JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

Computing for Data Science

Project Report

Topic - Laptop Price Prediction

GROUP MEMBER'S DETAIL

NAME	ENROLLMENT	BATCH
NAIVEDYA KHARE	19803002	B13
PRIYANSHU JAISWAL	19803016	B13

Department - Computer Science & Engineering
Project Supervisor - Dr. Megha Rathi

Introduction

A laptop computer, sometimes called a notebook computer by manufacturers, is a battery- or AC-powered personal computer generally smaller than a briefcase that can easily be transported and conveniently used in temporary spaces such as on airplanes, in libraries, temporary offices, and at meetings. A laptop computer is now all but essential for any business student. From registering for classes, to communicating with professors and classmates, to accessing notes and taking exams, a laptop provides necessary resources to be a successful Business Rocket.

It is predicted that the factors that have the largest effect on the price of the laptops are the RAM size, battery life, processor, GPU, Graphics Card, Flash memory or SSD/HDD, etc. We have taken some of these attributes into account to train a model that would predict the price of the laptop given the attributes, for example RAM and Processor.

Work Around

After doing some research, we found out that a random forest regressor is one of the best ways to train a model to predict laptop prices.

What is randomForestRegressor?

Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision

trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

Data set Attributes that are being taken into consideration:

- Company
- RAM
- Weight
- Laptop Type
- Storage
- Resolution
- Screen Size
- IPS
- Touchscreen

Steps Involved

- Collecting data for the dataset to train model
- Preprocess the data to remove noise and unnecessary attributes

- Find pattern and make model for prediction
- A web UI to enter details of the laptop to predict the price

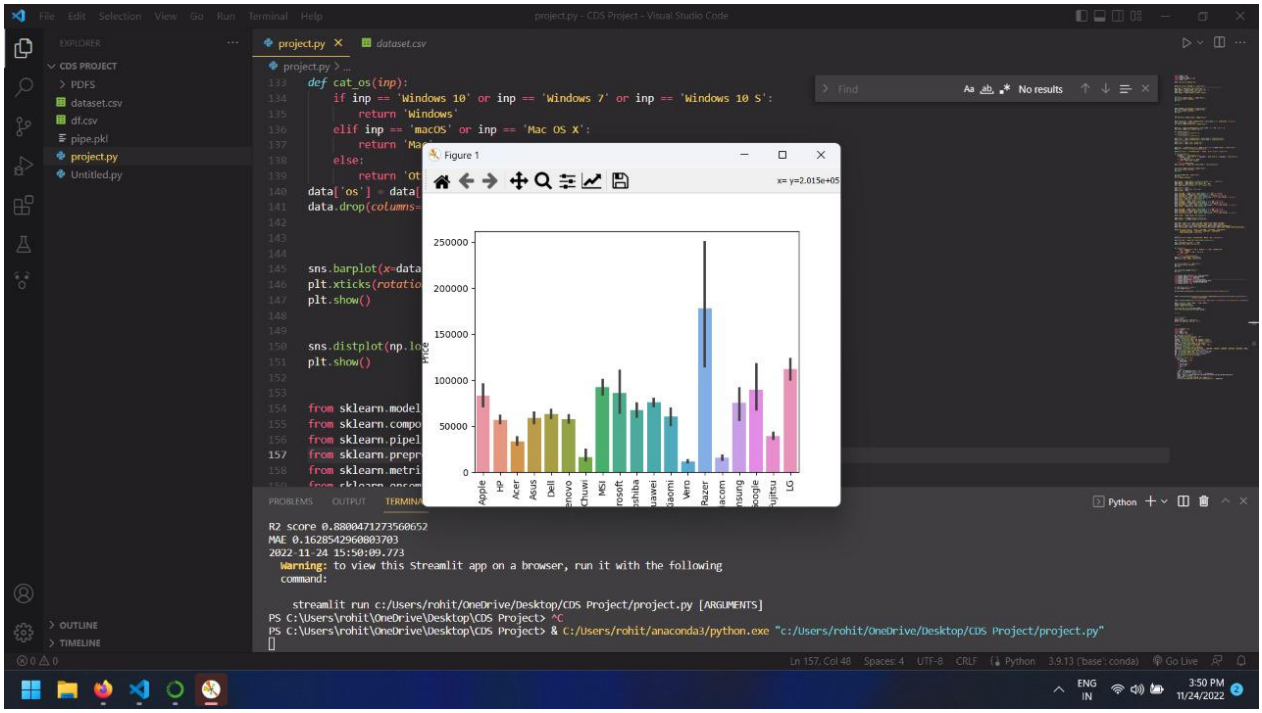
How does our code work?

- Firstly, we import the data that contains the information about laptops and their prices into a pandas dataframe. We have approximately **1302** data in our dataset.
- Showcasing various graphs between Price and other attributes (shown below) like
    - Price vs CPU
    - Price vs RAM
    - Price vs Operating System
- Then we apply various pre-processing on the dataset which includes -
    - Removing “GB” from RAM info
    - Processing CPU information to retrieve version of intel processor
    - Converting Yes/No data into binary form of attributes like Touchscreen and IPS.
    - Retrieving X and Y screen resolution from X*Y format from the dataset
- Then use various libraries to convert data into required form like converting categorical data into numerical data using oneHotEncoder.
- Then we trained our Machine Learning model using **RandomForestRegressor.**
- After training the model, we create a pickle file that stores the trained model for later use without needing to train the model

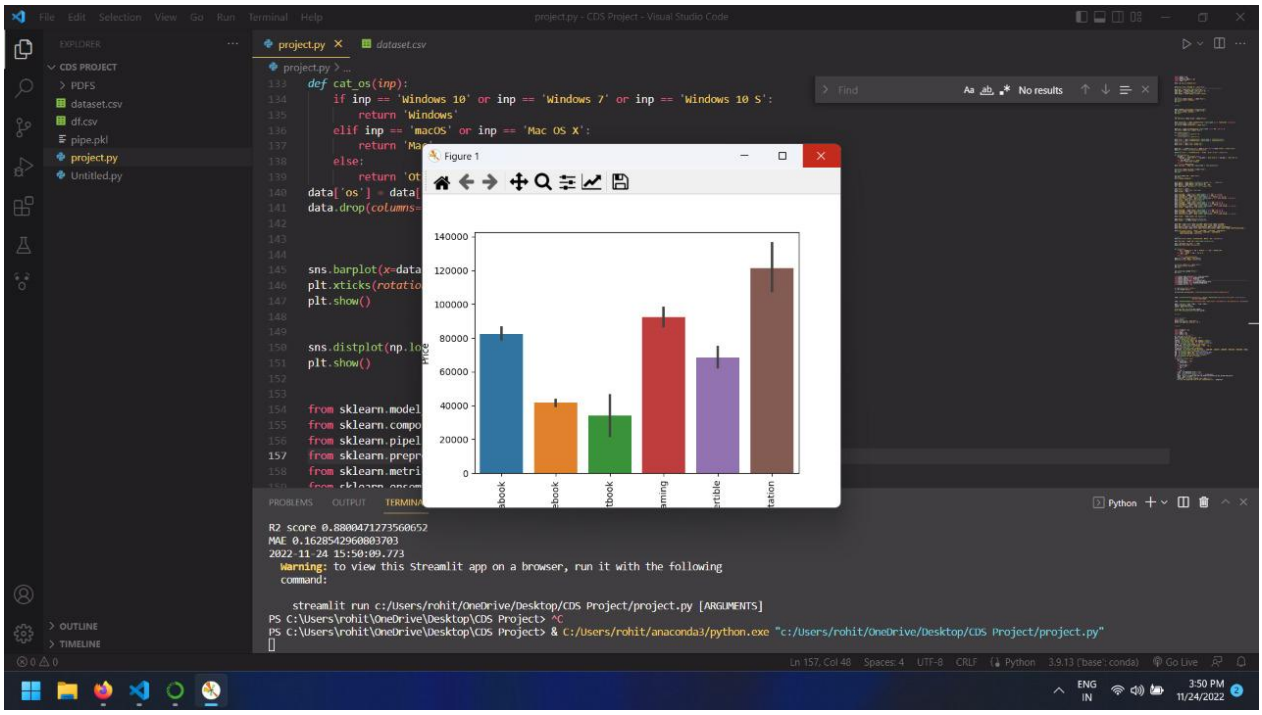
again.

- Then we create a UI to create an interface to enter details of laptops to predict its price using StreamLit.

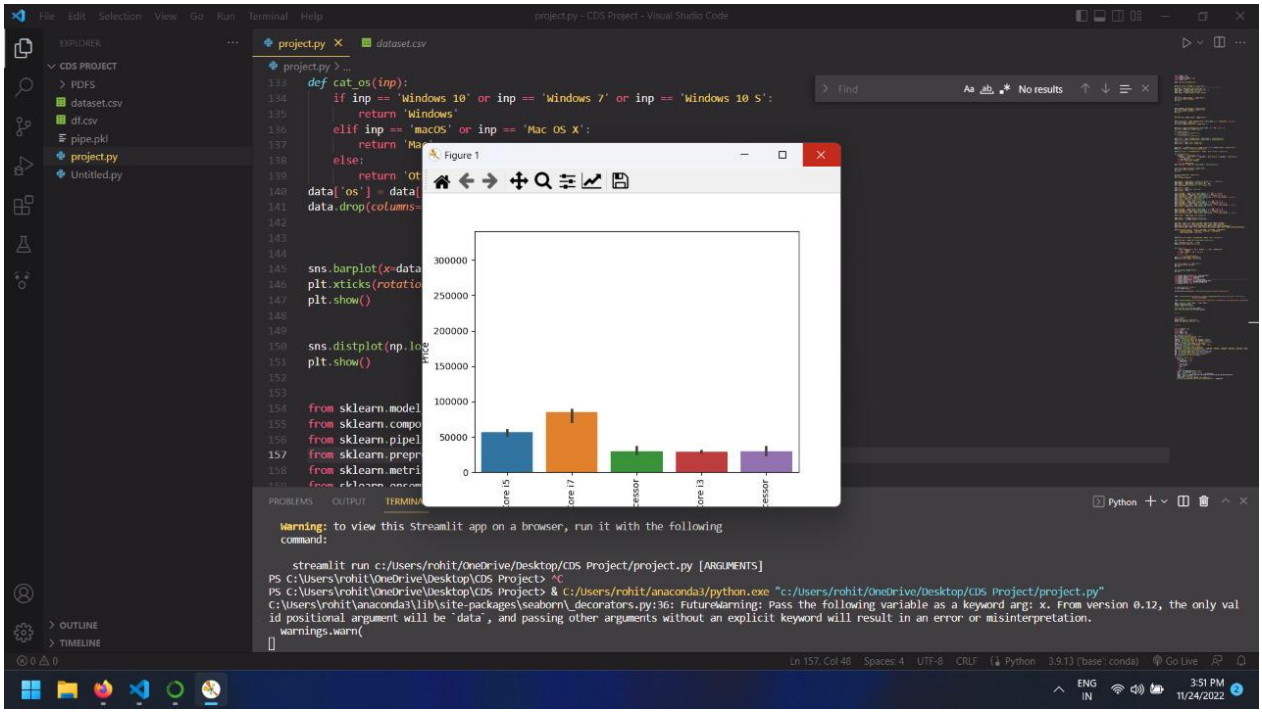
Price vs Company



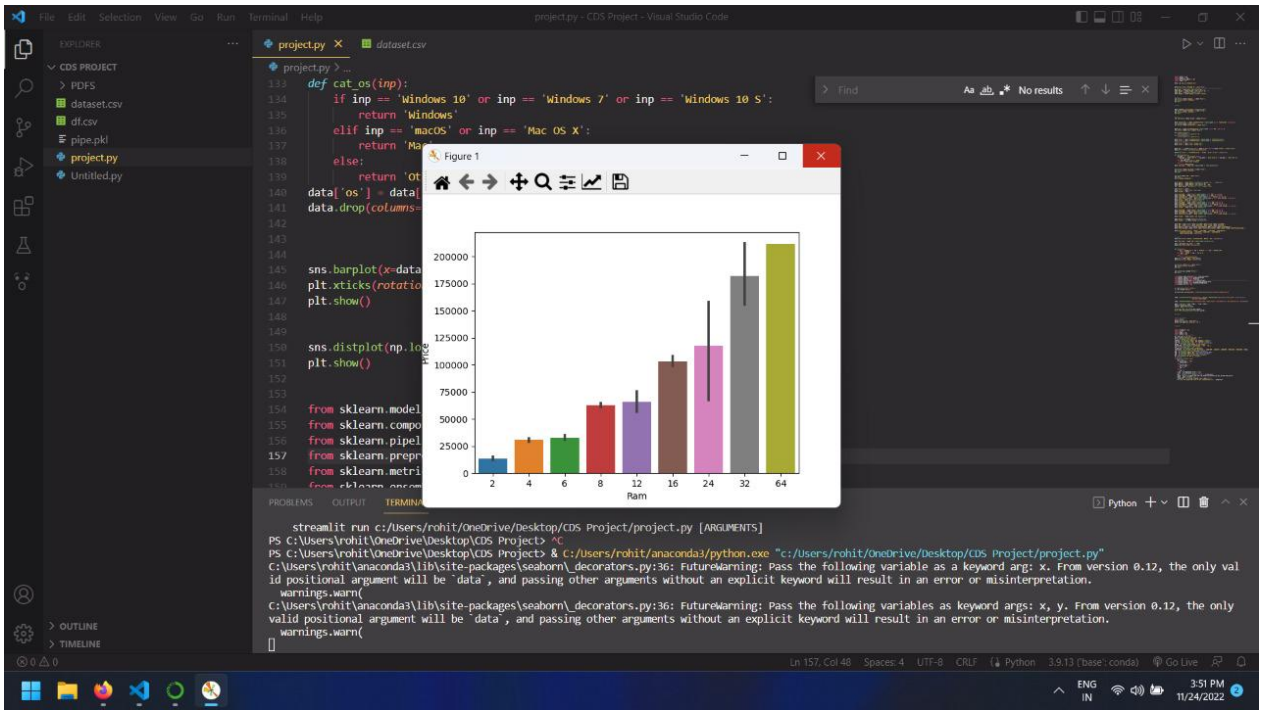
Price vs TypeName


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':
else:
data['os'] = data
data.drop(columns=
sns.barplot(x=data
plt.xticks(rotatio
plt.show()
sns.distplot(np.lo
plt.show()
from sklearn.model
from sklearn.compo
from sklearn.pipel
from sklearn.prepr
from sklearn.metri
Figure 1
Price
R2 score 0.8800471273560652
MAE 0.16285429600003703
2022-11-24 15:50:09.773
Warning: to view this Streamlit app on a browser, run it with the following
command:
streamlit run c:/Users/rohit/OneDrive/Desktop/CDS Project/project.py [ARGUMENTS]
PS C:\Users\rohit\OneDrive\Desktop\CDS Project> ^C
PS C:\Users\rohit\OneDrive\Desktop\CDS Project> & C:/Users/rohit/anaconda3/python.exe "c:/Users/rohit/OneDrive/Desktop/CDS Project/project.py"

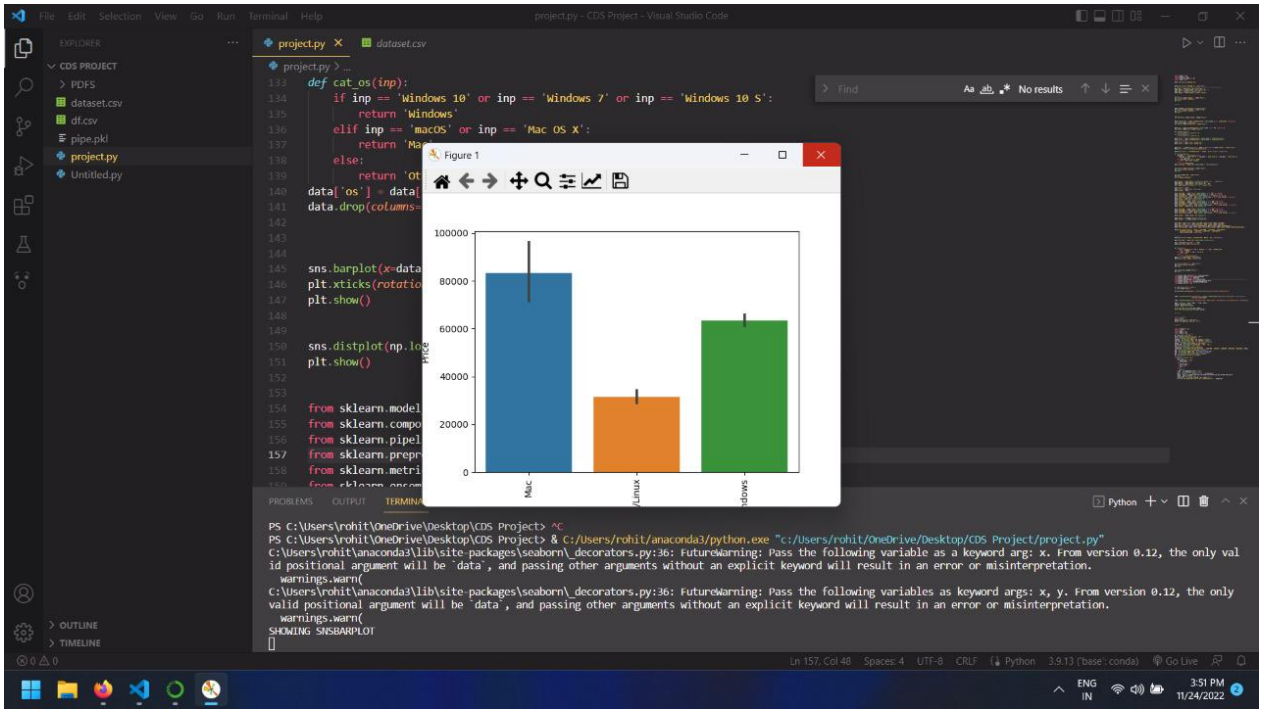
Price vs Cpu


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':
else:
data['os'] = data
data.drop(columns=
sns.barplot(x=data
plt.xticks(rotatio
plt.show()
sns.distplot(np.lo
plt.show()
from sklearn.model
from sklearn.compo
from sklearn.pipel
from sklearn.prepr
from sklearn.metri
Figure 1
Price
Warning: to view this Streamlit app on a browser, run it with the following
command:
streamlit run c:/Users/rohit/OneDrive/Desktop/CDS Project/project.py [ARGUMENTS]
PS C:\Users\rohit\OneDrive\Desktop\CDS Project> ^C
PS C:\Users\rohit\OneDrive\Desktop\CDS Project> & C:/Users/rohit/anaconda3/python.exe "c:/Users/rohit/OneDrive/Desktop/CDS Project/project.py"
C:\Users\rohit\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

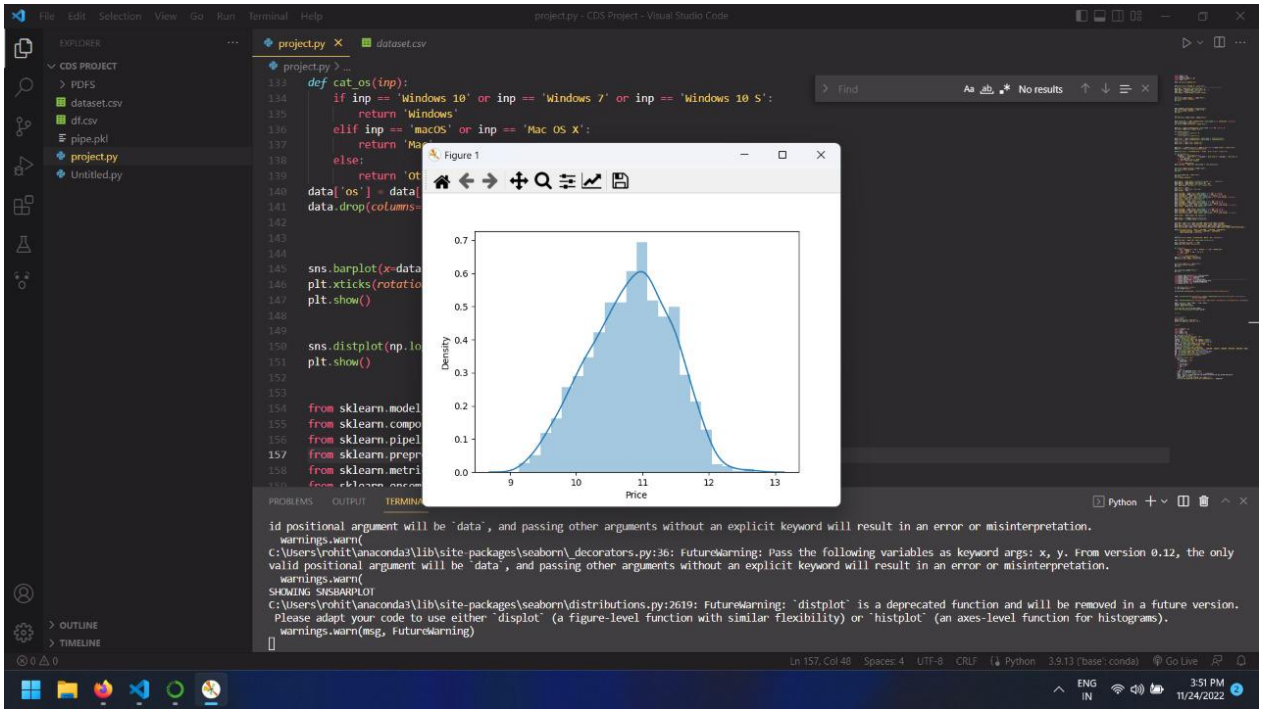
Price vs Ram



Price vs OpSys

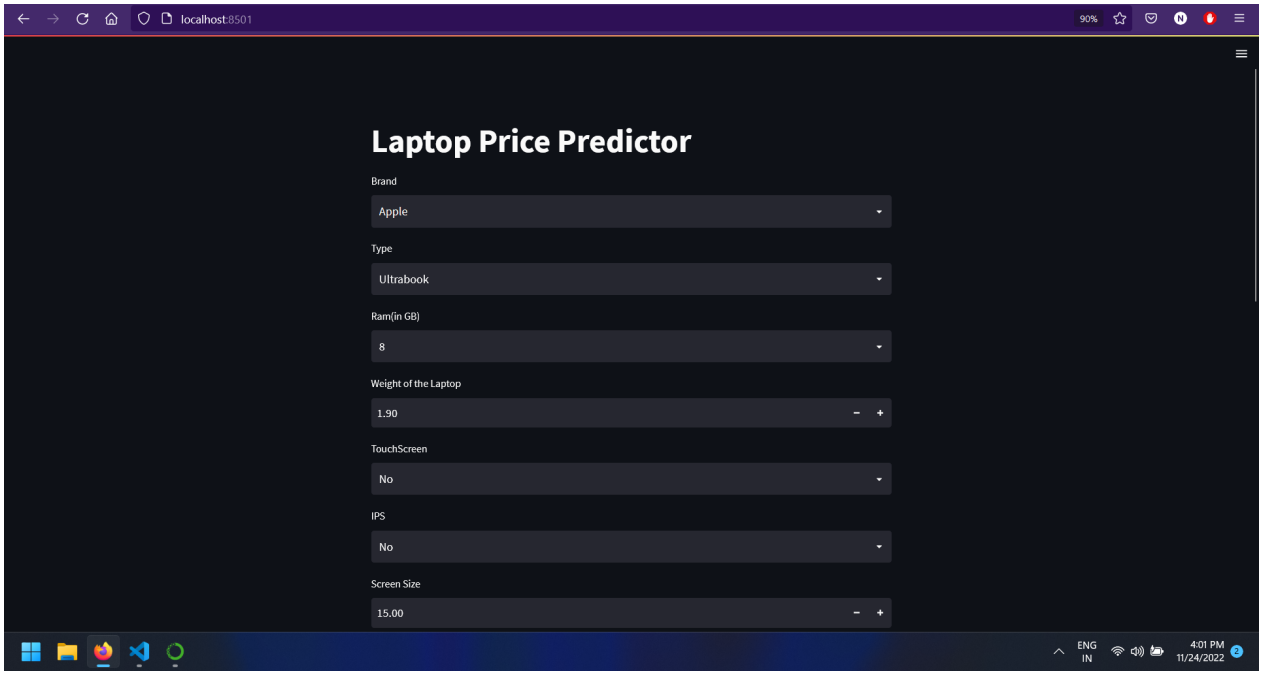


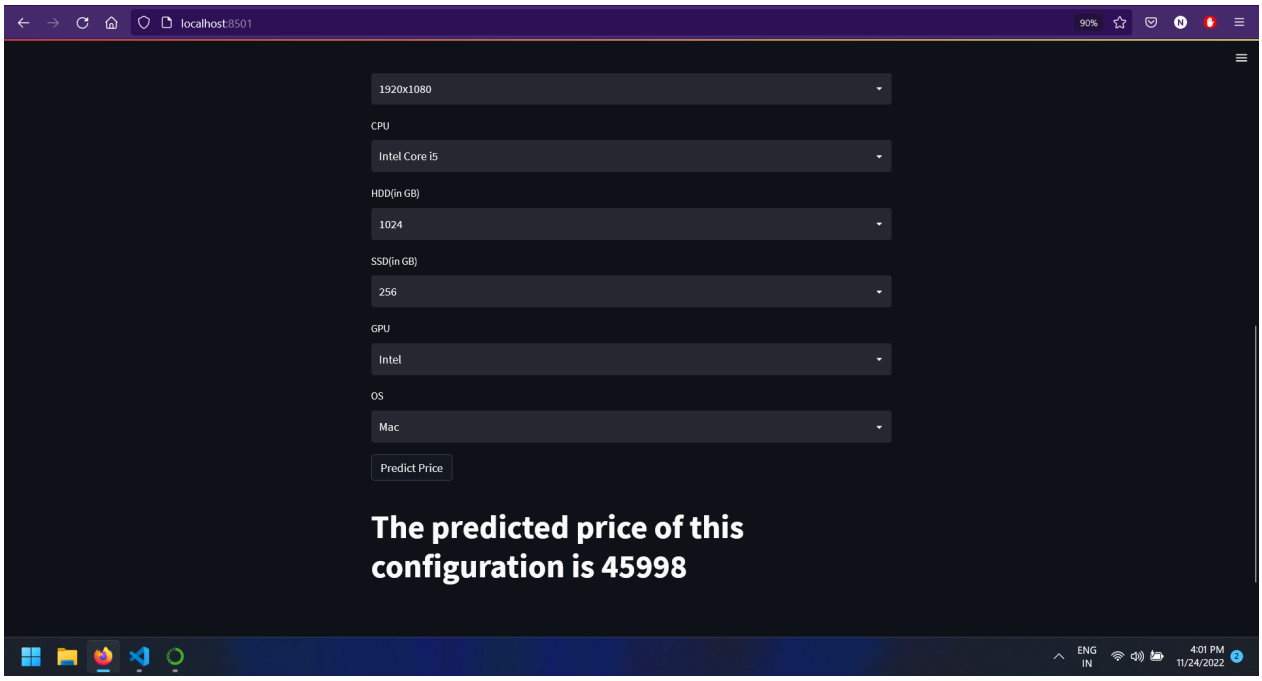
Graph of Logarithm of Price



UI of StreamLit

Laptop Price Predictor





Conclusion

Machine Learning is a technique of training machines to perform the activities a human brain can do, albeit a bit faster and better than an average human-being. Today we have seen that machines can beat human champions in games such as Chess or predict laptops prices. You have seen that machines can be trained to perform human activities in several areas and can aid humans in living better lives.

In this project we've demonstrated some of the fundamentals behind random forest models and more specifically how to apply sklearn's random forest regressor algorithm. We pointed out some of the benefits of random forest models.