**A**

**PROJECT REPORT**

**ON**

“**Laptop Price Predictions**”

**Submitted by**

**Sanjana Kakad Exam No. B150133807**

**Under the guidance of**

**Prof. D. D. Khartad**

****

**DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION** **ENGINEERING**

**K. K. WAGH INSTITUTE OF ENGINEERING EDUCATION &   
 RESEARCH, NASHIK.**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2021-2022**

# Dissertation Approval Sheet

This is to certify that the project work titled “Laptop Price Prediction”, has been submitted in partial fulfilment of the Bachelor’s degree in Electronics Engineering during the academic year of 2021- 2022 by :

Sanjana Kakad

This project confirms to the standards laid down by the Savirtibai Phule Pune University and has been completed in satisfactory manner as a partial fulfilment for the Bachelor’s degree in Electronics Engineering.

|  |  |  |
| --- | --- | --- |
| **External Examiner** | **Internal Guide** | **Head of Department** |
|  | **(Prof.K. Nirmala Kumari)** | **(Prof.D.M.Chandwadkar)** |

# Acknowledgement

I would like to express my deep gratitude to Prof.D.D. Khartad for his patient guidance, enthusiastic involvement and useful critiques of this project work. My grateful thanks extend to her for her advice and assistance in keeping our progress on schedule right from project synopsis to project report. I am very much thankful to her for encouraging me to acquire knowledge of LATEX system, which is a document preparation system for high-quality

typesetting.

I would also like to extend my thanks to the technicians of the laboratory of the E&TC department for their help in offering me the resources in running the program.

Finally, I wish to thank our parents for their support and encouragement throughout our study.

Sanjana Kakad

# Abstract

This paper presents a Laptop price prediction system by using the supervised machine learning technique. The research uses multiple linear regression as the machine learning prediction method which offered 81% prediction precision. Using multiple linear regression, there are multiple independent variables but one and only one dependent variable whose actual and predicted values are compared to find precision of results. This paper proposes a system where price is dependent variable which is predicted, and this price is derived from factors like Laptop’s model, RAM, ROM (HDD/SSD), GPU, CPU, IPS Display, and Touch Screen.

**Keywords— Multiple Linear regression, Laptop Price, Regression model, Machine Learning.**

# Index

[Dissertation Approval Sheet 2](#_Toc104854910)

[Acknowledgement i](#_Toc104854911)

[Abstract ii](#_Toc104854912)

[Index iii](#_Toc104854913)

[Introduction 1](#_Toc104854914)

[1.1 Overview 1](#_Toc104854915)

[1.2 RELATED WORK 1](#_Toc104854916)

[1.3 Problem Statement 1](#_Toc104854917)

[1.4 Dataset for Laptop Price Prediction 2](#_Toc104854918)

[EDA 3](#_Toc104854919)

[2.1 EDA of Laptop Price Prediction Dataset 3](#_Toc104854920)

[Feature Engineering and Pre-processing of Laptop Price Prediction Model 6](#_Toc104854921)

[Prepare cloud files for deployment 13](#_Toc104854922)

[Test Procedure and Results 17](#_Toc104854923)

[Conclusion and Future Scope 19](#_Toc104854924)

[5.1. Conclusion: 19](#_Toc104854925)

[5.2. Future Scope: 19](#_Toc104854926)

[References 19](#_Toc104854927)

# Introduction

# **Overview**

Laptop price prediction especially when the laptop is coming direct from the factory to Electronic Market/ Stores, is both a critical and important task. The mad rush that we saw in 2020 for laptops to support remote work and learning is no longer there. In India, demand of Laptops soared after the Nationwide lockdown, leading to 4.1-Million-unit shipments in the June quarter of 2021, the highest in the five years. Accurate Laptop price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, RAM, ROM, GPU, CPU, etc. In this paper, we applied different methods and techniques in order to achieve higher precision of the used laptop price prediction.

# **1.2** Related Work

Predicting price of laptops has been studied extensively in various researches. Listian discussed, in her paper written for Master thesis, that regression model that was built using Decision Tree & Random Forest Regressor can predict the price of a laptop that has been leased with better precision than multivariate regression or some simple multiple regression. This is on the grounds that Decision Tree Algorithm is better in dealing with datasets with more dimensions and it is less prone to overfitting and underfitting. The weakness of this research is that a change of simple regression with more advanced Decision Tree Algorithm regression was not shown in basic indicators like mean, variance or standard deviation.

# **1.3** Problem Statement

We will make a project for Laptop price prediction. The problem statement is that if any user wants to buy a laptop, then our application should be compatible to provide a tentative price of laptop according to the user configurations. Although it looks like a simple project or just developing a model, the dataset we have is noisy and needs lots of feature engineering, and pre-processing that will drive your interest in developing this project.

# **1.4** Dataset for Laptop Price Prediction

To support the application of machine learning using the Decision Tree algorithm, of course the sample data is needed. Table below contains data about various laptops and their prices depending on their configuration. Sample data were obtained from https://github.com/Raghavagr/Laptop\_Price\_Prediction/blob/main/laptop\_data.csv

The C4.5 algorithm (used as a Decision Tree Classifier which can be employed to generate a decision, based on sample dataset) starts with the process of selecting the highest gain attribute as the root of the tree, then creates a branch for each value, then divides the case in branches, then repeats the process for each branch until all cases in the branch have the same class.

# EDA

## **2.1** EDA of Laptop Price Prediction Dataset

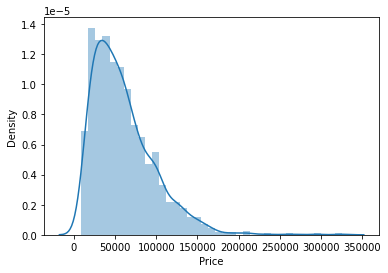
Exploratory analysis is a process to explore and understand the data and data relationship in a complete depth so that it makes feature engineering and machine learning modeling steps smooth and streamlined for prediction. EDA involves Univariate, Bivariate, or Multivariate analysis. EDA helps to prove our assumptions true or false. In other words, it helps to perform hypothesis testing. We will start from the first column and explore each column and understand what impact it creates on the target column. At the required step, we will also perform pre-processing and feature engineering tasks. our aim in performing in-depth EDA is to prepare and clean data for better machine learning modeling to achieve high performance and generalized models. so let’s get started with analyzing and preparing the dataset for prediction.

#### **2.1.1 Distribution of target column**

Working with regression problem statement target column distribution is important to understand.

sns.distplot(data['Price'])

plt.show()



The distribution of the target variable is skewed and it is obvious that commodities with low prices are sold and purchased more than the branded ones.

#### **2.1.2 Company column**

we want to understand how does brand name impacts the laptop price or what is the average price of each laptop brand? If you plot a count plot(frequency plot) of a company then the major categories present are Lenovo, Dell, HP, Asus, etc.

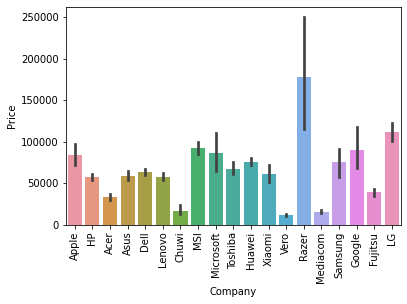
Now if we plot the company relationship with price then you can observe that how price varies with different brands.

#what is avg price of each brand?

sns.barplot(x=data['Company'], y=data['Price'])

plt.xticks(rotation="vertical")

plt.show()



Razer, Apple, LG, Microsoft, Google, MSI laptops are expensive, and others are in the budget range.

#### **3) Type of laptop**

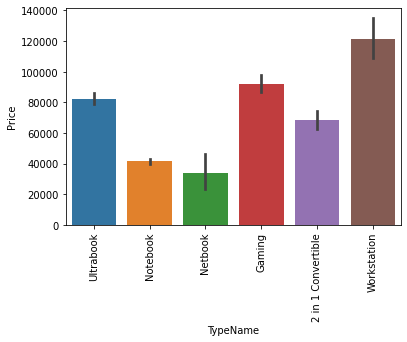
Which type of laptop you are looking for like a gaming laptop, workstation, or notebook. As major people prefer notebook because it is under budget range and the same can be concluded from our data.

#data['TypeName'].value\_counts().plot(kind='bar')

sns.barplot(x=data['TypeName'], y=data['Price'])

plt.xticks(rotation="vertical")

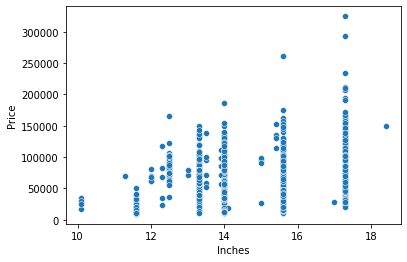
plt.show()



#### **4) Does the price vary with laptop size in inches?**

A Scatter plot is used when both the columns are numerical and it answers our question in a better way. From the below plot we can conclude that there is a relationship but not a strong relationship between the price and size column.

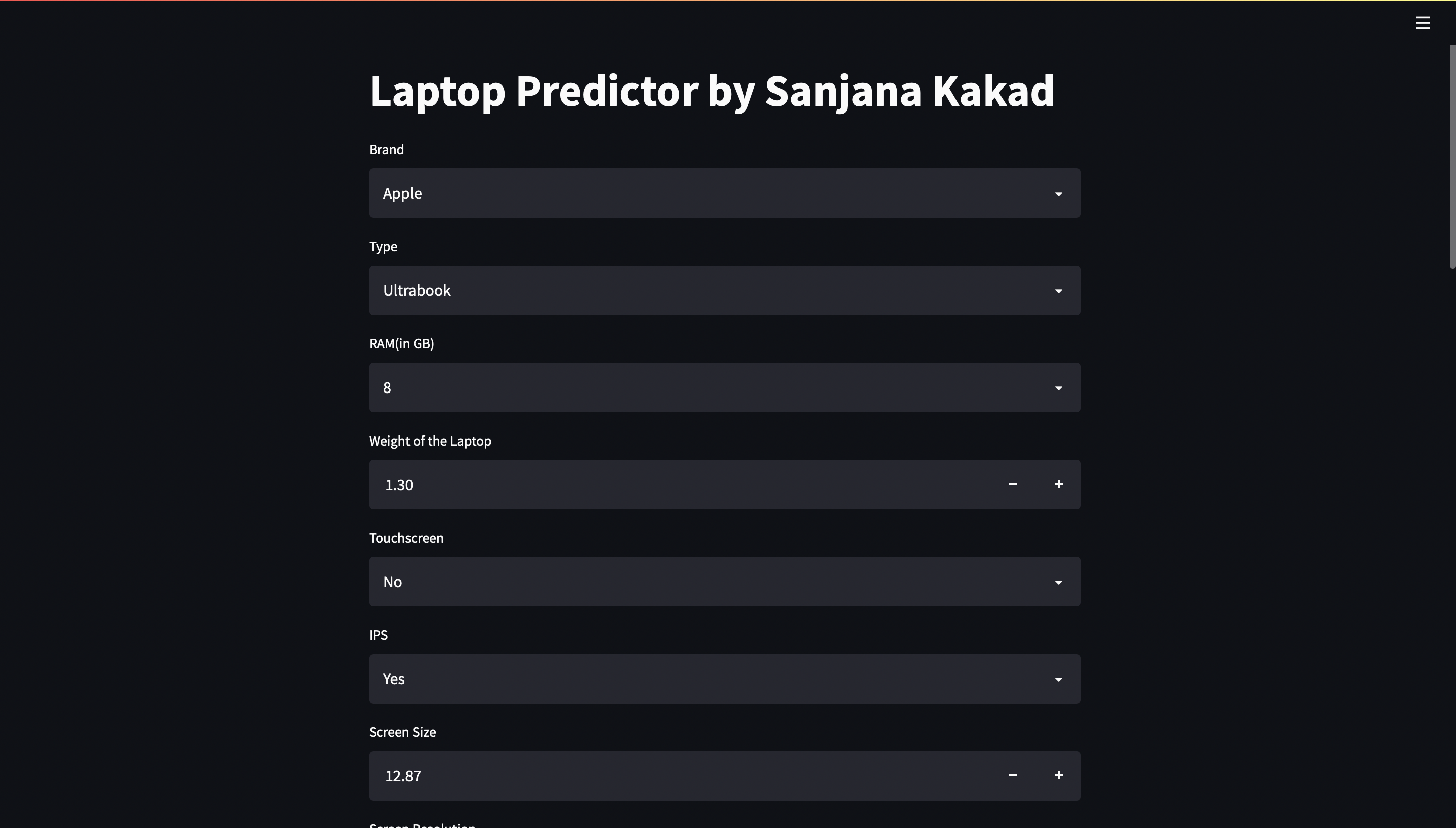
sns.scatterplot(x=data['Inches'],y=data['Price'])



## Feature Engineering and Pre-processing of Laptop Price Prediction Model

Feature engineering is a process to convert raw data to meaningful information. there are many methods that come under feature engineering like transformation, categorical encoding, etc. Now the columns we have are noisy so we need to perform some feature engineering steps.

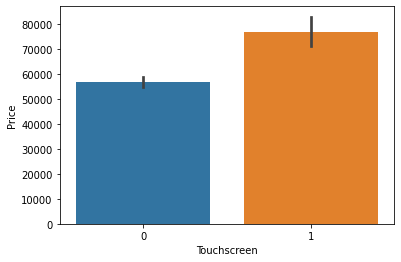
#### **5) Screen Resolution**

screen resolution contains lots of information. before any analysis first, we need to perform feature engineering over it. If you observe unique values of the column then we can see that all value gives information related to the presence of an IPS panel, are a laptop touch screen or not, and the X-axis and Y-axis screen resolution. So, we will extract the column into 3 new columns in the dataset.

**Extract Touch screen information**

It is a binary variable so we can encode it as 0 and 1. one means the laptop is a touch screen and zero indicates not a touch screen.

If we plot the touch screen column against price then laptops with touch screens are expensive which is true in real life.



**Extract IPS Channel presence information**

It is a binary variable and the code is the same we used above. The laptops with IPS channel are present less in our data but by observing relationship against the price of IPS channel laptops are high.

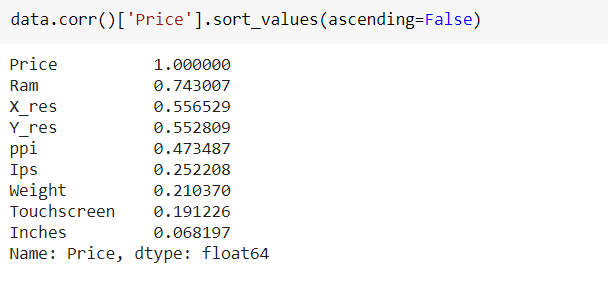
**Extract X-axis and Y-axis screen resolution dimensions**

Now both the dimension are present at end of a string and separated with a cross sign. So first we will split the string with space and access the last string from the list. then split the string with a cross sign and access the zero and first index for X and Y-axis dimensions.

**Replacing inches, X and Y resolution to PPI**

If you find the correlation of columns with price using the **corr** method then we can see that inches do not have a strong correlation but X and Y-axis resolution have a very strong resolution so we can take advantage of it and convert these three columns to a single column that is known as Pixel per inches(PPI). In the end, our goal is to improve the performance by having fewer features.

Now when you will see the correlation of price then PPI is having a strong correlation.



So now we can drop the extra columns which are not of use. At this point, we have started keeping the important columns in our dataset.

#### **6) CPU column**

If you observe the CPU column then it also contains lots of information. If you again use a unique function or value counts function on the CPU column then we have 118 different categories. The information it gives is about pre-processors in laptops and speed.

To extract the preprocessor we need to extract the first three words from the string. we are having an Intel preprocessor and AMD preprocessor so we are keeping 5 categories in our dataset as i3, i5, i7, other intel processors, and AMD processors.

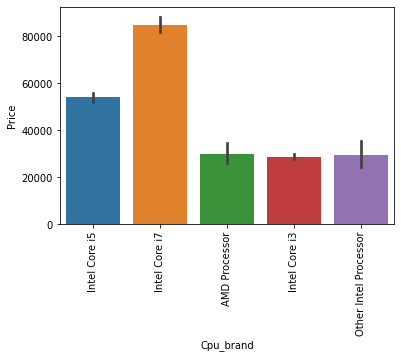
**How does the price vary with processors?**

we can again use our bar plot property to answer this question. And as obvious the price of i7 processor is high, then of i5 processor, i3 and AMD processor lies at the almost the same range. Hence price will depend on the preprocessor.

sns.barplot(x=data['Cpu\_brand'],y=data['Price'])

plt.xticks(rotation='vertical')

plt.show()

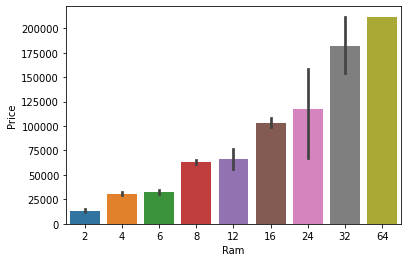


#### **7) Price with Ram**

Again Bivariate analysis of price with Ram. If you observe the plot then Price is having a very strong positive correlation with Ram or you can say a linear relationship.

sns.barplot(data['Ram'], data['Price'])

plt.show()



#### **8) Memory column**

memory column is again a noisy column that gives an understanding of hard drives. many laptops came with HHD and SSD both, as well in some there is an external slot present to insert after purchase. This column can disturb your analysis if not feature engineer it properly. So If you use value counts on a column then we are having 4 different categories of memory as HHD, SSD, Flash storage, and hybrid.

First, we have cleaned the memory column and then made 4 new columns which are a binary column where each column contains 1 and 0 indicate that amount four is present and which is not present. Any laptop has a single type of memory or a combination of two. so in the first column, it consists of the first memory size and if the second slot is present in the laptop then the second column contains it else we fill the null values with zero. After that in a particular column, we have multiplied the values by their binary value. It means that if in any laptop particular memory is present then it contains binary value as one and the first value will be multiplied by it, and same with the second combination. For the laptop which does have a second slot, the value will be zero multiplied by zero is zero.

Now when we see the correlation of price then Hybrid and flash storage have very less or no correlation with a price. We will drop this column with CPU and memory which is no longer required.

#### **9) GPU Variable**

GPU (Graphical Processing Unit) has many categories in data. We are having which brand graphic card is there on a laptop. we are not having how many capacities like (6Gb, 12 Gb) graphic card is present. so we will simply extract the name of the brand.

If you use the value count’s function then there is a row with GPU of ARM so we have removed that row and after extracting the brand GPU column is no longer needed.

#### **10) Operating System Column**

There are many categories of operating systems. we will keep all windows categories in one, Mac in one, and remaining in others. This is a simple and most used feature engineering method, you can try something else if you find more correlation with price.

#Get which OP sys

def cat\_os(inp):

if inp == 'Windows 10' or inp == 'Windows 7' or inp == 'Windows 10 S':

return 'Windows'

elif inp == 'macOS' or inp == 'Mac OS X':

return 'Mac'

else:

return 'Others/No OS/Linux'

data['os'] = data['OpSys'].apply(cat\_os)

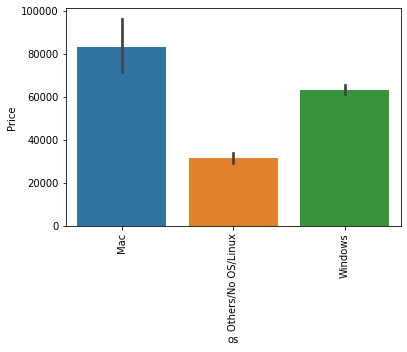
data.drop(columns=['OpSys'],inplace=True)

when you plot price aginst operating system then as usual Mac is most expensive.

sns.barplot(x=data['os'],y=data['Price'])

plt.xticks(rotation='vertical')

plt.show()

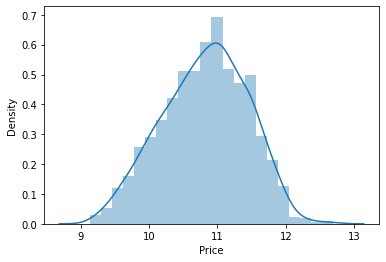


#### **Log-Normal Transformation**

we saw the distribution of the target variable above which was right-skewed. By transforming it to normal distribution performance of the algorithm will increase. we take the log of values that transform to the normal distribution which you can observe below. So while separating dependent and independent variables we will take a log of price, and in displaying the result perform exponent of it.

sns.distplot(np.log(data['Price']))

plt.show()



**3.4 Software Design**

**3.4.1 Modern Tools Used**

#### Deploy Application to Heroku

Now we are ready to deploy our website and make it available for the public to use.

### Prepare cloud files for deployment

**1) Procfile**

Create a file name Procfile which is an initiator file for Heroku. It only contains one line of code that says which file to run or it simply runs your python file.

web: sh setup.sh && streamlit run app.py

**2) requirements**

Create a file named requirements.txt. It is a text file that contains the name and version of a library that you have used to create your project. we need to define the libraries used to cloud so that when we deploy it creates a complete setup by installing required files. If you do not specify the version then it will install the current updated version of the library. we have used only four libraries for creating streamlit apps.

streamlit

sklearn

numpy

pandas

**3) setup file**

Create a file name *setup.sh* which contains how to create the directory structure in the cloud.

mkdir -p ~/.streamlit/

echo "

[server]n

port = $PORTn

enableCORS = falsen

headless = truen

n

" > ~/.streamlit/config.toml

#### **Upload Code to Github**

Log in to your GitHub account and create a new repository of the project name of your choice. Now you can either use the upload button to upload all files by selecting from a local file manager. And you can also use the GIT bash command as stated below to upload your code.

After creating a new repository copy the ssh link of a repository to connect with a repository. And then line by line follow the below commands.

git init #initialize empty repository

git remote add origin #connect to repository

git pull origin master #pull initial chnges

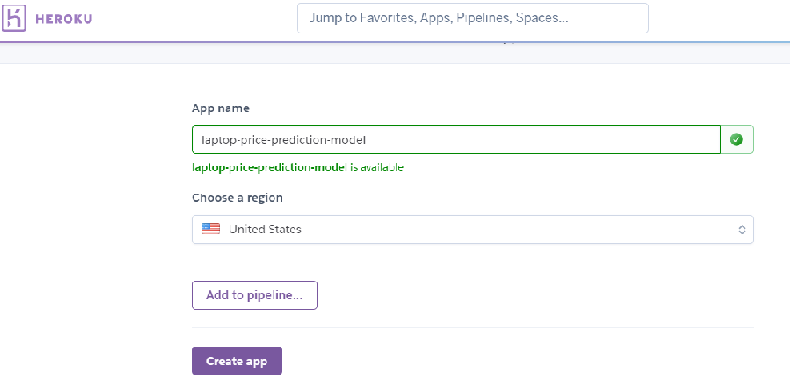
gid add -A #to add files in staging area

git commit -m initial commit

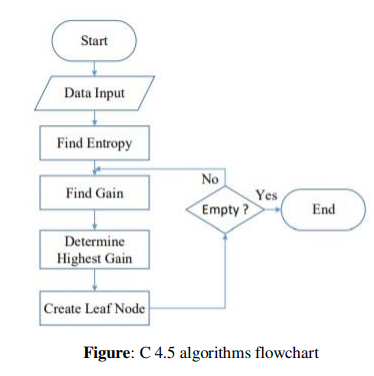
git push origin master #push all files to github

#### **Deploy to Heroku**

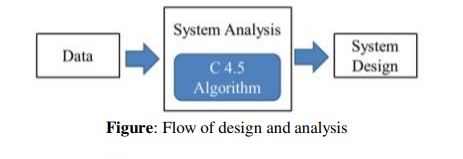
Log in or register to [Heroku](https://dashboard.heroku.com/) if you do not have an account. After you log in in the top-right corner you will have the option of new. create a new app. Give a unique name to your code and this name will be your website URL followed by the Heroku domain and let the region be united states only.



* + 1. **Algorithms**



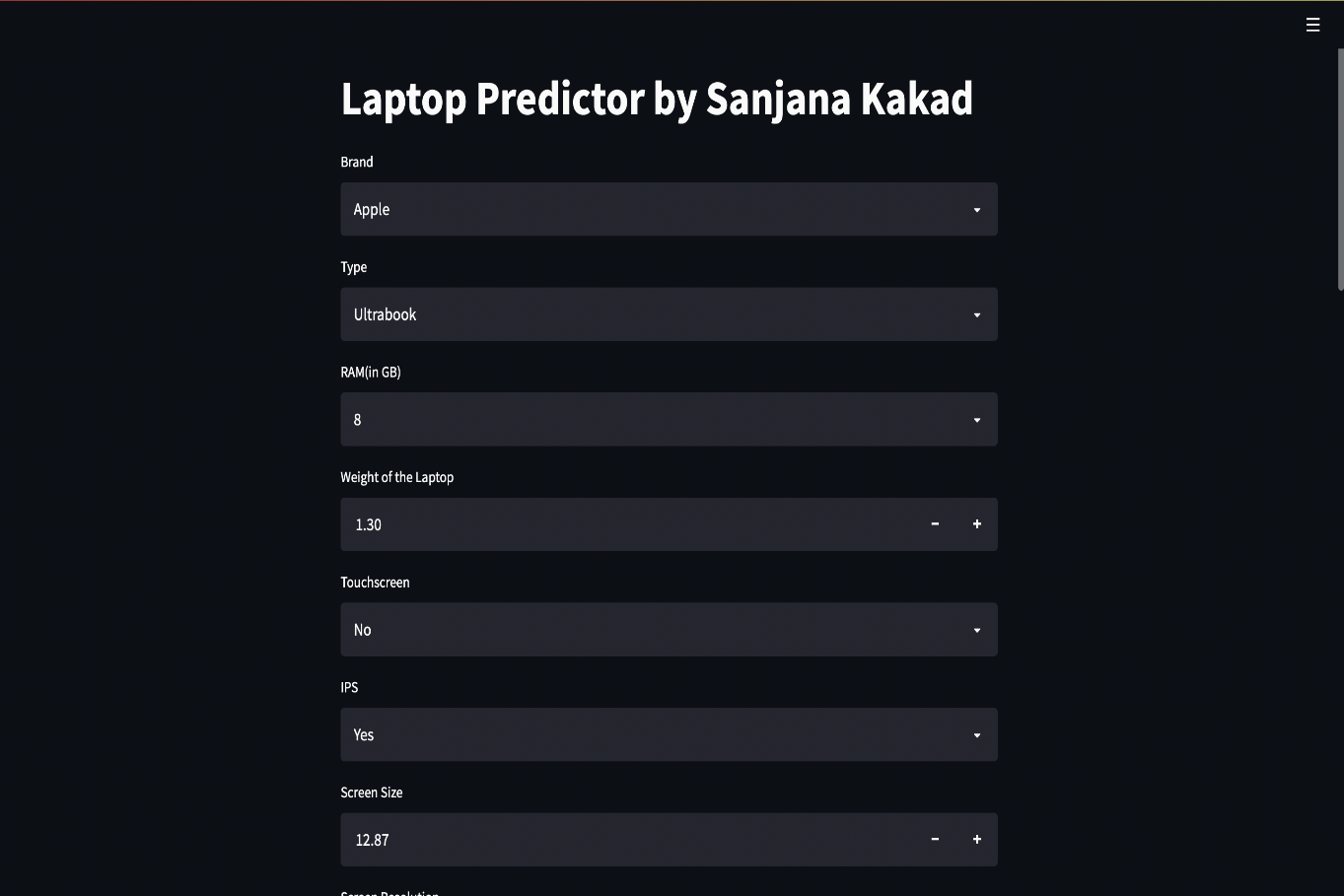
**3.4.3 Flowchart**

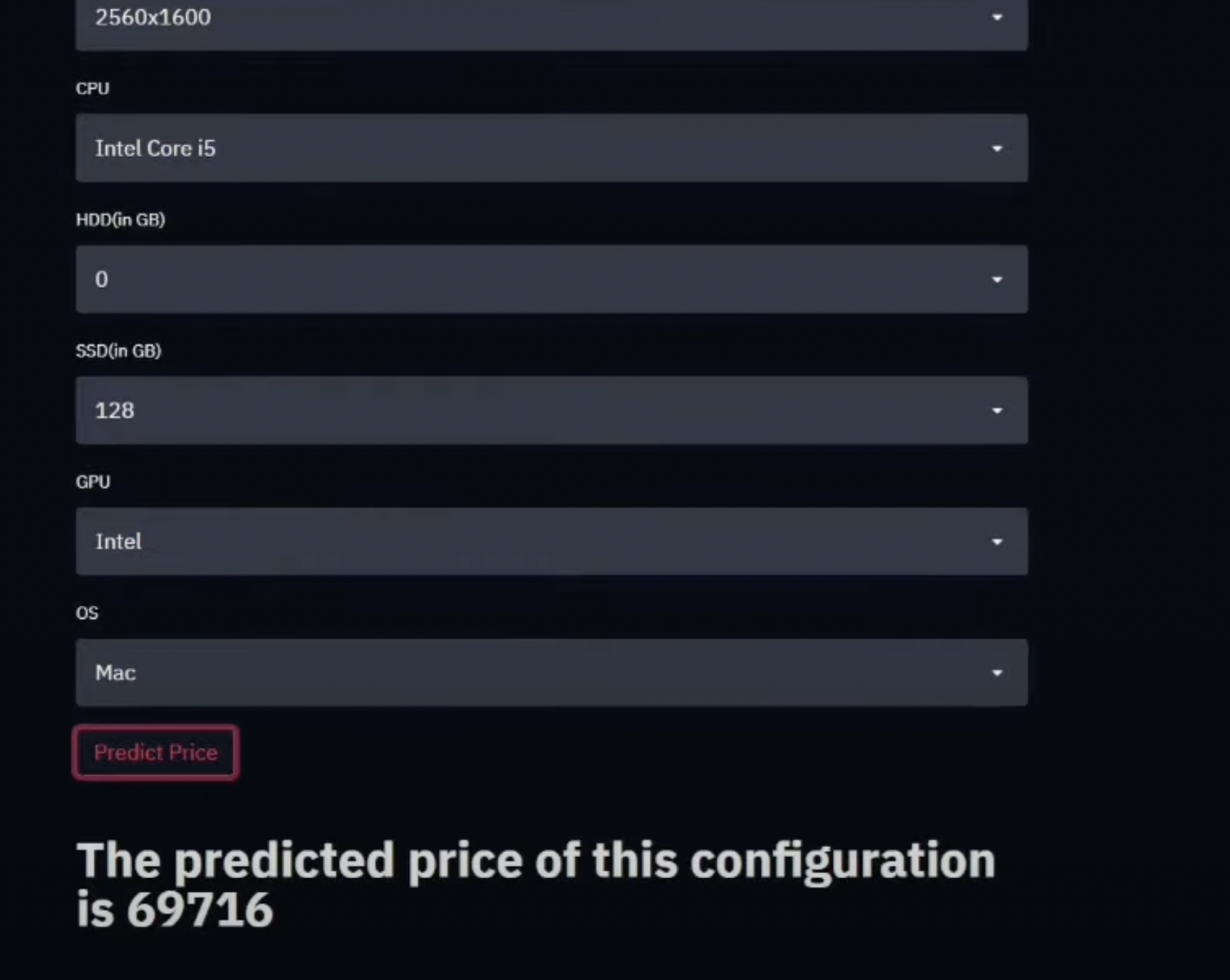


# 

# Test Procedure and Results

Streamlit library is used to build this WebApp UI. Streamlit is an (open-source Python library) that makes it easy to create and share, custom web apps for machine learning and data science. Result with backend code is shown in following figures.





# Conclusion and Future Scope

# 

## 5.1. Conclusion:

Predicting something through the application of machine learning using the Decision Tree algorithm makes it easy for students, especially in determining the choice of laptop specifications that are most desirable for students to meet student needs and in accordance with the purchasing power of students. Students no longer need to look for various sources to find laptop specifications that are needed by students in meeting the needs of students, because the laptop specifications from the results of the machine learning application have provided the most desirable specifications with their prices of laptops.

## 5.2. Future Scope:

We would like to extend this research by adding more company’s data and check the prediction accuracy. For those companies where availability of financial news is a challenge, we would be using twitter data for similar analysis. We can also incorporate similar strategies for algorithmic trading.

# 

# References

[1]. Sorower MS. A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis. 2010 Dec;18.

[2]. Pandey M, Sharma VK. A decision tree algorithm pertaining to the student performance analysis and prediction. International Journal of Computer Applications. 2013 Jan 1;61(13).

[3]. Priyama A, Abhijeeta RG, Ratheeb A, Srivastavab S. Comparative analysis of decision tree classification algorithms. International Journal of Current Engineering and Technology. 2013 Jun;3(2):334-7

[4]. Streamlit.io, Kaggle.com, Wikipedia.com