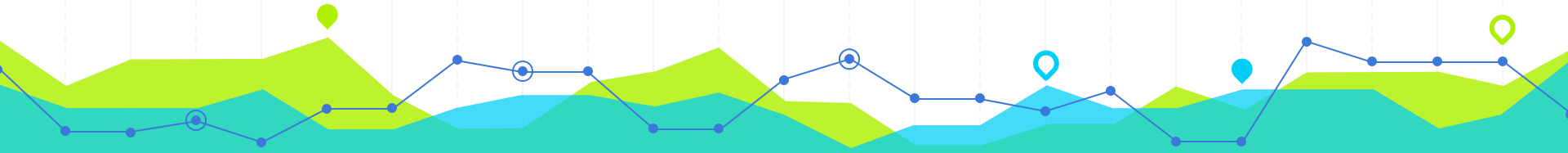


Python for Data Analysis : Project

VO Joakim

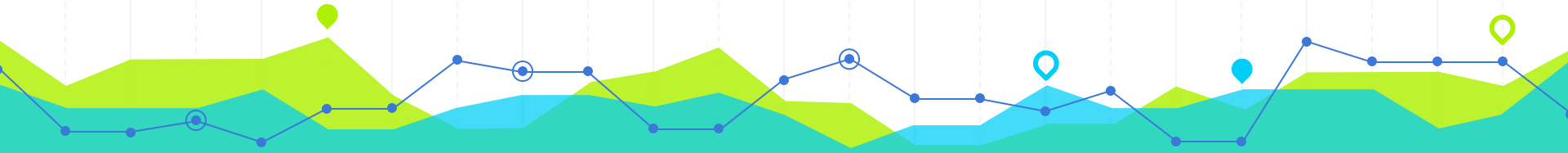
A5 IBO : Option Cybersécurité

Promotion 2020

PLAN

1. Contexte et enjeux du projet
2. Exploration, préparation et nettoyage des données
3. Analyse de la target et corrélation
4. Algorithmes de régression/modèles & parametres
5. Bilan de projet et cours

Ce powerpoint est un résumé explicatif simplifié du code github entier. Certaines parties du code (algorithmes et graphiques), n'ont donc pas étés mentionnés dans ce ppt.



Contexte et enjeux du sujet

1

Incident management process enriched event log Data Set

Contexte

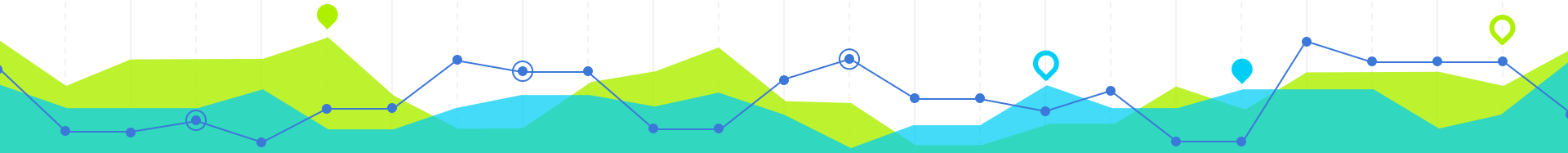
Une base de données des log incidents d'un système d'audit automatisé utilisé par une entreprise IT. Celle-ci est composé de 141,712 events (24,918 incidents) pour 36 attributs (1 case identifier, 1 state identifier, 32 descriptive attributes, 2 dependent variables)

Enjeu

L'objectif est de prédire le temps de résolution d'un incident ou événement, c'est à dire entre sa date d'ouverture du ticket, et sa résolution. Des algorithmes de machine learning (régression) seront utilisés pour créer et évaluer un modèle de prédiction le plus précis possible

Data Set Characteristics:	Multivariate, Sequential	**Number of Instances:**	141712	**Area:**	Business
Attribute Characteristics:	Integer	**Number of Attributes:**	36	**Date Donated**	2019-07-14
Associated Tasks:	Regression, Clustering	**Missing Values?**	Yes	**Number of Web Hits:**	14559

https://archive.ics.uci.edu/ml/datasets/Incident+management+process+enriched+event+log



Exploration et nettoyage des données

2

Premiere exploration et constatation

Quelques fonctions et méthodes utilisées:

df.isnull().sum()

df.info()

df.shape

df.head()

df[‘attribut’].value_counts()
df.describe()

etc...

A priori, il n’y a pas de valeurs nulles, mais beaucoup de données semblent manquantes, et certains attributs semblent peu important. On remarque également quelques valeurs erronées

Il y a beaucoup de données de type categorical, mais inutilisables dans la condition actuelle (sous forme de string), qu’il faudra transformer/convertir pour qu’elles soient utilisées correctement dans la régression.

Valeurs manquantes

- Les valeurs manquantes sont caractérisées par ‘?’ . Les columnes problem_id, rfc, vendor, caused_by et cmdb_ci ont plus de 95% de ces données manquantes.

- Les matrices de corrélations et régressions faites par la suite ont montrées en tests que ces colonnes n’avaient en effet pas d’impact sur la prédiction, donc on les a retirées.

- D’autres colonnes contenaient un certain nombre de ‘?’ (autour de 2000 lignes), mais nous avons cependant décidé de les garder

Valeurs erronées

- Il arrive que certaines colonnes ont des fausses données, alors on supprime complètement les lignes si celles ci sont peu nombreuses (image à droite)

```
Out[201]: Active                 30372
          New                    29631
          Resolved               20835
          Closed                 20303
          Awaiting User Info     10502
          Awaiting Vendor          535
          Awaiting Problem         355
          Awaiting Evidence         36
          -100                       3
```

- En créant une colonne nommée ‘duration’ qui correspond au temps de résolution d’un incident en jour (notre futur target), avec la date d’ouverture et date de fermeture, on remarque des temps négatifs. Il n’est pas possible qu’un incident ticket soit résolu avant que le ticket fut ouvert (probablement erreur de frappe quelconque), donc on supprime ces lignes (photo en bas). A gauche correspond a la date d’ouverture, et à droite la date de résolution.

0	True	Caller 2403	Opened by 397	2016-02-29 04:40:00	...	Do Not Notify	?	?	?	?	code 5	Resolved by 81	2016-01-03 09:52:00	6/3/2016 10:00	-56.8

Valeurs et attributs pièges

Pour ne pas fausser le modèle

Après avoir créé la target ‘duration’ qui correspond à (resolved_at - opened_at), il faut s’assurer de bien supprimer ces colonnes par la suite, y compris celles liées à la cloture/fin d’un incident, car on suppose à l’avance qu’on ne connaît aucune de ces données pendant qu’un incident est en cours. La résolution étant la phase finale de l’incident, alors closed_at ne nous intéresse pas à priori. En revanche si nous le gardons, on sait à l’avance que cet attribut va être extrêmement corrélé avec les dates d’ouverture, date de résolution, et duration. Non seulement cet attribut n’est pas censé être connu en situation d’incident (incident non terminé), mais il indique par lui même le temps approximatif qu’un incident met pour être résolu (cela serait considéré comme de la triche). On doit donc le supprimer. Les attributs concernant qui à fermé le ticket etc.. sont également supprimés.

On rappelle que l’objectif du projet est d’estimer le temps de résolution d’un incident pendant que celui-ci est en cours. Donc toute les données liées à la clôture/fin de résolution sont supprimées

Conversion des données

- Pour que les données soient acceptées par scikit-learn, il faut convertir les données de types catégories string vers des catégories int

- On prend bien soin de ne pas confondre les int categorical, avec les int classiques (par exemple le nombre de jour ‘duration’ , le nombre de fois qu’un ticket a été réouvert/réassigné etc...

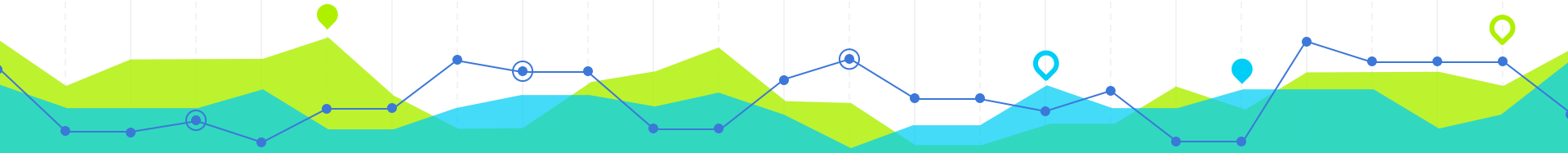
- Voici un exemple de fonction qui aide à convertir automatiquement

```
Entrée [225]: df['made_sla'] = df['made_sla'].astype('category')
              df['made_sla'] = df['made_sla'].cat.codes

Entrée [226]: df['caller_id'] = df['caller_id'].astype('category')
              df['caller_id'] = df['caller_id'].cat.codes
```

sys_mod_count	made_sla	caller_id	opened_by	sys_create
0	True	Caller 2403	Opened by 8	Created
2	True	Caller 2403	Opened by 8	Created
3	True	Caller 2403	Opened by 8	Created
4	True	Caller 2403	Opened by 8	Created
		Caller	Opened by	

sys_mod_count	made_sla	caller_id	opened_by	sys_created_by
0	1	1357	191	146
2	1	1357	191	146
3	1	1357	191	146
4	1	1357	191	146
0	1	2684	33	164



Analyse de la target et corrélations

3

‘A PICTURE IS WORTH A THOUSAND WORDS’

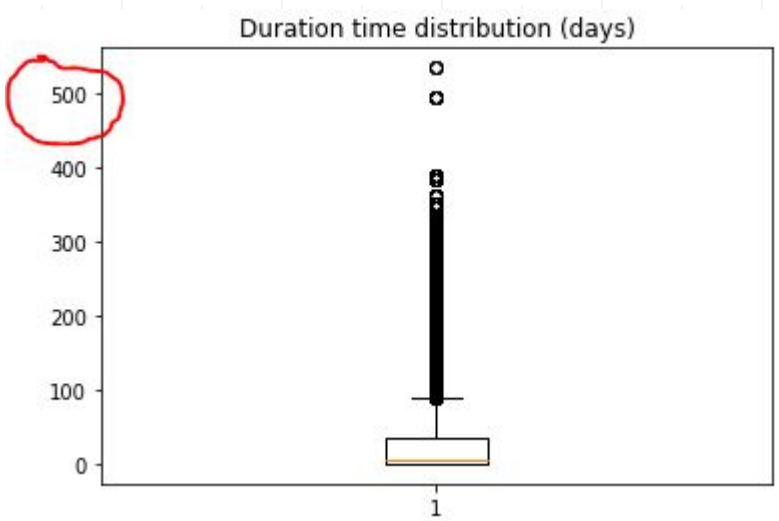
Utilisation de graphiques permettant d’estimer instinctivement et rapidement un problème et ses attentes en data science

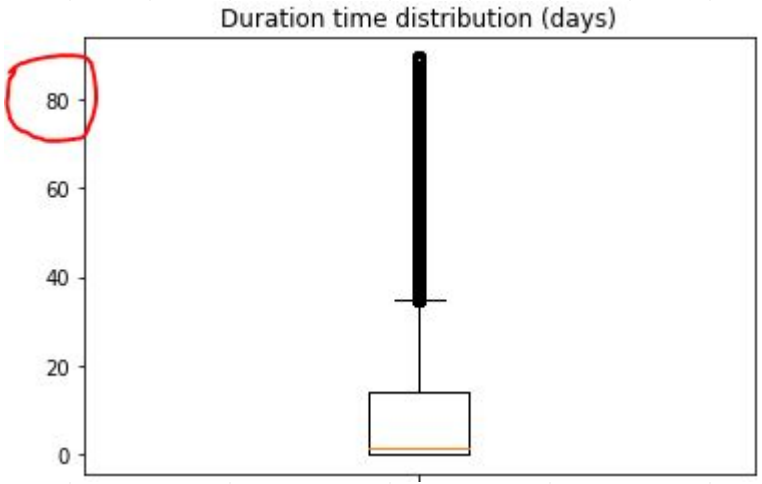
Usage des librairies matplotlib et seaborn

Target ‘duration’

- La cible ‘duration’ est définie en nombre de jour, caractérisée par un float

- On supprime une majorité des outliers en utilisant les fonction de scipy.stats. Au final, nous sommes passés de 141712 rows à 95167 rows après tout les nettoyages précédents (partie 2 du plan inclus)

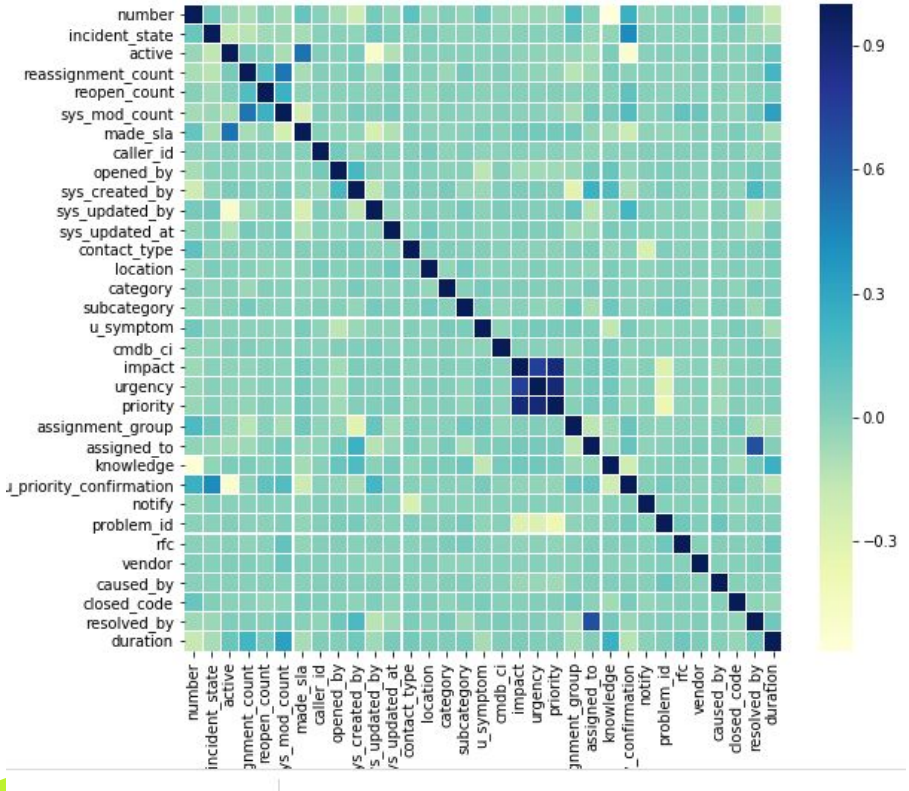




Matrice de corrélation globale

- A première vue, il semblerait que peu de variables soient fortement corrélées entre elles (les cubes étant en majorité tous très clair).

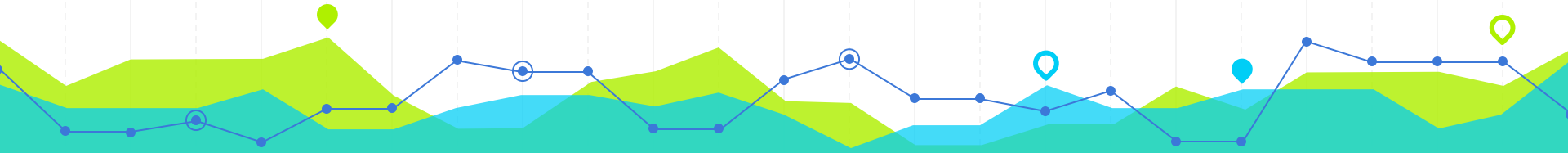
  On peut donc s'attendre qu'il sera difficile de construire un modèle de machine learning performant



Matrice de corrélation sur target

- Les correlation triées par ordre décroissant avec la target ‘duration’ montrent que peu d’entre-elles sont en lien avec celle-ci. Seulement 3 attributs ont une corrélation supérieure à 10% avec la target.

On s’attend donc à des difficultés sur le modèle de régression...



Algorithmes de régression et modèles

4

Algorithmes utilisés

- Régression linéaire simple : 25,3%, MSE=17 jours
- Cross-Validation avec cv = 5 et 10
- Lasso L1
- ElasticNet L1
- DecisionTreeRegressor: MSE=4 jours; 38,9% → Meilleur algo
- Ridge L2
- MLP regressor : MSE=9,8 jours
- RandomForest : MSE=11,41 jours

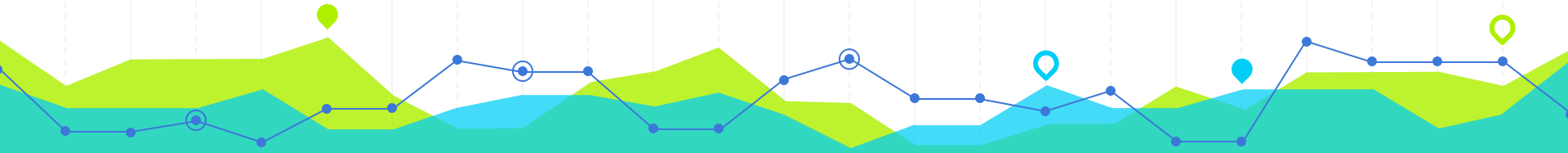
Tuning des hyperparametres

- On utilise RandomForestRegressor et GridSearchCV pour trouver les meilleurs parametres

```
bootstrap = [True, False]

random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}

rf = RandomForestRegressor()
rf_random = RandomizedSearchCV(estimator = rf, param_distributions
```



5

Bilan de projet et cours

Avis global

- Bien que les scores des modèles furent plutôt mauvais (au mieux environ 40%), ce dataset fut très intéressant de par les problématiques engendrée dans le contexte des log incidents. Instinctivement, prédire le temps de résolution d'un problème pendant qu'on le traite est assez complexe, car on peut rencontrer des nouveaux problèmes chaque jour, les technologies changent et évoluent. La base de donnée n'est donc pas homogène partout sur des particularités techniques qui ne sont pas visibles. Le nettoyage et préparation des données est sans surprise la partie la plus longue.

- Pour conclure, cette matière de Python For Data Analysis fut assez périlleuse pour ma part, notamment du fait que je suis dans la filière cybersécurité, et que ce cours fut le seul cours de data science avec ML pur que j'ai eu dans mon cursus. La partie pandas fut abordable, cependant la partie Machine learning était compliquée à comprendre, avec trop peu de temps de l'assimiler en seulement 2 séances pour des non initiés. Mes lacunes sont donc sur la partie ML. Cela ouvre cependant des ouvertures pour ma part sur des idées de machine learning appliqué en cybersécurité. Merci pour votre enseignement

FIN

joakim.vo@edu.devinci.fr