Examen 'IA et Predictive Analytics'

Instructions:

- Copier/coller ce texte dans un mail adressé à thomas.baudel@esiee.fr à la fin de l'examen, et remplir les réponses en dessous de chaque question.
- 5 à 6 lignes de texte par question sont généralement suffisantes pour obtenir une bonne note. Des points supplémentaires sont attribués pour des réponses plus détaillées. Chaque question apporte 2 points. Il n'est pas nécessaire de répondre aux questions marquées [BONUS] pour obtenir la note maximale mais des réponses justes à ces questions rapportent des points supplémentaires. Certaines questions sont beaucoup plus faciles que d'autres, et elles ne sont pas (toutes) par ordre de difficulté croissante...
- **Lorsque vous utilisez une réponse trouvée sur internet, donner l'hyperlien des sources utilisées.**

Ghidaglia Boris

Conduite de projets en science des données

1: IA

a) Décrire la différence entre approche réaliste et approche utilitariste dans la démarche scientifique.

L'idée de l'utilitarisme est de tenter d'agir et de réfléchir pour augmenter efficacement le bonheur général. Cependant, comment se défini la notion de bonheur avec une approche réaliste ? Le bonheur est souvent accepté comme étant relatif, mais un réaliste qui considère le monde comme existant et indépendant aura certainement du mal avec cette manière de penser. Dans une démarche scientifique ces questions se posent : le but est-il d'augmenter le bien être général comme le ferait un utilitariste ? ou bien quelqu'un (qui ?) doit-il définir la notion de bonheur tel qu'un réaliste estime qu'elle existe ? Ces deux paradigmes constitue la différence entre ces deux approches.

b) Watson Studio: décrire une caractéristique importante de Watson Studio comme environnement de développement en science des données, qui le rend utile en situation de développement en entreprise.

Dans un monde ou le volume des données ne cesse de croître et où les besoins en puissance de calcul sont de plus en plus importants, Watson Studio est parfaitement pertinent en proposant des solutions déployées sur le cloud. En effet, aujourd'hui un travail en local limiterait grandement les possibilités d'un projet en science des données. D'autre part, Watson Studio propose des modèles pré entraînés qui permettent de gagner un temps considérable : on connaît le temps requis pour entraîner un réseau de neurones sur un jeu de données d'images par exemple.

2: programmation logique/chainage avant
Dans un langage à base de règles simple en chainage avant (on appelle cela un système de production) on a le programme:

var *input[]=[], result[]=[], i=1, tmp=0;*

when *input.length>0* ***and*** *i >= input.length* ***then*** *result.append(input[tmp]), input.removeAt(tmp), i=1, tmp=0;*
when *input[i] < input[tmp]* ***then*** *tmp=i, i=i+1;*
when *input[i] >= input[tmp]* ***then*** *i=i+1;*

Lorsque l'instruction *input=[2,0,5,4,9];* est exécutée, que contiendront les variables *result* et *input* en retour? Expliquer l'algorithme.

En retour *: input = [] et result = [9, 5, 4, 2, 0]*
Explications *:*
On prépare une liste input, une result et on initialise deux compteurs : i à 1 et tmp à 0. Tant que l'on a des éléments dans input et que notre compteur i est supérieur ou égal à la longueur de input, on va ajouter à notre result la valeur qui se trouve à l'index tmp de input, puis on va supprimer par la même occasion la valeur se trouvant à l'index tmp dans input. Enfin, nous allons remettre i et tmp à leurs valeurs initiales. Pourquoi faisons nous cela ? Il nous faut regarder les autres when pour comprendre. Le premier couvre le cas ou un élément de input au rang i est strictement inférieur à la valeur de input au rang tmp. Dans ce cas là, tmp prends la valeur de i et i est incrémenté. Cela signifie que l'on a trouvé un élément plus petit que celui que l'on avait retenu. Dans tous les autres cas, on incrémente simplement i.

3: Smart City: Trouver sur internet 3 logiciels commerciaux **professionnels** destinés à remplir un rôle similaire à celui du projet SmartDeliveries. En vous basant

sur la présentation commerciale, identifiez leurs principales caractéristiques démarquantes (quels fonctionnalités mettent-ils particulièrement en avant par rapport à la concurrence). Fournir les références utilisées.

- https://routific.com/ : simplicité et rapidité de planifications d'itinéraires => notion de praticité
- https://www.badgermapping.com/ : mise en avant des gains auquel il faut s'attendre avec l'utilisation de cet outil (20-25% de volume de livraison en plus possible) entre autre grâce à leur planification intelligente
- https://www.smartmonkey.io/ : mise en avant de la possibilité de personnaliser les itinéraires

4: trafic routier

a- Quelles sont les principales variables mesurées par un détecteur de trafic?
https://www.swarco.com/products/detection-sensors/traffic-counting/tdc3-traffic-detectors

- Vitesse individuelle ou moyenne des véhicules
- Identification du type de véhicule
- Identification du nombre de véhicules
- Occupation de la voie : détection des distances entre les véhicules
- Détection de présence de véhicules en sens inverse

b- Qu'est ce que le diagramme fondamental d'un détecteur de trafic, pourquoi est-il utile pour mesurer et prévoir la congestion?
https://tel.archives-ouvertes.fr/tel-00801762/document
"Le diagramme fondamental du trafic donne une relation entre le débit routier et la densité routière. Il peut être utilisé pour prédire le comportement d'un tronçon routier."
Le document précise aussi un point important : pour prévoir la congestion, on peut, connaissant la "vitesse libre" ("définie comme la vitesse maximale recommandée pour le tronçon routier") d'un axe routier, déterminer si les usagers roulent plus ou moins vite que celle-ci, de manière régulière ou non, etc … Cela donne des indications sur la congestion future de cet axe.

c- quel est le débit typique maximal d'un tronçon à une voie
*en zone urbaine : **X***
*sur voie rapide ou autoroute : **Y avec Y > X***
*Pourquoi cette différence? Cette différence s'explique par la **vitesse maximale réglementaire** imposée, d'une part. En effet, celle-ci est plus import*ante sur voie rapide et autoroute. D'autre part, les voies rapides et autoroutes sont faites pour minimiser les variations brusques de conduite : il n'y a pas (peu) d'obstacles, les virages sont amples, etc ...

5: temps de parcours

a) Quelles sont les principales variables prédictives du temps de parcours d'un camion de livraison en ville, par ordre d'importance décroissante? (déterminées en cours)

Distance, missionName, staticDuration

b) Citer 2 facteurs potentiels affectant les temps de parcours et difficiles à mesurer avec les données fournies.

La congestion ainsi que les accidents (certains axes sont plus sujets aux accidents que d'autres)

6: géoréférencement

On a un fichier d'adresses tel que vu en cours:

ID, BASE, KIND, CITY, FROM, TO

1, Docteur Bouchut, Rue du, Lyon, 1, 100

etc...

a) Décrire la logique d'une fonction python qui permet de retrouver l'identifiant de tronçon (ID) correspondant à chaque adresse, en supposant que la base d'adresses est stockée dans un tableau, et que l'on a une fonction distance(a, b) qui retourne la distance de Levenshtein entre 2 chaines.

17, rue Bouchut, Lyon

17 rue du Docteur Bouchot, Lyon

L'idée serait de calculer la distance de Levenstein entre notre adresse en entrée et toutes celles de la base. La ligne du tableau avec la plus petite correspondrait à l'adresse que nous recherchons. Nous retournons donc son ID. On pourrait utiliser un dictionnaire liant un identifiant à sa distance de Levenstein pour simplifier le processus.

b) proposer cette fonction en python (améliorant celle vue en cours)

```
linkid(s, addresses):
    """ addresses contient une liste de tronçons [id, base, kind, city, from, to], s
l'adresse à trouver """
    res = {}
    for row in addresses:
        res[row[0]] = distance(s, ' '.join(row[1:]))
    return min(res, key=res.get)
```

Prescriptive Analytics

Question 1.
Les affirmations suivantes sont-elles vraies ou fausses:

a- Un problème de décision est dans la classe de complexité NP si et seulement si il n'existe pas d'algorithme polynomial pour le résoudre.
Faux : *si il n'existe pas un algorithme pour vérifier des solutions données pour ce problème, en temps polynomial.*

b- Dans l'industrie, la majorité des problèmes d'ordonnancement sont résolus grâce à des heuristiques.
Faux : les heuristiques sont utiles, mais c'est la programmation sous contrainte qui est majoritairement utilisée (celle-ci pouvant, parfois, utiliser des heuristiques)

c- Le problème suivant possède exactement trois solutions:

```
u in {1,3}
v in {1,2}
w in {3,4}
x in {1,5}
y in {4,5}
allDifferent(u,v,w,x,y)
```

u=1, v=2, w=3, x=5, y=4,
u=3, v=2, w=4, x=1, y=5,
u=, v=, w=, x=, y=

d- L'algorithme de résolution de CP-Optimizer est un algorithme exact: si un problème d'optimisation est faisable, il garantit de trouver une solution optimale.
FAUX : une solution possible oui, optimale non.

Question 2.

a- En cherchant sur internet, décrivez un problème d'optimisation combinatoire non vu dans le cours dont la version de décision est un problème NP-Complet.
Knapsack problem

b- Décrivez une petite instance particulière de ce problème d'optimisation (avec des valeurs pour chacune des données).
On un un set d'objets associés à une masse : livre 1kg, ordinateur: 3kg, repas: 1.5kg Notre sac peut porter au maximum 3kg. Quelle est la masse maximale que l'on peut charger dans notre sac ? quels sont les objets qui ont permis d'arriver à cette valeur ? Contrainte : un objet ne peut être présent qu'une fois dans notre sac.

c- Donnez une solution faisable non-optimale et une solution optimale de cette petite instance.
non optimale : livre et repas = 2.5kg
optimale : ordinateur = 3kg

Question 3.

Deux principes fondamentaux de la Programmation par Contraintes sont (1) la recherche arborescente et (2) le filtrage du domaine des variables. Décrivez brièvement ces principes, leurs rôles et la façon dont ils sont mis en oeuvre durant la résolution.

La recherche arborescente consiste à mettre le problème que l'on souhaite résoudre sous la forme d'un graphe (ou arbre). Ce graphe représentera l'ensemble des solutions possibles. Si l'on comparait tous les chemins de cet arbre, on trouverait la solution optimale. Cependant, ce processus est très long. C'est pourquoi le filtrage du domaine des variables est utile, il permet, comme d'autres techniques (heuristiques), d'élaguer notre arbre et donc de diminuer notre champs de recherche.

Question 4.

Un problème classique en ordonnancement est le problème d'open-shop pour lequel un ensemble de n jobs doivent être exécuté sur m machines. Chaque job consiste en un ensemble de m opérations de durée connue, devant être exécutées dans un ordre arbitraire sur les machines. Plus précisément, si l'opération o_ij désigne la jème opération du job i, cette opération utilise la machine j et sa durée est D_ij. L'ordre des opérations du job i n'est pas connu d'avance: les m opérations o_ij d'un job i donné doivent être ordonnées mais cet ordre est libre. D'autre part, une machine ne peut pas effectuer plus d'une opération à la fois. L'objectif est de déterminer les dates de début et de fin de chaque opération de manière à minimiser la date de fin du plan. Ce problème d'optimisation est NP-difficile.

Les données d'entrée du problème sont donc:

- n, le nombre de jobs
- m, le nombre de machines
- D_ij (i in [1,n], j in [1,m]), la durée de la jème opération du job i

a- Décrivez un modèle CP Optimizer à base de variables d'intervalles et de contraintes noOverlap pour résoudre le problème d'open-shop.

b- [BONUS] Décrivez un modèle CP Optimizer pour une variante du problème d'open-shop pour laquelle les machines peuvent effectuer au plus deux opérations en parallèle.