Implémenter un modèle de Scoring

Note Méthodologique

Projet n°7 - Data Scientist - OpenClassrooms – Yannick Quérin

Objectifs et Contexte:

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 du 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: Source

- *Dashboard interactif 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 (Streamlit Cloud). Tous les programmes sont accessibles sur Github: lien_github.

Sommaire

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 ce faire, les grands points qui seront évoqués et détaillés:
- Les phases préliminaires et méthodologie de la modélisation
- La fonction de cout métier (optimisation bayésienne), l'algorithme d'optimisation, ainsi que les métriques.
- L'interprétabilité du modèle (globale et locale).
- Les limites et pistes d'amélioration possibles

01

ETAPES PRELIMINAIRES A LA MODELISATION

Les 8 fichiers au format csv sont fournis et composent notre jeu de données. Ils contiennent 218 informations bancaires et personnelles anonymisées pour 307511 clients recueillies auprès de Home Crédit Group et auprès d'autres institutions financières.

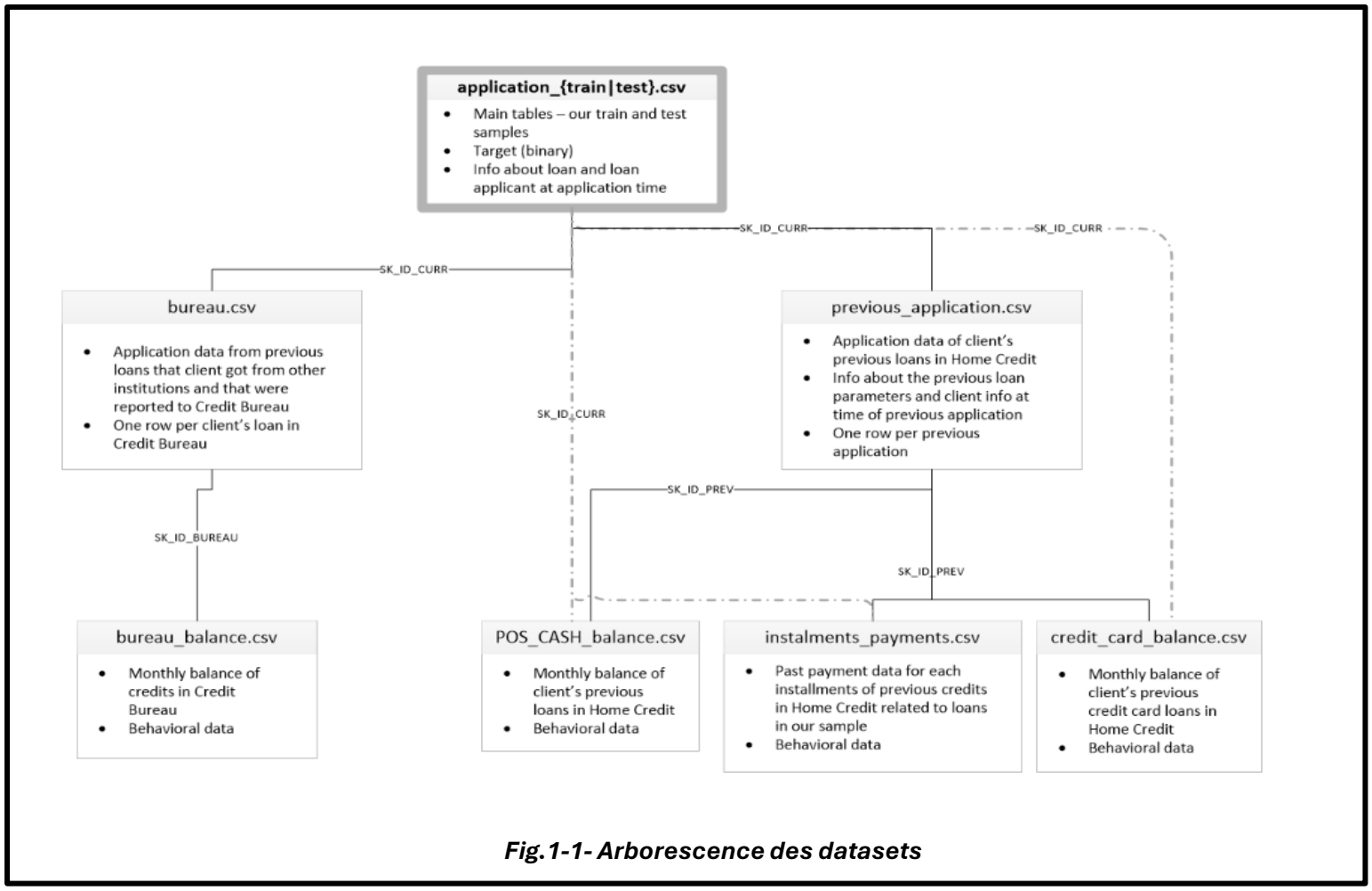


Fig.1-1- Arborescence des datasets

Globalement, la majorité des 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, les données sont au centre des algorithmes de Machine Learning et la plupart du temps, les *Data Set* proviennent avec des ordres de grandeurs différents ou dues à des données incorrectes ou incomplètes. Cette différence d'échelle peut conduire à des performances moindres. Il est donc souvent indispensable d'établir une stratégie de pré-traitement des données – autrement appelé Data Preprocessing – à partir de nos données brutes ,préparer au mieux ces données, permettra d'avoir de meilleures performances. Pour remédier à cela, des traitements préparatoires sur les données existent.

Data Cleaning

Cette étape consiste à un nettoyage des données incomplètes, incorrectes ou manquantes tel que les phases de:

- Optimisation de la mémoire en modifiant le typage des données (passage d'un format '*int64*' en '*int32*')
- Correction des valeurs aberrantes
- Imputation des données manquantes

Nous d'abord opéré à la transformation des valeurs aberrantes issues de l'EDA ou suppression valeurs uniques du train set ≠ test set, puis de la conservation des variables importantes de l'EDA (suppression des variables fortement colinéaires: seuil à 80 %), de la suppression des NaN ou des valeurs nulles sur les données manquantes pour terminer enfin sur la phase d'imputation en 2 étapes: médiane, NaNImputer. Ce procédé est itéré à la fois sur les données d'entrainement et test.

Data Transformation

Sur les 8 datasets sont appliquées la phase de ***Feature engineering :*** construction de nouvelles caractéristiques par méthode d'agrégation statistique (*Min*, *Max*, ...) , d'ajouts de variables métiers liés à la problématique du secteur bancaire, avec d'autre part, la transformation de variables catégorielles en variables numériques interprétables pour le modèle de Machine Learning (***One-hot encoder:*** consiste à encoder une variable à n états dont un seul prend la valeur 1, le numéro associé au segment catégorielle de la variable valant 1 étant le numéro de l'état pris par la variable; ***Label encoding***: attribution d'un nombre entier à chaque catégorie unique d'une variable catégorielle. Aucune nouvelle colonne n'est créée). Ce procédé se termine par 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.

Feature Selection

Dans le cadre de la construction de notre modèle prédictif, nous souhaitons conserver les variables qui contribuent le plus à notre variable cible ***TARGET*.** C'est là qu'intervient la sélection des caractéristiques, dont les méthodes connues et utilisées: la ***Permutation Importance*** qui mesure l'importance d'une caractéristique par rapport à la prédiction générale d'un modèle. En d'autres termes, dans quelle mesure le modèle serait affecté si vous supprimez sa capacité à tirer un enseignement de cette caractéristique. La métrique peut vous aider à affiner un modèle en modifiant les caractéristiques et les algorithmes à inclure. L'autre façon intervient avec l'algorithme ***LightGBM*** grâce à la potentielle multicolinéarite des variables: en effet, l'importance sera répartie entre les caractéristiques corrélées, et la somme des importances sera égale à l'importance de la caractéristique originale.

02

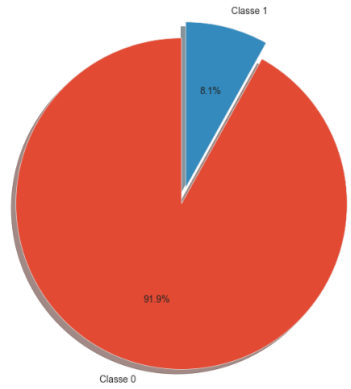
METHODOLOGIE SUR L'ENTRAINEMENT DU MODELE

Interprétation de la problématique

La variable cible à prédire prend 2 valeurs et est fortement déséquilibrée (8/92) :

⇨ **0 – positive – non défaillant** :
indique que le client a totalement remboursé son prêt

⇨ **1 – négative – défaillant** :
indique que le client n'a pas remboursé son prêt en totalité ou en partie



MODELISATION – Classification Binaire
CIBLE – Score de défaut de paiement (proba)
8 DATASETS - 218 VARAIBLES– 307511 CLIENTS

Modélisation

Rééchantillonnage des classes déséquilibrées: SMOTE, ADASYN

Méthodes de rééquilibrage

Une classe déséquilibrée peut avoir un impact négatif sur le modèle qui aura tendance à prédire la classe majoritaire (donc client non défaillant). Une modification de l'ensemble de données est possible avant d'entraîner le modèle prédictif afin d'équilibrer les données : le ***rééchantillonnage (re-sampling).***

Deux méthodes principales existent pour égaliser les classes :
- le sur-échantillonnage (***Oversampling***)
- le sous-échantillonnage (***Undersampling***)

- SMOTE & ADASYN:

Les 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.

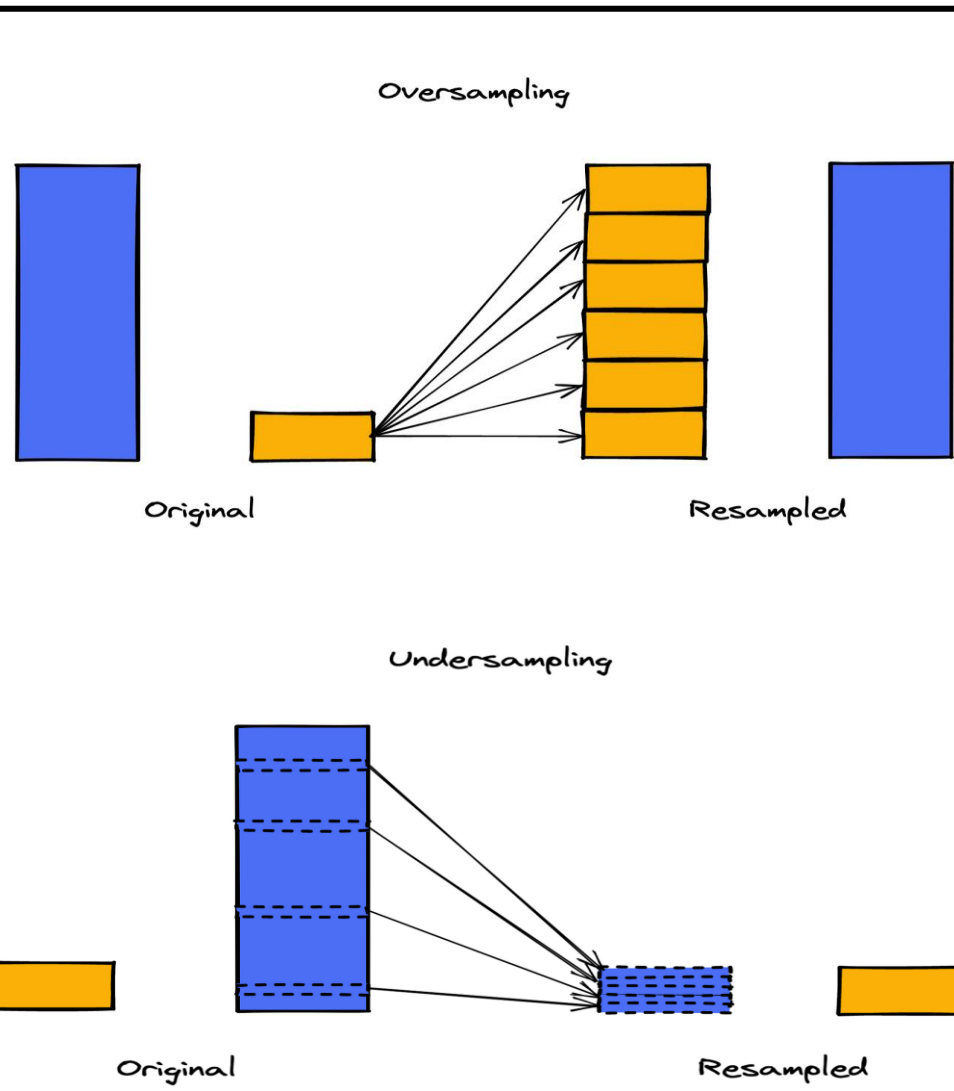


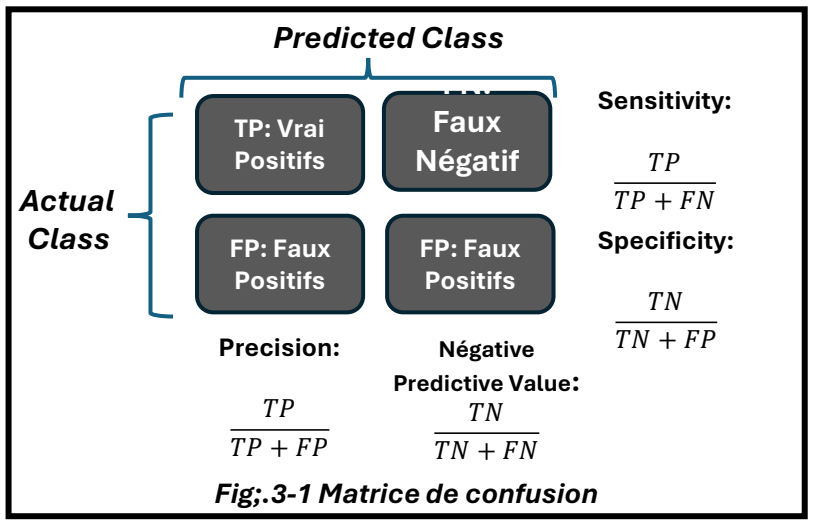
Fig.2-1-Fonctionnement du sur-échantillonnage, et du sous-échantillonnage

03 LA FONCTION COUT METIER, LES METRIQUES ET L'ALGORITHME D'OPTIMISATION

Bien choisir les métriques pour l'évaluation des modèles

En Machine Learning, le choix des métriques s'avère primordiale, et permet d'évaluer la performance du modèle prédictif et de garantir la qualité du modèle de classification. Ces métriques permettent de comparer les classes réelles aux classes prédites par le modèle.

En effet, En classification binaire il est d'usage d'utiliser le pourcentage de bonnes prédictions comme score. Cependant, 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.



Fig;.3-1 Matrice de confusion

•Pour notre problématique :

- Les **défaillants** forment la classe **positive**.
- Les **non-défaillants** forment la classe **négative**.

•Pour minimiser les pertes d'argent, nous devons :

- Nous efforcer de ne pas prédire un client non-défaillant s'il est défaillant ==> **minimiser le nombre de faux négatifs (erreur de type II)** (prédit non-défaillant mais client défaillant). Si un défaillant est prédit non-défaillant, le groupe Home Crédit aura perdu toute la somme prêtée à l'emprunteur. Cela constitue les plus grosses pertes pour l'entreprise.
- Nous efforcer de ne pas prédire de défaillant si le client n'est pas défaillant donc **minimiser les faux positifs (erreur de type I)** (classe 1 défaillant alors que non-défaillant dans la réalité). Si un non-défaillant est prédit défaillant, le groupe Home Crédit aura perdu les intérêts de la somme prêtée à l'emprunteur.

Métriques:

- **Recall** : la métrique pour déterminer le **taux de vrais positifs** est le Rappel (Sensibilité)/**Recall**. Elle mesure parmi toutes les observations positives combien ont été classées comme positives. Pour ne pas avoir de pertes, il faut détecter tous les défaillants (classe positive), donc **maximiser la métrique recall** :

$$\frac{TP}{TP+FN}$$

- **Precision** : elle mesure le nombre d'observations prédites comme positives (client défaillant) qui le sont en réalité. Si le client est prédit défaillant alors qu'il ne le sera pas, le prêt ne sera pas accordé et les intérêts ne seront pas empochés. Il faut donc **maximiser la Precision**:

$$\frac{TP}{TP+FP}$$

- **F-mesure ou F1** :
  La mesure de la F1 Score est définie comme la moyenne harmonique de la précision (precision) et du rappel (recall). Elle est donnée par la formule suivante :

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Ensemble, cela forme la mesure F1, qui combine à la fois la précision et le rappel en une seule métrique.

- Si on augmente le recall, la précision diminue. Il s'agit de faire un compromis entre ces deux métriques qui dépendent l'une de l'autre, en prenant en compte le métier/problème de l'entreprise et les coûts associés pour répondre à ces questions.

- Dans notre cas, il faut trouver le plus grand nombre d'observations réellement positives (client prédit défaillant et bien défaillant) et la perte est moins grande si on prédit un client défaillant mais qu'il ne l'est pas (faux positifs). Donc, **on donnera priorité à maximiser le recall au détriment de la précision** (on parle bien de précision, pas d'accuracy).

• Le réglage du paramètre beta pour le Fbeta score permet de donner plus de poids au recall (β>1) qu'à la précision (0<β<1).

• **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 :

$$F2\text{-}measure = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}$$

Séparation des données en entraînement/validation

Le jeu de données est séparé en deux :

• en un jeu d'entraînement (80%) servant à entraîner le modèle,
• et en un jeu de validation permettant d'évaluer la performance des différents modèles testés.

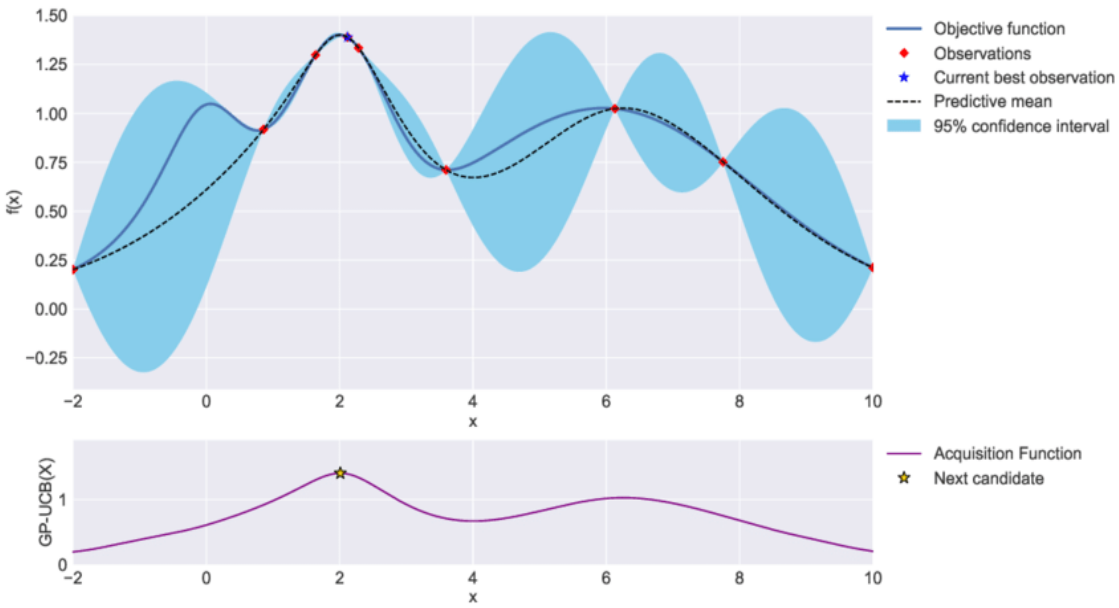
A noter : lors de la séparation, les 2 jeux de données devront conserver la répartition de départ des classes majoritaires (les clients non défaillants) et minoritaires (les clients défaillants).

Optimisation des hyperparamètres du modèle LightGBM

La technique retenue pour l'optimisation des hyperparamètres du modèle LightGBM est l'optimisation Bayésienne avec 3 librairies différentes (***bayes_op***t du MIT, ***skopt*** de scikit-learn et ***optuna***).

L'optimisation bayésienne est une méthode de réajustement automatisé (***tuning***) des hyperparamètres. Le réajustement automatisé des hyperparamètres vise à trouver les meilleures valeurs d'hyperparamètres pour un modèle d'apprentissage automatique sur un ensemble de données donné, sans que le data scientist n'ait à intervenir au-delà de la configuration initiale requise.

L'optimisation bayésienne utilise le raisonnement bayésien pour construire un modèle de probabilité de la fonction objective: $P(score|hyperparamètres)$ qui est ensuite utilisé pour sélectionner les prochaines valeurs d'hyperparamètres à évaluer.



Fig;.3-2 Fonctionnement de l'optimisation bayésienne

Le concept consiste à utiliser davantage d'itérations de recherche pour évaluer des valeurs d'hyperparamètres prometteuses en raisonnant à partir des résultats antérieurs. Il s'agit d'une méthode intuitive d'optimisation des hyperparamètres qui fonctionne à peu près de la même manière qu'un être humain pour s'améliorer dans n'importe quelle situation : apprendre à partir des expériences passées ! Si tout fonctionne comme prévu, l'optimisation bayésienne des hyperparamètres peut se traduire par une meilleure performance de généralisation sur l'ensemble de test avec moins d'itérations qu'une recherche aléatoire ou en grille, du type ***Grid Search*** ou ***Random Search***.

L'optimisation a été sur différentes métriques (Roc Auc, PR Auc, F10, Recall et la métrique métier) pour différents jeux de données (rééquilibrés avec ***Smote***, hyperparamètre class_weight de ***LightGBM*** ou non équilibrés, standardisés ou non…). Le but minimiser le nombre de faux négatifs tout en prédisant le plus de vrais positifs possibles tout en limitant le nombre de faux positifs. Le modèle LightGBM avec les paramètres de base sert de comparatif.

Modèle LightGBM optimisé final

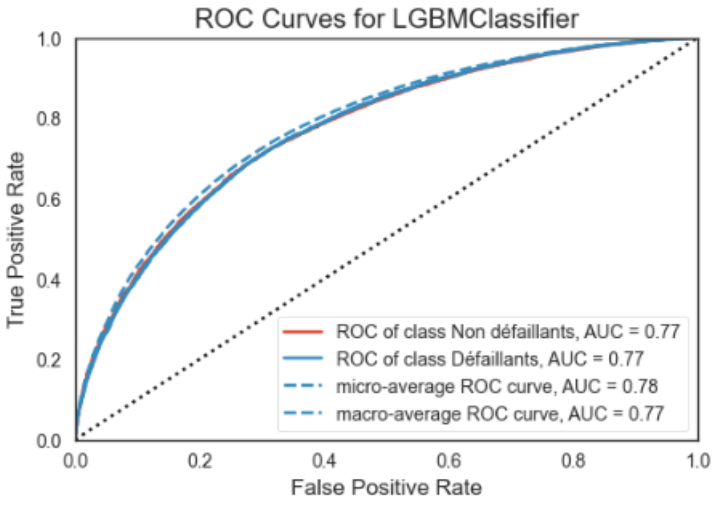
À la suite de différents tests selon les métriques et méthodes d'optimisation bayésienne, nous avons obtenu un classement des modèles ayant le score F10 le plus élevée, avec le taux de FN le plus bas possible.

Modèle	Jeu_donnees	FN	FP	TP	TN	Metrique	Optimisation	Class_weight	Rappel	Précision	F1	F5	F10	ROC_AUC	
lgbm_optuna_opt_F1	train	1500	16369	3465	40168	F1	optuna	oui	0.697885	0.174700	0.279447	0.625803	0.677788	0.770713	(
lgbm_optuna_opt_F5	train	1508	16182	3457	40355	F5	optuna	non	0.696274	0.176027	0.281011	0.625205	0.676479	0.771812	(
lgbm_optuna_opt_F10	train	1514	15871	3451	40666	F10	optuna	non	0.695065	0.178605	0.284185	0.625499	0.675720	0.776303	(
lgbm_optuna_opt_10_train	train	1544	15089	3421	41448	F10	optuna	oui	0.689023	0.184819	0.291459	0.623592	0.670902	0.780011	(

Fig;.3-3: Résultats optimisation LightGBM

```
LGBMClassifier(class_weight='balanced', colsample_bytree=0.883696173865355,
               force_col_wise=True, max_depth=4, min_child_samples=37,
               min_child_weight=0.9053832802852111, n_jobs=-1, num_leaves=8,
               objective='binary', reg_alpha=0.0013343227256418153,
               reg_lambda=1.1168060057563535e-06, subsample=0.876335534267455,
               subsample_freq=4, verbosity=-1)
```

Fig;.3-4: Hyperparamètres



Fig;.3-5:Courbe ROC

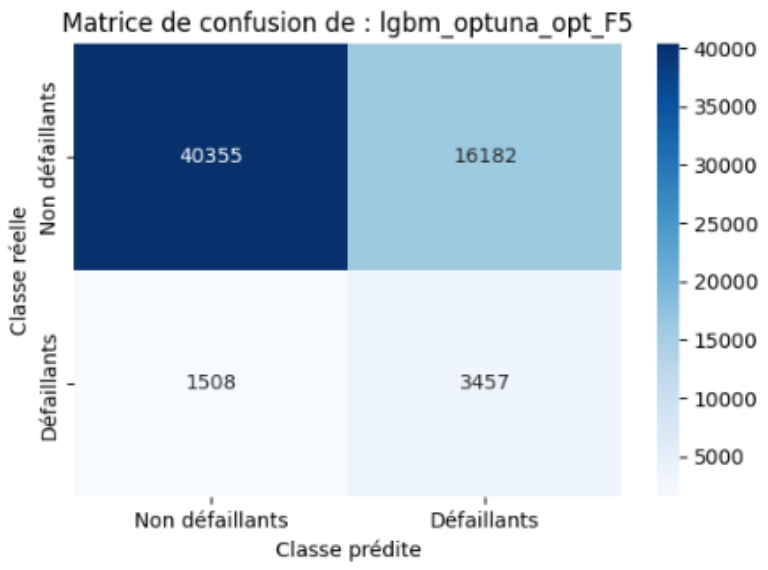


Fig. 3-6: Matrice Confusion

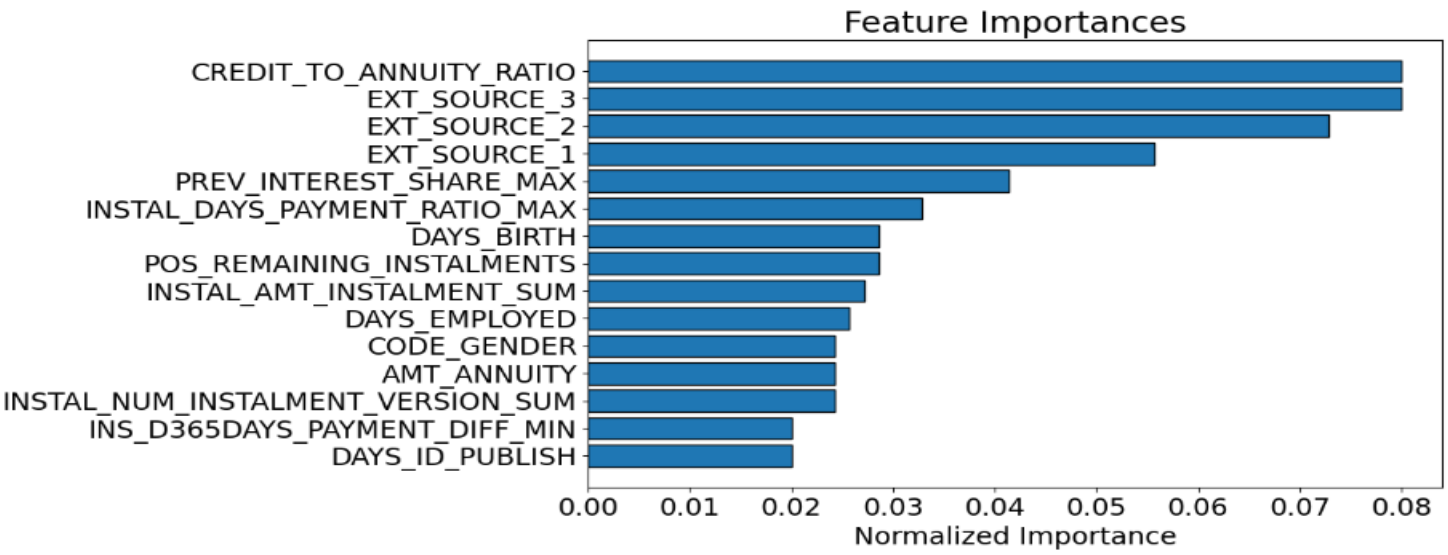
Le modèle le plus performant suivant les métriques, et nos critères de minimisation s'avère être le modèle optimisé avec la méthode bayésienne ***Optuna*** avec la ***métrique F5***, et l'hyperparamètre fixé pour le rééquilibrage des classes ***class_weight = 'balanced'.*** Il détecte le moins de faux négatifs, le plus de vrais positifs mais un taux plus élevé de faux positifs. Un compromis est à faire entre le taux de faux négatifs et le taux de faux positifs, une augmentation de l'un entraîne une diminution de l'autre…une collaboration sera nécessaire pour décider du réglage du seuil de décision d'un client défaillant (fixé par défaut à 0.5) et de la proportion de faux positifs acceptables

INTERPRETABILITE DU MODELE

Importance normalisée des variables

L'interprétation des modèles est importante en machine learning, afin d' expliquer – justifier – la prise de décision induite par le modèle prédictif. De ce fait, le sujet autour de la compréhension métier, et l'identification des features s'avère primordiale dans le but d'appréhender le mécanisme d'affectation de chaque feature, variable sous-jacent au modèle prédictif, ayant une importance relative (positive ou négative).

En effet, à partir de notre modèle LightGBM, nous avons pu pour chaque variable de calculer l'importance de cette variable pour le modèle. Une fois normalisées, nous pouvons comparer l'importance relative de chacune des variables et par un simple tri, afficher les 15 premières variables les plus importantes.



Fig;.4-1 - 15 variables les plus importantes

Parmi ces variables nous retrouvons les variables les plus corrélées avec la variable cible détectées lors de l'analyse exploratoire :

- Les informations ***bancaires*** en particulier ***CREDIT_ANNUITY_RATIO*** (ratio du montant du crédit du prêt sur l'annuité de prêt font partie des informations).

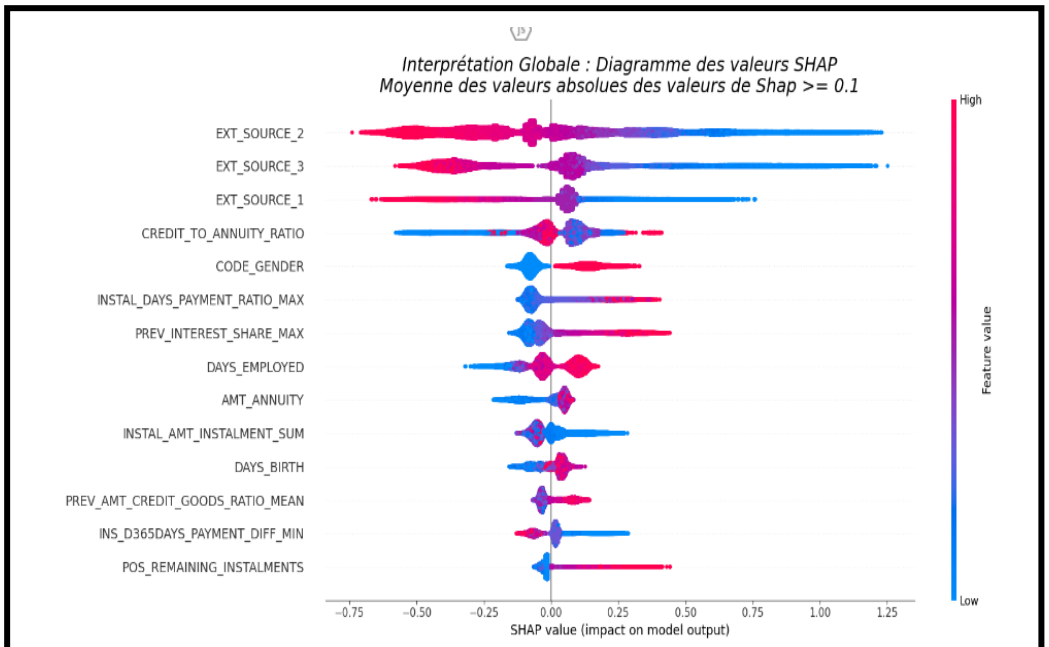
- Les ***données externes*** : EXT_SOURCE_1 EXT_SOURCE_2 et EXT_SOURCE_3.

- Les informations sur ***le calcul du taux d'intérêt***: ***PREV_INTEREST_SHARE_MAX*** (durée du crédit précédent (année précédente) pondérée avec la variation entre le montant de l'annuité et celui du crédit).

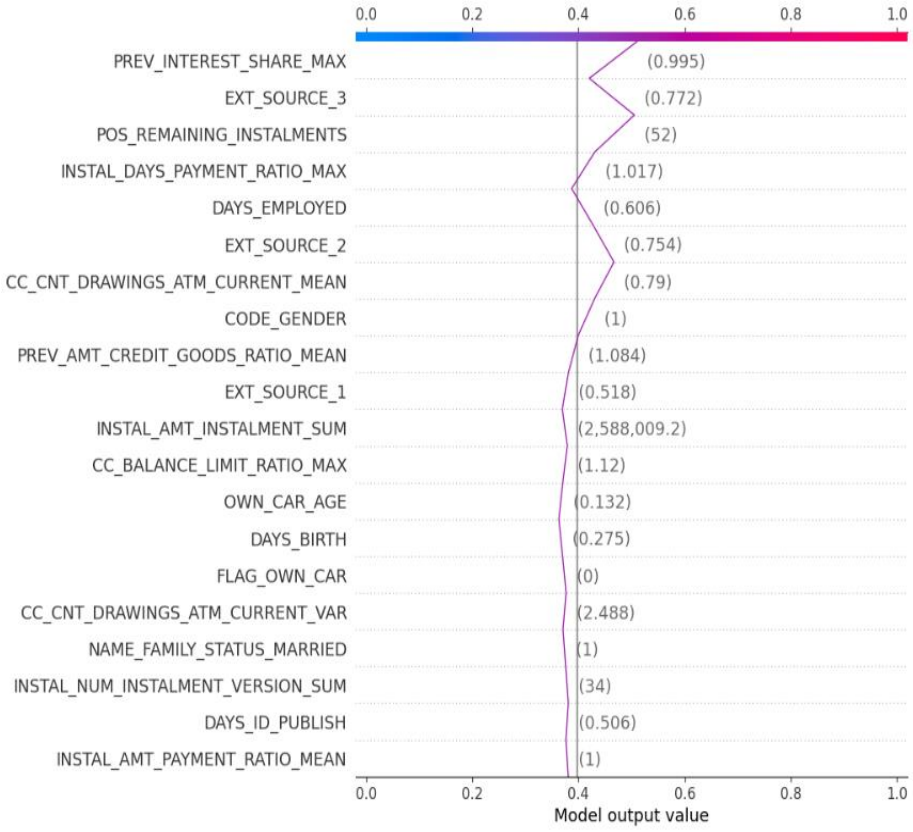
Interprétation globale et locale du modèle

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'***interprétabilité 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 features affectent le comportement global d'un modèle d'allocation de prêt ?



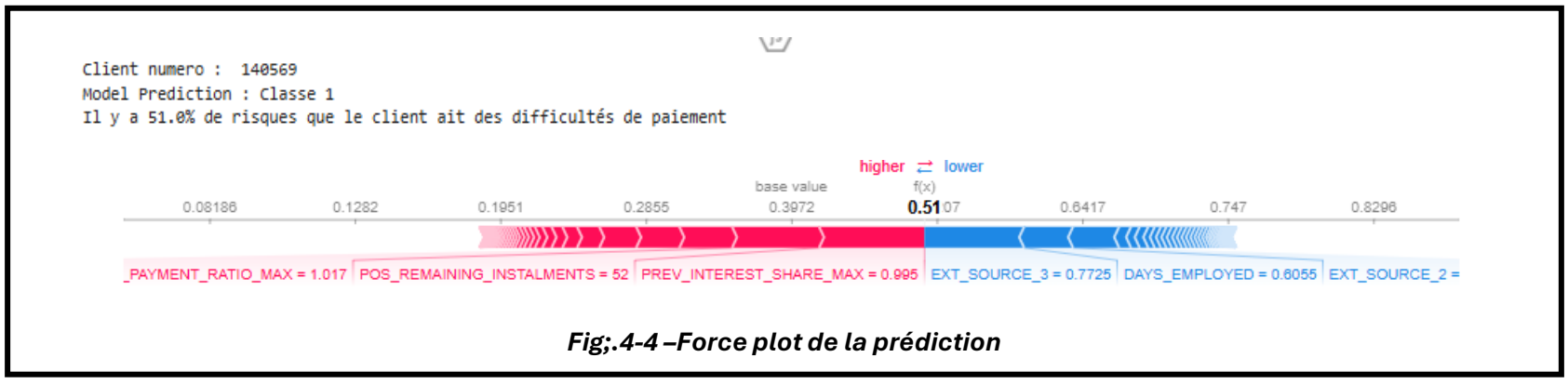
Fig;.4-2 –Diagramme valeurs Shapley



Fig;.4-3 –Interprétabilité locale des features

D'autre part, l' ***interprétabilité locale***, consiste à expliquer la prévision ***f(x)*** d'un modèle pour un individu x donné, dans notre cas métier pour un identifiant client donné. Par exemple, pourquoi la demande de prêt d'un client a-t-elle été approuvée ou rejetée ?

Ainsi, on peut visualiser pour une caractéristique, ou feature donnée sa probabilité correspondante qui tire dans le sens positif ou négatif dans la contribution à la probabilité de défaut de paiement.



Fig;.4-4 –Force plot de la prédiction

05

LIMITES ET PISTES POSSIBLES

Pour répondre au problème de classification binaire à partir des 8 datasets fournis par Prêt à dépenser, de nombreuses techniques de Machine Learning ont été nécessaires : rééquilibrage des classes, création de nouveaux features facilement explicables, sélection des variables pour rendre le modèle moins complexe, choix des métriques adaptées à notre problématique métier, réflexion sur le compromis taux de faux négatifs et taux de faux positifs et sur le réglage du seuil de décision

La première limite dans ce projet provient de ma méconnaissance du milieu bancaire et de la finesse du vocabulaire En effet, Une explication des données externe serait un plus puisqu'il est difficile d'être transparent en utilisant ces variables importantes pour le modèle mais inexplicables pour le client.

De plus, une collaboration avec l'équipe métier serait un gage d'amélioration du modèle, en termes d'affinement au niveau de la métrique d'évaluation, et sur la fonction de cout. Les experts métiers pourraient nous aider à créer une métrique bancaire plus efficace et adaptée et pourraient nous donner leur avis sur l'intérêt des nouvelles variables créées et pourquoi pas nous indiquer de nouvelles variables.

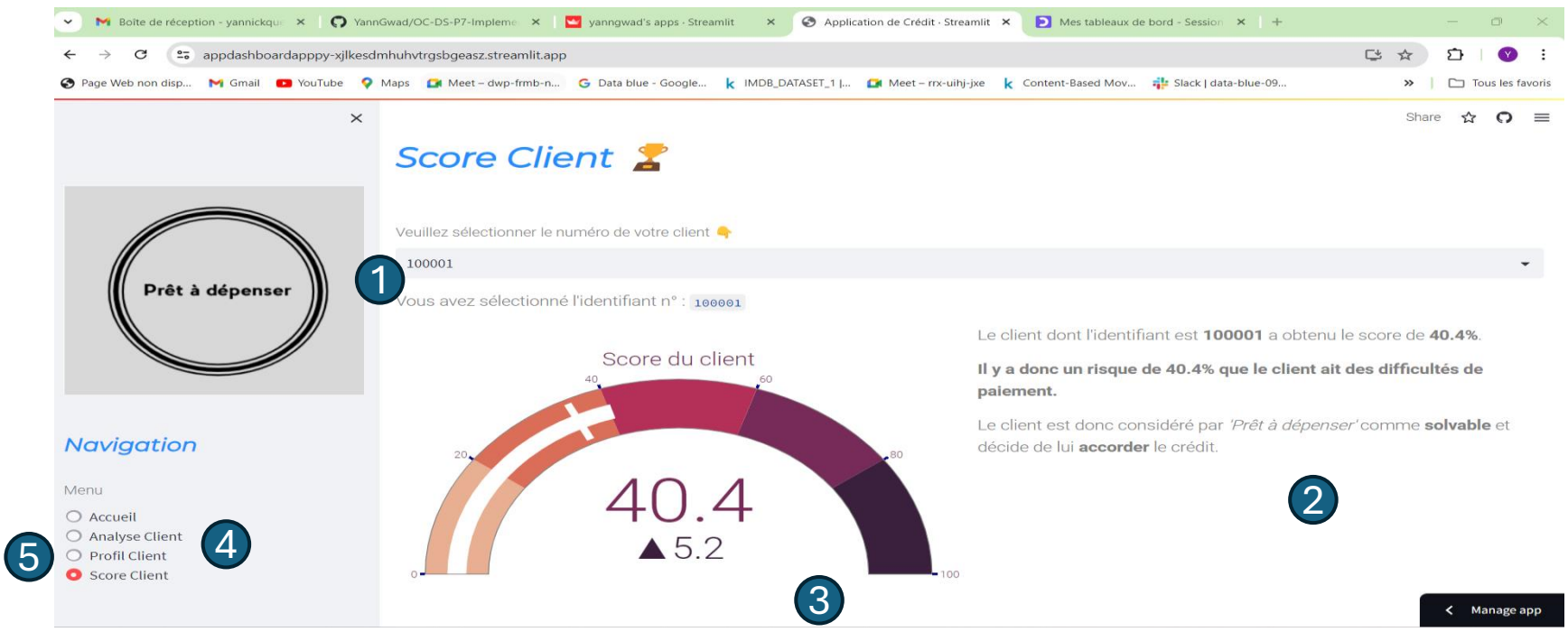
Un ***compromis*** est indispensable à effectuer entre la part de faux positifs et celle de faux négatifs. Le choix du seuil final d'acceptation ou refus de crédit à un poids important dans ce compromis, puisqu' augmenter le seuil tend à augmenter le nombre de crédits refusés, et donc augmenter le nombre de faux positifs alors que baisser le seuil à 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 au niveau de l'amélioration du Dashboard, Développer un dashboard avec une page ***Données bancaires*** coté décideur et une page ***Données Clients*** de façon à séparer, et sécuriser certaines données sensibles détenus uniquement du côté de la banque, sans pour autant certaines données au client.

Entrevoir une section de ***recommandation*** qui permettrait au client de voir quelle variable aurait pu influencer sur son obtention ou pas du prêt en question.

Présentation du Dashboard

Accès streamlit: https://yannickquerin-p07-dashboard.streamlit.app/
Accès Dépôt des sources*: https://github.com/YannickQuerin/OC-DS-P7-Implementez_modele_scoring_dashboard/blob/main/P7_Modelisation_risque_defaut_credit/App_dashboard_streamlit.py*



Fig;.5-1 – Extrait Dashboard

1. Liste de sélection d'un client parmi des nouveaux clients pour décision d'octroi de prêt (jeu de données application_test.csv prétraité de la même manière que le jeu de données d'entraînement ayant servi à entraîner le modèle).

2. Informations personnelles et bancaires minimales pour le client sélectionné

3. Jauge permettant de visualiser rapidement le score du patient

4. Volet 'Analyse Client': Graphes d'importance sur les features, Impact de chaque caractéristique sur la prédiction (diagrammes des valeurs Shap)

5. Volet 'Profil Client': descriptif du profil socio-économique, profil emprunteur, comparaison du profil client à celui des groupes de typologies clients.