





# Note méthodologique

Implémentez un modèle de scoring



Rim BAHROUN



Projet 7 parcours Data Scientist

Openclassrooms

Juin 2023

## Présentation de la démarche de modélisation

## Table des matières

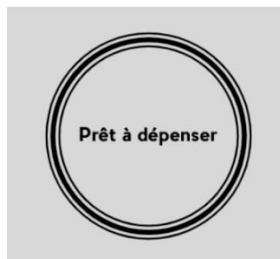
Introduction.....	3
1 Les données.....	3
2 Le traitement du déséquilibre des classes .....	4
3 Métriques d'évaluation .....	4
4 Méthodologie d'entraînement des modèles .....	6
5 Optimisation des modèles.....	6
6 Synthèse des résultats.....	7
7 Interprétabilité globale et locale du modèle.....	8
7.1 Interprétabilité globale du modèle .....	8
7.2 Interprétabilité locale du modèle.....	8
8 Analyse du data drift .....	10
9 Limites et améliorations possibles .....	10



## Introduction

---

**Prêt à dépenser** est une société financière qui propose des crédits à la consommation pour des personnes ayant peu ou pas du tout d'historique de prêt.



L'entreprise 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.). Cet algorithme implémentera un **Dashboard interactif** pour expliquer de façon la plus transparente possible les décisions d'octroi de crédit.

Cette note méthodologique décrit toute la démarche d'élaboration du modèle de scoring jusqu'à l'analyse du data drift.

## 1 Les données

---

Les données utilisées pour ce projet proviennent de <https://www.kaggle.com/competitions/home-credit-default-risk>.

Un premier prétraitement a été réalisé en utilisant le kernel Kaggle disponible sur <https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script>. Suite à ce prétraitement, les données ont été agrégées afin d'obtenir un seul jeu de données, les valeurs aberrantes ont été remplacées et de nouvelles variables ont été créées. On obtient alors un jeu de données d'entraînement de **797 variables et 356 251 clients**.

Un deuxième prétraitement a été effectué pour sélectionner les variables potentiellement intéressantes. 6 étapes ont été utilisées pour réduire la dimensionnalité en supprimant les variables :

- À plus de 75% de valeurs manquantes ; -> 755 variables gardées
- Colinéaires corrélées à plus de 90% ; -> 589 variables gardées/ AUC=0.79
- À faible importance, < de 5%, à partir d'un arbre de prédiction ; -> 250 variables
- À faible variance moins de 2% ; -> 99 variables gardées/AUC=0.77
- Par la méthode SelectKbest ; -> 66 variables gardées/ AUC=0.76
- Par la méthode Recursive Feature Elimination. -> 35 variables gardées/ AUC=0.75

A la fin de chaque étape, la métrique AUC a été calculée pour un modèle **lightgbm** avec une crose validation de 10 pour s'assurer de maintenir la valeur de cette métrique proche de la valeur avant réduction de dimension.



Le jeu de données ainsi obtenu est de **35 variables et 356 251 clients**.

Ce jeu de données a été séparé en deux :

- Un jeu d'entraînement (67% des clients soit **307 507 clients**)
- Un jeu de validation (33% des clients) pour l'évaluation du modèle.

## 2 Le traitement du déséquilibre des classes

Le problème est un problème de classification binaire : client à risque (classe 1) vs client fiable (classe 0) avec une classe sous représentée (8% clients à risque contre 92% de clients fiable).

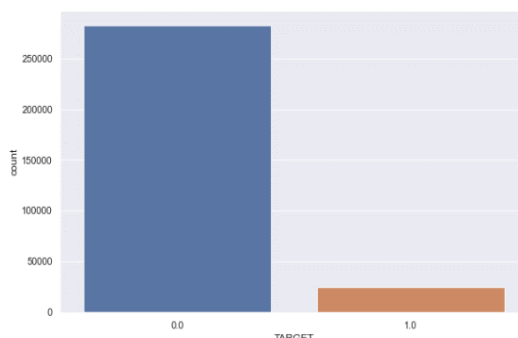


Figure 1 Répartition des individus selon la classe

Le déséquilibre des classes doit être pris en compte pour l'élaboration d'un modèle pertinent. En effet, un modèle naïf prédisant systématiquement le client en fiable aurait une accuracy de 0.92 et pourrait être considéré à tort comme performant alors qu'il ne permettrait pas la détection des clients à risque coûteux pour l'entreprise. **3** approches pour rééquilibrer les deux classes ont été testées :

- **Sous échantillonnage** : retirer aléatoirement des clients fiables (classe majoritaire). Le jeu de données d'entraînement est alors formé que de **33 310 clients**. Cette méthode fait perdre de l'information.
- **Sur échantillonnage SMOTE** : créer des données synthétiques de clients à risque (classe minoritaire) moyennant des données existantes. Le jeu de données d'entraînement est alors formé de **378 748 clients**. Cette méthode fait augmenter la taille du jeu de données et par suite le temps de calcul.
- **Class weights** : créer un modèle pénalisé en attribuant des poids différents pour chaque classe : pénalisation plus forte pour l'erreur de classification de la classe minoritaire. Le jeu de données d'entraînement ne change pas de dimension et est alors formé de **307 507 clients**. Cette méthode sera retenue pour la suite du projet.

## 3 Métriques d'évaluation

Pour évaluer les modèles et optimiser les hyperparamètres, un choix des métriques s'impose. La matrice de confusion est la suivante :

		Classe prédite	
		Client fiable	Client à risque
Classe Réelle	Client fiable	Vrais négatifs / TN	Faux positifs / FP
	Client à risque	Faux négatif / FN	Vrais positifs / TP



Les métriques d'évaluation prises en compte dans ce projet sont les suivantes :

- **AUC-ROC\_score** : Il mesure la capacité d'un modèle à classer correctement les exemples positifs par rapport aux exemples négatifs, quelle que soit la valeur du seuil de classification. L'AUC-ROC varie entre 0 et 1, où une valeur de 1 représente une performance parfaite et une valeur de 0,5 représente une performance aléatoire.
- **Accuracy\_score** : C'est un score d'exactitude calculé en divisant le nombre d'exemples correctement classés par le nombre total d'exemples. Une valeur de 1 indique que tous les exemples ont été classés correctement et une valeur de 0 indique une classification complètement incorrecte.
- **Recall\_score** : ou sensibilité. Il mesure la capacité d'un modèle à identifier correctement les exemples positifs (client à risque) parmi tous les exemples réellement positifs présents dans l'ensemble de données. Une valeur de 1 indique un rappel parfait, c'est-à-dire que tous les clients à risque ont été correctement identifiés, et une valeur de 0 indique que aucun client à risque n'a été correctement identifié.
- **Precision\_score** : Il mesure la capacité d'un modèle à classer correctement les exemples positifs parmi tous les exemples prédits comme positifs. Une valeur de 1 indique une précision parfaite, c'est-à-dire que tous les clients prédits à risque sont corrects, et une valeur de 0 indique que aucun client prédit à risque n'est correct.
- **Costum\_score** : Il s'agit d'un score créé pour répondre au besoin du métier. Le score est une somme pondérée et normalisée de faux négatifs et faux positifs. L'hypothèse prise en compte dans le calcul est la suivante : **le coût d'un faux négatif FN (clients à risque prédit fiable : donc crédit accordé et perte en capital) est dix fois supérieur au coût d'un faux positif FP (client fiable prédit à risque : donc refus crédit et manque à gagner en marge)**. Le costum\_score varie entre 0 et 1. Plus le costum\_score est grand, meilleur est le modèle.

$$\text{Costum\_score} = 1 - \frac{1*FP + 10*FN}{1*N + 10*P}$$

Avec :

FP : nombre de faux positifs,

FN : nombre de faux négatifs,

N : nombre de négatifs dans la population,

P : nombre de positifs dans la population,

1\*FP+10\*FN : coût de la prédiction,

1\*N+10\*P : coût maximal tous les vrais négatifs ont été prédits en positifs et inversement.

- **F\_beta\_score** : Il combine à la fois la précision et le rappel en une seule mesure en utilisant un paramètre beta pour contrôler le poids relatif de la précision par rapport au rappel.

$$f_{\beta} = \frac{(1 + \beta^2)TP}{(1 + \beta^2)TP + (\beta^2FN + FP)}$$

Afin de prendre en compte l'hypothèse qu'un FN coûte 10 fois plus qu'un FP, on prendra  $\beta^2=10$  et donc  $\beta = 3.16$ . Plus le  $f_{\beta}$  score est grand, meilleur est le modèle.



## 4 Méthodologie d'entraînement des modèles

---

Plusieurs modèles ont été testés :

- **DummyClassifier** : un classificateur naïf qui prédit la classe majoritaire tout le temps
- **LogisticRegression** : Le modèle de régression logistique utilise une fonction logistique (ou sigmoïde) pour transformer une combinaison linéaire des caractéristiques en une probabilité comprise entre 0 et 1. Cette probabilité est ensuite comparée à un seuil pour effectuer la classification finale. Si la probabilité est supérieure au seuil, l'exemple est classé comme positif, sinon il est classé comme négatif.
- **RandomForestClassifier** : un algorithme d'ensemble basé sur des arbres de décision. Il combine plusieurs arbres de décision pour former un modèle robuste et performant. Chaque arbre de décision est construit à partir d'un échantillon aléatoire de données et de variables d'entrée. L'algorithme utilise ensuite une combinaison des prédictions de tous les arbres pour effectuer la classification finale.
- **ExtraTreesClassifier** : un algorithme d'ensemble basé sur des arbres de décision. Il est similaire à RandomForestClassifier dans son fonctionnement, mais présente quelques différences clés. Lors de la construction de chaque arbre de décision, ExtraTreesClassifier sélectionne aléatoirement les seuils pour chaque variable d'entrée, contrairement à RandomForestClassifier qui utilise des seuils optimaux.
- **LGBMClassifier** : un algorithme d'apprentissage automatique basé sur le gradient boosting. Il utilise des arbres de décision pour créer un modèle prédictif. L'un des avantages clés de LGBMClassifier est sa rapidité d'exécution, car il utilise une technique d'optimisation pour sélectionner les exemples les plus informatifs lors de la construction des arbres. LGBMClassifier est particulièrement adapté aux problèmes avec de grandes quantités de données et un grand nombre de variables.

Pour chaque modèle, les entraînements ont été réalisés sur le jeu de données de 35 variables en utilisant les poids des classes. Le modèle final sélectionné a été choisi en comparant les scores obtenus sur le jeu de données de validation et les temps de traitement.

## 5 Optimisation des modèles

---

L'optimisation des hyperparamètres pour chaque modèle a été réalisée à l'aide de GridSearchCV. Les hyperparamètres sélectionnés seront ceux qui permettent d'obtenir le meilleur score métier (max costum\_score) en utilisant la validation croisée avec 10 plis.

Ensuite, la valeur seuil moyenne obtenue à partir de GridSearchCV pour le meilleur modèle a également été récupérée. L'application de ce seuil sur la méthode predict\_proba permettra de classer tout nouveau client en tant que client fiable ou client à risque, et ainsi de décider d'accorder ou non le crédit.



## 6 Synthèse des résultats

La comparaison des performances des modèles, comme illustré dans la figure ci-dessous, met en évidence les avantages de l’algorithme LightGBM. En effet, ce dernier a permis d’obtenir les meilleurs scores parmi tous les modèles évalués, tout en maintenant un temps de traitement raisonnable.

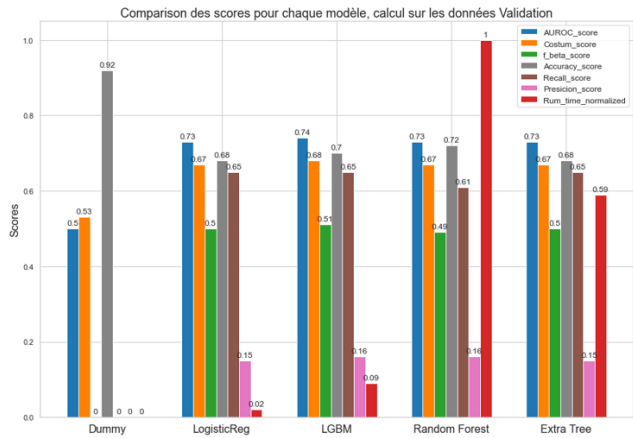


Figure 2 Comparaison des performances des modèles

Le modèle retenu pour notre projet est l’algorithme LightGBM, qui a été optimisé grâce à un processus de fine-tuning. Le tableau ci-dessous présente les scores obtenus par ce modèle sur les jeux de données d’entraînement et de test.

	Train set	Test set
AUROC	0.77	0.74
Score_métier	0.70	0.68
f_beta	0.54	0.51
Accuracy	0.69	0.69
Recall	0.70	0.66
Presicion	0.17	0.16

Tableau 1 Les scores du modèle retenu LightGBM

La figure suivante illustre le score métier obtenu sur le jeu de test par le modèle retenu. Pour maximiser le score métier, un seuil de classification de 0.5 a été utilisé. Ce seuil a été sélectionné afin d’optimiser la performance du modèle en termes de score métier.

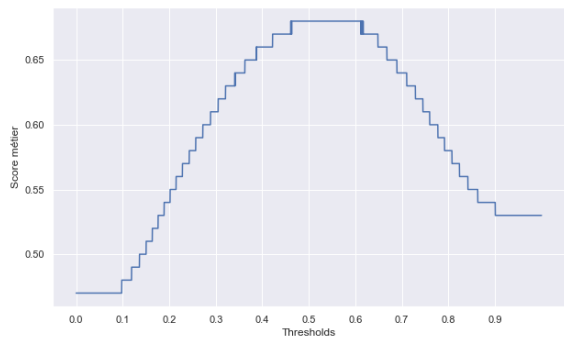


Figure 3 Score métier en fonction du seuil de classification



## 7 Interprétabilité globale et locale du modèle

### 7.1 Interprétabilité globale du modèle

L'importance des caractéristiques a été évaluée à l'aide du modèle LGBMClassifier. Cette mesure permet de déterminer l'influence relative de chaque caractéristique sur les prédictions du modèle. Les caractéristiques les plus importantes sont celles qui ont le plus grand impact sur les décisions de classification.

Dans notre cas, les résultats ont montré que les 3 scores externes et l'âge de client étaient parmi les caractéristiques les plus importantes pour le modèle sélectionné. Cela indique que ces variables jouent un rôle significatif dans les décisions d'octroi du crédit.

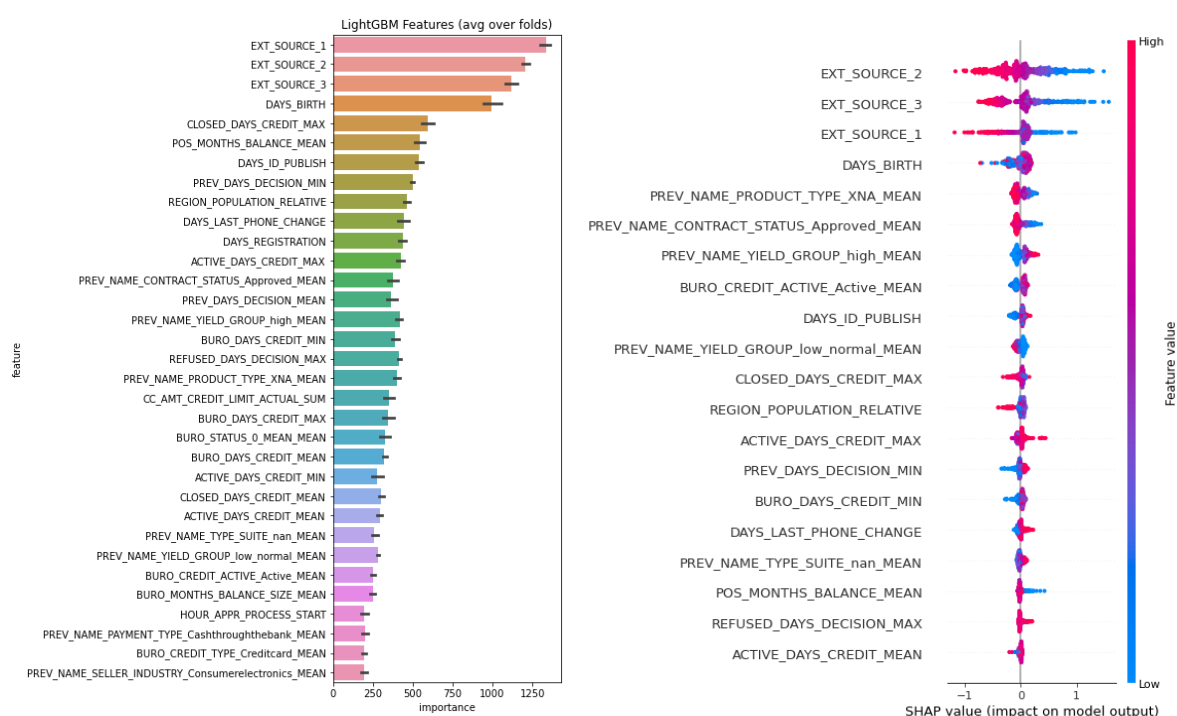


Figure 4 Importance globale des variables

### 7.2 Interprétabilité locale du modèle

Les valeurs SHAP sont utilisées pour évaluer l'importance de chaque variable dans la décision d'octroi de crédit pour un client donné. Les variables ayant une influence négative sur l'accord du crédit sont mises en évidence en rouge, tandis que celles ayant une influence positive sont en bleu. Cette distinction permet de mieux comprendre les facteurs qui défavorisent ou favorisent l'octroi du crédit. L'analyse des variables en rouge identifie les éléments ayant une influence défavorable, utile pour d'éventuels ajustements ou améliorations. De même, les variables en bleu indiquent les éléments favorisant l'accord du crédit. Cette visualisation combinée avec les valeurs SHAP offre une interprétation plus complète et intuitive de la contribution de chaque variable à la décision d'octroi de crédit pour le client donné.





Exemple de client à risque :

Client 100002.

Probabilité de remboursement 21%

Crédit refusé

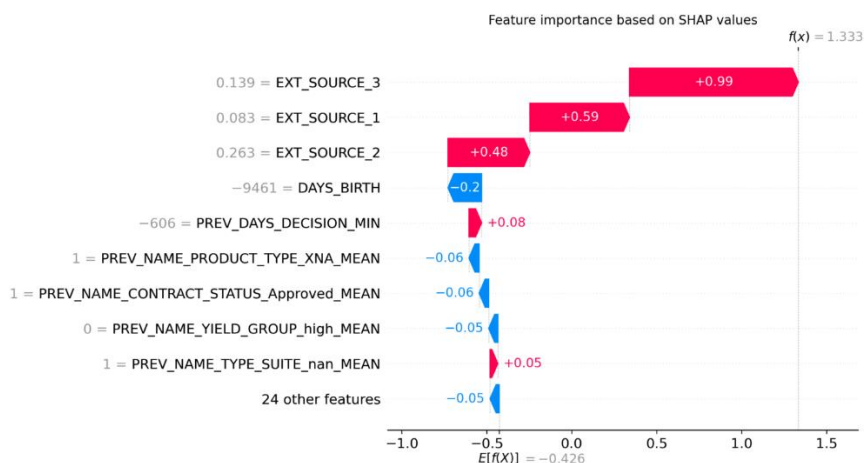


Figure 5 Importance locale des variables pour un client à risque

D'après la figure ci-dessus, il est clair que le crédit a été refusé principalement en raison des 3 scores externes du client. Ces scores externes ont une influence négative significative sur la décision d'octroi de crédit.

Exemple de client fiable :

Client 100004.

Probabilité de remboursement 79%

Crédit accordé

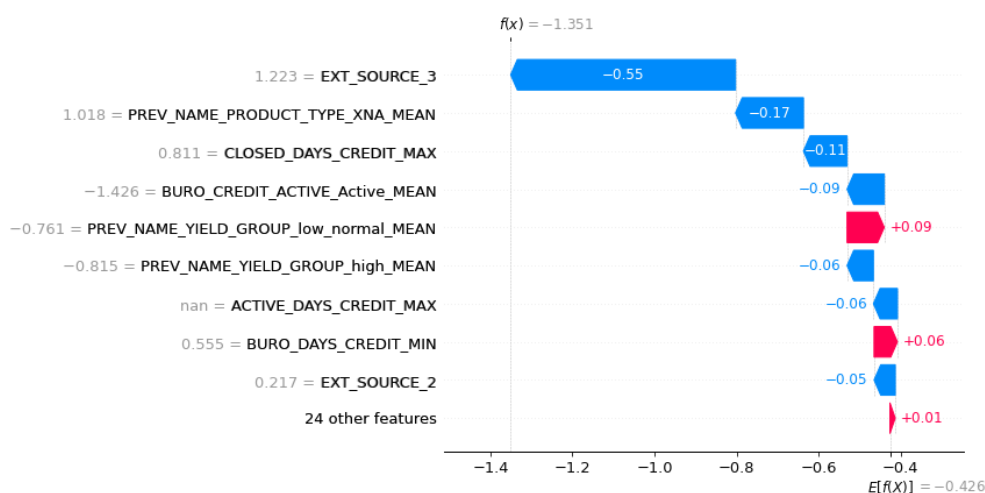


Figure 6 Importance locale des variables pour un client fiable

D'après la figure ci-dessus, le crédit a été accordé principalement en raison du score externe 3 du client. Ce score externe a une influence positive significative sur la décision d'octroi du crédit.



## 8 Analyse du data drift

La dérive des données fait référence à un phénomène dans lequel les caractéristiques ou les distributions des données changent au fil du temps. Cela peut se produire dans les ensembles de données utilisés pour l'apprentissage automatique lorsque les données d'entraînement et les données d'évaluation ne sont pas cohérentes.

La dérive des données peut se produire pour diverses raisons, telles que des changements dans l'environnement, des changements dans le comportement des utilisateurs ou des problèmes dans le processus de collecte des données. Elle peut entraîner une baisse des performances des modèles prédictifs, car ces derniers sont formés sur des données qui ne sont plus représentatives des données actuelles.

La détection et la gestion de la dérive des données sont des tâches importantes dans le domaine de l'apprentissage automatique. Cela peut impliquer la surveillance régulière des performances du modèle, la collecte continue de nouvelles données, l'adaptation du modèle aux changements et la réévaluation périodique du modèle pour maintenir sa précision et sa fiabilité dans des conditions changeantes.

Une analyse du Data Drift en production a été réalisée en utilisant la bibliothèque **evidently**. Cette bibliothèque permet de détecter d'éventuels changements de données (Data Drift) sur les principales caractéristiques entre les données d'entraînement et les données de test. Une synthèse est automatiquement générée sous la forme d'un tableau HTML d'analyse.

En comparant les distributions des 33 caractéristiques dans l'ensemble d'entraînement et l'ensemble de test, un drift a été détecté sur 6 caractéristiques, ce qui représente 18% du total. Avec un seuil de 50%, le Data Drift n'est donc pas détecté sur notre ensemble de test.

### Dataset Drift

Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

33	6	0.182
Columns	Drifted Columns	Share of Drifted Columns

Les 6 variables détectées ne sont pas les plus importantes pour le modèle sélectionné.

## 9 Limites et améliorations possibles

La partie de prétraitement et de création des variables a été réalisée de façon superficielle en se basant sur le kernel Kaggle fourni dans les ressources. Un travail plus approfondi, en collaboration avec les équipes métier, pourrait éventuellement améliorer les résultats de prédiction.



En raison de contraintes matérielles, la réduction de dimensionnalité a été poussée. Cette approche a été bénéfique pour obtenir un modèle simple, interprétable et avec des résultats convenables. Cependant, si nous souhaitons améliorer les prédictions, il serait envisageable de conserver plus de variables et d'adopter un modèle plus complexe.

La modélisation a été réalisée en se basant sur une hypothèse principale selon laquelle le coût d'un faux négatif est 10 fois supérieur à celui d'un faux positif. Cette hypothèse nécessite une meilleure exploration en collaboration avec les équipes métier.

Les variables les plus importantes pour le modèle sélectionné sont les scores externes. Afin d'expliquer de manière plus transparente les décisions d'octroi du crédit, il serait recommandé de fournir une note explicative détaillant les méthodes de calcul de ces scores.

Enfin, le tableau de bord interactif pourrait être amélioré pour répondre au mieux aux attentes et aux besoins des conseillers clients. Néanmoins, dans son état actuel, il est déjà informatif et les résultats de prédiction du modèle sont satisfaisants.

