Implémenter un modèle de Scoring :

Note Méthodologique

Projet n°7 - Data Scientist -
OpenClassrooms - Isabelle Contant

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.

- ***Implémentation d'un modèle de scoring :***

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.).

Les données originales sont téléchargeables sur Kaggle à cette **adresse**.

- ***Dashboard interactif réalisé avec Streamlit :***

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.

Le dashboard réalisé avec Streamlit est accessible en **cliquant ici**.

Tous les programmes sont accessibles sur **Github**.

Sommaire

01 Les étapes préalables à la modélisation

02 La méthodologie d'entraînement du modèle

03 La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation

04 L'interprétation globale et locale du modèle

05 Les limites et les améliorations possibles

Note Méthodologique

Cette note méthodologique présente les techniques employées pour la construction du modèle de scoring ainsi que les outils mis en place pour son interprétation.

Pour répondre à cet objectif, nous évoquerons succinctement :
- La méthodologie d'entraînement du modèle
- La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation
- L'interprétation globale et locale du modèle
- Les limites et les améliorations possibles

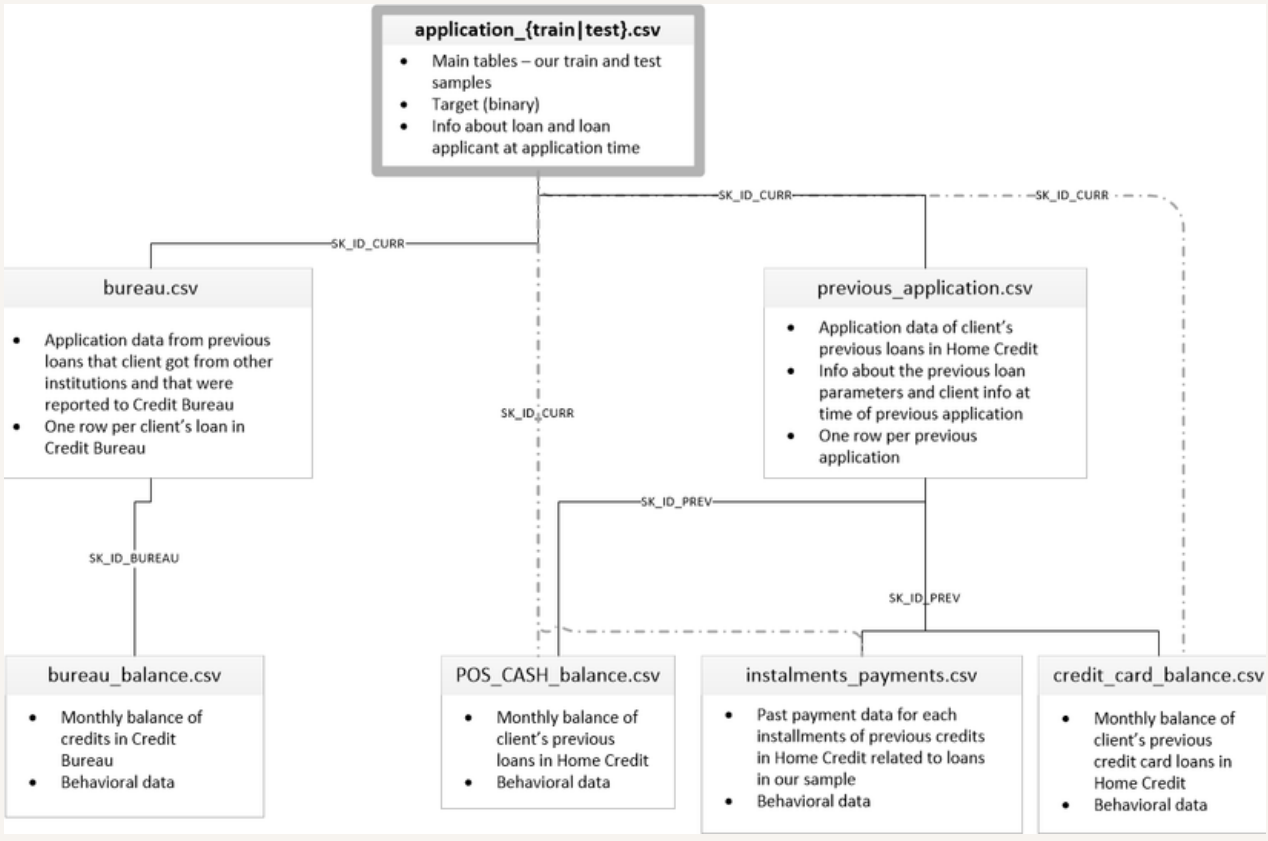
01 Les étapes préalables à la modélisation

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 »*.



Historique de prêt du client dans d'autres institutions financières

Historique de prêt du client chez *"Prêt à Dépense"*

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. Il faut donc les traiter préalablement pour pouvoir ensuite les utiliser : c'est l'étape de préprocessing.

En effet, des erreurs d'acquisition liées à des fautes humaines ou techniques peuvent corrompre notre dataset et biaiser l'entraînement. Parmi ces erreurs, nous pouvons citer des informations incomplètes, des valeurs manquantes ou erronées ou encore des bruits parasites liés à l'acquisition de la donnée. Il est donc souvent 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.

Data Cleaning

La première étape consiste en un nettoyage des données incorrectes, incomplètes ou manquantes.
Cette étape a été réalisée sur chaque table indépendamment des autres.
Nous avons d'abord supprimé toutes les variables avec plus de 0,1% de valeurs manquantes. Puis, nous avons décidé de remplir les variables avec moins de 0,1% de données manquantes en les remplaçant par la médiane pour les données numériques, ou par la modalité la plus fréquente dans le cas de variables catégorielles (autrement appelé le mode).

Data Reduction

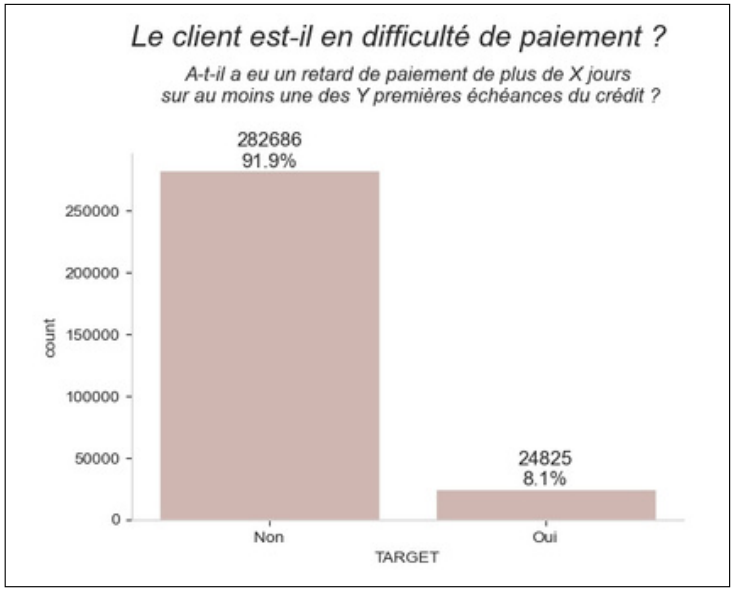
Lors de la construction d'un modèle prédictif, nous avons souvent de nombreuses fonctionnalités ou variables dans notre ensemble de données qui peuvent être utilisées pour former notre modèle. Cependant, ce n'est pas parce que la fonctionnalité existe dans notre jeu de données qu'elle est pertinente pour notre modèle et que nous devons l'utiliser.
Alors, comment savons-nous quelles fonctionnalités utiliser dans notre modèle ?
C'est là qu'intervient la sélection des caractéristiques. La sélection des caractéristiques est simplement un processus qui réduit le nombre de variables d'entrée, afin de ne conserver que les plus importantes.
Il existe trois catégories de méthodes de sélection de fonctionnalités, en fonction de la manière dont elles interagissent avec la variable à prédire, à savoir les méthodes de filtrage, d'encapsulation et intégrées. Nous les avons utilisées toutes les 3.

Data Transformation

Cette étape de prétraitement regroupe les changements effectués sur la structure même de la donnée. Ces transformations sont liées aux définitions mathématiques des algorithmes et à la manière dont ceux-ci traitent les données, de manière à optimiser les performances.
Nous avons essentiellement utilisé :

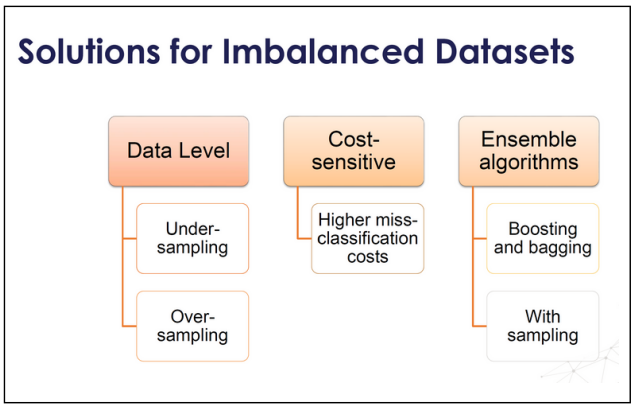
- La discrétisation de variables continues (à l'aide du découpage en intervalles) qui permet d'abaisser le nombre de modalités d'une variable et d'en supprimer les éventuelles valeurs aberrantes.
- La normalisation et la standardisation des données qui ramènent les données numériques à une échelle plus petite (par exemple entre -1 et 1), qui peuvent également centrer la moyenne et réduire la variance.

Au final, nous obtenons un jeu de données d'entraînement de 307 511 clients avec 56 variables descriptives.



02 La méthodologie d'entraînement du modèle

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. Une des causes de ces échecs est que les points de la classe minoritaire sont considérés comme des outliers qui ne contiennent aucune information.



Les différentes méthodes pour pallier le déséquilibre des classes

On sépare les approches pour contourner le déséquilibre des classes en 2. Les méthodes *data-level* qui consistent en des transformations opérées sur les données d'entraînement, et les méthodes *algorithm-level* (*Cost-Sensitive et Ensemble algorithmes*) qui reposent sur des modifications des modèles utilisés pour qu'ils soient plus adaptés à ce problème.

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 :**

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.

- **Tomek Links :**

Une autre approche de sous-échantillonnage de la classe majoritaire, un peu plus fine cette fois, propose un moyen plus naturel pour choisir les points à éliminer. L'idée est de chercher ceux de la classe majoritaire qui sont assez proches d'un point de la classe minoritaire. Les paires de points identifiées sont appelées Tomek Links. Dans chaque Tomek Link on va retirer le point qui appartient à la classe majoritaire. On peut remarquer au passage que l'identification des Tomek Links se fait comme l'identification des plus proches voisins dans l'algorithme KNN. Cette approche sera en pratique plus efficace que le sous-échantillonnage aléatoire. Elle va réduire la variance de la classe majoritaire et supprimer d'éventuels outliers, qui peuvent être une grande source de confusion.

- **SMOTE & ADASYN :**

Les autres approches d'échantillonnage que l'on retrouve souvent sont SMOTE (Synthetic Minority Oversampling Technic ou suréchantillonnage minoritaire synthétique) et ADASYN (ADAptive SYNthetic sampling ou échantillonnage synthétique adaptatif). Ce sont des techniques de suréchantillonnage cette fois. 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.

Néanmoins, il faut aussi savoir que certains algorithmes sont moins sensibles au problème des données déséquilibrées. Arbres de décision, forêts aléatoires et gradient boosting fonctionnent souvent bien avec de tels jeux de données.

2. Les Méthodes algorithm-level

La deuxième grande famille d'approches regroupe les méthodes dites algorithm-level. Elles reposent sur des adaptations des modèles de machine learning classiques afin qu'ils soient en mesure de mieux gérer le déséquilibre.

- **Apprentissage sensible aux coûts :**

Une des méthodes algorithm-level possible, est celle qui consiste à affecter un poids plus important à la classe minoritaire. On parle d'apprentissage sensible aux coûts. En pratique on va spécifier à notre modèle que le fait de bien classer un point de la classe minoritaire est plus important que de bien classer un point de la classe majoritaire. De cette façon on s'arrange pour qu'une erreur de classification d'un point de la classe minoritaire soit considérée par le modèle comme étant plus grave qu'une erreur de classification sur la classe majoritaire. Comme l'objectif des modèles de machine learning est toujours d'optimiser un paramètre prédéfini, on observe que l'optimisation du gain va inciter l'algorithme à donner plus d'importance à la classe minoritaire et donc à améliorer les prédictions sur celle-ci.

- **Apprentissage à une classe :**

Une autre approche qui peut donner de bons résultats dans certains cas est l'apprentissage à une classe. Plutôt que d'entraîner un modèle sur nos 2 classes, on entraîne un détecteur sur la classe majoritaire. Il sera ensuite capable de prédire si un point fait partie de la classe ou non. Même si les modèles d'apprentissage à une classe ne sont pas spécialement conçus pour contrer le déséquilibre des classes, ils peuvent offrir de gros gains de performances.

Mais avant d'appliquer ces différentes méthodes pour remédier au déséquilibre des classes, nous devons d'abord choisir le type de modèle de prédiction le plus adapté à nos données.

Pour ce faire, nous avons séparé notre jeu d'entraînement en 2 : un sous-échantillon d'entraînement regroupant 70% des données et un autre de validation. Chaque sous-échantillon contient le même mélange d'exemples par classe, c'est-à-dire environ 92% de classe 0 et 8% de classe 1.

Sur le sous-échantillon d'entraînement, nous avons évalué les modèles candidats à l'aide d'une validation croisée stratifiée répétée k-fold.

Le k-fold procédure de validation croisée fournit une bonne estimation générale de la performance du modèle qui n'est pas trop biaisée par l'optimisme, du moins par rapport à une seule séparation train-test.

Nous avons utilisé k = 10, ce qui signifie que chaque pli contient 92 254 / 10 soit environ 9 225 échantillons (clients).

Stratifié signifie que chaque pli contient le même mélange d'exemples par classe, c'est-à-dire environ 92% de classe 0 et 8% de classe 1.

Répété signifie que l'évaluation du processus sera effectuée plusieurs fois pour aider à éviter les résultats aléatoires et mieux capturer la variance du modèle choisi. Nous utilisons trois répétitions. Cela signifie qu'un seul modèle sera ajusté et évalué 10 × 3 (30) fois et la moyenne et l'écart type de ces essais seront calculés.

Sept modèles différents ont été testés : un classificateur naïf (Dummy), une régression logistique, deux classificateurs par forêt aléatoire, un modèle de Support Vector Machine et deux classificateurs par méthodes de descente de gradient : le classificateur de la librairie LightGBM et celui de la librairie XGBoost.

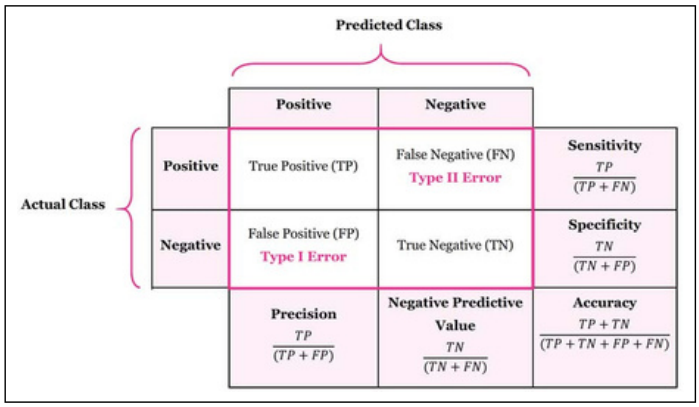
Le classificateur naïf (DummyClassifier) effectue des prédictions qui ignorent les entités d'entrée. Il sert de référence simple à comparer avec les autres classificateurs plus complexes.

La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation

Bien choisir les métriques pour mesurer les performances des modèles

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. Même si ça ne nous permet pas de contrer le problème, bien choisir les métriques nous permettra d'éviter des déconvenues.

En classification binaire il est d'usage d'utiliser le pourcentage de bonnes prédictions comme score. Sauf que ce pourcentage peut être élevé même si une grande partie des points de la classe minoritaire sont mal classifiés. Le score va lui aussi être affecté par le déséquilibre des classes. On pourrait croire que notre modèle est bon même lorsqu'il n'est bon que pour la classe majoritaire. Pour pallier ce problème, nous allons nous tourner vers des métriques qui ne seront pas affectées par la mauvaise répartition. On pourra utiliser des métriques qui seront beaucoup moins influencées par la classe majoritaire. **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. Plus le rappel est fort et moins il y a d'erreurs de type II, qui sont généralement les plus graves.

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 le nombre d'erreurs de type Hi (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 %.
Comme généralement les erreurs de type I ou de type II n'ont pas le même impact, il est possible de calculer le **Fβ-score** avec la valeur de β correspondant au coût relatif des erreurs de type II par rapport au type I. Le symbole β est en pratique remplacé par la valeur relative du rappel sur la précision.
Le score F1 en est donc un cas particulier où rappel et précision ont la même importance.
Un **score F2 indique que le rappel a deux fois plus d'impact que la précision, ou, autrement dit, qu'une erreur de type II est deux fois plus coûteuse qu'une erreur de type I.**

Sa formule générale est la suivante : $$F_\beta = \frac{(1+\beta^2) * \text{précision} * \text{rappel}}{(\beta^2 * \text{précision}) + \text{rappel}}$$

Choix de la fonction de coût

La tâche est centrée sur la classe positive (le client a des difficultés de paiement).
La précision et le rappel sont un bon point de départ.

- *Maximiser la précision minimisera les faux positifs* : le client n'a pas de difficultés de paiement mais est prédit comme tel. Ce qui signifie une perte de client pour la société de crédit (erreur de Type I).
- *Maximiser le rappel minimisera les faux négatifs* : le client a des difficultés de paiement mais est prédit comme n'en ayant pas. Ce qui engendre une perte de chiffre d'affaires pour la société de crédit (erreur de Type II).

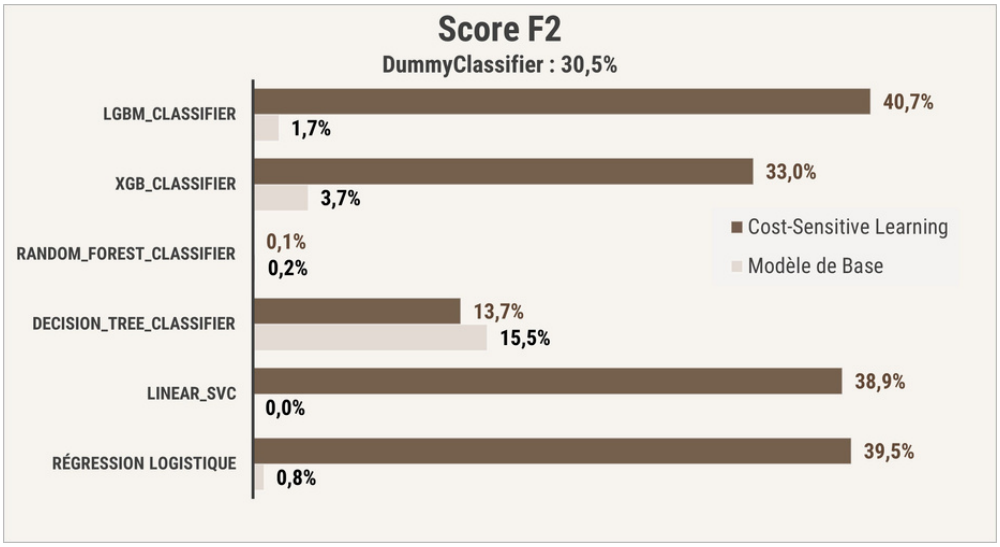
La difficulté ici réside dans le fait que les **faux négatifs sont plus dommageables que les faux positifs**.
Les faux négatifs sur cet ensemble de données sont des cas où un mauvais client est marqué comme un bon client et se voit accorder un prêt alors que les faux positifs sont des cas où un bon client est marqué comme un mauvais client et la société de crédit ne lui accorde pas de prêt. Les faux négatifs lui coûtent donc plus cher : Cost(FalseNegatives) > Cost(FalsePositives)

Autrement dit, nous nous intéressons à la F-mesure qui résumera la capacité d'un modèle à minimiser les erreurs de classification pour la classe positive tout en privilégiant les modèles qui minimisent les faux négatifs plutôt que les faux positifs. Ceci peut être réalisé en utilisant une version de la **F-mesure** qui calcule une moyenne harmonique pondérée de précision et de rappel mais **privilégie les scores de rappel supérieurs aux scores de précision**. C'est ce qu'on appelle la mesure **Fbeta**, une généralisation de la F-mesure, où bêta est un paramètre qui définit la pondération des deux scores.
La valeur de bêta dépend donc de manière indirecte de la fonction de perte que nous souhaitons appliquer sur chacun des deux cas erronés : les faux positifs et les faux négatifs.
Une valeur bêta de 2 accordera plus d'attention au rappel qu'à la précision et est appelée la mesure F2.

Prise en compte du déséquilibre des classes : Cost-Sensitive Learning

Comme l'objectif de l'apprentissage sensible aux coûts est de minimiser le coût d'un modèle sur l'ensemble de données d'apprentissage, où l'on suppose que différents types d'erreurs de prédiction ont une valeur Coût associée différente et connue, on peut dire qu'**il existe un lien étroit entre une classification déséquilibrée et un apprentissage sensible aux coûts.**
Plus précisément, un problème d'apprentissage déséquilibré peut être résolu à l'aide d'un apprentissage sensible aux 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.

Un bon point de départ pour les tâches de classification déséquilibrées consiste à **attribuer des coûts basés sur la distribution inverse des classes.**



Nous pouvons voir que 3 des modèles testés ont une mesure F2 supérieure à la valeur par défaut de prédire la classe majoritaire dans tous les cas (30.5%).
Les 3 meilleurs scores F2 sont obtenus avec les modèles **light GBM (41%)** puis la régression logistique (39.5%) et LinearSVC (38.9%).

Les 2 meilleurs modèles ont ensuite été réentraînés avec différents types de gestion du déséquilibre : échantillonnage par RandomOverSampler, SMOTE, ADASYN et BorderlineSMOTE.
Mais ces techniques n'ont pas amélioré les résultats de celle du "Cost-Sensitive Learning". En effet, un inconvénient général de l'approche est que des exemples synthétiques sont créés sans tenir compte de la classe majoritaire, entraînant éventuellement des exemples ambigus s'il y a un fort chevauchement entre les classes.

Le modèle final sélectionné est le Light Gradient Boosted Machine ou LightGBM en abrégé.

Optimisation de l'algorithme LightGBM

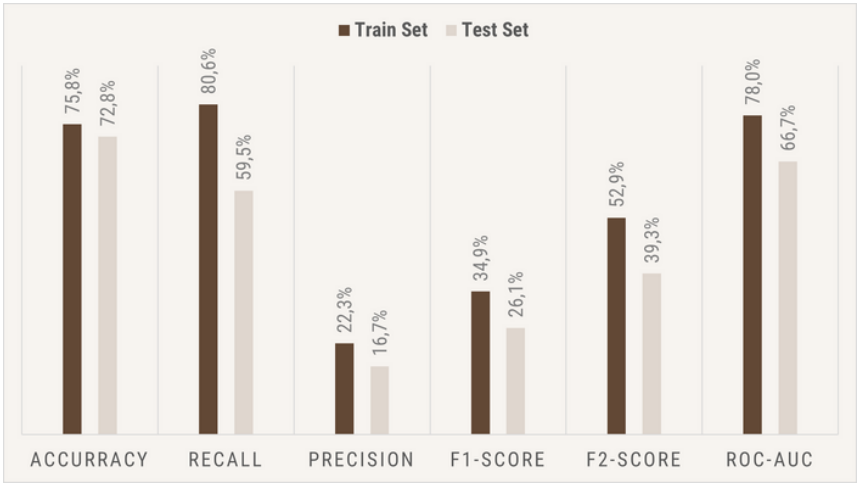
Il existe de nombreux hyperparamètres que nous pouvons examiner pour LightGBM. Ces hyperparamètres peuvent être regroupés en 4 catégories :

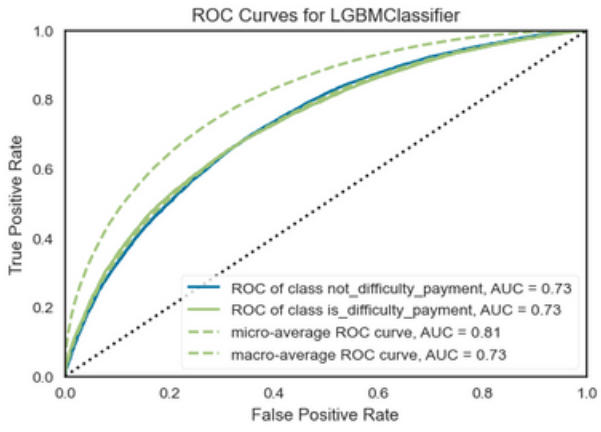
- Paramètres qui affectent la structure et l'apprentissage des arbres de décision
- Paramètres qui affectent la vitesse d'entraînement
- Paramètres pour une meilleure précision
- Paramètres pour lutter contre le surajustement

La plupart du temps, ces catégories se chevauchent beaucoup, et l'augmentation de l'efficacité dans l'une peut entraîner une diminution dans l'autre. C'est pourquoi **les régler manuellement est une erreur et doit être évité.**

Des frameworks comme **Optuna** peuvent automatiquement trouver le « juste milieu » entre ces catégories si on leur donne une grille de paramètres suffisamment bonne.
Nous avons réalisé 300 essais avec comme objectif la minimisation de la perte logarithmique (Log-loss) qui signifie que plus la probabilité de prédiction est éloignée de la valeur réelle, plus sa valeur de perte logarithmique est élevée.

Les "meilleurs paramètres" obtenus nous donnent les performances suivantes :





Le F2-score a augmenté de 40,7% à 52,9% sur notre jeu d'entraînement.

04 L'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.

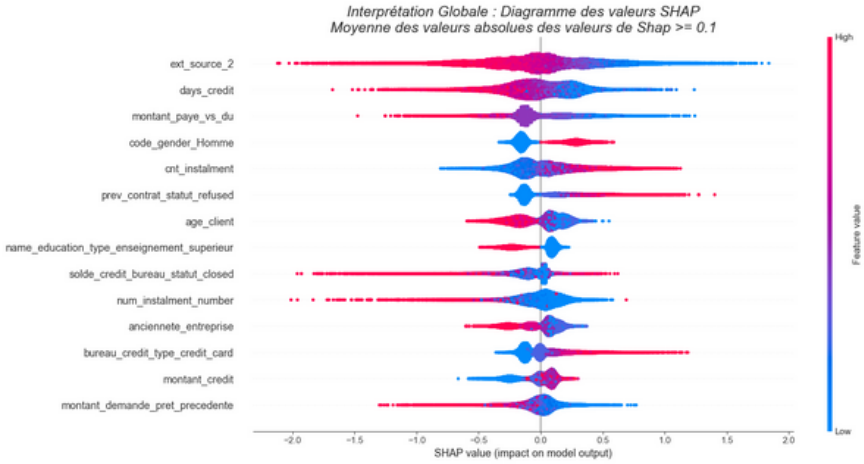
L'importance des variables mesure l'impact global de chaque descripteur dans le modèle. Elle peut être estimée en modélisation (sur l'échantillon d'apprentissage) ou en prédiction (sur l'échantillon test). Dans les deux cas, les principales étapes sont les mêmes :

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.

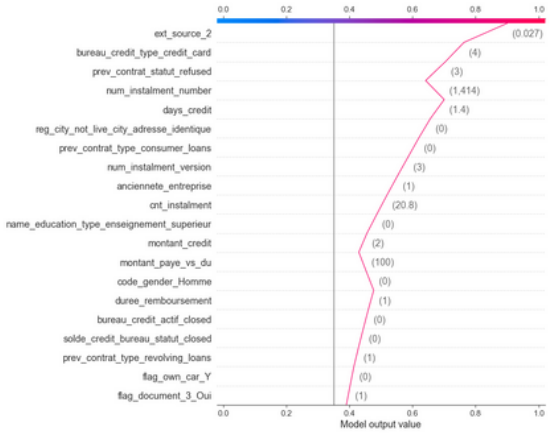
L'échantillon d'apprentissage permet de mettre en avant le rôle des variables dans la modélisation, alors que l'échantillon test éclaire sur l'influence de la variable sur les performances en généralisation (lorsque le modèle est déployé dans la population).

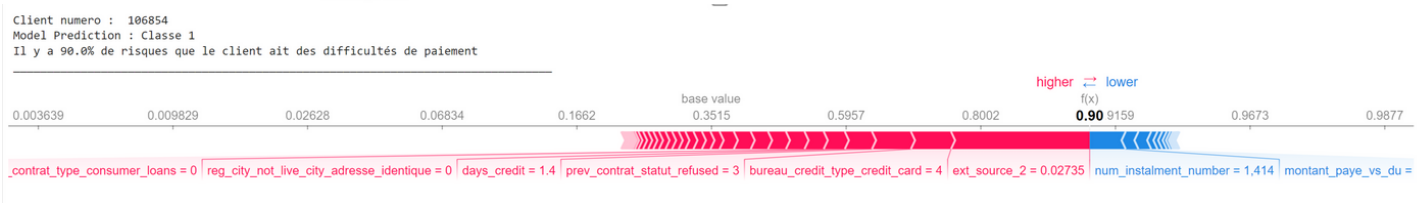
Les valeurs de Shapley calculent l'importance d'une variable en comparant ce qu'un modèle prédit avec et sans cette variable. Cependant, étant donné que l'ordre dans lequel un modèle voit les variables peut affecter ses prédictions, cela se fait dans tous les ordres possibles, afin que les fonctionnalités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

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 ?





05 Les limites et les améliorations possibles

La première limite dans ce projet provient de ma **méconnaissance du milieu bancaire** et de la finesse du vocabulaire. J'ai fait des sélections selon ma compréhension des variables et de leur impact potentiel mais sans forcément comprendre ce qu'elles impliquent. J'ai donc pu faire des sélections et des agrégations de variables incohérentes d'un point de vue "métier".
Il aurait donc été intéressant de disposer d'un véritable dictionnaire des variables et de pouvoir challenger ces choix auprès de *« Prêt à dépenser »* afin de **vérifier la cohérence du pré-process.**

De plus, le **choix et l'optimisation du modèle** de classification ont été réalisé sur la base d'une hypothèse forte concernant la **métrique d'évaluation** : le F Beta Score avec Beta fixé à 2 suivant certaines hypothèses non confirmées par le métier. 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.**
Pour cela 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.

Enfin, en termes d'amélioration du **Dashboard**, il serait intéressant de développer un dashboard avec une page « banque » et une page « client ». Cela permettrait à la personne de *« Prêt à dépenser »* qui explique la décision à un client d'avoir accès à certaines données permettant d'expliquer la réponse, sans nécessairement pouvoir les montrer au client. Par ailleurs, il serait intéressant de rajouter une partie interactive qui permettrait au client de voir quelle valeur sur quelle variable aurait pu lui permettre d'obtenir son crédit. En ce sens, on pourrait envisager une page « scenario » où l'on pourrait changer une ou plusieurs valeurs du profil du client et voir l'impact sur la réponse de la banque.

CLIQUER ICI POUR VOUS RENDRE SUR LE DASHBOARD