|  | **École nationale Supérieure d'Informatique (ESI), Alger**  **2CSSIQ2, 2CSSIQ3 et 2CSSIL2 2021/2022**  **Contrôle Final (CF)**  **Machine Learning (ML)** |
| --- | --- |

| **Jeudi 16 juin 2022 - [9h, 11h] - Partie 1**  *Documents non autorisés*  Il faut répondre sur cette feuille (pages 1 et 2) et la rendre cette feuille même vide  *Au cas d'insuffisance des copies, recopier les réponses sur une feuille d'examen séparée de celle des exercices* | |
| --- | --- |
| **Nom + Prénom :** ………………………………………………………….. | ◯ SQ2 ◯ SQ3 ◯ SL2 |

**QCM. Test de connaissance (6pts : 30mn)**

Pour chaque question, un choix erroné annule un autre correct. La note minimale par question est 0.

1) Pour chaque critère, sélectionner l'algorithme de classement qui la possède entre naive Bayes (NB), les machines à vecteurs de support (SVM), les forêts aléatoires (FA), les arbres de décision (AD) et la régression logistique (RL) **(0.25 X 6)** :

| **Critère** | **NB** | **SVM** | **FA** | **AD** | **RL** |
| --- | --- | --- | --- | --- | --- |
| **Utilise un algorithme d'optimisation** | ☐ | ☒ | ☐ | ☐ | ☒ |
| **Algorithme (original) déterministe** | ☒ | ☐ | ☐ | ☒ | ☐ |
| **Moins sensible au problème de sur-apprentissage (sans utiliser d'autres mécanismes** | ☒ | ☐ | ☒ | ☐ | ☐ |

2) Classer les modèles suivants du plus interprétable au moins interprétable **(0.25 X 3)** :

| **Modèle** | **Moins interprétable** | **Moyen** | **Plus interprétable** |
| --- | --- | --- | --- |
| **Régression logistique** | ◯ | ⬤ | ◯ |
| **Arbre de décision** | ◯ | ◯ | ⬤ |
| **Forêts aléatoires** | ⬤ | ◯ | ◯ |

3) Nous avons un dataset de 10 caractéristiques. Nous voulons sélectionner 7 caractéristiques. Dans ce cas, ordonner les algorithmes de sélection d'attributs en se basant sur leurs temps d'exécution **(0.25 X 3)** :

| **Modèle** | **Moins de temps** | **Moyen** | **Plus de temps** |
| --- | --- | --- | --- |
| **ANOVA** | ⬤ | ◯ | ◯ |
| **Forward Selection** | ◯ | ◯ | ⬤ |
| **Backward Elimination** | ◯ | ⬤ | ◯ |

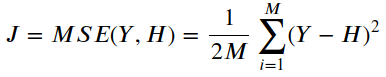
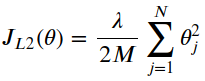
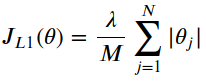
4) Nous voulons entraîner trois modèles de régression linéaire : un sans régularisation, un avec régularisation L2 et un avec régularisation L1. Initialement les trois modèles ont les mêmes paramètres θ. Nous avons utilisé le même dataset [X, Y].

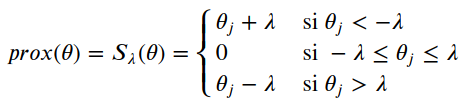
| X = | ⎡  ⎣ | 2 | 1 | ⎤  ⎦ | Y = | ⎡  ⎣ | 7 | ⎤  ⎦ | [θ0, θ1, θ2] = [1, 2, 1] |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 1 |  | 2 |  |
|  | 1 | 2 |  | 5 |  |

Etant donné 𝜆=1, 𝛼=1, la fonction du coût est MSE et pour L1 nous utilisons soft-thresholding, sélectionner les bonnes choix concernant le coût global (J) et les nouveaux paramètres : **(0.25 X 12)** :

| **Critère** | **Valeur** | **Reg-Lin** | **Reg-Lin + L2** | **Reg-Lin + L1** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Coût**  **J** | 5/6 | ☒ | ☐ | ☐ |
| 5/4 | ☐ | ☐ | ☐ |
| 5/3 | ☐ | ☒ | ☐ |
| 11/6 | ☐ | ☐ | ☒ |
| 5/2 | ☐ | ☐ | ☐ |
| 11/4 | ☐ | ☐ | ☐ |
| 15/4 | ☐ | ☐ | ☐ |
|  |  |  |  |  |
| **Nouveau**  **θ** | 0 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 | ☒ θ0 ☐ θ1 ☒ θ2 |
| 1/3 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☒ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 |
| 1/2 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 |
| 2/3 | ☒ θ0 ☐ θ1 ☒ θ2 | ☒ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 |
| 1 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☒ θ1 ☐ θ2 |
| 4/3 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☒ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 |
| 2 | ☐ θ0 ☒ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 | ☐ θ0 ☐ θ1 ☐ θ2 |
|  | | | | |
| ***Dans le cas où une valeur n'existe pas dans les choix, écrivez la ici :*** | | | | |
| **Coût J** |  | J= | J= | J= |
| **θ** |  | θ0= θ1= θ2= | θ0= θ1= θ2= | θ0= θ1= θ2= |

**Rappel :**



| **Jeudi 16 juin 2022 - [9h, 11h] - Partie 2**  *Documents non autorisés*  *Donner plusieurs réponses dont une est juste et le reste n'a pas de sens vaut un 0*  La réponse est sur la feuille d'examen ; Vous pouvez prendre cette feuille avec vous |
| --- |

**Exercice 01. Ingénierie des systèmes ML (7pts : 1h)**

Nous voulons concevoir un modèle qui estime la capacité d'un étudiant donné à réussir un semestre. Pour ce faire, nous avons utilisé la plateforme "talents" pour récupérer des informations comme le numéro d'inscription, le nom, le prénom, le nombre d'absences dans chaque module (15 c'est le max), la note de l'examen intermédiaire de chaque module, le coefficient de chaque module (max : 4), la wilaya, la spécialité du BAC (SN; SE et TM), le sex, l'age et le semestre (1 ou 2), année d'inscription et la moyenne finale du semestre. Afin de collecter le comportement des étudiants durant chaque module, nous avons envoyé des formulaires aux enseignants de chaque module. Le formulaire contient le numéro d'inscription, nom, prénom et photo de l'étudiant comme information. L'enseignant doit sélectionner un des choix : positif, neutre, négatif.

Nous n'avons pas reçu toutes les informations du comportement, mais nous ne voulons sacrifier ni cette caractéristique ni les échantillons sans réponse. Nous avons remarqué que la wilaya avait une petite corrélation avec la moyenne sauf pour une seule année. Aussi, l'âge est souvent entre 17 et 25 sauf pour quelques cas spéciaux (nous ne voulons pas les sacrifier aussi). En analysant les promotions, nous avons réalisé que chaque année la nouvelle promotion est similaire à celle d'il y a 3 années.

**A) Préparation de données (3pts)**

1. Décrire comment fusionner les données de "talents" et du formulaire en ligne **[0.25pt]**

| * **Nous appliquons une jointure sur le numéro d'inscription (0.25)** |
| --- |

1. Dans un premier temps, nous voulons entraîner un modèle pour estimer la réussite des 1CPI 1 semestre. Décrire les caractéristiques à garder (en justifiant pourquoi le reste n'est pas gardé), les opérations appliquées sur les caractéristiques et leurs représentations. La réussite est exprimée sous forme de notation ABCDEF et le modèle est discriminatif. Nous préférons des caractéristiques détaillées afin d'avoir plus de précision. **[1.75pt]**

| * **Supprimer numéro d'inscription, nom, prénom: ces caractéristiques sont inutiles puisque le numéro d'inscription est unique ; le nom et le prénom (même s'ils ne sont pas toujours uniques) n'ont aucune relation avec la réussite des étudiants, Supprimer wilaya puisque cette caractéristique ne présente aucune corrélation avec celle de la sortie sauf pour une année (une exception) (0.25)** * **Appliquer une coupure sur l'âge : si inférieur à 17 il sera égal à 17, si supérieure à 25 il sera égale à 25 (0.25)** * **Normaliser nombre d'absences\_module /15, note CI\_module / 20, coefficient\_module/4, (age - 16)/(25-16) (0.25)** * **Discrétiser la moyenne vers ABCDEF et appliquer OneHot (0.25)** * **Ajouter une catégorie "NA" (not available) à comportement\_module (0.25)** * **Ajouter une caractéristique "promotion" (A, B, C) calculée de l'année d'inscription et appliquer OneHot (0.25)** * **OneHot spécialité BAC (3 éléments), sex (2 éléments), comportement\_module (4 éléments) (0.25)** |
| --- |

1. Nous voulons généraliser le modèle sur le reste des niveaux et des semestres (avoir un seul modèle). Refaire la même chose. **[1pt]**

| * **En se basant sur les caractéristiques passées, ne ne pouvons pas utiliser les caractéristiques de chaque module à part (puisque les modules se changent : nombre, coefficient, etc.)** * **Ajouter moyenne CI (notes et coefficients des modules) puisque les modules de chaque niveau et semestre ne sont pas les mêmes (0.25)** * **Ajouter la moyenne des absences ou la somme (0.25)** * **Utiliser la catégorie majoritaire parmi celles =/= NA des comportement des modules (0.25)** * **Ajouter "semestre" et "niveau" (0.25)** |
| --- |

**B) Entraînement et validation (3pts)**

1. Nous avons remarqué une relation de causalité entre la catégorie TM de la caractéristique "spécialité du BAC" et la classe "F", le reste des catégories ont des corrélations égales avec toutes les classes. Donc, nous avons décidé d'utiliser une condition sur cette caractéristique (SI BAC = TM ALORS classe = F) et entraîner le modèle sur le reste des caractéristiques. Discuter cette solution : va-t-elle améliorer la performance ou non ? comment ? **[0.5pt]**

| **Oui, elle va améliorer la performance**   * **Moin de caractéristiques de bruit (noise features) donne plus de chance aux autres caractéristiques ; éviter le problème de sur-apprentissage** * **Moin de temps d'entraînement et d'estimation** * **Moin de paramètres (dans le cas de la régression logistique, SVM ou naïve Bayes)**   **(0.25) \* 2 raisons raisonnables (autres raisons peuvent être acceptées si elles sont logiques)** |
| --- |

1. Nous avons entraîné un arbre de décision CART sur les caractéristiques choisies.
   * Nous avons remarqué que certaines caractéristiques ne sont pas utilisées. Comment l'algorithme a fait pour les ignorer ? **[0.5pt]**

| * **Les arbres de décision utilisent une métrique (Gini ou entropie) afin de sélectionner la caractéristique et sa valeur qui peuvent mieux diviser l'ensemble des Y de ce nœud. (0.25)** * **L'opération est exécutée récursivement jusqu'à arriver à la condition d'arrêt. Dans ce cas, l'arbre ne prendra que les caractéristiques sélectionnées en considération lors de l'estimation (le reste sera ignoré) (0.25)** |
| --- |

* + Nous avons remarqué que le score de la validation est vraiment bas en le comparant avec celui de l'entraînement. Proposer deux raisons possibles et leurs solutions **[1pt]**

| * **Problème de sur-apprentissage (0.25) diminuer la profondeur max ou augmenter le nombre minimum des échantillons par noeud si le problème est causé par le fait d'avoir un échantillon par feuille, sinon utiliser les forêts aléatoires (0.25)** * **Problème de sous-apprentissage (même si le score de l'entraînement est mieux que celui de la validation, nous pouvons avoir ce problème) (0.25) augmenter la profondeur max ou diminuer le nombre minimum des échantillons par noeud si le problème est causé par le fait d'avoir beaucoups d'échantillons hétérogènes par feuille, sinon essayer d'avoir plus d'échantillons (0.25)** |
| --- |

1. Nous avons entraîné deux modèles SVM binaires (*nous avons fusionné les classes A B C ensembles et D E F ensembles*) : un avec hard-margin et un autre avec soft-margin. Nous avons remarqué que l'erreur du hard-margin est vraiment grande en la comparant avec celle du soft-margin malgrès que le F1-score est bon. Que peut-on dire à propos de la distribution du dataset ? **[0.5pt]**

| **Il y a un grand chevauchement entre les échantillons des deux classes : vu que soft-margin ignore les échantillons qui sont à l'intérieur de la marge dans le calcul de l'erreur, nous pouvons déduire que les échantillons chevauchés dans cette région (cercle ayant un diamètre égale à la marge) sont énormes (0.5)** |
| --- |

1. Nous avons entraîné un modèle de régression logistique. Le modèle a convergé rapidement et l'erreur a commencé à fluctuer. Proposer une solution. **[0.5pt]**

| **Une de ces solutions (0.5)**   * **diminuer le taux d'apprentissage (alpha)** * **utiliser un algorithme d'optimisation avec alpha adaptatif comme AdaGrad, Adam, etc.** |
| --- |

**C) Test et déploiement (1pt)**

Nous nous intéressons beaucoup plus à la classe "F" afin d'affecter ses étudiants à un programme de tutorat.

1. Quelle est la métrique utilisée pour sélectionner le modèle à déployer (P, R ou F1) et pourquoi ? **[0.5pt]**

| **Le rappel (R) (0.25) puisque nous voulons détecter le maximum possible des étudiants qui risquent de perdre leurs année, même si le système détecte quelques étudiants n'ayant pas de risque comme classe F, leur affecter un tutorat n'est pas considéré comme risque. (0.25)** |
| --- |

1. Quelle est la moyenne utilisée (Classe F, micro-avg, macro-avg ou weighted-avg) et pourquoi ? **[0.5pt]**

| **Classe F (0.25) Même si le système mélange entre les autres classe, la classe F est la plus alignée par rapport à nos objectifs (0.25)** |
| --- |

**Exercice 02. Réseaux de neurones (7pts : 30mn)**

***A) Réseaux de neurones à propagation avant***

Nous avons un problème de classement avec 5 classes de sortie. Afin de le résoudre, nous avons utilisé un réseau de neurones avec des couches cachées et une couche de sortie de 5 neurones. La fonction d'activation utilisée pour toutes les couches est la fonction logistique où les paramètres sont initialisés à 1.

* Expliquer les 3 limites de cette architecture en proposant une amélioration pour chacune **[1.5pts]**

| * **Initialiser les paramètres à une même valeur veut dire que les neurones de la même couche vont apprendre la même logique (0.25). La solution est d'initialiser les paramètres aléatoirement (0.25)** * **Utiliser la fonction logistique dans les couches cachées peut causer le problème de disparition du gradient (multiplication de plusieurs gradients proches à 0) (0.25). La solution est d'utiliser la fonction ReLu ou une de ces variantes (0.25).** * **Utiliser la fonction logistique dans la couche de sortie va nous générer une probabilité d'appartenance à chaque classe à part (la somme des probabilités n'est pas égale à 1) (0.25). La solution est d'utiliser la fonction softmax (0.25).** |
| --- |

***B) Réseaux de neurones convolutionnels (CNN)***

Voici une architecture CNN

|  |  |
| --- | --- |

1. Calculer le nombre des paramètres dans les deux couches Conv2D et MaxPool2D (nous utilisons seulement un canal d'entrée et un de sortie pour les deux) **[1pt]**

| **NParam(Conv2D) = 2 \* 2 + 1 = 5 (0.5)** | **NParam(MaxPool2D) = 0 (0.5)** |
| --- | --- |

1. Afin d'implémenter ces deux couches, combien de neurones devons-nous utiliser pour chacune ? **[0.5pt]**

| **NN(Conv2D) = 1 (0.25)** | **NN(MaxPool2D) = 0 (0.25)** |
| --- | --- |

1. Pour chaque neurone, le poids du biais est 1 et la fonction d'activation est ReLu. Calculer et dessiner les deux matrices X' et X'' (la structure de Conv2D est représentée dans le schéma ci-dessus). **[1pt]**

|  | **X'** | | |  | **X''** | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **3** | **1** | **4** |  | **5** | **5** | **5** |  |
|  | **1** | **5** | **0** |  | **5** | **6** | **6** |  |
|  | **0** | **0** | **6** |  | **5** | **6** | **6** |  |
|  | **(0.5)** | | |  | **(0.5)** | | |  |

1. Si nous supposons que la fonction d'activation utilisée est "ReLu" et le gradient passé par chaque cellule de X'' est de 1, calculer le gradient qui passe par X[3, 3] (colorée). **[1pt]**

| **Le gradient avant MaxPool est généré seulement par la cellule qui le génère (0.25)**  **X[3, 3] génère X'[2:3, 2:3]**   * **Le gradient de X'[2,2] est : 5 \* 1 / 9 = 0.56** * **Le gradient de X'[3, 3] est : 4 \* 1 / 4 = 1** * **Le reste est 0 (le max n'a pas été généré des deux 0)**   **Le gradient passé au Conv2D (par rapport ReLu) est la multiplication par son activation (0.25)**   * **Le gradient de X'[2,2] par rapport ReLu est : 0.56 \* 5 = 2.8** * **Le gradient de X'[3,3] par rapport ReLu est : 1 \* 6 = 6** * **Le reste est 0 (le max n'a pas été généré des deux 0)**   **Le gradient passé par ReLU, reste le même (0.25)**  **Le gradient passé par les poids de Conv2D est la multiplication par la source (0.25)**  **Puisque la source est toujours X[3, 3] = 1, on multiplie par 1**  **La moyenne est : (2.8+6 + 0 + 0)/4 = 2.2** |
| --- |

***C) Réseaux de neurones récurrents (RNN) et attention***

Voici l'architecture Transformer

|  |  |
| --- | --- |

1. Choisir la représentation détaillée du transformer (a, b, c ou d) étant donné une séquence d'entrée "s" de taille M et une séquence de sortie "d" de taille O. **[0.5pt]**

| **C (0.5)** |
| --- |

1. Expliquer comment le transformer est utilisé pour la traduction automatique du texte **[0.75pt]**

| * **L'entrée de l'encodeur est la phrase "s" complète (tous les mots) et sa sortie est un encodage des mots en utilisant l'attention (0.25)** * **L'entrée du décodeur est les derniers mots générés de "d" qui seront encodés en utilisant l'attention ; ce code est utilisé comme Q et le code de "s" est utilisé comme K et V pour avoir le mot suivant (0.25)** * **Le mot généré est fusionné avec les mots passés et utilisé comme entrée du décodeur (0.25)** |
| --- |

1. Expliquer comment les RNN sont utilisés pour la même tâche **[0.75pt]**

| * **L'encodeur utilise l'état passé et le mot actuel pour générer l'état actuel (0.25)** * **Le dernier état de l'encodeur est passé au décodeur (on peut utiliser les autres états avec attention) (0.25)** * **Le décodeur génère un mot et le passe comme entrée de l'encodeur afin de générer le mot suivant (0.25)** |
| --- |

**☙ Bonne chance**