Rapport de Projet

Smart Traffic Prediction (Casablanca)

Cloud Computing - Intelligence Artificielle

Réalisé par :
Fourari2004

Date :
4 février 2026

Casablanca, Maroc

Table des matières

Chapitre 1

Introduction

Ce projet vise à résoudre le problème de la congestion routière à Casablanca en utilisant l'**Intelligence Artificielle** et le **Cloud Computing**. L'objectif principal est de fournir une estimation en temps réel de l'état du trafic (Fluide, Modéré, Bloqué) basée sur des données historiques telles que la zone géographique, l'heure de la journée et le jour de la semaine.

Chapitre 2

Architecture Technique (Azure Cloud)

La solution repose sur une architecture **PaaS (Platform as a Service)** afin d'optimiser les coûts opérationnels et la maintenance.

2.1 Stockage de Données

— **Azure Blob Storage :** Ce service est utilisé pour stocker les fichiers bruts (notamment `dataset.csv`) qui servent à l'entraînement du modèle d'IA.
— **Azure SQL Database (Optionnel) :** Utilisé pour stocker l'historique des prédictions et assurer la gouvernance et la sécurité des données structurées.

2.2 Intelligence Artificielle

— **Azure Machine Learning (Designer) :** Nous avons utilisé le concepteur visuel pour créer un pipeline de **Régression Linéaire**. Ce pipeline gère l'ensemble du flux : ingestion des données, nettoyage, entraînement du modèle et évaluation des performances.

2.3 Déploiement & API

— **Azure Container Instance (ACI) :** Le modèle, une fois entraîné, est déployé sous forme de Web Service (API REST). Ce service est sécurisé par une clé d'authentification pour contrôler l'accès.

2.4 Interface Utilisateur

— **Application Client (Python/FastAPI) :** Une interface locale moderne, développée en Python avec le framework FastAPI, communique avec l'API Azure pour récupérer les prédictions et afficher les résultats en temps réel à l'utilisateur.

Chapitre 3

Méthodologie de Réalisation

3.1 Préparation des Données

La première étape a consisté en la collecte et la structuration des données de trafic. Les données (Zone, Heure, Jour, Niveau de Traffic) ont été consolidées dans un fichier CSV.

3.2 Entraînement (Training)

Nous avons utilisé **Azure ML Designer** pour entraîner un modèle de régression. Pour valider la précision du modèle, nous avons divisé les données en deux ensembles :

— 70% des données pour l'entraînement.
— 30% des données pour le test et la validation.

3.3 Déploiement (Deployment)

La mise en production du modèle s'est faite via un **"Real-time Inference Endpoint"**, permettant des appels API synchrones pour des prédictions immédiates.

3.4 Développement App

Nous avons créé un script Python intégrant une logique de **secours (Fallback)**. Cette fonctionnalité est cruciale pour une démonstration : elle garantit que l'application continue de fonctionner et de fournir des estimations (basées sur une simulation locale) même en cas de latence réseau ou d'indisponibilité temporaire du service Cloud Azure.

Chapitre 4

Optimisation des Coûts (Cost Management)

Le projet a été conçu pour respecter une contrainte budgétaire stricte (inférieure à 20$ par mois). Plusieurs stratégies ont été mises en œuvre :

- **Auto-shutdown :** Configuration de l'extinction automatique des clusters de calcul après 15 minutes d'inactivité.
- **Niveaux de service (Tiers) :** Choix des niveaux basiques (Basic Tier pour SQL, LRS - Locally Redundant Storage pour le stockage).
- **Azure Container Instance :** Utilisation de ce service qui facture à la seconde d'utilisation, idéal pour des besoins ponctuels ou de démonstration.

Chapitre 5

Procédure de Réentraînement (Mise à jour du Modèle)

Afin de maintenir la précision du modèle avec de nouvelles données (comme le dataset enrichi `traffic_data.csv` contenant Maarif, Sidi Maarouf, etc.), une procédure de réentraînement simple a été mise en place :

1. **Mise à jour des Données (Upload) :**
   — Accéder à l'Azure Portal → Storage Account → Containers → `datasets`.
   — Téléverser le nouveau fichier `traffic_data.csv` en sélectionnant l'option "Overwrite" pour remplacer l'ancien fichier.
2. **Relance de l'Entraînement (Retrain) :**
   — Dans Azure ML Studio (Designer), ouvrir le Pipeline existant.
   — Cliquer sur **"Submit"** en réutilisant l'Expérience existante (`traffic-exp-1`).
   — Azure va automatiquement ingérer les nouvelles données et ajuster le modèle (ex : apprendre que "Ain Diab" est saturé le vendredi soir).
3. **Mise à Jour du Service (Update Endpoint) :**
   — Une fois le Run terminé avec succès, déployer la nouvelle version sur l'Endpoint existant (`traffic-app`) pour que l'API utilise immédiatement le nouveau "cerveau".

Chapitre 6

Conclusion

Ce projet démontre la puissance et la flexibilité du Cloud Azure pour déployer rapidement des solutions d'Intelligence Artificielle évolutives et robustes. Il illustre comment, en quelques heures, il est possible de passer de la donnée brute à une application utilisateur fonctionnelle.