

Mission - Élaborez le modèle de scoring



Comment allez-vous procéder ?

Cette mission suit un scénario de projet professionnel.

Vous pouvez suivre les étapes pour vous aider à réaliser vos livrables.

Avant de démarrer, nous vous conseillons de :

- lire toute la mission et ses documents liés ;
- prendre des notes sur ce que vous avez compris ;
- consulter les étapes pour vous guider ;
- préparer une liste de questions pour votre première session de mentorat.

Prêt à mener la mission ?

Vous êtes Data Scientist au sein d'une société financière, nommée "**Prêt à dépenser**", 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.)



Vous aurez besoin de joindre les différentes tables entre elles.

Votre mission :

1. Construire un modèle de scoring qui donnera une prédition sur la probabilité de faillite d'un client de façon automatique.
2. Analyser les features qui contribuent le plus au modèle, d'une manière générale (feature importance globale) et au niveau d'un client (feature importance locale), afin, dans un soucis de transparence, de permettre à un chargé d'études de mieux comprendre le score attribué par le modèle.
3. Mettre en production le modèle de scoring de prédition à l'aide d'une API et réaliser une interface de test de cette API.
4. Mettre en œuvre une approche globale MLOps de bout en bout, du tracking des expérimentations à l'analyse en production du data drift.

Michaël, votre manager, vous incite à sélectionner un ou des kernels Kaggle pour vous faciliter l'analyse exploratoire, la préparation des données et le feature engineering nécessaires à l'élaboration du modèle de scoring.

Si vous le faites, vous devez analyser ce ou ces kernels et le ou les adapter pour vous assurer qu'il(s) répond(ent) aux besoins de votre mission.

Par exemple vous pouvez vous inspirer des kernels suivants :

- [Pour l'analyse exploratoire](#)
- [Pour la préparation des données et le feature engineering.](#)

C'est optionnel, mais nous vous encourageons à le faire afin de vous permettre de vous focaliser sur l'élaboration du modèle, son optimisation et sa compréhension.

De : Mickael

À : moi

Objet : Besoins et conseils pour l'élaboration d'un outil de credit scoring

Bonjour,

Afin de pouvoir faire évoluer régulièrement le modèle, je souhaite tester la mise en œuvre une démarche de type MLOps d'automatisation et d'industrialisation de la gestion du cycle de vie du modèle.

Vous trouverez en pièce jointe **la liste d'outils à utiliser** pour créer une plateforme MLOps qui s'appuie sur des outils Open Source.

Je souhaite que vous puissiez mettre en oeuvre au minimum **les étapes orientées MLOps** suivantes :

- Dans le notebook d'entraînement des modèles, générer à l'aide de MLFlow un tracking d'expérimentations
- Lancer l'interface web 'UI MLFlow" d'affichage des résultats du tracking
- Réaliser avec MLFlow un stockage centralisé des modèles dans un "model registry"
- Tester le serving MLFlow
- Gérer le code avec le logiciel de version Git
- Partager le code sur Github pour assurer une intégration continue
- Utiliser Github Actions pour le déploiement continu et automatisé du code de l'API sur le cloud
- Concevoir des tests unitaires avec Pytest (ou Unittest) et les exécuter de manière automatisée lors du build réalisé par Github Actions

J'ai également rassemblé des conseils pour vous aider à vous lancer dans ce projet !

Concernant **l'élaboration du modèle** soyez vigilant sur deux points spécifiques au contexte métier :

- Le déséquilibre entre le nombre de bons et de moins bons clients doit être pris en compte pour élaborer un modèle pertinent, avec une méthode au choix



- Vous pourrez supposer, par exemple, que le coût d'un FN est dix fois supérieur au coût d'un FP
- Vous créerez un score "métier" (minimisation du coût d'erreur de prédiction des FN et FP) pour comparer les modèles, afin de choisir le meilleur modèle et ses meilleures hyperparamètres. Attention cette minimisation du coût métier doit passer par l'optimisation du seuil qui détermine, à partir d'une probabilité, la classe 0 ou 1 (un "predict" suppose un seuil à 0.5 qui n'est pas forcément l'optimum)
- En parallèle, maintenez pour comparaison et contrôle des mesures plus techniques, telles que l'AUC et l'accuracy

D'autre part je souhaite que vous mettiez en œuvre une démarche d'élaboration des modèles avec **Cross-Validation et optimisation des hyperparamètres, via GridsearchCV ou équivalent.**

Un dernier conseil : si vous obtenez des scores supérieurs au 1er du challenge Kaggle (AUC > 0.82), posez-vous la question si vous n'avez pas de l'overfitting dans votre modèle !

Vous exposerez votre **modèle de prédiction sous forme d'une API** qui permet de calculer la probabilité de défaut du client, ainsi que sa classe (accepté ou refusé) en fonction du seuil optimisé d'un point de vue métier.

Le déploiement de l'API sera réalisée sur une plateforme Cloud, de préférence une solution gratuite.

Je vous propose d'utiliser un Notebook ou une application Streamlit pour réaliser en local **l'interface de test de l'API**.

Bon courage !

Mickael

Pièce-jointe :

- [Liste des outils MLOps à utiliser](#)

Il est temps de plonger dans le travail ! Vous pouvez suivre les étapes ci-dessous pour vous guider.

Étapes

Étape 1 : Préparez l'environnement d'expérimentation

- Initiez un environnement MLFlow permettant le tracking lors de l'entraînement des modèles.
- Mettez en place l'UI pour la visualisation et la comparaison des expérimentations, ainsi que le stockage de manière centralisée des modèles.

Étape 2 : Préparez vos données à la modélisation

- Choisissez un Kernel Kaggle pertinent et riche en feature engineering.
- Adaptez le Kernel à votre environnement technique.

Étape 3 : Créez un score métier pour l'entraînement de vos modèles

- Définissez votre score via une pondération sur les faux positifs et faux négatifs.



Étape 4 : Simulez et comparez plusieurs modèles



- Testez plusieurs modèles via une GridSearchCV par exemple tout en les comparant à une baseline.
- Gérez si nécessaire le déséquilibre entre les classes.
- Analysez la feature importance globale et locale du meilleur modèle retenu.

! [Avez-vous une suggestion pour nous ?](#)

[Suivant](#)