Questions Boostrap

lundi 20 mars 2023 19:24

Why use a separate dataset to measure the performance of an algorithm? What are your results when you test your algorithm on the same dataset used in training?

Lorsqu'on évalue la performance d'un algorithme, il est important d'utiliser des données séparées pour tester. En effet, utilisez le même dataset pour l'entraînement et le testing peut amener à du overfitting, on peut penser que l'algorithme performe bien sur les données d'entraînement mais agit pauvrement nullement sur les nouvelles données. En utilisant un dataset différent pour le test, on a une mesure plus précise de la performance de l'algorithme sur de nouvelles données. Cela est très utile lorsqu'on souhaite éviter overfitting(surévaluation) et être sûr que l'algorithme est capable de bien généraliser pour toutes les situations.

Les résultats peuvent être trompeurs. L'algorithme a déjà vu les données pendant l'entraînement. Il peut donc être biaisé vers ces données et fonctionner bien mieux qu'il ne le ferait sur de nouvelles données. Cela peut donner un faux sentiment de confiance dans les performances de l'algorithme.

En résumé, l'utilisation d'un ensemble de données différent pour les tests est essentiel pour évaluer avec précision les performances d'un algorithme, tester le même ensemble de données utilisées dans la dans l'entraînement peut conduire à des résultats OVERFITTES et MISLEADES

What are bias and variance?
What do they measure?
Which values should they take?

biais d'un modèle ==> sa capacité à **capturer la tendance générale des données**.
Si un modèle a un biais **élevé**, cela signifie qu'il est trop **simple** et qu'il ne peut pas **capturer les nuances des données**.
Un modèle avec un biais **faible** peut être plus **complexe** et mieux **adapté aux données**, mais il peut **souffrir d'un autre problème appelé** **variance**.

Variance d'un modèle ==> sa capacité à **généraliser à de nouvelles données**.
Si un modèle a une variance **élevée**, cela signifie qu'il est trop **complexe** et qu'il a **mémorisé les données d'entraînement** au lieu de **capturer les tendances générales**. Ainsi, lorsque le modèle est évalué sur des données de test, il ne performe pas aussi bien qu'il le devrait.

Bias ET **variance** sont 2 concepts importants en machine learning, qui permet de mesurer les la différence entre les values, les valeurs prédites et les valeurs réelles dans un dataset.

Bias fait référence à l'erreur introduite par l'approximation d'un problème du monde réel avec un modèle simplifié. Un biais élevé signifie que le modèle est trop simple et ne parvient pas à saisir la complexité du problème, ce qui entraîne un sous-ajustement.

Variance fait référence à l'erreur introduite par l'approximation d'un problème du monde réel avec un modèle complexe. Une variance élevée signifie que le modèle est trop complexe et capture le bruit dans les données, ce qui entraîne un overfitting (surajustement).

Idéalement, nous voulons obtenir **un faible biais et une faible variance**. Cependant, dans la pratique, il existe souvent un compromis entre biais et variance, et trouver le bon équilibre est important pour obtenir de bonnes performances sur de nouvelles données inédites.

Ces problèmes surviennent lorsque les données sont limitées
POUR RESOUDRE on utilise
validation croisée (cross-validation) ou la **sélection aléatoire de l'ensemble de test**
techniques utilisées pour évaluer la performance d'un modèle et s'assurer qu'il ne souffre pas de biais ou de variance.

What is cross validation?
What are the main advantages?
When can I use it?

https://datascientest.com/cross-validation

validation croisée (cross-validation)

Fonctionnement : diviser l'ensemble de données en plusieurs sous-ensembles, puis à entraîner et à évaluer le modèle sur chacun de ces sous-ensembles en utilisant différentes combinaisons d'entraînement et de test (nolds)
Utilisation :
obtenir une estimation plus fiable de la performance du modèle

Cette technique permet d'évaluer les performances d'un algorithme. Le principal avantage de la validation de création croisée est qu'elle permet une estimation plus précise de la perf. sur de nouvelles données inédites. Elle implique de diviser les données en plusieurs sous-ensembles et d'utiliser chaque sous-ensemble à la fois pour l'entrainement et les tests. Elle est ensuite testée sur des données qu'il n'a jamais vues auparavant. Ce qui permet d'éviter le overfitting. Et garantir que l'Algo est capable de bien se généraliser à de nouvelles situations.

Can you explain why it's important to normalize (scale) the data when using algos like KNN? Is it necessary in our specific study case?

Il est important de normaliser (mettre à l'échelle) les données lors de l'utilisation d'algorithmes tels que K-Nearest Neighbors (KNN), car l'algorithme est basé sur la distance entre les points de données.

La normalisation des données garantit que chaque entité contribue de manière égale au calcul de la distance. Cela peut aider à améliorer les performances de l'algorithme et à rendre les résultats plus significatifs.

Dans le cas de l'ensemble de données MNIST, chaque image est représentée sous la forme d'un tableau de 28 x 28 pixels, et chaque valeur de pixel est comprise entre 0 et 255, ce qui représente la luminosité du pixel. Avant d'utiliser ces données pour former un algo de ML, il est courant de normaliser les valeurs de pixel dans une plage comprise entre 0 et 1, en divisant chaque valeur de pixel par 255.

La normalisation est généralement considérée comme une bonne pratique et peut améliorer les performances du modèle et sa stabilité.

When you reshaped your image(like in mnist dataset), do you think the order of the columns (that means the order of the pixels) had an importance for the performance of your algorithm?

Quand on reshape une image, l'ordre des colonnes (pixels) n'a pas d'impact sur les performances de l'algorithme. En effet,le preprocessing garantit que chaque pixel se voit attribuer une position unique dans le tableau remodelé, quelle que soit sa position d'origine dans l'image.

Par exemple, dans le cas de MNIST, chaque image est représentée à l'origine sous la forme d'un tableau de 28 x 28 pixels. Lorsqu'ils sont remodelés en un tableau 1D de longueur 784, les pixels se voient attribuer une position unique dans le tableau en fonction de leur ligne et de leur colonne dans le tableau 2D d'origine.

Par conséquent, l'ordre des colonnes (pixels) n'a pas d'importance, puisque chaque pixel se voit attribuer une position unique dans le tableau remodelé.

Par conséquent, les performances de l'algorithme ne doivent pas être affectées par l'ordre des colonnes (pixels) dans le tableau remodelé.

Which metrics measure performance (in mnist dataset for exemple) ?
What does accuracy tell you?
Does accuracy penalize more one mistake over another?

Les mesures courantes utilisées pour mesurer les performances incluent l'exactitude, la précision, le rappel et le score F1.

La précision est une métrique qui mesure le pourcentage d'images correctement classées par rapport au nombre total d'images dans l'ensemble de données. Il vous indique à quelle fréquence l'algorithme est correct dans ses prédictions. Cependant, la précision ne tient pas compte de l'importance de certaines erreurs. Dans certains cas, la précision peut pénaliser une erreur plus qu'une autre.

Par exemple, dans un dataset où les classes sont déséquilibrées (c'est-à-dire qu'une classe a beaucoup plus d'échantillons que les autres), un score qui a une haute valeur peut être obtenu en prédisant simplement la classe majoritaire pour tous les échantillons. Dans ce cas, l'algorithme fonctionnerait mal sur la classe minoritaire, mais l'impact de cette erreur ne serait pas reflété dans le score de précision.

Autres métrique du coup : la précision, le recall et le F1 score

Ces mesures prennent en compte le compromis entre la précision (la fraction de vrais positifs parmi toutes les prédictions positives) et le rappel (la fraction de vrais positifs correctement identifiés).

Le score F1 est la moyenne harmonique de la précision et du rappel et fournit une mesure équilibrée des deux.

L'accurancy (precision) est une mesure utile pour **mesurer les performances globales**, mais elle peut ne pas **être suffisante dans les cas où les classes sont déséquilibrées ou lorsque certains types d'erreurs sont plus importants que d'autres.** D'autres mesures telles que la précision, le recall et le score F1 peuvent fournir une évaluation plus nuancée des performances de l'algorithme.

Qu'est-ce que le f1score ?

Il s'agit de la moyenne harmonique de précision et de rappel, qui sont elles-mêmes des mesures utilisées pour évaluer la fraction de vrais positifs parmi les prédictions positives et la fraction de vrais positifs correctement identifiés, respectivement.

Le score F1 fournit une mesure équilibrée de la précision et du rappel, ce qui peut être utile dans les cas où les deux paramètres sont importants.

Par exemple, dans une application de diagnostic médical, les faux positifs (prédire une maladie alors que le patient est en bonne santé) et les faux négatifs (manquer une maladie alors que le patient est malade) peuvent avoir de graves conséquences.

Dans ce cas, le score F1 peut fournir une évaluation plus significative des performances de l'algorithme que la précision seule.

Le score F1 va de 0 à 1.

Il est calculé comme suit :

$$ScoreF1 = \frac{2 \times (précision * rappel)}{(\text{précision} + \text{rappel})}$$

où la précision et le rappel sont calculés comme suit :

précision = vrais positifs / (vrais positifs + faux positifs)
rappel = vrais positifs / (vrais positifs + faux négatifs)

Can you give an explicit example where accuracy would not be a relevant metric? In that extreme case, can you propose a more suitable metric?

Un dataset vraiment déséquilibré => une classe a beaucoup plus d'échantillons que l'autre.

Par exemple, disons que nous avons un ensemble de données de 1000 images, où 990 images appartiennent à la classe A et seulement 10 images appartiennent à la classe B.

Si on forme l'algo qui prédit toujours la classe A, nous obtiendrions une précision de 0.99, mais ce ne serait pas une métrique utile pour évaluer les performances car on ignore complètement les performances de la classe B.

Du coup, le score F1 est utile.

La précision mesure la fraction de vrais positifs parmi toutes les prédictions positives, tandis que le rappel mesure la fraction de vrais positifs correctement identifiés.

Puisqu'on a un dataset déséquilibré on se concentre sur les résultats de B. Ce score nous donne une meilleure idée de la performance puisque on tient compte des faux positifs et des faux négatifs.

CONCLUSION : la précision c'est bien si non déséquilibre sinon F1

Do confusion matrix display all informations of algorithms' learning?

Matrice de confusion = tableau qui résume les performances d'un algorithme de classification en indiquant le nombre de vrais positifs, de vrais négatifs, de faux positifs et de faux négatifs pour chaque classe.

Elle fournit beaucoup d'infos mais ne donne pas tout :

- Pas d'informations sur la façon dont l'algorithme est arrivé à ses prédictions
- Pas d'infos sur les fonctionnalités qu'il a utilisées pour faire ces prédictions
- Pas d'infos sur le training process
- Pas d'infos performances de l'algorithme sur l'ensemble de données spécifique utilisé pour le générer
- Pas d'informations sur la façon dont l'algorithme fonctionnerait sur de nouvelles données invisibles.

CONCLUSION : bien qu'une matrice de confusion puisse être un outil utile pour évaluer les performances d'un algorithme de classification, elle n'est pas suffisante pour bien comprendre le processus d'apprentissage de l'algorithme.

Une matrice de confusion pour le MNIST dataset est une représentation des performances d'un algorithme de classification sur les images de chiffres dessinées à la main. La matrice de confusion est une table qui récapitule le nombre de vrais positifs, vrais négatifs, faux positifs et faux négatifs pour chaque classe (chiffre de 0 à 9) prédite par l'algorithme.
Voici un exemple de matrice de confusion pour un algorithme de classification appliqué au MNIST dataset:

	0	1	2	3	4	5	6	7	8	9
0	955	0	3	1	0	3	9	2	7	0
1	0	1107	5	1	0	1	5	0	16	0
2	9	5	901	13	8	0	17	11	65	3
3	3	1	17	912	0	29	3	14	25	6
4	1	0	5	0	922	0	13	0	10	31
5	9	2	1	23	5	788	17	3	40	4
6	14	3	4	0	6	9	917	0	5	0
7	1	9	20	5	5	0	1	942	7	38
8	6	6	6	16	6	6	6	7	899	16
9	8	6	1	9	28	3	0	13	8	933

Dans cet exemple, la diagonale représente le nombre de prédictions correctes pour chaque classe. Par exemple, la cellule (0,0) indique que l'algorithme a correctement prédit 955 images de chiffre 0. Les autres cellules de la matrice représentent les erreurs de classification, par exemple la cellule (0,2) indique que l'algorithme a prédit 3 images de chiffre 0 comme étant des 2.
La matrice de confusion peut être utilisée pour calculer d'autres métriques de performance telles que la précision, le rappel et le score F1. Ces métriques permettent d'évaluer la performance de l'algorithme de manière plus fine qu'avec la simple mesure de l'exactitude (accuracy).

Do you want to have a walk in the *forest* and observe *tree* growth?
Oui?