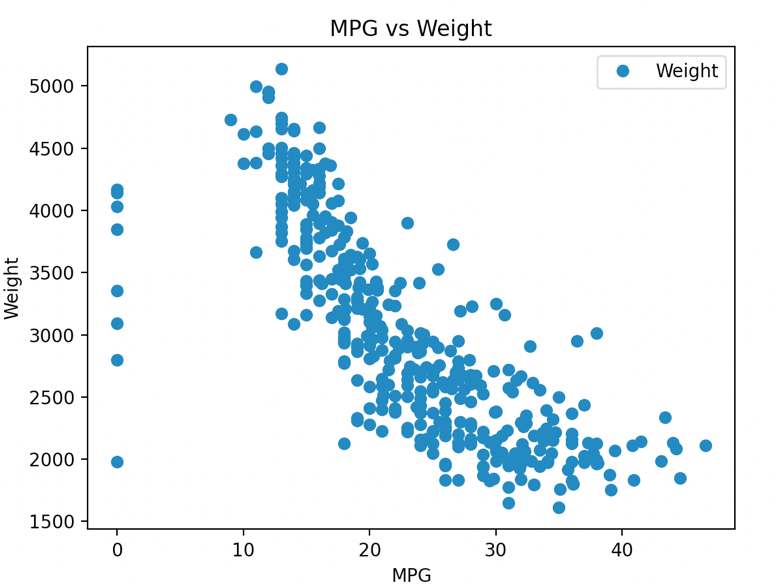
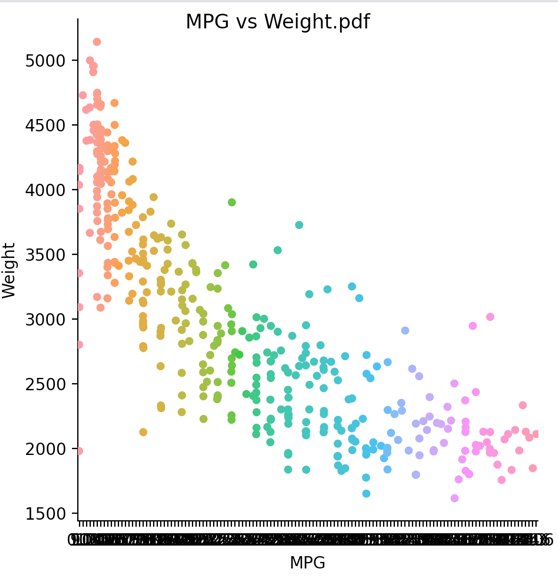
Comments

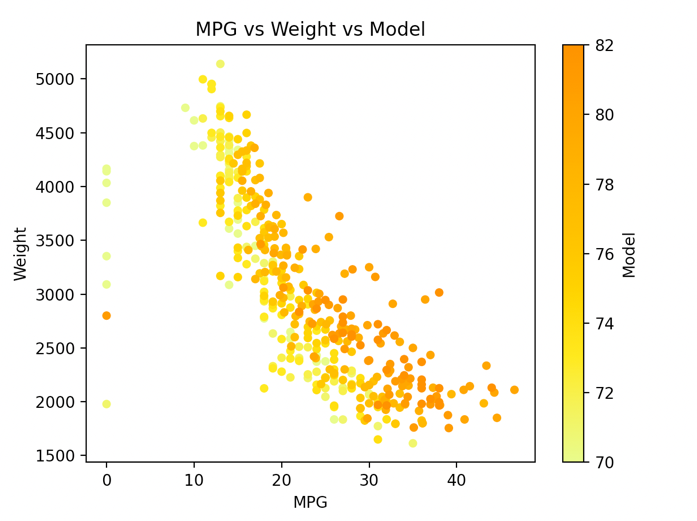
Pour ce projet nous avons utilisé un dataset de voiture. Le dataset comportait différente indication :

* Car
* MPG
* Cylinders
* Displacement
* Horsepower
* Weight
* Acceleration
* Model
* Origin

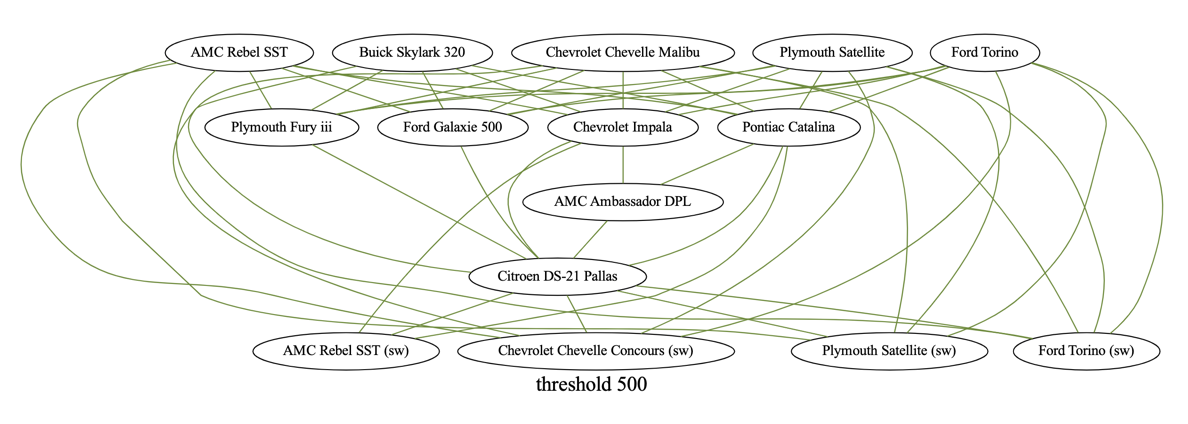
Nous nous sommes focalisés sur les MPG (Miles per Galon) et le poids (Weight)

Comment la consommation d’essence évolue en fonction du poids d’un véhicule ?





Pour le processing, dans le fichier build\_metrics.py. Nous avons fait un graph de similarité en fonction de leurs poids et de leurs consommations mais nous avons rajouté l’origine du véhicule dans le calcul de la distance euclidienne. Les valeurs de dissimilarité varient entre 0 et 1200 environ. Nous avons donc mis un seuil à 500.



Ensuite nous avons decidé de faire une clusterisation des mêmes valeurs avec notre dataset. Nous avons donc implété un algorithme K-mean.

L’algorithme de K-mean place un nombre de centroide. Ensuite il calcul la distance entre la data et les centroides et assigne au centroide le plus proche.

Pour un nombre d’itération définit, nous déplaçons les centroides pour optenir le resultat le plus précis.

Pour définir le nombre de centroide nous avons mis en place 2 methodes :

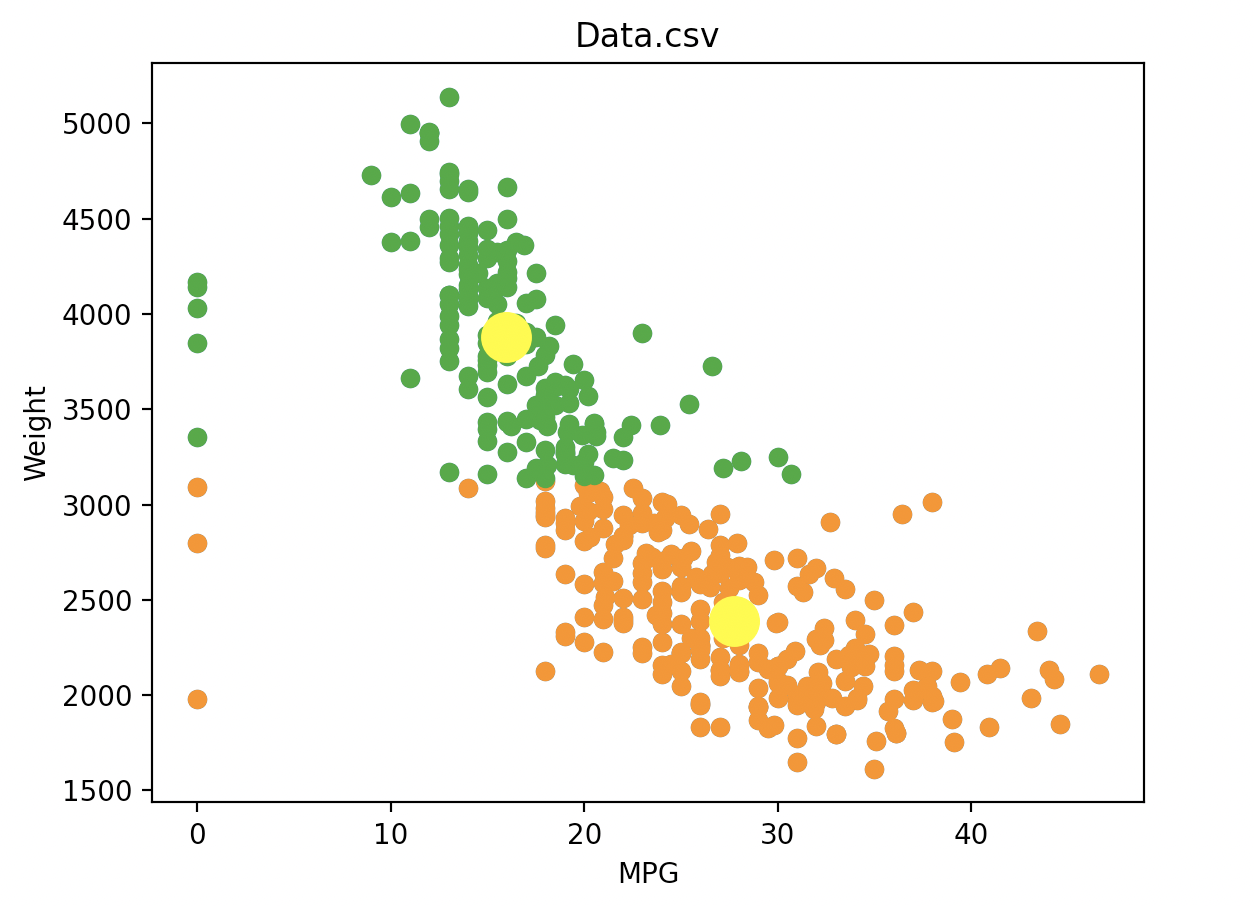
* L’elbow method
* Silhouette Analysis

Pour l’« elbow methow », il faut analyser la courbe afficher et regarder lorsque la courbe change de direction à la manière d’un coude.

Pour la « Silhouette Analysis », on calcule un coefficient de silhouette et lorsque le score et le plus élevé : il s’agit du nombre de cluster.

Les 2 methodes nous ont retourné le même resultat : 2 centroides.

Nous avons choisis le K-mean car il était rapide d’exécution et simple de compréhension.



On remarque après cette clusterisation, que les 2 clusters confirment que plus le poids du véhicule est élevé plus la consommation est haute.