TP : data mining et machine learning avec Orange

Ce TP donne lieu à un rendu noté !

— Date limite de rendu : 17/12/21

— Rapport individuel à envoyer à pierre-francois.gimenez@centralesupelec.fr

1 Installation d'Orange

Pour installer Orange, allez sur `https://orangedatamining.com/download/`

Sur Linux :

— peut-être aurez-vous besoin d'exécuter : `sudo apt install python3-pyqt5.qtsvg`

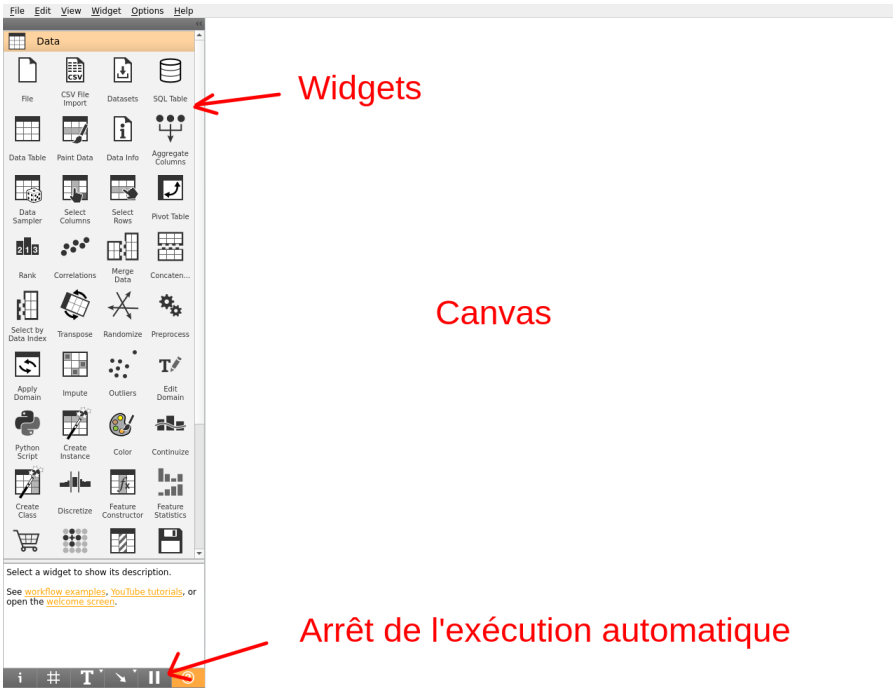
— pour lancer Orange : `orange-canvas` ou `python3 -m Orange.canvas`

2 Présentation d'Orange

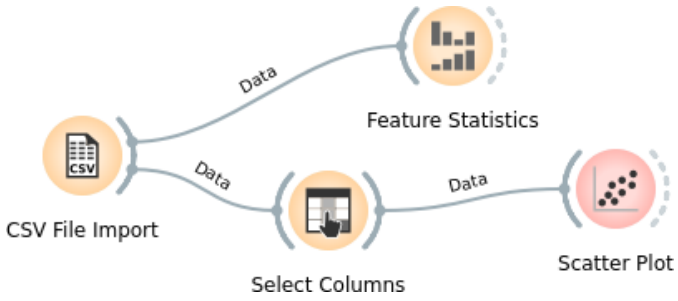
Orange est un logiciel gratuit et open source de data mining. Il s'appuie sur une représentation graphique ("workflow") des différentes transformations à appliquer à des données. Nous ne verrons qu'une petite partie du logiciel ; de plus, des extensions permettent d'avoir des outils spécialisés (traitement d'images, de texte, de séries temporelles, etc.).

Par défaut, toutes les opérations sont faites automatiquement en temps réel, c'est-à-dire que toute modification du workflow entraîne un recalcul immédiat des sorties.

Structure de l'interface graphique :



Dans Orange, les nœuds du workflow sont des unités de calcul (pour charger un fichier, apprendre un modèle, l'afficher, etc.). Pour ajouter un nœud, il suffit de le glisser depuis la colonne des widgets. Les arêtes contrôlent le flot de données. On peut double-cliquer dessus pour avoir plus d'options. Les canaux d'entrées (input) sont sur le côté gauche des nœuds, les canaux de sorties (output) sont sur le côté droit. Par exemple :



Les widgets sont rangés en catégories :

— "Data" : pour charger un modèle, faire du pre-processing, sauvegarder les données
— "Visualize" : propose plein de possibilités d'affichage
— "Model" : des modèles de machine learning
— "Evaluate" : évaluer les modèles appris
— "Unsupervised" : des techniques évoluées de traitement des données (clustering, réduction de dimensions...)

La documentation peut être utile ! `https://orangedatamining.com/widget-catalog/`

3 Le jeu de données

On va utiliser le jeu de données "KDD CUP 1999". C'est un jeu de données classique qui était très utilisé en détection d'intrusion. Chaque ligne du jeu de données est une connexion réseau qui peut être soit bénigne soit malveillante. On manipulera seulement un sous-ensemble du jeu de données car il est assez gros. Les attributs sont décrits à la fin de cet énoncé !

Les attaques du jeu de données sont réparties en quatre catégories :

— DOS (déni de service, par exemple du SYN flood)
— Probe (surveillance et sondage, par exemple scan de ports)
— U2R ("user to root", accès non-autorisé depuis un compte local, par exemple via une attaque en mémoire)
— R2L ("remote to local", accès non-autorisé depuis une machine extérieure, par exemple du bruteforce de mot de passe)

Ce jeu de données est pédagogiquement intéressant, mais sachez qu'il est vieux et critiqué.

4 Prise en main

Téléchargez le jeu de données sur Teams ou ici : `https://filesender.renater.fr/?s=download&token=014681bf-29f7-4bdc-b9cf-ff972e099034`

Ouvrez Orange, créez un nouveau projet et ouvrez le fichier de données avec le widget "CSV File Import". Explorez un peu les données avec les widgets "Data Table" (dans "Data"), "Distributions" (dans "Visualize") et "Feature Statistics" (dans "Data").

1. Combien y a-t-il d'instances dans ce jeu de données ? Combien y a-t-il d'attributs ? À votre avis, que signifient le C vert et le N rouge à côté du nom des attributs ? Que représente l'attribut "label" ?

2. Observez la distribution de l'attribut "label". Quelle classe est majoritaire ? Regardez les types d'attaques et leur proportion. Pensez-vous que ce dataset soit représentatif de mesures qu'on pourrait réaliser sur un réseau en 2021 ? Pensez-vous qu'il soit représentatif des attaques actuelles ?

5 Data mining et machine learning

Avec l'outil "Distributions", vous pouvez choisir de diviser les colonnes par l'attribut "Label" (en bas à gauche).

3. Pouvez-vous identifier des attributs dont certaines valeurs vous paraissent très liées à certaines attaques ?

Grâce au widget "Select columns", vous pouvez mettre l'attribut "Label" comme étant un attribut cible (target). Utilisez ensuite le widget "Rank" pour demander à Orange de classer les attributs par ordre de pertinence par rapport à la valeur à prédire (c'est-à-dire par rapport aux différentes attaques).

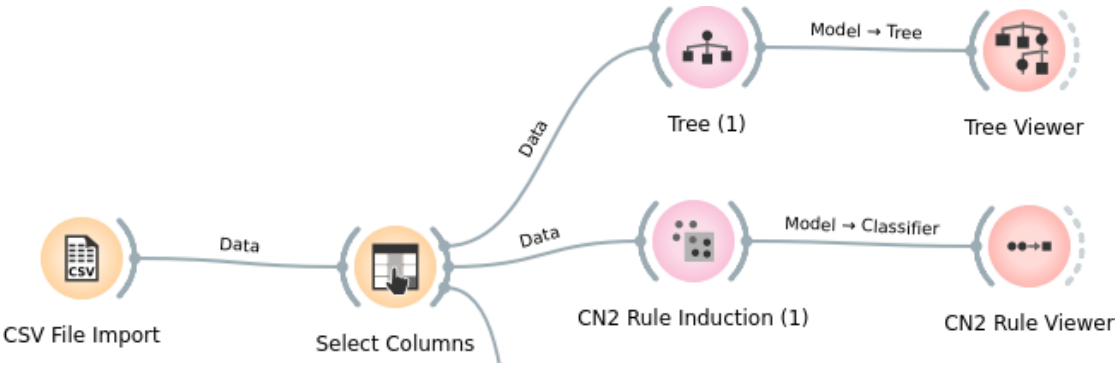
4. Listez les cinq attributs les plus corrélés aux attaques d'après leur "Gini coefficient".

Il n'est pas toujours évident de savoir pourquoi "Rank" indique qu'un attribut est corrélé à une attaque. Prenez l'un des attributs listé dans la question précédente et observez avec le widget "Distribution" en quoi il est effectivement corrélés aux attaques.

5. Quelle attaque précisément semble détectable avec cet attribut ? (Changez d'attribut si vous ne voyez rien.) Ajoutez une capture d'écran à votre rapport.

Vous pourrez certainement constater qu'il est difficile, en observant un seul attribut à la fois, de détecter correctement les attaques. Regardons comment des modèles appris automatiquement se débrouillent. Reproduisez le schéma ci-dessous qui va apprendre un arbre de décision et le visualiser (partie haute du schéma) et apprendre des règles de décision et les visualiser (partie basse du schéma).

Attention ! Le jeu de données est assez gros, donc l'apprentissage peut prendre un certain temps. Si l'application ne répond plus, c'est certainement que l'apprentissage est en cours !



Vous pouvez visualiser les modèles en double-cliquant sur les widgets "Viewer" de votre workflow.

6. Ajoutez à votre rapport deux captures d'écran : une capture pour la visualisation de l'arbre, une autre pour la visualisation des règles de décisions.

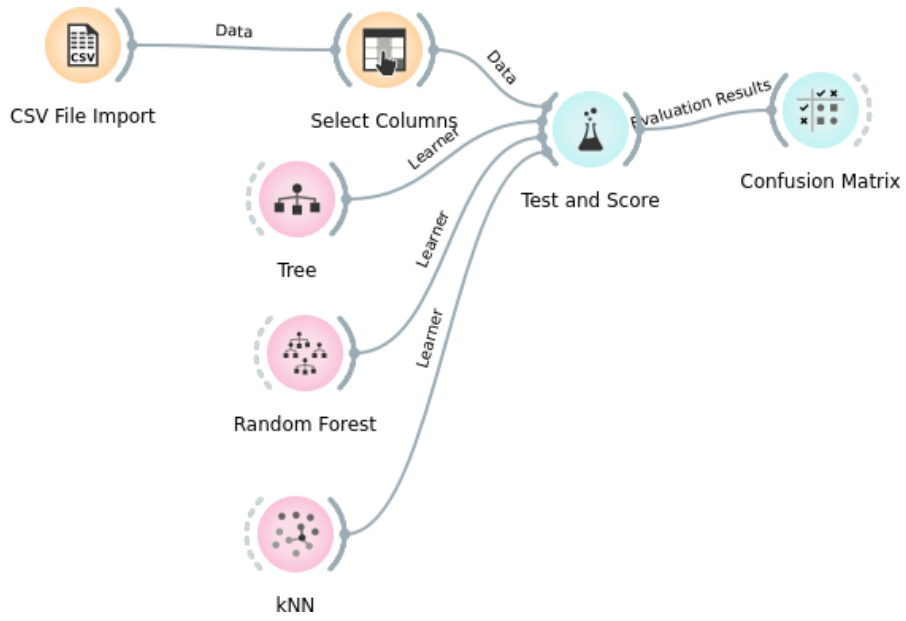
7. Pourriez-vous, à partir de l'arbre de décision appris, donner un exemple d'instance qui serait prédit comme une attaque "Probe" ? Même question avec les règles apprises.

Nous venons d'utiliser l'arbre de décision et les règles de décision comme un moyen de visualiser commet un modèle fonctionne et comment il prend une décision. Nous allons maintenant les évaluer, c'est-à-dire mesurer leur efficacité.

Les widgets de la catégorie "Model" permettent de mettre en œuvre des modèles de machine learning. Essayez les modèles suivants :

— "Tree". *Attention, il faut double-cliquer dessus et décocher "Induce binary tree" dans notre cas !*
— "kNN"
— "Random forest"
— "CN2 Rule Induction"

Utilisez le widget "Test and Score" pour évaluer ces modèles. Pour ce faire, on peut brancher des données et des modèles dans ce widget, comme ci-dessous :



Remarquez également l'utilisation du widget "Confusion matrix" qui permet d'analyser un peu plus finement les erreurs des modèles.

Pour chaque modèle, répondez aux questions suivantes :

8.1. Quel est l'accuracy [1] de ce modèle ?

8.2. Combien y a-t-il de faux positifs ? (instances normales considérées comme attaques) De faux négatifs ? (attaques considérées comme normales)

8.3. En supposant dix connexions légitimes par seconde, combien y aurait-t-il de faux positifs par jour en moyenne ? Trouvez-vous ce résultat satisfaisant pour une utilisation réelle ?

1. L'accuracy d'un modèle est sa proportion d'instances bien classées. Elle est notée "CA" (classification accuracy) dans le widget "Test and score"

Annexe : description des attributs

Nr	Features	
	Name	Description
1	duration	duration of connection in seconds
2	protocol_type	connection protocol (tcp, udp, icmp)
3	service	dst port mapped to service (e.g. http, ftp, ..)
4	flag	normal or error status flag of connection
5	src_bytes	number of data bytes from src to dst
6	dst_bytes	bytes from dst to src
7	land	1 if connection is from/to the same host/port; else 0
8	wrong_fragment	number of 'wrong' fragments (values 0,1,3)
9	urgent	number of urgent packets
10	hot	number of 'hot' indicators (bro-ids feature)
11	num_failed_logins	number of failed login attempts
12	logged_in	1 if successfully logged in; else 0
13	num_compromised	number of 'compromised' conditions
14	root_shell	1 if root shell is obtained; else 0
15	su_attempted	1 if 'su root' command attempted; else 0
16	num_root	number of 'root' accesses
17	num_file_creations	number of file creation operations
18	num_shells	number of shell prompts
19	num_access_files	number of operations on access control files
20	num_outbound_cmds	number of outbound commands in an ftp session
21	is_hot_login	1 if login belongs to 'hot' list (e.g. root, adm); else 0
22	is_guest_login	1 if login is 'guest' login (e.g. guest, anonymous); else 0
23	count	number of connections to same host as current connection in past two seconds
24	srv_count	number of connections to same service as current connection in past two seconds
25	serror_rate	% of connections that have 'SYN' errors
26	srv_serror_rate	% of connections that have 'SYN' errors
27	rerror_rate	% of connections that have 'REJ' errors
28	srv_rerror_rate	% of connections that have 'REJ' errors
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	srv_diff_host_rate	% of connections to different hosts
32	dst_host_count	count of connections having same dst host
33	dst_host_srv_count	count of connections having same dst host and using same service
34	dst_host_same_srv_rate	% of connections having same dst port and using same service
35	dst_host_diff_srv_rate	% of different services on current host
36	dst_host_same_src_port_rate	% of connections to current host having same src port
37	dst_host_srv_diff_host_rate	% of connections to same service coming from diff. hosts
38	dst_host_serror_rate	% of connections to current host that have an S0 error
39	dst_host_srv_serror_rate	% of connections to current host and specified service that have an S0 error
40	dst_host_rerror_rate	% of connections to current host that have an RST error
41	dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error
42	connection_type	