Programmer votre IA pour acheter votre voiture

Vous travaillez depuis un an en tant qu'expert en data et avez économisé assez d'argent pour acheter une voiture. En tant qu'expert en données et étant assez économe, vous voulez en avoir pour votre argent !!!

Imaginez que vous ayez également des données du site Web de voitures CarDekho, qui contient des informations sur une grande variété de voitures, y compris leur prix. Vous réalisez que vous pouvez utiliser ces données pour vous assurer d'obtenir une bonne affaire sur une nouvelle voiture. En particulier, vous pouvez déterminer exactement combien vous devriez payer pour un type de voiture spécifique. Cela peut être particulièrement utile si vous rencontrez un vendeur de voitures difficile!

Mais la question est de savoir comment utiliser les données pour déterminer combien vous devriez payer ?

1. La première étape est la récupération des données.
Nous utiliserons la bibliothèque de science des données appelée Pandas pour charger l'ensemble des données. Au travers de Pandas, il est possible de lire le fichier de données (carData.csv). Les données seront ensuite affectées et stockées dans une variable, par exemple : car_data.

2. Explorer les données.
Parcourir les colonnes pour s'approprier les données. Quelle est la taille du jeux de données. Calculer quelques statistiques de base (moyenne, médiane, quartile, tracer la distribution avec Matplotlib (histogramme).

3. Charger vos données dans une base de données MySQL. A partir de cette question, vos données devront être récupérées directement via la base de données.

4. Visualiser les données grâce à la librairie Seaborn.
Notamment avec le type de tracé `catplot`, expliquer dans quel cas ce type de tracé est pertinent.

4. Quantifier la relation entre l'âge et le prix de vente
Réaliser une veille sur la régression linéaire.

Pour rappel ... la régression linéaire est une méthode permettant de découvrir la relation entre deux variables de l'ensemble de données, telles que le prix de la voiture et l'année de fabrication. Les Data Scientists s'appuient sur cette méthode pour résoudre un large éventail de problèmes, notamment en matière de prédiction.

Est-ce que notre jeux de données est adapté à ce type d'algorithme (ou existe-t-il une corrélation linéaire entre les variables ?)
Proposer un outil de visualisation Matplotlib permettant d'appuyer votre argumentation.

1. **Appliquer l'algorithme de régression linéaire univariée** en vous aidant de la librairie **Numpy.**
2. **Appliquer l'algorithme de régression linéaire univariée** en vous aidant de la librairie **Scipy.**
3. **Appliquer l'algorithme de régression linéaire univariée** en vous aidant de la librairie **sklearn**
4. **Améliorer le modèle en utilisant plusieurs variables d'entrée**, telles que `Kms_Driven` et `Transmission` (réaliser une régression linéaire multiple en Python avec sklearn).
5. **Conclure**

5. Créer VOTRE Class LinearRegression. Ne pas utiliser de fonctions de régression linéaire existante (comme précédemment vue).

6. Pensez-vous possible de résoudre ce problème en implémentant un algorithme de Support Vector Machines (SVM) ? Justifier votre réponse.

7. Héberger vos sources sous github

8. Proposer un dashboard comme livrable de votre étude.

9. Question bonus : Vous avez trouvé votre voiture au meilleur prix, vous l'achetez. 3 jours après, vous êtes en panne. Quelles données manque-il à votre analyse ?