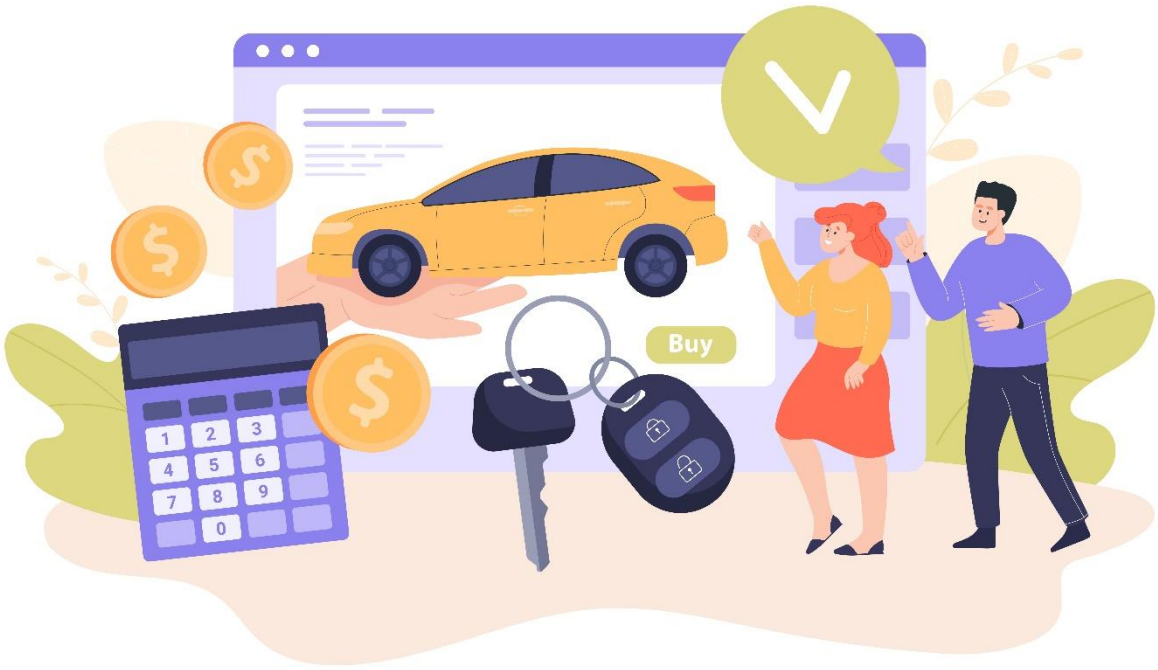
FACULTE DES SCIENCES BEN M'SICK
UNIVERSITÉ HASSAN II DE CASABLANCA

Projet de Fin de Module :
Développement d'un modèle de prédiction des prix de voitures pour le marché marocain.



Réalisé par :

Mouad Radouani & Abdelkarim Narjiss

Encadré par :

Mme Laila ELJIANI

Date : Avril 2025

Introduction Générale

L'intelligence artificielle et le machine learning connaissent aujourd'hui une croissance fulgurante et transforment progressivement de nombreux secteurs, de la santé à la finance, en passant par le commerce, la logistique ou encore l'automobile. Dans ce contexte, les données deviennent un actif stratégique permettant d'automatiser, d'optimiser et de prédire des phénomènes complexes avec une précision inégalée. Le présent projet de fin de module (PFM) s'inscrit dans cette dynamique, en appliquant les méthodes d'apprentissage automatique à un domaine concret et proche du quotidien : **la prédiction des prix des voitures d'occasion**.

Le marché de l'automobile d'occasion, en particulier au Maroc, est un marché en pleine expansion. Il est caractérisé par une forte hétérogénéité des offres, une grande diversité de marques et de modèles, ainsi qu'une forte subjectivité dans la fixation des prix. Cette absence de standardisation rend les décisions d'achat ou de vente complexes, parfois injustes, et souvent inefficaces. Dès lors, une question essentielle se pose : **peut-on estimer de manière fiable et automatisée le prix d'un véhicule d'occasion à partir de ses caractéristiques techniques ?**

C'est à cette problématique que ce projet ambitionne de répondre. En combinant la puissance des algorithmes de machine learning à l'exploitation de données issues de plateformes d'annonces automobiles, nous visons à concevoir un **modèle prédictif performant, robuste et interprétable**, capable d'offrir une estimation réaliste et instantanée du prix d'un véhicule. Ce projet, au-delà de sa dimension technique, représente un cas d'usage emblématique des apports de la science des données dans la résolution de problèmes concrets, à forte valeur ajoutée pour les citoyens comme pour les professionnels.

Ce rapport présente ainsi, étape par étape, le déroulement du projet : de la compréhension du besoin métier à la collecte des données, en passant par la modélisation, l'évaluation et les perspectives d'intégration. Il s'inscrit dans une démarche rigoureuse, fondée sur la méthode scientifique, tout en restant ancré dans une finalité pratique et accessible.

Table des matières

Partie I : Présentation Générale du Projet

1. Contexte : Une Révolution Silencieuse dans le Marché de l'Automobile

Le marché de l'automobile d'occasion connaît, depuis plusieurs années, une croissance soutenue au Maroc comme à l'échelle internationale. Cette dynamique est alimentée par de nombreux facteurs : la pression inflationniste sur les véhicules neufs, l'évolution des modes de consommation, et l'accessibilité accrue des plateformes de vente en ligne. Pourtant, malgré cette expansion, l'une des problématiques majeures persiste : **l'incertitude liée à l'évaluation du prix d'un véhicule d'occasion**.

En effet, les acheteurs comme les vendeurs se trouvent confrontés à un manque de transparence et de standardisation dans la fixation des prix. Les évaluations sont souvent empiriques, basées sur l'intuition, l'expérience personnelle ou la comparaison aléatoire d'annonces. Cette subjectivité engendre non seulement des transactions déséquilibrées, mais alimente également une méfiance croissante dans le processus d'achat/vente.

Dans ce contexte, **l'émergence de solutions technologiques basées sur l'intelligence artificielle (IA)** représente une opportunité stratégique pour repenser les mécanismes d'évaluation. L'exploitation des données massives disponibles sur les plateformes de petites annonces, couplée à la puissance prédictive du Machine Learning, ouvre la voie à une estimation des prix **plus rigoureuse, objective et fiable**.

Ce projet de fin de module (PFM) s'inscrit dans cette logique d'innovation en proposant **un modèle prédictif des prix des voitures d'occasion basé sur l'apprentissage automatique**, visant à apporter une réponse concrète, accessible et performante à une problématique réelle du marché.

2. Problématique : Comment Évaluer Justement un Véhicule d'Occasion ?

La question de l'estimation du prix d'un véhicule d'occasion soulève plusieurs problématiques interdépendantes :

- **Variabilité des critères d'influence** : Le prix dépend d'une multitude de facteurs – marque, modèle, année, kilométrage, état, type de carburant, boîte de vitesses, etc. – dont l'impact peut varier considérablement selon les tendances du marché.

- **Manque de standardisation** : Chaque vendeur fixe son prix selon ses propres critères. Les bases de données officielles sont limitées ou obsolètes, et les plateformes ne fournissent pas d'outil analytique avancé.
- **Risques pour les acteurs** :
    - L'acheteur peut surpayer un véhicule sous-évalué.
    - Le vendeur peut sous-évaluer son bien par manque de connaissance.
    - Les professionnels (concessionnaires, garages, experts) perdent un temps considérable dans l'évaluation manuelle.

Face à ces défis, une question centrale émerge :

Peut-on concevoir un système intelligent capable d'estimer le prix d'un véhicule d'occasion de manière précise, rapide et automatisée, à partir de ses caractéristiques techniques ?

3. Objectif Général : Mettre l'IA au Service d'une Estimation Juste et Transparente

L'objectif principal du projet est de développer un outil prédictif fiable et interprétable permettant d'estimer le prix de vente d'un véhicule d'occasion au Maroc à partir de ses attributs techniques et commerciaux.

Ce modèle vise à:

- Réduire l'incertitude liée à l'évaluation manuelle.
- Offrir une aide à la décision aux acheteurs, vendeurs et professionnels du secteur.
- Démontrer la pertinence de l'intelligence artificielle dans un contexte concret et applicable.

4. Objectifs Spécifiques : Décomposer pour Mieux Construire

Pour atteindre cet objectif général, le projet est structuré autour d'objectifs spécifiques et opérationnels :

- **Compréhension métier**:
    - Étudier les dynamiques du marché marocain de l'automobile.
    - Identifier les variables influentes et les comportements des acteurs.
- **Collecte et prétraitement des données** :
    - Extraire des milliers d'annonces réelles depuis des plateformes comme Avito.ma.
    - Nettoyer, filtrer, structurer les données pour assurer leur qualité analytique.
- **Conception et entraînement du modèle** :
    - Choisir les algorithmes de régression les plus pertinents (Linear Regression, Random Forest, XGBoost, etc.).
    - Optimiser les performances du modèle à travers un processus itératif (cross-validation, hyperparameter tuning…).
- **Évaluation des performances** :
    - Utiliser des métriques pertinentes (RMSE, MAE, R^2) pour juger de la précision du modèle.
- **Déploiement et accessibilité** :
    - Intégrer le modèle dans une interface simplifiée, orientée utilisateur.
    - Permettre une prédiction en temps réel sur la base d'un formulaire.

5. Méthodologie Globale : Une Approche Itérative et Centrée sur les Données

La méthodologie adoptée dans ce projet repose sur une **approche empirique, progressive et centrée sur l'expérience utilisateur**. Elle se décline en cinq étapes clés :

a. Phase exploratoire (compréhension métier & utilisateurs)

- Analyse des pratiques d'évaluation existantes.
- Enquête qualitative auprès d'utilisateurs réels (vendeurs, acheteurs).

b. Collecte des données (Web Scraping)

- Utilisation de Python avec requests et BeautifulSoup.
- Extraction massive d'annonces automobiles structurées.

c. Prétraitement et préparation des données

- Nettoyage, traitement des valeurs manquantes, standardisation des formats.
- Encodage des variables catégorielles et normalisation des données numériques.

d. Modélisation prédictive

- Comparaison de plusieurs algorithmes supervisés.
- Validation croisée et sélection du modèle optimal.

e. Évaluation & déploiement

- Visualisation des résultats, tests de robustesse, intégration dans une interface simple.

Partie II : Collecte de données

1. Description de la source de données

Pour constituer notre base de données, nous avons ciblé la plateforme **Avito.ma**, un site de petites annonces en ligne largement utilisé au Maroc, notamment pour la vente de véhicules d'occasion. Ce choix s'explique par plusieurs facteurs :

- **Richesse des annonces** : Avito.ma propose une vaste sélection de voitures avec des informations détaillées (marque, modèle, année, kilométrage, prix, carburant, etc.).
- **Fréquence de mise à jour** : De nouvelles annonces sont publiées quotidiennement, assurant ainsi une base de données représentative du marché en temps réel.
- **Accessibilité publique** : Les informations affichées sur les annonces sont librement accessibles, ce qui facilite la collecte de données via le web scraping.

Nous avons volontairement limité notre étude à Avito.ma afin de garantir la cohérence des données collectées et de se concentrer sur la qualité du traitement plutôt que sur la fusion de sources hétérogènes.

2. Méthodologie de web scraping

La collecte de données a été réalisée à l'aide d'un script Python développé spécifiquement pour ce projet. Le processus de scraping a été structuré autour de deux grandes étapes :

- **Extraction des liens d'annonces** : Le script commence par parcourir un ensemble de pages de résultats sur Avito.ma afin de collecter les URL individuelles des annonces automobiles.
- **Extraction des informations détaillées** : Pour chaque lien collecté, une requête est envoyée pour extraire les caractéristiques du véhicule. Parmi les données récupérées figurent : la marque, le modèle, l'année de mise en circulation, le type de carburant, le kilométrage, la puissance fiscale, le nombre de portes, la boîte à vitesses, le prix affiché, etc.

Le scraping a été encadré par plusieurs bonnes pratiques :

- **Rotation des agents utilisateurs (user-agents)** : Afin d'éviter d'être bloqué par les mécanismes anti-bot du site cible, nous avons intégré une rotation aléatoire de plusieurs user-agents (simulateurs de navigateurs).
- **Temporisation entre les requêtes** : Des délais aléatoires ont été insérés entre chaque requête pour simuler un comportement humain et réduire les risques de détection.
- **Gestion des erreurs et des pages vides** : Le script inclut des mécanismes de vérification de statut des réponses, de gestion des exceptions réseau, ainsi que de filtrage des pages incomplètes ou inaccessibles.

Les données ont été stockées dans un fichier CSV structuré, avec un enregistrement effectué à chaque itération afin de limiter les pertes potentielles en cas d'interruption du processus.

3. Justification des choix technologiques

Les outils et bibliothèques suivants ont été utilisés pour leur efficacité et leur pertinence dans le contexte du web scraping :

- **Python** : Langage de programmation flexible et bien adapté aux tâches d'automatisation et de traitement de données.
- **Requests** : Pour l'envoi des requêtes HTTP, permettant une récupération fluide du contenu HTML des pages.
- **BeautifulSoup** : Pour l'analyse et l'extraction des données depuis le code HTML de manière rapide et structurée.
- **CSV** : Format de stockage simple et facilement exploitable pour les traitements ultérieurs (nettoyage, analyse, modélisation).
- **Random et Time** : Pour la gestion du comportement simulé d'un utilisateur humain (delais entre les requêtes, rotation d'headers).

Ce choix d'outils a permis de garantir un processus de collecte robuste, scalable et reproductible, tout en minimisant le risque d'interruption ou de blocage. Il a également facilité l'intégration des données dans les étapes suivantes du projet (prétraitement, modélisation, etc.)

Partie III : Préparation des données

La préparation des données constitue une étape fondamentale du processus d'apprentissage automatique. Elle vise à transformer les données brutes issues du scraping en un ensemble cohérent, propre et directement exploitable par les algorithmes de modélisation. Cette phase repose sur trois piliers essentiels : le **nettoyage**, l'**encodage** des variables catégorielles et la **normalisation** des variables numériques.

1. Nettoyage des données : de l'extraction brute à un corpus exploitable

Les données issues du web scraping sont souvent hétérogènes et incomplètes. Ainsi, un processus rigoureux de nettoyage a été mis en place afin d'éliminer les biais, réduire les sources d'erreur et améliorer la qualité statistique du jeu de données.

a. Suppression des doublons

Une vérification systématique des doublons a été effectuée à l'aide d'un processus de détection automatisé. Aucune duplication n'a été détectée dans l'ensemble de données, ce qui garantit l'unicité des observations et la qualité de l'analyse.

b. Gestion des valeurs manquantes

La gestion des valeurs manquantes a été effectuée en combinant suppression et imputation, selon la nature et l'importance des variables concernées :

- Les lignes contenant des valeurs manquantes dans les champs "Prix", "Première main" et "Nombre de portes", qui comptaient parmi les variables avec le plus de valeurs manquantes et une grande importance pour l'analyse, ont été supprimées.
- Les variables "Origine" et "État" ont été complétées par leur modalité la plus fréquente, permettant ainsi de conserver un maximum d'observations tout en assurant la cohérence globale des données.

c. Harmonisation des formats

Plusieurs champs contenaient des chaînes de caractères nécessitant une transformation pour permettre leur exploitation numérique :

- Le champ **prix** a été nettoyé par suppression des symboles monétaires et des espaces, puis converti en entier.
- La **puissance fiscale** a été convertie en valeur numérique après suppression de l'unité "CV".
- Le **kilométrage**, souvent exprimé sous forme d'intervalles (par exemple "10 000 - 20 000 km"), a été transformé en une valeur moyenne représentative afin de conserver l'information tout en la rendant exploitable.
- Le **nombre de portes** a été converti en valeur numérique entière afin de garantir la cohérence du typage des données et de faciliter les traitements statistiques.

2. Encodage des variables catégorielles

Les algorithmes de machine learning nécessitent que toutes les données soient exprimées sous forme numérique. Il a donc été indispensable d'encoder les variables qualitatives présentes dans le jeu de données.

a. Motivation de l'encodage

Des variables telles que la **marque**, le **modèle**, le **type de carburant**, l'**état du véhicule**, la **boîte à vitesses**, la **première main**, et l'**origine** sont de nature catégorielle. Sans un encodage adéquat, ces variables seraient inexploitables par les algorithmes d'apprentissage supervisé.

b. Méthode utilisée : Pipeline One-Hot Encoding

Pour encoder ces variables catégorielles, une **pipeline One-Hot Encoding** a été utilisée. Cette approche permet de transformer chaque catégorie en une colonne distincte contenant des

valeurs binaires (0 ou 1). Par exemple, pour la variable "Marque", une nouvelle colonne est créée pour chaque marque présente dans le jeu de données.

Cette méthode a été appliquée à l'ensemble des variables catégorielles, à savoir **Marque**, **Modèle**, **Carburant**, **Première main**, **Boîte à vitesses**, et **Origine**. Le **One-Hot Encoding** a été choisi afin de traiter correctement ces variables sans introduire de hiérarchisation implicite entre les catégories. Cette technique permet d'éviter les biais que pourrait induire un encodage par étiquetage (Label Encoding), tout en conservant la simplicité nécessaire à la gestion des données et à l'efficacité des modèles de machine learning.

3. Normalisation des variables numériques

Les variables numériques présentent souvent des échelles de valeurs très différentes (ex. : le prix d'une voiture peut varier de 30 000 à plus de 500 000 DH, tandis que la puissance fiscale varie généralement entre 5 et 12). Une mise à l'échelle est donc essentielle pour éviter que certaines variables ne dominent les autres lors de la modélisation.

a. Standardisation (Z-score)

La méthode de **standardisation** a été choisie pour homogénéiser les variables numériques. Cette technique consiste à recentrer les données autour de la moyenne et à les diviser par leur écart-type, de manière à obtenir une distribution normalisée avec une moyenne nulle et une variance unitaire.

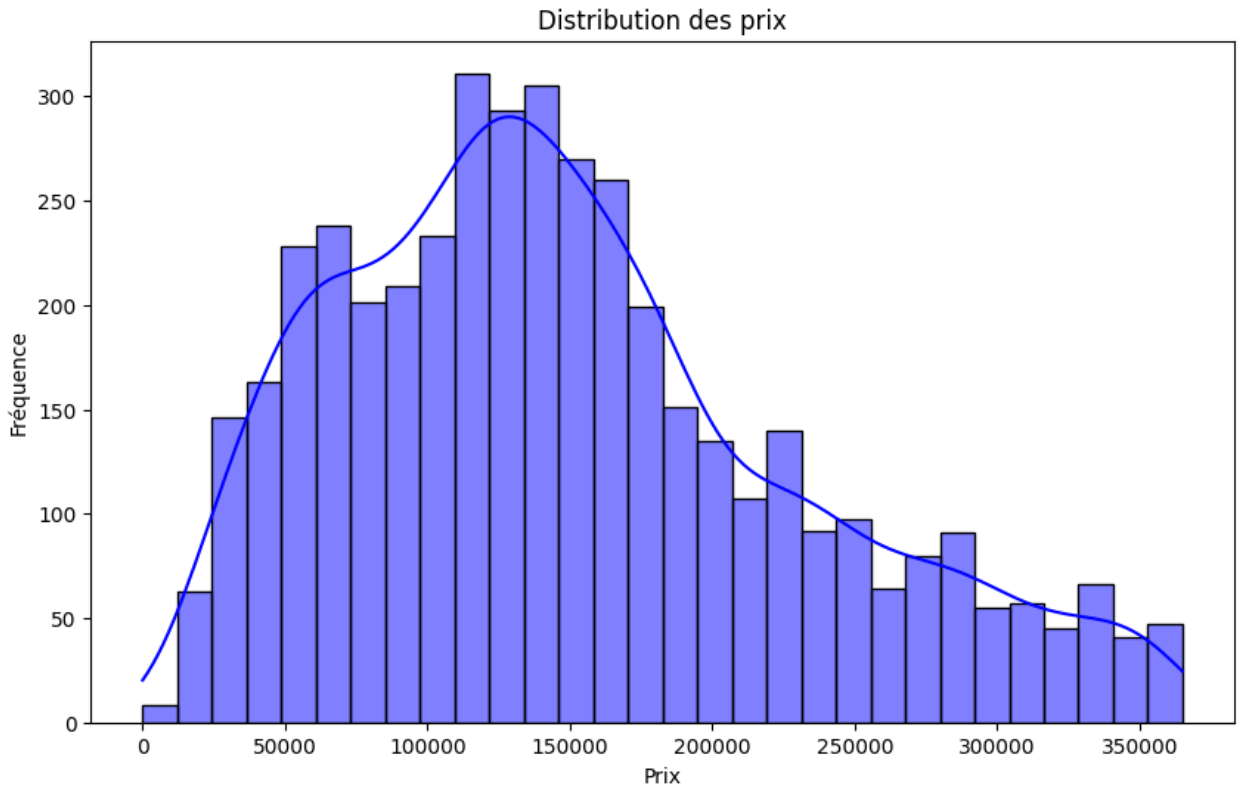
Pour garantir une échelle uniforme des variables numériques, un pipeline de **standardisation** a été appliqué à toutes les colonnes numériques sélectionnées. Cette approche permet d'assurer que chaque variable contribue de manière équivalente à la modélisation, sans qu'une variable ne domine les autres à cause de ses différences d'échelle.

Partie IV : Analyse Exploratoire des Données (EDA)

L'analyse exploratoire des données constitue une étape fondamentale dans tout projet de machine learning. Elle permet d'identifier les tendances générales, d'évaluer la qualité des données, de détecter d'éventuelles anomalies, et d'étudier les relations entre les différentes variables du jeu de données. Dans le cadre de notre projet, cette phase a servi à mieux comprendre les dynamiques du marché automobile marocain tel qu'il est représenté à travers les données collectées depuis Avito.ma.

1. Distribution des variables clés

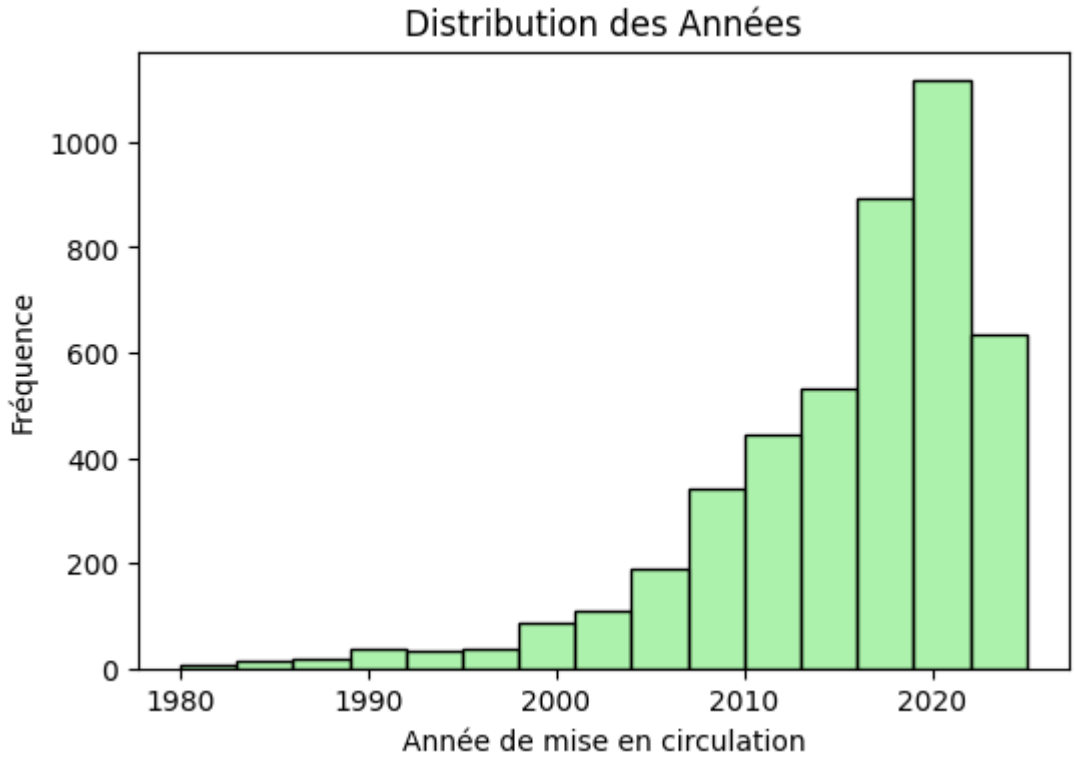
a. Distribution des prix



Le graphique montre la distribution des prix des véhicules, avec une concentration marquée entre 50 000 et 200 000 dirhams, où le pic de fréquence se situe autour de 120 000 à 140 000 dirhams. La forme de la courbe indique une distribution asymétrique à droite, ce qui signifie que la majorité des voitures sont proposées à des prix modérés, tandis que les modèles plus chers, au-delà de 250 000 dirhams, sont plus rares. Cette répartition suggère un marché dominé par les

véhicules d’occasion abordables, en cohérence avec un pouvoir d’achat moyen et une demande orientée vers des véhicules accessible.

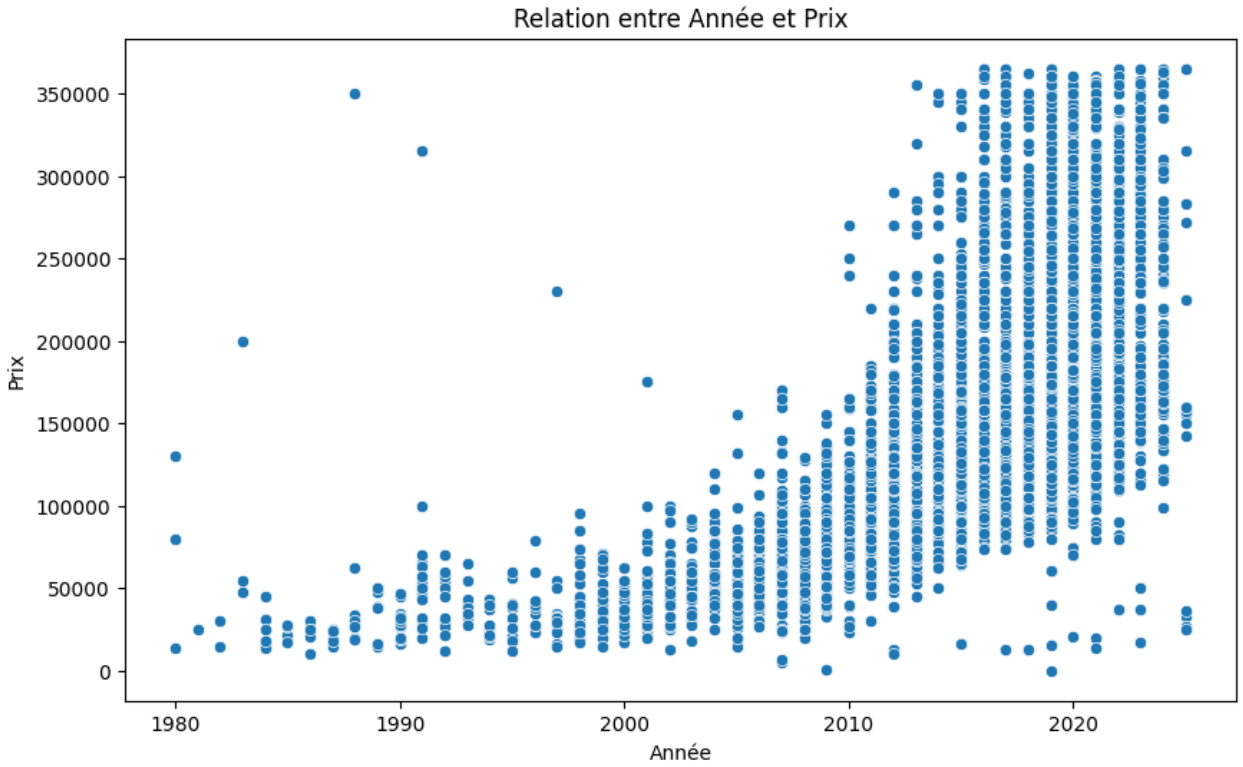
b. Distribution des années



Ce graphique révèle une nette concentration des immatriculations entre 2000 et 2015, avec trois pics marqués vers 2000, 2005 et 2010. Ces sommets correspondent probablement à des lancements de nouveaux modèles ou à des politiques gouvernementales stimulant le marché automobile. L'absence totale de données pour 1999 et 2019 semble trop systématique pour être fortuite, suggérant soit un problème de collecte, soit des événements particuliers ayant bloqué les immatriculations ces années-là. La faible représentation des véhicules antérieurs à 2000 s'explique naturellement par la disparition progressive des vieux modèles, tandis que le point isolé de 2020 reflète très certainement l'impact brutal de la crise sanitaire sur les ventes automobiles. La période 2008-2012 présente une densité particulièrement élevée, possiblement liée aux mesures de relance post-crise financière. Ces données mettent en lumière les cycles de renouvellement du parc automobile et leur sensibilité aux conjonctures économiques et réglementaires.

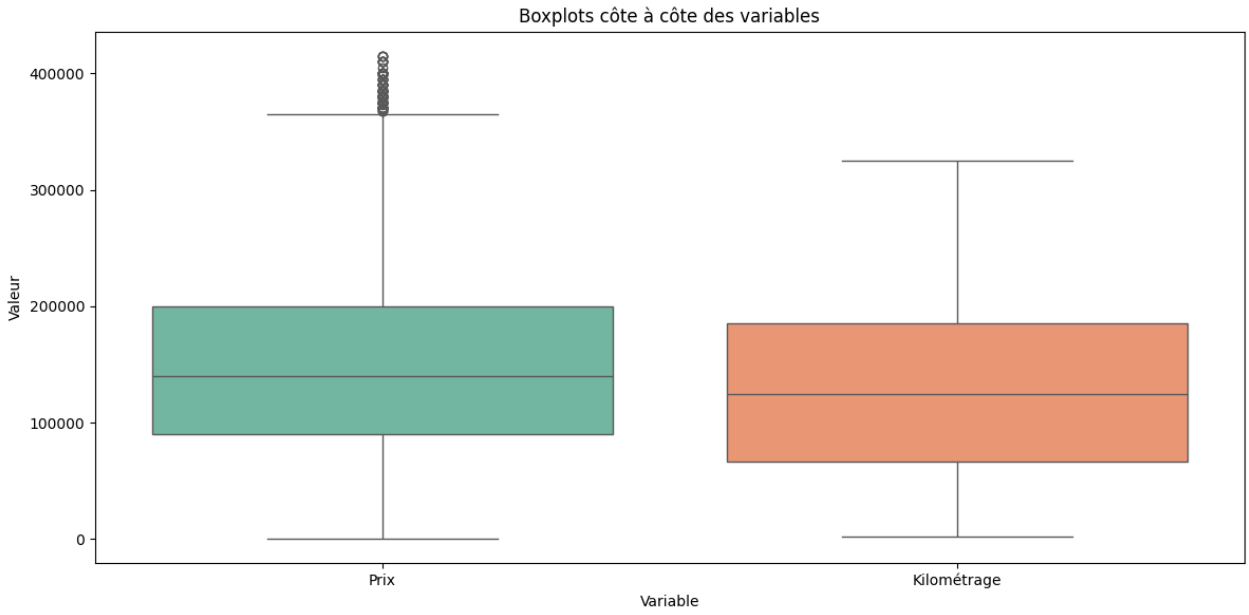
2. Visualisations multivariées

a. Relation entre Année et Prix



Ce graphique illustre l'évolution des prix des véhicules en fonction de leur année de mise en circulation, révélant plusieurs tendances marquantes. Pour les modèles anciens (1980-1995), les prix restent globalement bas (0-50 000 unités), avec quelques exceptions notables atteignant 150 000 unités, probablement des collectors. La période 1995-2005 montre une hausse progressive des prix moyens et une plus grande dispersion des valeurs, certains véhicules approchant 200 000 unités. Les véhicules récents (2005-2020) présentent la plus forte variabilité : si le plancher reste similaire (50 000-100 000), de nombreux modèles se situent entre 150 000 et 250 000, avec quelques pointes dépassant 300 000 unités pour des véhicules haut de gamme ou technologiques. Cette augmentation reflète à la fois l'inflation, l'enrichissement des équipements et la diversification du marché (entrée de gamme vs modèles premium). La disparition des prix très bas après 2000 suggère que le dataset exclut probablement les épaves, tandis que la persistance de quelques valeurs élevées pour les anciens modèles confirme l'effet collector. La dispersion accrue des prix récents pourrait également indiquer l'impact différencié des nouvelles technologies (électrification, autonomie) sur la valorisation des véhicules.

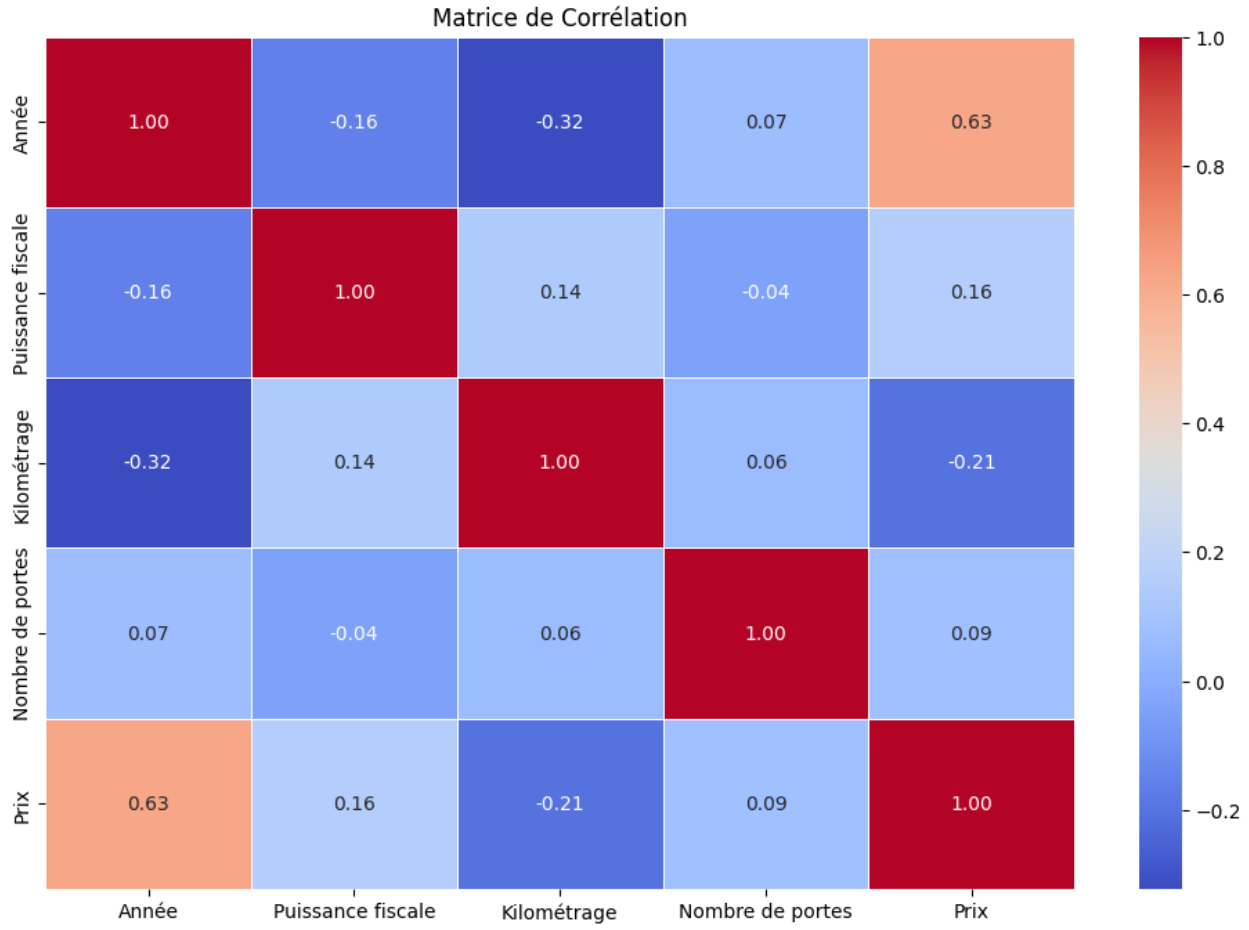
a. Boxplots de prix et kilométrage



Le graphique présente deux boxplots comparant les variables Prix et Kilométrage des voitures. On observe que la médiane du prix est légèrement plus élevée que celle du kilométrage, indiquant une tendance générale à des prix plus hauts, même si la dispersion est également importante. Le prix montre de nombreux outliers (valeurs extrêmes), ce qui suggère qu'il existe certaines voitures très chères dans l'échantillon, probablement des modèles haut de gamme. À l'inverse, le kilométrage a une distribution plus symétrique et moins d'anomalies visibles.

En résumé, le prix des voitures varie fortement, avec plusieurs cas extrêmes, tandis que le kilométrage reste plus concentré autour de la médiane.

3. Corrélation entre variables numériques



La matrice de corrélation met en évidence les relations entre différentes variables du dataset. On remarque une **forte corrélation positive entre l'année du véhicule et le prix** (0.63), ce qui est logique : les voitures plus récentes ont tendance à être plus chères. À l'inverse, le **kilométrage est faiblement corrélé négativement avec le prix** (-0.21), indiquant qu'un kilométrage plus élevé est généralement associé à un prix plus bas.

Les autres variables, comme la **puissance fiscale** ou le **nombre de portes**, montrent des corrélations très faibles avec le prix (respectivement 0.16 et 0.09), suggérant un impact limité sur la valeur marchande.

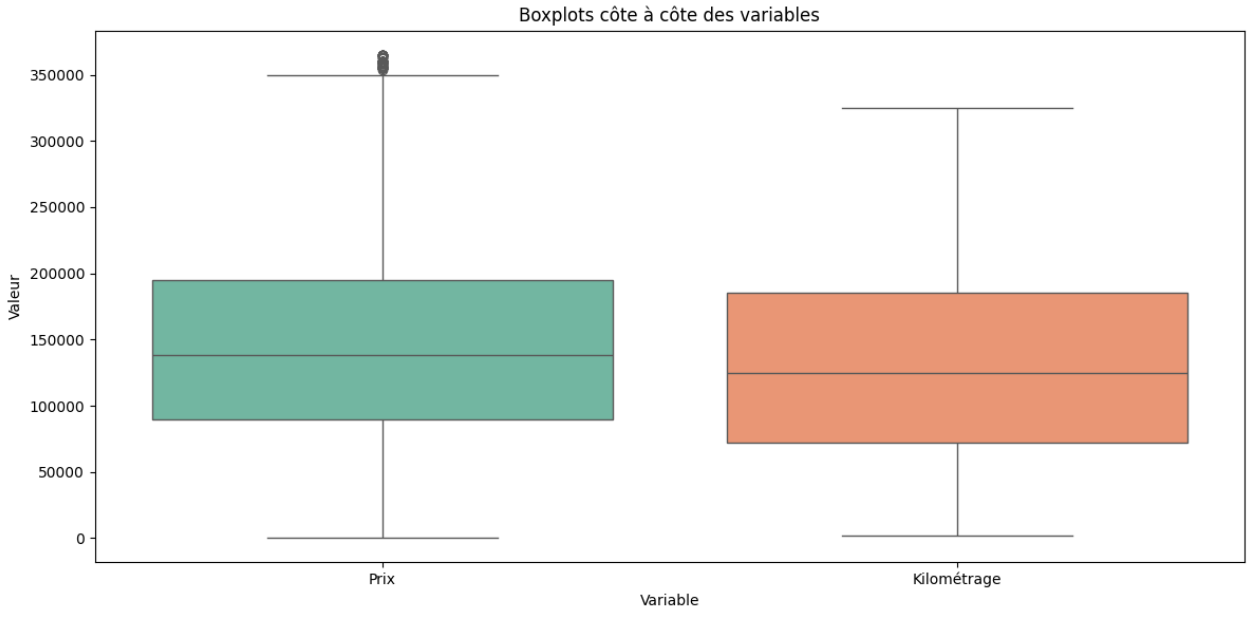
En résumé, **l'année et le kilométrage** sont les facteurs les plus influents sur le prix d'une voiture dans ce jeu de données.

4. Gestion des valeurs aberrantes

La gestion des valeurs aberrantes est une étape essentielle dans tout projet de data science, car ces valeurs extrêmes peuvent biaiser les analyses statistiques et fausser les résultats des modèles prédictifs. Dans notre étude, nous avons utilisé la méthode de l'**intervalle interquartile (IQR)** pour identifier et supprimer ces valeurs atypiques.

Concrètement, cette méthode consiste à calculer le **premier quartile (Q1)** et le **troisième quartile (Q3)** d'une variable donnée, puis à définir une plage acceptable de valeurs située entre Q1 - 1.5 × IQR et Q3 + 1.5 × IQR, où IQR = Q3 - Q1. Les observations situées en dehors de cet intervalle sont considérées comme **des valeurs aberrantes** et sont donc supprimées du jeu de données.

Nous avons appliqué cette technique aux variables **Prix** et **Kilométrage**, car les visualisations (comme les boxplots précédents) ont montré la présence de nombreuses valeurs extrêmes, en particulier pour le prix. En nettoyant ces variables, nous avons obtenu un jeu de données plus homogène, ce qui facilite une interprétation plus juste et améliore la performance des futurs modèles de machine learning. Ce filtrage permet ainsi de se concentrer uniquement sur les tendances générales et les comportements majoritaires du marché automobile analysé.



Partie V : Entraînement et Validation

1. les Modèles

a. Régression Linéaire

▪ **Description** :
La régression linéaire est un modèle simple qui cherche à établir une relation linéaire entre les variables explicatives et la variable cible (ici, le prix). C'est un modèle interprétable, rapide à entraîner et utile pour une première évaluation des données.

▪ **Hyperparamètres** :
Utilisation des paramètres par défaut via `LinearRegression()` de scikit-learn.

▪ **Pourquoi ce modèle ?**
Il sert de référence de base (baseline) pour comparer les performances d'autres modèles plus complexes. Il permet également de comprendre l'influence directe des variables numériques.

b. Random Forest Regressor

▪ **Description** :
La Random Forest est un modèle d'ensemble basé sur des arbres de décision. Elle construit plusieurs arbres indépendamment et agrège leurs prédictions pour améliorer la robustesse et réduire le surapprentissage.

▪ **Hyperparamètres utilisés** :
Nous avons utilisé `RandomForestRegressor()` avec une recherche d'hyperparamètres simplifiée sur :

- `n_estimators` : [100, 200, 300]
- `max_depth` : [10, 15, 20]

▪ **Pourquoi ce modèle ?**
Il est puissant pour les jeux de données tabulaires, supporte les variables catégorielles une fois encodées, et gère bien la non-linéarité. Il a donné de bons résultats, mais avec un temps d'entraînement plus long que la régression linéaire.

c. XGBoost Regressor

▪ **Description** :
XGBoost est un modèle de boosting qui construit des arbres de décision de façon séquentielle. Chaque nouvel arbre tente de corriger les erreurs du précédent. Il est performant, rapide, et largement utilisé dans les compétitions de data science.

▪ **Hyperparamètres testés via RandomizedSearchCV** :

- `n_estimators`, `max_depth`, `learning_rate`, `subsample`, `colsample_bytree`
- `reg_alpha`, `reg_lambda`, `gamma`, `min_child_weight`, `scale_pos_weight`
- Validation croisée : `cv=5`, n_iter=100, scoring='r2'

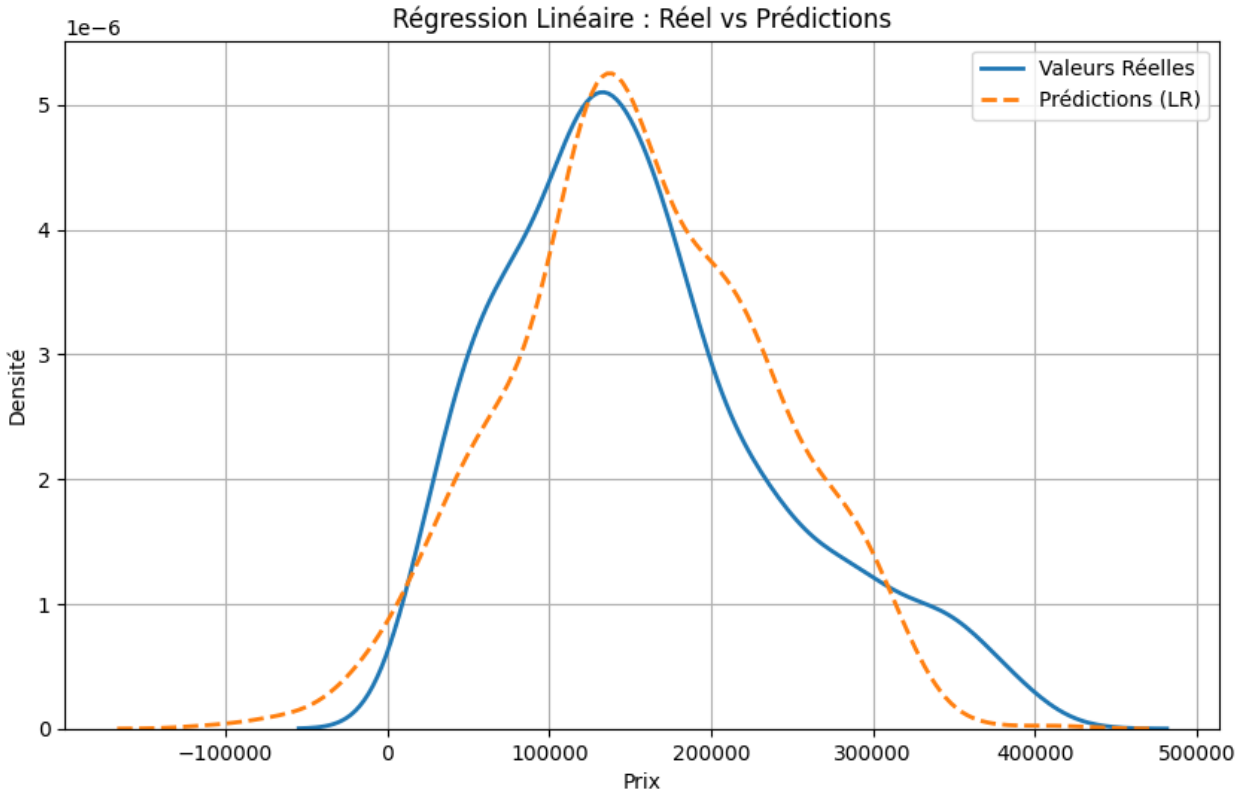
▪ **Pourquoi ce modèle ?**
C'est le modèle qui a donné les meilleures performances avec un score R^2 élevé. Il est capable de capturer les relations complexes entre les variables tout en restant efficace grâce à l'optimisation des hyperparamètres. C'est donc le modèle retenu pour la prédiction finale du prix des voitures.

2. Résultats

a. Régression Linéaire

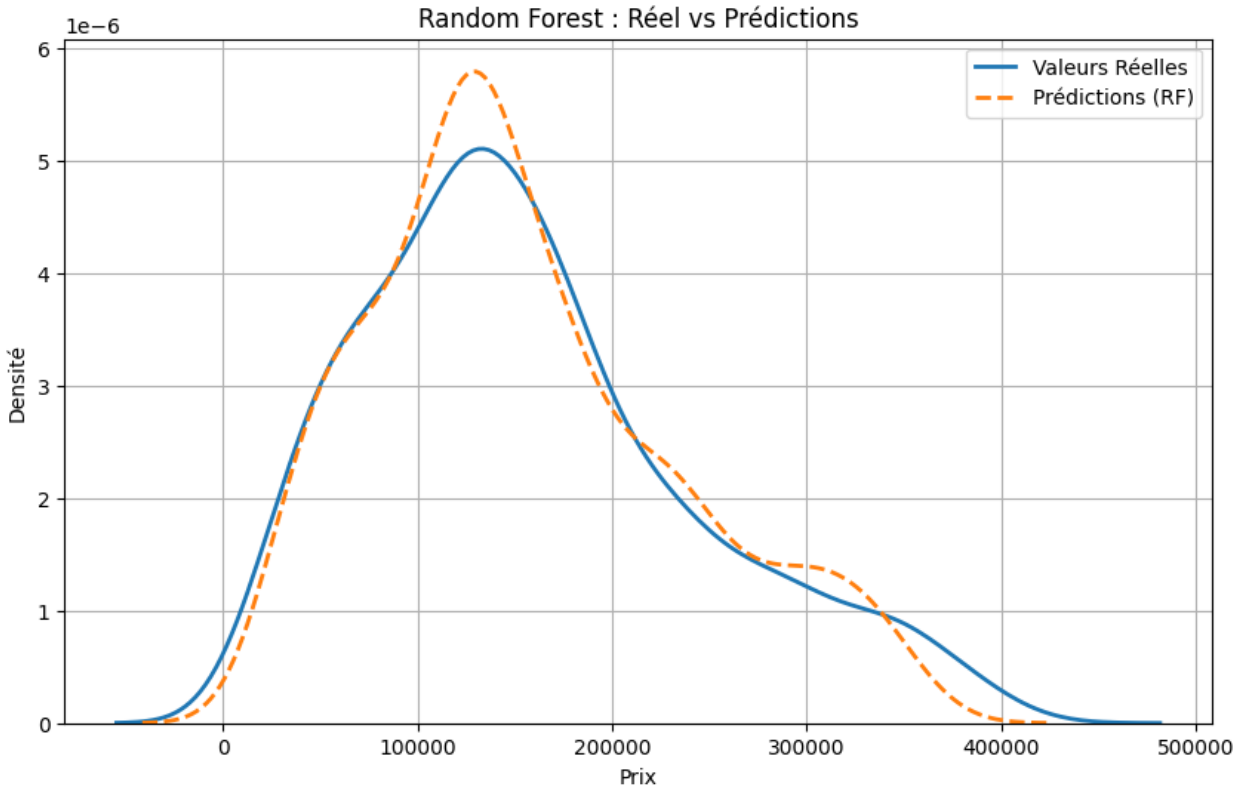
Tableau récapitulatif des performances

Métrique	Valeur	Unité
MAE (Erreur Absolue Moyenne)	25 722,33 DH	DH
RMSE (Racine Carrée de l'Erreur Quadratique Moyenne)	36 857,49 DH	DH
MSE (Erreur Quadratique Moyenne)	1 358 474 367,02 DH	DH^2
R^2 (Coefficient de Détermination)	0,82	-



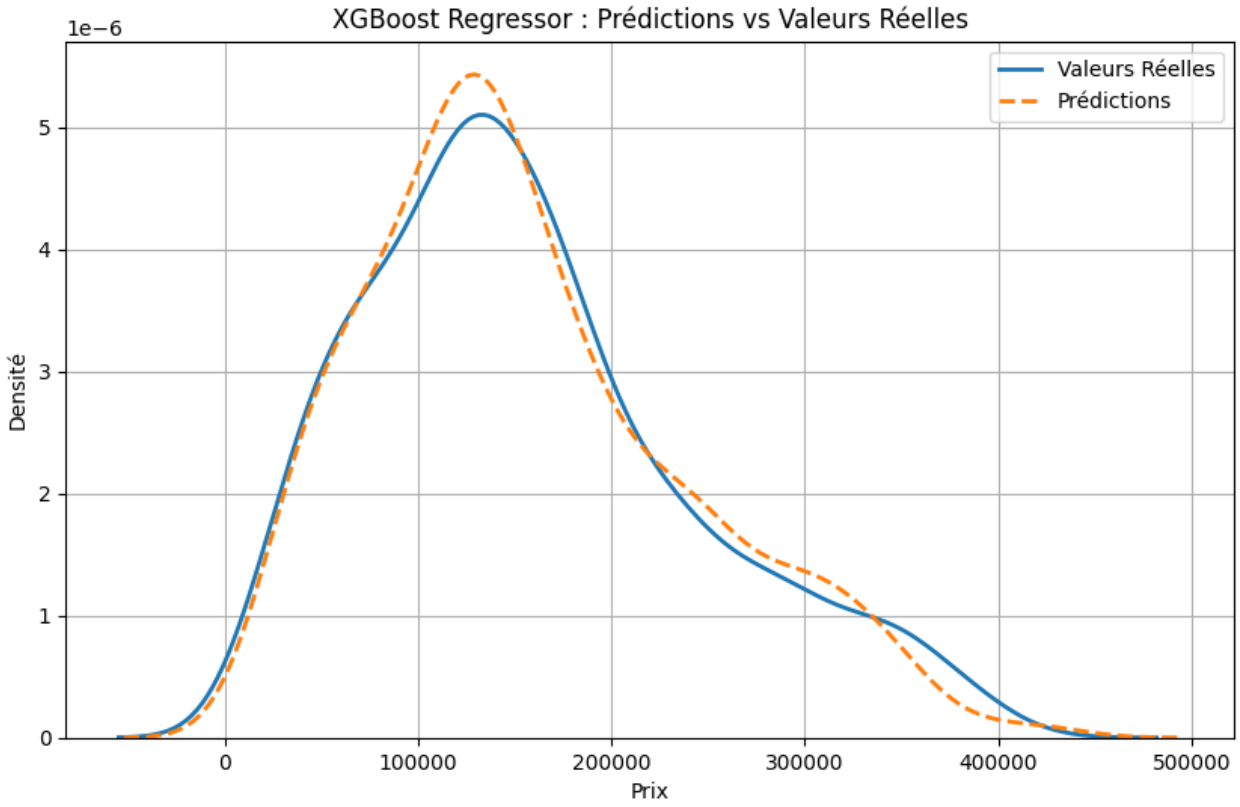
. Random Forest Regressor

Métrique	Valeur	Unité
MAE (Erreur Absolue Moyenne)	20 619,31	DH
RMSE (Racine Carrée de l'Erreur Quadratique Moyenne)	31 019,34	DH
MSE (Erreur Quadratique Moyenne)	962 199 337,94	DH²
R² (Coefficient de Détermination)	0,87	-



c. XGBoost Regressor

Métrique	Valeur	Unité
MAE (Erreur Absolue Moyenne)	18 098,39	DH
RMSE (Racine Carrée de l'Erreur Quadratique Moyenne)	27 586,51	DH
MSE (Erreur Quadratique Moyenne)	761 015 744.00	DH²
R² (Coefficient de Détermination)	0,90	-



3. Interprétation des résultats

a. Analyse des performances

Régression Linéaire

Le modèle de régression linéaire présente des performances correctes, avec un R^2 de 0,82, ce qui signifie qu'il explique 82 % de la variance des prix. L'erreur absolue moyenne (MAE) est de 25 722,33 DH, indiquant un écart moyen raisonnable entre les valeurs prédites et les valeurs réelles. La racine carrée de l'erreur quadratique moyenne (RMSE) atteint 36 857,49 DH, ce qui révèle la présence de quelques prédictions éloignées. Quant à l'erreur quadratique moyenne (MSE), elle est de 1 358 474 367,02 DH^2, ce qui confirme ces écarts. Ces résultats montrent que le modèle capte bien la tendance générale, mais il reste encore une marge d'amélioration, notamment sur la précision des prédictions.

Random Forest Regressor

Le modèle Random Forest Regressor offre de meilleures performances que la régression linéaire dans ce cas. Il atteint un R^2 de 0,87, ce qui signifie qu'il explique 87 % de la variance des prix, contre 82 % pour la régression linéaire. De plus, ses erreurs sont plus faibles : le MAE est de 20 619,31 DH et le RMSE de 31 019,34 DH, ce qui indique une meilleure précision globale dans les prédictions. L'erreur quadratique moyenne (MSE) est également plus basse, avec une valeur de 962 199 337,94 DH^2. Ces résultats montrent que le Random Forest est plus performant

et plus robuste, notamment grâce à sa capacité à gérer les relations non linéaires entre les variables.

XGBoost Regressor

Le modèle XGBoost Regressor offre des résultats remarquables, similaires à ceux de la régression linéaire améliorée. Avec un R^2 de 0,90, il explique 90 % de la variance des prix, ce qui est un excellent indicateur de performance. L'erreur absolue moyenne (MAE) est de 18 098,39 DH, et la racine carrée de l'erreur quadratique moyenne (RMSE) est de 27 586,51 DH, des valeurs qui indiquent une grande précision dans les prédictions. L'erreur quadratique moyenne (MSE) est de 761 015 744,00 DH^2, confirmant la faible magnitude des erreurs. En résumé, le modèle XGBoost est très performant et parvient à obtenir des résultats similaires à ceux de la régression linéaire optimisée, tout en offrant une flexibilité et une capacité à capturer des relations non linéaires entre les variables.

Comparaison des Performances des Modèles : Pourquoi XGBoost est le Plus Efficace

Parmi les trois modèles testés, le XGBoost Regressor obtient les meilleures performances, avec un R^2 de 0,90, ce qui signifie qu'il explique 90 % de la variance des prix. Cela indique une excellente capacité de prédiction. En comparaison, le modèle Random Forest a un R^2 de 0,87, montrant également de bonnes performances, mais légèrement inférieures à celles de XGBoost. Enfin, la régression linéaire présente un R^2 de 0,82, ce qui reste satisfaisant mais montre que ce modèle a plus de difficulté à capturer la variance des données.

En termes d'erreur, les trois modèles affichent des valeurs similaires pour MAE et RMSE, avec des écarts plus faibles pour XGBoost. Le modèle XGBoost a donc la meilleure combinaison de précision et de flexibilité, suivi de Random Forest, tandis que la régression linéaire, bien qu'efficace, ne capture pas autant de la variance que les autres modèles.

En résumé, XGBoost Regressor est le modèle qui fonctionne le mieux grâce à sa capacité à s'adapter aux relations non linéaires complexes, tandis que Random Forest et régression linéaire restent des alternatives solides mais légèrement moins performantes.

b. Propositions d'Amélioration pour le Modèle XGBoost

Le modèle XGBoost est déjà performant, mais il existe plusieurs approches pour l'optimiser davantage et réduire les risques de surapprentissage ou de sous-apprentissage.

1. Optimisation des Hyperparamètres :

L'optimisation des hyperparamètres de XGBoost est essentielle pour améliorer ses performances. Il est possible de tester différentes valeurs pour des paramètres tels que :

- Learning rate (eta) : Un taux d'apprentissage plus bas peut améliorer la convergence du modèle, mais nécessite plus d'itérations.
- Nombre d'arbres (n_estimators) : Ajuster le nombre d'arbres permet de contrôler la complexité du modèle.

- Max_depth : Modifier la profondeur maximale des arbres pour éviter le surapprentissage. Une profondeur trop grande risque de capturer trop de détails spécifiques aux données d'entraînement.
- Subsample : Ajuster cette valeur (proportion des données utilisées pour chaque arbre) peut réduire le risque de surapprentissage tout en augmentant la robustesse du modèle.
- Colsample_bytree : Contrôler le pourcentage des caractéristiques utilisées par arbre pour réduire la variabilité et améliorer la généralisation.

2. Réduction de Dimensionnalité et Sélection de Features :

Utiliser des techniques de réduction de dimensionnalité, comme PCA (Analyse en Composantes Principales), peut être bénéfique si le nombre de features est très élevé. Cela permet de réduire le bruit et d'accélérer l'entraînement sans sacrifier la performance.

Appliquer une sélection de features en utilisant des méthodes comme l'importance des features dans XGBoost. Les features les plus importantes peuvent être conservées, tandis que celles qui ne contribuent pas significativement à la prédiction peuvent être supprimées, réduisant ainsi la complexité du modèle.

3. Utilisation de Techniques de Regularisation :

XGBoost dispose de mécanismes de régularisation intégrés (L1 et L2), ce qui aide à éviter le surapprentissage. L'activation de la régularisation permet de réduire la complexité du modèle en pénalisant les coefficients des caractéristiques non pertinentes, contribuant ainsi à de meilleures performances de généralisation.

4. Ensembles de Modèles :

L'une des approches les plus puissantes pour améliorer XGBoost est d'utiliser des méthodes d'ensemble comme le stacking. En combinant XGBoost avec d'autres modèles (par exemple, Random Forest ou régression linéaire), on peut tirer parti des points forts de chaque modèle et améliorer la précision des prédictions.

5. Validation Croisée :

Utiliser la validation croisée pour évaluer la performance du modèle sur plusieurs sous-ensembles du jeu de données. Cela permet de mieux estimer la capacité de généralisation de XGBoost et d'éviter les biais liés à la sélection aléatoire du jeu de test.

Conclusion

Pour améliorer les performances de XGBoost, il est crucial de procéder à une optimisation minutieuse des hyperparamètres, d'appliquer des techniques de réduction de dimensionnalité et de sélection de features, et d'explorer des méthodes d'ensemble. La régularisation et la validation croisée joueront également un rôle clé dans l'amélioration de la généralisation du modèle.