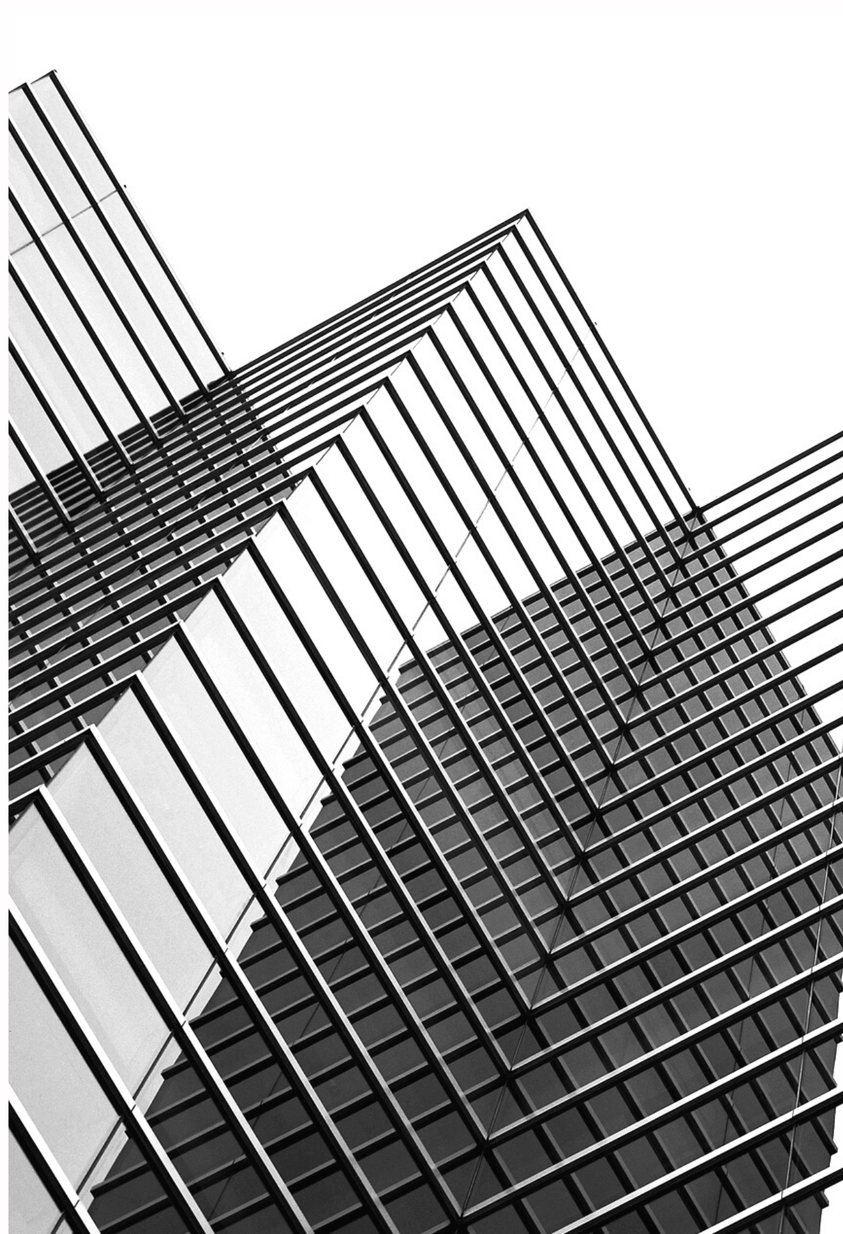
INCIDENT MANAGEMENT PROCESS ENRICHED EVENT LOG DATA SET



PROJET PYTHON-JORIS CHRISSACHOS

Pré-requis pour pouvoir lancer le code sur Jupyter

-Pour pouvoir lancer le code , il faut installer les différents packages : !pip install "le_package"

- Pour télécharger le dossier complet avec le modèle et le scaler déjà présent dans le fichier de l'API (APIPython/project/ipynb) car fichier trop volumineux pour GITHUB , je vous ai mis un lien vers mon drive dans le mail

Pré-requis pour pouvoir lancer le code sur Jupyter

-Si vous ne téléchargez le code via le lien de la diapositive précédente , il faut compiler le code(le code à été fait avec un pc 24threads/64GB).
- il y a n_job à changer en fonction du nombre de thread de son pc
-Pour pouvoir exécuter l'API , il faut mettre le modèle et le scaler dans le dossier (APIPython/project/ipynb)

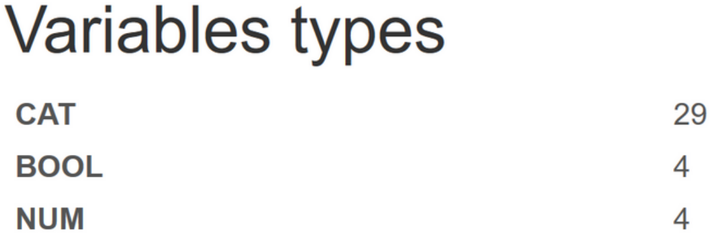
Introduction

Ce dataset est un journal des événements des processus de gestion des incidents extrait des données collectées à partir du système d'audit d'une instance de la plateforme ServiceNowTM utilisée par une société informatique.

Information sur le dataset

Ce dataset comporte 141,712 instances d'événement, 36 attributs pour un total de 24,918 incidents.

Après le calcul de temps entre les logs et la fermeture,nous avons ce type de variable présent dans la dataset:



Information sur le dataset

Il a fallut nettoyer le dataset car plusieurs variables avait des informations non exploitable.

cmdb_ci
Categorical

HIGH CARDINALITY

Distinct count	51
Unique (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	1.1 MiB

?	141267
cmdb_ci 31	32
cmdb_ci 49	24
cmdb_ci 7	24
cmdb_ci 11	21
Other values (46)	344

rfc
Categorical

HIGH CARDINALITY

Distinct count	182
Unique (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	1.1 MiB

?	140721
CHG0000132	20
CHG0001230	20
CHG0000047	18
CHG0001656	17
Other values (177)	916

caused_by
Categorical

Distinct count	4
Unique (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	1.1 MiB

?	141689
CHG0000097	11
CHG0000132	7
CHG0001327	5

problem_id
Categorical

HIGH CARDINALITY

Distinct count	253
Unique (%)	0.2%
Missing	0
Missing (%)	0.0%
Memory size	1.1 MiB

?	139417
Problem ID 14	184
Problem ID 2	147
Problem ID 52	82
Problem ID 239	48
Other values (248)	1834

vendor
Categorical

Distinct count	5
Unique (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	1.1 MiB

?	141468
code 8s	167
Vendor 1	69
Vendor 3	6
Vendor 2	2

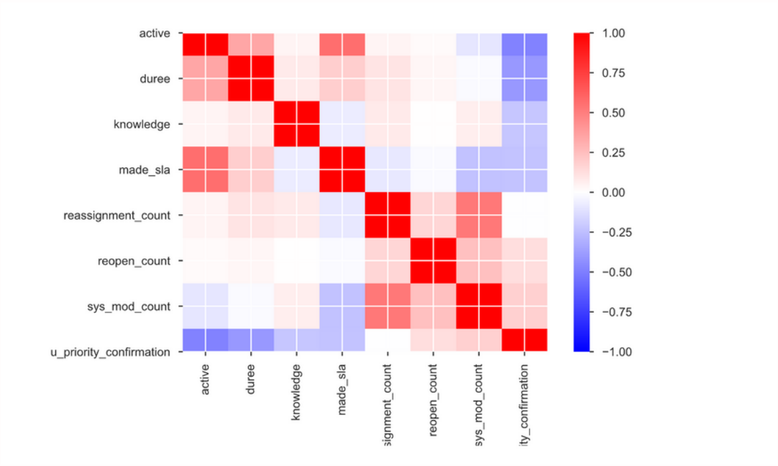
Calcul du temps restant avant complétion

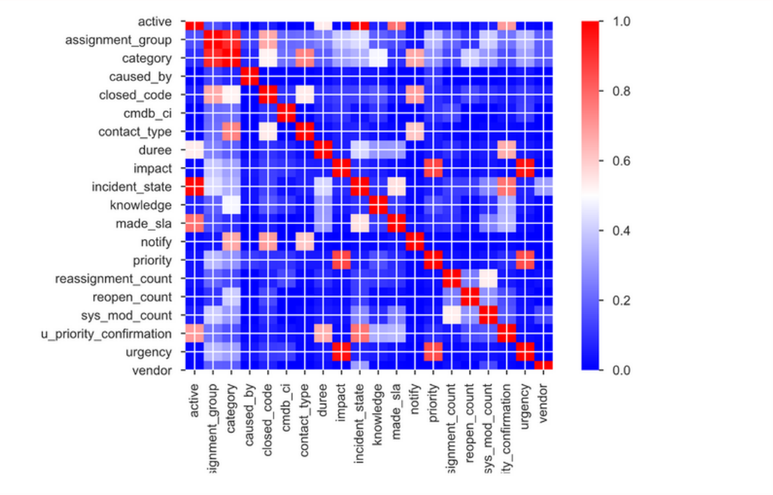
Pour le calcul du temps restant , j'ai utilisé comme variable "sys_updated-at" et "closed_at" pour avoir la différence entre chaque nouveau logs.

il a fallu transformé la date pour pouvoir effectuer une soustraction entre elle pour obtenir des secondes.

closed_at - sys_updated_at = duree

Matrice de corrélation





Grâce à la matrice de corrélation,on a pu déterminer quelles variables ont des liens avec les autres

les Features pour prédire le label(duree)

Pour le choix des features , je me suis basé sur la matrice de corrélation et de quelques variables qui me paraissait importante.

Pour les Features : on a "active","incident_state","resolved_at","opened_at","number","sys_mod_count","u_priority_confirmation'

Pour prédire : "duree"

les Features pour prédire le label(duree)

Pour le choix des features , je me suis basé sur la matrice de corrélation et de quelques variables qui me paraissait importante.

Pour les Features : on a "active","incident_state","resolved_at","opened_at","number","sys_mod_count","u_priority_confirmation'

Pour prédire : "duree"

Les différents modèle de régression

CatBoostRegressor: 34%

DecisionTreeRegressor: 11%

RandomForestRegressor: 10,6%

ExtraTreesRegressor: 46%

XgBoostRegressor: 30%

AdaBoostRegressor: 39%

RandomForestRegressor(avec paramètre optimisé): 48%

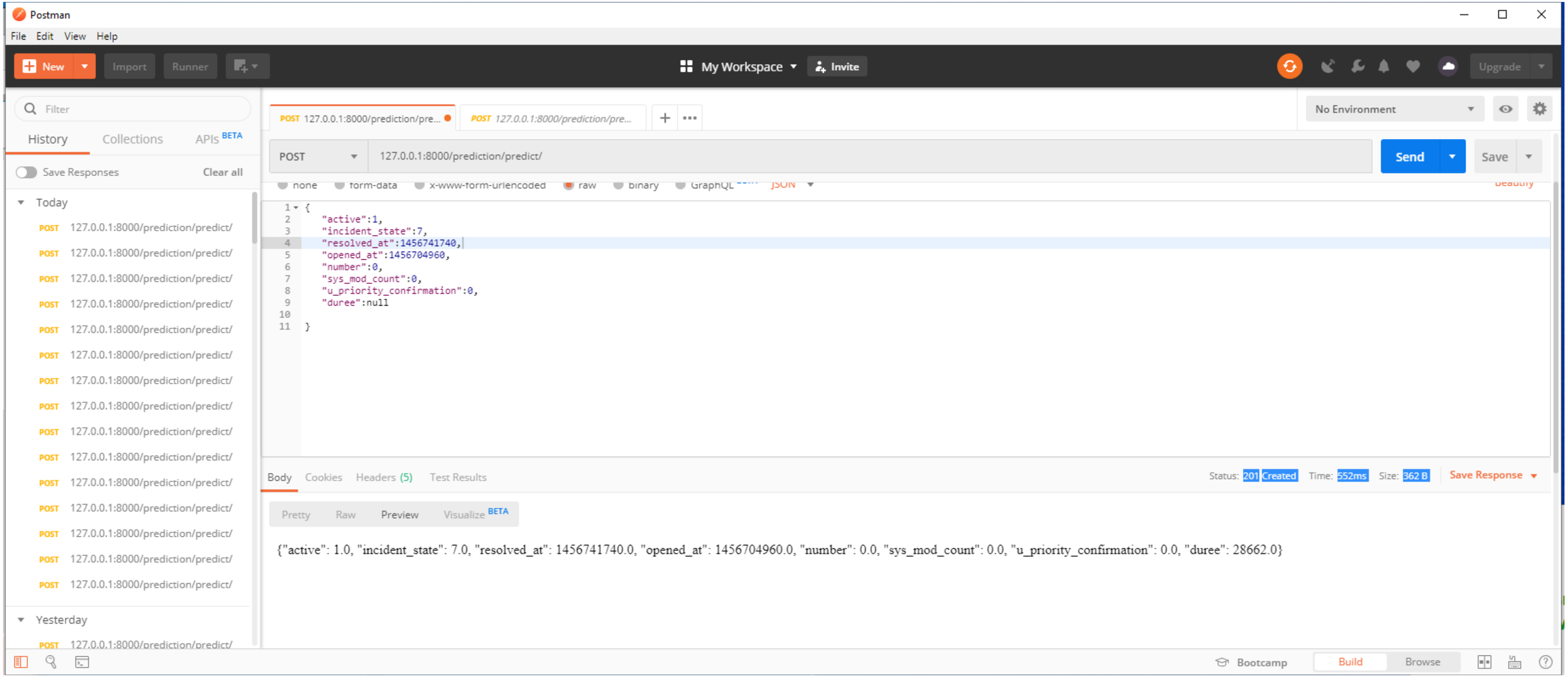
Mon choix s'est donc porté l'algorithme de régression RandomForest car cet algorithme à le score de performance le plus élevé

Mise en place de L'API

Pour la mise en place de l'API , j'ai dû utilisé Joblib pour pouvoir enregistrer le modèle(RandomForestRegressor) ainsi que le scaler.

Mon API se base principalement sur le TD8 disponible sur le github de Luc Bertin.

L'API fonctionnelle



L'API

On peut voir dans la précédente diapositive que l'api fonctionne car elle nous renvoie bien une prédiction de duree pour la résolution de l'incident.

Conclusion

Ce projet était très intéressant malgré le fait que la distance entre le cours vu et le projet est assez important.
Cependant j'ai réussi à réaliser plusieurs prédictions sous différents algorithme de régression avec une performance qui n'est pas optimale mais convenable.
L'api est fonctionnelle est retourne bien des prédictions en fonction du modèle de régression choisi.
Je voulais vous remercier pour ce projet , qui malgré la difficulté à été très enrichissant.