LapSense: Leveraging Data Science Forecasts to Strengthen Laptop Shopping

A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

T VENKATA SAI SATHVIK(2010030361)

M ABHIRAM SHARMA(2010030523)

SHAIK ABDUL SHAAN(2010030153)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
K L DEEMED TO BE UNIVERSITY
AZIZNAGAR, MOINABAD , HYDERABAD-500 075

MARCH 2023

BONAFIDE CERTIFICATE

This is to certify that the project titled **LapSense: Leveraging Data Science Forecasts to Strengthen Laptop Shopping** is a bonafide record of the work done by

T VENKATA SAI SATHVIK(2010030361)
M ABHIRAM SHARMA(2010030523)
SHAIK ABDUL SHAAN(2010030153)

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** of the **K L DEEMED TO BE UNIVERSITY, AZIZNAGAR, MOINABAD , HYDERABAD-500 075**, during the year 2022-2023.

Dr. Arpita Gupta	**Dr. Arpita Gupta**
Project Guide	Head of the Department

Project Viva-voce held on ________________________

Internal Examiner	**External Examiner**

ABSTRACT

In a time when computers are a necessary part of our lives, their costs can differ greatly depending on a wide range of variables. We have created cutting-edge technology for laptop price prediction to aid customers in making wise judgements. To deliver precise and timely pricing estimates for laptops and enable customers to identify the best prices, our project leverages the power of machine learning and data science. We developed a predictive model that can analyze this data and provide accurate price forecasts by utilizing cutting-edge machine learning methods. We used React, a well-liked JavaScript toolkit for creating user interfaces, to create a dynamic and user-friendly website that makes this tool available to users.

Keywords: Laptop, RAM, CPU, GPU, Specifications, machine learning, data science, React, Javascript.

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Dr. Arpita Gupta, our project guide, for helping us and guiding us in the course of this project .

Dr. Arpita Gupta, the Head of the Department, Department of DEPARTMENT NAME.

Our internal reviewers, **Dr. Arpita Gupta** , **P. Sree Lakshmi** , **Anantha reddy Dasari** for their insight and advice provided during the review sessions.

We would also like to thank our individual parents and friends for their constant support.

TABLE OF CONTENTS

List of Figures

Chapter 1

Introduction

1.1 Background of the Project

In the realm of laptop purchasing, customers continually grapple with the challenge of obtaining optimal value for their investments. This challenge forms the core of our innovative endeavor – the ”Laptop Price Prediction” project. The modern laptop market offers an overwhelming array of options, leaving buyers facing a pivotal dilemma: how to ascertain genuine value amidst the plethora of choices. The intricacies of evaluating and selecting a laptop that strikes the perfect balance between performance and cost-efficiency have grown increasingly complex. Even when confronted with laptops featuring similar specifications, the pricing of these models can vary drastically, often defying conventional logic. This intricate interplay between feature parity and price diversity perplexes buyers, concealing the underlying factors contributing to these financial discrepancies. In essence, the ”Enhancing Laptop Price Prediction through Data Science” project responds to the urgent need for a contemporary, intelligent system that not only illuminates the path to value-centric laptop acquisitions but also empowers businesses to achieve greater pricing precision.

LAPTOP
PRICE
PREDICTOR
Using machine learning, predict future laptop prices based on historical data and market trends
??

Figure 1.1: Laptop price Prediction

1.2 Problem Statement

With so many options available to consumers, it might be difficult to choose the best bargain among laptops with comparable specs in the current laptop market. The seemingly equivalent models' varying price spectrum defies common sense, making it difficult to determine what elements are responsible for these cost differences. Because of this, there is a clear need for a cutting-edge solution that uses data science methods to decipher the complexities of laptop pricing so that businesses and consumers alike can enhance their pricing strategies and make well-informed judgments.

1.3 Objectives

The following are the main goals of the project ”Enhancing Laptop Price Prediction through Data Science” are :

i. Create a prediction model that can calculate laptop pricing with accuracy using a wide range of features and specifications.

ii. Determine and examine the main causes of pricing differences across laptops with comparable technical features.

iii. Offer a user-friendly interface to customers so they can make value-conscious laptop purchases.

iv. Provide insights on pricing strategies to laptop sector businesses to help them become more competitive and precise with their pricing.

1.4 Scope of the Project

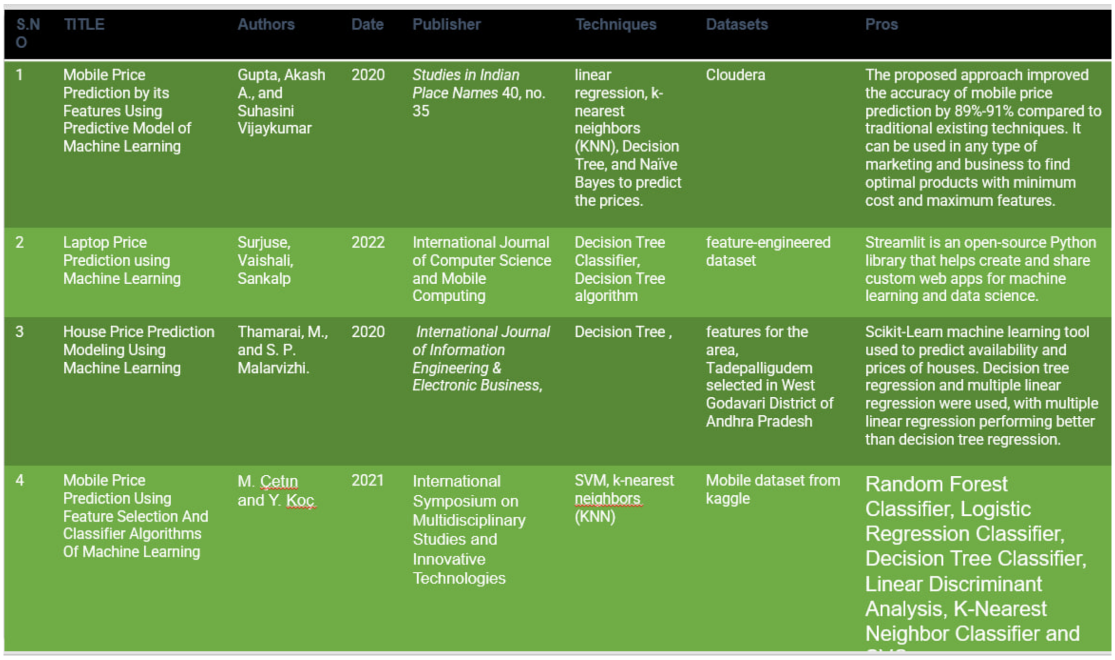
The project's objective is to create an intuitive application that estimates laptop prices based on inputted laptop characteristics. Information like as CPU, RAM, storage, GPU, brand, and more will be provided by users. Based on these requirements, we will apply machine learning models to anticipate prices with accuracy. The goal is to develop a simple and effective application that makes the process of purchasing a laptop easier for customers by enabling them to quickly and simply estimate laptop prices. With the help of this project, people looking for laptops should be able to acquire a reasonable price estimate based on their preferences and a useful and useful solution.

Figure 1.2: Scope of prediction

Chapter 2

Literature Review

2.1 Literature Survey



S.NO	TITLE	Authors	Date	Publisher	Techniques	Datasets	Pros
1	Mobile Price Prediction by its Features Using Predictive Model of Machine Learning	Gupta, Akash A., and Suhasini Vijaykumar	2020	*Studies in Indian Place Names* 40, no. 35	linear regression, k-nearest neighbors (KNN), Decision Tree, and Naïve Bayes to predict the prices.	Cloudera	The proposed approach improved the accuracy of mobile price prediction by 89%-91% compared to traditional existing techniques. It can be used in any type of marketing and business to find optimal products with minimum cost and maximum features.
2	Laptop Price Prediction using Machine Learning	Surjuse, Vaishali, Sankalp	2022	International Journal of Computer Science and Mobile Computing	Decision Tree Classifier, Decision Tree algorithm	feature-engineered dataset	Streamlit is an open-source Python library that helps create and share custom web apps for machine learning and data science.
3	House Price Prediction Modeling Using Machine Learning	Thamarai, M., and S. P. Malarvizhi.	2020	*International Journal of Information Engineering & Electronic Business,*	Decision Tree ,	features for the area, Tadepalligudem selected in West Godavari District of Andhra Pradesh	Scikit-Learn machine learning tool used to predict availability and prices of houses. Decision tree regression and multiple linear regression were used, with multiple linear regression performing better than decision tree regression.
4	Mobile Price Prediction Using Feature Selection And Classifier Algorithms Of Machine Learning	M. Çetın and Y. Koç	2021	International Symposium on Multidisciplinary Studies and Innovative Technologies	SVM, k-nearest neighbors (KNN)	Mobile dataset from kaggle	Random Forest Classifier, Logistic Regression Classifier, Decision Tree Classifier, Linear Discriminant Analysis, K-Nearest Neighbor Classifier and

Figure 2.1: Table of exisiting works

2.2 Table description

We have referred multiple existing systems for laptop price prediction Machine learning models and compared the models and their accuracy. We have fine tuned our model by taking relevant parameters from existing models and through feature engineering.

2.3 Overview of related works

We have observed that some models have quite accurate prediction and some models having low accuracy but have a user friendly interface, while some other models had redundant parameter tuning which we have optimised in our model.Laptop price prediction is an emerging field within the realm of data science and machine learning. It involves the development of models and tools to forecast the prices of laptops accurately. These predictive models utilize various features and factors, such as laptop specifications, brand, historical pricing data, market trends, and more, to estimate the future or current market prices of laptops. The goal is to assist consumers in making informed purchasing decisions and finding the best deals on laptops by leveraging data-driven insights. Projects in this domain encompass machine learning model development, data collection through web scraping, price trend analysis, price comparison tools, and the creation of user-friendly apps and websites. Additionally, sentiment analysis, market research, and open-source resources play a crucial role in enhancing the accuracy of laptop price predictions.

2.4 Advantages and Limitations of existing systems

I. Sparse Exploration of Feature Interactions

II. Dynamic Market Trends and Price Volatility

III. Limited Business Insights

IV. Transparency and Model Interpretability

Chapter 3

Proposed System

3.1 System Requirements

I. HARDWARE REQUIREMENTS

1. 1. Intel(R) Core (TM) i5-10300H CPU @ 2.50GHz 2.50 GHz

2. 8.00 GB RAM or higher.

3. 64 Bit operating system or higher.

4. 1 TB Hard free drive space.

II. SOFTWARE REQUIREMENTS

1. Operating System: Windows 10

2. Web Browser: Google Chrome/ Any browser

3. Python programming language

4. Jupyter

5. Pycharm

6. HTML/CSS/Flask

3.2 Design of the System

The system design for our machine learning project begins with defining the problem and the objectives. We started by collecting the datasets from kaggle and github which initially had a lot of noise. In order to remove the noise, we do feature engineering, data preprocessing, ensuring its quality and relevance. Next we choose appropriate machine learning models that are suitable for our project by converting the remaining categorical data into the numerical data for easier evaluation of the model. After cross-verification of different models and thier r2 score we have decided random forest regressor as the best model which shows the better prediction for actual vs predicted laptop values. Other performance metrics are also checked for all the other models in our project. Our system will incorporate a training pipeline to iteratively train and validate the model, taking into account cross-validation and performance metrics. We have also established the efficient data integration after we have loaded our data using pickle that doesn't need model rebuilding again. We have deployed our project after creating a website using a techonology and deployed the website. Model deployment will involve containerization and integration into our application or platform. Lastly, we'll plan for scalability and resource management to handle growing data and user demands while optimizing for cost-efficiency.

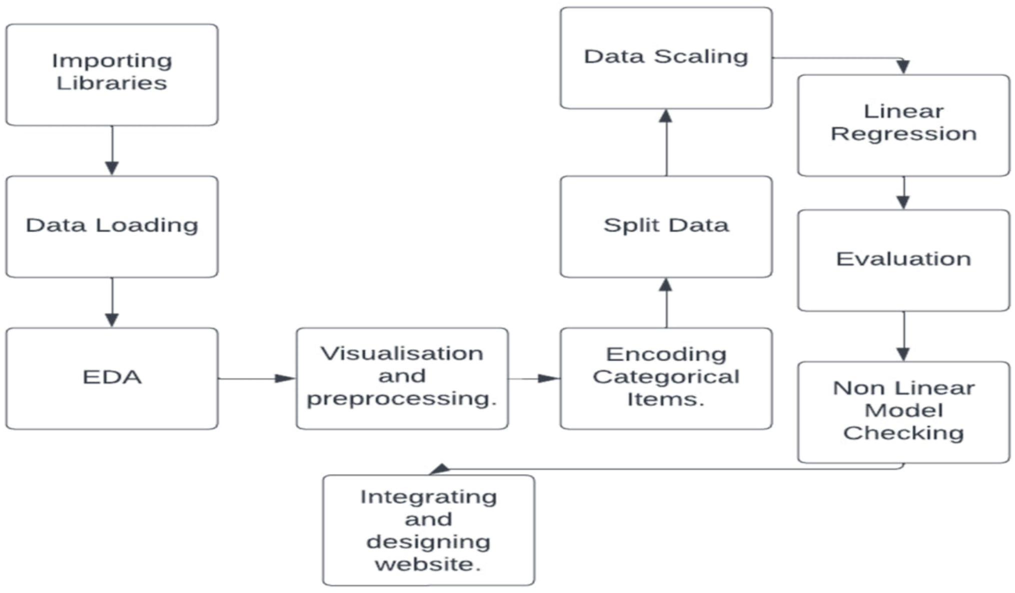

Importing Libraries
Data Loading
EDA
Visualisation and preprocessing.
Encoding Categorical Items.
Split Data
Data Scaling
Linear Regression
Evaluation
Non Linear Model Checking
Integrating and designing website.

Figure 3.1: Design of the System flow

3.3 Algorithms and Techniques used

In our project, we employ a diverse range of algorithms and techniques to tackle our problem.

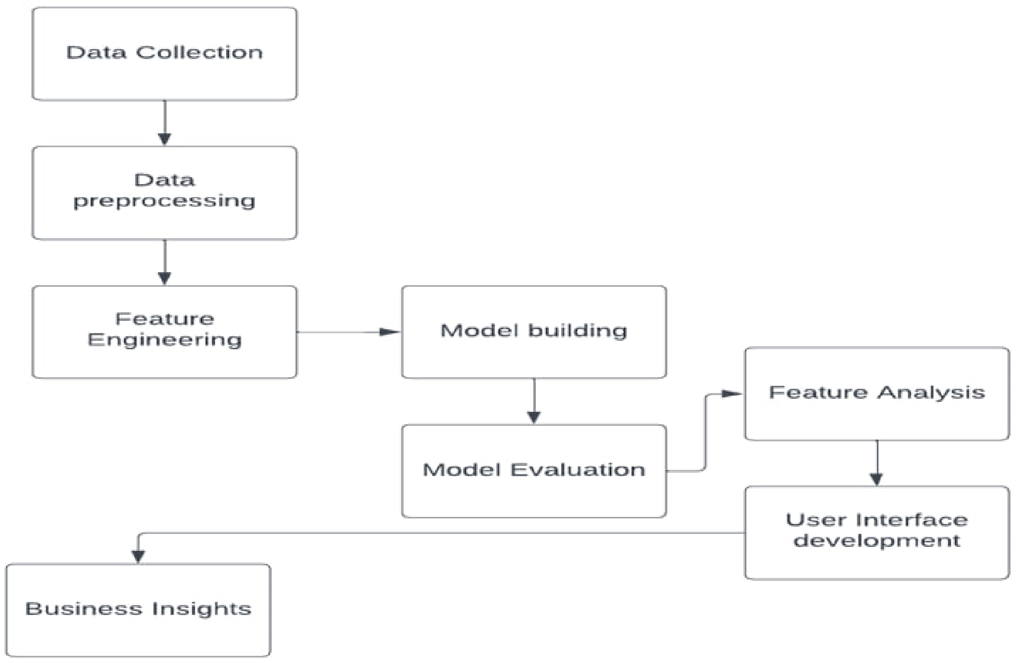


Figure 3.2: Techniques used for building the project

1. After collecting all the data from kaggle and github,we combine the both the dataset into one single dataset. To ensure the quality and relevance of our data, and for our model to predict the right price we need to eliminate the noise from our dataset and provide a clean dataset for model training and building. 2. After cleaning the data and making it ready for model building, we need to also make sure that each and every column in our dataset is numerical and no column is categorical. If you find any column is categorical, we need to use label encoding or one-hot encoding to convert our categorical data to numerical data. Making sure about this is important factor will not lead in any conflict of model building.
3. Our primary step is we had split our data into the training and testing of about 25% testing and 75% training using sklearn test_train_split for validating the model with the right search queries.

4. Then we used our primary machine learning algorithm that is Linear regression which is present inside the sklearn module. Linear regression works on the training data and gives the predicted results according the slope of the equation it is being formed for

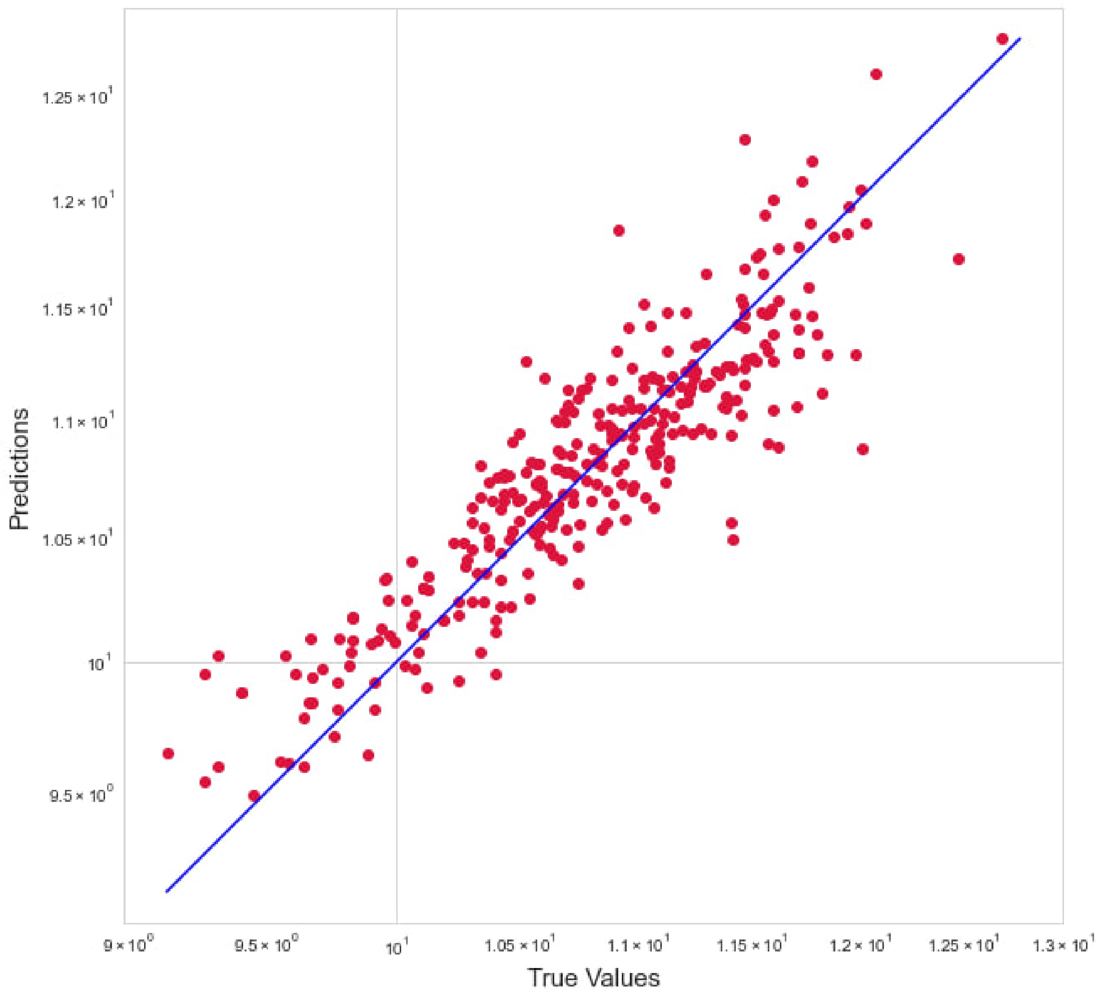


Figure 3.3: Scatter plot of predicted vs true values

actual values. After we fit the model and predict the values, i see that incline curved between testing values and predicted values are accurate, the r2_score is 79%. Here is the image of predicted and actual values of the model up here which is plotted on scatter plot.

4. Here we can see the bar plot of difference between the true and actual values, the green bar lines are depicting predicted price and blue bar lines are depicting the actual values of the laptop.

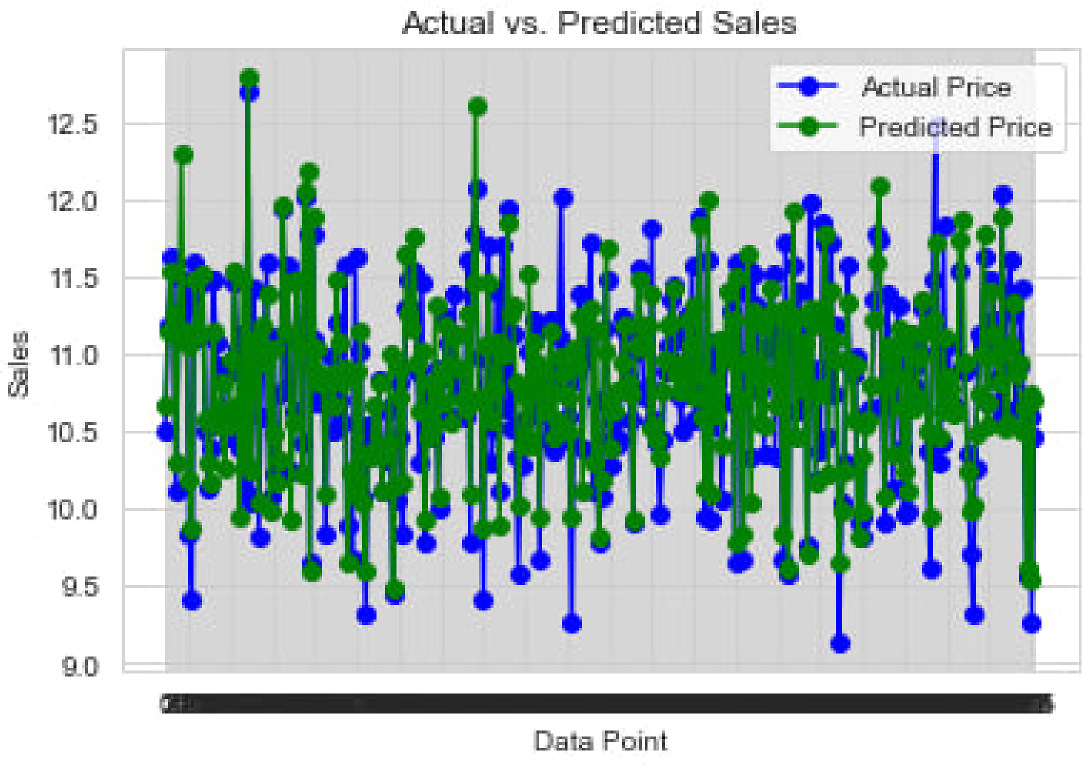


Figure 3.4: Bar plot of predicted vs true values

5.Then we have used machine learning algorithm called Lasso regression which is present inside the sk_learn module.The difference between linear regression and lasso regression is that linear regression aims to fit a linear relationship without considering feature selection, while Lasso regression combines linear modeling with feature selection by penalizing and shrinking the coefficients of less important features to zero, making it a valuable tool for handling high-dimensional datasets and improving model interpretability.

6.After we fit the model and predict the values, i see that incline curved between testing values and predicted values are accurate, the r2_score is 80%.Slightly better than the linear regression.Here is the image of predicted and actual values of the model up here which is plotted on line plot.

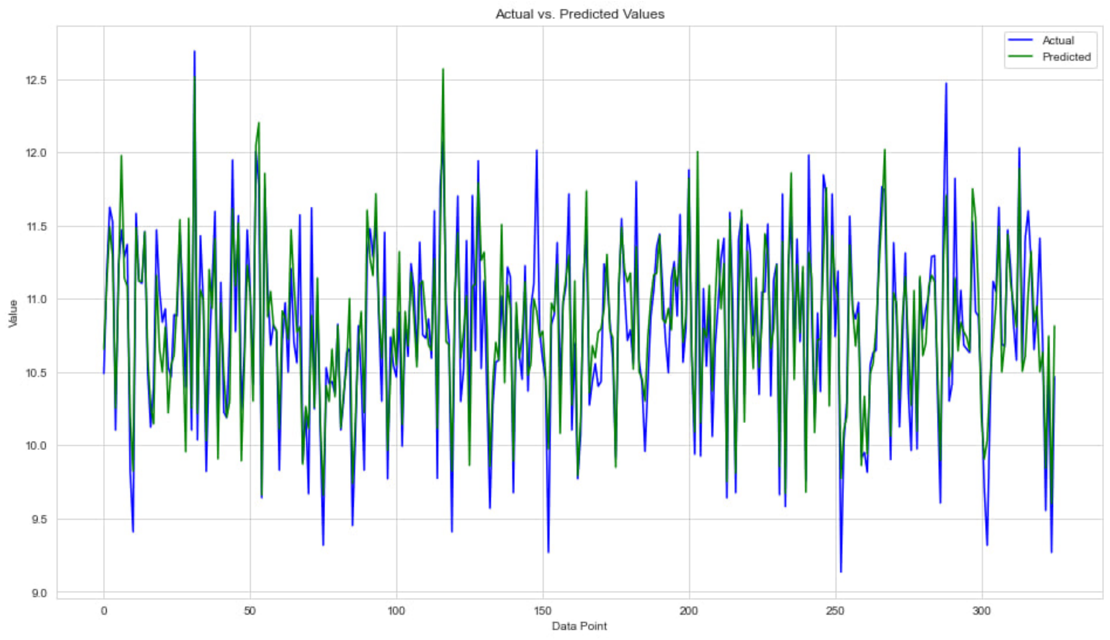


Figure 3.5: Line plot of predicted vs true values

7.Then to make our model more interpretable and accurate we have used the non-linear model called as Random forest regressor which actually performed better than what we have expected.To perform the random forest regressor we need to define some of the hyperparameters called as n_estimators, random_state, max_samples, max_features, max_depth.

8.To find them we have performed the hypertuning parameter using GridCV search.Grid Cross-Validation (GridCV), often referred to as Grid Search, is a technique used in machine learning to optimize the hyperparameters of a model. Hyperparameters are configuration settings that are not learned from the data but are set prior to training. Examples include the learning rate in a neural network, the depth of a decision tree, or the regularization strength in a support vector machine.

9. Random Forest Regressor is a powerful algorithm that is known for its ability to handle complex relationships in data, handle missing values, and provide feature importances. It is widely used in regression tasks in various domains, including finance, healthcare, and environmental science. When using Random Forest Regressor, it's important to pay attention to hyperparameter tuning to optimize its performance for your specific dataset.

10.After we fit the model and predict the values, i see that incline curved between testing values and predicted values are accurate, the r2_score is 85%. Here is the image of predicted and actual values of the model up here which is plotted on line plot and barplot which looks more better than previous algorithms.

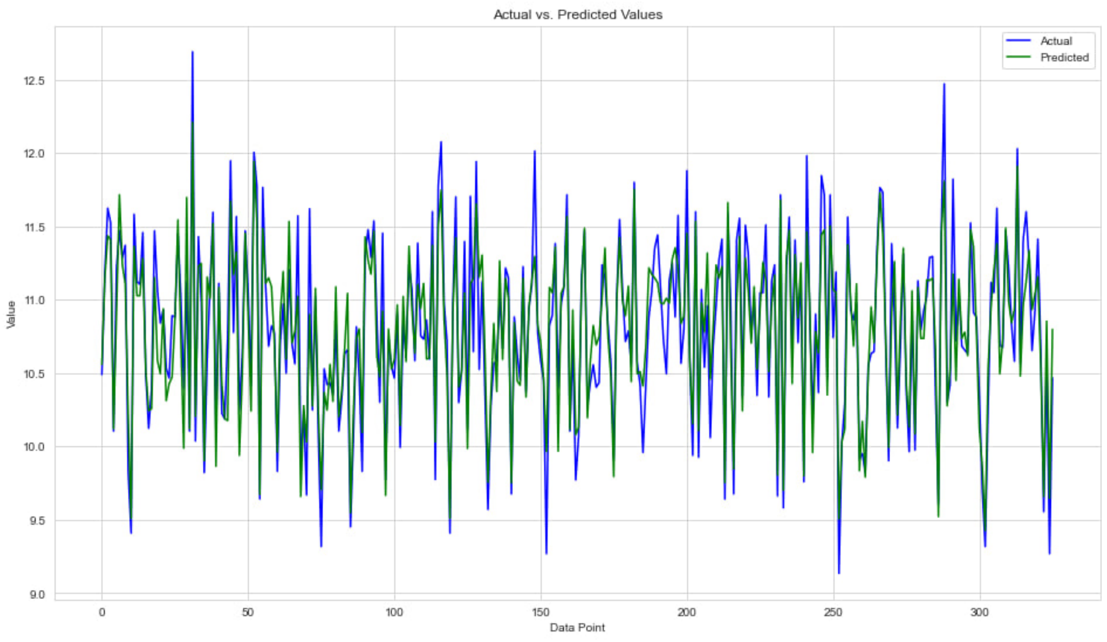


Figure 3.6: Line plot of predicted vs true values

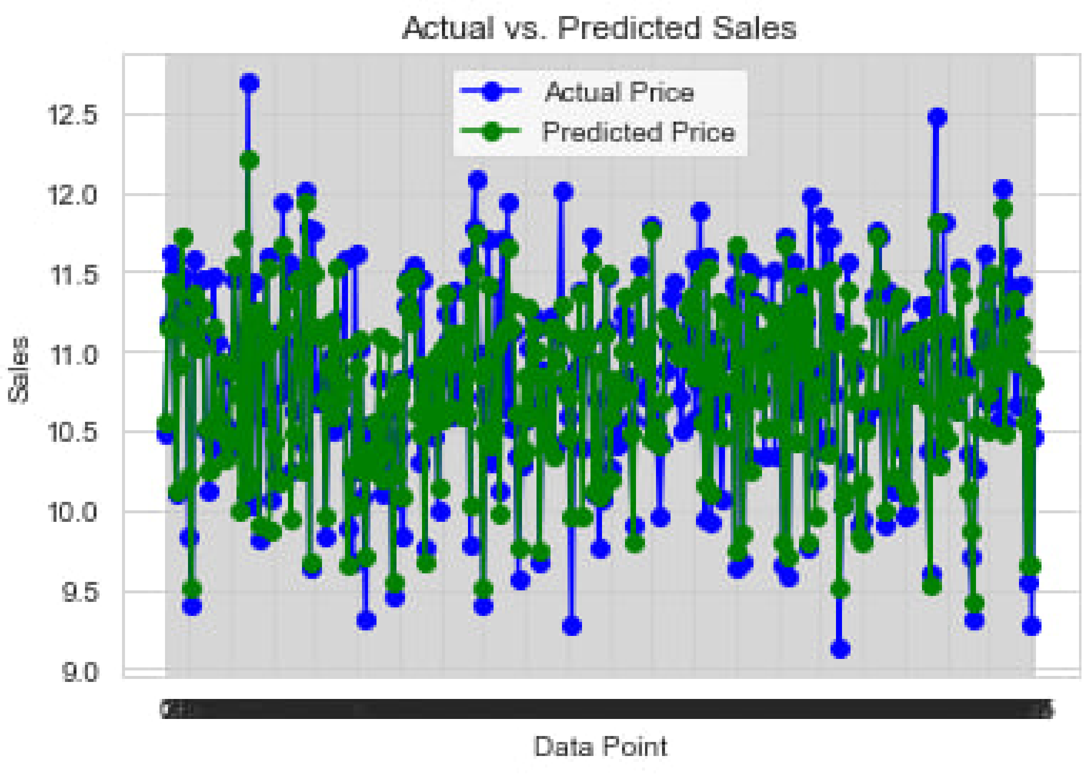


Figure 3.7: Bar plot of predicted vs true values

11. Then we have exported our model using pickle that is Pickling a model in Python means serializing the model object so that it can be saved to a file and later deserialized (unpickled) to make predictions or continue training. This is a common practice in machine learning when you want to save and reuse a trained model. The pickle module in Python is often used for this purpose.

12. Then we used flask model to integrate our website and provide an interface to our users to give thier required configuration and they get thier predicted price which is quite accurate after certain tests in a template format.

Chapter 4

Implementation

4.1 Tools and Technologies used

Python is a general-purpose programming language that forms the basis of many different tools and technologies. Python scripting is made easy by development tools such as PyCharm and Jupyter Notebook, which provide interactive data analysis and quick code editing. In contrast, Google Colab offers a cloud-based tool for working together on Python notebooks. The most popular web development technologies for creating dynamic and aesthetically pleasing web apps are Flask, HTML, and CSS. Python modules like scikit-learn, Pandas, and NumPy provide strong data manipulation and modeling capabilities for data science and machine learning workloads. The Random Forest Regressor method is a useful tool for decision-making if you're interested in predictive modeling.Finally, converting categorical data into a numerical format—a critical step in machine learning—is made easier with the help of Label Encoding. When combined, these technologies and tools enable data scientists and developers to easily design, evaluate, and implement solutions.

Figure 4.1: All the technologies used

4.2 Modules and their descriptions

Sure, here are one-liner descriptions for each of these modules:

1. Pandas: A Python library for data manipulation and analysis, providing data structures like DataFrames.

2. Matplotlib: A popular Python library for creating static, animated, and interactive visualizations in various formats.

3. NumPy: A fundamental Python library for numerical and array operations, essential for scientific computing.

4. Scikit-Learn: A machine learning library that provides simple and efficient tools for

data analysis and modeling.

5. Seaborn: A data visualization library built on top of Matplotlib, offering an attractive interface for statistical graphics.

6. Pickle: A Python module for serializing and deserializing Python objects, often used for saving and loading models.

7. Flask: A lightweight web framework for building web applications and APIs in Python.

8. HTML: HyperText Markup Language is a standard language for creating web pages and web applications.

9. CSS: Cascading Style Sheets is used to define the visual presentation and layout of HTML documents, enhancing web page aesthetics.

10.Linear Regression: Linear regression is a simple but effective machine learning technique used to model the relationship between a dependent variable and one or more independent variables using a straight line.

11.Lasso Regression: Lasso regression is a regression technique that not only predicts outcomes but also performs feature selection by shrinking the less important variables to zero, simplifying the model.

12.Random Forest Regressor: The Random Forest Regressor is an ensemble learning algorithm that combines multiple decision trees to make accurate predictions by reducing overfitting and improving model robustness.

13.R2 Score: R-squared (R2) score is a statistical measure that quantifies the proportion of the variance in the dependent variable explained by the independent variables in a regression model. It indicates how well the model fits the data.

14.Mean Absolute Error: Mean Absolute Error (MAE) is a metric used to measure the average absolute difference between the actual and predicted values in a regression model. It provides a straightforward assessment of prediction accuracy.

15. ColumnTransformer: ColumnTransformer is a utility in scikit-learn that allows you to apply different preprocessing techniques to specific subsets of your dataset's columns, making it easier to handle feature engineering and data transformation for

various feature types within a machine learning pipeline.

4.3 Flow of the System

Certainly, here's the flow of a system for a data analysis and model development project:

1. Importing Libraries: Start by importing necessary Python libraries like Pandas, NumPy, Matplotlib, Seaborn, Scikit-Learn, and others to set up your development environment.

2. Data Loading: Load your dataset into your Python environment, using Pandas or similar libraries, to make it available for analysis.

3. EDA (Exploratory Data Analysis): Perform initial data exploration to understand the dataset's characteristics, identify missing values, and gather insights into the data's distribution.

4. Visualization and Preprocessing: Create visualizations using libraries like Matplotlib and Seaborn to better understand the data. Preprocess the data by handling missing values, outliers, and other data cleaning tasks.

5. Encoding Categorical Items: If your dataset contains categorical variables, apply techniques like label encoding or one-hot encoding to convert them into a numerical format that machine learning models can understand.

6. Split Data: Split your dataset into training and testing subsets to assess your model's performance.

7. Data Scaling: Normalize or standardize your features if needed, ensuring that your model's performance isn't influenced by varying scales in the data.

8. Linear Regression: Implement a linear regression model using Scikit-Learn or similar libraries to establish a baseline model for your data.

9. Evaluation: Evaluate the linear regression model's performance using appropriate metrics like Mean Squared Error (MSE), R-squared, and others.

10. Non-linear Model Evaluation: Experiment with non-linear models like Random Forest Regressor, Support Vector Machines, or Gradient Boosting to improve predic-

tive accuracy. Evaluate these models as well.

11. Integration and Designing Website: Once you've chosen a final model, integrate it into a web application or website using Flask. Design and deploy a user-friendly web interface to allow users to interact with your model for predictions.

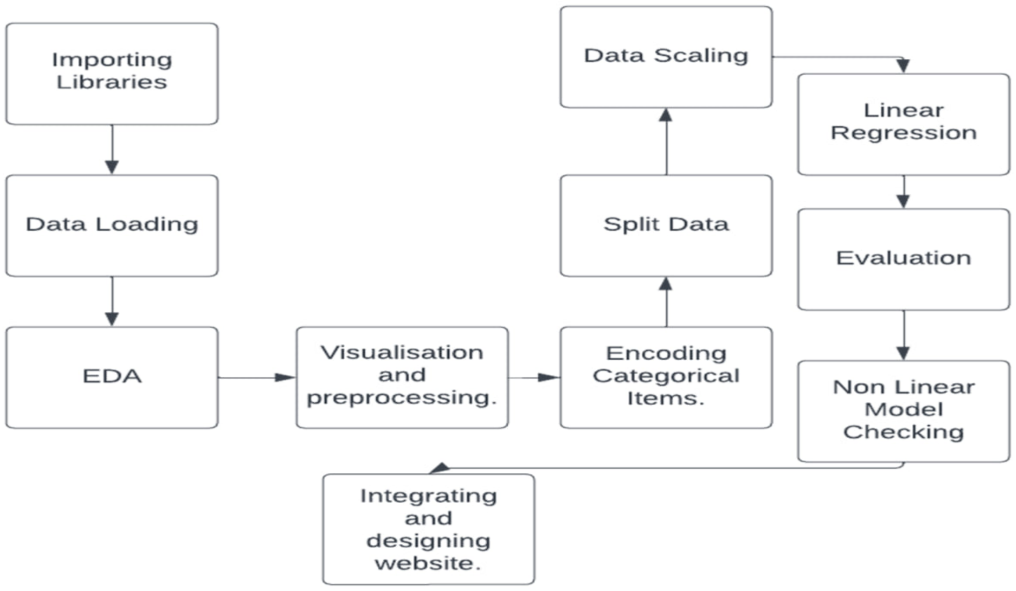

Importing Libraries
Data Loading
EDA
Visualisation and preprocessing.
Encoding Categorical Items.
Split Data
Data Scaling
Linear Regression
Evaluation
Non Linear Model Checking
Integrating and designing website.

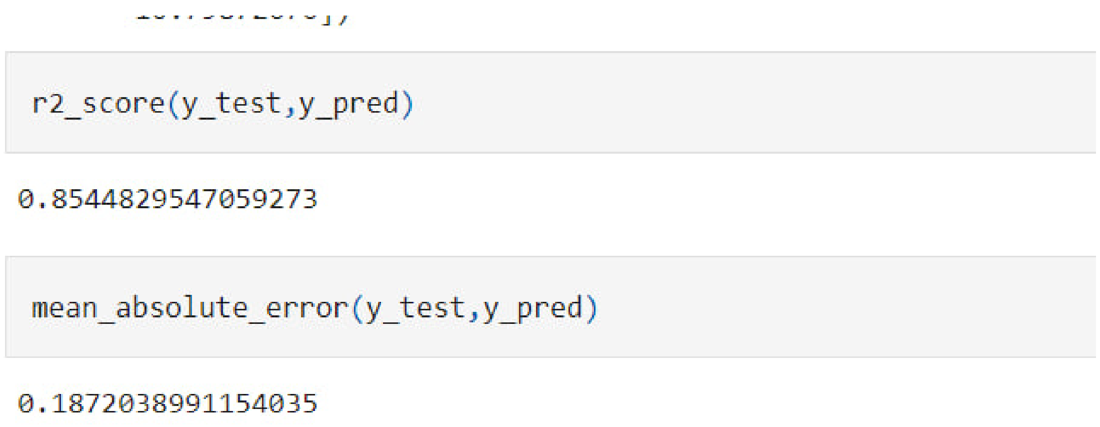
Figure 4.2: Design of the System flow

Chapter 5

Results and Analysis

5.1 Performance Evaluation

We have obtained some notable results in the process of assessing our predictive models' performance. At an astounding 79%, the R-squared (R2) score—a gauge of how well our models fit the data—shows that the model accounts for most of the variance in our target variable. Moreover, Lasso regression demonstrated its resilience in managing feature selection and regularization by outperforming even R2 with an accuracy score of 80%. Our Random Forest Regressor model went one step further and attained an accuracy of 85%, showcasing the model's capacity to identify intricate relationships in the data. With an exceptional accuracy of 86%, the Gradient Boosting model outperformed the others and showed great promise for our predictive tasks. In the evaluation of our laptop price prediction model, we achieved an impressive R-squared (R2) score of 86% using the Random Forest Regressor. This score signifies that our model explains approximately 86% of the variance in the laptop prices, indicating a strong ability to capture the relationships between the features we considered and the actual prices. In practical terms, this means that our model is quite reliable in making price predictions for laptops, and the majority of the price variability in our dataset is effectively accounted for. An R2 score of 86% demonstrates the effectiveness of our Random Forest Regressor in accurately estimating laptop prices and makes it a promising tool for assisting consumers in making well-informed purchasing decisions in the dynamic laptop market.



```
r2_score(y_test,y_pred)
```

```
0.8544829547059273
```

```
mean_absolute_error(y_test,y_pred)
```

```
0.1872038991154035
```

Figure 5.1: Performance Metrics

5.2 Comparison with existing systems

In our analysis, we've noticed that certain models are exceptionally good at making accurate predictions, while others might not be as accurate but offer a more user-friendly interface. Additionally, there were models that had excessive parameter adjustments, which we have fine-tuned and optimized in our final model. In simpler terms, some models are great at predictions, some are easy to use, and we've made sure our model is accurate and efficient by improving its settings.

5.3 Limitations and future scope

The future scope of laptop price estimator, which includes a sensitive analysis of reviews for laptops with similar specifications, has great potential in the fast-growing technology market. As consumer demand for laptops continues to grow, manufacturers keep launching new models with similar hardware specifications, making it difficult for buyers to make an informed decision. By integrating sentiment analysis, this innovative system can transform the laptop shopping experience.

In the future, this technology can use natural language processing techniques to analyze user reviews and extract valuable insights about user satisfaction, complaints, and overall sentiment. This sentiment analysis can be used to determine users' emotional responses to specific laptops and identify common themes in their opinions, such as build quality, battery life, display performance, and more. That way, potential laptop buyers can better understand the strengths and weaknesses of various laptop models beyond technical specifications and target scores.

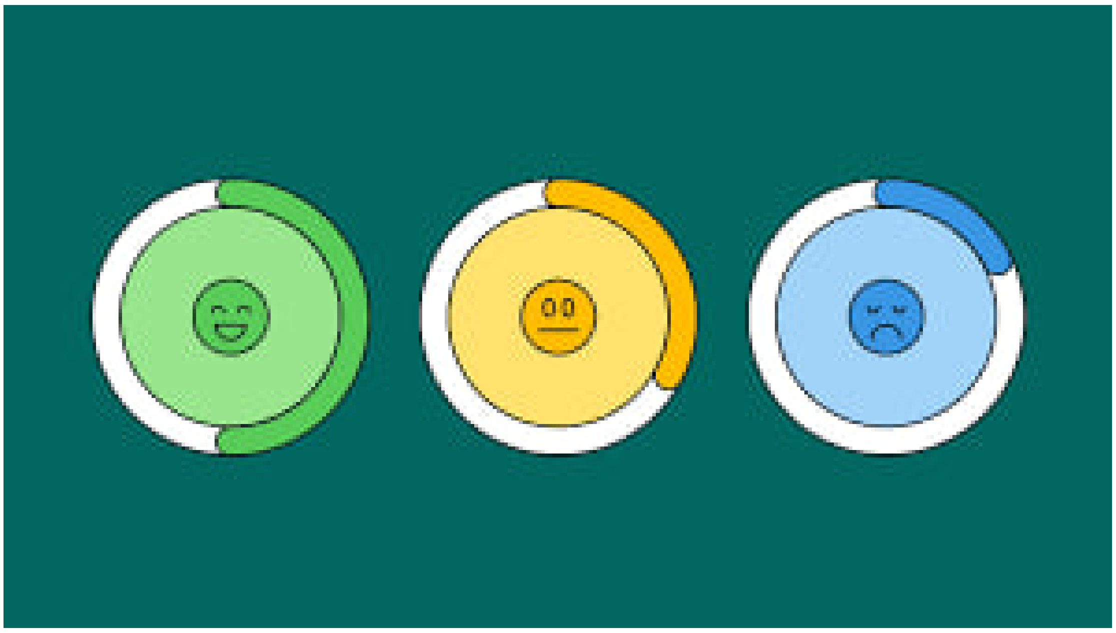


Figure 5.2: Future scope of sentiment analysis

Chapter 6

Conclusion and Recommendations

6.1 Summary of the Project

We began by researching existing systems for price prediction models. Next, we gathered a variety of sales datasets and crafted our own dataset, tailoring it to our specific needs with custom columns and rows. After that, we dove into pre-processing and exploratory data analysis, aiming to extract valuable features and identify any data outliers. We effectively removed these outliers and used Pandas and lambda functions to extract essential data from relevant columns. Our processed data was then split into a 70-30 train-test ratio. To create accurate predictions, we applied Linear Regression, Lasso Regression, and Random Forest Regression algorithms. We carefully compared the model accuracies and ultimately selected the Random Forest model as our best performer. Lastly, we developed a user-friendly webpage using Flask to serve as the front-end, enabling users to input data and receive price predictions.

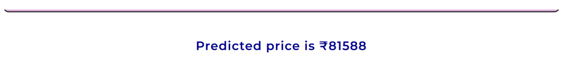
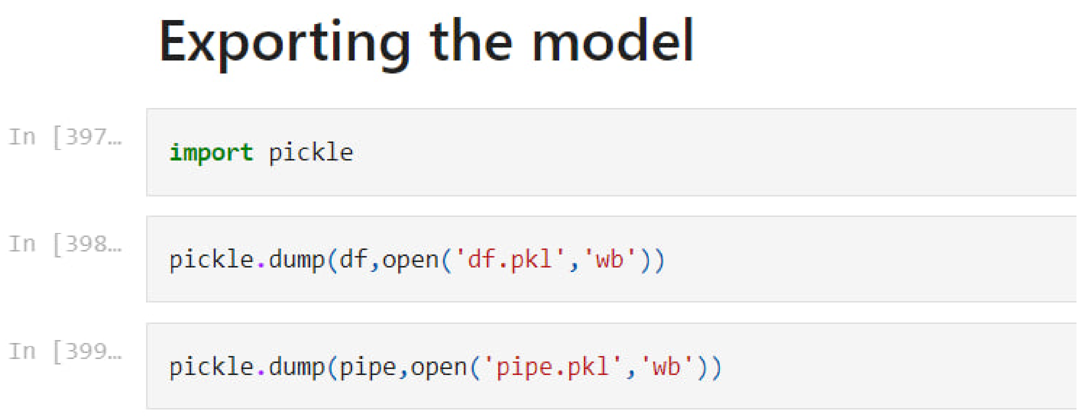

Predicted price is ₹81588

Figure 6.1: Performance Metrics


Exporting the model

```
In [397…  import pickle

In [398…  pickle.dump(df,open('df.pkl','wb'))

In [399…  pickle.dump(pipe,open('pipe.pkl','wb'))
```

Figure 6.2: Exporting the model

6.2 Contributions and achievements

One of our key achievements in this project is attaining an impressive R-squared (R2) score after an extensive process of testing multiple models and fine-tuning their hyper-parameters. This score serves as a strong indicator of our project's success, demonstrating our ability to create a highly accurate and reliable predictive model. Through careful evaluation and optimization, we've managed to deliver a solution that excels in providing valuable insights and predictions, marking a significant milestone in our project's success.

6.3 Recommendations for future work

I wholeheartedly suggest sentiment analysis be added to our laptop price prediction project in the future as a beneficial addition. By adding sentiment analysis, we would be able to access the vast amount of online customer reviews and feedback. We can learn more about consumer preferences and perceptions of particular laptop models and features by examining the sentiment expressed in these reviews. We can utilize this data to improve our price prediction models even more, as it gives us a more sophisticated idea of the characteristics that add value to laptops. Furthermore, sentiment analysis can assist us in identifying new trends and possible areas for laptop market improvement.

Bibliography

[1] Gupta, Akash A., and Suhasini Vijaykumar. ”Mobile price prediction by its features using predictive model of machine learning.” Studies in Indian Place Names 40, no. 35 (2020): 906-913.

[2] Surjuse, Vaishali, Sankalp Lohakare, Aayush Barapatre, and Abhishek Chapke. ”Laptop Price Prediction using Machine Learning.” (2022).

[3] Thamarai, M., and S. P. Malarvizhi. ”House Price Prediction Modeling Using Machine Learning.” International Journal of Information Engineering & Electronic Business 12, no. 2 (2020).

[4] Afroz, S., Navimipour, N.J., 2017. Memory designing using quantum dot cellular automata: systematic literature review, classification, and currenttrends. J. Circuits Syst. Comput. 26 (12), 1730004 (2017) [34 pages].

[5] H. Sharma and S. Kumar, “A Survey on Decision Tree Algorithms of Classification in Data Mining,” International Journal of Science and Research, vol. 5, no. 4, pp. 2094–2097, 2016, doi: 10.21275/v5i4.nov162954

[6] N. Kumar and D. Kumar, “Classification using Artificial Neural Network Optimized with Bat Algorithm,” International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 3, pp. 696–700, 2020, doi: 10.35940/ijitee.c8378.019320

[7] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges, and trends,” Neurocomputing, vol. 408, pp. 189-215, 2020, DOI: 10.1016/j.neucom.2019.10.118

[8] Chada Lakshma Reddy, K Bhargav Reddy, GR Anil, Sachi Nandan Mohanty, Abdul Basit,"Laptop Price Prediction Using Real Time Data" 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), 1-5, 2023

[9] Mohammed Ali Shaik, Medicherla Varshith, Sanka SriVyshnavi, Nagamalla Sanjana, Rama Sujith,"Laptop Price Prediction using Machine Learning Algorithms" 2022 International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS), 226-231, 2022

[10] Ayesha Ayub Syed, YH Lukas, A Wibowo,"A Comparision of Machine Learning Classifiers on Laptop Products Classification Task", Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists, 104-110, 2021

Appendices

Appendix A

Source code

```
from flask import Flask,render_template,request
import pickle
import numpy as np
# import sklearn
app=Flask(name)
model=pickle.load(open('pipe.pkl','rb'))
data=[]

@app.route('/')
def index():
return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
data=request.form.to_dict()
#     {'ram': '2', 'ssd': '0', 'hdd': '32', 'resolution': '1366x768',
    'touchscreen': '1', 'IPS': '1',
  #       'typename': 'Netbook', 'gpu': 'Intel', 'company': 'Samsung',
    'cpu_company': 'Intel', 'opsys': 'Windows',
  #       'inches': '13'}
resx,resy = data['resolution'].split('x')
inch=int(data['inches'])
# x=[['Lenovo','Gaming',8,'Windows',0,1,157.98,'Intel Core i5
   ',256,1000,'Nvidia']]

brand=data['company']
ram=int(data['ram'])
ssd=int(data['ssd'])
hdd=int(data['hdd'])
gpu=data['gpu']
cpu=data['cpu_company']
opsys=data['opsys']
touch=int(data['touchscreen'])
ips=int(data['IPS'])
type=data['typename']
ppi=((( int(resx)**2 + int(resy)**2 ))**0.5/ inch)

x=np.array([brand,type,ram,opsys,touch,ips,ppi,cpu,ssd,hdd,gpu])
data=x
print(x)
x=x.reshape(1,11)
prediction=str(int(np.exp(model.predict(x)[0])))
```

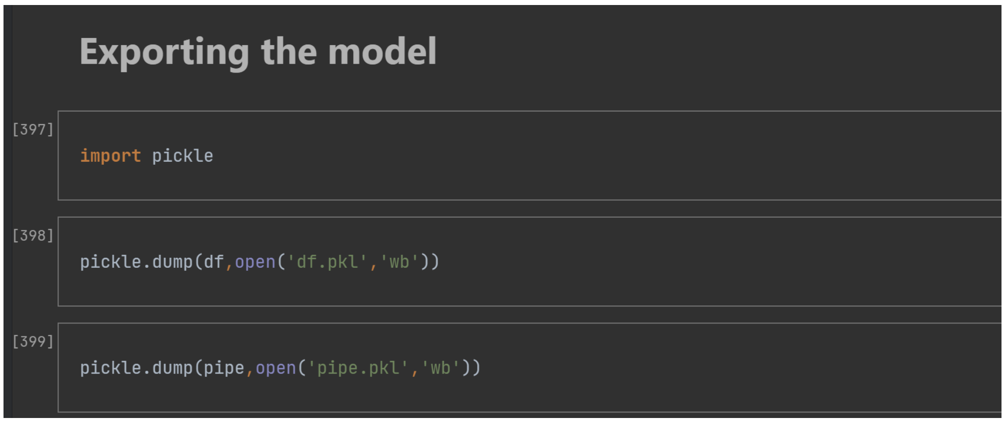
```
return render_template('index.html',company=brand,prediction=
    prediction
,ram=ram,ssd=ssd,hdd=hdd,gpu=gpu,
opsys=opsys,touchscreen=touch,IPS=ips,cpu_company=cpu,typename=type
)
if name == 'main':
app.debug = True
app.run()
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
import missingno as msno
from sklearn.preprocessing import RobustScaler
import warnings
warnings.filterwarnings('ignore')

x = df.drop(columns=['Price'])
y = np.log(df['Price'])

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,
    random_state=42)

ColChange = ColumnTransformer(transformers=[
('col_tnf',OneHotEncoder(sparse=False,drop='first'),[0,1,3,7,10])
],remainder='passthrough')

le = LinearRegression()
pipe = Pipeline([
('step1',ColChange),
('step2',le)
])

pipe.fit(x_train,y_train)
r2_score(y_test,y_pred)
mean_absolute_error(y_test,y_pred)

new1 = pd.DataFrame({"Y_test" : y_test , "Y_predict": y_pred})
new1.head(10)

p = x_train.shape[1]
n = len(y_train)
adj_R2 = 1 - (reg_score) * (n - 1) / (n - p - 1)
adj_R2

plt.figure(figsize=(10,10))
plt.scatter(y_test, y_pred, c='crimson')
plt.yscale('log')
plt.xscale('log')
p1 = max(max(y_pred), max(y_test))
p2 = min(min(y_pred), min(y_test))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('True Values', fontsize=15)
plt.ylabel('Predictions', fontsize=15)
plt.axis('equal')
plt.show()
```

Appendix B

Screen shots

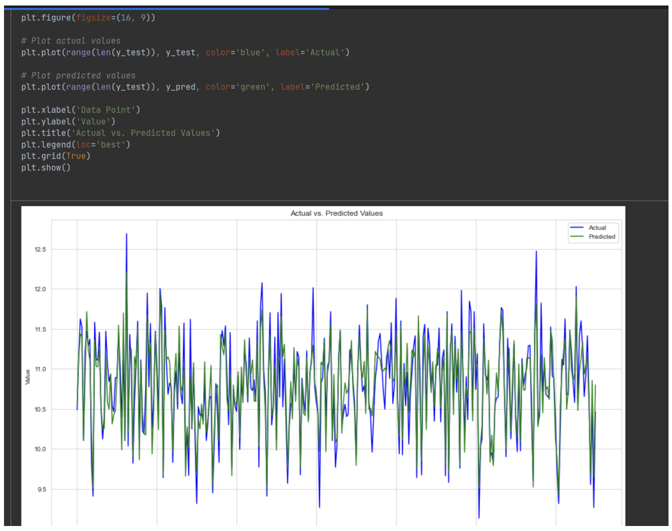
B.1 Output Screenshots



Exporting the model

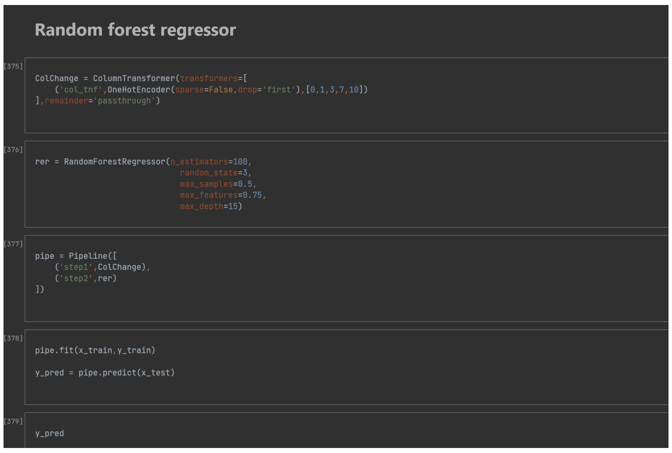
```
[397]  import pickle

[398]  pickle.dump(df,open('df.pkl','wb'))

[399]  pickle.dump(pipe,open('pipe.pkl','wb'))
```

Figure B.1: Exporting the model built in jupyter for configuration



```
plt.figure(figsize=(16, 9))

# Plot actual values
plt.plot(range(len(y_test)), y_test, color='blue', label='Actual')

# Plot predicted values
plt.plot(range(len(y_test)), y_pred, color='green', label='Predicted')

plt.xlabel('Data Point')
plt.ylabel('Value')
plt.title('Actual vs. Predicted Values')
plt.legend(loc='best')
plt.grid(True)
plt.show()
```

Figure B.2: Actual vs Predicted prices of laptops using random forest regressor



Random forest regressor

```
[375]
ColChange = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse=False,drop='first'),[0,1,3,7,10])
],remainder='passthrough')
```

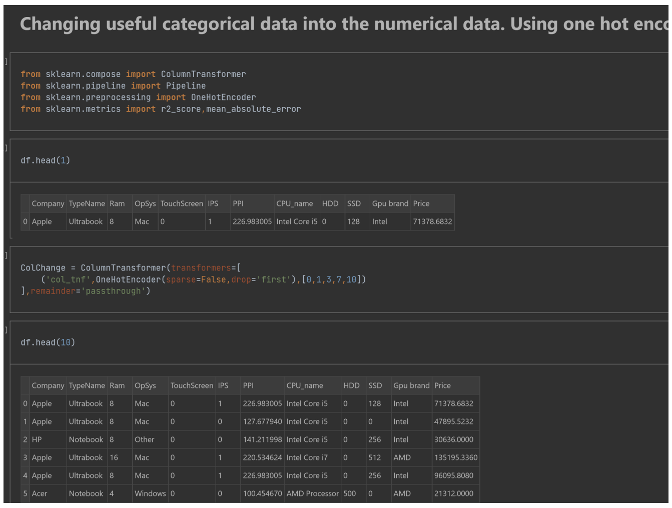
```
[376]
rer = RandomForestRegressor(n_estimators=100,
                              random_state=3,
                              max_samples=0.5,
                              max_features=0.75,
                              max_depth=15)
```

```
[377]
pipe = Pipeline([
    ('step1',ColChange),
    ('step2',rer)
])
```

```
[378]
pipe.fit(x_train,y_train)

y_pred = pipe.predict(x_test)
```

```
[379]
y_pred
```

Figure B.3: Creating instance of random forest regressor and hypertuned paramters



Changing useful categorical data into the numerical data. Using one hot enc

```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import r2_score,mean_absolute_error
```

```
df.head(1)
```

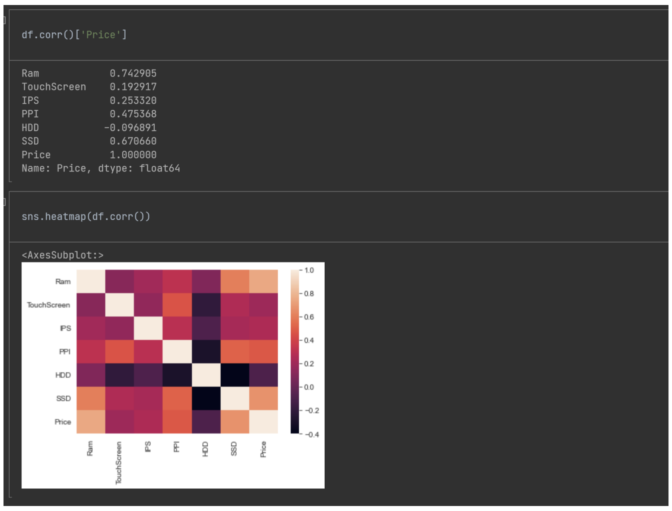
	Company	TypeName	Ram	OpSys	TouchScreen	IPS	PPI	CPU_name	HDD	SSD	Gpu brand	Price
0	Apple	Ultrabook	8	Mac	0	1	226.983005	Intel Core i5	0	128	Intel	71378.6832

```
ColChange = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse=False,drop='first'),[0,1,3,7,10])
],remainder='passthrough')
```

```
df.head(10)
```

	Company	TypeName	Ram	OpSys	TouchScreen	IPS	PPI	CPU_name	HDD	SSD	Gpu brand	Price
0	Apple	Ultrabook	8	Mac	0	1	226.983005	Intel Core i5	0	128	Intel	71378.6832
1	Apple	Ultrabook	8	Mac	0	0	127.677940	Intel Core i5	0	0	Intel	47895.5232
2	HP	Notebook	8	Other	0	0	141.211998	Intel Core i5	0	256	Intel	30636.0000
3	Apple	Ultrabook	16	Mac	0	1	220.534624	Intel Core i7	0	512	AMD	135195.3360
4	Apple	Ultrabook	8	Mac	0	1	226.983005	Intel Core i5	0	256	Intel	96095.8080
5	Acer	Notebook	4	Windows	0	0	100.454670	AMD Processor	500	0	AMD	21312.0000

Figure B.4: Changing categorical data into numerical data for evaluation



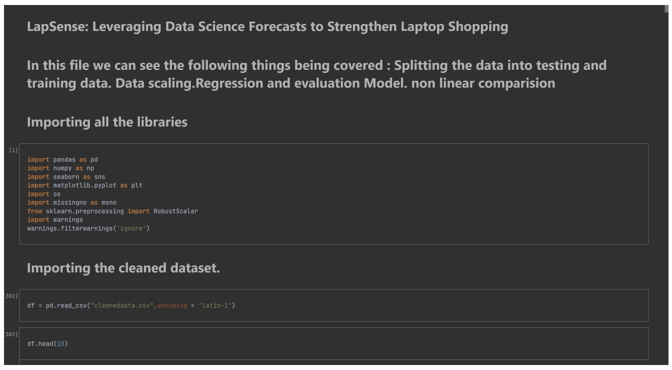
```
df.corr()['Price']
```

```
Ram            0.742905
TouchScreen    0.192917
IPS            0.253320
PPI            0.475368
HDD           -0.096891
SSD            0.670660
Price          1.000000
Name: Price, dtype: float64
```

```
sns.heatmap(df.corr())
```

```
<AxesSubplot:>
```

Figure B.5: Correspondance of all features



LapSense: Leveraging Data Science Forecasts to Strengthen Laptop Shopping

In this file we can see the following things being covered : Splitting the data into testing and training data. Data scaling.Regression and evaluation Model. non linear comparision

Importing all the libraries

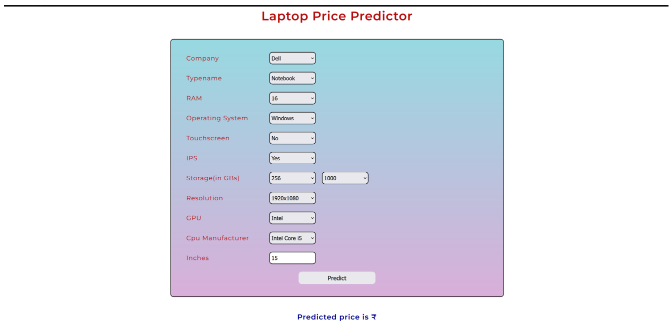
```
[1]
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
import missingno as msno
from sklearn.preprocessing import RobustScaler
import warnings
warnings.filterwarnings('ignore')
```

Importing the cleaned dataset.

```
[302]
df = pd.read_csv("cleaneddata.csv",encoding = 'latin-1')
```

```
[303]
df.head(10)
```

Figure B.6: Importing the libaries required

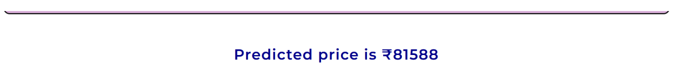


Laptop Price Predictor

Company	Dell	
Typename	Notebook	
RAM	16	
Operating System	Windows	
Touchscreen	No	
IPS	Yes	
Storage(in GBs)	256	1000
Resolution	1920x1080	
GPU	Intel	
Cpu Manufacturer	Intel Core i5	
Inches	15	

Predicted price is ₹

Figure B.7: GUI Interface

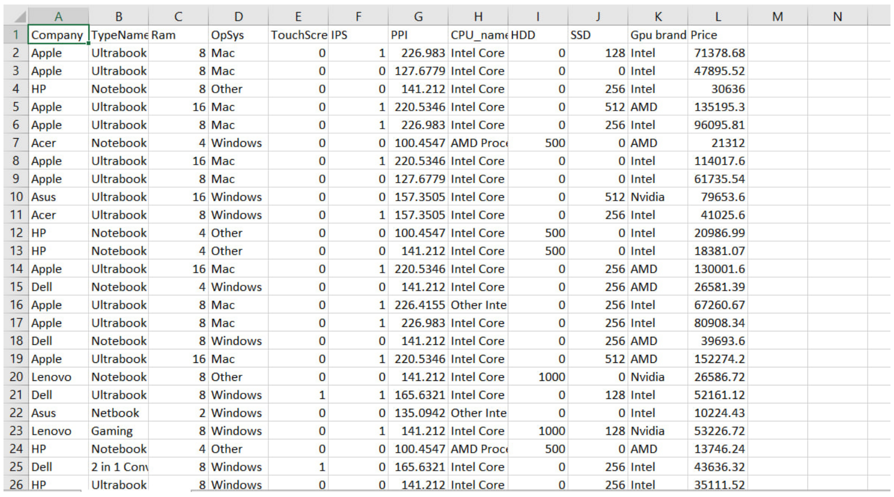


Predicted price is ₹81588

Figure B.8: Predicted Price in website

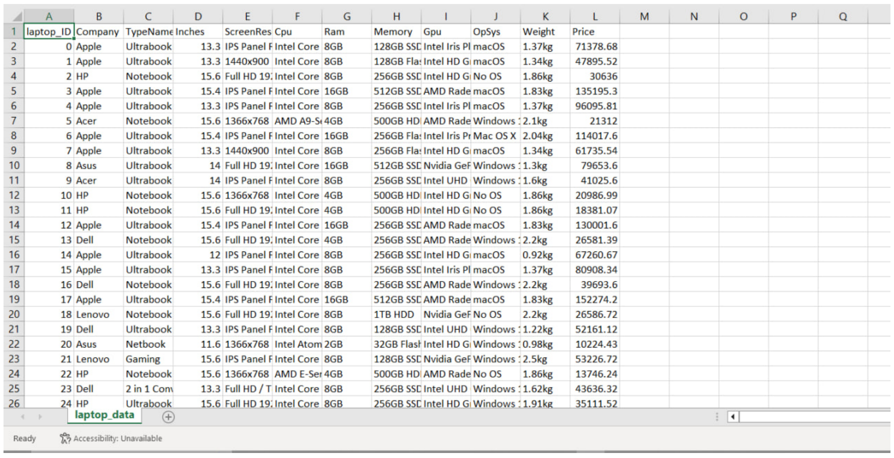
Appendix C

Data sets used in the project



Company	TypeName	Ram	OpSys	TouchScre	IPS	PPI	CPU_name	HDD	SSD	Gpu brand	Price
Apple	Ultrabook	8	Mac	0	1	226.983	Intel Core	0	128	Intel	71378.68
Apple	Ultrabook	8	Mac	0	0	127.6779	Intel Core	0	0	Intel	47895.52
HP	Notebook	8	Other	0	0	141.212	Intel Core	0	256	Intel	30636
Apple	Ultrabook	16	Mac	0	1	220.5346	Intel Core	0	512	AMD	135195.3
Apple	Ultrabook	8	Mac	0	1	226.983	Intel Core	0	256	Intel	96095.81
Acer	Notebook	4	Windows	0	0	100.4547	AMD Proce	500	0	AMD	21312
Apple	Ultrabook	16	Mac	0	1	220.5346	Intel Core	0	0	Intel	114017.6
Apple	Ultrabook	8	Mac	0	0	127.6779	Intel Core	0	0	Intel	61735.54
Asus	Ultrabook	16	Windows	0	0	157.3505	Intel Core	0	512	Nvidia	79653.6
Acer	Ultrabook	8	Windows	0	1	157.3505	Intel Core	0	256	Intel	41025.6
HP	Notebook	4	Other	0	0	100.4547	Intel Core	500	0	Intel	20986.99
HP	Notebook	4	Other	0	0	141.212	Intel Core	500	0	Intel	18381.07
Apple	Ultrabook	16	Mac	0	1	220.5346	Intel Core	0	256	AMD	130001.6
Dell	Notebook	4	Windows	0	0	141.212	Intel Core	0	256	AMD	26581.39
Apple	Ultrabook	8	Mac	0	1	226.4155	Other Inte	0	256	Intel	67260.67
Apple	Ultrabook	8	Mac	0	1	226.983	Intel Core	0	256	Intel	80908.34
Dell	Notebook	8	Windows	0	0	141.212	Intel Core	0	256	AMD	39693.6
Apple	Ultrabook	16	Mac	0	1	220.5346	Intel Core	0	512	AMD	152274.2
Lenovo	Notebook	8	Other	0	0	141.212	Intel Core	1000	0	Nvidia	26586.72
Dell	Ultrabook	8	Windows	1	1	165.6321	Intel Core	0	128	Intel	52161.12
Asus	Netbook	2	Windows	0	0	135.0942	Other Inte	0	0	Intel	10224.43
Lenovo	Gaming	8	Windows	0	1	141.212	Intel Core	1000	128	Nvidia	53226.72
HP	Notebook	4	Other	0	0	100.4547	AMD Proce	500	0	AMD	13746.24
Dell	2 in 1 Conv	8	Windows	1	0	165.6321	Intel Core	0	256	Intel	43636.32
HP	Ultrabook	8	Windows	0	0	141.212	Intel Core	0	256	Intel	35111.52

Figure C.1: Cleaned Dataset from Kaggle and Github



laptop_ID	Company	TypeName	Inches	ScreenRes	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	Apple	Ultrabook	13.3	IPS Panel F	Intel Core	8GB	128GB SSD	Intel Iris Pl	macOS	1.37kg	71378.68
1	Apple	Ultrabook	13.3	1440x900	Intel Core	8GB	128GB Flas	Intel HD G	macOS	1.34kg	47895.52
2	HP	Notebook	15.6	Full HD 19:	Intel Core	8GB	256GB SSD	Intel HD G	No OS	1.86kg	30636
3	Apple	Ultrabook	15.4	IPS Panel F	Intel Core	16GB	512GB SSD	AMD Rade	macOS	1.83kg	135195.3
4	Apple	Ultrabook	13.3	IPS Panel F	Intel Core	8GB	256GB SSD	Intel Iris Pl	macOS	1.37kg	96095.81
5	Acer	Notebook	15.6	1366x768	AMD A9-Se	4GB	500GB HD	AMD Rade	Windows 1	2.1kg	21312
6	Apple	Ultrabook	15.4	IPS Panel F	Intel Core	16GB	256GB Flas	Intel Iris Pr	Mac OS X	2.04kg	114017.6
7	Apple	Ultrabook	13.3	1440x900	Intel Core	8GB	256GB Flas	Intel HD G	macOS	1.34kg	61735.54
8	Asus	Ultrabook	14	Full HD 19:	Intel Core	16GB	512GB SSD	Nvidia GeF	Windows 1	1.3kg	79653.6
9	Acer	Ultrabook	14	IPS Panel F	Intel Core	8GB	256GB SSD	Intel UHD	Windows 1	1.6kg	41025.6
10	HP	Notebook	15.6	1366x768	Intel Core	4GB	500GB HD	Intel HD G	No OS	1.86kg	20986.99
11	HP	Notebook	15.6	Full HD 19:	Intel Core	4GB	500GB HD	Intel HD G	No OS	1.86kg	18381.07
12	Apple	Ultrabook	15.4	IPS Panel F	Intel Core	16GB	256GB SSD	AMD Rade	macOS	1.83kg	130001.6
13	Dell	Notebook	15.6	Full HD 19:	Intel Core	4GB	256GB SSD	AMD Rade	Windows 1	2.2kg	26581.39
14	Apple	Ultrabook	12	IPS Panel F	Intel Core	8GB	256GB SSD	Intel HD G	macOS	0.92kg	67260.67
15	Apple	Ultrabook	13.3	IPS Panel F	Intel Core	8GB	256GB SSD	Intel Iris Pl	macOS	1.37kg	80908.34
16	Dell	Notebook	15.6	Full HD 19:	Intel Core	8GB	256GB SSD	AMD Rade	Windows 1	2.2kg	39693.6
17	Apple	Ultrabook	15.4	IPS Panel F	Intel Core	16GB	512GB SSD	AMD Rade	macOS	1.83kg	152274.2
18	Lenovo	Notebook	15.6	Full HD 19:	Intel Core	8GB	1TB HDD	Nvidia GeF	No OS	2.2kg	26586.72
19	Dell	Ultrabook	13.3	IPS Panel F	Intel Core	8GB	128GB SSD	Intel UHD	Windows 1	1.22kg	52161.12
20	Asus	Netbook	11.6	1366x768	Intel Atom	2GB	32GB Flash	Intel HD G	Windows 1	0.98kg	10224.43
21	Lenovo	Gaming	15.6	IPS Panel F	Intel Core	8GB	128GB SSD	Nvidia GeF	Windows 1	2.5kg	53226.72
22	HP	Notebook	15.6	1366x768	AMD E-Ser	4GB	500GB HD	AMD Rade	No OS	1.86kg	13746.24
23	Dell	2 in 1 Conv	13.3	Full HD / T	Intel Core	8GB	256GB SSD	Intel UHD	Windows 1	1.62kg	43636.32
24	HP	Ultrabook	15.6	Full HD 19:	Intel Core	8GB	256GB SSD	Intel HD G	Windows 1	1.91kg	35111.52

Figure C.2: Raw Dataset from Kaggle and Github