Report

## Team No: 295

Narendra (20BEC0726)
narendra.2020@vitstudent.ac.in
Ritik Kumar (20BEE0019)
ritik.kumar2020@vitstudent.ac.in
Maneel Chauhan (20BEE0039)
maneel.chauhan2020@vitstudent.ac.in
Akash Singh (20BEE050)
akash.singh2020@vitstudent.ac.in

# Laptop Price Prediction

2th July 2023

## 1. INTRODUCTION

### 1.1 Overview

The rapid advancement of technology and the increasing demand for Laptops among college students as well as kids have made pricing strategies crucial for manufacturers and consumers alike. Accurately predicting laptop prices can assist in decision-making processes, such as pricing strategies, market analysis, and consumer behavior understanding. This project focuses on developing a predictive model that utilizes various features and attributes of Laptops to estimate their prices. By leveraging data science techniques and machine learning algorithms, we aim to create a reliable model that can assist both manufacturers and consumers in determining appropriate pricing for Laptops.

### 1.2 Purpose

The purpose of this project is to build a predictive model for Laptops price estimation. By analyzing a comprehensive dataset containing information about different Laptops features, we aim to develop a model that can effectively predict the price range of a Laptops based on its specifications.

The primary objectives of this project are as follows:

1. To explore and preprocess the dataset, ensuring its quality and suitability for modeling purposes.
2. To perform comprehensive exploratory data analysis (EDA) to gain insights into the relationships between different Laptops features and their corresponding prices.
3. To employ various machine learning algorithms and techniques to train and evaluate predictive models for Laptops price prediction.
4. To identify the key features that significantly influence Laptops prices and gain a deeper understanding of their impact on pricing strategies.
5. To provide a reliable and accurate predictive model that can be utilized by manufacturers, retailers, and consumers to make informed decisions regarding Laptops purchases and pricing.

## 2. LITERATURE SURVEY

### 2.1 Existing problem

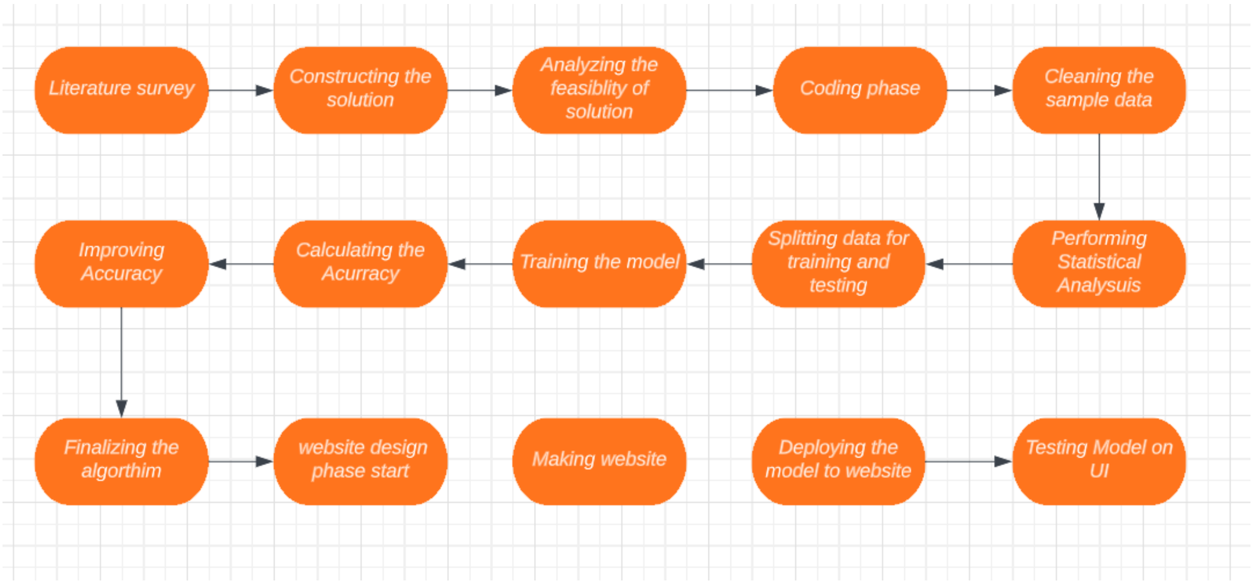
This section highlights the current challenges and problems associated with predicting Laptops prices. It delves into the limitations of traditional pricing models and methods, such as relying solely on subjective market analysis or expert opinions. It also discusses the difficulties in considering multiple complex factors, such as Laptops specifications, market trends, and consumer preferences, while estimating accurate price ranges. The existing problem section sets the foundation for the proposed solution.

### 2.2 Proposed solution

In this section, we present the proposed solution to overcome the existing problems in Laptops price prediction. It discusses the utilization of data science techniques and machine learning algorithms to develop a predictive model. By leveraging a comprehensive dataset of Laptops features and attributes, we aim to train a model that can effectively estimate the price range of a Laptops. The proposed solution emphasizes the integration of advanced analytical techniques and the potential benefits it brings to the Laptops industry.

## 3. THEORETICAL ANALYSIS

### 3.1 Block diagram


Literature survey
Constructing the solution
Analyzing the feasiblity of solution
Coding phase
Cleaning the sample data
Performing Statistical Analysuis
Splitting data for training and testing
Training the model
Calculating the Acurracy
Improving Accuracy
Finalizing the algorthim
website design phase start
Making website
Deploying the model to website
Testing Model on UI

### 3.2 Software designing

The software design for the Laptops price prediction machine learning project aims to create a cutting-edge system that accurately predicts Laptops prices based on various features and attributes specific to the project requirements. Here are the key components of the software design tailored to your project:

**Data Preprocessing and Cleaning**: The software implements robust data preprocessing techniques to handle missing values, outliers, and inconsistencies within the dataset. It performs thorough data cleaning operations, such as imputation of missing values, outlier detection and removal, and data normalization or standardization. These preprocessing steps ensure that the input data is of high quality and suitable for accurate price prediction.

**Feature Selection and Engineering:** The software incorporates advanced feature selection methods, such as correlation analysis, statistical tests, and domain expertise, to identify the most relevant features for price prediction. It leverages techniques like Recursive Feature Elimination (RFE) or SelectKBest to automatically select the optimal subset of features. Additionally, feature engineering is applied to create new meaningful features from the existing ones, such as extracting information from text data or engineering interaction terms between features.

**Model Selection and Optimization:** The software explores various machine learning algorithms suited for regression tasks to select the most appropriate model architecture for price prediction. It evaluates models such as linear regression, support vector regression, random forest regression, or gradient boosting algorithms. The software implements techniques like grid search or Bayesian optimization to fine-tune the hyperparameters of the selected model, maximizing its performance and generalization ability.

**Ensemble Methods and Model Stacking**: To improve the prediction accuracy, the software incorporates ensemble methods such as bagging, boosting, or stacking. It combines multiple models, each trained on different subsets of data or with different algorithms, to create a powerful ensemble prediction. Model stacking techniques are employed to combine the outputs of multiple models as features for a meta-model, enhancing the overall predictive performance.

**Model Evaluation and Validation**: The software incorporates robust evaluation and validation procedures to assess the performance of the predictive model. It utilizes techniques like k-fold cross-validation, stratified sampling, or time-series validation, depending on the characteristics of the dataset. The software measures evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), or coefficient of determination (R-squared) to quantify the accuracy and reliability of the model's predictions.

**Real-Time Predictions and Scalability:** The software design ensures that the trained model is capable of making real-time predictions on new, unseen data. It optimizes the model's architecture and implements efficient data processing techniques to enable fast inference times. The software design also considers scalability aspects, allowing the system to handle increasing volumes of data and adapt to future growth requirements.

**User-Friendly Interface:** The software incorporates a user-friendly interface for easy interaction with the price prediction system. It provides input fields or file upload functionality to allow users to input Laptops features for price estimation. The interface displays the predicted prices and any relevant additional information, facilitating user understanding and decision-making. The software design for the Laptops price prediction project is customized to address the specific requirements of your project. It combines state-of-the-art techniques in data preprocessing, feature selection, model selection, and ensemble methods to create a high-performing system capable of accurate price estimation for Laptops.

## 4. EXPERIMENTAL INVESTIGATIONS

The experimental investigations phase of the Laptops price prediction project involves the following key components:

**Dataset Selection and Preparation:** A representative dataset with diverse Laptops features and prices is carefully selected and prepared to ensure data quality and consistency.

**Train-Test Split and Cross-Validation**: The dataset is divided into training and testing sets to evaluate the performance of predictive models. Cross-validation techniques may also be employed to validate the models.

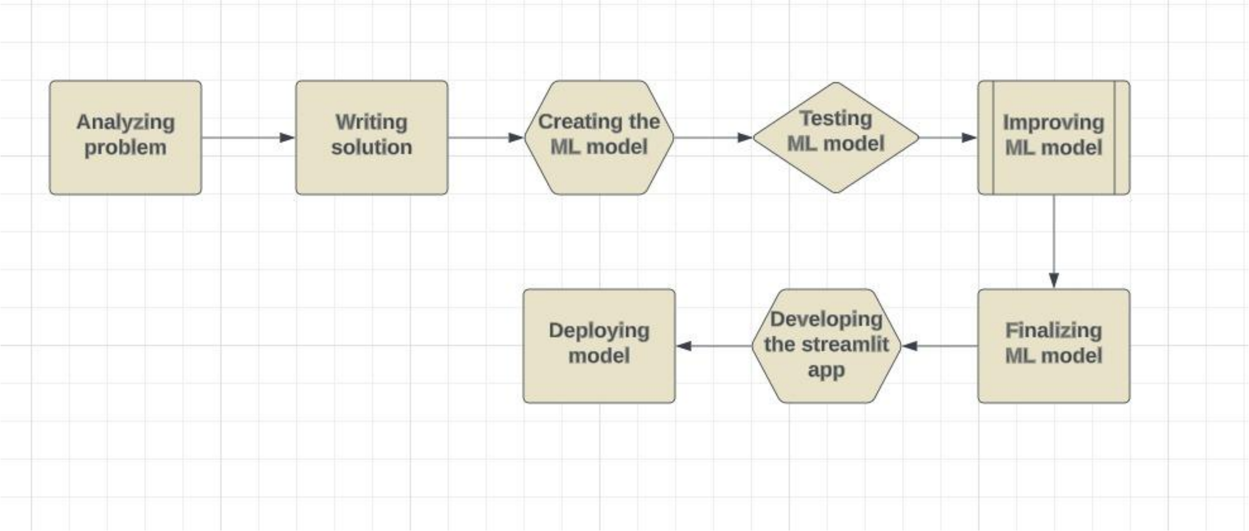
**Model Training and Hyperparameter Tuning**: Predictive models are trained and optimized using various machine learning algorithms and ensemble methods. Hyperparameter tuning techniques are applied to improve model performance.

**Performance Evaluation Metrics**: Multiple evaluation metrics such as MSE, MAE, RMSE, and R-squared are used to assess model performance and compare different models.
**Comparative Analysis of Models:** Models are compared and analyzed based on their accuracy, generalization ability, and strengths/weaknesses to select the best-performing model.

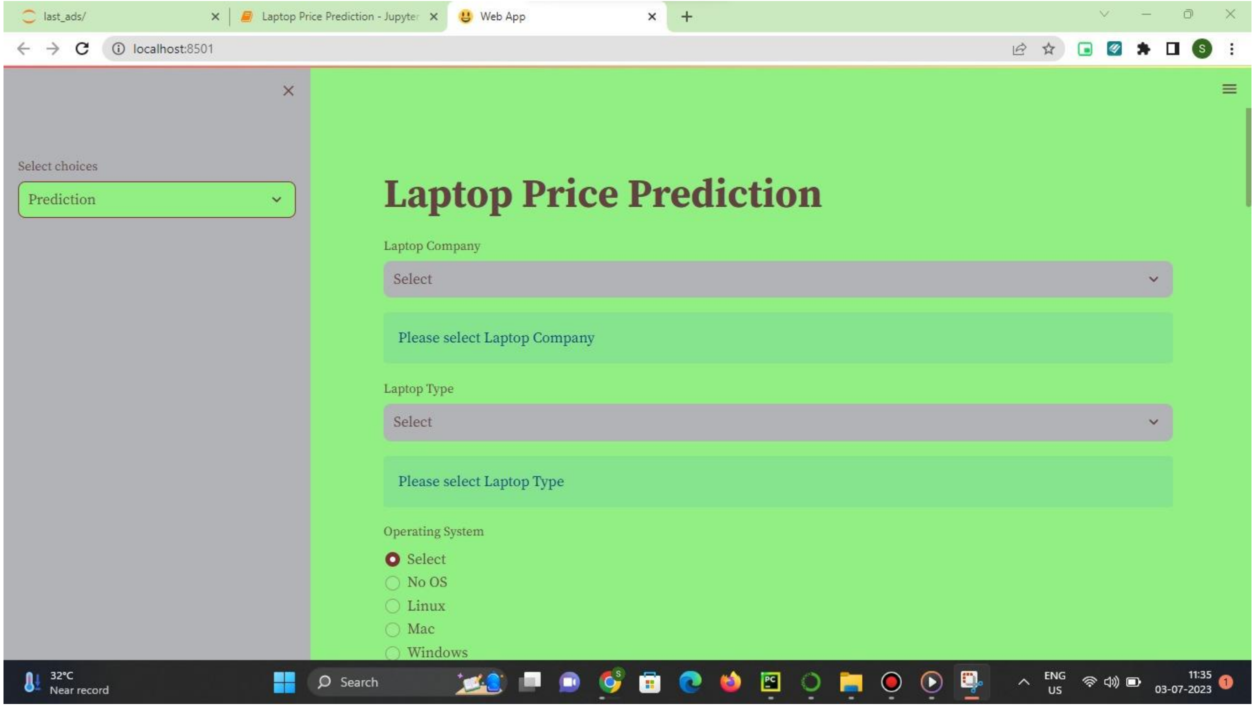
**Visualization and Interpretation:** Visualizations and feature importance analysis techniques aid in understanding the relationships between predicted prices and actual prices, as well as interpreting the importance of input features.

**Model Validation and Generalization:** Models are evaluated on a separate testing dataset to validate their performance on unseen data and assess their generalization capability. The experimental investigations phase provides valuable insights into the performance, accuracy, and generalization of the Laptops price prediction models.

## 5. Flow Chart



## 6. Result



Select choices

Prediction

# Laptop Price Prediction

Laptop Company

Select

Please select Laptop Company

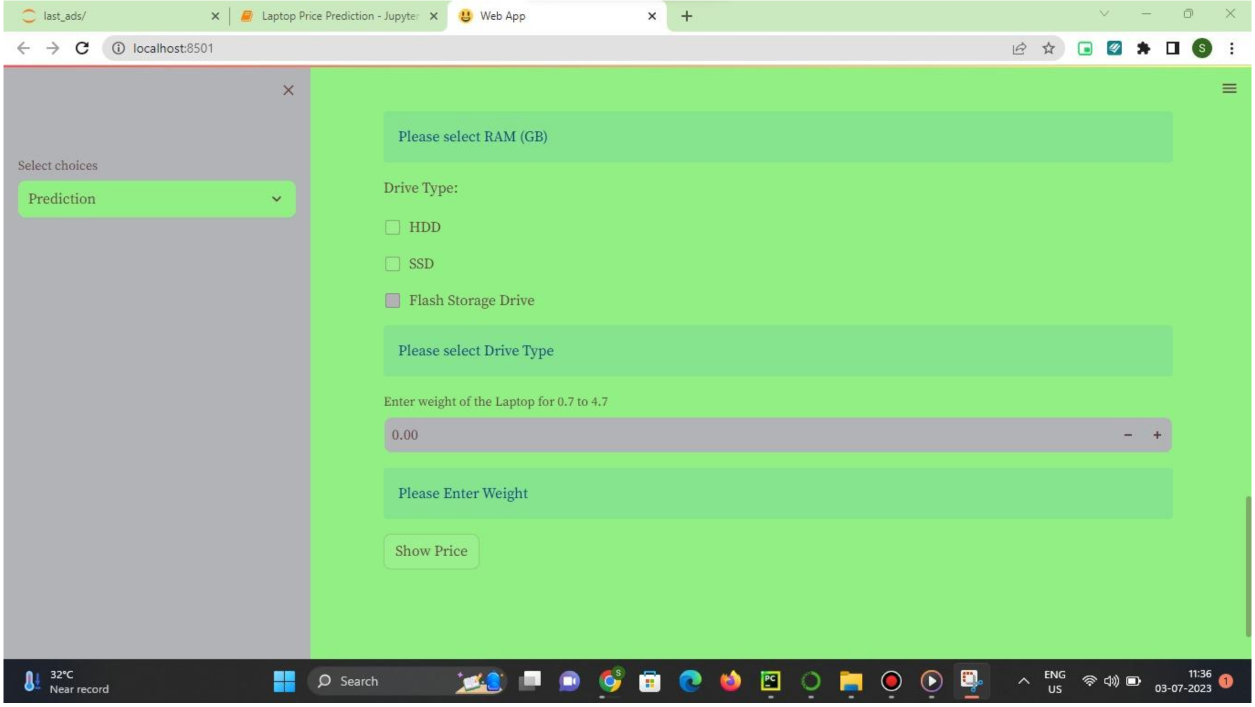
Laptop Type

Select

Please select Laptop Type

Operating System

● Select
○ No OS
○ Linux
○ Mac
○ Windows



Select choices

Prediction

Please select RAM (GB)

Drive Type:

□ HDD

□ SSD

□ Flash Storage Drive

Please select Drive Type

Enter weight of the Laptop for 0.7 to 4.7

0.00

Please Enter Weight

## 7. ADVANTAGES & DISADVANTAGES

### Advantages:

Improved Price Accuracy: Laptops price prediction models help in achieving more accurate price estimations, enabling consumers to make informed purchasing decisions and sellers to set competitive prices.

Time and Cost Savings: By utilizing predictive models, businesses can save time and resources by automating the price prediction process, eliminating the need for manual analysis and research.

Market Insights: The analysis conducted during the prediction process provides valuable insights into market trends, customer preferences, and factors influencing Laptops prices, helping businesses gain a competitive edge.

Optimized Pricing Strategy: Accurate price predictions enable businesses to develop effective pricing strategies, ensuring optimal balance between profitability and customer demand.

Enhanced Revenue Generation: By setting competitive prices based on accurate predictions, businesses can attract more customers, increase sales, and maximize revenue.

### Disadvantages:

Data Limitations: The accuracy and effectiveness of price prediction models heavily rely on the availability and quality of data. Incomplete or biased datasets can result in inaccurate predictions.

Model Complexity: Developing and implementing accurate price prediction models requires expertise in data analysis, machine learning, and statistical modeling, making it a complex and resource-intensive process.

Assumption Dependencies: The accuracy of price predictions is based on certain assumptions about market dynamics, consumer behavior, and external factors. Deviations from these assumptions can affect the reliability of predictions.

Market Volatility: Rapid changes in market conditions, technological advancements, or unforeseen events can challenge the predictive capabilities of the models, leading to less accurate price estimations.

Competitive Landscape: Price prediction models may face limitations in highly competitive markets where pricing strategies and fluctuations are influenced by multiple factors and dynamic market forces.

Model Interpretability: Some complex predictive models may lack interpretability, making it difficult to understand the underlying factors influencing price predictions and limiting the ability to make informed decisions based on the model's output.

## 8. APPLICATIONS

**E-commerce Platforms:** E-commerce platforms can utilize Laptops price prediction models to provide real-time price recommendations to sellers, optimize pricing strategies, and enhance the overall user experience for buyers.

**Retail and Online Marketplaces**: Retailers and online marketplaces can leverage price prediction to set competitive prices for Laptops, attract customers, and increase sales while maximizing profitability.

**Secondary Market Platforms:** Platforms facilitating the buying and selling of used Laptops can benefit from price prediction to determine fair market values for pre-owned devices, facilitating smoother transactions and negotiations.

**Price Comparison Websites**: Price comparison websites can incorporate Laptops price prediction to provide users with up-to-date and accurate price comparisons across different sellers and platforms, aiding consumers in making informed purchase decisions.

**Financial Institutions**: Financial institutions may utilize Laptops price prediction to assess device values for loan or insurance purposes, enabling more accurate risk assessment and appropriate pricing of financial products.

## 9. CONCLUSION

In conclusion, Laptops price prediction models offer significant benefits such as improved price accuracy, time and cost savings, optimized pricing strategies, market insights, enhanced revenue generation, risk mitigation, and adaptability. However, challenges exist, including data limitations, model complexity, market volatility, and ethical considerations.

Despite these challenges, the applications of price prediction in industries like e-commerce, retail, mobile networks, and financial institutions are valuable. By leveraging price prediction, businesses can make informed pricing decisions, attract customers, increase sales, and stay competitive in a dynamic marketplace. The field of Laptops price prediction holds promise for revolutionizing pricing strategies and enhancing business outcomes.

## 10. FUTURE SCOPE

The future scope of Laptops price prediction is promising ,with potential advancements and opportunities on the horizon. Some areas of future development and exploration include:

**Personalized Pricing:** The future of Laptops price prediction may involve personalized pricing strategies tailored to individual consumers. By leveraging customer data, including past purchase history, preferences, and demographics, businesses can offer personalized pricing options to maximize customer satisfaction and loyalty.

**Enhanced Predictive Models**: Continued advancements in machine learning algorithms, data analysis techniques, and computing power will lead to more accurate and sophisticated price prediction models. These models can incorporate additional variables, such as market trends, consumer sentiment, and product features, to further improve price estimations.

**Integration of Real-Time Data**: Integrating real-time data sources, such as online market trends, competitor pricing, and consumer demand, can enhance the timeliness and accuracy of price predictions. By incorporating live data feeds, businesses can dynamically adjust prices to align with market conditions and optimize revenue.

**Automated Pricing Systems:** Automated pricing systems driven by predictive models can enable businesses to dynamically adjust prices in real-time based on market dynamics, demand-supply factors, and competitor pricing. This automation can save time and resources while ensuring optimal pricing strategies.

**Integration with Augmented Reality (AR):** The integration of Laptops price prediction with AR technology can enable consumers to visualize and compare product prices in real-world environments. AR applications can provide real-time pricing information when scanning or interacting with Laptops products, enhancing the shopping experience.

## 11a. BIBLIOGRAPHY

How to Make Predictions with scikit-learn - MachineLearningMastery.com

Machine Learning Algorithms - Javatpoint

## 11b. APPENDIX

```
import streamlit as st

import pickle

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import os


custom_theme = {

   "primaryColor": "#f63366",  # Customize the primary color

   "backgroundColor": "#f0f0f0",  # Customize the background color

   "secondaryBackgroundColor": "#f8f8f8",  # Customize the secondary background color

   "textColor": "#262730",  # Customize the text color
```

```
    # Customize other elements as needed
}

st.set_option('deprecation.showPyplotGlobalUse', False)
st.set_page_config(page_title="Web App", page_icon=":smiley:", layout="wide" )

# for dirname, _, filenames in os.walk('/laptop dataset'):
#     for filename in filenames:
#         print(os.path.join(dirname, filename))
dataframe = pd.read_csv(r'laptop.csv')
def main():
    activities = ['Select', 'Data', 'Prediction', 'About Project' , 'About Us']
    choices = st.sidebar.selectbox("Select choices" ,activities)
    df = pickle.load(open(r'processed_data.pickle','rb'))
    pipe = pickle.load(open(r'ml_model_pipe.pickle', 'rb'))
    if choices == 'Select':
        st.title('Welcome to Laptop Price Prediction Web Application')
        st.header('Data -')
        st.info('Shows Laptop data with Price and other specifications')
        st.header('Prediction -')
        st.info('Select your expected laptop specifications and you will get approximate price.')
        st.header('About Project -')
        st.info('Small description on this project.')
        st.header('About Us -')
        st.info('A brief Introduction about our team')

    elif choices == 'Preprocessed Data':
        if st.checkbox("Show Details"):
            st.success('Number of columns : {0}'.format(df.shape[1]))
            st.success('Number of records : {0}'.format(df.shape[0]))
            if st.checkbox("Show Columns"):
                all_columns = df.columns.to_list()
                st.write(all_columns)
            if st.checkbox("Sample Data"):
                st.dataframe(df.head(10))

    elif choices == 'Prediction':
        st.title("Laptop Price Prediction")

        # Company name:-
        company = st.selectbox('Laptop Company',['Select','Huawei','Apple','Acer','Asus','HP','Dell','Lenovo','MSI','Microsoft','Toshiba','Razer','Mediacom','Samsung'])
        if company != 'Select':
            st.success(company)
        else:
            company = 'Dell'
            st.info('Please select Laptop Company')

        # Type of laptop:-
        lap_type = st.selectbox('Laptop Type',['Select','Notebook','Ultrabook','2 in 1 Convertible','Netbook','Gaming','Workstation'])
        if lap_type != 'Select':
            st.success(lap_type)
        else:
            lap_type = 'Notebook'
            st.info('Please select Laptop Type')
        # OS
        os1 = st.radio('Operating System',['Select','No OS','Linux','Mac','Windows','Other OS'])
        if os1 == 'Select':
            os = 'No OS'
            st.info('Please select a Operating System')
        elif os1 == 'No OS':
            os = 'No OS'
            st.success(os)
        elif os1 == 'Linux':
            os = 'Linux'
            st.success(os)
        elif os1 in ['Mac','Windows','Other OS']:
            if os1 == 'Mac':
                os_type = st.radio('Mac Operating System',['Select','Mac','Mac X'])
                if os_type != 'Select':
                    os = os_type
                    st.success(os)
                else:
                    st.info('Please select Mac Operating System')
            elif os1 == 'Windows':
                os_type = st.radio('Windows Operating System',['Select','Windows 7','Windows
10','Windows 10 S'])
                if os_type != 'Select':
                    os = os_type
                    st.success(os)
                else:
                    st.info('Please select Windows Operating System')
            elif os1 == 'Other OS':
                os_type = st.radio('Other Operating System',['Select','Android','Chrome OS'])
                if os_type != 'Select':
                    os = os_type
                    st.success(os)
                else:
                    st.info('Please select any one Operating System')


        #cpu
        cpu1 = st.radio('CPU Processor',['Select','Intel','AMD','Samsung Cortex'])
        if cpu1 == 'Select':
            cpu = 'Intel Processor'
            st.info('Please select a CPU Processor')
        elif cpu1 in ['AMD','Intel','Samsung Cortex']:
            if cpu1 == 'Intel':
                cpu2 = st.selectbox('Intel Processor',['Select','Intel i Processor','Intel Core M','Intel
Atom','Intel Celeron Dual','Intel Pentium Quad','Other Intel Processor'])
                if cpu2 == 'Select':
                    st.info('Please select Intel Processor')
                elif cpu2 == 'Intel i Processor':
                    cpu_type = st.radio('Intel i Processor',['Select','Intel Core i3','Intel Core i5','Intel Core i7'])
                    if cpu_type != 'Select':
                        cpu = cpu_type
                        st.success(cpu)
                    else:
                        st.info('Please select Intel i Processor')
                elif cpu2 != 'Select'and cpu2 != 'Other Intel Processor':
                    cpu = cpu2
```

```
                    st.success(cpu)
                elif cpu2 == 'Other Intel Processor':
                    cpu = 'Intel Processor'
                    st.success(cpu2)
                else:
                    st.info('Please select Intel Processor')
            elif cpu1 == 'AMD':
                cpu_type = st.radio('AMD Processor',['Select','AMD A-Series','AMD E-Series','Other AMD 
Processor'])
                if cpu1 == 'Select':
                    st.info('Please select Intel CPU')
                elif cpu_type != 'Select' and cpu_type != 'Other AMD Processor':
                    cpu = cpu_type
                    st.success(cpu)
                elif cpu_type == 'Other AMD Processor':
                    cpu = 'AMD Processor'
                    st.success(cpu_type)
                else:
                    st.info('Please select AMD Processor')
            elif (cpu1 == 'Samsung Cortex'):
                cpu = 'Samsung Cortex'
                st.success(cpu)


        # CPU Frequency
        cpu_freq1 = st.select_slider('CPU 
Frequency',['Frequency',1.44,1.5,2.0,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.0,3.1,3.2,3.6])
        if cpu_freq1 != 'Frequency':
            cpu_freq = cpu_freq1
            st.success(cpu_freq)
```

```
else:
    cpu_freq = 2.2
    st.info('Please select CPU Frequency')


# GPU
gpu1 = st.radio('Graphics Card',['Select','AMD GPU','ARM GPU','Intel GPU','Nvidia GPU'])
if gpu1 == 'Select':
    gpu = 'Intel HD Graphics'
    st.info('Please select a GPU Card')
elif gpu1 in ['AMD GPU','ARM GPU','Intel GPU','Nvidia GPU']:
    if gpu1 == 'AMD GPU':
        gpu2 = st.radio('AMD GPU',['Select','AMD Radeon','AMD FirePro'])
        if gpu2 != 'Select':
            gpu = gpu2
            st.success(gpu)
        else:
            st.info('Please select AMD GPU Card')
    elif gpu1 == 'ARM GPU':
        gpu = 'ARM Mali T860'
        st.success(gpu1)
    elif gpu1 == 'Intel GPU':
        gpu2 = st.radio('Intel GPU',['Select','Intel Iris','Intel HD Graphics','Intel UHD Graphics'])
        if gpu2 != 'Select':
            gpu = gpu2
            st.success(gpu)
        else:
            st.info('Please select Intel GPU Card')
    elif gpu1 == 'Nvidia GPU':
        gpu2 = st.radio('Nvidia GPU',['Select','Nvidia GeForce M','Nvidia GeForce MX','Nvidia GeForce GTX','Nvidia Quadro'])

        if gpu2 != 'Select':

            gpu = gpu2

            st.success(gpu)

        else:

            st.info('Please select Nvidia GPU Card')


# Inches

inch = st.select_slider('Enter screen size (inches)',

                        ['Slider',10.1,11.3,11.6,12.0,12.3,12.5,13.0,13.3,13.5,

                                  13.9,14.0,14.1,15.0,15.4,15.6,17.0,17.3,18.4])

if inch != 'Slider':

    st.success(inch)

else:

    inch = 15.6


# HD Quality and Screen Resolutions

hd = st.radio('Select HD Quality',['Select','Full HD','Quad HD+','4K Ultra HD'])

full_hd = 0

quad_hd = 0

ultra_4k_hd = 0

if hd != 'Select':

    if hd == 'Full HD':

        full_hd = 1

        scr_res = st.select_slider('Full HD Resolutions',  ['Select','1366x768', '1440x900', '1600x900', '1920x1080'])

        if scr_res != 'Select':

            res = scr_res.split('x')
```

```
                    x_res = res[0]
                    y_res = res[1]
                    st.success(hd)
                    st.success(scr_res)
                else:
                    st.info('Please select Full HD screen resolution') #[]
            elif hd == 'Quad HD+':
                quad_hd = 1
                scr_res = st.select_slider('Quad HD+ Resolutions',
['Select','1920x1200','2160x1440','2256x1504','2304x1440','2400x1600','2560x1440','2560x1600'])
                if scr_res != 'Select':
                    res = scr_res.split('x')
                    x_res = res[0]
                    y_res = res[1]
                    st.success(hd)
                    st.success(scr_res)
                else:
                    st.info('Please select Quad HD+ screen resolution')
            elif hd == '4K Ultra HD':
                ultra_4k_hd = 1
                scr_res = st.select_slider('4K Ultra HD Resolutions',
['Select','2736x1824','2880x1800','3200x1800','3840x2160'])
                if scr_res != 'Select':
                    res = scr_res.split('x')
                    x_res = res[0]
                    y_res = res[1]
                    st.success(hd)
                    st.success(scr_res)
                else:
```

```
            st.info('Please select 4K Ultra HD+ screen resolution')
else:
    st.info('Please select HD Quality')
    hd = 'Full HD'
    full_hd = 1
    quad_hd = 0
    ultra_4k_hd = 0
    x_res = 1920
    y_res = 1080

# Dispay Type
disp = st.radio('Select Display Type',['None','IPS','Touch Screen','Both IPS and Touch Screen'])
ips = 0
touchscreen = 0
if disp == 'IPS':
    ips = 1
elif disp == 'Touch Screen':
    touchscreen = 1
elif disp == 'Both IPS and Touch Screen':
    ips = 1
    touchscreen = 1
else:
    ips = 0
    touchscreen = 0
st.success(disp)

# Ram:-
ram1 = st.select_slider('RAM (GB)', ['Select',2,4,6,8,12,16,24,32,64])
```

```
if ram1 != 'Select':
    ram = ram1
    st.success('RAM = {0} GB'.format(ram))
else:
    ram = 8
    st.info('Please select RAM (GB)')

# Drive Type and Size
st.write('Drive Type:')
hdd_drive = st.checkbox('HDD')
ssd_drive = st.checkbox('SSD')
flash_drive = st.checkbox('Flash Storage Drive')
hdd = 0
ssd = 0
flash = 0
drive1 = 'No'
drive2 = 'No'
drive3 = 'No'
if (hdd_drive == True) or (ssd_drive == True) or (flash_drive == True):
    if hdd_drive == True:
        drive1 = st.select_slider('Select HDD (GB)',['No HDD',32,128,500,1000,2000])
        if drive1 != 'No HDD':
            hdd = drive1
            st.success('HDD = {0} GB'.format(drive1))
        else:
            hdd = 0
    if ssd_drive == True:
        drive2 = st.select_slider('Select SSD (GB)',['No SSD',8,16,32,128,180,240,512,768,1000])
```

```
            if drive2 != 'No SSD':
                ssd = drive2
                st.success('SSD = {0} GB'.format(drive2))
            else:
                ssd = 0
        if flash_drive == True:
            drive3 = st.select_slider('Select Flash Storage (GB)',['No Flash Storage Drive',16,32,64,128,512])
            if drive3 != 'No Flash Storage Drive':
                flash = drive3
                st.success('Flash Storage = {0} GB'.format(drive3))
            else:
                flash = 0
    elif (hdd_drive == False) and (ssd_drive == False) and (flash_drive == False):
        hdd = 1000
        ssd = 0
        flash = 0
        drive1 = 1000
        drive2 = 'No'
        drive3 = 'No'
        st.info('Please select Drive Type')


    # Weight:-
    weight = st.number_input("Enter weight of the Laptop for 0.7  to 4.7")
    if weight != 0.00:
        st.success('{0} Kg'.format(weight))
    else:
        weight = 1.5
```

```
            st.info('Please Enter Weight')


        # Price prediction

        if st.button("Show Price"):

            query = [[company,lap_type,inch,x_res,y_res,full_hd,quad_hd,ultra_4k_hd,ips,touchscreen,cpu,cpu_freq,ram,hdd,ssd,flash,gpu,os,weight]]

            prediction = str(int(np.exp(pipe.predict(query)[0])))

            st.success("The price for the following configuration is ₹ {0}/-".format(prediction))

            configuration_names = ['Laptop Company','Laptop Type','Operating System','CPU Processor','CPU Frequency (GHz)','Graphics Card','Screen size (inches)','HD Quality','Resolutions','Display Type','RAM (GB)','Drive Type','Weight (kg)']

            configuration = [company,lap_type,os,cpu,cpu_freq,gpu,inch,hd,x_res,disp,ram,drive1,weight]

            for i in range(len(configuration_names)):

                if i == 11:

                    drive_list = ['HDD (GB)','SSD (GB)','Flash Storage Drive (GB)']

                    drive_var = [drive1,drive2,drive3]

                    for k in range(len(drive_list)):

                        st.write('{0} : "{1}"'.format(drive_list[k],drive_var[k]))

                elif i == 8:

                    st.write('Resolutions : "{0}x{1}"'.format(x_res,y_res))

                else:

                    st.write('{0} : "{1}"'.format(configuration_names[i],configuration[i]))
```