# Incident Management Project

Prédire le temps restant avant complétion

Bruno BERNARD

# Sommaire

# Contexte

Sujet : Prédire le temps de complétion d'un incident

- Nous disposons d'un dataset de 141712 ligne à analyser. Chaque ligne représente un incident. Les incidents sont regroupé par id (Number) et sont datés (opened_at, closed_at).
- Il existe 24918 incidents différents, il y a donc plusieurs lignes par incident, et donc plusieurs date d'ouverture et de fermeture.
- Pour prédire le temps total de résolution d'un incident, il serait donc nécessaire de regrouper les lignes d'un même évènement.
- Nous allons donc nettoyer cette donnée, l'analyser, puis faire des prédictions.

# Cycle de vie d’un incident

Déclaration

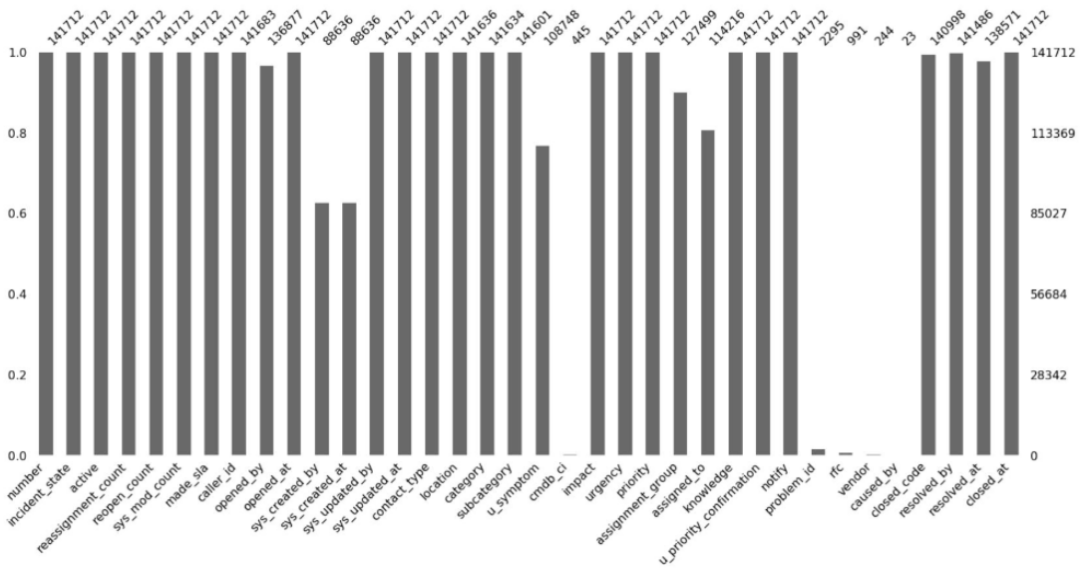
Prise en charge
( Initialisation de la date d’ouverture)

Modification de l’incident

Résolution de l’incident
( Initialisation de la date de fermeture)

# Modification du Dataset

- Le dataset est composé de beaucoup de valeur inconnu représenté par des « ? »
- Dans un premier temps, il faut remplacer les « ? » par NaN pour pouvoir analyser ces valeurs
- Grace à ce graphique nous remarquons que certaines colonnes dispose de trop peu d'information, nous allons donc les supprimés du dataset. (cmdb_ci, problem_id, rfc, vendor et caused_by)
- Les colonnes où il ne manque que quelques valeurs seront analysées et les lignes incomplètes seront supprimé. Etant donné le nombre de valeur dont on dispose, cette perte ne sera pas significative et permettra de mieux travailler la data.



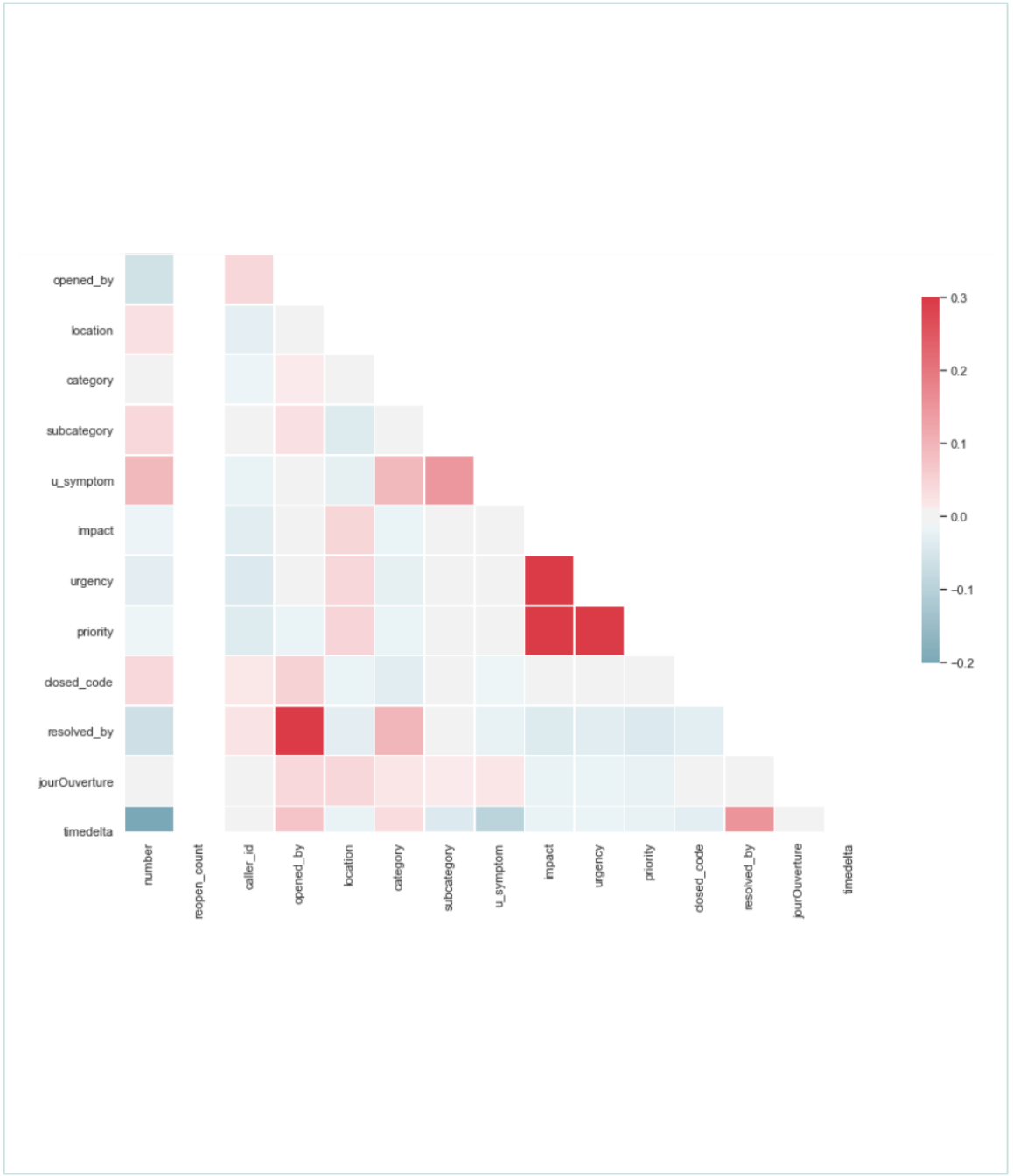
# Modification du Dataset

- Pour pouvoir analyser au mieux notre dataset, il est nécessaire le nettoyer. Pour cela nous allons tronquer les différentes valeurs pour ne garder que la partie chiffrée. Nous pouvons, par la suite, caster cette valeur en int.
- Un traitement est aussi nécessaire sur les dates. Nous nous intéresserons seulement aux dates d'ouverture et de fermeture. Nous calculerons le temps de traitement d'un ticket en soustrayant la date de fermeture avec celle d'ouverture pour avoir un résultat en jour que nous stockerons dans notre dataset.
- Nous ajoutons aussi une colonne qui renseigne le jour de la semaine pour ajouter de l'information
- La dernière modification consiste à supprimer les valeurs considéré comme absurdes pour ne passe fausser nos prédictions. Nous supprimons donc les incidents d'une durée supérieur a 50jours

# Etude

Correlation

- Pour TimeDelta on remarque une plus grande correlation avec resolved_by, open_by et u_symptom



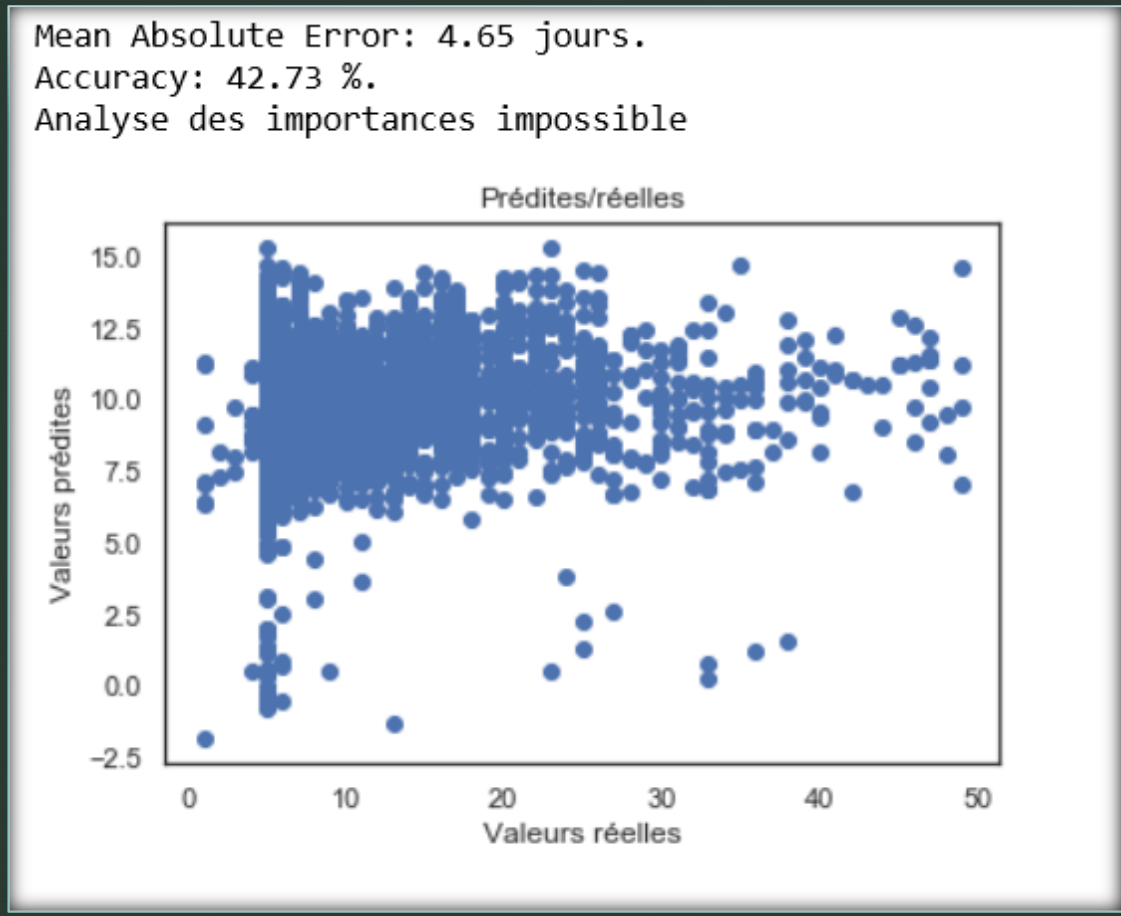
# Etude

Modèle testé :

- Régression linéaire
- Arbre de régression
- Réseau de neurone
- Random Forest
- ExtraTree Regressor
- GradientBoostingRegressor

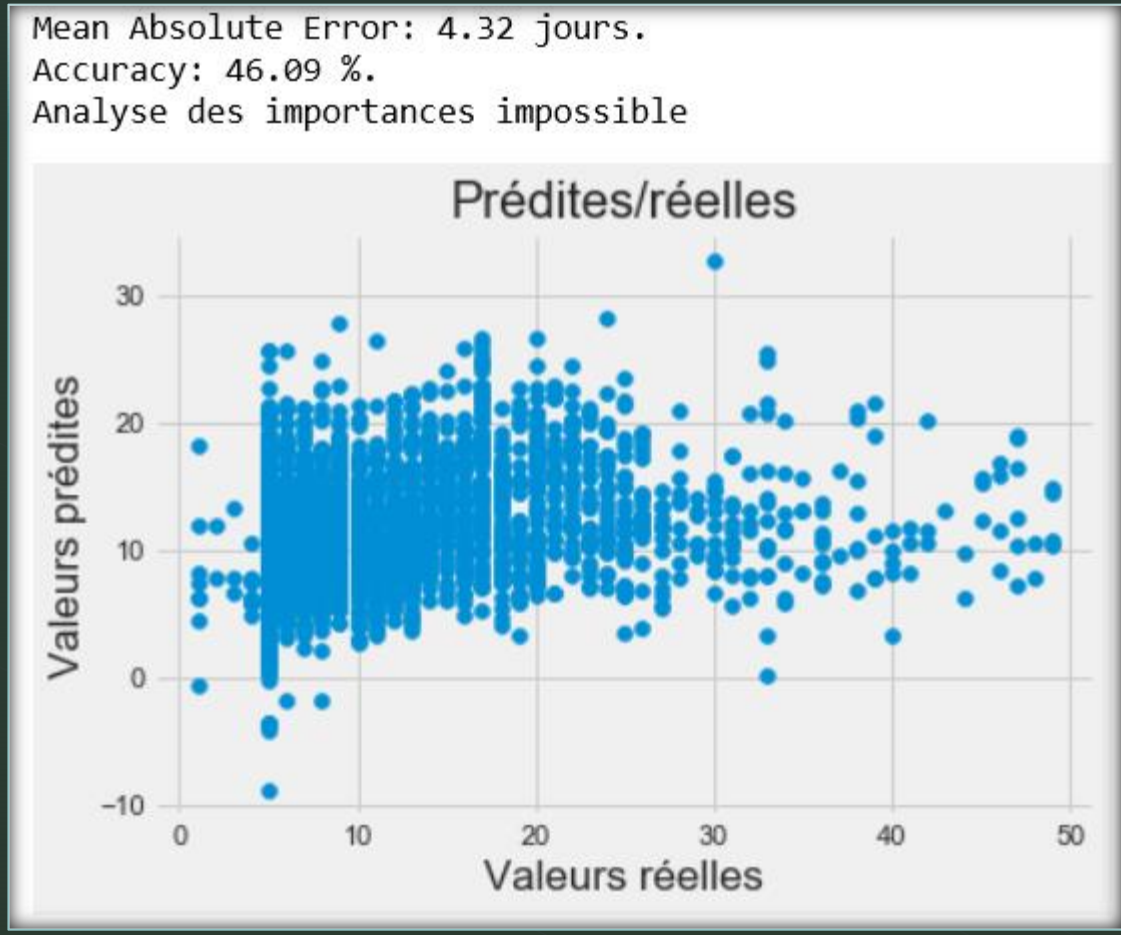
# Regression Linéaire

```
Mean Absolute Error: 4.65 jours.
Accuracy: 42.73 %.
Analyse des importances impossible
```



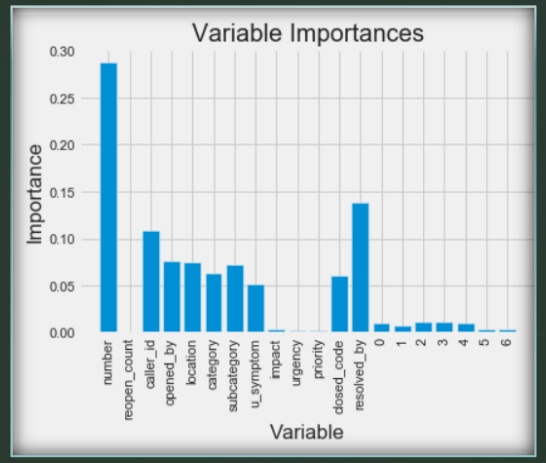
# Réseau de neurone

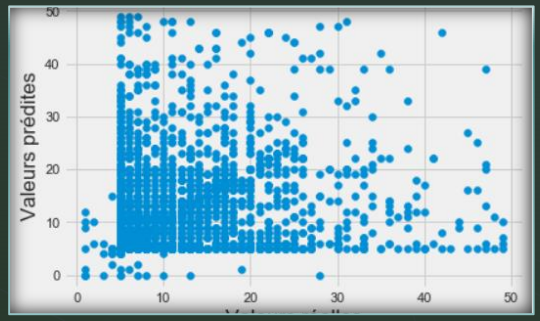
```
Mean Absolute Error: 4.32 jours.
Accuracy: 46.09 %.
Analyse des importances impossible
```



# Arbre de régression

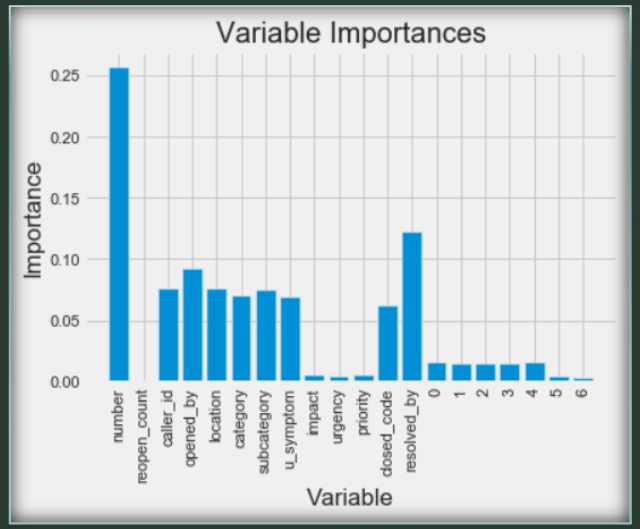
```
Mean Absolute Error: 4.14 jours.
Accuracy: 55.81 %.
Variable: number                Importance: 0.29
Variable: resolved_by           Importance: 0.14
Variable: caller_id             Importance: 0.11
Variable: opened_by             Importance: 0.08
Variable: location              Importance: 0.08
Variable: subcategory           Importance: 0.07
Variable: category              Importance: 0.06
Variable: closed_code           Importance: 0.06
```

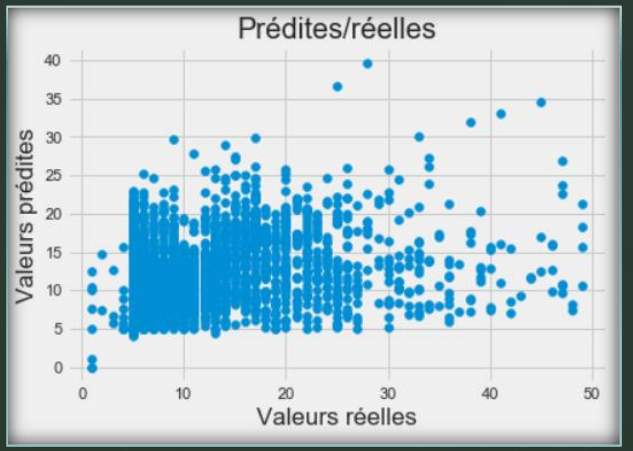


# ExtraTree Regressor

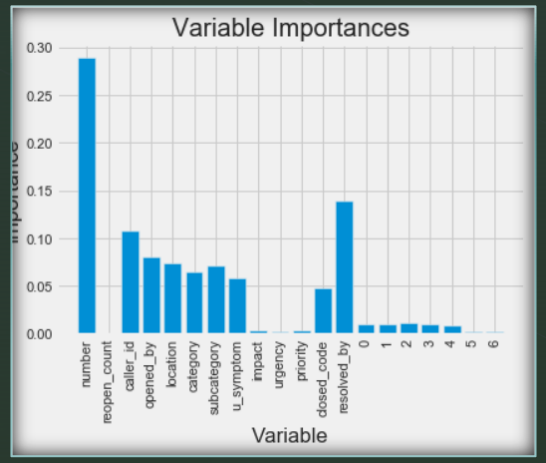
```
Mean Absolute Error: 3.57 jours.
Accuracy: 58.61 %.
Variable: number                Importance: 0.26
Variable: resolved_by           Importance: 0.12
Variable: opened_by             Importance: 0.09
Variable: caller_id             Importance: 0.08
```

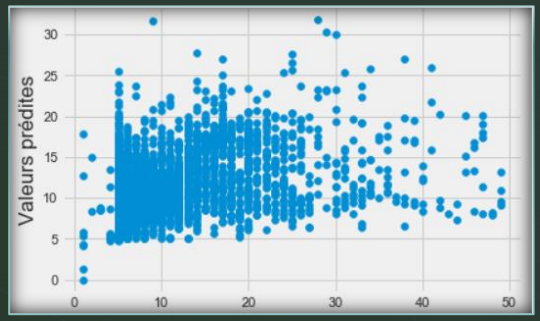


# Random Forest

```
Mean Absolute Error: 3.39 jours.
Accuracy: 60.81 %.
Variable: number                Importance: 0.29
Variable: resolved_by           Importance: 0.14
Variable: caller_id             Importance: 0.11
Variable: opened_by             Importance: 0.08
Variable: location              Importance: 0.07
Variable: category              Importance: 0.07
Variable: subcategory           Importance: 0.07
Variable: u_symptom             Importance: 0.06
```



# Résultat

- Après avoir essayer les différents modèles il se trouve que c'est avec RandomForest que l'on a la meilleurs performance avec Mean Absolute Error de 3.39 jours et une Accuracy de 60.81 %.
- Nous pouvons ensuite réalisé un hypertuning parameters avec une grille de recherche. Nous comparons 210fits.
- Voici les score finaux que l'on obtient suite à ca :
  Mean Absolute Error de 3.28 jours et une Accuracy de 61.52 %.

# Conclusion

- C’est une accuracy assez faible mais le modèle permet tout de même une prédication du nombre de jour que dure un incident à 3 jours près
- Bien sur, ce score peut-être plus élevé en rendant la dataframe plus performante ou en utilisant d’autres parametres