Jérôme TRAN E5

Conduite de projets en science des données

1: IA

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

Une approche utilitariste est une approche dans laquelle la valeur morale d'une action est déterminée par sa contribution à l'utilité générale. De son côté, une approche réaliste est une approche pouvant être mise en place avec des contraintes.
Dans une démarche scientifique, la différence entre ces deux approches se traduit par une approche logique pour l'approche réaliste, et une approche statistique pour l'approche utilitariste.

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.

Watson Studio est un environnement collaboratif basé sur le Cloud. Celui-ci met à disposition des outils de machine learning et de deep learning afin de démocratiser le développement de l'IA.

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.

Cet algorithme trie la liste donnée en entrée par ordre croissant.
L'algorithme parcourt la liste en associant à tmp l'indice de la valeur la plus petite trouvée. Une fois la liste entièrement parcourue, la valeur de i est égale à la longueur de la liste, ce qui va permettre d'ajouter à la variable result, le plus petit élément trouvé. On retire également cet élément de la liste de départ. Les variables i et tmp sont réinitialisées, permettant de ré-effectuer ce procédé jusqu'au tri complet de la liste.
L'exécution de l'algorithme avec comme entrée la liste [2,0,5,4,9], donne pour la variable result=[0,2,4,5,9] et la variable input=[]

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.

Urbantz (https://www.urbantz.com/fr) :

- Optimisation des livraisons grâce à un modèle prédictif
- Traçabilité du livreur depuis l'application
- Communication avec le livreur et le manager depuis l'application
- Informations en temps réel

TourSolver (https://fr.geoconcept.com/app-plan-tournees-cloud) :

- Solution d'optimisation de tournées dans le Cloud
- Personnalisation de l'interface utilisateur
- Depuis le web ou application mobile
- Met en avant la réduction d'émission de CO^2

MyBoxMan (https://myboxman.com/fr/) :

- Plateforme collaborative
- Bas coût
- Flexibilité

4: trafic routier

a- Quelles sont les principales variables mesurées par un détecteur de trafic?

Les variables principales mesurées sont :

- Le flux en fonction du temps (heure, jours, semaine, mois, année)
- Le nombre de véhicules et leur vitesse
- Le taux d'occupation des voies

b- Qu'est ce que le diagramme fondamental d'un détecteur de trafic, pourquoi est-il utile pour mesurer et prévoir la congestion?

Dans le cas d'un détecteur de traffic, le diagramme fondamental est un ensemble de graphique permettant d'analyser la congestion. En effet, celui-ci nous permet d'observer, en fonction du temps, le flux, l'occupation, ainsi que le débit. De ce fait, il est possible d'effectuer une analyse de la congestion, grâce aux données collectées. On peut donc mesurer et prévoir la congestion à l'aide du diagramme fondamental d'un détecteur de traffic, dans une certaine mesure.

c- quel est le débit typique maximal d'un tronçon à une voie

- en zone urbaine
- sur voie rapide ou autoroute
- Pourquoi cette différence?

Plus l'occupation est élevée, plus les différences entre zone urbaine et autoroute se réduisent. La vitesse sur voies rapide étant plus élevés, le débit augmente. Cependant, si l'occupation est élevée, le débit se réduit jusqu'à atteindre celui en zone urbaine dans les mêmes conditions.

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)

Les variables prédictives les plus importantes sont :
- La distance
- Type de véhicule
- Expérience du conducteur
- Motivation du conducteur
- Circonstances locales non prises en compte par le modèle

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

- La connaissance du terrain par le conducteur
- La fatigue du conducteur

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

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
"""
    return id
```

Jérôme TRAN E5

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.

VRAIE

b- Dans l'industrie, la majorité des problèmes d'ordonnancement sont résolus grâce à des heuristiques.

VRAIE

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)
```

FAUX

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.

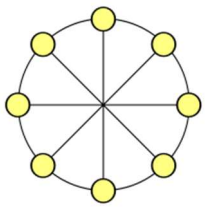
VRAIE

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.

Le problème algorithmique du chemin Hamiltonien est un problème NP-complet.
Un graphe hamiltonien possède un cycle hamiltonien, c'est-à-dire qu'il existe un chemin hamiltonien cyclique dans le graphe. Un chemin hamiltonien est un chemin passant par tous les sommets d'un graphe une seule et unique fois.

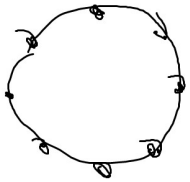
b) Décrivez une petite instance particulière de ce problème d'optimisation (avec des valeurs pour chacune des données).



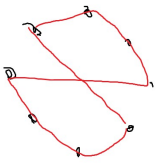
Ceci est un graphe hamiltonien à 8 sommets.

c- Donnez une solution faisable non-optimale et une solution optimale de cette petite instance.

Une solution optimale serait :



Une solution faisable non optimale serait :



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 en les tests successifs de toutes les valeurs possibles. Si une solution partielle ne rempli pas les conditions nécessaires à la résolution du problème, alors on revient à l'étape précédente. Le filtrage du domaine des variables, permet d'améliorer les capacités de résolution lors de la construction de l'arbre de recherche, en supprimant les valeurs inconsistantes des domaines de ces variables.

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.