La régression linéaire et le renforcement du gradient

Auteur : Ouedraogo Somkiéta Rahim Alex

Email : s.r.a.ouedraogo@gmail.com

Téléphone :57 74 21 32 / 72 58 78 71

Définition

La régression linéaire est un moyen de trouver une ligne droite qui correspond le mieux à un ensemble de points de données. C'est comme tracer une ligne sur un graphique qui passe par autant de points que possible. Cette ligne nous aide à prédire la valeur d'une chose en fonction de la valeur d'une autre chose. Par exemple, nous pouvons utiliser la régression linéaire pour prédire le montant du salaire d'une personne en fonction de son nombre d'années d'études. La pente de la ligne nous indique de combien la valeur prédite change pour chaque unité de changement dans l'autre valeur, et l'ordonnée à l'origine nous indique où la ligne croise l'axe des y. La régression linéaire est un outil simple et utile pour comprendre et prédire les relations entre les variables dans les données.

Le renforcement du gradient quant à lui est une puissante technique d'apprentissage automatique qui combine les prédictions de nombreux modèles faibles (tels que les arbres de décision) pour créer une prédiction globale solide. Cela fonctionne en entraînant à plusieurs reprises de nouveaux modèles pour corriger les erreurs commises par les modèles précédents.

Imaginez que vous essayez de prédire le prix d'une maison et que vous avez un groupe d'amis qui essaient également de faire des prédictions. Chacun de vos amis est doué pour prédire certains aspects du prix de l'immobilier, mais aucun d'eux n'est parfait. Donc, vous décidez de prendre leurs prédictions et de les combiner pour faire une prédiction plus précise.

Vous commencez par faire votre prédiction initiale en vous basant sur vos propres connaissances, mais vous savez qu'elle n'est peut-être pas parfaite. Vous demandez ensuite à votre premier ami de se concentrer sur les erreurs que vous avez commises et de faire sa propre prédiction pour les corriger. Vous prenez leur prédiction et mettez à jour votre prédiction initiale en conséquence.

Ensuite, vous demandez à votre deuxième ami de se concentrer sur les erreurs restantes et de faire sa prédiction. Vous continuez ce processus en demandant à chacun de vos amis de corriger les erreurs commises par les prédictions précédentes jusqu'à ce que vous soyez satisfait de la prédiction globale.

Ceci est similaire au fonctionnement de l'amplification de gradient. Chaque modèle faible (ami) se concentre sur la correction des erreurs commises par les modèles précédents et les prédictions de tous les modèles sont combinées pour créer une prédiction finale plus précise. La partie "gradient" fait référence aux ajustements apportés aux prédictions en fonction des erreurs ou des résidus des modèles précédents, et la partie "boosting" fait référence au processus itératif d'amélioration des prédictions avec chaque modèle suivant.

Awesome_Linear_Regression

Le référentiel GitHub "Awesome_Linear_Regression" (https://github.com/somkietacode/Awesome_Linear_Regression) est une collection de ressources liées à la régression linéaire, une technique statistique populaire utilisée dans l'apprentissage automatique et l'analyse de données.

En termes simples, la régression linéaire est une méthode utilisée pour trouver la relation entre deux variables, généralement désignées par X et Y, où X est la variable d'entrée et Y est la variable de sortie. La régression linéaire essaie d'ajuster une ligne droite aux points de données de manière à capturer au mieux la tendance ou le modèle sous-jacent dans les données.

Le référentiel « Awesome_Linear_Regression » sur GitHub contient divers documents liés à la régression linéaire, tels que des exemples de code, des didacticiels, des ensembles de données et des documents de recherche. Ces ressources visent à aider les débutants et les praticiens à comprendre et à mettre en œuvre la régression linéaire dans leurs propres projets.

En partageant ces ressources sur GitHub, je fournis une plate-forme gratuite et ouverte à toute personne intéressée à apprendre ou à travailler avec la régression linéaire. Les utilisateurs peuvent accéder, télécharger et contribuer au référentiel, ce qui en fait une ressource collaborative et communautaire pour l'apprentissage et l'amélioration des compétences en régression linéaire.

Auto_Gradient_Boosting

Le référentiel contient du code lié à une technique d'apprentissage automatique appelée "Gradient Boosting". Gradient Boosting est une méthode d'apprentissage d'ensemble populaire utilisée pour faire des prédictions ou créer des modèles pour des tâches telles que la régression, la classification et le classement. Il combine les prédictions de plusieurs modèles "faibles", tels que les arbres de décision, pour créer un modèle "fort" plus précis et plus puissant.

Le référentiel est étiqueté "Auto_Gradient_Boosting", ce qui suggère qu'il peut contenir du code pour une version automatisée ou optimisée de Gradient Boosting. Il inclut des fonctionnalités qui règlent automatiquement les hyperparamètres, gèrent les données manquantes ou effectuent une sélection de fonctionnalités pour optimiser les performances du modèle Gradient Boosting.

Le référentiel contient également de la documentation, des exemples de code et d'autres ressources pour aider les utilisateurs à comprendre et à utiliser l'implémentation Gradient Boosting fournie dans le référentiel. Il est destiné aux praticiens de l'apprentissage automatique, aux scientifiques des données ou à toute personne intéressée par l'utilisation de Gradient Boosting pour ses propres projets.

Veuillez noter que pour bien comprendre les spécificités du référentiel, il est recommandé de consulter le code et la documentation que j'ai fournis.

Application

Pour cette application vous devez vous assurer de disposer de python installer sur votre ordinateur si vous ne savez pas comment faire je vous prie de vous référer à ma publication précédente téléchargeable ici .

Une fois python installer vous pouvez installer les librairies à l'aide des commandes suivante dans votre terminale :

```
pip install Awesome-Linear-Regression
```

```
pip install Auto-Gradient-Boosting
```

Les données

Nous cherchons à prédire ici les risques de maladie cardiovasculaire chez un patient à partir de diagnostique précédent. Pour ce faire nous avons une base de données contenant des mesures et information prise sur 300 patients. Voir la figure ci-dessous :

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1

Explication des différentes colonnes :

- **Age** : âge du patient ;
- **Sex** : sex du patient (1 pour male & 0 pour femelle) ;
- **Cp** : Douleur thoracique (typique, asymptotique, non angineuse, non typique) ;
- **Trestbps** : Tension artérielle au repos ;
- **Chol** : Cholesteral sérique en mg/dl ;
- **Fbs** : Glycémie à jeun > 120 mg/dl (1 = vrai ; 0 = faux) ;
- **Restcg** : Résultats électrocardiographiques au repos ;
- **Thalach** : Fréquence cardiaque maximale atteinte ;
- **Exang** : Angor d'effort (1 = oui ; 0 = non) ;
- **Oldpeak** : Dépression ST induite par l'exercice par rapport au repos ;
- **Slope** : Pente du segment ST d'effort maximal ;
- **Ca** : Nombre de vaisseaux principaux colorés par fluoroscopie (0 - 3) ;
- **Thal** : 3 = normale ; 6 = défaut corrigé ; 7 = défaut réversible) ;
- **Target** : AHD - Diagnostic de maladie cardiaque (1 = oui ; 0 = non) ;

Le programme informatique

Configuration et initialisation :

```python
# Importation des bibliothèques nécessaire

import numpy as np
from Auto_Gradient_Boosting import AGB
from Awesome_Linear_Regression import linearregression as LR
import pandas as pd
import random

# Lecture des données
df = pd.read_csv('heart.csv')
df = df.apply(pd.to_numeric, errors='coerce')
df = df.dropna()

# Selection des paramètre de prédiction

X = np.matrix(df[["age","sex","cp","trestbps","chol","fbs","restecg","thalach","exang","oldpeak","slope","ca","thal"]].to_numpy() )
Y = np.matrix(df[["target"]].to_numpy())


# Parametrage du gradient boosting
learning_rate = 0.80  # Choix de la vitesse d'aprentissage
agb = AGB(X,Y,learning_rate)

# Parametrage de la régression linéaire

Lr = LR(X,Y)
Beta, rss = Lr.leastsquare()
```

Prédiction :

```python
# Prediction du risque de maladie cardiovasculaire pour deux patient avec les deux modèles

x1 = np.matrix([[63,1,3,145,233,1,0,150,0,2.3,0,0,1]])
x2 = np.matrix([[41,1,0,110,172,0,0,158,0,0,2,0,3]])

print("Patient 1 :")
print("Gradient Boost : ",agb.predict(x1))
print("Regression lineaire : ",Lr.predict(x1))
print("Patient 2 :")

print("Gradient Boost : ",agb.predict(x2))
print("Regression lineaire : ",Lr.predict(x2))
```

Résultat :

```
C:\Users\sraou\Desktop\PYTHON CODE>python tuto.py
Patient 1 :
Gradient Boost :  0.7203368746850547
Regression lineaire :  0.7642824794949011
Patient 2 :
Gradient Boost :  0.5867871744085285
Regression lineaire :  0.5973453541492566
```

Le patient 1 à au moins 72% de chance de développer ou d'avoir une maladie cardiovasculaire tant dis que le patient 2 en a 58.

Conclusion

G-Boost, également connu sous le nom de Gradient Boosting, est une technique d'apprentissage automatique puissante et populaire utilisée à la fois pour les tâches de régression et de classification, tandis que la régression linéaire est une méthode statistique simple et largement utilisée pour prédire la relation entre deux variables. Voici quelques différences clés entre G-Boost et la régression linéaire :

1. Complexité du modèle : G-Boost est une méthode d'ensemble qui combine plusieurs apprenants faibles, tels que des arbres de décision, pour créer un modèle prédictif solide. Il peut capturer des modèles non linéaires complexes dans les données et est moins sujet au surajustement. D'autre part, la régression linéaire suppose une relation linéaire entre les variables dépendantes et indépendantes, qui peut ne pas toujours être précise pour les modèles de données complexes.

2. Interprétabilité : la régression linéaire produit une équation linéaire simple qui peut être facilement interprétée pour comprendre la relation entre les caractéristiques d'entrée et la variable cible. Il fournit des coefficients pour chaque caractéristique, indiquant la force et la direction de la relation. G-Boost, d'autre part, implique une combinaison de plusieurs arbres de décision, ce qui le rend plus complexe et difficile à interpréter.

3. Traitement des valeurs manquantes : la régression linéaire nécessite des données complètes sans valeurs manquantes pour des prévisions précises. S'il manque des valeurs dans les entités en entrée, la régression linéaire peut ne pas fonctionner correctement. D'autre part, G-Boost peut gérer les valeurs manquantes en utilisant des techniques telles que le fractionnement de substitution et peut gérer efficacement les données manquantes pendant le processus de formation du modèle.

4. Robustesse aux valeurs aberrantes : la régression linéaire est sensible aux valeurs aberrantes dans les données, car les valeurs aberrantes peuvent affecter de manière significative la relation linéaire et les coefficients. G-Boost, d'autre part, est plus robuste aux valeurs aberrantes en raison de la nature d'ensemble de la combinaison de plusieurs arbres, ce qui peut aider à atténuer l'impact des valeurs aberrantes.

5. Performance : G-Boost est connu pour sa haute précision et ses performances prédictives, en particulier pour les ensembles de données complexes avec des relations non linéaires. Il peut souvent surpasser la régression linéaire lorsque les modèles de données sous-jacents sont complexes. La régression linéaire, en revanche, peut être plus appropriée pour des ensembles de données simples avec des relations linéaires.

6. Temps de formation : la régression linéaire est un algorithme simple et rapide qui peut être formé rapidement, même sur de grands ensembles de données. G-Boost, étant une méthode

d'ensemble avec plusieurs arbres, peut nécessiter plus de temps pour la formation, en particulier lorsqu'il s'agit d'un grand nombre d'apprenants faibles ou d'arbres profonds.

7. Évolutivité : la régression linéaire est un modèle linéaire et peut ne pas s'adapter correctement à de grands ensembles de données avec un grand nombre d'entités. G-Boost, d'autre part, peut gérer efficacement de grands ensembles de données et des espaces d'entités de grande dimension.

En résumé, G-Boost et la régression linéaire sont deux techniques utiles pour la prédiction, mais elles ont des forces et des faiblesses différentes. G-Boost est plus puissant, flexible et précis, mais peut être plus complexe et coûteux en calculs. La régression linéaire est plus simple, interprétable et adaptée aux relations linéaires, mais peut ne pas capturer les modèles complexes dans les données. Le choix entre G-Boost et la régression linéaire dépend des caractéristiques spécifiques des données et des objectifs de la tâche de prédiction.

Vous pouvez télécharger les documents qui vous permettrons de reproduire cette expérience chez vous en cliquant ici.