Kalpa Publications in Computing
Volume 19, 2024, Pages 61–72
Proceedings of 6th International Conference on Smart Systems and Inventive Technology

DEMATEL-Based Feature Selection for Improved Laptop Price Prediction

Amritha Gopakumar[1], Aathira Shine[2], Amina Ajim[3], and Anjali T[4]

[1] Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India
amrithagopakumarengoor@gmail.com
[2] Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India
aathirashine1999@gmail.com
[3] Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India
amina.ajimshah@gmail.com
[4] Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India
anjalit@am.amrita.edu

Abstract

Laptops have become an indispensable part of everyone's lives. They are beneficial for quickening the pace when it comes to any task. They serve the purpose of communication through online interactions and streamline other operations. The different kinds of laptops on the market have varying features designed precisely for your day to day needs. We have proposed a laptop price predictor that aims to predict the price of laptops. The results of the current study aid in understanding and identifying the factors that customers consider vital when purchasing a laptop. Further, the relationships amongst the important criteria were also identified. DEMATEL(Decision making trial and evaluation laboratory) method is employed to analyse the relationship between the variables and identify the most important factors. Mutual information regression (MIR) method was employed for feature selection and finally the selected features were input to the various Machine Learning and Deep learning algorithms for price prediction. Our study revealed that LightGBM Regressor and XGB Regressor performed the best, with the highest r2 scores of 0.81 after hyperparameter tuning.

1 Introduction

As a result of its convenience and portability, laptops have emerged as a necessary tool for both personal and professional use. The market for laptops has grown significantly in recent years, and users can now choose from a variety of models and brands at various price points. Although there are occasional swings, laptop prices have remained largely consistent thanks to the continual development of new technology and functionalities. In this analysis, we will examine the numerous aspects of laptop pricing, such as technological advancements, consumer demand, and manufacturer competitiveness. We will create a predictive model that will anticipate future laptop prices using previous pricing information and market patterns. We intend to offer useful insights into how laptop pricing may evolve over time by evaluating these elements. This investigation's main objective is to find a predictive model that can predict future costs and to provide a thorough understanding of the elements that affect laptop pricing. Making decisions regarding purchasing and making laptops can be aided by this knowledge for both customers and manufacturers. Because price is typically dependent on a variety of distinguishing features and elements, accurate laptop price forecast requires specialist expertise. Typically, the most significant ones are brand and model, RAM, ROM, GPU, CPU, etc. A variety of machine learning and deep learning techniques can be used to solve the complicated challenge of predicting laptop prices. We can better understand the underlying elements that affect laptop costs and make more precise and dependable predictions by investigating various methods and approaches. Both consumers and manufacturers may gain greatly from this in practical terms. In this paper, we applied different Machine learning [1], and deep learning methods and techniques in order to achieve higher precision of the used laptop price prediction. To perform laptop price prediction using machine learning and deep learning techniques [8], we need a dataset that contains laptop prices and features. The features can include specifications such as Screen resolution, RAM, memory, GPU and so on. The DEMATEL method and Mutual Information Regression method are employed for identifying the best features that can be used for laptop price prediction.

2 Related Work

A survey-based study was carried out by Vishesh et al.[5] to determine the main factors influencing customers' decisions to purchase laptops.They employed a systematic questionnaire to gather information from 250 laptop users in India. . Product features, brand image, after-sales service, and price were divided into four categories using factor analysis to find the important selection criteria. The findings indicated that product features, followed by brand image, after-sales support, and pricing, were the most crucial factors for laptop consumers.In order to look into the association between demographic traits and the important decision criteria, they also performed a chi-square test. The findings revealed a substantial correlation between the main decision factors and gender, age, income, and education level.The goal of Sankalp Lohakare et al.[3] is to forecast laptop prices using machine learning methods. They started by gathering a dataset with details on several laptop models, including their brand, processor type, screen size, storage capacity, and other characteristics.. The price of laptops was then predicted using four different machine learning algorithms: SVR, dt, random forests, and linear regression. They used metrics like mean squared error, mean absolute error, and R-squared to assess each algorithm's performance. With an R-squared value of 0.93, the results proved that the random forest algorithm outperformed better other three algorithms. This indicates that 93% of the variability in the data could be explained by the model. Using machine learning techniques, they were able to accurately anticipate the cost of laptops. In the context of eBay online trading, Ilya Igorevitch Raykhel et al.[11] want to create a model for real-time price prediction.The paper begins by outlining the issue of price volatility in online trading and the need of making accurate real-time price predictions. It then suggests a hybrid model for price prediction that combines statistics and machine learning methods. The hybrid model predicts prices in real-time by combining principal component analysis , multiple linear regression, and SVM. SVM was employed as the final prediction model after the dataset's relevant features and dimensionality were extracted using the multiple linear regression and PCA approaches. The analysis of the data revealed that the hybrid model performed better than the individual models and had a high level of prediction accuracy. Additionally, the model has the flexibility to make predictions in real time and adjust to shifting market conditions. Ikhlas F. Zamzami [4] wants to use 5G technology to create a deep learning model that can detect odd behaviour in elderly people. The study opens by emphasising the significance of elderly care and the difficulties in spotting aberrant behaviour in elderly people. In order to forecast atypical behaviour, the research suggests a deep learning-based model that incorporates sensor data from wearable devices. He gathered a collection comprising sensor data, such as temperature, heart rate, and accelerometer data, from wearable devices worn by older people.The model predicts atypical behaviour in elderly people using a combination of LSTM and CNN architectures. While the CNN architecture was utilised to extract spatial characteristics from the sensor data, the LSTM architecture was employed to model the temporal dependencies in the sensor data. In this study, the DEMATEL approach was also applied. Using DEMATEL and DLT, the most significant behavioural traits of elderly people were found. The findings demonstrated that the suggested model was highly accurate at predicting aberrant behaviour in elderly people. The model was also able to spot tiny alterations in behaviour patterns that can portend the beginning of health issues, one of which is depression, which is among the most prevalent. The research demonstrates that a model like this can be helpful for early detection of health issues in elderly people, enabling prompt intervention and care. The LSTM model with with 96% accuracy was found to be the best approach. To create a machine learning model for forecasting laptop costs, Astri Dahlia Siburian et al.[9] suggests a model that employs various regression techniques. To forecast laptop prices, the model combines linear regression, decision tree regression, and random forest regression algorithms. While decision tree regression and random forest regression methods were employed to capture non-linear correlations and interactions between features, the linear regression algorithm was utilised to describe the linear relationship between features and pricing.According to the findings, the suggested model performed better than the individual regression methods and had a high level of prediction accuracy.Additionally, the model was able to pinpoint the most crucial factors that influence laptop prices, such as processor speed, RAM, and storage capacity.

3 Proposed Methodology

This study's objective is to find an accurate model which can predict the price of a laptop based on its various features such as CPU, RAM, memory, screen resolution etc. To address this problem, we will use a dataset of laptops currently available in the market and apply feature selection methods such as DEMATEL and Mutual Information Regression to identify the most important features for price prediction. We will then apply various machine learning and Deep learning techniques to create a prediction algorithm that uses these variables to determine a laptop's pricing. The model will be evaluated using the standard regression metric R-squared and we will aim to achieve the highest possible performance.The successful development of this laptop price prediction model will have several benefits for consumers.

3.1 Dataset

The "Laptop Company Price List for Regression" dataset sourced from Kaggle comprises 1320 rows and 12 columns, with the target variable being the price in euros. Although the specific details of the dataset's features are not provided, it can be inferred that the dataset contains various attributes or characteristics of laptops, such as brand, processor type, RAM size, storage capacity, screen size, operating system, etc. These features likely play a crucial role in determining the price of a laptop. With the dataset size and target variable identified, further exploration and analysis, along with appropriate preprocessing techniques as mentioned earlier, can be applied to prepare the data for regression modeling tasks aimed at predicting laptop prices accurately. price_euros is the target variable. Tab. **??** displays an overview of the characteristics of the dataset in detail.

3.2 Data Preprocessing

The data preprocessing pipeline commenced with the meticulous removal of duplicate entries and handling of missing data to maintain the dataset's integrity and quality. Following this, normalization was applied using the standard scaler method to standardize the range of features, ensuring equitable contribution to the model and preventing any single feature from overshadowing others due to its scale. Feature selection was then executed through a multi-step process. Initially, the DEMATEL method was employed to discern highly correlated features with the target variable, specifically the laptop price,

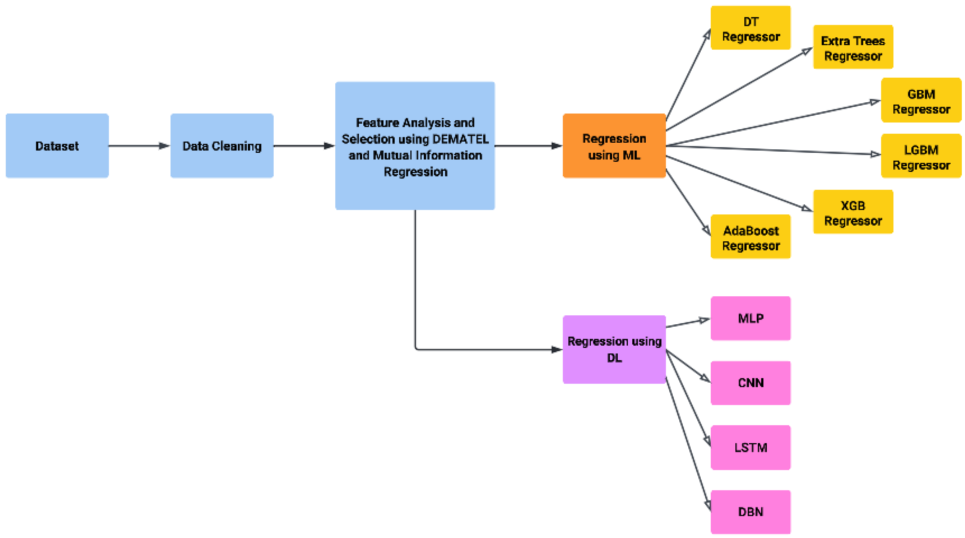

Figure 1: Proposed Model

Attributes	Description
Company	Laptop Manufacturer
Product	Brand and Model
TypeName	Type (Notebook, Ultrabook, Gaming, etc.)
Inches	Screen Size
ScreenResolution	Screen Resolution
Cpu	Central Processing Unit (CPU)
Ram	Laptop RAM
Memory	Hard Disk / SSD Memory
GPU	Graphics Processing Units (GPU)
OpSys	Operating System
Weight	Laptop Weight
price_euros	Price (Euro)

Table 1: Description of properties

offering insights into the causal relationships between variables and aiding in the identification of influential features. Additionally, mutual information regression was leveraged to assess the information gain of each feature concerning the target variable, thereby identifying the most relevant features while filtering out irrelevant or redundant ones. Finally, to accommodate categorical variables, one-hot encoding was utilized to transform them into binary values, enabling the model to comprehend and utilize them effectively. This comprehensive preprocessing approach not only enhanced the interpretability of the dataset but also contributed significantly to improving model accuracy and performance during subsequent training and evaluation phases.

3.3 DEMATEL

One of the techniques utilised for feature selection was the multicriteria decision-making process known as DEMATEL, or Decision-Making Trial and Evaluation Laboratory. It is a methodical strategy that lessens the complexity of the issue by assisting in the identification of linkages between the criteria. It is employed to assess the relative significance of the dataset's attributes and to pinpoint those that had a strong correlation with laptop costs. With the help of this approach, which is based on graph theory and matrix algebra, we may find and examine the direct and indirect links between the features.

Step 1: Construct the initial direct-relation matrix $[\mathrm{m_{ij}}]_{\mathrm{n}\times n}$

By averaging all individual N ratings ($\mathrm{m_k}$) using Eqs. (1) and (2), the individual feature pairwise rating ($\mathrm{m_{ij}}$) value is generated, which indicates the extent to which the factor i impacts the factor j. The resultant average table is presented in Fig. 2

$$m_{\mathrm{ij}} = \frac{\Sigma_k^N m_{\mathrm{k}}}{N} \tag{1}$$

$$M = [m_{\mathrm{ij}}]_{\mathrm{n\times n}} = \frac{1}{\mathrm{N}}\sum_{q=1}^{H}\left[x_{\mathrm{ij}}^{\mathrm{k}}\right]_{\mathrm{n\times n}} \tag{2}$$

	Company	Product	Type Name	Inches	Screenresolution	CPU	RAM	Memory	GPU	Opsys	Weight
Company	0	2	2	1.333333333	2.333333333	2	2.333333333	2.333333333	1.666666667	1	1.333333333
Product	3	0	3.333333333	1.666666667	3	2.666666667	3	3.333333333	2	2.333333333	2.333333333
Type Name	1.666666667	1.333333333	0	1.333333333	2.333333333	2	3.333333333	2	2.666666667	1.666666667	4
Inches	2	1.666666667	2	0	1.333333333	1.333333333	1	0	1.333333333	0.3333333333	2.666666667
Screenresolution	3.333333333	3.333333333	3.333333333	2.666666667	0	2.333333333	3.333333333	1	1.666666667	2	2
CPU	1.333333333	1	1.666666667	1.666666667	3.666666667	0	2	2.333333333	2.333333333	3	2
RAM	1.666666667	1.666666667	2	0	2	2.666666667	0	0.6666666667	2	2.666666667	1.666666667
Memory	2	1	2	0.6666666667	1	3	2.333333333	0	2	1.666666667	1.666666667
GPU	1.333333333	2	1.666666667	0.6666666667	3.666666667	3	3	3	0	2	0.3333333333
Opsys	2	0.6666666667	0.6666666667	0	0	0.6666666667	0	0	0	0	0.6666666667
Weight	1.666666667	1.666666667	1.666666667	0.6666666667	1	1.333333333	0.3333333333	2	1.333333333	0.3333333333	0

Figure 2: Average Direct Relation Matrix (M)

Step 2: Normalize the initial direct-relation matrix

The average matrix M is normalised in the second step to transform the direct matrix into a normalised matrix by using the formula R=M/s where

$$s = max\left[\max_{1\leq i\leq n}\sum_{j=1}^{n} m_{\mathrm{ij}},\ \max_{1\leq i\leq n}\sum_{i=1}^{n} m_{\mathrm{ij}}\right] \tag{3}$$

The calculated R matrix is presented in Fig. 3

	Company	Product	Type Name	Inches	Screenresolution	CPU	RAM	Memory	GPU	Opsys	Weight
Company	0	0.07499999999	0.07499999999	0.04999999999	0.08749999999	0.07499999999	0.08749999999	0.08749999999	0.06249999999	0.0375	0.04999999999
Product	0.1125	0	0.125	0.06249999999	0.1125	0.09999999999	0.1125	0.125	0.07499999999	0.08749999999	0.08749999999
Type Name	0.06249999999	0.04999999999	0	0.04999999999	0.08749999999	0.07499999999	0.125	0.07499999999	0.09999999999	0.06249999999	0.15
Inches	0.07499999999	0.06249999999	0.07499999999	0	0.04999999999	0.04999999999	0.0375	0	0.04999999999	0.0125	0.09999999999
Screenresolution	0.125	0.125	0.125	0.09999999999	0	0.08749999999	0.125	0.0375	0.06249999999	0.07499999999	0.07499999999
CPU	0.04999999999	0.0375	0.06249999999	0.06249999999	0.1375	0	0.07499999999	0.08749999999	0.08749999999	0.1125	0.07499999999
RAM	0.06249999999	0.06249999999	0.07499999999	0	0.07499999999	0.09999999999	0	0.025	0.07499999999	0.09999999999	0.06249999999
Memory	0.07499999999	0.0375	0.07499999999	0.025	0.0375	0.1125	0.08749999999	0	0.07499999999	0.06249999999	0.06249999999
GPU	0.04999999999	0.07499999999	0.06249999999	0.025	0.1375	0.1125	0.1125	0.1125	0	0.07499999999	0.0125
Opsys	0.07499999999	0.025	0.025	0	0	0.025	0	0	0	0	0.025
Weight	0.06249999999	0.06249999999	0.06249999999	0.025	0.0375	0.04999999999	0.0125	0.07499999999	0.04999999999	0.0125	0

Figure 3: Normalized Matrix (R)

Step 3: Find the total relation matrix T
Eq. (4) defines T

$$T = R(I - R)^{-1} = [t_{\mathrm{ij}}]_{\mathrm{n}\times\mathrm{n}} \tag{4}$$

Fig. 4 displays the calculated total relational matrix.

	Company	Product	Type Name	Inches	Screenresolution	CPU	RAM	Memory	GPU	Opsys	Weight
Company	0.1578625593	**0.2022232293**	**0.2323787665**	0.137002945	**0.2458072187**	**0.2390779589**	**0.2507721551**	**0.2175531183**	**0.1988646223**	0.1787307266	**0.1967355638**
Product	**0.3192481261**	0.1813576834	**0.3347763926**	0.1805444425	**0.3264928499**	**0.3225450281**	**0.3337420416**	**0.3004000783**	**0.2615513029**	**0.2746353821**	**0.2860491059**
Type Name	**0.2371359789**	**0.1994805674**	0.1827318029	0.1458720114	**0.266553266**	**0.2611593343**	**0.3009846045**	**0.2255032449**	**0.2494646411**	**0.2181450204**	**0.3023641893**
Inches	0.1882307805	0.1603145898	0.192612111	0.0689300206	0.1736209097	0.1722877242	0.1626800638	0.1093645263	0.1531748683	0.1160687324	**0.2069862229**
Screenresolution	**0.31985338**	**0.2865107458**	**0.3253403872**	**0.2085448133**	**0.2163162861**	**0.2982071264**	**0.3324275191**	**0.2154250883**	**0.2410364408**	**0.2529098981**	**0.2678836455**
CPU	**0.2183682548**	0.1796871715	**0.2305527958**	0.1536487314	**0.2954329982**	0.1777660451	**0.2467617012**	**0.2214735459**	**0.2257737462**	**0.2531677339**	**0.2254948291**
RAM	**0.1995287438**	0.1761451554	**0.2121287183**	0.08167482203	**0.2182403669**	**0.2399919226**	0.1491895389	0.1491105331	0.1924314578	**0.221572247**	0.1888318038
Memory	**0.2095437381**	0.1540399531	**0.2129473849**	0.1033435991	0.1882959211	**0.2549127797**	**0.2317712191**	0.1251916149	0.1965306702	0.1893720964	0.1914739159
GPU	**0.2262774605**	**0.2181361413**	**0.2409756274**	0.1259213067	**0.3091975248**	**0.2929497198**	**0.2942251089**	**0.2531679724**	0.1558497859	**0.2345235059**	0.1788235811
Opsys	0.1103500569	0.05788144912	0.06536492347	0.02444092831	0.04439795465	0.06605244852	0.04412575361	0.03919475172	0.03693268398	0.03476968361	0.06265317345
Weight	0.1656622389	0.1480628549	0.169399648	0.08656311215	0.1500174161	0.1633936568	0.1312253317	0.1677538446	0.1439238023	0.1086470284	0.1020121224

Figure 4: Total Relation Matrix (T)

Step 4: Determine the sum of rows and sum of columns of T
Eqs. (5) and (6) are used to calculate the vectors r and c, which respectively represent the sum of rows and columns.

$$r = \left[\left(\sum_{j=1}^{n} t_{\mathrm{ij}}\right)\right]_{\mathrm{n}\times 1} = [t_{\mathrm{i}}]_{\mathrm{n}\times 1} \tag{5}$$

$$c = \left[\left(\sum_{i=1}^{n} t_{\mathrm{ij}}\right)\right]_{1\times\mathrm{n}} = [t_{\mathrm{j}}]_{1\times\mathrm{n}} \tag{6}$$

The resultant matrix is provided in Fig 5. The magnitude of each criterion's relevance is represented by the vector r+c, which is also known as prominence. In contrast, the vector r-c, commonly referred to as relation, has the capacity to classify the variables into cause-and-effect groups. The criteria is recognised as a cause if the relation is positive. On the other hand, if the relation is negative, the criteria is recognised as an effect.

	R	C	R+C	R-C		Rank
Company	2.257008864	2.352061318	4.609070182	-0.09505245378	EFFECT	5
Product	3.121342433	1.963839541	**5.085181975**	1.157502892	CAUSE	2
Type Name	2.589394661	2.399208558	4.988603219	0.1901861031	CAUSE	3
Inches	1.70427055	1.316486733	3.020757282	0.387783817	CAUSE	10
Screenresolution	2.964455331	2.434372712	**5.398828043**	0.5300826185	CAUSE	1
CPU	2.428127553	2.488343745	4.916471298	-0.06021619111	EFFECT	4
RAM	2.02884531	2.477905038	4.506750347	-0.449059728	EFFECT	7
Memory	2.057422892	2.024138319	4.081561211	0.03328457368	CAUSE	8
GPU	2.530047735	2.055534022	4.585581756	0.474513713	CAUSE	6
Opsys	0.5861638073	2.082542055	2.668705862	-1.496378248	EFFECT	11
Weight	1.536661056	2.209308153	3.745969209	-0.672647097	EFFECT	9

Figure 5: Ranking summary of features

Step 5: Choose a threshold value (h).
To simplify the system and highlight the most significant effects, a threshold value is employed to eliminate insignificant effects. The threshold value is found using Eq. (7).

$$h = \frac{\Sigma_{i=1}^{n} \Sigma_{j=1}^{n} t_{\mathrm{ij}}}{N} \quad (7)$$

The h value for our problem is 0.1967251256.

Step 6: Create a relationship diagram

The threshold value is compared to the values in the total-relation matrix T and the causal chart is drawn. The resulting chart as shown in Fig. 6 displays the causal relationships between factors in the system and can aid in identifying the most significant causes and effects.

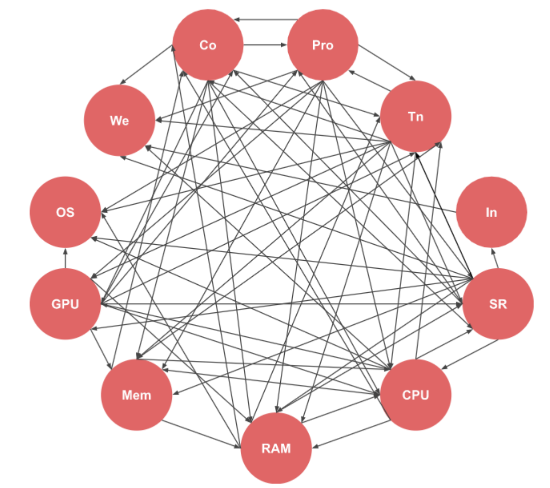

Figure 6: Causal chart

According to the findings of DEMATEL analysis, Screen resolution is the top influential factor with the highest (r+c) value as seen in Fig. 5. The other important factors are the Product, Type name, CPU and Company. These are the factors that greatly influence the other decision factors the most. Weight, on the other hand, has the biggest negative (r-c) value and is heavily impacted by other decision-making attributes. The features like Screen resolution, Product, Type Name, GPU, Memory, Inches are part of the cause group. CPU, Company, RAM, Weight and Opsys are members of the effect group.

3.4 Mutual Information Regression

Mutual information regression (MIR) is a feature selection technique frequently employed in statistical and machine learning applications [7]. By evaluating the amount of information that two variables share with one another, it is a statistical technique used to infer the link between two variables. By assessing the mutual information between each feature and the target variable, it seeks to determine the most informative features in a dataset. A measure of how much knowledge one variable offers about the other is called the mutual information between two variables.

The mutual information between feature and target variable is computed in MIR. The most informative features are those with the high mutual information ratings. The target variable is then predicted using a regression model that is constructed using these chosen features.

Using MIR as a feature selection method has the benefit of being able to handle both continuous and discrete variables. Additionally, it can handle high-dimensional data, capture non-linear and non-monotonic connections between variables, and is resistant to noise and outliers.

```
Cpu                 0.563153
Ram                 0.506978
Memory              0.506093
Gpu                 0.433819
TypeName            0.307423
Weight              0.251338
ScreenResolution    0.250832
Inches              0.223319
Product             0.188555
Company             0.133487
OpSys               0.107845
dtype: float64
```

Figure 7: MIR Values

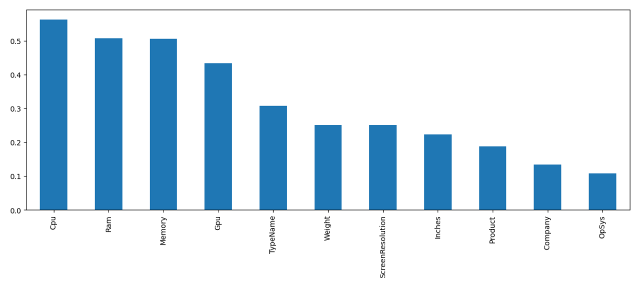

Figure 8: MIR Bar plot

After applying mutual information regression on our dataset, we got the most important features such as CPU, RAM, Memory, GPU etc. as in Figs. 7 and 8. We eliminated two of the least important features Opsys and Company and used the rest to build a regression model.

3.5 Using Machine Learning

We decided to employ machine learning for our project because it can analyse massive volumes of complicated and dynamic data, facilitate data-driven decision making, give adaptability and flexibility, provide competitive advantages, and be affordable and scalable [6]. We assessed the quality of fit of our regression model using the R2 score matrix. Hyperparameter tuning using Grid Search and Randomized Search was also done in case of Machine learning algorithms for getting the optimal parameters which resulted in increased performance of the model and helped to avoid overfitting. XGBoost, Gradient Boosting Regressor, LightGBM, Extra Trees Regressor, Decision Tree Regressor, and AdaBoost Regressor were some of the machine learning algorithms we tested on our preprocessed data after implementing the fundamental steps of data preparation, feature selection, and splitting the preprocessed data into training and testing sets.

3.6 Using Deep Learning

As Deep Learning can use ANN to automatically learn complicated features and representations from data to forecast laptop prices, it is the method we have chosen. It can scale to big datasets, captures nonlinear interactions, and enables flexibility in model building. We have also used the R2 score here to assess the quality of fit of our regression model. On our preprocessed data, we experimented with a variety of deep learning algorithms, including MLP, CNN, LSTM, and DBN.

Table 2: Comparison of Machine Learning Models

Machine Learning Models	R2 Score
XGB Regressor	0.774
Gradient Boosting Regressor	0.755
LightGBM Regressor	0.791
Extra Trees Regressor	0.785
Decision Tree Regressor	0.601
AdaBoost Regressor	0.595

Table 3: Comparison of Deep Learning Models

Deep Learning Models	R2 Score
MLP	0.647
LSTM	0.647
CNN	0.712
DBN	0.576

4 Experimental Results

To evaluate the effectiveness of the different Machine learning and Deep Learning models and find out the best model , we applied them on our data. The results of this experiment are displayed in Tabs. 2 and 3.

We also did hyperparameter tuning of the ML models using Grid Search and Randomized Search to further improve the execution of the models and results of this are shown in Tab. 4.

In our study, we found that the machine learning models[2], particularly the LGBM Regressor and XGB Regressor, outperformed other models and achieved the highest r2 scores of 0.81, followed closely by the Extra Trees Regressor model with a r2 score of 0.78. The best hyperparameters obtained for LGBM Regressor, XGB Regressor and Extra Trees Regressor are shown in Figs. 9, 10 and 11 respectively.

The fact that the LGBM Regressor and XGB Regressor models are both gradient boosting models, which are well-known to be strong and efficient for regression tasks, could be one explanation for their higher performance. These models can sequentially add weak learners to the ensemble, each time

Table 4: Machine Learning Models after Hyperparameter tuning

Machine Learning Models after Hyperparameter Tuning	R2 Score
XGB Regressor	0.816
Gradient Boosting Regressor	0.773
LightGBM Regressor	0.817
Extra Trees Regressor	0.785
Decision Tree Regressor	0.605
AdaBoost Regressor	0.67

```
Best hyperparameters: {'learning_rate': 0.1, 'n_estimators': 403, 'num_leaves': 98}
```

Figure 9: Best hyperparameters for LGBM Regressor

```
Best hyperparameters: {'colsample_bytree': 0.5, 'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 410, 'subsample': 1}
```

Figure 10: Best hyperparameters for XGB Regressor

```
Best hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'n_estimators': 890}
```

Figure 11: Best hyperparameters for Extra Trees Regressor

focusing on the mistakes produced by the prior weak learners to improve the model's predictions.

Additionally, the XGB Regressor model and the LGBM Regressor are both capable of handling high-dimensional data and can function effectively even when there are a lot of features. Because we had a reasonable amount of features to take into account in our investigation, this might have been especially significant.

It is significant to remember that the Extra Trees Regressor model worked admirably and obtained a decent r2 score of 0.78, despite having a lower r2 score than the LGBMRegressor and XGBRegressor models. This model, which incorporates several decision trees, is an example of an ensemble model and can be especially useful for handling noisy data.

In conclusion, our research reveals that gradient boosting models, in particular the LGBM Regressor and XGB Regressor, may be well suited for forecasting laptop pricing and may be useful tools for both businesses and consumers. When choosing the best model, as with any modelling approach, it is essential to consider the unique characteristics and requirements of the dataset and the relevant problem.

Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP) models with r2 scores of 0.71 and 0.64, respectively, fared the best among the deep learning models [10]. The MLPRegressor function from the scikit-learn library was used to implement MLP. We utilised an MLP model with two hidden layers, each having 50 neurons. The model summaries of CNN, LSTM and DBN are shown in Figs. 12, 13 and 14.

It is important to notice that, despite our dataset consisting of numerical data and not visual data, CNN, which is primarily used for picture and video data analysis, outperformed the other deep learning models in our study. This might occur as a result of the fact that CNNs are built to extract features from high-dimensional inputs, which may be advantageous for finding patterns and relationships in the numerical data in our dataset.

5 Conclusion

In this study, the goal was to predict laptop prices using machine learning algorithms. We first performed data cleaning to ensure the dataset was of high quality and free from errors. We then applied the DEMATEL method to identify the relationships between the variables and determine the importance of the features. This analysis helped us to understand the underlying factors that contribute to laptop prices. Next, we used the mutual information regression (MIR) method to perform feature selection and eliminated two features from the dataset. This helped us to reduce the complexity of the model and improve its accuracy. We then applied various machine learning and deep learning techniques and then evaluated their performance using r2 score. Our analysis revealed that LightGBM Regressor and XGB Regressor performed the best, with the highest r2 scores of 0.81 after hyperparameter tuning.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 7, 16)             64

 conv1d_1 (Conv1D)           (None, 5, 32)             1568

 conv1d_2 (Conv1D)           (None, 3, 64)             6208

 flatten (Flatten)           (None, 192)               0

 dense_3 (Dense)             (None, 64)                12352

 dense_4 (Dense)             (None, 32)                2080

 dense_5 (Dense)             (None, 16)                528

 dense_6 (Dense)             (None, 1)                 17

=================================================================
Total params: 22,817
Trainable params: 22,817
Non-trainable params: 0
_________________________________________________________________
```

Figure 12: Model Summary of CNN

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 9, 64)             16896

 lstm_1 (LSTM)               (None, 32)                12416

 dense (Dense)               (None, 32)                1056

 dense_1 (Dense)             (None, 64)                2112

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 32,545
Trainable params: 32,545
Non-trainable params: 0
_________________________________________________________________
```

Figure 13: Model Summary of LSTM

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_7 (Dense)             (None, 64)                640

 p_re_lu (PReLU)             (None, 64)                64

 dense_8 (Dense)             (None, 32)                2080

 p_re_lu_1 (PReLU)           (None, 32)                32

 dense_9 (Dense)             (None, 1)                 33

=================================================================
Total params: 2,849
Trainable params: 2,849
Non-trainable params: 0
_________________________________________________________________
```

Figure 14: Model Summary of DBN

In our study, we found that the machine learning models outperformed the deep learning models for predicting laptop prices. While both types of models showed promising results, the machine learning models achieved higher r2 scores than the deep learning models. One possible reason for this could be that the machine learning models were better suited to our dataset, which included a relatively small number of features and a moderate amount of data. Generally, huge datasets are required for deep learning models to train effectively, which may have limited their performance in our study. Additionally, the machine learning models are generally easier to train and interpret than deep learning models, which can require more complex architectures and longer training times. This can make it easier to fine-tune the hyperparameters of machine learning models, which can lead to better performance.

References

[1] Amritha Gopakumar; Aathira Shine; T Anjali. Analysis of alcoholic eeg signal using semantic technologies. 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2023.

[2] Likhith Prasanth; Deepak Sreekanthan; D Ajana Lakshmi; Gayathri Harikumar; M P Vissutha; T Anjali. Intelligent traffic control system using wsn: A perspectivek. Fourth International Conference on Microelectronics, Signals and Systems (ICMSS), 2021.

[3] V. Surjuse; S. Lohakare;A. Barapatre; A. Chapke. Laptop price prediction using machine learning. International Journal of Computer Science and Mobile Computing (IJCMC), 2022.

[4] Zamzami; I. F. The key criteria for predicting unusual behavior in the elderly with deep learning models under 5g technology, 2023.

[5] Vishesh;S. Bali;S. Mittal. A study of key decision criteria for buying decision making process with reference to smart phones using dematel. ISSN, 2019.

[6] Chethan Dev;Kripa Kumar;Arjun Palathil;T. Anjali;Vinitha Panicker. Machine learning based approach for detection of lung cancer in dicom ct image, 2019.

[7] Hemalatha Thirugnanam;Maneesha Vinodini Ramesh;Venkat P. Rangan. Enhancing the reliability of landslide early warning systems by machine learning, 2020.

[8] K. S. Aswin; Manav Purushothaman; Polisetty Sritharani; Angel T. S. Ann and deep learning classifiers for bci applications. Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), 2022.

[9] A. D. Siburian;D. R. H. Sitompul; D. J. Z. E. I. Stiven Hamonangan Sinurat Andreas Situmorang;Ruben. Laptop price prediction with machine learning using regression algorithms. Jurnal Sistem Informasi dan Ilmu Komputer Prima (JUSIKOM Prima), 2022.

[10] Rahan Manoj; S Abhishek; Anjali T. A strategy for identification and prevention of crime using various classifiers. 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2022.

[11] I. Raykhel; D. Ventura. Real-time automatic price prediction for ebay online trading, 2009.