26/04/2025

Projet machine learning : Prédiction du prix de diamonds

1er master BC

Kasraouiabdelhedi & Yosra Abdelwahed

Problématique du projet :

Dans un contexte où les technologies de l'intelligence artificielle révolutionnent de nombreux secteurs, l'industrie joaillière ne fait pas exception. Le marché des diamants repose sur des critères d'évaluation complexes mêlant caractéristiques physiques et qualités subjectives. La base de données exploitée dans ce projet comprend plus de 50 000 diamants, chacun décrit par des variables telles que le poids en carats, la qualité de la taille (*cut*), la couleur, la pureté (*clarity*), les proportions (profondeur, *table*) ainsi que les dimensions (longueur, largeur, hauteur), avec pour objectif final de prédire le **prix**.

La problématique centrale du projet consiste à identifier les attributs ayant une influence significative sur le prix et à développer un modèle de machine learning capable de prédire ce prix de manière fiable. Cette approche permettrait non seulement d'automatiser l'évaluation des diamants, mais aussi de détecter les facteurs sous-jacents aux écarts de valeur entre des pierres similaires. Plusieurs défis s'imposent : la gestion de variables qualitatives et quantitatives, la réduction de la multicolinéarité, la normalisation des données, ainsi que la sélection du modèle optimal (régression linéaire, arbres de décision, random forest, gradient boosting, etc.). Ce projet s'inscrit ainsi dans une logique de valorisation des données pour construire un outil intelligent et transparent, à la fois utile pour les acteurs du marché et illustratif du potentiel du machine learning appliqué à des problématiques économiques concrètes.

État de l'art des méthodes de prédiction du prix des diamants

1. Introduction

L'évaluation du prix des diamants constitue un défi important pour les professionnels de la joaillerie, tant les critères sont nombreux et complexes. Grâce aux avancées en machine learning, il est désormais possible de prédire ce prix avec une précision significative en se basant sur des caractéristiques objectives telles que le poids (*carat*), la qualité de la taille (*cut*), la couleur, la pureté (*clarity*) et les dimensions physiques. Cet état de l'art vise à analyser les

approches les plus couramment utilisées pour ce type de prédiction, tout en les confrontant aux résultats obtenus dans le cadre de notre projet personnel, développé sur Google Colab.

2. Méthode classique : Régression linéaire (scikit-learn)

La **régression linéaire** est l'un des modèles les plus simples et les plus interprétables. Elle cherche à établir une relation linéaire entre les caractéristiques explicatives (comme le carat ou la profondeur) et le prix.

Dans notre projet, cette approche a été implémentée comme référence de base. Si elle permet d'obtenir une vue rapide de l'impact de chaque variable, elle souffre de certaines limitations, notamment son incapacité à modéliser les relations non linéaires présentes dans les données des diamants. Cela se reflète dans les performances modestes observées (faible R^2, erreur moyenne élevée).

La régression linéaire cherche à modéliser la relation entre une variable dépendante continue (ici, le prix du diamant) et un ensemble de variables indépendantes (carat, cut, color, clarity, etc.).

Avantages :

- Simplicité d'implémentation et d'interprétation.
- Utile comme baseline (modèle de référence).
- Faible temps de calcul, même sur de grands datasets.

Limites :

- Suppose une relation linéaire entre variables indépendantes et cible.
- Sensible aux valeurs aberrantes.
- Ne capture pas les interactions complexes entre variables.

Résultat dans le projet

- R^2 relativement bas (faible pouvoir explicatif).
- Résultats incohérents pour les diamants très chers ou très bon marché.

- Modèle utile pour comparer l’apport des modèles plus avancés**.**

3. Arbre de décision (QUINLAN)

L’**arbre de décision** est une méthode non paramétrique qui segmente les données selon les variables les plus discriminantes. Il est particulièrement adapté pour des données mixtes (numériques et catégorielles) comme celles présentes dans notre dataset.

Lors de nos tests, ce modèle a montré une meilleure capacité d’adaptation que la régression linéaire, en particulier sur les extrêmes de prix. Cependant, il a tendance à surapprendre les données d'entraînement s'il n’est pas correctement régularisé (profondeur limitée, nombre minimal d’échantillons).

Un arbre de décision segmente les données par une succession de règles conditionnelles sur les variables. Chaque nœud représente une condition, et chaque feuille une prédiction.

Avantages :

- Modélise les relations non linéaires**.**
- Fonctionne bien avec des variables catégorielles et continues**.**
- Donne une interprétation visuelle du processus de décision.
- Peu de prétraitement requis.

Limites :

- Surapprentissage fréquent si l’arbre n’est pas limité en profondeur.
- Instabilité : une petite variation dans les données peut générer un arbre très différent.
- Moins performant qu’un ensemble d’arbres (ex. Random Forest, XGBoost).

Résultat dans le projet

- Meilleur R^2 que la régression linéaire.
- Mauvaise généralisation sur les données de test si l’arbre est trop profond.
- Bon outil pour identifier les variables influentes (importance des features).

4. Méthode d’ensemble : **XGBoost** (Tianqi Chen)

Le **XGBoost** (Extreme Gradient Boosting) est une méthode d’ensemble très puissante, combinant plusieurs arbres faibles pour en faire un modèle robuste. Il est connu pour ses performances exceptionnelles sur les problèmes de régression structurée.

Dans notre projet, XGBoost s’est révélé être le **modèle le plus performant** en termes de précision, avec une forte capacité à généraliser et à réduire l’erreur quadratique moyenne (RMSE). Il a surpassé les modèles précédents dans la majorité des métriques testées.

XGBoost est une technique d’**apprentissage par gradient boosting**, qui combine plusieurs **arbres de décision faibles** pour former un **modèle robuste**. Chaque nouvel arbre est entraîné pour corriger les erreurs du modèle précédent, en minimisant une fonction de coût via **descente de gradient**.

Avantages :

- Excellente précision sur les tâches de régression.
- Gère les données manquantes automatiquement.
- Intègre des mécanismes de régularisation (L1/L2) pour éviter le surapprentissage.
- Hautement optimisé pour la vitesse (parallélisation).

Limites :

- Plus complexe à configurer (nombre d’arbres, learning rate, profondeur max...).
- Moins interprétable qu’un seul arbre de décision.
- Peut devenir lourd en calcul sur des jeux de données très volumineux.

Résultat dans le projet :

- Meilleures performances globales (R^2 élevé, faible RMSE).
- Stabilité des prédictions même sur des extrêmes.
- A nécessité un grid search pour ajuster les hyperparamètres efficacement.

5. Réseaux de neurones artificiels (ANN) (Ian Goodfellow)

Les **réseaux de neurones** sont capables de modéliser des relations complexes et non linéaires. Dans notre expérimentation, un réseau de type *Multilayer Perceptron (MLP)* a été utilisé avec plusieurs couches cachées.

Ce modèle a montré des résultats compétitifs, proches de ceux de XGBoost. Il a cependant requis un **temps de calcul plus long** et un **réglage minutieux des hyperparamètres** (nombre de neurones, taux d'apprentissage, activation, etc.). Bien qu'efficace, son interprétabilité est plus faible que celle des arbres.

Les réseaux de neurones artificiels (ou MLP – Multi-Layer Perceptrons) sont composés de **couches de neurones** interconnectés. Chaque couche effectue une transformation non linéaire des données d'entrée à l'aide d'une fonction d'activation (ReLU, sigmoid, etc.).

Un modèle typique comprend :

- Une **couche d'entrée** (features),
- Une ou plusieurs **couches cachées** (intermédiaires),
- Une **couche de sortie** (valeur prédite du prix).

Avantages :

- Capacité à modéliser des relations complexes et des interactions non linéaires.
- Adapté à des données de grande dimension.
- Possibilité d'ajout de techniques de régularisation : dropout, early stopping.

Limites :

- Long temps d'entraînement si le réseau est profond.
- Peut surajuster facilement si les hyperparamètres ne sont pas bien choisis.
- Moins interprétable que les modèles à base d'arbres.

Résultat dans le projet :

- Performances proches de XGBoost après tuning.
- Training plus long**,** nécessité d'ajuster le taux d'apprentissage, le nombre de couches et de neurones.
- Bon compromis entre précision et généralisation, surtout après normalisation des données.

Critère	Régression linéaire	Arbre de décision	XGBoost	Réseau de neurones
Précision globale	Faible	Moyenne	Excellente	Très bonne
Temps d'entraînement	Très rapide	Rapide	Moyen	Long
Interprétabilité	Très bonne	Bonne	Moyenne	Faible
Sensibilité aux données	Élevée	Moyenne	Faible	Moyenne
Tuning requis	Faible	Faible	Élevé	Élevé
Capacité à modéliser le non-linéaire	Non	Oui	Oui	Oui

7. Limites et justification du projet

La littérature montre que des modèles avancés comme XGBoost et les réseaux de neurones donnent d'excellents résultats, mais au prix d'une complexité accrue. Les méthodes plus simples comme la régression linéaire restent utiles pour l'interprétation, mais sont insuffisantes seules.

Notre projet s'inscrit dans cette dynamique d'évaluation comparative. Il permet d'illustrer concrètement comment différents algorithmes traitent un même jeu de données, et met en évidence l'intérêt d'un arbitrage entre précision, coût computationnel et explicabilité selon les objectifs d'un utilisateur ou d'un professionnel.

Interprétation des modèles :

Nous avons fait tourner **quatre modèles de régression** pour prédire les **prix des diamants**, et nous avons obtenu leurs performances suivantes :

1. Arbre de décision

- **MSE** : 553 934.69
- **R^2** : 0.96

  => Modèle très bon, mais il reste une erreur significative. Les arbres de décision peuvent surajuster (overfit) les données d'entraînement, ce qui peut expliquer une erreur plus élevée que certains autres modèles.

2. XGBoost

- **MSE** : 286 536.69
- **R^2** : 0.98

  => Meilleur équilibre entre précision et robustesse. Très faible erreur, très bon pouvoir prédictif. XGBoost gère bien les interactions complexes et les non-linéarités. Il semble ici le plus performant.

3. Réseau de neurones (MLP)

- **MSE** : 1 129 637.12
- **R^2** : 0.93

  => Moins bon ici. Malgré un bon R^2, le MSE est le **plus élevé** de tous. Cela signifie que certaines prédictions sont beaucoup plus éloignées de la vérité. Peut être dû à un mauvais réglage du réseau (nombre d'epochs, architecture, normalisation, etc.).

4. Régression linéaire

- **MSE** : 307 728.19
- **R^2** : 0.9804
- **MAE** : 280.17

  => Très bonnes performances. Proche de XGBoost en R^2, mais un peu plus d'erreur moyenne (MSE). Cependant, très simple, interprétable, rapide à entraîner.

Modèle	MSE	R^2	Observations
XGBoost	**286 537**	**0.98**	**Le meilleur compromis précision/erreur**
Régression linéaire	307 728	0.98	Très bon, simple, interprétable
Arbre de décision	553 935	0.96	Bien, mais moins précis que XGBoost
Réseau de neurones	1 129 637	0.93	Mauvaise généralisation ici

=> **Le modèle recommandé est XGBoost**, car il offre le meilleur score R^2 ainsi qu'une erreur faible, ce qui en fait une solution hautement précise pour prédire les prix des diamants. Toutefois, si l'objectif est de disposer d'un modèle plus simple à interpréter, la régression linéaire constitue une alternative pertinente, bien qu'un peu moins performante sur le plan prédictif.

Bibliographie

s.d. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html>.

Ian Goodfellow, Yoshua Bengio,Aaron Courville. «Deep Learning.» (2016).

QUINLAN, JR. «Induction of Decision Trees .» (1986): 26.

Tianqi Chen, Carlos Guestrin. «XGBoost: A Scalable Tree Boosting System.» (2016).