Javascript par la pratique

Le Machine Learning

Apprentissage supervisé à l'aide d'un algorithme de classification

Travail de Maturité

Vibbeshan Nagalingam

Christopher Perritaz

Collège du Sud, Bulle

Mars 2021

Travail de maturité réalisé
sous la direction de
Monsieur Boetzel Sascha et Monsieur Charrière Jérome

(Informatique et Mathématiques)

Table des matières

Introduction

Description

Le *Machine Learning*, l'apprentissage automatique, est un domaine d'application de l'intelligence artificielle permettant à l'ordinateur d'accéder à des données et à les utiliser afin d'apprendre et de s'améliorer. L'objectif est de permettre à la machine d'effectuer des choix optimaux grâce à l'observation d'exemples, d'expériences ou d'instructions en y retrouvant un schéma particulier. L'apprentissage supervisé [1], dont il est question dans ce document, utilise un jeu de données à partir duquel l'algorithme KNN prédira un résultat pour une nouvelle donnée.

La fonction principale du *Machine Learning* est de fournir des prédictions offrant la possibilité d'optimiser et d'automatiser des choix. De nos jours, l'information ne cessant de gagner en ampleur, il est nécessaire de pouvoir l'analyser et de la traiter afin de pouvoir répondre de la meilleure des manières aux différents besoins. Ainsi, anticiper les pannes

1. L'apprentissage supervisé vise à la prédiction à partir d'exemples donnés en recourant à une fonction ou un algorithme. L'avantage d'un apprentissage supervisé est de prendre des décisions plus humaines. Mais il n'est pas capable de traiter d'autre type d'information. En utilisant des données ayant une classe qu'on lui donne pour entraîner un modèle, l'apprentissage supervisé ajuste sa précision en consommant de nouvelles données, apprenant ainsi sur une longue période de temps

et planifier des maintenances pour des machines, prévenir les blessures sportives et améliorer la cohésion d'une équipe, prédire les départs d'employés, burn-outs ou s'occuper de questions relatives à l'égalité des sexes, prévoir l'évolution d'épidémies ou de marché sont différents exemples d'applications importantes [2]. Il va sans dire que le *Machine Learning* est un domaine actuel et en plein essor.

Problématique, articulation du travail et plan du document théorique

Un des domaines à l'intérieur duquel l'algorithme K-NN excelle particulièrement est la recommendation d'un produit à partir d'autre produit du même type. Il est facile de s'imaginer la manière dont certains services recommande leurs produit. Une entreprise chargé de recommander des musiques à partir de récentes écoutes, pourrait facilement rechercher les écoutes récentes d'autre personnes ayant écouté les mêmes musiques et de les recommander. Similairement, il est intéresssant de se poser le même type de question pour un choix interne à une formation gymnasiale : le choix de l'option spécifique. En récoltant alors des données auprès d'élève ayant déjà fait leur choix. C'est ainsi que l'on arrive à la problématique de ce travail de maturité :

De quelle manière un algorithme de classification peut-il conseiller une OS à un élève de 1GY à partir de données recueillies auprès d'élèves de classes supérieures ?

Le but de ce travail de maturité est dès lors d'essayer de prédire à un élève de 1ère gymnase dans quelle option sont allées des personnes lui ressemblant. Pour ce faire, l'élève devra répondre à un questionnaire auquel auront déjà répondu d'autres élèves de classes supérieures ayant choisi leur option. En vue d'atteindre un résultat satisfaisant, les questions ont été préparées avec l'aide d'un orienteur professionnel : monsieur Bourquin Kevin. Le programme informatique associé à ce document implémentera un algorithme permettant d'attribua à une nouvelle donnée un groupe [3] lui correspondant. Pour finir, une analyse critique des résultats obtenus permettra de poser un point de vue objectif sur la performance fournie par le programme et la fiabilité des questions.

Le document théorique se présente en trois étapes. La première contient une explication concernant l'algorithme K-NN contenant la manière dont il fonctionne et quelques

2. Exemples tirés du site de la Swiss SDI(Swiss Statistical & Innovation), entreprise spécialisée dans le conseil analytique et stratégique. https://swiss-sdi.ch/en/use-cases/

3. Les groupes sont constitués de données similaires. De tailles variables, ils représentent un schéma particulier commun aux données y appartenant.

problème courant qu'il rencontre. La deuxième regroupe des techniques permettant de préparer un set de données et de permettre à l'algorithme d'obtenir une meilleure précision. Finalement, la dernière présente l'implémentation en Javascript de l'algorithme et une brève analyse des résultats obtenus.

Chapitre 1

K-NN

1.1 K-NN en théorie

1.1.1 Introduction

> “ Dis-moi qui tu fréquentes, je te dirais qui tu es. ”
>
> *- Proverbe populaire français*

Imaginons que nous recherchons une certaine marque de plume dans une papeterie. Il semble naturel de se rendre dans un rayon où l'on trouve du matériel pour écrire. En allant plus loin, comme l'on cherche une plume, il est évident qu'elle sera proche des autres plumes. Il est en effet bien pratique que les choses qui se ressemblent soient proches les unes des autres. Nous arrivons enfin devant le rayon des plumes et des stylos. Un chariot cependant bloque la vue sur le milieu du rayon, à gauche les plumes, à droite les stylos. Il y a fort à parier que la partie gauche du chariot cachera des plumes. De plus, nous pouvons remarquer que le rayon derrière nous est réparti à part égal entre des crayons de couleur et des feutres. Il y a alors peut-être une chance que celui des plumes et stylos le soit aussi. Il devient de plus en plus facile de deviner où se trouvent les plumes. Pourtant, il est assez intuitif d'affirmer qu'au centre du chariot, il est plus difficile de prédire quelle instrument d'écriture s'y trouvera, bien qu'il soit possible d'être presque

certain qu'à l'extrémité gauche il y aura des plumes. En y réfléchissant, il semblerait que lorsqu'il est possible de former un groupe qui se définit par les critères qu'on lui attribue (dans notre cas : la position latérale), il est naturel d'associer un objet répondant à ces mêmes critères au même groupe.

K-NN, abréviation de *k Nearest Neighbor* de la même manière prend en compte une notion de distance et part du simple principe que si une donnée ressemble à une autre, elles seront du même groupe.

les premières traces de K-NN apparaissent lors d'une analyse technique menée par Evelyn Fix (1904-1965) et Joseph Lawson Hodges Jr. (1922-2000) en 1951. L'idée fondamental est que K-NN soit une méthode de classification non paramétrique. Par la suite K-NN a évolué et de nouvelles études ont été menées. Encore de nos jours, des nouvelles approches continuent d'émerger.

1.1.2 Recherche des voisins les plus proches

Pour mieux comprendre la manière dont l'algorithme fonctionne, une explication est donnée au travers de l'exemple 1.1.2. Étant données la situation suivante, nous aimerions trouver l'avis de John sur un certain vote. Pour se faire, il y a trois critères à disposition, l'âge, le revenu et le sexe.

Nom	âge	revenu	sexe	vote
Richard	35	1600$	Homme	oui
Mélanie	20	1400$	Femme	non
Kim	25	3000$	Femme	non
George	25	3500$	Homme	oui
Annie	50	3600$	Femme	oui
Tom	50	6000$	Homme	non
Steve	55	9000$	Homme	non
David	30	9500$	Homme	oui
Danielle	60	10'000$	Femme	non
John	4060	4500$	Homme	?

TABLE 1.1 – Exemple : recherche d'un vote

La première étape est de trouver la personne à laquelle John s'assimile. Pour ce faire il est possible de s'intéresser à la distance entre John et les autres personnes [1]. Pour chaque critère, il est possible de calculer la distance entre la valeur qui est attribuée à John et celle qui est attribuée à une autre personne. Si l'on s'intéresse à l'âge, John a 40 ans, Kim 25 et Annie 50. La différence d'âge :

Entre Kim et John est : $40 - 25 = 15$

Entre Annie et John est :$50 - 40 = 10$

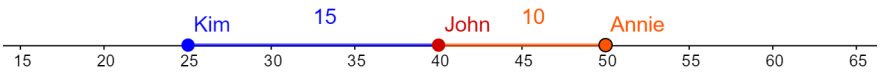

FIGURE 1.1 – Distance entre John et Annie et Kim

John ressemble alors plus à Annie qu'à Kim.

Le véritable intérêt de l'algorithme K-NN est non pas de rechercher un voisin qui ressemble le plus à une donnée pour chaque critère, mais de rechercher un nombre K de voisins pour tous les critères en même temps. En effet, K-NN n'associe pas pour chaque critère un voisin, mais pour tous les critère en même temps.

Autrement formulé, K-NN ordonne toutes les données en fonction de l'écart par rapport à la donnée dont ont recherche l'avis, en sélectionne les K données ayant les distances les plus petites, et retourne l'avis dominant.

Dans l'exemple ci-dessus, celà revient à :

1. L'exemple de l'introduction permet d'illustrer cette idée : si les stylos sont à droite, plus l'on se rapproche du côté droit, plus il est probable que c'est des stylos également

1. Choisir une valeur K
2. Calculer la distance entre chaque point : La distance Euclidienne [[3]] est la distance la plus courante utilisée.
3. Prendre l'avis des K voisins ayant les plus petites distances et retenir la classe [2] la proche

A l'intérieur de la figure 1.2, lorsque K = 3, K-NN sélectionne les trois voisins les plus proches ; deux sont de la classe verte et un est de la classe rouge. L'algorithme prédira que le point central est de la classe verte . Pour K= 6, 4 sont de la classe rouge et deux de la classe verte. L'algorithme prédira que le point central est de la classe rouge

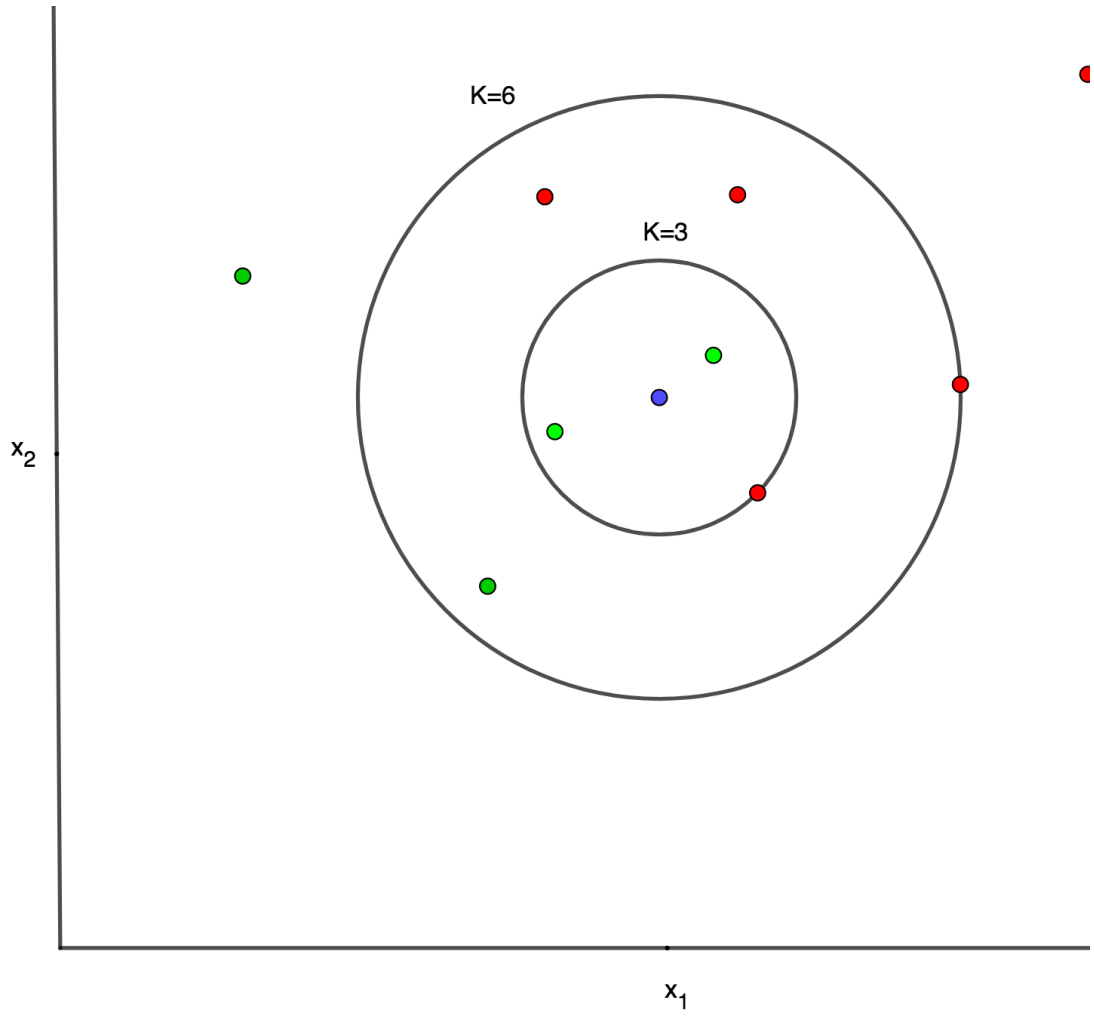

FIGURE 1.2 – Exemple de fonctionnement de K-NN

Bien choisir le nombre de voisin est alors important pour obtenir des résultats optimaux.

2. On considère comme classe, le nom de la catégorie dans laquelle tombe une donnée. Une classe peut posséder plusieurs groupes de données mais un groupe ne peut pas posséder plusieurs classes.

1.1.3 Balancer les données

Lorsque des distances sont utilisées, un problème ressort rapidement : si les distances ne sont pas du même ordre de grandeur, K-NN ne sera pas fiable car un critère sera bien plus pondérant que les autres. Dans notre exemple, si l'on utilise le critère de l'âge et du revenu, l'âge n'aura d'influence uniquement si le revenu est à distance égale entre deux personnes :

(a) distance entre Tom et John : $\sqrt{(50-40)^2+(6000-4500)^2} \cong 1500.03333296$

(b) distance entre Kim et John : $\sqrt{(40-25)^2+(4500-3000)^2} \cong 1500.07499813$

(c) distance entre Richard et John : $\sqrt{(40-35)^2+(4500-1600)^2} \cong 2900.00431034$

Une première solution à ce problème, si l'on souhaite conserver la manière dont K-NN fonctionne serait d'exprimer les distances entre chaque critère comme des valeurs comprises entre 0 et 1, où 0 signifierait que la distance est nulle et 1 que la distance est maximale :

$$DistancePonderee = \frac{Distance}{DistanceMax}$$

Tel que *Distance* soit la distance entre la le critère de base et la donnée recherchée et que DistanceMax soit la distance la plus grande possible, soit la distance entre le critère avec la plus grande valeur et celui avec la plus petite.

Autrement formulé : par rapport à la distance la plus grande possible, à quel point sera grande la distance entre les deux valeur du critère.

Dès lors, la distance entre Richard et John serait de :

$\sqrt{(\frac{40-35}{60-20})^2+(\frac{4500-1600}{10000-1400})^2} \cong 0.35963191401$

Et celle entre kim et John serait de :

$\sqrt{(\frac{40-25}{60-20})^2+(\frac{4500-3000}{10000-1400})^2} \cong 0.413578$

Il est désormais possible de déterminer une liste plus ou moins fiable des voisins les plus proches de John.

1.2 K-NN en pratique

1.2.1 Introduction

La stratégie de K-NN ressemble à de la mémorisation. Celà ressemble à simplement se souvenir de la réponse quand une question possède certaines caractéristiques, étant donné que cette même question est comprise par le moyen de certaines règles de classifications. C'est là un trait d'apprentissage préalable/faible (lazy algorithm), car en réalité, K-NN n'apprend rien durant la période de test.

Cela implique non seulement que K-NN est très rapide à entraîner, alors que la vitesse de prédiction est au contraire très lente.La plupart des calculs et la recherche des voisins s'effectuent en effet à ce moment-là. Mais encore que K-NN est gourmand en mémoire : toutes les données du set sont emmagasinées à l'intérieur de la mémoire imposant une limite lors de travaux avec des données relativement importantes en taille. Le véritable avantage de K-NN est d'être capable de fonctionner aisements avec de très nombreuse classes (par exemple : des tags pour des réseaux sociaux) [3].

1.2.2 Trouver le meilleur K et validation croisée

Le paramètre K se veut être le nombre de voisins pris en compte lors d'une tentative de réponse par l'algorithme K-NN.

Ce paramètre représente le niveau d'abstraction des tendances entre les classes : plus k est petit, plus l'algorithme s'adaptera aux données et sera sensible à de nombreuses tendances. Bien que certaines séparations et liens entre les classes soient mieux représentées, des données non appropriées (bruits [4]) sont très influentes et amènent une grande perte en terme de précision. Avoir un trop petit k mène alors au sur-apprentissage de l'algorithme (*overfitting*). Inversement, plus k est grand, plus l'algorithme suivra des tendances générales. Un grand K a comme conséquence de mal représenter des groupes minoritaires (voir ne pas les représenter du tout) mais il permet de supprimer les bruits. Avoir un trop grand K mène au sous-apprentissage (*underfitting*).

Il est facile de se représenter l'importance de la valeur du paramètre K. Par exemple, si l'on s'intéresse à l'état d'une ville imaginaire qui possède un certain nombre d'habi-

3. Tiré *de Machine Learning for Dummies* [JPM18]
4. cf. chapitre 2 : recherche de cas particuliers

tants. Et que l'on se fonde sur les types d'habitants de villes actuelles, Il est possible de s'imaginer le sens de certaines valeurs de K :

(a) K = 1 : La ville ressemble à un seul habitant : Une seule personne déterminerais l'entièrereté de la ville. Bien qu'il y ait utilité à ce genre de cas, ici, voulant représenter des types d'habitant, c'est absurde d'assumer qu'il n'y en ai qu'un seul.

(b) K = 3-6 : La ville ressemble à un foyer. (Mais les individus peuvent n'avoir aucun trait en communs). On pourra dire par exemple que si l'on fait passer un test à 100 habitants, 25 auraient les mêmes réponses. Ce genre de valeur pourrait par contre être utile pour définir des groupes politiques dominants (en ignorant les minorités).

(c) K = 10-50 : la ville ressemble à un groupe de la taille d'une famille.

Évidemment, ces genres de valeur sont extrêmes, mais elles représentent les risques du sur-apprentissage. Si pour K=5, nous avions un meurtrier, le modèle assumerait que 1 personne sur 5 est un meurtrier. La sensibilité aux bruit est un des plus gros problèmes du sur-apprentissage.

Pour d'autres k :

(a) K≅50'000'000 : la ville ressemble à un pays de moyenne taille.

(b) K ≅ 1 400'000'000 : la ville ressemble à un continent ou aux plus grands pays

(c) K≅ 8'000'000'000 : la ville ressemble au Monde entier.

.

Également extrême, ces valeurs représentent les risques du sous-apprentissage. Pour k ≅ 8'000'000'000, le Vatican, État souverain le moins Peuplé du monde n'aurait aucune influence. Pour un K plus petit, certes, le Vatican aurait peu de chances d'être sélectionné (et cela représentent mieux la réalité en quelque sorte), mais s'il était, il aurait un impact.

Il n'y a pas de méthode prédéfinie pour choisir un certain K. Cependant, il est possible de comparer chacun d'entre eux afin de trouver le meilleur. Ainsi, le set de données doit être séparé en 2 parties : une d'"entraînement" et une de "test". La première contient les données de référence que l'algorithme va utiliser. Elle est constituée d'à peu près 90% [5] des données. Le reste est utilisé afin de tester l'algorithme pour chaque k. Étant donné que la classe de ces données est déjà connue, en la comparant avec les réponses de k-nn, il est possible, pour chaque k, de calculer son pourcentage de succès.

5. valeur obtenue utilisant la validation croisée à 10-blocs

Le pourcentage peut être calculé en divisant le nombre de prédictions correctes par le nombre de prédictions réalisées.

Il est souvent important de définir l'efficacité d'un algorithme.Et c'est d'autant plus vrai pour K-NN. Ayant besoin d'estimer la précision pour trouver le meilleur K, il ne sagit pas d'une simple comparaison entre différents algorithmes. La validation croisée (*cross-validation*) est un modèle de calcul d' "à quel point le modèle est prédictif". Voici la méthodes utilisée :

La validation croisée à K-blocs (*K-fold cross-validation*). Selon le guide de Jason Brownlee [4], la procédure se déroule en 4 étapes :

I. Mélanger le set de données
II. Diviser le set en K groupes (K= 10 sera utilisé.Il s'agit d'une des valeur qui permet, de manière générale, d'obtenir les meilleurs résultats. Un onzièmes bloc est créé si le set de données contient un reste après la division entière par 10)
III. Pour chaque groupe :
    i. séparer le groupe et l'utiliser comme set de "test"
    ii. utiliser le reste comme set d'"entraînement"
    iii. évaluer puis enregistrer le score.
IV. Combiner les scores de réussite

Une explication plus exhaustive de la validation croisée à K-blocs peut être trouvée à l'intérieur du chapitre 5.1.3 de *An Introduction to Statistical Learning with Applications in R* [GJ13]

Chapitre 2

Préparation des données

2.1 Analyse exploratoire des données

Dans un monde idéal, un set de données serait composé de milliers de données toutes complètes et bien réparties entre tous les groupes. Cependant, le monde n'est pas idéal et les données ne sont pas prêtes à un emploi direct. Une préparation spécifique des données est ainsi nécessaire. Non seulement pour éviter des résultats imprécis mais bien plus pronfondément, pour assurer que certains algorithmes tel que K-NN recoivent les données formatées selon leurs besoins et puissent fonctionner. Ainsi, ce chapitre s'occupera de présenter les différentes étapes de la préparation de données brutes pour obtenir un set de données immédiatement utilisable par un algorithme. Les différentes étapes ont été inspirées du guide de préparation de données de Terence Shin [1]. L'analyse exploratoire des données revient à mieux comprendre chacune des données afin de simplifier les prochaines étapes.

2.1.1 Type des données

Déterminer le type des variables est crucial pour la suite. De manière générale, il convient de choisir le modèle d'apprentissage après avoir effectué cette étape. Mais étant donné que l'inverse a été effectué, il est nécessaire d'utiliser des variables adaptées (ou pouvant le devenir) à KNN.

Dans le questionnaire (Cf. annexe C), il y a 4 types de variables :

- Des variables catégoriques (Option spécifique, option complémentaire et Langue 3)

- Des variables binaires (trois ans de latin, Homme/Femme)

- Une variable catégorique mais pouvant être transformée en une donnée ordinale en gardant une unité de sens (métier en tête : non, plutôt non, plutôt oui, oui)

- Des variables ordinales (sur une échelle de 1 à 6)

KNN, comme vu précédemment n'utilise que des variables ordinales. En effet, KNN utilise une notion de distance [3]. Dès lors, il est impossible pour l'algorithme de traiter avec sens les autres types de données. C'est pourquoi les relations non linéaires sont exclues à cause de leur difficulté à être converties numériquement.

2.1.2 Recherche de cas particuliers

Comme présenté précédemment, le choix d'une valeur k influence fortement ce que représente le modèle. Malencontreusement, dans des sets de données de petite taille (140 données ont été recueilles (cf. annexe B)), la valeur k se retrouve de même réduite. Cela implique que le modèle est très sensible à certaines tendances et que des données atypiques ont une influence élevée. Ces données sont appelées bruits et peuvent ou non révèler une tendance particulière.

Ces cas particuliers peuvent être détectés de plusieurs manières

- Rechercher à l'intérieur de la feuille de données : il est possible que certaines données possèdent des valeurs erronées ou absurdes.
- Calculer le Score Z : le Score Z quantifie la singularité d'une variable par rapport à une distribution normale. il est calculé ainsi :

$$Z = \frac{x-\mu}{\sigma}$$

où

$$x = \text{la valeur observée}$$

$$\mu = \text{la moyenne de toutes les valeurs}$$

$$\sigma = \text{l'écart-type}\ [1]\ \text{de toute les données}$$

Plus Z est grand, et plus la valeur s'écarte de la distribution normale. De manière générale, lorsque Z est plus grand que 3, la données est considérée comme un cas extrême.

1. L'écart-type représente la dispersion des données autour de la moyenne. Elle est calculée en prenant la racine carré de la variance qui se calcule en prenant la somme de toutes les distances au carré entre chaque point et la moyenne de tous ces points.

Les bruits peuvent être alors supprimé ou non. D'autres manières plus complexes (Notamment : IQR et le test d'hypothèse) sont possibles mais ne seront pas abordées.

2.2 Traitement préalable des données

2.2.1 Imputation des éléments manquants

Que vaut 5 soustrait à quelque chose qui n'existe pas ? Knn s'appuye sur le calcul de distance et pour se faire, il a besoin d'avoir deux valeurs numériques qui existent. Dans certains cas, des sets de données peuvent être incomplets. C'est alors un problème majeur pour KNN. Bien qu'il soit possible de supprimer des colonnes entières de données, il est également possible d'imputer de nouvelles valeurs.

Voici différentes possibilités :

- Ajouter la valeur moyenne de la classe de la donnée : À première vue, cela fait sens. Assurément, c'est la manière de réduire au maximum l'impact de la valeur, car la valeur moyenne représente toutes les valeurs possibles équitablement (si une donnée se distance de 0,5 de la moyenne, elle le sera aussi de la valeur imputée). Mais cette manière de procéder est loin d'être fiable. Pour donner un exemple, que se passerait-il si les valeurs sont répartie entre deux sous-groupe [2] ? la moyenne représenterait de toute manière la classe dominante et la valeur perdrait du sens.
- Utiliser l'algorithme pour trouver la valeur manquante. Même si cette solution paraît évidente, elle reste néanmoins efficace. A deux conditions : l'algorithme doit avoir un pourcentage de fiabilité assez élevé et le nombre de valeur à trouver minime. Il est facile de comprendre la perte de précision en s'imaginant une multiplication de pourcentages ($70\% * 70\% = 49\%$) . Accessoirement, 1-NN [3] se prête relativement bien pour cette tâche.
- Utiliser la représentation de la valeur : En fonction de ce que numérise la valeur, il fait du sens de choisir une valeur de la même manière. Une illustration de cette idée serait la valeur manquante obtenue par un lancer de dé. Il ferait donc sens de choisir une valeur aléatoire avec le même pourcentage de chance que les dés. Ou alors : Admettons que

2. L'exemple de classification entre bon et mauvais client d'une banque em est une excellent preuve. Prenant en compte 2 variables qui sont le capital et la dette de chaque client, est un bon client la personne qui possède une petite dette et un petit capital ou une grande dette et un grand capital. Considérant 10% des clients comme ayant un grand capital, La valeur moyenne n'aurait que 90% de réussite car le capital serait toujours considéré comme petit. Le pourcentage de réussite empire avec le nombre de sous-groupes.
3. K-NN avec une valeur K=1.

98% des travailleurs d'une entreprise sont des hommes. Il est raisonnable de créer une nouvelle valeur qui aurait 98% de chance d'être "homme" .

2.2.2 Encodage des éléments

L'encodage des élements se résume à la conversion des valeurs "brutes" en valeurs numériques utilisables par un algorithme. Les valeurs peuvent être réparties en deux catégories. L'une est constituée de valeurs ordinales, l'autre du reste. Numériser des valeurs se fait aisément.

Pour les valeurs ordinales, une seule attention est nécessaire, convertir les valeurs dans un système décimal d'une même unité si l'algorithme utilise un concept de distance (En système binaire $111 - 101 = 010$ (2) ne vaut pas la même chose que $111 - 101 = 10$)

Pour le reste, il suffit d'ajouter un nombre représentant la hiérarchie (si la répartition des valeurs de la variable est linéaire, l'incrément serait égal entre chaque nombre) pour les valeurs en ayant une (pas du tout -> 1, un peu -> 2, beaucoup -> 3, entièrement -> 4).

Et si l'algorithme le permet (et ce n'est pas le cas de KNN), il suffit d'ajouter un nombre différent pour chaque valeur (acide -> 1, sucré ->2, amer -> 3, salé ->4, umami -> 5).

L'autre option est d'utiliser l'encodage "1 parmis n" (*One Hot encoding*) où pour chaque valeur possible une valeur binaire est associée.(Pour un bonbon acidulé : acide ->1 | sucré -> 1 | amer->0 | salé -> 0 | umami -> 0). Le gros désavantage de cette technique est d'augmenter énormément la dimensionalité.

2.2.3 Balancer le set de donnée

Comme mentionné ci-dessus, un set de données est très rarement équilibré. Il est même très probable que certain algorithme tel que KNN puisse obtenir un bon score de réussite en associant toujours la valeur la plus représentée (cf. exemple du test d'un cancer ci-dessous). C'est, en quelque sorte un sous-apprentissage extrême (cf chap1). Les solutions ci-dessous proviennent d'un tutoriel proposé par Elite data science [5]

Le véritable problème provient notamment du petit nombre des données sous représentées. La solution la plus efficace serait de simplement récolter plus de données mais ce n'est pas toujours réalisable. Voici donc quelques manières de gérer ce problème :

- Sur-échantillonage (*Up-Sampling*) : Très simplement, il s'agit de dupliquer les données minoritaires jusqu'à obtenir un ratio 1 :1 avec les autres classes.

- Sous-échantillonage (*Down-Sampling*) : Inversement, si le set de donnée est assez grand, il s'agirait alors de supprimer aléatoirement des données majoritaires jusqu'à obtenir un ratio 1 :1.

D'un autre côté, Il est courant de se concentrer sur la manière d'obtenir un pourcentage de réussite (Par exemple, la méthode utilisée pour trouver le meilleur K (cf chap1) ne prend que le résultat final en considération. Si l'algorithme devait être utilisé pour déterminer si un patient est atteint d'une maladie qui apparait toutes les 1000 radiographies et qu'il prédirait non à chaque fois, il aurait une précisions de 99.9% ! Mais en réalité 100% des cas négatifs seraient trouvés tandis que 0% des cas positifs seraient trouvés).

2.2.4 Changer la méthode de calcul de la précision

> “ Tout le monde est un génie. Mais si vous jugez un poisson sur sa capacité à grimper à un arbre, il passera sa vie entière à croire qu'il est stupide ”
>
> *- Albert Einstein*

2 procédés, se basant sur le changement de "l'unité de mesure" de la précision peuvent résoudre ce problème :

Courbe *AUROC*[4] : La courbe *AUROC* est une des mesures les plus importantes pour calculer la performance d'un modèle.

Pour rester simple, il faut d'abord s'imaginer une situation de classification entre deux classes : positif et négatif. Les prédictions du modèle sont alors soit vraies, soit fausses. Il y a donc 4 états abrégés en VP, VN, FP, FN (cf. Table 2.1)

	Vrai	Faux
Positif	VP	FP
Négatif	VN	FN

TABLE 2.1 – les 4 états possibles d'une prédiction

La courbe ROC peut être calculée à partir des taux de faux positif et vrais positifs.

4. Explication tirée de l'article de Sarang Narkhede [2]

Le taux de vrai positif(TVP, *True Positiv Rate : TPR*) représente le taux de réussite de l'algorithme lorsqu'il répond vrai. Il est calculé ainsi :

$$\text{TVP} = \frac{VP}{VP+FN}$$

Inversement, le taux de faux positif (TFP, *False Positiv Rate : FPR*) représente le taux d'échec de l'algorithme lorsqu'il répond vrai. Il est calculé ainsi :

$$\text{TFP} = \frac{FP}{VN+FP}$$

En trançant le graphe de la courbe ROC obtenu par l'expression de la TVP en fonction de la TFP, l'aire entre le dessous de la courbe et la droite AUC représente AUROC. Comme le montre le graphique 2.1 l'aire est grande, meilleur est le score AUROC.

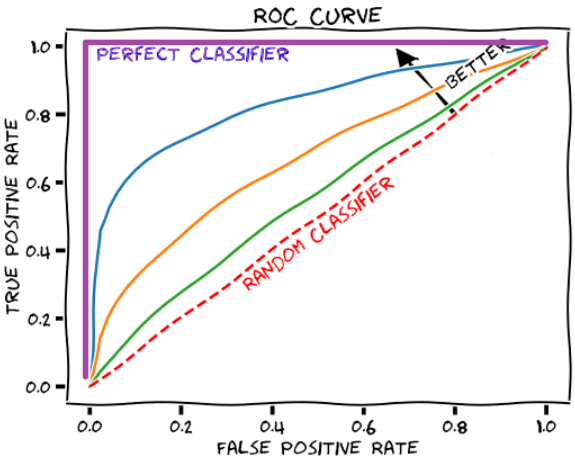

FIGURE 2.1 – Courbe ROC

L'autre solution est de pénaliser les erreurs de l'algorithme sur les classes proportionnellement à leur représentation. Se tromper sur une classe représentée 10 fois moins aura alors un impact 10 fois plus grand sur le score de précision. [5]

2.3 Séparation des données

La dernière étape revient à séparer le set de donnée en plusieurs groupes afin de pouvoir entraîner l'algorithme. La répartition se fait le plus souvent en deux groupes pricipaux

5. Un exemple d'un modèle pénalisé est présenté à l'intérieur du chapitre 6.4 d'*Applied Predicting Modeling*[MK13]

nommés *Train* et *Test*. Le groupe *Train* contient plus de 80% des données et sert de base pour l'algorithme. Le groupe *test* contenant le reste des données permet de tester la précision de l'algorithme avec des données non-utilisées. La séparation peut se faire avant le lancement de l'algorithme, cependant certaines validations croisées comme la validation croisée à k bloc séparent elles-même leur données.

Chapitre 3

Implémentation javascript et Prédiction d'une options spécifique

3.1 Algorithme KNN

Cette section s'occupe de présenter une implémentation en javascript de l'algorithme K-NN présenté au chapitre 1.

Les données sont formatées de la même manière que le code 3.1. C'est à dire, toutes les variables à la suite et en dernier le groupe auxquel appartient la donnée.

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
```

Code 3.1 – Exemple de données attendues

L'implémentation fondamentale de KNN se déroule en quatre étapes :

(a) Une fonction distance

(b) Classement de tous les voisins les plus proches à l'aide de la fonction distance

(c) Pour chaque k, recherche de la classe la plus représentée

(d) Test de précision pour chaque K et choix du meilleur K

La fonction getDistance présentée dans le code 3.2 retourne la distance Euclidienne [3] entre deux point. Pour se faire, elle calcule pour chaque coordonnée la distance au carré et l'additionne, retournant finalement la racine carrée du résultat .

```
1 getDistance(train,target){
        let result = null ;
3       for ( let n = 0; n < this.params.dataPoint.length-1; n++){

5           result += ((Number(train[n])-Number(target[n])))**2;
        };
7       return Math.sqrt(result);
      };
```

Code 3.2 – Fonction getDistance

La recherche des voisins les plus proches est assez triviale. Notons juste que le k (Code 3.3, ligne 3) fait partie des paramètres de l'algorithme et qu'il représente le nombre maximal de voisins qui seront traités.

```
2 getNN(){

4       for ( let j = 0; j < this.params.k ; j++){

6

8
            let nearest = set[0];
10          let var3 = null

12          for ( let i=1; i < set.length; i++){

14              let elementActuel = set[i];

16

                if ( this.getDistance(elementActuel,this.params.
                    dataPoint) < this.getDistance(nearest,this.params.
                    dataPoint)){
18                  nearest = elementActuel;
                    var3=i ;
20              };
            };
22          set.splice(var3,1);
            this.params.nN.push(nearest);
24      };

26  };
```

Code 3.3 – Fonction getNN

Pour trouver la classe la plus représentée, un tableau contenant un zero pour chaque valeur est incrémenté de 1 à chaque occurence. La suite permet de gérer des classes également présentes. Dans ce genre de cas, un tableau contenant les deux réponses est retourné. Si ce n'est pas le cas, seul le nom de la classe est retourné.

```
2
arrayToClass(arr, className, flag) {
4     let compare = [];
      for (let p = 0; p < className.length; p++) {
6       compare.push(0);
      }
8
      for (let i = 0; i < arr.length; i++) {
10      for (let j = 0; j < className.length; j++) {
          if (arr[i][arr[0].length - 1].trim() == className[j].trim()) {
12          compare[j] += 1;
          }
14      }
        let highestNumber = Math.max.apply(null, compare);
16    function numberOfOccurence(array, value) {
        var count = 0;
18      array.forEach((v) => v === value && count++);
        return count;
20    }

22    if (1 == numberOfOccurence(compare, highestNumber) && flag != true)
          {
        return className[compare.indexOf(highestNumber)];
24    } else if (1 == numberOfOccurence(compare, highestNumber) && flag
         == true) {
        return [className[compare.indexOf(highestNumber)]];
26    } else if (1 != numberOfOccurence(compare, highestNumber) && flag
         == true) {

28      let i = 0;
        let numberOfOccurences = numberOfOccurence(compare, highestNumber
            );
30      let classes = [];
        while (numberOfOccurences > i) {
32        classes.push(className[compare.indexOf(highestNumber)]);
          compare[compare.indexOf(highestNumber)] = 0;
34        i += 1;
        }
36      return classes;
```

```
    } else {
38    return "undefined";

40


42  return classes;

44  }

46  ]);
    }
```

Code 3.4 – Fonction arrayToClass

Finalement le test se fait en calculant le nombre de prédiction correcte pour chaque k. L'argument fold se réfère à la validation croisée à 10 blocs du code 3.6

```
1
  getSuccess(fold, success, dataTest, dataSet, kMax, distMax) {
3   for (let i = 0; i < dataTest.length; i++) {
      let nearest = this.getKnn(dataSet, dataTest[i], kMax, distMax);
5     for (let k = 0; k < kMax; k++) {
        let arr1 = Null;
7       let arr2 = [];

9       arr1 = nearest.slice(0, k + 1);

11      arr2.push(this.arrayToClass(arr1, this.classes));

13      if (arr2[0].trim() == dataTest[i][dataTest[0].length - 1].trim
            ()) {
          success[fold][k] += 1;
15      }
      }
17  }
  }
```

Code 3.5 – Méthode getSuccess

la validation croisée répartit les données en 10 blocs (avec la possibilité de créer un onzième bloc contenant le reste si le nombre de donnée n'est pas divisible par 10) puis lance le test sur chaque k.

```
2
  crossValidation(numberDataPerFold, numberOfFolds, kMax, distMax) {
```

```
4       this.data.arrayShuffle();
        let index = 0;
6       let dataSet;
        let dataTest;
8       for (let i = 0; i < numberOfFolds; i++) {
          if (i == 10) {
10
            this.data.getDataTest(this.data.data.length \% 10, index);
12        } else {
            this.data.getDataTest(numberDataPerFold, index);
14        }
          dataSet = this.data.dataSet;
16        dataTest = this.data.dataTest;
          this.getSuccess(i, this.success, dataTest, dataSet, kMax, distMax
              );
18
          this.data.resetData();
20        index += numberDataPerFold;
        }
```

Code 3.6 – Méthode crossValidation

3.2 Prédiction d'une option

Lancé sur des données sans aucune préparation , l'algorithme obtient un taux de réussite médiocre : autour des 30% pour le meilleur k. La figure 3.1 représente toutes les valeurs de k et leur pourcentage de réussite (également 1-NN obtient un taux de réussite trop élevé par rapport au reste des données).

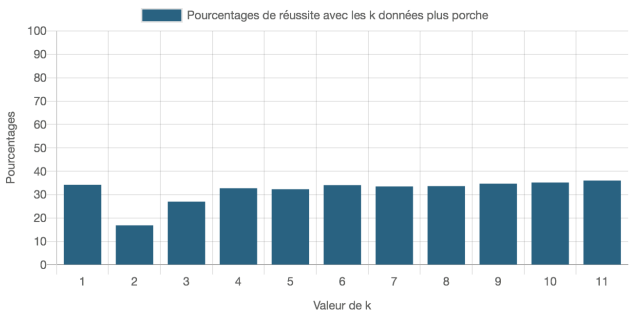

Figure 3.1 – Poucentages de réussite lors du premier essai

Le taux de réussite très faible ainsi que le manque de données laissent penser qu'il serait très difficile de traiter les données pour augmenter la précision.

Une autre possibilité de prédire une option est d'utiliser la distance par rapport à la moyenne de chaque groupe et d'en faire un pourcentage d'affinités (où une distance = 0 serait 100% et une distance maximum serait 0%). Mais même cette solution n'est pas efficace.Par exemple, la dérivation standard de l'option spécifique Biologie et Chimie (BIC) est tellement élevée qu'il n'est pas possible de tirer une quelconque tendance sans créer de sous groupe (et la répartion des valeurs a tendance à se séparer en deux groupes distincts sur plusieurs questions).(NB. les valeurs de dérivations standards suivantes sont multipliées par trois) Pour donner une idée de la représentation des valeurs de la dérivation standard, la figure 3.2 exprime une valeur plus ou moins maximale sans que la moyenne perde trop de sens.

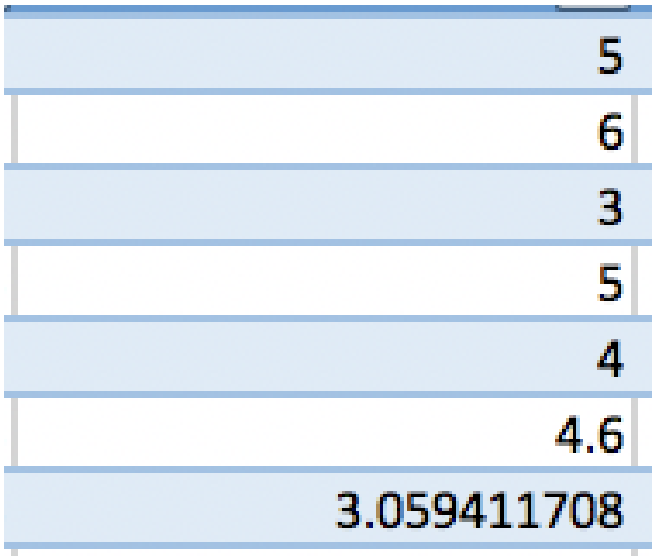

FIGURE 3.2 – Exemple de valeur d'une dérivation standard (valeur multipliée par trois)

Pour revenir à l'exemple de l'option spécifique Biologie et Chimie, 55 données ont été recueillies. Malheureusement, pour pouvoir diviser un groupe en sous-groupe, il faut au moins 100 données. Comme le montre l'image 3.3, bien qu'il semblerait que quelques questions soient révélatrices d'une tendance, la majorité des valeurs ne sont pas fiables. De plus, de nombreuses options comme l'option spécifique art visuel ne possèdent qu'une minorité de donnée(3 données). Il parait alors très difficile d'obtenir un meilleur résultat grâce aux techniques du chapitre 2. La meilleure solution est de récolter plus de données pour balancer le set de données, éliminer les bruits, créer des sous-groupes.

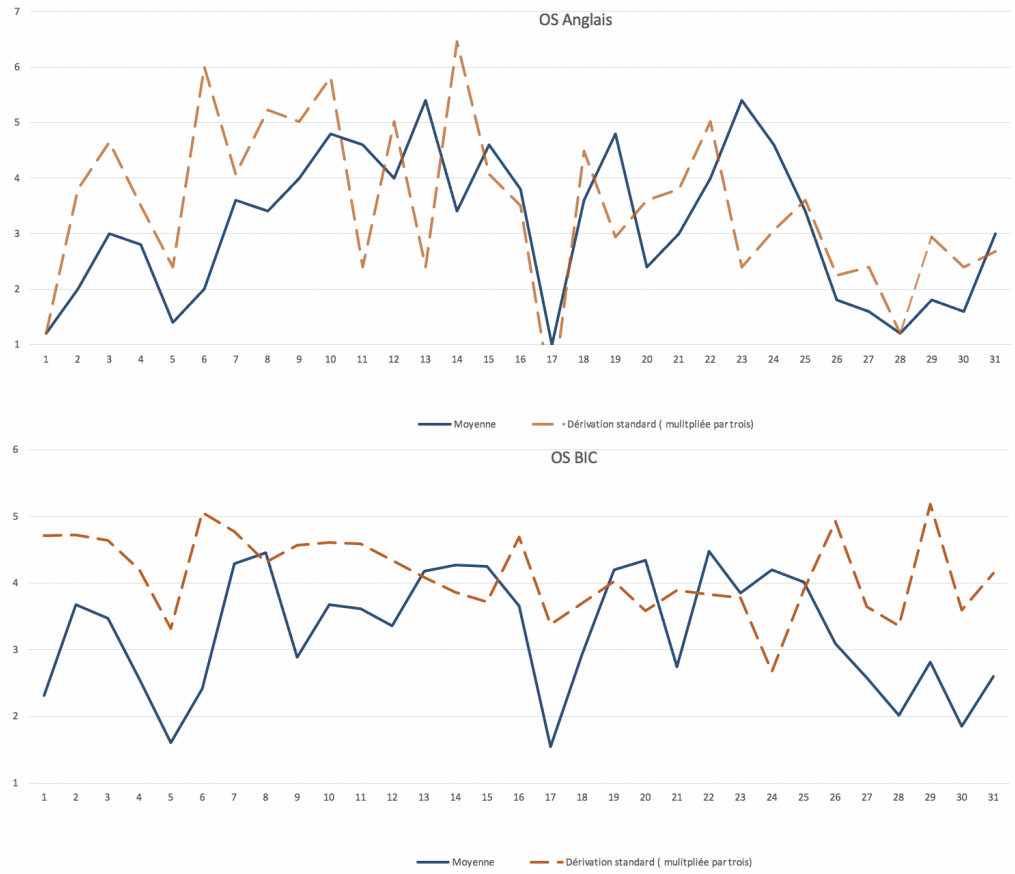

FIGURE 3.3 – Moyenne et dérivation standard de deux groupes (OS Anglais : 4 données, OS BIC : 55 données)

Conclusion

L'algorithme K-NN semble pouvoir bien s'adapter à la problématique. Son avantage de pouvoir se concentrer sur les individus permet de mieux associer un groupe sur ce genre de tâche. 1-NN pourrait même être une bonne solution en ayant suffisamment de données.

Cependant, beaucoup trop peu de données ont été récoltées. Celà implique qu'il serait présomptueux de se permettre d'utiliser ces bases pour prédire une option.

Il est certain qu'une bonne partie des questions fait du sens et que l'algorithme est un bon modèle pour la problématique. Dès lors, obtenir un nombre plus conséquent de données et pouvoir appliquer les différentes techniques du chapitre 2 serait la voie à suivre pour obtenir un résultat satisfaisant.

Bibliographie

[GJ13] Trevor Hastie Robert Tibshirani Gareth James, Daniela Witten. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013. Ouvrage exhaustif. Une très bonne référence en la matière.

[JPM18] Luca Massaron John Paul Mueller. *Machine Learning for Dummies*. learning made easy, 2018. Ouvrage simple à comprendre mais donnant un très bon point de vue sur la matière. 14

[MK13] Kjell Johnson Max Kuhn. *Applied Predictive Modeling*. Springer, 2013. Ouvrage complexe présentant les solutions à des problèmes de modélisation appliqués.

Webographie

[1] How To Prepare Your Data for Your Machine Learning Model, Towards data science, A Medium publication sharing concepts, ideas and codes. https://towardsdatascience.com/how-to-prepare-your-data-for-your-machine-learning-model-b4c9fd4 (dernière consultation le 25 mars 2021).

[2] Understanding AUC - ROC Curve, Towards data science, A Medium publication sharing concepts, ideas and codes. . https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (dernière consultation le 25 mars 2021).

[3] Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Euclidean_distance (dernière consultation le 22 mars 2021).

[4] Machine Learning Mastery, Making developers awesome at Machine Learning. https://machinelearningmastery.com/k-fold-cross-validation/ (dernière consultation le 29 mars 2021).

[5] Elite data science. https://elitedatascience.com/imbalanced-classes (dernière consultation le 22 mars 2021).

Table des figures

Liste des codes sources

Annexe A

Questionnaire

Questions Options Spécifiques

Les réponses seront utilisées afin de permettre à un algorithme de conseiller à des élèves de 1GY des options spécifiques.

Les questions se présentent sous deux formes: à choix multiple et sur une échelle de 1 à 6 où 1 est le plus petit et 6 le plus grand.

Il n'est pas nécessaire de trop réfléchir aux questions, le questionnaire ne devrait pas prendre plus de 5 minutes

Merci de votre participation !

1. Quelle est votre option spécifique?

- ◯ Latin
- ◯ Grec
- ◯ Italien
- ◯ Anglais
- ◯ Espagnol
- ◯ Physique+ application des mathématiques
- ◯ Biologie + chimie
- ◯ Economie + droit
- ◯ Arts visuels
- ◯ Musique

2. Quelle est votre option complémentaire?

- ◯ Application des mathématiques
- ◯ Physique
- ◯ Chimie
- ◯ Biologie
- ◯ Informatique
- ◯ Géographie
- ◯ Histoire
- ◯ Economie et droit
- ◯ Psychologie et pédagogie
- ◯ Philosophie
- ◯ Sciences religieuses
- ◯ Arts visuels
- ◯ Musique
- ◯ Sport

3. Quelle est votre langue 3 ?

- ◯ Anglais
- ◯ Italien
- ◯ Latin

4. Avez-vous suivi les trois ans de latin au CO?

- ◯ Oui
- ◯ Non

5. Êtes-vous de sexe masculin ou féminin ?

- ◯ Masculin
- ◯ Féminin

6. Avez-vous déjà un métier en tête ?

- ◯ Oui
- ◯ Plutôt oui
- ◯ Plutôt non
- ◯ Non

7. Quelle est votre affinité avec les arts martiaux et les sports de combats (boxe / taekwondo / karaté / ...) ?

1	2	3	4	5	6
◯	◯	◯	◯	◯	◯

8. Quelle est votre affinité avec les sports extrêmes (alpinisme / parachutisme / sport de glisse extrême : ski / snow / skate / ...) ?

1	2	3	4	5	6
◯	◯	◯	◯	◯	◯

9. Quelle est votre affinité avec les sports collectifs (football / hockey / baseball / ...)?

1	2	3	4	5	6
◯	◯	◯	◯	◯	◯

10. Quelle est votre affinité avec les sports de précision (tir à l'arc / billard / golf / ...)?

1	2	3	4	5	6
◯	◯	◯	◯	◯	◯

11. Quelle est votre affinité avec les sports mécaniques (moto-cross, ...)

1 2 3 4 5 6

12. Quelle est votre affinité avec les sports animaliers (équitation, agility,...)

1 2 3 4 5 6

13. Quelle est votre affinité avec les sports individuels (Athlétisme / cyclisme / sports nautiques / sport de glisse/ sport aériens/ ...) ?

1 2 3 4 5 6

14. Quelle est votre affinité avec les sports libres (sport en plein air / renforcement / ...) ?

1 2 3 4 5 6

15. Quelle est votre affinité avec la méditation et la détente spirituelle ?

1 2 3 4 5 6

16. Quelle est votre affinité avec la lecture ?

1 2 3 4 5 6

17. Quelle est votre affinité avec les créations visuelles (photographie / peinture / graphisme) ?

1 2 3 4 5 6

18. Quelle est votre affinité avec les créations manuelles (bricolage / mécanique / construction / DIY (Do It Yourself : créations , constructions et réparations faites maison)) ?

1 2 3 4 5 6

19. Quelle est votre affinité avec la musique ?

1 2 3 4 5 6

20. Quelle est votre affinité avec la cuisine, pâtisserie,...?

1 2 3 4 5 6

21. Quelle est votre affinité avec les jeux de sociétés ?

1 2 3 4 5 6

22. Quelle est votre affinité avec les jeux de stratégie et réflexions (échecs, ...) ?

1 2 3 4 5 6

23. Quelle est votre affinité avec l'e-trading et l'investissement (économie locale et mondiale / bourse /...) ?

1 2 3 4 5 6

24. Quelle est votre affinité avec la politique et l'actualité ?

1 2 3 4 5 6

25. Comment qualifieriez-vous votre organisation ? (par exemple : j'oublie mes affaires et perds mes document = 1, je fais toujours mes devoirs à la dernière minutes = 3, ...)

1 2 3 4 5 6

26. Quelle est votre affinité avec les mathématiques ?

1 2 3 4 5 6

27. A quel point aimez-vous lire et vous imprégnez de pensée d'auteurs philosophiques ?

1 2 3 4 5 6

28. A quel point éprouvez-vous du plaisir à trouver les réponses à des problèmes vous demandant de vous creuser la tête ?

1 2 3 4 5 6

29. A quel point est-il important pour vous d'apprendre de nouvelles langues ?

1 2 3 4 5 6

30. Quel serait la taille de la charge de travail que vous pensez vouloir assumer ? (par ex: le moins possible =1, je suis prêt à assumer une grosse charge de travail = 5 ou 6)

1 2 3 4 5 6

31. A quel point êtes-vous à l'aise avec les ordinateurs ?

1 2 3 4 5 6

32. A quel point la médecine, la chimie et la biologie sont présentes à l'intérieur de votre cadre familial ?

1 2 3 4 5 6

33. A quel point la physique, l'astronomie et l'ingénierie(mécanique, électronique,...) sont présentes à l'intérieur de votre cadre familial ?

1 2 3 4 5 6

34. A quel point les domaines du droit et de l'économie sont présents à l'intérieur de votre cadre familial ?

1 2 3 4 5 6

35. A quel point les sciences sociales (psychologie,...) et l'enseignement sont présents à l'intérieur de votre cadre familial ?

1 2 3 4 5 6

36. A quel point le secteur primaire est présent à l'intérieur de votre cadre familial ?

1 2 3 4 5 6

37. A quel point le secteur secondaire est présent à l'intérieur de votre cadre familial ?

1 2 3 4 5 6

Annexe B

Résultat du questionnaire

Voici le lien vers le document .xlsx en ligne :

https://eduetatfr-my.sharepoint.com/:x:/g/personal/perritazc03_studentfr_ch/Edls_ZyMp35JuTtBnxp-4aABwVX-XX7OGbGjDl5ek7i9TA?e=yrFDji

Annexe C

Code

Remerciements

Nous adressons nos remerciements à :

Monsieur Xavier Bays, expert en analyse de données

Monsieur Sacha Boetzel, enseignant responsable du travail de maturité

Monsieur Kevin Bourquin, orienteur professionnel

Monsieur Jérôme Charrière , enseignant responsable du travail de maturité

Toutes les personnes ayant pris part au questionnaire

Déclaration personnelle

Vibbeshan Nagalingam
Collège du Sud
Rue de Dardens 79
1630 Bulle

Christopher Perritaz
Collège du Sud
Rue de Dardens 79
1630 Bulle

(1) Nous certifions que le travail

Le Machine Learning
Apprentissage supervisé à l'aide
d'un algorithme de classification

été réalisé par nous conformément au « Guide de travail » des collèges et aux « Lignes directrices » de la DICS concernant la réalisation du Travail de Maturité.

(2) Nous prenons connaissance que notre travail sera soumis à une vérification de la mention correcte et complète de ses sources, au moyen d'un logiciel de détection de plagiat. Pour assurer ma protection, ce logiciel sera également utilisé pour comparer mon travail avec des travaux écrits remis ultérieurement, afin d'éviter des copies et de protéger mon droit d'auteur. En cas de soupçon d'atteintes à mon droit d'auteur, je donne mon accord à la direction de l'école pour l'utilisation de mon travail comme moyen de preuve.

(3) Nous nous engageons à ne pas rendre public notre travail avant l'évaluation finale

(4) Nous nous engageons à respecter la Procédure d'archivage des travaux de maturité en vigueur dans notre école.

(5) Nous autorisons la consultation de notre travail par des tierces personnes à des fins pédagogiques et/ou d'information interne à l'école

Lieu, date : Signatures :