Prototype pour un système de livraison à la demande

## URL du dépôt du projet

https://github.com/leonard0371/projetlivraisondemande

## Membres de l’équipe

* Léonard Benamara – 20246822
* Aiman Rahmani - 20196280

## Partenaires ou collaborateurs

Aucun partenaire externe actuellement. Si une entreprise de livraison accepte de collaborer en partageant des données, nous en informerons le responsable et le superviseur.

## Titre du projet

Système de livraison à la demande

## 1. Contexte

Les systèmes de livraison actuels sont souvent basés sur des données fixes et des itinéraires définis après réception des commandes. Cela provoque des délais supplémentaires et une inefficacité logistique, surtout en zones urbaines comme Montréal. Notre projet vise à concevoir un prototype intelligent capable de prédire les commandes et de préparer un stock mobile, afin de livrer rapidement et efficacement en temps réel. L’objectif est de réduire les coûts logistiques et les délais de livraison en misant sur la prédiction et la réactivité.

## 2. Problématique et motivation

Les systèmes classiques ont de la difficulté à répondre à la demande immédiate. Ils manquent de flexibilité et de capacité d’anticipation.  
Notre prototype a pour objectifs :

• D’anticiper les commandes journalières via un modèle prédictif.  
• De préparer un stock mobile chargé à l’avance.  
• D’optimiser l’itinéraire des livreurs et leur positionnement pour exécuter rapidement les commandes potentielles.  
• D’assurer des livraisons avant la péremption des produits (je travaille avec des produits périssables).

## 3. Objectifs

1. Prédire dynamiquement les commandes à venir.  
2. Préparer un stock mobile fictif à charger chaque jour en fonction des prédictions.  
3. Définir un itinéraire optimal pour les livreurs basé sur ces prédictions. (si j’ai du temps)  
4. Générer et afficher les points stratégiques où les livreurs doivent se positionner pour répondre efficacement à la demande (si le temps le permet).

## 4. Méthodologie

1. Rôle utilisateur : Le gestionnaire de livraisons de la compagnie de vente (marchand) qui simule les opérations à partir d’une interface.  
2. Génération de données (via Faker Python Package) :  
 - Types de produits périssables.  
 - Adresses client dans le Grand Montréal.  
 - Fréquence d’achat par secteur.  
 - Durée de vie des produits.  
 - Chronologie des commandes.  
 - (Trafic routier si le temps le permet).

3. Interface React :

* Déclenche manuellement la prédiction des produits à stocker pour la journée.
* Affichage des produits prédits avec leur zone prédite correspondante.
* Affiche les zones de livraison et les points stratégiques sur carte.
* Section de validation de la prédiction en comparant les données prédites avec les données du jeu de données (données réelles)

4. Modèle prédictif (Python/ML) :  
 - Entraîné sur l’historique de commandes.  
 - Mis à jour à chaque nouvelle entrée.  
 - Affiche la liste des produits à stocker selon la demande anticipée et l’affiche au front-end.  
5. Optimisation de la position des livreurs :  
 - Utilisation de K-Means ou DBSCAN pour localiser les points stratégiques (si le temps le permet). Affichage sur carte via Google Maps API  
 - S’inspirer du problème ***Vehicle Routing Problem*** et du ***problème des journaux***.  
6. Évaluation :  
 - Comparaison entre les commandes prédictives et les commandes réelles.  
 - Mesure du temps gagné et de la précision du système.

## 5. Hypothèses simplificatrices

• Une seule flotte de véhicules fictifs partant d’un point de collecte unique.  
• Le système ne tient pas en compte les commandes enregistrées la veille comme dans un système de flotte normale. (Post-commande)  
• Le déclenchement de la prédiction se fait à minuit chaque jour avant le début des livraisons.

## 6. Technologies utilisées

• Frontend : React.js  
• Backend : Python

* Github CI/CD
* Azur static web apps (React)
* Azur functions
* Données simulées : Faker
* Machine Learning : Scikit-learn, TensorFlow
* Cartographie : Google Maps JavaScript API