INTERNSHIP AT ENLIGHTEN INFOSYSTEMS

A PROJECT REPORT

Submitted

by

Maulik S Raval

200050131090

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

Computer Science and Engineering

Babaria Institute of Technology, Vadodara

Gujarat Technological University, Ahmedabad

Academic year: 2023-24

Babaria Institute of Technology

Vadodara-Mumbai NH 8, Varnama, Vadodara, Gujarat

391243

CERTIFICATE

This is to certify that the Internship report submitted along with the project entitled **Adaptive and Predictive Modelling** has been carried out by **Maulik Sanjaykumar Raval** under my guidance impartial fulfillment for the degree of Bachelor of Engineering in Computer and Science Engineering, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2023-24.

Ms. Jalpa P Bhatt	Dr. Rock Upadhyay
Internal Guide	Head of the Department

INDUSTRY COMPLETION CERTIFICATE

Enlighten Infosystems
Create | Innovate | Share

2B, Neptune Tower, Near
BPC House, BPC Road, Alkapuri,
Vadodara - 390 007.
+91 81402 31918
www.enlighteninfosystems.com
contact@enlighteninfosystems.com

ISO 9001:2015

18th April, 2024

Certificate

To Whomsoever It May Concern

This is to certify that **Mr. Maulik S. Raval** (Enrollment No.: 200050131090), student of **B.E (CSE), BIT, Vadodara** affiliated to Gujarat Technological University, Ahmedabad (Gujarat) has carried out **Internship** in the domain of **Data Science** at our organization from 16th January, 2024 to 18th April, 2024.

During this period he was found to be hardworking, honest, motivated and a committed individual. He took keen interest in the development assignments given to him, which met our expectations.

We wish him great success in all his future endeavors.

For, Enlighten Infosystems

A.Z. Surti

Dr. Akil Z. Surti
Project Mentor

ENLIGHTEN INFOSYSTEMS VADODARA

Product Development | Software Development | Mobile App. Development
Website Development | Training & Consultancy | Domain & Hosting

UNIVERSITY COMPLETION CERTIFICATE

GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2022-2023

Date of certificate generation : 03 April 2024 (08:58:05)

This is to certify that, ***Raval Maulik Sanjaykumar*** (Enrolment Number - 200050131090) working on project entitled with ***Adeptive and Predictive Modeling*** from ***Computer Science & Engineering*** department of ***BABARIA INSTITUTE OF TECHNOLOGY, VARNAMA*** had submitted following details at online project portal.

Internship Project Report	**Completed**

Name of Student : Raval Maulik Sanjaykumar

Name of Guide : Mrs. Jalpa Poojan Bhatt

Signature of Student : _____

*Signature of Guide : _____

Disclaimer :
This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received a copy of the data that you have uploaded and submitted as your project work.

*Guide has to sign the certificate, Only if all above activities has been Completed.

Babaria Institute of Technology
Mumbai NH 8, Varnama, Vadodara-391243,Gujarat

DECLARATION

I hereby declare that the Internship / Project report submitted along with the Internship entitled **Data Science** submitted in partial fulfillment for the degree of Bachelor of Engineering in Computer and Science Engineering to Gujarat Technological University, Ahmedabad is a bonafide record of original project work carried out by meat Enlighten Infosystems under the supervision of Dr. Akil Z Surti and that no part of this report has been copied from any student's reports or taken from any other source, without providing due reference.

Name of the Student — Sign of Student

Maulik Sanjaykumar Raval

ACKNOWLEDGEMENT

Any achievement doesn't depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. Several personalities have helped me. I would like to take this opportunity to thank all of them.

I would like to extend my deepest gratitude to Enlighten Infosystems and Dr. Akil Surti, who gave me the golden opportunity to do this wonderful internship.

I extend warm thanks to Dr. Rocky Upadhyay, H.O.D., Dept. of Computer and Science Engineering BITS, Vadodara for his constant encouragement, motivation, and guidance.

Further, I would like to thank my internal guide Ms. Jalpa P Bhatt for giving me constant guidance during the internship and helping me a lot in finalizing this internship within the limited time frame.

Thanking you

Maulik S Raval

ABSTRACT

This report encapsulates the experiences and learnings from a 12-week internship at Enlighten Infosystems in Vadodara, Gujarat, India, as part of the requirements for the completion of the Bachelor of Engineering degree in Computer Science and Engineering from Gujarat Technological University, Ahmedabad.

The internship focused on the theme of "Adaptive and Predictive Modelling," delving into the realms of data science and machine learning. It involved understanding the core concepts of these fields and their practical applications in real-world scenarios.

The report begins with an introduction to Enlighten Infosystems, the host organization for the internship, providing insights into its operations, services, and clientele. It further explores the significance of adaptive and predictive modeling, elucidating their roles in data analysis, forecasting, and pattern recognition.

The learning journey during the internship is divided into two phases. In the first phase, the emphasis was on grasping the fundamentals of data science and machine learning, including learning Python programming, exploring various libraries such as Numpy, Pandas, Matplotlib, and Seaborn, and gaining proficiency in data manipulation and visualization.

The second phase delved deeper into advanced concepts, including database exploration with MySQL and the exploration of machine learning algorithms like Histogram of Oriented Gradients (HOG), Region-based Convolutional Neural Networks (R-CNN), Fast R-CNN, Single Shot-Detector (SSD), and You Only Look Once (YOLO), Random Forest, linear Regression, LightGBM, etc. Each algorithm was examined in detail, along with its applications, advantages, and limitations.

Throughout the internship, practical projects and hands-on exercises were undertaken to apply the acquired knowledge and skills. The experience not only enhanced technical competencies but also fostered soft skills such as communication, teamwork, and problem-solving.

In conclusion, the internship at Enlighten Infosystems provided invaluable insights into the dynamic field of data science and machine learning. It equipped the intern with the requisite knowledge, skills, and practical experience, laying a strong foundation for future endeavors in the realm of technology and analytics.

-Maulik S Raval

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW OF THE COMPANY

Company Name: Enlighten Infosystems

Address: 2B, Neptune Tower, Near BPC House, BPC Road, Alkapuri, Vadodara - 390007

Contact No: +91 8140231918

Email ID: contact@enlighteninfosystems.com

Enlighten Infosystems, based in Vadodara, is a dynamic IT software development, training, professional service, and consulting firm that works with corporations, firms, and businesses. Enlighten Infosystems, a privately held company, provides software design, development, services, and support to the following areas and verticals: Education, Retail, Finance, Manufacturing and Engineering, Health Care Technology, and Travel and Hospitality. The company provides several IT software development services while also meeting the needs of its clientele. Its strength is delivering services on time and to the client's expectations. Enlighten Infosystems adheres to a few principles: Focus on quality and use cutting-edge technologies and technology development updates for our valued clients. The staff has demonstrated their abilities on medium- to large-scale projects with over 500 users. They also have specialized knowledge in the development of novel products. Enlighten Infosystems has worked on more than ten projects in the previous

years and has a customer base of more than 30 clients, providing satisfactory services. Aside from Vadodara, they have clientele from Gujarat, Maharashtra, Rajasthan, Delhi NCR, and Madhya Pradesh.

The internship aims to provide direction to the activities, help to focus on a result, and to assess the result achieved.

The two factors influencing my development were taken into account when formulating the goal before beginning the internship:

1. Understand what I have learned (theoretical and practical knowledge on my subject field) with actual work experience to complement my field-specific skills and gain new skills.
2. Apply and analyze at least one future skill.

1.2 WHAT IS ADEPTIVE AND PREDICTIVE MODELLING?

Adaptive modeling is a technique used in data analysis, statistics, and machine learning where the model structure or parameters are adjusted dynamically based on incoming data or changing conditions. The primary goal of adaptive modeling is to create models that can continuously learn and improve over time without the need for manual intervention.

Predictive modeling is a statistical technique used to forecast future outcomes or trends based on historical data and patterns. It involves building a mathematical model that captures the relationships between input variables (also known as predictors or features) and the target variable (the variable to be predicted). The primary goal of predictive modeling is to make accurate predictions about future events or behaviors.

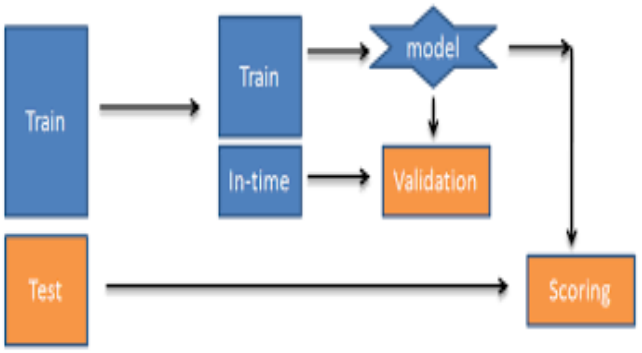


Figure 1.2 Adaptive and Predictive Modelling

1.3 DATA SCIENCE AND MACHINE LEARNING

1.3.1 What is Data Science?

It is the complicated analysis of enormous amounts of data in a company's or organization's repository. This study includes where the data came from, an analysis of its substance, and how this data may help the organization expand in the future. Organizational data is always in one of two forms: structured or unstructured. When we analyse this data, we gain significant insights into business or market patterns, which gives the company an advantage over its competitors because they have boosted their efficacy by spotting patterns in the data collection.

1.3.2 What is Machine Learning?

Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and techniques that enable computers to learn from data and improve their performance on a specific task without being explicitly programmed. In other words, machine learning algorithms learn patterns and relationships from data, allowing them to make predictions, decisions, and inferences based on new or unseen data.

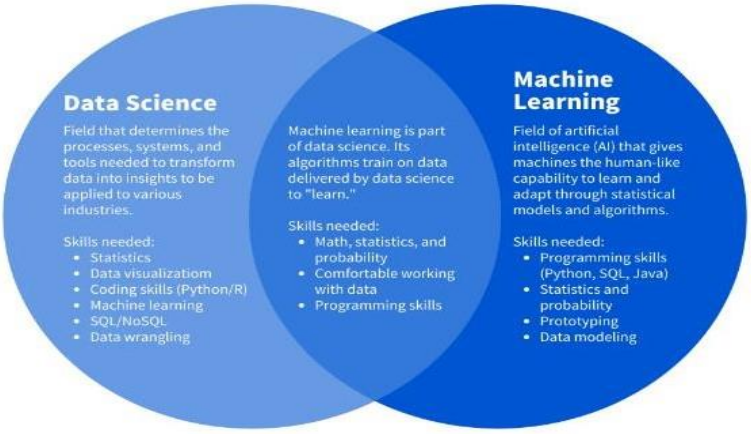


Figure 1.3.2 Data Science and Machine

CHAPTER 2: LEARNING PHASE-I

2.1 LEARNING FUNDAMENTALS

At the start of my internship, I got to know the company better. An early stage of my life saw me go through the process of basic information transfer. I got my ideas about the Python language clear. From setting up the environment to learning the basics of any programming language, like data types, arrays, different kinds of looping, and so on. I learned a lot about data science and machine learning, such as the different kinds of data, datasets, methods, and different libraries.

2.2 GETTING STARTED WITH DATA SCIENCE AND MACHINE LEARNING

I learned more about what Data Science and Machine Learning are. In addition, I learned about the ideas behind machine learning (ML) and its two types, supervised learning and unsupervised learning. I then used PyCharm to put the different methods into action. I also learned a few more things, such as how to clean data, do exploratory data analysis, use graphics, prepare data for analysis, and do small amounts of statistics. As I worked on the newer jobs, I also learned the basics of NumPy, Pandas, Matplotlib, Seaborn, Data frames, and other tools. Finally, I worked on datasets from the real world to figure out how to use ML techniques on them.

2.2.1 NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

```python
import numpy as np

a = np.array([10, 20, 30])
b = np.array([40, 50, 60])
print(a)
print(b)

# Different mathemetical Operations  on  two different arrays
print(np.add(a, b))
print(np.subtract(a, b))
print(np.multiply(a, b))
print(np.divide(a, b))
print(np.mod(a, b))
print(np.power(a, b))
print(np.sqrt(b))

# it is an N-dimentional Array
arr = np.array([[10, 20, 30], [20,30,40],[30,40,50]])
print(type(arr))
# output - <class 'numpy.ndarray'> here, nd stands for N dimentional array

#Slicing in array
arr1 = np.array([10, 20, 30, 40])
print(arr1[0:3])
print(arr1[0:])
print(arr1[:4])
print(arr1[3:])
```

Figure 2.2.1.1

```python
#slicing for 2D array
arr2 = np.array([[10, 20, 30], [20,30,40]])
print(arr2[0:2,0:2])
print(arr2[0:2,0:3])
print(arr2[0,1:4])
print(arr2[1,1:3])
print(np.shape(arr2)) #Shape function will show the number of rows and columns(here output will 2 rows and 3 columns)
print(np.size(arr2)) # Size function will show you the total size of the array including all rows and columns
print(np.ndim(arr2)) # ndim function will show you the number of dimentions of specific array
print(arr2.dtype) #Dtype will give you the data type which is get used in specified array
print(arr2.astype(float)) #astype function wil used to change the datatype of specified array
print(len(arr2)) # number of neasted values

# Combining and Splitting arrays
    #methods to concanate array
arr3 = np.array([10, 20, 30])
arr4 = np.array([40, 50, 60])
print(np.concatenate((arr3, arr4)))
    #method to concanate according to axis
arr5 = np.array([[1,2,3],[4,5,6]])
arr6 = np.array([[7,8,9],[11,12,13]])
print(np.concatenate( arrays: (arr5, arr6),axis=1)) #vertically
print(np.concatenate( arrays: (arr5, arr6),axis=0)) #horizontally
print(np.hstack((arr5, arr6))) # hstack = horizontally concatanation
print(np.vstack((arr5, arr6))) # vstack = vertically concatanation
```

Figure 2.2.1.2

```
    #methods to split array
arr7 = np.array([1,2,3,4,5,6,7,8,9])
print(np.array_split(arr7, indices_or_sections: 3))
c = np.array_split(arr7, indices_or_sections: 3)
print(c[0])
print(c[1])

#Adding and Removing elements from array
arr8 = np.array([1,2,3,4,5,6])
arr9 = np.array([10,11,12,13,14,15,16])
print(np.append(arr8, values: 10))
print(np.insert(arr8, obj: 1, values: 40))
print(np.insert(arr5, obj: 1, values: [100,200],axis=1))
print(np.insert(arr5, obj: 1, values: [100,200,300],axis=0))
print(np.delete(arr9, obj: 1))
print(np.delete(arr5, obj: 1,axis=1))

#sort,filter,search
    #Sorting
arr10 = np.array([1,22,5,3,9,7,11,0])
print(np.sort(arr10))
print(np.sort(arr6))
    #search
arr11 = np.array([1,222,45,2,6,8])
s = np.where(arr11 == 222)
print(s)
```

Figure 2.2.1.3

2.2.2 Pandas

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's Active Python.

```python
import numpy.random
import pandas as pd
dict1 = {'Name': ['John', 'Carol', 'David', 'Floki']
         , 'Age': [20, 30, 40, 50]
         , 'Gender': ['M', 'F', 'M', 'M']
         ,  'Phone': ['023456789', '7895641230','5731594560', '987654321']
         }
df = pd.DataFrame(dict1)
print(df)

#to create a CSV file
df.to_csv('pandas_example.csv')
df.to_csv( path_or_buf: 'pandas_NoIndex_example.csv',index = False)
print("")
#to see the initial row
print(df.head(2))

print("")
#to see last few rows
print(df.tail(2))

print("")
# to Do all statistical operations
print(df.describe())
```

Figure 2.2.2.1

```python
#dataframe
new_dataframe = pd.DataFrame(numpy.random.rand(334,5), index=numpy.arange(334))
print(new_dataframe)
print(type(new_dataframe))
print(new_dataframe.describe())
# new_dataframe[0][0] = "Maulik"
# print(new_dataframe.head())
print(new_dataframe.to_numpy())

#Transpose= row into column and column into row
print(new_dataframe.T)

# sorting index on row
print(new_dataframe.sort_index(axis= 0, ascending=False))

#sorting index on columns
print(new_dataframe.sort_index(axis= 1, ascending=False))

#view
new_dataframe2 = new_dataframe
new_dataframe2[0][1] = "Hello World"
print(new_dataframe)

#LOC function to edit dataframe
new_dataframe.loc[0,0] = 7980
print(new_dataframe)

new_dataframe.columns = list("ABCDE")
print(new_dataframe)
new_dataframe.loc[0,'B'] = 779
print(new_dataframe)
```

Figure 2.2.2.2

2.2.3 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, Python shells, web application servers, and various graphical user interface toolkits.

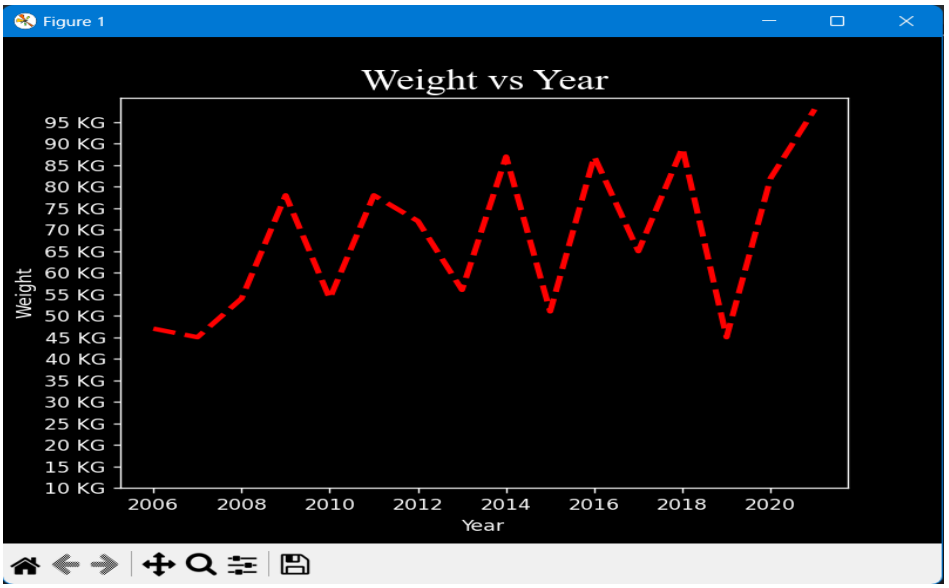


Figure 2.2.3.1

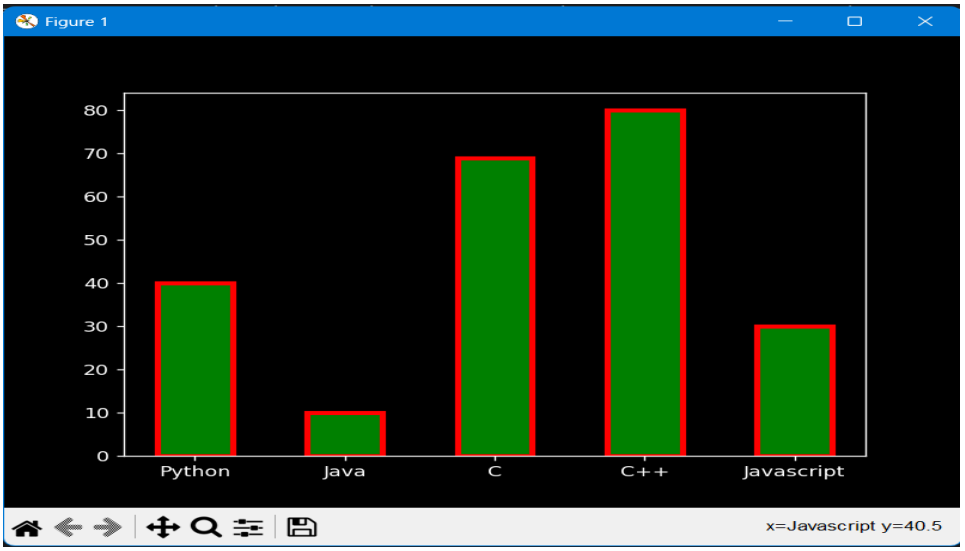


Figure 2.2.3.2

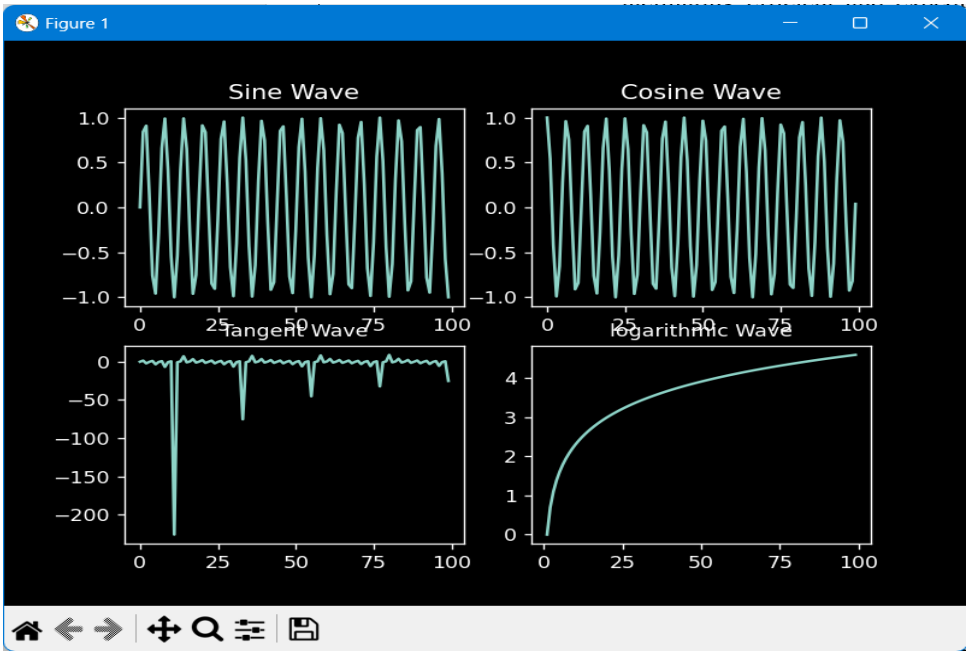


Figure 2.2.3.3

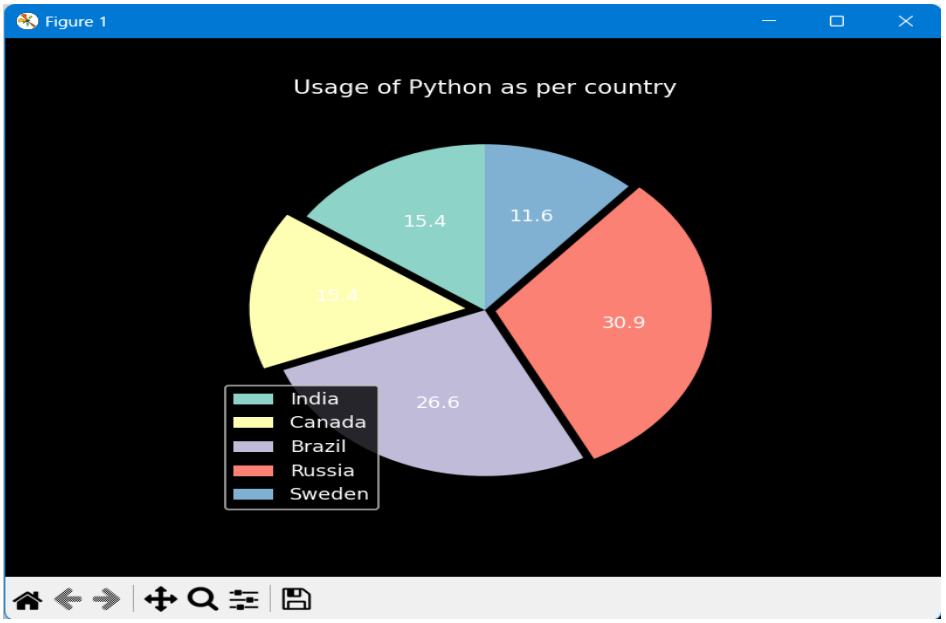


Figure 2.2.3.4

2.2.4 Seaborn

Seaborn is a widely recognized Python library for data visualization that is constructed upon Matplotlib. It provides a user-friendly interface for generating visually captivating and useful statistical images. The software is specifically engineered to seamlessly integrate with Pandas dataframes, facilitating efficient and expedient visualization and exploration of data.

Seaborn provides a diverse range of robust tools for information visualization, encompassing scatter plots, line plots, bar plots, heat maps, and numerous more options. Additionally, it offers assistance for computationally sophisticated statistical analysis techniques, including regression analysis, distribution plots, and categorical plots.

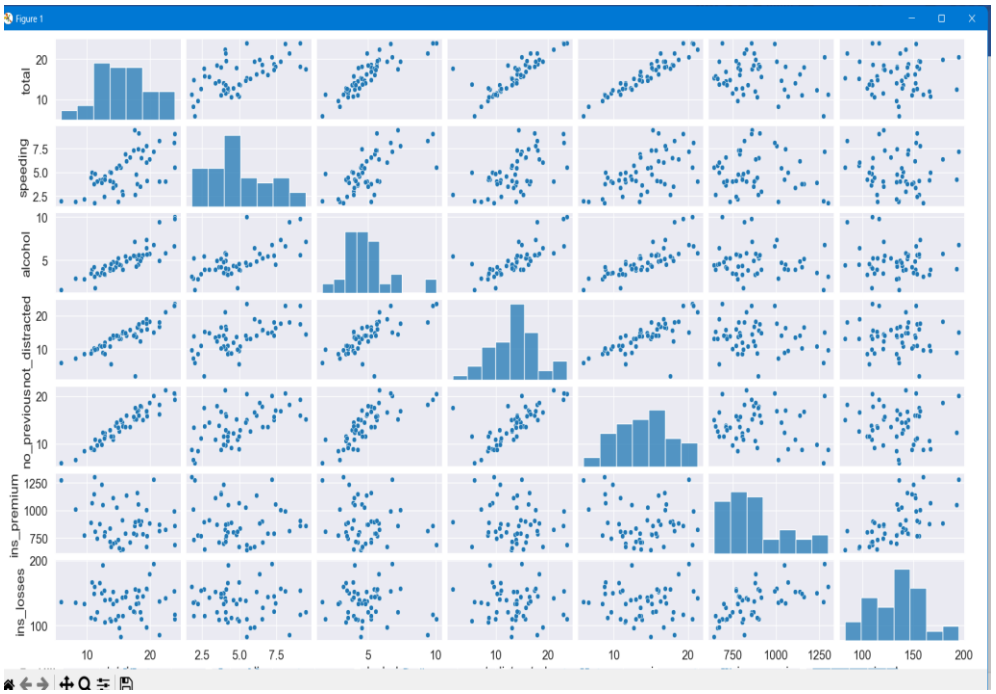


Figure 2.2.4.1

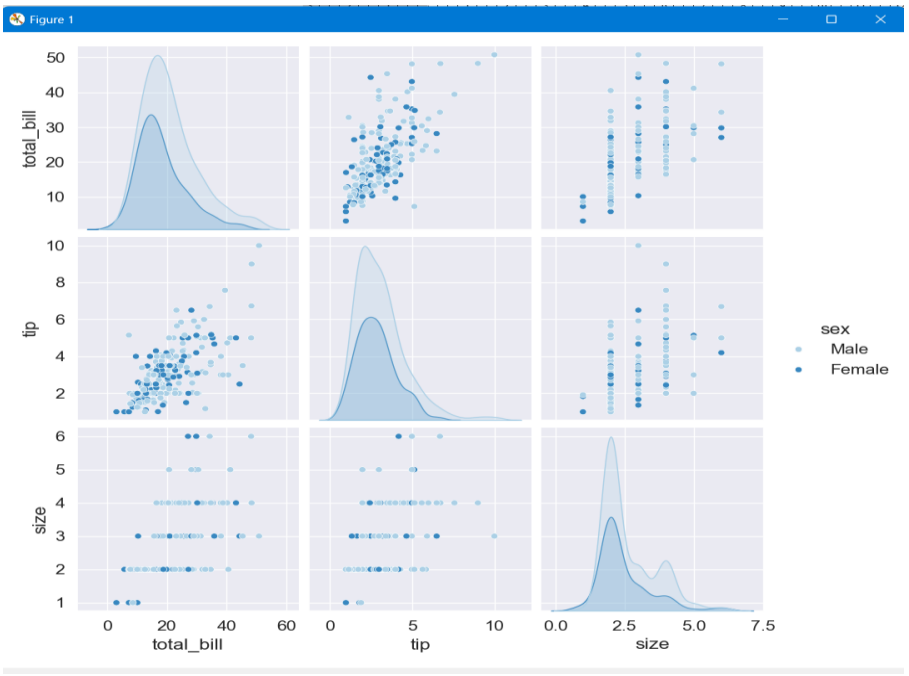


Figure 2.2.4.2

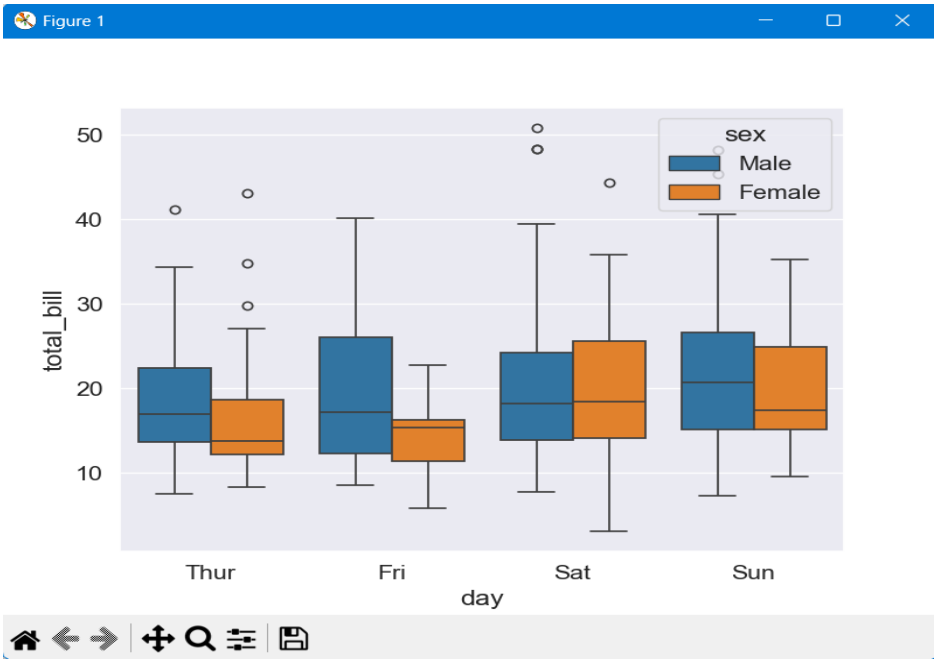


Figure 2.2.4.3

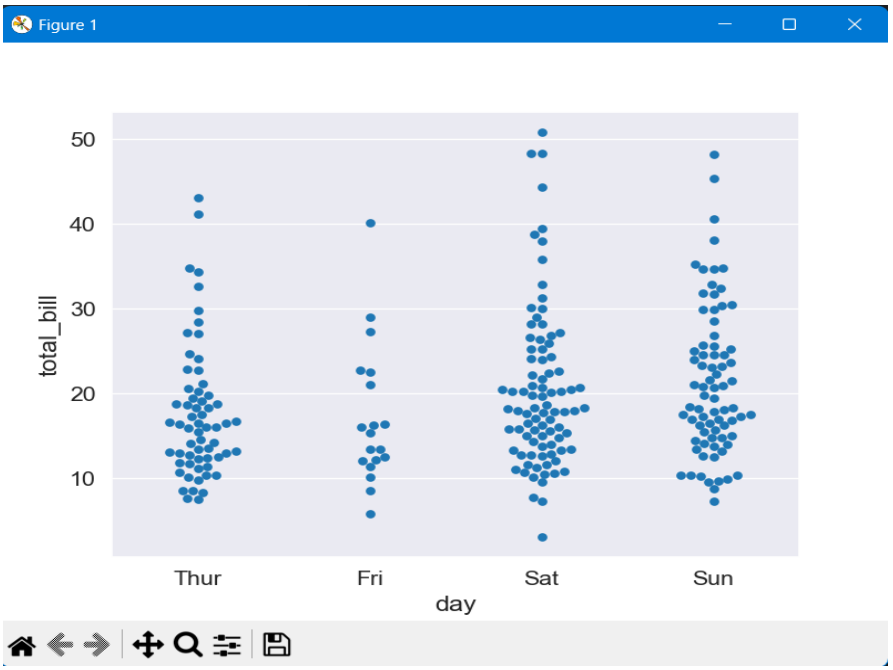


Figure 2.2.4.4

2.2.5 SciPy

SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands. As mentioned earlier, SciPy builds on NumPy and therefore if you import SciPy, there is no need to import NumPy.

```python
from scipy.optimize import minimize


new *
def f(x):
    return (x - 3) ** 2


res = minimize(f, x0: 2)
print(res)

# Constant Sub Packages

# c = speed of vacuum
# h = Plank's const.
# G = Newton's gravitational const.
# e = Elementry charge
# R = Molar gaws const.
# Avogradro = Avogradro const.
# k = Boltzman const
# m_e = Mass of electron
# m_p = Mass of proton
# m_n = Mass of newtron
# etc....

# Mathematical Constants
from scipy.constants import pi, golden_ratio

print(f"Pi = {pi}")
print(f"Golden Ratio = {golden_ratio}")
radius = 5
print(f"Area = {pi * radius * radius}")
from scipy.constants import m_n, Avogadro
```

Figure 2.2.5.1

```python
print(f"Mass of Neutron: {m_n} Kg")
print(f"Avogadro's Number: {Avogadro}")

# Values of Physical Constants
from scipy.constants import physical_constants

print(physical_constants["alpha particle mass"])
for key, value in physical_constants.items():
    print(f"{key}: {value}")

# Conversion of Unit
from scipy.constants import kilo, yotta, deci

print(f"Kilo = {kilo} gram")
meters = 800
print(f" In Kilometers: {meters / yotta}")
print(f" In Kilometers: {meters / deci}")

# Conversation of temprature
from scipy.constants import convert_temperature

temprature_in_fahrenheit = 90
temprature_in_celsius = convert_temperature(temprature_in_fahrenheit, old_scale: "F", new_scale: "C")
print(f"Temprature in Celsius: {temprature_in_celsius}")

temprature_in_celsius = 0
temprature_in_kelvin = convert_temperature(temprature_in_fahrenheit, old_scale: "C", new_scale: "K")
print(f"Temprature in Kelvin: {temprature_in_kelvin}")
```

Figure 2.2.5.2

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.vq import kmeans2

a = np.random.multivariate_normal( mean: [0, 1],  cov: [[2, 1], [1, 1.5]], size=200)
b = np.random.multivariate_normal( mean: [0, 8],  cov: [[1, -1], [-1, 3]], size=400)
c = np.random.multivariate_normal( mean: [6, 4],  cov: [[5, 0], [0, 1.2]], size=250)
z = np.concatenate((a, b, c))
np.random.shuffle(z)
# plt.scatter(z[:,0], z[:,1])
# plt.show()

K = 3

centroids, labels = kmeans2(z, K, minit="points")
print(f"Cenroids: {centroids}")
print(f"Z: {z}")
print(labels)

cluster_1 = z[labels == 0]
cluster_2 = z[labels == 1]
cluster_3 = z[labels == 2]
plt.scatter(cluster_1[:, 0], cluster_1[:, 1], label="cluster_1")
plt.scatter(cluster_2[:, 0], cluster_2[:, 1], label="cluster_2")
plt.scatter(cluster_3[:, 0], cluster_3[:, 1], label="cluster_3")
plt.scatter(centroids[:, 0], centroids[:, 1], label="Centroids", color="black")
plt.show()
```

Figure 2.2.5.3

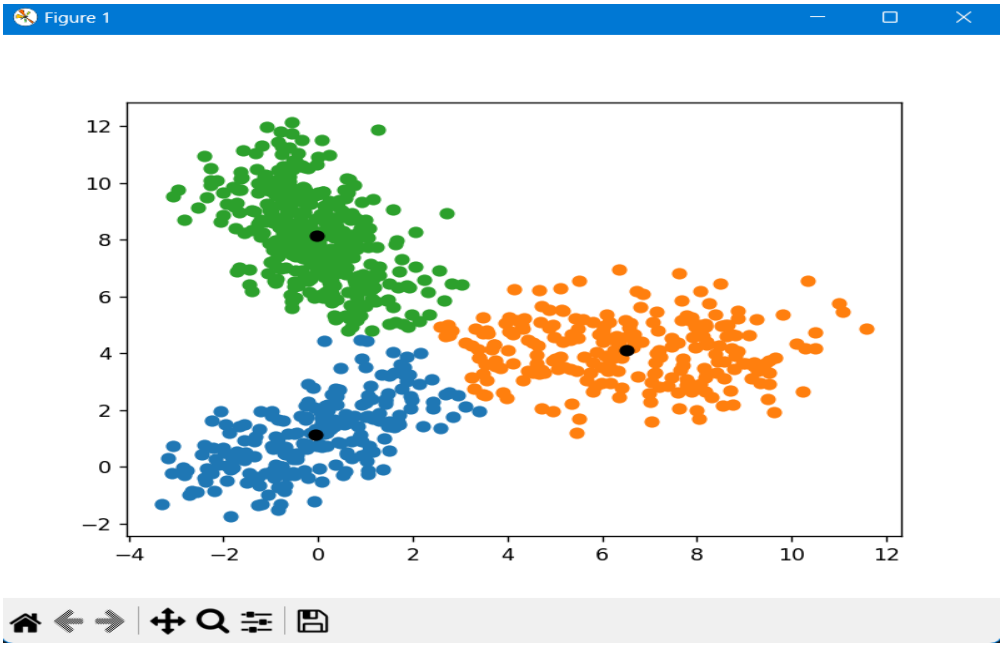


Figure 2.2.5.4

2.2.6 Scikit learn

Scikit-learn, also known as sklearn, is an open-source, machine learning and data modelling library for Python. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python libraries, NumPy and SciPy.It supports a standardized and concise model interface across models. For example, Scikit-learn makes use of a simple fit/predict workflow model for its classification algorithms.

2.2.7 OpenCV

OpenCV (Open-Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

```python
import cv2
import numpy as np

img = cv2.imread(r"E:\data science Internship\pythonProject1\archive\Vegetable Images\test\Tomato\1031.jpg")
print(img.shape)

#How to convert RGB Image to a Gray scale Image

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
print(img_gray.shape)
cv2.imshow( winname: "Vegetable_Gray", img_gray)
cv2.waitKey(0)

# Manipulate RGB
img_blue = img[:, :, 0]
img_green = img[:, :, 1]
img_red = img[:, :, 2]

new_img = np.hstack((img_blue, img_green, img_red))  # To show multiple images in same tab = hstack -- NumPy

# Image Resize
img_resize = cv2.resize(img,  dsize: (256, 256))
print(img_resize.shape)

cv2.imshow( winname: "Vegetable", img_resize)
cv2.waitKey(0)
```

Figure 2.2.7.1

```python
# Flip Image
img_flip = cv2.flip(img, flipCode: 0) # flipcode : 0 -- vertical flip, flipcode : 1 -- horizontal

cv2.imshow( winname: "Vegetable", img_flip)
cv2.waitKey(0)

# Cropping
img_crop = img[0:224, 0:200]
cv2.imshow( winname: "Vegetable", img_crop)
cv2.waitKey(0)

# Save Image
cv2.imwrite( filename: 'cropped_tomato.png', img_crop)

#Draw Shape ( which is used to scan the shape like square, circle, etc.
raw_img = np.zeros((512,512,3)) # to create a black image in which we will create shapes and te
#Rectangle
cv2.rectangle(raw_img, pt1=(100,100), pt2=(300,300),color=(255,0,0),thickness=3) #If you want t
#Circle
cv2.circle(raw_img, center=(100,400), radius=50, color=(0,0,255), thickness=-3)
#Line
cv2.line(raw_img,pt1=(0,0), pt2=(512,512),thickness=2, color=(0,255,0))
#Text
cv2.putText(raw_img, org=(350,100), fontScale=4, color=(0,255,255), thickness=2,
            lineType=cv2.LINE_AA, text="Hello", fontFace= cv2.FONT_HERSHEY_PLAIN)
```

Figure 2.2.7.2

```python
cv2.imshow( winname: "Vegetable", raw_img)
cv2.waitKey(0)

# Live Direct drawing
new *
def draw(event, x, y, flags, params):
    pass
cv2.namedWindow(winname = "Window")
img_raw = np.zeros((512,512,3))

while True:
    cv2.imshow( winname: "Image",img_raw)

    if cv2.waitKey(1) & 0xFF == ord('x'):
        break
cv2.destroyAllWindows()
```

Figure 2.2.7.3

CHAPTER 3: LEARNING PHASE - II

3.1 EXPLORING DATABASE

After going through the basics, I decided to add an authentication factor to the project. For that purpose, I explored and learned about one of the database systems – MySQL. It is based on relational database management systems (RDBMS).

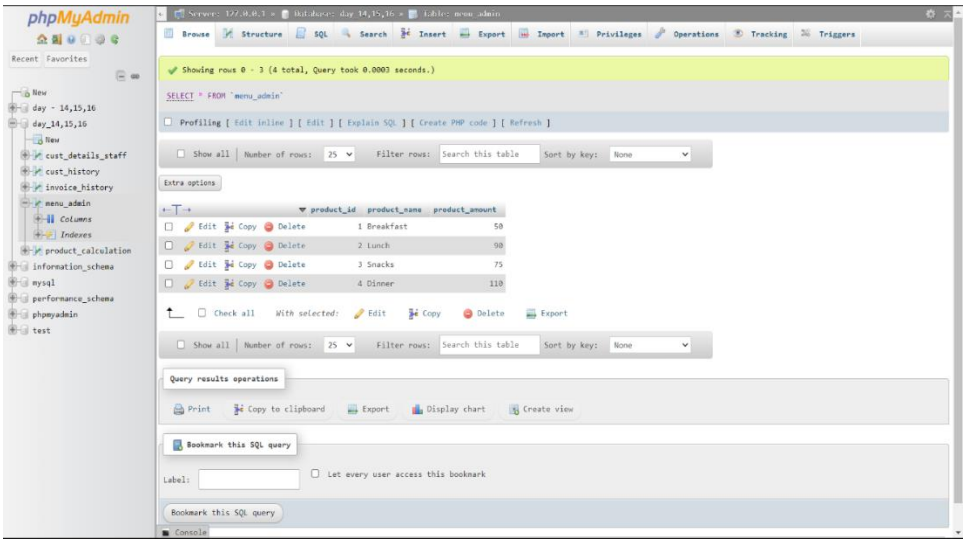


Figure 3.1.1

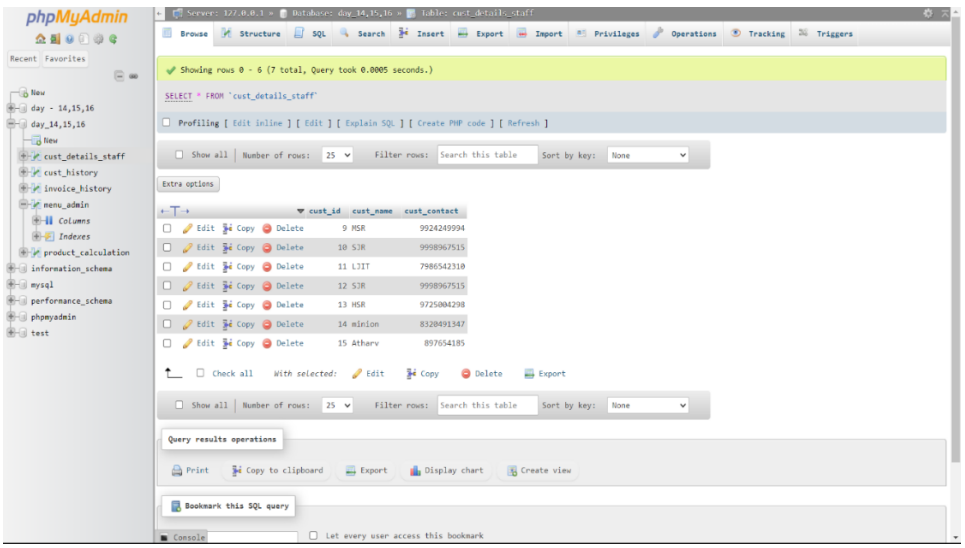


Figure 3.1.2

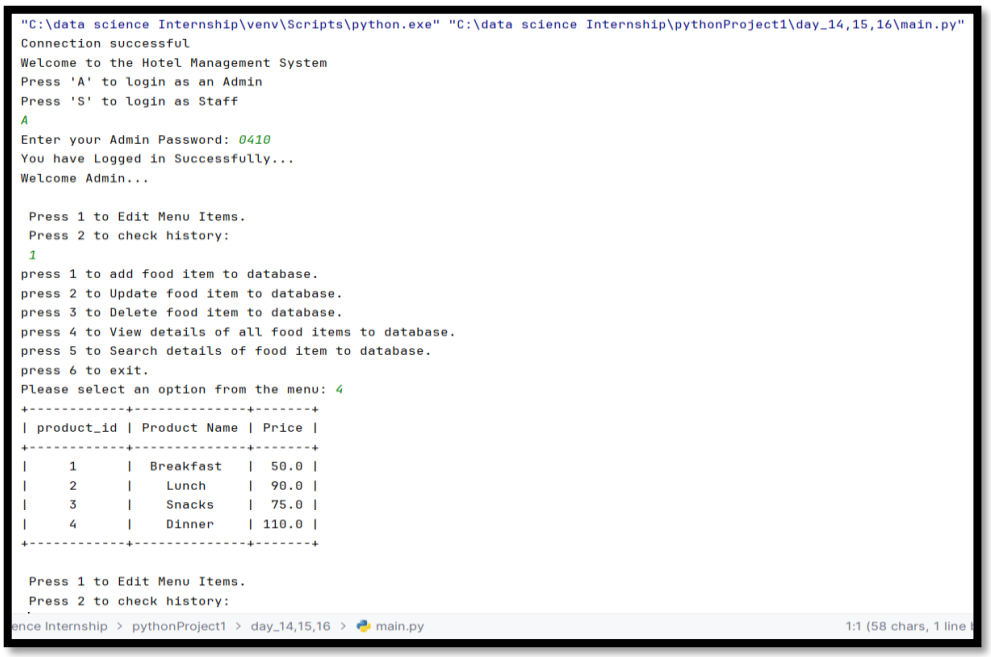


Figure 3.1.3

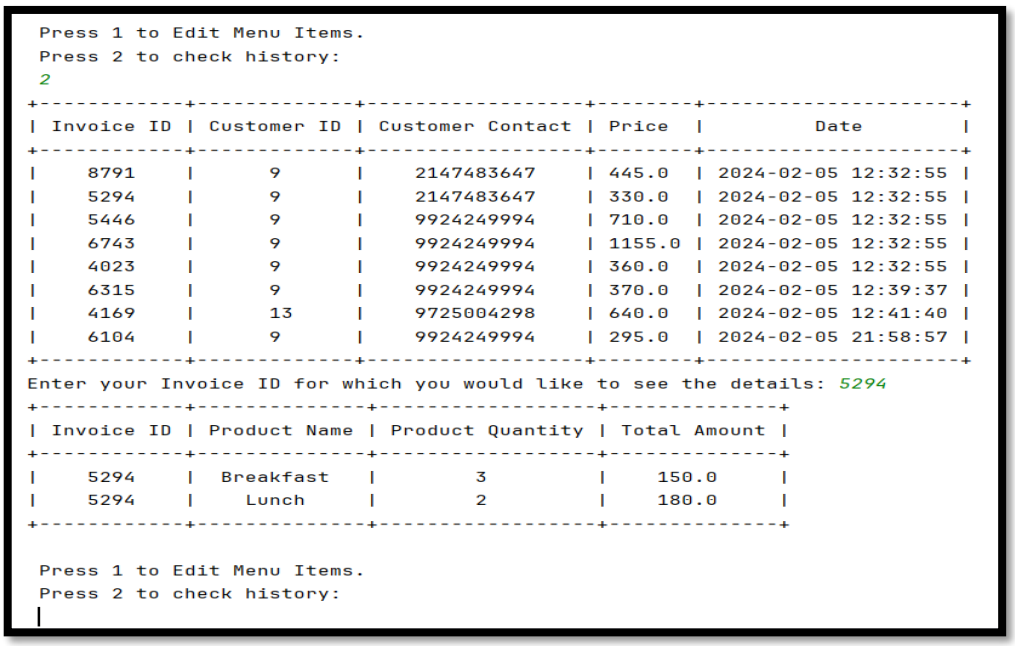


Figure 3.1.3

```
Logging out as an Admin...
Press 'A' to login as an Admin
Press 'S' to login as Staff
S
Enter your Password to login: 0410
Welcome to Hotel Management System

    Press 1 if you are going to add new user
    Press 2 if you want to access old user

Enter your choice: 1


....ADD NEW CUSTOMER DETAILS....

Enter Customer Name: Atharv
Enter Customer Contact Number: 897654185

Customer Details Added Successfully...

Press 1 to Buy Products...
Press 2 to Log out...
 1
```

Figure 3.1.4

```
Enter your contact number : 897654185
+------------+--------------+-------+
| Product ID | Product Name | Price |
+------------+--------------+-------+
|     1      |  Breakfast   |  50.0 |
|     2      |    Lunch     |  90.0 |
|     3      |    Snacks    |  75.0 |
|     4      |    Dinner    | 110.0 |
+------------+--------------+-------+
What would you like to order?
Please type the Product ID from the menu: 2
Enter the quantity of the product: 3
Order is taken. Thank you!
Do you want to add more products to your order? (yes/no): yes
+------------+--------------+-------+
| Product ID | Product Name | Price |
+------------+--------------+-------+
|     1      |  Breakfast   |  50.0 |
|     2      |    Lunch     |  90.0 |
|     3      |    Snacks    |  75.0 |
|     4      |    Dinner    | 110.0 |
+------------+--------------+-------+
What would you like to order?
Please type the Product ID from the menu: 4
Enter the quantity of the product: 1
Order is taken. Thank you!
Do you want to add more products to your order? (yes/no): no
Order processing completed. Final Amount: $380.0
Here's your Invoice ID for the payment:  3465
Press 'A' to login as an Admin
Press 'S' to login as Staff
```

Figure 3.1.5

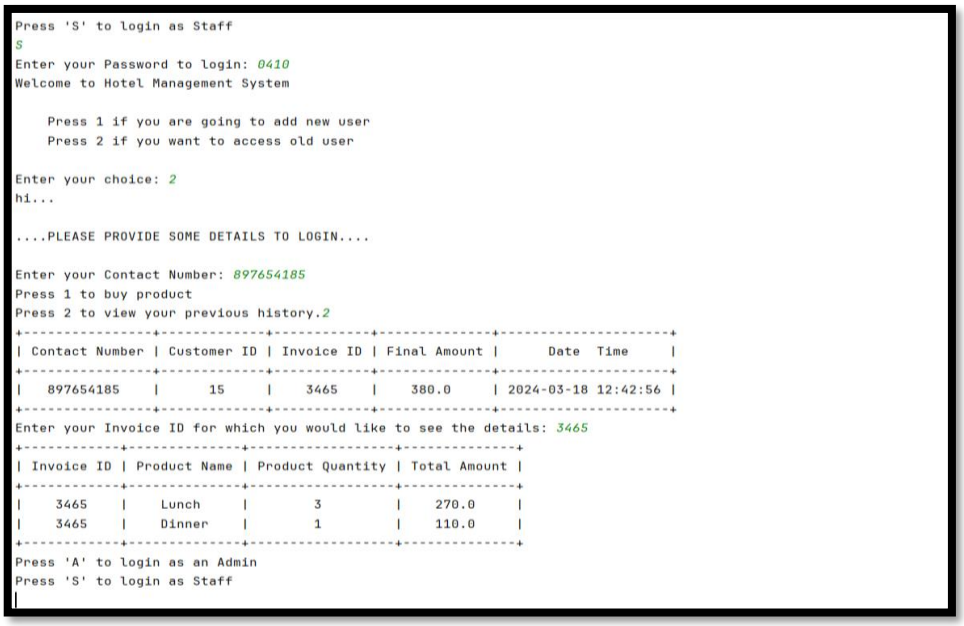


Figure 3.1.6

3.2 EXPLORING MACHINE LEARNING ALGORITHMS

Without the need for explicit programming, machine learning algorithms are models that enable computers to recognize patterns, forecast, or form opinions based on data. To lift towards final project, I have explored various machine learning algorithms Such as HOG, R-CNN, SSD, YOLO including their pros and cons.

1.1.1 Histogram of Oriented Gradients (HOG)

A feature extractor is what HOG uses to find things in a picture. In HOG, a feature description is a picture of a part of an image from which we only take out the most important details and ignore the rest. This is what the feature descriptor does: it turns the image's total size into an array or feature vector. The gradient orientation method is used in HOG to find the most important parts of an image.

There are some issues with the Histogram of Oriented Gradients (HOG): Even though it was very new when it first came out, this method had a lot of issues. It takes a long time to do complicated calculations on pixels in pictures, and it doesn't always work well when you need to find things in small spaces.

What is the best time to use HOG? – HOG is often the best way to try other object detection algorithms and see how well they work. No matter what, HOG is useful for finding most objects and recognizing face landmarks with a fair amount of accuracy.

HOG is often used to find people going down the street because it has smooth edges. Object recognition of certain things is another general use.

3.2.2 Region-based Convolutional Neural Networks (R-CNN)

We use selective features in the R-CNN models to try to get the most important features out, which are usually around 2000 features. A selective search algorithm can help you figure out how to choose the most important extractions. This can lead to these more important regional ideas.

There are some drawbacks of this algorithm: this algorithm takes long time to get all the region ideas and then the best areas, even though the CNN models that have already been trained do a good job of getting features. Not only does the R-CNN model learn slowly, but it also takes a long time to make predictions. The answer needs a lot of computer power, which makes it more likely that the process will work. The building as a whole can be thought of as quite expensive because of this.

When Should You Use R-CNN? – An R-CNN model that works like the HOG object detection method should be used as a starting point to check how well the object detection models work. Most of the time, the more recent versions of R-CNN are better because the time it takes to guess images and objects is longer than expected.

R-CNN can be used in a number of different ways to solve different kinds of object recognition problems. Tracking things from a camera on a drone, finding text in an image, and turning on object detection in Google Lens are some examples.

3.2.3 Fast R- CNN

In Fast R- CNN, all of the picture is sent through the pre-trained Convolutional Neural Network in the fast R-CNN method, rather than each sub-segment. The region of interest pooling method is unique because it uses two inputs—a pre-trained model and a selective search algorithm—to create an output layer that is fully linked. We will learn more about the Faster R-CNN network in this part. This network is better than the fast R-CNN mode.

Issues: One of the main problems with the Faster R-CNN method is that it takes too long for different items to be given. The type of system used can sometimes change the speed. Other CNN methods take longer to use. This one takes less time. R-CNN needs about 40 to 50 seconds to figure out what things are in a picture most of the time. Fast R-CNN only takes two seconds, but it finds the best answer in 0.2 seconds.

3.2.4 Single Shot Detector (SSD)

The single-shot detector for multi-box predictions is one of the fastest ways to do object detection jobs in real time. The Faster R-CNN methods can make accurate predictions, but the whole process takes a long time and needs the real-time job to run at about 7 frames per second, which is not good.

Limitations of SSD: The single-shot detector for multi-box predictions is one of the fastest ways to do object detection jobs in real time. The Faster R-CNN methods can make accurate predictions, but the whole process takes a long

time and needs the real-time job to run at about 7 frames per second, which is not good.

The single-shot detector for multi-box predictions is one of the fastest ways to do object detection jobs in real time. The Faster R-CNN methods can make accurate predictions, but the whole process takes a long time and needs the real-time job to run at about 7 frames per second, which is not good.

Different datasets, like the PASCAL VOC, COCO, and ILSVRC datasets, can be used to train and test the Single-shot detection. Larger objects, like people, tables, chairs, and other similar things, can be found quickly and accurately by them.

3.2.5 YOLO (You Only Look Once)

Many people know that YOLO is one of the best model designs and ways to find things. When you search Google for algorithms on object recognition, the YOLO design is often the first thing that comes up. Next, we'll talk about the different types of YOLO. With one of the best neural network designs, the YOLO model is able to be very accurate and work quickly. A lot of people like it because it works quickly and correctly.

One of the best things about YOLO is that it can do a lot of work very quickly compared to other object recognition and training methods. The YOLO algorithm not only has a fast-computing speed, but it also has a high level of accuracy because it reduces the background mistakes that other methods make. YOLO's design makes it easier for the model to learn about and understand many things at once.

Nevertheless, there are some limitations of YOLO such as, The memory rate is low, so you can't see small details in a picture or movie. You can't find two things that are very close to each other because of how bounding boxes work

3.2.6 Random Forest

Random Forest is a powerful ensemble learning algorithm in machine learning that operates by constructing a multitude of decision trees during training and outputting the mode (classification) or mean prediction (regression) of the individual trees. Each tree in the forest is trained on a random subset of the training data and a random subset of the features.

This randomness helps to decorrelate the individual trees, reducing the risk of overfitting and improving the model's generalization ability. During prediction, the algorithm aggregates the predictions of all the trees to produce a final prediction, making Random Forest robust and accurate in various tasks. It is widely used for classification and regression problems, as it can handle high-dimensional data, large datasets, and noisy data effectively.

Additionally, Random Forest provides valuable insights into feature importance, making it a popular choice for data analysis and predictive modeling tasks.

3.2.7 Linear Regression

Linear regression stands as a cornerstone within supervised learning, serving to anticipate continuous outcomes. Its essence lies in depicting the interconnection between a reliant variable, or target, and one or more autonomous variables, or features, through the assembly of a linear equation upon observed data. At its core, the algorithm posits that this relationship can be aptly captured by a straight line, imbuing it with simplicity and interpretability.

Through the process of training, the algorithm discerns coefficients, encompassing slope and intercept, which are honed to minimize the dissonance between predicted and actual values within the training set,

typically employing the method of least squares. Following this phase, the model stands primed to extrapolate predictions onto novel data by way of multiplying input features by their respective coefficients and aggregating them, yielding the anticipated outcome.

Linear regression's pervasive utility spans across disciplines such as economics, finance, and social sciences, underpinning essential functions like sales prognostication, trend delineation, and risk evaluation.

3.2.8 Light Gradient Boosting Model

LightGBM (Light Gradient Boosting Machine) is a powerful gradient boosting framework that has gained popularity in the machine learning community due to its efficiency and scalability. It is based on the gradient boosting decision tree (GBDT) algorithm but introduces novel techniques to improve training speed and accuracy.

LightGBM employs a leaf-wise growth strategy, where it grows the tree node-wise, prioritizing nodes with the maximum loss reduction, resulting in a deeper and potentially unbalanced tree. Additionally, it uses histogram-based algorithms to discretize continuous feature values into discrete bins, reducing memory usage and speeding up the training process.

LightGBM also implements techniques such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to further enhance efficiency while maintaining high predictive performance. These features make LightGBM well-suited for large-scale datasets and real-world applications where speed and accuracy are crucial.

3.2.9 Decision Tree

The decision tree algorithm is a highly adaptable supervised learning method utilized across machine learning tasks encompassing classification and regression. It functions by iteratively segmenting datasets into subsets

based on attributes deemed most informative.

At each node, the algorithm identifies the attribute offering optimal data segregation, with the objective of maximizing homogeneity within each subset. This iterative process persists until predefined termination conditions, such as reaching a specified maximum tree depth, are met. Decision trees present notable simplicity and interpretability, rendering them conducive to visualization and comprehension even among non-specialists.

Nevertheless, they are susceptible to overfitting, particularly evident in instances of deep trees or data containing substantial noise. Despite these limitations, decision trees maintain relevance across diverse domains, including finance, healthcare, and marketing, owing to their capacity to handle diverse data types and intrinsic feature selection capabilities.

3.2.10 Naïve Base

The Naive Bayes algorithm serves as a fundamental probabilistic classifier extensively utilized in machine learning applications such as text classification, spam detection, and recommendation systems. It relies on Bayes' theorem to estimate the probability of a hypothesis given observed evidence. Despite its straightforward nature, Naive Bayes demonstrates considerable efficacy, particularly in scenarios involving high-dimensional datasets.

The "naive" descriptor pertains to its assumption of feature independence, positing that features are conditionally independent given the class label. This assumption simplifies probability calculations, rendering the algorithm computationally efficient, especially for large datasets.

Although the independence assumption may not always hold in real-world contexts, Naive Bayes frequently yields satisfactory results and serves as a benchmark model for evaluating more sophisticated algorithms.

3.3 EXPLORING FLASK FRAMEWORK

Flask stands out as a lightweight and adaptable web framework tailored for Python, engineered to streamline the process of constructing web applications swiftly and effortlessly. Recognized for its straightforwardness, adaptability, and minimalist design ethos, Flask empowers developers to craft web applications with a notably reduced burden of repetitive code.

Fundamental to Flask are its provisions for crucial web development functionalities including URL routing, template rendering, and request management. By adhering to the WSGI (Web Server Gateway Interface) specification, Flask ensures compatibility across a spectrum of web servers and deployment configurations.

Moreover, Flask boasts extensive extensibility, boasting a rich ecosystem of extensions that enable the integration of additional features such as authentication mechanisms, database functionality, and RESTful API support. The inherent simplicity and modular architecture of Flask render it a preferred choice for both novices and seasoned developers seeking to construct web applications efficiently.

CHAPTER 4: SYSTEM ANALYSIS

After learning enough about my training modules, I was introduced to a task. I had to make views and templates for admin interface. I had to look after whether all the implemented concepts of Data Science were working well or not**.**

4.1 INTRODUCTION

The benefits of this website were to give a user a new feeling of using and learning data science and machine learning. Where he can explore more about this domain and can look after a few amazing projects that works on those concepts.

4.2 GOALS

The primary objective of the project is to empower users with the capability to perform analysis on any dataset through interactive interfaces, eliminating the need for them to develop algorithms from the ground up.

4.4 FEASIBILITY STUDY

Feasibility study is a process that identifies, describes and evaluates proposed system and selects the best system. During the study, the problem definition is solved and all aspects of the problem to be included in the system are determined. Size of project, cost of benefits is also estimated with greater accuracy. A good feasibility study will show the strengths and defects before the project is planned or budgeted.

4.4.1 Technical Study

The feasibility study evaluates the availability and integration of necessary technology for the proposed system, considering its stakeholders, particularly users. The technical assessment examines whether users possess the requisite expertise to comprehend and utilize the new system effectively.

This evaluation relies on preliminary system requirements design to ascertain the technical and legal viability of the proposal. It involves scrutinizing the hardware and software components to ensure they align with the needs of the proposed system. Furthermore, it examines whether the technology can be developed using existing equipment and possesses the technical capability to accommodate the data requirements of the current system.

4.4.2 Economic Study

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification. It identifies the financial benefits and cost associated with the development of the system. Economic feasibility is often known as the cost benefit analysis. To carry out an economic feasibility study it is necessary to estimate actual money value against activities needed for implementing the system.

While implementing our system we can ensure that the cost of the prospective new venture will ultimately be profitable to the people. So, we can say it is financially feasible.

4.4.3 Operational Study

Operational feasibility evaluates the extent to which a proposed system effectively addresses identified problems and leverages opportunities outlined during scope definition. It assesses how well the system meets the requirements identified in the requirement analysis phase of system development. This evaluation centers on the alignment of the proposed project with the existing environment and objectives, particularly concerning the development schedule. Operational feasibility scrutinizes aspects related to human resources, organizational structure, and political considerations.

CHAPTER 5: SYSTEM DESIGN

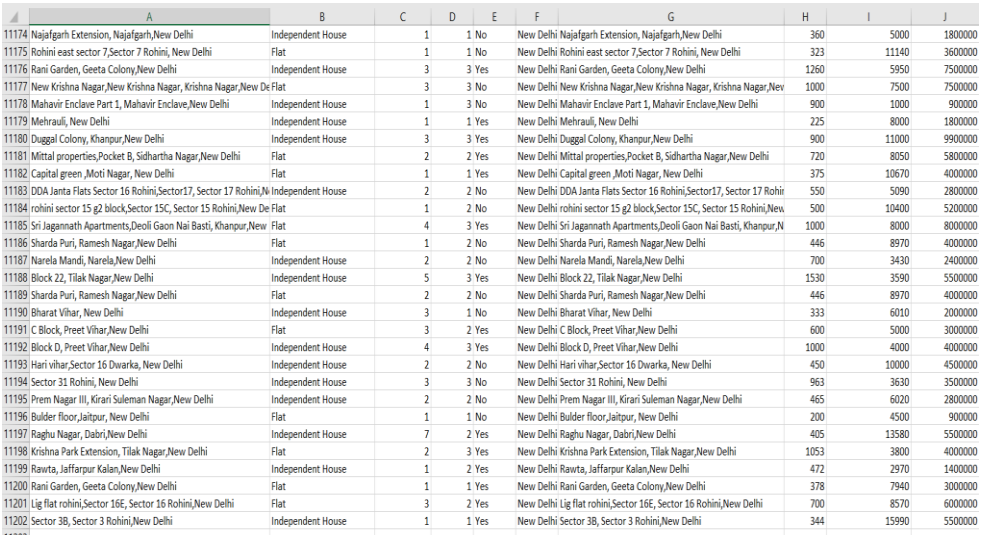
The purpose of this document is to give a detailed description of the prerequisite steps for the project. It will illustrate the complete declaration for the development of the system.

5.1 EXPLORING DATASET

Understanding the dataset holds paramount importance in constructing a precise predictive model for housing prices. The primary objective during data analysis is to identify patterns, anomalies, and other pertinent factors that can contribute to predicting housing prices accurately. A comprehensive exploration of all variables influencing housing prices, encompassing features like balcony presence, bedroom and bathroom count, is integral to this process. Additionally, categorical variables such as location and availability might also be included in the dataset. This systematic approach facilitates the extraction of invaluable insights from the dataset, paving the way for the development of robust predictive models for house price estimation.

	A	B	C	D	E	F	G	H	I	J
1	Name	Property_Type	Bedrooms	Baths	Balcony	City	Location	Total_Area	Price_per_SQFT	Price
2	DAC Prapthi	Flat	3	3	No	Chennai	Kasthuribai Nagar, West Tambaram,Chennai	1320	7580	10000000
3	DRA 90 Degrees	Flat	3	3	No	Chennai	Rose Avenue, Pallikaranai,Chennai	1444	9000	13000000
4	CasaGrand Vistaaz	Villa	3	3	No	Chennai	Kolapakkam - Vandalur, Chennai	1566	6900	10800000
5	Vasantham	Flat	3	3	Yes	Chennai	Maruthi Nagar, Sembakkam,Chennai	1712	6720	11500000
6	Golden Jubilee Flats	Flat	3	3	Yes	Chennai	Thirumangalam, Anna Nagar West,Chennai	1100	9090	10000000
7	Prestige Bella Vista	Flat	3	3	Yes	Chennai	Iyyappanthangal, Chennai	1791	7260	13000000
8	BSR Nerkundram	Independent House	4	3	No	Chennai	Tirumalai Nagar, Koyambedu,Chennai	1400	8210	11500000
9	Pacifica Happiness Towers	Flat	4	4	Yes	Chennai	Padur, Chennai	2232	5600	12500000
10	Pacifica Aurum Villas	Villa	4	4	Yes	Chennai	Padur, Chennai	2117	6610	14000000
11	Prestige Courtyards by Prestige Estates Projects Ltd.	Flat	3	3	Yes	Chennai	Sholinganallur, Chennai	1689	7820	13200000
12	Casagrand Arena	Villa	3	3	Yes	Chennai	Oragadam Ambattur, Chennai	1545	8090	12500000
13	Pacifica Aurum Villas by Pacifica Companies	Flat	4	4	Yes	Chennai	Padur, Chennai	2200	6820	15000000
14	Navins Kaushik	Flat	3	3	Yes	Chennai	Ramakrishna Puram, West Mambalam,Chennai	1269	12610	16000000
15	Sudharshan Homes	Flat	2	3	No	Chennai	Periyar Nagar, Velachery,Chennai	1152	9000	10400000
16	Krishna Flats	Flat	2	3	No	Chennai	Natesa Nagar, Virugambakkam,Chennai	1365	8420	11500000
17	Vijay Raja Classic	Villa	3	3	Yes	Chennai	Unamancheri, Kolapakkam, Kolapakkam - Vandalur,Chennai	1734	6340	11000000
18	VGN Coasta	Flat	3	3	No	Chennai	Muttukadu, Chennai	1540	8440	13000000
19	Tulip Apartment	Flat	2	3	No	Chennai	Kamdar Nagar, Nungambakkam,Chennai	1150	11740	13500000
20	Man Pearl Reside	Flat	3	3	Yes	Chennai	Thiruvenkatapuram, Choolaimedu,Chennai	1260	8730	11000000
21	Jain West Minster	Flat	3	3	Yes	Chennai	Pushpa Colony, Saligramam,Chennai	1385	10830	15000000
22	Jadhav Jain Atisaya	Flat	2	3	No	Chennai	Trustpuram, Kodambakkam,Chennai	1120	10710	12000000
23	VGN LA PARISIENNE	Flat	3	3	Yes	Chennai	Nolambur, Ambattur Industrial Estate,Chennai	1309	8400	11000000
24	Mahalakshmi Apartments	Flat	3	3	No	Chennai	Rajaji Nagar Extention, Madipakkam,Chennai	1149	10880	12500000
25	Sri Rama Nivas	Flat	3	3	Yes	Chennai	Reddi Kuppam, West Mambalam,Chennai	1225	10610	13000000
26	Appaswamy Arcot Terrace	Flat	2	3	No	Chennai	Somasundara Bharathi Nagar, Vadapalani,Chennai	1231	13000	16000000
27	Radiance Icon	Flat	2	3	No	Chennai	Nerkundram, Koyambedu,Chennai	1002	11980	12000000
28	Purva Windermere	Flat	3	4	Yes	Chennai	Pallikaranai, Chennai	2160	6480	14000000
29	Nakar Navkar Residency by Nakar Homes	Flat	3	3	Yes	Chennai	Korukkupet, Washermanpet,Chennai	1750	7260	12700000

Figure 5.1.1



	A	B	C	D	E	F	G	H	I	J
11174	Najafgarh Extension, Najafgarh,New Delhi	Independent House	1	1	No	New Delhi	Najafgarh Extension, Najafgarh,New Delhi	360	5000	1800000
11175	Rohini east sector 7,Sector 7 Rohini, New Delhi	Flat	1	1	No	New Delhi	Rohini east sector 7,Sector 7 Rohini, New Delhi	323	11140	3600000
11176	Rani Garden, Geeta Colony,New Delhi	Independent House	3	3	Yes	New Delhi	Rani Garden, Geeta Colony,New Delhi	1260	5950	7500000
11177	New Krishna Nagar,New Krishna Nagar, Krishna Nagar,New De	Flat	3	3	No	New Delhi	New Krishna Nagar,New Krishna Nagar, Krishna Nagar,Nev	1000	7500	7500000
11178	Mahavir Enclave Part 1, Mahavir Enclave,New Delhi	Independent House	1	3	No	New Delhi	Mahavir Enclave Part 1, Mahavir Enclave,New Delhi	900	1000	900000
11179	Mehrauli, New Delhi	Independent House	1	1	Yes	New Delhi	Mehrauli, New Delhi	225	8000	1800000
11180	Duggal Colony, Khanpur,New Delhi	Independent House	3	3	Yes	New Delhi	Duggal Colony, Khanpur,New Delhi	900	11000	9900000
11181	Mittal properties,Pocket B, Sidhartha Nagar,New Delhi	Flat	2	2	Yes	New Delhi	Mittal properties,Pocket B, Sidhartha Nagar,New Delhi	720	8050	5800000
11182	Capital green ,Moti Nagar, New Delhi	Flat	1	1	Yes	New Delhi	Capital green ,Moti Nagar, New Delhi	375	10670	4000000
11183	DDA Janta Flats Sector 16 Rohini,Sector17, Sector 17 Rohini,N	Independent House	2	2	No	New Delhi	DDA Janta Flats Sector 16 Rohini,Sector17, Sector 17 Rohir	550	5090	2800000
11184	rohini sector 15 g2 block,Sector 15C, Sector 15 Rohini,New De	Flat	1	2	No	New Delhi	rohini sector 15 g2 block,Sector 15C, Sector 15 Rohini,New	500	10400	5200000
11185	Sri Jagannath Apartments,Deoli Gaon Nai Basti, Khanpur,New	Flat	4	3	Yes	New Delhi	Sri Jagannath Apartments,Deoli Gaon Nai Basti, Khanpur,N	1000	8000	8000000
11186	Sharda Puri, Ramesh Nagar,New Delhi	Flat	1	2	No	New Delhi	Sharda Puri, Ramesh Nagar,New Delhi	446	8970	4000000
11187	Narela Mandi, Narela,New Delhi	Independent House	2	2	No	New Delhi	Narela Mandi, Narela,New Delhi	700	3430	2400000
11188	Block 22, Tilak Nagar,New Delhi	Independent House	5	3	Yes	New Delhi	Block 22, Tilak Nagar,New Delhi	1530	3590	5500000
11189	Sharda Puri, Ramesh Nagar,New Delhi	Flat	2	2	No	New Delhi	Sharda Puri, Ramesh Nagar,New Delhi	446	8970	4000000
11190	Bharat Vihar, New Delhi	Independent House	3	1	No	New Delhi	Bharat Vihar, New Delhi	333	6010	2000000
11191	C Block, Preet Vihar,New Delhi	Flat	3	2	Yes	New Delhi	C Block, Preet Vihar,New Delhi	600	5000	3000000
11192	Block D, Preet Vihar,New Delhi	Independent House	4	3	Yes	New Delhi	Block D, Preet Vihar,New Delhi	1000	4000	4000000
11193	Hari vihar,Sector 16 Dwarka, New Delhi	Independent House	2	2	No	New Delhi	Hari vihar,Sector 16 Dwarka, New Delhi	450	10000	4500000
11194	Sector 31 Rohini, New Delhi	Independent House	3	3	No	New Delhi	Sector 31 Rohini, New Delhi	963	3630	3500000
11195	Prem Nagar III, Kirari Suleman Nagar,New Delhi	Independent House	2	2	No	New Delhi	Prem Nagar III, Kirari Suleman Nagar,New Delhi	465	6020	2800000
11196	Bulder floor,Jaitpur, New Delhi	Flat	1	1	No	New Delhi	Bulder floor,Jaitpur, New Delhi	200	4500	900000
11197	Raghu Nagar, Dabri,New Delhi	Independent House	7	2	Yes	New Delhi	Raghu Nagar, Dabri,New Delhi	405	13580	5500000
11198	Krishna Park Extension, Tilak Nagar,New Delhi	Flat	2	3	Yes	New Delhi	Krishna Park Extension, Tilak Nagar,New Delhi	1053	3800	4000000
11199	Rawta, Jaffarpur Kalan,New Delhi	Independent House	1	2	Yes	New Delhi	Rawta, Jaffarpur Kalan,New Delhi	472	2970	1400000
11200	Rani Garden, Geeta Colony,New Delhi	Flat	1	1	Yes	New Delhi	Rani Garden, Geeta Colony,New Delhi	378	7940	3000000
11201	Lig flat rohini,Sector 16E, Sector 16 Rohini,New Delhi	Flat	3	2	Yes	New Delhi	Lig flat rohini,Sector 16E, Sector 16 Rohini,New Delhi	700	8570	6000000
11202	Sector 3B, Sector 3 Rohini,New Delhi	Independent House	1	1	Yes	New Delhi	Sector 3B, Sector 3 Rohini,New Delhi	344	15990	5500000

Figure 5.1.2

5.2 DATA CLEANING

Data cleaning serves as the fundamental cornerstone of any data-centric undertaking, pivotal in ensuring precision, reliability, and credibility in subsequent analyses. This process is instrumental in upholding dataset consistency and completeness through meticulous handling of missing values, standardization of formats, and error detection. By addressing discrepancies and rectifying inconsistencies, data quality is significantly enhanced, rendering it more conducive to insightful modeling and analysis. Moreover, the elimination of outliers and redundant data not only enhances the efficacy and performance of predictive models but also streamlines the analytical workflow. Ultimately, the rigorous practice of data cleaning facilitates informed decision-making and facilitates the derivation of actionable insights by averting erroneous conclusions and instilling confidence in the accuracy of outcomes.

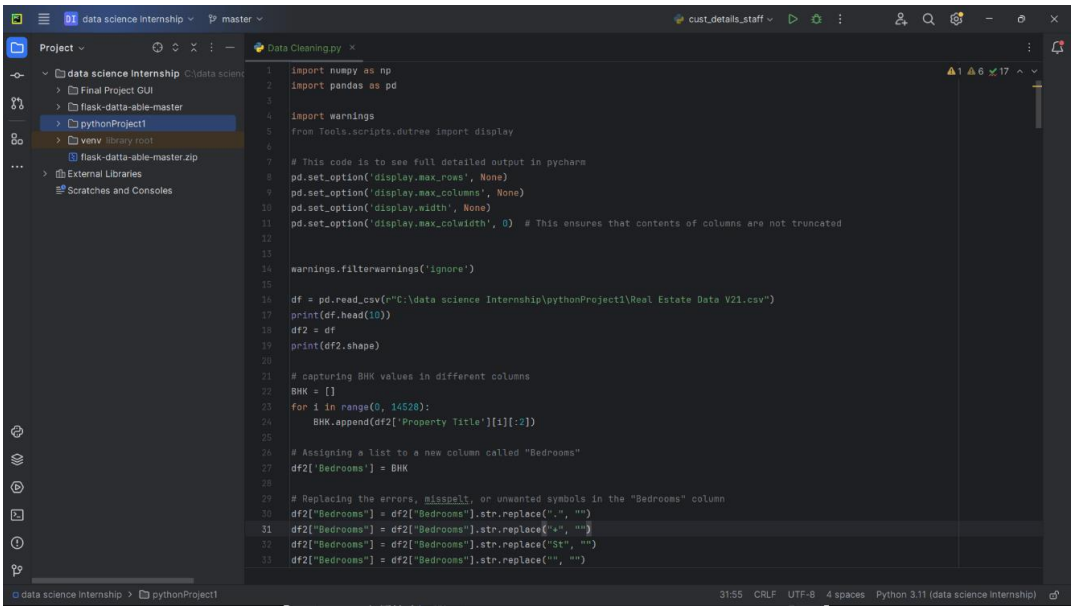


Figure 5.2.1

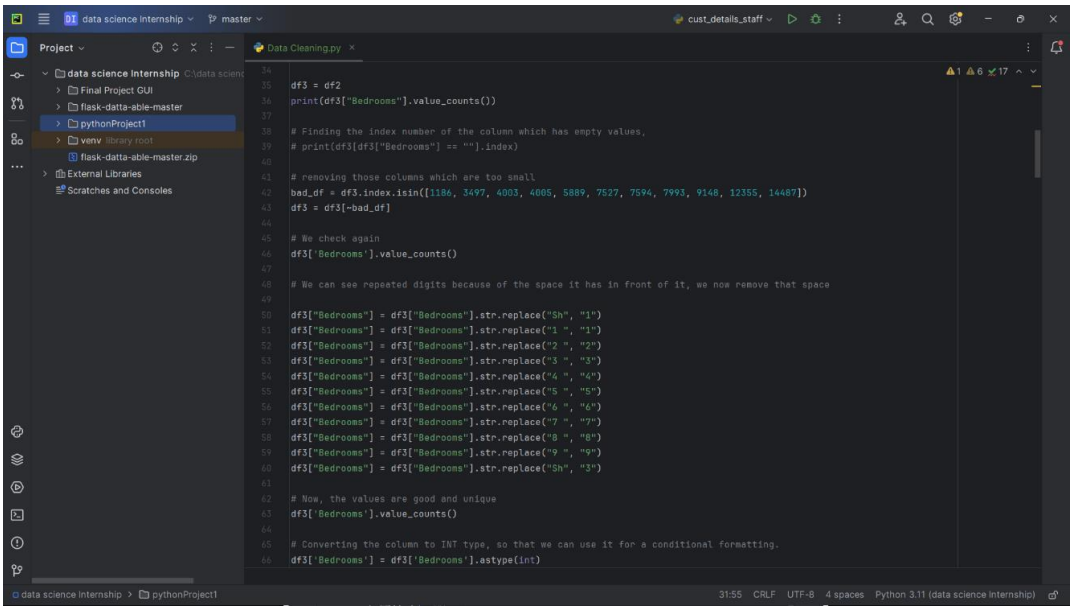


Figure 5.2.2

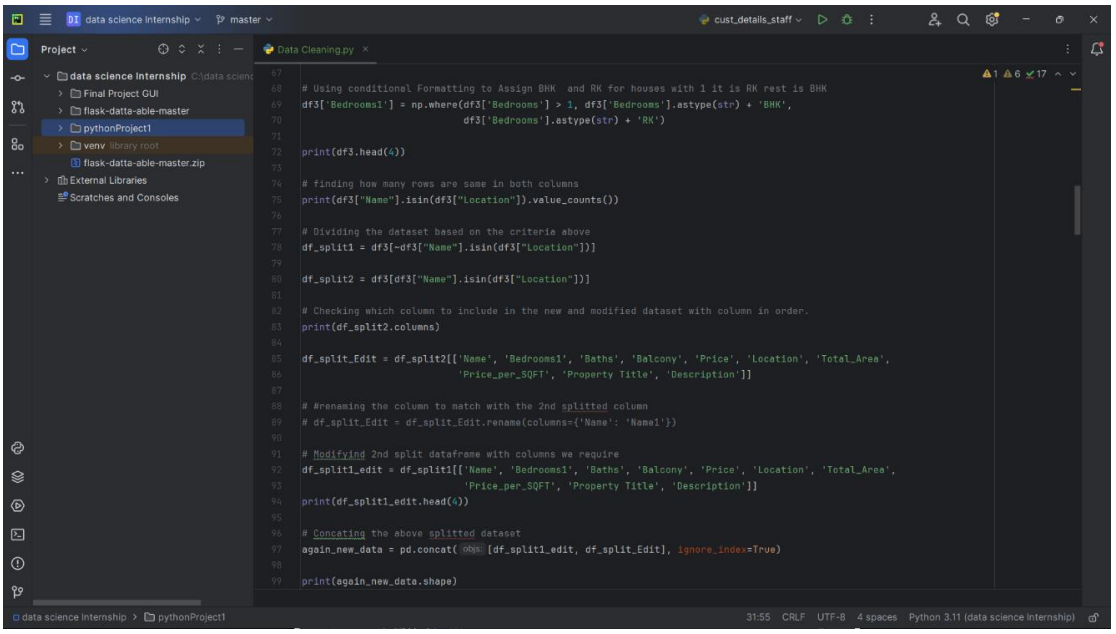


Figure 5.2.3

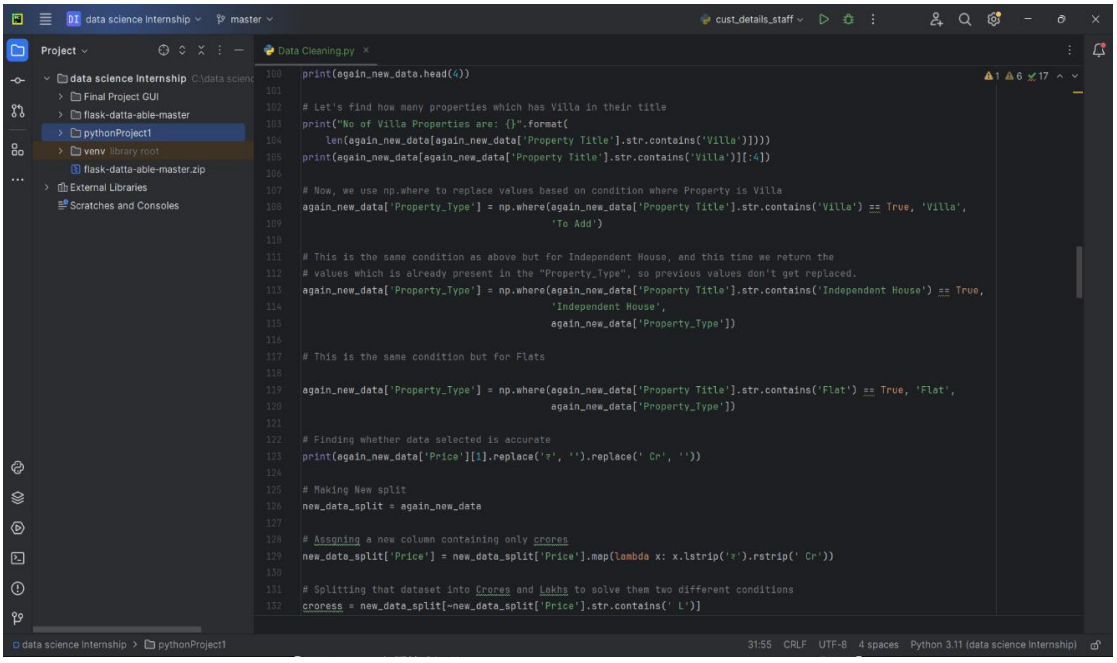


Figure 5.2.4

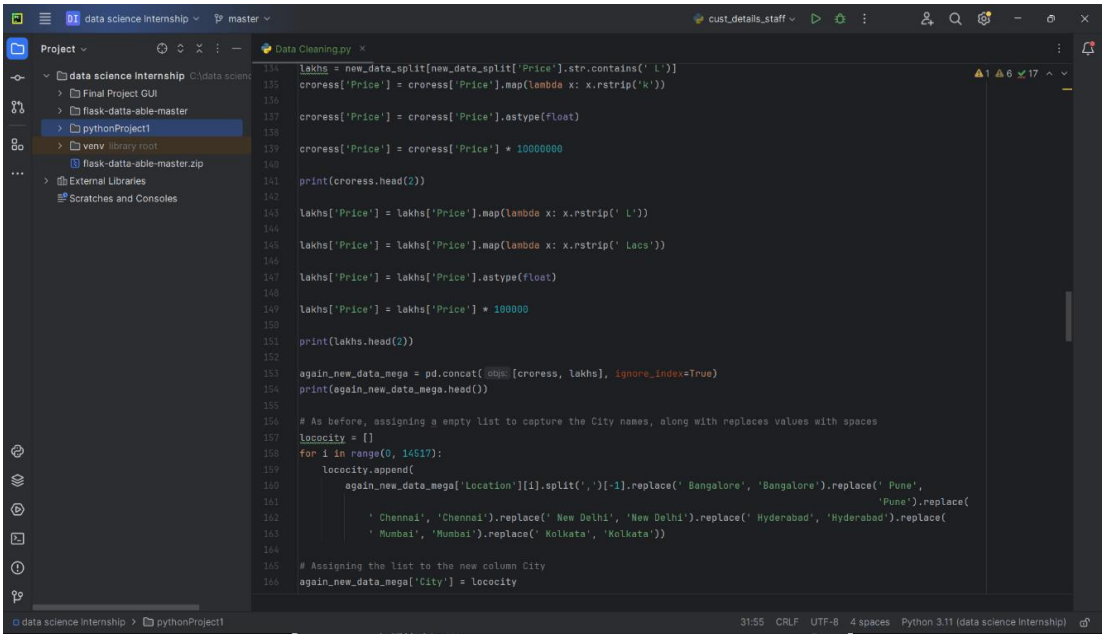


Figure 5.2.5

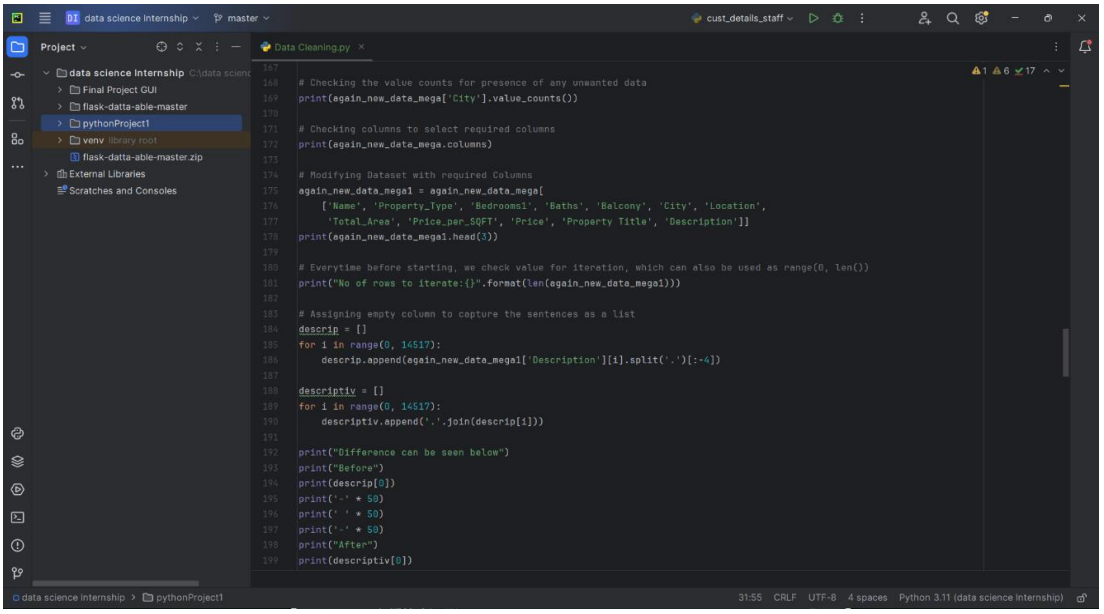


Figure 5.2.6

```python
# Assigning list to the column
again_new_data_mega1['Description1'] = descriptiv

# Dropping not needed columns
again_new_data_mega2 = again_new_data_mega1.drop('Description', axis=1)

# Finalized and clean Dataset
print(again_new_data_mega2.head(4))

again_new_data_mega2.to_csv('again_new_data_mega2.csv', index=False)
print(again_new_data_mega2.head())

# to check Null values
print(again_new_data_mega2.isna().sum())
```

Figure 5.2.7

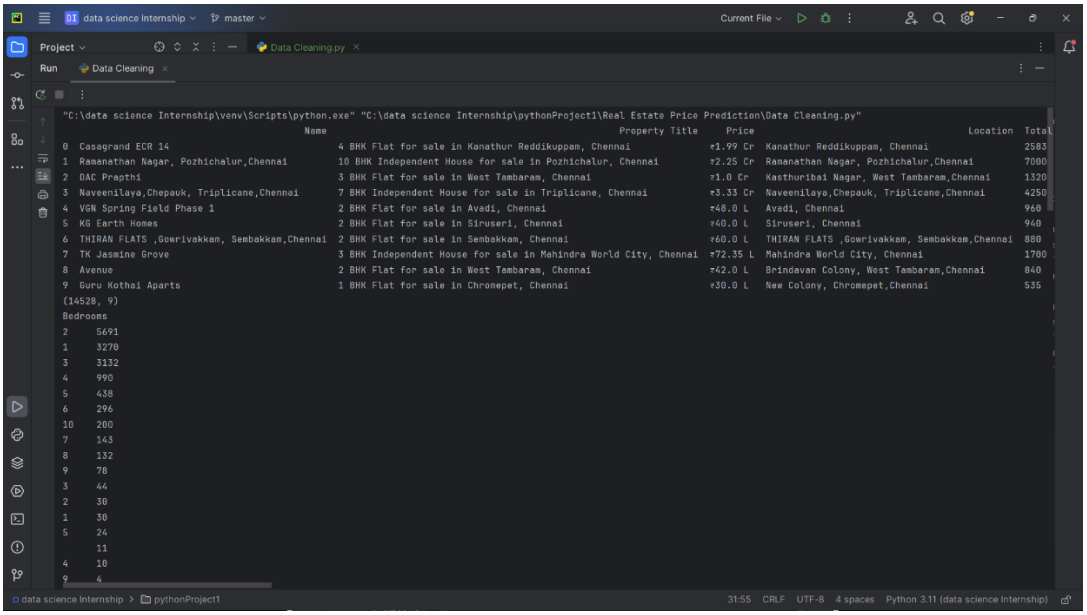


Figure 5.2.8

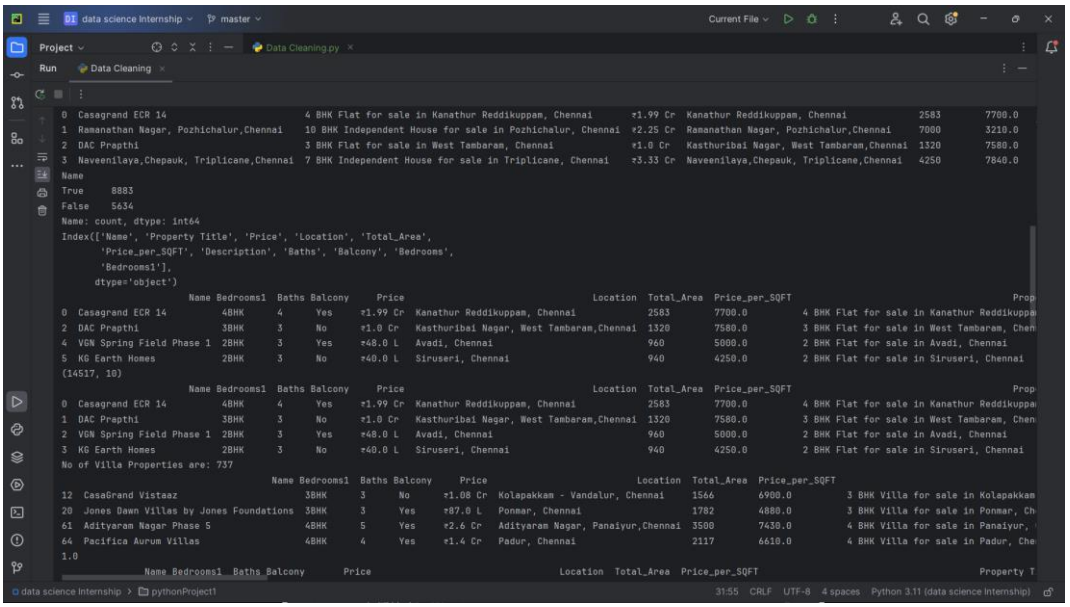


Figure 5.2.9

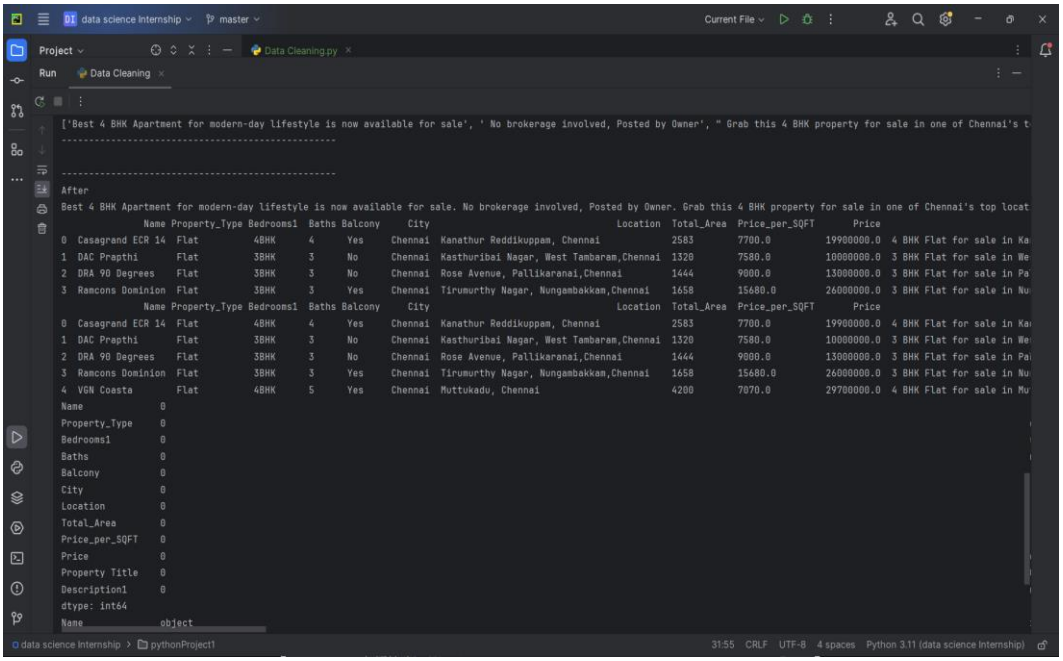


Figure 5.2.10

5.3 EXPLORATORY DATA ANALYSIS (EDA)

A vital first step in the data analysis process is exploratory data analysis (EDA), which seeks to comprehend the underlying patterns, trends, and correlations within a dataset. It entails analyzing the data both statistically and visually in order to draw conclusions and develop theories for additional research. EDA is crucial because it gives analysts a thorough picture of the data, enabling them to spot outliers, missing numbers, and other problems with data quality early on. Analysts can decide on data cleaning, feature engineering, and modeling methodologies by developing a deeper grasp of the dataset's properties. Typically, exploratory statistical analyses like correlation matrices and clustering algorithms are used in conjunction with descriptive statistics and data visualization tools like histograms, scatter plots, and box plots to undertake exploratory data analysis (EDA).

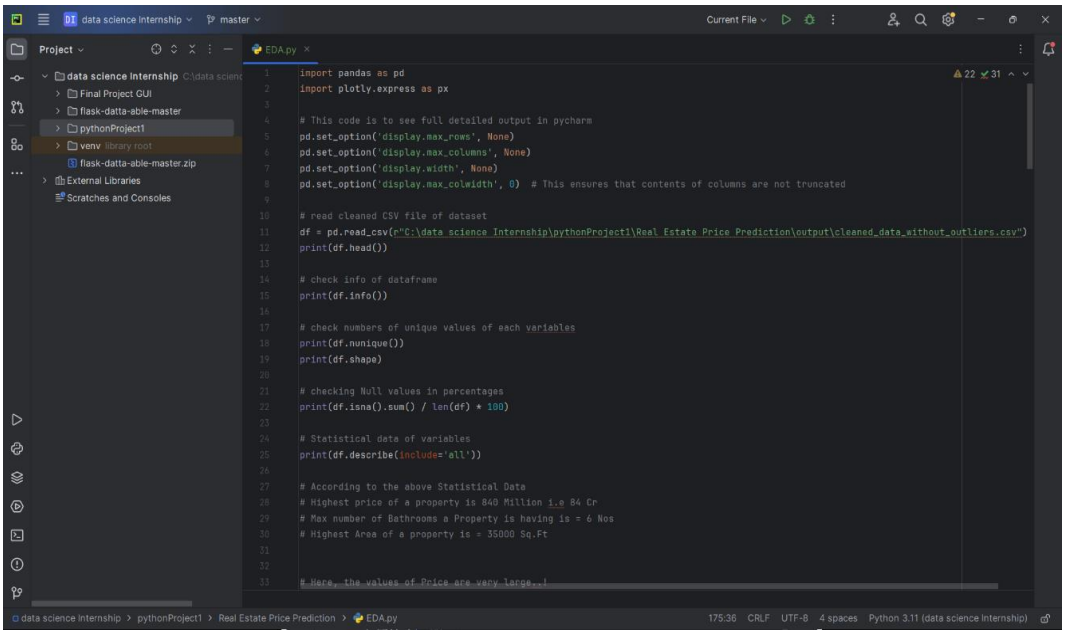


Figure 5.3.1

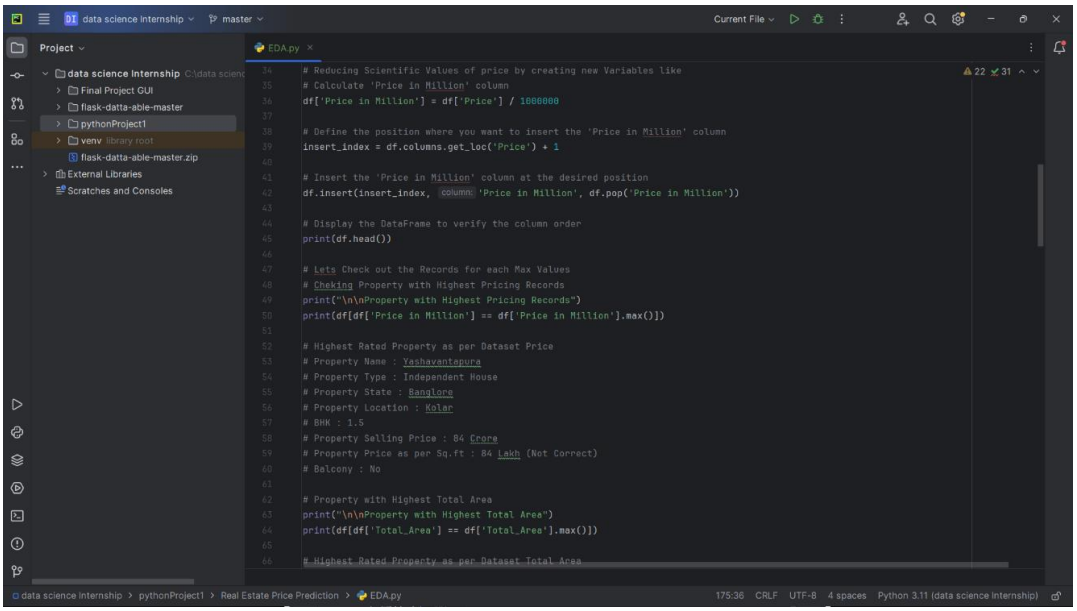


Figure 5.3.2

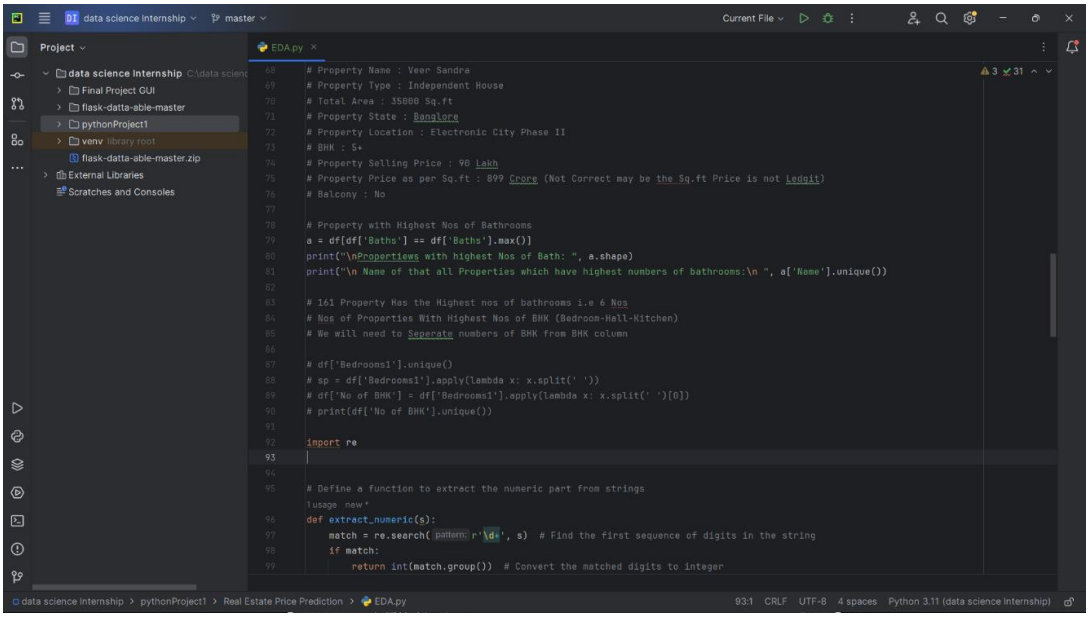


Figure 5.3.3

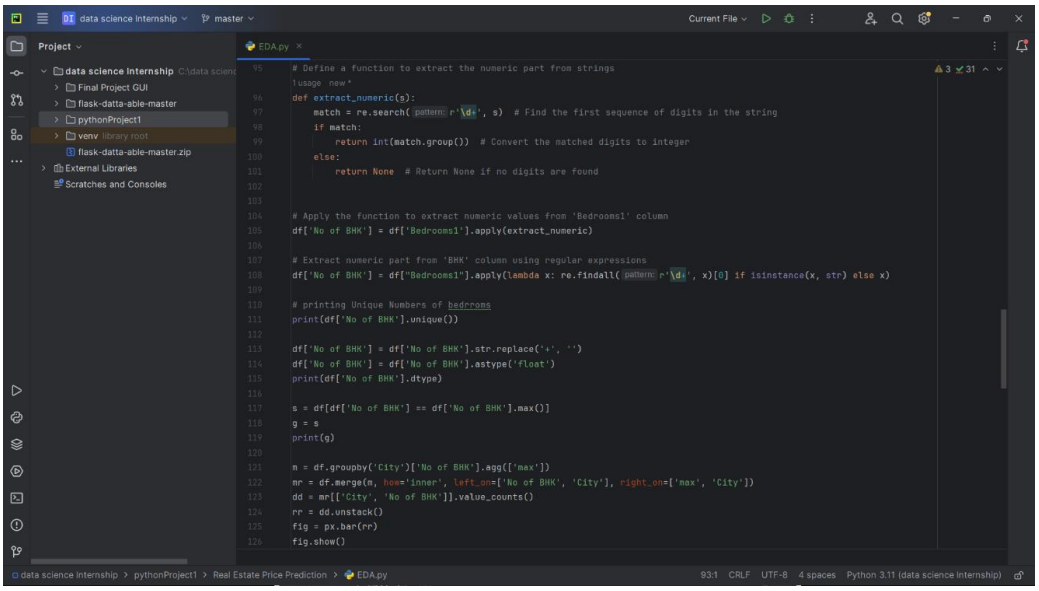


Figure 5.3.4

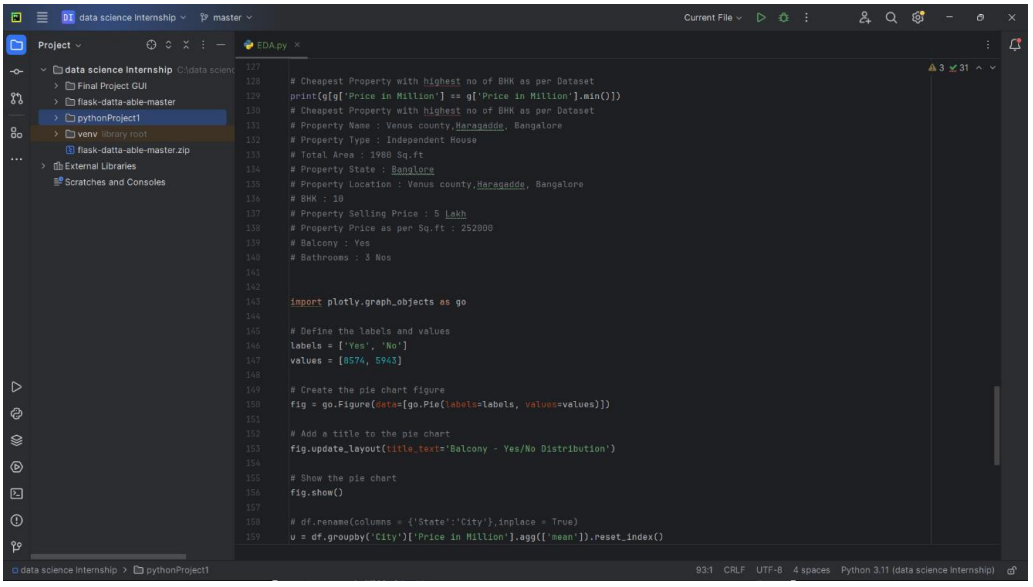


Figure 5.3.5

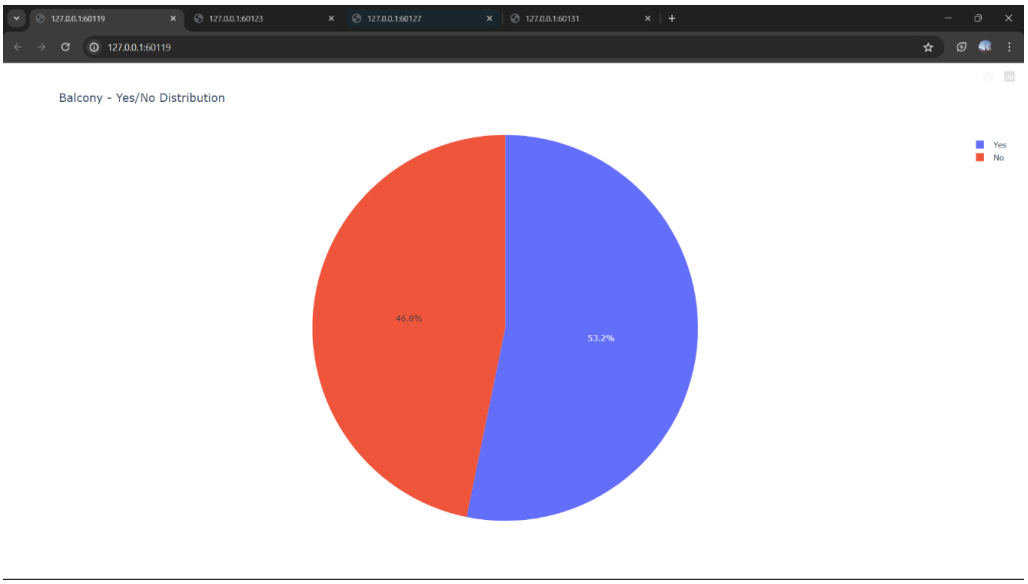


Figure 5.3.6

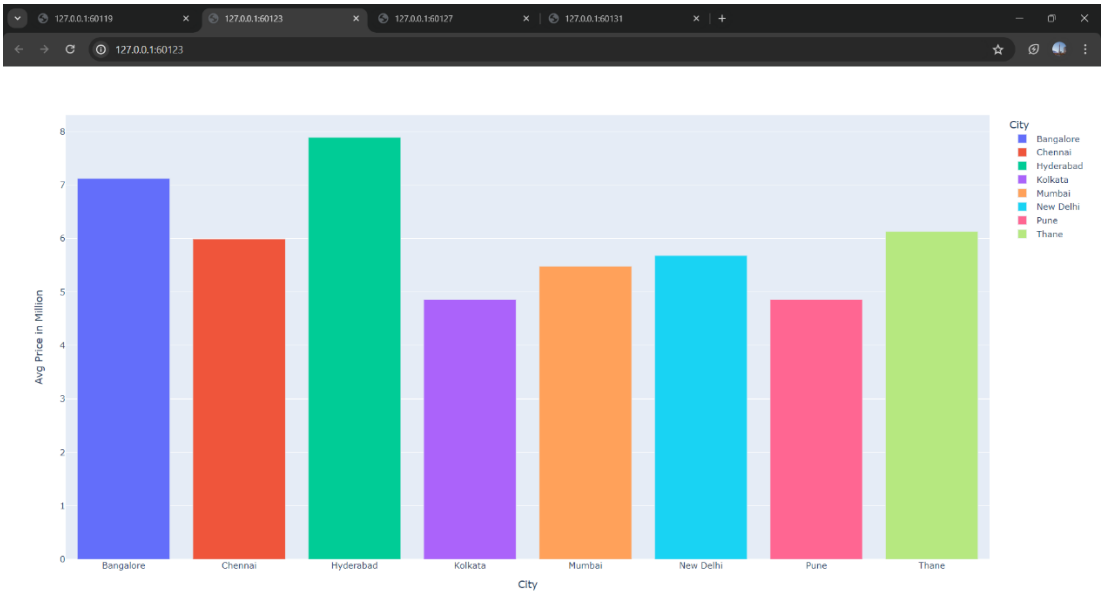


Figure 5.3.7

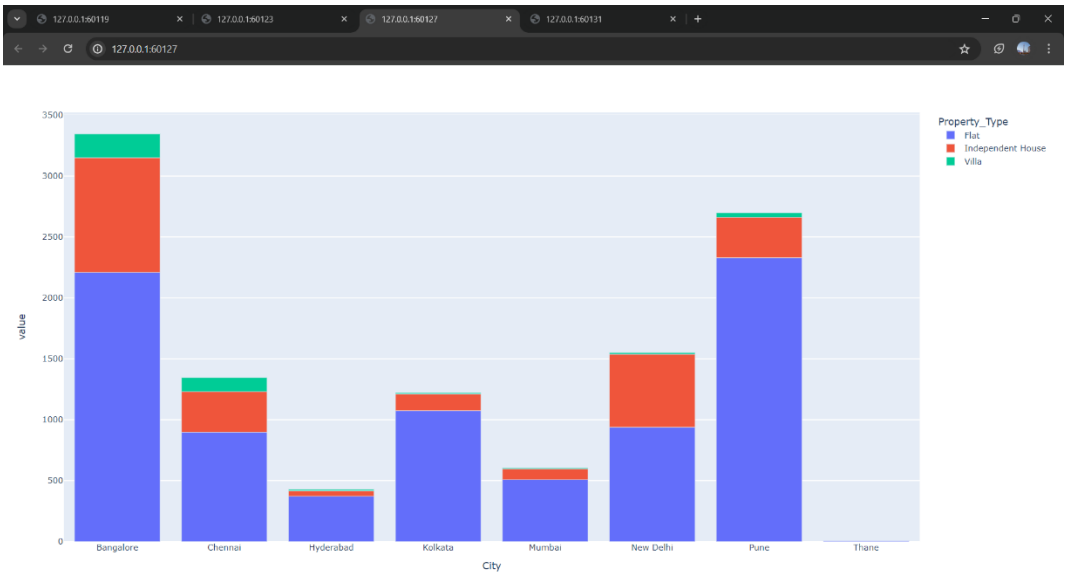


Figure 5.3.8

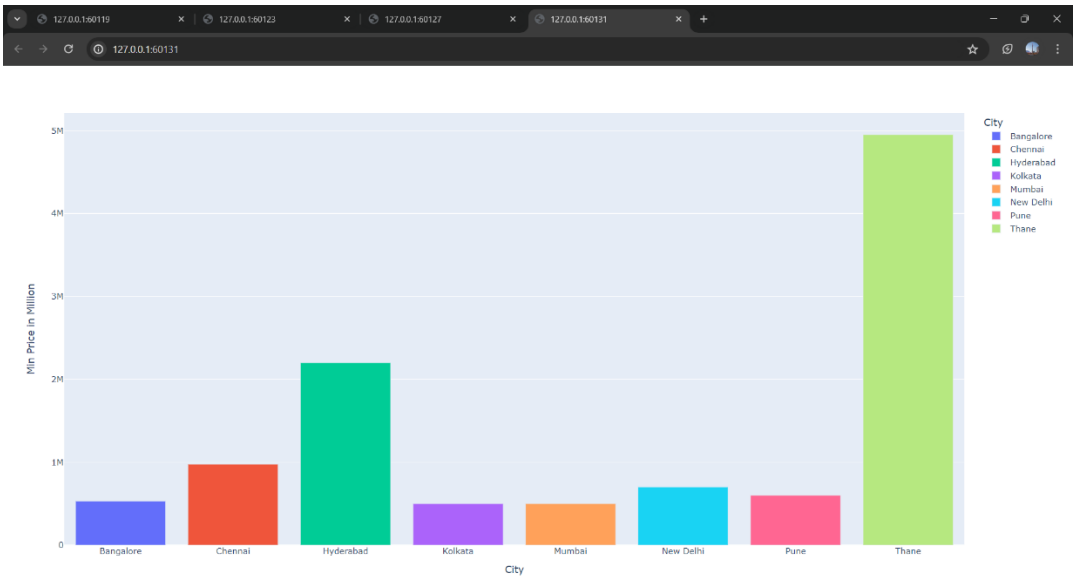


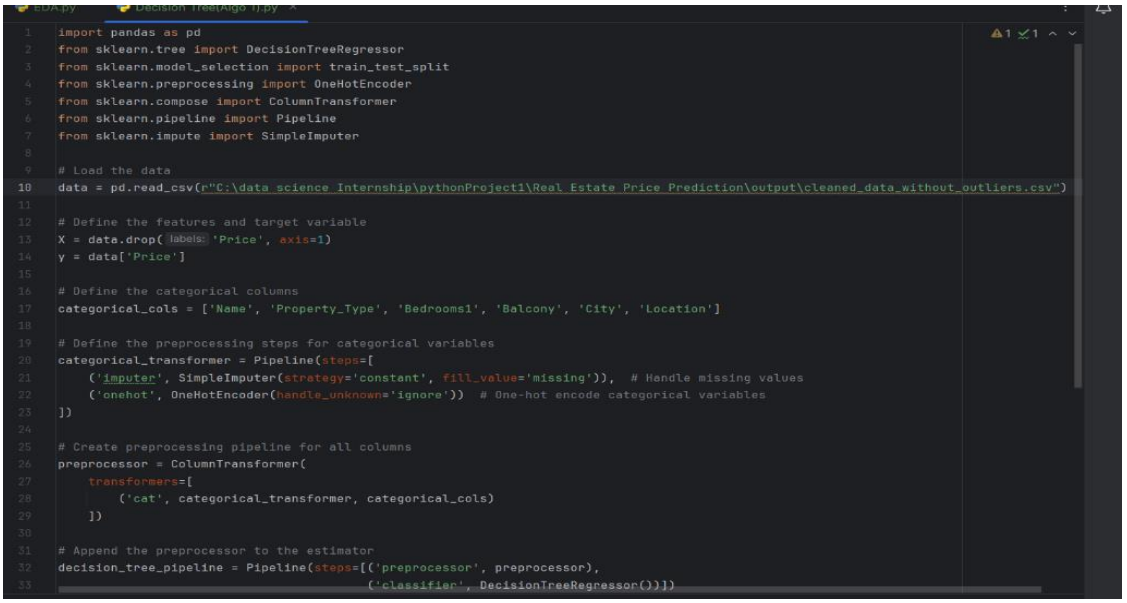
Figure 5.3.9

In any data analysis or machine learning endeavor, the selection of algorithms assumes paramount importance as it fundamentally shapes the success trajectory and modeling approach of the project. Leveraging my dataset, I conducted an array of algorithms to forecast pricing and subsequently assessed the accuracy post model training.

5.4 ALGORITHMS

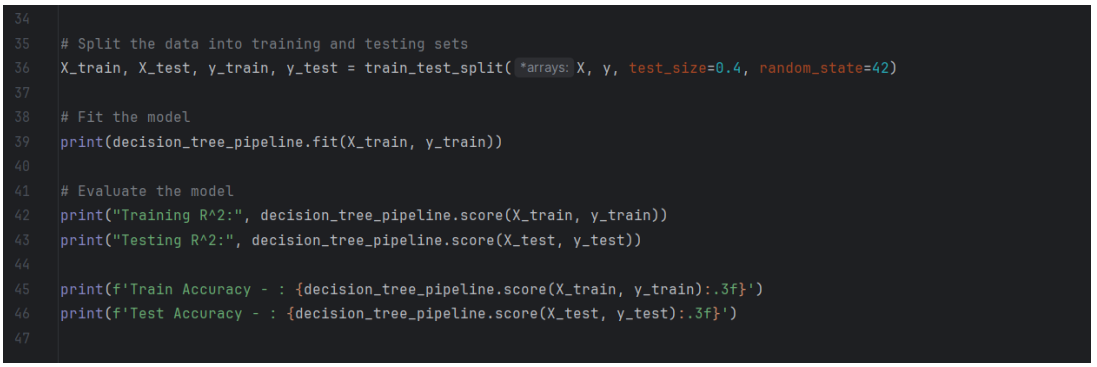
An approach that typically uses recursive operations to solve a mathematical problem in a limited number of steps

5.4.1 Decision Tree



```python
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer

# Load the data
data = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")

# Define the features and target variable
X = data.drop(labels: 'Price', axis=1)
y = data['Price']

# Define the categorical columns
categorical_cols = ['Name', 'Property_Type', 'Bedrooms1', 'Balcony', 'City', 'Location']

# Define the preprocessing steps for categorical variables
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),  # Handle missing values
    ('onehot', OneHotEncoder(handle_unknown='ignore'))  # One-hot encode categorical variables
])

# Create preprocessing pipeline for all columns
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', categorical_transformer, categorical_cols)
    ])

# Append the preprocessor to the estimator
decision_tree_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                         ('classifier', DecisionTreeRegressor())])
```

Figure 5.4.1.1



```python

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.4, random_state=42)

# Fit the model
print(decision_tree_pipeline.fit(X_train, y_train))

# Evaluate the model
print("Training R^2:", decision_tree_pipeline.score(X_train, y_train))
print("Testing R^2:", decision_tree_pipeline.score(X_test, y_test))

print(f'Train Accuracy - : {decision_tree_pipeline.score(X_train, y_train):.3f}')
print(f'Test Accuracy - : {decision_tree_pipeline.score(X_test, y_test):.3f}')

```

Figure 5.4.1.2

```
Training R^2: 0.9904563145340908
Testing R^2: 0.22782508762903408
Train Accuracy - : 0.990
Test Accuracy - : 0.228
```

Figure 5.4.1.3

5.4.2 Random Forest

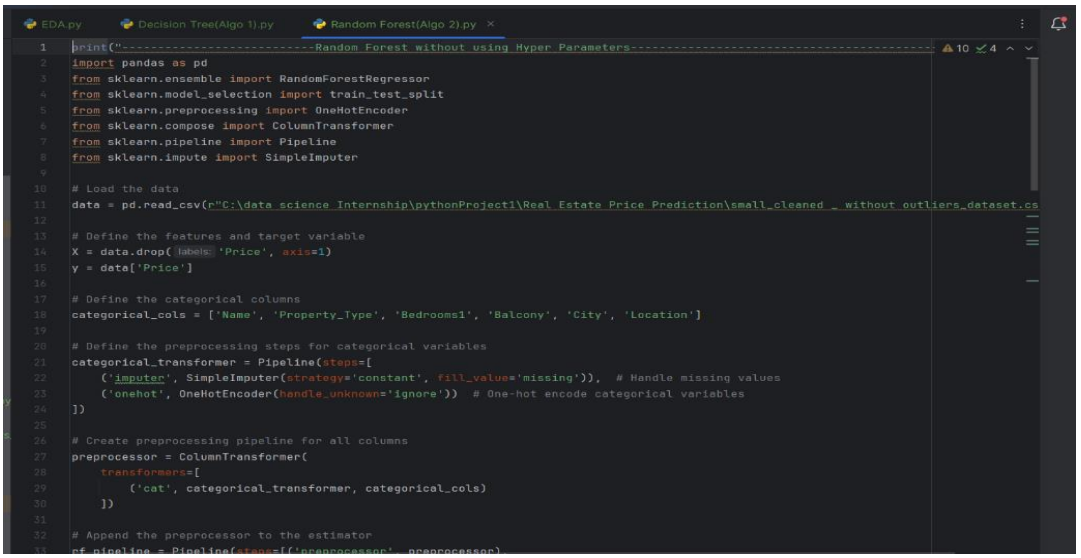


Figure 5.4.2.1

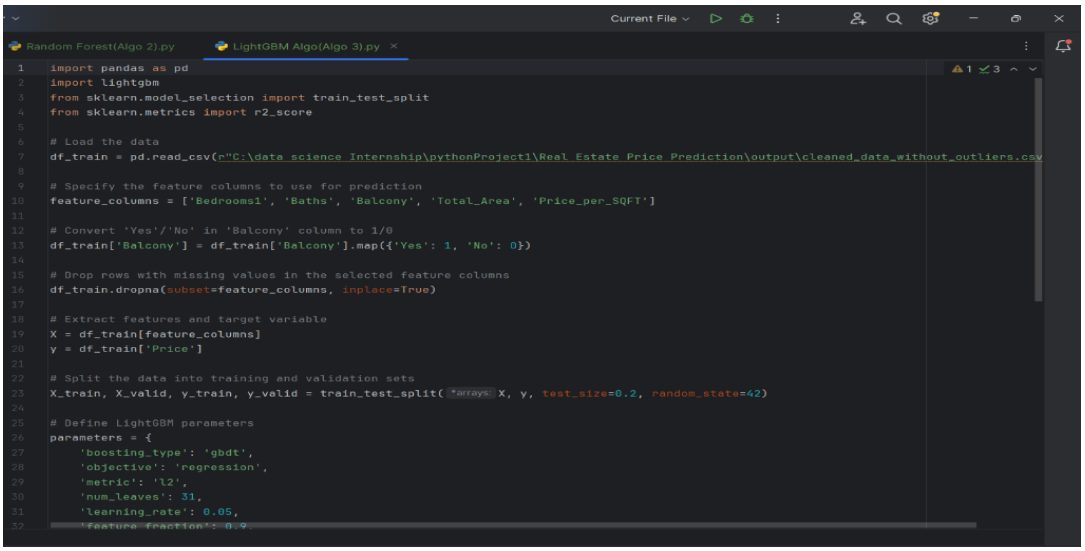
```
# Append the preprocessor to the estimator
rf_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                              ('classifier', RandomForestRegressor())])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size=0.4, random_state=42)

# Fit the model
print(rf_pipeline.fit(X_train, y_train))

# Evaluate the model
print("Training R^2:", rf_pipeline.score(X_train, y_train))
print("Testing R^2:", rf_pipeline.score(X_test, y_test))

print(f'Train Accuracy - : {rf_pipeline.score(X_train, y_train):.3f}')
print(f'Test Accuracy - : {rf_pipeline.score(X_test, y_test):.3f}')


# OUTPUT
# Training R^2: 0.9060332750431791
# Testing R^2: 0.347827611955025
# Train Accuracy - : 0.906
# Test Accuracy - : 0.348
```

Figure 5.4.2.2

```
Training R^2: 0.9441759123453746
Testing R^2: 0.6030151291599787
Train Accuracy - : 0.944
Test Accuracy - : 0.603
```

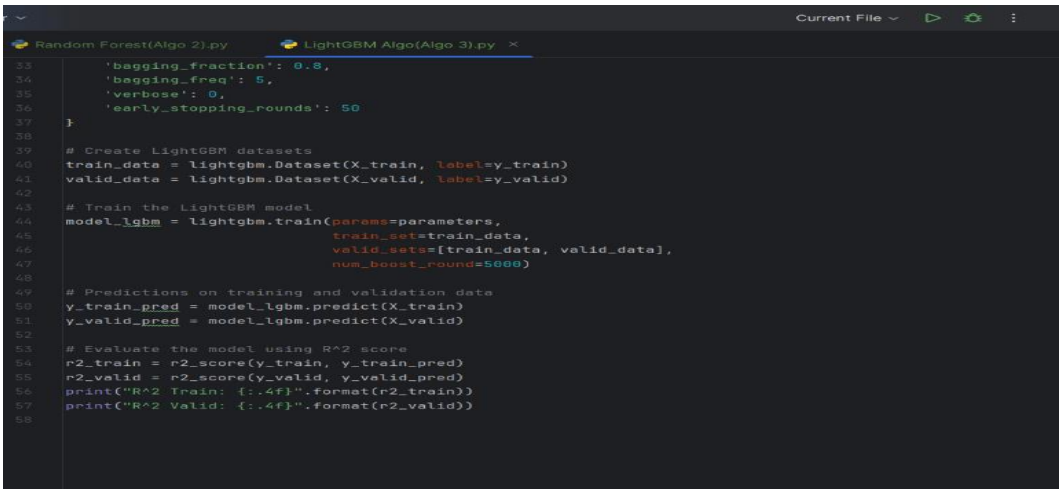
Figure 5.4.2.3

5.4.3 Light Gradient Boosting Model



```python
import pandas as pd
import lightgbm
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

# Load the data
df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")

# Specify the feature columns to use for prediction
feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'Total_Area', 'Price_per_SQFT']

# Convert 'Yes'/'No' in 'Balcony' column to 1/0
df_train['Balcony'] = df_train['Balcony'].map({'Yes': 1, 'No': 0})

# Drop rows with missing values in the selected feature columns
df_train.dropna(subset=feature_columns, inplace=True)

# Extract features and target variable
X = df_train[feature_columns]
y = df_train['Price']

# Split the data into training and validation sets
X_train, X_valid, y_train, y_valid = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)

# Define LightGBM parameters
parameters = {
    'boosting_type': 'gbdt',
    'objective': 'regression',
    'metric': 'l2',
    'num_leaves': 31,
    'learning_rate': 0.05,
    'feature_fraction': 0.9
```

Figure 5.4.3.1



```python
    'bagging_fraction': 0.8,
    'bagging_freq': 5,
    'verbose': 0,
    'early_stopping_rounds': 50
}

# Create LightGBM datasets
train_data = lightgbm.Dataset(X_train, label=y_train)
valid_data = lightgbm.Dataset(X_valid, label=y_valid)

# Train the LightGBM model
model_lgbm = lightgbm.train(params=parameters,
                            train_set=train_data,
                            valid_sets=[train_data, valid_data],
                            num_boost_round=5000)

# Predictions on training and validation data
y_train_pred = model_lgbm.predict(X_train)
y_valid_pred = model_lgbm.predict(X_valid)

# Evaluate the model using R^2 score
r2_train = r2_score(y_train, y_train_pred)
r2_valid = r2_score(y_valid, y_valid_pred)
print("R^2 Train: {:.4f}".format(r2_train))
print("R^2 Valid: {:.4f}".format(r2_valid))
```

Figure 5.4.3.2

```
"C:\data science Internshi
R^2 Train: 0.9992
R^2 Valid: 0.9990
```

Figure 5.4.3.3

5.4.4 Linear Regression

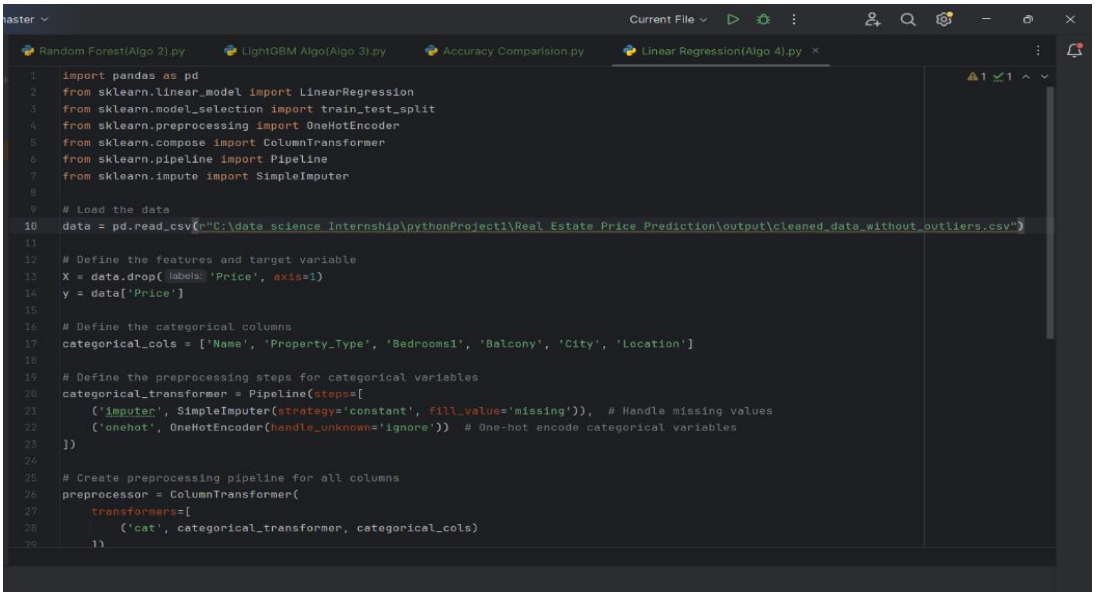


Figure 5.4.4.1

```python
# Create preprocessing pipeline for all columns
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', categorical_transformer, categorical_cols)
    ])

# Append the preprocessor to the estimator
linear_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                  ('classifier', LinearRegression())])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.4, random_state=42)

# Fit the model
print(linear_pipeline.fit(X_train, y_train))

# Evaluate the model
print("Training R^2:", linear_pipeline.score(X_train, y_train))
print("Testing R^2:", linear_pipeline.score(X_test, y_test))

print(f'Train Accuracy - : {linear_pipeline.score(X_train, y_train):.3f}')
print(f'Test Accuracy - : {linear_pipeline.score(X_test, y_test):.3f}')
```

Figure 5.4.4.2

```
Training R^2: 0.9338286255782781
Testing R^2: 0.38710968837148185
Train Accuracy - : 0.934
Test Accuracy - : 0.387
```

Figure 5.4.4.3

5.4.5 Naïve Base

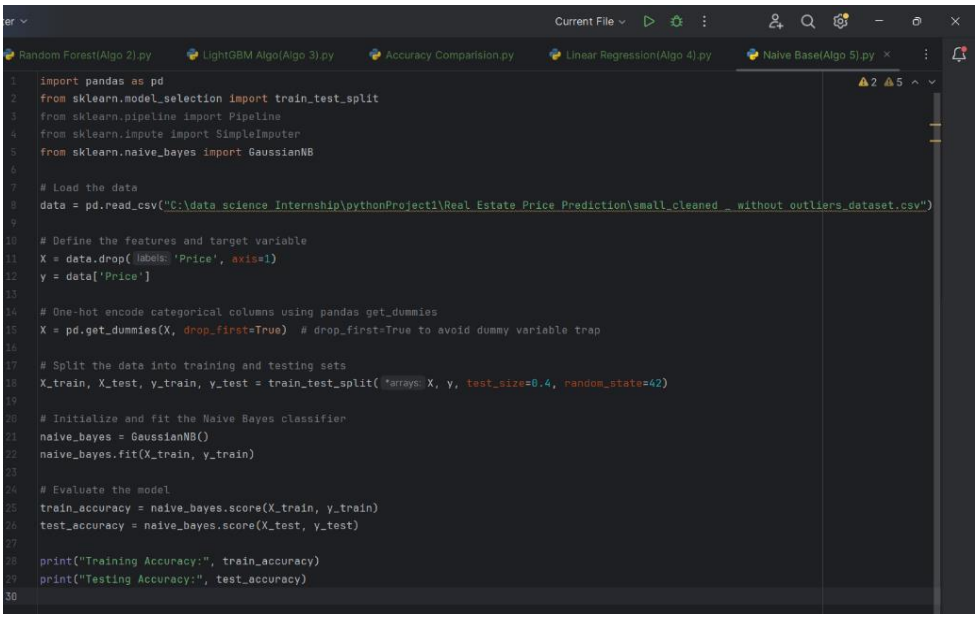


Figure 5.4.5.1

```
Training Accuracy: 0.8511582060128142
Testing Accuracy: 0.05617147080561715

Process finished with exit code 0
```

Figure 5.4.5.2

5.5 ACCURACY COMPARISON

A key component of this evaluation process is accuracy comparison, which makes it possible to compare the predictive capacities of various models. I utilized matplotlib to create a bar chart, enabling a comparative analysis of the accuracy across multiple algorithms. Through this examination, it was discerned that the decision tree algorithm yielded the most favorable outcomes..

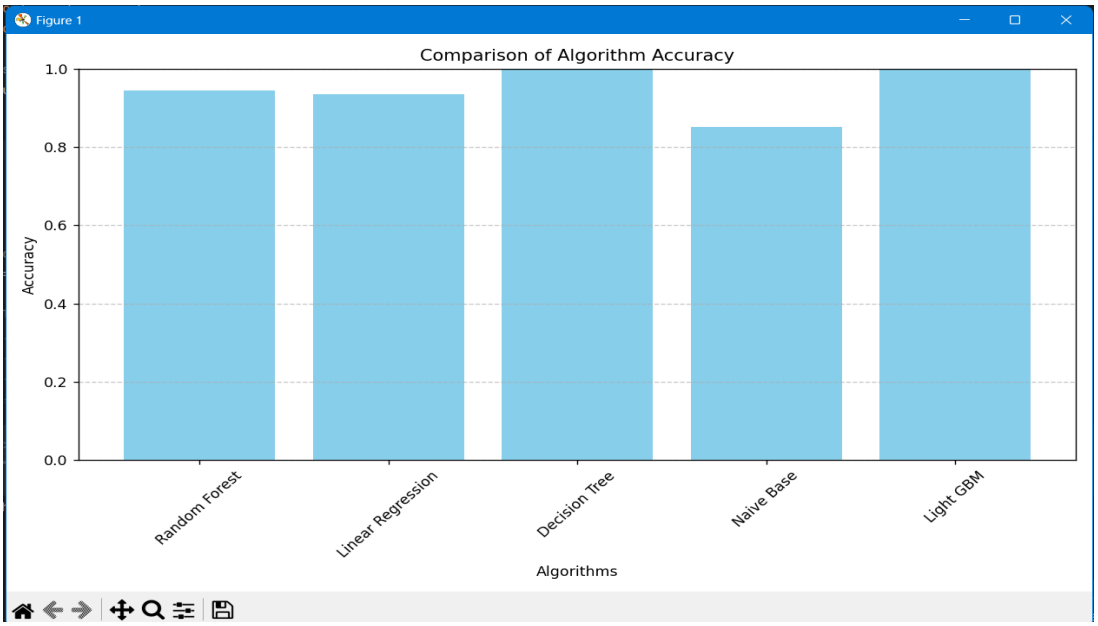
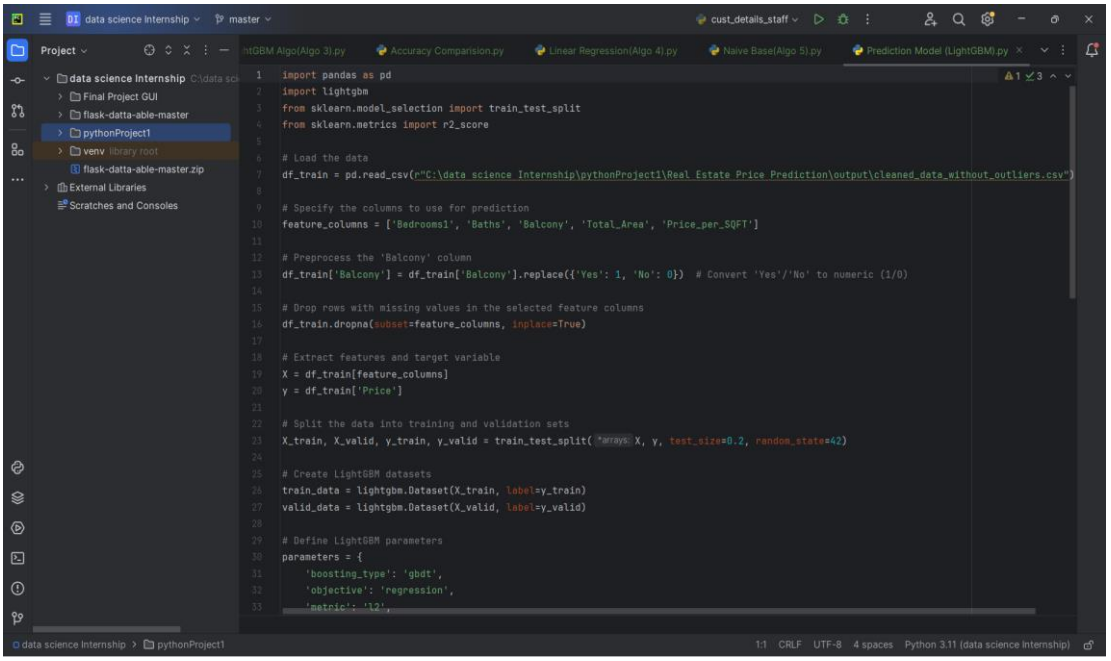


Figure 5.5.1

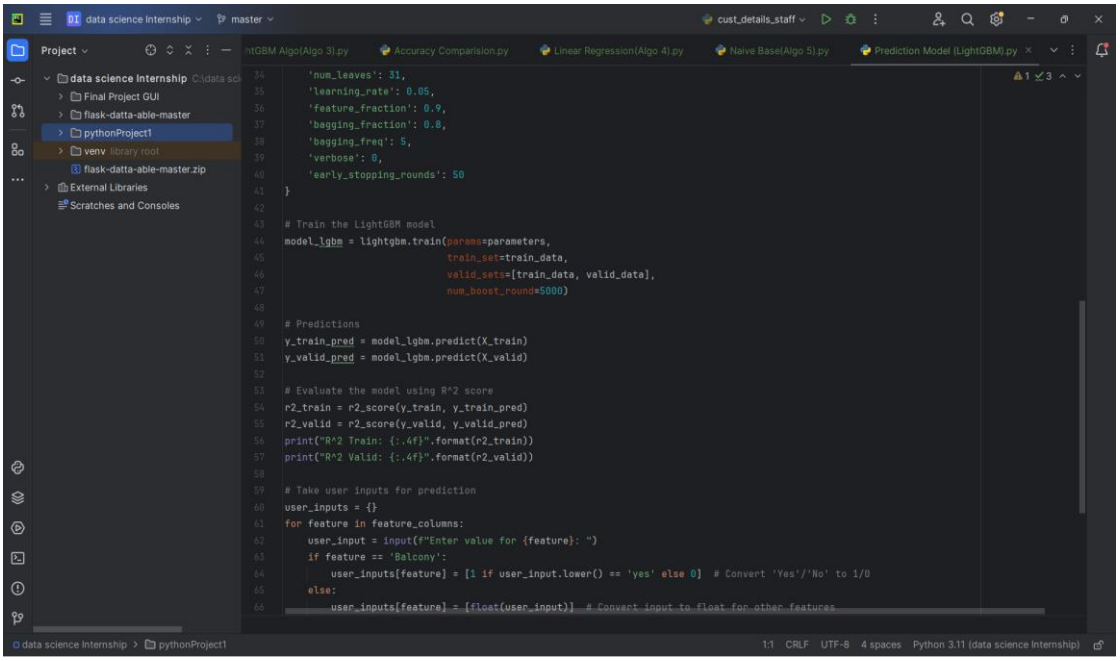
5.6 CREATING REAL ESTATE PRICE PREDICTION MODEL

In the real estate sector, a price prediction model is a potent instrument that provides insights into the dynamic swings in property values. Since a model is required to accomplish the aim, I chose one that matches my dataset flawlessly. I was able to develop a price prediction model with it and get an accurate estimation of the chosen model.



```
import pandas as pd
import lightgbm
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

# Load the data
df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned_data_without_outliers.csv")

# Specify the columns to use for prediction
feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'Total_Area', 'Price_per_SQFT']

# Preprocess the 'Balcony' column
df_train['Balcony'] = df_train['Balcony'].replace({'Yes': 1, 'No': 0})  # Convert 'Yes'/'No' to numeric (1/0)

# Drop rows with missing values in the selected feature columns
df_train.dropna(subset=feature_columns, inplace=True)

# Extract features and target variable
X = df_train[feature_columns]
y = df_train['Price']

# Split the data into training and validation sets
X_train, X_valid, y_train, y_valid = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)

# Create LightGBM datasets
train_data = lightgbm.Dataset(X_train, label=y_train)
valid_data = lightgbm.Dataset(X_valid, label=y_valid)

# Define LightGBM parameters
parameters = {
    'boosting_type': 'gbdt',
    'objective': 'regression',
    'metric': 'l2',
```

Figure 5.6.1



```
    'num_leaves': 31,
    'learning_rate': 0.05,
    'feature_fraction': 0.9,
    'bagging_fraction': 0.8,
    'bagging_freq': 5,
    'verbose': 0,
    'early_stopping_rounds': 50
}

# Train the LightGBM model
model_lgbm = lightgbm.train(params=parameters,
                            train_set=train_data,
                            valid_sets=[train_data, valid_data],
                            num_boost_round=5000)

# Predictions
y_train_pred = model_lgbm.predict(X_train)
y_valid_pred = model_lgbm.predict(X_valid)

# Evaluate the model using R^2 score
r2_train = r2_score(y_train, y_train_pred)
r2_valid = r2_score(y_valid, y_valid_pred)
print("R^2 Train: {:.4f}".format(r2_train))
print("R^2 Valid: {:.4f}".format(r2_valid))

# Take user inputs for prediction
user_inputs = {}
for feature in feature_columns:
    user_input = input(f"Enter value for {feature}: ")
    if feature == 'Balcony':
        user_inputs[feature] = [1 if user_input.lower() == 'yes' else 0]  # Convert 'Yes'/'No' to 1/0
    else:
        user_inputs[feature] = [float(user_input)]  # Convert input to float for other features
```

Figure 5.6.2

```
# Create DataFrame from user inputs
user_df = pd.DataFrame(user_inputs)

# Make predictions on user input data
predicted_price = model_lgbm.predict(user_df)

# Print the predicted price
print("Predicted Price:", predicted_price[0])
```

Figure 5.6.3

CHAPTER 6: IMPLIMENTATION

6.1 SETTING-UP FLASK

For my application, I have developed a graphical user interface using the Flask framework. Setting up the file structure was the first step, and I utilized it to make a fantastic file structure for my entire application. I created using Flask Framework.

I made two distinct directories in the file structure. one is called template, and inside the template folder, I've made a ton of HTML files to make a beautiful user interface. In the static folder, I've added a variety of javascript and bootstrap files to enhance the online program.

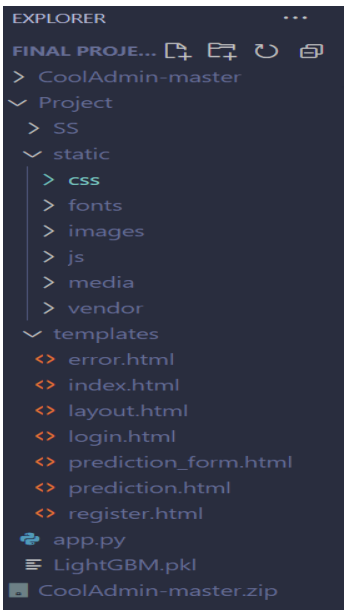


Figure 6.1.1

6.2 REGISTRATION

To register any user on the local server, I have made a registration form. To do this, I have added registration.html to the template dictionary and appended this HTML file with the app.py application file.

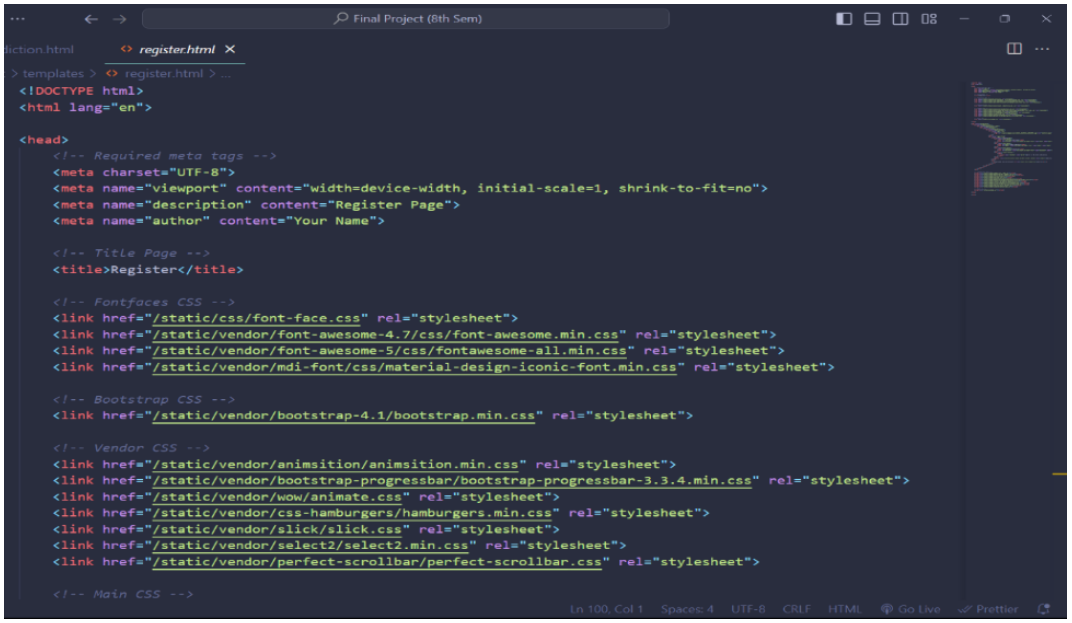


Figure 6.2.1

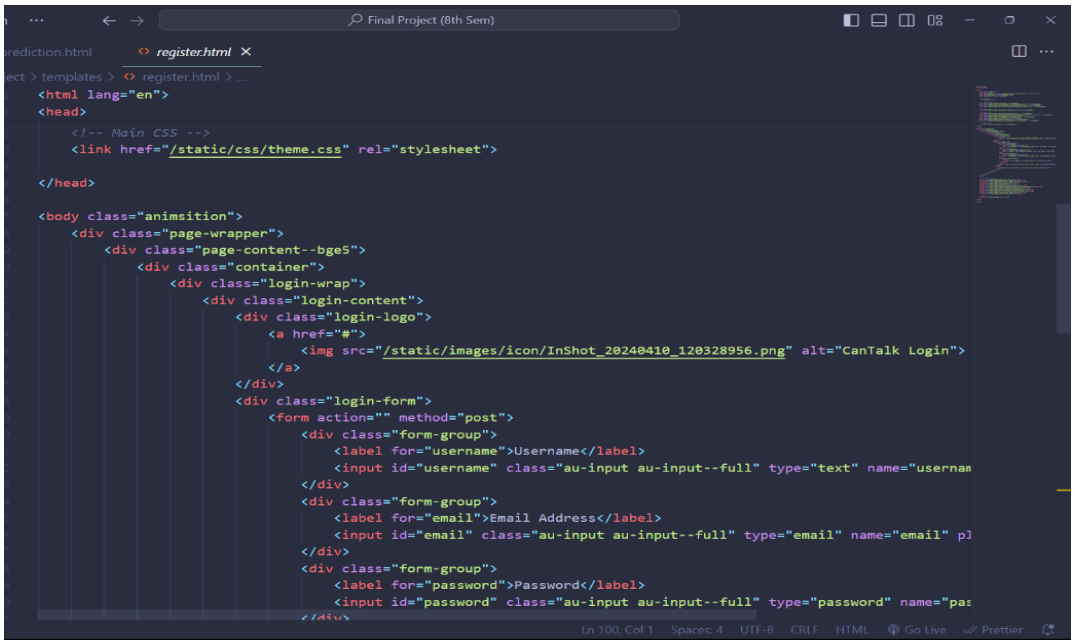


Figure 6.2.2

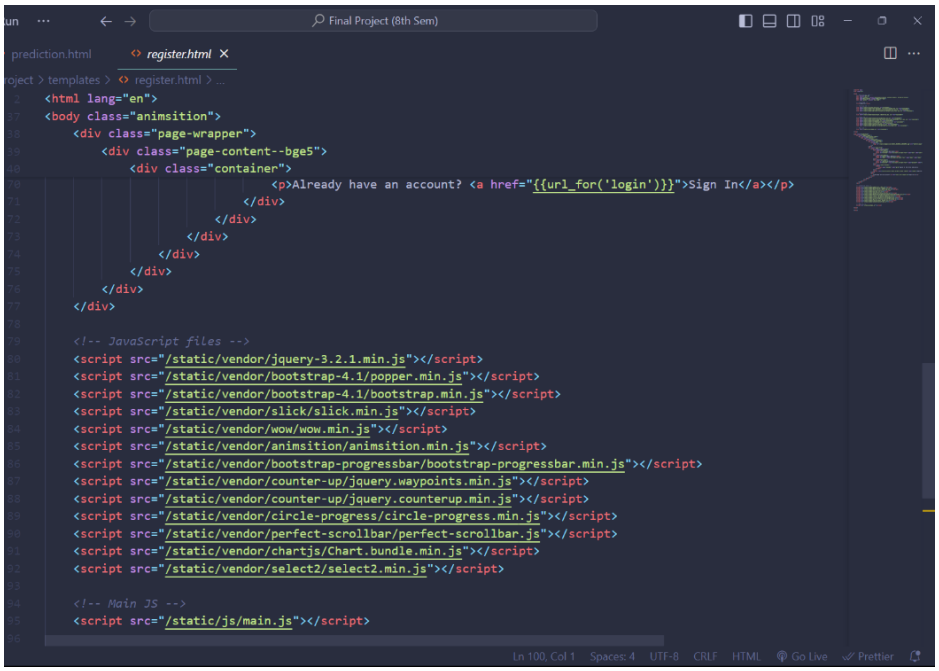


Figure 6.2.3

6.3 LOGIN

The purpose of the login form is to allow any existing user to log in to the program. To that end, I have connected the login form to the local server, allowing it to retrieve and validate user data.

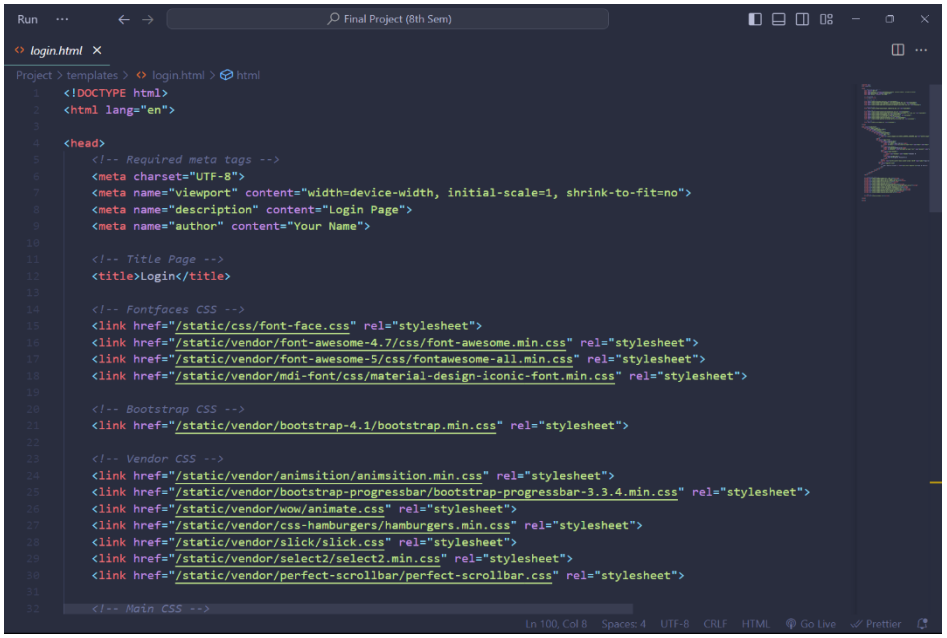


Figure 6.3.1

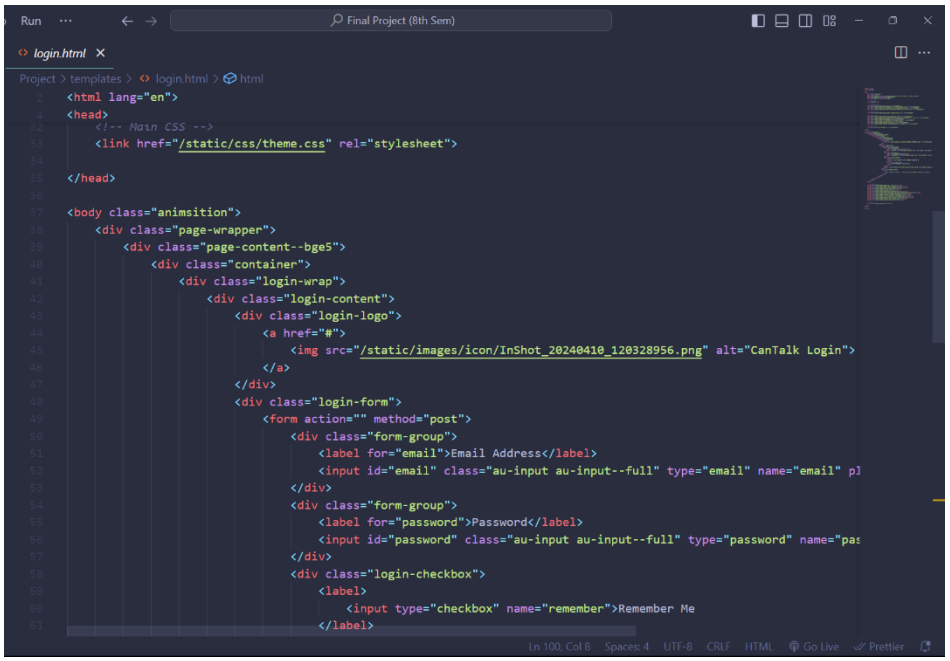


Figure 6.3.2

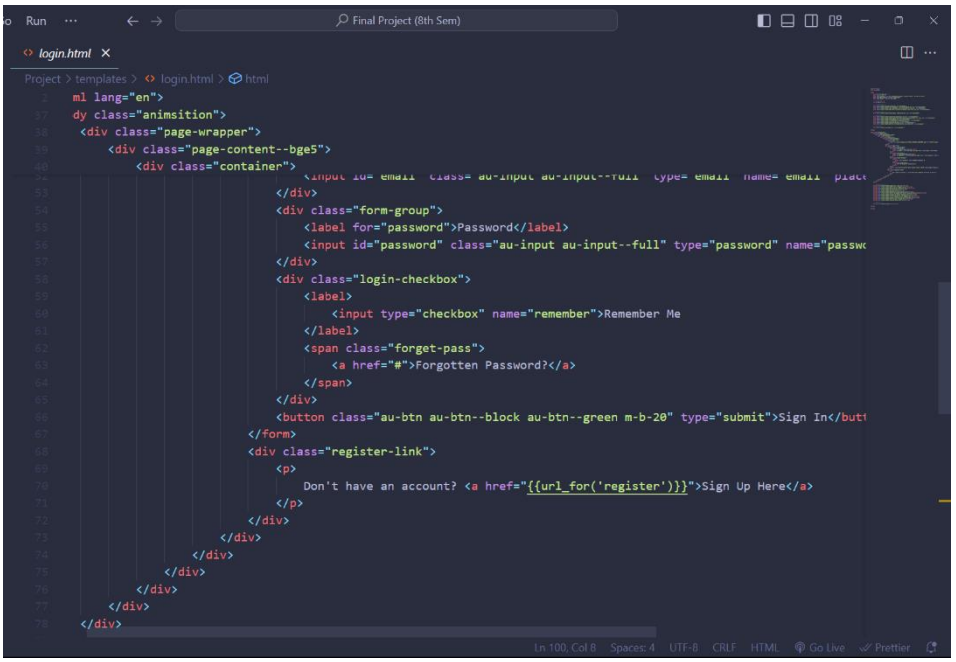


Figure 6.3.3

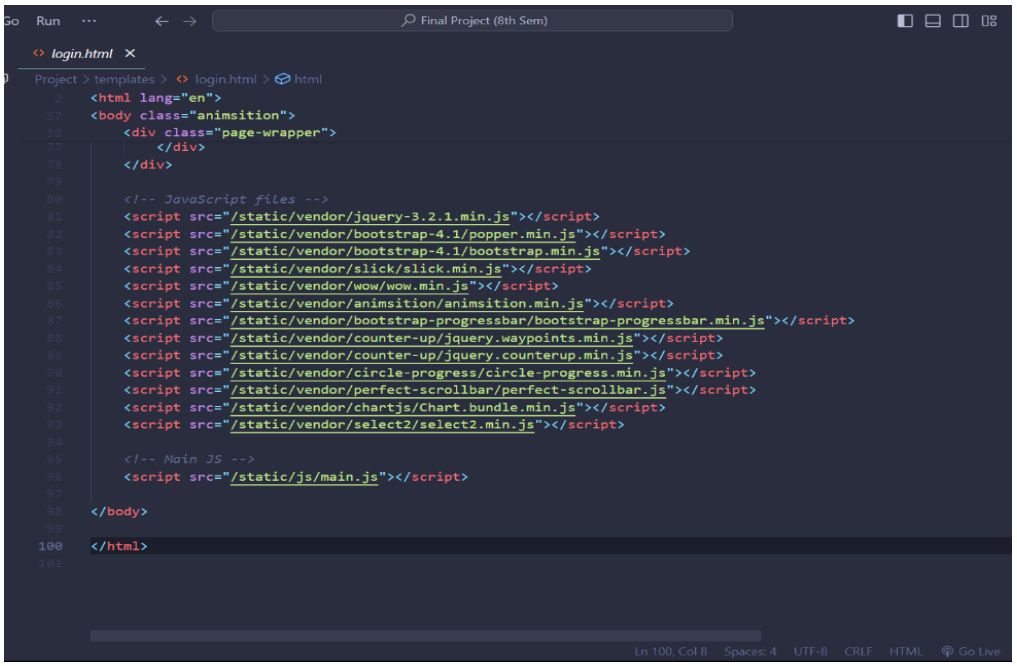


Figure 6.3.4

6.4 PIKAL FILE

The pickle file, which contains all of the program's logic and allows it to run the prediction model, is the heart and soul of this application.

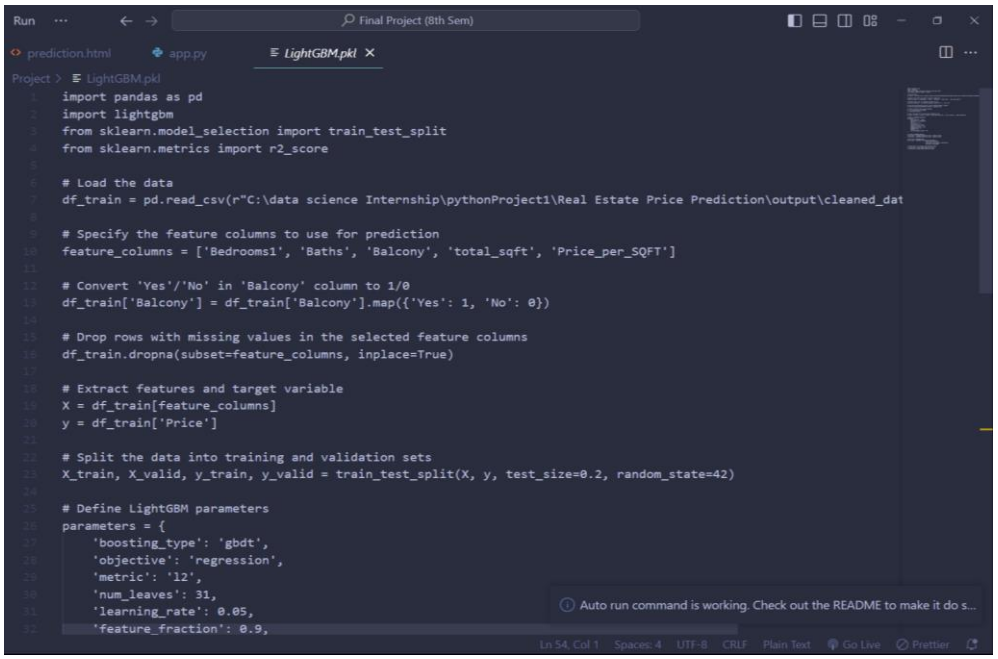


Figure 6.4.1

```
    parameters = {
        'learning_rate': 0.05,
        'feature_fraction': 0.9,
        'bagging_fraction': 0.8,
        'bagging_freq': 5,
        'verbose': 0,
        'early_stopping_rounds': 50
    }

    # Create LightGBM datasets
    train_data = lightgbm.Dataset(X_train, label=y_train)
    valid_data = lightgbm.Dataset(X_valid, label=y_valid)

    # Train the LightGBM model
    model_lgbm = lightgbm.train(params=parameters,
                                train_set=train_data,
                                valid_sets=[train_data, valid_data],
                                num_boost_round=5000)

    # Predictions on training and validation data
    y_train_pred = model_lgbm.predict(X_train)
    y_valid_pred = model_lgbm.predict(X_valid)
```

Figure 6.4.2

6.5 INDEX FILE

The index file is used to retrieve user-provided values. It has a form where the user can enter various user values for the type of property, such as price per square foot, number of rooms, number of bathrooms, and total square feet.

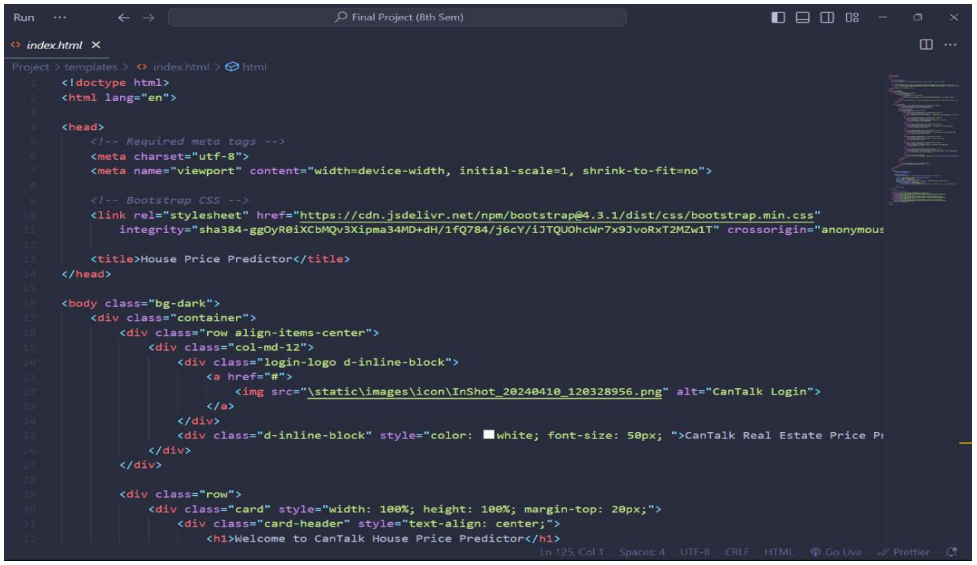


Figure 6.5.1

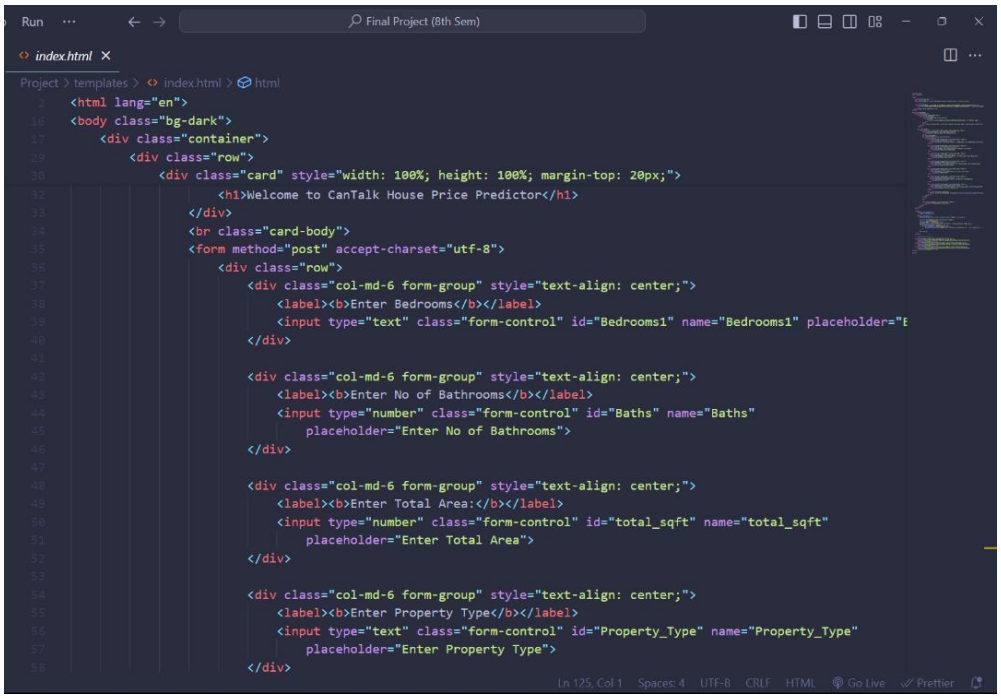


Figure 6.5.2

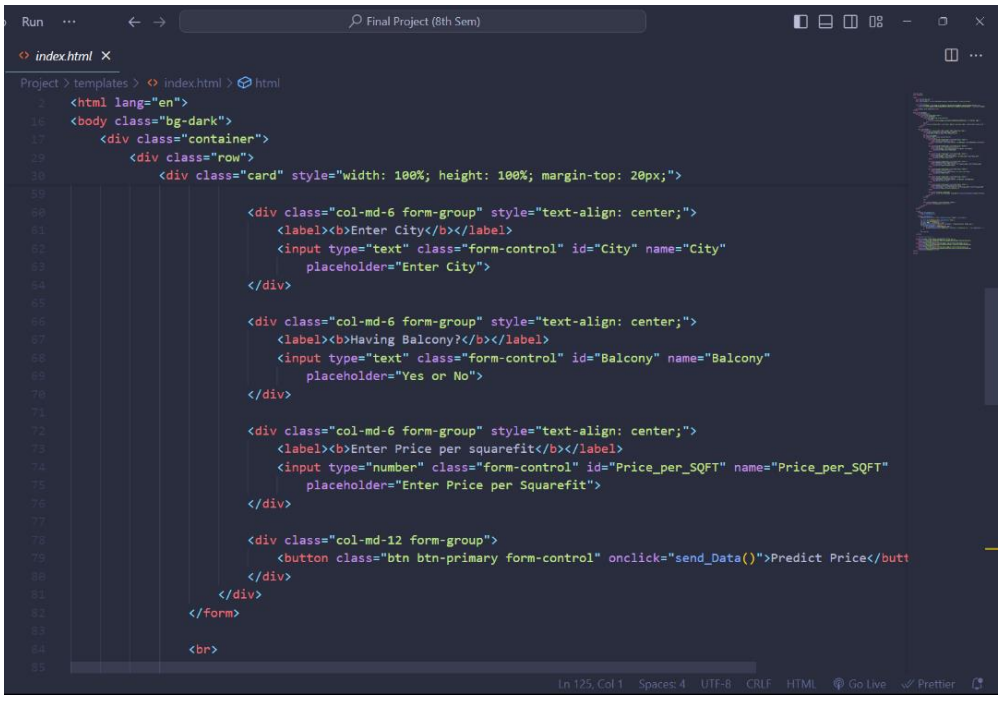


Figure 6.5.3

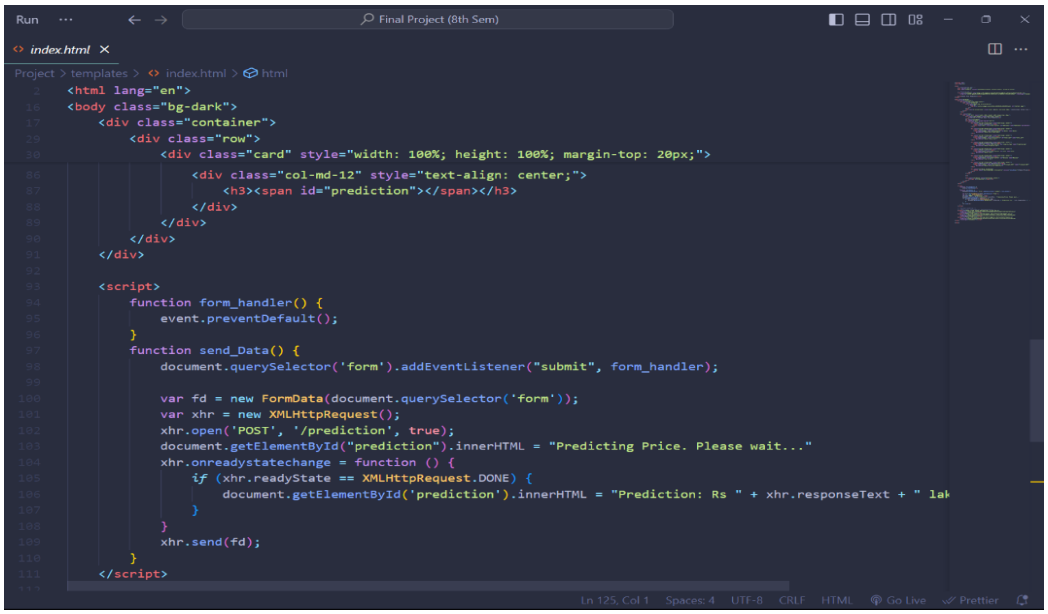


Figure 6.5.4

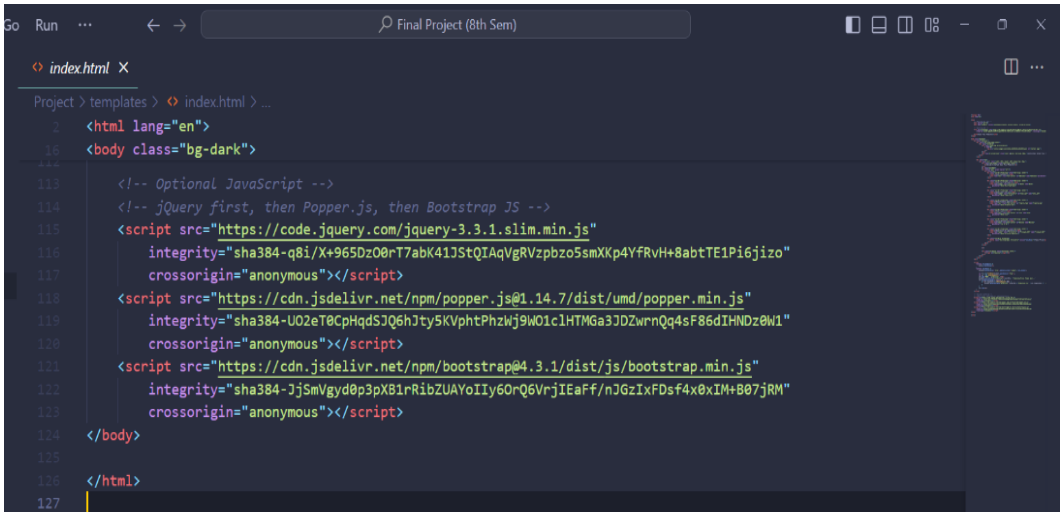


Figure 6.5.5

6.6 APP.PY

The main file that links all of the application's connections in the project is called app.py. It was made with the help of several libraries, including flask, mysqldb, pandas, scikit-

learn, light gbm, logging, and OS, among others. It loads the model, pickle file, and other HTML files that are contained in the template directory.

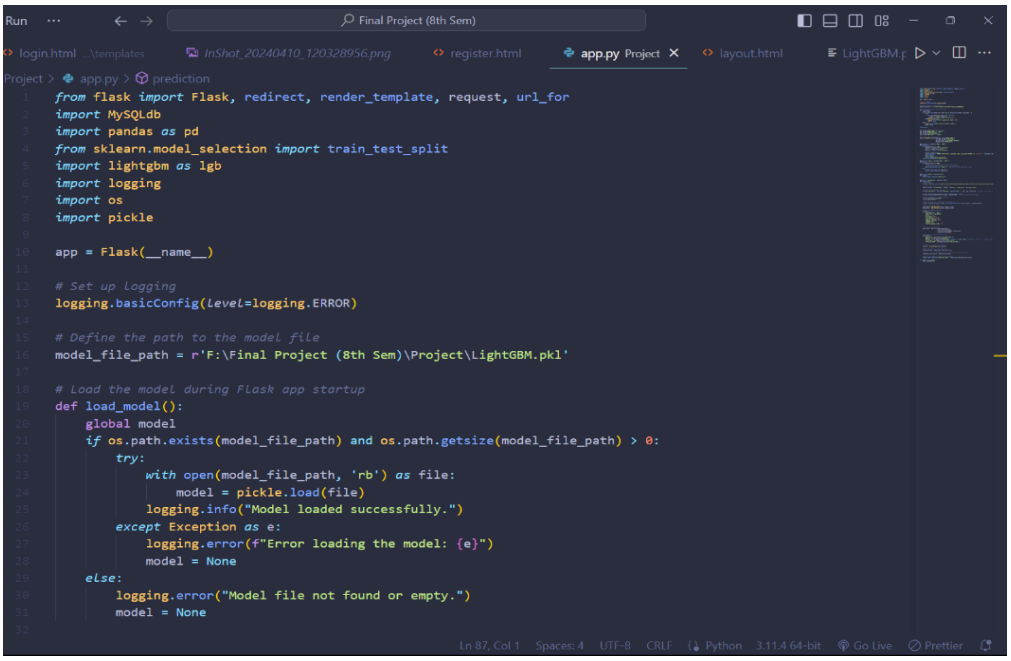


Figure 6.6.1

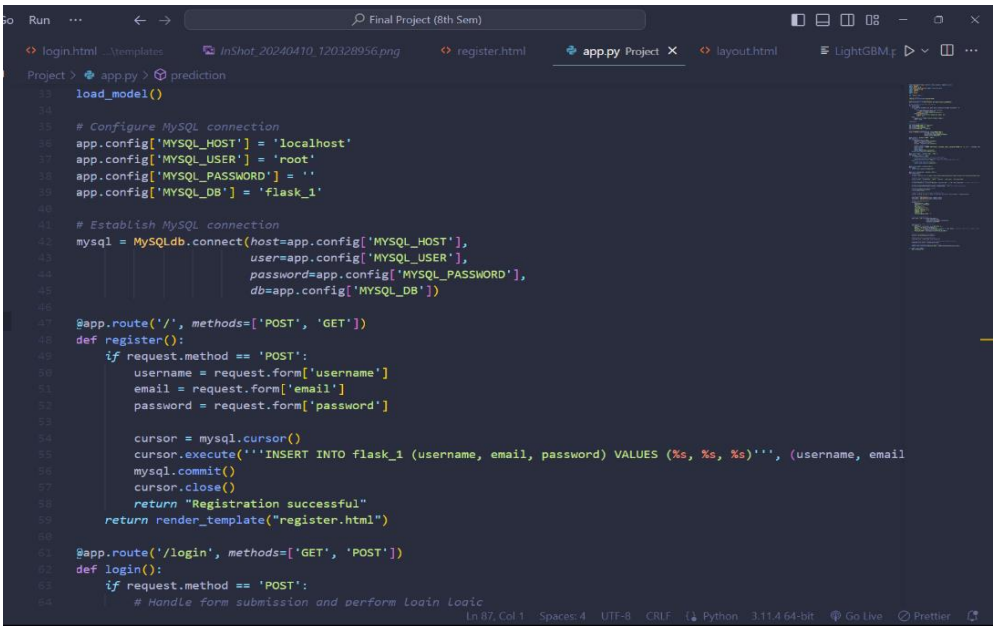
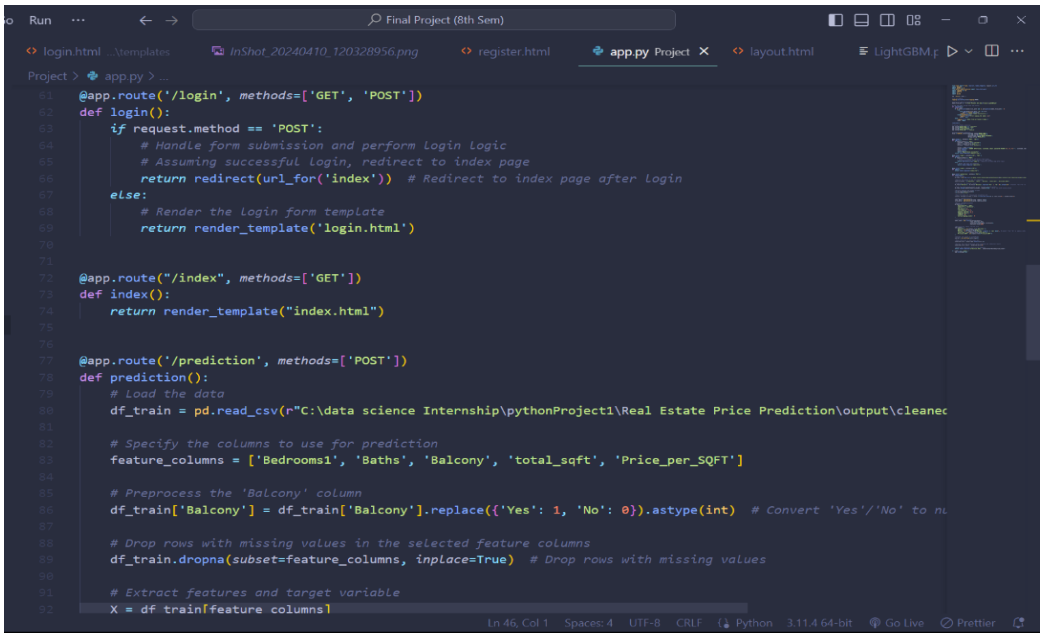
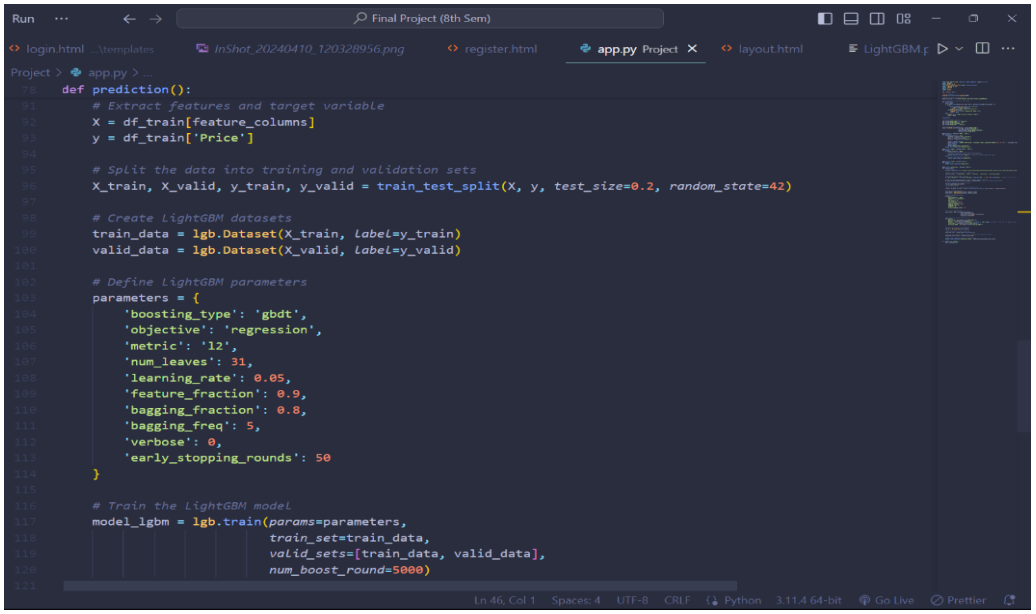


Figure 6.6.2



```python
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        # Handle form submission and perform login logic
        # Assuming successful login, redirect to index page
        return redirect(url_for('index'))  # Redirect to index page after login
    else:
        # Render the login form template
        return render_template('login.html')


@app.route("/index", methods=['GET'])
def index():
    return render_template("index.html")


@app.route('/prediction', methods=['POST'])
def prediction():
    # Load the data
    df_train = pd.read_csv(r"C:\data science Internship\pythonProject1\Real Estate Price Prediction\output\cleaned

    # Specify the columns to use for prediction
    feature_columns = ['Bedrooms1', 'Baths', 'Balcony', 'total_sqft', 'Price_per_SQFT']

    # Preprocess the 'Balcony' column
    df_train['Balcony'] = df_train['Balcony'].replace({'Yes': 1, 'No': 0}).astype(int)  # Convert 'Yes'/'No' to nu

    # Drop rows with missing values in the selected feature columns
    df_train.dropna(subset=feature_columns, inplace=True)  # Drop rows with missing values

    # Extract features and target variable
    X = df_train[feature_columns]
```

Figure 6.6.3



```python
def prediction():
    # Extract features and target variable
    X = df_train[feature_columns]
    y = df_train['Price']

    # Split the data into training and validation sets
    X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2, random_state=42)

    # Create LightGBM datasets
    train_data = lgb.Dataset(X_train, label=y_train)
    valid_data = lgb.Dataset(X_valid, label=y_valid)

    # Define LightGBM parameters
    parameters = {
        'boosting_type': 'gbdt',
        'objective': 'regression',
        'metric': 'l2',
        'num_leaves': 31,
        'learning_rate': 0.05,
        'feature_fraction': 0.9,
        'bagging_fraction': 0.8,
        'bagging_freq': 5,
        'verbose': 0,
        'early_stopping_rounds': 50
    }

    # Train the LightGBM model
    model_lgbm = lgb.train(params=parameters,
                           train_set=train_data,
                           valid_sets=[train_data, valid_data],
                           num_boost_round=5000)
```

Figure 6.6.4

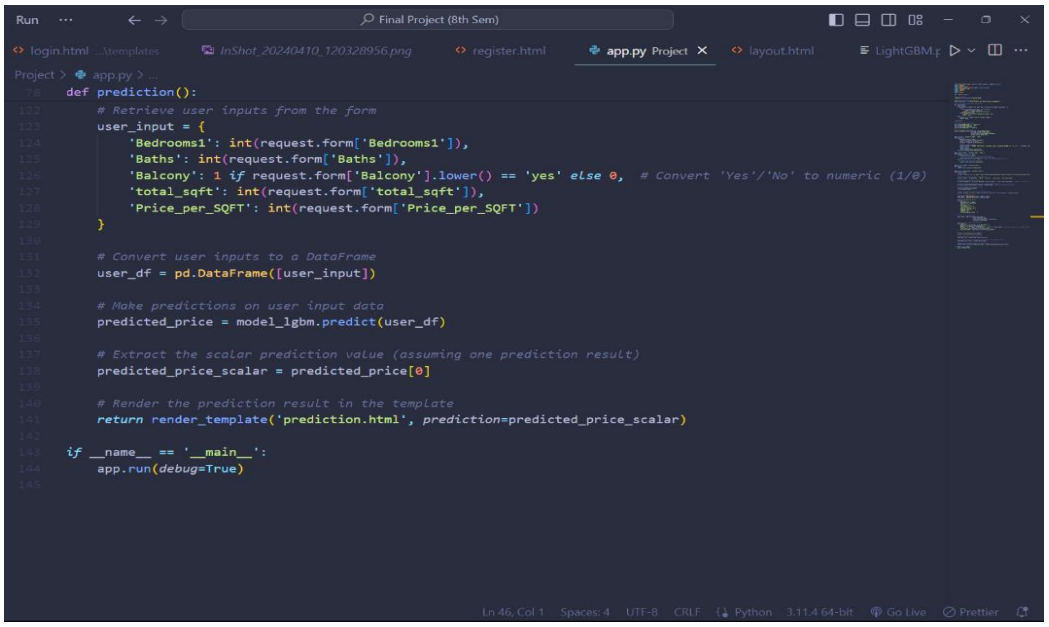


Figure 6.6.5

6.7 GUI

The graphical user interface (GUI) is what the user can see and utilize. When utilizing this form, the user can first view the registration form as shown here. If the user is already registered, he can proceed straight to the login form by clicking the sign in button at the bottom of the registration form. Otherwise, the user can register himself to the local server.

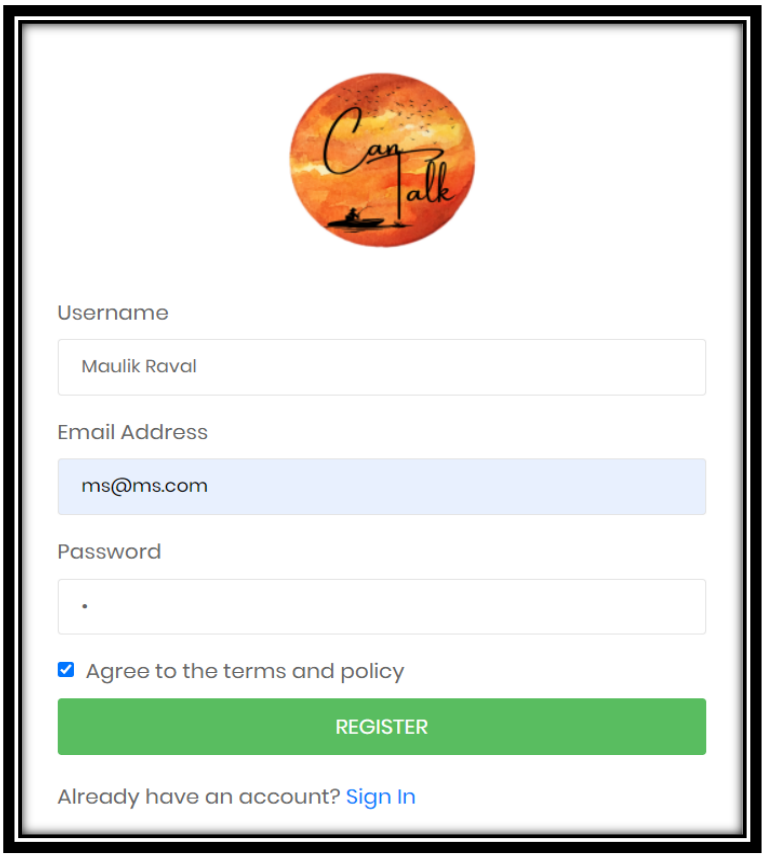


Figure 6.7.1

As soon as the user enters their login, email address, and password, the information is saved in the localhost database, as seen below.

	ID	username	email	password
Edit Copy Delete	1	msr	ms@ms.com	1
Edit Copy Delete	2	cantor	cantor@cantalk.ca	123
Edit Copy Delete	3	maulik.raval	maulik.raval@mail.ca	121
Edit Copy Delete	4	maulik.raval	maulik.raval@mail.ca	121
Edit Copy Delete	5	bits	campus@bits.in	098

Figure 6.7.2

The next time they visit, the existing user will be taken straight to the login page, where they will need to enter their email address and password to log in. The application will compare the user's information entered by the user with that of the local host database.

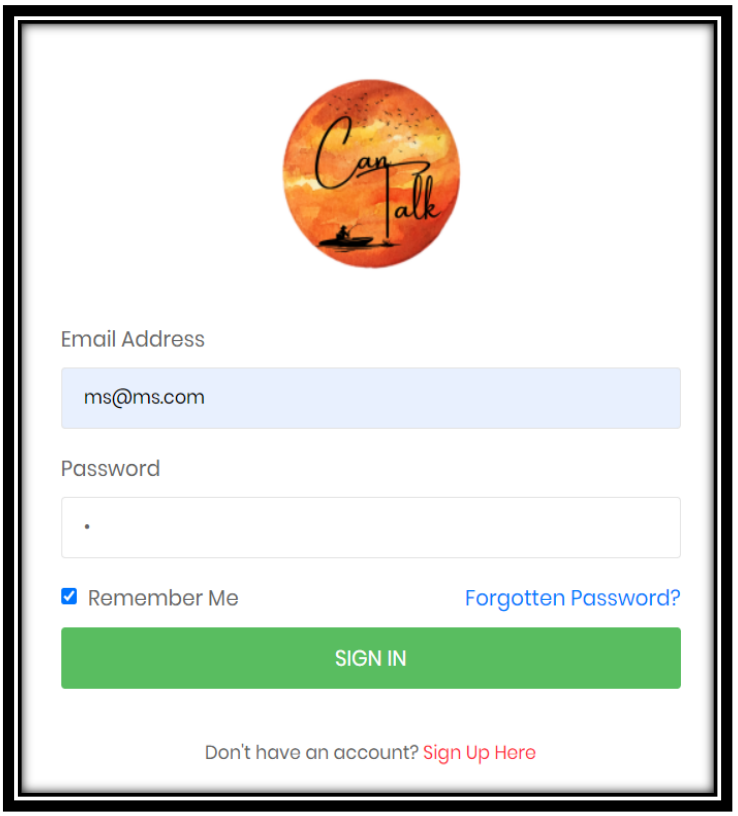


Figure 6.7.3

The user will engage with the Cantalk real estate price prediction system after their login credentials have been verified. Here, they will be able to submit various details, including the number of bedrooms, baths, and total space of the property, as well as whether or not it has a balcony and how much it will cost per square foot.

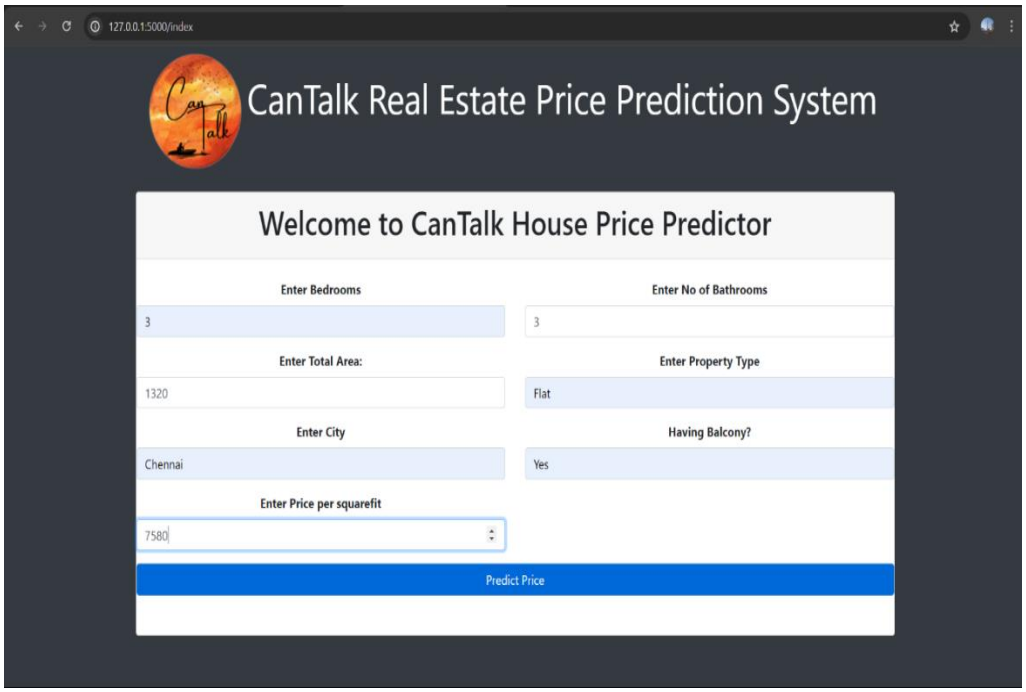


Figure 6.7.4

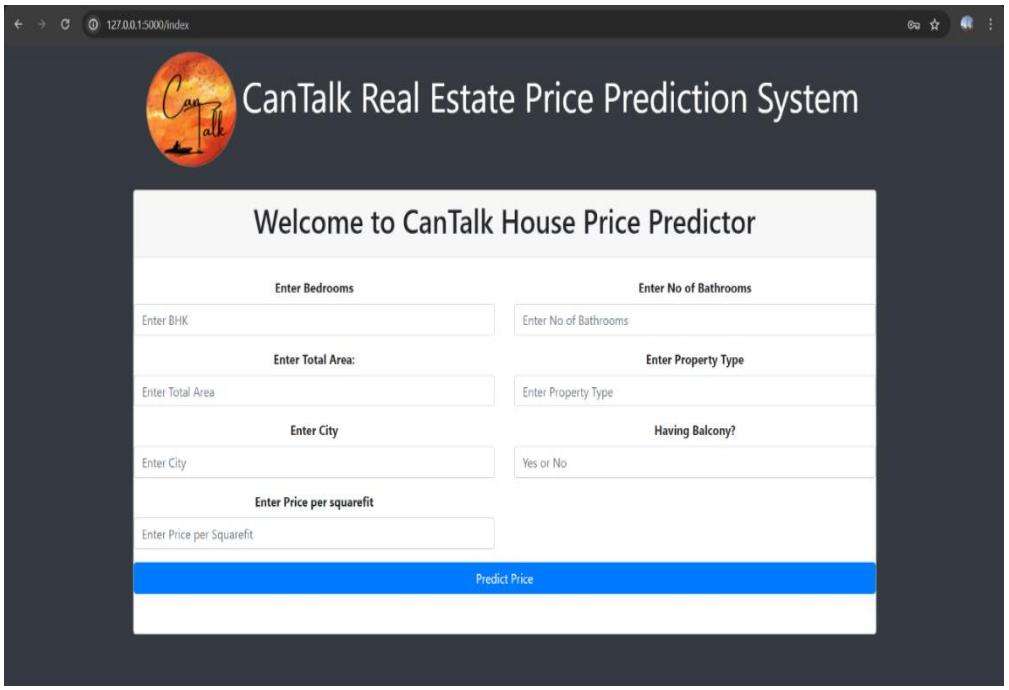


Figure 6.7.5

Figure 6.7.6

Figure 6.7.7

CHEPTER 7: CONCLUSION

In wrapping up my price prediction project, it's evident that it holds considerable promise for future forecasting endeavours. By harnessing the power of advanced algorithms and meticulously scrutinizing historical data, I've successfully engineered a robust model adept at accurately delineating price trends. The culmination of my research not only yields actionable insights but also furnishes valuable resources for companies and investors navigating the complexities of market behaviour.

This venture represents more than just a static snapshot; it's a dynamic journey marked by ongoing refinement and evolution. As I continue to fine-tune my methodologies and incorporate fresh streams of data, the efficacy of my predictive capabilities will undoubtedly ascend. This iterative process not only fortifies the reliability of my model but also augments the discernment it provides to stakeholders seeking to make informed decisions in the financial landscape.

Moreover, the implications of this endeavour extend far beyond the confines of my immediate research. The insights garnered from this project hold the potential to catalyse paradigm shifts in how we approach market analysis and decision-making. By leveraging the symbiotic relationship between cutting-edge technology and historical market data, we can forge a path toward a more nuanced understanding of market dynamics.

In essence, my price prediction project encapsulates the fusion of data-driven ingenuity and strategic foresight. It serves as a beacon illuminating the possibilities that lie ahead in the realm of financial forecasting. As I embark on the next phase of this journey, I remain steadfast in my commitment to pushing the boundaries of predictive analytics and empowering stakeholders with the knowledge they need to navigate the ever-changing tides of the market.

CHAPTER 8: REFERENCES

https://www.kaggle.com

www.github.com

https://seaborn.pydata.org

https://www.python.org

https://www.w3schools.com

https://stackoverflow.com