

Documentation Générale et Technique - POC ML Carrefour

1. Contexte et Objectif

Ce projet est un POC (Proof of Concept) pour Carrefour Market. L'objectif est de démontrer de manière simple et interactive comment le Machine Learning peut être utilisé pour répondre à trois besoins métiers stratégiques :

- Anticiper les comportements clients (via un modèle de classification).
- Optimiser la logistique (via un modèle de prévision de la demande).
- Recommander des produits (via un système de recommandation basé sur la similarité entre clients).

2. Données utilisées

Les données utilisées dans ce projet sont simulées afin de représenter des cas réalistes. Elles sont générées automatiquement à chaque exécution du code pour chaque section :

- **Comportement client** : données de profil client (âge, revenu, fréquence d'achats, panier moyen).
- **Prévision logistique** : données de demande journalière simulées sur une année.
- **Recommandation produit** : matrice d'achats client-produit (binaire).

3. Technologies utilisées

- **Streamlit** : pour créer l'interface interactive et l'exécution du POC dans un navigateur web.
- **Pandas / NumPy** : pour la manipulation de données tabulaires.
- **Scikit-learn** : pour les modèles de Machine Learning (classification, régression, recommandation).
- **Matplotlib** : pour la visualisation graphique simple.

4. Documentation Technique

4.1 Structure du projet

- **`app.py`** : fichier principal contenant tout le code de l'application avec trois sections principales.
- **`logo.png`** : logo utilisé dans la barre latérale de navigation.
- **`requirements.txt`** : liste des bibliothèques nécessaires pour exécuter le projet.

4.2 Détail par section

◆ Navigation:

Utilise `st.sidebar.radio()` pour basculer entre les différentes pages (Accueil, Comportement client, Logistique, Recommandation). La navigation est stylisée avec CSS pour appliquer les couleurs Carrefour (bleu et rouge).

◇ Section 1 - Classification comportement client:

- Génère un dataset client simulé (âge, revenu, etc.).
- Applique un modèle `RandomForestClassifier` pour prédire si un client achètera un produit.
- Affiche la prédiction et la probabilité à l'utilisateur selon les entrées choisies.

◇ Section 2 - Prévision logistique:

- Génère une série temporelle simulée de demande journalière.
- Utilise un modèle `LinearRegression` pour faire des prédictions simples.
- Visualise la courbe de demande et permet de prédire une date future.

◇ Section 3 - Recommandation produit:

- Crée une matrice client-produit binaire aléatoire.
- Applique `NearestNeighbors` avec une distance cosinus pour trouver des clients similaires.
- Suggère des produits non achetés par le client mais achetés par ses voisins proches.

4.3 Personnalisation graphique (UI)

L'apparence de l'application a été adaptée avec CSS :

- Barre latérale colorée avec dégradé bleu/rouge (codes couleur de Carrefour).
- Boutons stylisés avec surbrillance au survol ou sélection.
- Ajout du logo de Carrefour via `st.sidebar.image()` et insertion HTML.

5. Auteur

Réalisé par Dyhia BERKHOUCHE.