Analyse de Données

Classification, Validation et Métriques

Antoine Lavault[1,2]

[1]Apeira Technologies

[2]UMR CNRS 9912 STMS, IRCAM, Sorbonne Université

19 janvier 2024

Contenu

Contenu

Rappels

- $\bar{y}$: dans le contexte du cours, la moyenne empirique (mean) sur les données y
- $\hat{y}$: un estimateur sur les données y
- Problème dual : Décrit plus tard dans le cours.

Contenu

Contenu

Définition

La classification est une technique supervisée qui permet de prédire la classe d'un objet à partir de caractéristiques ou de variables d'entrée choisie au préalable.
Cette technique est largement utilisée dans de nombreux domaines tels que la reconnaissance de texte, la reconnaissance d'image, la détection de fraudes, l'analyse de sentiment, etc.

Définition

La classification binaire est un type de classification où le modèle doit prédire une des deux classes possibles pour chaque objet. Par exemple, reconnaître un chat ou un chien ou encore reconnaître un spam d'un message acceptable.

Remarque

Pour entraîner un modèle de classification binaire, on aura besoin d'un ensemble de données étiquetées qui contient des exemples des deux classes possibles (contrairement au clustering, ou cette information n'est pas nécessaire).
Le processus d'entraînement consiste à ajuster les paramètres du modèle choisi. Une fois que le modèle est entraîné, il peut être utilisé pour prédire la classe de nouveaux objets. Pour ce faire, le modèle utilise les caractéristiques de l'objet pour calculer une probabilité que l'objet appartienne à chaque classe. La classe avec la plus haute probabilité est alors choisie comme la prédiction du modèle.

Définition

La classification multi-classe est un type de classification où le modèle doit prédire une des plusieurs classes possibles pour chaque objet. Par exemple, le modèle peut être entraîné à prédire le type de fleur en fonction de leurs caractéristiques.

Remarque

Pour entraîner un modèle de classification multi-classe, vous avez besoin d'un ensemble de données étiquetées qui contient des exemples de toutes les classes possibles. Le processus d'entraînement est similaire à celui de la classification binaire, mais le modèle doit maintenant prédire la probabilité d'appartenance à chaque classe.
Une approche courante pour résoudre le problème de la classification multi-classe est la classification one-vs-all (ou one-vs-rest). Dans cette approche, un modèle est entraîné pour chaque classe, en considérant cette classe comme positive et toutes les autres comme négatives. Lors de la prédiction, le modèle qui prédit la plus haute probabilité est choisi comme prédiction finale.

Contenu

D'où ça sort ?

On cherche donc un hyperplan séparateur qui maximise la marge. La marge est la distance entre l'hyperplan et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports.
L'hyperplan qui maximise la marge est donné par

$$\arg\max_{w,b} \min_{i} \{||x - x_i|| : x \in \mathbb{R}^N, w^T x + b = 0\}$$

Il s'agit donc de trouver w et b remplissant ces conditions, afin de déterminer l'équation de l'hyperplan séparateur :

$$h(x) = w^T x + b = 0$$

La marge est la plus petite distance entre les échantillons d'apprentissage et l'hyperplan séparateur qui satisfasse la condition de séparabilité (à savoir $y_i(w^T x_i + w_0) \geq 0$).
La distance d'un échantillon x_i à l'hyperplan est donnée par sa projection orthogonale sur le vecteur de poids :

$$\frac{y_i(w^T x_i + w_0)}{||w||}$$

L'hyperplan séparateur (w,w0) de marge maximale est donc donné par :

$$\arg\max_{w,w_0} \left\{ \frac{1}{||w||} \min_i \left[y_i(w^T x_i + w_0) \right] \right\}$$

Afin de faciliter l'optimisation, on choisit de normaliser w et w0, de telle manière que les échantillons à la marge (x^+_{marge} pour les vecteurs supports sur la frontière positive, et x^-_{marge}pour ceux situés sur la frontière opposée) satisfassent :

$$\begin{cases} w^T x^+_{marge} + w_0 = 1 \\ w^T x^-_{marge} + w_0 = -1 \end{cases}$$

D'où pour tous les échantillons, $k = 1, \ldots, p$

$$y_i(w^T x_i + w_0) \geq 1$$

Cette normalisation est parfois appelée la forme canonique de l'hyperplan, ou hyperplan canonique.

On suppose que la séparation est parfaite dans ce cas. Le cas "imparfait" suit un cheminement similaire, avec quelques conditions supplémentaires.
L'entraînement du SVM pour trouver w et b consiste à résoudre le problème d'optimisation suivant :

$$\max_{w,b} marge \Leftrightarrow \min_{w,b} \frac{1}{2} \|w\|^2 \tag{1}$$

avec $y_i(w^T \cdot x_i + b) \geq 1$ pour tout i et où x_i est le i-ème vecteur d'entrée, y_i est l'étiquette de classe correspondante (1 ou -1 pour une classification binaire) et les w sont considérés normalisés.
A noter, les contraintes garantissent que tous les exemples sont classés correctement avec une marge minimale de 1.
Une fois que w et b ont été trouvés, la frontière de décision est déterminée par :

$$\mathcal{H} = \{x \in X | w^T \cdot x + b = 0\}$$

La classe d'un nouvel exemple est déterminée en évaluant la fonction de décision $f(x)$ et en prenant le signe de sa valeur : $f(x) \geq 0$ pour la classe positive, $f(x) < 0$ pour la classe négative.

La forme primale du problème d'optimisation est la suivante :

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

sous la contrainte $y_i(w^T x_i + b) \geq 1$ pour tout $i = 1, \ldots, n$, où w est le vecteur de poids, b est le biais, $\|.\|^2$ est la norme Euclidienne..
En utilisant la théorie de la dualité, on peut transformer le problème en forme duale en introduisant des coefficients de Lagrange $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_n$ tels que :

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

sous la contrainte $\sum_{i=1}^{n} \alpha_i y_i = 0$, $0 \leq \alpha_i$ pour tout $i = 1, \ldots, n$

- Écrire la fonction objectif de la forme primale (dans le cas imparfait), qui est de la forme :

$$\min_{w,b} \frac{1}{2}|w|^2 + C\sum_{i=1}^{n} \xi_i$$

où w est le vecteur de poids, b est le biais, ξ_i est la marge d'erreur pour la i-ème instance, et C est un hyperparamètre qui contrôle le compromis entre la marge et la violation des contraintes dans le cas de données non-linéairement séparables.

- Ajouter les contraintes pour la forme primale :

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \text{ et } \xi_i \geq 0$$

pour $i = 1, \ldots, n$.

- Former la fonction Lagrangienne en introduisant les multiplicateurs de Lagrange α_i pour chaque contrainte :

$$\mathcal{L}(\mathrm{w}, b, \xi, \alpha, \mu) = \frac{1}{2}|\mathrm{w}|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(\mathrm{w}^T\mathrm{x}_i + b) - (1-\xi_i)] - \sum_{i=1}^{n}\mu_i\xi_i$$

où α et μ sont les vecteurs de multiplicateurs de Lagrange pour les contraintes d'inégalité et de positivité, respectivement.

- Trouver les dérivées partielles de la fonction Lagrangienne par rapport à w, b et ξ, et les égaler à zéro :

$$\frac{\partial \mathcal{L}}{\partial \mathrm{w}} = \mathrm{w} - \sum_{i=1}^{n} \alpha_i y_i \mathrm{x}_i = 0 \quad \Rightarrow \quad \mathrm{w} = \sum_{i=1}^{n} \alpha_i y_i \mathrm{x}_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad \Rightarrow \quad \mu_i = C - \alpha_i$$

- Substituer les résultats obtenus pour w, b, et ξ en fonction de α et μ, dans la fonction Lagrangienne, pour obtenir la forme duale du SVM :

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathrm{x}_i, \mathrm{x}_j \rangle$$

$$\text{s.c.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C \quad \text{pour} \quad i = 1, \ldots, n$$

où $\langle \mathrm{x}_i, \mathrm{x}_j \rangle$ est le produit scalaire, que l'on peut aussi écrire $\mathrm{x}_i^T \mathrm{x}_j$

Demo Time

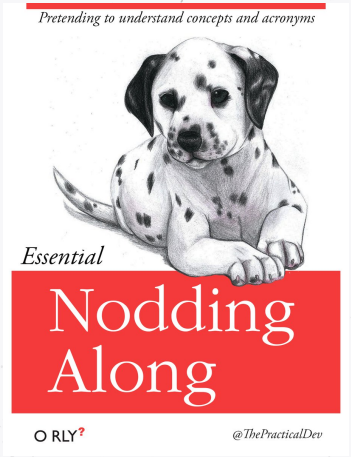

Figure – J'espère que ce n'est pas le cas...

Contenu

Définition

Le kernel trick est une technique utilisée pour étendre des méthodes de classification linéaire à des problèmes de classification non linéaires.
En utilisant le kernel trick, il est possible de transformer les données d'entrée dans un espace de dimension supérieure (voire infinie) où elles sont linéairement séparables, sans avoir à calculer explicitement cette transformation.

Application aux SVM - Cas parfait

Le kernel trick est basé sur l'utilisation de la représentation duale du problème d'optimisation du SVM, qui est un problème d'optimisation quadratique sous contraintes. La fonction lagrangienne associée au problème d'optimisation SVM linéaire parfait est :

$$L(w, b, \alpha) = \frac{1}{2}|w|^2 - \sum_{i=1}^{n} \alpha_i y_i (w^T x_i + b) + \sum_{i=1}^{n} \alpha_i$$

où α_i est le coefficient de Lagrange associé à l'échantillon x_i.
Ensuite, en utilisant les conditions de Karush-Kuhn-Tucker (KKT, existence du minimum sous conditions), on peut montrer que les vecteurs de support, c'est-à-dire les échantillons x_i pour lesquels $\alpha_i > 0$, sont ceux qui se trouvent sur la marge de séparation. Mais ça on s'en fout.

En utilisant une fonction noyau, nous pouvons calculer le produit scalaire entre deux vecteurs de données x_i et x_j dans un espace de caractéristiques transformé sans avoir à calculer explicitement la transformation elle-même (c'est bien). La fonction noyau est définie comme suit :

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

où ϕ est la transformation de l'espace des caractéristiques. Le noyau le plus couramment utilisé est le noyau gaussien, également connu sous le nom de noyau RBF (Radial Basis Function) :

$$K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2)$$

où γ est un paramètre qui contrôle la forme de la fonction noyau. La fonction noyau permet de calculer le produit scalaire dans l'espace de caractéristiques transformé, sans avoir à effectuer explicitement la transformation.

Remarque

C'est le théorème de Mercer qui permet d'obtenir ce résultat.

En utilisant la fonction noyau, nous pouvons remplacer le produit scalaire entre les vecteurs de données dans le lagrangien par le produit scalaire dans l'espace transformé. Cela nous permet de résoudre des problèmes de classification non linéaires en utilisant un SVM linéaire dans l'espace de caractéristiques transformé.
Le problème d'optimisation de la forme duale avec un noyau devient :

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

sous la contrainte de à $\sum_{i=1}^{n} \alpha_i y_i = 0$ et $\alpha_i \geq 0$ pour tout i.

Conséquence pour les SVM

En résolvant ce problème d'optimisation, nous pouvons trouver les valeurs optimales de α, qui sont ensuite utilisées pour calculer les poids du modèle de SVM. Les échantillons pour lesquels les coefficients de Lagrange sont non nuls sont les vecteurs de support qui définissent la frontière de décision du SVM. Cette frontière de décision est donnée par :

$$f(x) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b)$$

où b est le biais et est déterminé à partir des échantillons de support.

Remarque

Le kernel trick permet de résoudre des problèmes de classification non linéaires en transformant implicitement l'espace de caractéristiques des données d'entrée vers un espace de dimension supérieure, sans avoir à effectuer explicitement cette transformation. Cela permet d'économiser du temps et de la mémoire, car la transformation peut être très coûteuse en termes de calcul. De plus, le kernel trick permet de travailler avec des espaces de caractéristiques de grande dimension, qui peuvent être difficiles à manipuler et à visualiser.

Demo Time

Figure – Probabilité de 78.35% que ce soit vrai

Contenu

Figure – Niveau 1 : Cacher la poussière sous le tapis.

Capacité de généralisation et séparation des données

La séparation des ensembles de données en ensembles d'entraînement, de test et de validation est une pratique courante et nécessaire dans l'apprentissage automatique et l'analyse de données. Cela permet d'évaluer les performances d'un modèle d'apprentissage automatique et de s'assurer qu'il généralise bien à de nouvelles donnée.
Le jeu de données d'entraînement est utilisé pour ajuster les paramètres du modèle. Le modèle est ensuite évalué sur un ensemble de données de validation distinct pour ajuster les hyperparamètres et éviter l'overfitting. L'ensemble de test est utilisé pour évaluer les performances finales du modèle et pour estimer comment il se comportera sur de nouvelles données.
La séparation des ensembles de données en ensembles d'entraînement, de test et de validation permet de mesurer la performance d'un modèle sur des données qu'il n'a pas vu auparavant, ce qui est crucial pour évaluer la capacité de généralisation du modèle. En utilisant une séparation appropriée, il est possible de s'assurer que les résultats obtenus sont fiables et reflètent la capacité du modèle à généraliser à de nouvelles données.

Exemple

On peut faire une analogie avec les exemples du cours (entraînement), les TD (entraînement, validation) et l'examen (test).

Validation Croisée

Définition

La validation croisée est une technique d'évaluation de la performance des modèles de machine learning qui consiste à diviser l'ensemble de données en plusieurs sous-ensembles, à entraîner le modèle sur certains de ces sous-ensembles et à évaluer sa performance sur les autres. La validation croisée permet d'obtenir une estimation plus fiable de la performance du modèle sur des données inconnues que l'utilisation d'un seul ensemble de données de test.

Remarque

L'ensemble de test ne doit JAMAIS être utilisé dans le pipeline d'entraînement.

K-Fold

Définition

La méthode K-fold est une technique de validation croisée courante utilisée pour évaluer la performance des modèles d'apprentissage automatique.
Cette méthode consiste à diviser le jeu de données en k sous-ensembles ou "plis" (folds) de taille égale, ou à peu près égale.
Le modèle est alors entraîné k fois, chacune avec un pli différent utilisé comme ensemble de test et les autres plis comme ensemble d'entraînement. Les résultats sont ensuite moyennés pour obtenir une estimation de la performance moyenne du modèle.

Remarque

La méthode K-fold permet de maximiser l'utilisation des données d'entraînement tout en permettant une évaluation fiable de la performance du modèle. Elle est également utile pour évaluer la variabilité des performances du modèle en fonction des différentes partitions des données.
Il est important de noter que la méthode K-fold ne doit pas être utilisée pour ajuster les hyperparamètres du modèle car cela peut conduire à un sur-apprentissage. Pour l'ajustement des hyperparamètres, il est recommandé d'utiliser une technique de validation croisée sur l'ensemble d'entraînement ou de validation uniquement.

Shuffle & Split

Définition

La méthode "shuffle and split" est une technique courante pour diviser un ensemble de données en deux parties : une partie pour l'entraînement du modèle et une autre partie pour le test du modèle. Dans cette méthode, l'ensemble de données est d'abord mélangé aléatoirement (shuffled) pour éviter tout biais dans la répartition des données. Ensuite, les données sont divisées en deux parties en utilisant une proportion prédéfinie (par exemple 80% pour l'entraînement et 20% pour la validation).

Remarque

Cette méthode est simple et rapide à mettre en œuvre et convient particulièrement aux ensembles de données de petite à moyenne taille. Elle est également utile lorsque l'on a besoin d'une seule partition pour l'entraînement et la validation, sans avoir besoin d'itérations multiples comme dans la méthode K-fold.
Cependant, il est important de noter que la méthode "shuffle and split" peut entraîner une forte dépendance à la répartition aléatoire des données, en particulier pour les ensembles de données de petite taille. Il est donc recommandé de répéter la division aléatoire plusieurs fois et de prendre la moyenne des résultats pour réduire l'impact de cette variabilité.

Grid Search

Définition (Hyperparamètres)

Les hyperparamètres sont des paramètres qui ne sont pas appris à partir des données d'entraînement, mais qui doivent être définis avant l'entraînement du modèle. Par exemple, le nombre de couches dans un réseau de neurones, le taux d'apprentissage, le nombre d'arbres dans une forêt d'arbres de décision, etc.

Définition (GridSearch)

La méthode GridSearch consiste à créer une grille de toutes les combinaisons possibles d'hyperparamètres à tester. Pour chaque combinaison d'hyperparamètres, le modèle est entraîné sur les données d'entraînement et évalué sur les données de validation. Le meilleur ensemble d'hyperparamètres est alors sélectionné en fonction de la performance sur les données de validation.

Remarque

Le GridSearch ne garantit pas que les hyperparamètres sélectionnés sont les meilleurs pour de nouvelles données, il est donc important de tester le modèle final sur un ensemble de test distinct pour évaluer sa performance.

Contenu

Contenu

Évaluation d'un modèle de classification

Une fois que le modèle est entraîné, il est important de mesurer sa performance sur un ensemble de données de test pour déterminer sa capacité à généraliser à de nouveaux exemples. Les mesures de performance courantes pour la classification comprennent :

- Précision : la proportion de prédictions positives qui sont correctes. Il est possible d'étendre cette définition aux classes négatives.
- Rappel (Recall) : la proportion d'exemples positifs qui sont correctement prédits
- F-score : la moyenne harmonique de la précision et du rappel

Ce qui nous donne comme équations :

- Précision : $\frac{vp}{vp+fp}$
- Rappel : $\frac{vp}{vp+fn}$
- F-score : $F_1 = \frac{2}{\mathrm{rappel}^{-1}+\mathrm{precision}^{-1}} = 2\frac{\mathrm{precision}\cdot\mathrm{rappel}}{\mathrm{precision}+\mathrm{rappel}} = \frac{2\mathrm{vp}}{2\mathrm{vp}+\mathrm{fp}+\mathrm{fn}}$.

Avec *vp* le nombre de vrais positifs, *fp* les faux positifs (items négatifs classifiés commes positifs) et *fn* les faux négatifs (les positifs classés comme négatifs).

Matrice de confusion

La matrice de confusion est un outil important pour évaluer les performances d'un modèle de classification. Elle permet de comparer les prédictions du modèle avec les vraies étiquettes des données et de calculer des métriques de performance telles que la précision, le rappel, le score F1, etc.
La matrice de confusion est une matrice carrée qui contient quatre éléments : les vrais positifs (VP), les faux positifs (FP), les vrais négatifs (VN) et les faux négatifs (FN). Chaque ligne de la matrice représente les vraies étiquettes des données et chaque colonne représente les prédictions du modèle.
Voici un exemple de matrice de confusion pour un modèle de classification binaire :

	Prédiction positive	Prédiction négative
Vraie étiquette positive	VP	FN
Vraie étiquette négative	FP	VN

Table – Exemple de matrice de confusion

Avec la matrice de confusion, nous pouvons calculer différentes métriques de performance pour le modèle de classification. Voici quelques-unes des métriques les plus couramment utilisées :

- La précision (accuracy) que l'on définira ici sur les deux classes : $\frac{VP+VN}{VP+FP+VN+FN}$
- Le rappel (recall) est défini comme la proportion d'exemples positifs correctement classés par rapport au nombre total d'exemples positifs : $\frac{VP}{VP+FN}$.
- Le score F1 est une mesure pondérée de la précision et du rappel, défini précédemment

Remarque

En général, on cherchera à maximiser la précision et le rappel simultanément. Cependant, ces métriques peuvent être en concurrence, ce qui signifie que l'amélioration de l'une peut entraîner une diminution de l'autre. Le score F1 est souvent utilisé pour trouver un compromis entre la précision et le rappel. La connaissance de la matrice de confusion peut également aider à identifier les erreurs de classification commises par le modèle et à améliorer sa performance.

Définition

La courbe ROC (Receiver Operating Characteristic) est une courbe de performance d'un modèle de classification binaire qui représente le taux de vrais positifs (sensibilité) en fonction du taux de faux positifs (1 - spécificité) pour différents seuils de classification.
En d'autres termes, la courbe ROC représente la capacité d'un modèle à distinguer les classes positives et négatives en fonction du seuil de décision utilisé pour séparer les deux classes.

Pour construire la courbe ROC, on calcule le taux de vrais positifs et le taux de faux positifs pour différents seuils de classification, et on représente ces mesures sur un graphique avec le taux de vrais positifs sur l'axe des ordonnées et le taux de faux positifs sur l'axe des abscisses. Plus la courbe ROC est proche du coin supérieur gauche du graphique, plus la performance du modèle est considérée comme élevée.

Cont.

En plus de la courbe ROC elle-même, une mesure de performance couramment utilisée est l'aire sous la courbe ROC (AUC-ROC), qui mesure la performance globale du modèle sur toute la plage de seuils de classification possibles. L'AUC-ROC est une mesure de performance entre 0 et 1, où une valeur de 1 indique une performance parfaite, une valeur de 0,5 indique une performance aléatoire et une valeur de 0 indique une performance complètement erronée.
La courbe ROC est souvent utilisée en combinaison avec d'autres mesures de performance, telles que la précision, le rappel et la F1-score, pour évaluer les performances d'un modèle de classification binaire.
Le Detection Error Tradeoff est une variante du ROC qui permet d'évaluer le modèle sous l'angle du compromis faux-négatifs/faux positifs.

Figure – Et c'est pour ça que l'on utilise des métriques

Contenu

- Erreur quadratique moyenne (MSE) : Cette métrique calcule la moyenne des carrés des différences entre les prédictions du modèle et les vraies valeurs de la variable cible. Elle est définie comme suit :

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

où y_i est la vraie valeur de la variable cible pour l'observation i, $\hat{y}_i$ est la prédiction du modèle pour l'observation i, et n est le nombre total d'observations.

- Erreur absolue moyenne (MAE) : Cette métrique calcule la moyenne des valeurs absolues des différences entre les prédictions du modèle et les vraies valeurs de la variable cible. Elle est définie comme suit :

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Coefficient de détermination (R^2) : Cette métrique mesure la proportion de la variance totale de la variable cible qui est expliquée par le modèle. Il varie entre 0 et 1, où 1 indique une correspondance parfaite entre les prédictions du modèle et les vraies valeurs de la variable cible. Il est défini comme suit :

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

où $\bar{y}$ est la moyenne des vraies valeurs de la variable cible.

- Erreur de pourcentage absolue moyenne (MAPE) : Cette métrique calcule la moyenne des pourcentages absolus des différences entre les prédictions du modèle et les vraies valeurs de la variable cible. Elle est définie comme suit :

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

Remarque

Il est important de choisir la métrique appropriée en fonction du contexte et de la nature des données. Par exemple, le MSE est sensible aux valeurs aberrantes, tandis que le MAE est plus robuste à celles-ci. De même, le R^2 est utile pour comparer la performance de différents modèles, mais ne donne pas d'indication sur la magnitude des erreurs de prédiction.

- Indice de Rand (Rand Index - RI) : Cette métrique mesure la similarité entre les groupes obtenus par le modèle de clustering et les groupes réels (si disponibles). De manière plus générale, c'est une mesure entre deux partitions.

$$RI = \frac{a+b}{a+b+c+d}$$

Où

- a est le nombre de paires de données qui sont dans le même groupe par les deux méthodes (vrais positifs),
- b est le nombre de paires de données qui sont dans des groupes différents par les deux méthodes (vrais négatifs),
- c est le nombre de paires de données qui sont dans le même groupe par la première méthode mais pas par la seconde (faux positifs),
- d est le nombre de paires de données qui sont dans des groupes différents par la première méthode mais dans le même groupe par la seconde (faux négatifs).

- Coefficient de silhouette (Silhouette Coefficient) : Cette métrique mesure à quel point chaque observation est similaire aux autres observations dans son propre cluster par rapport aux observations des autres clusters. Il varie entre -1 et 1, où une valeur plus proche de 1 indique que l'observation est bien classée dans son propre cluster et mal classée dans les autres clusters. Il est défini comme suit :

$$s(i) = \frac{b(i) - a(i)}{\max a(i), b(i)}$$

où $a(i)$ est la distance moyenne entre l'observation i et toutes les autres observations dans le même cluster, et $b(i)$ est la distance moyenne entre l'observation i et toutes les observations dans le cluster le plus proche (autre que son propre cluster).

Remarque

Il est important de choisir la métrique appropriée en fonction du type de données et du contexte. Par exemple, le coefficient de silhouette est utile pour évaluer la qualité de la segmentation, tandis que l'indice Rand est plus adapté pour évaluer la similarité entre les groupes obtenus par le modèle de clustering et les groupes réels.

Contenu

Ce que vous devriez être

Figure – Un équivalent de Von Neumann

Figure – Les lunettes ne donnent pas de points d'intelligence.

Rappel

DM/Examen pour le 9/04 à 23h59, sur Teams. Pas de remise tardive autorisée.

Figure – Maintenant miam miam.