Mini Project Report

TITLE: Laptop Price Prediction

AIM: Implement an application for selected problem statements using a suitable frontend language such as Java, Python, C++, Netbeans, and .Net.

OBJECTIVE: To develop a mini project using suitable technology to provide a solution to the selected problem statement.

OUTCOME: Develop an application using suitable technology to provide a solution to the selected problem statement.

TEMPLATE FOR MINI PROJECT REPORT

(Font: Times New Roman, Font Size: for title-14, for Content-12)

Page 1: Cover page

Page 2: Certificate page with guide name

Page 3: Abstract

Page 4: Index (Refer to below template)

Chapter	Contents		Page No.
1.		Introduction	
	a.	Problem Statement	
	b.	Project Idea	
	c	Motivation	
	d	Scope	
	e	Literature Survey / Requirement Analysis	
2.		Project Design	
	a	H/W, S/W, resources, requirements & their detailed explanation	
	b	Dataset Design	
	c	Hours estimation	
3.		Module Description	
	a	Block diagram with an explanation of each module	
4.		Results & Discussion	
	a	Source code	
	b	Module Description	
	c	Test cases	
5	a	Conclusion References	

Chapter 1: Introduction

a) Problem Statement

The objective of this machine learning project is to create a predictive model that accurately estimates the price of laptops based on various features and specifications. The model should take into account a wide range of attributes and specifications, such as processor speed, RAM, storage capacity, brand, screen size, and more, to make reliable price predictions.

b) Project Idea

The project aims to develop a machine learning application that predicts laptop prices based on factors such as

- RAM
- Processor
- Storage Capacity
- Brand
- Screen Size
- Other

This project idea not only predicts laptop prices but also demonstrates the impact of sentiment analysis on pricing, potentially uncovering valuable insights for e-commerce platforms and laptop manufacturers.

c) Motivation

The motivation behind creating a Laptop Price Prediction tool stems from the following key factors:

- Empowering Software Developers
- Enhance Technology
- Data-Driven Decision-Making
- Educational Value
- Career Guidance

d) Scope

The scope of the "Laptop Price Prediction" project is broad and holds significant potential for various stakeholders, including Students, Seller, and the technology industry as a whole

The key areas within the project's scope:

- Price Insights of laptops for users
- Price Benchmarking of laptop for users
- Educational Tool
- Data-Driven Decision-Making
- Continuous Learning

e) Literature Survey/ Requirement

Analysis Literature Survey:

- Market Basket Analysis
- Machine Learning Algorithms
- User Interface Design
- Data Sources

Requirement Analysis:

- User Needs
- Data Collection
- Machine Learning Model
- User Interface
- Data Visualization
- Customization
- Educational Value
- Deployment
- Scalability

Chapter 2: Project Design

a) H/W , S/W , resources, requirements & their detail explanation

Hardware Requirements:

- Server/Cloud Hosting : The project requires a server or cloud hosting service to deploy the web application for user access.A scalable solution is preferred to accommodate potential traffic.
- Storage : Sufficient storage space is needed to store the application code, datasets, and machine learning model.
- Web Hosting : A web hosting service is necessary to host the project's website and databases.

Software Requirements:

- Operating System :The project can be developed on various operating systems, including Windows, Linux, or macOS. The application should be compatible with multiple OS environments.
- Programming Languages: The primary programming languages for development include Python for machine learning and web development, HTML/CSS for frontend for interactive features.
- Development Tools :Code editors or integrated development environments (IDEs) like Visual Studio Code, Jupyter Notebook.
- Database Management System (DBMS) : LocalDatabase
- Machine Learning Libraries : Python libraries such as scikit-learn, NumPy, and pandas for developing the salary prediction model
- Web Development Libraries
- Version Control : Virtual environment, docker and git.

Resources:

- Development Team : The project requires a team of developers, data analysis.
- Data Sources: Reliable datasets containing information on software developers, including their location, education, experience, and salaries.
- Educational Material: Educational resources on machine learning, web development, and UI/UX design for team members.
- Server/Cloud Service: Access to a server or cloud hosting service to deploy the web application.

b) Dataset Design

- The dataset design involves selecting and structuring data related to software developers, including attributes like location, education, years of experience, and salary.
- Data preprocessing will include handling missing values, encoding categorical variables, and scaling numeric features.
- Datasets will be collected from public surveys and may be further curated to ensure data quality.

c) Hours estimation

Development of the project, including machine learning model creation, dataset design, web application development, and UI/UX design, is estimated to require approximately 1200-1500 hours.

Chapter 3: Module Description

a) Block diagram with explanation of each module.

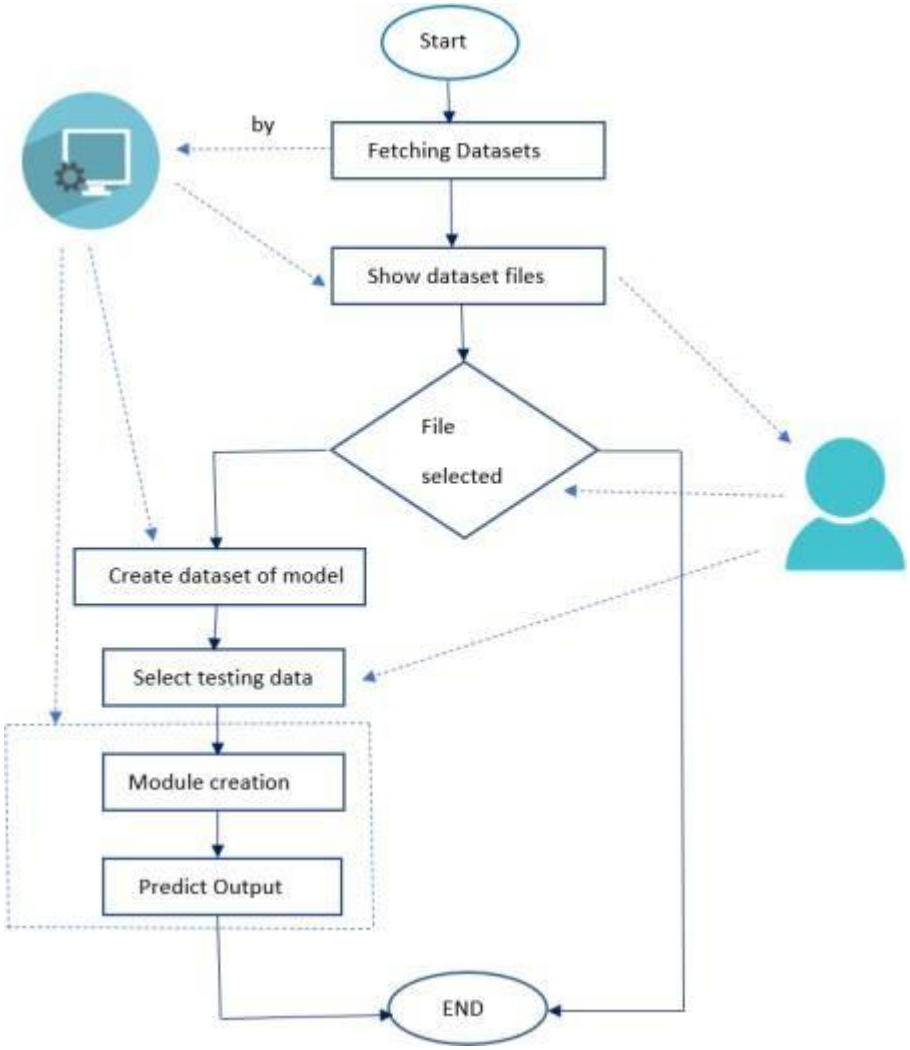


Fig: Architecture

Chapter 4: Results & Discussion

a) Source code:

laptop_price_predictor.ipynb

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("survey_results_public.csv")
df.head()
df = df[["Country", "EdLevel", "YearsCodePro",
"Employment", "ConvertedComp"]]
df = df.rename({"ConvertedComp": "Salary"}, axis=1)
df.head()
df = df[df["Salary"].notnull()]
df.head()
df.info()
df = df.dropna()
df.isnull().sum()
df = df[df["Employment"] == "Employed full-time"]
df = df.drop("Employment", axis=1)
df.info()
df['Country'].value_counts()
def shorten_categories(categories, cutoff):
    categorical_map = {}
    for i in range(len(categories)):
        if categories.values[i] >= cutoff:
            categorical_map[categories.index[i]] = categories.index[i]
        else:
            categorical_map[categories.index[i]] = 'Other'
    return categorical_map
country_map = shorten_categories(df.Country.value_counts(),
400) df['Country'] = df['Country'].map(country_map)
df.Country.value_counts()
fig, ax = plt.subplots(1,1, figsize=(12, 7))
df.boxplot('Salary', 'Country', ax=ax)
plt.suptitle('Salary (US$) v Country')
plt.title('')
plt.ylabel('Salary')
plt.xticks(rotation=90)
plt.show()
df = df[df["Salary"] <= 250000]
df = df[df["Salary"] >= 10000]
df = df[df['Country'] != 'Other']
fig, ax = plt.subplots(1,1, figsize=(12, 7))
df.boxplot('Salary', 'Country', ax=ax)
plt.suptitle('Salary (US$) v Country')
plt.title("plt.ylabel('Salary')
plt.xticks(rotation=90) plt.show()
df["YearsCodePro"].unique() def
clean_experience(x):
    if x == 'More than 50 years':
```

```
        return 50
    if x == 'Less than 1 year':
        return 0.5
    return float(x)

df['YearsCodePro'] =
df['YearsCodePro'].apply(clean_experience)
df["EdLevel"].unique()
def clean_education(x):
    if 'Bachelor’s degree' in x:
        return 'Bachelor’s degree'
    if 'Master’s degree' in x:
        return 'Master’s degree'
    if 'Professional degree' in x or 'Other doctoral' in x:
        return 'Post grad'
    return 'Less than a Bachelors'

df['EdLevel'] = df['EdLevel'].apply(clean_education)
df["EdLevel"].unique()
from sklearn.preprocessing import LabelEncoder
le_education = LabelEncoder()
df['EdLevel'] = le_education.fit_transform(df['EdLevel'])
df["EdLevel"].unique()

le_country = LabelEncoder()
df['Country'] =
le_country.fit_transform(df['Country'])
df["Country"].unique()X = df.drop("Salary", axis=1)
y = df["Salary"]
from sklearn.linear_model import LinearRegression
linear_reg = LinearRegression()
linear_reg.fit(X, y.values)
y_pred = linear_reg.predict(X)

from sklearn.metrics import mean_squared_error,
mean_absolute_error import numpy as np
error = np.sqrt(mean_squared_error(y,
y_pred)) error
from sklearn.tree import DecisionTreeRegressor
dec_tree_reg =
DecisionTreeRegressor(random_state=0)
dec_tree_reg.fit(X, y.values)
y_pred = dec_tree_reg.predict(X)
error = np.sqrt(mean_squared_error(y,
y_pred)) print("${:,.02f}".format(error))
from sklearn.ensemble import RandomForestRegressor
random_forest_reg = RandomForestRegressor(random_state=0)
random_forest_reg.fit(X, y.values)
y_pred = random_forest_reg.predict(X)
error = np.sqrt(mean_squared_error(y,
y_pred)) print("${:,.02f}".format(error))
from sklearn.model_selection import GridSearchCV

max_depth = [None, 2,4,6,8,10,12
```

parameters = {"max_depth": max_depth}

```
regressor = DecisionTreeRegressor(random_state=0)
gs = GridSearchCV(regressor, parameters,
scoring='neg_mean_squared_error') gs.fit(X, y.values)
regressor = gs.best_estimator_

regressor.fit(X, y.values)
y_pred = regressor.predict(X)
error = np.sqrt(mean_squared_error(y,
y_pred)) print("${:,.02f}".format(error))
X
# country, edlevel, yearscode
X = np.array([["United States", 'Master's degree', 15
]]) X
X[:, 0] = le_country.transform(X[:,0])
X[:, 1] = le_education.transform(X[:,1])
X = X.astype(float)
X
y_pred =
regressor.predict(X) y_pred
import pickle
data = {"model": regressor, "le_country": le_country, "le_education":
le_education} with open('saved_steps.pkl', 'wb') as file:
   pickle.dump(data, file)
with open('saved_steps.pkl', 'rb') as file:
   data = pickle.load(file)

regressor_loaded = data["model"]
le_country = data["le_country"]
le_education = data["le_education"]
y_pred = regressor_loaded.predict(X)
y_pred
```

app.py

```
import streamlit as st

import pickle

import xgboost

import numpy as np



# import the model

pipe = pickle.load(open('pipe.pkl', 'rb'))
```

```
df = pickle.load(open('df.pkl', 'rb'))


st.title("Laptop Price Predictor")


# brand

company = st.selectbox('Brand',df['Company'].unique())

# print(type(company))


# type of laptop

type = st.selectbox('Type',df['TypeName'].unique())


# Ram

ram = st.selectbox('RAM(in GB)',[2,4,6,8,12,16,24,32,64])


# weight

weight = st.number_input('Weight of the Laptop')


# Touchscreen

touchscreen = st.selectbox('Touchscreen',['No','Yes'])


# IPS

ips = st.selectbox('IPS',['No','Yes'])
```

```
# screen size

screen_size = st.number_input('Screen Size')


# resolution

resolution = st.selectbox('Screen
Resolution',['1920x1080','1366x768','1600x900','3840x2160'
,'3200x1800','2880x1800','2560x1600','2560x1440','2304x144
0'])


#cpu

cpu = st.selectbox('CPU',df['Cpu brand'].unique())


hdd = st.selectbox('HDD(in GB)',[0,128,256,512,1024,2048])


ssd = st.selectbox('SSD(in GB)',[0,8,128,256,512,1024])


gpu = st.selectbox('GPU',df['Gpu brand'].unique())


os = st.selectbox('OS',df['os'].unique())


if st.button('Predict Price'):

# query

ppi = None

if touchscreen == 'Yes':
```

```
touchscreen = 1

else:

touchscreen = 0



if ips == 'Yes':

ips = 1

else:

ips = 0



X_res = int(resolution.split('x')[0])

Y_res = int(resolution.split('x')[1])

ppi = ((X_res**2) + (Y_res**2))**0.5/screen_size

query =
np.array([company,type,ram,weight,touchscreen,ips,ppi,cpu,
hdd,ssd,gpu,os],dtype=object)



query = query.reshape(1,12)

st.title("The predicted price of this configuration is " +
str(int(np.exp(pipe.predict(query)[0]))))
```

misc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
<component name="Black">
<option name="sdkName" value="Python 3.10 (LaptopPricePrediction)" />
</component>
<component name="ProjectRootManager" version="2" project-jdk-name="Python 3.10
(LaptopPricePrediction)" project-jdk-type="Python SDK" />
</project>
```

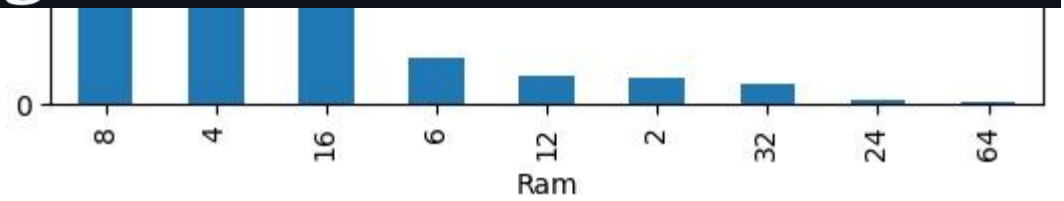
modules.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
<component name="ProjectModuleManager">
<modules>
<module fileurl="file://$PROJECT_DIR$/.idea/LaptopPricePrediction.iml"
filepath="$PROJECT_DIR$/.idea/LaptopPricePrediction.iml" />
</modules>
</component>
</project>
```

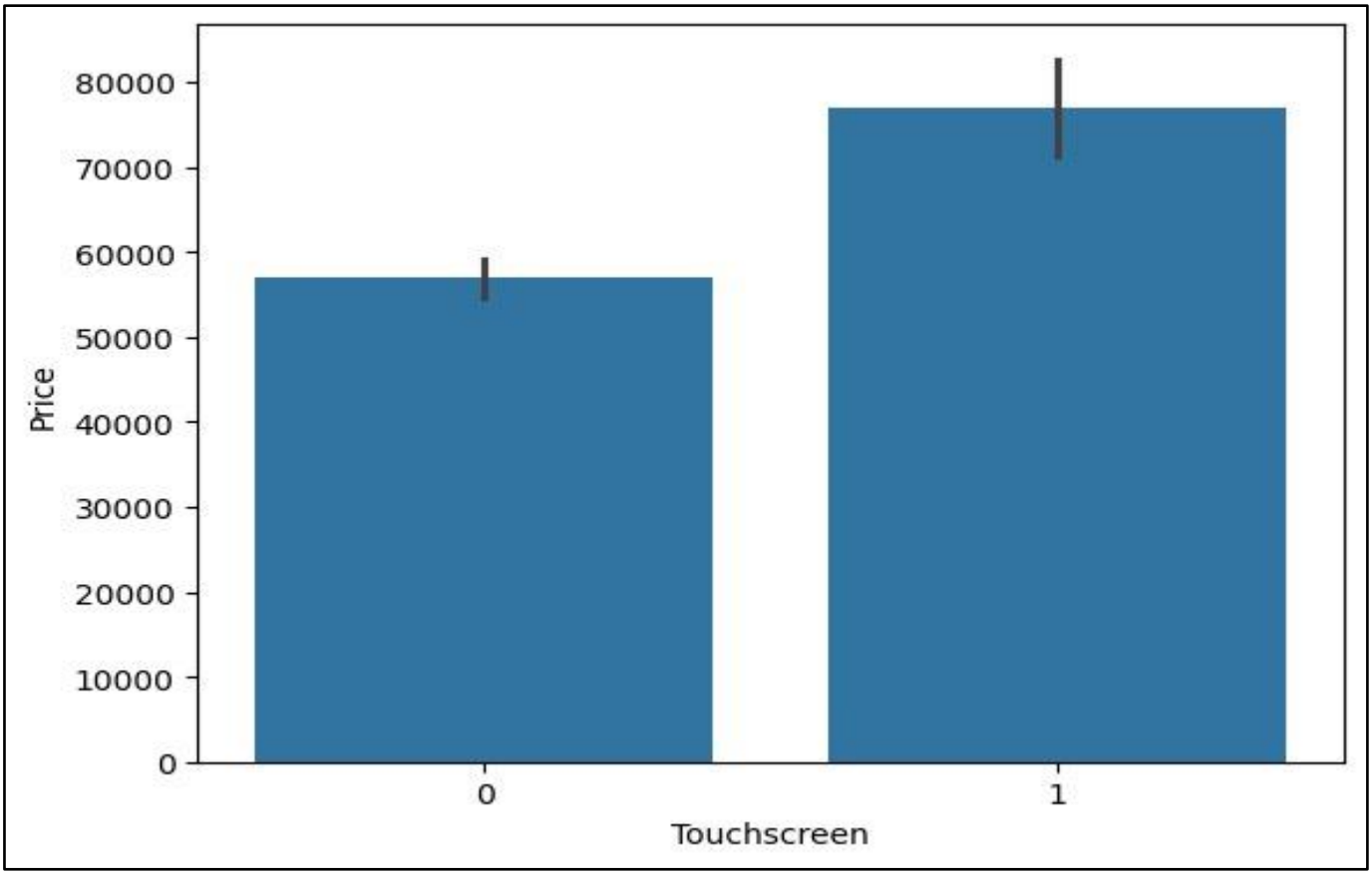
LaptopPricePrediction:

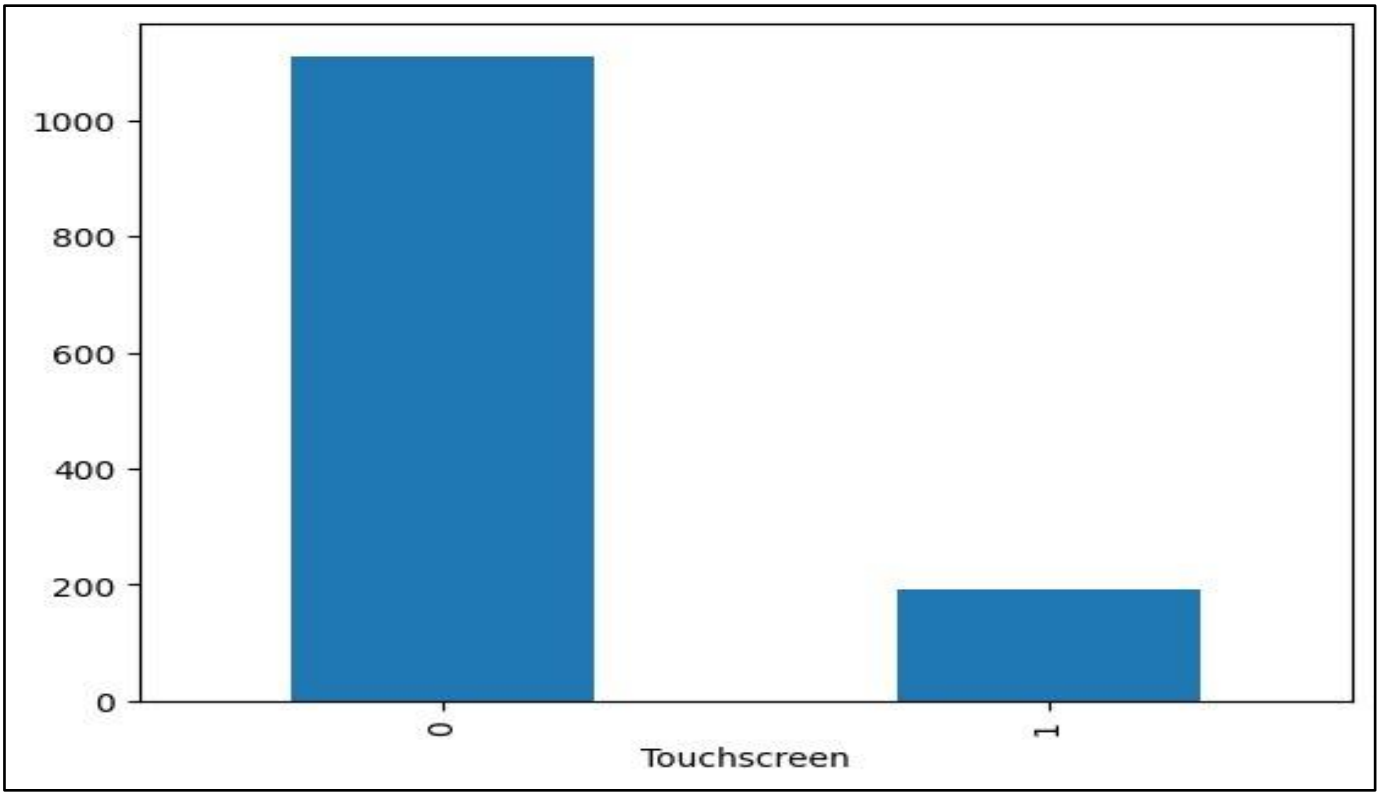
```
<?xml version="1.0" encoding="UTF-8"?>
<module type="PYTHON_MODULE" version="4">
<component name="NewModuleRootManager">
<content url="file://$MODULE_DIR$">
<excludeFolder url="file://$MODULE_DIR$/venv" />
</content>
<orderEntry type="jdk" jdkName="Python 3.10 (LaptopPricePrediction)" jdkType="Python SDK" />
<orderEntry type="sourceFolder" forTests="false" />
</component>
</module>
```

b) Output Screenshots:



Command for run : - streamlit run app.py
http://localhost:850





Chapter 5: Conclusion/References

Conclusion

I this report, we have presented the culmination of our efforts in developing a Laptop Price Prediction Machine Learning Project. Our primary objective was to create a robust and accurate model capable of forecasting laptop prices, with the potential to revolutionize the way consumers make purchasing decisions and assist sellers in setting competitive prices.

Over the course of this project, we have addressed various facets of the problem and achieved notable milestones:

References

1. Brownlee, J. (2021). Machine Learning Mastery: A Gentle Introduction to Predictive Modeling. [Online Book]. Available at: https://machinelearningmastery.com/start-here/#algorithms

2. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). DOI: 10.1145/2939672.2939785

3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.

4. McKinney, W. (2017). Python for Data Analysis. O'Reilly Media.

5. Python Software Foundation. (2021). Python Language Reference, Version 3.10. [Online Documentation]. Available at: https://docs.python.org/3.10/

6. The Laptop Database. (n.d.). [Online Database]. Available at: https://laptopmedia.com/laptop-specs/

A PRELIMINARY MINI PROJECT REPORT ON

"Laptop Price Prediction"

SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (T.Y. B. Tech.)

Academic Year: 2022-23

By

Name of the students in the group with Roll No.

Suraj Dalvi - TYCOA44

Ronak Dagale - TYCOA43

Prathmesh Chougale - TYCOA40

Under The Guidance of

Prof. Mahalakshmi Bodireddy

DEPARTMENT OF COMPUTER ENGINEERING,

PIMPRI CHINCHWAD COLLEGE OF

ENGINEERING SECTOR 26, NIGDI, PRADHIKARAN

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that, the project entitled **"Laptop Price Prediction**" is successfully carried out as a mini project successfully submitted by the following students of "PCET's Pimpri Chinchwad College of Engineering, Nigdi, Pune-44**".**

Under the guidance of Prof. Mahalakshmi Bodireddy

In the partial fulfillment of the requirements for the T.Y. B. Tech. (Computer Engineering)

Name of the students in the group with Roll No.

Suraj Dalvi - TYCOA44

Ronak Dagale - TYCOA43

Prathmesh Chougale - TYCOA40