2022

Formation Data Scientist

Soutenance

Projet 7

Implémentez un modèle de scoring

SOMMAIRE

Projet 7

Implémentez un modèle de scoring

SOMMAIRE

Contexte

Société financière qui propose des crédits à la consommation aux personnes ayant peu d'historique de prêt.

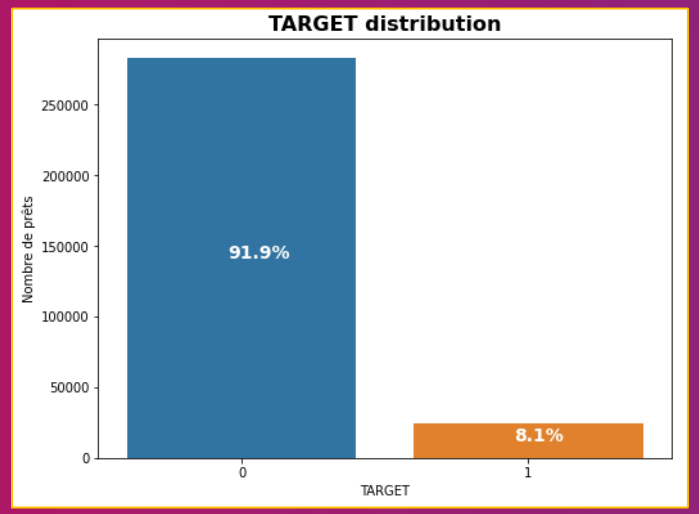
BESOIN

Développer un outil d'aide à la décision d'octroi de crédit.

LIVRABLES

- Un modèle de scoring
- Un dashboard restituant :
  - La probabilité de défaut de paiement d'un client
  - Des éléments justifiant le score
  - Ses informations personnelles

Problématique



Contrainte technique

Problème de classification binaire de classes déséquilibrées.

Entraînement des modèles en appliquant différentes méthodes d'ajustement de la distribution des classes.

Contrainte métier

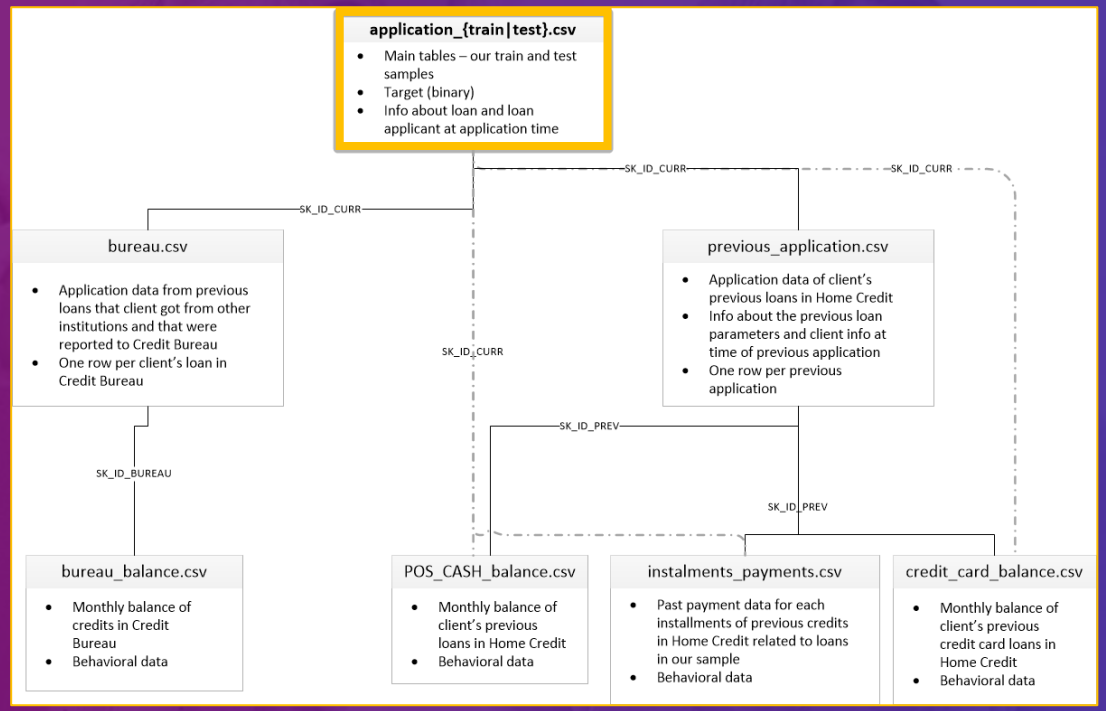
Limiter les pertes engendrées par les clients en défaut de paiement.

--> Restriction des faux négatifs (clients considérés comme solvable à tord) par le choix de métriques adaptées.

Les données

Caractéristiques

- 7 fichiers (Informations personnelles et détails crédit des clients)
- Fichier principal : application_train.csv
- Plus de 307 000 clients
- Client en défaut de paiement: 1
  Client sans défaut de paiement: 0



	Rows	Columns	Missing Values (%)	Duplicate (%)	object_dtype	float_dtype	int_dtype	bool_dtype
application_test.csv	48744	121	23.81	0.0	16	65	40	0
application_train.csv	307511	122	24.40	0.0	16	65	41	0
bureau.csv	1716428	17	13.50	0.0	3	8	6	0

Les données

Nettoyage et features engineering

- Le nettoyage et le features engineering des données ont été effectués à l'aide du kernel LightGBM_with_simple_features
- Remplacement de valeurs incohérentes par NaN values
- Encodage de variables catégorielles
- Création de variables agrégées par client
- Création de nouvelles variables

Format fichier d'entrainement : 79060, 555

Projet 7

Implémentez un modèle de scoring

SOMMAIRE

Modélisation

Démarche

Afin d'éviter le data leakage, mise en place d'une pipeline de 3 étapes :

- Etape 1 : Méthode de traitement du déséquilibre des données
- Etape 2 : Méthode de recalibrage des données (StandardScaler)
- Etape 3 : Modèle à entrainer

Les opération ne seront donc appliquées que sur les données d'entrainement.

Cette pipeline sera entrainée via un GridSearchcv (cv=5) avec modification d'un unique paramètre par modèle.

Modélisation

Méthodes de traitement du déséquilibre des données

1

Ajustement des poids de classe

Un poids est affecté à chaque classe. La classe minoritaire aura le poids le plus élevé. Cette méthode est mise en place à l'aide de l'option class_weight = 'balanced'.

2

Sous-échantillonnage ou *Undersampling*

La classe majoritaire est réduite à l'effectif de la classe minoritaire. La réduction est faite à l'aide de la fonction RandomUnderSampler.

3

Sur-échantillonnage ou Oversampling

La classe minoritaire est augmentée au niveau de l'effectif de la classe majoritaire. De nouvelles observations sont créées. Test de deux algorithmes : SMOTE et ADASYN.

Modélisation

Modèles à entrainer

Classifieur binaire

Logistic Regression

B A S E L I N E

Classifieur multi-classe

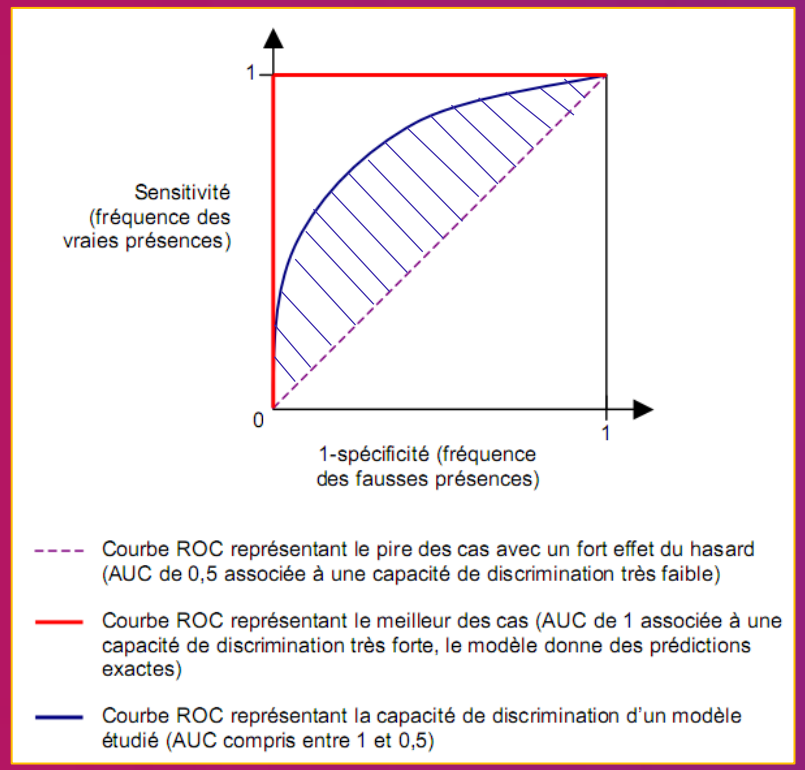
RandomForest

GradientBoosting

LGBM

Modélisation

Métrique d'évaluation – ROC_AUC



(source : Decout, 2007)

Courbe **R**eceiver **O**perator **C**haracteristic ou ROC est utilisée pour montrer la capacité de prédiction d'un classifieur binaire.

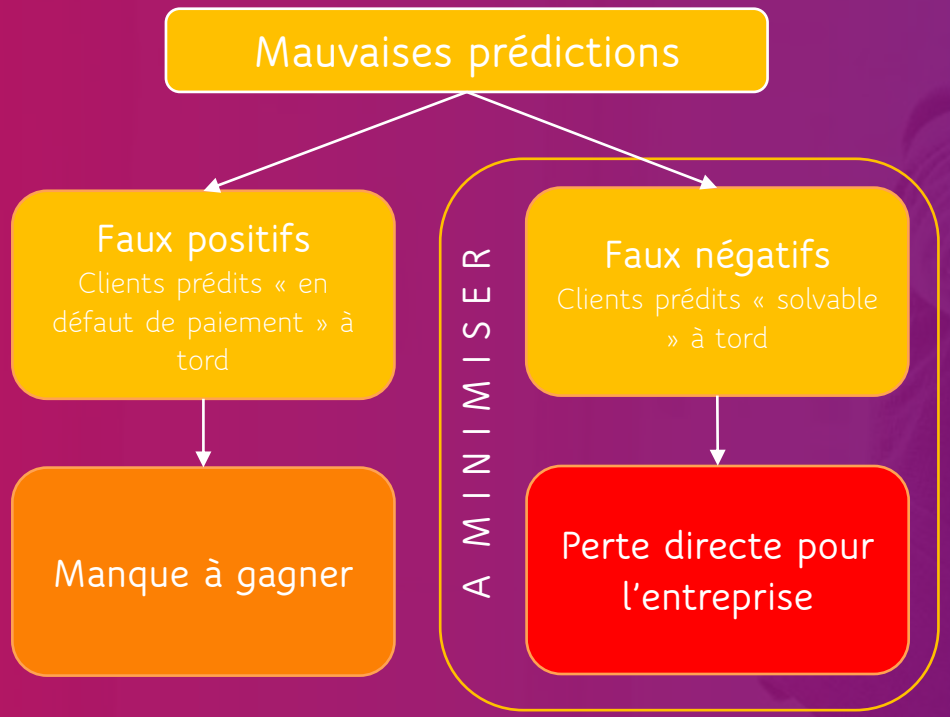
Sensitivité = Taux de vrais positifs ou *Recall*
Spécificité = Taux de faux positifs

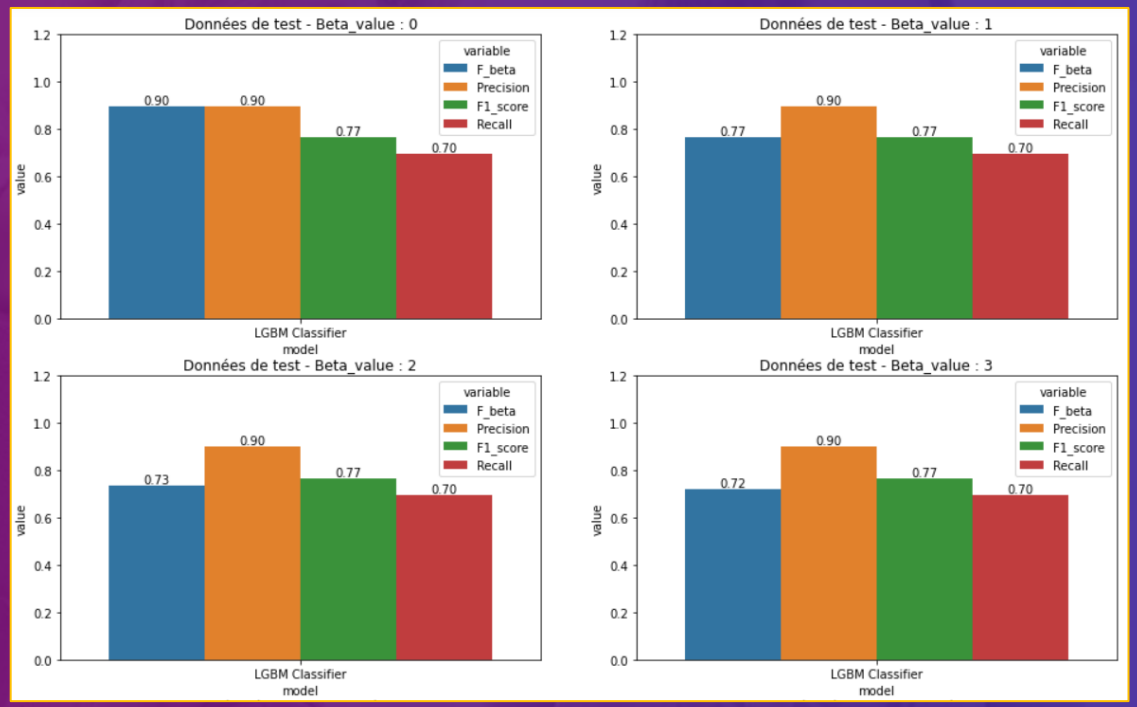
La mesure ROC_AUC est l'aire sous la courbe ROC, elle est comprise entre 0 et 1.

Un modèle dont les prédictions sont 100% fausses -> ROC_AUC =0
Un modèle dont les prédictions sont 100% vraies -> ROC_AUC = 1

Modélisation

Métriques d'évaluation – Fonction de coût - Fβ score



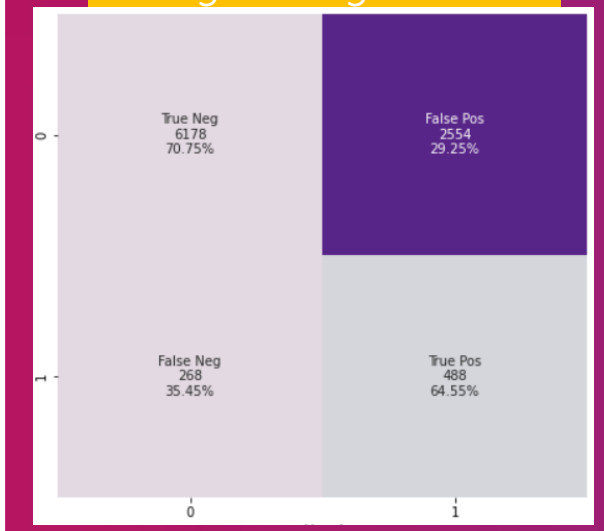


Minimiser les faux négatifs ==> Accorder plus d'importance au recall. - **Choix : beta = 3**

Modélisation

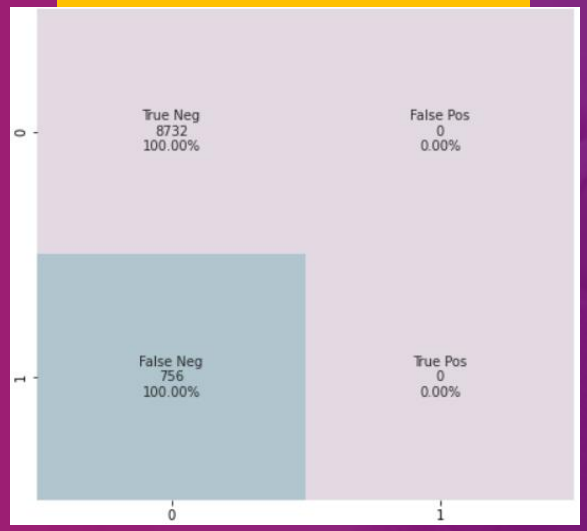
Résultat - Pondération

Logistic Regression



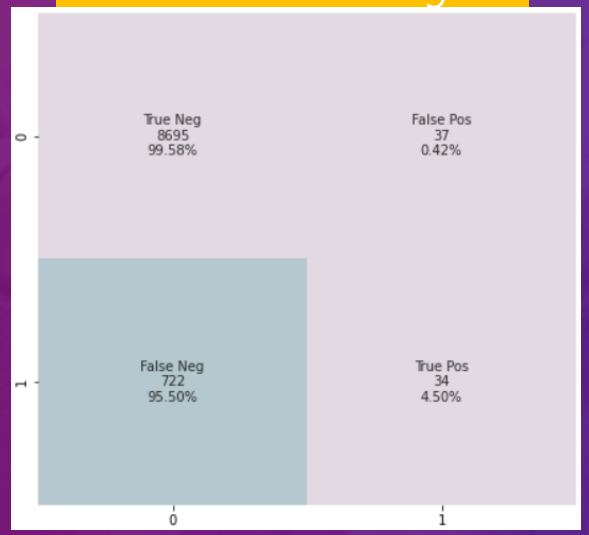
ROC_AUC	0,74
Fβ score	0,71
Exécution	1 min 8s

Random Forest



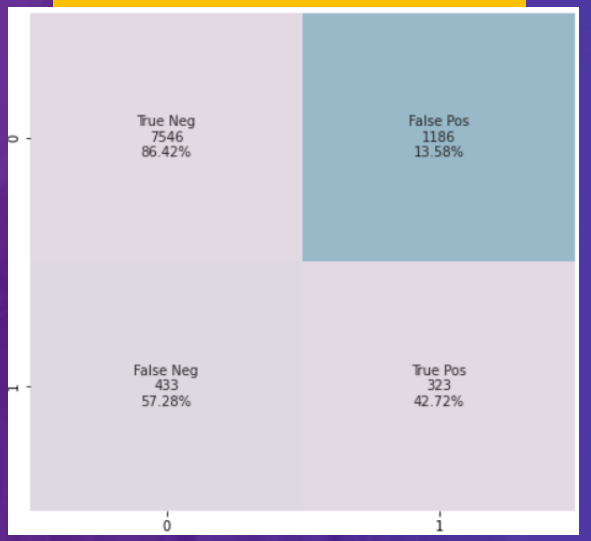
ROC_AUC	0,72
Fβ score	0,91
Exécution	12min 11s

Gradient Boosting



ROC_AUC	0,75
Fβ score	0,91
Exécution	37 min

LGBM

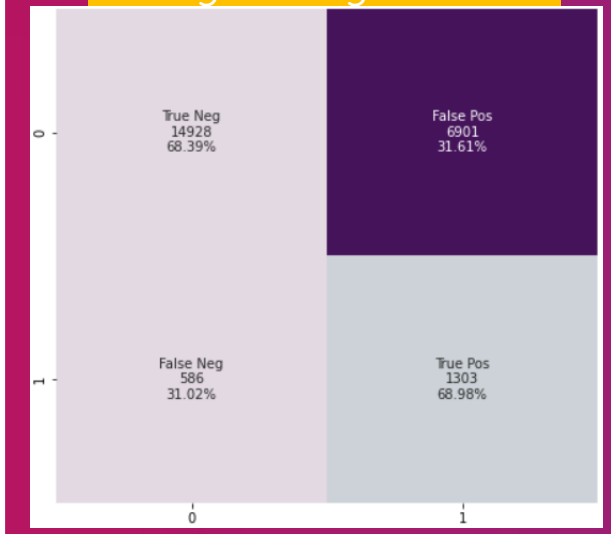


ROC_AUC	0,75
Fβ score	0,83
Exécution	34,1 s

Modélisation

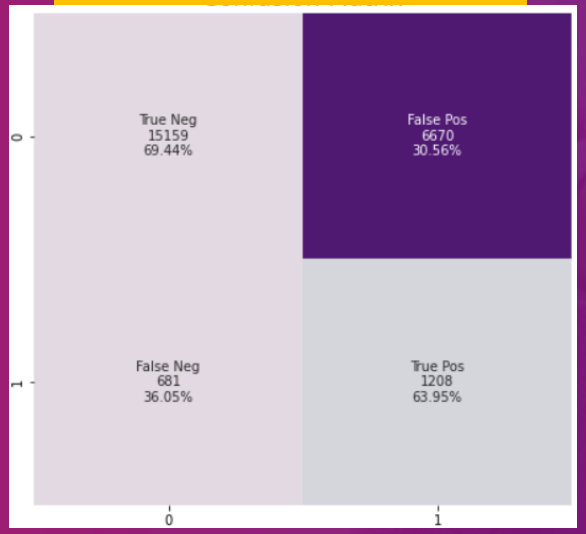
Résultat – Sous-échantillonnage (RandomUnderSampler)

Logistic Regression



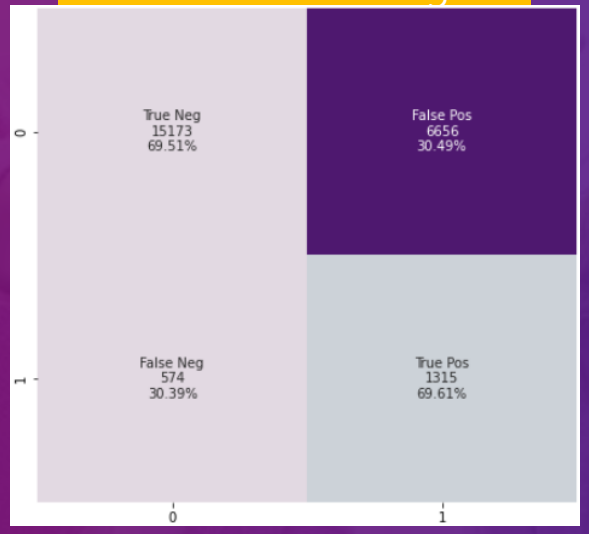
ROC_AUC	0,75
Fβ score	0,69
Exécution	22,6 s

Random Forest



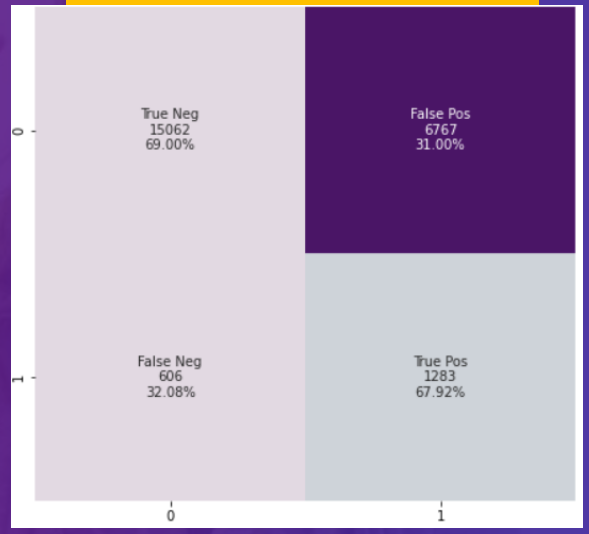
ROC_AUC	0,74
Fβ score	0,69
Exécution	2min 44s

Gradient Boosting



ROC_AUC	0,76
Fβ score	0,70
Exécution	13min 4s

LGBM



ROC_AUC	0,74
Fβ score	0,69
Exécution	28,1s

Modélisation

Résultat – Sur-échantillonnage (SMOTE)

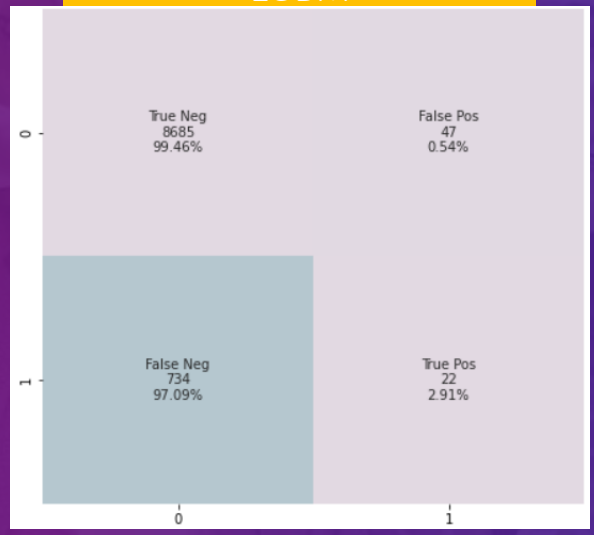
Logistic Regression

ROC_AUC	0,74
Fβ score	0,91
Exécution	1min 18s

Random Forest

ROC_AUC	0,69
Fβ score	0,91
Exécution	16min 47s

LGBM

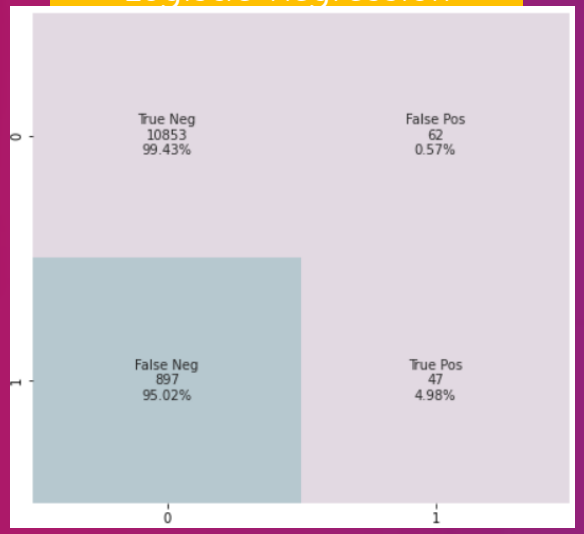


ROC_AUC	0,74
Fβ score	0,91
Exécution	1min 14s

Modélisation

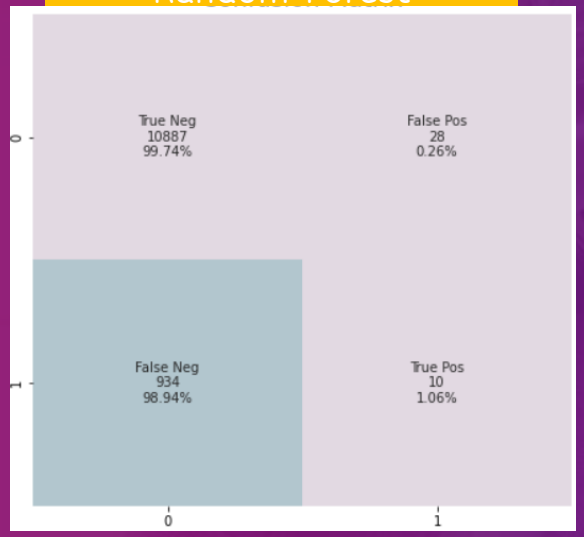
Résultat – Sur-échantillonnage (ADASYN)

Logistic Regression



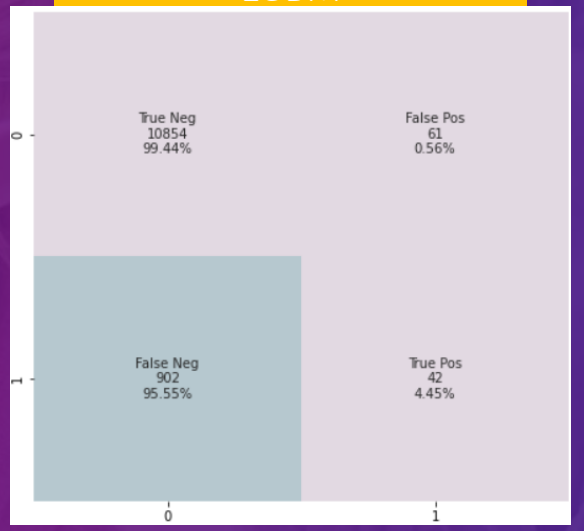
ROC_AUC	0,74
Fβ score	0,91
Exécution	2min 11s

Random Forest



ROC_AUC	0,70
Fβ score	0,91
Exécution	16min 47s

LGBM



ROC_AUC	0,75
Fβ score	0,91
Exécution	1min 14s

Modélisation

Résultat – Conclusion

Nous obtenons les meilleures performances avec la méthode d'ajustement de classe **sous-échantillonnage**.
Les modèles les plus performants sont Logistic Regression et LGBM.

Notre choix va se porter sur le modèle **LGBM** car il laisse davantage de place à l'optimisation.

Modélisation

Optimisation - Hyperopt

Paramètres à optimiser	Valeurs à tester
n_estimators	[100, 200, 300, 400, 500, 600]
learning_rate	[0,001, 0,03, 0,05]
max_depth	[3, 4, 5, 6, 7]
num_leaves	[5, 10, 15, 20, 25, 30, 35]

→

Valeurs sélectionnées

n_estimators: 600
learning_rate: 0,039
max_depth:7
num_leaves: 5

↓

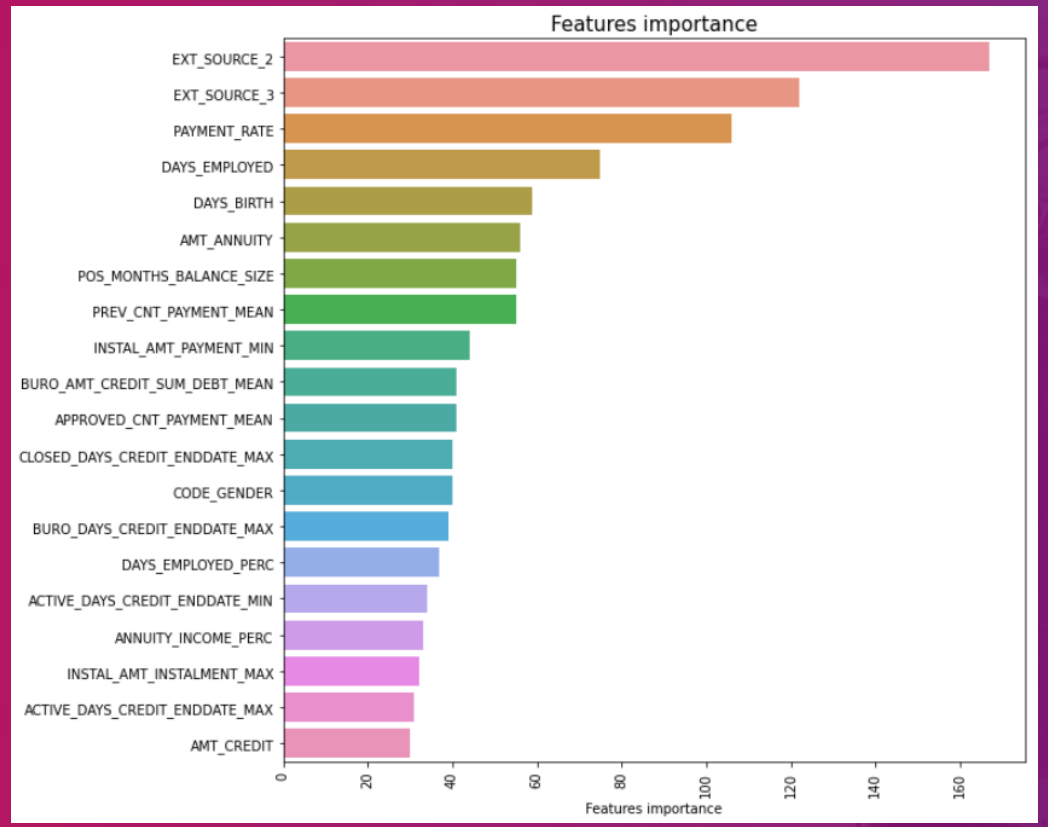
Performances

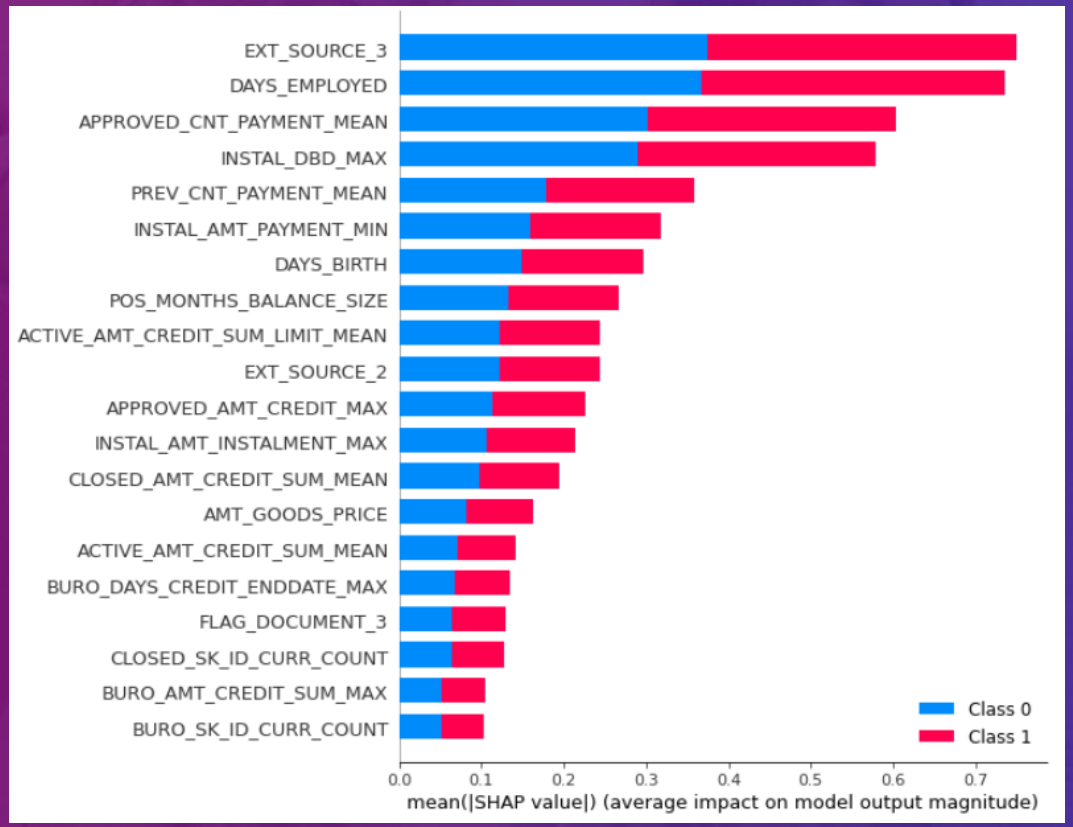
ROC_AUC	0,76
Fβ score	0,69
Exécution	10,5s

OBJECTIF : Maximiser la métrique **ROC_AUC**

Modélisation

Features importance (global)

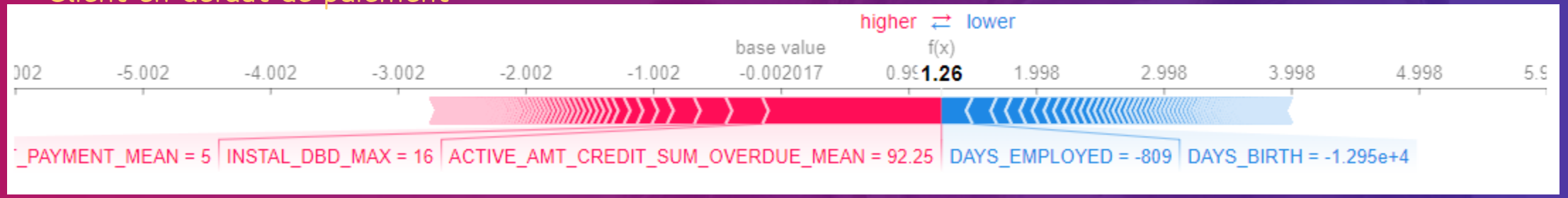




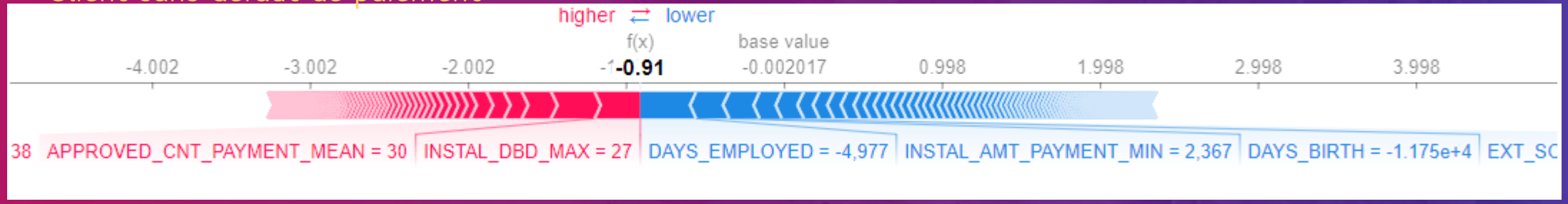
Modélisation

Features importance (local)

Client en défaut de paiement



Client sans défaut de paiement



Projet 7

Implémentez un modèle de scoring

SOMMAIRE

Dashboard

Caractéristiques techniques

Le tableau de bord est composé de 2 applications:

- Une API de type flask dont le but est calculer le score crédit . Elle est hébergée chez Heroku.
- Une application streamlit qui va récupérer ce score crédit et afficher toutes les informations d'aide à la décision d'octroi de crédit. Elle est également hébergée chez Heroku : https://scoring-credit-dashboard.herokuapp.com/

Le code est disponible chez GitHub.

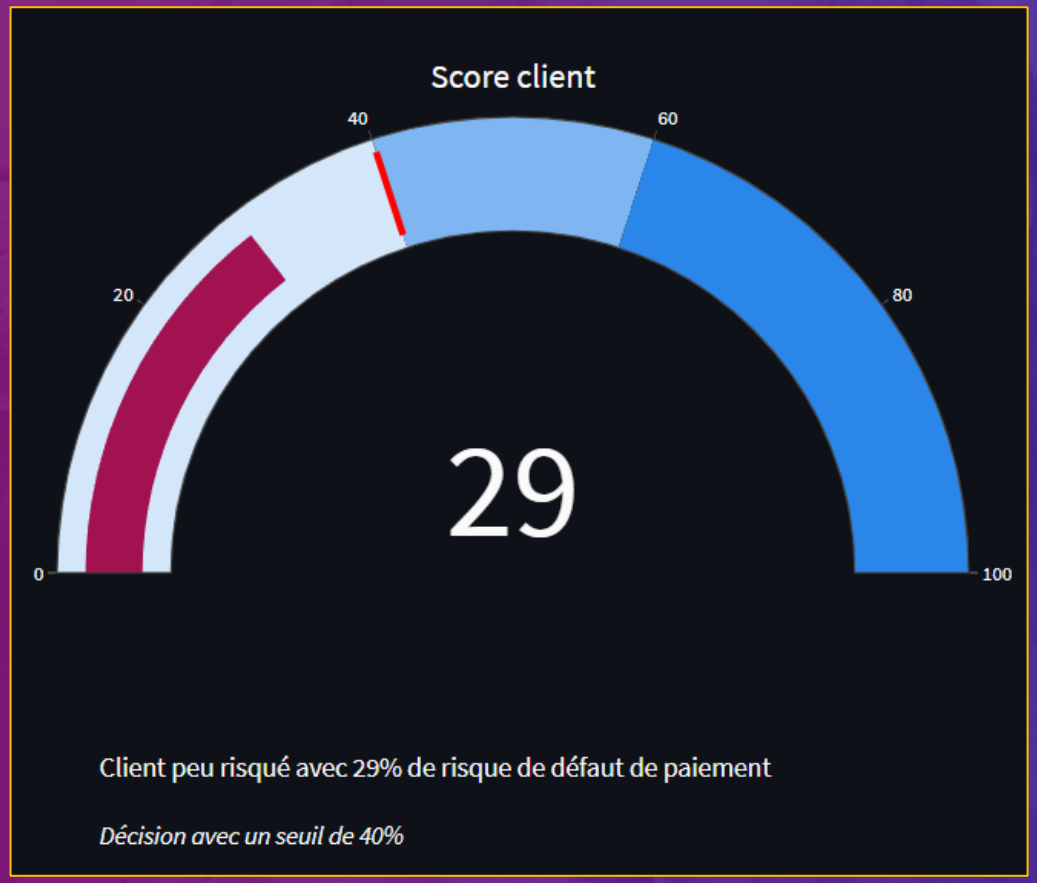
Dashboard

Présentation – Score client

Le score client correspond à sa probabilité d'être en défaut de paiement.

Le client est topé peu risqué si son score est inférieur à 40 %

Entre 40 % et 60 %, il est à risque et au-delà, il est très risqué.



Dashboard

Présentation – Features importance

Features importances locale

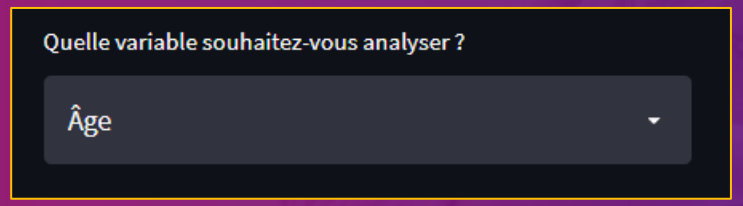
Features importance globale

Dashboard

Présentation – Comparaison client – Analyse simple

Deux variables à analyser:

- Âge
- Revenu total



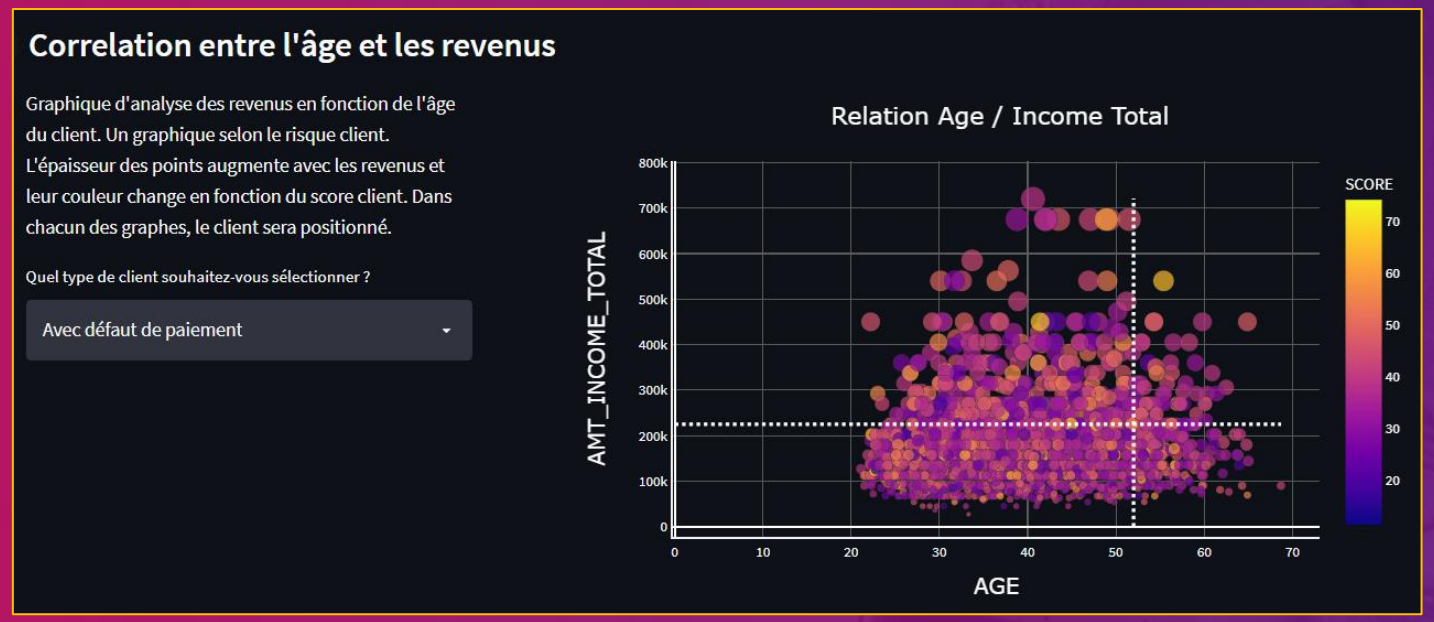
Un graphe par classe et le client est positionné.

Dashboard

Présentation – Comparaison client – Analyse bi-variée

Relation entre les deux variables pré-traitée:

- Âge
- Revenu total



Caractéristiques des points:

- Taille = montant des revenus
- Couleur = score de solvabilité

2021

MERCI DE VOTRE ATTENTION !