

Groupe 5

Benjamin Colson, Edgar Saiz, Isra Friia, Luxon Masseau et Papelard Charlotte

**Combien vaut votre voiture ?**

Une application d’estimation du prix

de revente pour votre voiture

Une image contenant roue, voiture, véhicule, Véhicule terrestre

Description générée automatiquement

Master SEP

Promotion 2023-2024

Table des matières

[Remerciements 3](#_Toc153485000)

[Introduction 4](#_Toc153485001)

[1. L’interface utilisateur 5](#_Toc153485002)

[1.1. L’interface d’accueil 5](#_Toc153485003)

[1.2. Le formulaire 5](#_Toc153485004)

[1.3. L’interface de sortie 7](#_Toc153485005)

[2. Préparation des données 9](#_Toc153485006)

[2.1. La base de données est construite avec une méthode de scraping 9](#_Toc153485007)

[2.2. Nettoyage des données 12](#_Toc153485008)

[2.3. Lien VBA-Python 13](#_Toc153485009)

[3. Méthode de prédiction 14](#_Toc153485010)

[3.1. Essais de méthode 14](#_Toc153485011)

[3.2. Méthode sélectionnée 19](#_Toc153485012)

[Conclusion 21](#_Toc153485013)

[Annexe 22](#_Toc153485014)

[Présentation de la base 22](#_Toc153485015)

[Architecture du projet 24](#_Toc153485016)

# Remerciements

Il est indéniable que la réussite de notre projet repose sur les épaules de ceux qui ont généreusement partagé leur expertise et leur savoir-faire.

* Nous tenons à exprimer nos sincères remerciements à **Morgan Cousin**, dont les contributions méthodologiques et les outils innovants ont profondément marqué notre approche. Ses conseils éclairés ont joué un rôle déterminant dans notre progression, influant tant sur nos compétences verbales que sur notre structuration interne.
* Nous sommes également reconnaissants envers **Amor Keziou**, dont les avis éclairés sur le choix des modèles de prédiction ont apporté une lumière critique à nos méthodes. Sa guidance a été un pilier essentiel pour naviguer dans le domaine complexe de la modélisation et de la prédiction.
* Enfin, nous saluons avec gratitude **Arona Diene** pour son expertise précieuse dans la conception et le développement de notre interface. Son engagement et son savoir-faire ont contribué de manière significative à la création d'une expérience utilisateur exceptionnelle.

Nous exprimons nos sincères remerciements à chacun de nos professeurs dévoués, dont l'impact positif sur notre projet demeurera mémorable. Leur contribution constitue une source d'inspiration précieuse pour les défis à venir que nous comptons relever.

# Introduction

L'industrie automobile en France demeure un secteur d'activité incontournable, suscitant un intérêt soutenu et occupant une place centrale dans la vie quotidienne des Français. Avec plus de 37,8 millions de voitures particulières en circulation, le paysage automobile français évolue constamment. Malgré cela, l'achat d'une voiture neuve représente souvent un investissement financier considérable, incitant de nombreuses personnes à se tourner vers le marché des véhicules d'occasion. La question de l'achat et de la revente de voitures d'occasion est ainsi cruciale et concerne des millions de citoyens. C'est dans ce contexte que notre équipe s'est engagée dans la création d'une application visant à simplifier le processus d'estimation du prix de revente des véhicules d'occasion, afin de répondre aux besoins des propriétaires de voitures d'occasion en France.

Permettez-moi de vous présenter les membres dévoués de notre équipe qui ont mis leurs compétences au service de cette application :

* **Benjamin Colson (Product Owner)** : Il guide l'équipe en définissant la vision du produit, en priorisant les fonctionnalités et en veillant à ce que l'application réponde aux besoins des utilisateurs. Fort de son expertise en gestion de projets, il est le moteur derrière notre application de prédiction sur le prix des voitures ;
* **Edgar Saiz (Scrum Master)** : En tant que Scrum Master, il assure l'application des méthodologies agiles, favorise la collaboration au sein de l'équipe et garantit que les objectifs du projet sont atteints dans les délais ;
* **Charlotte Papelard (Développeuse VBA)** : Spécialisée dans le développement de fonctionnalités et d'outils pour l'analyse des données, elle crée des modèles de prédiction en utilisant VBA, automatisant ainsi le traitement des données ;
* **Isra Friaa (Data Engineer)** : Son rôle essentiel est la collecte, la préparation et la structuration des données alimentant l'application, assurant ainsi la disponibilité de données de qualité pour des prédictions précises en matière de prix du véhicule ;
* **Luxon Masseau (Data Scientist)** : Il se concentre sur l'élaboration de l'algorithme de prédiction au cœur de l'application, développant des modèles analytiques pour faire de notre application un outil fiable pour prédire le prix des voitures des automobilistes.

Cette équipe vous propose un rapport explorant l’interface utilisateur (1), la préparation des données (2) et la méthode de prédiction (3) derrière une telle application.

# L’interface utilisateur

Cette partie explore tout le travail effectué sous Excel et VBA, *i.e.* qu’elle passe par l’interface d’accueil (1.1), le formulaire (1.2.) ainsi que l’interface de sortie (1.3.).

## L’interface d’accueil

Dans un premier temps, l’utilisateur est accueilli sur une page d’accueil (*cf.* Figure 1) qui rappelle le contexte, le but ainsi que le fonctionnement de l’application. Elle explique le principe du bouton « Formulaire » qui sera la première étape de la prédiction du prix du véhicule ainsi que le bouton « Mise à jour » qui va permettre de remettre à jour la base de données (*cf.* 2.1.).

Une image contenant texte, roue, véhicule, Véhicule terrestre

Description générée automatiquement

Figure  : Interface d’accueil de l’application

Après une éventuelle mise à jour des données de l’application, l’utilisateur lance le formulaire.

## Le formulaire

Une fois que l’utilisateur lance le bouton « Formulaire », il arrive sur une première page nommée « Voiture » (*cf.* Figure 2).

Dans la section « Voiture » :

* Pour trouver la **marque et le modèle**, une aide à la sélection est proposée. Dès lors, lorsqu’on commence à taper la marque d’une voiture, une liste déroulante apparait et toutes les correspondances avec les premières lettres tapées apparaissent. L’utilisateur va pouvoir choisir sa voiture dans la liste ;
* Pour les caractéristiques qui suivent (**carburant**, **boite de vitesse**, **catégorie** et **année**) une liste déroulante incluant toutes les réponses possibles est proposée afin de faciliter et sécuriser la saisie. A noter que si la saisie de la boite de vitesse est « automatique », la caractéristique **nombre de vitesses** s’enlève car une voiture automatique n’a pas de vitesse ;
* Pour les informations de type numérique comme le **kilométrage**, le **nombre de places**, le **nombre de portes** ou le **nombre de vitesses**, une sécurité empêche de mettre des caractères ou autre symboles spéciaux – seuls des chiffres peuvent être inscrits.

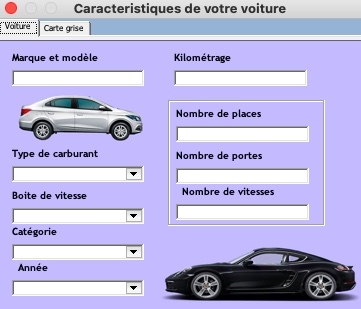


Figure 2 : Page « Voiture » du formulaire de saisie de l’application

Par ailleurs, quand l’utilisateur fini de remplir la première page, il passe intuitivement dans la section « Carte grise ».

Dans la section « Carte grise » (*cf.* Figure 3) :

* Pour chaque caractéristique à remplir à l’aide de sa carte grise (**cylindrée**, **puissance fiscale** et **puissance** **physique**) une aide visuelle pour trouver l’information s’affiche. Si l’utilisateur retourne sur la première page, puis reviens de nouveau sur celle-ci, les images ont disparues. Une sécurité est également mise en place pour forcer l’utilisateur à inscrire des chiffres en interdisant caractères et symboles spéciaux ;
* Le bouton « Valider » enregistre les caractéristiques remplies par l’utilisateur. Un message d’erreur s’affiche lorsqu’une valeur est manquante et bloque la validation des valeurs ;
* Une fois que les valeurs sont enregistrées, l’utilisateur clique sur le bouton « Prédire » qui lance l’estimation du prix de la voiture.



Figure 3 : Page « Carte grise » du formulaire de saisie de l’application

Le multipage soulage le formulaire et donne à l’utilisateur une vision intuitive de ce qu’il va devoir faire. D’une part, il sait qu’il va devoir remplir toutes les caractéristiques qu’il est censé connaître sur sa voiture, et d’autre part il devra aller chercher sa carte grise pour compléter les informations qu’il lui manque. Une fois la prédiction lancée, l’utilisateur est renvoyé sur l’interface de sortie.

## L’interface de sortie

Une fois le formulaire complété, l’utilisateur est renvoyé sur l’interface de sortie, là où il retrouvera l’estimation du prix de sa voiture (*cf.* Figure 4). Il y trouvera également :

* Le prix moyen de la marque de la voiture de l’utilisateur, qui lui donne une idée du prix pour des voitures de la même marque que la sienne ;
* La quantité de voiture de la même marque que la sienne dans la base de données. Cela lui donne une information sur la rareté de la présence de sa voiture sur le site Spoticar ;
* Un récapitulatif du top 3 des caractéristiques techniques les plus explicatives du prix (*cf.* 3. Méthode de prédiction) *i.e.* du kilométrage, des chevaux et de l’année ;
* Des informations sur la performance du modèle : précision et erreur ;
* Un histogramme avec le prix estimé de sa voiture en comparaison avec les prix moyens des 4 marques de voiture les plus représentées dans la base de données.

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

Description générée automatiquement

Figure  : Interface de sortie de l’application

Derrière la partie visible de l’iceberg se trouve tout un travail de préparation des données.

# Préparation des données

Une fois le formulaire complété par l’utilisateur, l’application doit renvoyer une estimation du prix de sa voiture. Toutefois, pour que l’estimation soit fiable, il faut un modèle prédictif qui soit construit en parallèle. Ce modèle a besoin de données solides sur lesquelles apprendre (2.1.) et propres (2.2.). Par ailleurs, les caractéristiques fournies par l’utilisateur doivent être envoyées sur Python (2.3.) afin de lancer la prédiction.

## La base de données est construite avec une méthode de scraping

Après de nombreuses recherches, aucune base de données européenne recensant les caractéristiques et les prix des voitures n'a été trouvée ; seules des données indiennes, chinoises ou américaines sont disponibles. Or, nous voulions une application qui pourrait nous être utile, et pas seulement pour l’exercice. La prédiction en euro était indispensable.

Ainsi, nous sommes partis pour construire notre propre base à partir d’une méthode de scraping qui propose plusieurs avantages tels que :

* **Le contrôle des données** – la source des données est connue ;
* **L’actualisation de la base** – ce qui permet à l’application de ne pas devenir obsolète grâce à la mise à jour pour corriger les prix de l’inflation ;
* **La personnalisation de la base** qui facilite le nettoyage des données – seules les variables intéressantes à étudier sont récupérées.

Le scraping est une technique informatique qui consiste à extraire des données à partir de sites web de manière automatisée. Dans le cadre de ce projet, deux méthodes ont été testées.

1. Première Méthode (Selenium pour l'API JSON de Spoticar)

La première méthode a privilégié l'utilisation de Selenium pour récupérer des données à partir de l'API JSON du site Spoticar. Selenium est une suite d'outils conçue pour automatiser les navigateurs web, facilitant le test automatisé des applications web. Cette approche a permis d'extraire directement des données structurées au format JSON, comprenant des informations telles que la **marque**, le **modèle**, le **prix**, le type de **carburant**, la boîte de vitesses, **l'année de mise en circulation**, et le **kilométrage**. Ces données ont été stockées dans un fichier Excel.

2. Deuxième Méthode (Selenium pour l'API JSON dynamique)

La deuxième méthode a également utilisé Selenium, mais cette fois-ci pour extraire des données à partir d'une API JSON dynamique. Le script a automatisé le navigateur pour accéder à une page web contenant des données JSON, puis a transformé ces données en un dictionnaire Python. Les informations sur les offres de véhicules, telles que la marque, le modèle, la boîte de vitesses, la couleur, etc., ont été extraites et stockées dans un DataFrame, puis sauvegardées dans un fichier Excel. C’est sur cette méthode que nous sommes restés car elle possède de nombreux avantages tels que :

* **Sa rapidité et son efficacité** : En utilisant l'API directement, le scraping est plus rapide car il contourne le rendu visuel de la page **;**
* **Sa moindre utilisation de ressources** : Cette méthode nécessite moins de ressources car elle ne charge pas l'intégralité de la page web ;
* **Sa facilité de maintenance** : En utilisant une API officielle, le scraping est plus stable face aux changements potentiels dans la structure de la page ;
* **Sa souplesse d'extension** : La méthode permet d'extraire un ensemble plus large de données en cliquant sur les éléments pour obtenir des informations détaillées.

Le scraping a permis de considérablement réduire la partie nettoyage des données en sélectionnant uniquement les variables voulues. Toutefois, cette partie est restée obligatoire.

## Nettoyage des données

Comme les données ont été récupérées directement à la source, la base était plus simple à nettoyer. Néanmoins, il a fallu :

* Retirer les valeurs manquantes ;
* Retirer les doublons pour éviter le sur-apprentissage ;
* Retirer les accents ;
* Mettre en minuscule tous les caractères.

Une fois ce travail fait, la base est enregistrée sous le nom « data\_clean.xlsx » et prête pour les modèles de prédiction.

## Lien VBA-Python

Une fois les données de l’utilisateur récupérées, l'étape suivante consiste à déclencher notre algorithme de prédiction pour obtenir le résultat final. Mais comment établir ce lien ? Dès que nous amorçons la mise à jour, le script du modèle est activé, générant ainsi les coefficients de chaque variable grâce à l'algorithme RandomForest (*cf.* Méthode de prédiction). Ces informations cruciales sont ensuite enregistrées dans un fichier Pickle mis constamment à jour à chaque nouvelle opération de scraping. Lorsque nous effectuons des prédictions sur le prix d’un véhicule, nous extrayons ce fichier Pickle pour le connecter aux données fournies par l'utilisateur. Le résultat prédit est ensuite enregistré dans une cellule dédiée d'une feuille de calcul Excel, créant ainsi une traçabilité claire du processus de prédiction. Le processus est schématisé ci-dessous.

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

Description générée automatiquement

Figure  : Architecture du projet

Une fois le nettoyage effectué, l’algorithme de prédiction se lance.

# Méthode de prédiction

Dans cette dernière partie, nous allons expliquer la mise en œuvre de notre algorithme, des premiers essaies à la sélection finale par RandomForest.

## Essais de méthode

Aux vues de notre variable cible qui est je le rappelle le prix du véhicule, une variable quantitative, nous sommes partis sur une régression linéaire multiple. La régression linéaire multiple est une technique statistique utilisée pour modéliser la relation entre une variable dépendante et plusieurs variables indépendantes. Contrairement à la régression linéaire simple qui se concentre sur une seule variable prédictive, la régression linéaire multiple prend en compte plusieurs prédicteurs, permettant ainsi de mieux capturer la complexité des relations entre les variables. L'objectif principal de cette méthode est de créer un modèle mathématique qui représente au mieux la variation de la variable dépendante en fonction des différentes variables indépendantes. Chaque variable indépendante est associée à un coefficient, indiquant l'ampleur de son impact sur la variable dépendante tout en tenant compte des autres prédicteurs. L'ajustement du modèle est effectué par minimisation des erreurs résiduelles, optimisant ainsi la précision des prédictions. La régression linéaire multiple trouve des applications dans divers domaines, notamment l'économie, la finance et les sciences sociales, offrant une approche puissante pour analyser et modéliser des systèmes complexes avec plusieurs facteurs explicatifs.

La formule d'une régression linéaire multiple est généralement exprimée comme suit :

Où :

- est la variable dépendante que vous essayez de prédire.

- est l'ordonnée à l'origine (constante).

- sont les coefficients des variables indépendantes respectivement.

- sont les variables indépendantes.

- est le terme d'erreur, représentant les erreurs non capturées par le modèle.

En terme plus simples, la valeur de la variable dépendante () est estimée comme une combinaison linéaire des valeurs des variables indépendantes, ajustée par des coefficients et une constante. L'objectif de la régression linéaire multiple est de trouver les valeurs optimales des coefficients () qui minimisent les erreurs de prédiction.

Nous avons ensuite créé une sélection de modèle selon plusieurs critères. Le processus de sélection des modèles est une étape cruciale dans la modélisation statistique, visant à identifier le modèle le plus approprié parmi plusieurs candidats. Différents critères, tels que l'AIC (Critère d'Information d'Akaike) et le BIC (Critère d'Information Bayésien), sont souvent utilisés pour évaluer la qualité des modèles. L'AIC prend en compte la qualité du modèle et la complexité, favorisant les modèles qui parviennent à bien ajuster les données tout en restant relativement simples. Le BIC, quant à lui, pénalise davantage la complexité, encourageant une plus grande parcimonie dans la sélection du modèle.

L'AIC (Critère d'Information d'Akaike) et le BIC (Critère d'Information Bayésien) sont des mesures utilisées pour évaluer la qualité d'un modèle statistique, y compris les modèles de régression. Ces critères prennent en compte la performance du modèle tout en pénalisant la complexité pour éviter le surajustement.

**1. AIC (Critère d'Information d'Akaike) :**

- : Le nombre de paramètres dans le modèle.

- : La fonction de vraisemblance maximale du modèle.

L'AIC favorise les modèles qui ajustent bien les données tout en ayant un nombre minimal de paramètres. Un AIC plus bas indique un modèle potentiellement meilleur.

**2. BIC (Critère d'Information Bayésien) :**

- : Le nombre de paramètres dans le modèle.

- : Le nombre d'observations.

- : La fonction de vraisemblance maximale du modèle.

Le BIC pénalise davantage la complexité du modèle que l'AIC. Comme pour l'AIC, un BIC plus bas indique généralement un modèle préférable.

Dans ces formules, représente le logarithme naturel. L'idée est de trouver un équilibre entre l'ajustement du modèle aux données et la complexité du modèle, afin d'éviter le surajustement. Lorsque vous comparez différents modèles, celui avec la valeur d'AIC ou de BIC la plus basse est généralement considéré comme le meilleur, compte tenu de la balance entre ajustement et complexité.

Par ailleurs, les méthodes de régularisation telles que Ridge et Lasso sont couramment employées pour améliorer la performance des modèles de régression en introduisant des termes de pénalité sur les coefficients. Le Ridge ajoute une pénalité quadratique à la somme des carrés des coefficients, tandis que le Lasso introduit une pénalité absolue. Ces approches sont particulièrement utiles pour traiter la multicolinéarité et la sélection automatique des variables, améliorant ainsi la généralisation du modèle. En fin de compte, le choix entre ces méthodes dépend du contexte spécifique du problème et des caractéristiques des données, chaque critère et méthode apportant sa propre perspective pour guider la sélection du modèle optimal.

Les méthodes de régularisation, telles que Ridge et Lasso, sont utilisées pour éviter le surajustement dans les modèles de régression. Les formules pour les termes de pénalité dans Ridge et Lasso sont ajoutées à la fonction de coût de la régression linéaire ordinaire.

**1. Ridge Regression (L2 regularization) :**

Dans Ridge Regression, on ajoute le terme de régularisation L2 à la fonction de coût :

- : Fonction de coût.

- : Somme des carrés des résidus, comme dans la régression linéaire ordinaire.

- : Paramètre de régularisation (lambda dans certaines notations).

- : Terme de régularisation L2, qui pénalise les coefficients en les poussant vers zéro.

**2. Lasso Regression (L1 regularization) :**

Dans Lasso Regression, on ajoute le terme de régularisation L1 à la fonction de coût :

- : Fonction de coût.

- : Somme des carrés des résidus, comme dans la régression linéaire ordinaire.

- : Paramètre de régularisation.

- : Terme de régularisation L1, qui pénalise les coefficients en les poussant vers zéro. La particularité de Lasso est qu'il peut conduire à des coefficients exactement égaux à zéro, réalisant ainsi une sélection de variables.

Dans ces formules, représente les coefficients de régression, est le nombre de prédicteurs, et est le paramètre de régularisation qui contrôle l'intensité de la régularisation. Plus est grand, plus la pénalité est forte, conduisant à des coefficients de régression plus petits.

Ces termes de régularisation sont ajoutés à la fonction de coût afin de trouver une solution qui équilibre la meilleure adéquation aux données et la complexité du modèle.

Lors de la réalisation d'une régression linéaire multiple (RLM), la validation des hypothèses sous-jacentes est essentielle pour garantir la fiabilité des résultats obtenus. Plusieurs hypothèses fondamentales doivent être vérifiées pour s'assurer que le modèle est robuste et que ses conclusions sont interprétables. Tout d'abord, l'homoscédasticité, qui suppose une variance constante des résidus à tous les niveaux des variables indépendantes, doit être évaluée. Des graphiques de résidus ou des tests statistiques peuvent être utilisés pour détecter d'éventuelles violations de cette hypothèse.

Deuxièmement, la normalité des résidus est une hypothèse importante. Idéalement, les résidus devraient suivre une distribution normale pour que les tests d'inférence soient valables. Des graphiques quantiles-quantiles et des tests statistiques comme le test de Shapiro-Wilk peuvent être appliqués pour évaluer la normalité des résidus.

Troisièmement, la linéarité entre les variables indépendantes et la variable dépendante doit être vérifiée. Des graphiques de régression partielle ou des graphiques de résidus peuvent aider à identifier des tendances non linéaires.

En outre, il est crucial de s'assurer que les résidus sont indépendants les uns des autres. La présence de dépendance sérielle peut compromettre l'efficacité des estimations.

Enfin, la multicolinéarité, qui mesure la corrélation entre les variables indépendantes, doit être évaluée. Une forte multicolinéarité peut rendre les estimations des coefficients peu fiables. Des diagnostics tels que le VIF (variance inflation factor) peuvent aider à identifier ce problème.

En résumé, la validation des hypothèses lors d'une RLM est cruciale pour s'assurer de la validité des conclusions tirées du modèle. Elle nécessite une combinaison de diagnostics graphiques et de tests statistiques pour évaluer la conformité aux différentes hypothèses sous-jacentes à la régression linéaire multiple.

Rajouter les graphs

## Méthode sélectionnée

La forêt aléatoire, ou RandomForest en anglais, est une méthode d'apprentissage automatique qui repose sur l'agrégation de multiples arbres de décision pour améliorer la robustesse et la précision des prédictions. Cette approche présente plusieurs avantages, notamment sa capacité à gérer des ensembles de données complexes, non linéaires et comportant de nombreuses variables. Un aspect fondamental de la forêt aléatoire réside dans la diversité des arbres de décision qui la composent, obtenue en introduisant des éléments d'aléatoire lors de la construction de chaque arbre.

Chaque arbre de décision est formé sur un sous-ensemble aléatoire des données d'entraînement et sur un sous-ensemble aléatoire des variables explicatives. Cette diversification réduit le surajustement (overfitting) et améliore la généralisation du modèle aux données inconnues. Lors de la prédiction, chaque arbre contribue à la décision finale, et le résultat est déterminé par un vote majoritaire dans le cas de classifications ou par une moyenne dans le cas de régressions.

La forêt aléatoire est particulièrement efficace pour traiter des problèmes complexes, identifier des interactions non linéaires entre les variables et gérer des jeux de données de grande dimension. Elle est également robuste face au bruit et aux valeurs aberrantes. Cependant, il est important de noter que son interprétation peut être plus complexe par rapport à des modèles plus simples comme la régression linéaire.

En résumé, la forêt aléatoire est une technique d'apprentissage automatique puissante et polyvalente, souvent privilégiée pour sa capacité à fournir des prédictions précises et à gérer des données complexes tout en limitant le surajustement.

Le critère de Gini est une mesure d'impureté utilisée dans la construction d'arbres de décision, y compris ceux utilisés dans l'algorithme RandomForest. Il mesure l'homogénéité des classes dans un nœud donné d'un arbre de décision.

La formule du critère de Gini pour un nœud donné est la suivante :

- : le critère de Gini pour le nœud .

- : le nombre de classes dans le problème de classification.

- : la proportion d'échantillons du nœud qui appartiennent à la classe .

Le critère de Gini est compris entre 0 et 1. Un score de 0 indique une pureté maximale, ce qui signifie que tous les échantillons du nœud appartiennent à la même classe. Un score de 1 indique une impureté maximale, ce qui signifie que les échantillons sont uniformément répartis entre les différentes classes.

Lors de la construction d'un arbre de décision, le critère de Gini est utilisé pour évaluer la qualité de la séparation des données à chaque nœud. Plus la diminution du critère de Gini est importante après une division, plus la division est considérée comme bonne. Cela permet à l'algorithme de prendre des décisions pour maximiser la pureté des nœuds dans l'arbre.

# Conclusion

En conclusion, nous avons réussi à créer une application qui permet de prédire le prix d’un véhicule « de tous les jours ». Notre formulaire n’est en effet pas adapté pour les supercars ou les hypercars. Il permet cependant de remplir sa mission initiale, qui était aux usagers d’avoir un outil simple et intuitif pour prédire le prix de votre véhicule grâce à sa carte grise.

L’autre limite de notre application, c’est le fait qu’elle ne prend pas en compte si le véhicule nécessite ou non des réparations, l’état des pneus et en partant du principe d’un contrôle technique a été fait moins de 6 mois auparavant.

Si nous souhaitons prendre ces paramètres en compte pour pouvoir mieux prédire le prix d’un véhicule, cela nécessiterait des données supplémentaires ou même une base de données différente.

# Annexe

## Présentation de la base

Afin de construire une telle application, nous utiliserons une base de données obtenue par *scraping* sur le site [www.spoticar.fr](http://www.spoticar.fr), leader européen du marché de l’occasion, qui recense plus de 70 000 offres de voitures d’occasion. Le *scraping* possède plusieurs avantages :

**Contrôle** – Il permet d’avoir la main mise sur les données ;

**Actualisation** – Il permet la mise à jour de la base de données en direct, ce qui va permettre de ne pas rendre le produit obsolète vis-à-vis des prix pour ainsi suivre les variations du marché ;

**Personnalisation** – Il permet de faciliter le nettoyage de la base car les données sont récupérées à la source *i.e.* que seules les variables d’intérêt seront récupérées. C’est un vrai avantage car les variables et individus peuvent être augmentées au besoin.

Les variables

La variable que nous cherchons à prédire, *i.e.* notre **variable d’intérêt**, est le prix du véhicule. Les autres variables présentes dans notre base de données seront utilisées pour définir cette variable.

Ainsi, nos **variables explicatives**, qui seront sollicitées auprès du client pour une analyse approfondie de sa voiture seront :

Les données obtenues se présentent de la manière suivante :

|  |  |
| --- | --- |
| Marque | Marque de la voiture (Peugeot, Opel, etc.) |
| Modèle | Modèle de la voiture (308, 107, etc.) |
| Marque et modèle | Fusion des deux premières variables |
| Boite de vitesse | Automatique/Manuel |
| Couleur | Couleur de la voiture |
| Crit’air | Vignette sur la pollution de la voiture |
| Catégorie | Catégorie du véhicule (4x4, berline, etc.) |
| Cylindrée | Nb de cylindrée du véhicule |
| Kilométrage | Nb de kilométrage du véhicule |
| Nb place | Nb de places du véhicule |
| Nb porte | Nb de portes du véhicule |
| Nb vitesse | Nb de vitesse du boitier |
| Puissance fiscale | Nb de chevaux fiscaux |
| Puissance physique | Nb de chevaux physiques |
| Carburant | Type de carburant (essence, diesel, etc.) |
| Année | Année de mise en circulation du véhicule |
| Nombre de roues motrices | Nb de roues motrices |
| Prix | Prix du véhicule |

Cette base contient initialement 10 000 lignes qu’on nettoiera dans un premier temps avant de déployer notre modèle de sélection de variables puis de prédiction afin d’obtenir le meilleur résultat possible dans le prix du véhicule du client.