# Exam Maths

Arnaud Forasacco

01/02/2021

## 1er travail : Tree-based pipeline optimization fait par LAMTI Olfa

https://github.com/OlfaLmt/PSBX/blob/main/Thèses/Automating%20biomedical%20data%20science%20through%20tree-based%20pipeline%20optimization.pdf

Ce travail traite de l'optimisation automatisée de pipelines à partir d'un arbre de décision. Ces pipelines sont des séries d'action réalisée afin de récupérer de la data brute, puis la traiter pour l'exploiter par la suite. Dans ce travail, il est présenté le fonctionnement de l'algorithme TPOT permettant de réaliser des pipelines avec diverses configurations possibles. L'algorithme implémente des pipelines basés sur des arbres où chaque nœud représente un «opérateur». TPOT implémente 4 «opérateurs» principaux : préprocesseur, décomposition, sélection de fonctionnalités et modèle. Ces opérateurs sont orchestrés comme une arborescence où les feuilles sont une ou plusieurs copies des données d'entrée. L'ensemble de données parcourt l'arborescence où les entités évoluent opérateur par opérateur se trouvant dans le nœud final où le modèle (classification ou régression) est généré. Dans l'exemple ici, il est utilisé des algorithmes génétiques, c'est un type d'algorithme évolutif (EA), un sous-ensemble de l'apprentissage machine. La nature adaptative des EA peut générer des solutions comparables ou meilleures que ce que l'Homme puisse faire. La GP peut être utilisée pour découvrir une relation fonctionnelle entre les caractéristiques des données (régression symbolique), pour regrouper les données en catégories (classification) et pour aider à la conception de circuits électriques, d'antennes et d'algorithmes quantiques.

Le travail est clair et bien structuré, il est facile de comprendre le fonctionnement de l'algorithme, notamment grâce au schéma. C'était une découverte pour moi et je trouve cet algorithme très intéressant et il pourrait s'avérer utile à étudier plus profondemment.

## 2ème travail : Algorithme génétique fait par Ramya

https://github.com/RamyaHTDJ/Psb_Ramya/blob/main/Algo_genetique.Rmd

Le travail portant sur les algorithmes génétiques de Ramya nous présente le fonctionnement et le déroulement d'un process génétique. Dans les de son travail, nous trouvons une liste de termes propres au domaine qui nous permettent de mieux comprendre le document. La génétique étant un domaine qui va beaucoup évoluer dans les prochaines années, il est très interessant d'en apprendre plus sur la façon dont les algoritmes génétiques fonctionnent.

Les algorithmes génétiques utilisent la théorie de Darwin sur l'évolution des espèces. Elle repose sur trois principes : le principe de variation, le principe d'adaptation et le principe d'hérédité.

Ici il est présenté les différentes étapes par lesquelles passent les "solution candidate" à travers l'algorithme. Tout d'abord, nous commençons par l'initialisation d'une population de solutions possibles au problème donné. Cette population peut être générée au hasard ou générée en utilisant les fonctions heuristiques.

La prochaine étape est la définition de la fonction de fitness faite pour trouver les individus qui seront éventuellement solutions au problème posé.

Ensuite, arrive la phase de représentation, où une représentation incorrecte peut entraîner de mauvaises performances de l'AG. Parmis ces représentations, nous trouvons différents modèles : binaire, en valeur réelle, en nombres entiers, en permutation...

Après cela, il y a la sélection des parents de cette population pour l'accouplement basé sur une valuer d'aptitude, conforme à la théorie darwinienne de la survie du plus fort. Différentes séléctions peuvent s'appliquer : condition physique, tournois K-Way, rang ou aléatoire.

La phase de croisement est la phase suivante qui contribue au brassage génétique lors de la reproduction (Recombinison génétique). On applique le principe d'hérédité de la théorie de Darwin. Dans ce cas, plus d'un parent est sélectionné et une ou plusieurs descendances sont produites en utilisant le matériel génétique des parents. Le crossover est généralement appliqué dans un GA avec une probabilité élevée.

Vient ensuite la phase de mutation qui peut être définie comme un petit ajustement aléatoire du chromosome, pour obtenir une nouvelle solution. Elle est utilisée pour maintenir et introduire de la diversité dans la population génétique.

Pour terminer, il y a la selection des survivants qui peut se faire en fonctio de l'âge ou basée sur l'aptitude physique

La condition de fin d'un algorithme génétique est importante pour déterminer le moment de la fin d'un cycle d'AG. Il a été observé qu'au début, l'AG progresse très rapidement avec de meilleures solutions à quelques itérations d'intervalle, mais cela tend à saturer dans les étapes ultérieures où les améliorations sont très faibles. Il est généralement voulu que la condition de fin soit telle que la solution soit proche de l'optimum, à la fin du cycle.

Ces descendants remplacent donc les individus existants dans la population et le processus se répète. De cette façon, les algorithmes génétiques tentent en fait d'imiter l'évolution humaine dans une certaine mesure.

Pour moi, ceci est un travail bien détaillée et aide parfaitement à comprendre comment les AG fonctionnent même si l'on a pas le vocabulaire nécessaire. Les étapes de l'algorithme sont claires et complètes et j'i beaucoup apris sur un sujet qui va être beaucoup étudié à l'avienir.

## 3ème travail : La Régression linéaire fait par ZOUMANIGUI Nina

https://github.com/Nina809/PSBX/blob/main/Regression.Rmd

La régression linéaire est un concept très utilisé dans le cadre des études statistiques et économétriques. Elle est utilisée pour prédire un résultat quantitatif y sur la base d'une seule variable prédictive x. Le but est de construire un modèle mathématique (ou une formule) qui définit y en fonction de la variable x. Une fois que l'on construit un modèle statistiquement significatif, il est possible de l'utiliser pour prédire les résultats futurs sur la base de nouvelles valeurs x. Cela peut se représenter par une droite traversant un nuage de points, un modèle linéaire parfait serait une droite traversant tous les points dans une représenattion de nuage de points. Pour le cas de la régression multiple, c'est une généralisation du modèle de régression simple lorsque les variables explicatives sont en nombre fini.

```
plot(mpg~wt,pch=20)
fit= lm(mpg~wt,data=data)
fit
abline(fit,col="red",lwd=2)
```

Dans l'exemple présenté, on voit la fonction "lm" signifiant linear model et des "weights" s'appliquant à la fonction.

Le travail est pour moi bien expliqué et montre le fonctionnnement de la régression simple et multiple à trvaers un exemple. Il est donc compréhensible, claire et bien struturé et le choix du sujet est pertinent car la régression est un sujet commun en data et statistiques.

## 4ème travail : ML for financial products fait par EL GHALDY Soukaina

https://github.com/soukainaElGhaldy/PSB-X/blob/main/Mathematics/maths.pdf

Le machine learning en finance est devenu plus important récemment en raison de la disponibilité de grandes quantités de données et d'une puissance de calcul plus abordable. Les principales banques et sociétés de services financiers utilisent de plus en plus l'IA, pour rationaliser leurs processus, optimiser leurs portefeuilles, réduire les risques ou encore souscrire des prêts.

L'algorithme "Experts Network" presenté ici l'exemple tournera autour de la question suivante :

« À une date t, quel investisseur est intéressé par l'achat/vente de quel actif financier ?»

Cet algoritme, prennant la forme d'un réseau neuronal va faire ressortir un "Supervised clustering" des investisseurs. Il faudra d'abord examiner l'hétérogénéité des investisseurs et y introduire une modélisation mathématique. Ensuite, il s'agira d'effectuer l'architecture du réseau neuronal pour analyser les groupes hétérogènes d'investisseurs. Ceci ayant pour but de prédire des stratégies d'investissement adoptées par les clients.

La modélisation mathématique consiste à présenter l'ensemble des stratégies de placement distinctes qu'un investisseur pourrait choisir et appartennant à un groupe M de l'ensemble des stratégies pouvant exister. Il existe selon un mapping f qui va de l'information actuelle à l'expression d'intérêt à un achat ou une vente d'un actif.

Nous définissons D comme l'ensemble des stratégies de placement distinctes qu'un investisseur pourrait choisir et appartennant à un groupe M (fictif) de l'ensemble des stratégies pouvant exister. D existe selon un mapping f qui va de x à y, où x est l'information actuelle et y l'expression d'intérêt à un achat ou une vente d'un actif.

L'algorithme va donc devoir déduire l'ensemble des stratégies de placement.

Le réseau neuronal est conçu dans le but de suivre l'hypothèse formulée dans la modélisation ci-dessus

Il est donc composé d'un réseau neuronal indépendant (Gating block) dont le rôle est d'affecter les investisseurs aux différents groupes d'investisseurs(n) appelés "experts". Ces blocs reçoivent de différentes données d'entrée.

Ce réseau possède aussi une autre bloc (Experts block) composé de n sous-réseaux indépendants, appelés experts. Chaque expert reçoit en entrée les données des experts correspondant aux caractéristiques utilisées pour résoudre.

Chaque expert apprendra une cartographie f qui correspond le mieux aux actions des investisseurs attribués. Par conséquent, le rôle d'un expert est de récupérer un f(k) donné, correspondant à l'un des groupes d'investisseurs sous-jacents K que nous avons supposé.

Les résultats de ces deux blocs sont combinés par l'équation suivante :

$$f(x|a) = \sum_{i=1}^{n} p(i|a) f_i(x),$$

Pour terminier, grâce à la fonction "softmax", l'algoritme va pouvoir s'entrainer à calculer la probabilité que tel investisseur soit affecté à tel expert, et trouvant la plus grande probabilité, il est ainsi possible de prédire le comportement de cet investisseur.

$$p(x|a) = Softmax\left(W_{experts} * x\right)$$

Le travail est très intéressant et structuré, il présente le contexte et le monde financier avant de commencer à parler de l'algorithme "ExNet". Il est facilement compréhensible même sans beaucoup de notions mathématiques et nous voyons parfaitement comment fonctionne l'algorithme, comment il est utilisé et exploité.

## 5ème travail : Naïve Bayes Classifier fait par HOUNSINOU Jordy

https://github.com/Jordyhsn/PSB_Hounsinou/blob/main/Naïve-Bayes.pdf

Ce travail traite de la classification “Naive Bayes”, un ensemble d’algorithmes couramment utilisé dans le Machine Learning. Son objectif est donc de pouvoir résoudre les problématiques de classification dont on fait face dans la vie courante en se basant sur des variables totalement indépendantes entre elles, d’où son appellation “Naive”.

La loi de Bayes se définit par la formule suivante :

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)}$$

Avec $A$ et $B$ des événements, $P(A)$ la probabilité de $A$ et $P(A \mid B)$ la probabilité conditionnelle de A sachant B.

Afin d’illustrer et de mieux comprendre comment fonctionne cette loi, il nous sera présenté le cas du COVID-19. Ces données seront appliquées au cas :

- 1 personne sur 1000 attrape le covid-19
- La précision du test génique PCR est de 99%

Après utilisation de la loi de Bayes, nous constatons qu’il y a une probabilité de 9 % d’avoir le virus si l’on est positif, ce qui nous dévoile que même si l’on ressort positif sur le test, il y a une faible chance de réellement l’avoir.

Voici ensuite la présentation d’une application de la méthode naive Bayes :

Le fait que la méthode Naive s’applique à plusieurs variables indépendantes complexifie le calcul de la probabilité. Voici la formule applicable :

$$\mathrm{P}(\mathrm{C} \mid F1, \ldots, Fn) = \frac{\mathrm{P}(\mathrm{C}) * \mathrm{P}\left(\mathrm{C}|_{F1,\ldots,Fn}\right)}{\mathrm{P}(F1, \ldots, Fn)}$$

Pour l’application de cette méthode, il sera pris en compte plusieurs symptômes :

- La personne a des symptômes de Difficultés respiratoires
- La personne a des symptômes de Perte de l’odorat
- La personne a de la Fièvre
- La personne a des Maux de tête

Grâce à un jeu de données et l’application de cette formule, on remarque que la probabilité que notre personne soit porteuse du Covid-19 est largement plus grande que les autres. On classifie notre individu inconnu comme étant porteur du Covid.

Les avantages d’un tel algorithme sont nombreux, nous pouvons relever ces 2 avantages majeurs ci :

- Il est relativement simple à comprendre et n’exige aucune volumétrie de données, il pourrait même s’appliquer aux petits jeux de données.
- Il est très rapide pour les enjeux de classification et pas très couteux.

Ce travail est très exlicite et bien structuré, ce qui permet de comprendre avec aisance l’utilisation de cette formule de calcul de probablilité. Cette présentation de l’exemple du COVID-19 est très clair et permet de l’appliquer sur d’autres domaines.