

LRT

LINEAR REGRESSIO...

GitHub - codeonym/LRT: Linear Regre...

▼ IMPORTS

- ARGUMENTPARSER
- TIME
- NUMPY
- SYS
- CSV
- PANDAS
- MATHPLOTLIB
- OS
- COLORAMA

▼ FUNCTIONS

▼ is_valid_csv()

- ARG: chemin_fichier
- RETURN: TRUE ou FALSE

▼ FONCTIONALITÉ

- Tente de lire le fichier CSV spécifié par `chemin_fichier`.
- Vérifie les erreurs courantes (format CSV, existence du fichier, autorisations).

▼ ALGORITHM

- Essayer d'ouvrir le fichier à `chemin_fichier`.
- Utiliser `csv.reader` pour lire la première ligne, validant le format CSV.
- Gérer les erreurs potentielles telles que `csv.Error`, `FileNotFoundError` et `PermissionError`.

▼ load_dataset()

- ARG: chemin_fichier
- RETURN: X, y, labels

- ▼ **FONCTIONALITÉ**
 - Lit un fichier CSV à l'aide de pandas.
 - Extrait les caractéristiques (``X``), la variable cible (``y``) et les noms de colonnes.
- ▼ **ALGORITHM**
 - Utiliser pandas pour lire le fichier CSV.
 - Extraire les caractéristiques (``X``) et la variable cible (``y``).
 - Obtenir les noms de colonnes et créer un dictionnaire d'étiquettes.
- ▼ **compute_cost()**
 - ARG: X, y, theta
 - RETURN: cost
- ▼ **FONCTIONALITÉ**
 - Lit un fichier CSV à l'aide de pandas. Calcule la fonction de coût pour la régression linéaire.
 - Utilise la fonction hypothèse et la somme des erreurs au carré.
- ▼ **ALGORITHM**
 - Calculer l'hypothèse en utilisant le produit scalaire de ``X`` et ``theta``.
 - Calculer la somme des erreurs au carré entre l'hypothèse et ``y``.
 - Multiplier par ``1/2`` et renvoyer le résultat.
- ▼ **h_theta_func()**
 - ARG: X, theta
 - RETURN: h_theta
- ▼ **FONCTIONALITÉ**
 - Calcule la fonction hypothèse pour la régression linéaire.
 - Renvoie le produit scalaire de ``X`` et ``theta``.
- ▼ **ALGORITHM**
 - Calculer le produit scalaire de ``X`` et ``theta``.
 - Renvoyer le résultat.
- ▼ **batch_gradient_descent()**
 - ARG: X, y, learning_rate, iterations, convergence_threshold

- RETURN: theta, nb_iterations

▼ FONCTIONALITÉ

- Implémente la descente de gradient par lots pour la régression linéaire.
- Met à jour itérativement les paramètres (`theta`) pour minimiser la fonction de coût.

▼ ALGORITHM

- Initialiser `theta` à zéro et définir le coût précédent à l'infini.
- Itérer jusqu'à la convergence ou atteindre le nombre maximum d'itérations.
- Mettre à jour chaque paramètre en utilisant la formule de descente de gradient.
- Vérifier la convergence en comparant les coûts actuel et précédent.
- Sortir si convergé ; sinon, mettre à jour le coût précédent.
- Renvoyer `theta` et le nombre d'itérations.

▼ stochastic_gradient_descent()

- ARG: X, y, learning_rate, iterations, convergence_threshold
- RETURN: theta, nb_iterations

▼ FONCTIONALITÉ

- Implémente la descente de gradient stochastique pour la régression linéaire.
- Met à jour itérativement les paramètres (`theta`) en utilisant un point de données aléatoire.
- Renvoie les paramètres optimisés et le nombre d'itérations.

▼ ALGORITHM

- Initialiser `theta` à zéro et définir le coût précédent à l'infini.
- Itérer jusqu'à la convergence ou atteindre le nombre maximum d'itérations.
- Pour chaque point de données, mettre à jour chaque paramètre en utilisant la formule de descente de gradient stochastique.
- Vérifier la convergence en comparant les coûts actuel et précédent.
- Sortir si convergé ; sinon, mettre à jour le coût précédent.
- Renvoyer `theta` et le nombre d'itérations.

▼ plot_data()

- ARG: X, y, theta, nom_fichier, labels, dim, action
- RETURN: NULL
- ▼ FONCTIONALITÉ
 - Choisi la fonction de tracé appropriée en fonction de la dimension.
 - Enregistre ou affiche le graphique en fonction de l'action spécifiée.
- ▼ ALGORITHM
 - Vérifier la dimension et appeler `plot_data_2d` ou `plot_data_3d`.
 - Enregistrer ou afficher le graphique en fonction de l'action spécifiée.
- ▼ plot_data_2d()
 - ARG: X, y, theta, fichier, labels
 - RETURN: plt
 - ▼ FONCTIONALITÉ
 - Génère un graphique de dispersion 2D de l'ensemble de données et de la droite de régression.
 - ▼ ALGORITHM
 - Graphique de dispersion des points de données en utilisant `X[:, 0]` et `y`.
 - Tracer la droite de régression en utilisant `h_theta_func(X[:, 0], theta[0])`.
 - Étiqueter les axes et ajouter une légende.
 - Renvoie l'objet du graphique.
- ▼ plot_data_3d()
 - X, y, theta, fichier, labels
 - RETURN: plt
 - ▼ FONCTIONALITÉ
 - Génère un graphique de dispersion 3D de l'ensemble de données et du plan de régression.
 - ▼ ALGORITHM
 - Graphique de dispersion des points de données 3D en utilisant `X[:, 0]`, `X[:, 1]` et `y`.
 - Tracer le plan de régression en utilisant `theta[0] * x_mesh + theta[1] * y_mesh`.
 - Étiqueter les axes et ajouter une légende.

- Renvoie l'objet du graphique.

▼ print_statistics()

- ARG: y, predictions, iterations, theta, exec_time
- RETURN: NULL

▼ FONCTIONALITÉ

- affiche un résumé des statistiques.

▼ ALGORITHM

- Calculer le coefficient de détermination en utilisant `calculate_r_squared`.
- Afficher les statistiques incluant le nombre total d'itérations, `theta`, le temps d'exécution et le coefficient de détermination.

▼ calculate_r_squared()

- ARG: y, predictions
- RETURN: coef

▼ FONCTIONALITÉ

- Calcule le coefficient de détermination pour les prédictions du modèle.

▼ ALGORITHM

- Calculer la moyenne de `y`.
- Calculer la somme totale des carrés et la somme résiduelle des carrés.
- Renvoyer le coefficient de détermination.

▼ test_algo()

- ARG: theta, labels, n
- RETURN: NULL

▼ FONCTIONALITÉ

- Permet à l'utilisateur d'entrer des caractéristiques personnalisées et prédit la cible.

▼ ALGORITHM

- Inviter l'utilisateur à entrer des caractéristiques.
- Calculer les prédictions en utilisant le modèle entraîné.
- Afficher le résultat.

▼ Main

▼ FONCTIONNALITÉ

- Analyse les arguments de la ligne de commande, charge les données et exécute la régression linéaire.

▼ ALGORITHM

- Initialiser l'analyseur d'arguments et définir les arguments de la ligne de commande.
- Analyser les arguments de la ligne de commande.
- Valider le fichier CSV.
- Charger l'ensemble de données.
- Exécuter l'algorithme de descente de gradient choisi.
- Afficher les statistiques.
- Éventuellement, exécuter un test de caractéristique personnalisée.
- Éventuellement, tracer les données.