

Score de Brier

Le score de Brier est une fonction de Score qui mesure l'exactitude des prédictions probabilistes. Pour les prédictions unidimensionnelles, elle est strictement équivalente à l'erreur quadratique moyenne aux probabilités prédites.

Définition générale :

Dans le cas où une variable peut prendre plus de 2 valeurs. Le score de Brier est alors défini par :

$$B_s = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m (p_{i,j} - o_{i,j})^2$$

où R est le nombre de classes possibles dans lesquelles l'évènement peut tomber, et N le nombre total d'instances de toutes les classes. $p_{i,j}$ représente la probabilité prédite pour la classe i , $o_{i,j}$ vaut 1 si la i ème observation est de la catégorie j et 0 sinon.

Le score de Brier peut être décomposé en 3 composantes additives : incertitude, fiabilité et résolution

$$B_s = F - R + I$$

I : terme d'incertitude qui prend en compte la dispersion des observations F : terme de fiabilité qui mesure dans quelle circonstances les probabilités prévues sont proches des probabilités réelles compte tenu d'une prévision. R : terme de résolution qui mesure la distance entre les probabilités d'occurrence

Interprétation et décomposition :

Plus la valeur du score de Brier sera faible plus la prédiction sera bonne et une prévision parfaite obtiendra un score de 0. A l'inverse le plus mauvais score sera de 1. Code de la fonction :

```

# code de la fonction

import numpy as np

def brier_score_f(y_true, y_prob):
    # Args:
    # - y_true : les vraies étiquettes
    # - y_prob : les probabilités prédites pour la classe positive
    n = len(y_true)

    brier_score = np.sum((y_prob - y_true) ** 2) / n

    return brier_score

import numpy as np

# Exemple de prédictions et de vraies étiquettes
y_true = np.array([0, 1, 1, 0])
y_prob = np.array([0.1, 0.9, 0.8, 0.3])

n = len(y_true)

# Calcul du score de Brier
brier_score = brier_score_f(y_true, y_prob)

print("Score de Brier:", brier_score)

```

Score de Brier: 0.03749999999999999