Gyalpozhing College of Information Technology
Royal University of Bhutan
Kabjisa, Chamjekha, Thimphu

School of Computing
Bachelor of Computer Science

Machine Learning Project

Spring Semester 2024

Laptop Price Prediction

Submitted by

NELAM RAI (12210065)
JIGME DEMA (12210053)
KINLEY NAMGAY (12210059)
YESHI DORJI (12210040)

Supervisor: Ms. Pema Yangden

1. Introduction

The laptop has become an important item in everyday life, but it can be difficult to sell and choose. Laptops come in all sorts of prices, features, and specifications. Price estimation and prediction are an important part of consumer strategy. Deciding on the correct price of a product is very important for the market success of a product. A laptop price prediction model is needed to predict the price based on user input such as brand, processor, memory, and storage.

i. Aim

To develop a machine learning model for accurate laptop price prediction

ii. Goal

To develop a machine learning model and accurately predict the price of the laptop.

iii. Problem Statement

Laptop prices tend to differ due to the rising demand from workers, students, and gamers. Laptop prices are subject to significant variations based on various factors such as their specifications, brand, and other variables. Customers may face challenges in assessing the reasonable market value of a laptop or determining if they are receiving a reasonable deal. So, our platform will offer the solution to purchase a laptop at a reasonable price.

iv. Scope

a) System scope:

The system will predict the price of the laptop.

b) User scope:

The price prediction will be targeted at all the customers who want to purchase a laptop.

2. Literature Review

a) Keepa:

Keepa is a web application that provides price history charts and price drop alerts for millions of products on Amazon. The Keepa website and browser extension offer a range of features that can help you track prices and make informed purchasing decisions. With Keepa, you can view the price history of a product over a specific period of time, ranging from a few days to several years. The price history chart shows the lowest and highest prices, as well as the average price, for the product during the selected time period. You can also view the price history for different Amazon marketplaces, such as Amazon US, Amazon UK, and Amazon Germany. Keepa also offers price drop alerts. Keepa offers a free version with limited features, as well as a paid version with additional features such as advanced price history charts and alerts for out-of-stock products. Keepa is a popular tool among Amazon shoppers and can be a useful resource for predicting the future price of a laptop or other products on Amazon.

b) **PriceSpy**:

PriceSpy is a website and app that allows you to compare prices of products from various online retailers. The website was founded in 2000 in Sweden, and it has since expanded to other countries, including the UK, Australia, and New Zealand. PriceSpy is popular among shoppers looking to find the best deals on products such as laptops, smartphones, and home appliances. The website has a large database of products from thousands of retailers, making Page 3 it easy to find the best prices and deals. One of the key features of PriceSpy is the price history graph, which shows you how the price of a product has changed over time. This information can be helpful in determining whether the current price is a good deal or if you should wait for a price drop. In addition to comparing prices, PriceSpy also provides user reviews, product specifications, and ratings to help you make an informed decision. You can also create price alerts for specific products, which will notify you when the price drops to a certain level.

PriceSpy is free to use, and it earns revenue through affiliate links. When you click on a link to a retailer and make a purchase, PriceSpy earns a commission. Overall, PriceSpy is a useful tool for anyone looking to save money on their online purchases.

c) Buyvia:

Buyvia is a mobile app and website that offers price comparisons and alerts for a wide range of products, including laptops. The site provides users with access to real-time pricing information from multiple retailers, allowing them to easily compare prices and find the best deal.

3. Dataset

The dataset we collected has 1303 rows and 12 columns. It has features like Unnamed, Company, TypeName, Inches, ScreenResolution, Cpu, Ram, Memory, Gpu, OpSys, Weight, and the target column/feature Price.

```
df=pd.read_csv('laptop_data.csv')
df.head()
```

]:

	Unnamed: 0	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	71378.6832
1	1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	47895.5232
2	2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	30636.0000
3	3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	135195.3360
4	4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	96095.8080

```
df.shape
```

]: (1303, 12)

During the preprocessing steps, we extracted important information from the columns into new columns and dropped the columns after extracting the information. After the preprocessing steps were done, we were left with 15 features including the target feature.

Our final dataset:

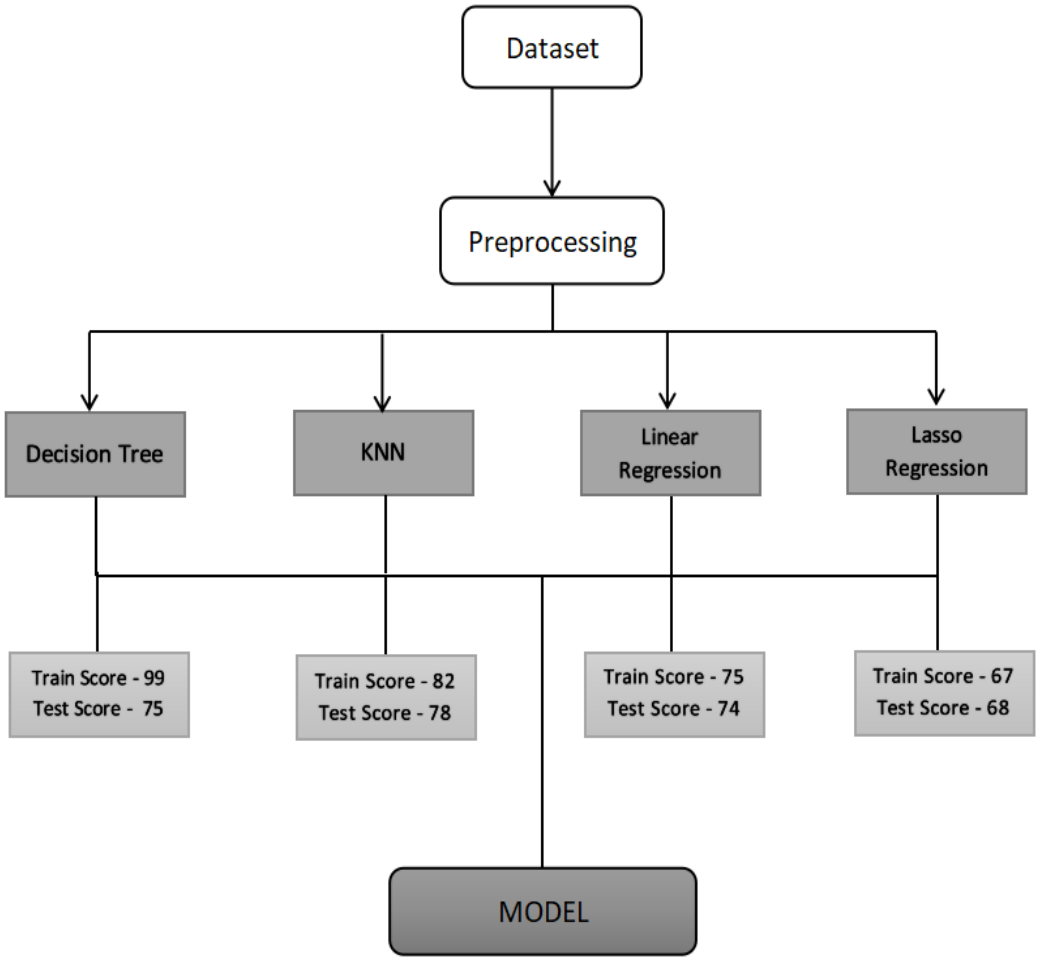
	Company	TypeName	Ram	Weight	Price	TouchScreen	Ips	ppi	Cpu Brand	HDD	SSD	Hybrid	Flash_Storage	Gpu brand	os
0	Apple	Ultrabook	8	1.37	71378.6832	0	1	226.983005	Intel Core i5	0	128	0	0	Intel	Mac
1	Apple	Ultrabook	8	1.34	47895.5232	0	0	127.677940	Intel Core i5	0	0	0	128	Intel	Mac
2	HP	Notebook	8	1.86	30636.0000	0	0	141.211998	Intel Core i5	0	256	0	0	Intel	Others/No Os/Linux
3	Apple	Ultrabook	16	1.83	135195.3360	0	1	220.534624	Intel Core i7	0	512	0	0	AMD	Mac
4	Apple	Ultrabook	8	1.37	96095.8080	0	1	226.983005	Intel Core i5	0	256	0	0	Intel	Mac
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1298	Lenovo	2 in 1 Convertible	4	1.80	33992.6400	1	1	157.350512	Intel Core i7	0	128	0	0	Intel	Windows
1299	Lenovo	2 in 1 Convertible	16	1.30	79866.7200	1	1	276.053530	Intel Core i7	0	512	0	0	Intel	Windows
1300	Lenovo	Notebook	2	1.50	12201.1200	0	0	111.935204	Other Intel Processor	0	0	0	64	Intel	Windows
1301	HP	Notebook	6	2.19	40705.9200	0	0	100.454670	Intel Core i7	1000	0	0	0	AMD	Windows
1302	Asus	Notebook	4	2.20	19660.3200	0	0	100.454670	Other Intel Processor	500	0	0	0	Intel	Windows

1302 rows × 15 columns

4. Methodology

a) Algorithm

We have implemented the following algorithms to our datasets and checked the accuracy for each.



1. Decision Tree's goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piece wise constant approximation.

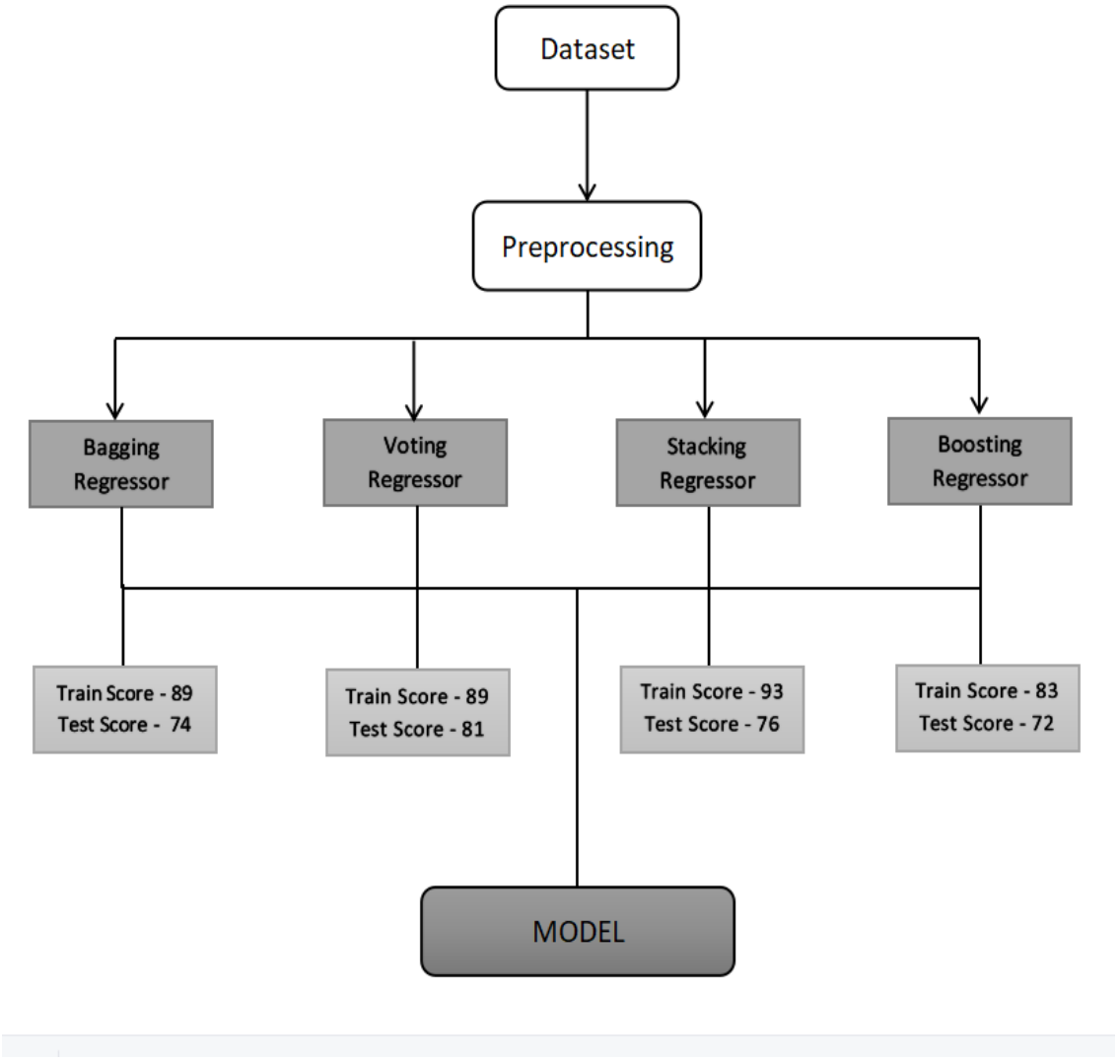
2. K-Nearest Neighbors It is also known as KNN or k-NN, is a non-parametric, supervised learning classifier that employs proximity to classify or anticipate how a certain data point will be grouped.

3. Linear Regression uses independent variables to model a goal prediction value. It mostly serves to determine how factors and forecasting interact.

4. The lasso regression allows you to shrink or regularize these coefficients to avoid overfitting and make them work better on different datasets.

b) Ensemble Learning

After that, we perform ensemble learning to improve the overall predictive performance of our model by combining the predictions of multiple model.



1. Bagging regressor: combines multiple regression model through the process of bagging (bootstrap aggregating).
2. Voting regressor: combines the predictions of multiple model to generate the final prediction.
3. Stacking regressor: combines multiple model in hierarchical manner to make prediction.
4. Boosting regressor: combines multiple weak model to create a strong predictive model.

c) Evaluation Metrics

Once the machine learning model is trained we should see its efficiency to make sure that our model is well trained to make good predictions on newly observed data.
Once the model is trained, we checked the accuracy of the model.

- R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively.
- Mean Absolute Error (MAE): MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.
- Mean Squared error (mse): MSE is a common metric used to measure the average squared difference between the predicted value and the actual value.

d) Evaluation of model

1) Bagging ensemble learning

```
Mean Absolute Error: 11998.887674730204
Mean Squared Error: 294171788.8111861
Root Mean Squared Error: 17151.43693138234
R-squared: 0.7473898399948364
```

```
print(pipelinemodel.score(X_train, y_train))
print(pipelinemodel.score(X_test, y_test))
```

```
0.8952380052839735
0.7473898399948364
```

2) Voting ensemble learning

```
Mean Absolute Error: 10497.328548316722
Mean Squared Error: 221421129.56575245
Root Mean Squared Error: 14880.226126163287
R-squared: 0.8098620292783089
```

```
print(pipelinemodel1.score(X_train, y_train))
print(pipelinemodel1.score(X_test, y_test))
```

```
0.8999076972875585
0.8098620292783089
```

3) Stacking ensemble learning

```
Mean Absolute Error: 11524.439097186807
Mean Squared Error: 298336471.9401203
Root Mean Squared Error: 17272.41940030754
R-squared: 0.7438135580004873
```

```
print(pipelinemodel2.score(X_train, y_train))
print(pipelinemodel2.score(X_test, y_test))
```

```
0.9379755242109136
0.7438135580004873
```

4) Boosting ensemble learning (AdaBoostRegressor)

```
Mean Absolute Error: 13417.618642737472
Mean Squared Error: 317039216.9585812
Root Mean Squared Error: 17805.595102623814
R-squared: 0.727753202822508
```

```
print(pipelinemodel3.score(X_train, y_train))
print(pipelinemodel3.score(X_test, y_test))
```

```
0.839535963687277
0.727753202822508
```

e) Final model selection

The final model selection was voting regressor ensemble learning with an accuracy score of 80%, train score of 89%, and 81% of test score. We chose the voting regressor learning model because compared to other estimators with and without ensemble learning, it gives a high accuracy score and has the least difference between train and test scores. Without ensemble learning, linear regression has the best performance with 75% train score and 74% test score. After implementing voting ensemble learning, accuracy improves to 89% and 81% train and test scores respectively.

5. Result and Discussion

a) Experimental Setup

1. Anaconda

An open-source distribution of Python and R for data research called Anaconda seeks to make package management and deployment easier. Packages for data science are available in the distribution that work with Windows, Linux, and macOS.

2. Jupyter Notebook

The objective of Project Jupyter is to create open-source software, open standards, and interactive computing services for a variety of programming languages. The first web application for producing and sharing computational documents was called a Jupyter Notebook. It provides a straightforward, efficient, document-focused experience.

3. Scikit Learn

It is a Python programming language machine learning package that is available for free. Support-vector machines, random forests, gradient boosting, k-means, and DBSCAN are just a few of the classification, regression, and clustering algorithms it offers. It is also built to work with Python's NumPy and SciPy scientific and numerical libraries.

4. Matplotlib and seaborn

For plotting, we use Matplotlib and Seaborn. On top of matplotlib, seaborn offers an API that provides reasonable options for plot style and color defaults, outlines straightforward high-level functions for popular statistical plot kinds and connects with Pandas DataFrame's features.

b) Programming language: Python

1. Python Libraries for Data Processing and Model

• Pandas: This is one of those libraries for data analysis that contains high-level data structuresand tools to manipulate data in a simple way.

Sci-Kit Learn: It has supervised and unsupervised machine learning algorithms for production applications. Sci-Kit Learn focuses on code quality, documentation, ease of use, and performance as this library provides learning algorithms.

• NumPy: Is a Python library for numerical calculations and scientific computations. NumPy provides numerous features which Python enthusiasts and programmers can use to work with high-performing arrays and matrices.

c) Web Deployment

We integrated the model with the web using Flask and deployed it in the python anywhere platform as it consists of the libraries we required.

https://laptoppriceprediction-i7w0.onrender.com/

Conclusion

Based on a few particular specifications, the goal of this research was to estimate the cost of laptops from manufacturers like Apple, Dell, Lenovo, Acer, HP, etc.

We used a variety of supervised regression strategies to solve this problem and determined the most effective one. To determine which algorithm provides the greatest prediction, we tested many evaluation metrics such as mse, mae, accuracy_score, and rmse.

Among different algorithms, we chose voting ensemble learning with an accuracy score of 80%, and train score of 89%, and a test score of 81%.

Bibliography

Chapke, A., Barapatre, A., & Lohakare, S. (2022, January). Laptop Price Prediction using Machine Learning. *11*(1), 164–168. doi:10.47760/ijcsmc. 2022.v11i01.021

Journal, I. J. C. S. M. C. (2022). Laptop Price Prediction using Machine Learning. IJCSMC, (1), 164–168. https://doi.org/10.47760/ijcsmc.2022.v11i01.021

S., Subhiksha & Thota, Swathi & Sangeetha, J.. (2020). Prediction of Phone Prices Using Machine Learning Techniques. 10.1007/978-981-15-1097-7_65. Retrieved from https://www.researchgate.net/publication/338471736_Prediction_of_Phone_Prices_Using_Machine_Learning_Techniques

BuyVia. (n.d.-b). BuyVia. https://www.buyvia.com/ Kopischke, N. (2022, March 19). *How to Use Keepa (Amazon Sales Rank + Price History Charts)*. Online Selling Experiment.

https://onlinesellingexperiment.com/how-to-use-keepa/#:~:text=Keeps%20is%20a%20tool%20that,area%20on%20a%20listing%20page

PriceSpy Ltd. (n.d.). PriceSpy UK. https://pricespy.co.uk/