Centre Universitaire de Mila Institut: ST, Département: MI Master 1 – STIC / I2A	**Corrigé type de l'interrogation en « Apprentissage Automatique »** Durée de l'examen : 1h00 / Mila le : 02-05-2023

Corrigé type de l'interrogation 2022-2023

Exercice 1 : QCM (5 points)

Cocher la (les) bonne (s) réponse (s) :

1) Quelle est la fonction d'erreur la plus couramment utilisée en régression linéaire?

a) La fonction de perte de Hinge

b) La fonction de perte logistique

c) La fonction d'erreur quadratique moyenne

d) La fonction de perte de Perceptron

Réponse: c) La fonction d'erreur quadratique moyenne. La fonction d'erreur quadratique moyenne est couramment utilisée en régression linéaire car elle mesure l'écart entre les prédictions du modèle et les valeurs réelles en utilisant une mesure de distance quadratique.

2) Qu'est-ce qu'un arbre de décision?

a) Un algorithme de classification linéaire

b) Un algorithme de régression linéaire

c) Un algorithme de classification non linéaire

d) Un algorithme de clustering

Réponse: c) Un algorithme de classification non linéaire. Un arbre de décision est un algorithme de classification non linéaire qui utilise une hiérarchie de décisions basées sur des tests de seuil pour prédire la classe d'un exemple.

3) Comment est déterminée la meilleure variable de division dans un arbre de décision?

a) En maximisant l'information gain

b) En minimisant l'information gain

c) En maximisant l'impureté de Gini

d) En minimisant l'impureté de Gini

Réponse: a) En maximisant l'information gain. La meilleure variable de division dans un arbre de décision est déterminée en maximisant l'information gain, qui mesure la réduction de l'entropie (ou de la variance) après la division.

4) Comment est déterminée la profondeur optimale d'un arbre de décision?

a) En maximisant l'information gain

b) En minimisant l'information gain

c) En maximisant l'impureté de Gini

d) En minimisant l'impureté de Gini

Réponse: b) En minimisant l'information gain. La profondeur optimale d'un arbre de décision est déterminée en minimisant l'information gain ou toute autre mesure de complexité, pour éviter le surapprentissage.

5) Qu'est-ce que la règle de Bayes?

a) Un théorème mathématique qui permet de calculer la probabilité conditionnelle

b) Une heuristique utilisée pour choisir la meilleure variable de division dans un arbre de décision

c) Une méthode d'optimisation utilisée pour minimiser la fonction de coût dans la régression linéaire

d) Une méthode d'optimisation utilisée pour minimiser l'impureté de Gini dans un arbre de décision

Réponse: a) Un théorème mathématique qui permet de calculer la probabilité conditionnelle. La règle de Bayes est un théorème mathématique qui permet de calculer la probabilité conditionnelle de la classe étant donné les caractéristiques d'un exemple, en utilisant la probabilité a priori et la vraisemblance.

6) Quel est l'avantage de la régression linéaire par rapport à d'autres algorithmes de régression?

a) Elle est plus rapide à entraîner

b) Elle peut modéliser des relations non linéaires

c) Elle est plus précise que les autres algorithmes

d) Elle est plus facile à comprendre et à interpréter

Réponse: d) Elle est plus facile à comprendre et à interpréter. La régression linéaire est souvent préférée à d'autres algorithmes de régression car elle est plus facile à comprendre et à interpréter, en plus d'être plus rapide à entraîner.

7) Qu'est-ce que la descente de gradient?

a) Une méthode pour minimiser la fonction de coût dans la régression linéaire

b) Une méthode pour maximiser l'information gain dans les arbres de décision

c) Une méthode pour calculer la probabilité conditionnelle dans le classificateur de Bayes

d) Une méthode pour maximiser la précision de la classification linéaire

Réponse: a) Une méthode pour minimiser la fonction de coût dans la régression linéaire. La descente de gradient est une méthode d'optimisation utilisée pour minimiser la fonction de coût dans la régression linéaire en ajustant les poids de manière itérative.

8) Qu'est-ce que l'entropie?

a) Une mesure de la pureté d'un nœud dans un arbre de décision

b) Une mesure de la distance entre les données dans l'espace des caractéristiques

c) Une mesure de l'erreur de prédiction pour un modèle de classification

d) Une mesure de la qualité d'un modèle de régression linéaire

Réponse: a) Une mesure de la pureté d'un nœud dans un arbre de décision. L'entropie est une mesure de la pureté d'un nœud dans un arbre de décision, utilisée pour déterminer quelle fonction de division est la meilleure.

9) Qu'est-ce que le classificateur de Bayes naïf?

a) Un algorithme de classification linéaire qui utilise une approche probabiliste

b) Un algorithme d'apprentissage non supervisé qui recherche des motifs dans les données

c) Un algorithme de classification qui utilise des règles conditionnelles simples et indépendantes

d) Un algorithme de régression linéaire qui utilise une approche bayésienne

Réponse: c) Un algorithme de classification qui utilise des règles conditionnelles simples et indépendantes. Le classificateur de Bayes naïf est un algorithme de classification qui suppose que toutes les caractéristiques sont indépendantes les unes des autres et utilise des règles conditionnelles simples pour prédire les classes.

10) Quelle est la différence entre l'overfitting et l'underfitting?

a) L'overfitting se produit lorsque le modèle est trop simple pour les données, tandis que l'underfitting se produit lorsque le modèle est trop complexe pour les données.

b) L'overfitting se produit lorsque le modèle s'adapte trop bien aux données d'entraînement, tandis que l'underfitting se produit lorsque le modèle ne s'adapte pas suffisamment bien aux données d'entraînement.

c) L'overfitting se produit lorsque le modèle a une variance élevée, tandis que l'underfitting se produit lorsque le modèle a une erreur de biais élevée.

d) L'overfitting et l'underfitting sont des termes interchangeables qui décrivent tous deux une mauvaise adaptation du modèle aux données.

Réponse: b) L'overfitting se produit lorsque le modèle s'adapte trop bien aux données d'entraînement, tandis que l'underfitting se produit lorsque le modèle ne s'adapte pas suffisamment bien aux données d'entraînement. L'overfitting et l'underfitting sont des problèmes courants en apprentissage automatique qui peuvent être résolus en utilisant des techniques telles que la régularisation pour réduire la complexité du modèle ou en ajustant les hyperparamètres pour trouver le bon équilibre entre biais et variance.

Exercice 2 : Classificateur de Bayes (5 points)

Nous souhaitons réaliser un classifieur bayesien permettant de classifier les emails en « Spam » ou « Ham (not spam) ». Pour ce faire, chaque mot w_i d'un e-mail, quel que soit l'endroit où il se trouve dans l'e-mail, est supposé avoir une probabilité $P(W= w_i|Y)$, où W prend des mots dans un dictionnaire prédéterminé (la ponctuation est ignorée). Y prend une valeur binaire (Spam ou ham).

I. Supposons que nous avons trois emails comme ensemble d'apprentissage.

(Spam) dear sir, if you could answer my questions I would be most grateful.
(Ham) see you at 12
(Ham) well, prepare it for tomorrow.

A partir de cet ensemble d'entrainement, calculer les probabilités bayésiennes suivantes.

- $P(W=sir \mid Y = spam)$
- $P(W=see \mid Y = ham)$
- $P(W=today \mid Y = ham)$
- $P(Y = ham)$

II. Le tableau suivant montre les probabilités estimées d'un ensemble de mots spams entrainés sur un large corpus d'emails.

W	*good*	*to*	*fine*	*luck*	*pay*
P(W|Y=spam)	*1/6*	*1/8*	*1/4*	*1/8*	*1/4*
P(W|Y=ham)	*1/8*	*1/3*	*1/4*	*1/12*	*1/12*

On vous donne un nouvel email à classer, avec seulement deux mots :

Good luck

1. Calculer la décision estimée pour cet email, sachant que :
   $P(Y = spam)=1/5$.
2. Quelle est l'intervalle de probabilités de $P(Y = spam)$ pour lequel le classifieur bayesien classe ce nouvel email comme spam ?

Solution

I. Calcul des probabilités

Intuitivement, la probabilité conditionnelle est *P(mot | c'est un mot dans un email de type Y)*. Nous estimons cette probabilité en comptant le nombre de mots :

- $P(W=sir \mid Y = spam)$

$$P(W = sir|Y = spam) = \frac{Card_{mot}(W = sir, Y = spam)}{Card_{mot}(Y = spam)} = \frac{1}{13} = 0{,}07$$

- *P(W=see | Y = ham)*

$$P(W = see|Y = ham) = \frac{Card_{mot}(W = see, Y = ham)}{Card_{mot}(Y = ham)} = \frac{1}{9} = 0.11$$

- *P(W=today | Y = ham)*

  Le mot "*today*" n'apparaît pas dans nos e-mails d'apprentissage, par conséquence, nous estimons sa probabilité conditionnelle à 0.

- *P(Y = ham)*

  Estimer la probabilité P(Y = ham) ne nécessite que de compter les emails :

$$P(Y = ham) = \frac{Card_{email}(Y = ham)}{Card_{email}(Total)} = \frac{2}{3} = 0.67$$

II. 1. Calcul de la décision estimée pour cet email

$$P(Y = spam|w_1 = Good, w_2 = luck)$$
$$= P(w_1 = Good \mid Y = spam)P(w_2 = luck \mid Y = spam)P(Y = spam)$$
$$= \frac{1}{6} \times \frac{1}{8} \times \frac{1}{5} = 0{,}0042$$

$$P(Y = ham \mid w_1 = Good, w_2 = luck)$$
$$= P(w_1 = Good \mid Y = ham)P(w_2 = luck \mid Y = ham)P(Y = ham)$$
$$= \frac{1}{8} \times \frac{1}{12} \times \frac{4}{5} = 0{,}0083$$

Donc, le modèle estime que l'email est normal (ham)

2. L'intervalle de probabilités de P(Y = spam) pour lequel le classifieur bayesien classe ce nouvel email comme spam

$$P(Y = spam|w_1 = Good, w_2 = luck) > P(Y = spam|w_1 = Good, w_2 = luck)$$
$$\Rightarrow P(w_1 = Good \mid Y = spam)P(w_2 = luck \mid Y = spam)P(Y = spam)$$
$$> P(w_1 = Good \mid Y = ham)P(w_2 = luck \mid Y = ham)P(Y = ham)$$
$$\Rightarrow \frac{1}{6} \times \frac{1}{8} \times P(Y = spam) > \frac{1}{8} \times \frac{1}{12} \times P(Y = ham)$$
$$\Rightarrow \frac{1}{48} \times P(Y = spam) > \frac{1}{96} \times (1 - P(Y = spam))$$
$$\Rightarrow \frac{1}{48} \times P(Y = spam) > \frac{1}{96} \times (1 - P(Y = spam))$$
$$\Rightarrow \frac{2}{96} \times P(Y = spam) > \frac{1}{96} - \frac{1}{96} \times P(Y = spam)$$

$$\Rightarrow \frac{3}{96} \times P(Y = spam) > \frac{1}{96}$$
$$\Rightarrow P(Y = spam) > \frac{1}{96} \times \frac{96}{3}$$
$$\Rightarrow P(Y = spam) > \frac{1}{3}$$
$$\Rightarrow P(Y = spam) \in \left]\frac{1}{3}, 1\right]$$

Exercice 3 Régression linéaire avec la bibliothèque scikit-learn:

Nous avons un ensemble de données comprenant deux variables, X et Y. Nous souhaitons créer un modèle de régression linéaire pour prédire la variable Y en fonction de la variable X.

Les données sont les suivantes :

X = [23, 45, 12, 67, 87, 43, 65, 34, 56, 78]

Y = [450, 678, 340, 980, 1200, 600, 900, 520, 800, 1100]

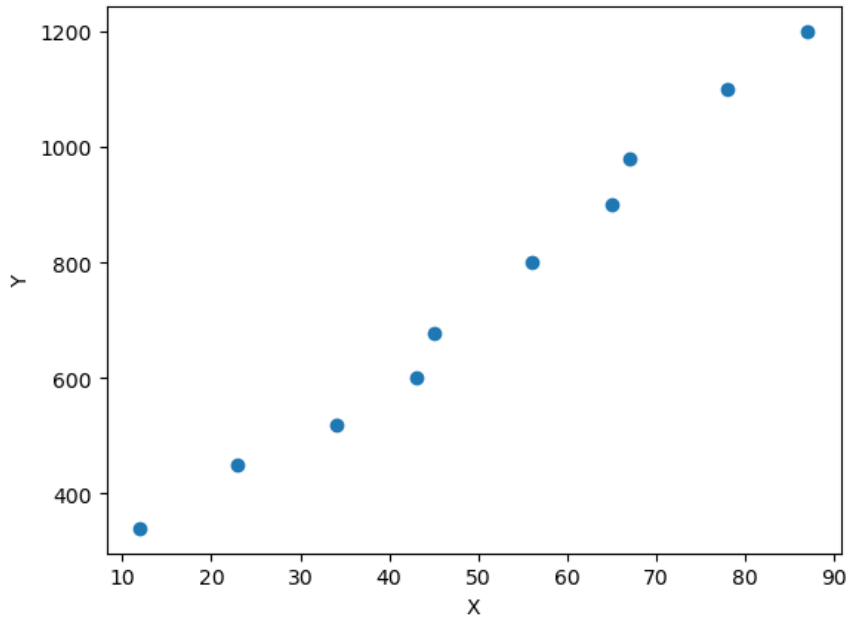
1. Tracer un graphique représentant les données.
2. Diviser les données en ensembles d'entraînement (80%) et de test (20%).
3. Créer un modèle de régression linéaire à l'aide de l'ensemble d'entraînement.
4. Prédire les valeurs de Y pour l'ensemble de test en utilisant le modèle de régression linéaire.
5. Évaluer les performances du modèle en utilisant la métrique de l'erreur quadratique moyenne (Mean Squared Error - MSE).
6. Tracer un graphique représentant les prédictions du modèle par rapport aux vraies valeurs pour l'ensemble de test.

Solution :

1. Tracer un graphique représentant les données.

```
import matplotlib.pyplot as plt

X = [23, 45, 12, 67, 87, 43, 65, 34, 56, 78]

Y = [450, 678, 340, 980, 1200, 600, 900, 520, 800, 1100]

plt.scatter(X, Y)

plt.xlabel('X')

plt.ylabel('Y')

plt.show()
```

Le graphique représentant les données est :



2. Diviser les données en ensembles d'entraînement (80%) et de test (20%).

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

3. Créer un modèle de régression linéaire à l'aide de l'ensemble d'entraînement.

```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit([[x] for x in X_train], Y_train)
```

4. Prédire les valeurs de Y pour l'ensemble de test en utilisant le modèle de régression linéaire.

```
Y_pred = reg.predict([[x] for x in X_test])
```

5. Évaluer les performances du modèle en utilisant la métrique de l'erreur quadratique moyenne (Mean Squared Error - MSE).

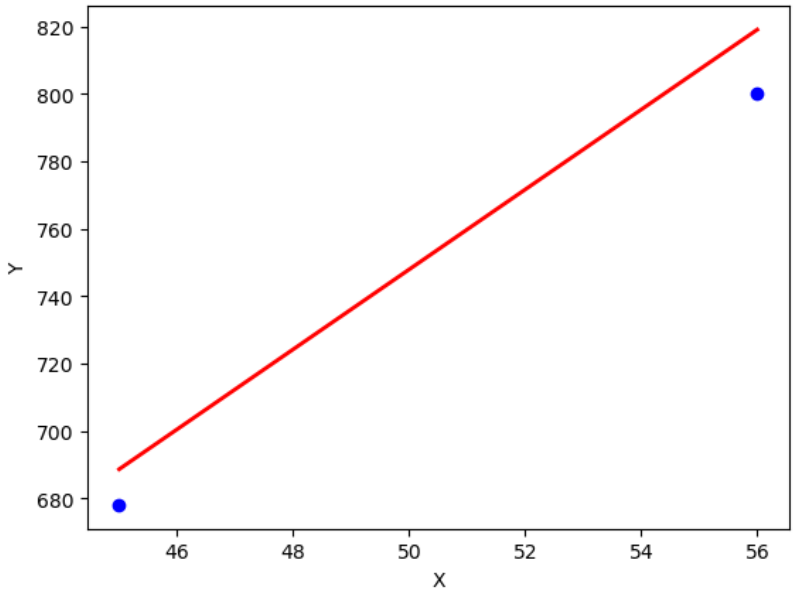
```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(Y_test, Y_pred)
print('MSE :', mse)
```

MSE : 238.14899651676146

6. Tracer un graphique représentant les prédictions du modèle par rapport aux vraies valeurs pour l'ensemble de test.

```
plt.scatter(X_test, Y_test, color='blue')
plt.plot(X_test, Y_pred, color='red', linewidth=2)
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

Le graphique résultant montre les valeurs prédites (en rouge) par rapport aux vraies valeurs (en bleu) pour l'ensemble de test :



On peut voir que le modèle de régression linéaire prédit raisonnablement bien les valeurs de Y en fonction de X, mais qu'il y a une certaine variabilité dans les prédictions.