Quelles sont les émissions de CO2 d’une voiture ? Une image contenant noir, obscurité

Description générée automatiquement

Table des matières

[Introduction 3](#_Toc148618061)

[Partie I : Exploration des données 3](#_Toc148618062)

[Etape II : Pre-processing 9](#_Toc148618063)

[Sélection des variables 10](#_Toc148618064)

[Etude des variables numériques 13](#_Toc148618065)

[Etude des variables catégorielles 16](#_Toc148618066)

[Etape III : Modélisation 19](#_Toc148618067)

[Modèles simples 20](#_Toc148618068)

[La régression linéaire 20](#_Toc148618069)

[L’arbre de décision 21](#_Toc148618070)

[Le Support à Vecteur de Machine 21](#_Toc148618071)

[Modèles avancés 22](#_Toc148618072)

[Random Forest (modèle de base entraîné : arbre de décision) 22](#_Toc148618073)

[Boosting 22](#_Toc148618074)

[Voting Regressor 23](#_Toc148618075)

[Stacking Regressor 26](#_Toc148618076)

[Conclusion 26](#_Toc148618077)

# 

# Introduction

Dans cet exposé, nous allons nous exercer à prédire les émissions de CO2 d’un véhicule. L’idée sous-jacente est d’identifier les véhicules polluants.   
  
En montrant aux autorités compétentes quels sont les véhicules polluants, celles-ci peuvent mettre en place des mesures visant à réduire leur production.   
Les bonus/malus à l’achat sont un type d’action visant clairement à favoriser la production de certaines voitures plutôt que d’autres.   
  
En amont de toute prise de décision se trouve l’étape d’analyse qui a pour but de prouver (par des données réelles) une hypothèse.   
Il est commun de penser que les véhicules roulant au Diesel sont les plus polluants.   
Il s’agira ici de ne pas s’appuyer sur des ouï-dire mais sur des données factuelles.

Nous allons nous intéresser à deux jeux de données :

* Les données du gouvernement français sur les véhicules commercialisés en France sur une année civile.
* Les données de l’European Environment Agency sur les véhicules commercialisés en Europe sur une année civile.

Avant d’analyser des données, il convient d’en vérifier la fiabilité car les résultats de modélisation et la conclusion des études en dépendent.   
Les deux jeux de données cités plus haut sont issus d’institutions gouvernementales. Nous allons donc considérer chaque donnée retranscrite comme étant fiable.

# Partie I : Exploration des données

Les deux jeux de données comportent des variables différentes.  
Le dataset « Europe » contient 38 variables initiales contre 26 pour le dataset « France ».   
La variable cible est bien évidemment la variable qui décrit la quantité d’émission de CO2 pour une voiture en gramme par kilomètre.

Voici une exploration succincte pour évaluer la qualité des jeux de données :

* Le dataset « France » sur l’année 2013 sur-représente les véhicules DIESEL (=GO) et la catégorie de véhicule MINIBUS

Une image contenant texte, capture d’écran, affichage, nombre

Description générée automatiquementUne image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

On retrouve dans la catégorie Minibus, les modèles Jumper, Jumpy, Ducato, Scudo, Trafic, etc…

La sur-représentation est identique sur les données publiées en 2014 et en 2012. Cette sur-représentation de certaines catégories aura un impact négatif sur le fonctionnement de l’algorithme. En effet, l’algorithme se basera sur des échantillons restreints de données pour les catégories minoritaire, ce qui peut entraîner un modèle plus approximatif.

Voici un graphique qui représente l’impact moyen des émissions de CO2 en g/km des différentes catégories de véhicules :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement  
Ce seul critère semble dégager une tendance mais ne se suffit pas à lui-même pour expliquer quelle catégorie de véhicule émet le plus de CO2.

Concernant la gamme, une tendance semble aussi se dessiner :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

La modélisation d’une heatmap sur le dataset « Europe » (données 2020) permet de tirer quelques conclusions sur les relations linéaires entre variables (explicatives et cible).

* Voici les corrélations entre variables explicatives numériques :
  + La masse de la voiture est bien corrélée avec l’empattement (Wheel base) et le gabarit du véhicule (Axle width). Le coefficient positif de corrélation est de 0.8.
  + La masse de la voiture est aussi corrélée avec la puissance du moteur/batterie du véhicule (coefficient positif de 0.7)
  + La masse du véhicule a un impact sur l’énergie électrique consommée par kilomètre (coefficient positif de 0.64)
  + Plus la masse augmente, moins la réduction de Co2 induite par l’intégration de nouvelles technologies dans le véhicule est importante (coefficient négatif d’environ 0.2).
  + L’empattement de la voiture augmente l’autonomie des voitures électrique (coefficient positif de 0.15). Ce résultat est surprenant mais on peut comprendre que plus le gabarit de la voiture est important, plus la batterie qui y est introduite est importante aussi (et donc, maximise l’autonomie du véhicule).
  + L’empattement de la voiture augmente la consommation d’essence pour les voitures thermiques (ce qui est logique, coefficient positif de 0.14).
  + Concernant les véhicules électriques, plus la puissance est élevée, plus l’autonomie augmente. Ce résultat est surprenant mais on peut comprendre que les batteries puissantes sont destinées au plus gros véhicule. Les gros véhicules ont pour objectif de faire de plus longues distances. De ce point de vue, il est logique que l’autonomie pour les batteries puissantes soient plus importantes (coefficient positif = 0.22).
  + Quant aux véhicules thermiques, plus la puissance du moteur est élevée, plus la consommation au km est importante (coefficient positif = 0.36).
  + Plus la batterie est puissante, plus la consommation d’énergie en Wh est importante (coef positif 0.38).
  + La consommation d’essence est plus importante pour les véhicules dotés d’un moteur puissant et d’un gabarit important.
  + Les réductions de Co2 suite aux innovations technologiques sont plus importantes pour les véhicules électriques avec une plus faible autonomie. Comme expliqué précédemment, les réductions sont plus importantes lorsque la voiture a un gabarit peu volumineux. Lorsque l’autonomie est grande, le gabarit de la voiture est plus important.
* Voici les corrélations entre variables explicatives et la variable cible :
  + Plus la consommation d’essence augmente, plus les émissions augmentent (coefficient positif de corrélation très fort de 0.87)
  + Plus l’autonomie des véhicules électriques est importante, moins il y a d’émissions de Co2 (coefficient négatif de 0.82).  
    Lorsque les autonomies sont faibles, il est fort probable que le véhicule soit hybride. Donc, le lien de corrélation est exact dans la mesure où les véhicule hybrides consomment tout de même de l’essence.
  + Si la puissance motrice augmente, les émissions augmentent (coefficient positif de 0.28).
  + Si la consommation électrique en Wh/km augmente, les émissions de Co2 aussi (coefficient positif de 0.17).   
    Lorsqu’un véhicule hybride n’a plus d’énergie dans sa batterie, elle va se propulser à l’essence ou au diesel. Plus un véhicule consommera d’énergie électrique, plus elle aura tendance à s’appuyer sur son moteur à combustion (donc engendrera des émissions de CO2).

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

* Voici la corrélation entre les différentes modalités de la variable explicative catégorielle « type de carburant » et la variable explicative numérique « Consommation de carburant/km » :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement  
On observe que, selon la typologie de carburant, la consommation moyenne de carburant (au 100km) diffère. Pour les véhicules électriques, on parle de consommation de Wh (et non de carburant).

* Voici le lien de corrélation entre les différentes modalités de la variable type de carburant et l’impact sur les émissions de CO2 :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement  
Le type de carburant à une incidence directe sur les émissions de Co2. Selon le type, les rejets de pollution sont différents.

Les émissions de CO2 par pays (Europe) diffèrent également :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

La moyenne de rejet Co2 est différente selon les pays. La Norvège est le pays qui rejette le moins de Co2 avec son parc automobile.

Ce n’est pas le pays qui a un impact sur la variable cible mais les voitures présentes dans ces pays. Les caractéristiques des voitures sont différentes entre pays, notamment la Norvège par rapport au reste de l’Europe.   
La culture de la voiture est donc différente entre pays d’Europe. Les habitudes de consommation se distinguent.

Enfin, voici la représentation des différentes catégories de la variable « type de carburant » du jeu de donnée « Europe » :   
Une image contenant texte, capture d’écran, affichage, nombre

Description générée automatiquement

Les différentes catégories y sont plutôt bien représentées. Les quelques catégories mineures sont marginales.

Voici une boîte à moustache qui représente la distribution de la variable cible sur le jeu de données « Europe » :

Une image contenant ligne, diagramme, capture d’écran, Tracé

Description générée automatiquement  
Au-delà de 500, les valeurs paraissent aberrantes. En dessous de 50, les valeurs sont extrêmes mais non aberrantes (un véhicule électrique n’émet pas de CO2).

Si on compare la boîte à moustache avec le jeu de données « France », on peut faire la même observation :  
 Une image contenant diagramme, ligne, capture d’écran, Rectangle

Description générée automatiquement

# Etape II : Pre-processing

A ce stade, la question se pose de savoir sur quel jeu de données nous allons baser notre analyse :

Le Dataset France sur-représente les catégories MINIBUS et DIESEL (GO). Il faut envisager de supprimer une partie des données sur-représentées pour ré-équilibrer le jeu.   
Avec une perte de données d’entrées conséquente, il faut alors envisager l’intégration de plusieurs années dans le jeu pour avoir un nombre d’entrée significatif car la quantité de données joue un rôle sur la qualité de prédiction de l’algorithme.

Le Dataset Europe ne fournit pas d’information sur les catégories de véhicules comme dans le dataset France. C’est dommage car c’est une variable intéressante à prendre en compte dans l’analyse.

Le Dataset Europe sur-représente les catégories de véhicules essence par rapport aux autres mais cette sur-représentation est moins marquée que sur le Dataset France.

Il faut envisager de supprimer une partie des données essence pour qu’elle soit à peu près équivalente à la 2ème modalité. Même avec la suppression de données, le jeu comptera certainement un très grand nombre d’entrée (> 1million).   
  
Il est important qu’une modalité de variable ne soit pas sur-représentée. Sinon, l’algorithme sera bien entrainé sur la modalité sur-représentée mais pas sur les autres, ce qui entrainera beaucoup d’erreurs de prédictions sur les modalités minoritaires.

En conclusion, le jeu de données Europe présente l’avantage d’avoir un grand nombre d’entrées et une répartition des données entre les différentes modalités plus équilibrée.   
  
Dans l’ensemble, les 2 datasets présentent les mêmes variables :

* Gabarit du véhicule
* Puissance motrice
* Type de carburant
* Emission de CO2

Etant donné nos limites computationnelles de nos équipements, nous allons poursuivre l’étude avec les données du gouvernement Français.   
Avec des conditions d’étude optimales, il nous aurait paru plus judicieux de travailler sur un grand nombre de données (plutôt que moins).

Afin d’avoir un jeu de données avec un nombre d’entrées intéressant, nous allons récupérer les données des années 2012, 2013 et 2014.

Dans un premier temps, il a fallu récolter les jeux de données, harmoniser les noms de leur colonne (non identique d’une année à l’autre) et harmoniser les types de chaque Series (colonne pandas).

## Sélection des variables

Une fois le jeu de données prêt à être transformer, il convient de procéder à une étude sur les variables à conserver ou à supprimer du jeu.   
En appliquant un raisonnement classique, nous pouvons déjà prendre des décisions :

Par exemple : La marque va-t-elle déterminer si le véhicule est polluant ? Non. Cela paraît logique. Chaque marque présente un grand nombre de véhicule aux caractéristiques différentes.   
Nous pourrions répondre oui à cette question uniquement dans le cas où chaque marque fabriquerait un seul type de véhicule (où tous les modèles auraient des caractéristiques identiques).   
  
Nous sélectionnerons des variables avec un nombre limité de modalités et un taux de valeur manquantes faibles.   
  
Vous trouverez ci-après un tableau justifiant les choix de sélection/non sélection de chacune d’entre elles :

|  |  |  |
| --- | --- | --- |
| Variable | Pertinence | Justification |
| Marque | Non Pertinente | La marque ne détermine pas si un véhicule est polluant car elle ne conditionne pas les caractéristiques techniques du véhicule. Il y a plus de 10 modalités, ce qui n’apporterait que de la confusion à l’analyse. |
| Modèle dossier | Non Pertinente | Le modèle ne détermine pas si un véhicule est polluant. Le modèle ne conditionne pas les caractéristiques techniques du véhicule. Il y a plus de 10 modalités, ce qui n’apporterait que de la confusion à l’analyse. |
| Modèle UTAC | Non Pertinente | Le modèle ne détermine pas si un véhicule est polluant (ne conditionne pas les caractéristiques techniques du véhicule). Il y a plus de 10 modalités, ce qui n’apporterait que de la confusion à l’analyse. |
| Désignation commerciale | Non Pertinente | La désignation ne détermine pas si un véhicule est polluant (ne conditionne pas les caractéristiques techniques du véhicule). Il y a plus de 10 modalités, ce qui n’apporterait que de la confusion à l’analyse. |
| CNIT | Non Pertinente | Le code est un numéro d’identification unique pour chaque véhicule. Il y a autant de modalités que d’entrées. Cette information n’est pas pertinente. |
| Type Variante Version (TVV) | Non Pertinente | La variante ne détermine pas si un véhicule est polluant (ne conditionne pas les caractéristiques techniques du véhicule). Il y a 40k modalités. |
| Carburant | Pertinente | Le carburant est une des caractéristiques techniques du véhicule. |
| Hybride | Pertinente | La fonctionnalité d’hybridation est une caractéristique technique. |
| Puissance administrative | Pertinente | Caractéristique technique du véhicule. |
| Puissance maximale (kW) | Pertinente | Caractéristique technique du véhicule. |
| Boîte de vitesse | Non Pertinente | C’est une caractéristique technique du véhicule mais elle présente un nombre trop important de modalités (20). |
| Consommation urbaine (l/100km) | Pertinente | Données sur la performance (de consommation) du véhicule. |
| Consommation extra-urbaine (l/100km) | Pertinente | Données sur la performance (de consommation) du véhicule. |
| Consommation mixte (l/100km) | Pertinente | Données sur la performance (de consommation) du véhicule. |
| CO2 (g/km) | Pertinente | Variable cible, celle qu’on souhaitera prédire par la suite. |
| CO type I (g/km) | Pertinente | Données sur la performance (d’émission de CO1) du véhicule. |
| HC (g/km) | Non Pertinente | Données sur la performance du véhicule. Mais 80% des données sont manquantes, ce qui rend la colonne inutilisable. |
| NOX (g/km) | Pertinente | Données sur la performance du véhicule. |
| HC+NOX (g/km) | Non Pertinente | Données sur la performance du véhicule. Mais 20% des données sont manquantes. Il vaut mieux écarter cette colonne de notre analyse. |
| Particules (g/km) | Pertinente | Données sur la performance du véhicule. |
| masse vide euro min (kg) | Pertinente | Caractéristique technique du véhicule. |
| masse vide euro max (kg) | Pertinente | Caractéristique technique du véhicule. |
| Champ V9 | Non Pertinente | Nombre de modalités trop important (40) |
| Date de mise à jour | Non Pertinente | La temporalité n’est pas à prendre en compte dans notre cas d’étude. |
| Carrosserie | Pertinente | Type de véhicule (caractéristique technique générale). |
| gamme | Pertinente | Classe du véhicule (caractéristique technique générale). |
| Unnamed: 26 | Non Pertinente | Variable vide |
| Unnamed: 27 | Non Pertinente | Variable vide |
| Unnamed: 28 | Non Pertinente | Variable vide |
| Unnamed: 29 | Non Pertinente | Variable vide |

Après ce retraitement de colonnes, nous obtenons un dataframe sur lesquel nous allons approfondir notre analyse.   
En tout, il y a environ 140 000 entrées. Après suppression des doublons, nous n’obtenons plus que 15 000 entrées.

En ce qui concerne les valeurs manquantes, nous avons :

* 15% pour la variable Particules
* 0.9% pour les variables NOX et CO type I
* <0.5% pour les variables CO2 et Consommations (urbaines, extra-urbaines et mixtes)

En observant la relation des Particules avec la cible CO2, on constate qu’elle n’est pas pertinente pour notre modèle (car elle ne possède pas de relation linéaire avec celle-ci) :

Une image contenant texte, capture d’écran, diagramme, ligne

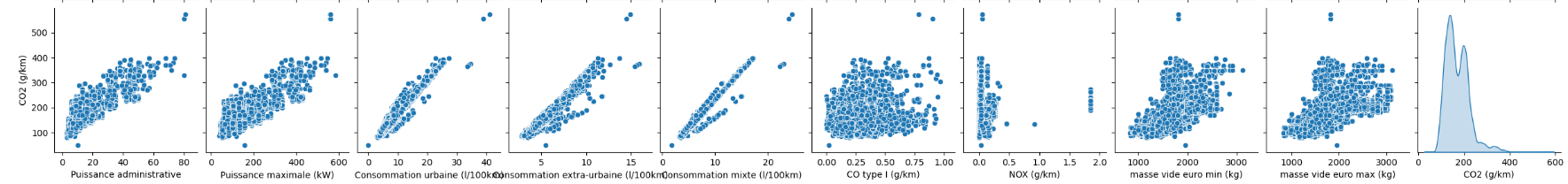
Description générée automatiquement

Les variables CO2 et Consommations qui contiennent des valeurs manquantes ont été supprimées (seul 0.2% des données ont été « supprimées).

Les lignes dont les données sont manquantes pour les variables NOX et CO type I (env 1%) ont été remplacées par leur médiane.

### Etude des variables numériques

Pour approfondir et conforter notre première vague de sélection de variable, nous avons réalisé un pairplot.

Figure : Analyse de la corrélation (linéaire) entre les variables (explicatives) numériques et la variable cible CO2

Nous constatons par les données de notre jeu la pertinence des variables *a priori* sélectionnées.

Ici, les variables CO type I et NOX ne sont pas corrélées avec la variable CO2. Il faut donc les supprimer.   
par ailleurs, la consommation de carburant à une corrélation quasi identique à la variable cible.   
  
Pour se faire une idée précise des interactions entre variables explicatives, nous avons réalisé une heatmap :   
Une image contenant texte, capture d’écran, nombre, Caractère coloré

Description générée automatiquement

Les variables numériques suivantes sont très corrélées entre elles :

* Puissances
* Consommations
* Masse à vide

La colinéarité des variables explicatives peut entraîner des résultats moins performants.

Ayant l’objectif de modéliser un algorithme aux performances optimales, il faut garder ces éléments en tête pour revenir dessus par la suite si besoin.

Ayant une base de travail cohérente pour les variables numériques, traitons les outliers présents sur notre jeu de données.

Le seuil de valeur extrême observé pour la variable CO2 concerne toutes les valeurs supérieures à 305.5 g de CO2 émis par km (représentant 2% des données) :

Une image contenant capture d’écran, texte, diagramme, ligne

Description générée automatiquement

Les valeurs extrêmes ne semblent pas être aberrantes, sauf pour les valeurs d’émissions supérieure à 500g/km. Nous obtenons donc 6 valeurs qui semblent aberrantes. MAIS, ces valeurs ne concernent qu'un modèle de voiture : l'Aston Martin one-77.   
Après quelques recherches en ligne, ces émissions de CO2 sont bien estimées à plus de 500g/km.

Nous ne sommes pas en présence de valeurs aberrantes mais bien de valeurs extrêmes.

Nous avons généré des boxplots concernant les variables numériques :

Une image contenant diagramme, texte, Rectangle, Plan

Description générée automatiquement

Toutes possèdent des valeurs extrêmes.   
Cela dit, pour toutes les valeurs extrêmes, seule la catégorie LUXE (de la variable gamme) y est représentée (avec des modèles très sportif).   
Nous pouvons en juger que toutes les valeurs sont extrêmes mais non aberrantes.

### Etude des variables catégorielles

De la même manière que nous avons vérifié la pertinence des variables numérique à l’aide de graphiques, il convient de vérifier la pertinence des variables dans notre modèle.

Premièrement, dans un souci d’harmonisation, toutes les variables ont été regroupée (pour réduire le nombre de catégories).

**Carrosserie :**

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement   
En étudiant le graphique, on peut réduire le nombre de classes en regroupant certaines d'entre elles :

* SPORT = COUPE + CABRIOLET
* VOLUMINEUSE = MONOSPACE + BREAK + COMBISPACE
* COMPACTE = BERLINE + MONOSPACE COMPACT + MINISPACE

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Gamme**

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

À la vue du graphique et du nombre d'entrée pour chacune des classes de Gamme, on peut regrouper les valeurs comme suit :

* LUXE = LUXE
* MOYENNE = MOY-INFER + MOY-SUPER + SUPERIEURE + SPORT
* INFERIEURE = INFERIEURE + ECONOMIQUE

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Carburant**

**Une image contenant texte, capture d’écran, affichage, nombre

Description générée automatiquement**

Ici, nous allons préserver tous les combustibles propres à eux-même.   
Dès qu’il y a une combinaison de technologies, nous allons la regrouper dans une catégorie spécifique.

Voici les définitions des différentes catégories :

* GO = DIESEL
* ES = ESSENCE
* EH = ESSENCE-ELECTRICITE (hybride non rechargeable)
* ES/GP = ESSENCE-GPL (Gaz pétrole liquéfié - butane ou propane)
* FE = SUPERETHANOL
* ES/GN = ESSENCE-GAZ NATUREL
* GH = DIESEL-ELECTRICITE (hybride non rechargeable)
* GN = GAZ NATUREL
* GL = DIESEL-ELECTRICITE (hybride rechargeable)

Voici le regroupement réalisé :

* GO = DIESEL
* ES = ESSENCE
* FE = SUPERETHANOL
* GN = GAZ NATUREL
* GO+ = GH, GL (DIESEL-ELECTRICITE)
* ES+ = EH, ES/GP, ES/GN

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

De plus, la p-value du test de statistique ANOVA de la variable carburant avec la variable cible est de 4.951873e-91. Cette variable est donc étroitement liée à la variable cible.

Pour rendre les variables catégorielles exploitables dans les différents modèles, nous devons encoder celles-ci.   
Un encodage ONE-HOT a été réalisé pour les variables Carburant, Gamme et Carrosserie, et Hybride.

# Etape III : Modélisation

Après ces diverses étapes de nettoyage, le but est maintenant de trouver un modèle approprié qui permettra de prédire, avec le plus d’exactitude possible, les émissions de CO2 d’une voiture.

Pour que le modèle soit jugé, nous avons partitionné les données en 2 sets (1 set d’entraînement, 1 set de test).

Le modèle sera entraîné sur le set dédié, puis évalué à partir du set de test.   
Ce set de test est important pour juger de la performance car nous souhaitons savoir si les prédictions sont bonnes lorsque les données sont inconnues du modèle.

Nous sommes dans un cas d’apprentissage supervisé (les labels, càd les émissions produites par les voitures, du jeu de données sont disponibles).

Le label est quantitatif. Nous allons donc appliquer des algorithmes qui conviennent, tel que :

* La régression linéaire
* L’arbre de décision
* Le support à vecteur de machine

Dans un premier temps, nous allons faire une modélisation simple. Puis, nous élaborerons des modèles plus complexes visant à optimiser les performances.

## Modèles simples

A des fins de comparaisons, nous avons sauvegardé un set avec des données standardisées et un set sans standardisation.

### La régression linéaire

La régression linéaire, entraînée avec données standardisées et non standardisées donnent des résultats identiques.

Ces résultats sont très bons puisque la prédiction moyenne est de 99,28% (pour le set d’entraînement) et 99,35% (pour le set de test.

La différence quasi-nulle entre le set d’entraînement et set de test démontre aussi que nous ne sommes pas en présence d’un problème de sur-apprentissage.

La régression linéaire est une modélisation très adaptée à notre problème. Pour étayer mes propos, regardons la droite de régression (prédiction) et les point réels observés (en orange sur le graphique ci-dessous) :

Une image contenant texte, capture d’écran, ligne, Tracé

Description générée automatiquement

Enfin, l’hypothèse d’homoscédasticité est vérifiée puisque la variation des prédictions par rapport au points réels est très faible (les points sont concentrés autour de 0).

Une image contenant texte, capture d’écran, ligne, diagramme

Description générée automatiquement

### L’arbre de décision

Comparativement à la régression linéaire, la modélisation du problème au travers d’un arbre de décision permet d’obtenir de meilleur résultat lorsque les données sont standardisées :

* 0.9972 sur set de test et données non standardisées
* 0.9979 sur set de test et données standardisées.

La différence n’est toutefois pas significative.

Concernant les résultats obtenus sur set d’entraînement et set de test sont plus significatif que ceux obtenus lors de la régression linéaire. La différence est toutefois négligeable et ne permet pas de conclure que le modèle est over-fitté.

### Le Support à Vecteur de Machine

Enfin le modèle SVM sans standardisation est mauvais.   
Le score sur set d’entraînement et de test est de 0.55.   
L’erreur quadratique moyenne est de 35 points. A chaque prédiction, nous sommes donc loin des valeurs réelles observées.

Par contre, le modèle standardisé propose un score intéressant (score de 0.92, RMSE de 14).

Cette modélisation laisse présager qu’une optimisation des hyperparamètres permettra d’atteindre un très bon score.

L’entraînement du modèle SVM dure environ 40 secondes.   
En appliquant la fonction GridSearchCV pour chercher les meilleurs hyperparamètres, le temps de calcul sera conséquent (comparativement aux autres modèles entraînés jusqu’ici).

Dans une démarche d’exhaustivité, nous allons chercher les meilleurs paramètres du modèle SVM pour le comparer aux autres.

## Modèles avancés

Etant donné les scores entre données standardisées et non standardisées, nous allons poursuivre notre analyse avec données standardisées.   
  
Pour approfondir notre concept de régression linéaire, nous pourrions nous intéresser aux régressions Ridge, Lasso ou ElasticNet (combinaisons flexibles des deux premiers types de régression).

Toutefois, ces types de régression sont utilisées lorsque nous faisons face à un problème de sur-apprentissage.   
Comme nous l’avons vu avec le modèle de régression simple, nous ne sommes pas dans un cas de sur-apprentissage. Il n’est donc pas nécessaire de se pencher sur ces variantes.

Nous allons entrainer :

* Une forêt aléatoire (cas particulier de Bootstrapping/Bagging avec un arbre de décision comme modèle de référence)
* Un Boosting (base régression linéaire et arbre de décision)
* Un Voting Regressor (prenant en compte les modèles de régression, arbre de décision et SVM)
* Un Stacking Regressor (en prenant en compte les best\_params de chaque modèle référence).

### Random Forest (modèle de base entraîné : arbre de décision)

Le résultat sur set de test est meilleur avec le modèle Random Forest que sur l'arbre de décision "simple" :

* Score = 0.99863 pour Random Forest
* Score = 0.99792 pour l'arbre de décision

C'est léger mais on observe que le régresseur Random Forest réduit le sur-apprentissage de l'arbre de décision "simple" :

* Différence de score entre train et test pour le Random Forest : 0.001
* Différence de score entre train et test pour l’arbre de décision : 0.002

Le modèle Random Forest est donc meilleur à tout point de vue par rapport à l'arbre de décision.

L'échelle de comparaison est infime parce que les 2 modèles obtiennent la quasi-justesse de prédictions.

### Boosting

La modèle AdaBoost entraîné avec une base d’arbre de décision et de régression linéaire n'est pas une alternative convaincante.

Le Boosting, dans notre problème, n'est pas la solution adaptée :

* Score sur set de test pour la régression linéaire : 0.9935
* Score sur set de test par Boosting (base régression linéaire) : 0.9880
* Score sur set de test pour l’arbre de décision : 0.9963
* Score sur set de test par Boosting (base arbre de décision) : 0.9506

### Voting Regressor

Avant d’élaborer un Voting Regressor, il convient de trouver les meilleurs paramètres à entraîner dans ce méta-classifieur pour obtenir les meilleures performances.

Il est possible d’intégrer la recherche des meilleurs paramètres directement dans la fonction du Voting Regressor mais par soucis de clarté et de détail, nous allons faire appel à GridSearchCV. Nous commenterons ensuite les résultats.

#### GridSearchCV sur Forêt Aléatoire

Les paramètres ont été définis comme suit :

params = {'n\_estimators': [20, 100, 200],

          'max\_depth': [3, 6]}

Pour rappel, voici les paramètres par défaut de la forêt aléatoire lorsque les paramètres ne sont pas précisés :

params :  
'n\_estimators'= 100,

'max\_depth' = None

Nous avons défini une cross-validation sur 3 partitions (du set d’entrainement).

Comparons maintenant les résultats obtenus sur le set d’entrainement :

* Score Forêt Aléatoire par défaut : 0.9997
* Score best\_params\_ Forêt Aléatoire : 0.9895

Le score lorsque l’on réalise un GridSearchCV est plus faible que le modèle par défaut.   
En voici les raisons :

* Le modèle par défaut ne propose pas de profondeur maximum pour chaque arbre. Le partitionnement peut être plus pointu et permet d’obtenir de meilleurs segment (une meilleure entropie)

Cette profondeur maximum conditionne la performance du modèle. Et pour sûr, les best\_params\_ trouvés sont ceux-ci :

params :  
'n\_estimators'= 200,

'max\_depth' = 6

Cela indique que le modèle a eu besoin de plus d’itérations d’arbre pour obtenir une meilleure performance (comparativement au modèle de base à 100 itérations). Et même avec ces 200 itérations, la profondeur maximale a été choisie.

On observe d’ailleurs que le critère de profondeur a plus d’impact sur la performance que le nombre d’itérations :

|  |  |  |
| --- | --- | --- |
| params | mean\_test\_score | rank\_test\_score |
| {'max\_depth': 3, 'n\_estimators': 20} | 0.942872 | 4 |
| {'max\_depth': 3, 'n\_estimators': 100} | 0.942170 | 6 |
| {'max\_depth': 3, 'n\_estimators': 200} | 0.942581 | 5 |
| {'max\_depth': 6, 'n\_estimators': 20} | 0.989374 | 3 |
| {'max\_depth': 6, 'n\_estimators': 100} | 0.989405 | 2 |
| {'max\_depth': 6, 'n\_estimators': 200} | 0.989552 | 1 |

Pourquoi limiter la profondeur des arbres ?   
Dans le cas d’arbres de décisions, nous faisons très souvent face à un modèle qui s’adapte parfaitement aux données d’entraînement (overfitting).

Limiter sa profondeur, c’est limiter son adaptation aux données, ce qui permet de prévenir le sur-apprentissage.

Cependant, dans le cas de forêts aléatoires, nous mettons en œuvre le Bagging (càd un tirage aléatoire pour chaque arbre d’un nombres x d’observations ainsi qu’un nombre n de variables présentes dans l’échantillon initial).   
Cette méthode (sous-cas de Bagging) permet de prévenir l’overfitting.

Dans ce cas, on peut en conclure que la restriction de profondeur de l’arbre pour les forêts aléatoires n’est pas nécessaire.

*Nota* : Toutes choses n’étant pas égales par ailleurs car nous n’avons pas réalisé une cross-validation sur le modèle entraîné par défaut.

Toujours est-il que pour le Voting Regressor, nous utiliserons le modèle de forêt aléatoire par défaut qui nous a largement satisfait.

#### GridSearchCV sur SVM

Les paramètres ont été définis comme suit :

params = {'C': [0.1,1, 10],

          'gamma': [0.001, 0.01, 0.5],

          'kernel': ['linear', 'poly', 'rbf']}

Pour rappel, voici les paramètres par défaut de la forêt aléatoire lorsque les paramètres ne sont pas précisés :

params :  
'C' = 1  
'gamma' = par défaut, calculé selon caractéristiques  
'kernel' = 'rbf'

Nous avons défini une cross-validation sur 4 partitions (du set d’entrainement).

*Nota :* L’option ‘poly’ du kernel a été écarté à cause d’un temps de calcul trop onéreux.

Comparons maintenant les résultats obtenus sur le set d’entrainement :

* Score SVM par défaut : 0.9347
* Score best\_params\_ SVM : 0.9930

Le jeu en valait la chandelle. Les résultats obtenus par grille de recherche sont bien meilleurs qu’en utilisant le modèle définit par défaut.

Avant de figer le choix best\_params\_ pour avancer dans notre analyse, jetons un œil au tableau des résultats :

|  |  |  |  |
| --- | --- | --- | --- |
| params | mean\_test\_score | std\_test\_score | rank |
| {'C': 0.1, 'gamma': 0.001, 'kernel': 'linear'} | 0.990649 | 0.003021 | 2 |
| {'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'} | 0.236469 | 0.006500 | 18 |
| {'C': 0.1, 'gamma': 0.01, 'kernel': 'linear'} | 0.990649 | 0.003021 | 2 |
| {'C': 0.1, 'gamma': 0.01, 'kernel': 'rbf'} | 0.766665 | 0.014365 | 15 |
| {'C': 0.1, 'gamma': 0.5, 'kernel': 'linear'} | 0.990649 | 0.003021 | 2 |
| {'C': 0.1, 'gamma': 0.5, 'kernel': 'rbf'} | 0.350852 | 0.015144 | 17 |
| {'C': 1, 'gamma': 0.001, 'kernel': 'linear'} | 0.990478 | 0.003068 | 5 |
| {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'} | 0.925487 | 0.002507 | 13 |
| {'C': 1, 'gamma': 0.01, 'kernel': 'linear'} | 0.990478 | 0.003068 | 5 |
| {'C': 1, 'gamma': 0.01, 'kernel': 'rbf'} | 0.951551 | 0.016392 | 12 |
| {'C': 1, 'gamma': 0.5, 'kernel': 'linear'} | 0.990478 | 0.003068 | 5 |
| {'C': 1, 'gamma': 0.5, 'kernel': 'rbf'} | 0.674538 | 0.021722 | 16 |
| {'C': 10, 'gamma': 0.001, 'kernel': 'linear'} | 0.990395 | 0.003043 | 8 |
| {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'} | 0.993086 | 0.001410 | 1 |
| {'C': 10, 'gamma': 0.01, 'kernel': 'linear'} | 0.990395 | 0.003043 | 8 |
| {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'} | 0.986946 | 0.007782 | 11 |
| {'C': 10, 'gamma': 0.5, 'kernel': 'linear'} | 0.990395 | 0.003043 | 8 |
| {'C': 10, 'gamma': 0.5, 'kernel': 'rbf'} | 0.919524 | 0.016602 | 14 |

On constate que :

* Le kernel ‘linear’ est constant dans ses performances.
* Gamma n’a aucun impact sur les résultats (lorsque kernel = linear)
* Plus C croît, plus la performance se dégrade (lorsque kernel = linear)
* Le kernel ‘rbf’ propose des performances très disparates.

Donc, au vu de la stabilité de performance du kernel ‘linear’, notre choix se porte sur le 2ème meilleur dont les paramètres sont les suivants :

params :  
'C' = 0.1  
'gamma' = par défaut, calculé selon caractéristiques  
'kernel' = 'linear'

Entraînons donc le VotingRegressor.

Sont inclus dans le VotingRegressor la régression linéaire, le random forest et le meilleur modèle SVM.

Voici les résultats obtenus :

* Score sur set d'entrainement : 0.99642
* Score sur set de test : 0.99638

Ce modèle fait mieux qu’une régression linéaire simple mais moins bien qu’un arbre de décision ou une forêt aléatoire.

Notons toutefois que ce modèle fait preuve d’une stabilité déconcertante car les scores entre set d’entrainement et set de test sont quasi-identiques (différence 0,00004).

Lorsque nous entrainons ce modèle en y ajoutant l’arbre de décision, la performance augmente légèrement et l’écart de performance entre set d’entrainement et set de test s’accroît plus que proportionnellement.

### Stacking Regressor

Les résultats obtenus par le Stacking sont impressionnants :

|  |  |  |
| --- | --- | --- |
| Variantes | Train score | Test score |
| Régression linéaire, Random Forest, SVM | 0.99963 | 0.99864 |
| Régression linéaire, Random Forest, SVM, Arbre | 0.99976 | 0.99845 |

On voit encore plus clairement l’effet de l’arbre de décision dans ce modèle. Le fossé entre le set d’entrainement et set de test est bien plus important quand on intègre l’arbre de décision.

# Conclusion

Nous avons analysé les résultats dans leur détail. Procédons maintenant à une analyse synthétique, plus globale de ceux-ci pour orienter notre décision de modèle final.

|  |  |  |  |
| --- | --- | --- | --- |
| Modèles | Entrainement | Test | Delta Entrainement-Test |
| rf\_200 | 0.999817 | 0.998732 | 0.001085 |
| StackingReg1 | 0.999636 | 0.998641 | 0.000994 |
| rf | 0.999792 | 0.998639 | 0.001153 |
| StackingReg | 0.999759 | 0.998444 | 0.001315 |
| VotingReg | 0.997987 | 0.997344 | 0.000643 |
| dt | 0.999997 | 0.997296 | 0.002701 |
| VotingReg1 | 0.996424 | 0.996382 | 0.000042 |
| lr | 0.992898 | 0.993547 | -0.000649 |
| best\_svm | 0.990662 | 0.992284 | -0.001622 |
| ada\_lr | 0.989122 | 0.988851 | 0.000270 |
| ada\_dt | 0.945214 | 0.943162 | 0.002053 |
| svm\_model | 0.934714 | 0.924312 | 0.010402 |

La version RandomForest avec 200 itérations est le meilleur modèle proposé en termes de performance.

Les comparaisons se basent sur des infinitésimales.  
A ces niveaux de comparaisons, tous les modèles sont pertinents (du RandomForest jusqu’au meilleur modèle SVM).

Le choix du modèle est donc assez ouvert.   
S’il y a plusieurs modèles parmi lesquels choisir (car les performances s’équivalent), deux critères de plus peuvent être pris en considération :

* Temps de calcul
* Interprétabilité des résultats retournés

Voici un rapide commentaire pour chaque modèle :

|  |  |  |  |
| --- | --- | --- | --- |
| **Modèles** | **Performance** | **Temps de calcul** | **Interprétabilité** |
| Random Forest | Très bonne | Assez important | Faible\* |
| Stacking | Très bonne | Important | Faible\* |
| Voting | Très bonne | Important | Faible \* |
| Arbre de décision | Bonne ( /!\ overfitting) | Rapide | Forte |
| Régression linéaire | Très bonne | Rapide | Forte |
| Support à Vecteur de Machine | Très bonne | Très important | Faible \* |

*\*Les modèles ne proposent pas de récupérer l’ordre d’importance des features quant à leur impact sur la sortie du label (contrairement à l’arbre de décision, la régression et le randomforest). Il faudrait utiliser des méthodes plus approfondies pour obtenir les informations d’interprétabilité (via SKATER, LIME ou SHAP).*

Selon une analyse succincte, on peut retenir la régression linéaire comme étant la plus adaptée à notre problème.

Les bonnes performances s’expliquent par les variables de consommation qui sont quasi-corrélées à 100% avec la variable cible CO2 :

Une image contenant texte, capture d’écran, nombre, Caractère coloré

Description générée automatiquement

Voici le top10 des critères qui impactent le plus la régression linéaire :

|  |  |
| --- | --- |
| **Features** | **Coefficient** |
| Consommation mixte (l/100km) | 38.914359 |
| Consommation extra-urbaine (l/100km) | 7.481250 |
| Carburant\_FE | -5.871244 |
| Consommation urbaine (l/100km) | 5.394725 |
| Carburant\_GO | 4.513410 |
| Carburant\_ES+ | -3.410181 |
| Carburant\_ES | -2.712111 |
| Carburant\_GN | -2.243793 |
| masse vide euro min (kg) | 1.790132 |
| Carrosserie\_MINIBUS | 1.608209 |

2 critères expliquent majoritairement les émissions :

* La consommation au 100km (mixte)
* Le type de carburant

En valeur absolue, voici les features qui impactent le plus la sortie CO2 :

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

En ce qui concerne l’arbre de décision, voyons si les features qui impactent le résultat sont les mêmes.Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Les importances divergent mais dans l’ensemble, les 2 méthodes proposent les mêmes importances aux features.

Dans ce projet, nous avons pu mettre en œuvre un grand nombre d’algorithmes permettant de prédire la sortie d’émissions de CO2 avec une grande précision.   
Cependant, l’échelle de comparaison étant infinitésimale, nous n’avons pas pu mettre en « évidence » de façon claire les comportements de chacun des algorithmes.

Bien que nous n‘avons pas une quantité conséquente de variables dans notre jeu de données, nous aurions pu tester des modèles sans les prendre toutes en compte.   
Nous savons notamment que la colinéarité entre variable peut dégrader les performances d’un modèle.   
*A priori*, nous aurions pu choisir les variables dont la p-value est la plus faible par rapport à la variable cible (f\_regression, SelectKBest). Cependant, il faut bien veiller à ne pas inclure les variables explicatives trop corrélées entre elles dans le modèle.

*A posteriori*, pour un arbre de décision, il aurait tout à fait pu être possible d’entraîner les variables les plus importantes (SelectFromModel).  
Nous faisons le même constat quant à la régression linéaire. Il aurait été possible de ne prendre que les 5 coefficients les plus importants dans la modélisation.   
NB : Veiller à toujours exclure les variables explicatives trop corrélées entre elles en plus de la préconisation.

Une option alternative consiste à modéliser une régression linéaire régularisée. Nous sommes bien en présence d’un cas où des variables explicatives sont extrêmement corrélées entre elles.