Projet P7 – Implementer un modele de Scoring

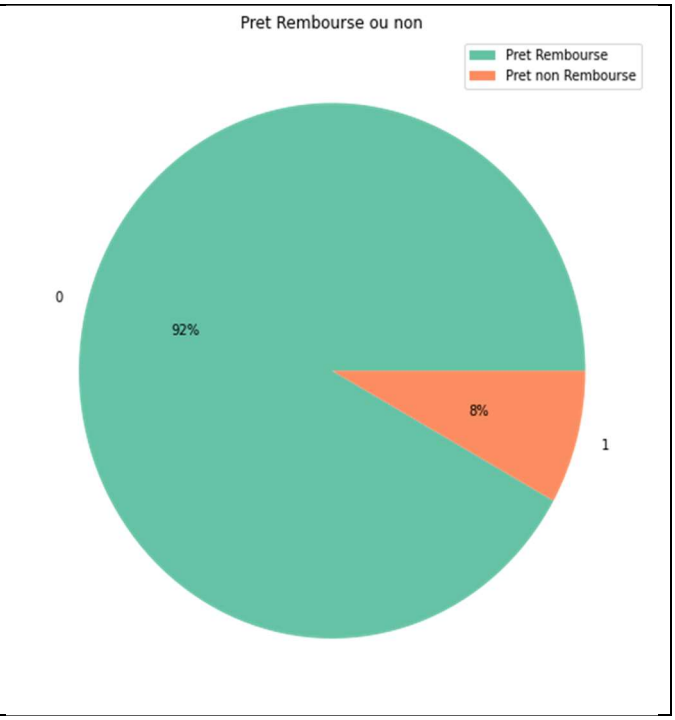
Note Méthodologique

Cette note présente les points suivants :

- La méthodologie d'entraînement du modèle (2 pages maximum)
- La fonction coût métier, l'algorithme d'optimisation et la métrique d'évaluation (1 page maximum)
- L'interprétabilité globale et locale du modèle (1 page maximum)
- Les limites et les améliorations possibles (1 page maximum)

La méthodologie d'entraînement du modèle

La valeur à prédire est contenue dans la variable « TARGET ». Elle peut prendre 2 valeurs : - 1 qui indique que le client n'a pas remboursé son crédit. (Défaillant, 8%) - 0 qui indique que le client l'a remboursé. (Non défaillant, 92%)



Nous avons un problème de classification binaire avec un fort déséquilibre des classes.

Ce déséquilibre devra être pris en compte lors de la construction du modèle, pour éviter des prédictions erronées.

La démarche suivie est la suivante :

1. Utiliser PYCARET avec SMOTE pour sélectionner les 3 modèles les plus pertinents
2. Réaliser une modélisation avec un RandomSearchCV - pipeline avec SMOTE et un FBeta Score adapte aux modèles imbalanced
3. Choix du beta = 3. Cette valeur donnera plus de poids au Recall et moins à la Précision.

4. Le Résultat est un best modèle qu'on améliore avec un threshold moving. On trouve **optimal_threshold = 0.358**

Modelisation avec PYCARET(SMOTE)

Les modèles ensemblistes rf, lightgbm, gbc et ada ont un Accuracy trop élevé, ce qui rappelle l'**accuracy paradox** de plus leurs Recall est pratiquement nul ce qui signifie que les faux négatifs sont très élevés ce qui est contraire au résultat qu'on cherche. - Par contre LogisticRegression, Ridge et Naive Bayes ont des recall éleve ce qui nous incite a les choisir pour la suite. - Remarque : Ridge n'offre pas la possibilité de faire un threshold moving car il n'a pas de predict_proba. Les classes sont predites en 0 ou 1 directement.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.9195	0.7298	0.0103	0.5158	0.0202	0.0170	0.0637	21.5340
rf	Random Forest Classifier	0.9194	0.6805	0.0006	0.3869	0.0012	0.0009	0.0119	52.6200
gbc	Gradient Boosting Classifier	0.9191	0.6831	0.0089	0.3946	0.0173	0.0138	0.0489	112.0040
ada	Ada Boost Classifier	0.9096	0.6549	0.0420	0.2095	0.0666	0.0417	0.0568	38.6870
dt	Decision Tree Classifier	0.8432	0.5260	0.1480	0.1190	0.1319	0.0469	0.0472	100.2020
lr	Logistic Regression	0.6667	0.7051	0.6286	0.1430	0.2330	0.1172	0.1700	47.0290
ridge	Ridge Classifier	0.6634	0.0000	0.6335	0.1425	0.2326	0.1164	0.1700	31.2730
nb	Naive Bayes	0.3594	0.5734	0.7503	0.0888	0.1587	0.0172	0.0441	18.7800

Modélisation - RandomSearchCV - pipeline avec SMOTE. Nous choisirons une valeur beta egale a 3. Cette valeur donnera plus de poids au recall et moins a la Precision.

	Model	Accuracy_Score	AUC_Score	Precision_Score	Recall_Score	F1_Score	Fbeta_Score
0	RidgeClassifier	0.696047	0.676639	0.160496	0.653494	0.257693	0.499906
1	LogisticRegression	0.698359	0.674737	0.160473	0.646566	0.257117	0.496217
2	GaussianNB	0.300266	0.545956	0.089844	0.838953	0.162275	0.457231

Choix du meilleur modele

Les faux negatifs sont de : 8 721 pour lr, 8 510 pour ridge, 4 104 pour Naive Bayes. Malgre ca l'etude suivante va montrer que le modele lr est mieux adapte pour notre cas, car ce qui compte aussi c'est de ne pas avoir trop de faux positifs avec une precision trop faible pour le modele Naive Bayes.
Best Model : LR

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

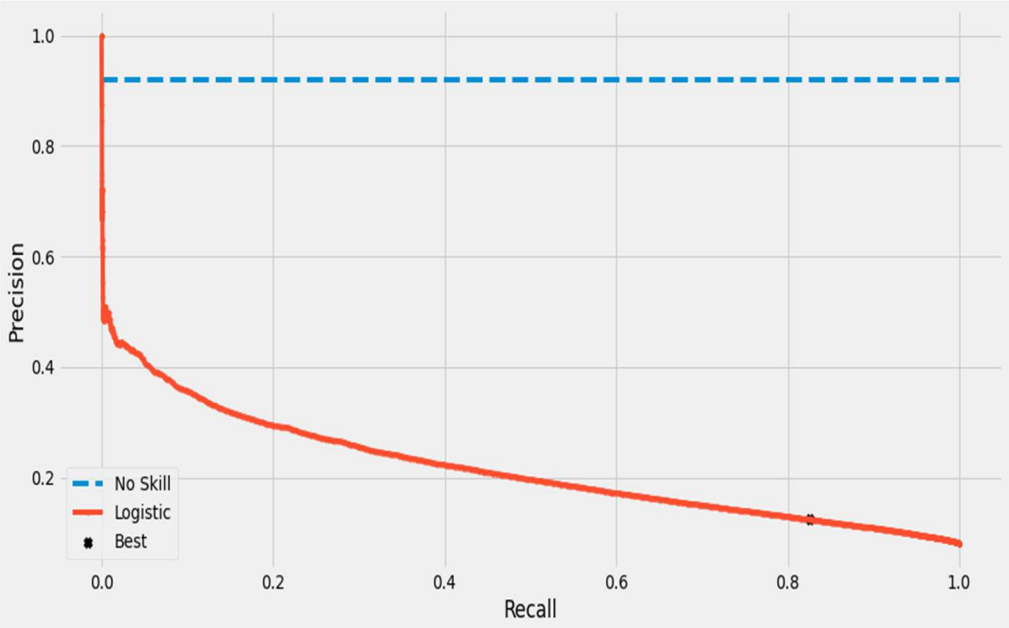
- La métrique d'evaluation est FBeta Score
- L'optimisation de la solvabilite est réalisé avec une technique de Threshold Moving

Optimal Threshold for Precision-Recall Curve

ix : 358 Optimal Threshold : 0.358
Fbeta-Score : 0.5273

recall : 0.8256 precision : 0.1240

Les faux negatifs sont passes de 8 721 a 4 327 avec threshold optimal = 0.358

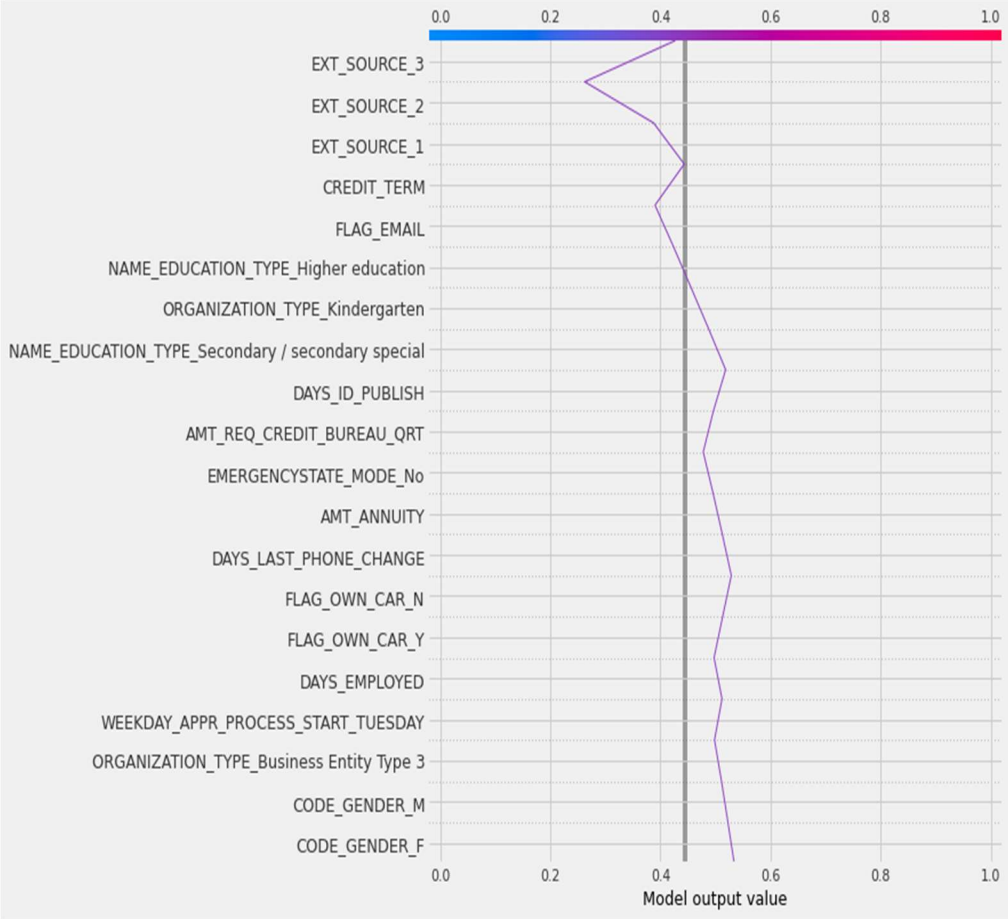


L’interprétabilité globale et locale du modèle

Interpretation : SHAP (SHapley Additive exPlanations)

Locale

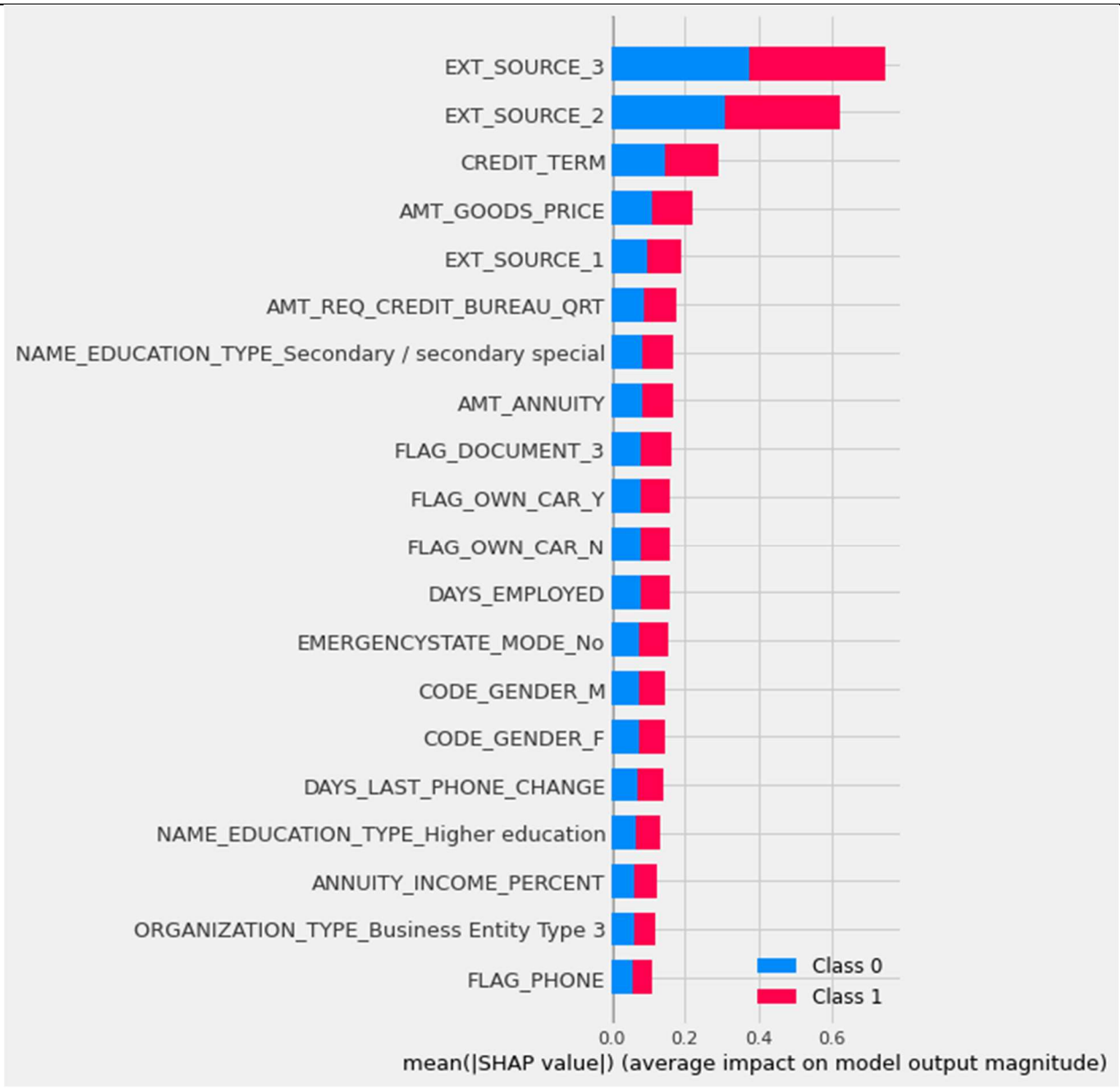
Decision Plot pour interpretation de la probabilite de defaut d’un client




```
# plot the SHAP values for the
shap.initjs()
shap.force_plot(expected_value[1], shap_values[1][0,:], X_test_1000[0].reshape(1, -1), link="logit",\
                feature_names=feature_names)
```

Globale

Summary Plot



Les limites et les améliorations possibles

Les limites :

- Temps de calcul important des modèles ensemblistes
- Pour des données plus importantes la technique de Oversampling SMOTE peut consommer beaucoup de ressources
- L'interprétabilité avec SHAP explainer est couteuse en temps de calcul. Pour la réduire nous avons utilise KernelExplainer car le LinearExplainer ne donne pas de résultats satisfaisants

Les améliorations possibles :

- Le modèle **Naive Bayes** peut être une bonne alternative a cause du Recall important qu'elle fournit (>80%) a condition de trouver un moyen algorithmique afin d'améliorer la faible Precision qu'elle donne.
- Comprendre et améliorer le temps de calcul des méthodes ensemblistes qui sont réputés meilleurs en terme de prédiction