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.**

Dan Seban

Conduite de projets en science des données

1: IA

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

En ce qui concerne la démarche scientifique, l'approche utilitariste est une approche qui tend à optimiser une fonction d'objectif. Par exemple, dans le cas du biomimétisme, observer le comportement des oiseaux pour élaborer un avion est suffisant à une approche utilitariste car on peut se servir de cette observation et s'en inspirer pour atteindre un objectif. Mais cela est peu suffisant pour une approche réaliste car il faut alors dépasser ce simple mimétisme et adapter notre solution au domaine informatique.

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 donne accès à des notebooks permettant d'embarquer des outils d'IBM. De plus il permet une mise en production rapide car il permet l'élévation de la solution au cloud d'IBM. Il permet un développement de la forme Drag&Drop avec des modèles prés à l'emploi. Il s'agit d'un environnement mettant à disposition de nombreux outil le rendant comme étant une solution tout en un.

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.

L'algorithme ci-dessus, parcours la liste en entrée, et ajoute l'élément le plus petit de cette dernière dans un résultat. Les valeurs de i et tmp étant réinitialisées à chaque ajout dans la liste résultat, le programme continu jusqu'à obtenir une input vide. Il s'agit d'un algorithme de tri par ordre croissant. L'algorithme s'arrête lorsqu'il n'y a plus de cas possibles.
La sortie sera donc result = [0,2,4,5,9] et 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.

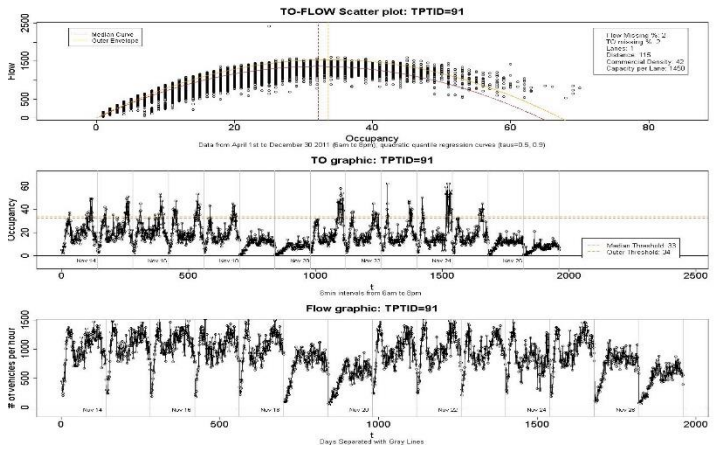
- Milestonesys est un logiciel dédier au trafic. Il s'agit d'une solution permettant l'accès à de nombreuses caméras routières. https://www.milestonesys.com/fr/solutions/industries/circulation/?gclid=Cj0KCQjwrrXtBRCKARIsAMbU6bGLyhmr79XklinN9XHeJuO6mvJEG6wotM6xf7PhfJUyB1ikgzp5kA8aAkOnEALw_wcB
- Karos, optimisation des points de rencontre des covoitureurs en fonction des itinéraires https://www.karos.fr/magazine/appli-karos/decouvrez-kronos-nouvel-outil-de-prediction-de-temps-de-parcours-333

4: trafic routier

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

Nous avons : Le temps d'occupation, le nombre de véhicule par heure et le flux

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



Le diagramme fondamental permet de comprendre les tendances d'occupation qui ont lieu via ce capteur. Nous pouvons donc rapidement comprendre si notre capteur détecte une saisonnalité dans le flux du trafic. Cela donne une indication sur la pertinence du capteur.

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

- en zone urbaine : 750 véhicules/heure
- sur voie rapide ou autoroute : 1500 véhicules/heure
- Pourquoi cette différence?
  La zone urbaine est soumise a des feux rouges et a des limitations de vitesse il y a donc des ralentissements. De plus la zone urbaine est le point de convergence des usagers. La voie rapide elle est un moyen d'accéder à la zone urbaine.

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)

L'heure de la journée, l'indicateur de flux maximum relatif au capteur, le type de détecteur et enfin le jour du mois. Il s'agit des variables démontrées en TP.

Cependant, les variables endogènes tels que le type du véhicule, l'expérience du conducteur, l'urgence de la course et enfin des circonstances locales apportent une réel valeur et son plus difficile à capturer.

b) Citer 2 facteurs potentiels affectant les temps de parcours et difficiles à mesurer avec les données fournies dans les fichiers fournis en TP.

Facteurs extérieurs, crevaison de pneux.

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

Tout ce que je demande, ce sont les différentes étapes à suivre, pas forcément le programme exact.

Dans un premier temps il faut comparer l'entrée, à savoir notre cible, avec notre base de données et ainsi retrouver la rue exacte. Pour cela nous calculons la distance de Levenshtein entre notre chaine de caractère d'entrée et l'ensemble des chaines de caractères de notre base de données pour en récupérer la plus proche. Ainsi, nous obtenons la rue. Par la suite il faut vérifier que le numéro indiqué se situe bien dans le tronçon isolé. Si c'est le cas alors il s'agit de la bonne adresse, sinon il y a une erreur et il faut alors proposer différentes alternatives comme par exemple les numéros proches de celui en entrée, ou bien vérifier si dans les tronçons voisins, le numéro d'entrée appartient à une intersection entre un tronçons voisin et le tronçon actuel.

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
```

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, plus précisément, un problème est dit de classe de complexité NP, si et seulement si il n'est pas possible de trouver une solution permettant de le résoudre en temps polynomial. Toutefois il est facile de vérifier une solution rapidement.

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

Faux, les problèmes d'ordonnancement industriels sont très complexe et demande une solution adaptée

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.

Vrai ! CP-Optimizer utilise des méthode hybrid

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 du sac-à-dos multidimensionnel.

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

On considère ici que le sac à dos a *d* dimensions, avec *d > 0* (*d-KP*). Par exemple, on peut imaginer une boîte. Chaque objet a trois dimensions, et il ne faut déborder sur aucune des dimensions.

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

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 programmation par contraintes repose sur le principe de satisfaction de contraintes. En effet, il est question ici de réduire la taille de l'espace des solutions, en propageant les contraintes.

Le filtrage du domaine des variables quant à lui, repose sur une base de connaissance et utilise cette base pour établir une solution. Le filtrage du domaine des variables fonctionne tel une recherche en profondeur dans un arbre de possibilités.

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.

```
dvar interval op[j in Jobs][p in Pos] size Ops[j][p].pt;
dvar sequence mchs[m in Mchs] in
  all(j in Jobs, p in Pos: Ops[j][p].mch == m) op[j][p];

minimize max(j in Jobs) endOf(op[j][nbPos]);
subject to {
  forall(m in Mchs)
    noOverlap(mchs[m]);
  forall(j in Jobs, p in 2..nbPos)
    endBeforeStart(op[j][p-1],op[j][p]);
}
```

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.