MATH60629. Devoir.

Date de remise: 18 octobre 2024

Instructions :

- Veuillez inclure votre matricule HEC sur votre soumission.
- Le devoir est à rendre pour le 18 octobre (n'importe quand avec le lendemain). Vous devez en téléverser une version PDF sur ZoneCours.
- Ce travail vaut 20% de votre note finale et doit être fait individuellement.

1 Principes de base de l'apprentissage automatique (3pt)

Vos réponses à ces questions devraient être courtes (au plus 5 lignes).

1. (1pt) Quelle est la différence entre l'erreur d'entraînement et celle de généralisation ? Votre réponse doit aussi décrire une approche pour évaluer l'erreur de généralisation en pratique incluant les limites de cette approche (en d'autres mots, vous devez décrire les conditions sous lesquelles cette approche est fiable).
2. (1pt) Expliquez l'effet de varier le taux de régularisation (par exemple la régularisation Ridge) sur le dilemme biais-variance.
3. (1pt) Un collègue vous explique avoir entraîné un modèle pour prédire le prix médian des maisons à partir de données du marché montréalais. Le modèle fonctionne bien selon son ensemble de validation. Par contre, ce modèle obtient de mauvaises performances sur un autre ensemble constitué à partir de données du marché immobilier à Toronto. Expliquez ce résultat et suggérez une façon d'obtenir un meilleur modèle pour les données de Toronto.

2 Classification (16pt)

- Pour ce problème, nous allons utiliser le jeu de données synthétiques que j'ai généré et qui est disponible ici.

  Une fois les données accessibles dans votre répertoire courant, vous pouvez les charger avec les trois lignes suivantes :

```
data = np.load("a24_devoir_q2-classification.npz")
X = data["X"]
y = data["y"]
```

Les données sont encodées dans un vecteur numpy (*numpy array*).

La tâche est de prédire la classe de chaque donnée à partir de ses deux caractéristiques.

1. (4pt) À la suite d'une exploration des données, que remarquez-vous ? Que pouvez-vous dire de la performance (taux de bonne classification) de test d'un modèle n'utilisant que des frontières de décision linéaire pour ce problème ?
2. (2pt) Divisez votre jeu de données en entraînement et test. Le jeu de test doit faire 20% des données totales et vous devez utiliser ce paramètre pour la fonction de sklearn `random_state=123`.

   Votre réponse doit comprendre les quelques lignes de code pour diviser les données.
3. (1pt) Qu'est-ce que le fait de ne pas avoir gardé un ensemble de validation vous indique ?

Les prochains questions vous demandent d'utiliser un modèle bayésien naïf (naive Bayes).

4. (1pt) Quel modèle bayésien naïf devrait-on utiliser ? En d'autres termes, quelle est une bonne distribution pour modéliser les probabilités conditionnelles des classes ? Justifiez.
5. (2pt) Entraînez le modèle sur les données d'entraînement et obtenez son taux de bonne classification en entraînement et en test.

   Veuillez fournir les quelques lignes de code pour faire cela ainsi que les taux de bonne classification d'entraînement et de test.
6. (4 pt) Ré-entraînez le modèle en utilisant des prieurs de classe uniformes (fournissez le code pour cela). Que remarquez-vous en ce qui concerne le taux de bonne classification d'entraînement et de test ? Expliquez ce résultat.
7. (2 pt) Pensez-vous que cette observation est susceptible de se vérifier en général ?

3 Régression (38pt)

Dans cette question, vous allez entraîner un modèle k-NN et un réseau de neurones pour une tâche de prédiction d'une note en fonction d'un critique d'un item.

Les données à télécharger sont ici. Chaque donnée est une critique en format texte (en anglais) d'un produit disponible sur Amazon. [1]

1. Pour votre information, les jeux de données complets sont disponibles ici. Nous utilisons un sous-ensemble de la catégorie *Toys and Games*.

Dans le fichier chaque ligne correspond à une donnée. Chaque donnée contient une variable cible (y) suivie d'un court texte (x). La variable cible dénote la note de l'utilisateur pour un produit. C'est un entier pouvant prendre une valeur entre 1 et 5. Le texte est la critique. Il vous faudra d'abord diviser les caractéristiques des cibles.

Indice : vous pouvez utiliser la fonction `split('\t')`, pandas vous permettra aussi de facilement utiliser ces données.

1. (2pt) Nous allons modéliser ce problème comme un problème de régression (et vous pouvez utiliser l'erreur moyenne au carré pour l'évaluation de la performance). Donnez un avantage et un inconvénient de plutôt modéliser le problème comme en étant un de classification à 5 classes.

Prétraîtement des données (5pt)

2. (2pt) On vous demande de diviser les données en un ensemble d'entraînement (80% des données), un ensemble de validation (10%) et un ensemble de test (10%). Fixer la racine du générateur de nombres aléatoires à 1234 (`random_state=1234`)

   Veuillez fournir les quelques lignes de code pour diviser le jeu de données en deux.

3. (3pt) On vous demande ensuite d'obtenir une représentation sac à mots (bag-of-words) des caractéristiques (*features*). sklearn offre des fonctions pour y arriver.

   Pour limiter le temps d'entraînement requis, utilisez un maximum de 2 000 mots dans votre vocabulaire (`max_features=2000`) et la liste des mots vides de sklearn (`stop_words="english"`). Cette liste permet de retirer des mots qui à priori ne seront pas utiles à la prédiction. Utilisez les autres paramètres par défaut de la fonction.

   Nous vous demandons les quelques lignes de code de sklearn que vous avez utilisées pour encoder (et seulement encoder) les données d'entraînement, de validation et de test.

k plus proches voisins (k-nearest neighbhours) (13pt)

4. (3pt) Laquelle de ces trois fonctions de distance 'cosine', 'euclidean' et 'manhattan', vous semble la plus appropriée pour ce problème et pourquoi ?

5. (5pt) En fonction de votre réponse précédente, entraînez le modèle k-NN pour cette tâche. On vous demande d'essayer avec 1, 10, 50, 100 et 1000 voisins.

   Quelle est la performance de chaque modèle sur l'ensemble d'entraînement et de validation ?

   Veuillez fournir les quelques lignes de codes utilisées pour entraîner le modèle et évaluer sa performance.

6. (5pt) Pour quelle valeur de l'hyperparamètre obtenez-vous le meilleur résultat ? Expliquez ce résultat.

Réseau de neurones (10pt)

Quand vous instanciez vos réseaux de neurones, veuillez fixer la racine aléatoire à la valeur 1234 (c'est-à-dire `random_state=1234`).

7. (5pt) Vous allez maintenant entraîner une série de réseaux de neurones avec différents hyperparamètres. Utilisez l'option `early_stopping=True` et trouvez les hyperparamètres qui donnent les meilleurs résultats sur l'ensemble de validation (j'imagine que vous entraînerez environ 50 modèles différents). Je vous suggère d'explorer ces trois hyperparamètres : le taux d'apprentissage (*learning rate*), la taille du réseau et le taux de régularisation L2 (*L2 regularization term*).

   Dans votre réponse, veuillez inclure le code utilisé pour entraîner ces réseaux de neurones et pour obtenir la performance sur l'ensemble de validation (seulement cette partie du code).

   Quelle est la meilleure combinaison d'hyperparamètres et quelle est la performance sur l'ensemble de validation du modèle résultant ?

8. (5pt) Qu'avez-vous constaté par rapport à l'importance des différents hyperparamètres essayés ?

Comparaison (8pt)

9. (4pt) Si vous ne deviez conserver qu'une seule caractéristique (c.-à-d. un seul mot), lequel utiliseriez-vous et pourquoi ?

10. (2pt) Quelle est la performance finale de chaque modèle (k-NN, réseau de neurones) ?

    Veuillez fournir les quelques lignes de codes utilisées pour obtenir ce résultat.

11. (2pt) Trouvez un exemple pour lequel les prédictions des deux modèles diffèrent de plus de 2.0 et expliquez cette différence. (Notez que vous pouvez inventer de nouveaux exemples qui ne sont pas dans les données pour y arriver.)

Système en ligne (11pt)

Dans cette question, on vous demande de trouver un système en ligne qui pourrait utiliser l'apprentissage automatique pour faire des prédictions (bien sûr, vous ne saurez peut-être pas avec certitude s'il en utilise ou non). Un exemple d'un tel système est Google Traduction, mais vous ne pouvez pas utiliser cet exemple. Nous vous demanderons d'explorer le système en répondant aux questions ci-bas.

1. (2 pt) Décrivez la tâche d'apprentissage automatique que le système essaie de résoudre.
2. (3 pt) Décrivez les données (ou leur type) qui serait nécessaire pour entraîner ce modèle ainsi qu'une mesure de performance appropriée (vous n'avez pas nécessairement besoin d'utiliser une mesure existante).
3. (2 pt) Quelles sont les limitations du système ? Trouvez-en au moins deux, idéalement vous démontreriez leur effet sur le système.
4. (4 pt) Pour chaque limitation ci-dessus, en vous basant sur vos connaissances en apprentissage automatique, comment pourrait-elle être corrigée ?