 Confidentiel

Projet professionnel

Formation Data Analyst

Jean-François Massonnié,

Michaël Mathieu,

Christian Etien

Prédiction du temps de réponse d’un véhicule de la Brigade des Pompiers de Londres

#### TABLE DES MATIERES

[1 *Introduction* 5](#_Toc99888444)

[1.1 *Objet du document* 5](#_Toc99888445)

[*1.2* *Contexte* 5](#_Toc99888446)

[1.3 *Objectif* 6](#_Toc99888447)

[1.4 *Réalisation* 7](#_Toc99888448)

[2 *Les données* 7](#_Toc99888449)

[2.1 *Vision métier* 7](#_Toc99888450)

[Graphique du nombre d’intervention par type d’incident. 8](#_Toc99888451)

[Graphique du nombre d’intervention par type d’incident sans les fausses alarmes. 9](#_Toc99888452)

[Graphique du nombre d’intervention par district. 10](#_Toc99888453)

[Graphique du temps d’attente moyen par district. 11](#_Toc99888454)

[Graphique du temps d’attente moyen par type de jour. 12](#_Toc99888455)

[Graphique du temps d’attente moyen par type d’intervention. 13](#_Toc99888456)

[Graphique du temps d’attente moyen par type / mois. 14](#_Toc99888457)

[Graphique du nombre d’intervention moyen par mois. 15](#_Toc99888458)

[2.2 *Description* 16](#_Toc99888459)

[2.3 *Analyse* 16](#_Toc99888460)

[2.3.1 Exploration 16](#_Toc99888461)

[2.3.2 AttendanceTimeSeconds 16](#_Toc99888462)

[Corrélations 16](#_Toc99888463)

[*2.4* *Transformations* 18](#_Toc99888464)

[*2.4.1 Suppression de variables avec des valeurs manquantes* 18](#_Toc99888465)

[*2.4.2 Séparation du dataset en variables indicatrices et explicatives* 20](#_Toc99888471)

[Transformation des variables explicatives 21](#_Toc99888472)

[*2.4.2* *Transformation des variables indicatrices* 26](#_Toc99888473)

[2.4.3 Fusion des 2 datasets 27](#_Toc99888474)

[*3* *Modèles Machine Learning* 28](#_Toc99888475)

[*3.1* Schéma 28](#_Toc99888476)

[28](#_Toc99888477)

[*3.2* Méthode 29](#_Toc99888478)

[Modèles 29](#_Toc99888479)

[Gestion de l’aléatoire 29](#_Toc99888480)

[Stabilité 30](#_Toc99888481)

[Fiabilité 30](#_Toc99888482)

[Performance 30](#_Toc99888483)

[Variables les plus utiles 30](#_Toc99888484)

[3.2 Métriques 31](#_Toc99888485)

[Figure 12 Table de confusion 31](#_Toc99888486)

[3.3 Modèles 31](#_Toc99888487)

[3.3.1RandomForestRegressor 31](#_Toc99888488)

[Interprétation 31](#_Toc99888489)

[Hyperparamètres 31](#_Toc99888490)

[3.3.2LinearRegressor 33](#_Toc99888491)

[Interprétation 33](#_Toc99888492)

[Paramètres 33](#_Toc99888493)

[3.3.2XGBRegressor 34](#_Toc99888494)

[Interprétation 34](#_Toc99888495)

[Hyperparamètres 34](#_Toc99888496)

[Stabilité 34](#_Toc99888497)

[Fiabilité 34](#_Toc99888498)

[Performances 34](#_Toc99888499)

[Variables utiles 35](#_Toc99888500)

[4 Interprétation des résultats 36](#_Toc99888501)

[*4.1* Interprétabilité du modèle avec LIME 36](#_Toc99888502)

[*4.1.1* Interpréter la première observation 36](#_Toc99888503)

[*4.1.2* Comment une variable a contribué à une prédiction 37](#_Toc99888504)

[5 Conclusion 38](#_Toc99888505)

[6 Perspectives 39](#_Toc99888506)

### 1 Introduction

### 

### 1.1 Objet du document

### 

Ce document présente le résultat du projet professionnel effectué par Jean-François Massonnié, Michael Mathieu et Christian Etien dans le cadre de la formation Data Analyst dispensée par DataScientest (promotion DA01 – 2021/2022).

### 

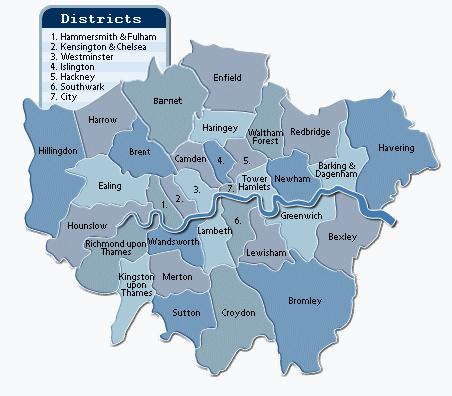
### Contexte

Le projet qui nous a été assigné est : Prédiction du temps de réponse d’un véhicule de la Brigade des Pompiers de Londres. (Vous pourrez trouver les jeux de données ici : <https://data.london.gov.uk/dataset/london-fire-brigade-incident-records>

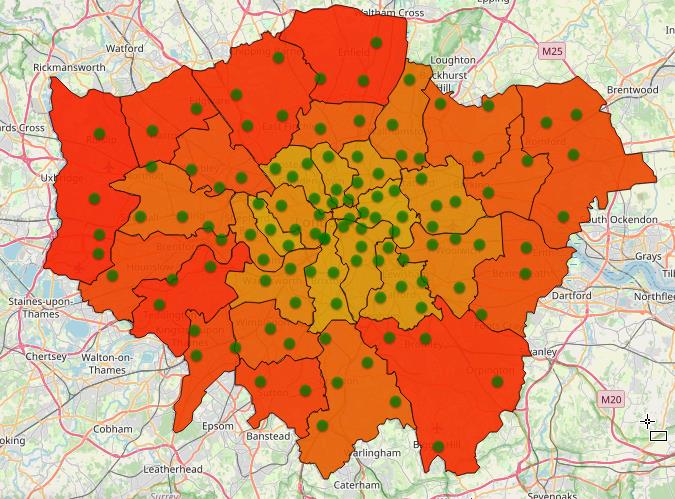
et <https://data.london.gov.uk/dataset/london-fire-brigade-mobilisation-records> )

A Londres, il y a 102 brigades de pompiers sur les 33 districts. ( source : <https://www.london-fire.gov.uk/community/your-borough/> )

Les districts sont distribués ainsi :



Localisation des brigades de pompiers à Londres :



### 1.3 Objectif

L’objectif de notre projet est l’analyse des documents fournit, nettoyage des données et créer une prévision du temps d’intervention selon plusieurs caractéristiques notamment en fonction des différents quartiers, du type d’urgence, donc du type de ressources mobilisées

Nous avons d’abord analysé la nature des variables contenues dans les 2 Datasets, puis réalisé un important travail de pré-processing des données aidées par une matrice de corrélation suivi de graphes de visualisation pour faire ressortir les variables les plus pertinentes pour notre problématique.

### 1.4 Réalisation

Le projet a été réalisé en Python, avec l’utilisation des librairies :

* Numpy
* Pandas
* Matplotlib et seaborn (visualisations)
* Pandas-profiling (très utile dans le cycle analyse/nettoyage)
* Scikit-learn, LinearRegression, RandomForestRegressor et XGBoost (modèles de régression)
* Shap et Lime pour l’interprétation des résultats

Vu le nombre élevé de variables initiales, nous avons mis en place une configuration décrivant le traitement à effectuer pour chacune d’entre elles :

* Chargement ou non lors de la lecture du fichier ?
* Utilisation ou non pour la phase d’analyse et/ou de prédiction ?
* Comportement face à des valeurs non renseignées : cas nominal ou non ?
* Passage ou non en indicatrices ? Avec encodage préalable ou non ?
* …

### 2 Les données

### 

### 2.1 Vision métier

Les données proviennent du site « Mayor of London », un site gouvernemental qui recensent toutes les données et statistiques sur les différents évènements de la ville de Londres et notamment les incidents de la brigade des pompiers de Londres.

Elles sont découpées en 2 jeux :

* Le premier jeu rassemblent les temps d’intervention de la brigade des pompiers de Londres
* Le deuxième jeu regroupe les ressources en personnes, véhicules et matériels mobilisées pour

ces interventions.

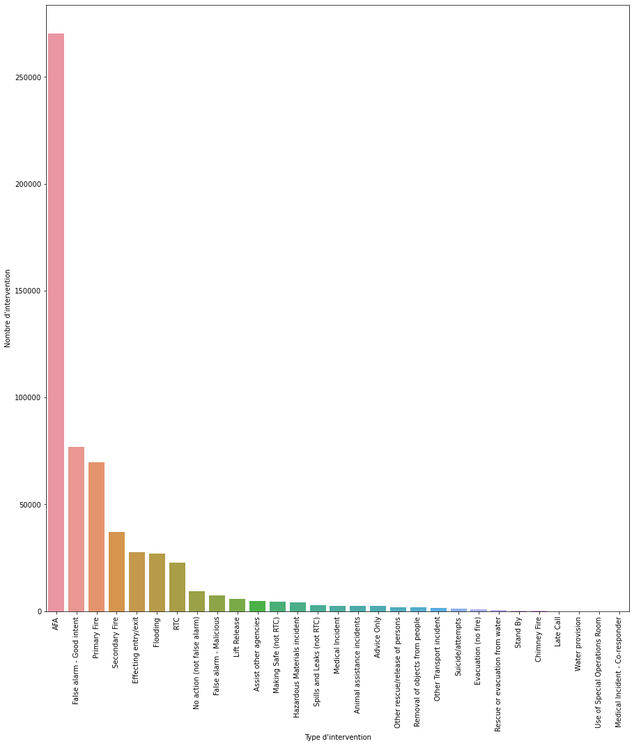
<https://data.london.gov.uk/dataset/london-fire-brigade-incident-records>

Notre choix pour l’analyse des données s’est porté sur un ensemble de données des 3 dernières années afin de simplifier l’analyse et l’apprentissage du Machine Learning.

Ci-dessous les différents graphes de visualisation qui nous ont emmenés aux différents choix des variables explicatives pour entrainer le modèle d’apprentissage.

Les graphes présentés ci-dessous sont organisés de la vision générale vers une vision de plus en plus précise en fonction des caractéristiques.

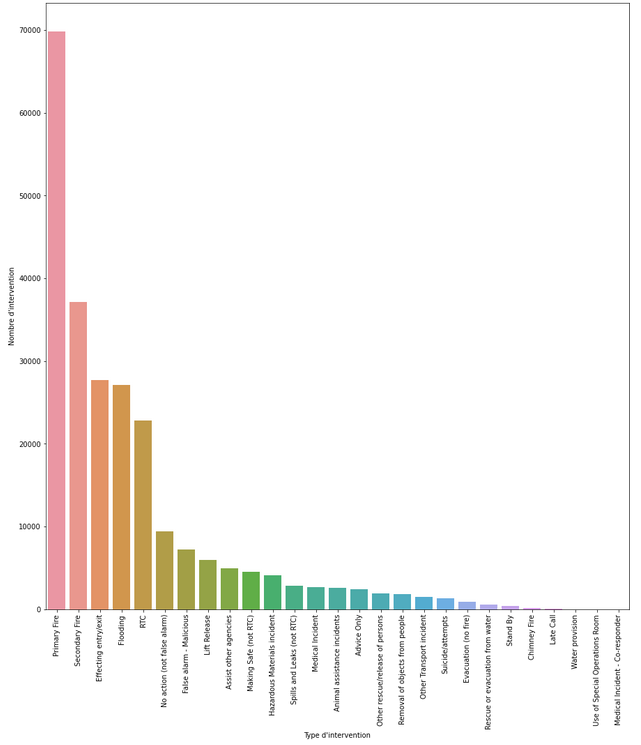
### Graphique du nombre d’intervention par type d’incident.



On constate que les fausses alertes représentent un nombre important d'incident !

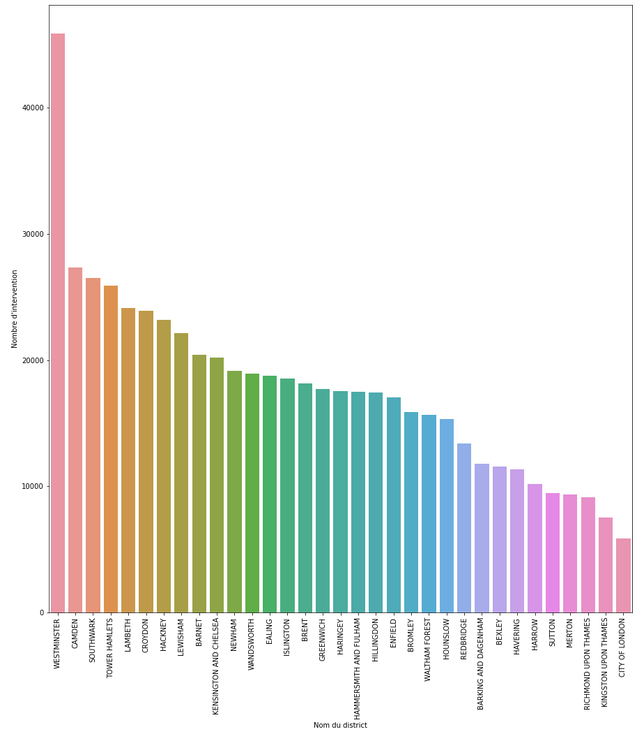
Suivis par les feux et les inondations.

### Graphique du nombre d’intervention par type d’incident sans les fausses alarmes.



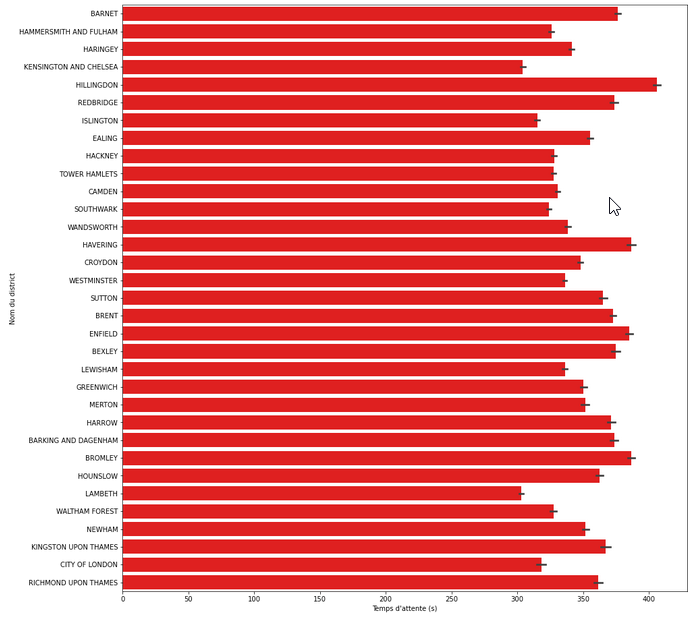
En retirant les fausses alarmes, ce sont les feux qui sont majoritaire sur les interventions.

### Graphique du nombre d’intervention par district.



Nous remarquons aussi que l'arrondissement de WESTMINSTER qui est le quartier le plus dynamique et le plus actif de Londres représente un nombre important d’intervention.

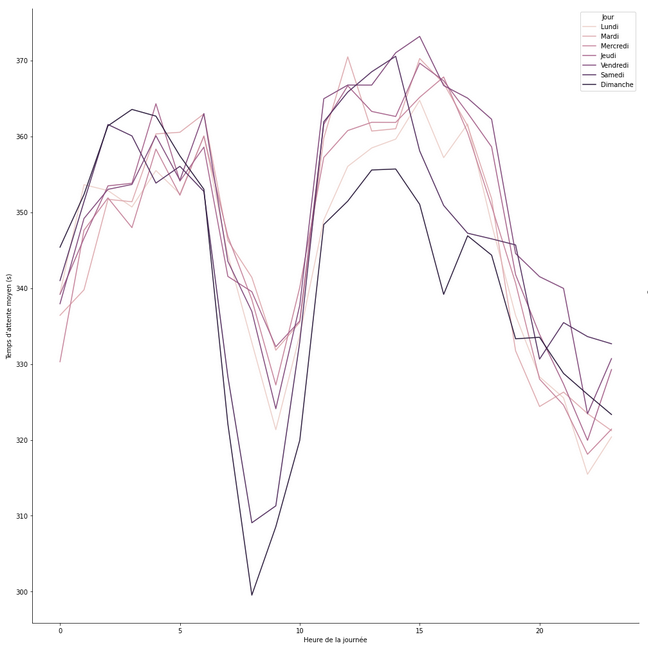
### Graphique du temps d’attente moyen par district.



Le district d'Hillingdon, qui se situe à L'Ouest de Londres, représente le lieu où les délais d'intervention sont les plus longs !

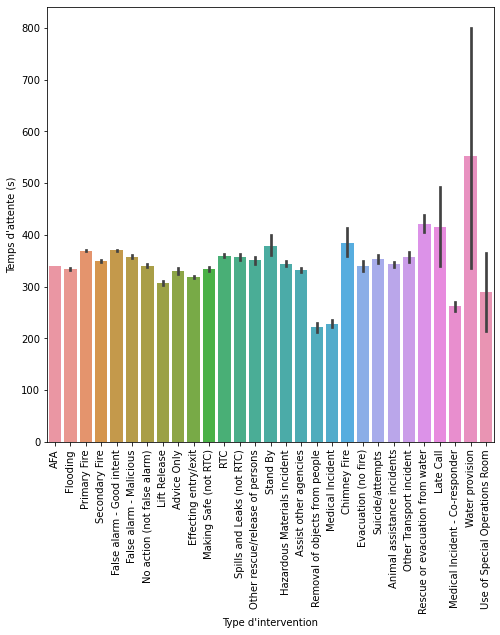
Et l'arrondissement de Lambeth qui se situe en centre-ville de Londres représente les délais les plus courts.

### Graphique du temps d’attente moyen par type de jour.



Si le jour ne semble pas impacter énormément le temps d’attente moyen, avec seulement une légère diminution le Week-End sur la période 6-10h, en revanche le type d’intervention influence plus fortement ce délai.

### Graphique du temps d’attente moyen par type d’intervention.



Ce graphe met en évidence quatre groupes de temps d’attente :

* Un premier groupe dont les temps d’attente, les plus courts du panel, se situe entre 200 et 250 secondes :

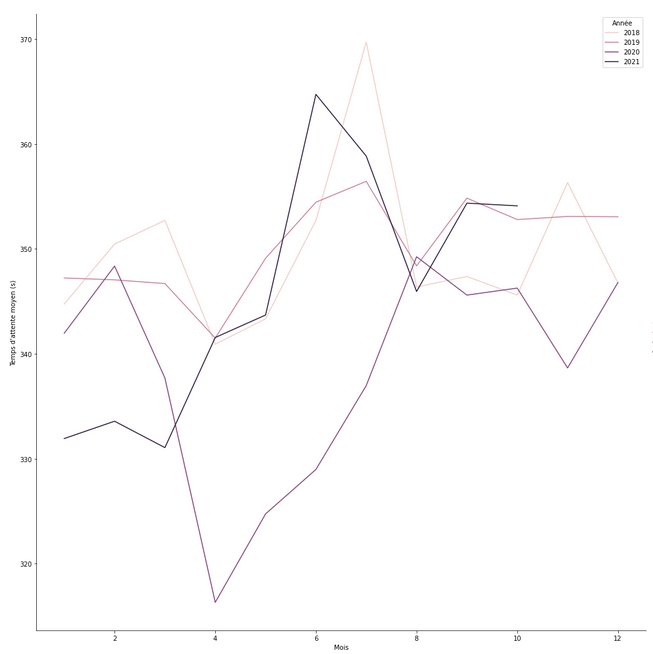
ce premier groupe concerne des incidents de personne : Enlèvement des objets aux personnes, incidents médicaux de personne et incidents médicaux de personnes collatérales.

* Un deuxième groupe avec des temps d’attente compris entre les bornes 300 et 400 secondes qui concernent soit des fausses alarmes soit des incidents de départ d’incendies.
* Le troisième groupe avec des temps supérieur à 400 secondes concernent des évacuations pour inondations
* Le quatrième et dernier, dont les temps sont les plus long, concerne des interventions d’approvisionnement en eau.

Nous pouvons conclure que plus les incidents nécessitant de mobiliser des véhicules lourds équipés de matériels importants, comme des camions citernes en intervention sur des départs de feu ou des camions d'approvisionnement en eau, plus les délais d'interventions sont longs.

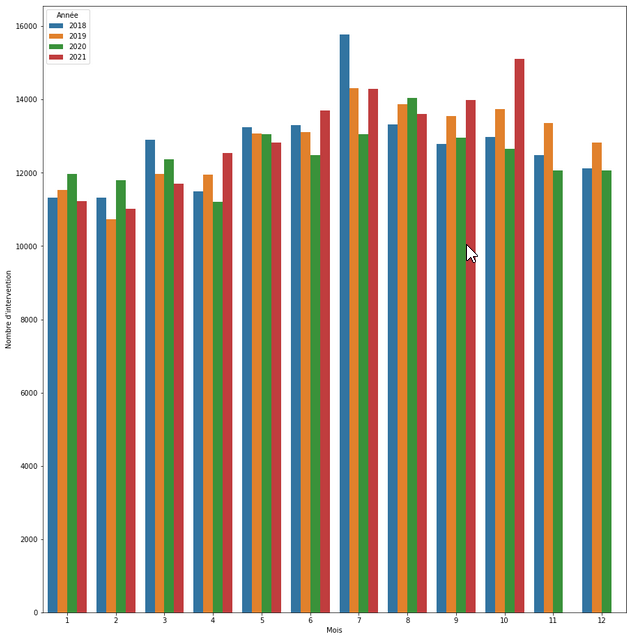
A l’opposé, les incidents médicaux de personne concentrent les délais d’intervention les plus court,

### Graphique du temps d’attente moyen par type / mois.



Nous avons constaté clairement une chute du temps d’attente entre avril et juillet 2020 lors du premier confinement peut-être en raison d’une circulation plus fluide à ce moment.

### Graphique du nombre d’intervention moyen par mois.



Exception faite des mois de juillet 2018 et d’octobre 2021, le nombre d’intervention a été stable.

### Description

Les variables sont de 3 catégories :

* Des caractéristiques sur la localisation géographique des différents quartiers

d’intervention.

* Des caractéristiques sur le type d’incident avec un détail sur le type de

bâtiment touché et l’importance de l’incident (incendie, ascenseur bloqué,

inondation etc …)

* Des caractéristiques sur les casernes et brigades d’interventions.

### 2.3 Analyse

### 2.3.1 Exploration

### 

Pour l’équilibre des données d’apprentissage face aux données de test nous avons pris le choix d’une méthode empirique en prenant un pourcentage du volume de données total sachant que nous avons 399 880 lignes nous avons pris 25 % (99 970) au départ mais ce volume paraissait trop important au risque de favoriser l’overfitting donc nous avons réduis le volume à 10% (39 988).

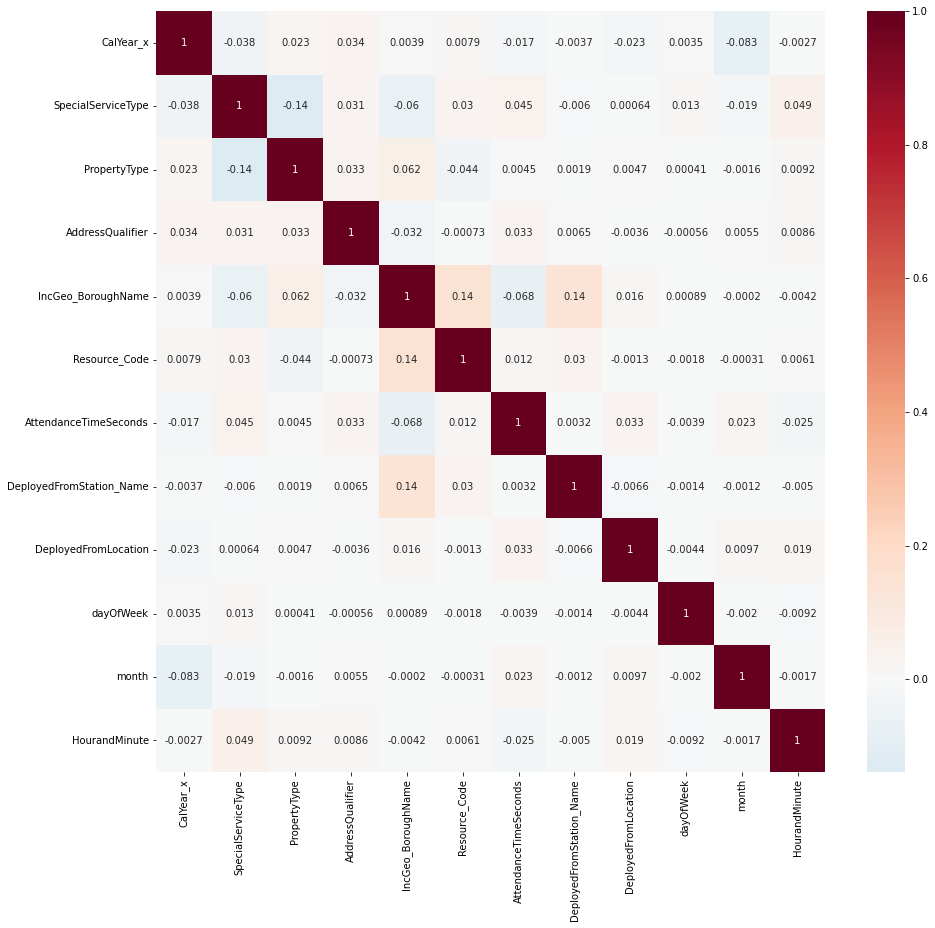
Ceci étant nous avons quand même utilisé la méthode des KMeans pour assurer notre choix de l’équilibre des données.

### 2.3.2 AttendanceTimeSeconds

‘AttendanceTimeSeconds’ est la variable qui porte l’indicateur du délai d’intervention. C’est la valeur que l’on va chercher à prédire. Cependant la prédiction à la seconde près semble très compliqué donc nous avons décidé de rajouter une variable « minute » pour optimiser la prédiction.

### Corrélations

Ici sont seules représentées les variables les plus corrélées en positif ou négatif à ‘AttendanceTimeSeconds’, on voit que les corrélations ne sont pas très fortes.



### Transformations

*2.4.1 Suppression de variables avec des valeurs manquantes*

Des transformations ont été nécessaires pour que les données soient exploitables pour la construction du modèle de machine learning, et ont été effectuées au cas par cas ou en plusieurs phases selon les variables.

Tout d’abord, une identification des variables présentant des valeurs manquantes ainsi qu’un contrôle du pourcentage de valeurs manquantes pour chacune, a été réalisée :

Liste des variables avec le ratio de valeurs manquantes :

IncidentNumber 0.000000

DateOfCall 0.000000

CalYear\_x 0.000000

TimeOfCall 0.000000

HourOfCall\_x 0.000000

IncidentGroup 0.000000

StopCodeDescription 0.000000

SpecialServiceType 78.636899

PropertyCategory 0.000000

PropertyType 0.000000

AddressQualifier 0.000000

Postcode\_full 58.263793

Postcode\_district 0.000000

UPRN 0.000000

USRN 0.000000

IncGeo\_BoroughCode 0.000000

IncGeo\_BoroughName 0.000000

ProperCase 0.000000

IncGeo\_WardCode 0.000170

IncGeo\_WardName 0.000170

IncGeo\_WardNameNew 0.000170

Easting\_m 58.263793

Northing\_m 58.263793

Easting\_rounded 0.000000

Northing\_rounded 0.000000

Latitude 58.263793

Longitude 58.263793

FRS 0.000000

IncidentStationGround 0.000000

FirstPumpArriving\_AttendanceTime 0.000341

FirstPumpArriving\_DeployedFromStation 0.002555

SecondPumpArriving\_AttendanceTime 38.120472

SecondPumpArriving\_DeployedFromStation 38.122005

NumStationsWithPumpsAttending 0.000000

NumPumpsAttending 0.000000

PumpCount 0.000000

PumpHoursRoundUp 0.026402

Notional Cost (£) 0.026402

CalYear\_y 0.000000

HourOfCall\_y 0.000000

ResourceMobilisationId 0.000000

Resource\_Code 0.000000

PerformanceReporting 0.000000

DateAndTimeMobilised 0.000000

DateAndTimeMobile 0.647790

DateAndTimeArrived 0.000000

TurnoutTimeSeconds 0.648812

TravelTimeSeconds 0.661417

AttendanceTimeSeconds 0.000000

DateAndTimeLeft 0.043776

DateAndTimeReturned 100.000000

DeployedFromStation\_Code 0.002214

DeployedFromStation\_Name 0.002214

DeployedFromLocation 0.057233

PumpOrder 0.000000

PlusCode\_Code 0.000000

PlusCode\_Description 0.000000

DelayCodeId 77.761880

DelayCode\_Description 77.761880

Cet inventaire réalisé, nous avons procédé à la suppression de toutes les variables contenant plus de 50% de valeurs manquantes et qui ne représentaient qu’un intérêt mineur pour l’apprentissage du modèle comme pour le résultat final. Nous avons exclus de cette suppression la variable « SpecialServiceType », conservée pour pouvoir être retravaillée ultérieurement de manière spécifique :

Liste des variables initiales conservées :

0 IncidentNumber 587073 non-null object

1 DateOfCall 587073 non-null object

2 CalYear\_x 587073 non-null int64

3 TimeOfCall 587073 non-null object

4 HourOfCall\_x 587073 non-null int64

5 IncidentGroup 587073 non-null object

6 StopCodeDescription 587073 non-null object

7 SpecialServiceType 125417 non-null object

8 PropertyCategory 587073 non-null object

9 PropertyType 587073 non-null object

10 AddressQualifier 587073 non-null object

11 Postcode\_district 587073 non-null object

12 UPRN 587073 non-null object

13 USRN 587073 non-null int64

14 IncGeo\_BoroughCode 587073 non-null object

15 IncGeo\_BoroughName 587073 non-null object

16 ProperCase 587073 non-null object

17 IncGeo\_WardCode 587072 non-null object

18 IncGeo\_WardName 587072 non-null object

19 IncGeo\_WardNameNew 587072 non-null object

20 Easting\_rounded 587073 non-null int64

21 Northing\_rounded 587073 non-null int64

22 FRS 587073 non-null object

23 IncidentStationGround 587073 non-null object

24 FirstPumpArriving\_AttendanceTime 587071 non-null float64

25 FirstPumpArriving\_DeployedFromStation 587058 non-null object

26 SecondPumpArriving\_AttendanceTime 363278 non-null float64

27 SecondPumpArriving\_DeployedFromStation 363269 non-null object

28 NumStationsWithPumpsAttending 587073 non-null float64

29 NumPumpsAttending 587073 non-null float64

30 PumpCount 587073 non-null float64

31 PumpHoursRoundUp 586918 non-null float64

32 Notional Cost (£) 586918 non-null float64

33 CalYear\_y 587073 non-null int64

34 HourOfCall\_y 587073 non-null int64

35 ResourceMobilisationId 587073 non-null int64

36 Resource\_Code 587073 non-null object

37 PerformanceReporting 587073 non-null object

38 DateAndTimeMobilised 587073 non-null object

39 DateAndTimeMobile 583270 non-null object

40 DateAndTimeArrived 587073 non-null object

41 TurnoutTimeSeconds 583264 non-null float64

42 TravelTimeSeconds 583190 non-null float64

43 AttendanceTimeSeconds 587073 non-null int64

44 DateAndTimeLeft 586816 non-null object

45 DeployedFromStation\_Code 587060 non-null object

46 DeployedFromStation\_Name 587060 non-null object

47 DeployedFromLocation 586737 non-null object

48 PumpOrder 587073 non-null int64

49 PlusCode\_Code 587073 non-null object

50 PlusCode\_Description 587073 non-null object

dtypes: float64(9), int64(10), object(32)



### *2.4.2 Séparation du dataset en variables indicatrices et explicatives*

Nous avons ensuite procédé à la séparation des variables en 2 groupes :

* Un groupe correspondant aux variables indicatrices
* Un groupe correspondant aux variables explicatives de type qualitative.

Les variables indicatrices contiennent donc les données de type numérique ou quantitative.

Les variables explicatives contiennent donc les données de type objet.

La séparation du dataset donne donc les 2 dataframe de variables suivants :

* df\_indicatrice
* df\_catégorielle

Cette opération avait pour objectif de préparer la phase de discrétisation des données explicatives.

Cette séparation est une étape nécessaire dans cette phase de preprocessing pour préparer la discrétisation des variables explicatives afin de pouvoir les rendre compatible à l’apprentissage de notre modèle de machine learning.

### Transformation des variables explicatives

* + 1. *Suppression des variables non nécessaire à notre modèle*

Après séparation du jeu de données en 2 dataframes, un dataframe des variables explicatives ou catégorielles, et un dataframe des variables indicatrices, nous nous sommes attachés à réaliser ici toutes les transformations qui nous ont semblé nécessaire du dataframe des variables explicatives.

Liste des variables explicatives conservées :

0 IncidentNumber 587073 non-null object

1 DateOfCall 587073 non-null object

2 CalYear\_x 587073 non-null int64

3 TimeOfCall 587073 non-null object

4 HourOfCall\_x 587073 non-null int64

5 IncidentGroup 587073 non-null object

6 StopCodeDescription 587073 non-null object

7 SpecialServiceType 125417 non-null object

8 PropertyCategory 587073 non-null object

9 PropertyType 587073 non-null object

10 AddressQualifier 587073 non-null object

11 Postcode\_district 587073 non-null object

12 UPRN 587073 non-null object

13 USRN 587073 non-null int64

14 IncGeo\_BoroughCode 587073 non-null object

15 IncGeo\_BoroughName 587073 non-null object

16 ProperCase 587073 non-null object

17 IncGeo\_WardCode 587072 non-null object

18 IncGeo\_WardName 587072 non-null object

19 IncGeo\_WardNameNew 587072 non-null object

20 Easting\_rounded 587073 non-null int64

21 Northing\_rounded 587073 non-null int64

22 FRS 587073 non-null object

23 IncidentStationGround 587073 non-null object

24 FirstPumpArriving\_AttendanceTime 587071 non-null float64

25 FirstPumpArriving\_DeployedFromStation 587058 non-null object

26 SecondPumpArriving\_AttendanceTime 363278 non-null float64

27 SecondPumpArriving\_DeployedFromStation 363269 non-null object

28 NumStationsWithPumpsAttending 587073 non-null float64

29 NumPumpsAttending 587073 non-null float64

30 PumpCount 587073 non-null float64

31 PumpHoursRoundUp 586918 non-null float64

32 Notional Cost (£) 586918 non-null float64

33 CalYear\_y 587073 non-null int64

34 HourOfCall\_y 587073 non-null int64

35 ResourceMobilisationId 587073 non-null int64

36 Resource\_Code 587073 non-null object

37 PerformanceReporting 587073 non-null object

38 DateAndTimeMobilised 587073 non-null object

39 DateAndTimeMobile 583270 non-null object

40 DateAndTimeArrived 587073 non-null object

41 TurnoutTimeSeconds 583264 non-null float64

42 TravelTimeSeconds 583190 non-null float64

43 AttendanceTimeSeconds 587073 non-null int64

44 DateAndTimeLeft 586816 non-null object

45 DeployedFromStation\_Code 587060 non-null object

46 DeployedFromStation\_Name 587060 non-null object

47 DeployedFromLocation 586737 non-null object

48 PumpOrder 587073 non-null int64

49 PlusCode\_Code 587073 non-null object

50 PlusCode\_Description 587073 non-null object

dtypes: float64(9), int64(10), object(32)

En complément à la suppression des variables avec un pourcentage de valeurs manquantes supérieur à 50%, nous avons procédé à la suppression des variables suivantes :

* La variable « IncidentNumber » qui ne représentait pas une notion métier et était donc peu influente sur l’apprentissage du modèle et sur l’analyse du résultat final.
* Les variables dont les valeurs de date et heures apparaissent en doublon pour ne conserver que celles utiles à notre apprentissage et à notre interprétation du résultat.

En revanche, les variables « DateAndTimeMobilised » et « DateAndTimeArrived » ont été conservées pour pouvoir opérer en résultat final une interprétation sur le délai d'intervention.

Liste des 27 variables explicatives conservées :

0 IncidentGroup 587073 non-null object

1 StopCodeDescription 587073 non-null object

2 SpecialServiceType 125417 non-null object

3 PropertyCategory 587073 non-null object

4 PropertyType 587073 non-null object

5 AddressQualifier 587073 non-null object

6 Postcode\_district 587073 non-null object

7 UPRN 587073 non-null object

8 IncGeo\_BoroughCode 587073 non-null object

9 IncGeo\_BoroughName 587073 non-null object

10 ProperCase 587073 non-null object

11 IncGeo\_WardCode 587072 non-null object

12 IncGeo\_WardName 587072 non-null object

13 IncGeo\_WardNameNew 587072 non-null object

14 FRS 587073 non-null object

15 IncidentStationGround 587073 non-null object

16 FirstPumpArriving\_DeployedFromStation 587058 non-null object

17 SecondPumpArriving\_DeployedFromStation 363269 non-null object

18 Resource\_Code 587073 non-null object

19 PerformanceReporting 587073 non-null object

20 DateAndTimeMobilised 587073 non-null object

21 DateAndTimeArrived 587073 non-null object

22 DeployedFromStation\_Code 587060 non-null object

23 DeployedFromStation\_Name 587060 non-null object

24 DeployedFromLocation 586737 non-null object

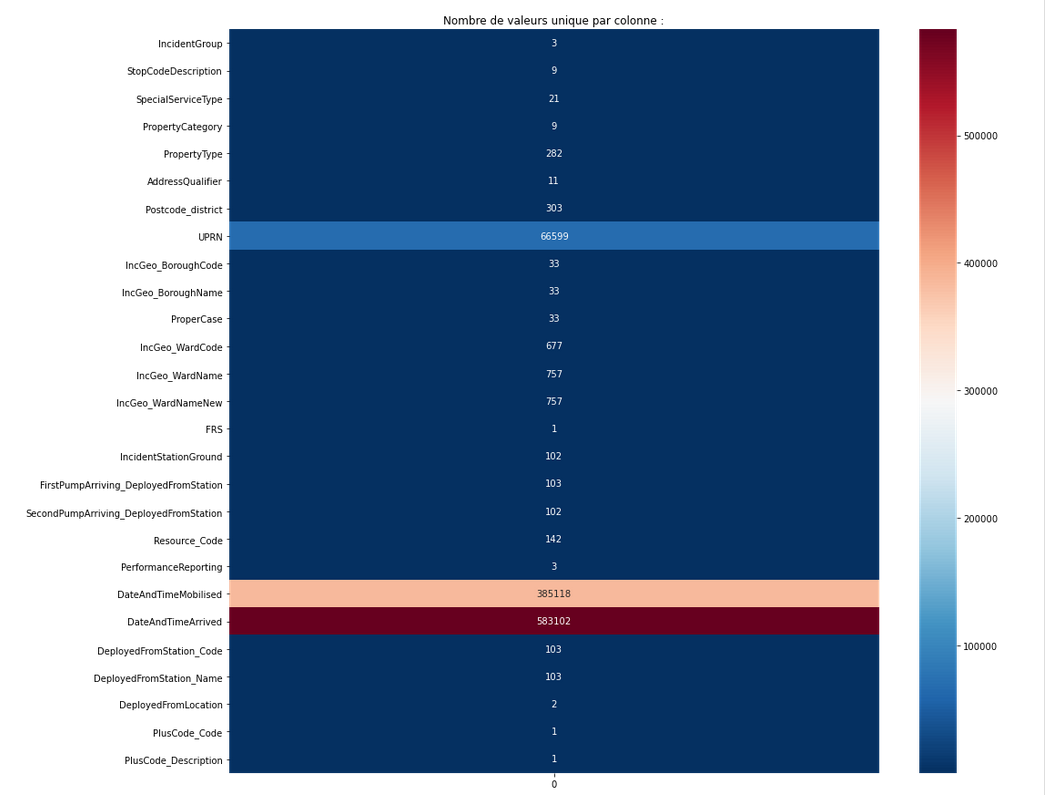
25 PlusCode\_Code 587073 non-null object

26 PlusCode\_Description 587073 non-null object

dtypes: object(27)

memory usage: 125.4+ MB

A ce stade, nous avons réalisé un graphe en heatmap du nombre de valeurs uniques par variable afin de vérifier pour chaque, le nombre de leurs modalités :



Ce graphe a mis en évidence plusieurs variables à retravailler :

1. La variable UPRN présentait un grand nombre de modalité. Or cette variable correspondant à un numéro unique de référence par bâtiment pour pouvoir localiser un bâtiment indépendamment de ses habitants, ne constituait pas une information importante pour notre objectif car sans influence spécifique sur le délai d’intervention.

Nous avons préféré la supprimer pour uniformiser notre modèle et éviter une problématique de sur-apprentissage.

1. Une différence apparaissait sur le nombre de modalité entre les variables « IncGeo\_WardName », « IncGeo\_WarNameNew » et « IncGeo\_WardCode ».

Après une analyse comparée des différences entre les valeurs des variables « IncGeo\_WarNameNew » et « IncGeo\_WardName », est décidé de ne conserver que la variable « IncGeo\_WardNameNew ».

1. La variable «IncGeo\_WardCode» étant moins représentative et présentant variables ne constituent pas des informations essentielles avec un impact pour l’apprentissage de notre modèle. Celles-ci ont donc été supprimées de notre dataframe.
2. En revanche les variables « PropertyType » et « PropertyCategory » ont été comparées en détail. En effet la variable « PropertyCategory » avec 9 modalités contre 282 pour la variable « PropertyType », a une information similaire mais moins détaillée.

Pour garder une cohérence sur le nombre de modalités sur l’ensemble du jeu de données, nous avons procédé à la suppression de la variable « PropertyType » en raison de son nombre de modalités.

Dans notre démarche de nettoyage des variables catégorielles, les données numériques présentaient moins de pertinence que les données apportant une information littérale.

Dans cette optique les variables suivantes n’ont pas été prises en compte :

- « SecondPumpArriving\_DeployedFromStation »,

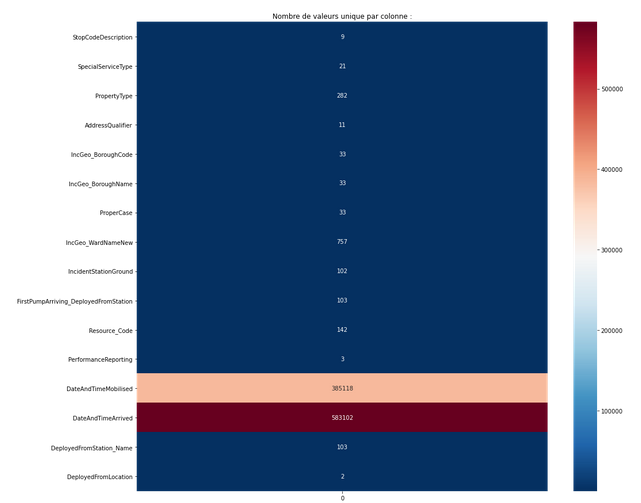
- « DeployedFromStation\_Code »

- « Postcode\_district »

En revanche la variable « Resource\_Code », représentant la Brigade intervenue, est conservée pour pouvoir affiner l’interprétation des résultats.

1. Les variables « FRS », « PlusCode\_Code », « Plus\_Code\_Description » et « IncidentGroup » ne présentaient qu’une seule modalité, donc sans intérêt. Celles-ci ont été supprimées.

Graphe des variables conservées avant l’étape de dichotomisation :



* + - 1. Exploration des variables catégorielles

En premier, nous avons alors procédé à l’analyse des variables « DeployedFromLocation » et « PerformanceReporting » :

* La variable « DeployedFromLocation » est transformée en variable indicatrice. Ses modalités explicatives « Home Station » et « Other Station » sont remplacées par des valeurs numériques 0 et 1 .
* La variable « PerformanceReporting » présente une modalité « not\_used » que l’on remplace par « 0 ».

Ces 2 variables, transformées en variables indicatrices, sont migrées dans le dataframe des variables indicatrices et supprimées de leur dataframe initial.

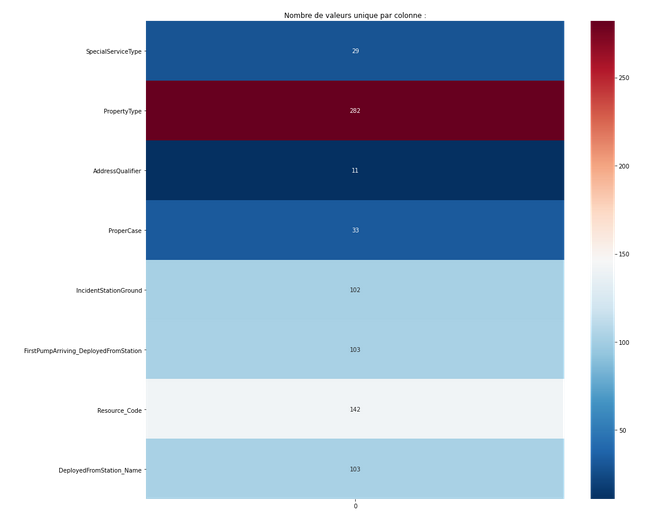
En second, les variables « IncGeo\_BoroughName » et « ProperCase » présentant des informations redondantes sur l’arrondissement de localisation de l’incident, nous avons décidé de ne conserver que la variable « ProperCase ».

Les valeurs « Nans » de la variable « SpecialServiceType » ont été remplacées par la modalité de la variable « StopCodeDescription ».

La variable « StopCodeDescription » devenue inutile, a été supprimé de notre jeu de données.

En troisième, pour préparer la discrétisation, nous avons supprimé les 2 variables de date « DateAndTimeMobilised » et « DateAndTimeArrived », ainsi que la variable « INcGeo\_WardNameNew » car leurs nombres de modalités trop important entraineraient un nombre de colonnes trop important pour pouvoir obtenir un modèle performant.

Graphe final des variables catégorielles utilisées pour notre modèle :



* + - 1. Discrétisation des variables catégorielles

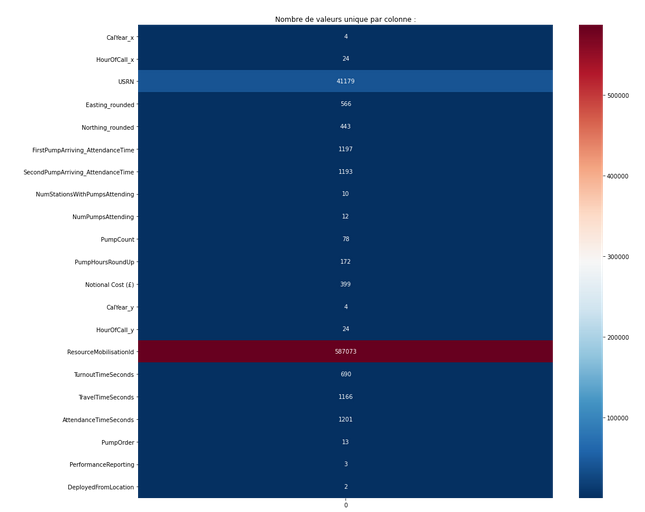
Pour terminer cette phase de pre-processing, nous avons procédé à la discrétisation de ces variables.

## *Transformation des variables indicatrices*

Après cette transformation des variables catégorielles ou explicatives, la seconde étape a consisté à opérer le même type de manipulation sur les données indicatrices.

L’objectif du projet étant de prédire un temps d’intervention en tenant compte des caractéristiques qui présentent le plus d’influence sur notre modèle, nous n’avons conservé ci-dessous que les variables indicatrices de type numérique et ayant un pourcentage de valeurs manquantes < 50%.

Graphe des variables indicatrices initiales avant pre-processing :



Dans une première phase de préparation de ces variables, nous avons supprimé les variables présentant peu d’intérêts pour l’apprentissage du modèle et pour l’interprétation des résultats.

Les variables non rattachées à des notions métier comme les identifiants ont aussi été supprimées du dataframe.

Les variables suivantes n’ont donc pas été prise en compte :

* « USRN »
* « ResourceMobilisationId »
* « FirstPumpArriving\_AttendanceTime »
* « SecondPumpArriving\_AttendanceTimme »
* « PumpOrder »
* « Easting\_rounded »
* « Northing\_rounded »

Les variables « PumpHoursRoundUp », « Notional Cost », et « DeployedFromLocation » qui ne contiennent pas de données pouvant influer sur le temps d’intervention des véhicules, ont été supprimées.

Enfin les variables « TurnoutTimeSeconds », « TravelTimeSeconds », représentant le délai global calculé dans la variable cible, donc sans importance réelle sur le modèle de machine learning, n’ont pas été conservées pour notre jeu de données final.

Au final, seules 9 variables indicatrices des 21 variables initiales ont été préservées pour entrainer et tester notre modèle de machine learning :

Liste des 9 variables indicatrices gardées :

0 CalYear\_x 587073 non-null int64

1 HourOfCall\_x 587073 non-null int64

2 NumStationsWithPumpsAttending 587073 non-null float64

3 NumPumpsAttending 587073 non-null float64

4 PumpCount 587073 non-null float64

5 CalYear\_y 587073 non-null int64

6 HourOfCall\_y 587073 non-null int64

7 AttendanceTimeSeconds 587073 non-null int64

8 PerformanceReporting 587073 non-null object

## Fusion des 2 datasets

Après l’achèvement de la phase nettoyage des données séparées en données catégorielles et indicatrices, nous avons réalisé au réassemblage des données finales en un seul jeu.

Une vérification de l’absence correcte valeurs nulle a été faite.

La variable cible « AttendanceTimeSeconds » a été séparée du reste du dataframe.

### Modèles Machine Learning

### Schéma

### 

### Méthode

### 

Au vu du nombre élevé de variables, l’enjeu est double : il faut prédire au mieux la correction avec le moins de variables possibles pour comprendre ce qui se passe. Une ACP n’aiderait pas car elle ferait perdre le lien avec les variables initiales et donc complexifierait l’interprétation des résultats. Je dois donc compter sur les modèles pour réduire le nombre de variables.

## Modèles

Une recherche de modèles à été réalisée qui permettent d’obtenir une courbe associant la variable cible et ses propres caractéristiques avec une interprétabilité la plus claire possible :

* Régression linéaire simple : LinearRegression de scikit-learn
* Forêt aléatoire : RandomForestRegressor de scikit-learn
* Ensemble d’arbres boostés : XGBRegressor de xgboost

D’autres modèles ont été testés, mais abandonnés :

* Régression logistique : LogisticRegression de scikit-learn
* Régression linéaire avec méthode Lasso : LinearRegression et Lasso de scikit-learn
* Régression linéaire avec méthode Ridge : LinearRegression et RidgeCV de scikit-learn
* Arbre de décision : DecisionTreeRegressor de scikit-learn

## Gestion de l’aléatoire

Pour chacun des modèles, les actions suivantes ont été effectuées :

* Le jeu de données initial a été séparé en 2 DataSets, les variables indicatrices d’un côté et les

variables qualitative d’un autre côté dans le but de réaliser un traitement spécifique.

* Réduction de dimensions pour le jeu de données qualitatif avec un choix de conserver les

variables qui possèdent un nombre de modalité unique ne dépassant pas un seuil de 300 afin

de limiter le nombre de variables lors de la phase de discrétisation.

* Suppression des variables qui ont peu d’importance dans l’apprentissage du modèle comme

tous les Ids et le numéro de l’incident etc … Et celles qui ont un nombre de modalités uniques

très élevées ( >1000 ) qui conduirait à la rupture de l’apprentissage du modèle.

* Discrétisation sur le jeu de données qualitatif.
* Le jeu de données initial a été coupé en 2 parties apprentissage/test avec un ratio 80/20.
* Pour certains modèles et dans un souci d’optimisation une validation croisée a été effectuée

sur la partie apprentissage afin d’optimiser les hyperparamètres.

* La performance du modèle a été évaluée sur la partie test.
* Les variables les plus influentes/importantes ont été extraites.

## Stabilité

Une fois que le DataSet a été travaillé, les variables utiles ont été filtrées afin d’identifier celles qui sont stables, c’est-à-dire qui sont importantes. Enfin, les variables utiles et stables ont été visualisées avec une Heatmap et boite à moustaches.

La notion de stabilité a mis en évidence le besoin de réduire les variables du jeu de données utilisé par les modèles. Pour ce faire, plusieurs méthodes ont été testées :

* Réduction dite ‘manuelle’ car ne s’appuyant pas sur des outils de scikit-learn.
* Méthode essais / erreur avec toujours la même démarche qui était de conserver les variables

jugées d’importance haute au niveau de la problématique et du métier conjugué avec un

nombre de modalités unique ne dépassant pas le seuil de 300.

Ces méthodes de réduction ont vite montré leur limite selon les modèles utilisés car peu optimisé.

Cependant à force d’itération un compromis à été trouvé pour atteindre un score satisfaisant au regard du volume de données et des caractéristiques à prendre en compte pour la prédiction du temps d’intervention.

## Fiabilité

La fiabilité du modèle a été évaluée par comparaison des accuracy d’entrainement.

* Comparaison du score R² entre les données réelles et de test
* MAE et MAPE (Marge d’erreur entre la valeur prédite et le score réel)
* RMSE

## Performance

Les performances du modèle stabilisé ont été évaluées

* Table de confusion
* Boîtes à moustaches

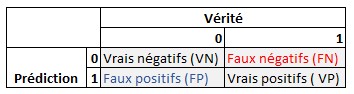
## Variables les plus utiles

Une fois le jeu de données stabilisé pour le modèle, les variables stables les plus importantes sont affichées avec leurs coefficients respectifs.

### 3.2 Métriques

### 

Les métriques sont ici détaillées dans l’optique de permettre la compréhension de la démarche à un public non averti, comme le métier, à qui ce rapport est également destiné.



# Figure 12 Table de confusion

La position métier étant :

1. Qu’il faut comprendre ce qui se passe
2. Il faut maximiser le taux de bonnes prédictions ET minimiser les faux négatifs au détriment des

faux positifs.

De même que l’accuracy à été évalué pour remonter la performance du modèle ainsi que les différentes marge d’erreur.

### 3.3 Modèles

## 3.3.1RandomForestRegressor

### Interprétation

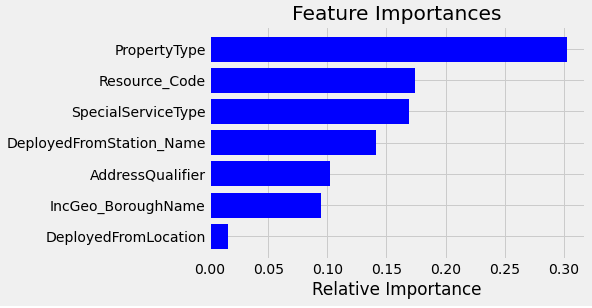
Comme pour les arbres, l’attribut feature\_importances\_ du modèle RandomForestRegressor contient une liste de coefficients donnant l’importance des variables. Un coefficient important signifie que la variable est intervenue de façon importante dans les nœuds des différents arbres de la forêt. Un coefficient faible signifie l’inverse.

Par contre la notion d’importance n’explique pas dans quel sens la variable influe sur la décision : Y=0 ou 1 ?

### Hyperparamètres

* Criterion : gini, sa valeur par défaut
* max\_depth : None, sa valeur par défaut
* max\_features : auto, sa valeur par défaut
* n\_estimators : 100, sa valeur par défaut

Feature Importances



Performances

score train R² : 0.46774935463828704

score test R² : 0.11313665992881194

Mean Absolute Error: 98.89

Mean Absolute Percentage Error : 65.87 %.

RMSE: 139.053624

Changement de l’hyperparamètre n\_estimators :

n\_estimators = 10

score train R² : 0.4470932076949665

score test R² : 0.09370932028465573

Mean Absolute Error: 100.03

Mean Absolute Percentage Error : 67.24 %.

RMSE: 140.698104

n\_estimators = 50

score train R² : 0.46165410692057973

score test R² : 0.11538946819875917

Mean Absolute Error: 98.95

Mean Absolute Percentage Error : 66.96 %.

RMSE: 139.005038

n\_estimators=150

score train R² : 0.4640448935151765

score test R² : 0.11828330577547852

Mean Absolute Error: 98.79

Mean Absolute Percentage Error : 66.82 %.

RMSE: 138.777488

n\_estimators=200

score train R² : 0.464351892175895

score test R² : 0.11858134099534468

Mean Absolute Error: 98.77

Mean Absolute Percentage Error : 66.81 %.

RMSE: 138.754031

La modification de max\_depth n’a pas donnée d’amélioration.

Ce modèle a été vite abandonné et n’a pas été optimisé.

## 3.3.2LinearRegressor

### Interprétation

La régression linéaire est un modèle qui permet de réaliser des prédictions ou des estimations sur la base de valeurs existantes. À partir d'un algorithme d'apprentissage supervisé, une relation linéaire est établie entre une variable expliquée et une variable explicative

Aussi appelée modèle linéaire, la régression linéaire est un modèle statistique qui effectue des fonctions prédictives. Pour réaliser des estimations pertinentes, le processus s’appuie sur des valeurs numériques afin de dégager une tendance ou une évolution prévisible. Par le biais d’un dataset, le système permet ainsi de les extrapoler et d’anticiper des valeurs futures.

La régression linéaire présente plusieurs champs d’application. Cela vaut pour développer une intelligence artificielle de type machine learning, capable d’apprendre de nouvelles règles et fonctions par elle-même.

### Paramètres

* n\_jobs : None, sa valeur par défaut
* normalize : False, sa valeur par défaut
* fit\_intercept : True, sa valeur par défaut

Performances

score train : 0.29399609802017945

score test : 0.2863378589517179

## 3.3.2XGBRegressor

## Interprétation

L’attribut feature\_importances\_ du modèle XGBRegressor contient une liste de coefficients donnant l’importance des variables. Un coefficient important signifie que la variable est intervenue de façon importante dans le modèle. Un coefficient faible signifie l’inverse.

Par contre la notion d’importance n’explique pas dans quel sens la variable influe sur la décision : Y=0 ou 1 ?

### Hyperparamètres

Pour le choix des hyperparamètres la stratégie ci-dessous à été mise en place :

L’ hyperparamètre max\_depth a été optimisé par validation croisée pour les valeurs (2, 3, 4, 5).

D’autres ont été fixés :

* le ratio d’apprentissage learning\_rate pour booster les arbres à chaque itération = 0.1.
* n\_estimators est resté à 100, sa valeur par défaut
* num\_parallel\_tree est resté à 1
* verbosity est resté à 0, sa valeur par défaut

Ces différents choix n’ont pas montré un résultat satisfaisant par conséquent nous avons décidé de laisser les paramètres par défaut. Le résultat est nettement meilleur.

### Stabilité

La stabilité du modèle avec toutes les variables n’étant pas satisfaisante, j’ai effectué une réduction manuelle qui aboutit à 11 variables qualitatives utiles. Tout en conservant l’intégralité de toutes les variables numériques.

### Fiabilité

Le modèle réduit présente une bonne fiabilité : l’entrainement monte et la validation stagne.

### Performances

En comparant les performances des modèles avant et après réduction sur l’ensemble de test pour variables, on voit que le modèle XGB est :

* Globalement meilleure et plus optimisé pour l’accuracy : 41 % contre 1% à 20% en moyenne

Training score R²: 0.4113613746125049

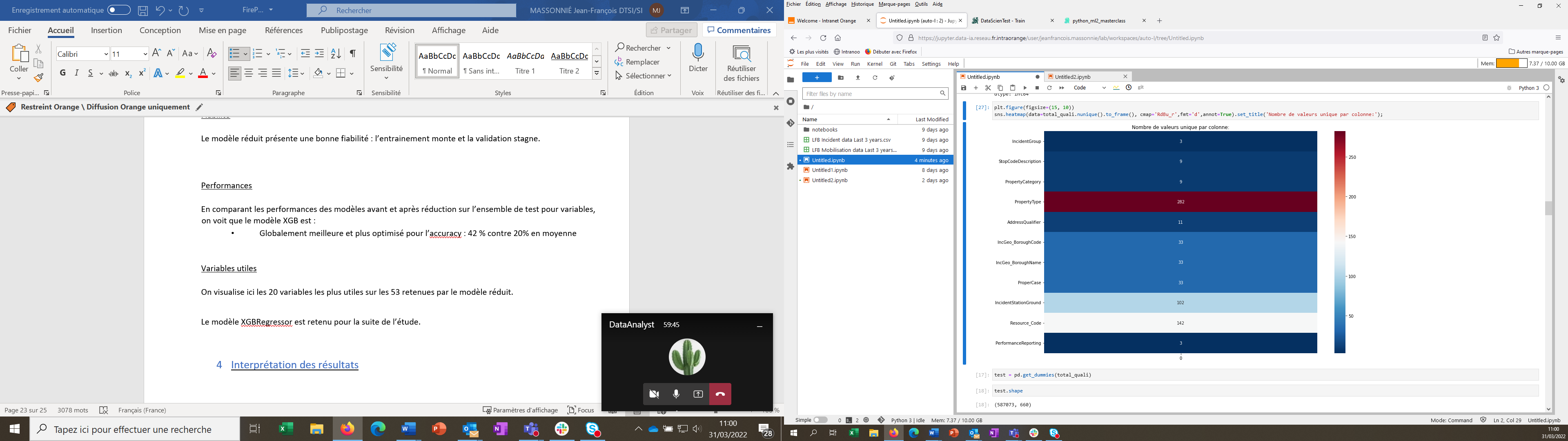
Test score R²: 0.37771337811825967

MSE: 3.77

RMSE: 1.94

### Variables utiles

On visualise ici les 11 variables les plus utiles sur les 59 retenues par le modèle réduit.



Le modèle XGBRegressor est retenu pour la suite de l’étude.

### Interprétation des résultats

### Interprétabilité du modèle avec LIME

Pour cette interprétation nous avons utilisé la librairie « Lime\_tabular » qui est utilisé pour générer les explications pour des données structurées et qui peut être utilisé pour des modèles de régression.

### Interpréter la première observation

Score prédit : 5.5134444

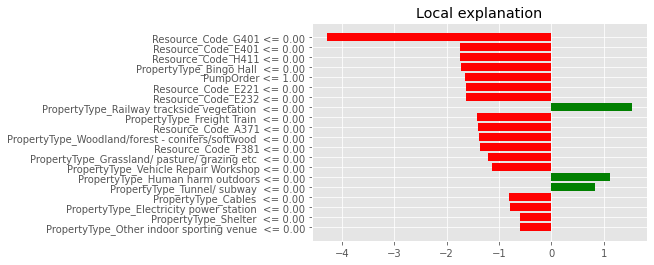
La valeur actuelle : 7

###### *La contribution de chaque variable dans la prédiction de la première observation*

Intercept 27.002428785508563

Prediction\_local [5.35665792]

Right: 5.5134444



###### *Les variables qui sont corrélées positivement avec la target sont illustrées en vert*

###### *Les variables qui sont corrélées négativement avec la target sont illustrées en rouge*

##### *Les valeurs faibles de PropertyType trackside vegetation (<=0.0) sont corrélées positivement avec une bonne qualité de prédiction du temps d’intervention.*

##### *Prédiction local de LIME = intercept + somme des coefficients de toutes les variables*

La somme des contributions de toutes les variables :

-21.645770864230844

[('Resource\_Code\_G401 <= 0.00', -4.282458327848167),

('Resource\_Code\_E401 <= 0.00', -1.7453302579629646),

('Resource\_Code\_H411 <= 0.00', -1.7374967575158984),

('PropertyType\_Bingo Hall <= 0.00', -1.7314710059478573),

('PumpOrder <= 1.00', -1.6464190543517894),

('Resource\_Code\_E221 <= 0.00', -1.6341084615701593),

('Resource\_Code\_E232 <= 0.00', -1.633082125996627),

('PropertyType\_Railway trackside vegetation <= 0.00', 1.5345841334623878),

('PropertyType\_Freight Train <= 0.00', -1.411752023943695),

('Resource\_Code\_A371 <= 0.00', -1.410238633749457),

('PropertyType\_Woodland/forest - conifers/softwood <= 0.00',

-1.390657727556499),

('Resource\_Code\_F381 <= 0.00', -1.3669624993451066),

('PropertyType\_Grassland/ pasture/ grazing etc <= 0.00',

-1.2020665398391952),

('PropertyType\_Vehicle Repair Workshop <= 0.00', -1.1435884439396677),

('PropertyType\_Human harm outdoors <= 0.00', 1.1270153957700553),

('PropertyType\_Tunnel/ subway <= 0.00', 0.8314334479615865),

('PropertyType\_Cables <= 0.00', -0.8079274795982265),

('PropertyType\_Electricity power station <= 0.00', -0.7840331716890833),

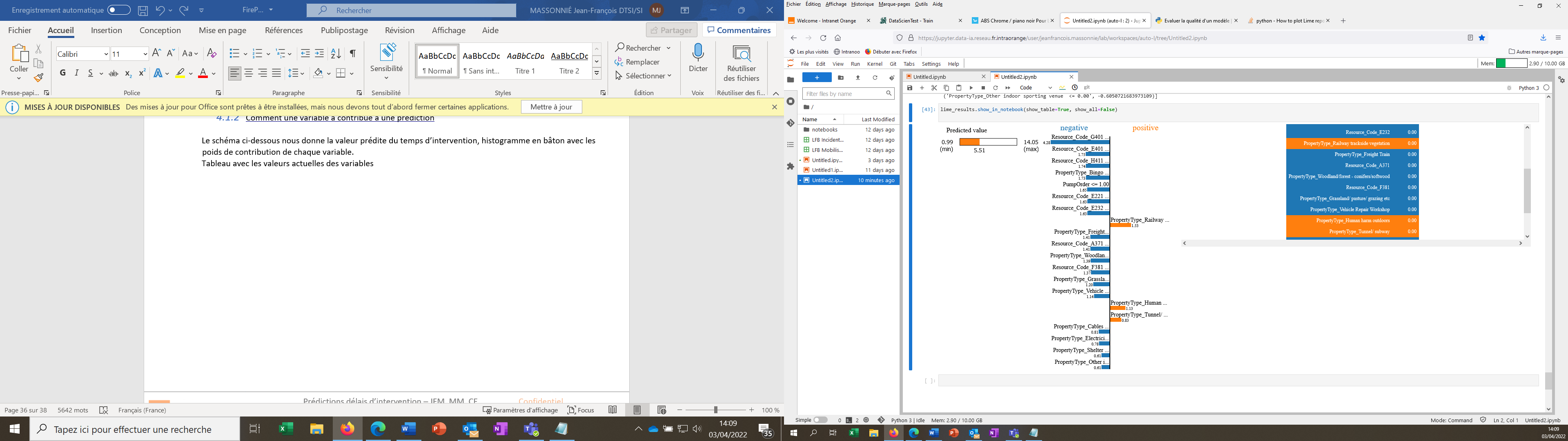
('PropertyType\_Shelter <= 0.00', -0.6061391621731699),

('PropertyType\_Other indoor sporting venue <= 0.00', -0.6050721683973109)]

### Comment une variable a contribué à une prédiction

Le schéma ci-dessous nous donne la valeur prédite du temps d’intervention, histogramme en bâton avec les poids de contribution de chaque variable.

Tableau avec les valeurs actuelles des variables

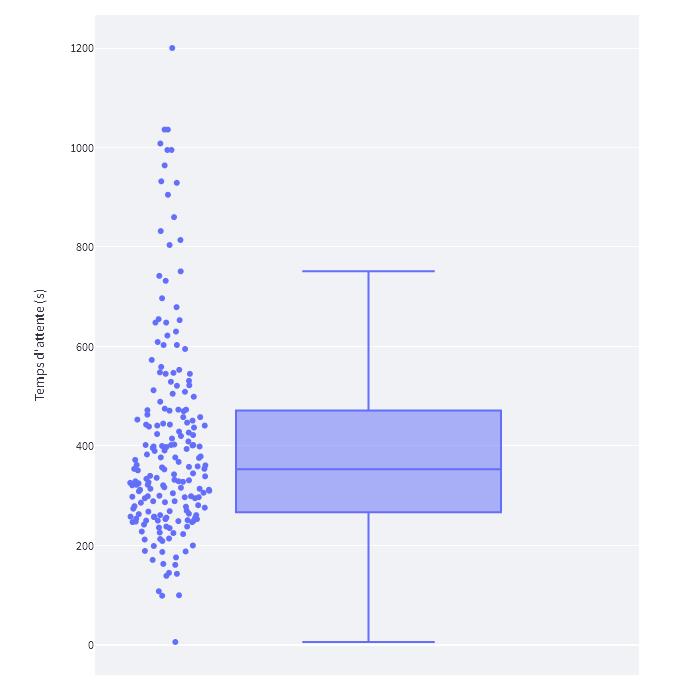


### Conclusion

Ce projet a demandé un gros travail en plusieurs temps :

* + La préparation des données a été chronophage et a demandé de nombreuses itérations.
  + Le choix des modèles par leur interprétabilité a nécessité beaucoup de lectures (cours, documentation scikit-learn et littérature sur le web).
  + L’entrainement des modèles n’a pas posé de problèmes. Optimiser les hyperparamètres dans l’objectif d’avoir les meilleures métriques est la partie facile du travail.
  + Cependant l’extraction des variables a été très compliquée. A partir du moment où l’on s’est rendu compte que ces variables étaient nombreuses et toutes semblaient pertinentes il nous a fallu beaucoup de réflexions et d’essais avant de trouver une méthode stabilisant ces variables pour un modèle. Cette difficulté a été pour nous révélatrice du fait que l’on avait au départ un volume de données très important. Les nombreuses caractéristiques à prendre en compte pour la prédiction du temps d’intervention s’est révélé très compliqué voire impossible à mesurer à la seconde prés.
  + Nous pouvons supposer que le temps d'attente dépend également de données extérieures que nous n'avons pas :
* Fluidité du trafic routier
* Météo
* Ressources de la caserne
* Congés ....
* Distance

Exemple de temps d'attente sur le district de Barnet avec une intervention sur un incendie principale d'un véhicule :



Ce gros travail a été extrêmement formateur, on est maintenant à l’aise avec bien des concepts dont on ne soupçonnait pas l’existence il y a 6 mois.

Malheureusement, les résultats sont assez maigres avec des performances devenues intéressantes seulement avec le modèle du XGBoost. On peut juste dire que le volume de données très importants avec toutes les caractéristiques à prendre en compte dans la prédiction nous a obligé à avoir recours à un nettoyage de données très avancés. On a dû trouver un compromis et avoir une stratégie entre conserver un maximum de variables significatives et supprimer celles qui semblais cannibalisé l’apprentissage du modèle.

### Perspectives

Des actions complémentaires peuvent être envisagées mais n’ont pu être lancées par manque de temps :

* + Revoir la pertinence des variables indicatrices car certaines peuvent faire défaut lors de l’apprentissage du modèle et rejouer le modèle XGBRegressor.
  + Il peut être intéressant de revoir les hyper paramètres car ceux-ci peuvent être optimisés.
  + Il peut être intéressant d’explorer aussi la piste des modèles ARIMA et SARIMA.
  + Le manque de temps nous a contraint de stopper nos investigations mais la problématique mérite d’aller plus loin pour améliorer la prédiction du modèle.