Projet Python For Data Analysis

Sujet: Incident management process enriched event log Data Set

Objectif: Prédire le temps restant avant complétion

Introduction

- Etat des lieux : Il s'agit d'un journal des événements d'un processus de gestion des incidents extrait des données recueillies à partir du système d'audit d'une instance de la plateforme ServiceNowTM utilisée par une société informatique. Le journal d'événements est enrichi de données chargées à partir d'une base de données relationnelle sous-jacente à un système d'information correspondant, conscient des processus. Les informations ont été rendues anonymes pour des raisons de confidentialité.
- Compréhension du problème: Le sujet de ce projet est la création d’une API basé sur une algorithme de machine learning permettant la prédiction de la durée avant résolution d’un incident lors de l’apparition de celui-ci.

I Exploration des données

Information sur les attributs

1. number : identifiant de l'incident (24 918 valeurs différentes) ;
2. Incidentstate: huit niveaux contrôlant les transitions du processus de gestion des incidents entre l'ouverture et la fermeture du dossier ;
3. active : attribut booléen qui indique si le dossier est actif ou fermé/annulé ;
4. Reassignment_count : nombre de fois que l'incident a entraîné un changement de groupe ou d'analystes de soutien ;
5. reopen_count : nombre de fois que la résolution de l'incident a été rejetée par l'appelant ;
6. sys_mod_count : nombre de mises à jour de l'incident jusqu'à ce moment ;
7. made_sla : attribut booléen qui indique si l'incident a dépassé l'ANS cible ;
8. caller_id : identifiant de l'utilisateur affecté ;
9. opened_by : identificateur de l'utilisateur qui a signalé l'incident ;
10. opened_at : date et heure d'ouverture de l'utilisateur ayant signalé l'incident ;
11. sys_created_by : identificateur de l'utilisateur qui a enregistré l'incident ;
12. sys_created_at : date et heure de création du système d'incident ;
13. sys_updated_by : identifiant de l'utilisateur qui a mis à jour l'incident et généré l'enregistrement de journal actuel ;
14. sys_updated_at : date et heure de mise à jour du système d'incidents ;
15. contact_type : attribut catégorique qui indique par quel moyen l'incident a été signalé ;
16. location : identifiant de l'emplacement du lieu affecté ;
17. category : description de premier niveau du service affecté ;
18. subcategory : description de deuxième niveau du service affecté (liée à la description de premier niveau, c'est-à-dire à la catégorie) ;
19. u_symptôme : description de la perception de l'utilisateur quant à la disponibilité du service ;
20. cmdb_ci : identifiant (élément de confirmation) utilisé pour signaler l'élément affecté (non obligatoire) ;
21. impact : description de l'impact causé par l'incident;
22. urgency : description de l'urgence signalée par l'utilisateur pour la résolution de l'incident ;
23. priority : calculée par le système sur la base de l'impact et de l'urgence ;
24. assignment_group : identificateur du groupe de soutien en charge de l'incident ;
25. assigned_to : identifiant de l'utilisateur en charge de l'incident ;
26. knowledge : attribut booléen qui indique si un document de la base de connaissances a été utilisé pour résoudre l'incident ;
27. u_priority_confirmation : attribut booléen qui indique si le champ de priorité a été vérifié deux fois ;
28. notify : attribut catégorique qui indique si des notifications ont été générées pour l'incident ;
29. problem_id : identificateur du problème associé à l'incident ;
30. rfc (demande de changement) : identificateur de la demande de changement associée à l'incident ;
31. vendor : identificateur du fournisseur en charge de l'incident ;
32. caused_by : identificateur du RFC responsable de l'incident ;
33. close_code : identificateur de la résolution de l'incident ;
34. resolved_by : identificateur de l'utilisateur qui a résolu l'incident ;
35. resolved_at : date et heure de la résolution de l'incident par l'utilisateur (variable dépendante) ;
36. closed_at : date et heure de fermeture de l'utilisateur de l'incident (variable dépendante)

Exploration des données

- On observe:
  - Que les valeurs absentes ont été remplis par des points d’interrogations,
  - De nombreuses dates et variables catégorielles,
  - Et des variables booléennes et numériques.

number	INC0000045	INC0000045	INC0000045	INC0000045	INC0000047
incident_state	New	Resolved	Resolved	Closed	New
active	True	True	True	False	True
reassignment_count	0	0	0	0	0
reopen_count	0	0	0	0	0
sys_mod_count	0	2	3	4	0
made_sla	True	True	True	True	True
caller_id	Caller 2403	Caller 2403	Caller 2403	Caller 2403	Caller 2403
opened_by	Opened by 8	Opened by 8	Opened by 8	Opened by 8	Opened by 397
opened_at	29/2/2016 01:16	29/2/2016 01:16	29/2/2016 01:16	29/2/2016 01:16	29/2/2016 04:40
sys_created_by	Created by 6	Created by 6	Created by 6	Created by 6	Created by 171
sys_created_at	29/2/2016 01:23	29/2/2016 01:23	29/2/2016 01:23	29/2/2016 01:23	29/2/2016 04:57
sys_updated_by	Updated by 21	Updated by 642	Updated by 804	Updated by 908	Updated by 746
sys_updated_at	29/2/2016 01:23	29/2/2016 08:53	29/2/2016 11:29	5/3/2016 12:00	29/2/2016 04:57
contact_type	Phone	Phone	Phone	Phone	Phone
location	Location 143	Location 143	Location 143	Location 143	Location 165
category	Category 55	Category 55	Category 55	Category 55	Category 40
subcategory	Subcategory 170	Subcategory 170	Subcategory 170	Subcategory 170	Subcategory 215
u_symptom	Symptom 72	Symptom 72	Symptom 72	Symptom 72	Symptom 471
cmdb_ci	?	?	?	?	?
impact	2 - Medium	2 - Medium	2 - Medium	2 - Medium	2 - Medium
urgency	2 - Medium	2 - Medium	2 - Medium	2 - Medium	2 - Medium
priority	3 - Moderate	3 - Moderate	3 - Moderate	3 - Moderate	3 - Moderate
assignment_group	Group 56	Group 56	Group 56	Group 56	Group 70
assigned_to	?	?	?	?	Resolver 89
knowledge	True	True	True	True	True
u_priority_confirmation	False	False	False	False	False
notify	Do Not Notify	Do Not Notify	Do Not Notify	Do Not Notify	Do Not Notify
problem_id	?	?	?	?	?
rfc	?	?	?	?	?
vendor	?	?	?	?	?

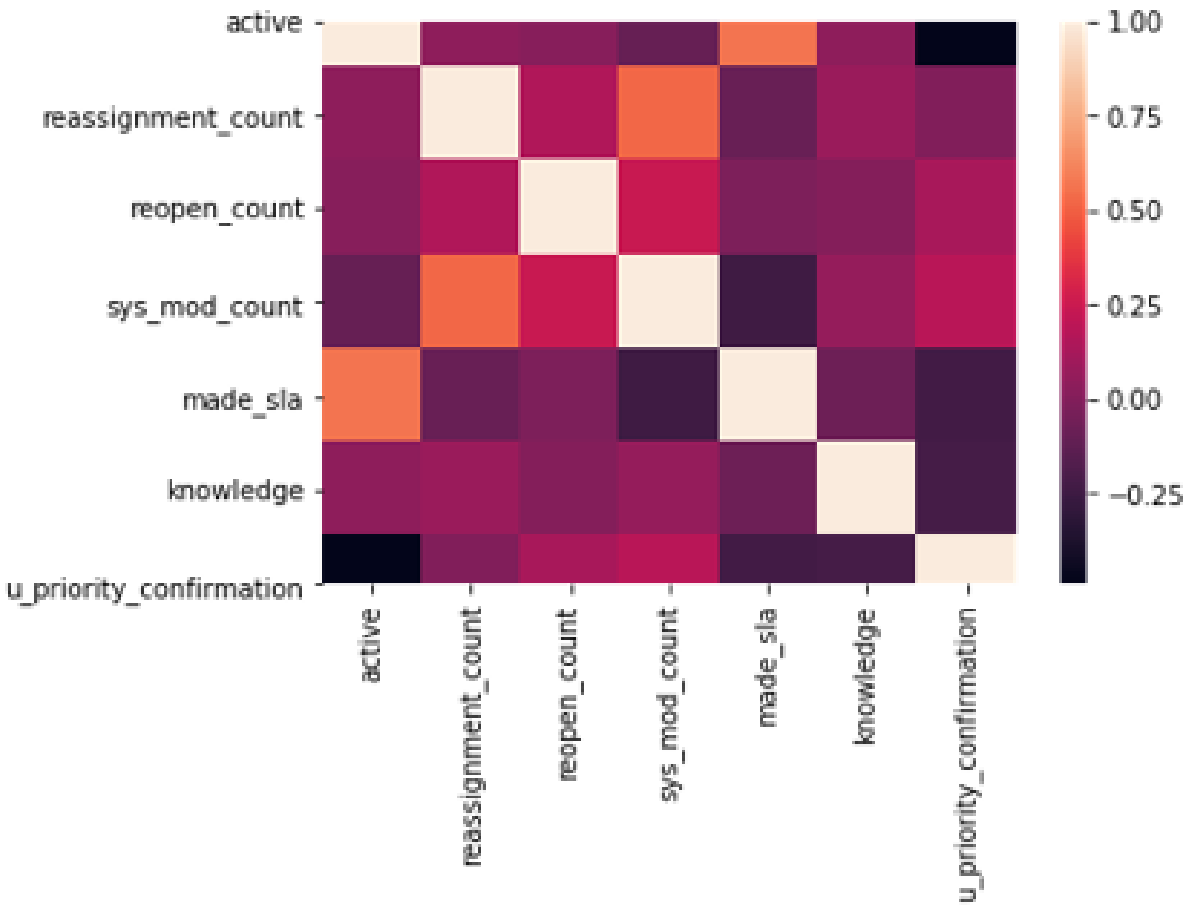
Statistiques des données numériques

- On observe:
  - La variance est faible,
  - Beaucoup de données sont nuls,
  - Quelques données avec une valeurs importantes augmentant la moyenne

	reassignment_count	reopen_count	sys_mod_count
count	141712.000000	141712.000000	141712.000000
mean	1.104197	0.021918	5.080946
std	1.734673	0.207302	7.680652
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	1.000000
50%	1.000000	0.000000	3.000000
75%	1.000000	0.000000	6.000000
max	27.000000	8.000000	129.000000

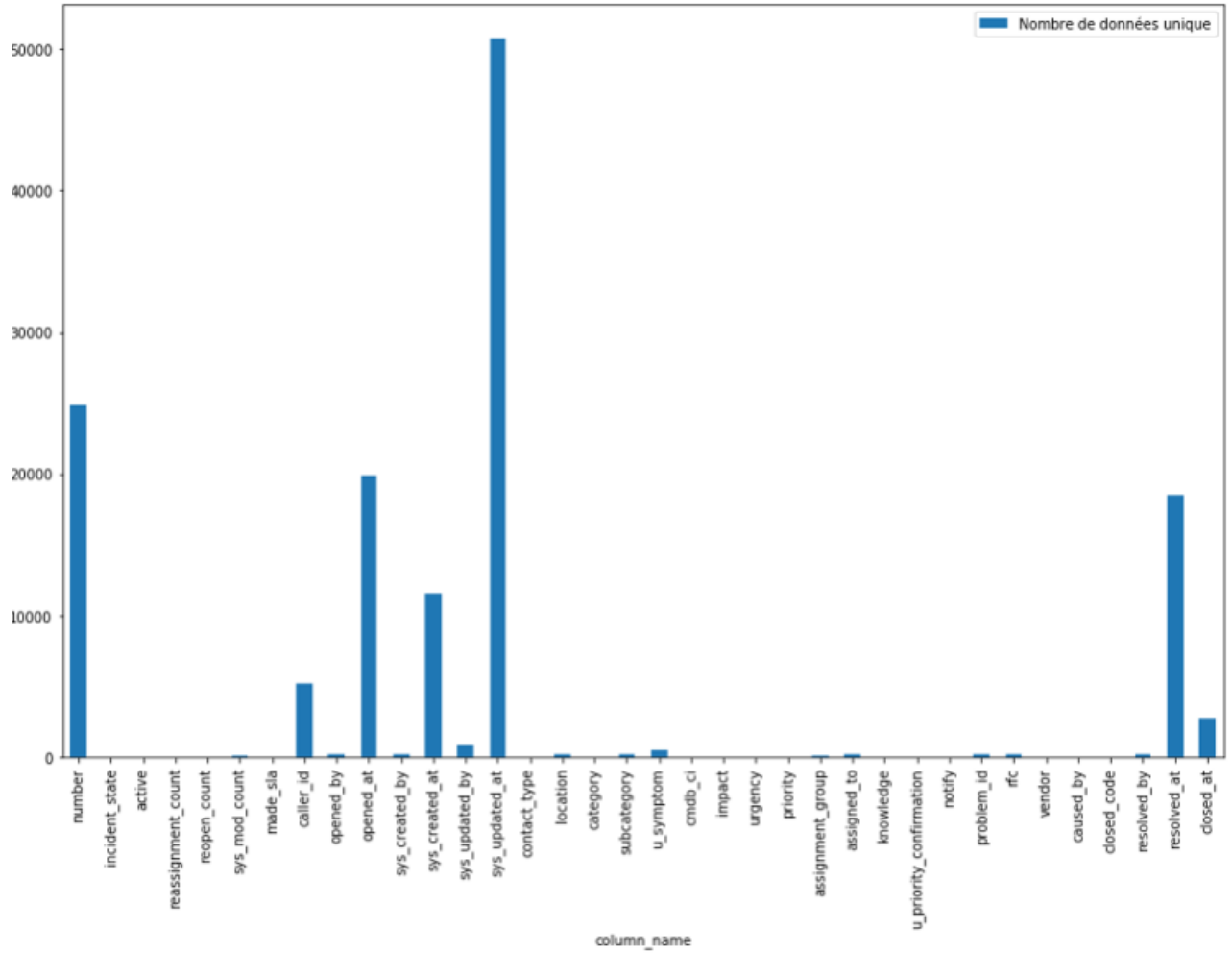
HeatMap de Corrélation

- On observe:
  - Corrélation importante entre made_sla et active
  - Corrélation importante entre sys_mod_count et reassignement_count
  - Les autres variables ont des correlations faibles voir inexistante (active et u_priority)



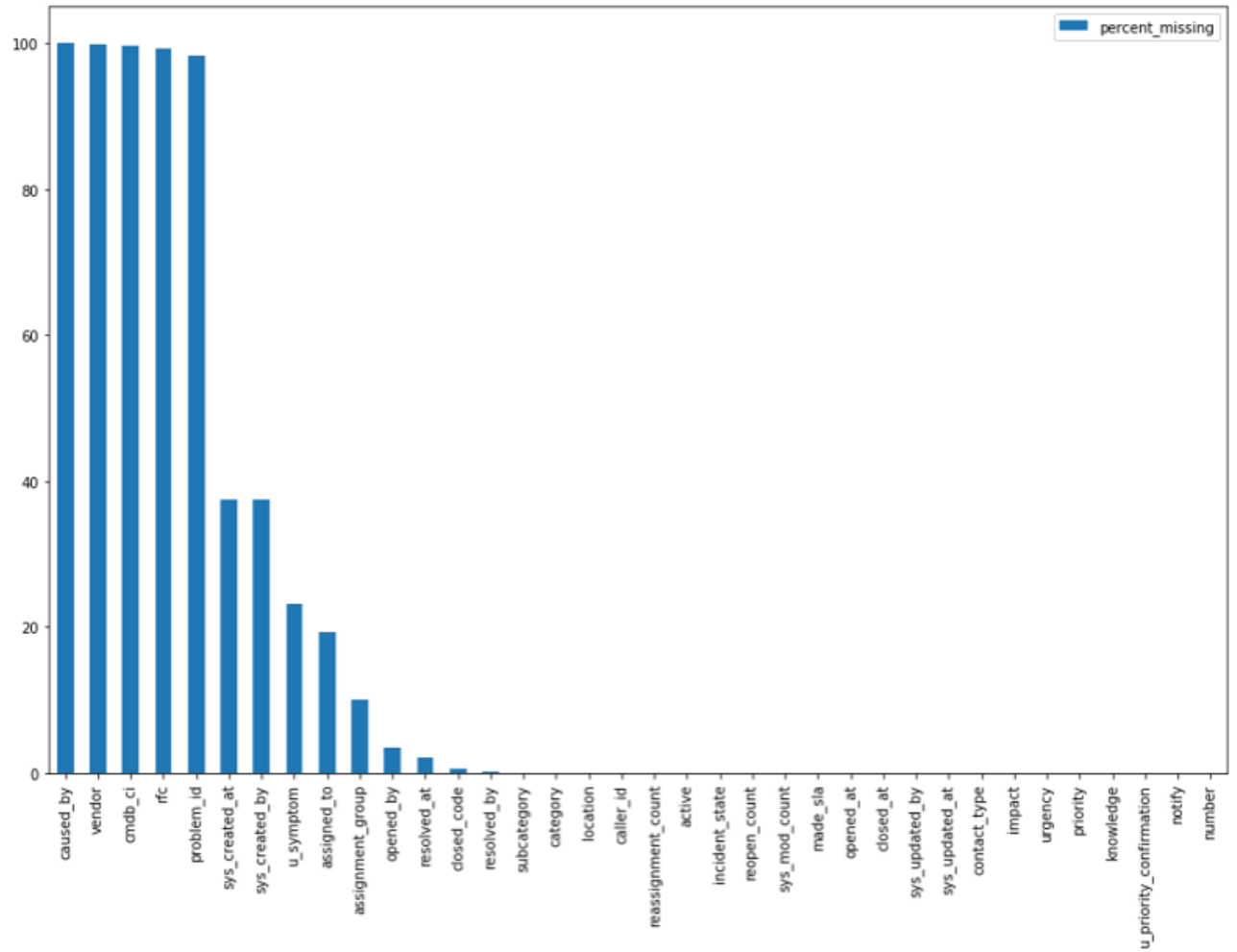
Nombre de valeurs uniques

- On observe:
  - 24 918 id unique sur 140 000 données,
  - On observe une faible diversité de valeurs uniques autre que dans les variables de date
- Action:
  - Nous devons donc faire une agrégation ou vérifier après préprocessing le nombre de redondance pour s'assurer de ne pas biaiser le modèle



Pourcentage valeur manquantes

- On observe:
  - 5 variables avec un pourcentage de valeurs manquantes supérieur à 98%
  - Ainsi que 9 autres variables avec un pourcentage de valeurs manquantes inférieures à 50%
- Action:
  - Suppression de la colonne caused_by car corrélé à rfc (cf diapo2)
  - Conversion des 4 autres colonnes en un booléen présence ou absence de l'information (afin de ne pas perdre trop d'information)
  - Dans les 9 autres variables remplissages des NA par une nouvelle catégorie « Non renseigner » (afin de ne pas perdre ou biaiser le modèle)



Suppression des lignes non closes ou résolus

- On supprime ces lignes afin d'avoir uniquement des valeurs permettant la création puis l'analyse/prédictions de la **cible**
- On choisi de faire la cible sur la résolution et non la clôture par le client car celui-ci peut oublier de clôturer son incident alors qu'il a été résolu.

Suppressions des lignes autres que closes ou resolved

```
data = data[data['incident_state'].map(lambda x: str(x)!="Active" and
                                       str(x)!="New" and str(x)!='Awaiting User Info' and str(x)!='Awaiting Vendor' and
                                       str(x)!='Awaiting Problem' and str(x)!='Awaiting Evidence' and str(x)!='-100')]
```

Mise au bon format de date et création de la cible time_before_completion

```
data['closed_at']   = pd.to_datetime(data['closed_at'], format='%d/%m/%Y %H:%M')
data['resolved_at']= pd.to_datetime(data['resolved_at'], format='%d/%m/%Y %H:%M')
data['opened_at']   = pd.to_datetime(data['opened_at'], format='%d/%m/%Y %H:%M')
data['sys_created_at']= pd.to_datetime(data['sys_created_at'], format='%d/%m/%Y %H:%M')
data['sys_updated_at']   = pd.to_datetime(data['sys_updated_at'], format='%d/%m/%Y %H:%M')
data['time_before_completion']=data['resolved_at'].sub(data['opened_at'], axis=0).dt.days
```

Modification des variables de temps

- On supprime ces colonnes pouvant faussé le modèles en effet les information sur la clôture, la résolution, les mises à jours etc… ne seront pas fournis lors d'une véritable prédiction.
- On extrait les informations l'ouverture de l'incident au format float64 puis on supprime la colonne afin de pouvoir faire entre ces informations dans un modèles

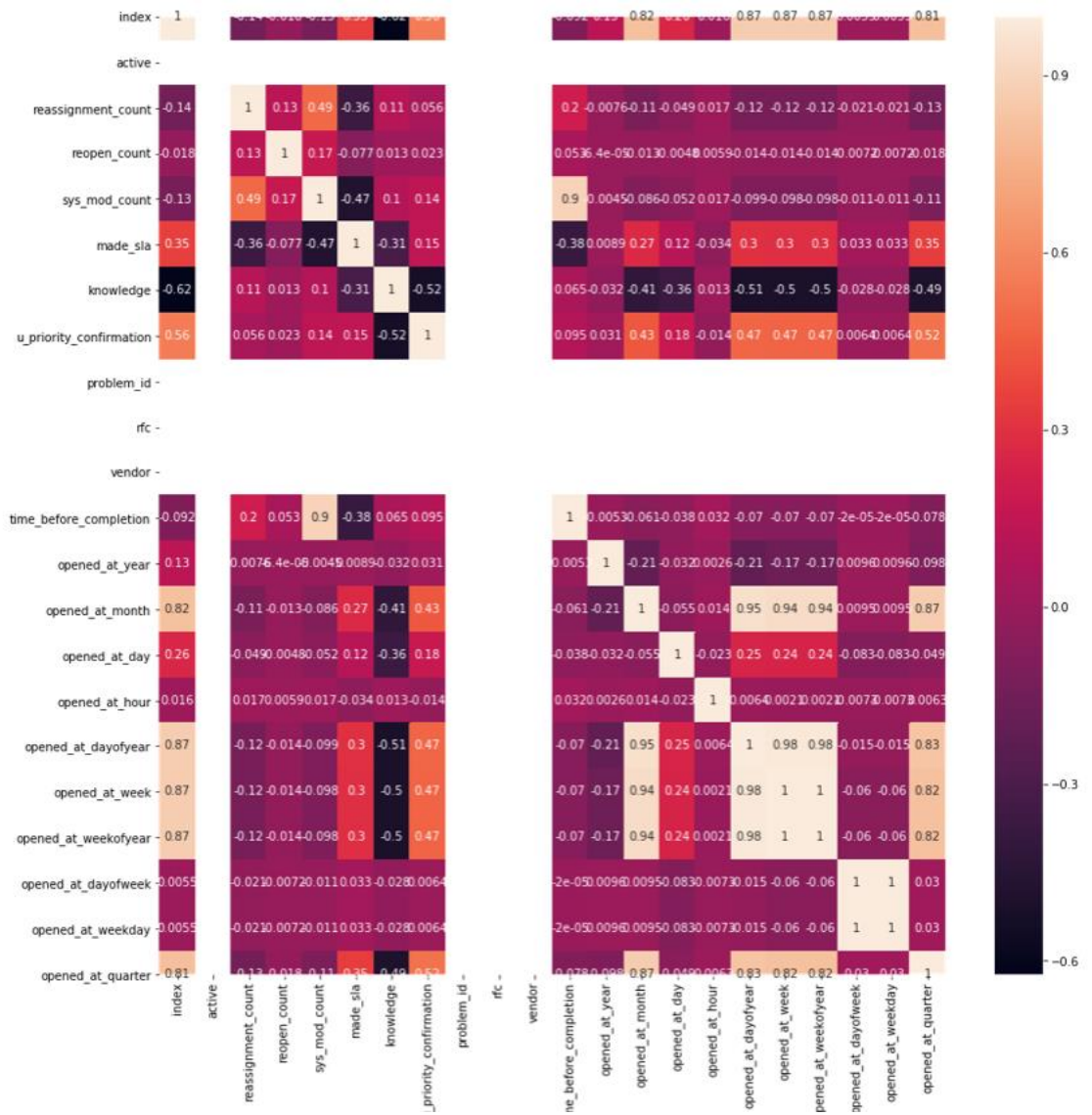
```
data=data.drop('closed_at',axis=1)
data=data.drop('resolved_at',axis=1)
data=data.drop('sys_created_at',axis=1)
data=data.drop('sys_updated_at',axis=1)

# On supprime ces quatres colonnes afin de ne pas donner la réponse à l'algorithme

data["opened_at_year"       ]=data["opened_at"].dt.year
data["opened_at_month"      ]=data["opened_at"].dt.month
data["opened_at_day"        ]=data["opened_at"].dt.day
data["opened_at_hour"       ]=data["opened_at"].dt.hour
data["opened_at_dayofyear"  ]=data["opened_at"].dt.dayofyear
data["opened_at_week"       ]=data["opened_at"].dt.week
data["opened_at_weekofyear" ]=data["opened_at"].dt.weekofyear
data["opened_at_dayofweek"  ]=data["opened_at"].dt.dayofweek
data["opened_at_weekday"    ]=data["opened_at"].dt.weekday
data["opened_at_quarter"    ]=data["opened_at"].dt.quarter
data=data.drop('opened_at',axis=1)
```

Matrice de corrélation et cible

- On observe:
  - Corrélation importante la cible et sys_mod_count
  - Corrélation importante entre la cible et reassignement_count
  - Les autres variables ont de faible corrélation avec la cible



Test de khi2 corrélation variable catégorielle et cible

- On observe:
  - Des variables avec une indépendance complète avec la cible et deux variables fortement décorrélé (number et notify)
- Action:
  - Suppression des colonnes ayant une indépendance complète ou importante avec la cible

```
number P-value:  0.47249080491912276
incident_state P-value:  1.0
caller_id P-value:  1.0
opened_by P-value:  0.0
sys_created_by P-value:  0.0
sys_updated_by P-value:  1.0
contact_type P-value:  1.0
location P-value:  1.0
category P-value:  0.0
subcategory P-value:  0.0
u_symptom P-value:  0.0
cmdb_ci P-value:  0.257942747556982 77
impact P-value:  5.565447492624639e-34
urgency P-value:  1.065115867404221 1e-43
priority P-value:  5.38440611679639e-38
assignment_group P-value:  0.0
assigned_to P-value:  0.0
notify P-value:  0.849281893116857
closed_code P-value:  0.0
resolved_by P-value:  0.0
```

Test de khi2 corrélation variable catégorielle et cible

- Création des dummies afin de pouvoir implémenter les variables catégorielles dans le modèle sans créer de biais

```
str_cols = data.columns[data.dtypes=='category']
df=pd.get_dummies(data[str_cols])
frame=[data,df]
data=pd.concat(frame, axis=1, sort=False)
```

II Modèle

- Utilisation du RandomForestRegressor pour son faible risque d'overfitting grâce au bootstrap et la possibilité de faire une cross validation ainsi que pour sa relative rapidité de mise en place et de paramétrage via un gridsearch.
- Résultat :
  - MSE = 22,56, score = 0,96 (après optimisation via GridSearch)
  - MSE= 96 , score = 0,70 (avant optimisation)
- On observe aussi que la majorité de la prédiction est faite grâce à seulement trois variables (~91%):

made_sla	0.028299
reassignment_count	0.033182
sys_mod_count	0.860346

III API

- Utilisation d’une api Django permettant la réalisation d’une prédiction lors de l’envoie des informations de bases lors de l’incident.

Conclusion

- Problèmes rencontrer: Manque de puissance ordinateur empêchant la réalisation et le paramétrage de plusieurs modèles complexes.

- Qualité des prédiction: Nous avons des prédictions de qualités supérieur au hasard avec un mse de 22 alors que la variance de la cible est de 22.