Présentation des modèles

DummyRegressor

Le DummyRegressor est une approche simple de régression qui prédit la valeur moyenne, médiane ou la plus fréquente des caractéristiques d'entraînement, ou utilise une constante prédéfinie. Il est utilisé comme baseline pour comparer avec d'autres modèles de régression plus complexes. La stratégie 'stratified' prédit un échantillon de manière probabiliste (probabilité des classes).

AdaBoost

AdaBoost, ou "Adaptive Boosting", est un algorithme d'apprentissage ensembliste qui combine plusieurs modèles faibles pour former un modèle robuste.

Les paramètres clés :

n_estimators **:** le nombre de modèles faibles à utiliser,

learning_rate **:** affecte la contribution de chaque modèle au modèle final,

algorithm **:** 'SAMME' ou 'SAMME.R', affectant la manière dont les poids sont ajustés

LightGBM

LightGBM est une implémentation efficace de l'algorithme de gradient boosting qui utilise des arbres basés sur des histogrammes pour accélérer l'entraînement et réduire la consommation de mémoire.

Les paramètres clés :

n_estimators : le nombre d'arbres,

max_depth : la profondeur maximale de chaque arbre,

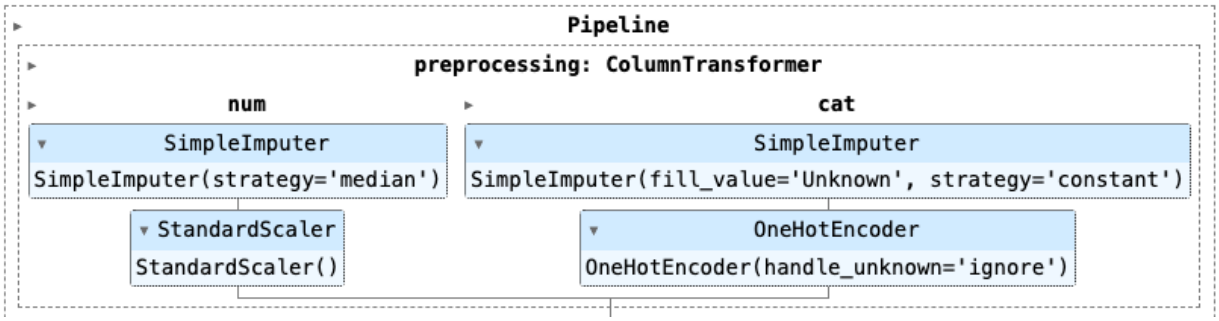
learning_rate : la taille du pas,

subsample : fraction d'échantillons utilisés pour l'entraînement de chaque arbre, colsample_bytree : fraction de caractéristiques à utiliser par arbre

num_leaves : le nombre maximum de feuilles dans chaque arbre

reg_alpha et reg_lambda aident à contrôler le surajustement

La méthodologie d'entraînement du modèle



1. **Imputation des données d'entraînement**

   Pour les variables numériques :

   SimpleImputer(strategy='median') : les valeurs manquantes sont remplacées par la médiane de la colonne. Cela permet de gérer les données manquantes en attribuant une valeur centrale qui ne dévie pas trop les estimations de tendance centrale de la distribution.

   Pour les variables catégorielles

   SimpleImputer(fill_value='Unknown', strategy='constant') : les valeurs manquantes sont remplacées par la c haîne de caractères "Unknown". Cette stratégie attribue une catégorie fixe aux valeurs manquantes, ce qui permet de les traiter comme une catégorie à part entière dans les analyses ultérieures.

2. **Standardisation des données d'entraînement**

   Pour les variables numériques

   StandardScaler() : les données sont normalisées en retirant la moyenne et en divisant par l'écart-type. Cette étape assure que les variables numériques contribuent équitablement à l'analyse en mettant toutes les variables à la même échelle.

   Pour les variables catégorielles

   OneHotEncoder(handle_unknown='ignore') : les variables catégorielles sont transformées en un format numérique où chaque catégorie est représentée par une colonne binaire. L'option handle_unknown='ignore' permet d'ignorer les catégories inconnues lors de la transformation, évitant ainsi des erreurs si de nouvelles catégories apparaissent dans les données de test ou après le déploiement du modèle.

3. **Validation croisée (StratifiedKFold) – Nombre de folds = 5**

   Utilisation de la validation croisée stratifiée (StratifiedKFold) avec 5 subdivisions Pour les tâches de classification présentant un déséquilibre significatif des classes cibles, comme c'est le cas dans ce projet, il est crucial d'assurer que chaque subdivision de la validation croisée reflète la composition globale de l'ensemble des données. C'est pourquoi les données ont été distribuées de sorte que chaque subdivision maintienne une proportion similaire d'échantillons par classe, en conformité avec le jeu de données entier. Cette approche est connue sous le nom de stratification, d'où l'utilisation de la méthode "StratifiedKFold" de la bibliothèque scikit-learn.

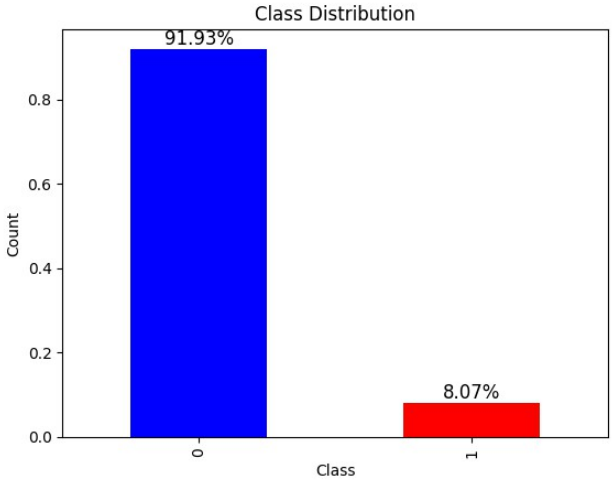
4. **Intégration de l'hyperparamètre seuil (threshold)**

   Ajustement du seuil d'hyperparamètre (threshold) Les modèles de classification produisent une probabilité qui indique si un prêt peut être accordé à un client. Habituellement, le seuil par défaut est fixé à 0,5. Toutefois, il est possible que ce seuil ne soit pas le plus approprié dans certains cas. En modifiant ce seuil, on peut affiner l'équilibre entre les faux positifs et les faux négatifs pour optimiser les performances du modèle.

5. **Enregistrement des résultats avec MLflow**

   Sauvegarde des performances avec MLflow MLflow est un outil de gestion du cycle de vie des modèles de machine learning. Il offre la possibilité de documenter, suivre et comparer les différents essais et configurations de modèles, contribuant ainsi à leur reproductibilité et facilitant leur déploiement. Dans le cadre de ce projet, MLflow est utilisé pour consigner les hyperparamètres, les performances et les modèles eux-mêmes, qui sont sauvegardés au format pickle.

Le traitement du déséquilibre des classes



Dans la gestion du déséquilibre de classes, nous sommes confrontés à des données de formation qui présentent une disproportion entre les classes. Ce déséquilibre induit un biais lors de l'entraînement des modèles, car la classe majoritaire est plus présente pendant l'apprentissage. Pour remédier à ce déséquilibre, plusieurs stratégies peuvent être employées : diminuer le nombre d'exemples dans la classe majoritaire, augmenter le nombre d'exemples dans la classe minoritaire, ou encore utiliser un facteur de correction interne si le modèle l'autorise.

Diminuer les cas dans la classe majoritaire consiste à réduire sa taille afin d'équilibrer les classes. Cela peut impliquer la suppression aléatoire d'individus jusqu'à obtenir une parité.

Pour augmenter les cas dans la classe minoritaire, on peut se servir d'outils comme ceux de la bibliothèque Imbalanced Learn, qui créent de nouveaux éléments en combinant les caractéristiques d'éléments existants, en s'appuyant notamment sur les techniques de plus proches voisins.

La littérature suggère que la combinaison d'une augmentation de la classe minoritaire avec une réduction de la classe majoritaire tend à donner de meilleurs résultats. L'utilisation de techniques comme SMOTE nécessite l'absence de valeurs manquantes, ce qui nous a amenés à utiliser un pipeline pour remplacer les valeurs manquantes par la médiane.

Voici les résultats obtenus pour le modèle LightGBM

LightGBM	Undersampling	201	0.70	0.27	0.39	0.31	0.492	299000
	SMOTE	302	0.72	0.28	0.35	0.31	0.652	319000
	SMOTE 0.6	251	0.70	0.1	0.91	0.19	0.492	299000
	SMOTE 0.4	162	0.71	0.24	0.432	0.31	0.566	281000
	SMOTE 0.2	193	0.69	0.23	0.39	0.29	0.070	245000

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

La fonction de coût du métier

L'objectif principal d'une banque lors de l'attribution de crédits est d'optimiser ses profits.

Toutefois, deux cas peuvent engendrer des pertes financières :

- **Les faux positifs (FP)** : des situations où la banque prédit à tort qu'un client sera défaillant dans le remboursement de son prêt.
- **Les faux négatifs (FN)** : des cas où la banque considère, à tort, qu'un client remboursera le crédit alors qu'il finira par être en défaut de paiement.

Il est important de reconnaître que le coût d'un faux négatif est considérablement plus élevé pour la banque qu'un faux positif. Pour illustrer, on peut envisager que le coût lié à un FN soit dix fois plus important que celui d'un FP.

Par ailleurs, les vrais positifs (TP) représentent les cas où la banque a correctement identifié un client qui ne remboursera pas son prêt, et les vrais négatifs (TN) ceux où la banque a correctement identifié un client qui remboursera son prêt. Dans ce contexte, l'objectif est de minimiser une fonction appelée "coût du métier", qui est définie par la formule suivante :

Coût du métier = Somme(-10*FN - FP + TN)

L'algorithme d'optimisation

Dans le domaine de l'apprentissage automatique, l'optimisation des hyperparamètres est essentielle pour améliorer les performances d'un modèle. La méthode de l'arbre de Parzen, ou l'estimateur structuré en arbre Parzen (Tree-structured Parzen Estimator, TPE), est une technique couramment utilisée pour cette optimisation.

Le TPE adopte une stratégie bayésienne, évaluant la probabilité que certains hyperparamètres mènent à de bonnes performances, basée sur les résultats des tests précédents. Afin de guider cette optimisation, une fonction "objective", que l'on cherche à minimiser ou à maximiser, est définie. Le TPE affine ses recommandations au fur et à mesure des itérations, se concentrant sur les zones de l'espace des hyperparamètres où les performances sont susceptibles d'être les meilleures. Le package Hyperopt a été utilisé pour mettre en œuvre cette méthode d'optimisation.

La métrique d'évaluation

Comme mentionné dans la section sur l'algorithme d'optimisation, l'arbre de Parzen vise à optimiser une fonction "objective" afin de déterminer le meilleur ensemble d'hyperparamètres. Dans notre cas, cette fonction objective est le coût du métie

Résultats des modèles

Modèle		Durée, s	RocAUC	Precision	Recall	F1score	Threshold	Buisnes score
Dummy Regressor		6	0.5	0.09	0.49	0.15		-13000
AdaBoost	Undersampling	184	0.7	0.27	0.33	0.29	0.496	317000
	SMOTE	2985	0.7	0.28	0.31	0.29	0.499	271000
	SMOTE 0.6	1277	0.72	0.28	0.35	0.31	0.426	313000
	SMOTE 0.4	669	0.72	0.28	0.35	0.31	0.397	318000
	SMOTE 0.2	370	0.72	0.29	0.33	0.31	0.459	315000
LightGBM	Undersampling	201	0.70	0.27	0.39	0.31	0.492	299000
	SMOTE	302	0.72	0.28	0.35	0.31	0.652	319000
	SMOTE 0.6	251	0.70	0.1	0.91	0.19	0.492	299000
	SMOTE 0.4	162	0.71	0.24	0.432	0.31	0.566	281000
	SMOTE 0.2	193	0.69	0.23	0.39	0.29	0.070	245000

RocAUC (L'aire sous la courbe ROC)
L'aire sous la courbe ROC (Receiver Operating Characteristic) est une mesure de la capacité d'un modèle de classification à distinguer entre les classes. La courbe ROC est un graphique représentant le taux de vrais positifs (sensibilité) en fonction du taux de faux positifs (1-spécificité) à différents seuils de décision. La formule n'est pas directe, car l'AUC est une mesure de l'intégrale de la courbe ROC

Précision (Precision)
La précision est le rapport des vrais positifs sur le total des cas classés positifs (vrais positifs plus faux positifs). C'est une mesure de la qualité des prédictions positives du modèle.
Precision=TP/(TP+FP)

Rappel (Recall)
Le rappel, également connu sous le nom de sensibilité ou taux de vrais positifs, est le rapport des vrais positifs sur le total des cas réellement positifs (vrais positifs plus faux négatifs).
Rappel=TP/ (TP+FN)

Score F1 (F1 Score)
Le score F1 est la moyenne harmonique de la précision et du rappel. Il est utile lorsque vous voulez trouver un équilibre entre la précision et le rappel.
F1 Score=2×Precision×Rappel/(Precision+Rappel)

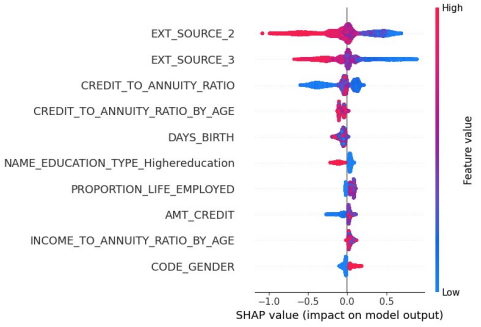
Pourquoi l'Exactitude (Accuracy) n'est pas adaptée pour les classes très déséquilibrées?
L'accuracy est le nombre total de prédictions correctes divisé par le nombre total de prédictions. Cependant, dans le cas de classes très déséquilibrées, l'accuracy peut être trompeuse. Par exemple, si 95% des échantillons appartiennent à une classe, un modèle peut simplement prédire cette classe pour tous les cas et atteindre une accuracy de 95%, même s'il n'a aucune capacité à identifier correctement les 5% restants. Cela signifie que l'accuracy ne tient pas compte de la performance du modèle sur la classe minoritaire, qui est souvent la plus importante à prédire correctement dans de telles situations.

L’interprétabilité globale et locale du modèle

En matière d'intelligence artificielle, l'interprétabilité d'un modèle de machine learning (ML) est essentielle pour plusieurs raisons cruciales :

- **Sur le plan professionnel**, comprendre le fonctionnement interne d'un modèle renforce la confiance en sa fiabilité et sa pertinence.
- **Éthiquement**, certains modèles, comme ceux utilisés dans notre projet, nécessitent une justification claire, notamment lorsque des décisions importantes, telles que le refus de crédit, doivent être expliquées aux clients concernés.
- **Légalement**, conformément à l'article 22 du RGPD, une personne a le droit de ne pas être soumise à une décision entièrement automatisée. Des mesures adéquates doivent être prises pour garantir l'intervention humaine, permettant ainsi de contester ou de discuter de telles décisions.

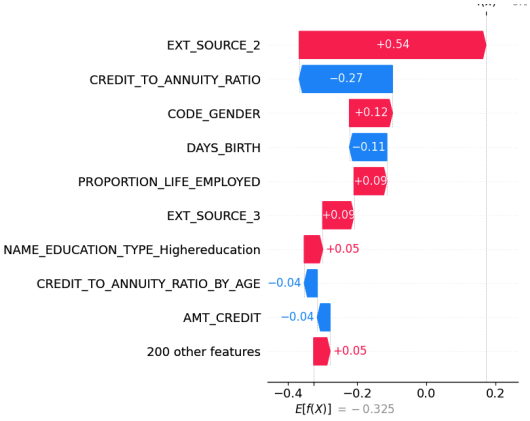
L’interprétabilité globale



SHAP, résumé en deux points, clarifie la contribution de chaque variable aux prédictions d'un modèle, que ce soit à une échelle globale pour un large éventail d'observations ou localement pour des groupes d'observations similaires. À travers une représentation visuelle, il est possible de discerner l'impact des variables les plus significatives sur les prédictions du modèle.

Prenons l'exemple de la variable **CODE_GENDER_M** : les valeurs attribuées à cette variable tendent à influencer la décision de crédit, favorisant les femmes lorsqu'elle est évaluée à 0. De même, on observe que des valeurs plus élevées pour **EXT_SOURCE_2 et EXT_SOURCE_3** augmentent la probabilité d'une classification vers la classe 0, c'est-à-dire un crédit accordé.

L’interprétabilité locale



Pour un cas spécifique, ce sont les caractéristiques les plus significatives qui ont été affichées, celles qui ont le plus influencé la prédiction. On remarque que la caractéristique ayant le plus d'impact est « EXT_SOURCE_2 », qui a joué un rôle significatif en inclinant la prédiction vers la classe 1.

Analyse de la Variation des Données (Datadrift)

Contexte

Avec le temps, les données servant à l'entraînement des modèles d'apprentissage automatique (ML) peuvent évoluer, rendant ces modèles obsolètes. Par exemple, les modèles prédictifs basés sur les habitudes d'achat avant la récession de 2008 deviendraient moins fiables après, car les comportements des consommateurs ont tendance à évoluer vers plus d'économies et moins de dépenses, modifiant ainsi les tendances précédemment observées. C'est ce qu'on appelle le datadrift, ou la dérive des données. Les modèles basés sur ces données dépassées doivent être réentrainés avec des informations récentes et pertinentes.

Méthodologie

L'analyse du datadrift s'effectue en comparant les distributions de deux ensembles de données similaires, en prenant comme référence celui sur lequel le modèle initial a été formé. Pour étudier l'impact de la dérive sur les variables ciblées, nous pouvons les inclure dans les colonnes analysées, en veillant à comparer des éléments comparables et à prévoir la variable cible dans les deux jeux de données.

Dans le cas où les jeux de données sont trop volumineux, nous pouvons limiter notre attention aux caractéristiques les plus significatives en utilisant l'importance des caractéristiques (feature_importance_) ou les valeurs SHAP. Si le volume des individus est trop conséquent, un échantillonnage peut être effectué pour réduire la taille des données à un volume gérable.

Solution

Nous employons la bibliothèque Evidently, et plus spécifiquement la configuration DataDriftPreset, pour analyser la dérive. Selon la nature des ensembles de données, différents indicateurs sont utilisés pour mesurer le datadrift.

StatTest	Applicable to	Default method for	Drift score
ks	données uniquement numériques	méthode par défaut pour les données numériques, si ≤ 1000 objets	retourne p_value la dérive détectée lorsque p_value < seuil seuil par défaut: 0.05
chisquare	données uniquement catégorielles	méthode par défaut pour catégoriel avec > 2 étiquettes, si ≤ 1000 objets	retourne p_value la dérive détectée lorsque p_value < seuil seuil par défaut: 0.05
z	données uniquement catégorielles	méthode par défaut pour les données binaires, si ≤ 1000 objets	retourne p_value la dérive détectée lorsque p_value < seuil seuil par défaut: 0.05
wasserstein	données uniquement numériques	méthode par défaut pour les données numériques, si > 1000 objets	retourne la distance la dérive est détectée lorsque la distance ≥ seuil seuil par défaut: 0.1
jensenshannon	données numériques et catégorielles	méthode par défaut pour catégoriel, si > 1000 objets	retourne la distance la dérive est détectée lorsque la distance ≥ seuil seuil par défaut: 0.1

Résultats

Dans notre cas, l'approche SHAP a été appliquée pour ne retenir que les 80 attributs les plus significatifs, représentant 65 % de l'explication globale. Les prédictions de la variable cible (feature TARGET) ont également été incluses. Le rapport indique qu'en général, en conservant les seuils par défaut, il n'y a pas de dérive significative des données ni de la variable cible prédite.

Les limites et les améliorations possibles

1) L'un des principaux obstacles rencontrés a été le manque de connaissance spécifique au domaine bancaire. La compréhension nuancée de chaque variable s'est révélée complexe, soulignant l'intérêt d'une collaboration étroite avec un expert en crédit bancaire. Une telle expertise aurait pu éclairer la pertinence des features et optimiser les méthodes de traitement des données.
2) les processus d'apprentissage ont été limités par les capacités de calcul

Améliorations possibles

Pour pallier ces difficultés, l'intégration d'une expertise métier aurait été bénéfique pour affiner la sélection des features pertinentes et pour élaborer un score métier plus adéquat. De plus, un renforcement des capacités de calcul aurait permis de mener des expérimentations parallèles, d'explorer diverses techniques de rééquilibrage des classes et d'élargir l'éventail des modèles testés.

En conclusion, bien que le modèle actuel offre des résultats satisfaisants, il existe une marge significative pour l'amélioration. En associant expertise métier et ressources computationnelles accrues, nous pourrions significativement rehausser la précision et la robustesse de nos modèles prédictifs.