**Documentation : Application de Prédiction d'Acceptation d'Offre de Dépôt à Terme**

**Contexte**

Cette application a été développée pour prédire si un client acceptera une offre de dépôt à terme. Elle repose sur un modèle de machine learning entraîné à partir des données d’une campagne marketing bancaire. Le modèle vise à améliorer l'efficacité des campagnes en ciblant les clients les plus susceptibles d'accepter l'offre.

**Fichiers impliqués**

1. **Modèle ML : bank\_marketing\_model.pkl**
   * Contient un modèle de machine learning entraîné, basé sur une **régression logistique**.
   * Ce choix est justifié par la simplicité, l'interprétabilité et l'efficacité de ce modèle dans les problématiques de **classification binaire**, particulièrement pertinentes pour les campagnes marketing.
2. **Script Streamlit : deployment\_bank\_model.py**
   * Permet de déployer une interface interactive pour tester le modèle de manière intuitive.
3. **Dataset original : bank-additional-full.csv**
   * Jeu de données utilisé pour entraîner le modèle.
   * Il contient des informations sur les clients, leurs interactions avec la banque, et des indicateurs économiques.

**Préparation des Données**

**Variables Clés**

Le dataset contient les catégories suivantes :

* **Variables personnelles du client** :
  + age, job, marital, education
* **Informations financières** :
  + default (défaut de crédit), housing (prêt immobilier), loan (prêt personnel)
* **Détails de la campagne marketing** :
  + contact, month, day\_of\_week, duration, campaign, pdays, previous, poutcome
* **Indicateurs macroéconomiques** :
  + emp.var.rate (taux de variation de l'emploi), cons.price.idx (indice des prix à la consommation), cons.conf.idx (indice de confiance des consommateurs), euribor3m, nr.employed (nombre d'employés)

**Variable Cible**

* **y** : Indique si le client a accepté l'offre :
  + **yes (1)** : Offre acceptée
  + **no (0)** : Offre refusée

**Prétraitement des Données**

* **Encodage des variables catégoriques** : Pour convertir les variables non numériques en une représentation utilisable par le modèle.
* **Normalisation des données numériques** : Pour aligner les échelles et améliorer la performance du modèle.
* **Gestion des valeurs manquantes** :
  + Imputation des valeurs fréquentes pour les variables catégoriques.
  + Remplacement des valeurs manquantes par la médiane pour les données numériques.

**Fonctionnement de l'Application Streamlit**

**1. Chargement du Modèle**

* Le fichier bank\_marketing\_model.pkl est chargé à l’aide de la bibliothèque joblib.
* Si le fichier est introuvable, l'application affiche un message d'erreur et interrompt son exécution pour éviter des prédictions incorrectes.

**2. Saisie des Données**

* Une **barre latérale interactive** permet de saisir les caractéristiques du client via des champs :
  + Menus déroulants pour les valeurs catégoriques (par exemple, job, marital).
  + Champs numériques pour les données continues (age, duration, etc.).
* Chaque champ est accompagné de **valeurs par défaut** pour simplifier l'expérience utilisateur.

**3. Prédiction**

* Une fois les données saisies, l’utilisateur clique sur **Prédire**.
* Le modèle renvoie :
  + Une **interprétation en pourcentage** de la probabilité que le client accepte ou refuse l'offre.
  + Une présentation claire du résultat :
    - Par exemple : "Le client est susceptible d'accepter l'offre à 85 %."
    - Ou : "Le client est susceptible de refuser l'offre à 90 %."

**Utilisation de l’Application**

**1. Pré-requis**

* Installer **Streamlit** :

pip install streamlit

* Placer les fichiers suivants dans un **même dossier** :
  + deployment\_bank\_model.py
  + bank\_marketing\_model.pkl

**2. Exécution**

* Lancer l’application via la commande :

streamlit run deployment\_bank\_model.py

**3. Interface Utilisateur**

* Renseigner les informations utilisateur via la **barre latérale**.
* Cliquer sur **Prédire** pour afficher les résultats.

**Conclusion**

Cette application offre une méthode simple et intuitive pour exploiter un modèle de machine learning dans le contexte des campagnes marketing bancaires.  
Elle peut être améliorée avec :

* L’intégration de modèles supplémentaires pour comparer les performances.
* Des visualisations des performances du modèle (courbe ROC, matrice de confusion) directement dans l'interface Streamlit.
* Une option d’export des prédictions pour une analyse plus approfondie.

Avec ces améliorations, cette application pourrait devenir un outil précieux pour les équipes marketing désireuses de maximiser l'efficacité de leurs campagnes.