

1/ CONSIGNES GÉNÉRALES

L'épreuve d'informatique du concours CCINP proposait de découvrir des algorithmes d'intelligence artificielle appliquée à la détection de maladie. À partir d'une base de données comportant des informations sur le bassin et rachis lombaire, on cherchait à déterminer si un nouveau patient pouvait être considéré comme sain ou contractait une spondylolisthésis ou une hernie discale.

Le sujet permettait de balayer l'ensemble des compétences décrites dans le programme d'informatique en CPGE. Beaucoup de questions étaient des questions de cours.

2/ REMARQUES GÉNÉRALES

L'épreuve a été très bien réussie par la majorité des candidats. Les correcteurs ont pu constater assez peu d'erreurs de syntaxe sur les langages Python et SQL. Cette épreuve a été la première en version numérique pour les correcteurs : ils encouragent les candidats à utiliser des encres foncées qui permettent d'avoir une bonne visibilité, les encres pâles sont à éviter. L'absence d'utilisation de correcteur (blanc, effaceur) ne doit pas pour autant laisser penser que les copies doivent être raturées exagérément, l'utilisation du brouillon pour les algorithmes plus compliqués peut permettre de proposer une copie plus propre.

Les correcteurs conseillent cependant de bien lire les questions et ne pas oublier de répondre à certaines parties de celles-ci, de répondre succinctement aux questions d'analyse, de proposer des algorithmes simples à comprendre avec des noms de variables compréhensibles.

On retrouve cependant quelques erreurs ou maladresses :

- confusion entre / et //
- enchaînement de if alors qu'un elif serait plus judicieux
- mise en place de else inutile avec a = a
- problème d'affectation valeur = variable...

3/ REMARQUES SPÉCIFIQUES

Q1 : cette question portait sur une requête SQL simple. Aucune jointure n'était nécessaire pour répondre à la question si on analysait bien les données. Quelques candidats oublient les guillemets pour les chaînes de caractères ou au contraire mettent des guillemets pour les champs des différentes tables.

Q2 : il fallait réaliser une jointure dans cette question. Certains ne précisent pas à quelle table appartiennent les champs, ce qui était indispensable ici pour ne pas les confondre.

Q3 : cette question nécessitait de compter les états et de les grouper. Il faut connaître les mots clés particuliers tels que COUNT et GROUP BY.

Q4 : le module Numpy permet d'optimiser les opérations matricielles. Beaucoup de candidats se contentent de dire que cela permet uniquement d'avoir accès plus facilement à un élément ou que c'est optimisé en mémoire, voire que cela permettait de simplifier l'affichage dans la console.

Q5 : il est important de connaître les différents stockages (entier sur 4 octets, réel double précision sur 8 octets...) pour pouvoir estimer une capacité de stockage. La moitié des candidats ont réussi cette question. Beaucoup de candidats ne connaissent pas le lien entre 1 bit et 1 octet.

Q6 : cette question permettait de comprendre le stockage proposé dans Python. Il est important de savoir utiliser les listes. Il est nécessaire d'éviter de recopier plusieurs fois les mêmes lignes, ce qui veut dire qu'il est possible de remplacer cela par une itération finie.

Q7 : cette question nécessitait de comprendre une documentation, ce qui est une compétence indispensable en informatique. Il fallait également avoir bien appréhendé le stockage proposé et savoir utiliser les instructions de la librairie Numpy pour extraire des données (`groupes[k][ : , i ]`). Cette question a été mal traitée dans l'ensemble.

Q8 : l'interprétation des diagrammes a été bien faite dans l'ensemble.

Q9 : l'opération qui consiste à normaliser une liste de valeurs pour la ramener entre 0 et 1 est classique et doit être maîtrisée.

Q10 : la fonction `min_max` demandée est une fonction classique vue en cours. Il ne fallait faire qu'une seule boucle pour trouver à la fois le minimum et le maximum. Un nombre non négligeable de candidats n'arrivent pas à réaliser cette fonction correctement.

Q11 : pour calculer une distance euclidienne, il faut utiliser la formule suivante : en adaptant les vecteurs aux listes définies dans l'énoncé. De trop nombreux candidats ne pensent pas à initialiser la somme à chaque itération.

Q12 : le tri proposé était un tri fusion qui possède une complexité inférieure à celle du tri par insertion mais n'est pas en place. Les différents tris doivent être maîtrisés et leurs propriétés bien connues. Trop de candidats confondent la complexité rédactionnelle du programme avec sa complexité algorithmique.

Q13 : il s'agissait de compléter, par analogie avec les éléments donnés, les lignes manquantes de la fonction récursive. La dernière ligne a souvent été mal écrite par les candidats.

Q14 : les candidats devaient commenter les parties de la fonction proposée. Il ne s'agissait pas d'expliquer en détail chaque ligne mais de donner le principe de chaque partie en une ou deux lignes. Tout commentaire trop long montre un mauvais esprit de synthèse ou une mauvaise compréhension de l'algorithme.

Q15 : la question d'interprétation de la matrice de confusion a été comprise par la majorité des candidats.

Q16 : il fallait, dans cette question, déterminer quel paramètre `K` était optimal et critiquer l'algorithme KNN.

Q17 : l'écriture des fonctions `moyenne` et `ecarttype` est un résultat de cours. Une complexité linéaire étant demandée, il ne fallait pas réutiliser la fonction `moyenne` à l'intérieur de la boucle sur les composantes mais la définir en dehors de la boucle. Beaucoup de candidats se sont trompés sur ce point.

Q18 : l'écriture de la fonction `synthèse` était compliquée car il fallait bien maîtriser le stockage retenu et la donnée de sortie de la fonction. L'utilisation de la syntaxe propre à Numpy était conseillée ici.

Q19 : il s'agissait ici de réécrire sous forme de fonction Python une expression mathématique. Cette question n'a pas posé de difficultés.

Q20 : ce fut la question la moins bien traitée du sujet. En effet, il fallait bien comprendre la spécification pour ne pas oublier d'initialiser correctement les probabilités et boucler correctement sur les différentes grandeurs.

Q21 : la fonction `prédiction` était quasiment un résultat de cours similaire à la fonction `min_max` et n'a pas posé de réelle difficulté.

Q22 : l'utilisation du logarithme permet d'avoir des nombres mieux répartis et évite d'avoir des nombres trop petits proches de la précision machine.

Q23 : il était attendu, en plus de l'analyse des résultats, une critique des méthodes, qui en l'absence de nombreuses données ne peuvent donner des résultats très pertinents.

4/ CONCLUSION

La nature de l'épreuve d'informatique va rester du même ordre en termes de difficulté et de couverture du programme. Les candidats sont donc invités à lire les derniers rapports et à s'entraîner sur les derniers sujets afin de bien se préparer.

À noter qu'à partir de la session 2020, l'intégralité des réponses devra être rédigée sur un document réponse.