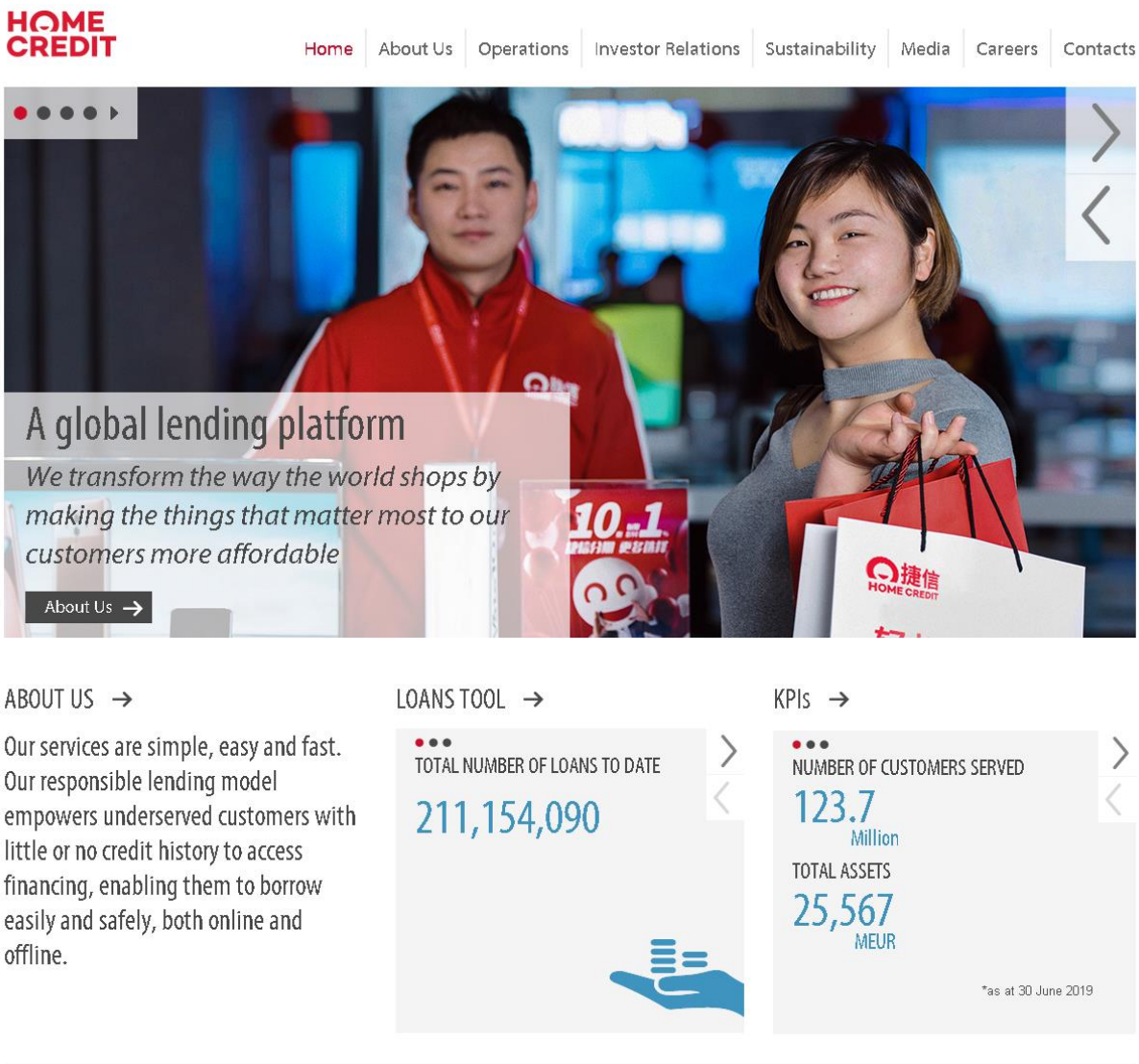
Mise en situation

Vous êtes *Consultant Data* au sein d'une société financière, nommée "**home Credit**", qui propose des *crédits* pour des personnes ayant peu ou pas du tout d'historique de prêt



Fraîchement embauché depuis une semaine avec ce salaire annuel, vous avez fait connaissance avec vos collègues et votre nouveau bureau. Mais revenons à vos missions : il est temps de mettre les mains dans le cambouis ! Le **DSI** vous a donné l'accès à la base de données. L'entreprise souhaite **développer un modèle de credit scoring** et de le **mettre en production**. Les données à disposition sont variées : *données comportementales*, *données provenant d'autres institutions financières*, etc. (**à vous de vous familiariser avec cette data** !!!).

Mission 1

Créer la base de données de la Banque

- MCD, tables, jointures, PK, FK, connexion sécurisée, ..., backup (création dans l'art en respectant les compétences **RNCP** à valider, voir ci-dessous).

Mission 2

Créer le modèle de scoring et le mettre en production

- Construire un modèle de scoring et le déployer via une `API` sur le Web où en saisissant l'identifiant client (ou num de dossier de crédit), l'API renvoie bien la prédiction correspondante.

- Pensez à utiliser un outil gratuit et disponible plusieurs mois *en vue de vos entretiens techniques*, *jury*,…. par exemple Heroku (netlify : alternative gratuite) où vous déployez `Flask` ou `Django` : help1 et help2 ; en passant par un fichier pickle contenant votre modèle sérialisé.

Le focus pour le modèle de machine learning sera mis sur :

- La conception du modèle, son évaluation et son interprétation compréhensible pour les métiers

- La systématisation de la création de ***features***, via des *jointures, groupby, LabelEncoder, OneHotEncoder,…* ou via la combinaison de features (rapport de 2 features, notamment montants, ...)

- Dans le cadre de l'optimisation du modèle, penser à utiliser **SMOTE** (génération de lignes pour ré-équilibrer le nombre de valeur cible à 1 par rapport à 0) et **Hyperopt** (optimisation des hyperparamètres).

- N'oublier pas de mettre en œuvre une **matrice de coût** adaptée au contexte de crédit afin de proposer une optimisation orientée métier et non pas technique :

    - Par exemple : le coût d'un *faux positif* (bon crédit considéré comme mauvais constitue un manque à gagner modéré pour la banque, une perte de marge) est différent d'un faux négatif (mauvais crédit considéré comme bon = constitue une perte importante pour la banque, un défaut de paiement et/ou une perte de capital non remboursé). Idéalement, vous montrez que l'optimum « métier » est différent de l'optimum du **fscore** ou autres mesures purement « techniques ».

Mission 3 : Créer un dashbord interactif

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 va tout à fait dans le sens des valeurs que l'entreprise veut incarner. Votre **manager** décide donc de **développer un dashboard interactif** pour que les conseillers puissent expliquer de façon la plus transparente possible les décisions

Cahier des charges rédigé par le manager pour le dashboard :

Les spécifications du Dashboard devront a minima contenir les fonctionnalités suivantes :

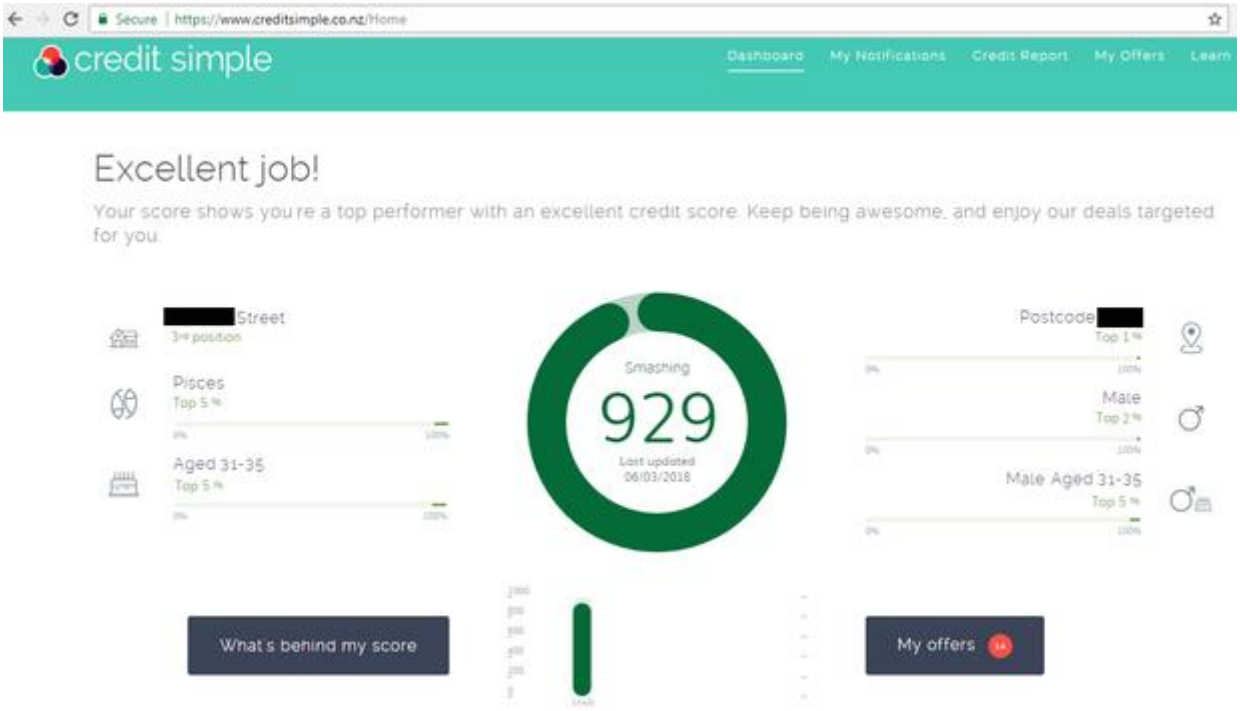
- *Permettre de visualiser des informations descriptives relatives à un client (via un système de filtre).*
- *Permettre de visualiser le score et l'interprétation de ce score pour chaque client de façon intelligible pour une personne non experte en data science.*
- *Permettre de comparer les informations descriptives relatives à un client à l'ensemble des clients ou à un groupe de clients similaires.*

Le focus sera mis :

- Le dashboard est accessible pour d'autres utilisateurs sur leurs postes de travail (déploiement dans le web)

- Les graphiques réalisés sont pertinents : ils permettent de répondre à la problématique métier

- Vous réalisez au moins deux graphiques interactifs permettant aux utilisateurs d'explorer les données clients



Livrables attendus

1. Une organisation DevOps Azure qui contient :
   - Un **board** où vous répartissiez les tâches entre les membres de votre groupe suivant votre méthode de travail `Agile`, `Scrum`
   - Les **pipelines** pour le CI/CD
2. Un `repo Github` contenant :
   - Un fichier README (où vous expliquez comment lancer les scripts, …),
   - Un notebook Python (non cleané, pour comprendre votre démarche :
     - Les problèmes rencontrés sur le jeu de données

        - Comment vous avez nettoyé les données
        - Votre modélisation (du preprocessing à la prédiction).
    - Le code générant le dashboard et permettant de déployer le modèle sous forme d'API

- L'URL de la WebApp mise en ligne et répondant au cahier des charges précisé ci-dessus.
    - Un support de présentation (environ 10 slides) :
        - De La démarche de modélisation et la méthodologie d'entraînement du modèle
        - De La fonction **coût**, l'algorithme d'optimisation et la métrique d'évaluation
        - L'interprétabilité du modèle est explicitée (1 page max). N'oubliez pas : la façon d'interpréter l'importance des variables n'est pas la même pour une régression logistique que pour un random forest (par exemple). Préciser les limites éventuelles ?
        - Les limites et les améliorations possibles pour gagner en performance et en interprétabilité (1 page max)

Modalités de présentation du travail

Votre présentation pourra prendre cette forme :

5 min.	Présentation de la problématique, de l'exploration des données, du cleaning effectué, du feature engineering
10 min	Présentation des différentes pistes de modélisation effectuées
10 min	Présentation du dashboard
10 min	Séance de questions-réponses

Ressources complémentaires

- Un article donnant quelques bonnes pratiques pour le design de dashboard.
- Un document décrivant les bonnes pratiques pour réaliser des graphiques clairs et pertinents.

- Des informations sur deux librairies permettant de construire des dashboards interactifs en Python : Dash et Bokeh.

- Ce lien, et celui-ci pour aider à construire l'interprétabilité du modèle.

Compétences à valider

- Déployer un modèle de Machine Learning via une API dans le Web
- Réaliser un dashboard pour présenter son travail
- Rédiger une note méthodologique afin de communiquer sa démarche de modélisation
- Utiliser un logiciel de version de code pour assurer l'intégration du modèle
- Présenter son travail de modélisation à l'oral