Laptop Price Prediction using ML

🗣️ "Hey, these are the specifications of the laptop I'm looking for. Can you tell me the cost of this laptop? Great! Just a minute — let me ask my ML model!"

By

Dudekula Abid Hussain

Email - dabidhussain2502@gmail.com | Kaggle | Github

Introduction

This is a regression-based machine learning project where the primary objective is to predict the price of a laptop based on its specifications. The dataset consists of multiple rows and columns, with each row representing a laptop and each column representing a specific feature such as manufacturer, GPU, RAM, CPU, etc. The target column is 'price', which we aim to predict using the remaining features. While with a small amount of data, one might estimate the price of a laptop fairly easily, in real-world scenarios, especially for major manufacturers like HP, Apple, or Dell, determining the price of a product involves a deeper understanding of customer demand, current market trends, competitive pricing, and production costs.

These companies rely on large-scale data and advanced analytics to strategically price their products to balance profitability and market competitiveness. Although the dataset we are using in this project is relatively small and simplified, the goal is to simulate this real-world challenge and highlight how machine learning can be leveraged to make accurate pricing decisions. The core idea is to demonstrate the process of building, training, and evaluating regression models that can learn the relationships between laptop specifications and their market prices.

In this project, I will explore and compare the performance of various regression algorithms to determine which model offers the most accurate price predictions. Through proper preprocessing, feature engineering, model evaluation using metrics like RMSE and R^2 score, and tuning, this project will not only provide insight into laptop price prediction but also showcase the practical application of supervised learning in a consumer-focused industry.

Data Overview

This dataset is imported from the IBM Coursera Data Science Professional Certification course. The file is in CSV format and contains 12 columns that describe various specifications of laptops.

- The '**Manufacturer**' column indicates the brand or company that manufactures the laptop.
- '**Category**' describes the classification or type of laptop.
- The '**Screen'** column refers to the type of display used.

- ‘**GPU**’ represents the number of Graphic Processing Units present in the laptop.
- ‘**OS**’ refers to the Operating System installed.
- The ‘**CPU_Core**’ column indicates the number of central processing unit cores.
- ‘**Screen-size_cm**’ gives the screen size in centimeters.
- The ‘**CPU_frequency**’ column tells us how fast the processor can execute instructions, measured in gigahertz (GHz); higher frequency generally means better performance.
- The ‘**RAM_GB**’ column provides information about the size of the Random Access Memory in gigabytes—more RAM usually translates to better multitasking and performance.
- ‘**Storage_GB_SSD**’ represents the amount of Solid-State Drive storage available, again in gigabytes, and a higher value means more storage capacity.
- ‘**Weight_kg**’ indicates the weight of the laptop in kilograms.
- Lastly, the ‘**Price**’ column is our target variable, representing the cost of the laptop.

This dataset will be used to build a machine learning model by analyzing the relationships and trends among the features to accurately predict laptop prices. It is a structured and clean dataset suitable for regression analysis, with no missing contextual information or assumptions, making it ideal for training and evaluating regression models.

Methodology

In this project, we will follow a structured and step-by-step approach, starting from importing the dataset (in CSV format) from the IBM Cloud database. The process begins with understanding the dataset, identifying any missing values, and exploring the relationships and correlations among features using essential Python libraries such as Pandas and NumPy. This is followed by a detailed Exploratory Data Analysis (EDA), where we visualize trends and patterns in the data using Matplotlib and Seaborn. These visualizations help in identifying which features are significant, how they behave, and how they might impact the target variable. After developing a solid understanding of the data through EDA, we move on to the model development phase. Here, we split the data into training and testing sets for validation purposes and apply various regression models. Finally, we evaluate and refine the models by calculating their accuracy to determine which model performs best. This structured approach ensures a thorough analysis and reliable predictive performance for the laptop price prediction task.

Exploratory Data Analysis

The data has been imported and cleaned with now no missing values. Statistical Analysis is performed on the numeric data to understand the summary statistics by using ‘df.describe()' before going further it is important to understand the importance of this summary statistical information. This table gives the statistical information only for the numeric columns, this table contains 8 rows for each numeric column these are ‘count’ - gives the total number of values of the data, if the count is less it meaning there are missing values, ‘mean’ - describes the mean value, ‘std’ - standard deviation, how spread out the values are, ‘min’ minimum value, ‘25%’ 25% of values fall below this number, ‘50%’ 50% of values fall below this number, ‘75%’ - 75% of values fall below this number, ‘max’ - maximum value.

some insights that can be drawn are - if max is much higher than 75% or if min is way lower than 25%, it might indicate outliers. Standard deviation (std) tells you how much variability there is — high std means values vary widely. If mean ≈ median (50%), the data is likely symmetric. If not, the data might be skewed (e.g., right-skewed if mean > median).

	Category	GPU	OS	CPU_core	Screen_Size_cm	CPU_frequency	RAM_GB	Storage_GB_SSD	Weight_kg	Price
count	238.000000	238.000000	238.000000	238.000000	238.000000	238.000000	238.000000	238.000000	238.000000	238.000000
mean	3.205882	2.151261	1.058824	5.630252	37.269615	2.360084	7.882353	245.781513	1.862232	1462.344538
std	0.776533	0.638282	0.235790	1.241787	2.946184	0.411393	2.482603	34.765316	0.489090	574.607699
min	1.000000	1.000000	1.000000	3.000000	30.480000	1.200000	4.000000	128.000000	0.810000	527.000000
25%	3.000000	2.000000	1.000000	5.000000	35.560000	2.000000	8.000000	256.000000	1.472500	1066.500000
50%	3.000000	2.000000	1.000000	5.000000	38.100000	2.500000	8.000000	256.000000	1.862232	1333.000000
75%	4.000000	3.000000	1.000000	7.000000	39.624000	2.700000	8.000000	256.000000	2.200000	1777.000000
max	5.000000	3.000000	2.000000	7.000000	43.942000	2.900000	16.000000	256.000000	3.600000	3810.000000

Fig 1 - Statistical Summary of Numeric Variables

the figure above is the statistical summary for my numerical data, from this we can say that, there are no missing values. Most laptops have similar specs (8 GB RAM, 256 GB SSD, ~38cm screen, 2 GPUs, 1 OS). High-priced laptops likely correlate with higher CPU cores, frequency, RAM, and weight (suggesting gaming or workstations). **Target variable (Price)** is highly spread — preprocessing with normalization or log scaling might improve model performance. 'Storage_GB_SSD' have almost no variation, so this feature might not help in model training, we have to consider this during feature selection.

A correlation heat map on numeric data was analyzed, before diving into the insights let understand what is a correlation and how it helps - A correlation heat map visually represents the strength and direction of relationships between numerical features in a dataset. The main insight to draw is how each feature is correlated with the target variable (in this case, 'Price')—strong positive correlations (close to +1) indicate that as one variable increases, so does the other, while strong negative correlations (close to -1) show that one decreases as the other increases. Features with strong correlation to target are usually the most influential in prediction models. Additionally, the heat map helps us identify multicollinearity, where two or more features are highly correlated with each other, which can negatively impact certain models like linear regression. Features with low or no correlation might not add much predictive value, although this depends on the model type. Overall, the heat map helps in feature selection, understanding relationships, and detecting possible redundancies or data quality issues.

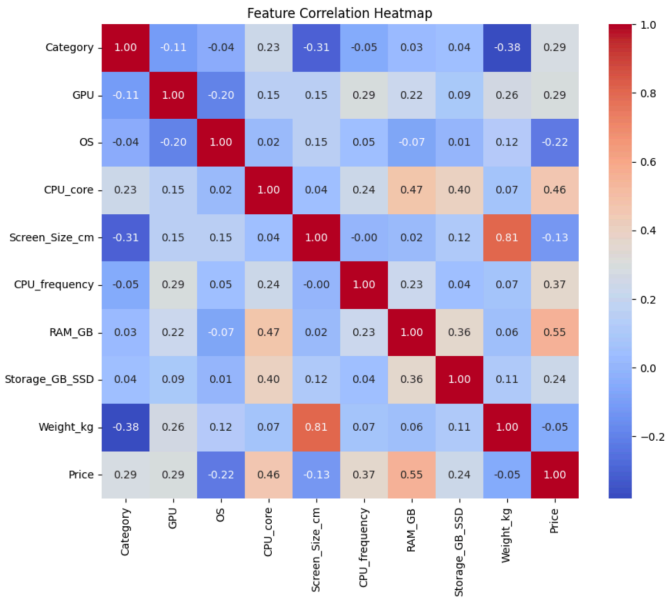

Fig 2 - Correlation Heat map of Numeric Variables

From the above heat map, we can clearly see that 'RAM_GB' and 'CPU_Frequency' directly proportional to the price, if these values increase then the price increases, then we have 'weight_kg', which is inversely proportional which means as the weight of the laptop increases then the price decrease and vise versa.

The price range of the laptops has been binned into three groups: **high**, **medium**, and **low**. It was observed that the dataset contains a higher number of laptops in the low and high price ranges, while there are relatively fewer laptops in the medium price range. This indicates that most laptops in the dataset are either budget-friendly or high-end, with fewer mid-range options.

To understand how key features influence price, scatter plots and box plots were performed. Using polynomial regression (n-degree curves), it was observed that features like **CPU_Core**, **RAM_GB**, and **CPU_frequency** fit well and show a clear pattern with price. However, the **Weight_kg** feature displayed a widely scattered distribution with several outliers, indicating that its relationship with price is less straightforward and needs careful handling during feature engineering (Fig 4).

For categorical features, box plots were used. In the **Manufacturer** category, brands like **Dell**, **HP**, and **Lenovo** showed a broad price range with notable outliers, while brands such as **Huawei**, **Razer**, and **Xiaomi** appeared to have a more consistent price range. Overall, the price distribution across manufacturers is quite varied and scattered, which is important to consider when encoding and modeling these features.

Regarding the **Screen** feature, the price distribution across different screen types was relatively consistent and centered around the median price range of $1300 to $1500. However, some outliers were present here as well.

Both the scatter plots and box plots, as attached below, help visualize these trends and guide the next steps in feature engineering and model selection.

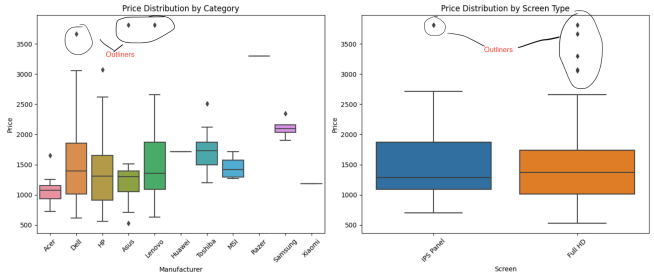

Fig 3- Box Plot of Price distribution by Manufacturer and Screen Type

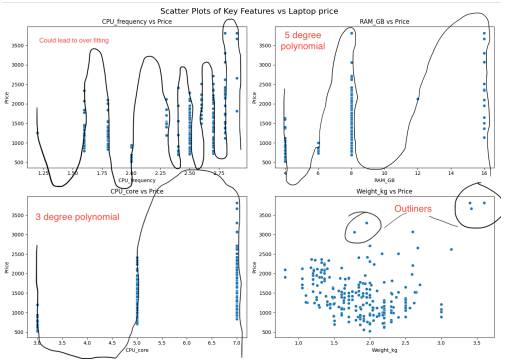

Fig 4 - Scatter Plot of Price Distribution for key features

Model Development and Evaluation

To accurately predict the laptop price the dataset has divided into train and test splits on 70 -30 split, 70% of data for training and 30 of data for testing. The train data is then fit with 10 regression models and predicted on test data, below fig - 5 give the full insights of each model it's R2 Score, MAE and RMSE. As the data is limited with only 238 values if we perform hyper parameter tuning then the model tends to overfitting, therefore in this project normal basic model fit has performed for accuracy analysis.

Among the 10 regression models tested on the laptop pricing dataset, Ridge Regression performed the best with the highest R^2 score of 0.86 and the lowest MAE and RMSE, indicating strong predictive accuracy and minimal large errors. Linear and Lasso Regression followed closely, performing nearly as well. These linear models are particularly well-suited for the small dataset (like ours with only 238 rows), as they generalize better without overfitting. Ensemble models like Gradient Boosting and Random Forest achieved decent results ($R^2 \approx 0.80–0.83$), but didn't outperform linear models—likely due to the limited data and absence of hyper parameter tuning. Adaboost underperformed, possibly due to its sensitivity to outliers. Poorly performing models included Decision Tree ($R^2 = 0.63$), which likely overfitted due to its tendency to create deep splits in small data, and SVR ($R^2 \approx -0.01$), which struggled likely due to poor scaling and lack of tuning. Polynomial Regression models (degree 2 and 3) drastically failed with extreme negative R^2 values, highlighting severe overfitting from excessive curve-fitting on such a small dataset. In conclusion, **Ridge, Linear, and Lasso are the most reliable choices** here, while more complex or high-variance models tend to overfit or underperform without enough data or tuning.

	Model	R2 Score	MAE	RMSE
3	Ridge Regression	8.597915e-01	1.376553e+02	1.775319e+02
0	Linear Regression	8.580715e-01	1.384262e+02	1.786175e+02
4	Lasso Regression	8.568263e-01	1.400237e+02	1.793993e+02
7	Gradient Boosting	8.324176e-01	1.537363e+02	1.940901e+02
6	Random Forest	8.026574e-01	1.690411e+02	2.106200e+02
8	Adaboost	7.616075e-01	1.898278e+02	2.314917e+02
5	Decision Tree	6.265599e-01	2.146528e+02	2.897341e+02
9	SVR	-1.040267e-02	3.854669e+02	4.765807e+02
2	Polynomial (Deg = 3)	-1.622569e+03	1.175328e+04	1.910401e+04
1	Polynomial (Deg = 2)	-5.821046e+22	3.552180e+13	1.143904e+14

Fig 5 - ML Models Results

Key Insights / Conclusion

The Laptop Prices dataset used in this project contains 238 entries, each representing a unique laptop with 12 features. During the exploratory data analysis phase, it was observed that certain features significantly influence the price of a laptop. Among them, RAM size (in GB), CPU frequency, CPU core, and the weight of the laptop had the most notable

impact. In general, **laptops with higher RAM, faster CPUs (e.g., frequencies of 2.5 GHz or more), and lighter weight tend to be priced higher.** Additionally, the distribution of prices was found to be right-skewed, meaning that most laptops are in the affordable price range, while a smaller number fall into the high-end category.

To predict the price of a laptop, several regression models were applied and evaluated based on **R^2 score, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).** Among the models tested, Ridge Regression, Linear Regression, and Lasso Regression performed the best, achieving high R^2 scores and low error values. These models demonstrated good predictive performance and generalization on the test data. Given the relatively small size of the dataset, hyperparameter tuning methods like GridSearchCV were intentionally avoided. Applying such techniques on limited data may lead to overfitting or underfitting, as the model could make incorrect assumptions due to insufficient variability in the dataset.

After comparing all models, **Ridge Regression** emerged as the most accurate and stable, making it the final choice for this project. It provides a reliable way to predict laptop prices based on key specifications without overfitting the data. With this model, new laptops' prices can be estimated effectively, helping consumers, retailers, or manufacturers understand pricing trends and make informed decisions. This concludes the project with a well-performing regression model that can be further refined or scaled if more data becomes available.

> **Thanks for stopping by! If you found this helpful or have suggestions, feel free to leave feedback here. Happy learning and exploring new data!** 👏