Note Méthodologique

Projet n°7 : Implémenter un modèle de Scoring

Contexte et Objectifs

La société financière Prêt à dépenser propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt.

L'entreprise souhaite mettre en œuvre un outil de scoring crédit qui calcule la probabilité qu'un client rembourse son crédit, puis classifie la demande en crédit accordé ou refusé. Elle souhaite donc développer un algorithme de classification en s'appuyant sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.).

De plus, les chargés de relation client ont fait remonter le fait que les clients sont de plus en plus demandeurs de transparence vis-à-vis des décisions d'octroi de crédit. Cette demande de transparence des clients va tout à fait dans le sens des valeurs que l'entreprise veut incarner. *Prêt à dépenser* décide donc de développer un dashboard interactif pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d'octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

Table des matières :

I - Les étapes préalables à la modélisation

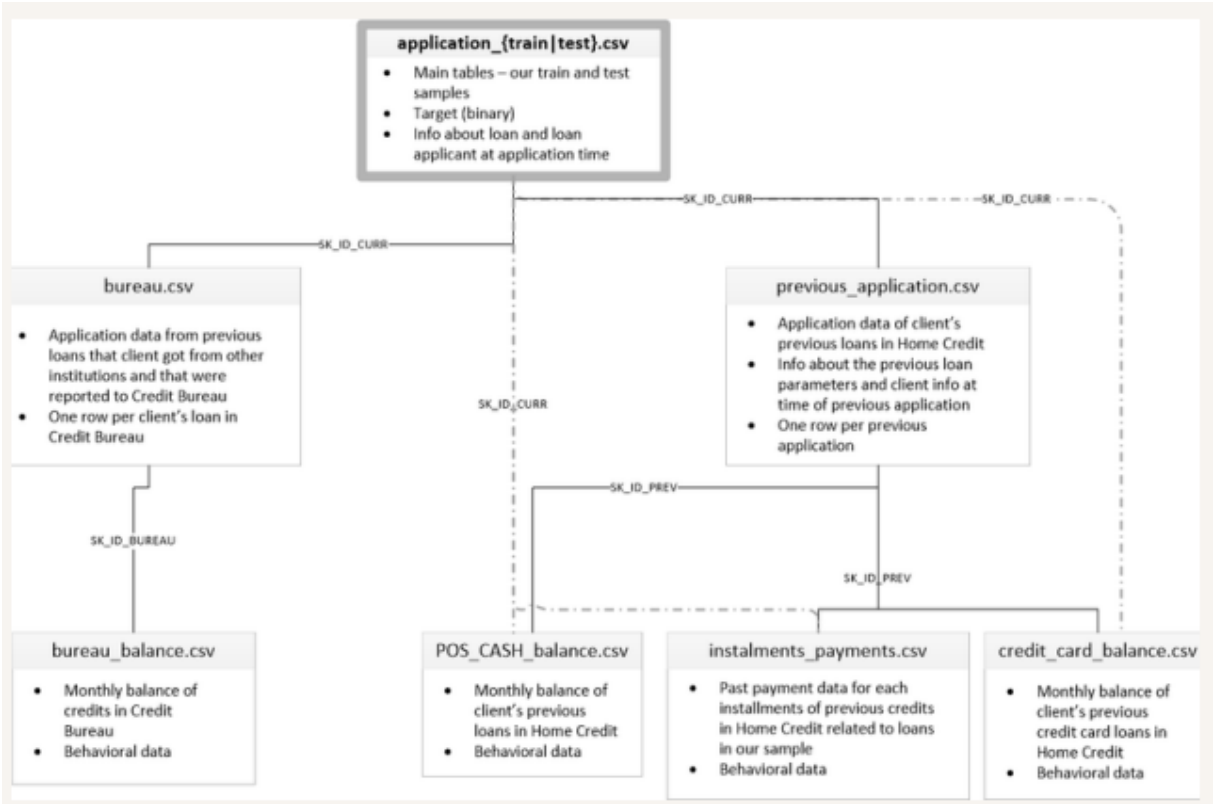
Les données sont dispatchées dans 7 fichiers liés entre eux selon le schéma ci-dessous. La table « application » regroupe les informations personnelles des clients actuels ainsi que les données relatives au crédit qu'ils demandent.

Cette table est séparée en 2 jeux de données :

- l'application "train" regroupant 307 511 clients dont on connaît la décision de « Prêt à Dépenser » sur l'octroi du crédit (variable "Target") et
- l'application "test" dont on ne connaît pas cette décision.

Les autres fichiers contiennent les données historiques de prêt de ces mêmes clients : Les tables « bureau » et « balance_bureau » contiennent les informations des crédits passés dans des institutions autres que « Prêt à Dépenser »

La table « Previous_application » reporte les donnés des crédits passés auprès de « Prêt à dépenser ».



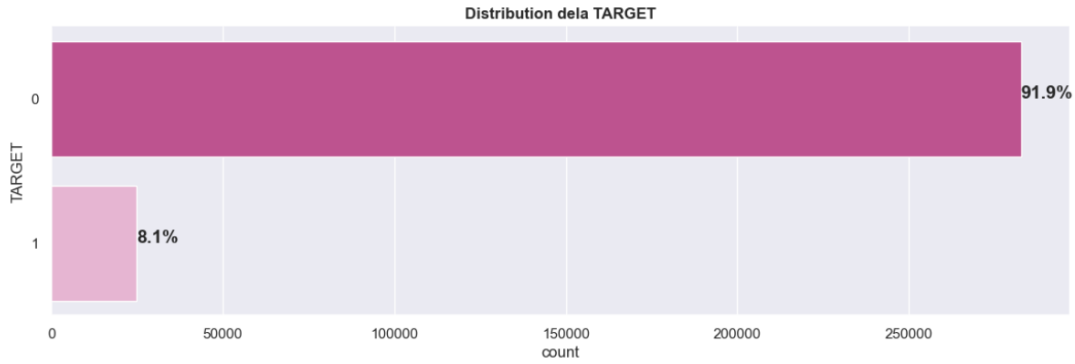
Bien que l'on puisse penser qu'il suffit d'un grand nombre de données pour avoir un algorithme performant, les données dont nous disposons sont souvent non adaptées, voire insuffisantes ou erronés (informations incomplètes, valeurs manquantes…). Il est donc indispensable d'établir une stratégie de pré-traitement des données – autrement appelé Data Preprocessing – à partir de nos données brutes pour arriver à des données exploitables qui nous donneront un modèle plus performant

Après une première phase de compréhension, **d'analyse et de nettoyage des 7 fichiers distincts,** une **data globale unique a été construite** :

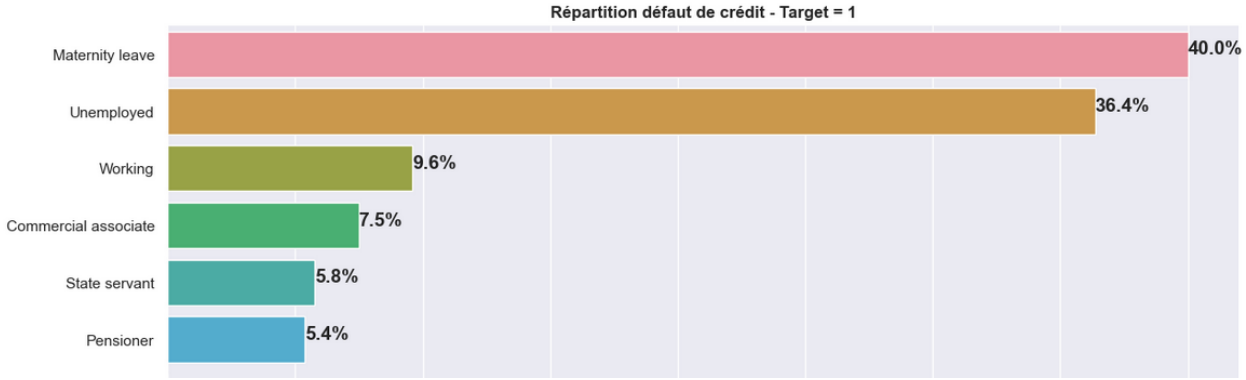
- Avec comme point de départ une **table regroupant la data Application train et test**
- Agrégation et fusion des autres tables à cette data globale à partir des clés uniques liant les différentes tables – schéma ci-dessous
- Il a été fait le choix de supprimer les valeurs manquantes sur cette base agrégée avec (simpleImputer – Médiane)

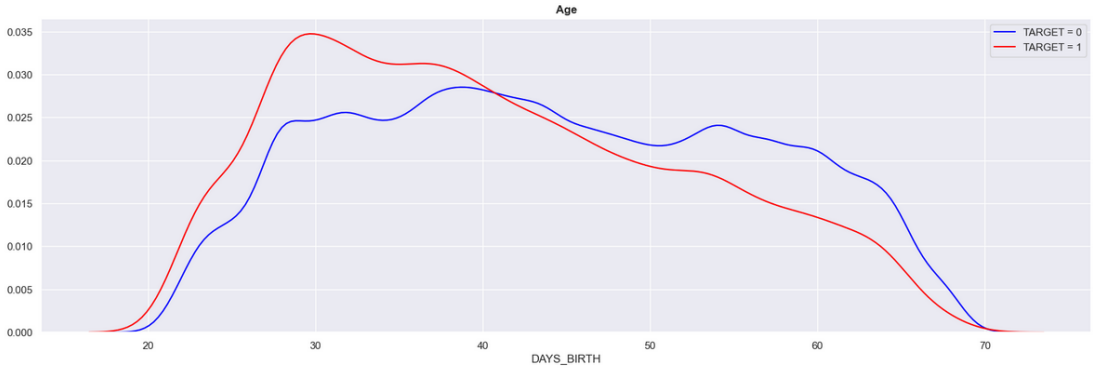
L'analyse exploratoire de cette base a permis de mettre en avant :

✓ un déséquilibre des classes (91.9% sont étiquetés « 0 » à savoir ayant honoré leur crédit, et 8.1% sont étiquetés « 1 », c'est-à-dire ne l'ayant pas honoré)



✓ des comportements de risque intéressant , comme la répartition du défaut de crédit par les grandes catégories d'activité (Sans Emploi, retraité, travailleur …) et selon l'âge:





Enfin, afin de disposer de données analysables, uniformes et optimales, une phase de features engineering a été réalisé selon les 4 axes suivants :

Création de 4 nouvelles variables métier	•CREDIT_INCOME_PERCENT: % du montant du crédit par rapport au revenu d'un client •ANNUITY_INCOME_PERCENT: % de la rente de prêt par rapport au revenu d'un client •CREDIT_TERM: Durée du paiement en mois •DAYS_EMPLOYED_PERCENT: % des jours employés par rapport à l'âge du client
Encodage des variables catégorielles	•La majorité des algorithmes fonctionnent sur des données numériques. •Encodage des variables catégorielles avec LabelEncoder
Standardisation des données	•Standardisation avec MinMaxScaler sur une plage de [0,1] pour une meilleure comparaison des données, faciliter la convergence des algorithmes par une réduction de l'echelle et réduire l'impact des outliers
Features Selection	•Mise en oeuvre d'une réduction de dimension avec RFECV •Ce qui a permis de réduire le nombre de features à 112

II - Méthodologie d'entrainement du modèle

1. Les différents modèles

La modélisation a d'abord été appréhendé avec DummyClassifier qui est un classifieur naïf, c'est-à-dire qu'il effectue des prédictions sans essayer de trouver des modèles dans les données.

Cette première classification sert de baseline, c'est-à-dire qu'elle sert de référence pour mesurer la performance des autres modèles.

Il a ensuite été intégré différents types d'algorithme :

- **KNN Classifier** (algorithme des plus proches voisins) qui classifie les variables d'un jeu de données en **analysant les similitudes entre elles**. Pour cela, le KNN utilise un graphique et calcule la distance entre les différents points. Ceux qui sont les plus proches sont enregistrés dans la même catégorie.

- **La régression logistique** : permet de **prédire la probabilité** qu'un événement arrive (valeur de 1) ou non (valeur de 0) à partir de l'optimisation des coefficients de régression.

- **CatBoostClassifier** est un algorithme Gradient Boosting Machine (GBM) conçu spécifiquement pour créer des modèles d'apprentissage automatique hautes performances lors de l'utilisation de types de données catégoriels. Il utilise également une stratégie d'apprentissage spécialement conçue pour lisser les irrégularités dans l'ensemble de données.

- **Light Gradient Boosting Machine (LGBM)** est un dérivé de l'algorithme Gradient Boosting Machine (GBM). LGBM est conçu pour créer des modèles d'apprentissage automatique hautes performances sur de grands ensembles de données. LGBM utilise une technique d'apprentissage basée sur l'histogramme en divisant les données verticalement et horizontalement. Cela permet de traiter les données plus rapidement et d'utiliser moins de mémoire.

2. Le déséquilibre des classes

Les clients en difficulté de paiement sont largement sous-représentés (8.1%) dans les données d'entraînement. Or, les méthodes de machine learning classiques ne sont pas toujours adaptées pour la classification sur des données déséquilibrées. Elles donnent souvent de mauvais résultats et, pire encore, elles peuvent induire en erreur avec des scores trop optimistes.

1 Les Méthodes data-level

L'idée derrière les approches data level est toujours la même. Il s'agit de transformer les données d'entraînement du modèle pour atténuer le déséquilibre. On va souvent utiliser des techniques d'échantillonnage pour ajouter des représentants dans la classe minoritaire et/ou en retirer de la classe majoritaire.

Sous-échantillonnage aléatoire (Undersampling):

La première approche consiste en un sous-échantillonnage de la classe majoritaire. On cherche à réduire la taille de la classe majoritaire pour atténuer le déséquilibre des classes. On va choisir les points à retirer de manière très naïve, simplement en retirant des points de façon aléatoire.

Sur-échantillonnage aléatoire (Ovesampling):

il s'agit cette fois d'**augmenter le nombre de données appartenant à la classe minoritaire**, jusqu'à atteindre un certain équilibre. Ou du moins un taux satisfaisant pour réaliser des prédictions fiables.

SMOTE :

Une autres approche d'échantillonnage que l'on retrouve souvent est SMOTE (Synthetic Minority Oversampling Technic ou suréchantillonnage minoritaire synthétique). Il s'agit d'une techniques de suréchantillonnage. Plutôt que de réduire la taille de la classe majoritaire, on cherche à agrandir celle de la classe minoritaire. Pour cela, on va sélectionner des points de la classe que l'on souhaite agrandir et en créer de nouveaux.

2 Les Méthodes algorithm-level

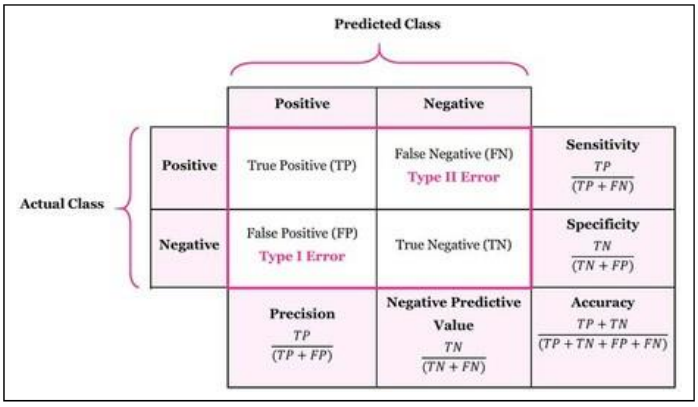
L'apprentissage sensible aux coûts pour la classification déséquilibrée se concentre d'abord sur l'attribution de différents coûts aux types d'erreurs de classification erronées qui peuvent être faites, puis en utilisant des méthodes spécialisées pour prendre ces coûts en compte. Les différents coûts de mauvaise classification sont mieux compris en utilisant l'idée d'une matrice de coûts. Contrairement aux méthodes de suréchantillonnage et de sous-échantillonnage, les méthodes des pondérations équilibrées ne modifient pas le rapport des classes minoritaires et majoritaires. Au lieu de cela, il pénalise les mauvaises prédictions sur la classe minoritaire en donnant plus de poids à la fonction de perte.

3. Fonction coût métier et métrique d'évaluation

Généralement en machine learning, il faut s'attarder sur le choix des métriques de mesure des performances. Cette règle est encore plus fondamentale lorsque l'on travaille sur des données déséquilibrées pour éviter de mauvaises interprétations des résultats.

En classification binaire il est d'usage d'utiliser le pourcentage de bonnes prédictions (accuracy) comme score mais celui est affecté par le déséquilibre des classes. En effet, ce pourcentage peut être élevé même si une grande partie des points de la classe minoritaire est mal classifié.

Il faut donc utiliser des métriques moins influencées par le déséquilibre des classes comme le recall (sensibilité) ou le F-Score sont de bons exemples.



La Matrice de Confusion

Dans la tâche de classification, la matrice de confusion est le principal indicateur de la qualité d'un modèle. Il s'agit d'un tableau à double entrée mettant en correspondance les classes réelles et les classes prédites par le modèle.Elle sert de base à tous les calculs d'indicateurs

Comme l'accuracy ne distingue pas le type des erreurs commises, elle est souvent complétée par deux indicateurs : le rappel (recall en anglais) et la précision.

- **Le rappel** est la proportion de la classe positive détectée (compris entre 0 et 1). Un fort rappel indique donc que presque tous les cas de la classe positive ont été détectés, par exemple qu'une très grande partie des patients réellement malades ont été classés comme tels.
- **La précision** est, elle, la proportion des vrais positifs dans l'ensemble des positifs détectés (aussi comprise entre 0 et 1). Elle permet d'analyser, dans les cas qui sont sortis positifs par le modèle, quelle proportion l'est réellement. Cela permet d'estimer les faux positifs.

En fonction des problèmes métiers, l'un des deux indicateurs aura un plus fort impact que l'autre. Il peut donc être intéressant de ne pas choisir un modèle selon son accuracy mais selon son rappel et/ou sa précision.

- Le but du **F1-score** (ou "score F1") est de donner un unique indicateur qui prenne en compte le rappel et la précision, sans tomber dans les pièges de l'accuracy.

  Sa définition est la moyenne harmonique de la précision et du rappel. En pratique cela donne la formule suivante : 2 * (precision * recall) / (precision + recall). Sa valeur varie de 0 à 1. Un score de 1 indique une précision et un rappel de 100 %.

Afin de prendre en compte le risque métier, un **score métier** a été mis en place pour prendre en compte le fait que le risque de refuser un crédit à tort (faux négatifs) n'a évidemment pas le même poids que d'accorder un crédit à un client n'ayant pas honoré son engagement (faux positifs).

Ainsi, un nouvel indicateur normalisé de performance est défini en pondérant les prédictions par des poids de la façon suivante :

- Poids de 1 pour les bonnes prédictions (dont TP et TN)
- Poids de -1 pour les faux négatifs (refus du crédit à tort)
- Poids de -10 pour les faux positifs (attribution du crédit pour les mauvais payeurs)

Ce qui traduit mathématiquement par :

$$SCORE_{METIER} = \frac{(J - Jmin)}{(Jmax - Jmin)}$$

Où $J = 1 * TP + 1 * TN - 1 * FP - 10 * FN$

Où $Jmax = (FP + TN) * 1 + (FN + TP) * 1$

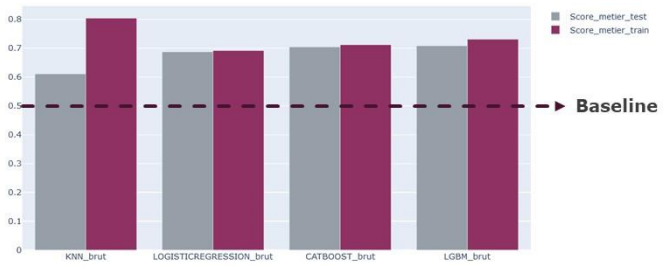
Et $Jmin = (FP + TN) * (-1) + (FN + TP) * (-10)$

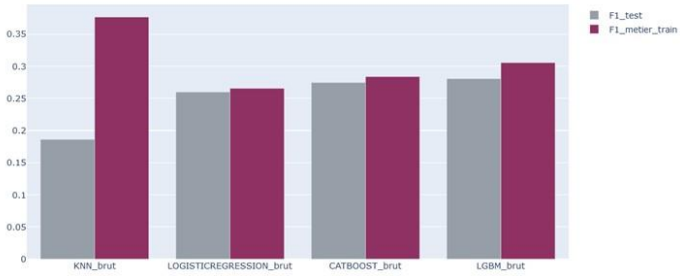
4. Comparaison des résultats

Pour commencer, il est important de préciser que les modèles ont été élaborés avec GridsearchCV et 15 cross validation, ce qui signifie que le modèle est tester 15 fois pour chaque ensemble d'hyperparamètres sélectionnés.

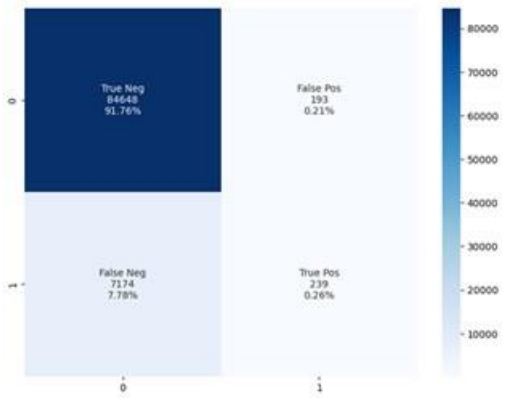
Les mêmes résultats ont été réalisés en appliquant les différentes méthodologies pour palier aux déséquilibres des classes, sans pour autant fournir de résultats meilleurs. Il s'agit ci-dessous des résultats sur les données sans traitement du déséquilibre.

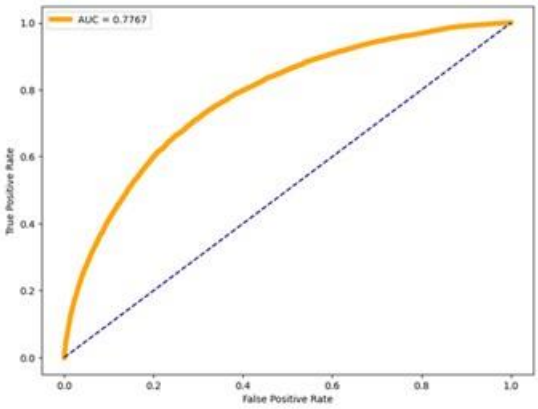
Au vu des résultats, l'algorithme permettant d'optimiser le score métier et le F1 score, sans créer de sur apprentissage le modèle LGBM Classifier.

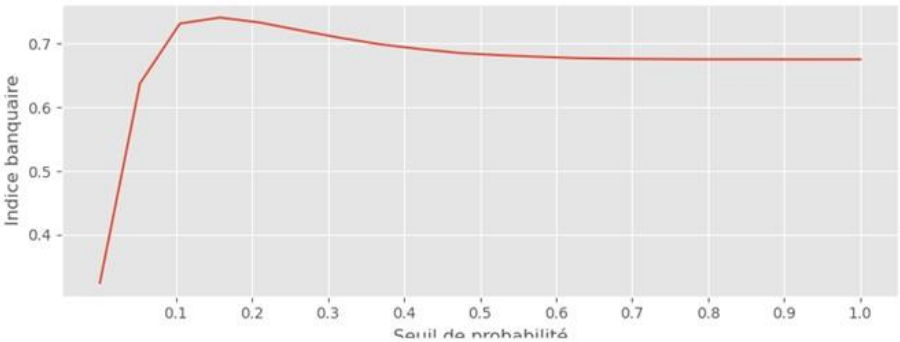




Ci-dessous la matrice de confusion et la courbe ROC, qui montre un AUC de 0.7767. On peut voir également que le seuil de probabilité aux alentours de 0.10.







III - Interprétation globale et locale du modèle

L'interprétation des modèles est importante en machine learning, ne serait-ce que parce que dans de nombreux domaines, il faut expliquer – justifier – la prise de décision induite par le modèle prédictif.

La question de l'identification des variables pertinentes reste centrale, ne serait-ce que pour comprendre le mécanisme d'affectation sous-jacent au modèle.

L'importance des variables mesure l'impact global de chaque descripteur dans le modèle. Elle peut être estimée en modélisation (sur l'échantillon d'apprentissage) ou en prédiction (sur l'échantillon test). Dans les deux cas, les principales étapes sont les mêmes :

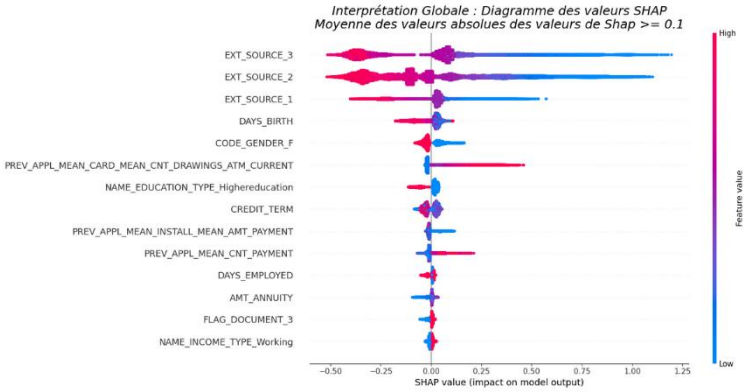
1.calculer le taux d'erreur de référence,

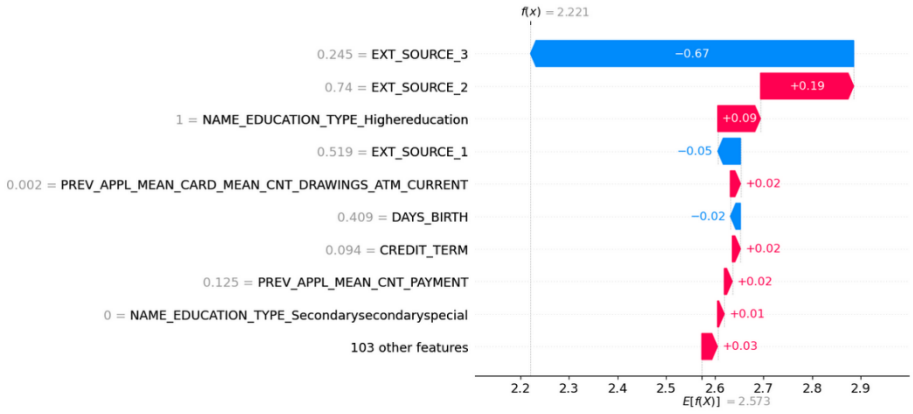
2.calculer ensuite le même indicateur en neutralisant tour à tour chaque variable prédictive,

3.former le ratio entre les deux valeurs.

Les valeurs de Shapley calculent l'importance d'une variable en comparant ce qu'un modèle prédit avec et sans cette variable. Cependant, étant donné que l'ordre dans lequel un modèle voit les variables peut affecter ses prédictions, cela se fait dans tous les ordres possibles, afin que les fonctionnalités soient comparées équitablement

L'intelligibilité globale cherche à expliquer le modèle dans sa globalité. C'est-à-dire quelles sont les variables les plus importantes en moyenne pour le modèle. Par exemple, quelles caractéristiques affectent le comportement général d'un modèle d'allocation de prêt ?





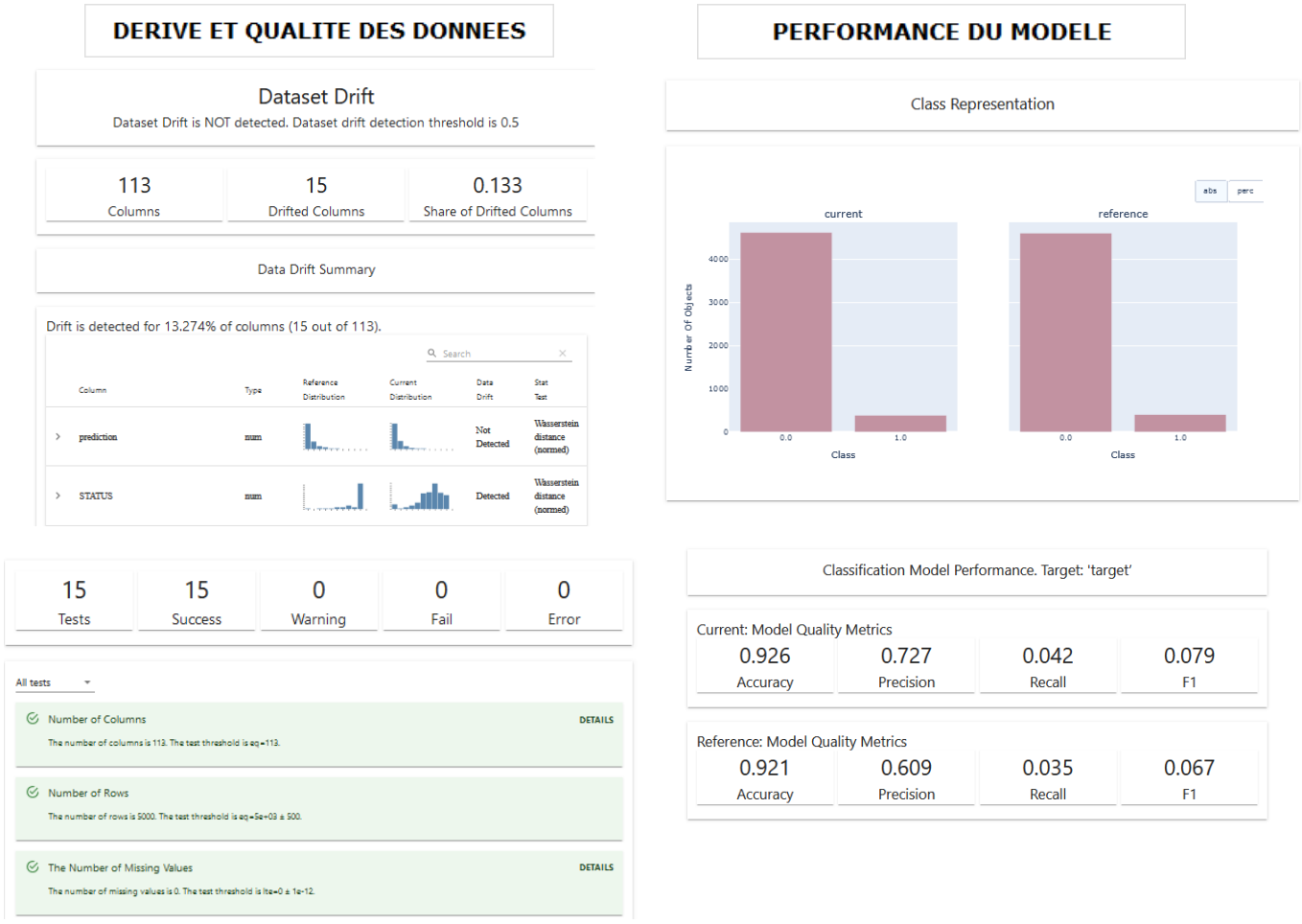
A contrario, **l'intelligibilité locale,** consiste à expliquer la prévision f(x) d'un modèle pour un individu x donné. Par exemple, pourquoi la demande de prêt d'un client a-t-elle été approuvée ou rejetée

IV - Analyse du Data Drift

Dans le cadre du déploiement d'un modèle de machine learning en production, il est essentiel de mettre en place un suivi des data et prédictions en production afin de s'assurer :

- **Dérive des données :** détecte les modifications dans la distribution des entités en entrée.
- **Qualité des données :** fournit des statistiques détaillées sur les fonctionnalités et une vue d'ensemble du comportement.
- **Dérive de la cible** : détecte les changements dans la sortie du modèle pour des données cibles numériques et Catégorielles
- **Performances du modèle** : évalue la qualité du modèle et les erreurs du modèle dans le cadre de classification, Classification probabiliste ou Modèles de régression

Concrètement, les différentes analyses portent sur la comparaison entre un jeu de données de référence ayant servi à la construction du modèle et un jeu de données dit « current » ou « prod » correspond aux remontées de la production.



V - Limites et améliorations du modèle

La première limite dans ce projet provient de la méconnaissance du milieu bancaire et de la finesse des informations transmises. J'ai donc garder le maximum d'information, agréger des informations qui n'étaient peut être pas les plus cohérentes d'un point de vue métier.

Par ailleurs, il est ressorti de l'analyse des features importantes appelées EXT-SOURCE 1, EXT-SOURCE 2, et EXT-SOURCE 3 qui sont des scores normalisés provenant d'une source de données externes. Sans aucune autre information. Elles ne permettent pas de bien interpréter.

Par ailleurs, le choix et l'optimisation du modèle de classification ont été réalisé sur la base d'une hypothèse forte concernant la métrique d'évaluation : un poids de 1 contre 10 entre les faux négatifs et les faux positifs. L'axe principal d'amélioration serait de définir plus finement la métrique d'évaluation et la fonction de coût en collaboration avec les équipes métier.

Enfin, il aurait été primordial de pouvoir discuter avec « Prêt à dépenser » sur les points clés de la métrique. En effet, il y a un compromis indispensable à faire entre la part de faux positifs et celle de faux négatifs. Le choix du seuil final d'acceptation ou refus de crédit à un poids important dans ce compromis, puisqu' augmenter le seuil tend à augmenter le nombre de crédits refusés, et donc augmenter le nombre de faux positifs alors que baisser le seuil a l'effet inverse. Ce seuil devrait donc être discuté et fixé avec le client, par exemple en lien avec une analyse financière des pertes dues aux erreurs d'attribution.