NOTE METHODOLOGIQUE

Projet

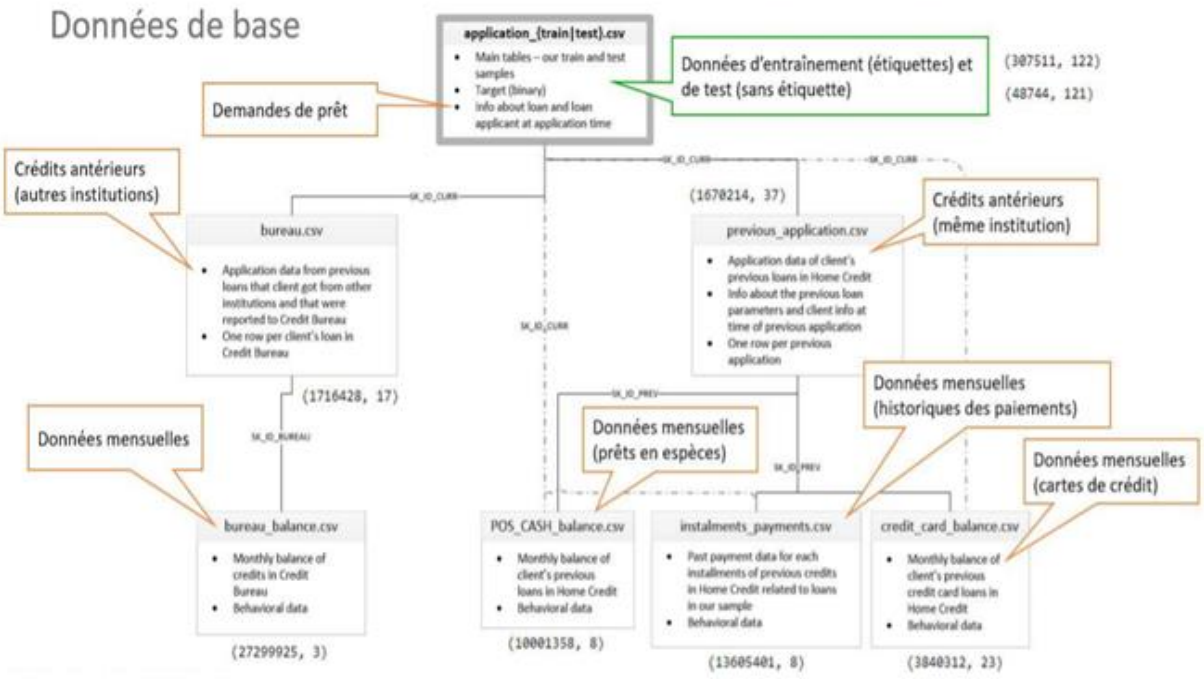
Implémenter un modèle de **scoring**

Contexte

Cette note constitue l'un des livrables du parcours Data Scientist d'Openclassrooms.

Il présente le processus de modélisation et d'interprétabilité du modèle mis en place dans le cadre du projet. Le projet consiste à développer pour la société « Prêt à Dépenser », une société de crédit à la consommation, un modèle de scoring de la probabilité de défaut de paiement d'un client avec pas ou peu d'historique de prêt.

Les données utilisées pour ce projet sont une base de données de 307 511 clients comportant 121 « features » (âge, sexe, emploi, logement, revenus, informations relatives au crédit, notation externe, etc.). La probabilité de classification est essentielle car nous voulons être très sûrs lorsque nous classons quelqu'un en tant que non-défaut, car le coût d'une erreur peut être très élevé pour l'entreprise.



NOTE METHODOLOGIQUE

Objectifs

Le modèle doit permettre de définir la probabilité de défaut de remboursement d'un crédit sur la base d'informations relatives au client.

Il doit également offrir un certain niveau de transparence concernant les données et leurs traitements en vue d'implémenter des méthodes d'interprétabilité des variables.

Modélisation

Utilisation de mlflow

MLflow est une plateforme open source de gestion du cycle de vie des modèles de machine learning, développée par Databricks.
Elle vise à simplifier et à organiser le processus de modélisation en fournissant des outils pour le suivi des expériences, la gestion des modèles, le déploiement et la centralisation de la documentation des modèles. MLflow est conçu pour fonctionner avec n'importe quelle bibliothèque d'apprentissage automatique, langage de programmation et environnement d'exécution, ce qui le rend extrêmement flexible.
Grâce à son composant MLflow Tracking, il est possible d'enregistrer automatiquement les paramètres, les métriques, les artefacts et même le code source de chaque exécution d'expérience, facilitant ainsi la comparaison entre différents runs et la reproduction des résultats.

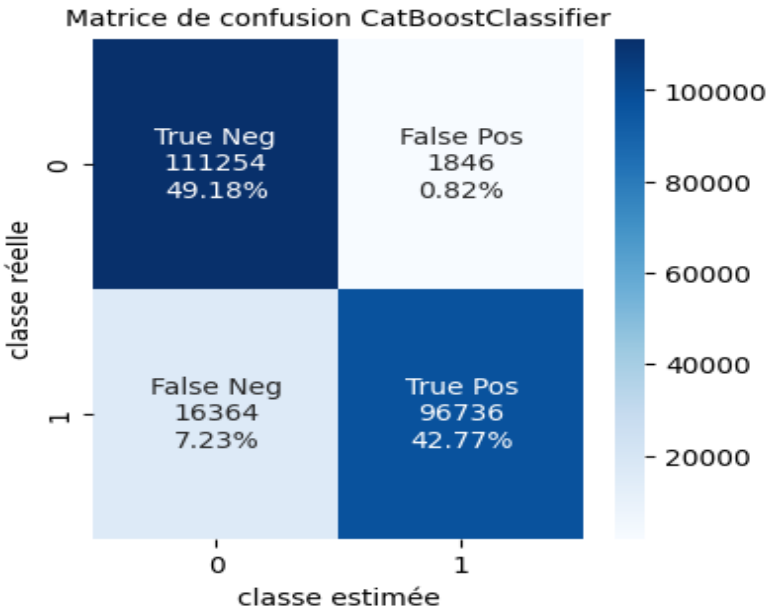
Méthodologie d'entraînement

Nous avons sélectionné quatre algorithme (en plus du modèle naïf) :
Régression Logistique, DecisionTreeClassifier, LGBMClassifier, et CatBoost.

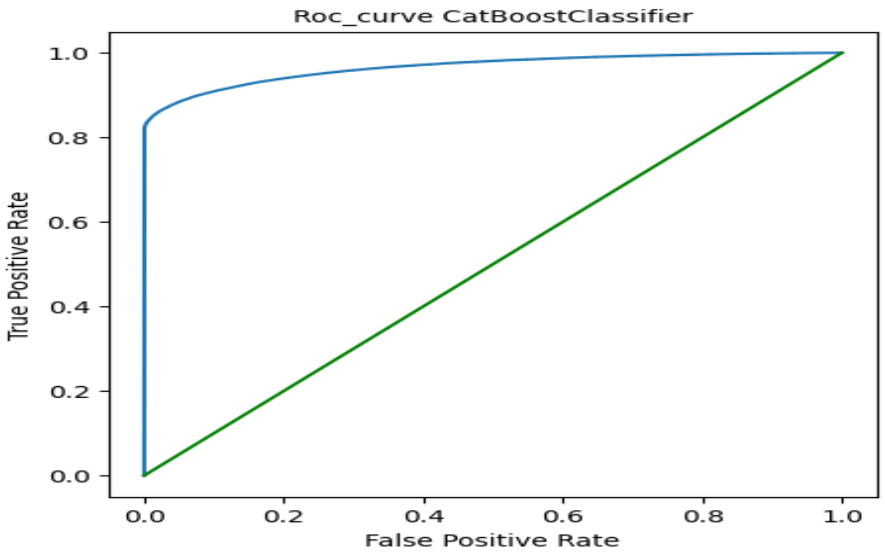
Pour optimiser les hyperparamètres de chaque modèle et évaluer sa performance de manière exhaustive, nous avons employé la méthode **GridSearchCV** de scikit-learn, intégrant une validation croisée à 5 plis. Cette approche méthodique nous a permis de systématiquement explorer une grille prédéfinie d'hyperparamètres pour chaque modèle, en cherchant la combinaison qui maximise la performance du modèle selon deux critères principaux : l'aire sous la courbe ROC (AUC) et notre score métier personnalisé.

NOTE METHODOLOGIQUE

Pour illustrer le résultat nous avons présenté une matrice de confusion :



Ainsi que la courbe ROC :



Une courbe ROC caractérise le classifieur qui a produit les résultats sous forme de probabilités par variation du seuil de classification. Un modèle idéal a une sensibilité et une spécificité = 1.

Plus la courbe se rapproche de cet idéal, meilleurs sont les indicateurs.

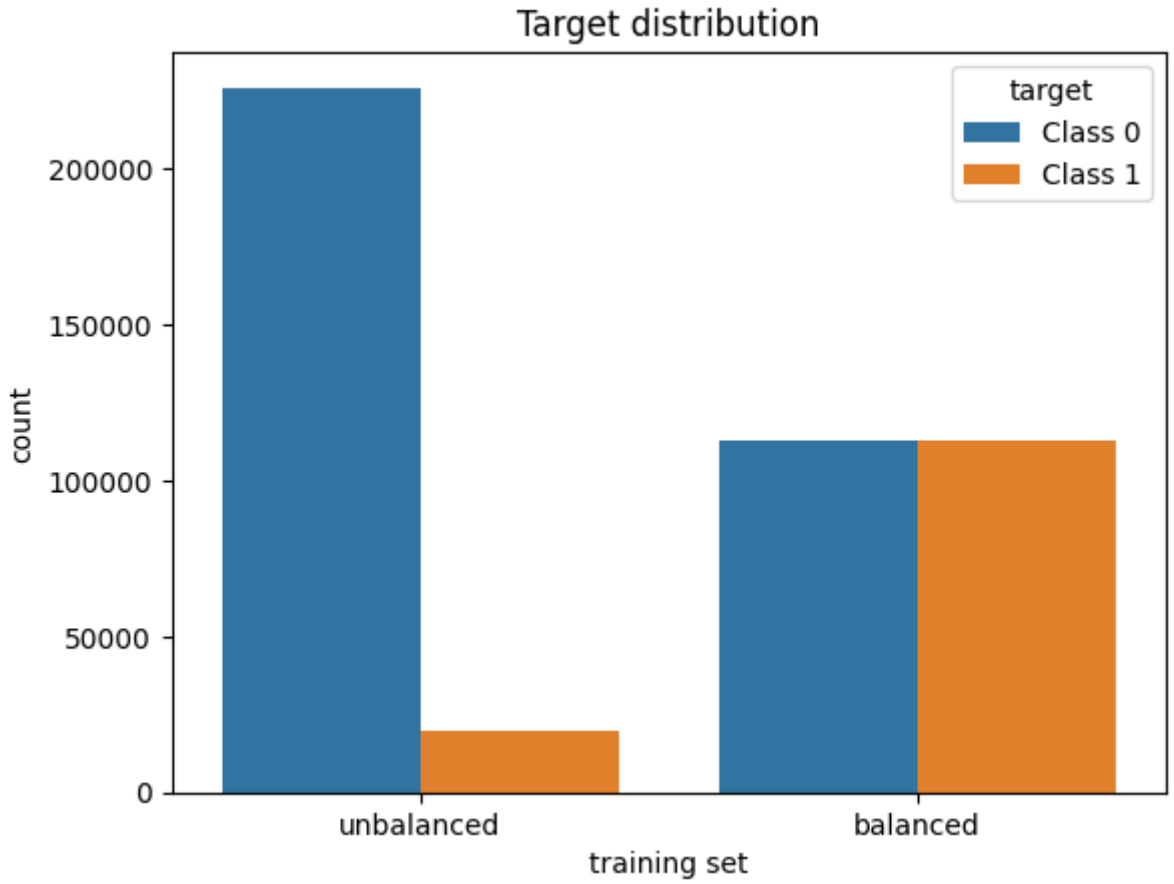
¬ On peut résumer la mesure de cette performance par l'aire sous la courbe (Area Under the Curve).

NOTE METHODOLOGIQUE

Equilibrage des classes

Stratégie :

Pour traiter le déséquilibre des classes, nous avons opté pour une combinaison de suréchantillonnage avec SMOTE et de sous-échantillonnage avec RandomUnderSampler, atteignant un ratio final de 1:1 entre les classes dans notre ensemble d'entraînement. Cette approche a non seulement amélioré l'équité des prédictions mais a également contribué à augmenter la sensibilité du modèle aux instances de la classe minoritaire.



NOTE METHODOLOGIQUE

Fonction de coût, algorithme et métrique d'évaluation

Fonction de Coût

Ma principale préoccupation dans ce projet était de minimiser l'impact des faux négatifs (FN), qui dans le contexte du risque de crédit, peuvent signifier la perte d'opportunités importantes ou l'exposition à des risques financiers significatifs :

```
score=(10*fn + fp)/len(y_true)
```

Cette fonction pénalise les FN dix fois plus que les FP. Les conséquences d'un FN étant nettement plus préjudiciables que celles d'un FP selon l'énoncé. Cette fonction nous permet de guider l'optimisation du modèle

Algorithme : LGBMClassifier

LGBMClassifier, un composant de LightGBM, une bibliothèque de boosting de gradient réputée pour sa vitesse et son efficacité, surtout sur de grands ensembles de données. LightGBM utilise des techniques de fractionnement des feuilles basées sur le gradient, ce qui le rend particulièrement adapté aux ensembles de données complexes et de grande dimension comme le nôtre. Son efficacité computationnelle, sa capacité à gérer le surajustement, et sa facilité d'utilisation en font un choix idéal pour notre projet, permettant une exploration rapide de l'espace des hyperparamètres et l'obtention de modèles performants.

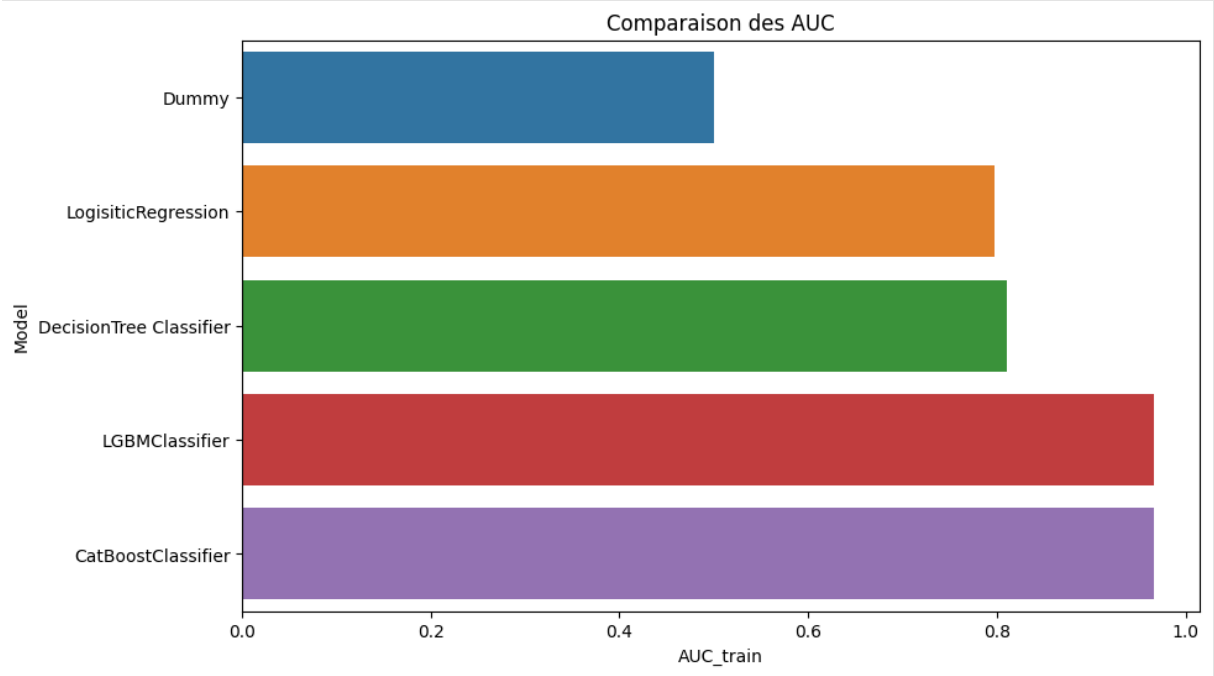
Métriques d'Évaluation : AUC ROC et Score Métier

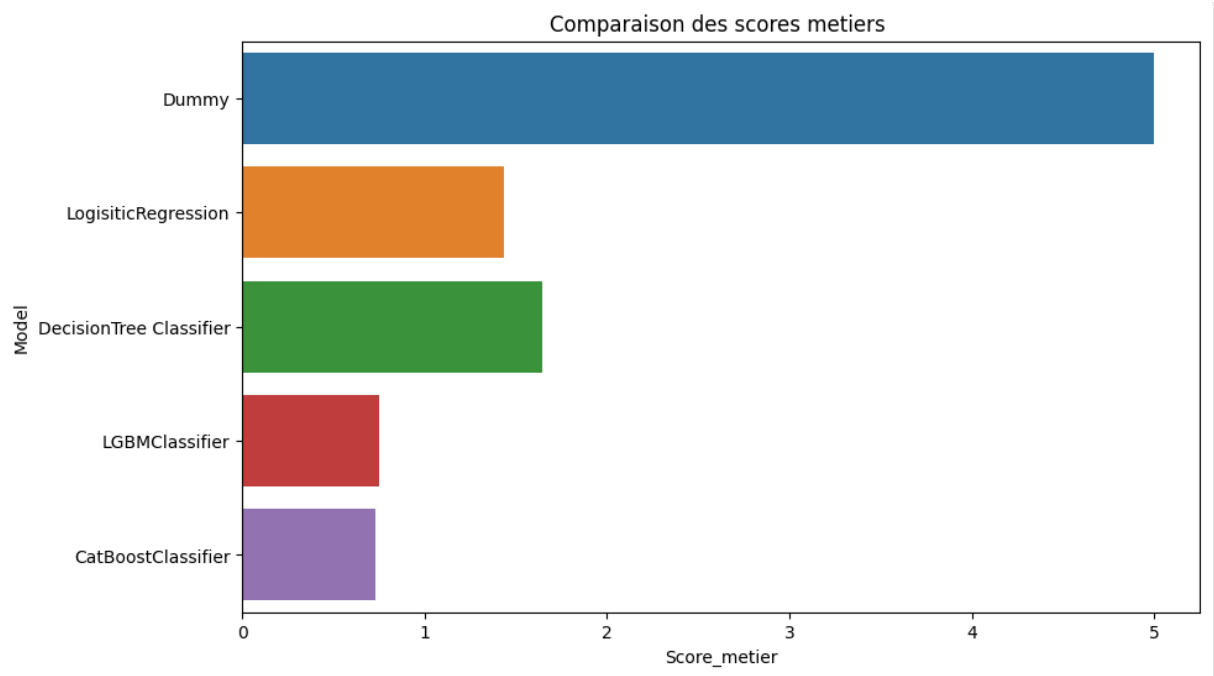
L'AUC de la courbe ROC : Cette métrique fournit une mesure globale de la capacité du modèle à distinguer entre les classes positives et négatives à tous les seuils de classification. Une valeur proche de 1 indique une excellente performance, tandis qu'une valeur proche de 0.5 suggère une performance équivalente à une classification aléatoire.

Le Score Métier : Grâce à ma fonction de coût métier personnalisée j'évalue la performance du modèle en termes d'impact métier des erreurs de classification.

NOTE METHODOLOGIQUE

Synthèse des résultats





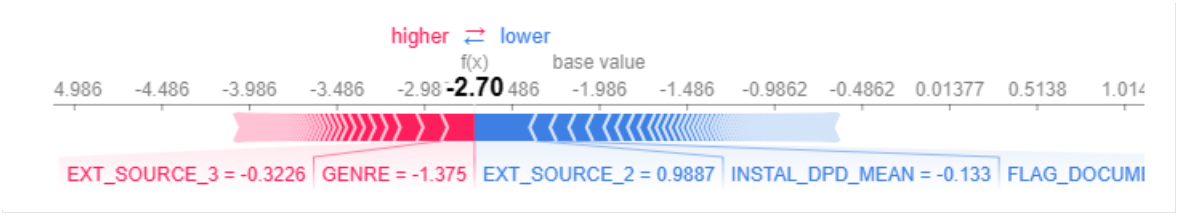
Techniquement Catboost est légèrement plus performant que LGBMClassifier mais pour des questions de facilité d'interprétation des résultats par le chargé de compte il m'a semblé plus pertinent de garder LGBMClassifier

NOTE METHODOLOGIQUE

Interprétabilité du modèle

Nous avons implémenté la méthode SHAP (SHapley Additive exPlanations) qui consiste à calculer la valeur de Shapley pour toutes les variables de tous les individus c'est-à-dire la moyenne de l'impact d'une variable (sur la sortie, donc la prédiction) pour toutes les combinaisons de variables possibles.

¬ La somme des effets de chaque variable explique alors la prédiction. Le graphe ci-dessous représente les impacts des valeurs des variables importantes sur la prédiction. Les variables en rouge contribuent le plus à une prédiction positive (dossier non remboursé), les variables en bleu contribuent le plus à une prédiction négative (dossier remboursé).



Dans cet exemple, la valeur SHAP est inférieure à la valeur de base, le dossier est classé négatif. Les variables en bleu expliquent cette décision.

SHAP permet également une analyse globale. En effet, en moyennant les valeurs absolues des valeurs de chaque variable (niveau local), on remonte à l'importance globale des variables.

NOTE METHODOLOGIQUE

Limites et les améliorations possibles

La réalisation du modèle a nécessité la conception de nombreux blocs de transformation et de traitement des données. Chaque bloc fait appel à des méthodes paramétrables. De fait, les résultats sont dépendants des paramètres choisis. L'architecture du code permet d'optimiser les blocs indépendamment.

Sélection des variables

Les informations disponibles relatives à l'importance des variables sont débattues avec les experts métier en vue de définir les stratégies techniques à tester dans les différents blocs concernés :

-Valeurs manquantes
-Corrélations entre variables
-Seuil de variance
-Réduction de dimensions (RFE)

Equilibrage des données

L'équilibrage des données introduit des données artificielles donc la possibilité d'incohérences. Des tests peuvent être réalisés en variant certains paramètres (ratios classes).

Fonction d'évaluation du gain

Les règles métier et les critères financiers relatifs aux pertes et profits doivent être communiqués en vue d'établir une fonction d'évaluation du gain adaptée.

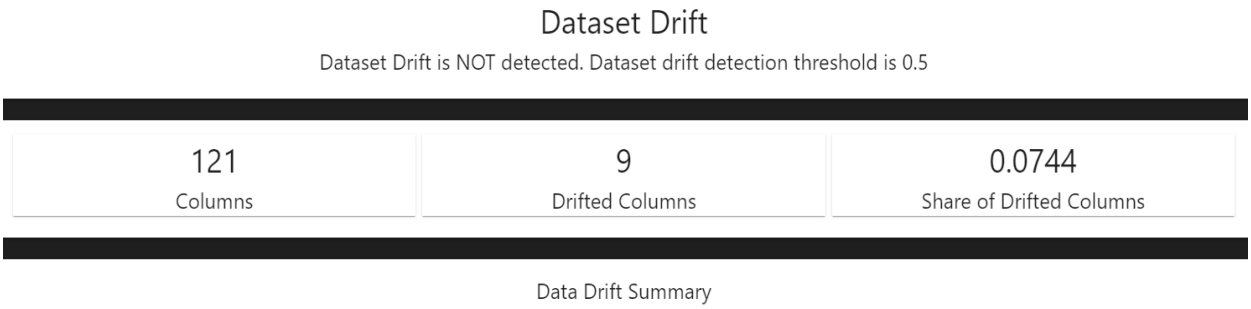
Choix de l'algorithme

Nous avons testé plusieurs classifieurs et retenu LightGBM pour sa rapidité d'exécution. D'autres classifieurs comme XGBoost peuvent potentiellement apporter de meilleures performances techniques. Il s'agit de les tester dans le pipeline complet si les contraintes de temps en production le permettent

NOTE METHODOLOGIQUE

L'analyse du Data Drift

Nous avons testé le data drift entre les données d'entrainement et de test



Du data drift a été constaté dans 9 colonnes sur 121

Voici la copie d'écran des 5 premières

Drift is detected for 7.438% of columns (9 out of 121).

Search

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
AMT_REQ_CREDIT_BUREAU_QRT	num			Detected	Wasserstein distance (normed)	0.359052
AMT_REQ_CREDIT_BUREAU_MON	num			Detected	Wasserstein distance (normed)	0.281765
AMT_GOODS_PRICE	num			Detected	Wasserstein distance (normed)	0.210785
AMT_CREDIT	num			Detected	Wasserstein distance (normed)	0.207334
AMT_ANNUITY	num			Detected	Wasserstein distance (normed)	0.161102

Rows per page: 5 rows 1-5 of 121

Conclusion :

Le seuil de data drift étant de 7.4% nous considérons qu'il n'y a pas de variations significatives entre les données d'entrainement et de test