %),
- Du point de vue de la structure familiale, la catégorie *Husband* est la plus représentée (40,4 %), suivie des individus *Not-in-family* (25,8 %).
- Des individus avec un niveau d'éducation suffisamment élevé. Le niveau d'éducation est également marqué par une forte concentration sur certains diplômes, notamment *HS-grad* (32,3 %) et *Some-college* (22,3 %).
- Les variables liées à l'activité professionnelle montrent une forte hétérogénéité. Les catégories *Prof-specialty*, *Craft-repair* et *Exec-managerial* concentrent chacune environ 12 % des individus, tandis que d'autres catégories restent marginales.
- Une majorité d'individus avec gain et sans gain, ce qui permet de distinguer les individus, caractérisées par une prédominance de valeurs nulles (respectivement 91,7 % et 95,3 %).
- Une concentration importante autour d'un nombre standard d'heures travaillées par semaine (46,7% pour 40h/sem en moyenne, 99h/sem pour les temps de travail atypiques).

cf. Dossier Figures/exploration.

E) Premières hypothèses

Ces constats suggèrent que certaines variables pourraient présenter un pouvoir discriminant plus important que d'autres dans la prédiction du revenu. Il est probable que le revenu élevé soit associé à un niveau d'éducation supérieur, à certaines catégories professionnelles spécifiques, à une intensité de travail plus élevée et à la présence de gains en capital.

III. Pré-traitement des données

La phase de pré-traitement a pour objectif de transformer le jeu de données brut en un ensemble cohérent, exploitable et compatible avec les algorithmes de modélisation. Elle garantit que les transformations appliquées aux données d'entraînement pourront être reproduites de manière identique lors de la phase de prédiction sur de nouvelles données.

A) Recodage, nettoyage et regroupement des variables

Recodage

Une première étape a consisté à harmoniser les noms des variables afin d'éviter toute ambiguïté ou incohérence technique. Les caractères spéciaux tels que les tirets ont été remplacés par des underscores, facilitant ainsi leur manipulation en Python et leur intégration dans les pipelines de modélisation.

Nettoyage

La variable *fnlwgt*, jugée non pertinente pour l'objectif prédictif, a été supprimée. Cette décision repose sur le fait qu'elle correspond à un poids d'échantillonnage statistique et non à une caractéristique socio-économique directement interprétable dans le cadre de la prédiction du revenu.
Les modalités codées sous forme de « ? » dans certaines variables catégorielles ont été remplacées par une catégorie explicite (« Non renseigné »), permettant d'éviter la perte d'observations tout en conservant l'information liée à l'absence de réponse.

Regroupement

Pour le regroupement, certaines variables catégorielles présentaient un nombre élevé de modalités, dont plusieurs faiblement représentées. Afin de limiter la fragmentation de l'information et de réduire la complexité du modèle, des regroupements ont été réalisés de manière raisonnée.
Les regroupements ont été effectués en tenant compte :

- de la cohérence sémantique des catégories,
- de leur fréquence dans la population,
- et de leur potentiel explicatif vis-à-vis du revenu.

Regroupement :

- l'âge a 5 classes définies : « Moins de 18 », « Entre 18 et 30 ans », « Entre 31 et 50 ans », « Entre 51 et 65 ans » et « Plus de 65 ans ».
- *hours_per_week* a été transformée en 3 classes représentatives du niveau d'emploi : *under_employed* (moins de 40 heures), *normally_employed* (entre 40 et 50 heures) et *over_employed* (au-delà de 50 heures).
- Les variables *capital_gain* et *capital_loss* ont été binarisées selon la présence ou non d'un gain/perte de capital, la majorité des individus présentant une valeur nulle : « Gain de capital » / « Pas de gain de capital » et « Perte de capital » / « Pas de perte de capital ».
- La variable *native_country* a été simplifiée en deux modalités (*USA* et *Not USA*) afin de réduire la cardinalité et de conserver une lecture synthétique.
- La variable *relationship* a été regroupée de manière raisonnée : les catégories *Husband* et *Wife* ont été regroupées sous « Married », tandis que *Not-in-family* et *Other-relative* ont été rassemblées sous « Others », afin de limiter l'impact des catégories minoritaires ; les modalités *Own-child* et *Unmarried* ont été conservées.

- La variable *race* a été ramenée à trois modalités : « White », « Black » et « Other », pour obtenir une catégorisation plus stable statistiquement.
- La variable *occupation* a fait l'objet d'un regroupement en 6 catégories professionnelles plus larges, notamment « White Collar », « Blue Collar », « Agriculture », « Sales », « Protective Services » et « Other »
- Dans la même logique, *workclass* a été regroupée en 3 grands ensembles : les différents statuts gouvernementaux ont été fusionnés sous « Gov », les statuts indépendants sous « Self_emp », et les modalités très rares (*Without-pay*, *Never-worked*) regroupées sous « Other », tout en conservant la catégorie « Non renseigné » lorsqu'elle est présente.
- le niveau d'éducation (*education*) a été synthétisé en 4 catégories ordonnées : « Primary », « Middle_Low », « HighSchool_SomeCollege » et « Higher ». Pour une hiérarchie cohérente.
- La variable *marital_status* a également été regroupée sous « Married », tandis que « Married-spouse-absent » a été rapproché de « Separated »

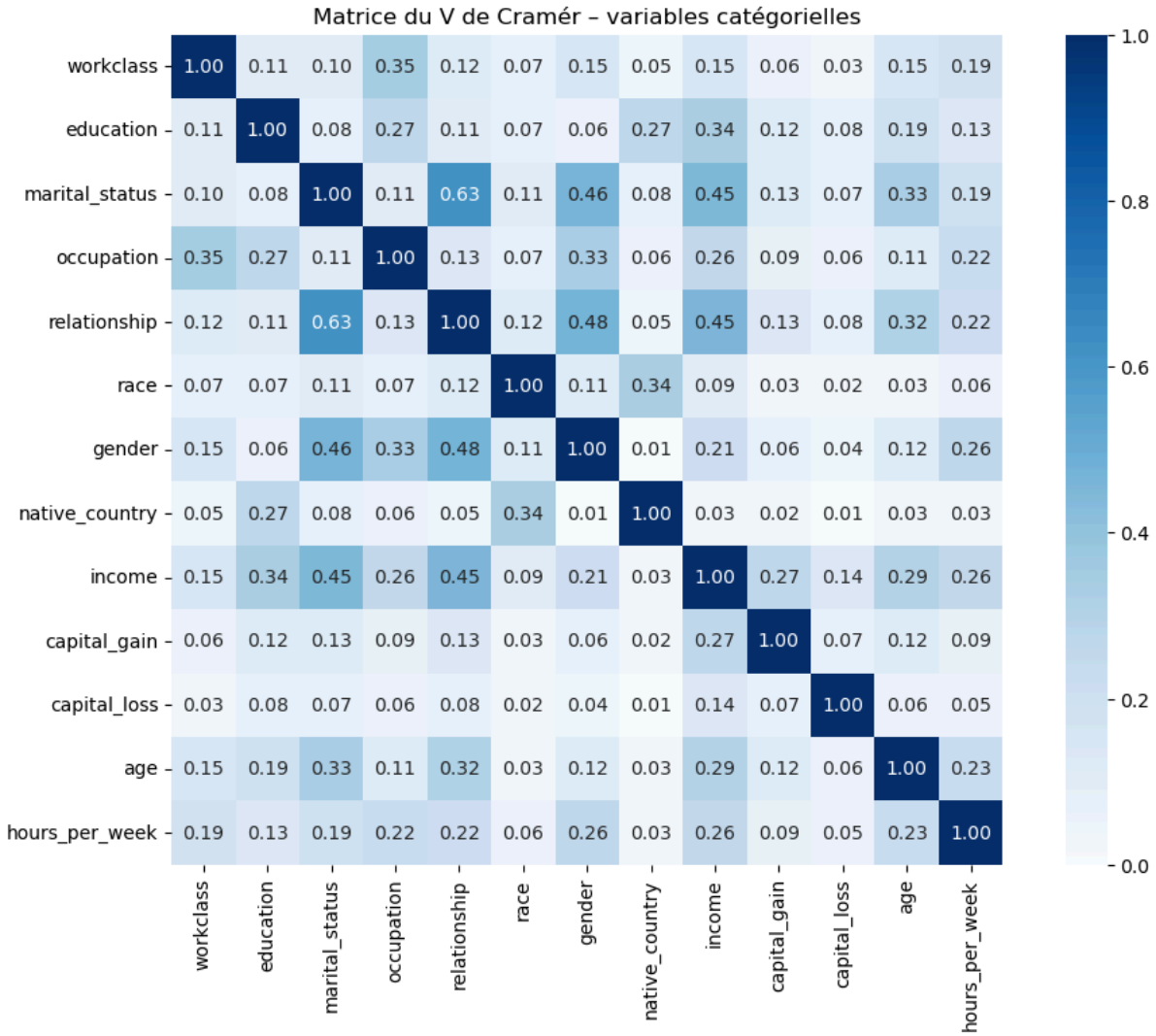
Les variables numériques telles que l'âge et le nombre d'heures travaillées par semaine ont été ainsi transformées variable catégorielle interprétable. Cette transformation permet :

- d'introduire une lecture plus métier des profils,
- de capter d'éventuelles non-linéarités,
- et de faciliter l'interprétation des résultats.

B) Préparation à la modélisation : Matrice de VCramer

La matrice du V de Cramer a été réalisée afin de mesurer l'intensité des liens entre les variables catégorielles, notamment entre les variables explicatives et la variable cible (*income*). Cette analyse permet d'identifier les variables les plus fortement associées au revenu, mais également de détecter d'éventuelles redondances entre variables explicatives. Elle constitue ainsi un outil d'aide à la sélection des variables en vue de la modélisation, tout en limitant les risques de multicolinéarité et de sur-représentation d'informations similaires dans le modèle.

Figure 3 - Matrice de Vcramer



L'analyse de la matrice du V de Cramér met en évidence plusieurs variables présentant une association notable avec la variable cible income, traduisant des liens structurels entre le niveau de revenu et certaines caractéristiques socio-démographiques et professionnelles.

Fortement lié :

Les variables marital_status et relationship apparaissent comme les plus fortement liées au revenu (V ≈ 0.45).

Modérément lié :

- La variable education présente également une association significative avec income (V ≈ 0.34), traduisant le rôle central du capital humain dans la structuration des inégalités de revenus.
- L'âge (V ≈ 0.29) suggère un effet du cycle de vie professionnel, où l'accumulation d'expérience et l'ancienneté peuvent influencer le niveau de rémunération.
- Les variables hours_per_week (V ≈ 0.26) et occupation (V ≈ 0.26) indiquent que l'intensité du travail ainsi que le type d'activité professionnelle constituent également des déterminants importants du revenu.

- Enfin, la variable capital_gain (V ≈ 0.27) révèle l'impact des revenus du capital dans la distinction des niveaux de revenus, en particulier pour les individus appartenant aux catégories de revenus élevés.

Faiblement lié :
Les variables native_country (0.03), race (0,09), gender(0,21), workclass(0.15) et capital_loss (0.14) présentent des valeurs de V de Cramér faibles, traduisant une association limitée avec la variable income lorsqu'elles sont considérées isolément.

L'étude des dépendances entre variables explicatives met en évidence certaines redondances susceptibles d'influencer la construction du modèle.

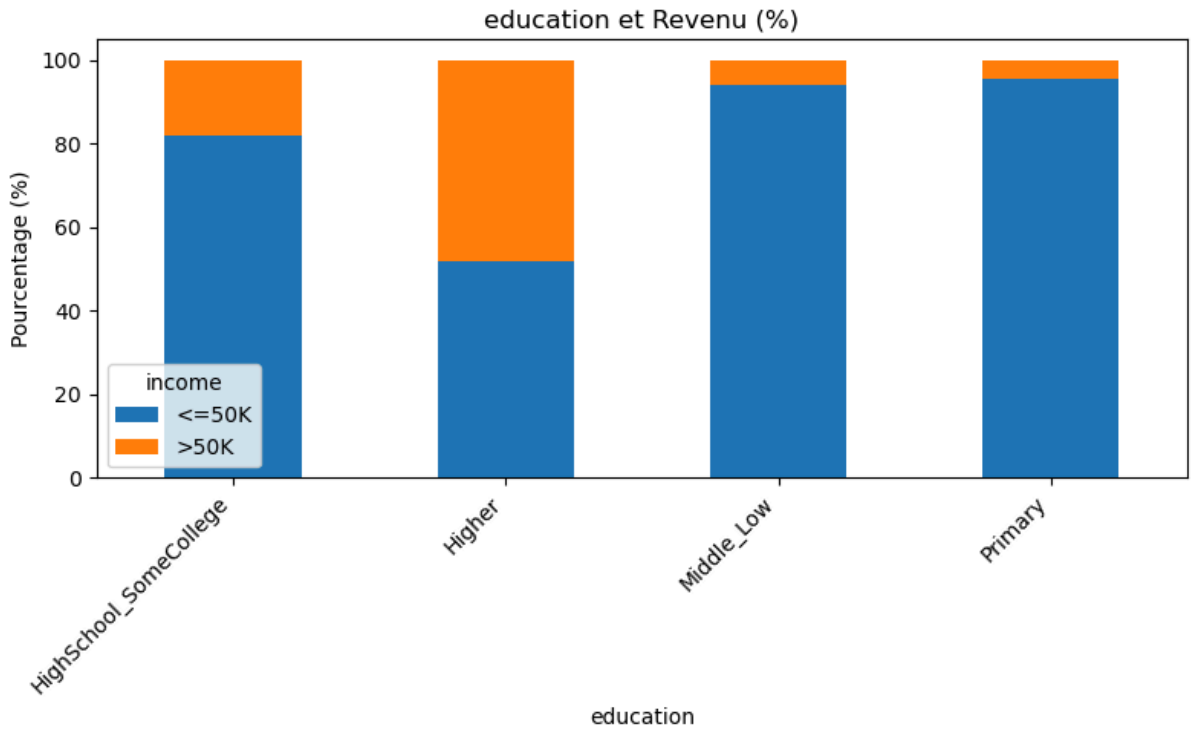
La relation la plus marquée :
les variables marital_status et relationship (V ≈ 0.63). Cette forte association indique que ces deux variables véhiculent une information très proche sur la situation familiale des individus.
De fortes associations sont également observées entre relationship et gender (V ≈ 0.48), ainsi qu'entre marital_status et gender (V ≈ 0.46), traduisant des structures sociales genrées dans les rôles familiaux.

Des liens modérés :
Entre education et occupation (0.27), ainsi qu'entre occupation et workclass(0.35), ce qui est cohérent avec la structuration du marché du travail, sans pour autant constituer des redondances strictes.

À l'issue de cette analyse, les variables *education, age, hours_per_week, occupation, capital_gain et une seule variable décrivant la situation familiale : marital_status* apparaissent comme les plus pertinentes pour la suite de l'analyse bivariée et la phase de modélisation. Les variables présentant à la fois une faible association avec la variable cible et une redondance limitée pourront être écartées ou intégrées de manière secondaire selon les performances observées lors de la modélisation.

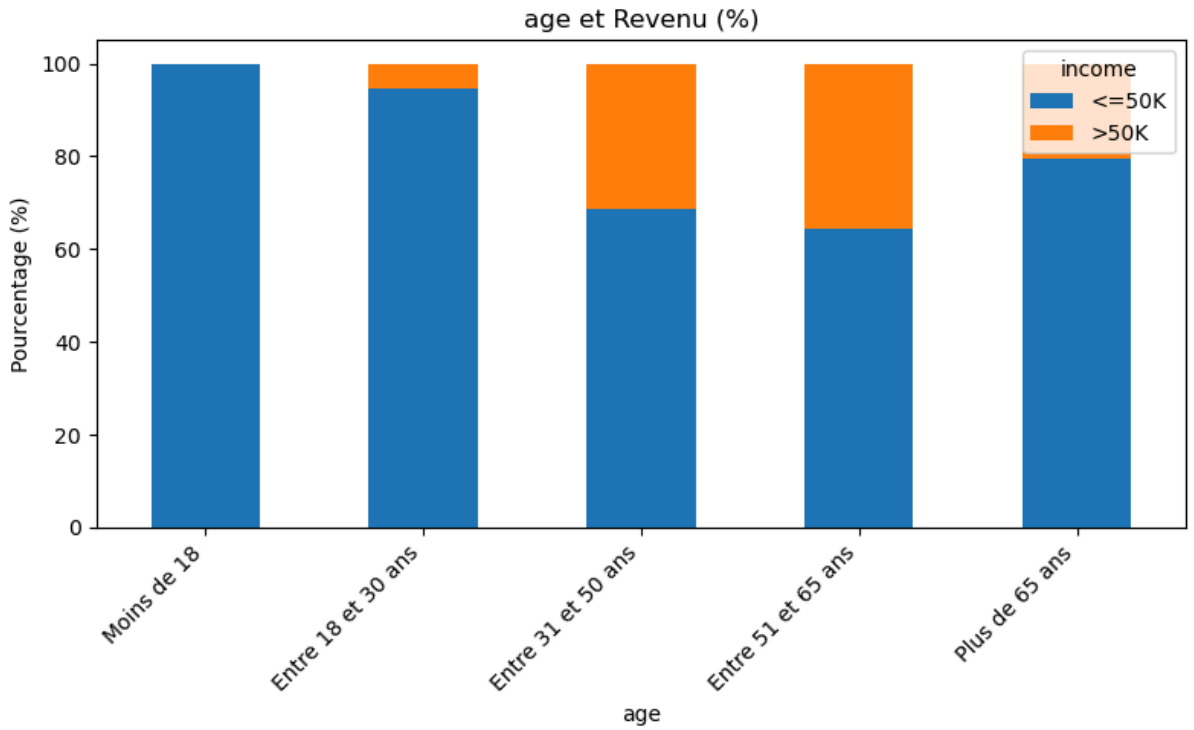
C) Analyse bivariées

Education et revenu
Plus le niveau d'étude est élevé (Higher) plus le niveau de revenu l'est aussi (48,08%) :



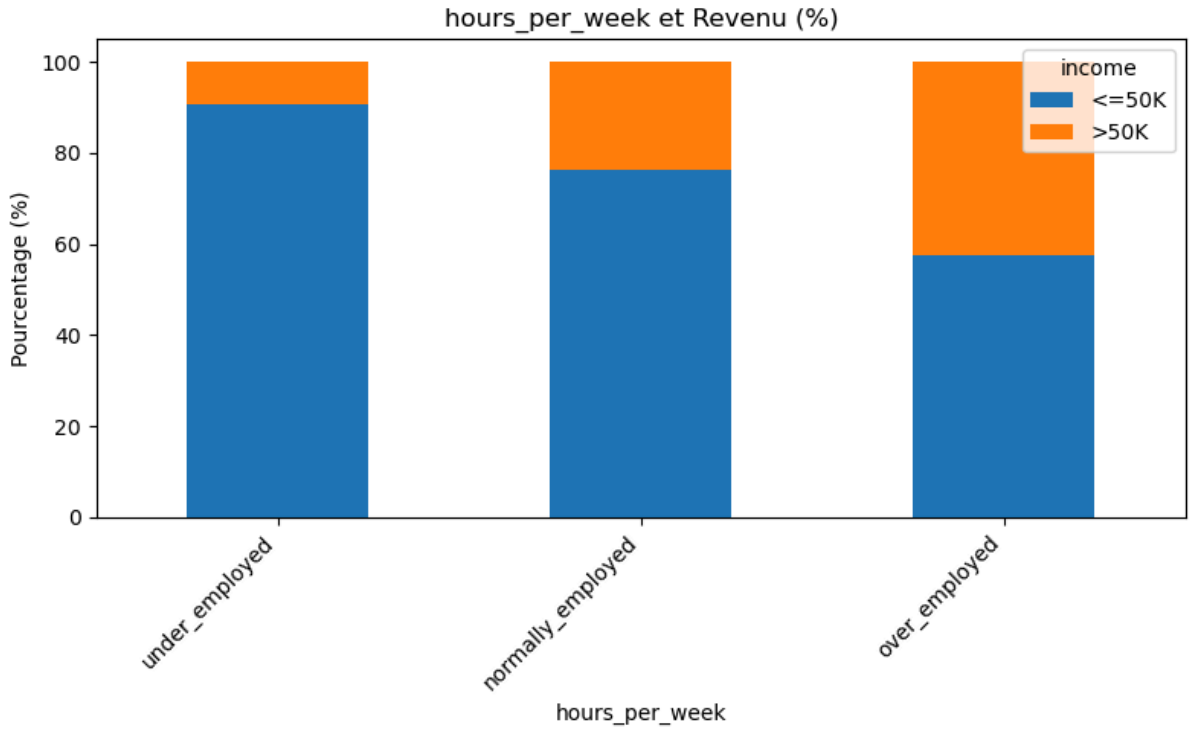
Âge et revenu

Plus l'âge est élevé (entre 51 et 65 ans) plus le niveau de revenu l'est aussi (35,72%) :



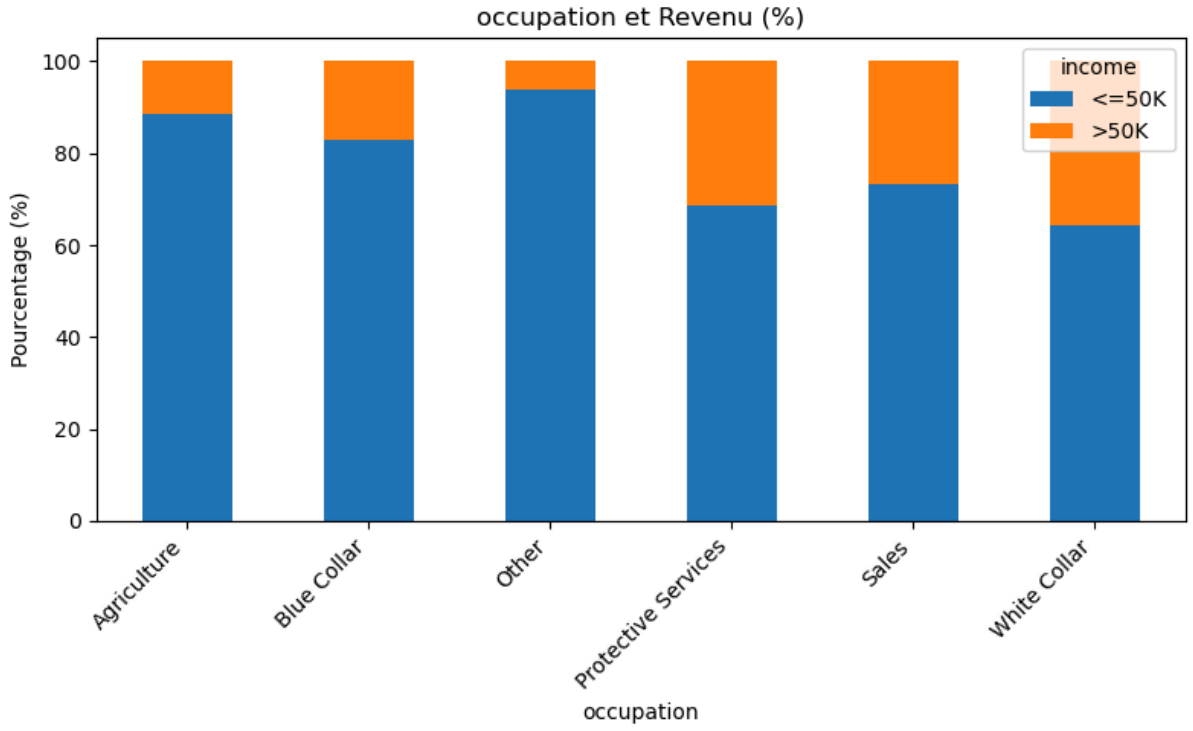
Hours_per_week et revenu :

Plus le temps de travail à la semaine est élevé (over_employed, >40h/sem) plus le niveau de revenu l'est aussi (42,60%) :



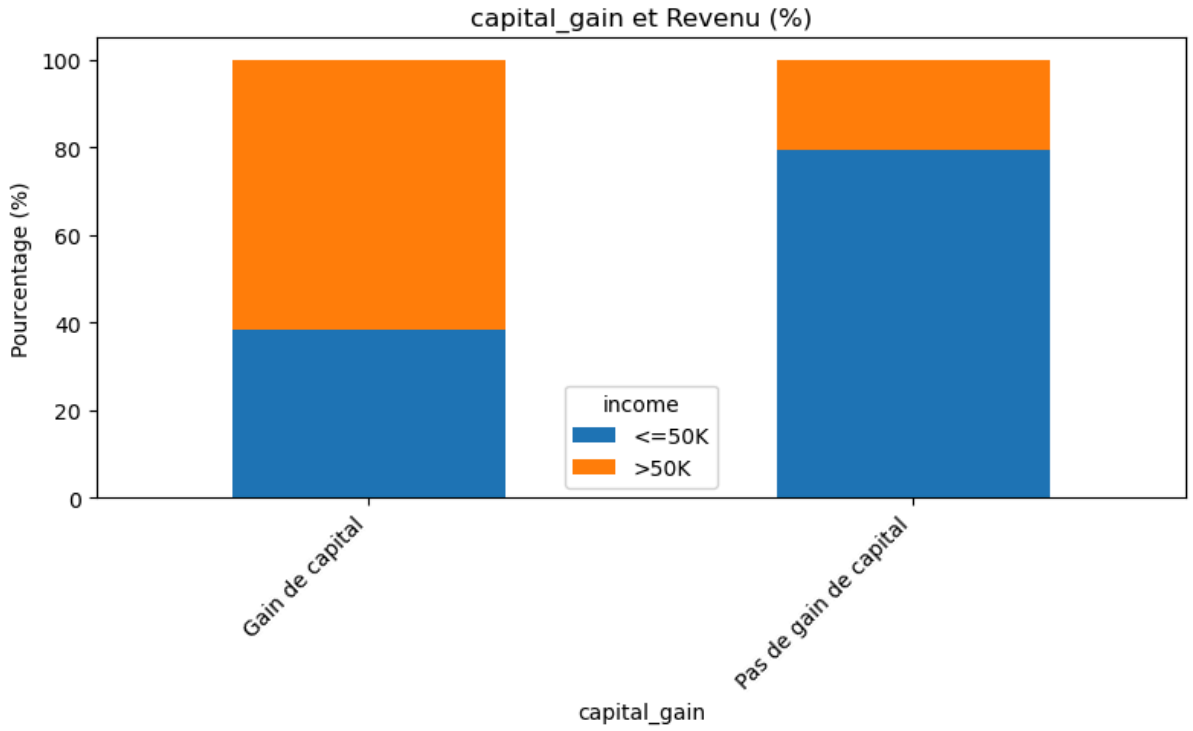
Occupation et revenu

Plus l'individu aura un travail orienté bureau (White collar) plus le niveau de revenu sere élevé (35,62%)



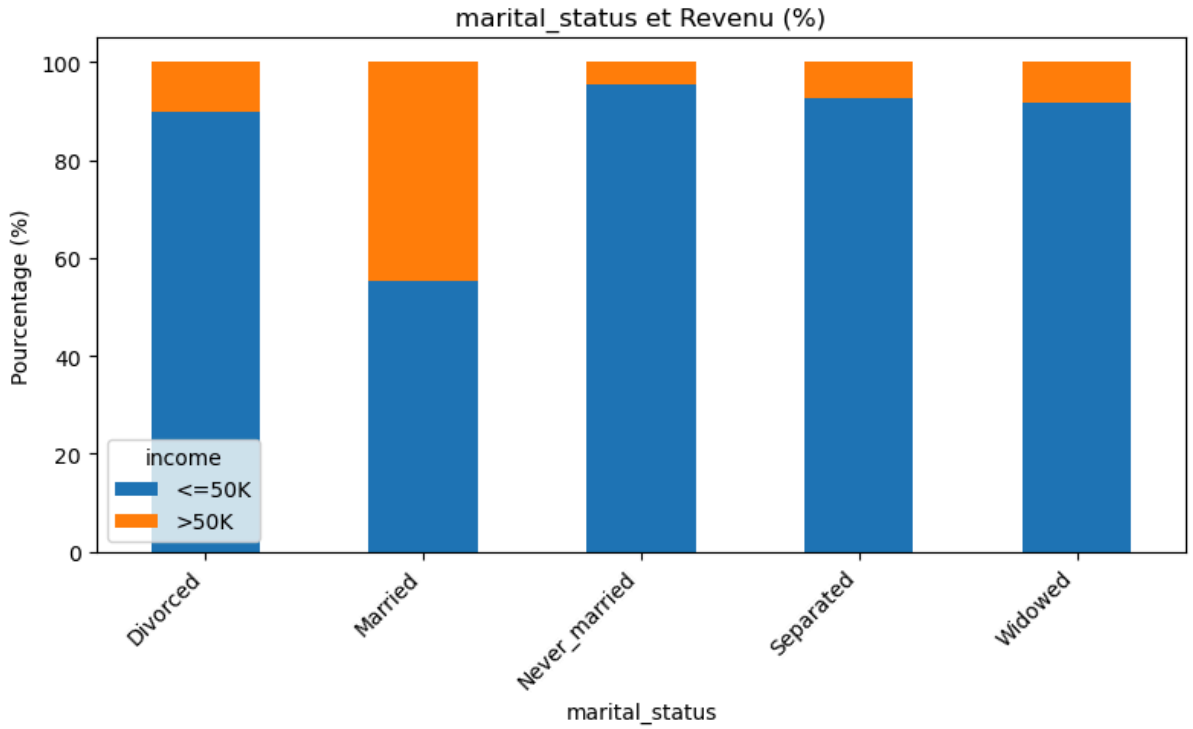
Capital_gain et revenu

Plus l'individu aura un gain de capital plus le niveau de revenu sera élevé (61,73%)



Marital_status et revenu

Plus l'individu aura une situation familial plus le niveau de revenu sera élevé (44,60%)



Ces analyses bivariées confirment les résultats observés lors de l'étude de la matrice de dépendance.

cf. Dossier Figure/prétraitement

IV. Modélisation

La phase de modélisation vise à construire un modèle capable de prédire l'appartenance d'un individu à la catégorie de revenu supérieur à 50 000 dollars (>50K) à partir de ses caractéristiques socio-démographiques. Cette étape repose sur une approche comparative et d'optimisation permettant d'évaluer plusieurs algorithmes de classification et de sélectionner le modèle le plus performant au regard des objectifs du projet.

A) Les modèles

Deux familles de modèles ont été retenues :
La **régression logistique**, utilisée comme modèle de référence, présente l'avantage d'être interprétable et robuste. Elle permet d'établir une relation linéaire entre les variables explicatives et la probabilité d'appartenir à la classe cible. Ce modèle est le point de comparaison essentiel dans tout projet de classification binaire.
Le **Random Forest**, modèle d'ensemble non linéaire, a été sélectionné pour sa capacité à capturer des relations complexes entre les variables. En combinant plusieurs arbres de décision, il réduit le risque de surapprentissage et améliore la performance prédictive, notamment en présence d'interactions entre variables.

B) Entraînement

Le jeu de données utilisé pour la modélisation comporte 48 842 observations. Afin d'évaluer correctement la capacité de généralisation des modèles, une séparation des données en deux sous-ensembles a été effectuée : un ensemble d'entraînement (80 %) et un ensemble de test (20 %). Cette répartition permet de conserver un volume suffisant d'observations pour l'apprentissage tout en réservant un échantillon indépendant destiné à l'évaluation finale des performances.
La séparation a été réalisée de manière stratifiée sur la variable cible *income*. Cette stratification est essentielle dans le contexte étudié, car la distribution des classes est déséquilibrée (environ 76 % pour la classe ≤50K contre 24 % pour la classe >50K). En maintenant cette proportion dans les ensembles d'entraînement et de test, on garantit une évaluation plus fiable et représentative des performances du modèle.
Les données ont ensuite été séparées en variables explicatives (X) et variable cible (y). La variable cible a également été transformée en version binaire (1 pour >50K, 0 pour ≤50K) afin de faciliter le calcul de certains indicateurs, notamment la courbe ROC et l'aire sous la courbe (AUC).
L'ensemble d'entraînement comprend 39 073 observations, tandis que l'ensemble de test en contient 9 769. La vérification des proportions dans chacun des sous-ensembles confirme que la distribution des classes est parfaitement conservée, ce qui assure la cohérence du protocole d'évaluation.
Enfin, les variables explicatives étant catégorielles ou discrétisées lors du pré-traitement, un encodage de type One-Hot Encoding a été appliqué via un

ColumnTransformer. Cet encodage transforme chaque modalité en variable binaire, permettant ainsi leur utilisation par les algorithmes de classification. L'option handle_unknown="ignore" a été activée afin de garantir la robustesse du pipeline en présence de modalités non observées lors de l'entraînement, notamment lors de la phase de prédiction sur de nouvelles données.
L'intégration du pré-traitement et du modèle dans un pipeline unique assure la cohérence entre les phases d'entraînement et d'inférence, condition indispensable à une mise en production suivant les principes MLOps.

C) Résultats

1) Modèle de régression logistique

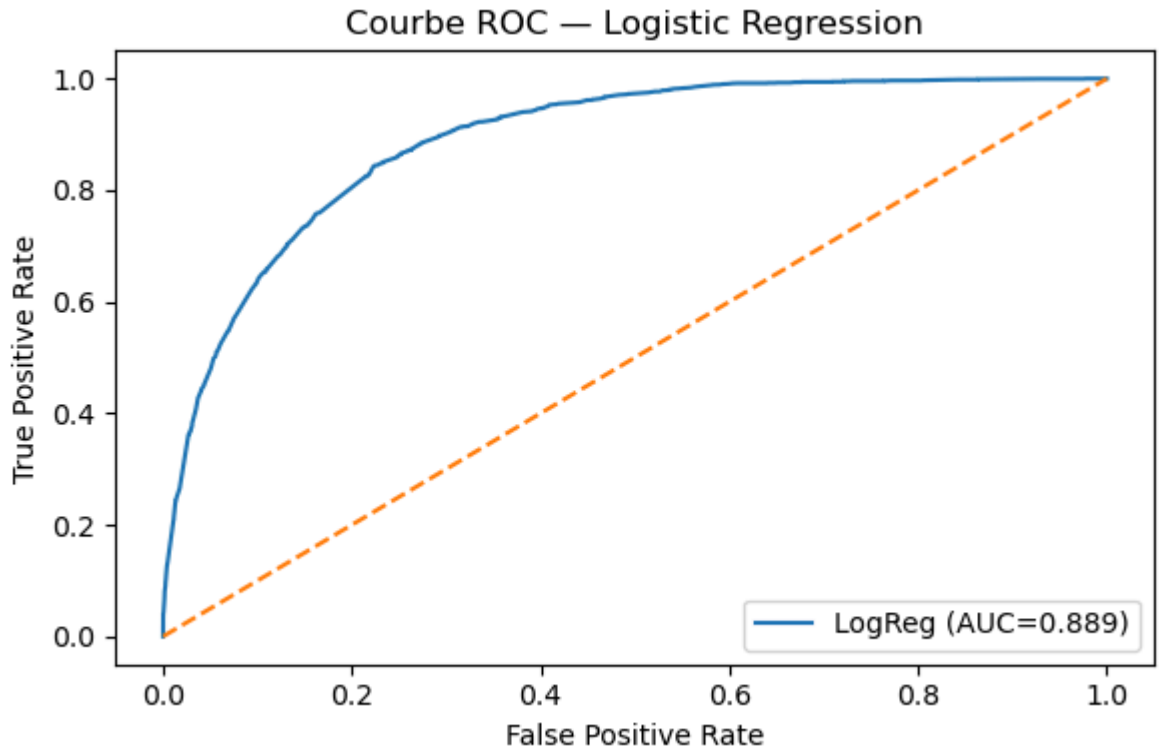
Modèle de référence (M1) :

Le modèle de reégression logistique de référence (M1) obtient une **accuracy de 84 %**, indiquant une bonne capacité globale de classification.

```
Accuracy  : 0.840
Precision : 0.705
Recall    : 0.570
F1-score  : 0.630
ROC-AUC   : 0.889

Classification report :
              precision    recall  f1-score   support

       <=50K       0.87      0.92      0.90      7431
        >50K       0.70      0.57      0.63      2338

    accuracy                           0.84      9769
   macro avg       0.79      0.75      0.76      9769
weighted avg       0.83      0.84      0.83      9769

Matrice de confusion (lignes = vrai, colonnes = prédit) :
 [[6873  558]
 [1005 1333]]
```

Le **score ROC-AUC de 0,889** montre une excellente capacité du modèle à discriminer les individus ayant un revenu supérieur à 50K de ceux ayant un revenu inférieur ou égal à 50K, indépendamment du seuil de décision.
Concernant la classe minoritaire >50K, le modèle présente une **précision de 70,5 %**, ce qui signifie que la majorité des individus prédits comme ayant un revenu élevé le sont effectivement. Le **rappel de 57 %** indique toutefois que certains individus à revenu élevé ne sont pas détectés, ce qui traduit une tendance prudente du modèle.
La **matrice de confusion** met en évidence un nombre limité de faux positifs mais un volume plus important de faux négatifs pour la classe >50K, ce qui est cohérent avec le déséquilibre des classes observé dans le jeu de données.
Ce modèle constitue ainsi une **baseline robuste**, offrant un bon compromis entre interprétabilité et performance.

Figure 4 : Courbe ROC (M1)



Modèle optimisée (M2)

L'objectif de ce second modèle est d'améliorer les performances de la régression logistique initiale, en particulier sur la classe minoritaire correspondant aux individus ayant un revenu strictement supérieur à 50K.

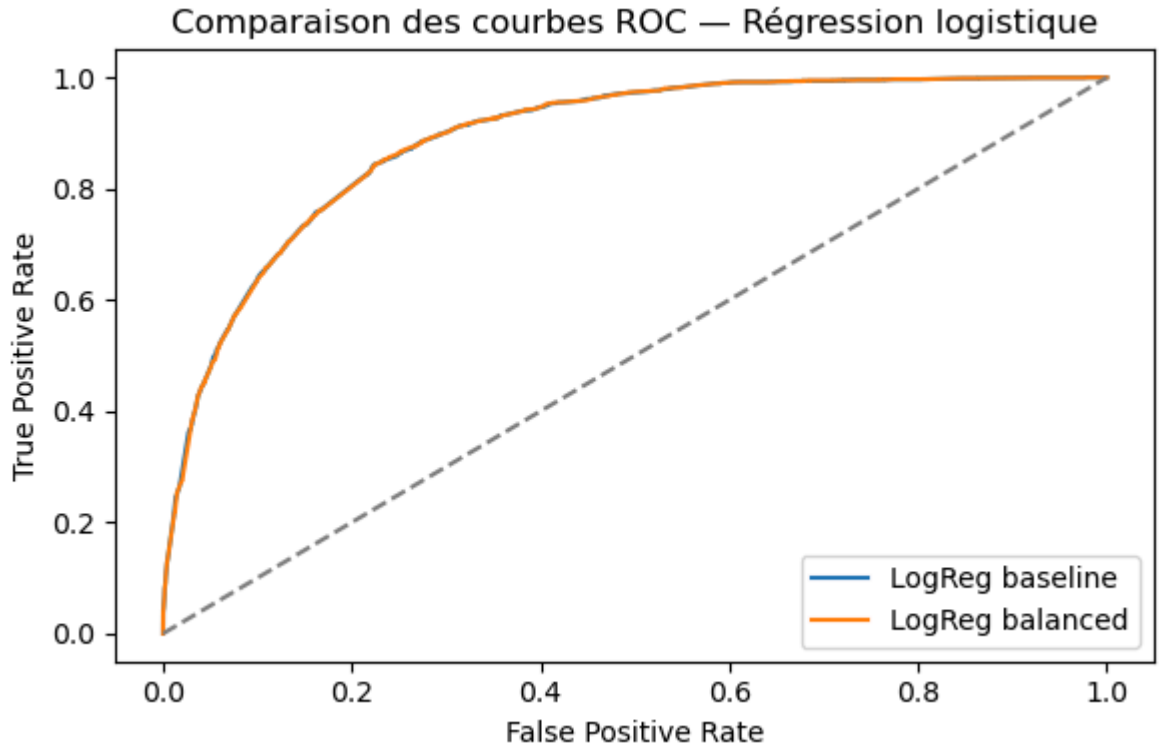
```
Accuracy  : 0.788
Precision : 0.537
Recall    : 0.848
F1-score  : 0.657
ROC-AUC   : 0.889

Classification report :
              precision    recall  f1-score   support

       <=50K       0.94      0.77      0.85      7431
        >50K       0.54      0.85      0.66      2338

    accuracy                           0.79      9769
   macro avg       0.74      0.81      0.75      9769
weighted avg       0.84      0.79      0.80      9769
```

Dans le modèle de référence, les résultats obtenus étaient globalement satisfaisants, notamment avec un score ROC-AUC élevé, traduisant une bonne capacité de discrimination. Toutefois, le rappel de la classe >50K restait limité (environ 57 %), ce qui signifie qu'une partie des individus à revenu élevé n'était pas correctement identifiée.
Afin de répondre à cette problématique, un ajustement des paramètres de la régression logistique a été envisagé. Les axes d'amélioration retenus sont les suivants : la prise en compte du déséquilibre des classes via la pondération (`class_weight`), ainsi que l'étude de l'effet de la régularisation à travers le paramètre C, qui contrôle la complexité du modèle.

Ces ajustements visent principalement à améliorer le rappel et le F1-score de la classe >50K, tout en conservant un modèle stable et interprétable.

L'utilisation d'une pondération des classes dans la régression logistique entraîne des changements significatifs dans les performances du modèle, en particulier pour la classe minoritaire correspondant aux individus dont le revenu est supérieur à 50K.
Le rappel de la classe >50K progresse fortement, passant de 57 % pour le modèle de référence à 85 % pour le modèle équilibré. Cette amélioration indique que le modèle équilibré identifie beaucoup mieux les individus à revenu élevé, réduisant ainsi le nombre de faux négatifs.
En contrepartie, la précision associée à cette classe diminue, passant de 70 % à 54 %. Cette baisse est attendue et s'explique par une augmentation du nombre de faux positifs, effet classique de l'utilisation du paramètre `class_weight="balanced"`.
L'accuracy globale du modèle diminue également, de 84 % à 79 %. Cette évolution est cohérente avec la correction du déséquilibre des classes, l'accuracy étant une métrique sensible à la classe majoritaire.

Figure 5 - Courbe ROC (M2)



Enfin, le score ROC-AUC reste strictement identique (0,889) pour les deux modèles. Ce résultat est particulièrement important, car il montre que le pouvoir discriminant global du modèle ne change pas. Seul le seuil de décision effectif est modifié par la pondération des classes.

Un ajustement du paramètre de régularisation C a été réalisé afin d'évaluer la sensibilité de la régression logistique équilibrée à la pénalisation des coefficients. Trois

valeurs ont été testées (C = 0.1, 1 et 10), tout en conservant une pondération des classes (`class_weight="balanced"`).
Les résultats montrent que le F1-score de la classe >50K est strictement identique quelle que soit la valeur de C (0,657). Cette stabilité indique que la régularisation n'est pas un facteur limitant pour ce modèle et que les variables sélectionnées portent déjà l'essentiel de l'information nécessaire à la prédiction.
Ce comportement est un signal positif, suggérant l'absence de sur-apprentissage ainsi qu'une faible sensibilité du modèle aux hyperparamètres. En l'absence de gain de performance, la valeur par défaut C = 1 est conservée pour la suite de l'étude.

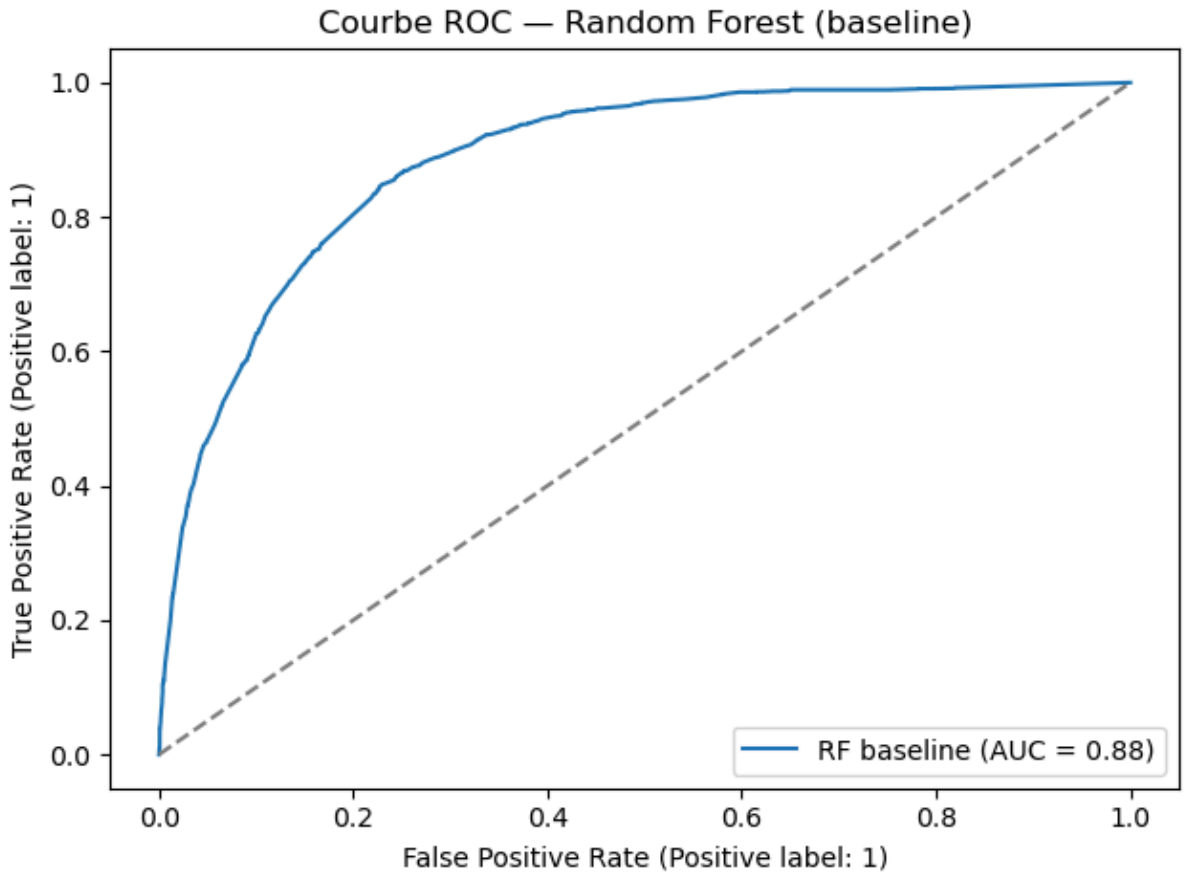
2) Modèle Random Forest

Modèle de référence (M3) :

```
Accuracy  : 0.833
Precision : 0.673
Recall    : 0.586
F1-score  : 0.627
ROC-AUC   : 0.884

Classification report :
              precision    recall  f1-score   support

       <=50K       0.87      0.91      0.89      7431
        >50K       0.67      0.59      0.63      2338

    accuracy                           0.83      9769
   macro avg       0.77      0.75      0.76      9769
weighted avg       0.83      0.83      0.83      9769

Matrice de confusion (lignes = vrai, colonnes = prédit) :
 [[6764  667]
 [ 967 1371]]
```

Le modèle Random Forest de référence présente des performances globales solides, avec une accuracy de 83,3 % et un score ROC-AUC de 0,884, indiquant une bonne capacité de discrimination entre les individus ayant un revenu supérieur à 50K et ceux dont le revenu est inférieur ou égal à ce seuil.
Concernant la classe minoritaire (>50K), le modèle obtient une précision de 67,3 % et un rappel de 58,6 %, conduisant à un F1-score de 0,627. Ces résultats traduisent une performance comparable à celle de la régression logistique de référence, avec une légère amélioration du rappel, mais sans gain significatif sur le F1-score.
La matrice de confusion met en évidence un nombre modéré de faux négatifs pour la classe >50K, indiquant que certains individus à revenu élevé ne sont pas détectés par le modèle. Ce comportement est cohérent avec l'absence de pondération des classes dans cette version de la Random Forest.

Figure 6 - Courbe ROC (M3)



La courbe ROC confirme ces observations, avec une trajectoire nettement au-dessus de la diagonale aléatoire et un AUC proche de 0,88. Malgré la capacité du Random Forest à modéliser des relations non linéaires, ses performances restent proches de celles de la régression logistique, suggérant que des ajustements de paramètres pourraient être nécessaires pour exploiter pleinement son potentiel.

Modèle optimisée (M4) :

Les résultats du Random Forest de référence montrent des performances globales comparables à celles de la régression logistique, mais mettent également en évidence une sous-détection persistante de la classe >50K, qui reste minoritaire dans le jeu de données.

Afin d'améliorer la performance du modèle sur cette classe d'intérêt, une version optimisée du Random Forest est mise en place en appliquant les bonnes pratiques de gestion du déséquilibre des classes. Le paramètre class_weight="balanced" est introduit afin d'accorder davantage d'importance aux observations de la classe >50K lors de l'apprentissage.

Par ailleurs, plusieurs paramètres sont ajustés pour contrôler la complexité du modèle et limiter le risque de sur-apprentissage :

- max_depth = 15 permet de restreindre la profondeur des arbres et d'éviter des règles trop spécifiques,
- min_samples_leaf = 20 garantit des feuilles plus robustes et mieux généralisables,
- n_estimators = 300 augmente la stabilité du modèle en s'appuyant sur un plus grand nombre d'arbres.

Ce choix de paramètres repose sur un tuning raisonné, visant à améliorer la capacité du modèle à identifier correctement les revenus élevés tout en maintenant un bon équilibre entre performance et robustesse.

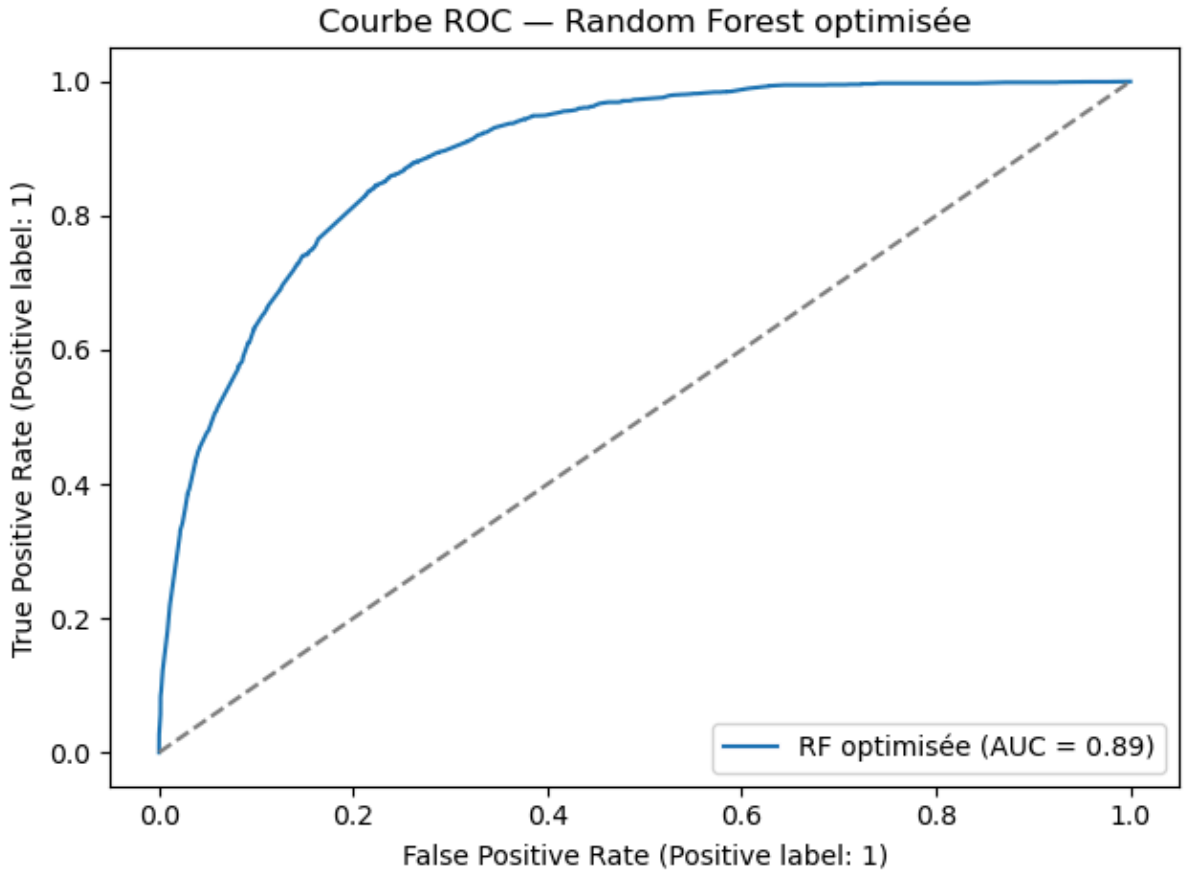
Le modèle Random Forest optimisé présente des performances globalement comparables aux modèles précédemment testés, avec un ROC-AUC élevé (0,890), indiquant un excellent pouvoir discriminant entre les individus percevant un revenu inférieur ou supérieur à 50K.

```
Accuracy  : 0.786
Precision : 0.533
Recall    : 0.856
F1-score  : 0.657
ROC-AUC   : 0.890

Classification report :
              precision    recall  f1-score   support

       <=50K       0.94      0.76      0.84      7431
        >50K       0.53      0.86      0.66      2338

    accuracy                           0.79      9769
   macro avg       0.74      0.81      0.75      9769
weighted avg       0.85      0.79      0.80      9769

Matrice de confusion (lignes = vrai, colonnes = prédit) :
 [[5676 1755]
 [ 337 2001]]
```

L'optimisation a principalement permis d'améliorer significativement le rappel de la classe >50K, qui atteint 85,6 %, contre environ 58–59 % pour les modèles baseline. Cela signifie que le modèle identifie désormais correctement une très grande majorité des individus à revenus élevés, ce qui est particulièrement pertinent dans un contexte où l'objectif est de détecter au maximum les revenus >50K.
En contrepartie, la précision sur la classe >50K diminue (53,3 %). Cette baisse était attendue : le modèle génère davantage de faux positifs, conséquence directe d'un compromis en faveur du rappel. Cette logique est cohérente avec une stratégie orientée vers la détection plutôt que la certitude absolue.
L'accuracy globale diminue (≈ 79 %) par rapport au Random Forest baseline, ce qui est également un effet classique lorsque l'on corrige le déséquilibre des classes. Cependant, cette métrique est moins pertinente dans un contexte de classes déséquilibrées et ne remet pas en cause la qualité du modèle.

Le F1-score de la classe >50K atteint 0,657, soit un bon équilibre entre précision et rappel, très proche de celui obtenu avec la régression logistique équilibrée. Cela confirme que le modèle est robuste et stable après optimisation.
Enfin, la matrice de confusion montre une forte augmentation des vrais positifs (>50K correctement prédits), validant l'intérêt de l'optimisation lorsque l'objectif métier est d'identifier les individus à hauts revenus, même au prix de quelques erreurs supplémentaires.

Figure 7 - Courbe ROC (M4)



D) Choix du modele

L'objectif principal de cette modélisation était de prédire le niveau de revenu (>50K ou ≤50K) dans un contexte de déséquilibre des classes, la classe >50K étant minoritaire.

Modèle	Accuracy	Precision (>50K)	Recall (>50K)	F1-score (>50K)	ROC-AUC	Objectif
Régression logistique (M1)	0.84	0.70	0.57	0.63	0.889	référence
Régression logistique – optimisée (M2)	0.79	0.54	0.85	0.66	0.889	Maximiser la détection >50K
Random Forest (M3)	0.83	0.67	0.59	0.63	0.884	Modèle non linéaire
Random Forest – optimisée (M4)	0.79	0.53	0.86	0.66	0.890	Détection optimale >50K

Les modèles de référence (régression logistique et random forest baseline) présentent de bonnes performances globales, avec une accuracy élevée et un ROC-AUC proche de 0.89. Toutefois, ces modèles montrent une capacité limitée à détecter correctement les individus à revenus élevés, avec un rappel autour de 57–59 % pour la classe >50K.

Afin de corriger ce biais, des versions équilibrées et optimisées ont été testées. L'introduction du paramètre class_weight="balanced" a permis une amélioration très significative du rappel de la classe >50K, atteignant environ 85–86 %, aussi bien pour la régression logistique que pour le random forest. Cette amélioration s'accompagne logiquement d'une baisse de la précision et de l'accuracy globale, phénomène attendu dans un contexte de rééquilibrage des classes.

L'analyse comparative montre que :

- le ROC-AUC reste stable, indiquant que le pouvoir discriminant global du modèle ne change pas ;
- le F1-score de la classe >50K est maximisé (~0.66) pour les modèles équilibrés ;
- les modèles optimisés sont beaucoup plus adaptés à un objectif de détection des revenus élevés.

Choix final du modèle : Random Forest optimisée (M4) :
Le modele est retenu car il présente le meilleur compromis recall / F1-score pour la classe >50K, il atteint le rappel le plus élevé (≈ 86 %), critère central de l'étude, il conserve un excellent ROC-AUC (0.890) et il est capable de capturer des relations non linéaires entre les variables explicatives et le revenu.
Ce modèle est donc le plus pertinent pour une utilisation opérationnelle, lorsque l'objectif est d'identifier un maximum d'individus à revenus élevés, même au prix de quelques faux positifs supplémentaires.

E) Prédiction

Afin d'illustrer l'usage opérationnel du modèle, un jeu de données distinct (nouvelle_data.csv) a été chargé. Ce jeu contient 9 681 observations décrivant des individus selon les mêmes caractéristiques socio-démographiques que celles utilisées lors de l'entraînement.

Conformément aux principes de reproductibilité adoptés dans le projet, le **même processus de pré-traitement** a été appliqué à ces nouvelles données via la fonction clean_data. Cette étape garantit :

- l'harmonisation des noms de variables,
- le regroupement des modalités,
- la transformation des variables numériques en classes,
- la cohérence totale entre les données d'entraînement et les données d'inférence.

Seules les 6 variables effectivement utilisées par le modèle final ont ensuite été sélectionnées :

- education
- age
- hours_per_week
- occupation
- capital_gain
- marital_status

Cette sélection assure ainsi une compatibilité stricte avec le pipeline entraîné.

Le modèle Random Forest optimisé a été utilisé pour produire :

- la **classe prédite** (income_predicted)
- la **probabilité associée à la classe >50K** (proba_>50K)

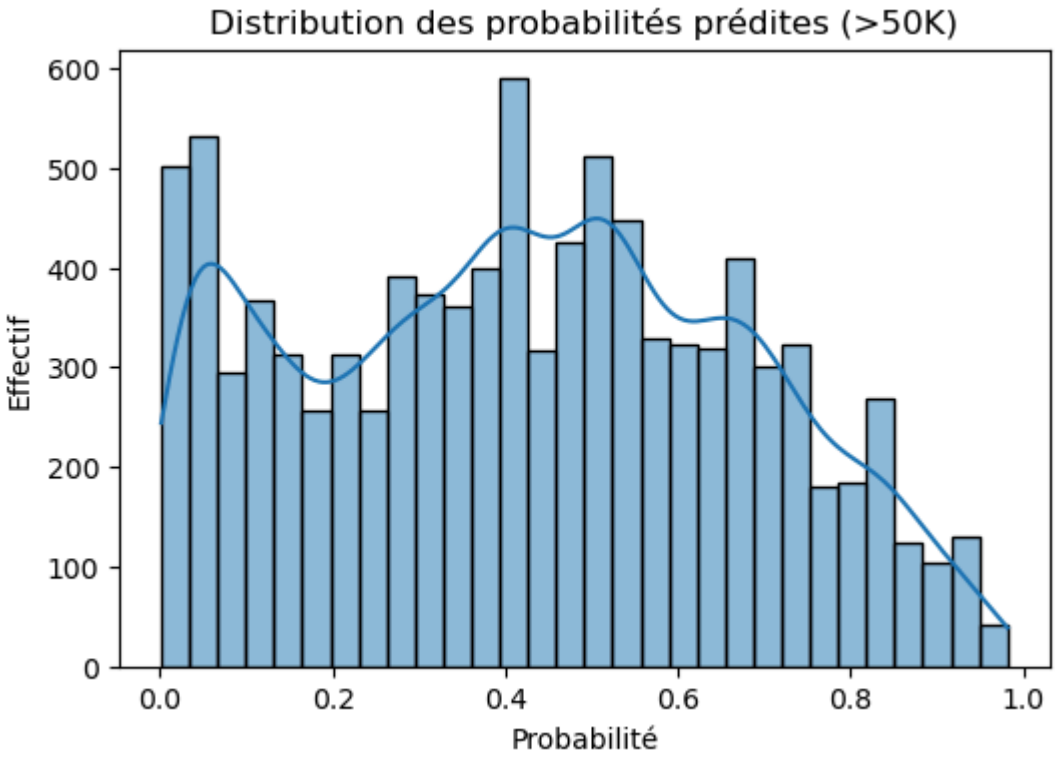
Les résultats montrent que :

- environ **60 %** des individus sont prédits ≤50K
- environ **40 %** sont prédits >50K

Cette proportion est cohérente avec l'orientation du modèle vers la détection accrue de la classe minoritaire, résultant de l'utilisation du paramètre class_weight="balanced" lors de l'entraînement

L'analyse de la distribution des probabilités prédites met en évidence une dispersion relativement large.

Figure 8 - Distribution des probabilités prédites (>50K)



Certaines observations présentent des probabilités supérieures à 0,8, traduisant une forte confiance du modèle, tandis que d'autres se situent dans une zone intermédiaire (entre 0,4 et 0,6), indiquant une incertitude plus élevée

Cette distribution permet d'introduire une réflexion métier sur le **choix du seuil de décision**.

Par défaut, la classification repose sur un seuil de 0,5. Toutefois, dans un contexte opérationnel, il peut être pertinent d'adopter un seuil plus strict.

Un seuil de **0,7** a ainsi été testé :

- Si proba_>50K ≥ 0.7 → prédiction >50K
- Sinon → prédiction ≤50K

Cette approche permet de :

- réduire le nombre de faux positifs,
- ne retenir que les prédictions les plus fiables,
- adapter le modèle aux contraintes métier (gestion du risque, ciblage marketing, sélection de profils, etc.)

Figure 9 - Distribution des probabilités prédites (>50K)

.

	income_predicted	proba_>50K	income_predicted_strict
0	>50K	0.869370	>50K
1	<=50K	0.257304	<=50K
2	<=50K	0.479623	<=50K
3	<=50K	0.466755	<=50K
4	<=50K	0.035149	<=50K

Le choix du seuil devient ainsi un levier stratégique dépendant de l'objectif opérationnel : maximiser la détection ou maximiser la précision.

L'analyse des individus présentant les probabilités les plus élevées à avoir un revenu >50K montre une cohérence avec certaines tendances observées lors de la phase exploratoire. Les profils fortement associés à un revenu >50K combinent généralement :

- un âge compris entre 51 et 65 ans,
- sexe féminin,
- un niveau d'éducation élevé,
- un statut marital marié,
- un gain de capital,
- un temps de travail important,
- non américain
- une profession classée "White Collar" dans le secteur privé, employé autres

V. Automatisation du pipeline d'inférence

La dernière étape du projet consiste à mettre en œuvre une logique d'automatisation conforme aux principes du MLOps. L'objectif n'est plus d'entraîner un modèle, mais de rendre le processus de prédiction reproductible, paramétrable et exécutable indépendamment des notebooks précédents.

Contrairement à la phase de modélisation, où l'accent est mis sur la performance et la comparaison des algorithmes, cette étape vise à transformer le modèle sélectionné en un composant opérationnel capable de produire des prédictions sur de nouvelles données sans intervention manuelle.

A) Parématrisation et séparation des responsabilités

L'automatisation repose sur une gestion explicite des chemins d'accès aux fichiers :

- chemin du jeu de données d'entrée : DATA_NEW_PATH = Path("../data/prediction/nouvelle_data.csv")

- chemin du modèle sauvegardé (fichier .joblib) : MODEL_PATH = Path("../outputs/jobs/rf_optimized.joblib")
- chemin du fichier de sortie contenant les prédictions. : OUT_PATH = Path("../data/prediction/prediction_automated/newpredictions.csv")

Cette approche présente un avantage majeur : il suffit de modifier les paramètres de chemin pour changer le jeu de données ou le modèle utilisé, sans modifier la logique du code. Le pipeline devient ainsi configurable et adaptable.

B) Chargement du modèle, génération des prédictions et sauvegarde

Après avoir intégré le processus de pré-traitement dans le notebook, le modèle final (Random Forest optimisé) est chargé depuis le dossier outputs/jobs à l'aide de joblib. Une fois chargé, il est appliqué aux nouvelles observations pour produire :

- une prédiction de classe (<=50K ou >50K),
- une probabilité associée à la classe >50K.

La probabilité permet une exploitation plus fine que la simple classe prédite, notamment pour adapter un seuil décisionnel selon des contraintes métier (par exemple, privilégier la réduction des faux positifs).

Les résultats sont ensuite exportés automatiquement vers un fichier CSV dans un dossier dédié. Le code crée dynamiquement le dossier de sortie si nécessaire, ce qui renforce la robustesse du pipeline.

Le fichier généré contient :

- les variables d'entrée nettoyées,
- la classe prédite,
- la probabilité associée.

C) Intérêt Mlops

Cette automatisation constitue une première étape vers une industrialisation du pipeline. En pratique, ce script pourrait être :

- exécuté régulièrement via une tâche planifiée (cron, planificateur Windows),
- intégré dans un job d'orchestration,
- déployé dans un environnement applicatif.

La séparation claire entre entraînement, sauvegarde du modèle et inférence sur nouvelles données illustre les principes fondamentaux du MLOps :

- modularité,
- reproductibilité,
- traçabilité
- facilité de maintenance.

VI. Discussion

A) Limites du projet

Malgré la mise en place d'un pipeline structuré et conforme aux principes fondamentaux du MLOps, ce projet présente plusieurs limites, tant sur le plan méthodologique que technique.

Sur le plan méthodologique, l'évaluation du modèle repose sur une séparation train/test classique (80/20), sans validation croisée approfondie ni validation temporelle. Bien que la stratification ait permis de préserver la distribution des classes, une validation croisée (k-fold) aurait peut être permis d'obtenir une estimation plus robuste des performances et de réduire la dépendance aux spécificités d'un seul découpage des données.

Par ailleurs, le tuning des hyperparamètres reste raisonné mais limité. Les paramètres du Random Forest optimisé (profondeur maximale, nombre d'arbres, taille minimale des feuilles, pondération des classes) ont été ajustés de manière empirique, sans recours à une recherche systématique de type GridSearch ou RandomSearch. Une optimisation plus exhaustive aurait potentiellement permis d'améliorer encore le compromis entre précision et rappel.

Une autre limite concerne l'absence de validation sur des données réellement indépendantes d'un point de vue structurel (autre source, autre période, autre contexte socio-économique). Le jeu de données utilisé pour la prédiction illustre l'inférence, mais il reste issu du même cadre que les données d'entraînement. La capacité du modèle à généraliser dans un contexte différent n'a donc pas été évaluée.

D'un point de vue MLOps, bien que la structuration du projet soit conforme aux bonnes pratiques (séparation des notebooks, dossier outputs, sauvegarde du modèle), l'utilisation de GitHub est restée limitée. Le versioning du code et des modèles n'a pas été exploité de manière avancée (gestion fine des branches, pull requests, historique détaillé des modifications, tagging de versions de modèle). Le projet repose davantage sur une organisation locale que sur une collaboration industrielle complète. Une intégration continue (CI/CD) ou des workflows automatisés n'ont pas été mis en place.

Enfin, aucun mécanisme de suivi post-déploiement n'a été implémenté. En pratique, un modèle déployé devrait faire l'objet d'un monitoring continu afin de détecter d'éventuelles dérives de données (data drift) ou de performances (model drift).

B) Difficultés rencontrées

Plusieurs difficultés ont été rencontrées au cours du projet.

La première concerne la gestion de la cohérence entre les différentes étapes du pipeline, notamment entre le pré-traitement et la prédiction. Garantir que les transformations appliquées aux données d'entraînement soient strictement identiques lors de l'inférence constitue un enjeu central en MLOps. La nécessité d'encapsuler ces transformations dans une fonction dédiée a été un apprentissage clé du projet.

La seconde difficulté a porté sur la gestion du déséquilibre des classes. L'amélioration du rappel de la classe >50K s'est faite au prix d'une baisse de précision, imposant un arbitrage méthodologique. Cette tension entre détection maximale et réduction des faux positifs illustre les compromis classiques en classification binaire.

Enfin, la structuration GitHub et l'organisation du projet selon une logique proche de l'industrialisation ont représenté un défi. Passer d'un travail exploratoire en notebook à une logique plus modulaire et paramétrable nécessite une rigueur supplémentaire dans la gestion des fichiers, des chemins et des dépendances.

3. Usage de l'intelligence artificielle dans le projet

L'intelligence artificielle a été utilisée comme outil d'assistance méthodologique et technique tout au long du projet.

Elle a notamment contribué à :

- clarifier certaines notions liées aux indicateurs de performance (ROC-AUC, F1-score, recall),
- structurer le pipeline de manière cohérente avec les standards MLOps,
- résoudre des erreurs techniques (gestion des chemins, importations, cohérence des variables),
- améliorer la rédaction du rapport dans un cadre académique structuré.

Cependant, l'IA n'a pas remplacé la réflexion méthodologique. Les choix de variables, les regroupements de modalités, les arbitrages entre modèles, ainsi que la sélection finale du modèle reposent sur une analyse critique des résultats obtenus. L'outil a servi de support, mais la responsabilité scientifique et les décisions finales relèvent du travail des étudiants.

Cette utilisation illustre une évolution des pratiques en data science : l'IA devient un assistant de productivité et de structuration, mais ne se substitue ni à l'analyse ni à la compréhension des résultats.

4. Perspectives d'amélioration

Plusieurs axes d'amélioration peuvent être envisagés dans une logique d'industrialisation progressive.

Sur le plan méthodologique :

- mettre en place une validation croisée systématique,
- implémenter une recherche automatisée d'hyperparamètres (GridSearchCV),
- tester d'autres algorithmes (Gradient Boosting, XGBoost),
- intégrer des métriques métier plus spécifiques selon le contexte d'usage.

Sur le plan MLOps :

- versionner explicitement les modèles (model versioning),
- intégrer un système de suivi des performances en production,
- automatiser entièrement le pipeline via un script Python exécutable indépendamment des notebooks,
- mettre en place un outil de suivi des dérives de données,
- intégrer le projet dans une logique CI/CD (par exemple via GitHub Actions).

Enfin, dans une perspective plus avancée, le pipeline pourrait être encapsulé sous forme d'API (par exemple avec FastAPI) afin de rendre le modèle accessible via une interface applicative.

Conclusion

Ce projet, réalisé dans le cadre du cours d'introduction au MLOps, avait pour objectif de concevoir un pipeline complet de data science, depuis l'exploration des données jusqu'à l'automatisation de la prédiction, dans une logique de structuration, de reproductibilité et de préparation à un usage opérationnel.

À partir d'un jeu de données socio-démographiques, nous avons cherché à répondre à la problématique suivante : **quelles caractéristiques sont associées à un revenu élevé (>50K) et comment construire un modèle capable de prédire cette probabilité de manière fiable ?**

L'analyse exploratoire a permis de mettre en évidence un déséquilibre significatif entre les classes et d'identifier les variables les plus liées au revenu, notamment le niveau d'éducation, la situation matrimoniale, le capital gain et le temps de travail hebdomadaire. Le pré-traitement a joué un rôle central dans la qualité de la modélisation : harmonisation des variables, regroupements raisonné des modalités, transformation de variables numériques en classes interprétables et encapsulation des traitements dans une fonction réutilisable.

Deux familles de modèles ont été comparées : la régression logistique, utilisée comme modèle de référence interprétable, et le Random Forest, capable de capturer des relations non linéaires. L'analyse des indicateurs (accuracy, recall, F1-score, ROC-AUC) a montré que le Random Forest optimisé offrait le meilleur compromis, notamment en matière de détection de la classe minoritaire >50K. Ce modèle a donc été retenu comme modèle final et sauvegardé pour un usage ultérieur.

Au-delà des performances, la contribution principale du projet réside dans la structuration du pipeline selon une logique MLOps. La séparation claire des notebooks (exploration, pré-traitement, modélisation, automatisation), l'organisation des dossiers (data, outputs, jobs, figures) et la sauvegarde du modèle permettent de réutiliser le pipeline sur de nouvelles données sans modifier la logique interne du code. La phase d'automatisation illustre cette dimension opérationnelle en permettant de charger un nouveau jeu de données, d'appliquer les mêmes transformations et de générer automatiquement des prédictions exportables.

Ainsi, ce projet ne se limite pas à la construction d'un modèle prédictif performant ; il constitue une première étape vers une démarche d'industrialisation de la data science. Il met en évidence l'importance de la reproductibilité, de la modularité du code et de la séparation des responsabilités dans la conception de projets analytiques destinés à évoluer vers des environnements de production.