

Prédiction de la sévérité d'accidents corporels de la route

11 janvier 2024



Sommaire

01 Traitement des données

02 Entraînement des 1ers modèles

03 Représentation one-hot

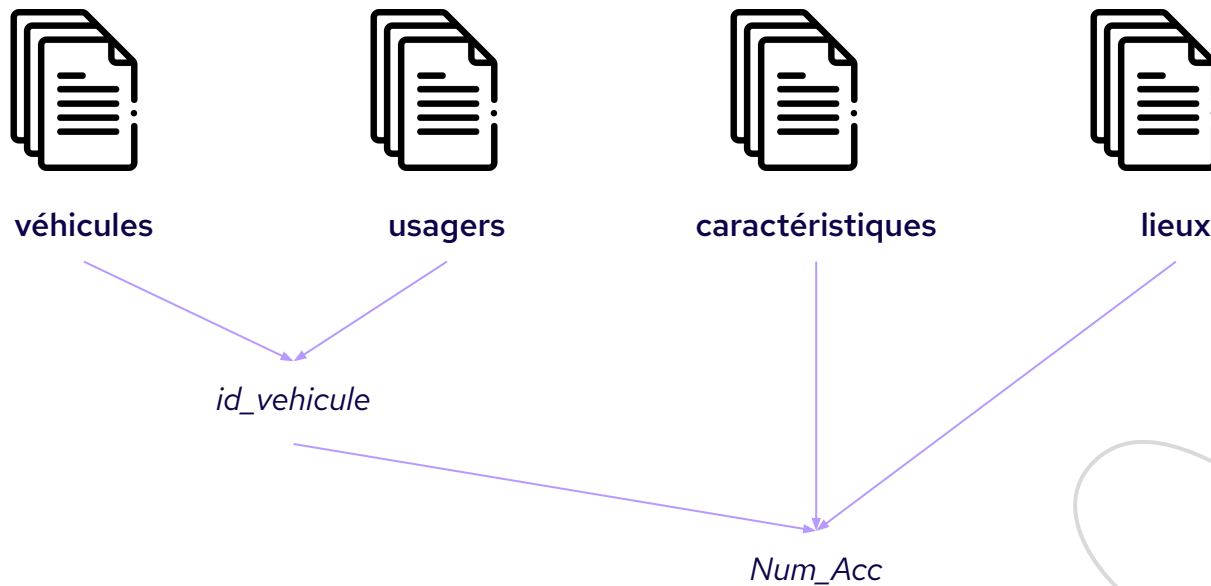
04 Binarisation de la cible

05 Idées d'améliorations

01

Traitement des données

Fusion des BD



→ **127k usagers**

Suppression de features

→ 54 features

- ❖ Suppression des **IDs** : **-4**
- ❖ Suppression des features de **date** et **lieu** : **-14**
- ❖ Suppressions de celles avec trop de valeur manquantes : **-3**
 - *occutc* (98.6%)
 - *lartpc* (99.95%)
 - *larroul* (93.5%)
- ❖ Suppression de *trajet* (25%) : **-1**
- ❖ Suppression des features intrinsèques aux piétons (93%) : **-3**

→ 29 features

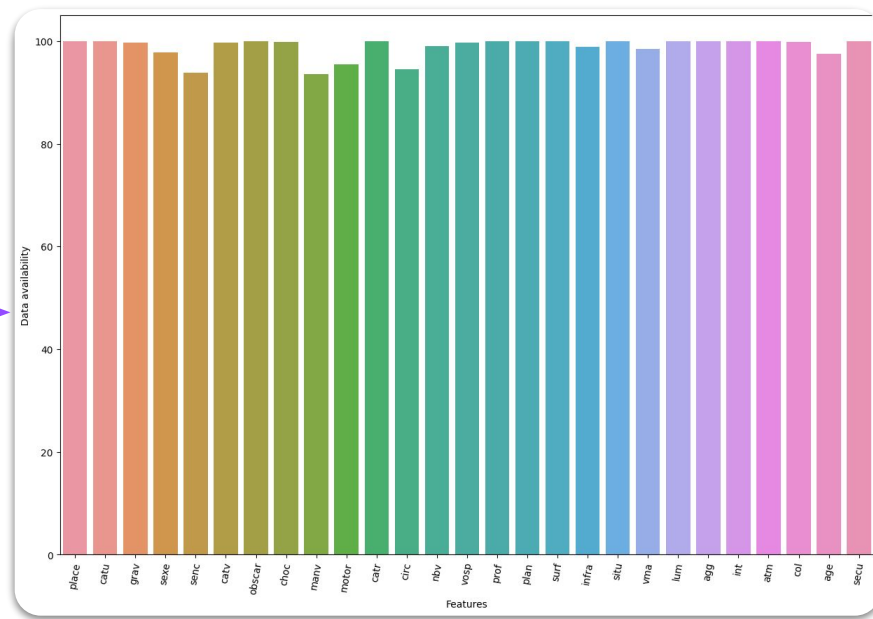
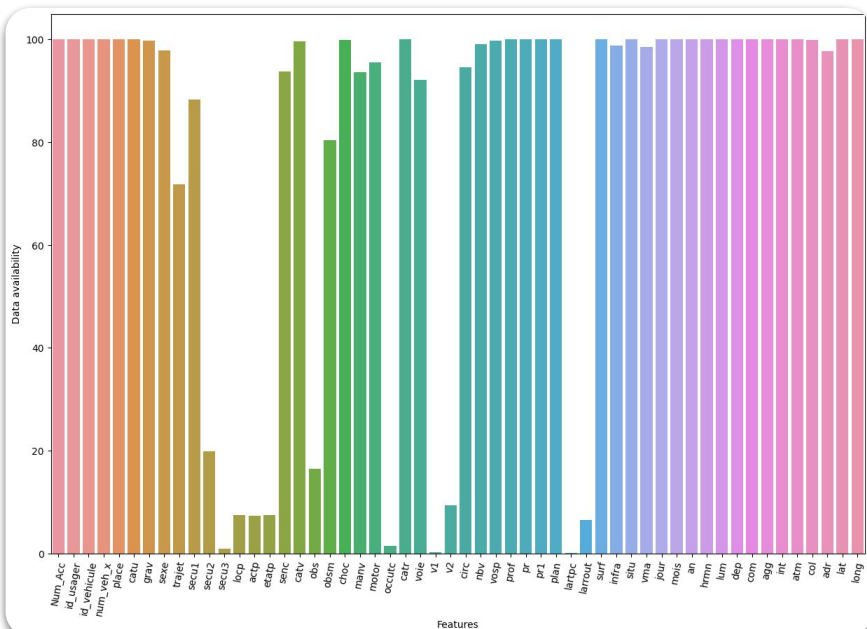
Remplacement de features

→ **29 features**

- ❖ Pour réduire le taux de données manquantes :
 - Remplacement de *obs* et *obsm* par *obscar* qui les regroupe : **-1**
 - Remplacement de *secu1*, *secu2* et *secu3* par *secu* qui compte le nombre d'équipements : **-2**
- ❖ Remplacement de *an_nais* par *age* pour mieux représenter les valeurs extrêmes

→ **26 features**

Taux de remplissage des features



Formatage des données

- ❖ Peu d'**outliers** (données catégorielles)
- ❖ Remplacement des **lettres** par des nombres
- ❖ **Ajout de 1** à toutes les valeurs de :
 - *infra, choc, nbv* et *vosp*

Pour maintenir une sémantique cohérente (-1 ou 0 : manquant)
et pour différencier le **manque d'info** de l'**info négative** (ex : *nbv*)

Filtrage des données

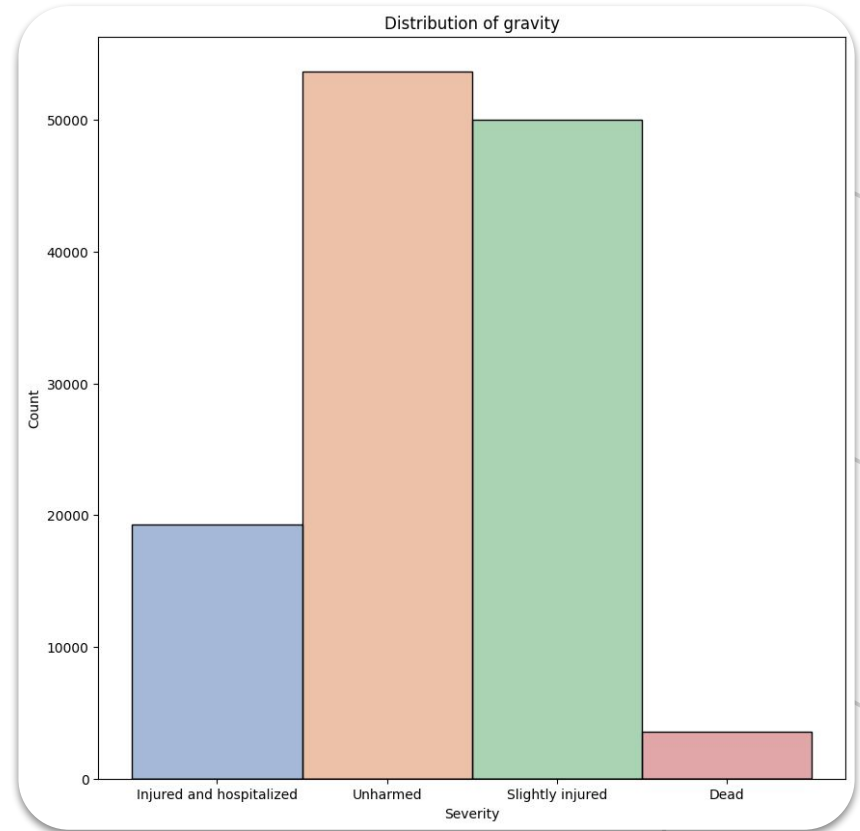
On conserve l'utilisateur si aucune valeur manquante :

127k usagers → 98k usagers (vs 75k avec *trajet*)

54 features → 26 features

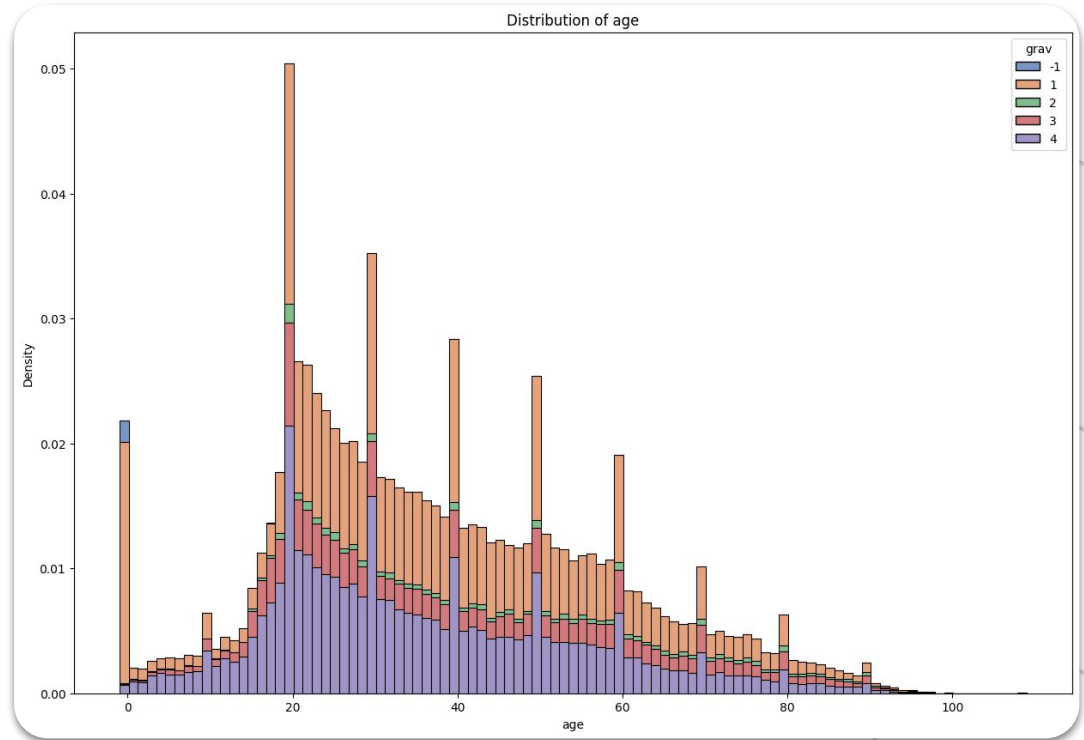
Exploration des données

La **cible** *grav* présente 4 classes très **déséquilibrées**



Exploration des données

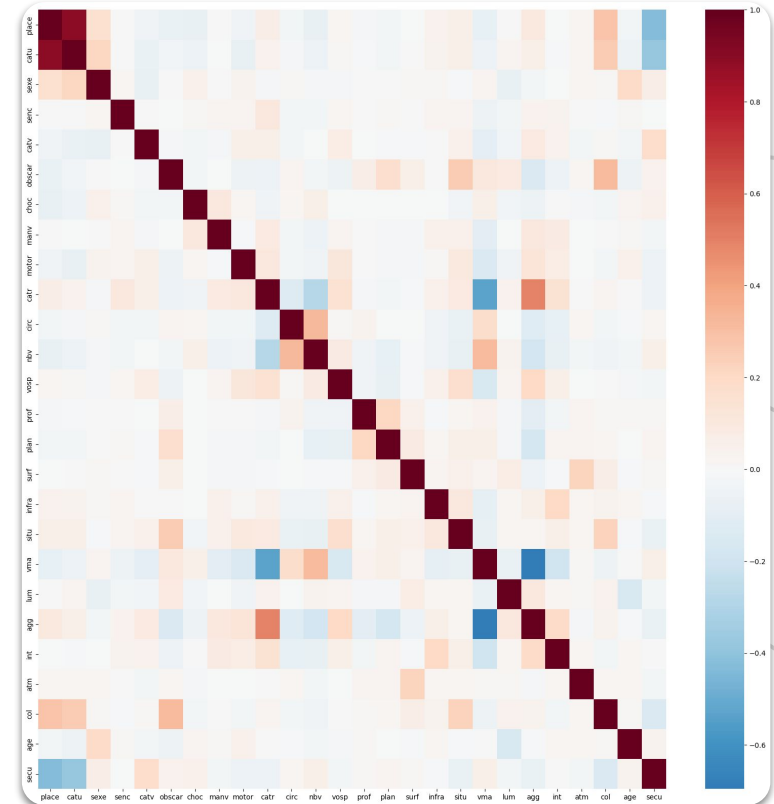
age présente une distribution étrange



Réduction de dimension

Les 26 features conservées contiennent quelques **corrélations** assez importantes :

- ❖ *secu, place* et *catu*
- ❖ *vma, agg, nbv* et *catr*
- ❖ ...



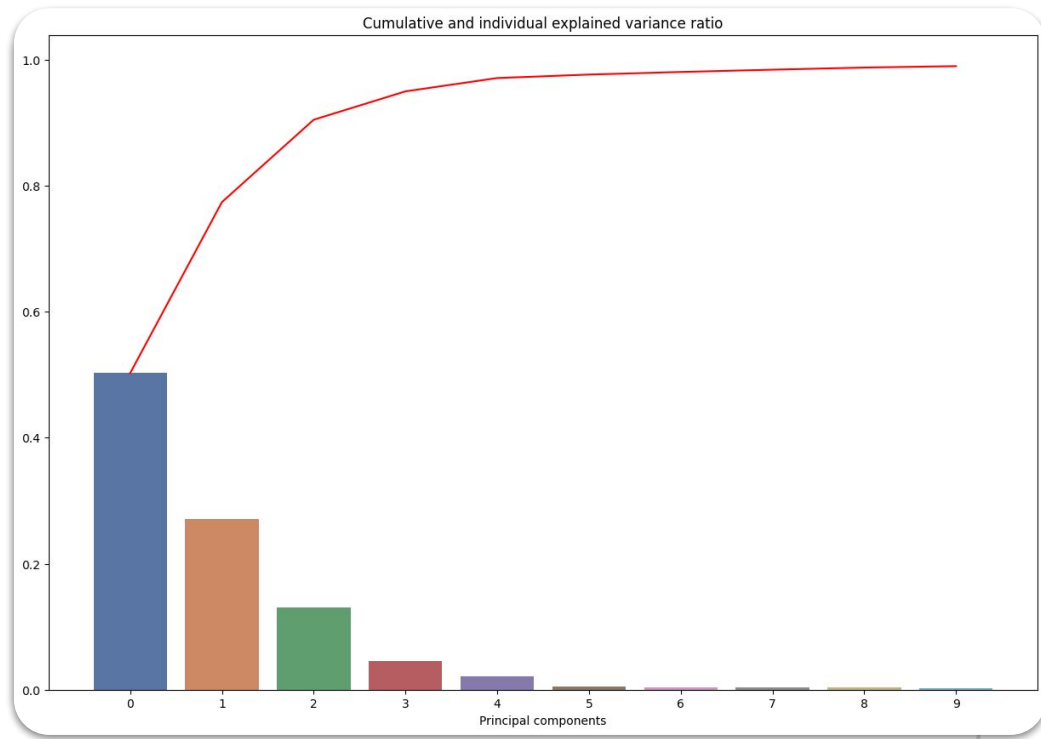
Réduction de dimension

Pour éliminer la redondance et réduire le temps de calcul : **PCA** (ou t-SNE ou UMAP)

→ **5 composantes** suffisent à expliquer **99+%** de la variance

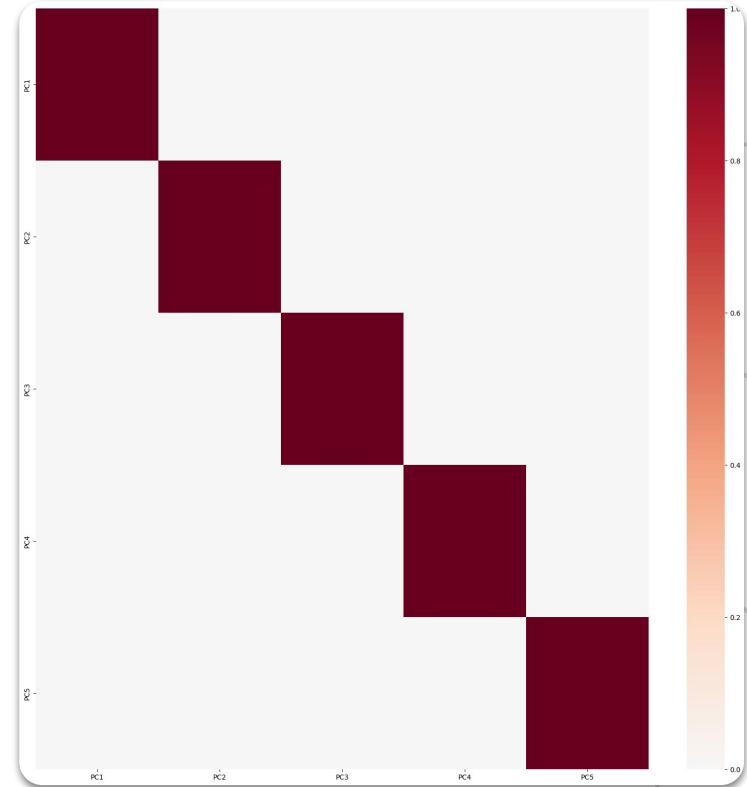
→ chaque composante est composée à plus de 95% par une seule feature :

- *vma*
- *age*
- *catv*
- *manv*
- *obscar*

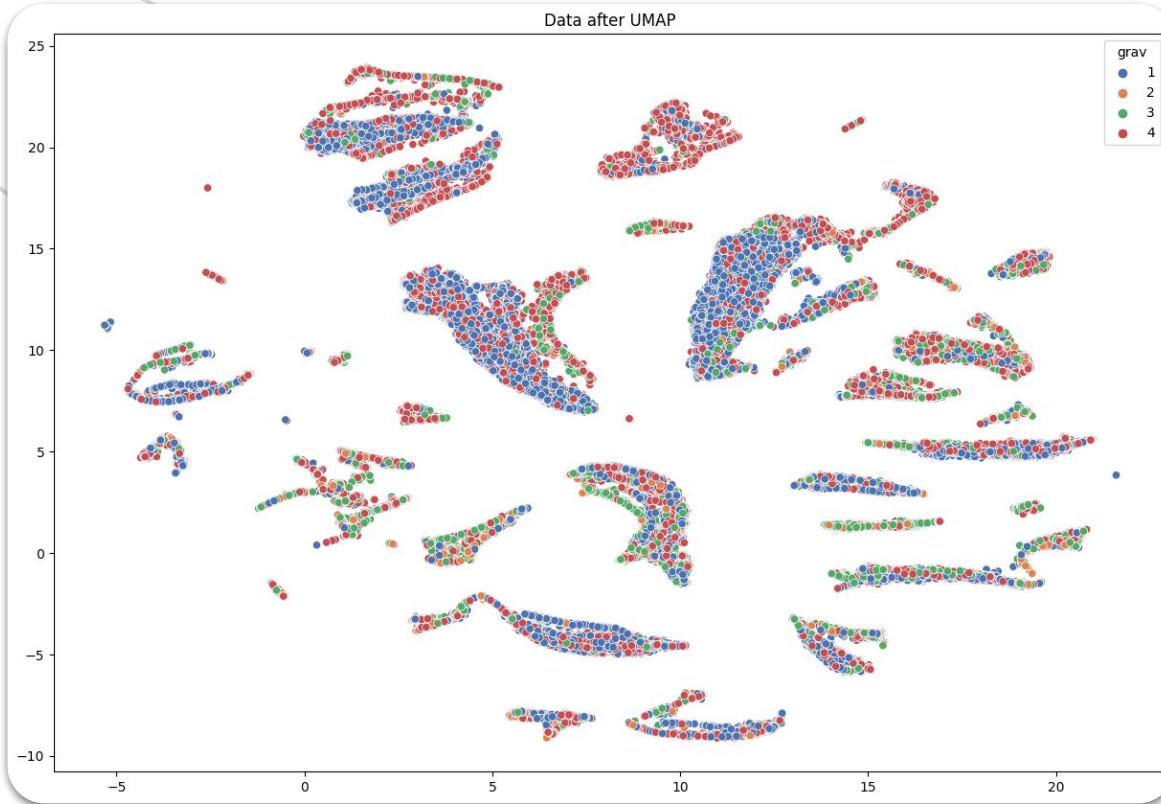


Réduction de dimension

→ la **corrélation** entre les 5 composantes principales est bien **nulle**



Séparation en dimension inférieure



→ **en 5D**, après PCA :
silhouette = -0.02
(min -1, max 1)

En résumé

→ 3 ensembles de features pour un même usager

- ❖ 26 features extraites/construites à partir du dataset
 - le plus proche des données originales
- ❖ 5 composantes principales expliquant 99+% de la variance
 - autant d'info que dans les 26 features, moins de calculs
- ❖ 5 features du dataset choisies par PCA (ou XGBoosting)
 - les plus importantes, facilite l'interprétation

02

Entraînement des 1ers modèles

Modèles sélectionnés

→ 5 modèles de familles différentes

- ❖ RandomForestClassifier
 - Forêt aléatoire
- ❖ GaussianNB
 - Réseau bayésien gaussien (inférence probabiliste)
- ❖ LogisticRegression
 - Régression logistique
- ❖ KNeighborsClassifier
 - Vote des k plus proches voisins
- ❖ MLPClassifier
 - Perceptron multi-couches (réseau de neurones)

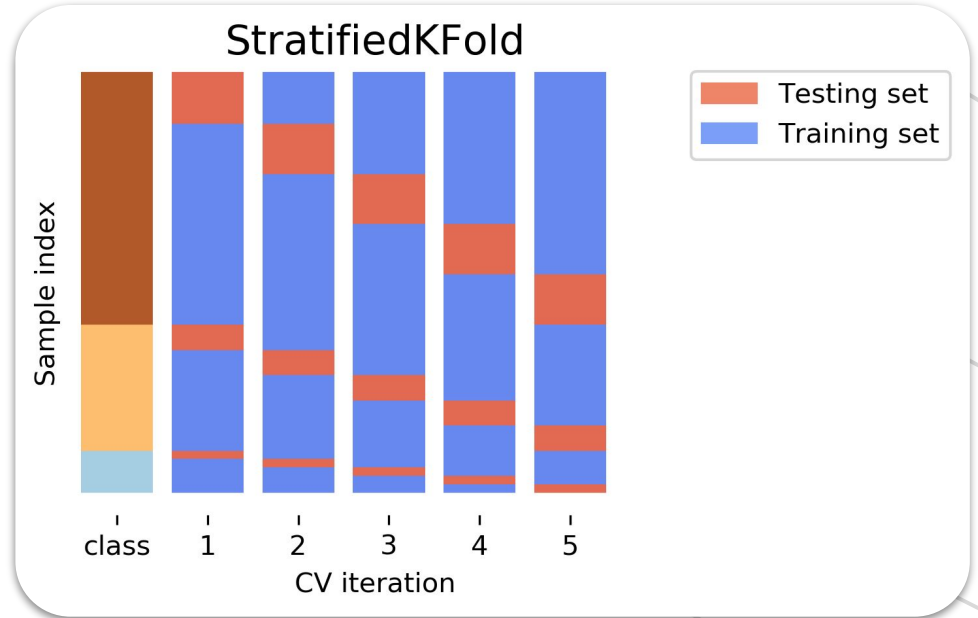
Recherche d'**hyperparamètres**

- **GridSearch** sur quelques HP bien choisis
 - seulement quelques valeurs pour chaque HP
 - possibilité de RandomSearch pour affiner le choix

Entraînement des modèles

→ Cross-validation :

- afin de réduire l'effet du hasard
- 5 folds (80% train + 20% test)
- proportions des classes conservées

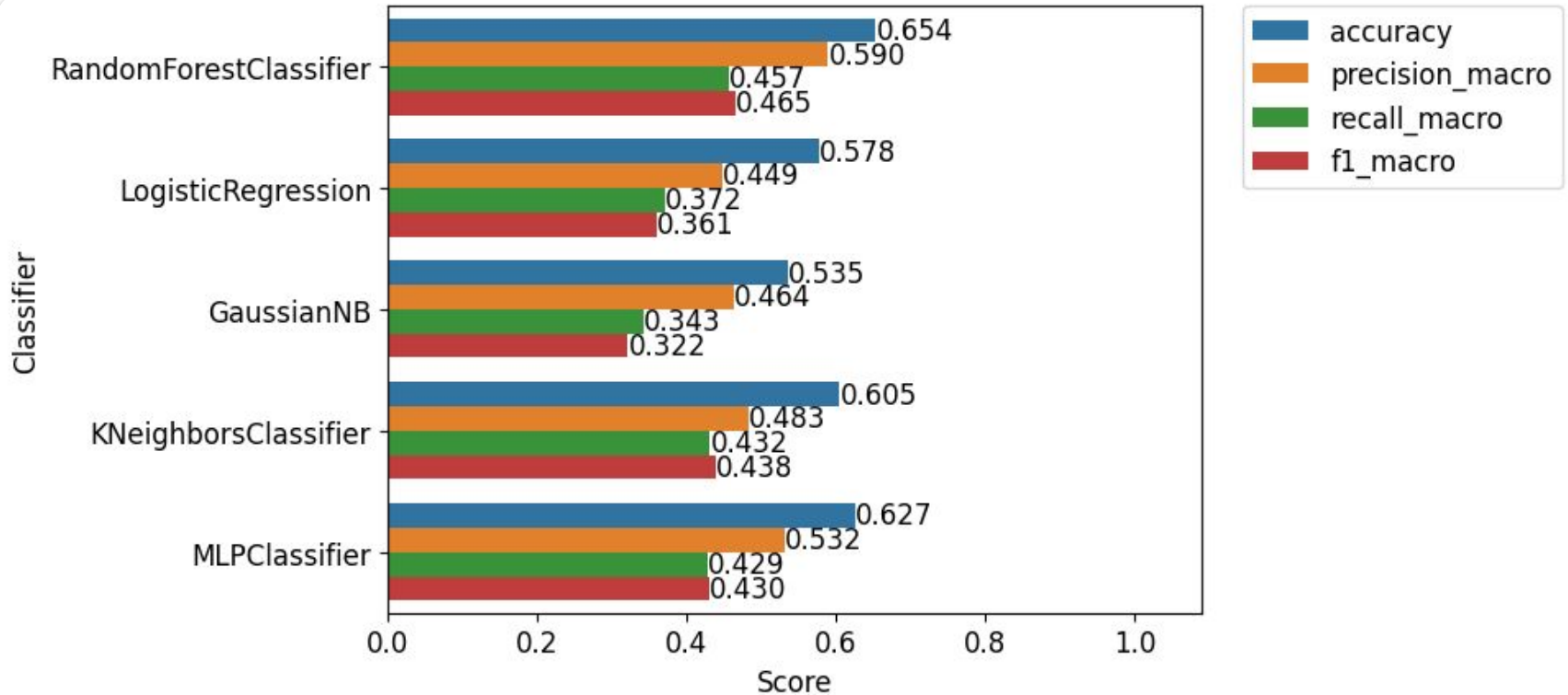


Métriques de performance

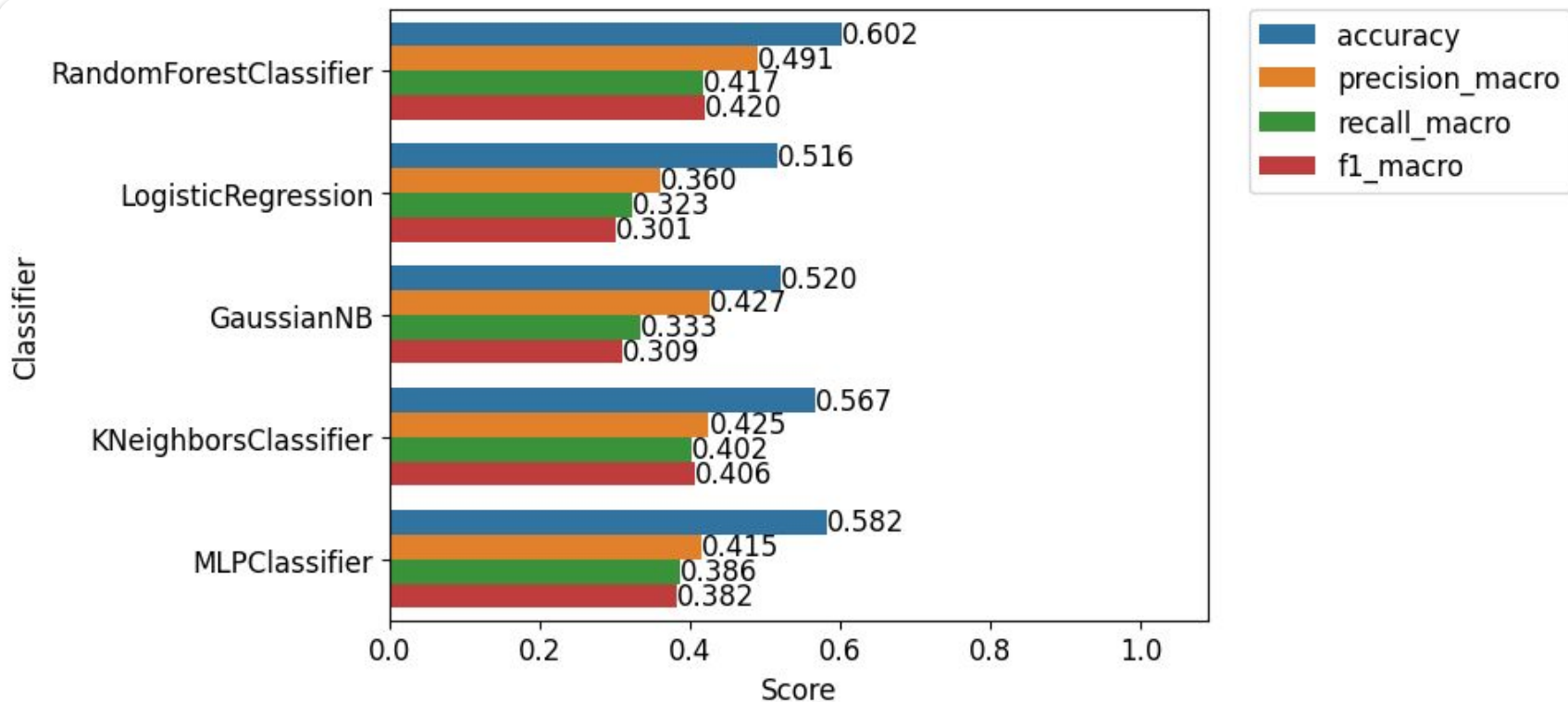
→ 4 métriques

- ❖ Accuracy
 - permet de traquer la performance globale
- ❖ Precision (macro)
- ❖ Recall (macro)
- ❖ F1-score (macro)
 - permettent de prendre en compte le déséquilibre des classes

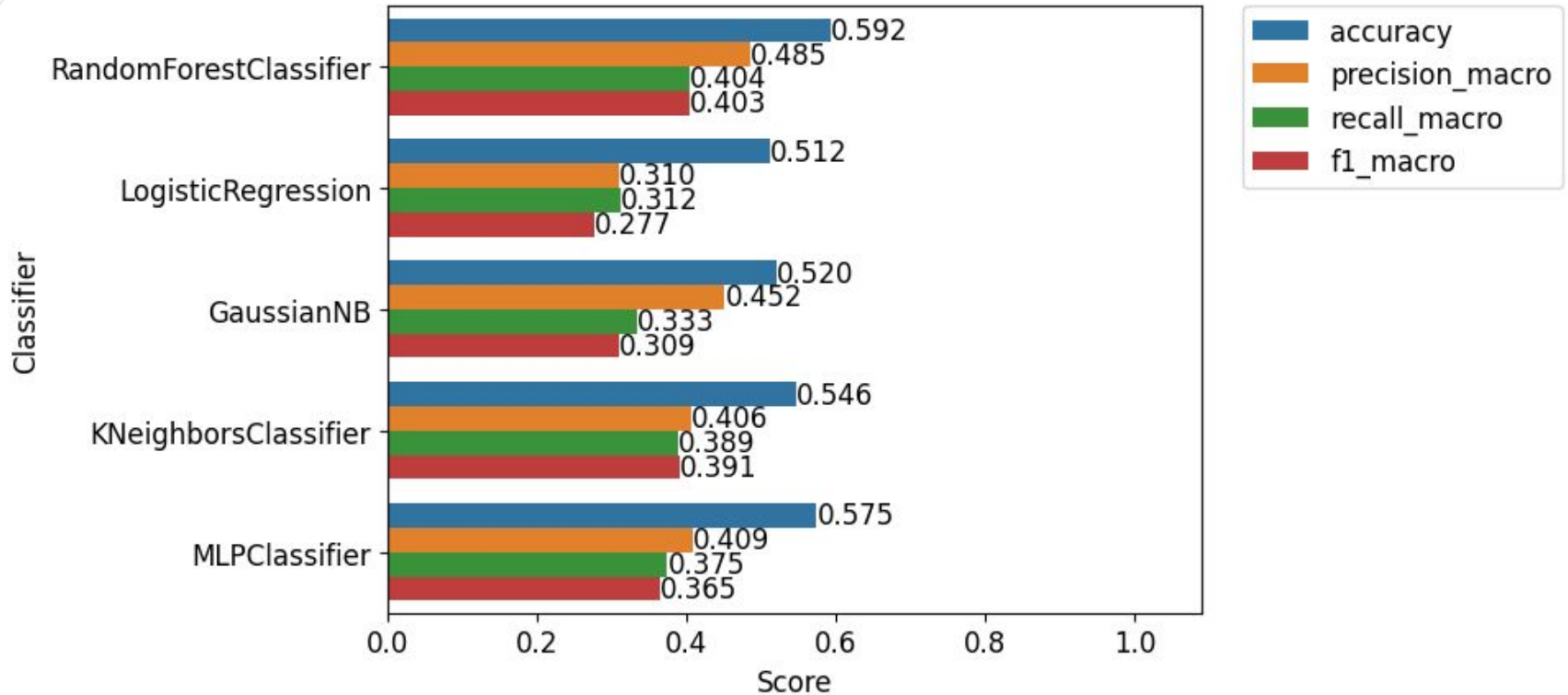
Résultats 26 features



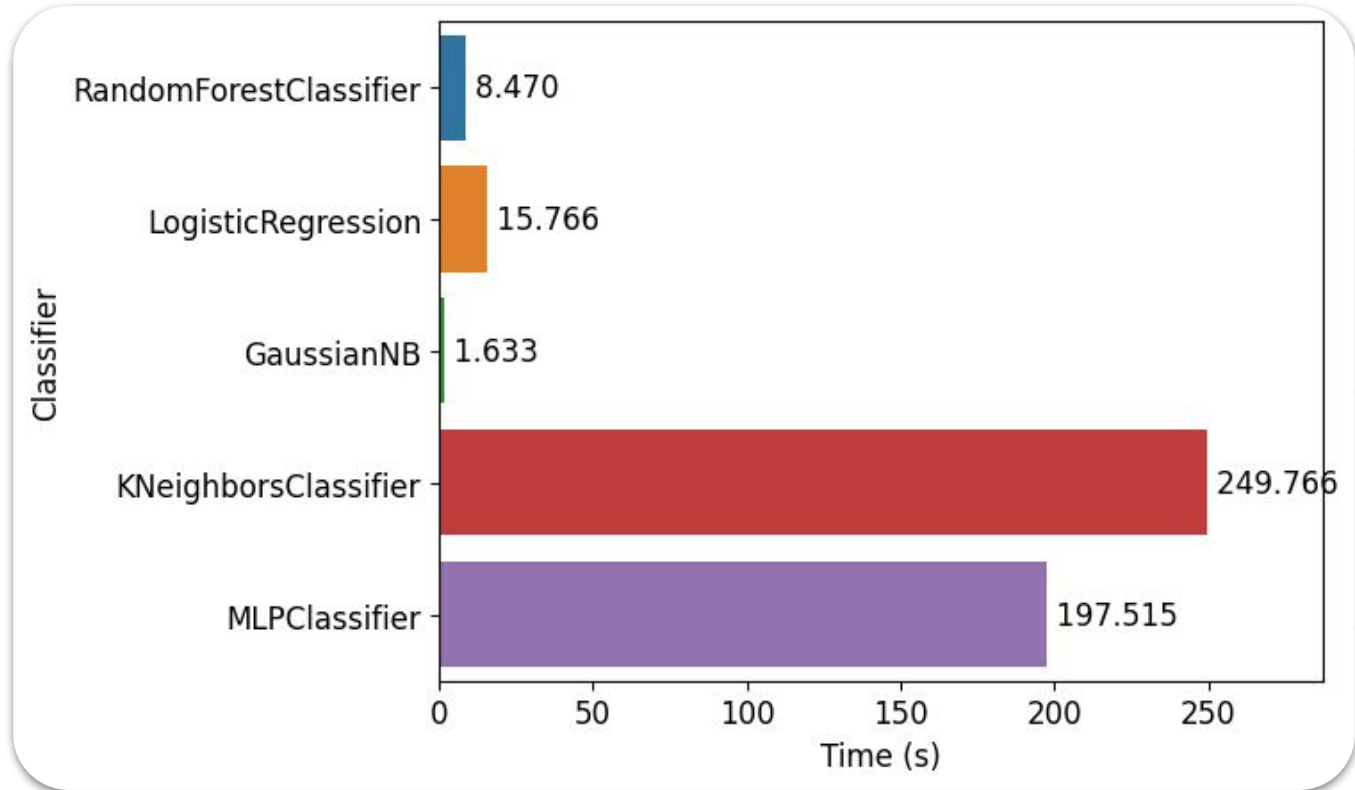
Résultats 5 PCA



Résultats 5 features principales

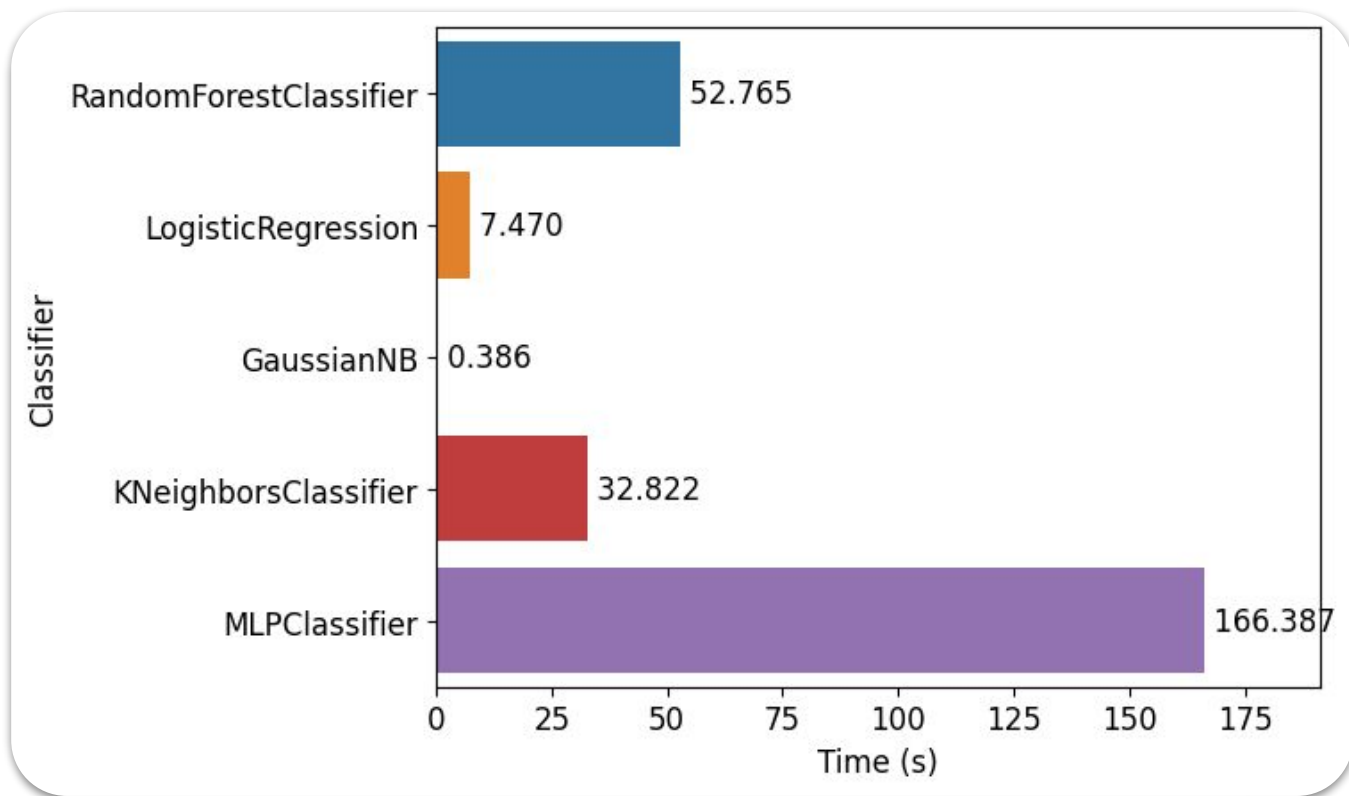


Temps 26 features



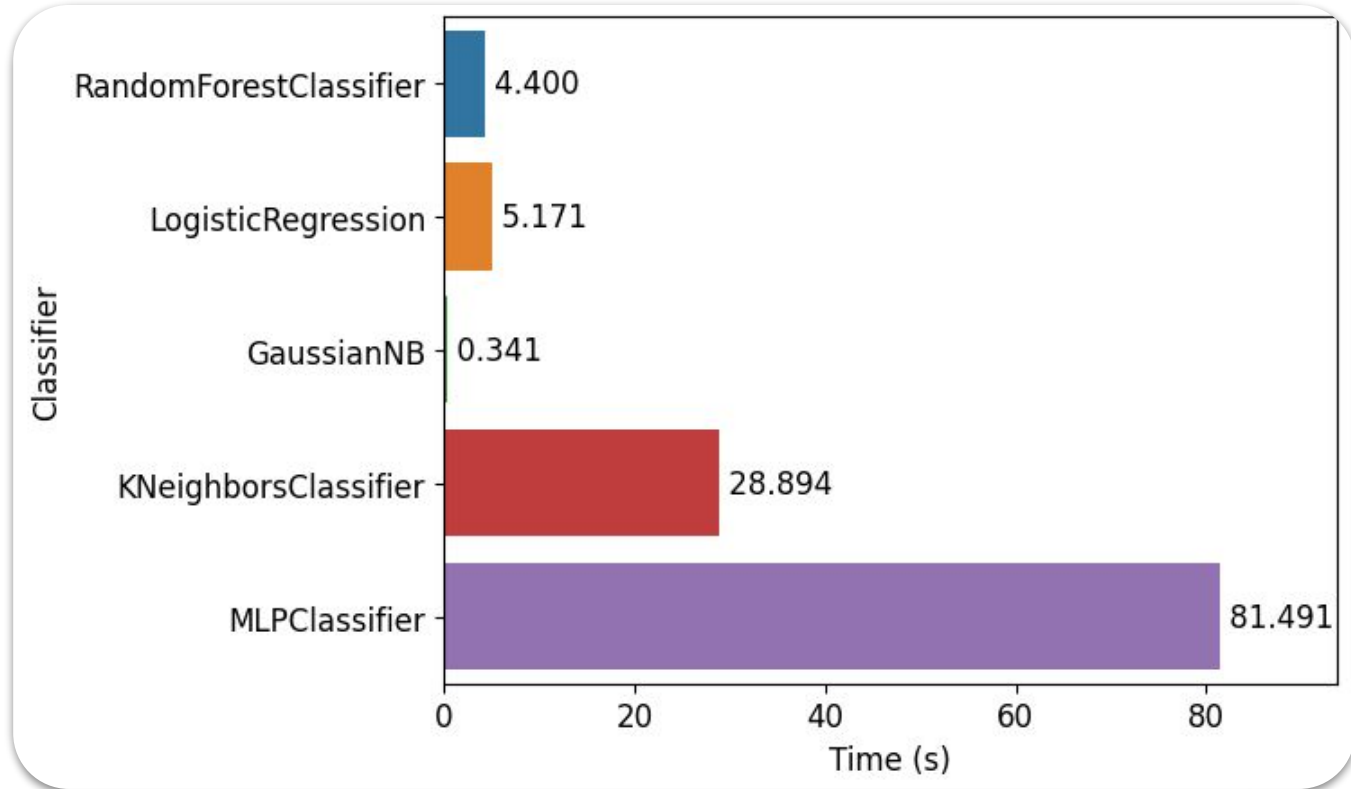
Temps 5 PCA

→ KNN --
→ RandomForest ++



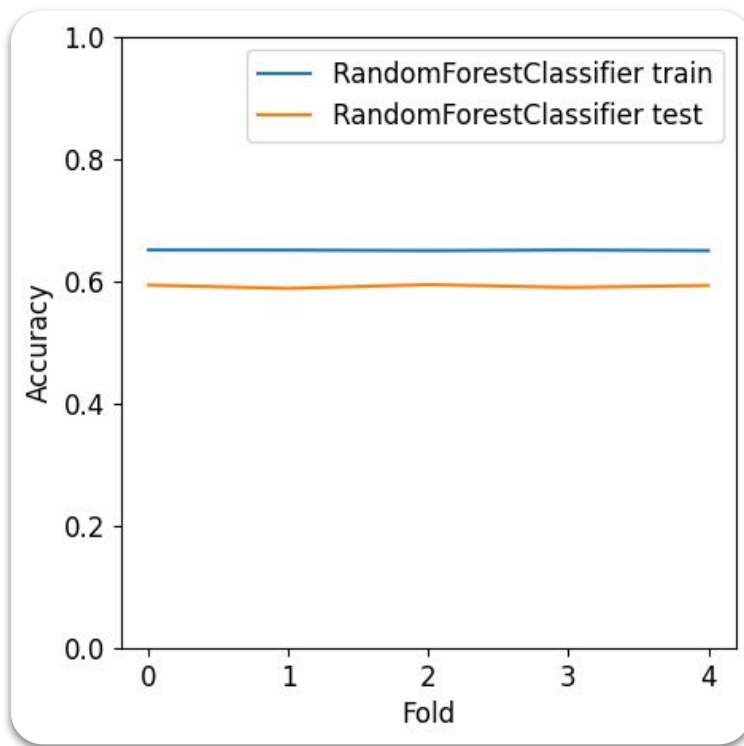
Temps 5 features principales

→ - pour tous



Surapprentissage ? (non)

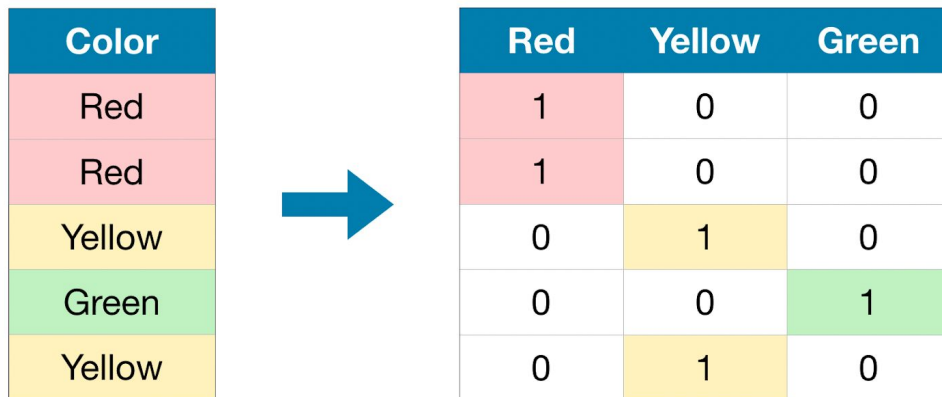
→ accuracy de train \approx accuracy de test



03

Représentation one-hot

One-hot



The diagram illustrates the one-hot encoding process. On the left, a table with a 'Color' column contains five rows: 'Red', 'Red', 'Yellow', 'Green', and 'Yellow'. A blue arrow points to the right, where a new table is shown. This table has three columns: 'Red', 'Yellow', and 'Green'. Each row in the new table corresponds to a row in the original table, with a '1' in the column matching the color and '0' in the others. For example, the first row (Red) has a '1' under 'Red' and '0' under 'Yellow' and 'Green'.

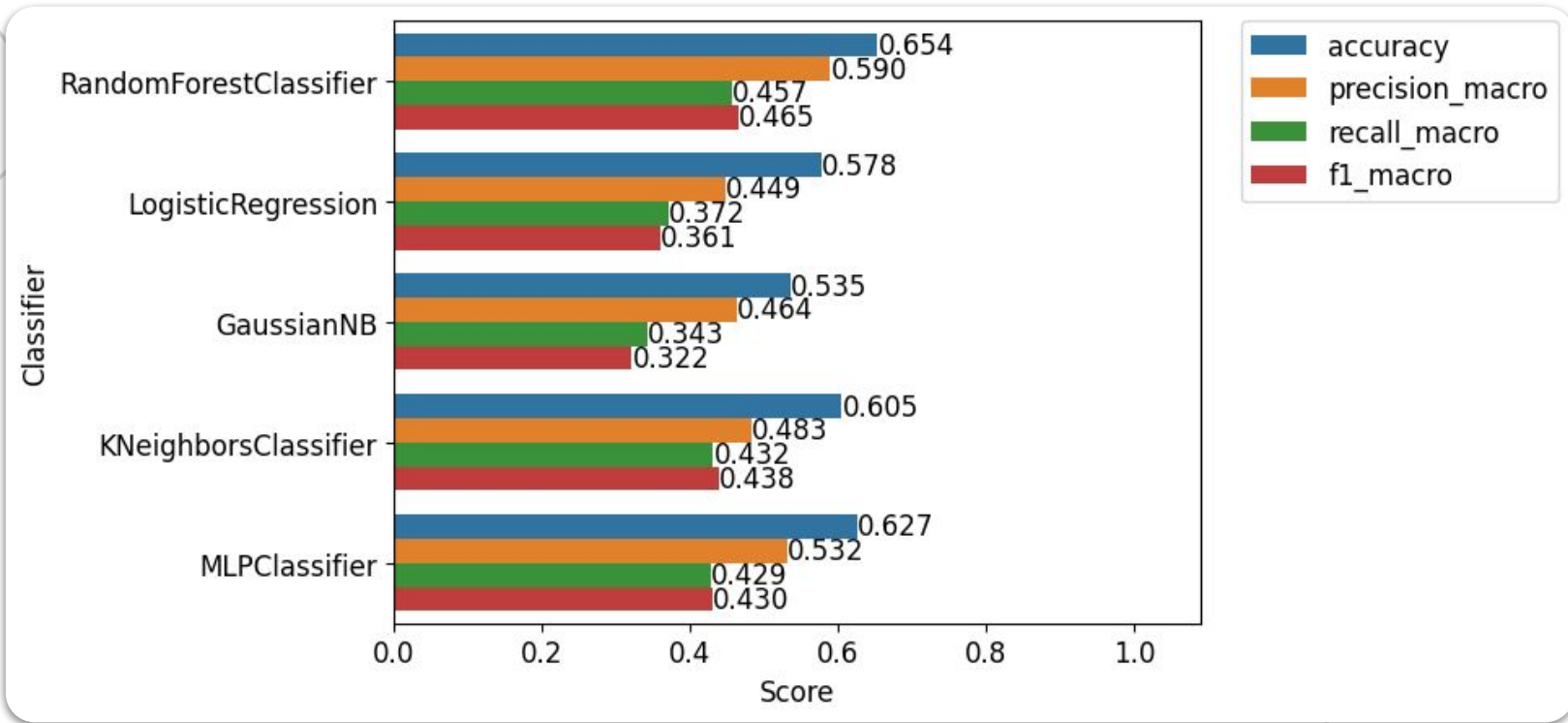
Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

- création d'autant de features qu'il y a de valeurs possibles pour la feature originale
 - **suppression de l'ordre** entre les catégories
 - meilleures performances mais plus de calculs

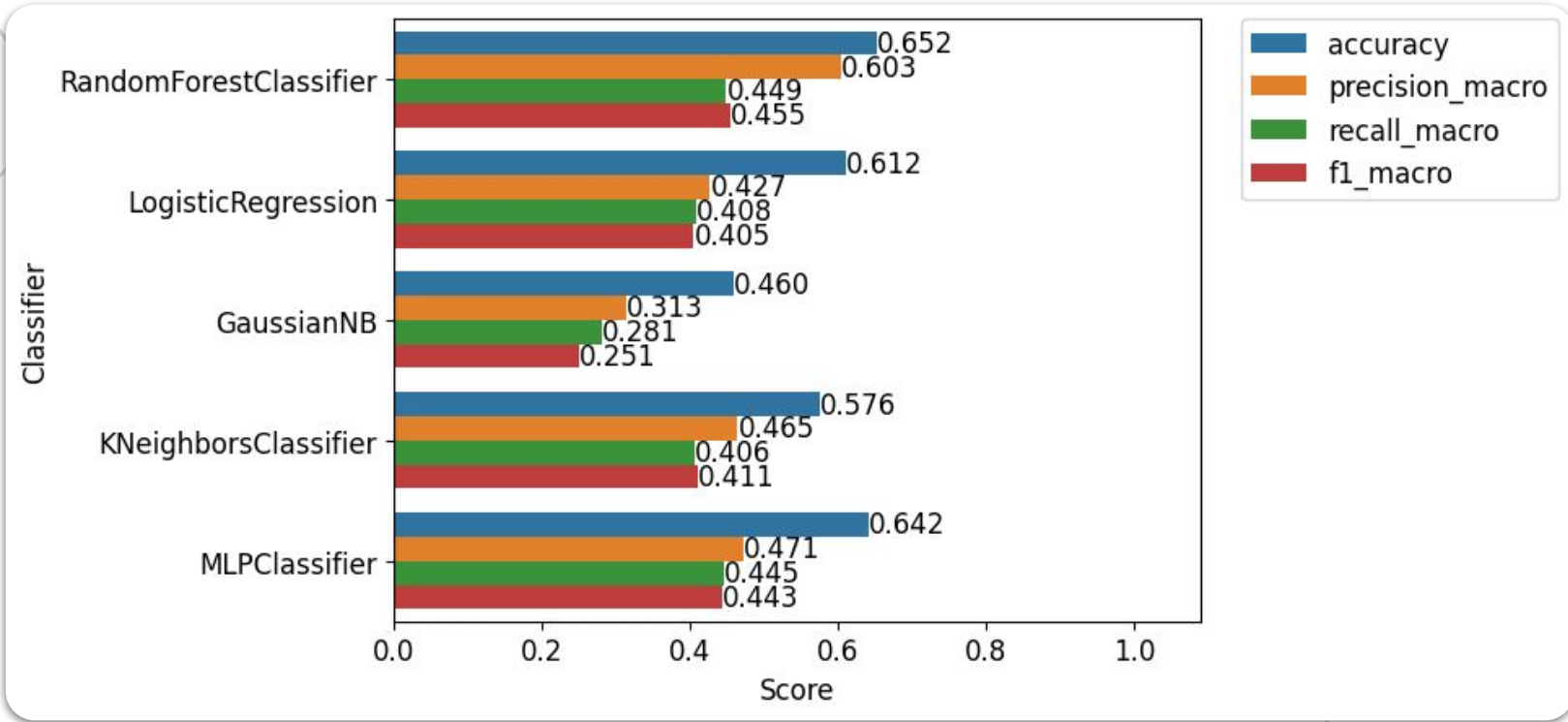
26 features → 200 features

Résultats 26 features

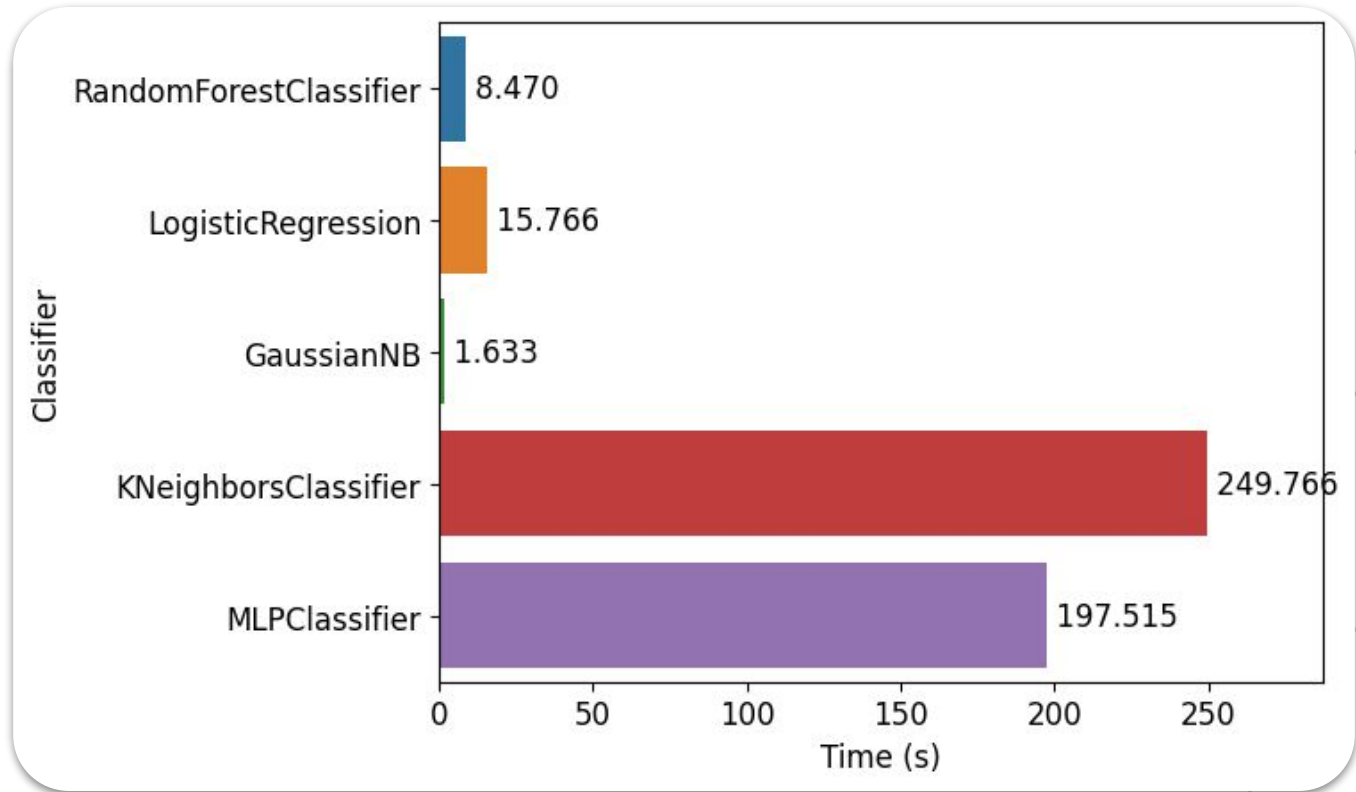


Résultats 200 one-hot features

→ NB -- car
plus d'indé

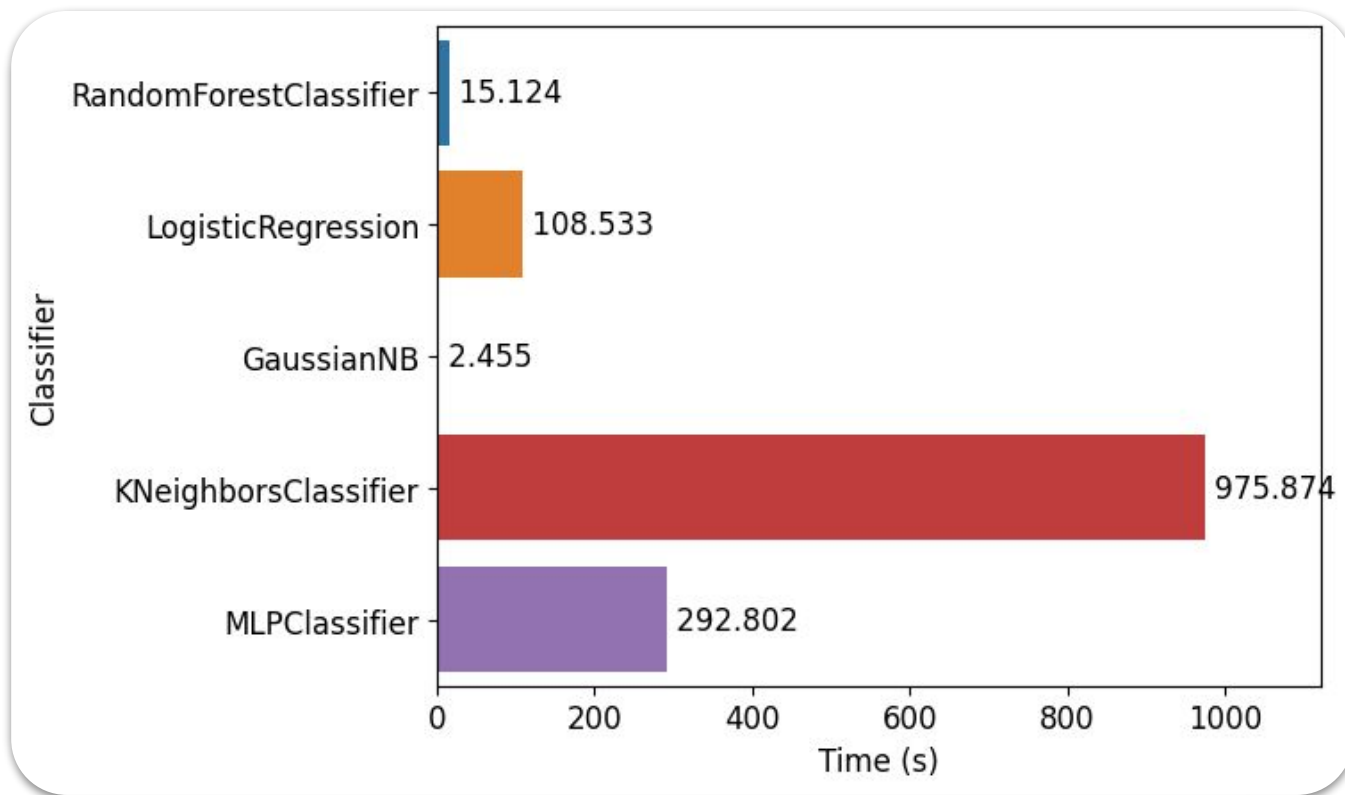


Temps 26 features



Temps 200 one-hot features

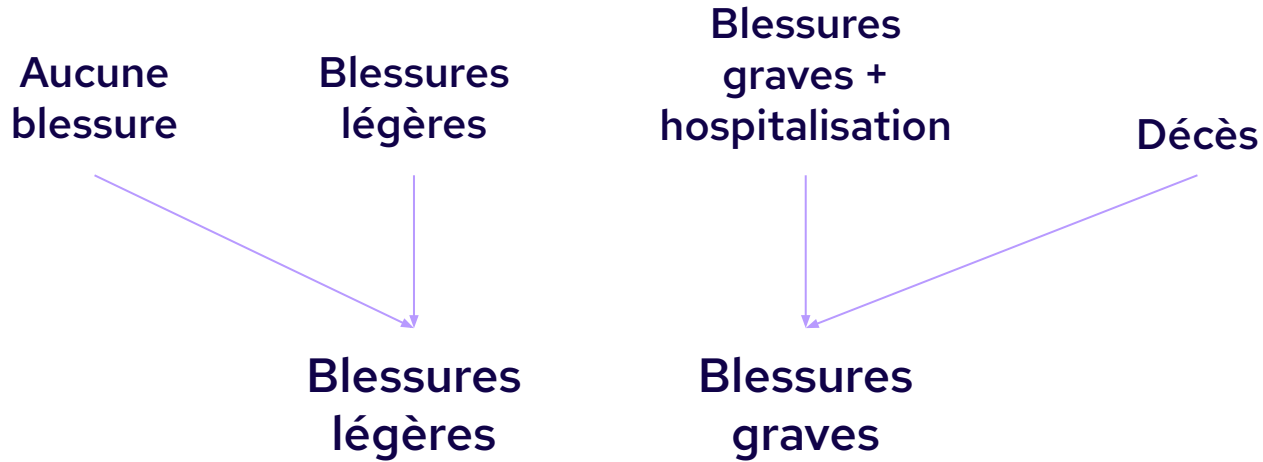
- Regression +
- MLP +
- KNN +++



04

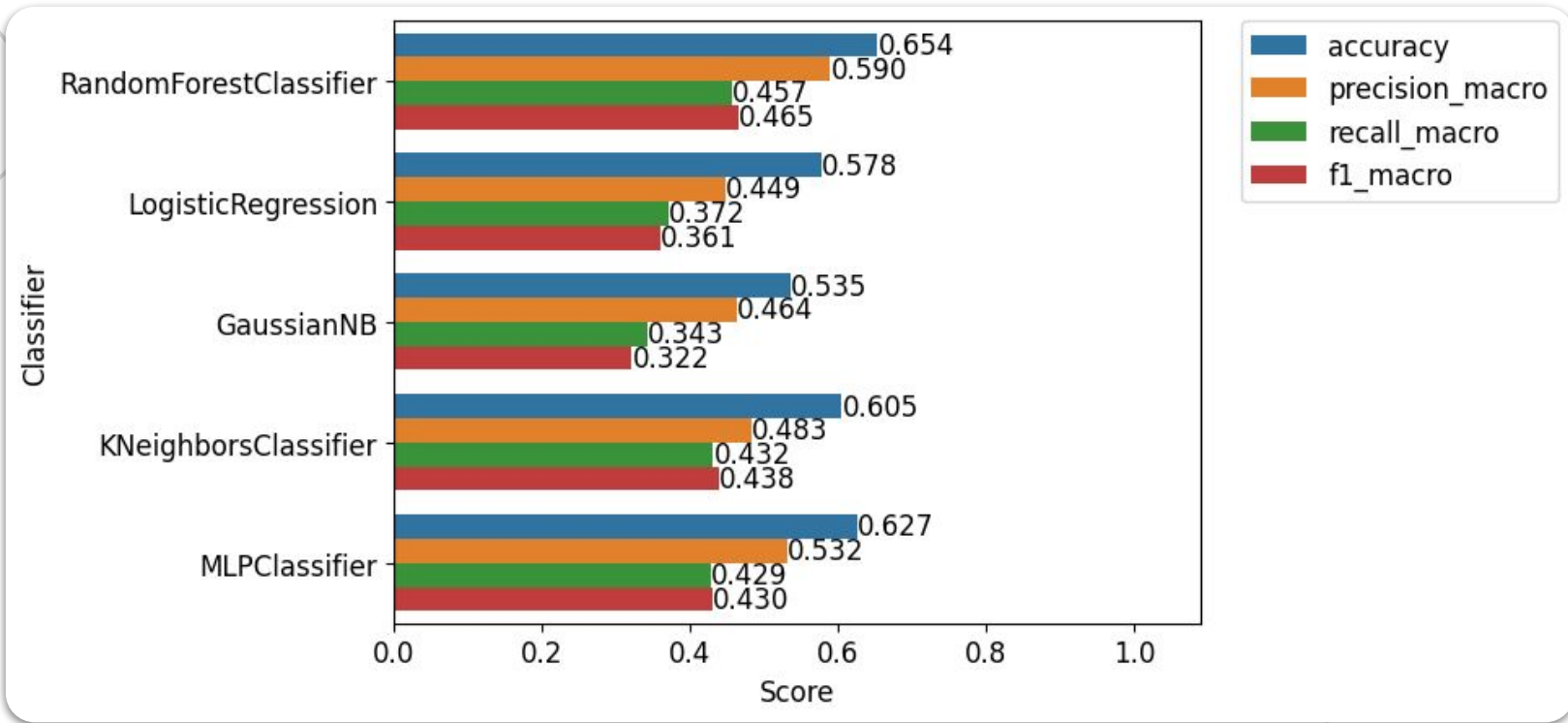
Binarisation de la cible

2 classes au lieu de 4



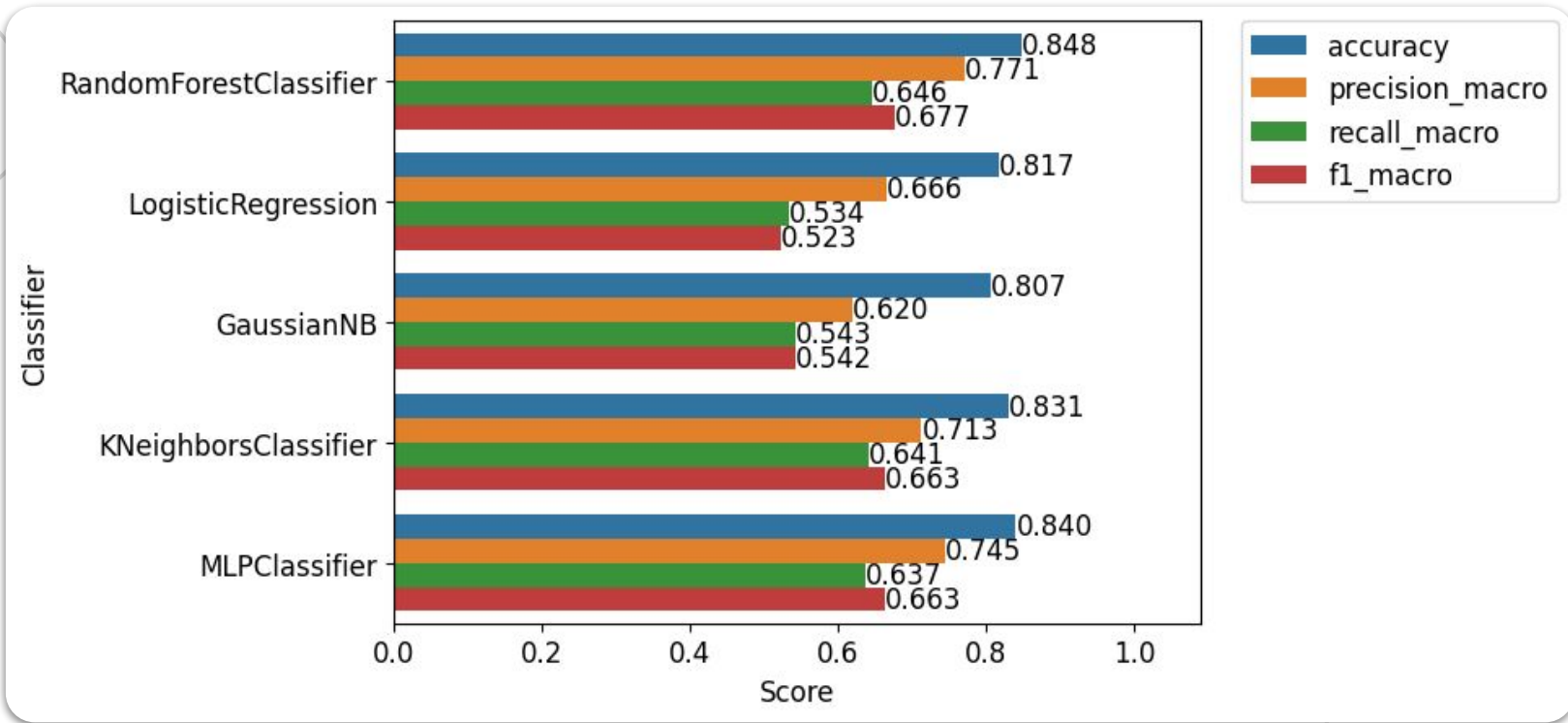
- Réduction du **déséquilibre** des classes
- Permet de voir si le modèle fait des **grosses erreurs**
- Réduit l'impact de l'ordre

Résultats 4 classes (26 features)

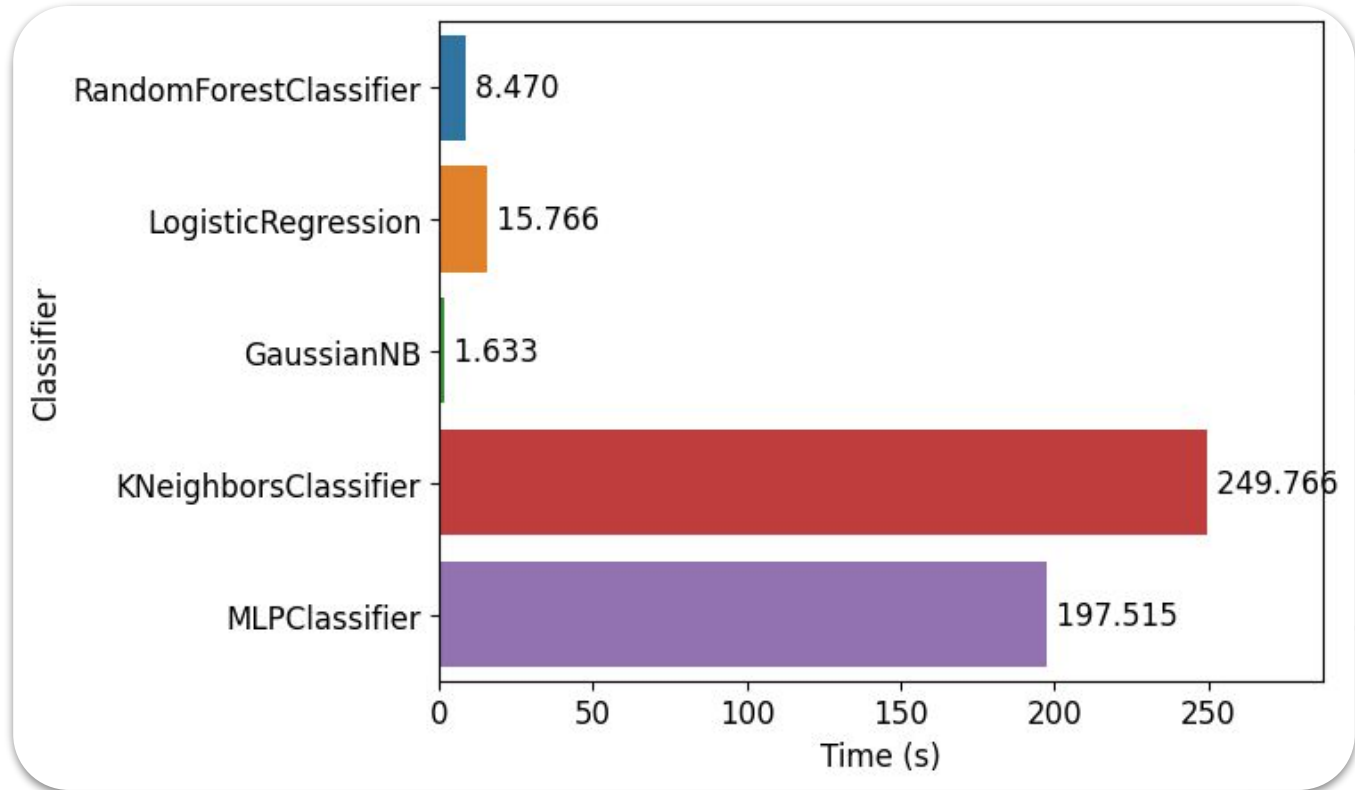


Résultats 2 classes (26 features)

→ tout ++

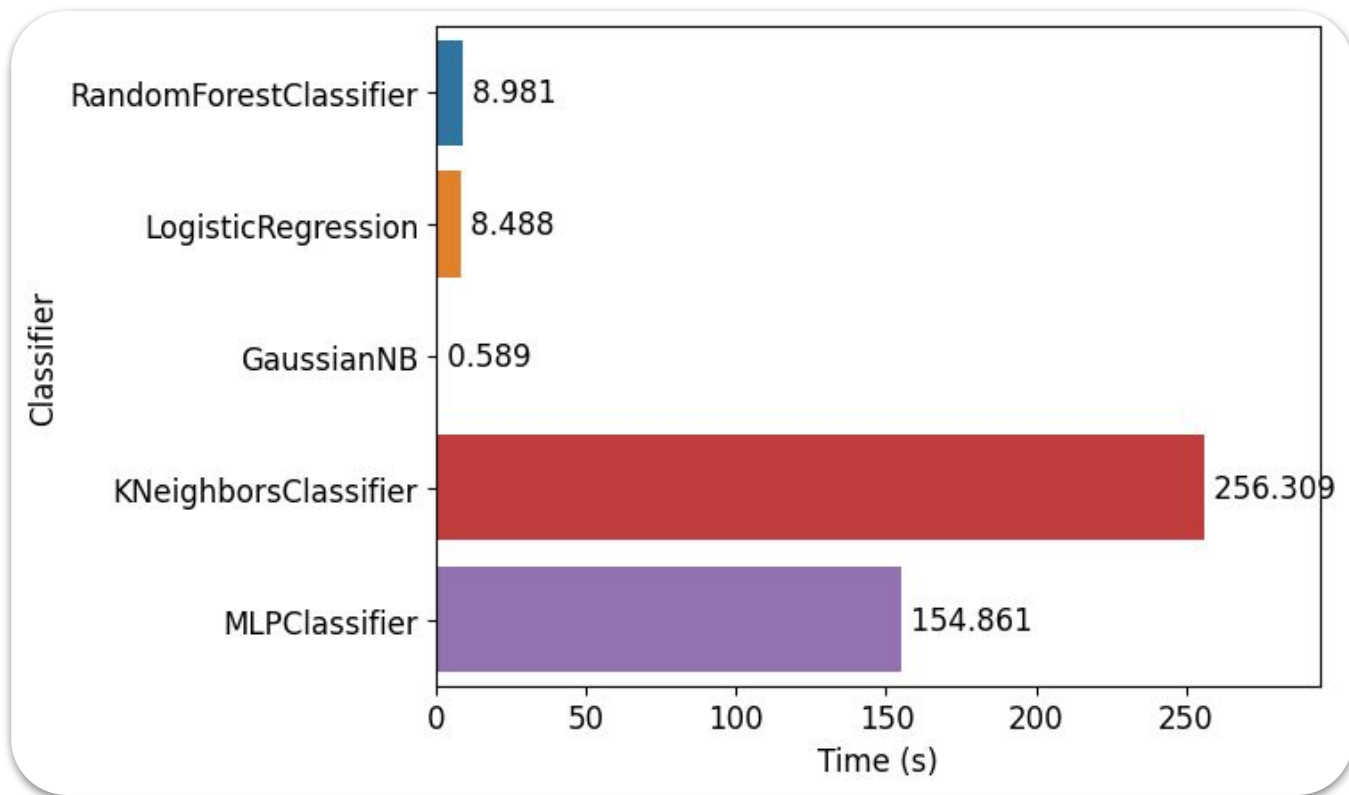


Temps 4 classes (26 features)



Temps 2 classes (26 features)

→ tout pareil

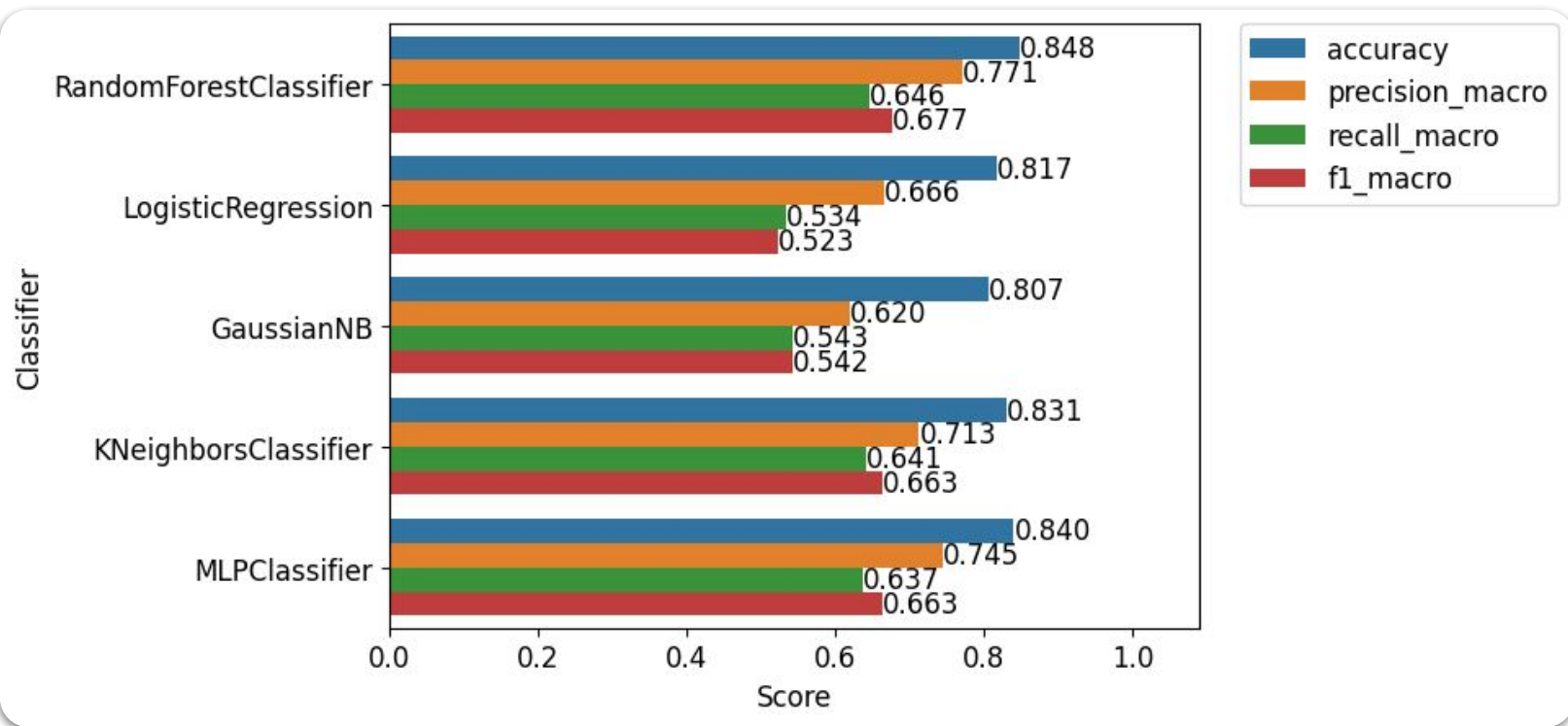


Suréchantillonnage

SMOTE (Synthetic Minority Oversampling TEchnique)

- création de nouveaux usagers pour équilibrer la classe minoritaire
- utilisation des usagers existants de la classe
- passage de 25% de l'effectif de l'autre classe à **75%**

Résultats 2 classes

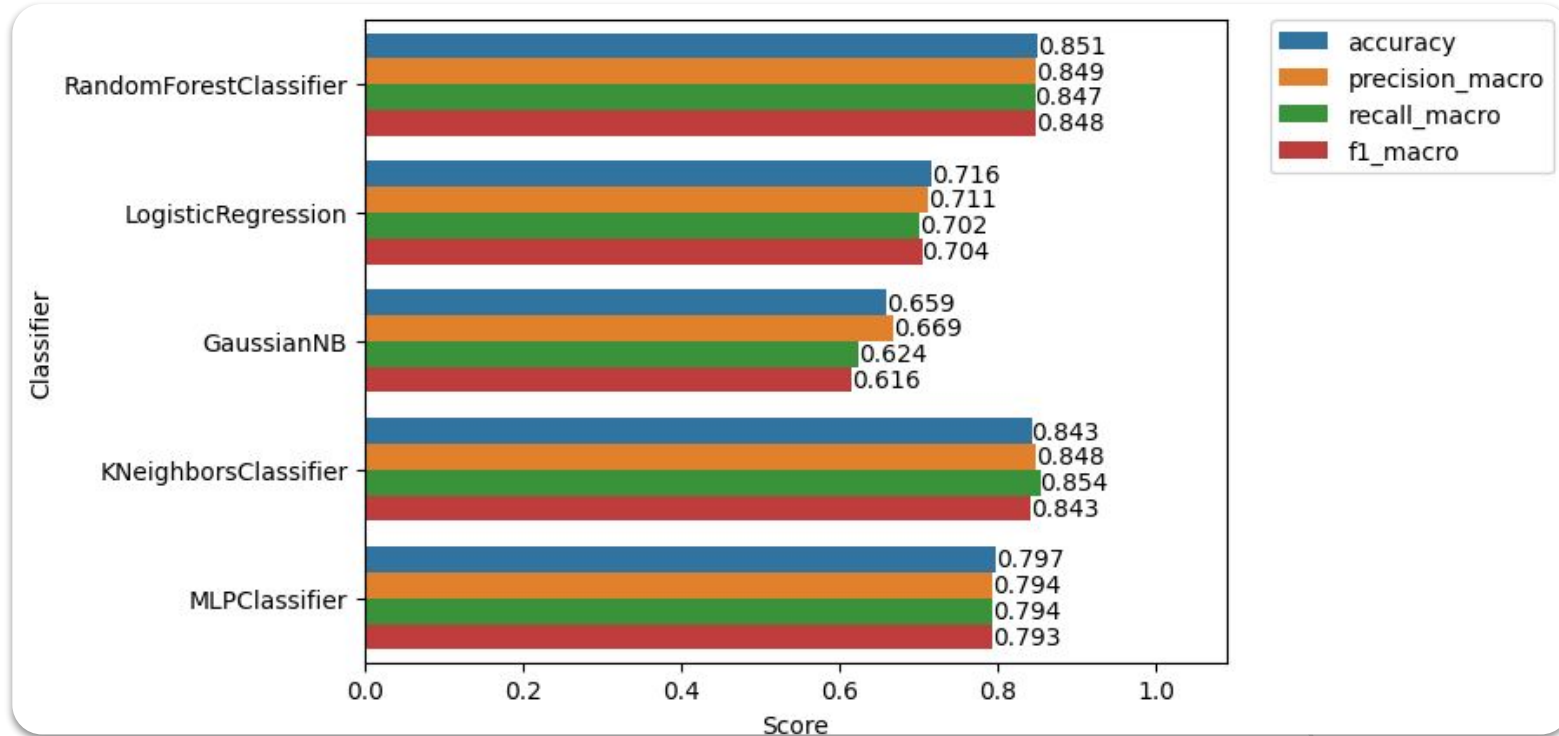


Résultats 2 classes suréchantillonnées

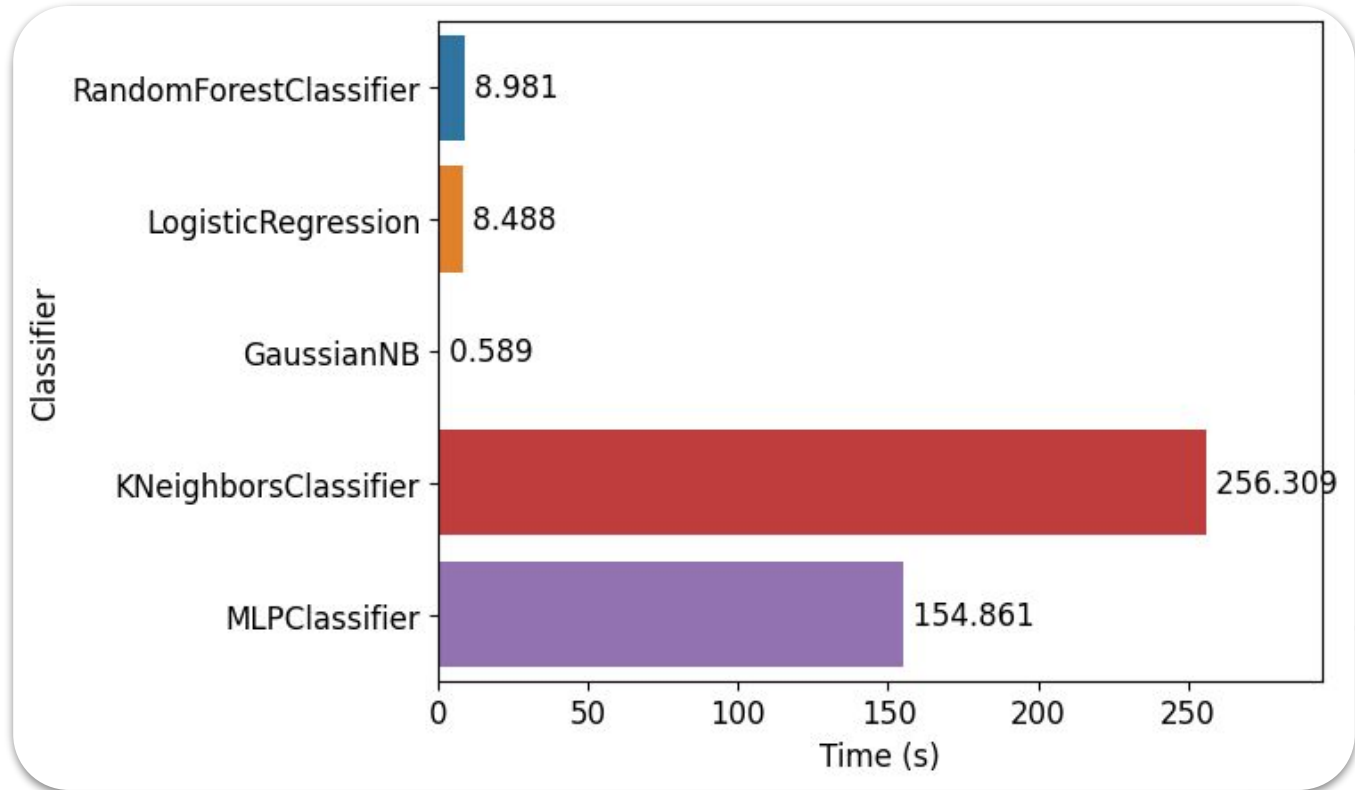
→ Regression -

→ NB --

→ Meilleures
perfs sur classe
minoritaire

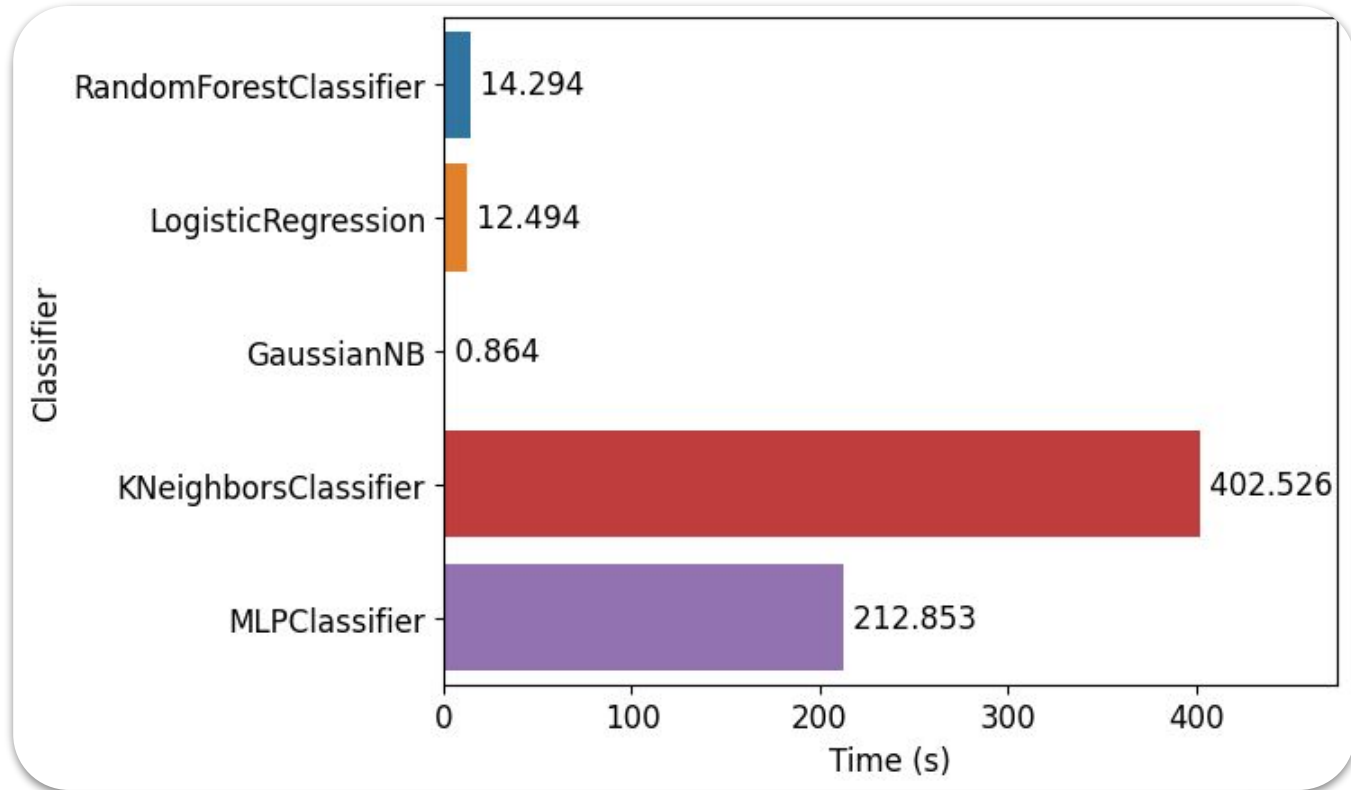


Temps 2 classes (26 features)



Temps 2 classes suréchantillonnées

→ KNN + car plus d'échantillons
(temps en $O(n^2)$)



05

Idées d'améliorations

Idées d'améliorations

- ❖ **Plusieurs modèles** pour les différentes situations

- 1 modèle pour les piétons
- 1 modèle pour les voitures seules
- 1 modèle pour les collisions de voitures

Permet de modéliser plus précisément l'accident :

- Utilisation des 3 features des piétons
- Utilisation d'infos sur les autres véhicules impliqués (ex : sens de circulation relatif, type de véhicule, etc.)
- Découplage des piétons renversés et de passagers, qui sont actuellement rattachés à la même voiture

- ❖ **Combinaison de modèles**

- ❖ Exploration plus exhaustive des HP avec **RandomSearch**

06

Annexes

