


Projet Streamlit : Prédiction du Risque de Crédit avec Machine Learning

 **Titre du projet : Détection du Risque de Crédit avec Machine Learning**

 **Objectif :** Construire un modèle de machine learning pour prédire si un client peut être considéré comme risqué ou non en fonction de ses informations financières et personnelles.

Contexte

Les institutions financières, comme les banques et les organismes de crédit, doivent évaluer le risque associé à l'octroi d'un prêt à un client. Un mauvais choix peut entraîner des pertes importantes, tandis qu'un bon choix permet d'optimiser la rentabilité.

L'intelligence artificielle et le machine learning offrent des solutions avancées pour analyser les données des clients et prédire leur probabilité de remboursement ou de défaut.

Dans ce projet, nous allons exploiter un jeu de données de **crédit bancaire** provenant de Kaggle afin de construire un modèle capable d'évaluer si un client est susceptible d'avoir des difficultés à rembourser son crédit.

Étapes du Projet

1- Collecte et Préparation des Données

- Télécharger un jeu de données de crédit bancaire depuis **Kaggle**.
- Nettoyer les données : gestion des valeurs manquantes, transformation des variables catégorielles en variables numériques, normalisation des données.
- Division des données en **ensemble d'entraînement (train)** et **ensemble de test (test)**.

2- Analyse Exploratoire des Données (EDA) et Visualisation

- Étudier la distribution des variables (âge, revenus, historique de crédit, etc.).
- Analyser les corrélations entre les variables et la variable cible (**client risqué ou non**).
- Visualiser les tendances à l'aide de **matplotlib** et **seaborn** (histogrammes, heatmaps, boxplots, etc.).

3- Construction de Modèles de Machine Learning

- Entraîner un **modèle de régression logistique** et un **arbre de décision** pour classifier les clients en fonction du risque.
- Comparer la performance des modèles avec des métriques telles que :
 - **Accuracy** (Précision)
 - **Recall** (Rappel)
 - **F1-score**
 - **AUC-ROC**

4- Choix du Meilleur Modèle

- Comparer les performances des modèles.
- Sélectionner le modèle le plus performant pour l'application

5- Déploiement en Local avec Streamlit

- Créer une **application interactive avec Streamlit**.
- Permettre aux utilisateurs d'entrer les caractéristiques d'un client et d'obtenir une **prédiction en temps réel**.
- Afficher des **visualisations dynamiques** des données.

6- Déploiement sur Streamlit Cloud

- Préparer un fichier **requirements.txt** pour gérer les dépendances.
- Héberger l'application sur **Streamlit Cloud** pour qu'elle soit accessible en ligne.

Livrables

Code source et fichiers :

- app.py → Script principal de l'application Streamlit.
- model.joblib → Modèle de machine learning sauvegardé.
- dataset.csv → Jeu de données utilisé.
- notebook.ipynb → Notebook Jupyter contenant l'analyse exploratoire et la construction du modèle.
 - Commenter les résultats (tableaux, graphiques)
 - Commenter les codes aussi
- images/ → Dossier contenant les visualisations générées.
- requirements.txt → Liste des dépendances nécessaires au projet.

Lien de déploiement :

 [Lien vers l'application Streamlit Cloud](#)