esprit Se former autrement HONORIS UNITED UNIVERSITIES	**EXAMEN** Semestre : 1 ☒ 2 ☐ Session : Principale ☒ Rattrapage ☐

Module: Principes fondamentaux de Machine Learning **Classes:** 3A-3B **Date:** 17/1/2025
Heure: 9h **Durée:** 1h30

Documents autorisés: OUI ☐ NON ☒ **Calculatrice autorisée:** OUI ☒ NON ☐
Internet autorisé: OUI ☐ NON ☒ **Nombre de pages:** 10

ETUDIANT(E) Identifiant: . . . . . . Noms & Prénom:	Classe: . . . . . . Salle: . . . . .

Questions à choix unique : (4 points)

Pour chaque question, cocher la bonne réponse.

Question 1: Quelle fonction est assurée par l'apprentissage supervisé?

- ☐ Classification.
- ☐ Détection d'anomalie.
- ☐ Clustering.
- ☐ Réduction de la dimensionnalité.

Question 2: Quelle est la fonction principale de l'indice de Gini dans un arbre de décision?

- ☐ Mesurer la pureté des nœuds de l'arbre.
- ☐ Calculer la probabilité d'erreur de classification.
- ☐ Déterminer la profondeur optimale de l'arbre.
- ☐ Évaluer la variance des données dans les nœuds.

Question 3: Quelle est la définition du clustering en machine learning?

- ☐ Une technique supervisée de classification des données.
- ☐ Une méthode d'apprentissage non supervisée pour regrouper des données similaires.
- ☐ Un algorithme pour prédire des valeurs continues.
- ☐ Une technique pour réduire la dimensionnalité des données.

Question 4: Comment peut-on valider le choix du nombre optimal de clusters dans l'algorithme K-means?

- ☐ En traçant la courbe de variance cumulative.
- ☐ En calculant la matrice de confusion.
- ☐ En calculant l'erreur quadratique moyenne (MSE).
- ☐ En traçant la courbe courbe coude (Elbow).

Exercice 1 : (10 points)

Le dataset **"cars.csv"** contient des informations techniques sur des voitures, permettant d'analyser leurs performances, leur consommation de carburant et d'autres caractéristiques mécaniques. Il est composé des colonnes suivantes :

- **mpg** : *Miles per gallon* - Mesure de l'efficacité énergétique d'une voiture, exprimée en distance parcourue par gallon d'essence. Plus la valeur est élevée, meilleure est l'efficacité énergétique.
- **cylinders** : Nombre de cylindres dans le moteur. Un nombre plus élevé de cylindres est souvent associé à une plus grande puissance, mais aussi à une consommation de carburant plus importante.
- **displacement** : Cylindrée du moteur (en pouces cubes). Elle représente le volume total déplacé par les pistons dans le moteur, influençant la puissance et la consommation.
- **horsepower** : Puissance du moteur (en chevaux). Cette mesure indique la force motrice générée par le moteur, influençant directement les performances et l'accélération.
- **weight** : Poids du véhicule (en livres). Le poids impacte la consommation de carburant et les performances globales de la voiture.
- **acceleration** : Temps (en secondes) pour passer de 0 à 60 mph. Cette colonne reflète la rapidité du véhicule, généralement corrélée à la puissance du moteur.

Le dataset est chargé à l'aide de la bibliothèque Pandas en Python.

[1]:
```
import pandas as pd
df = pd.read_csv("cars.csv")
df.head()
```

[1]:

	mpg	cylinders	displacement	horsepower	weight	acceleration
0	18.0	8	307.0	130.0	3504	12.0
1	15.0	8	350.0	165.0	3693	11.5
2	18.0	8	318.0	150.0	3436	11.0
3	16.0	8	304.0	150.0	3433	12.0
4	17.0	8	302.0	140.0	3449	10.5

Partie I : Préparation des données

1. (**0.5 point**) Remplir le champ vide ("_____") par la commande permettant d'obtenir les dimensions du dataset **"cars.csv"**.

[2]:
```
___________
```

[2]:
```
(398, 6)
```

2. (**0.5 point**) Remplir le champ vide ("_____") par la commande permettant d'obtenir les noms des colonnes du dataset **"cars.csv"**.

[3]:
```
___________
```

[3]:
```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration'],
      dtype='object')
```

3. (**0.5 point**) Remplir les champs vides ("_____") par la commande permettant d'obtenir le nombre de valeurs manquantes dans chaque colonne du dataset **"cars.csv"**.

[4]:
```
df._________.__________
```

[4]:
```
mpg             0
cylinders       0
displacement    0
horsepower      6
weight          0
acceleration    0
dtype: int64
```

4. (**0.5 point**) Remplir le champ vide ("_____") par la commande permettant d'imputer les valeurs manquantes dans la colonne **'horsepower'** par la moyenne de cette colonne.

[5]:
```
# Remplacer les valeurs manquantes par la moyenne
df['horsepower'] = df['horsepower'].fillna(df['horsepower'].____________)
```

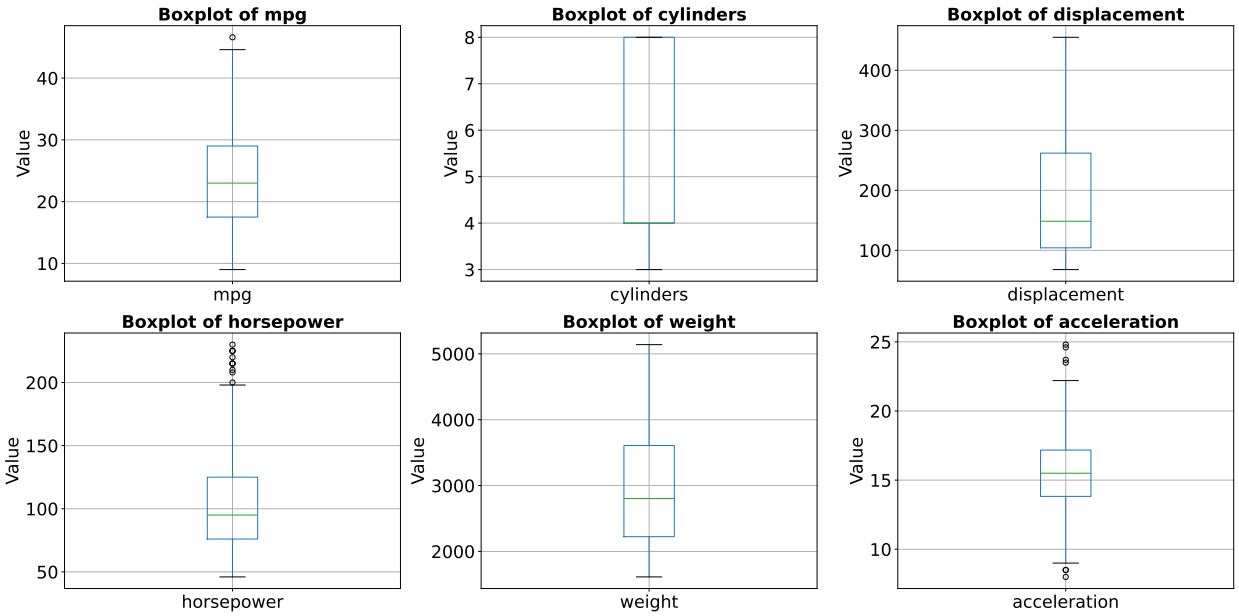
5. (**1 point**) À partir des résultats affichés par `df.describe()`, déterminez les valeurs du premier quartile (**Q1**), de la médiane, du troisième quartile (**Q3**) et l'écart-type pour la colonne **weight**.

[6]:
```
df.describe()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration
count	398.000	398.000	398.000	398.000	398.000	398.000
mean	23.515	5.455	193.426	104.469	2970.425	15.568
std	7.816	1.701	104.270	38.199	846.842	2.758
min	9.000	3.000	68.000	46.000	1613.000	8.000
25%	17.500	4.000	104.250	76.000	2223.750	13.825
50%	23.000	4.000	148.500	95.000	2803.500	15.500
75%	29.000	8.000	262.000	125.000	3608.000	17.175
max	46.600	8.000	455.000	230.000	5140.000	24.800

- **Q1** = ... **Q3** = ...
- **Médiane** = ... **écart-type** = ...

6. (**1 point**) Interpréter les boxplots ci-dessous des variables `mpg`, `cylinders`, `displacement`, `horsepower`, `weight` et `acceleration`.



.....

.....

.....

.....

.....

.....

ETUDIANT(E)	
Identifiant:	Classe: . . . . .
Noms & Prénom:	Salle: . . . . .

Partie II : Régression linéaire

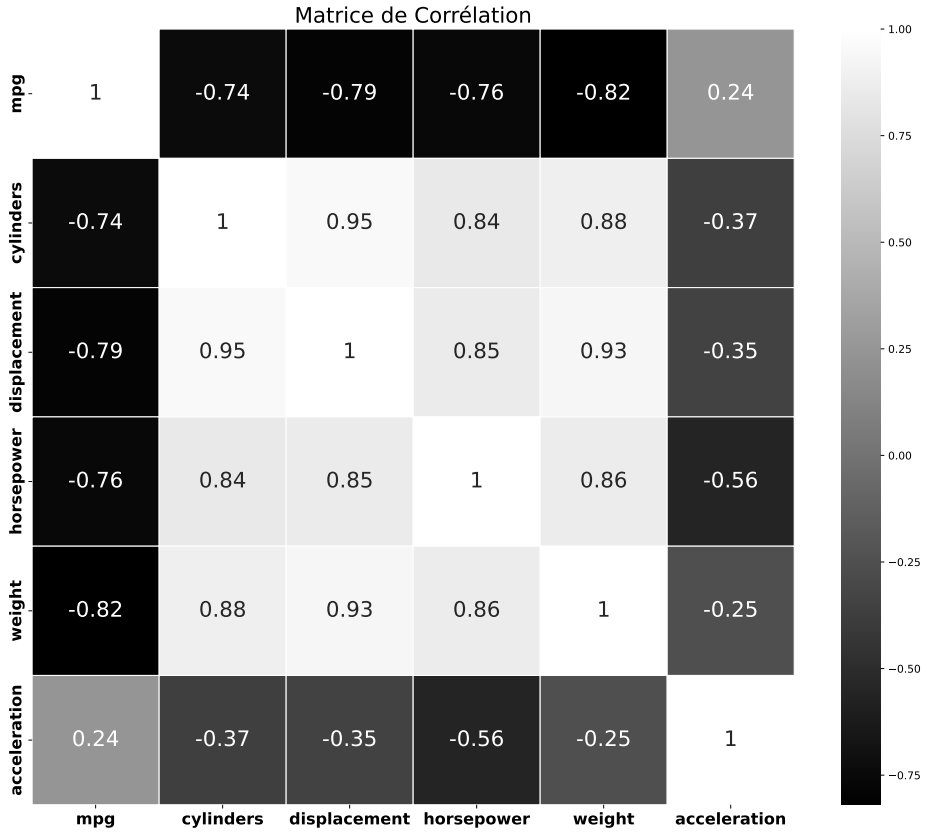
Les valeurs aberrantes ont été supprimées et les valeurs manquantes ont été remplacées par la moyenne. Le dataset nettoyé a été sauvegardé dans **'cars_cleaned.csv'**.

[7]:
```
# Charger le dataset sans outlayers
df = pd.read_csv('cars_cleaned.csv')
# Afficher la nouvelle dimension de dataset nettoyé
df.shape
```

[7]: (347, 6)

7. (**1 point**) À partir de la matrice de corrélation ci-dessous, interprétez les résultats concernant les relations entre les variables suivantes :

   - La **corrélation entre** `weight` **et** `displacement`.
   - La **corrélation entre** `weight` **et** `mpg`.



.....

.....

.....

.....

.....

.....

.....

Dans la suite, une régression linéaire sera effectuée afin de prédire la variable mpg (miles per gallon) en fonction de weight (poids).

[8]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Sélectionner les variables indépendantes (X) et dépendantes (y)
X = df[['weight']]
y = df['mpg']
```

8. (**0.5 point**) Remplir les champs vides ("_____") afin de diviser les données avec 20% pour le test et fixer l'aléatoire à 42.

[9]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=_______,
random_state=_________)
```

9. (**0.5 point**) Remplir le champ vide ("_____") pour créer un modèle de régression linéaire.

[10]:
```
# Créer le modèle de régression linéaire
model = ___________________
```

10. (**0.5 point**) Compléter le champ vide ("_____") par la méthode correcte pour entraîner un modèle de régression linéaire.

[11]:
```
# Entraîner le modèle de régression linéaire
model._____(X_train, y_train)
```

11. (**0.5 point**) Compléter le champ vide ("_____") par la méthode correcte pour effectuer la prédiction des observation de l'ensemble de test.

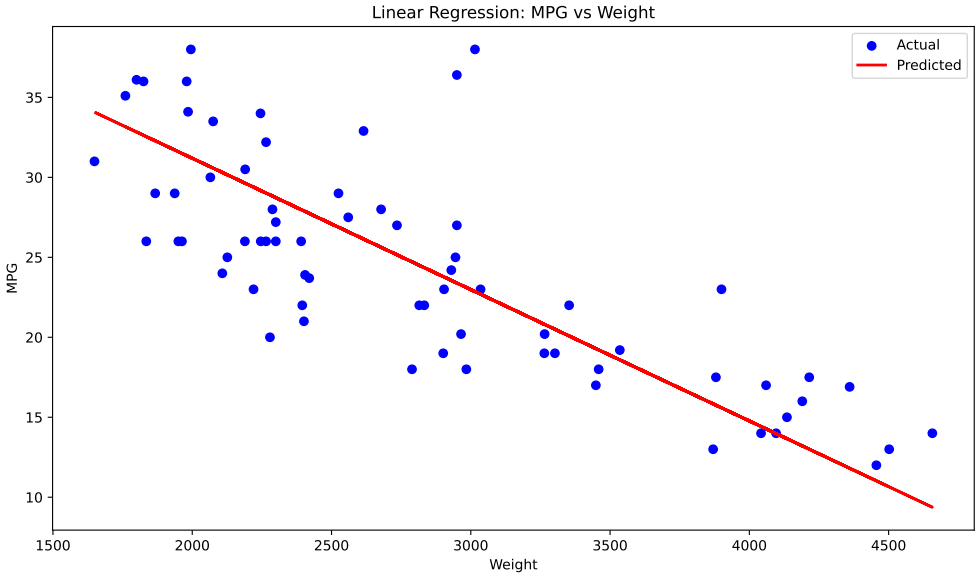
[12]:
```
# Prédire les valeurs sur l'ensemble de test
y_pred = model._____(X_test)
```

12. (**1 point**) Le code suivant est utilisé pour visualiser les résultats d'un modèle de régression linéaire. Complétez les champs manquants pour tracer la droite de régression, représentée dans la figure ci-dessous, basée sur les prédictions du modèle.

[13]:
```
# Visualiser les résultats
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='blue', label='Actual')

plt.plot(____________, _______________, color='red', label='Predicted')

plt.title('Linear Regression: MPG vs Weight')
plt.xlabel('Weight')
plt.ylabel('MPG')
plt.legend()
plt.show()
```



Vous obtiendrez les deux valeurs suivantes en exécutant le code suivant :

[14]:
```
print(model.intercept_)
print(model.coef_[0])
```

[14]:
```
47.6188791369011
-0.008212236375687649
```

13. (**1 point**) Donner l'équation de la droite de régression.

.....

Après avoir exécuté le code suivant, vous obtiendrez les valeurs de l'erreur quadratique moyenne (MSE) et du coefficient de détermination (R^2) :

[15]:
```
# Évaluer le modèle
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R^2 Score: {r2}")
```

[15]:
```
Mean Squared Error: 19.839240063680197
R^2 Score: 0.5638028451538022
```

Considérons les deux formules mathématiques suivantes, qui représentent des métriques utilisées pour évaluer la performance d'un modèle de régression :

(a) **Métrique 1** :

$$métrique_1 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

(b) **Métrique 2** :

$$métrique_2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

où :

- y_i sont les valeurs réelles observées,
- $\hat{y}_i$ sont les valeurs prédites par le modèle,
- $\bar{y}$ est la moyenne des valeurs réelles,
- n est le nombre total d'observations.

14. (**1 point**) À quelles métriques d'évaluation de la régression linéaire correspondent chacune des formules de $métrique_1$ et $métrique_2$?

.....

Exercice 2 : (6 points)

Dans cet exercice, nous allons travailler avec un dataset contenant des informations médicales sur les patients, notamment leur **Âge**, leur **Taux de cholestérol** (L'unité est en milligrammes par décilitre) et leur **fréquence cardiaque** (L'unité est battements par minute). L'objectif est de développer un modèle prédictif capable de déterminer l'état de santé des patients.
Le tableau ci-dessous présente les informations de cinq patients (nous travaillerons uniquement avec ces cinq patients) :

Patient	Âge	Taux de cholestérol (mg/dL)	Fréquence cardiaque (bpm)	Etat
A	45	200	160	Malade
B	50	220	NAN	Non-Malade
C	NAN	250	180	Malade
D	60	240	170	Non-Malade
E	55	NAN	150	Malade

ETUDIANT(E)	
Identifiant:	Classe: .....
Noms & Prénom:	Salle: .....

1. (a) (**0.5 point**) Remplacez les valeurs manquantes dans Âge et Taux de cholestérol par **les moyennes des colonnes** correspondantes en calculant les nouvelles valeurs .

   .....

   .....

   .....

   (b) (**0.5 point**) Remplacez la valeur manquante dans Fréquence cardiaque par **la médiane** de cette colonne en calculant la nouvelle valeur.

   .....

   .....

   .....

2. (**1 point**) Quel type d'apprentissage automatique convient à notre étude de cas ? Justifiez votre réponse.

   .....

   .....

   .....

3. (**0.5 point**) Précisez la variable cible (target) ainsi que les caractéristiques (features).

   .....

   .....

   .....

4. Nous avons entraîné deux algorithmes d'apprentissage automatique pour résoudre ce problème.

   - Algorithme 1 : un modèle basé sur les k plus proches voisins avec $k = 3$, en utilisant les autres hyperparamètres par défaut.
   - Algorithme 2 : un modèle d'arbre de décision avec les hyperparamètres par défaut. Ce modèle atteint un score de 100% sur l'échantillon d'apprentissage et de 70% sur l'échantillon de test.

(a) (**1 point**) Expliquez le fonctionnement de l'algorithme 1.

.....

.....

.....

.....

(b) (**0.5 point**) Interprétez les résultats du modèle de l'algorithme 2. Décidez si vous préférerez le conserver ou non en justifiant votre décision.

.....

.....

.....

.....

5. (**1 point**) Nous souhaitons classer un nouveau patient X, dont les attributs sont: **Âge = 52**, **Taux de cholestérol= 230** et **Fréquence cardiaque = 165**, en utilisant l'algorithme des k plus proches voisins. En utilisant la distance euclidienne, les distances entre le patient X et chaque patient du tableau A, B, C, D et E sont fournies dans le tableau ci-dessous.

Distance	d(A,X)	d(B,X)	d(C,X)	d(D,X)	d(E,X)
Valeurs	31.23	10.20	25	13.75	15.5

Utilisez l'algorithme des k plus proches voisins (avec k = 3) pour prédire si le patient X souffre d'une maladie cardiaque ou non.

.....

.....

.....

6. (**1 point**) Pour éviter le surapprentissage et le sous-apprentissage dans le modèle KNN, quelles sont les recommandations générales concernant le choix du paramètre k (nombre de voisins) ?

.....

.....

.....