

Compte rendu



Visualization of Massive Data

/ Table des matières

/ Présentation de l'échantillon	2
/ Librairies utilisées	2
/ Correlations	2
/ Explications de nos analyses	3
/ Conclusion	6

Depuis les différents liens indiqués sur le sujet, nous avons trouvé un échantillon de données correspondant à une liste de voitures classées principalement par modèle et région de production. Et nous nous sommes posé cette question :

Est-ce que les caractéristiques d'une voiture sont influencées par le pays d'où elle provient ?

/ Présentation de l'échantillon

Nom du fichier : data.csv (./data.csv)

Description : Ensemble de voitures produites à travers le monde et différentes statistiques à propos de leur puissance.

Liste des champs : **(1)**Nom du modèle, **(2)**MPG (Miles per Gallon, consommation d'essence du véhicule), **(3)**Nombre de cylindres, **(4)**Cylindrée (La cylindrée est le volume balayé par le déplacement d'une pièce mobile dans une chambre hermétiquement close pour un mouvement unitaire.), **(5)**Puissance en chevaux, **(6)**Accélération (nombre de secondes pour passer de 0 à 60 MPH), **(7)**Année de production du modèle, **(8)**Origine du modèle.

Source : <https://perso.telecom-paristech.fr/eagan/class/igr204/datasets> (Link 2 dans le sujet)

/ Librairies utilisées

Pour réaliser nos analyses nous avons utilisé plusieurs fonctions de librairies externes:

Nom de la fonction : LabelEncoder

Paramètres : Pas de paramètres

Librairie : "sklearn.preprocessing"

Description : Initialise le modèle qui encode les étiquettes cibles avec une valeur comprise entre 0 et n_valeurs-1.

Justificatif : Nous a permis de traduire les données étiquetées en nombre entier pour pouvoir les utiliser plus facilement dans notre algorithme (car un algorithme de machine learning ne traite pas de données étiquetées).

Nom de la fonction : `LabelEncoder().fit_transform(param1)`

Paramètres : param1: Valeurs étiquetées à transformer en valeurs

Librairie : "sklearn.preprocessing"

Description : Encode les étiquettes cibles avec une valeur comprise entre 0 et n_valeurs-1.

Justificatif : Nous a permis de traduire les données étiquetées en nombre entier pour pouvoir les utiliser plus facilement dans notre algorithme (car un algorithme de machine learning ne traite pas de données étiquetées).

Nom de la fonction : `DecisionTreeClassifier()`

Paramètres : Pas de paramètres

Librairie : "sklearn.tree"

Description : Génère un modèle de machine learning utilisant l'algorithme de classification "decision tree".

Justificatif : Nous a permis de générer le modèle qui a classifié nos données et repérer les données "aberrantes" de notre échantillon de données.

Nom de la fonction : `DecisionTreeClassifier().fit(inputs_n, target)`

Paramètres : inputs_n: Données permettant à l'algorithme de s'entraîner à créer un arbre de décision afin de classifier chaque donnée de l'échantillon. target: Données permettant à l'algorithme de connaître le résultat souhaité ce qui va lui permettre d'apprendre et donc de classifier l'ensemble de notre échantillon de données.

Librairie : "sklearn.tree"

Description : Génère un arbre de décision permettant de classifier des données.


Justificatif : Nous a permis d'entraîner notre modèle de machine learning sur une partie de notre échantillon de données afin qu'il soit le plus fiable possible lorsque nous utiliserons le modèle sur la totalité de l'échantillon de données.

Nom de la fonction : `DecisionTreeClassifier().predict(total_inputs_n)`

Paramètres : total_inputs_n: Totalité de notre échantillon de données après les avoir prétraitées

Librairie : "sklearn.tree"

Description : Génère un arbre de décision permettant de classifier des données.

Justificatif : Nous a permis de prédire si un véhicule était efficace ou non en fonction de nos critères (cf.  DATAVIZ).

Nom de la fonction : `pearsonr(x, y)`

Paramètres : x: Tableau d'entrées, y: Tableau d'entrées

Librairie : "scipy.stats"

Description : Calcule le coefficient de pearson pour x par rapport à y.

Justificatif: Nous a permis de calculer facilement les corrélations entre chaque colonnes de notre échantillon de données.

Nom de la fonction KMeans(n_clusters=options.clusters, init='k-means++', random_state=0)

Paramètres : n_clusters: Le nombre de clusters que l'on cherche à former

init: permet le choix d'un mode optimisé d'initialisation des centroïdes afin d'accélérer la convergence

random_state: permet de donner la seed pour le random (ici 0)

Librairie : "sklearn.cluster"

Description : Génère un modèle de machine learning utilisant l'algorithme de classification "K-Moyenne".

Justificatif: Nous a permis de générer le modèle qui a classifié nos données

Nom de la fonction KMeans.fit(df) & KMeans.predict(df)

Paramètres : dataframe: Le jeu de données en entrée

Librairie : "sklearn.cluster"

Description : Génère une liste donnant l'appartenance de chaque entrée à un cluster, permettant de classifier des données.

Justificatif: Nous a permis de catégoriser facilement les données en plusieurs clusters distincts

/ Répartition du travail

Fichier : analysis.py

Contributeurs : Benoît Yver de la Bruchollerie, Luc Braun-Exposito.

Liste des fonctions : cars_production_by_country (Luc), preprocess_dataset (Luc), potential_outliers (Benoît), corr_horsepower_cylinder(Luc), corr_horsepower_weight (Luc), corr_horsepower_displacement (Luc), corr_horsepower_origin (Luc), corr_horsepower_acceleration (Luc), corr_horsepower_MPG (Luc), corr_horsepower_model (Luc), average_horse_power_by_country (Benoît), average_car_weight_by_country (Benoît), average_displacement_by_country (Benoît)

Fichier : artificial_dataset.py

Contributeurs : Christophe Chauvot

Fichier : supervised_learning.py

Contributeurs : Benoît Yver de la Bruchollerie, Luc Braun-Exposito.

Liste des fonctions : preprocess_dataset (Benoît), predict_weight (Luc)

Fichier : unsupervised_learning.py

Contributeurs : Christophe Chauvot

Fichiers : visualization_1.py/visualization_2.py

Contributeurs : Benoît Yver de la Bruchollerie, Luc Braun-Exposito

Fichier : Report.pdf

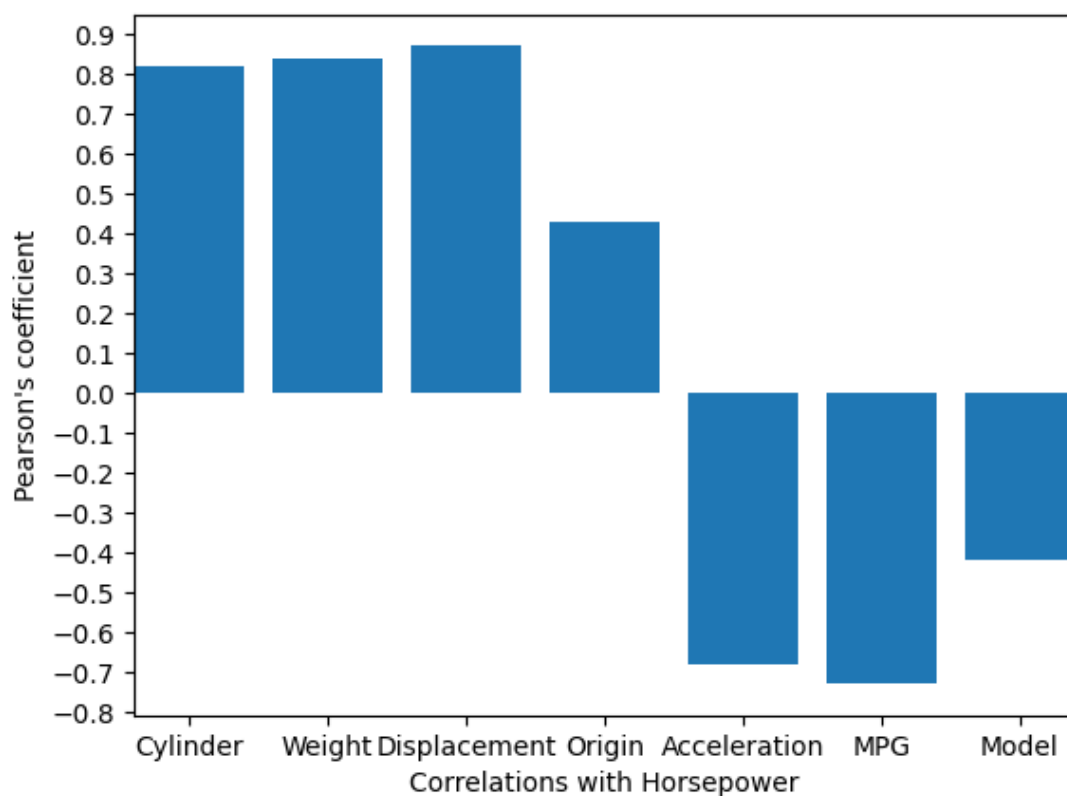
Contributeurs : Benoît Yver de la Bruchollerie, Luc Braun-Exposito, Christophe Chauvot

/ Correlations

Nom des fichiers/scripts : analysis.py, supervised_learning.py, unsupervised_learning.py, vizualization_1.py, vizualization_2.py

Langage : Python 3

Afin de définir les différents degrés de corrélations entre nos valeurs nous avons décidé d'utiliser le coefficient de Pearson, celui-ci nous a permis de remarquer que la puissance en chevaux d'une voiture était principalement liée à sa cylindrée, son poids et le nombre de cylindres. En moindre mesure mais de manière non négligeable l'origine a un impact sur la puissance de la voiture.

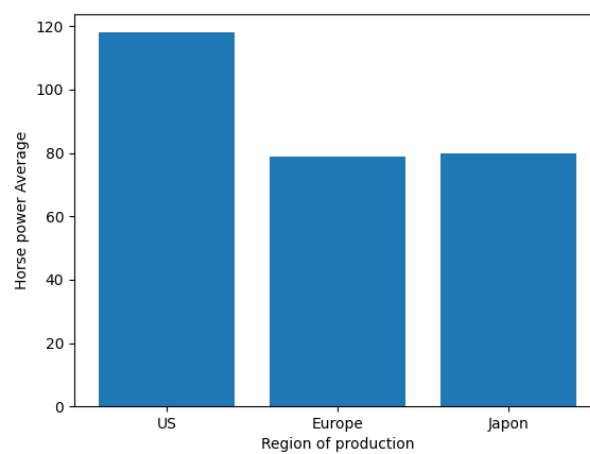
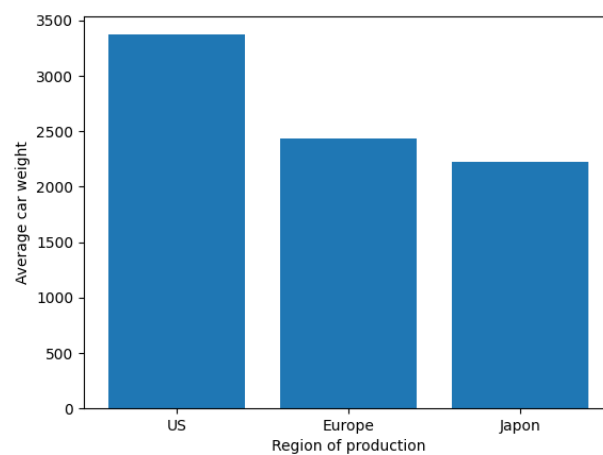
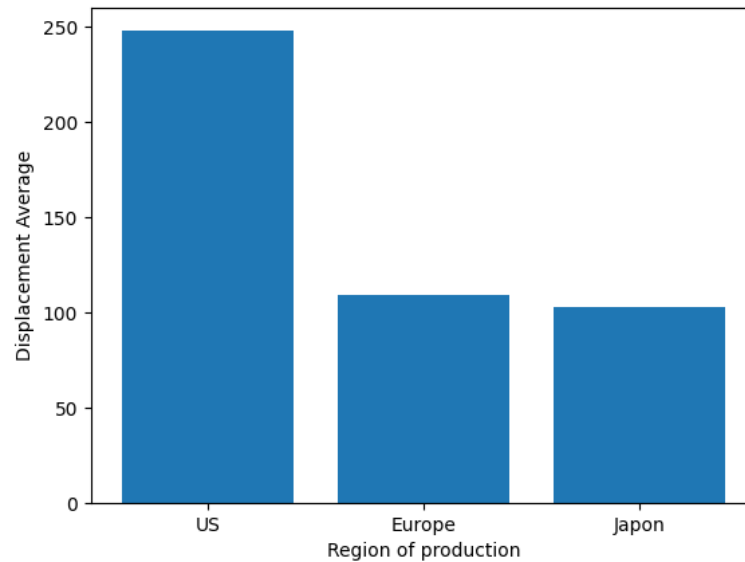


/ Explications de nos analyses

En plus du calcul de corrélations, nous avons effectué différentes analyses permettant d'apporter des éléments de réponses à notre question principale. Nous avons donc repris les

résultats des calculs de corrélation et avons regardé la moyenne de chaque région de production sur les caractéristiques influençant le plus une voiture.

De cette analyse nous en retirons que les Etats Unis produisent en moyenne des voitures bien plus grosses, puissantes et avec de plus grosses cylindrées qu'en Europe ou au Japon.



Pour finir nous avons décidé arbitrairement des critères définissant ce qu'est une voiture "peu efficace" (une voiture a un moteur "peu efficace") si :

- Son accélération est inférieure à 10 pour une cylindrée de 400 et +
- Son accélération est inférieure à 12.5 pour une cylindrée de 100 et -).

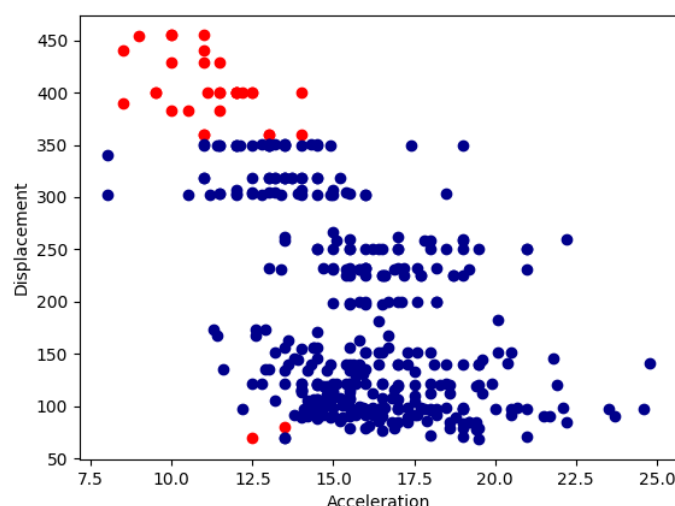
Et avec ces critères nous avons créé un échantillon de données d'entraînement pour un arbre de décision permettant la classification automatique de nos données en ajoutant une colonne "Performance" qui vaut 0 si la voiture est peu efficace et 1 si elle est efficace.

Nom du fichier : data_algo.csv (./data_algo.csv)

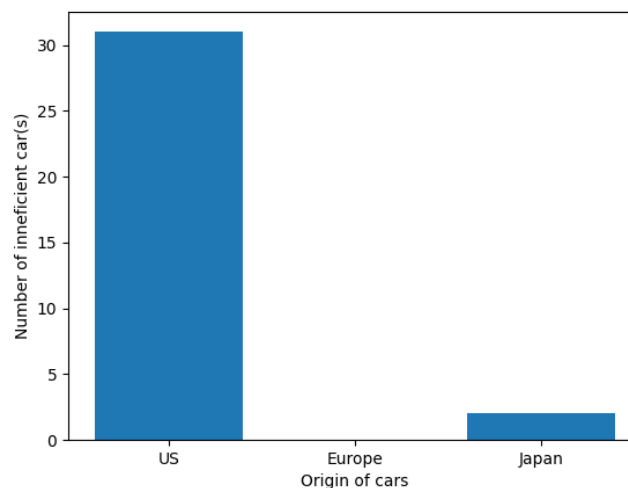
Description : Échantillon de données d'entraînement pour un arbre de décision

Liste des champs : **(1)**Nom du modèle, **(2)**MPG (Miles per Gallon, consommation d'essence du véhicule), **(3)**Nombre de cylindres, **(4)**Cylindrée (La cylindrée est le volume balayé par le déplacement d'une pièce mobile dans une chambre hermétiquement close pour un mouvement unitaire.), **(5)**Puissance en chevaux, **(6)**Poids, **(7)**Accélération (nombre de secondes pour passer de 0 à 60 MPH), **(8)**Année de production du modèle, **(9)**Origine du modèle, **(10)**Performance

Une fois entraîné, nous avons utilisé ce modèle sur notre échantillon de données principal. Nous sommes parvenus à avoir un score de 0.88 sur notre modèle de prédiction après plusieurs itérations. En effet, pour améliorer l'algorithme de notre modèle, nous avons essayé d'ajouter ou/et supprimer certains paramètres de notre échantillon afin de voir si notre score de prédiction pourrait augmenter. Par exemple, si nous laissons la colonne "Année de production du modèle", notre score descendait à 0.82. Ainsi, nous voulions prédire les voitures "peu efficaces" selon nos critères. Donc nous avons supprimé les colonnes qui diminuent notre score de prédiction : Nom du modèle, l'origine du modèle, l'année de production du modèle et le nombre de cylindres.

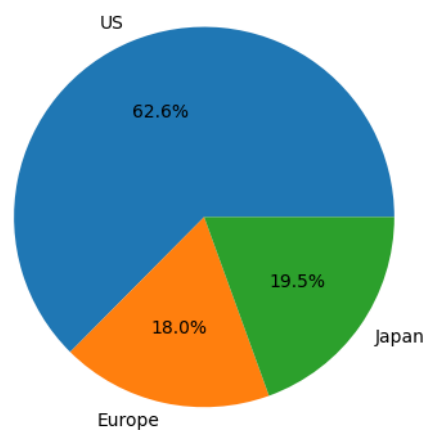


Et pour finir nous avons récolté les voitures que l'algorithme a repéré comme étant "peu efficaces" et nous les avons regroupées par région.



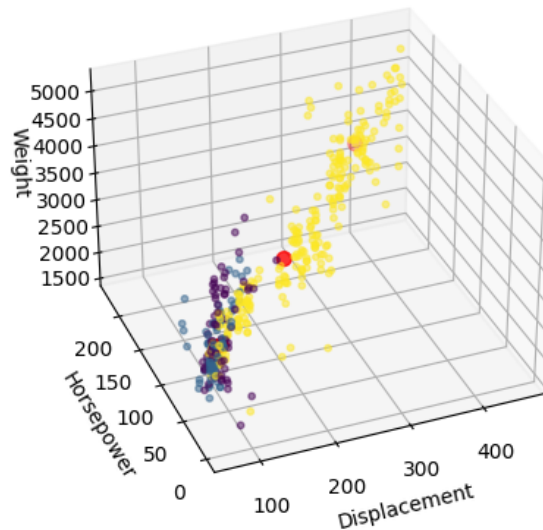
Ceci s'explique notamment car dans nos critères d'efficacité nous ne prenons pas en compte le poids de la voiture, or, beaucoup de voitures produites aux Etats Unis sont des gros véhicules ce qui explique la grosse cylindrée comparé à la faible accélération de la voiture.

Comparative of the number of produced cars in different regions

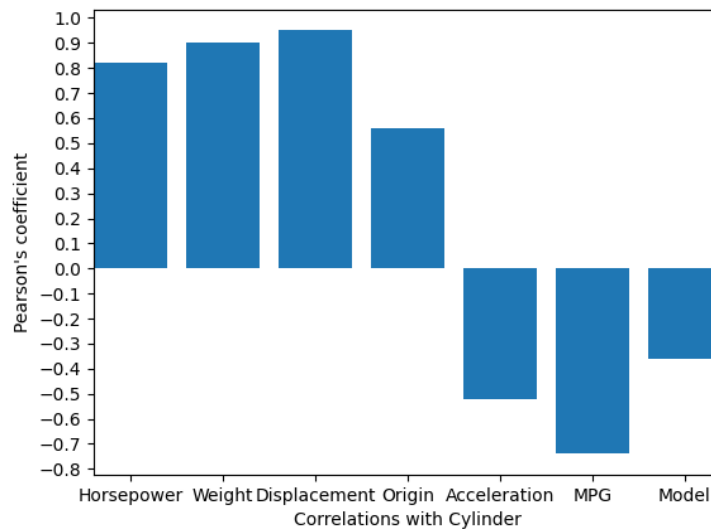


De plus, nous avons essayé de voir s'il était possible de déterminer l'origine de la voiture en utilisant ses principales caractéristiques par le biais d'un algorithme de clusterisation, or il est clair qu'en fonction de ces paramètres, nous avons fini par déterminer que cela était impossible de différencier un véhicule japonais d'un européen, la corrélation entre eux étant trop forte, même s'il est tout de même possible de les différencier des véhicules américain (ici en jaune sur le schéma)

Displacement vs Horsepower vs Weight



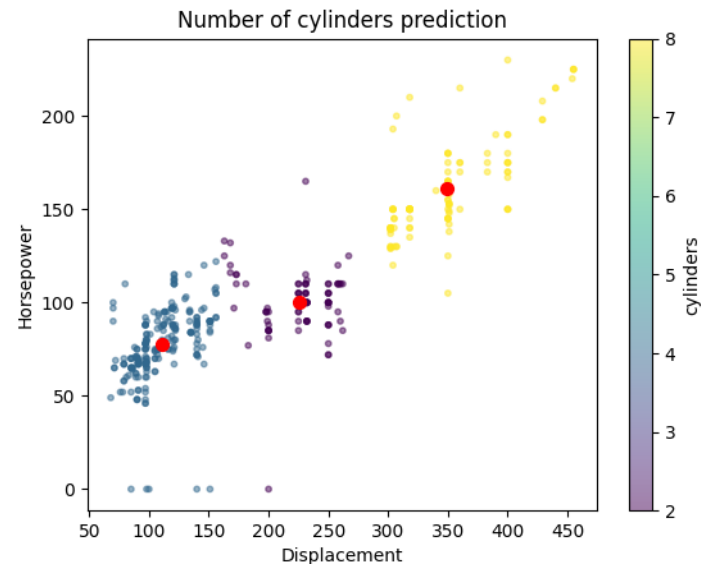
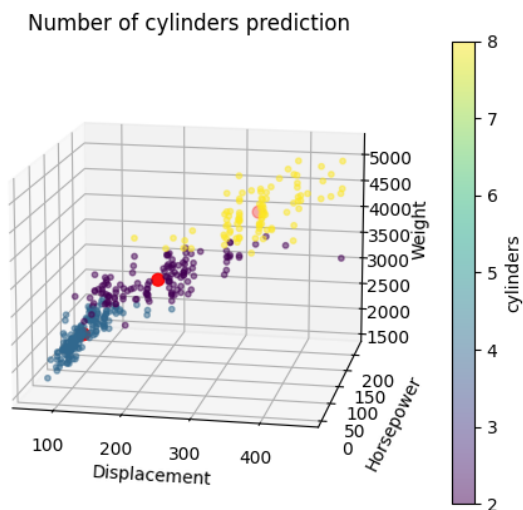
Toutefois, ces mêmes données nous permettent tout de même de déterminer une autre caractéristique avec une bonne fiabilité : le nombre de cylindres. Afin d'utiliser uniquement les caractéristiques ayant le plus d'influence sur ce dernier, on en calcule le coefficient de Pearson.



Ainsi, on voit bien que le poids la puissance et le couple sont les caractéristiques à choisir en priorité. A noter la présence de quelques voitures aux cylindres atypiques (3 et 5), que nous ignorons ainsi, étant des valeurs aberrantes. Nous avons donc 3 cylindres, donc 3 clusters à former.

Donc en appliquant toujours le même algorithme de classification, on les obtient, avec leur centroïdes, c'est à dire la moyenne des points appartenants au cluster (étant denotes en rouge ci-après). En attribuant ainsi chaque cluster a un nombre de cylindres, on peut ainsi en faire la prédiction.

Après l'utilisation de plusieurs paramètres différents, nous avons obtenu un précision de 76.35% (valeurs aberrantes comprises) en se délestant du poids (comparaison entre les 2 résultats ci-dessous). Ce qui est plutôt significatif pour une méthode non supervisée.



/ Conclusion

Pour revenir à notre question de base, les caractéristiques d'une voiture sont bel et bien influencées par le pays d'où elle provient.

En effet grâce à nos analyses nous avons pu constater que plus une voiture est puissante, plus sa cylindrée sera grande, ce qui est notamment le cas des constructeurs américains, qui ont plus tendance à produire des voitures lourdes type 4x4 avec une faible accélération mais une grosse cylindrée pour compenser le poids du véhicule. Pour les constructeurs européens et japonais on constate un meilleur équilibre entre poids et accélération avec des véhicules plus citadins.