

**Documentation**

**Challenge :** L'automatisation des processus de crédit bancaire à l'aide de l'IA

**Proposer Par :** PROXYM

**Réaliser Par :** EPI\_GRINTA

EDITION 2023

**Sommaire**

Description de challenge ……………………………………. 3

Description de DataSet…………………………………………4

Description de model……………………………………………5

Le fonctionnement de prototype………………………….6

**Description de challenge**

L'automatisation des processus de crédit bancaire à l'aide de l'IA : comment les banques peuvent utiliser l'IA pour analyser les données financières des clients, automatiser le processus de décision de crédit et réduire les délais de traitement.

**Description de DataSet**

Notre dataset est nommée german\_credit\_data .

elle est de forme CSV.Elle est importe a partir de lien suivant : https://www.kaggle.com/datasets/uciml/german-credit

Cette dataset décrit les clients des banques allemandes qui demandent des crédits . on a 9 caractéristiques présentes dans notre base, ils sont comme suit :

|  |  |
| --- | --- |
| Nom | Type |
| Age | int |
| Sex | String |
| Job | int |
| Housing | String |
| Saving\_accounts | String |
| Checking\_accounts | String |
| Credit\_amount | int |
| Duration(en mois) | Int |
| Purpose | String |

Les caractéristiques suivants on peut classifie ses choix selon des catégories :

- Sex : female , male

- Job : 0,1,2,3

- Housing : own ,free , rent

- Saving\_accounts : NA , little , quite rich , rich , moderate

- Checking\_account : NA , little , quite rich , rich , moderate

- Purpose : radio/TV , education , furniture/equipment , car , business , domestic appliances , repairs , vacation/others

Les dimensions sont : (1000,11)

**Description de model**

Pour réaliser notre model, on a divisé notre dataset avec un taux de 25% entre train et test. Puis on a utilisé l’algorithme de classification DecisionTreeClassifier .

Le model va repondre par ‘Good ’ si la situation est favorable sinon il va repondre par ‘Bad’

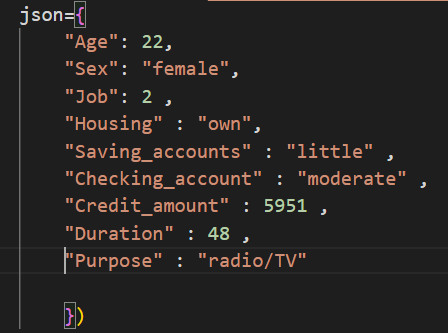
on a évaluer notre model a travers les métriques et on a eu les résultats suivants :

* Accuracy = 0.696
* Recall = 0.778
* Precision = 0.787

Puis on a exporté notre model à travers la bibliothèque Joblib sous le nom (’’ model.pkl ’’)

**Le fonctionnement de prototype**

Apres avoir exporte le model, on a réalisé une application flask(python) et on a intégré notre model afin d’avoir un API fonctionnel qui prend comme input de type Json avec 9 données comme l’exemple suivant :



**Figure 1 :**exemple requête POST avec data type json

pour consommer l’api , l’url dédiée est ‘’server/port/predict ‘’. Elle est de type post.

L’api va répondre par ‘Good’ si le client a des bonnes chances d’avoir un crédit sinon il va répondre avec ‘Bad’