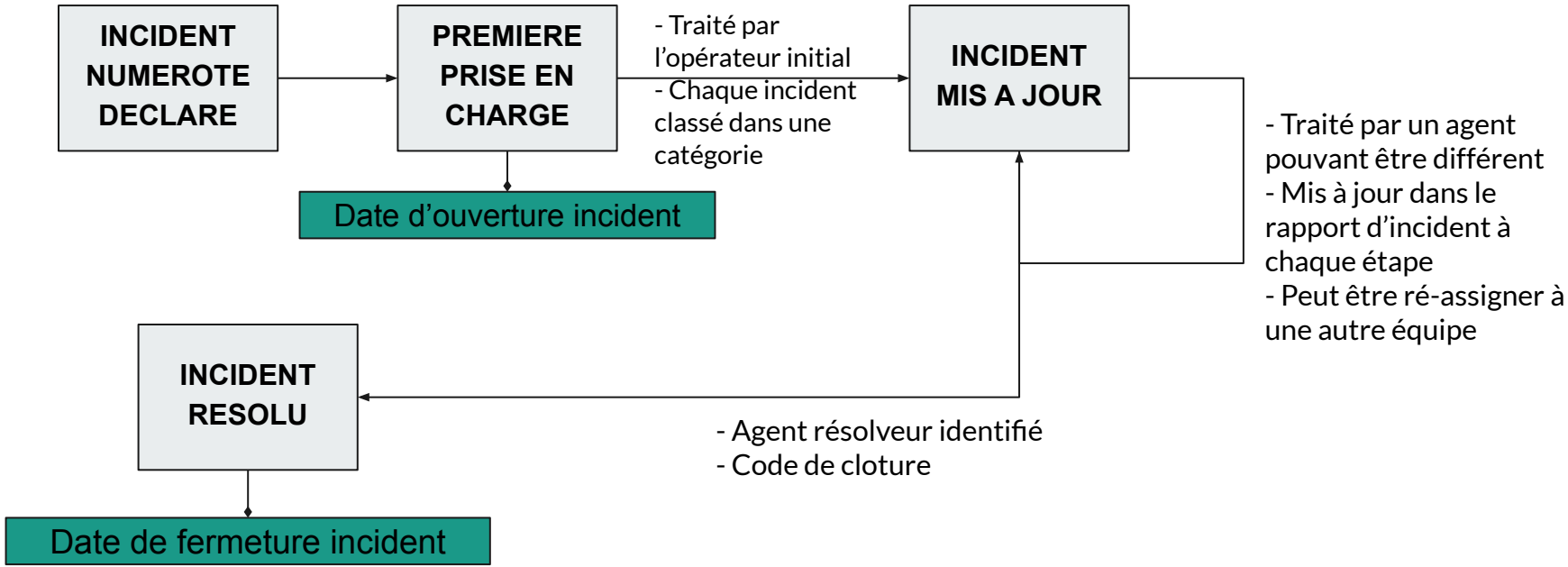
Incident Management Process

Prédire le temps de complétion d'un incident

Olivier DUPAIN

Processus de gestion de l'incident



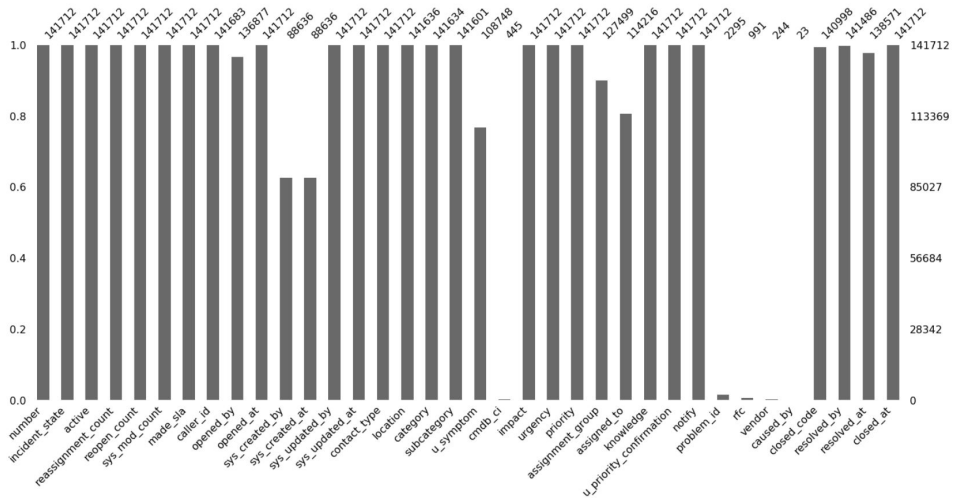
24 918 incidents

Sur les 141 712 données dont nous disposons dans cette DataFrame, il n'y a que 24 918 incidents uniques. On a donc en moyenne 6 lignes par incidents, 58 au maximum et 2 au minimum.

L'approche choisie consiste à regrouper plusieurs lignes d'un même évènement en 1 afin de prédire le **temps total** de résolution d'un incident.

Cette approche pose plusieurs problèmes pour fusionner les différentes lignes.

Valeurs manquantes



Si la colonne contient plus de 97% de valeurs manquantes, nous avons décidé de ne pas la sélectionner pour le traitement du problème car elle n'apporte pas assez d'information.

Les colonnes 'problem_id', 'rfc', 'vendor', 'caused_by', 'cmdb_ci' ont donc été supprimés pour cette raison.

Pour les autres colonnes contenant des valeurs différentes pour des mêmes incidents, nous expliqueront ensuite la démarche utilisée.

Passage en quantitatif

La grande majorité des colonnes de nos données sont de type qualitatives, boolean, date puis quantitatives. Nous décidons de garder un maximum d'information en transformant :

- La colonne ID de l'event en int
- L'ensemble des boolean en int
- Avec prétraitement et suppression des lignes ne contenant pas d'informations sur ces colonnes : caller_id, opened_by, location, category, subcategory, impact , urgency, priority, resolved_by, close_code, u_symptom en int

Gestion des dates

Pour les dates, nous avons décidé de sélectionner la date d'ouverture de l'incident et la date de fermeture. Nous ignorons dès lors, toutes les dates se rapportant à une mise à jour de l'incident.

Nous créons dès lors la variable target timedelta qui donnera l'information du temps de traitement de l'incident en jour. Pour cela, on soustrait les deux colonnes et affichons le temps de traitement dans cette nouvelle colonne.

Pour ajouter de l'information qui nous semble être pertinente : on crée une colonne jourdelasemaine qui contient l'information en int du jour et réalisons ensuite un **dummies pour ne pas donner de relation d'ordre entre les jours.**

TIMEDELTA
(en jour)

=

Date fermeture incident

-

Date d'ouverture incident

Valeurs absurdes

Au cours de l'exploration de la dataset, on s'aperçoit que certaines valeurs sont **extrêmes et viennent fausser les prédictions.**

Un incident se résout en effet en moyenne en 16,5 jours.

3 % des incidents durent plus que 50 jours.

On décide donc de supprimer ces incidents de notre étude.

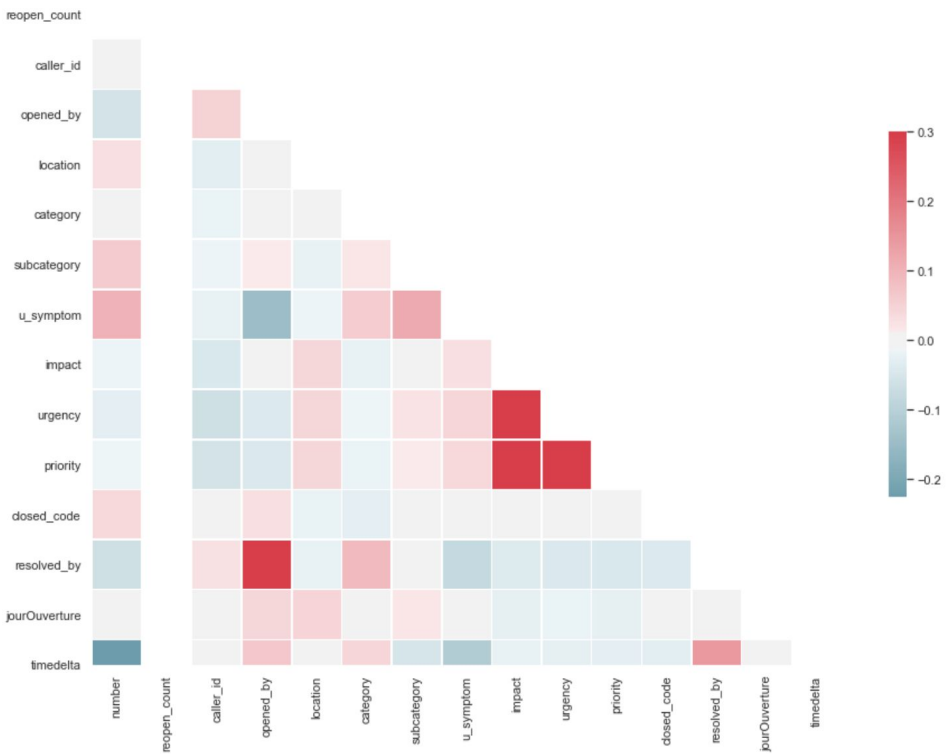
Début de l'étude

Nous avons donc notre dataframe prête, elle contient :

'number','reopen_count','caller_id','opened_by','location','category','subcategory','u_symptom','impact','urgency','priority','closed_code','resolved_by','jourOuverture'

Ainsi que l'information sur le jour de la semaine en dummies dans une colonne numérotée de 0 à 5.

Corrélation entre les variables



Timedelta semble être le plus corrélé avec la variable number, opened_by, u_symptom, resolved_by.

Ces variables devraient donc être plus décisives dans la construction du modèle.

Modèles testés

Voici l'ensemble des modèles testés :

- Régression linéaire
- Arbre de régression
- Réseau de neurone (MLP Regressor)
- RandomForest
- ExtraTreeRegressor
- GradientBoostingRegressor

Résultats et hypertuning

RandomForest est le modèle le plus performant. On réalise un hypertuning parameters avec une grille de recherche. Au total, 210 fits sont comparés.

On gagne + 0,5% en accuracy grâce à la grille de recherche.

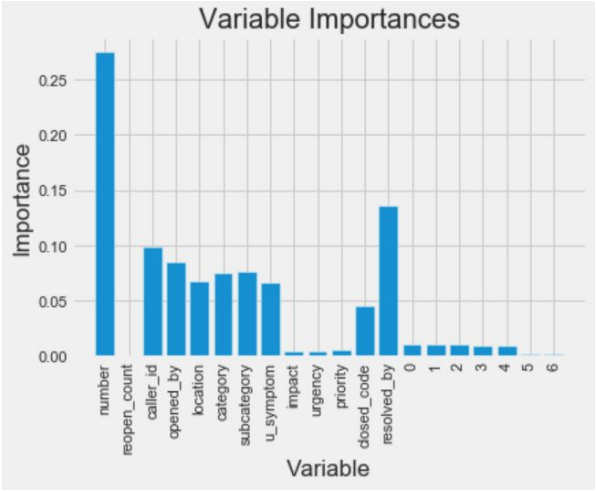
Voici les scores finaux :

MAE = 3,38 jours

Accuracy = 59,64%

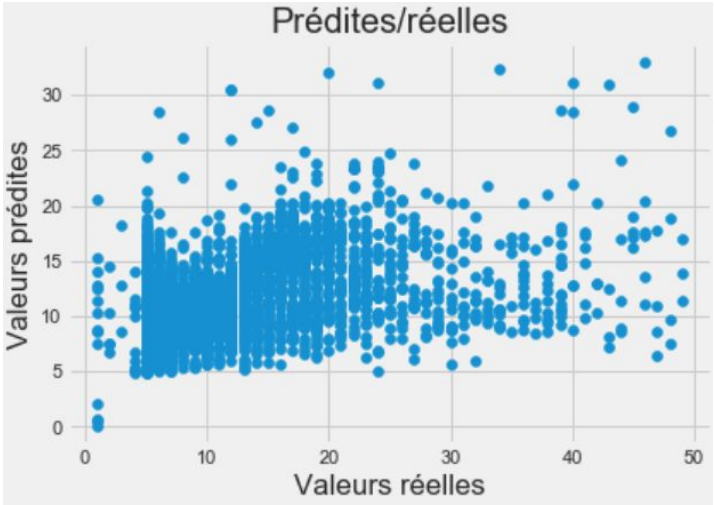
Résultats en graphiques

Importance des variables :

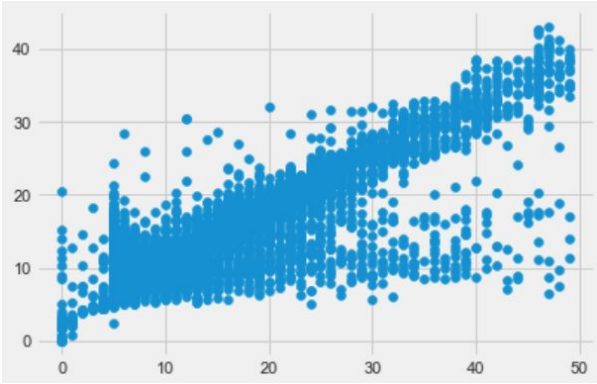


...

Balance prédites/réelles sur jeu de test…



la dataset :



Remarques et conclusion

L'accuracy finale n'est pas très élevée. Cependant le modèle fonctionne relativement bien car il permet de prédire avec une erreur moyenne de 3,5 jours le bon nombre de jours alors qu'un incident est réglé en moyenne en 16 jours.

Il permet donc d'approximer le résultat ou au moins d'en donner un ordre d'idée.

Pour améliorer, j'aimerai essayer par la suite :

- Comprendre pourquoi la feature number à tant d'importance
- Essayer de garder l'information updated_by en gardant l'opérateur qui a le plus de fois mis à jours l'incident
- Voir comment modifier la dataframe pour la rendre encore plus performante