|  |  |  |  |
| --- | --- | --- | --- |
| **Machine Learning (ML)**  **Examen Final**  **2h :**   * **15mn préparation** * **1h45 travail** |  | * Documents autorisés * Il faut rendre :   + QCM (cette feuille) ;   + une réponse sur papier (TP) ;   + un fichier Jupyter notebook avec votre nom. | |
| Nom et Prénom : | | | ○2CSSIQ3 ○2CSSIL2 |

**Questions (7pts : 15mn ~2mn/question)**

Pour chaque question, la faute sanctionne le juste et la note minimale locale est 0.

**Q1)** Sélectionner pour chaque méthode de sélection d’attributs son approche **(1pt)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Filtrage | Intégrée | Enveloppante | Pas une méthode de sélection |
| SVM | ○ | ○ | ○ | ⦿ |
| Arbres de décision | ○ | ⦿ | ○ | ○ |
| L1 | ○ | ⦿ | ○ | ○ |
| Pearson | ⦿ | ○ | ○ | ○ |

**Q2)** Choisir pour chaque proposition, les méthodes de sélection d’attributs **(1pt)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Filtrage | Intégrée | Enveloppante |
| Réduire le nombre des paramètres (RegLog) | ☒ | ☐ | ☒ |
| Améliorer la vitesse d’entraînement | ☒ | ☒ | ☒ |
| Éviter le problème de sur-apprentissage | ☒ | ☒ | ☒ |
| Faire une étape de pré-entraînement | ☒ | ☐ | ☒ |

**Q3)** Dans SVM avec HingeLoss, forcer une marge grande (petite valeur de C) veut dire : **(1pt)**

a) Plus d’observations aberrantes seront prises en compte dans l’erreur ⦿Vrai ○Faux

b) La norme euclidienne du vecteur des paramètres (forme «primal») sera plus grande ⦿Vrai ○Faux

c) Les valeurs de tous les paramètres seront plus grandes ○Vrai ⦿Faux

d) L’effet de la régularisation sera plus petit ⦿Vrai ○Faux

**Q4)** Dessiner le plan de séparation dans les deux cas **(1pt)**

|  |  |  |
| --- | --- | --- |
| SVM linéaire avec une marge max qui accepte 3 échantillons dans l’erreur J (avec marges) |  | SVM avec RBF avec une séparation sans erreur (parfaite) |
| y  x |  | y  x  **L’essentiel un cercle ou demi-cercle** |

**Q5)**Mentionner deux intérêts de l’utilisation de la fonction ReLu dans les couches cachées**(1pt)**

|  |
| --- |
| 1) Eviter le problème de la disparition des gradients (gradients vanishment)  En utilisant la fonction sigmoid, les gradients vont avoir de petites valeurs après chaque mise à jour |
| 2) Accélérer l’apprentissage  Calculs non complexes |

**Q6)** Sélectionner les propositions correctes à propos des auto-encodeurs **(1pt)**

☒ Un auto-encodeur est un modèle d’apprentissage non supervisée

☐ Un auto-encodeur doit toujours avoir des couches cachées de taille inférieure à celle de l’entrée

☒ Un auto-encodeur est un modèle génératif

☐ Dans un auto-encodeur la sortie est toujours l’entrée elle-même

**Q7)**Etant donné une couche Conv2D avec une taille = 4X4, stride = 1, padding = 0 et k=4. L’entrée de cette couche est un tenseur de taille 20X20X3 où 3 réfère aux valeurs RGB.

Sélectionner les deux propositions correctes **(1pt)**

☐ Nombre de paramètres est (20\*20\*3+1) \* 4 = 4804

☐Nombre de paramètres est 20\*20\*3 \* 4 = 4800

☒ Nombre de paramètres est (4\*4\*3+1) \* 4 = 196

☐ Nombre de paramètres est 4\*4\*3 \* 4 = 192

☐La sortie est de taille 20X20X4

☐La sortie est de taille 4X4X4

☐ La sortie est de taille 16X16X4

☒ La sortie est de taille 17X17X4

**TP (13pts : 1h30mn)**

Afin de classer les personnes en féminin/masculin selon des caractéristiques, nous avons utilisé deux datasets. Dans les deux datasets, chaque candidat possède un code. Notre but est de fusionner les deux schémas pour utiliser le dataset résultant pour l’entrainement seulement.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **sexe.csv** | | | |  | **gender.csv** | | | | |
| **candidat** | **taille(cm)** | **poids(kg)** | **sexe** | **code** | **size(feet)** | **weight(kg)** | **footSize(cm)** | **gender** |
| **M1** | **182** | **81.6** | **masculin** | **F3** | **5.41339** | **59.0** | **18** | **feminin** |
| **M2** | **180** | **86.2** | **masculin** | **M1** | **5.97113** | **81.6** | **30** | **masculin** |
| **M3** | **170** | **77.1** | **masculin** | **F4** | **5.74147** | **68.0** | **23** | **feminin** |
| **F1** | **152** | **45.4** | **feminin** | **M4** | **5.90551** | **74.8** | **25** | **masculin** |
| **F2** | **168** | **68.0** | **feminin** |  | | | | |
| **F3** | **165** | **59.0** | **feminin** |

**Mot de passe : 2esiMLMise1**

Pour chaque étape du cycle de vie d’un projet ML, nous présentons deux types de questions :

* Questions « a » : il faut répondre sur Jupyter notebook.
* Questions « b » : il faut répondre sur la feuille de l’examen.

Les questions ne sont pas bloquantes ; vous pouvez répondre à une question de type « b » sans programmer « a ». Aussi, vous pouvez répondre à une question dans une étape sans répondre à l’étape qui la précède.

**1- Préparation des données (4pts)**

1.a- Lire les fichiers, prétraiter les données afin d’avoir un seul schéma (table de données) et afficher la table finale. *Ce n’est pas la peine de faire des vérifications avant d’appliquer les prétraitements puisqu’on peut détecter les anomalies visuellement* **(2pts)**

|  |
| --- |
| ***# 1- Préparation de données***  ***# ===========================***  ***import pandas as pd***  ***# lecture d'au moins une table (0.25)***  ***s = pd.read\_csv("datasets/sexe.csv", skipinitialspace=True)***  ***g = pd.read\_csv("datasets/gender.csv", skipinitialspace=True)***  ***# suppression de l'attribut en plus (0.25)***  ***g.drop(["footSize(cm)"], axis=1, inplace=True)***  ***# renomage des colonnes (0.25)***  ***colonnes = {"code": "candidat", "size(feet)": "taille(cm)", "weight(kg)": "poids(kg)", "gender": "sexe"}***  ***g.rename(columns=colonnes, inplace=True)***  ***# Problème d'échelle (0.25)***  ***g["taille(cm)"] = g["taille(cm)"] \* 30.48***  ***# concaténation (0.25)***  ***sg = pd.concat([s, g], ignore\_index=True)***  ***# suppression des doublons (0.25)***  ***sg.drop\_duplicates("candidat", keep="first", inplace=True)***  ***# suppression de l'attribut en plus (0.25)***  ***sg.drop(["candidat"], axis=1, inplace=True)***  ***# affichage d'une table (0.25)***  ***Sg*** |

1.b- Quels traitements avez-vous utilisé et pourquoi en indiquant les problèmes qui peuvent se poser si on n’applique pas ce prétraitement ? **(2pts)**

|  |
| --- |
| * Suppression des attributs inutiles **(0.25)**: puisque l’attribut « *footSize(cm)* » n’existe pas dans la première table et l’attribut « *candidat/code* » est inutile après la fusion. Si on n’applique pas ce traitement, on aura des codes qui n’ont aucune signification dans le premier cas et des valeurs absentes (NaN) dans le deuxième **(0.25)** * Unification d’échelle **(0.25)** : puisque l’attribut « taille » est en « cm » dans la première table et en « feet » dans la deuxième. Si on n’applique pas ce traitement, on aura des valeurs non homogènes **(0.25)** * Nommage des attributs **(0.25)** : les attributs doivent avoir les mêmes noms avant la fusion. Si on n’applique pas ce traitement, on aura des attributs doublés (avec des NaN dans un ou l’autre) **(0.25)** * Suppression des doublons **(0.25)** : « F3 » et « M1 » existent dans les deux tables. Si on n’applique pas ce traitement, on aura des échantillons doublés qui peuvent affecter les modèles **(0.25)** |

**2- Encodage des attributs (1pt)**

2.a- Encoder les attributs d’entrée afin d’utiliser les données par naïve Bayes gaussien, arbre de décision (CART) et SVM **(0.5pt)**

|  |
| --- |
| ***# Rien à encoder ici (0.5)***  ***#Si vous avez encodé les classes ce n’est pas faux (côté programmation)***  ***#vous aurez (0.5) dans la partie « a »***  ***#mais pas « b » puisque c’est indiqué : attributs d’entrée*** |

2.b- Justifier votre choix **(0.5pt)**

|  |
| --- |
| Tous les attributs sont numériques et les 3 algorithmes supportent ce type d’attributs. Donc, on n’a pas besoin d’encodage. |

**3- Échantillonnage et division des données (1.5pts)**

3.a- On veut utiliser la totalité du dataset pour l’entrainement. Diviser le dataset en X\_train et Y\_train **(0.5pt)**

|  |
| --- |
| ***X\_train = sg.iloc[:, :-1].values # Premières colonnes (0.25)***  ***Y\_train = sg.iloc[:,-1].values # Dernière colonne (0.25)*** |

3.b- En se basant sur le dataset d’entrainement :

* Est-ce que le dataset d’entraînement est déséquilibré ? Justifier la réponse en indiquant la classe majoritaire en cas de déséquilibre **(0.5pt)**
* Mentionner une méthode pour régler le problème de déséquilibre lorsque le dataset est petit. Décrire comment elle fonctionne dans un paragraphe **(0.5pt)**

|  |
| --- |
| * Le dataset n’est pas déséquilibré **(0.25)** puisque les deux classes ont le même nombre d’observations (ou un pourcentage proche entre les deux classes) **(0.25)** * Une des méthodes de sur-échantillonnage (Duplication aléatoire, SMOTE, ADASYN, etc.) **(0.25)**. L’explication de cette méthode **(0.25)** par exemple :   + Duplication aléatoire : dupliquer aléatoirement les échantillons de la classe minoritaire   + SMOTE : chercher les K-Voisins de chaque point de la classe minoritaire et définir un nouveau point entre ce point et ses voisins   + ADASYN : comme SMOTE mais en favorisant les points dans les espaces non homogènes |

**4- Entraînement (5pts)**

4.a- Entraîner trois modèles : naïve Bayes gaussien (***nbg***), arbre de décision (***cart***) avec une profondeur max égale à 1, SVM (***svm***) avec le noyau linéaire en utilisant scikit-learn**(0.75pt)**

|  |
| --- |
| ***fromsklearn.svm import SVC***  ***fromsklearn.naive\_bayes import GaussianNB***  ***fromsklearn.tree import DecisionTreeClassifier***  ***# 0.25***  ***nbg = GaussianNB()***  ***nbg.fit(X\_train, Y\_train)***  ***#0.25***  ***cart = DecisionTreeClassifier(max\_depth=1)***  ***cart.fit(X\_train, Y\_train)***  ***#0.25***  ***svm = SVC(kernel="linear")***  ***svm.fit(X\_train, Y\_train)*** |

4.b- Discussion des algorithmes :

* Pourquoi nous avons choisi d’utiliser la variante gaussienne de naïve Bayes et pas les deux autres variantes (multinomiale, Bernouli) ? **(0.25pts)**
* Comment la profondeur max peut affecter le modèle CART**? (0.25pts)**
* Le temps d’entraînement *T(nbg, cart, svm) = (0.0016, 0.0009, 0.0012)*. Expliquer comment chaque algorithme fonctionne en motivant ces durées **(1.5pts)**
* Si on duplique un échantillon plusieurs fois, quel est l’effet de la duplication sur le temps d’entrainement de chacun des 3 algorithmes ? Expliquer en se basant sur leur algorithme. **(0.75pts)**
* Quel est l’effet de la duplication sur les modèles finaux en terme de prédiction ? Expliquer en se basant sur leur algorithme. **(1.5pts)**

|  |
| --- |
| * Puisque les attributs sont numériques **(0.25)** * On peut avoir des feuilles avec des classes non homogènes ; ça sert comme une condition d’arrêt **(0.25)** * Fonctionnement des algorithmes et motivation des durées :   + Dans naïve bayes gaussien, on essaye de sauvegarder la moyenne et l’écart type de chaque attribut par rapport à chaque classe, en plus de la probabilité à-priori **(0.25).** Donc, on aura besoin de calculer ces deux paramètres pour chaque attribut et chaque classe **(0.25)**   + CART essaye de trouver la division qui minimise la diversité Gini dans chaque nœud **(0.25).** Normalement CART est plus lent que NBG, mais puisque la profondeur max est 1, donc nous avons fait une seule division, ce qui peut motiver le temps **(0.25)**.   + SVM essaye de minimiser une fonction objectif en essayant de maximiser la marge entre les classes **(0.25).** Normalement SVM est plus lente que NBG, mais dans ce cas elle est plus rapide puisque le nombre d’échantillons est minimal et le résultat est peut-être dû au fait que les deux classes sont bien séparées **(0.25)** * Les 3 algorithmes vont prendre plus de temps :   + dans naïve bayes gaussien, cela va affecter le calcul des paramètres pour la classe de l’échantillon dupliqué pour chaque attribut **(0.25)**   + dans CART, cela va affecter le calcul de l’index de diversité Gini **(0.25)**   + dans SVM, cela va affecter le calcul de la fonction objectif et l’estimation en cas de la forme « dual » où on utilise un noyau MxM **(0.25)** * L’effet de la duplication sur les algorithmes en termes de prédiction :   + dans naïve bayes gaussien, cela va affecter la moyenne et l’écart-type de la classe de l’échantillon en question ; i.e., on est en train de donner plus d’importance à cet échantillon (naive Bayes se base sur la distribution des échantillons : un modèle génératif)  **(0.5)**   + dans CART, cela va affecter le calcul de l’index de diversité Gini, mais même si sa valeur est modifiée, l’algorithme se base sur la séparation par attribut/valeur ; c.-à-d., la division ne change pas (seulement, on aura plus d’échantillons dans la feuille du duplicata) **(0.5)**   + dans SVM, cela va affecter le calcul de la fonction objectif (moyenne des erreurs de chaque échantillon) : il va attirer J totale vers J du duplicata. Mais, avec plusieurs itérations, la solution va converger de la même façon, puisque dans SVM ce qui compte c’est la position des échantillons en terme de leurs attributs. (**0.5)** |

**5- Test (1.5pts)**

5.a- Prédire les classes d’entrainement en utilisant un des modèles précédents. **(0.25pt)**

|  |
| --- |
| ***# 0.25***  ***Pred = nbg.predict(X\_train)*** |

5.b- Etant donné que le test a été appliqué sur les données d’entrainement :

* Toutes les classes de tous les modèles ont eu une précision et un rappel égalent à 1. Que peut-on dire à propos du dataset ? **(0.25pt)**
* Que peut-on déduire à partir du modèle cart (avec une profondeur max égale à 1) ? **(0.5pt)**
* Si on utilise un noyau radial (RBF) avec SVM, est-ce que cela va détériorer le résultat ? Expliquer pourquoi **(0.5pt)**

|  |
| --- |
| * Les deux classes du dataset sont bien séparables **(0.25)** * Il a fait une seule division, donc on peut dire qu’il a utilisé un seul attribut. En plus, il a donné des résultats parfaits, donc cet attribut est représentatif et suffisant pour séparer les deux classes **(0.5)** * Non **(0.25),** puisque un problème linéairement séparable peut être séparé d’une façon radiale **(0.25)** |

**ANNEXE**

Les fonctions et les classes qui n’existent pas dans les TPs :

**sklearn.naive\_bayes.GaussianNB** : naive bayes gaussien. Les fonctions sont les mêmes : fit, predict