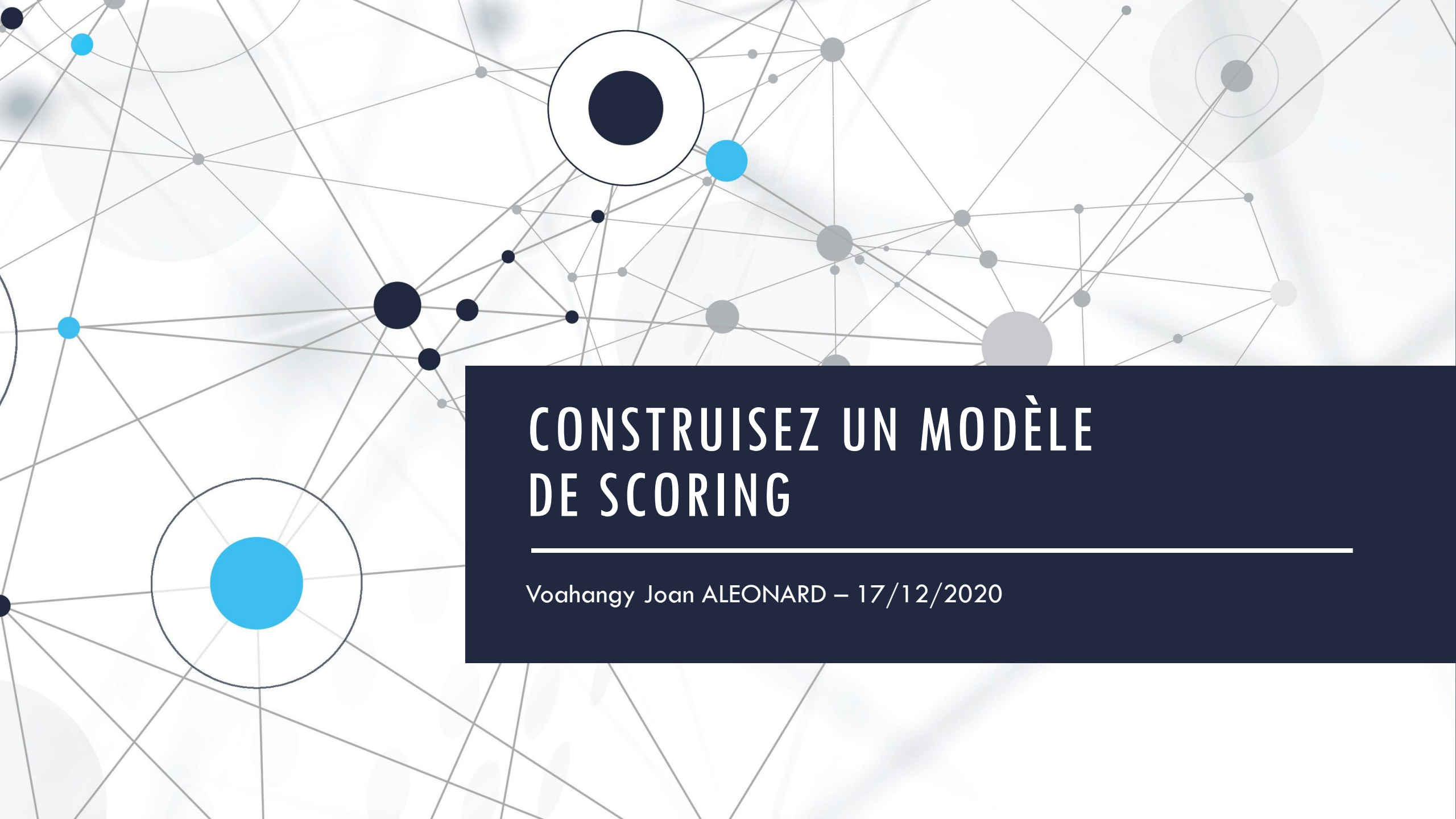


CONSTRUISEZ UN MODÈLE DE SCORING

Voahangy Joan ALEONARD – 17/12/2020

AGENDA DU JOUR

PRÉSENTATION DU PROJET

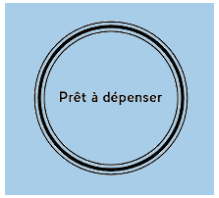
PRÉPARATION DES DONNÉES

MODÉLISATION ET OPTIMISATION

INTERPRÉTABILITÉ GLOBALE ET LOCALE

PRÉSENTATION DU PROJET

PRÉSENTATION DU PROJET



ENTREPRISE
- ACTIVITE ET OBJECTIFS -

Société de **crédits à la consommation** pour des personnes ayant **peu ou pas d'historique de prêt**, souvent sans compte en banque

Besoin critique

Sélectionner les **clients solvables** pour assurer la profitabilité de l'entreprise

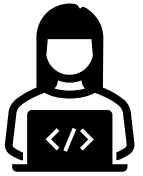
CHARGÉS DE RELATION CLIENT
- BESOIN(S) METIER -

Accord ? Rejet

Besoin critique

Disposer d'un outil:

- **facilement interprétable** pour prendre une décision éclairée
- Transparent concernant la **mesure de l'importance des variables** afin de pouvoir justifier leur décision vis-à-vis du client



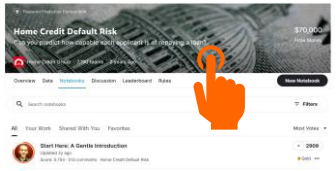
DATA SCIENTIST
- RÔLE -

Développer un **modèle de scoring** permettant de **prédire une probabilité de défaut (PD) de paiement**

Proposer des **métriques adaptés**, à la fois métier et business

Faire adhérer les équipes métier au **processus d'apprentissage supervisé** afin d'améliorer régulièrement le modèle

JEU DE DONNÉES ET APPROCHE DE MODÉLISATION



Base de données principale
« application_train »

~307.500 clients

121 features

⚠ **Données personnelles (sensibles)**

Sexe – Revenus – Situation familiale –Nombre d'enfants – Age – Temps travaillé avant la demande de prêt…

Données relatives au crédit (en cours et échus)

Identifiant client - Montant du crédit - Montant des annuités…

Des données externes

Scores achetés à des institutions financières

Variable à prédire : TARGET

Implémentation d'un modèle de classification

binaire

0 = Client solvable

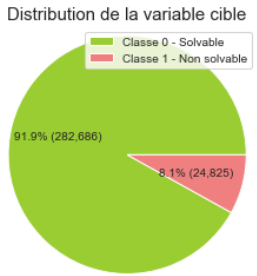
1 = Client non solvable

⚠ *Sortie* = Probabilité défaut paiement

en apprentissage supervisé

Des exemples de la valeur cible (target) sont fournis au modèle pour l'apprentissage

Spécificités
Classes déséquilibrées

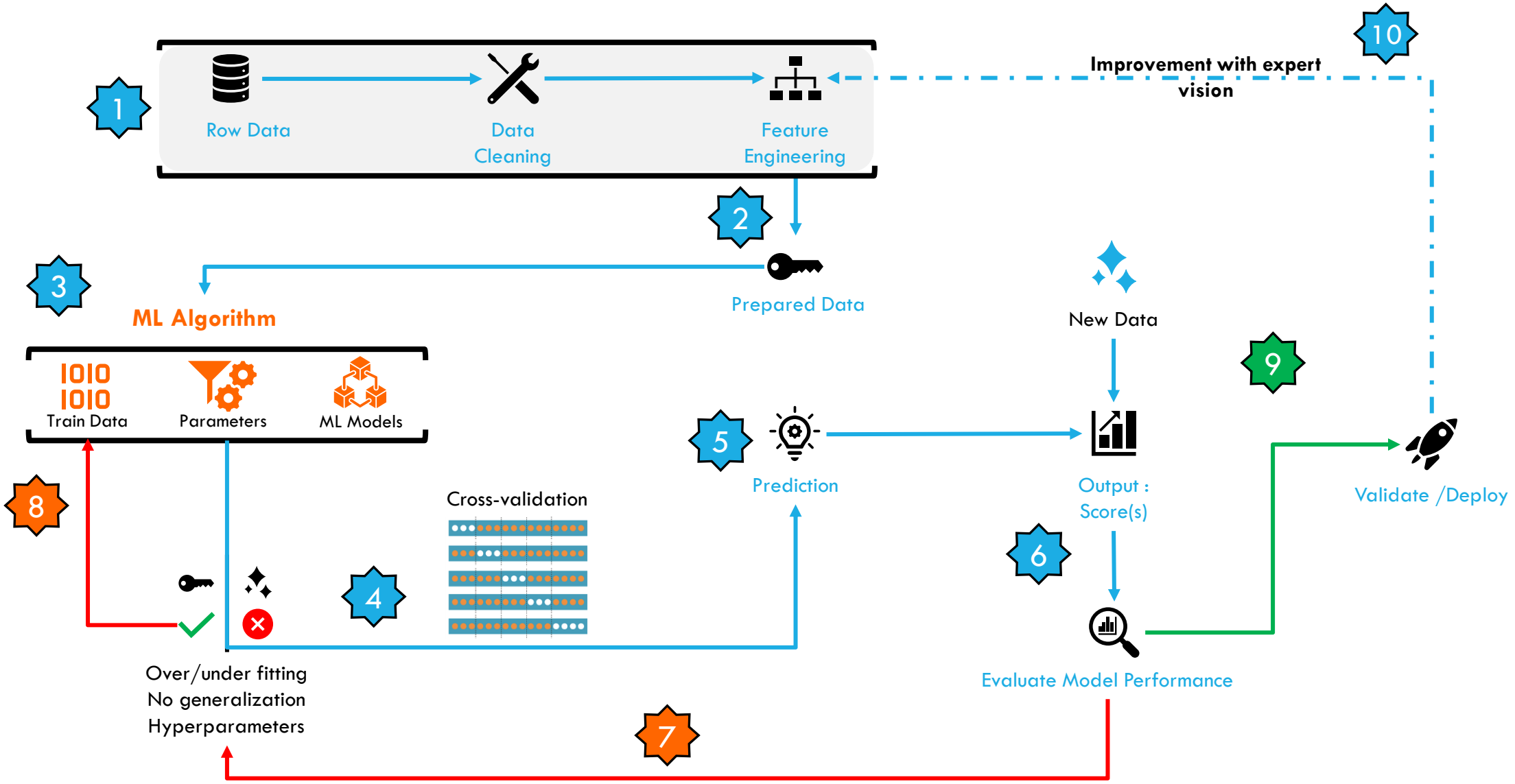


⚠ *imbalanced data*

Utilisation de techniques de **rééquilibrage des données**

Influence sur le **choix des métriques d'évaluation de performance des modèles**

PROCESSUS DE L'APPRENTISSAGE SUPERVISÉ



PRÉPARATION DES DONNÉES

ÉTAPES DE PRÉPARATION DES DONNÉES

Cleaning

- **Traitement des anomalies**
  - Suppression des étiquettes non présentes sur le TEST set
  - Remplacement des valeurs extrêmes par NaN
- **Traitement valeurs manquantes**
  - Suppression des colonnes ayant plus de 60% de valeurs manquantes
  - Imputation par la médiane pour le reste

Feature engineering

- **Encodage variables catégorielles**
  - Utilisation des LabelEncoder pour les variables binaires
  - Encodage par les fréquences pour les variables non binaires
- **Suppression/ajout de features**
  - Suppression des features à faible variance
  - Création de 6 features pour améliorer le modèle

Dimension TRAIN : 307.500 lignes, 90 features + TARGET

Modeling (split, normalisation)

Evaluation de différents modèles → **Optimisation des hyperparamètres** → **Prédictions et évaluation** → **Interprétabilité et décision**

- **train_test_split**
  test_size = 20%
- **Normalisation**
  (Mise à l'échelle)

MODÉLISATION ET OPTIMISATION

MÉTHODOLOGIE D'ÉVALUATION DES MODÈLES

Objectifs = Mesurer la capacité de généralisation et les temps de calcul

5 modèles évalués

Baseline = Naïve Bayes

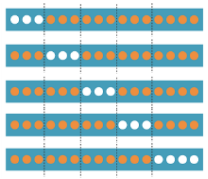
+ 2 linéaires : Régression logistique et Stochastic Gradient Descent

+ 2 non-linéaires : Random Forest et Light Gradient Boosting Machine

Rééquilibrage des classes

Utilisation du paramètre `class_weight = « balanced »` pour rééquilibrer les classes de la variable cible TARGET

Cross validation



Utilisation de l'intégralité du TRAIN set pour l'entrainement ET la validation

Choix de 3 folds pour nos modèles

StratifiedKFold

Création des sous-ensembles de validation croisée en gardant la même proportion d'exemples pour chaque classe (à l'image du jeu de données complet)

Métriques d'évaluation

AUC

Probabilité de défaillance (*0 → 100%)*

Recall

Capacité du modèle à détecter tous les clients non-solvables

F1-Score

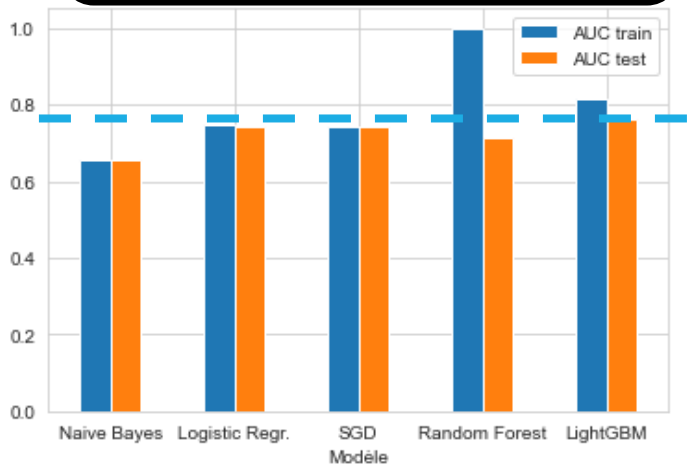
Moyenne harmonique du **recall** et de la **precision** (capacité du modèle à détecter les VRAIS non-solvables)

Temps d'apprentissage et de prédiction

Les temps d'exécution doivent être raisonnables

COMPARAISON DES MODÈLES PAR MÉTRIQUE

Score AUC : TRAIN et TEST



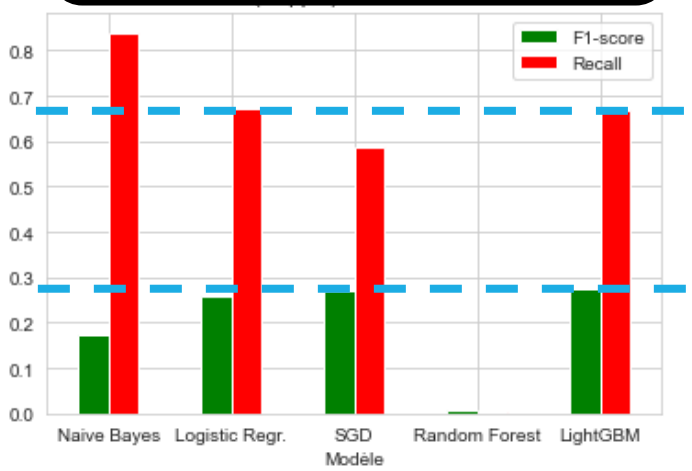
Meilleure performance AUC en TEST pour Light Gradient Boosting Machine (LightGBM).

Performances similaires sur pour Logistic Regression et Stochastic Gradient Descent.

Sur-apprentissage pour Random Forest

Naive Bayes, étant la baseline

Métriques - Rappel et F1-score



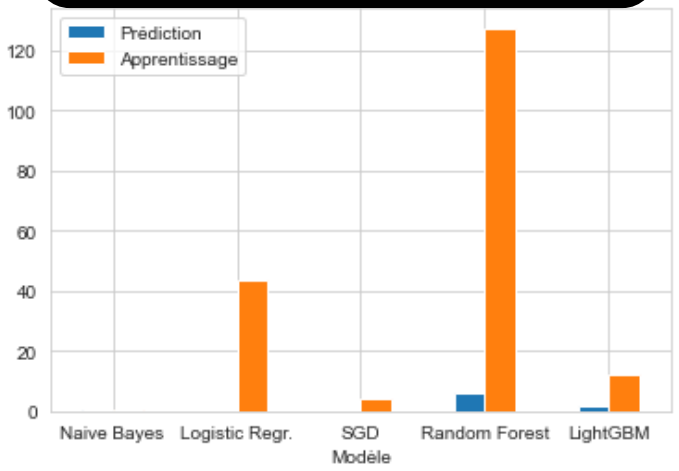
Le compromis Rappel/Précision avec F1-score et le rappel sont maximisés pour Logistic Regression et LightGBM

SGD a une performance correcte

Random Forest ne semble pas adapté

Naive Bayes sanctionne fortement les solvables, qu'il prédit non-solvables ; on voudrait qu'il sanctionne plutôt les non-solvables prédits solvables

Temps apprentissage et prédiction



LightGBM, SGD et Naïve Bayes sont très rapides

Logistic Regression prend un peu plus de temps mais rien de rhédibitoire

Random Forest est très gourmand en temps de calcul

LightGBM

OPTIMISATION DES HYPERPARAMÈTRES : LIGHTGBM

Objectif

Augmenter les **performances du modèle sélectionné** en optimisant la métrique AUC.

GridSearchCV

Méthode permettant d'**évaluer la meilleure combinaison** d'hyperparamètres

```
# Créer le modèle à optimiser
lightgbm = LGBMClassifier(random_state=random_state,
                          class_weight=class_weight,
                          objective="binary")
```

```
# Créer les espaces de recherche (space search) des hyperparamètres
params_grid_lgbm = {"n_estimators":[500, 1000],
                    "max_depth":[8, 12],
                    "learning_rate":[0.01, 0.02],
                    "num_leaves":[30, 50]}
```

```
# Créer le Grid Search
gscv_lgbm = GridSearchCV(lightgbm,
                         params_grid_lgbm,
                         cv=cv,
                         scoring=scoring,
                         refit="auc",
                         verbose=verbose,
                         n_jobs=n_jobs)
```

Avant / Après

Mesure d'AUC avant :
0.7604

Mesure d'AUC après :
0.7672

ÉVALUATION DÉTAILLÉE DU MODÈLE CHOISI

La matrice de confusion

La matrice de confusion mesure les erreurs de prédictions du modèles par rapport à la classification réelle

Classes réelles		Classes prédites	
		0	1
	0	Vrai négatif (VN)	Faux positif (FP)
	1	Faux négatif (FN)	Vrai positif (VP)

VN : Prédit négatif et réellement négatif
FN : Prédit négatif MAIS positif en réalité
FP : Prédit positif MAIS négatif en réalité
VP : Prédit positif et réellement positif

Confusion Matrix - LightGBM :

	Prédit 0	Prédit 1	
Réel 0	40410	16125	56535
Réel 1	1591	3374	4965
	42001	19499	

Le rapport de classification

Le rapport de classification analyse les métriques par classe

classe	précision	rappel	F1-score	support
0	0.96	0.72	0.82	56535
1	0.17	0.68	0.28	4965
				61500

Recall = VP/(VP+FN)
Precision = VP/(VP + FP)
F1 Score = 2*(Recall * Precision) / (Recall + Precision)
Specificity = VN / (VN+FP)
1-Specificity = FP / (FP+VN)

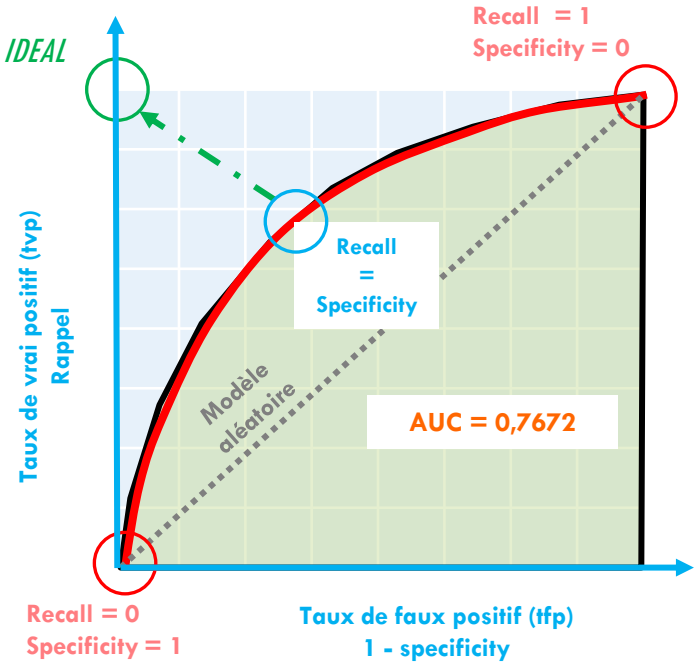
Specificity : capacité du modèle à détecter les clients solvables

⚠ Le modèle a tendance à sanctionner des clients solvables

La courbe ROC

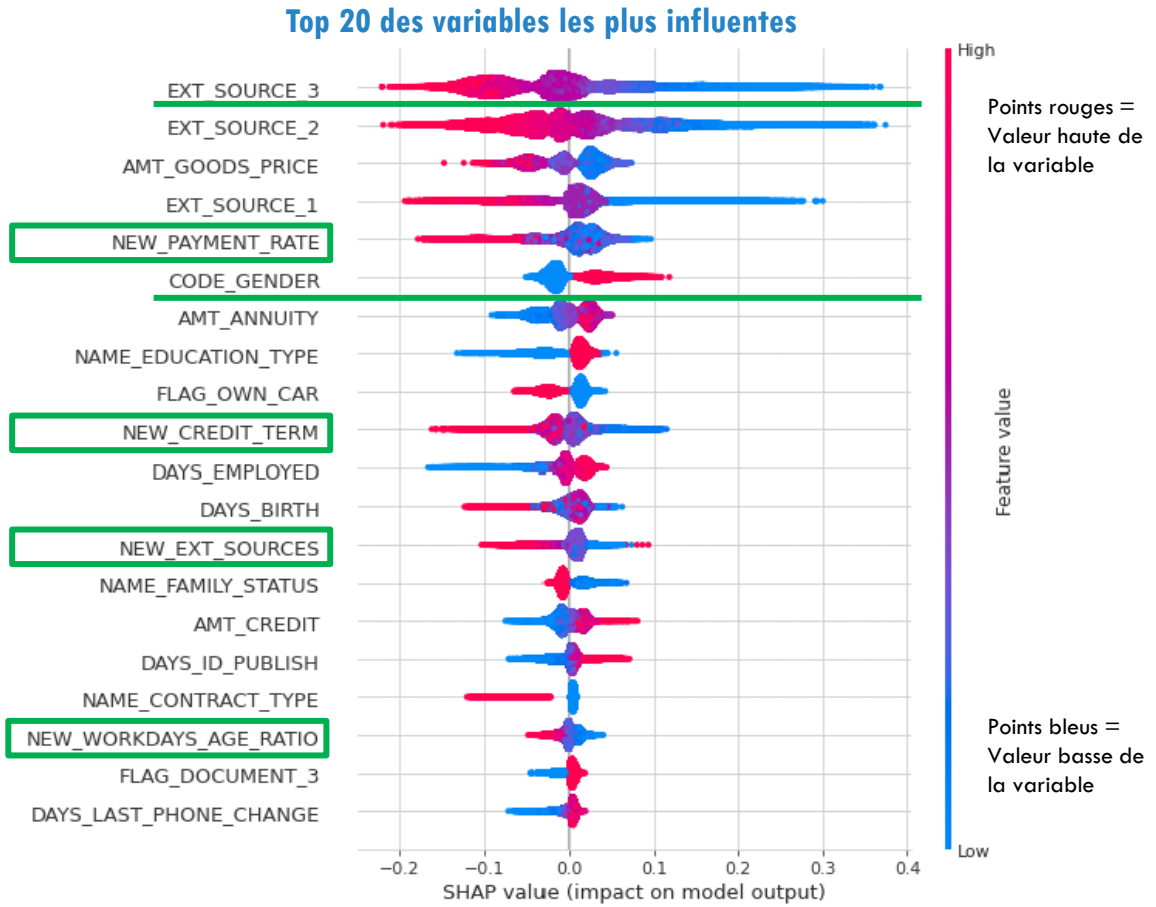
La courbe ROC représente le Recall et la Specificity en fonction du seuil de classification (ici, threshold = 0,5 par défaut)

Un modèle IDEAL a un recall = 1 et une specificity = 1



INTERPRÉTABILITÉ GLOBALE ET LOCALE

MESURE DE L'IMPORTANCE GLOBALE DES VARIABLES



EXT_SOURCE_3 : impact négatif quand la valeur de variable est élevée

CODE_GENDER : impact positif quand la valeur de variable est élevée

La présence des 4 nouvelles features sur 6 dans le top 20

Ce graphique nous apporte des informations sur les **features qui ont influencées GLOBALEMENT les valeurs prédites.**

En effet, chaque feature peut augmenter ou diminuer la probabilité de défaut de paiement, en fonction du sens de leur influence.

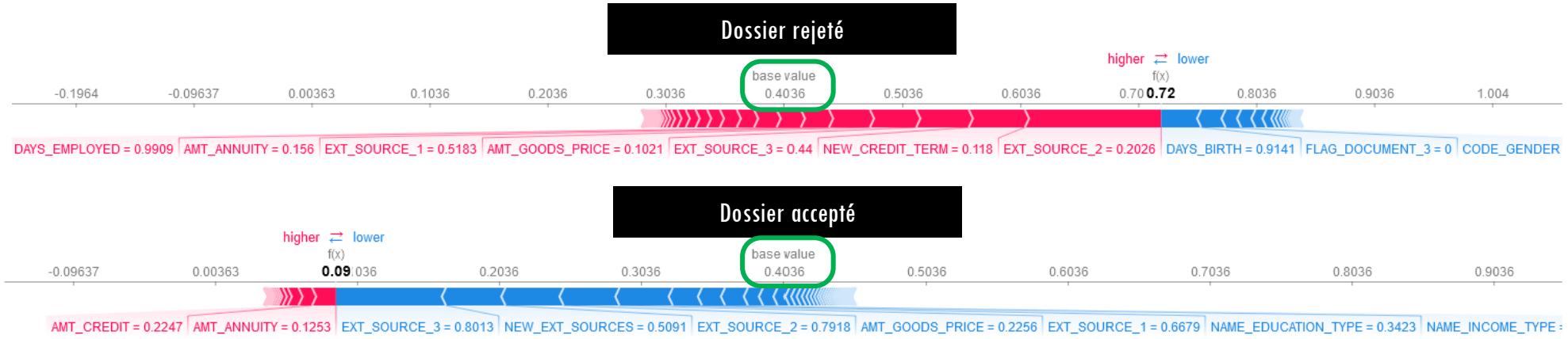
Ex: si la couleur rouge est à gauche de 0, cela veut dire que plus la valeur de la variable diminue, plus elle va contribuer à prédire un défaut de paiement

Pourquoi la connaissance des variables influentes au global est essentielle ?

Elle permet de **vérifier** la cohérence du modèle avec les experts métier, et de leur **donner confiance** dans les prédictions.

Pour nous en tant que Data Scientist, c'est le moyen de **comprendre les mécanismes sous-jacents** du modèle, de le **valider** (voire l'**améliorer**), ou de le **corriger.**

INTERPRÉTABILITÉ LOCALE



Les graphiques ci-dessus représentent l'impact des variables importantes sur la **prédiction de probabilité de Défaut pour un client donné (=local).**

La **valeur de base** *(base value)* est la moyenne de prédiction de tous les individus.

Dossier rejeté : les grandes valeurs des variables en rouge contribuent à augmenter la proba de défaut.

Dossier accepté : les petites valeurs des variables en bleu contribuent à baisser la proba de défaut.

Pourquoi la connaissance des variables influentes en local est essentielle ?

Elle permet aux métiers de **prendre des décisions** éclairées, basées sur des critères objectifs.

Elle permet également de **justifier** les raisons d'un rejet de demande de prêt auprès d'un client.

Elle permet de **se conformer à la réglementation** en vigueur : par ex, RGDP interdit les décisions émanant uniquement de machines.

UNE MÉTRIQUE BUSINESS D'AIDE À LA DÉCISION

Le concept

Utilité

Donner une **vision globale business et de la hauteur** aux chargés de clientèle, en complément de la probabilité prédite par le modèle.

Elle donne une fonction coût subie par l'entreprise en cas de mauvaise décision

Principes

Evaluation du risque en fonction des différents seuils de classification – risqué provenant :

- d'une **perte réelle** due à l'acceptation d'un client non-solvable
- d'un **manque à gagner** dû au rejet d'un client solvable

Les hypothèses

Données d'entrée

- Le montant du prêt demandé : ***M***
- La perte (estimée) sur un prêt non remboursé : ***pe = 70%***
- Le gain (estimé) sur un prêt remboursé : ***gn = 20%***
- La proportion de faux négatifs sur le total individus : ***p(FN)***
- La proportion de faux positifs sur le total individus : ***p(FP)***

Données de sortie

(basée sur les erreurs de classification de la matrice de confusion)

- Perte réelle = **M x pe x p(FN)**
- Manque à gagner = **M x gn x p(FP)**
- **Perte totale** = Perte réelle + manque à gagner

Exemple

- Pour un prêt ***M = 100.000***
- **Perte réelle** : 100.000 x 70% x (1.591/61.500) = 1.813
- **Manque** : 100.000 x 20% x (16.125/61.500) = 5.244
- **Perte totale** : 1.813 + 5.244 = 7.057

```
Confusion Matrix - LightGBM :
          Prédit 0  Prédit 1
Réel 0      40410     16125
Réel 1       1591      3374
```

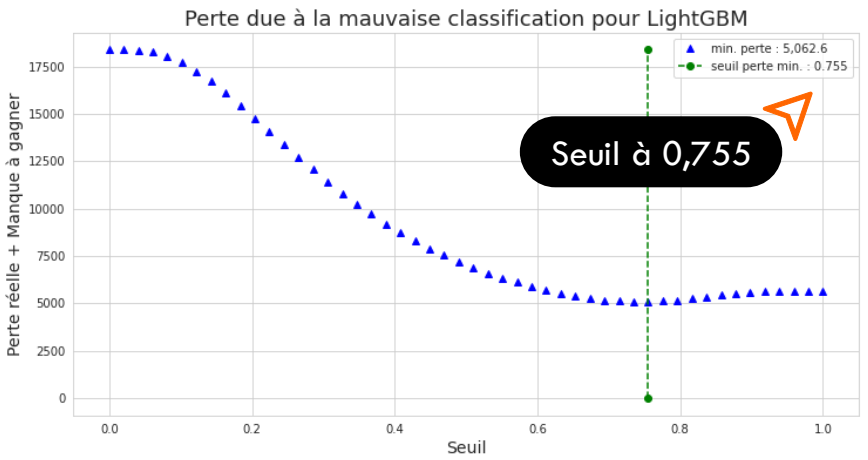
p(FN) = 2,59%
p(FP) = 26,22%

La modélisation

Estimation des pertes par seuil de classification

	seuil	perte nette	opport. manquée	perte ttle
0	0.000000	0.00	18385.37	18385.37
1	0.020408	0.00	18385.37	18385.37
2	0.040816	0.00	18372.68	18372.68
3	0.061224	0.00	18285.53	18285.53
4	0.081633	2.28	18055.28	18057.56
5	0.102041	15.93	17695.61	17711.54
...				
45	0.918367	5636.42	1.63	5638.05
46	0.938776	5651.22	0.00	5651.22
47	0.959184	5651.22	0.00	5651.22
48	0.979592	5651.22	0.00	5651.22
49	1.000000	5651.22	0.00	5651.22

Recherche de seuil de perte 'optimal'



SYNTHÈSE

QUE POUVONS-NOUS EN CONCLURE ?

Mise en place des bases …

- Sélection des métriques
- Equilibrage des données
- Optimisation des hyperparamètres

… avec des résultats …

dépendant fortement des transformations et traitement effectués

ET des paramètres de(s) modèle(s) choisis

… pouvant être améliorés par …

Une **meilleure compréhension du business avec les équipes métier** pour une meilleure préparation des données (traitement des valeurs manquantes, création/suppression de features, etc.)

Une **approfondissement des mécanismes sous-jacents des modèles** afin de mieux choisir les paramètres influents.

L'utilisation de la **complétude des données** (avec un meilleur séquençage du code pour optimiser les temps d'exécution)

QUESTIONS / RÉPONSES

ANNEXES

RÉFÉRENCES

- Home Credit Default Risk : a gentle introduction (Will Koehrsen) – Kaggle
- Les classes déséquilibrées
- scikit-learn : GridSearchCV
- LightGBM
- L'interprétabilité en machine learning
- Interprétation des modèles (SHAP)

Ce document a été produit dans le cadre de la soutenance du projet n°4 du parcours Ingénieur IA d'OpenClassrooms :
« Construisez un modèle de scoring »

Mentor : Thierno DIOP
Evaluateur : Bertrand BEAUFILS

OPENCLASSROOMS
En partenariat avec
CentraleSupélec
Microsoft