Cahier des charges

# **Projet ML Ops** : Prédiction du temps de réponse des pompiers à Londres

## Contexte et Objectifs

La London Fire Brigade (LFB) est le plus grand corps de sapeurs-pompiers dans le monde avec 103 casernes et plus de 5000 sapeurs-pompiers professionnels. La Brigade couvre les 13 circonscriptions de Londres qui est composé de 8 millions d’habitants. La Brigade traite entre 100000 et 130000 appels d’urgences chaque année et intervient sur un territoire de 1587m².

En cas d’urgence chaque minute est comptée, il est donc primordial que les pompiers interviennent le plus rapidement possible sur le site. Dans ce contexte, il serait intéressant pour la brigade, et notamment pour le centre d’appels de pouvoir anticiper et prédire le temps d’intervention des pompiers suite à un appel.

Problématique : Lors d’un appel d’urgence, pouvoir prédire en temps réel avec quelques informations récoltées par téléphone le temps que mettra le premier camion de pompiers pour arriver sur les lieux.

Commanditaire : La direction des services opérationnels de la London Fire Brigade

Utilisateur : l’équipe de régulation en charge de la réception des appels (Une centaine de personnes se relaient pour la réception des appels d'urgence)

Administrateur :

Dans quel contexte l’application devra-t-elle s’intégrer :

Via quel support l’application sera-t-elle utilisée :

## Modèle

Il s’agit d’un modèle de régression dont l’objectif est de prédire au plus proche de la réalité le temps d’intervention des pompiers suite à un appel d’urgence.

Features : (afficher quelques dataviz ?)

Variable cible :

Algorithme utilisé :

Performance du modèle :

Métriques d’évaluations :

## Base de données :

Les données utilisées :

* Données sur les incidents**:** <https://data.london.gov.uk/dataset/london-fire-brigade-incident-records>
* Données sur les forces de la Brigade qui ont été mobilisés suite à l’incident : <https://data.london.gov.uk/dataset/london-fire-brigade-mobilisation-records>

Outil de gestion de base de données : MySQL

Architecture de la base de données : (ajouter des images)

Gestion des nouvelles données : Chaque nouvelle donnée passera par un pipeline ETL avant d’être chargée dans la base de données, ceci afin de garder une architecture propre.

## API

Cette API assure le lien entre la base de données, le modèle entrainé et l’utilisateur.

Outil utilisé : FastAPI

Liste des fonctionnalités de l’API :

* Authentification
  + Users
  + Admin
* Model
  + Results
  + Accuracy (ou autre métrique)
* Base de données
  + Get\_data : récupérer certaines données
  + Put\_data : ajouter de nouvelles données (admin uniquement)

## Testing & Monitoring

Liste des tests unitaires à mettre en œuvre :

* Bon fonctionnement du modèle lors de l’entraînement
* Bon fonctionnement du modèle lors de la prédiction
* Bon fonctionnement des endpoints de l’API
* Bon fonctionnement du process d’ingestion de nouvelles données

Monitoring :

La LFB fournit chaque mois les nouvelles données concernant les interventions du mois précédent.

Après chaque nouvelle entrée mensuelle, le modèle devra s’évaluer pour vérifier les métriques. Si l’accuracy du modèle n’atteint pas le seuil de **80%** sur les dernières données, alors on procèdera à un nouvel entrainement du modèle sur les données des 3 dernières années.

Après une nouvelle évaluation du modèle entrainé sur les 3 dernières années, si le seuil de précision souhaité (**80%**) n’est toujours pas atteint, alors il y a deux solutions :

* Si l’accuracy est comprise entre 70% et 80% : L’outil reste en production mais une alerte mail devra être envoyée aux administrateurs pour trouver des pistes d’amélioration du modèle.
* Si l’accuracy est strictement inférieure à 70% : L’application devra être bloquée.