Les douzes travaux d'AstéRix

Sous la direction de M.LAUDE Henri

ARSIC Marko

02/02/2021

Contents

1 Introduction

Ce dossier est réalisé dans le cadre des enseignements de Programmation R et Mathématiques pour le Big Data délivrés par M. LAUDE Henry. Il s'agit d'une synthèse de 12 travaux réalisés par la promotion du MSc Data Management 2020-2022 de l'école Paris School of Business comportant du code R et des notions mathématiques.

Ainsi, mon devoir se présente en deux parties distinctes. La première abordant la partie Programmation R dans laquelle je présente les travaux réalisés sur les packages, et la seconde dédiée à la partie Mathématiques dans laquelle je traite des notions mathématiques qui ont été étudiées.

Ce devoir implique une évaluation des devoirs réalisés, c'est pourquoi l'ensemble des travaux sera jugé en fonction des critères suivant :

- Vulgarisation du concept ;
- Le caractère synthétque ;
- La clarté des propos ;
- La pertinence avec la Data Science ;
- La propreté et la structure.

Enfin, à la fin du docuement nous retrouverons une évaluation objective de deux travaux auxquels j'ai participé par rapport aux autres travaux, sur ces mêmes critères d'évaluation.

2 Partie programmation R

2.1 Leaflet

- Auteur : Arnaud FORASACCO
- Découvrez Leaflet, en cliquant ici.

2.1.1 Synthèse du travail

Le package Leaflet est spécialisé dans la création de cartes géographiques interactives avec R. Leaflet est l'une des librairies JavaScript les plus populaires, très largement utilisée de nos jours, par sa simplicité et son intuitivité à créer des maps interactives.

Ce package permet en quelques lignes de code très simples, d'afficher de façon visuelle une diversité d'informations. On peut notamment le comparer, par sa simplicité à afficher des visualisations, à ggplot2 qui est la référence en termes de graphiques.

Leaflet offre une multitude de possibilités, comme ajouter des surfaces, différents types de marqueurs, pour personnaliser et générer des cartes des cartes faciles à comprendre. A travers son travail, Arnaud nous montre les fonctions utiles pour débuter avec ce package et générer ses premières cartes. En effet, Arnaud nous explique tout d'abord comment générer simplement une carte, puis il va utiliser une base de données pour montrer les fonctions basiques de visualisation. On comprend que le package fonctionne par étapes. Une fois que la carte ait été générée avec OpenStreetMap, il suffit d'ajouter des couches, des arguments, pour personnaliser sa visualisation comme bon nous semble.

2.1.2 Extrait commenté

Avant de pouvoir effectuer toute visualisation avec Leaflet, il est nécessaire de pouvoir installer le package sur notre machine. Bien que cela puisse paraître évident pour certains, étant donné qu'il s'agit d'un tutoriel pour démarrer avec le package, il me semble important de préciser la commande pour installer le package. C'est un point de détail, mais que mon camarade a omis de préciser.

Par conséquent pour installer leaflet il suffit d'exécuter la commande suivante (avant de pouvoir l'exécuter comme cela a très bien été montré) :

```
install.packages('leaflet')
```

Leaflet permet non seulement d'afficher une map avec une vue globale, mais également de choisir mettre en évidence un point ou une région à l'aide des coordonnées GPS. C'est ce que nous montre Arnaud en pointant une visualisation sur le Taj Mahal avec la fonction setView, les coordonnées GPS ainsi que le degré de zoom souhaité. Également la fonction AddMarkers est utilisé pour rajouter du texte sur le point qu'on a choisi d'afficher « Taj Mahal » en glissant le curseur sur le point :

```
map <- leaflet() %>%
  addTiles() %>%
  setView(lng= 78.0419, lat=27.1750 ,zoom=15)
  addMarkers(map= map,lng= 78.0419, lat=27.1750 ,popup="Taj Mahal")
```

Une autre fonction pertinente que permet le package est de pouvoir regrouper des points sur une carte par couleur. Cela peut sembler bien utile lorsqu'il y a plusieurs points sur une même carte afin d'améliorer la lisibilité. Pour ce faire, mon camarade a utilisé le data frame quakes et la fonction addCircleMarkers afin d'ajouter des marqueurs circulaires en choisissant le format des clusters avec clusterOptions :

```r
m3 <- leaflet(data = quakes) %>%
  addTiles() %>%
  addCircleMarkers(lng=~longitude, lat= ~latitude,clusterOptions = markerClusterOptions())
m3
```

2.1.3 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (4/5)
- Clarté des propos (3/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (3/4)
  - NOTE GLOBALE 3.6/5

2.1.4 Conclusion

Le travail effectué par Arnaud m'a permis dès la première lecture de comprendre l'intérêt et e fonctionnement globale du package leaflet. Je trouve qu'il s'agit d'un très bon tutoriel pour débuter avec la visualisation des cartes interactives, et cela m'a donné envie de découvrir plus en détail les options qu'offre leaflet. Les explications sont claires et concises à la portée de tout un chacun. Petit point de détail, il manque les références à la fin du document.

2.2 LateX sous R Markdown

- Auteur : Jiayue LIU
- Découvrez LateX, en cliquant ici.

2.2.1 Synthèse du travail

Le second travail que nous nous proposons d'étudier est le tutoriel réalisé par mon camarade Jiayue concernant le langage informatique LateX et son utilisation sous R Markdown. Comme nous l'explique Jiayue, LateX est spécialisé pour produire des documents à caractère scientifique, comportant des expressions mathématiques, en respectant au mieux les normes typographiques. Ce langage qu'on ne présente plus, peut sembler difficile à appréhender, et la prise en main peut nécessiter plusieurs heures avant d'arriver à des résultats dignes de ce nom. Ainsi, cela nous donne la possibilité, avec un peu de maîtrise, de rédiger des formules complexes dont la mise en forme est réalisée automatiquement.

Je trouve ce tutoriel à la fois très synthétique et très complet, et il m'a été d'une bonne utilité pour la réalisation de mes documents comportant des notions scientifiques, dans le cadre des enseignements de ce semestre.

En effet, on y retrouve un condensé des possibilités qu'offre ce langage, et ce pour tous les niveaux. Un petit rappel de la syntaxe de base nous permet de découvrir la structure de l'environnement Latex, et les possibilités typographiques. On retrouve également des rappels des symboles et caractères spéciaux, ainsi qu'une explication des différents environnements pour la présentation de nos équations et des matrices. Une dernière partie est consacrée à l'insertion de graphiques qui est une option que j'ai découvert avec ce tutoriel.

2.2.2 Extrait commenté

Etant donné qu'il ne s'agit pas véritablement d'un package, dans cette partie il n'y aura pas réellement d'aspect très technique et d'explication de code R standard. Toutefois, je vais essayer de démontrer l'intérêt et la pertinence des commandes LateX et des packages sous-jacents, qui me paraissent les plus utiles.

Les sections 1 et 2 ne présentent pas d'intérêt particulier à être expliquées étant donné qu'elles ne présentent que les rudiments de ce langage, ainsi qu'une énumération des différents caractères spéciaux propre à celui-ci. Néanmoins, outre les équations et les formules standards, LateX nous permet de produire des matrice grâce à l'environnement `array`. La commande `\begin{array}` (`\end{array}`) permet de générer une matrice en utilisant les commandes `\left(` et `\right`). Les éléments d'une ligne sont séparés par l& symbole « & » et un retour à la ligne et symbolisé par un `\\.` :

```
\[
\left(
\begin{array}{ccc}
     x & y & z
  \\ \alpha & \beta & \gamma
\end{array}
\right)
\]
```

Le package asmmath permet également de modifier le délimiteur si nécessaire.

J'aimerai porter une attention particulière à l'insertion graphique dans LateX qui sedistingue du langage R Markdown. En effet, le package graphicx intégré dans LateX permet d'inclure de façon très simple, des éléments graphiques dans ces documents. Pour ce faire il suffit d'utiliser la commande `includegraphics[options]{nom du fichier}` à laquelle nous pouvons apporter des modifications de mise en page (positionnement, numérotation) pour un meilleur rendu :

```
begin{figure}
  \centering \includegraphics[angle=0,scale=0.25]{donald.png}
  \caption{"Donald Duck"}
  \label{duck}
\end{figure}
```

2.2.3 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (4/5)
- Clarté des propos (4/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (5/4)
  - NOTE GLOBALE 4.2/5

2.2.4 Conclusion

Le tutoriel réalisé par Jiayue m'a semblé vraiment très pertinent bien structuré et très parlant, que l'on soit débutant ou que l'on ait déjà quelques notions en Latex. Ce devoir m'a permis d'appréhender un peu mieux les expressions de ce langage et m'a guidé lors de la construction de certains de mes documents. Les sources pertinents apparaissent à la fin du document.

2.3 NetworkD3

- Auteurs : Claire MAZZUCATO & Adrien JUPITER
- Découvrez NetworkD3, en cliquant ici.

2.3.1 Synthèse du travail

A l'instar du premier document étudié, le package NetwordD3 qui nous a été présenté par Claire et Adrien, est issu d'une librairie JavaScript open-source. Ce package fournit des fonctionnalités diverses pour la représentation et la visualisation dynamique des données sous forme de graphe. Il permet de produire différents types de réseaux selon les besoins que l'on exprime, grâce aux fonctions qui lui sont intégrées. Ce devoir est articulé autour des fonctions principales de NetworkD3, chacune permettant de réaliser un modèle de graphique de réseau. On y retrouve notamment les graphiques fondés sur la force, les diagrammes de Sankey, ainsi que les diagrammes de réseau radial.

Dans leur devoir, mes camarades rappellent l'intérêt de l'étude des réseaux à travers un petit rappel historique avec le problème des sept ponts de Königsberg posé par Euler. De plus, ils mettent en évidence la pertinence d'utiliser R pour l'analyse des réseaux ; qui apparaît comme un outil performant, flexible, permettant de générer des graphiques de haute qualité rapidement.

2.3.2 Extrait commenté

L'installation du packages est bien expliquée, et nous comprenons que networkD3 est un package puissant qui offre différents modèles de visualisation. Une fois installé, il suffit donc tout simplement de charger la librairie networkD3, et de créer ou importer un jeu de données afin de réaliser les graphes.

Claire et Adrien ont eu la bonne idée de créer un jeu de données fictif, avec les prénoms de plusieurs camarades de classe, en utilisant les arguments src (source) et target (cible), afin d'illustrer ce à quoi peut ressembler un réseau basique.

```
# Chargement du package
library(networkD3)
# Création de données fictives
src <- c("Claire", "Claire", "Claire", "Adrien",
        "Adrien", "Adrien", "Claude", "Claude", "Claude","Siva", "Siva", "Siva","Thuy","Thuy","Thuy")
target <- c("Adrien", "Claude", "Siva", "Arnaud",
            "Siva", "Claude", "Claire", "Arnaud", "Siva","Claude","Adrien","Claire","Claire",
            "Arnaud","Adrien")
networkData <- data.frame(src, target)
# Plot
simpleNetwork(networkData)
```

De la même façon, ils nous ont montré les fonctionnalités supplémentaires que permet la fonction forceNetwork, en créant un graphe avec les personnages des Misérables, en utilisant les jeux de données MisLinks et MisNodes. Je trouve dommage le fait que les graphes n'apparaissent pas dans les tutoriels ; les représentations visuelles contribuent également à la compréhension.

Le diagramme de Sankey est également intéressant à comprendre, et c'est ce qui nous est proposé dans ce tutoriel à travers l'exemple du référendum britannique. Après avoir chargés les librairies adéquates, regroupés et transformés les données ; les liens et les nœuds ont été créés en utilisant les régions ainsi que les résultats des votes.

```r
# création des noeuds
regions <- unique(as.character(results$Region))
nodes <- data.frame(node = c(0:13),
                    name = c(regions, "Leave", "Remain"))
#création des liens
results <- merge(results, nodes, by.x = "Region", by.y = "name")
results <- merge(results, nodes, by.x = "result", by.y = "name")
links <- results[ , c("node.x", "node.y", "vote")]
colnames(links) <- c("source", "target", "value")
```

Enfin, une fois que tout les traitements on été effectués et que les données sont dans le format souhaité, nous pouvons tracer le diagramme en utilisant la fonction sankeyNetwork et en précisant les arguments Links, Nodes et Source :

```r
#draw sankey network
networkD3::sankeyNetwork(Links = links, Nodes = nodes, Source = 'source',
                         Target = 'target', Value = 'value', NodeID = 'name',
                         units = 'votes')
```

2.3.3 Evaluation

- Vulgarisation du concept (3/5)
- Caractère synthétique (4/5)
- Clarté des propos (3/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (4/5)
  - NOTE GLOBALE 3.6/5

2.3.4 Conclusion

Ce tutoriel m'a permis d'aborder la visualisation des graphes avec R en mettant en exergue les fonctions principales que délivre le package networkD3. Ce dossier fait le tour des éléments importants à connaître afin de débuter au mieux avec la notion de réseaux. Une explication plus précise de certaines commandes aurait été souhaitée, ainsi que la présence des visualisations dans le dossier. Dans son ensemble le sujet est bien traité et les notions clairement synthétisées.

2.4 Lubridate

- Auteurs : Gaspard PALAY
- Découvrez Lubridate, en cliquant ici.

2.4.1 Synthèse su travail

Analyser des données contenant des dates peut s'avérer comme étant une véritable épreuve si l'on si prend mal ou si on ne dispose pas d'un outil performant. En tant qu'étudiant au sein du Master Data Management et alternant au sein du pôle CyberSécurité d'un grand groupe bancaire, j'ai pour habitude de traiter des logs de sécurité comportant des dates ; et je ne comprends que trop peu la problématique que cela peut poser.

Les données comportant des dates peuvent être de différents formats et cela peut vite créer des erreurs si on ne dispose pas du programme adapté. Comme l'explique mon camarade Gaspard, le package lubridate fournit une multitude de fonctionnalités qui rendent plus agréables les traitements effectués sur les dates. Ce package, présent dans la libraire TidyVerse, comprend un grand nombre de fonction et s'impose comme un outil très puissant et performant pour manipuler les dates.

Lubridate comprends son lot de fonctions triviales, très intuitives et permet de jouer avec le format des données. De plus, le package permet d'effectuer des manipulations sur les dates et les heures, les intervalles mais également de calculer des périodes de temps, et des Instants. Une syntaxe simple et un caractère intuitif fait de lubridate un outil très puissant pour traiter vos dates.

2.4.2 Extrait commenté

Dans la mesure ou ce package comprend une multitude de fonctions très simples, la pertinence de mes commentaires concernant le code sera moindre. Néanmoins, je vais tenter de faire un focus sur quelques aspects et quelques fonctions que je trouve particulièrement utiles afin d'en préciser l'intérêt.En préambule, une attention particulière est portée aux possibilités d'installation du package ce qui est apprécié.

Une des fonctions principales de lubridate dmy() permet de convertir un vecteur de type chaîne de caractère en un vecteur de type date. Pour ce faire il suffit d'identifier l'ordre dans lequel le jour le mois et l'année apparaissent dans la chaîne de caractère et ensuite mettre les lettres dans le bon ordre :

```
jourJ <- lubridate::dmy("30 may 2020")
class(jourJ)
```

Certaines fonctions permettent d'effectuer des arrondis, c'est le cas notamment de ceilng_date(), floor_date(), ou encore round_date().

Le package lubridate nous donne la possibilité d'effectuer des calculs sur les dates de façon simple et précise, et ainsi de prendre en compte automatiquement la multitude de paramètres que cela englobe(années bissextiles, secondes intercalaires etc) :

```
t1+months(9) # t1 + 9 mois
t1+ddays(287) # t1 + exactement 287 jours
ddays(287)/dweeks(1) # combien de semaines (exactement) pour 287 jours?
t2-dweeks(7) # t2 - 7 semaines
```

2.4.3 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (4/5)

- Clarté des propos (4/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (3/5)
  - NOTE GLOBALE 3.8/5

2.4.4 Conclusion

Ce dossier m'a interpellé au premier abord étant donné que c'est un sujet qui me parle et auquel j'ai déjà dû à faire. Bien quel le package soit un peu simple, il englobe un grand nombre de fonction dont un florilège a été exposé dans ce devoir. Un petit point d'amélioration serait apporté aux exemples, qui auraient pu être tirés de cas concret pour susciter davantage de pertinence. Également, il est dommage qu'il n'y ait pas les références en fin de devoir.

2.5 Rpart

- Auteurs : Siva CHANEMOUGAM & Maxime ALLAKERE
- Découvrez Rpart, en cliquant ici.

2.5.1 Synthèse du travail

Le dernier package que j'ai décidé d'appréhender aujourd'hui est le package Rpart (Recursive Partioning And Regression Trees), dont le tutoriel a été réalisé par Maxime et Siva. Il s'agit d'une librairie inspirée de l'approche CART (Classification and Regression Trees), qui permet de produire des modèles de prédiction représentés sous forme d'arbre de décision.

Une première partie du dossier est consacrée à la partie théorique, permettant notamment d'introduire des notions telles que les modèles de classification et de régression, de machine learning (apprentissage automatique) et d'arbre de décision. Dans cette partie, mes camardes prennent le parti de présenter les intérêts de cette méthode – la facilité d'utilisation et de compréhension - ainsi que les limites qu'elles présentent, comme le sur-apprentissage.

La seconde partie de ce tutoriel constitue une mise en pratique da la librairie Rpart, à travers un jeu de données sur le Titanic, fourni dans la librairie rpart.plot. Dans cette partie davantage pratique, on trouve un exemple d'application de rpart sur un cas concret ; avec notamment la création d'un jeu d'apprentissage pour le modèle et la construction de l'arbre de décision. Puis une partie d'optimisation de l'arbre construit avec la méthode de l'élagage.

Pour rappel, ce package est intimement lié à la méthode de cross validation, que j'ai présenté avec mon camarde Rindra au cours de ce semestre (vidéo disponible : . Effectivement la fonction rpart réalisé par défaut une estimation des performances de l'arbre de décision en appliquant la méthode de validation croisée.

2.5.2 Extrait commenté

Selon moi la compréhension d'un package passe par la compréhension des commandes de bases qui caractérisent la spécialité du package. En effet, outre le chargement des données et les modifications préalables de ces dernières pour construire le jeu d'apprentissage ; il me paraît important de mettre en avant les fonctions principales pour élaborer un arbre de décision.

Pour ce faire, il suffit d'utiliser la fonction rpart, en précisant la variable à expliquer à gauche et les variables explicatives à droite. Les commandes `rpart.control` et les arguments `minsplit` et `cp` permettent de déterminer la façon dont l'arbre sera découpé. Ensuite il suffit d'afficher le résultat avec la fonction `plot` et `text` :

```
#construction de l'arbre
ptitanic.Arbre <- rpart(survived~.,data= ptitanic.apprt,
                        control=rpart.control(minsplit=5,cp=0))
#affichage de l'arbre
plot(ptitanic.Arbre, uniform=TRUE, branch=0.5, margin=0.1)
text(ptitanic.Arbre,all=FALSE, use.n=TRUE)
```

On remarque que la visualisation produite avec l'exemple que nous propose Maxime et Siva est saturée. Par conséquent, une étape d'élagage s'impose afin d'optimiser le rendu de notre visualisation. Pour cela, il est nécessaire de déterminer le cp optimal (complexity parameter) sur lequel je décide de ne pas porter une attention particulière étant donné que cela ne relève pas d'une grande complexité.

Néanmoins cet indicateur va permettre de réaliser un représentation graphique optimisée en adoptant un mettant en évidence les informations les plus utiles. Pour cela il suffit d'utiliser la fonction `prune` sur notre arbre, avec le `cp` optimal trouvé précédemment et les arguments qui conviennent :

```
ptitanic.Arbre_Opt <- prune(ptitanic.Arbre,cp=ptitanic.Arbre$cptable
                            [which.min(ptitanic.Arbre$cptable[,4]),1])
```

Et d'afficher le résultat obtenu. Beaucoup plus pertinent n'est-ce pas ?

```
prp(ptitanic.Arbre_Opt,extra = 1)
```

2.5.3 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (4/5)
- Clarté des propos (5/5)
- Pertinence avec la DS (5/5)
- Propreté et structure (3/5)
  - NOTE GLOBALE 4.2/5

2.5.4 Conclusion

Ce devoir m'a permis de me remémorer de quelques notions abordées plus tôt dans le semestre, et même dans ma vie, afin de les rafraîchir. Le tutoriel dans son ensemble est très clair, les explications sont adaptées à la compréhension de tous. Certaines commandes sont découpées dans l'affichage pdf et celui nuit quelque peu à la compréhension. Également, la présence des références aurait été souhaitable.

3 Partie Mathématiques

3.1 KNN (K-Nearest Neighbors)

- Auteurs : Antoine SERREAU & Corentin BRETONIERE & Benjamin GUIGNON
- Découvrez KNN, en cliquant ici.

3.1.1 Synthèse du travail

Dans ce dossier, Antoine Corentin et Benjamin nous proposent de découvrir l'algorithme KNN (K-Nearest Neighbors) qui est une méthode d'apprentissage supervisé parmi les plus répandue en Machine Learning. On comprend que ce modèle sert aussi bien pour les cas de classification que pour la régression. Ce modèle va exploiter les données d'un échantillon afin de pouvoir prédire le positionnement d'une nouvelle observation ne faisant pas parti de l'échantillon. Cette méthode présente l'avantage d'être rapide, facile à interpréter et ayant un fort pouvoir de prédiction.

Le principe de cette méthode peut s'apparenter à l'analogique « dis moi qui sont tes voisins, je te dirais qui tu es ». En effet, pour effectuer des prédictions l'algorithme, contrairement aux méthodes de régression logistique ou linéaire, ne va pas construire un modèle prédictif. Ce dernier va se baser sur les k instances les plus proches de notre observation, pour produire une prédiction. Cela est d'ailleurs très bien illustré graphiquement dans le devoir.

3.1.2 Extrait commenté

L'algorithme KNN, comme nous pouvons nous en douter, nécessite une fonction de calcul de distance entre deux observations. On retrouve une multitude de fonction de calcul de distance, chacune est utilisée en fonction du type de données que nous avons à exploiter.

Je décide de mettre en avant la distance euclidienne qui est généralement très utile lorsqu'on traite des données quantitatives et du même type. Cette fonction permet de calculer la racine carrée de la somme des différences des carrées entre les coordonnées des points x et y, soit la distance euclidienne :

$$D_e(x,y) = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2}$$

3.1.3 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (4/5)
- Clarté des propos (5/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (4/5)
  - NOTE GLOBALE 4.2/5

3.1.4 Conclusion

Le devoir est clair, concis et structuré. Le principe de KNN et son intérêt et bien mis en évidence, en particulier ces applications actuelles. L'illustration du principe permet de comprendre facilement en quoi consiste la méthode KNN et les représentations graphiques également. Les références sont présentes en fin de dossier.

3.2 TPOT

- Auteur : Olfa LAMTI
- Découvrez TPOT, en cliquant ici.

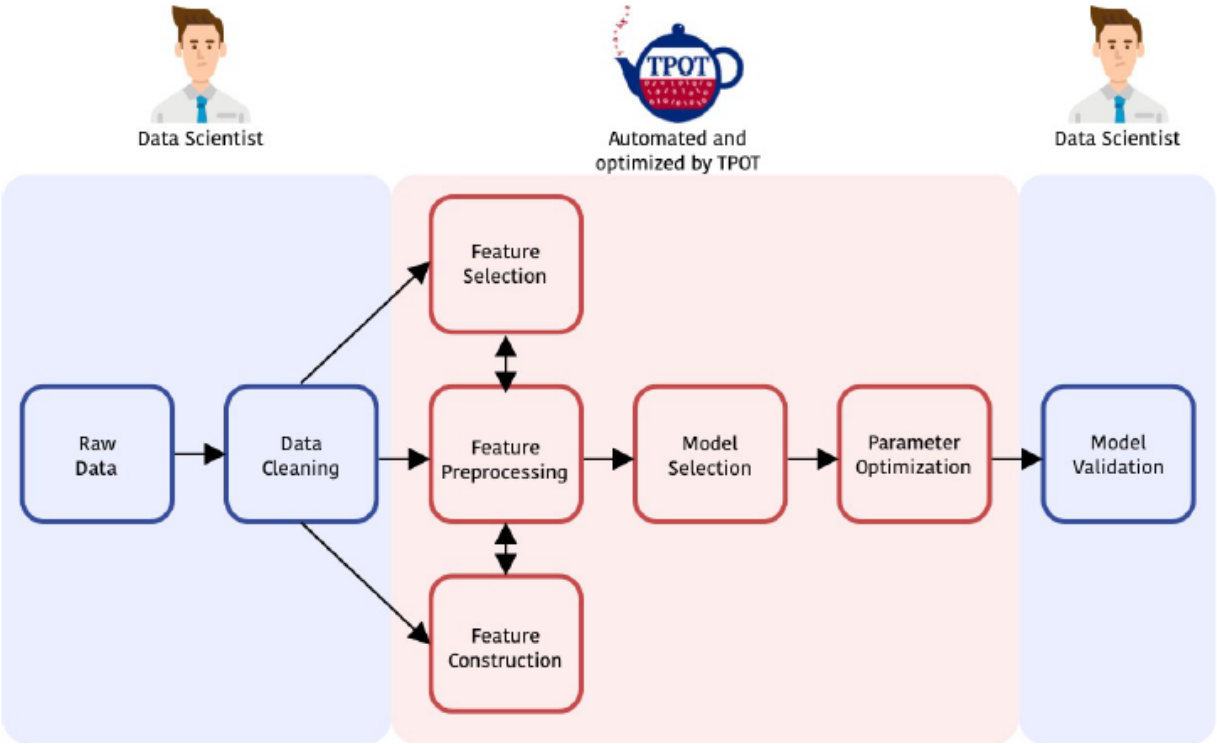
3.2.1 Synthèse du travail

Nous avons débuté avec un sujet de Machine Learning et nous allons poursuivre avec l'application de l'outil TPOT (Tree-bases Pipeline Optimization Tool) pour la data science biomedical. Pour rappel, un pipeline est un concept informatique qui correspond aux étapes de transport des données d'une source vers une cible. L'outil TPOT va permettre de construire de manière automatique des pipelines qui combinent les phases de pré-traitement, de sélection d'algorithme et d'optimisation. Il faut donc le voir comme une assistance au travail des data-scientists.

L'optimisation des modèles TPOT décrits par des arbres vont utiliser la programmation génétique. Cette technique d'optimisation s'inspire donc des mécanismes d'évolution biologique d'une population, énoncés par Darwin. Les algorithmes sous-jacents possèdent trois caractéristiques importantes : la sélection, le croisement et la mutation.

3.2.2 Extrait commenté

Ce devoir ne présente pas tout à fait des aspects mathématiques bruts. En effet, on ne retrouve pas de formule ni d'expressions mathématiques expliquant le fonctionnement théorique des algorithmes sous-jacents. Néanmoins, Olfa nous présente un schéma intéressant illustrant le fonctionnement de ce procédé. En effet, il s'agit d'un schéma représentant les étapes d'un pipeline classique, de la récupération des données brutes à la validation de ce dernier. La partie rouge met en évidence « la partie automatisable » du pipeline correspondant aux étapes de sélection et transformation des variables, choix du modèle et optimisation de ce dernier :



3.2.3 Evaluation

- Vulgarisation du concept (3/5)
- Caractère synthétique (4/5)
- Clarté des propos (4/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (4/5)
  - NOTE GLOBALE 3.8/5

3.2.4 Conclusion

Ce dossier m'a permis de découvrir un outil que je ne connaissais pas du tout, et étant donné que j'ai pour habitude de développer des pipelines à mon alternance, j'y trouve un intérêt tout particulier. Le devoir ne comporte malheureusement pas réellement d'aspect mathématiques bien qu'il soit très intéressant. Néanmoins, le sujet est bien exposé avec clarté et la structure donne une lecture agréable.

3.3 GAN (Generative Adversarial Network)

- Auteur : Jeremie SAYAG
- Découvrez GAN, en cliquant ici.

3.3.1 Synthèse du travail

Un réseau de neurones antagonistes génératifs ou GAN (Generative Adversarial Network est une technique de Machine Learning et plus généralement d'intelligence artificielle. Il permet de créer des imitations parfaites d'images et de toutes autres données.

Comme nous l'indique Jeremie, cette méthode repose sur la dualité de deux réseaux de neurones qu'on appelle « générateur » et « discriminateur ». Le premier a pour but de créer de nouvelles instances, tandis que le deuxième à pour tâche de valider l'authenticité de l'objet créé par le premier. Ainsi, ces deux entités se confrontent et c'est cela qui va leur permettre de s'améliorer mutuellement, jusqu'à atteindre un point d'équilibre. Ce genre de réseau peut avoir des utilités diverses et variées. Par exemple, on peut s'en servir afin générer des données, mais encore dans des domaines tel que l'art l'architecture et bien d'autres encore.

3.3.2 Extrait commenté

Les notions mathématiques constitutives de ce devoir sont les expressions mathématiques du générateur et du discriminateur. Ce sont les éléments centraux des réseaux de neurones antagonistes génératifs, et la fonction d'optimisation caractérise la dualité de leurs objectifs respectifs. Le discriminateur, dont le rôle est de distinguer réel de l'artificiel s'exprime en cherchant à maximiser $D(x)$ et $1 - D(G(z))$. A l'inverse, le générateur a pour but de l'induire en erreur ce qui revient à maximiser $D(G(z))$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

3.3.3 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (5/5)
- Clarté des propos (4/5)
- Pertinence avec la DS (5/5)
- Propreté et structure (4/5)
  - NOTE GLOBALE 4.4/5

3.3.4 Conclusion

Ce dossier m'a permis de découvrir un outil que je ne connaissais pas du tout, et étant donné que j'ai pour habitude de développer des pipelines à mon alternance, j'y trouve un intérêt tout particulier. Le devoir ne comporte malheureusement pas réellement d'aspect mathématiques bien qu'il soit très intéressant. Néanmoins, le sujet est bien exposé avec clarté et la structure donne une lecture agréable.

3.4 EPEARS

- Auteurs : Thuy AUFRERE & Claire MAZZUCATO & Claude REN
- Découvrez EPEARS, en cliquant ici.

3.4.1 Synthèse du travail

Le sujet que nous nous proposons d'étudier et le troisième papier de recherche du dossier réalisé par mes camarades Claire, Claude et Thuy, qui traite de la prédiction dans le domaine du décrochage scolaire. En effet, ces derniers ont décidé de présenter trois papiers de recherche sur un sujet d'actualité qui particulièrement la jeune population et s'impose comme un débat social et sociétal majeur.

Ce document nous questionne sur la possibilité de prédire qui sont les étudiants à risque, et implicitement le décrochage scolaire, à travers leurs comportements d'apprentissage, afin de mettre en place des mesures de préventions. Pour cela, ils suggèrent l'algorithme EPEARS qui permettrait la modélisation des habitudes d'apprentissage à travers les logs des plateformes d'apprentissage. Le constat est le suivant : les étudiants à risque ne disposent pas d'une routine de travail, et les amis de ces derniers sont davantage exposé au décrochage scolaire également.

3.4.2 Extrait commenté

Dans le papier de recherche que nous présentent mes camarades, l'algorithme qui se cache derrière cette expérience permet de construire des modèles en se basant sur les comportements d'apprentissage des étudiants. Les variables explicatives sont la régularité d'apprentissage et le facteur d'homophilie sociale, qui correspond au fait de s'affilier avec des individus possédant partageant des caractéristiques semblables. La fonction présentée permet justement d'exprimer ce phénomène, en explorant les nœuds voisins de l'individu concerné. La méthode d'apprentissage repose sur l'estimateur du maximum de vraisemblance, c'est pourquoi on cherche à maximiser la fonction de vraisemblance dans l'équation :

$$\max_f \sum_{u \in V} log(\prod_{v_i \in N_s(u)} \frac{exp(f(u).f(v_i))}{\sum_{v \in V} exp(f(u).f(v))})$$

3.4.3 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (5/5)
- Clarté des propos (5/5)
- Pertinence avec la DS (5/5)
- Propreté et structure (5/5)
  - NOTE GLOBALE 4.8/5

3.4.4 Conclusion

Les papiers de recherche que mes camarades ont décidé de présenter une pertinence particulière en cette période d'une ampleur inédite. Les explications sont claires et concises, les notions mathématiques bien expliquées et les résultats de l'étude sont très pertinent. Le fait d'avoir choisi trois sujets très liés est une très bonne idée. Le document est bien structuré et les références bien présentées également.

3.5 Arbres de décision

- Auteurs : Rindra LUTZ & Nicolas ALLIX
- Découvrez les arbres de décision, en cliquant ici.

3.5.1 Synthèse du travail

Nicolas et Rindra nous présente la représentation en arbre de décision. En effet il s'agit d'un outil d'aide à la décision qui permet de visualiser un ensemble de choix sous la forme d'un graphique composé de branche, qu'on appelle généralement un arbre. Cette méthode d'apprentissage supervisée permet donc de modéliser une hiérarchie de test afin de faire des prédictions. Les deux principaux types sont les arbres de régression et les arbres de classification.

Cet outil est largement utilisé de nos jours du fait de sa simplicité. Le but étant d'expliquer une variable à expliquer à partir d'autres variables explicatives. Pour cela, l'algorithme cherche à partitionner les individus en groupes d'individus les plus similaires possibles du point de vue de la variable à prédire. On retrouve trois éléments principaux dans un arbre : les nœuds racines, les nœuds internes et les nœuds terminaux. Son fonctionnement est bien expliqué avec la représentation graphique d'un cas concret d'une stratégie de développement.

3.5.2 Extrait commenté

Comme nous pouvons le comprendre, l'arbre de décision est un outil itératif, c'est-à-dire qu'à chaque itération, donc à chaque nœud, les individus vont être séparés en k sous-groupes et cela jusqu'à ce que le processus s'arrête. La notion de pureté est selon moi très importante. Celle-ci est utilisée pour caractériser un ensemble d'individu associés à une valeur lorsqu'ils appartiennent à la classe en question ; et s'exprime avec l'indice de Gini :

$$G_i = 1 - \sum_{k=1}^{n} P_i\,, k^2$$

Également la notion de coût est cruciale dans le processus de décision et s'expriment avec la mesure d'impureté des nœuds descendants ainsi que la proportion de la population sur les nœuds gauche et droite :

$$J(k) = (\frac{m_{gauche}}{m})G_{gauche+}(\frac{m_{droite}}{m})G_{droite}$$

3.5.3 Evaluation

- Vulgarisation du concept (3/5)
- Caractère synthétique (4/5)
- Clarté des propos (3/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (3/5)
  - NOTE GLOBALE 3.4/5

3.5.4 Conclusion

La notion est bien expliquée et l'illustration avec l'exemple de choix d'une stratégie de développement paraît très pertinente. Le concept est simplement expliqué et est à la portée de tous. Quelques répétitions dans le texte peuvent sommes toutes être corrigées et les références intégrées à la fin du dossier.

4 Auto-évaluation

4.1 Flexdashboard

- Auteurs : Marko ARSIC & Nicolas ALLIX
- Découvrez les arbres de décision, en cliquant ici.

Ce devoir réalisé plus tôt dans le semestre permet d'introduire la notion de dashboard qui est un outil de data science largement répandu. En effet cette librairie R, facile et intuitive offre de nombreuses fonctionnalités qui permettent de personnaliser ses tableaux de bords. Dans ce rapport, nous présentons un florilège de ces options tel que les dispositions, l'organisation des pages, ajustement des graphiques etc. C'est le package parfait pour construire ses premiers tableaux de bords.

4.1.1 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (4/5)
- Clarté des propos (3/5)
- Pertinence avec la DS (4/5)
- Propreté et structure (3/5)
  - NOTE GLOBALE 3.6/5

4.2 Cryptohraphie et théorie des nombres

- Auteurs : Marko ARSIC & William ROBACHE
- Découvrez les arbres de décision, en cliquant ici.

Nous avons également choisi d'aborder l'aspect mathématiques de la cryptographie et notamment son lien avec a théorie des nombres. Dans ce dossier nous avons essayé d'expliquer le plus simplement possible ce sujet qui peut s'avérer très complexe, étant donné que les méthodes de cryptohraphie sont des processus élaborés. Nous distinguons la cryptographie à clé symétrique de la cryptographie à clé asymétrique, et c'est sur cette dernière que nous nous sommes intéressés en particulier. Fonction a sens unique,principe de chiffrement, système RSA : toutes ces notions qui peuvent paraître asbtraite sont explicitées dans notre rapport.

4.2.1 Evaluation

- Vulgarisation du concept (4/5)
- Caractère synthétique (4/5)
- Clarté des propos (4/5)
- Pertinence avec la DS (5/5)
- Propreté et structure (4/5)
  - NOTE GLOBALE 4.2/5