****Openclassrooms**

**Formation Data Science**

**Procédure de la mise en place d’un modèle de scoring ainsi que d’un DASHBORD.**

**Projet 07 :**

**Implémenter un modèle de scoring**

**Rédigé par : KARIM YAHIATENE**

**Fait le 01/07/2021**

# Introduction et contexte du projet :

Dans le cadre de la formation de data science, ce projet est réalisé afin de mettre en pratique les compétences acquises dans le tri des données et leurs exploitation, la mise en place de modèle de scoring ainsi que la création d’un Dashbord pour l’interaction client banquier.

L’entreprise souhaite développer un modèle de scoring de la probabilité de défaut de paiement du client pour étayer la décision d'accorder ou non un prêt à un client potentiel en s’appuyant sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.) .

La mission principale de ce projet est de prédire le risque de faillite d'un client pour une société de crédit. Pour cela, nous devons configurer un modèle de classification binaire et d'en analyser les différentes métriques.

Ce projet consiste à créer une API web avec un Dashboard interactif. Celui-ci devra à minima contenir les fonctionnalités suivantes :

- 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 visualiser des informations descriptives relatives à un client (via un système de filtre).

- Permettre de comparer les informations descriptives relatives à un client à l’ensemble des clients ou à un groupe de clients similaires.

Les données peuvent être téléchargées à partir du lien suivant: https://www.kaggle.com/c/home-credit-default-risk/data

# La méthodologie d'entraînement du modèle

La méthodologie suivi lors de ce projets est de commencer par appréhender les données afin de voir ce qu'on a comme entrée pour le travailles à effectuer. Pour commencer j'ai fait un constat des différents fichiers csv, voir leurs contenu ainsi qu'une clé afin de les lier les uns par rapport aux autre.

* Une étude univariée et mulltivariée a été réalisée afin de quantifier les features.
* Suppression des outliers afin d'enlever les valeurs aberrantes et statistiquement trop grande pour l’étape de modélisation
* Feature engineering en faisant des groupements selon des critères spécifiques
* One hot encooding appliquée sur les features qualitatifs
* Un rééquilibrage des données a été appliqué pour avoir un jeu de données équivalent en termes de refus et d'acceptation de crédit.
* Mise à l'échelle des données
* Réduction de dimension en utilisant la PCA
* Comparaison entre diffèrent modèles [dummy classifiers, XGBoust, RandomForest, Regression logictique]
* Optimisation et amélioration des modèles.

# La fonction coût, l'algorithme d'optimisation et la métrique d'évaluation

Nous allons mesurer différentes métriques pour bien comprendre les classifieurs :

AUC : signifie "aire sous la courbe ROC". Cette valeur mesure l'intégralité de l'aire à deux dimensions située sous l'ensemble de la courbe ROC (par calculs d'intégrales) de (0,0) à (1,1). Elle mesure l'aptitude du modèle à prédire un score plus élevé pour les exemples positifs par rapport aux exemples négatifs. Comme la métrique AUC est indépendante du seuil sélectionné, elle nous permet de nous faire une idée des performances de prédictions de notre modèle, sans choisir de seuil.

Accurancy: mesure la fraction de prédictions correctes. Le taux de positifs prédits mesure la fraction de positifs observés parmi les exemples prédits comme positifs.

f1: présente la moyenne harmonique entre le taux de positifs prédits et la sensibilité. Cette dernière mesure le nombre de positifs observés qui ont été prédits comme positifs

La précision : quelle proportion d'identifications positives était effectivement correcte.

Le rappel : quelle proportion de résultats positifs réels a été identifiée correctement.

J'ai utilisé une gridSearch pour optimiser mon modèle.

# Modèle de classification

Le modèle retenu pour cet exercice est le modèle XGBoost. Pour plusieurs raisons, parmi elles:

-Modèle moins gourmand en termes de temps de calcul

-Le plus performant d'après les scores

-Facilement optimisable

# L’interopérabilité du modèle

Les résultats obtenu avec ce modèle quantifier avec les metrics suivants :

GradientBoostingClassifier amélioré: f1=0.595 auc=0.647

GradientBoostingClassifier amélioré: Accurancy=0.618

GradientBoostingClassifier amélioré: precision=0.630 rappel=0.563

# Les limites et les améliorations possibles

Le modèle peut probablement être amélioré, pour raison de certaines contraintes imposées par le Dashbord, j'ai dû limiter le nombre d'inputs a renseigner pour l'accord du prêt, le modèle peut sans doute être amélioré avec plus de features.

# Les livrables

Les livrables attendus doivent contenir :

1-Le dashboard interactif répondant au cahier des charges précisé.

2-Un dossier sur un outil de versioning de code contenant :

\*Le code de la modélisation (du prétraitement à la prédiction)

\*Le code générant le dashboard

\*Le code permettant de déployer le modèle sous forme d'API

# Conclusion

Le dataset est pertinent pour répondre à la problématique car il possède d’assez de données pour faire une étude qualitative et quantitatives.

Il a fallu faire un rééquilibrage des données car dispose d’une portion de données avec une acceptation de crédit à 92% tandis que le taux de refus été de 8%.

Les méthodes d’analyse de composantes principales fonctionnent bien pour la simplification de la problématique, et permets d’avoir une visualisation 2D/3D efficace pour le clustering.

Plusieurs modèles ont été testés, celui qu’on a retenu est le XGBoost.

Plusieurs métriques ont été utilisées pour la qualification des modèles.

Une gridSearch a été utilisée pour optimiser le modèle sélectionné, ce qui a permis de donner un modèle plus efficace pour la classification