


	<p><u>Projet 7 :</u></p> <p>Implémentez un modèle de scoring</p>	
---	--	---

Projet 7 : Implémentez un modèle de scoring

Note méthodologique

Nom : TRABIS
Prénom : Mohamed
Intitulé de formation : Data Scientist
Mentor: Mr. Christian NOUMSI

	<p style="text-align: center;"><u>Projet 7 :</u></p> <p style="text-align: center;">Implémentez un modèle de scoring</p>	
---	--	---

Sommaire

1.	LA METHODOLOGIE D'ENTRAINEMENT DU MODELE	3
1.1.	INTRODUCTION.....	3
1.2.	DEFINIR LES DONNEES D'ENTRAINEMENT ET DE TEST.....	3
1.3.	LE DESEQUILIBRE DES CLASSES.....	3
1.4.	MODELES DE PREDICTION	4
2.	COUT METIER ET METRIQUE D'EVALUATION.....	5
2.1.	LA FONCTION COUT METIER.....	5
2.2.	METRIQUE D'EVALUATION	5
3.	INTERPRETABILITE DU MODELE.....	7
4.	LES LIMITES ET LES AMELIORATIONS.....	8
4.1.	MODELE	8
4.2.	DASHBOARD	8

1. La méthodologie d'entraînement du modèle

1.1.Introduction

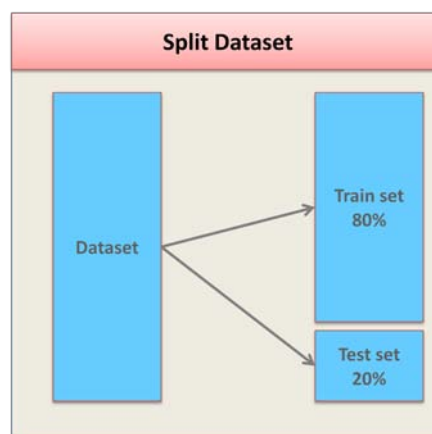
L'entreprise "Prêt à dépenser" souhaite mettre en œuvre un outil de “scoring crédit” pour calculer la probabilité qu’un client rembourse son crédit, puis classifie la demande en crédit accordé ou refusé. Elle souhaite donc développer un algorithme de classification en s’appuyant sur des sources de données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

Le point le plus importants de la mission est d’assurer une certaine transparence quant à la décision afin de permettre au responsable client de mieux comprendre pourquoi la demande d’un de ses clients est acceptée ou refusée. Ce modèle doit être accessible via un dashboard interactif.

1.2.Définir les données d'entrainement et de test

Les valeurs explicatives contiennent 307505 lignes et 286 colonnes, et la cible contient 307505 lignes.

Les données sont divisées en données d'entrainement (80%) et de test (20%) avec *train_test_split()* de Scikit-Learn:



1.3.Le déséquilibre des classes

Une première exploration des données révèle un problème de déséquilibre des classe qu'il faut régler.

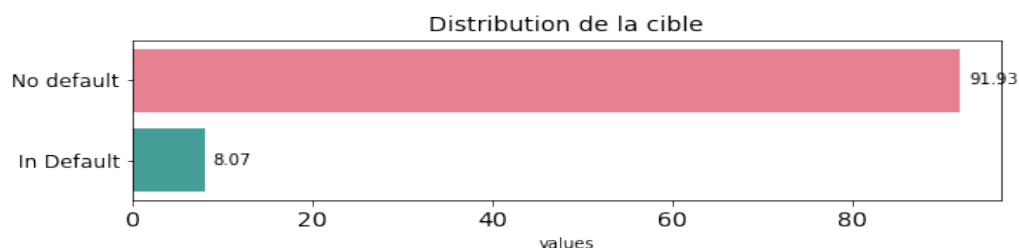




Figure1 : Distribution de la cible

	Projet 7 : Implémentez un modèle de scoring	
--	--	--

Le déséquilibre des classes est un problème courant en apprentissage automatique, en particulier dans les problèmes de classification. Les données de déséquilibre peuvent nuire considérablement à la précision de notre modèle.

Il existe trois méthodes pour régler ce déséquilibre :

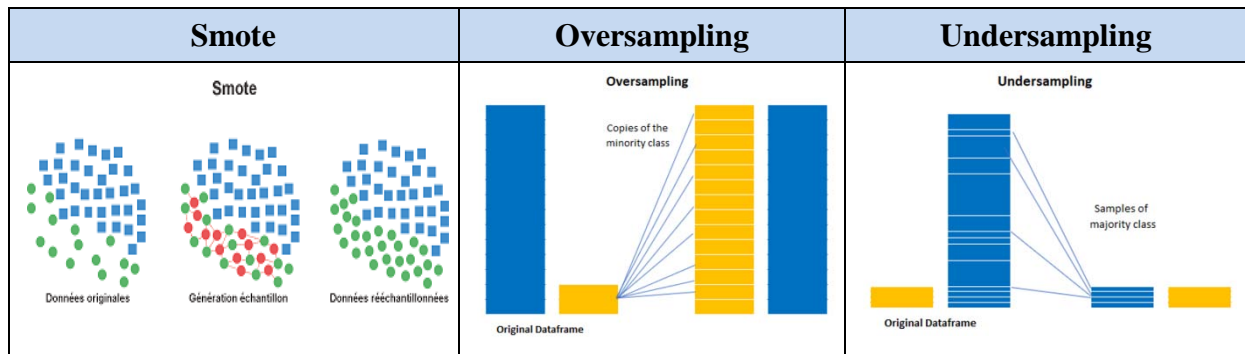


Figure2 : Smote - Oversampling - Undersampling

1.4.Modèles de prédiction

Pour cette étape on a utilisé deux modèles de prédiction : **LGBMClassifier** et **RandomForestClassifier**.

Pour chacun de ces deux algorithmes, nous avons effectué un "**RandomizedSearchCV**" sur le jeu de données avec une validation croisée de 5 KFold afin d'optimiser le score "**roc_auc_score**."

Cette validation croisée est effectuée après chaque méthode de rééchantillonnage.

Exemple : rééchantillonnage Smote

```

pipeline = Pipeline([
    ('smote', SMOTE()),
    ('lgbm', LGBMClassifier())
])

params_lgbm = {'lgbm__' + key: param_distributions[key] for key in param_distributions}

random_search_smote = RandomizedSearchCV(estimator=pipeline,
                                          param_distributions=params_lgbm,
                                          scoring='roc_auc',
                                          cv=kf)

random_search_smote.fit(X_train, y_train)

```

Figure 3 : validation croisée RandomizedSearchCV (Smote)

2. Coût métier et métrique d'évaluation

2.1. La fonction coût métier

La fonction coût métier consiste à calculer le gain obtenu pour l'ensemble des individus du jeu de données.

Pour ce faire on va fixer un poids pour chacune des prédictions par rapport à leurs valeurs réelles. Les valeurs des poids sont les suivantes :

- **FN = -10**
- **TP = 0**
- **TN = 1**
- **FP = 0**

		Reality	
		Negative : 0	Positive : 1
Prediction	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

De ce fait, les prêts accordés aux individus qui ne sont finalement pas solvables sont dotés d'une pénalisation négative de -10, alors que les prêts accordés aux individus finalement solvables rapportent 1. Ce rapport 10 est totalement arbitraire et il est tout à fait possible de changer ces valeurs à la convenance de l'optique métier.

2.2. Métrique d'évaluation

Pour évaluer la performance d'un modèle de prédiction, le choix de la bonne métrique est donc crucial.



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.

On pourra aussi utiliser des métriques qui seront beaucoup moins influencées par la classe majoritaire. la métrique ROC AUC , le Recall (sensibilité) ou le F-Score sont de bons exemples.

Avant de calculer les scores, le réglages d'hyperparamètres est une étapes importante pouvant avoir un impact dans l'amélioration de la métrique d'évaluation.

```
param_distributions = {
    'learning_rate': [0.01, 0.1, 0.09],
    'reg_alpha': [0, 7, 10, 20],
    'n_estimators': [100, 400, 800, 1000],
    'max_depth': [5, 10, 20],
    'colsample_bytree': [0.01, 0.1, 0.2, 0.3],
    'min_child_samples': [10, 20, 50],
    'min_child_weight': [5, 10, 20],
    'num_leaves': [10, 15, 20],
    'reg_lambda': [10, 50, 70, 100],
    'scale_pos_weight': [6, 7, 8, 10, 11, 12, 15, 17],
}
```

Figure 4 : Réglage d'hyperparamètres

	<p align="center"><u>Projet 7 :</u></p> <p align="center">Implémentez un modèle de scoring</p>	
--	--	--

Ci-dessous les scores après chaque rééchantillonnage et validation croisée :

Modèle	Méthode de rééchantillonnage	Score AUC	Score F1	Recall
LGBMClassifier	Smote	0.780	0.286	0.705
	Oversampling	0.782	0.283	0.722
	Undersampling	0.780	0.289	0.690
RandomForestClassifier	Smote	0.731	0.241	0.726
	Oversampling	0.759	0.268	0.708
	Undersampling	0.762	0.268	0.715

3. Interprétabilité du modèle

L'interprétabilité vient **après l'étape de modélisation** (celle où l'on construit le modèle prédictif). Cette phase ne fait plus appel à des algorithmes prédictifs entraînés mais à des **algorithmes d'intelligibilité**, pour expliquer les prédictions du modèle prédictif précédemment construit.

L'objectif est de mieux comprendre les modèles ou leurs prévisions. Cela consiste à pouvoir expliquer simplement les prévisions d'un modèle.

Les valeurs de Shapley (SHapley Additive exPlanations) peuvent être utilisées pour expliquer la sortie d'un modèle d'apprentissage automatique :

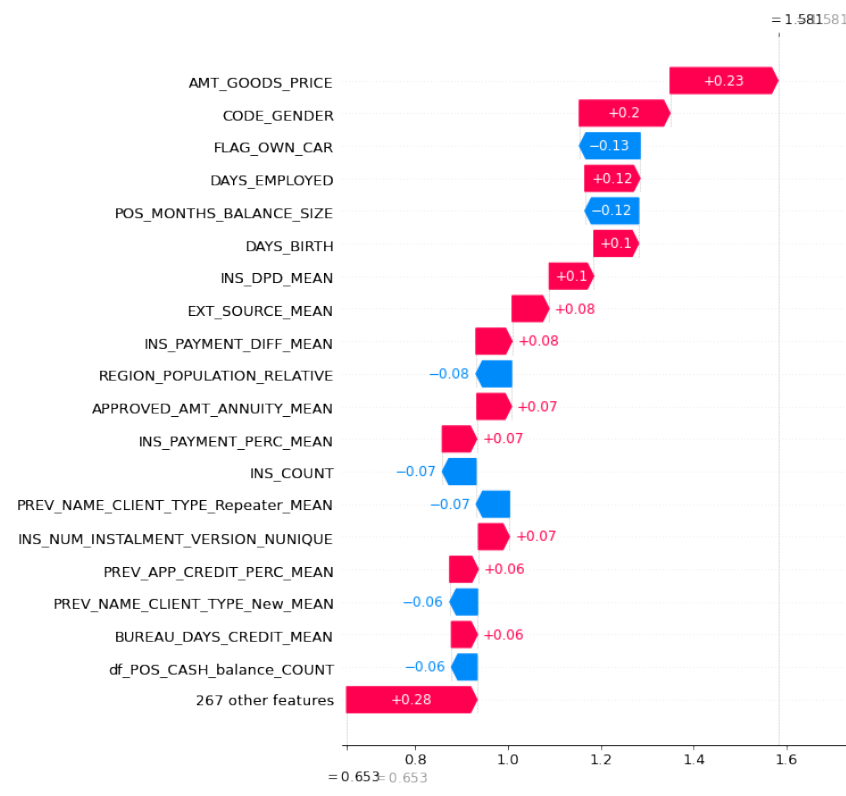




Figure 5 : Waterfall de Shapley

Pourquoi les valeurs Shapley sont-elles importantes ?

L'explicabilité du modèle nous permet d'examiner la prise de décision d'un modèle à la fois au niveau global et local. Au niveau global, nous pouvons comprendre quelles caractéristiques contribuent au résultat du modèle et dans quelle mesure elles influencent la décision du modèle. Au niveau local (chaque point de données individuel), nous pouvons déterminer pourquoi le modèle a pris une certaine décision et fournir des raisons si nécessaire.

Cette approche va permettre aux chargés de relation client d'expliquer de façon la plus transparente possible les décisions de la banque.

	<p style="text-align: center;"><u>Projet 7 :</u></p> <p style="text-align: center;">Implémentez un modèle de scoring</p>	
--	--	--

4. Les limites et les améliorations

Il existe d'autres approches pour améliorer le modèle et le Dashboard interactif.

4.1.Modèle

Ci-dessous les points d'amélioration de notre modèle :

- Améliorer le data engineering avec une analyse comptable plus poussée, pour générer d'autres variables.
- Affiner l'analyse avec l'équipe métier pour déterminer le seuil optimal.
- Optimiser la performance du modèle :
 - ✓ Affiner le réglage d'hyperparamètres.
 - ✓ Avoir un matériel puissant pour répondre aux exigences du jeu de données particulièrement lourd (307505 lignes et 286 colonnes).
- Essayer d'autres techniques d'encodage de données.

4.2.Dashboard

- Optimiser les performances pour un chargement de données plus rapide.
- Ajouter d'autres onglets pour une analyse complète.
- Intégrer des graphiques interactifs.
- La récupération des données à l'aide d'une interface API qui permet notamment l'automatisation, la communication standardisée et sécurisée.