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 vitale, toute minute perdue amoindrit les chances de survie ou augmente le risque de séquelles. Côté incendie, un fameux dicton le dit : la première minute, un feu s’éteint avec un verre d’eau, la deuxième avec un seau, et la troisième avec une citerne ! 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. Cela permettrait notamment de rassurer les personnes appelant les secours, de mieux optimiser la logistique des forces de secours ou de challenger l’organisation du départ et les trajets pour réduire encore le temps de prise en charge de l’urgence.

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 lle temps d’intervention des pompiers suite à un appel d’urgence.

Features :

* IncidentGroup object
* StopCodeDescription object
* PropertyCategory object
* IncGeo\_BoroughCode object
* IncGeo\_WardCode object
* Easting\_rounded int64
* Northing\_rounded int64
* IncidentStationGround object
* FirstPumpArriving\_AttendanceTime float64
* FirstPumpArriving\_DeployedFromStation object
* NumStationsWithPumpsAttending int32
* NumPumpsAttending int32
* PumpMinutesRounded int64
* NumCalls int32
* PumpOrder int64
* DelayCodeId int32
* Year int32
* DayOfWeek int32
* Hour int32

Variable cible :

FirstPumpArriving\_AttendanceTime (temps en secondes entre le moment de l’appel d’urgence et le moment d’arrivée des premiers secours sur les lieux)

Algorithme utilisé :

LightGBM Regressor (Gradient Boosting)

Performance du modèle :

* Mean Squared Error (MSE): 7452.791247584479
* Mean Absolute Error (MAE): 60.75463757169317
* R-squared (R²): 0.6151402501948497
* Root Mean Squared Error (RMSE): 86.32955025704975

Métriques d’évaluations :

Nous choisissons d’observer la RMSE pour évaluer la performance de notre modèle, car celle-ci est facilement interprétable. Ici nous avons une RMSE de 86.33, soit environ 1 minute et 26 secondes de différence moyenne entre la prédiction du temps d’intervention et le temps réel..

## 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 entraîné et l’utilisateur.

Outil utilisé : FastAPI

Liste des fonctionnalités de l’API :

* Authentification
  + Users
  + Admin
* Model
  + Results(prédictions)
  + RMSE
* 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 la RMSE du modèle dépasse le seuil de **120** sur les dernières données, alors on procède 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 (**120 maximum**) n’est toujours pas atteint, alors il y a deux solutions :

* Si la RMSE est comprise entre 120 et 180 : L’outil reste en production mais une alerte mail devra être envoyée aux administrateurs pour trouver des pistes d’amélioration du modèle.
* Si la RMSE est strictement supérieure à 180 : L’application devra être bloquée.