La régression linéaire

Le concept et l'algorithme LinearRegression

L'objectif de la régression linéaire est de définir la relation entre une valeur cible et les autres valeurs d'un jeu de données. Pour cela, l'algorithme va voir pour chaque valeur explicative son rapport à la valeur cible en regardant la somme des écarts au carré, permettant de supposer au mieux quelles valeurs semblent avoir un impact sur la prédiction souhaitée. Une fois cette information obtenue et perfectionnée via l'entrainement sur une partie du jeu de données, l'algorithme va pouvoir être testé sur le reste de celui-ci. Il va s'agir, pour chacune des valeurs explicatives du jeu de test de prédire quelle serait la valeur de la variable cible, puis de faire une moyenne des différents résultats prenant en compte l'importance supposée de chacun.

Pourquoi cet algorithme ?

Notre problématique nous demande s'il est possible de prédire la note moyenne d'un élève durant une année à partir d'un ensemble de données. En clair nous cherchons à savoir comment, en associant nos différentes valeurs explicatives, nous pourrions retrouver la moyenne de l'élève. Nous venons de voir dans la partie précédente qu'il s'agit exactement de ce que fait cet algorithme. Il semble donc logique de l'utiliser.

Les paramètres, méthodes et attributs de l'algorithme

L'algorithme possède des paramètres mais ceux-ci ne seront pas utilisés dans ce projet et laissés à leurs valeurs par défaut :

- fit_intercept = True
- normalize = False
- copy_X = True
- n_jobs = None
- positive = False

L'algorithme possède également des attributs dont certains nous serviront :

- coef_ : il s'agit d'un tableau qui pour chaque valeur explicative estime son coefficient par rapport à la valeur cible. En clair, il s'agit de l'importance présumée de la valeur dans le calcul de la cible.
- Intercept_ : il s'agit de l'ordonnée à l'origine c'est-à-dire le point ou le modèle linéaire coupe l'axe des ordonnées pour x = 0.

Enfin, nous retrouvons certaines méthodes que nous utiliserons :

- fit : il s'agit sans doute de la méthode la plus importante de cet algorithme. En effet, on va ici permettre à ce dernier de s'entraîner sur un jeu de données d'entraînement pour lequel on lui donne la cible à atteindre. Ces deux variables sont les paramètres X et y de la méthode. Il faut se méfier en entrainant notre algorithme de lui fournir suffisamment de données pour être effectif mais pas trop pour ne pas le surentraîner ce qui l'empêcherait de généraliser pour de nouveaux jeux de données. Dans notre cas, le jeu de données n'étant pas immense (~650 lignes) nous créerons un jeu d'entrainement comprenant 80% des valeurs.
- score : cette méthode retourne le R^2 de notre prédiction, c'est-à-dire en simplifié, le taux de précision de celui-ci dont le meilleur score est donc 1. La méthode score n'entraîne pas notre algorithme, il n'est pas problématique de lui fournir de grosses portions du jeu de données à vérifier.
- predict : il s'agit en quelques sortes de la finalités de notre modèle de prédiction. Seul les variables explicatives sont données à la méthodes et celle-ci nous retourne ses prédictions basée sur l'algorithme perfectionné par l'entraînement.