****

**Project Report: Diamond Price Prediction with AdaBoost**

**Group member :**

* **Hergi DIANGUE DI MOUNGUET**
* **LEVI JUNIOR**

**Under the supervision of :**

* Mr Burak Civicioglu

**Academic year : 2024-2025**

**Introduction**

In this project, we used the AdaBoost algorithm to build a model capable of predicting diamond prices from a specific dataset. The main aim was to deepen the understanding and implementation of the AdaBoost algorithm, while evaluating its performance with appropriate metrics.

**Project objectives**

Apply AdaBoost to a regression task to predict diamond prices. and analyze data (EDA stage). Implement a data pre-processing pipeline. Optimize model hyperparameters to improve performance. Provide clear visualizations and interpretations of results**.**

**Methodology**

For this study, I followed a methodical approach consisting of five main stages: data exploration, pre-processing, model building, optimization and evaluation. Here is a detailed presentation of each step.

1. Data Exploration and Analysis (DEA)

First, I explored the data to gain a better understanding of their structure and characteristics:

Analysis of variable distributions: I examined quantitative and categorical variables for anomalies, striking trends or extreme values.

Exploring correlations: I used correlation matrices to identify relationships between different characteristics and their influence on the target variable (price).

Visualizing key relationships: I used graphs to highlight the links between important characteristics and prices, making it easier to identify patterns or trends.

2. Data pre-processing

Once the data had been explored, I focused on preparing it for the model:

Handling missing values: I identified missing data and applied appropriate imputation techniques or deleted observations that were too incomplete.

Encoding categorical variables: To make categorical data usable by the model, I used encoding methods such as one-hot encoding.

Normalization and standardization: I harmonized the scales of quantitative variables where necessary to improve model performance.

3. Building the AdaBoost model

For the model, I chose AdaBoost because of its ability to combine several weak models and gradually improve accuracy:

Weak model: I used a shallow decision tree as the basis for each iteration.

Initial training: Preliminary training with default hyperparameters was carried out to establish a benchmark.

4. Hyperparameter optimization

In this step, I optimized the hyperparameters to achieve the best possible performance:

Searching for optimal hyperparameters: Using GridSearchCV, I searched for ideal combinations of the following parameters:

Number of estimators (“n\_estimators”).

Learning rate (“learning\_rate”).

Tree depth (“estimator\_\_max\_depth”).

This optimization enabled me to considerably increase the model's efficiency.

5. Evaluation and visualization

To evaluate the model's performance, I adopted an approach combining metrics and visualizations:

Evaluation metrics:

Mean square error (MSE).

Mean absolute error (MAE).

Coefficient of determination (R²).

Visualization of results: I compared model predictions with actual values using graphs to better understand the accuracy and robustness of predictions.

Visualization ofpredictionsversusactualvalues**.**

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

Description générée automatiquement

Carat and price have a strong correlation (0.91), meaning that the higher the weight, the higher the price. In contrast, depth and price have almost no correlation (-0.00095), showing that depth has no real impact on price.Une image contenant texte, capture d’écran, diagramme, Tracé

Description générée automatiquement

his chart shows how diamond prices are distributed. Most diamonds cost less than 2,500, as shown by the big peak at the beginning. After that, the number of diamonds decreases as prices rise. A few diamonds are very expensive, up to 17500, but these are rare. The shape shows a majority of low prices, with a queue stretching towards the higher prices.

Une image contenant texte, capture d’écran

Description générée automatiquement

There's a general tendency for the points to rise, indicating a positive relationship. However, for the same weight, prices can vary, probably due to other criteria such as quality. Some carat values are less common, creating gaps between the points.

**Résultats**

We used an AdaBoostRegressor that combines several small models (decision trees) to improve overall performance. The base model is a DecisionTreeRegressor, and we configured AdaBoost to use 100 trees ( n\_estimators=100) with a learning rate of 0.1 ( learning\_rate=0.1).

Performance evaluation:

MSE (Mean Square Error):

This measures the mean squared deviation between our predictions and the actual values. Here, the MSE is 285282.94, which means our model is relatively accurate. A lower value indicates even better predictions.

R² (Coefficient of determination):

The R² indicates the proportion of data variance explained by our model.

With an R² of 0.98, we explain 98% of the variation in the data. This shows that our model captures the relationships between the variables very well.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

**Discussion**

The AdaBoost Regressor model performed very well, with an R² of 0.98, explaining 98% of the data variance, and an MSE of 285,282.94, demonstrating low mean error and high prediction accuracy.

However, to guarantee its robustness and generalizability, several points deserve particular attention:

Independent validation: Test the model on an independent dataset to ensure that it is not affected by overfitting.

Analysis of important variables: Identify key features influencing predictions to better understand underlying relationships.

Optimization of hyperparameters: Adjust parameters such as the number of estimators or the learning rate to improve the balance between accuracy and generalization.

In conclusion, the model provides a solid basis for meeting the objectives, but further validation and optimization are required to guarantee even more robust performance, adapted to a variety of contexts.

**Conclusion**

The AdaBoost Regressor model, built on the basis of a Decision TreeRegressor, demonstrated excellent performance with an R² of 0.98 and an MSE of 285,282.94. These results attest to its ability to efficiently capture complex relationships between variables and deliver accurate, reliable predictions.

However, while these performances are promising, certain actions remain necessary to reinforce the model's robustness and generalizability:

Validation on independent data: Evaluating the model on a completely independent dataset will confirm its ability to generalize and avoid the risk of overlearning.

In-depth analysis of variables: Identifying and studying the most influential characteristics will help to better understand the underlying dynamics and further optimize the model.

Optimization of hyperparameters: A more refined search for hyperparameters could further improve performance and guarantee a more suitable solution.

All in all, this model represents a sound basis for addressing the problem under study. Nevertheless, further adjustments and in-depth evaluation are recommended to ensure optimum performance and reliable use in a variety of contexts.

**Deliverables**

- Source code

The Python code is available in a GitHub repository, including notebooks, scripts and instructions for reproducing the results.

https://github.com/dianguehergi

- Documentation

A README file explaining the steps to replicate the experiment.

- Presentation

Slides summarizing methodologies, results and perspectives.