



# Introduction

## Objectif du projet :

Je suis 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.).

De plus, les chargés de relation client ont fait remonter le fait que les clients sont de plus en plus demandeurs de **transparence** vis-à-vis des décisions d'octroi de crédit. Cette demande de transparence des clients va tout à fait dans le sens des valeurs que l'entreprise veut incarner.

**Prêt à dépenser** décide donc de **développer un dashboard interactif** pour que les chargés de relation client puissent à la fois expliquer de façon la plus transparente possible les décisions d'octroi de crédit, mais également permettre à leurs clients de disposer de leurs informations personnelles et de les explorer facilement.

Ce dashboard répond à des impératifs métiers, il doit être lisible, rapidement interprétable et doit pouvoir être montré au client. Les informations doivent donc être conformes à la RGPD et compréhensible aussi bien par un conseiller non informaticien et non data scientist. La grande partie du travail sera simplifiée par la mise en place du modèle de machine Learning, mais le conseiller doit toujours être capable d'octroyer un crédit de son propre gré.

Afin de tenir compte de cette contrainte, les scores compris entre 45 % et 55 % seront affichés en orange, et à la main du conseiller.

Les dossiers créés et leur fonction sont les suivants :

## Architecture du repository et contenu des dossiers

**.github/workflows** : ce dossier est composé du fichier .yml permettant le déploiement et le test en continu de l'api.

**api** : contient les dossiers **\_\_pycache\_\_**, **static** qui contient les fichiers images du dashboard, **templates** avec le fichier index.html, structure html de l'api et enfin le dossier **test** : fichiers de test du projet

**Les fichiers de codes pour l'api sont :**

api2.py : code de l'api

/test/test\_api.py : fichiers de test pytest

**appli** : contient les dossiers **static** qui contient les fichiers images du dashboard, **templates** avec le fichier index.html, structure html de l'api

**Les fichiers de codes pour l'api sont :**

dash\_api.py : code du dashboard

/test/test\_app.py

**data\_drift** : contient les fichiers code et html pour le datadrift

**mlartifact** : ce dossier contient les artefacts des modèles enregistrés sur MLFLOW

**mlruns** : ce dossier est contient les runs MLFLOW

**models** : Les modèles sont dans ce dossier. Ils sont enregistrés en version « normale », sans hyperparamètres, et en version pré entraînée, avec les hyperparamètres accessibles. Il y a la version finale du modèle LightGbm

**notebooks** : les notebooks sont les suivants :

EDA.ipynb : Notebook de l'analyse de donnée

feature\_engineering : Notebook de creation du dataframe et feature engineering

Modelisation\_MLFLOW : Notebook de modélisation et de création du modèle

Pipeline : Notebook de test du pipeline

Les fichiers .py sont les fichiers de fonctions personnalisées