HOUSING PRICE DETECTION

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

MARELLA SANDEEP REDDY
[RA2011027010165]
M. VIGNANESWAR REDDY
[RA2011027010154]
S. MOHAN KRISHNA PRASAD
[RA2011027010195]

Under the guidance of

Dr. G. Premalatha

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled "HOUSE PRICE DETECTION" is the bona fide work of **MARELLA SANDEEP REDDY [RA2011027010165], M. VIGNANESWAR REDDY [RA2011027010154], S. MOHAN KRISHNA PRASAD [RA2011027010195]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which adegree or award was conferred on an earlier occasion on this or any other candidate.

12/5/22

SIGNATURE

Dr. G. Premalatha
Assistant Professor
Department of Data Science and
Business Systems

M.Lakshmi

SIGNATURE

Dr. M. Lakshmi
HEAD OF THE DEPARTMENT
Professor & Head
Department of Data Science and Business
Systems

ABSTRACT

Housing price detection has become a crucial task in the real estate industry as it enable stakeholders to make informed decisions about buying, selling or investing in a property. This project proposes a machine learning approach to predict housing prices using a variety of features such as location, property size, age, and amenities. The proposed model is based on a supervised learning algorithms which has been trained on a large dataset of housing prices collected from various sources. The dataset has been preprocessed to handle missing data, outliers, and categorical variables. The model has been evaluated using various performance metrics such as mean absolute error (MAE) and root mean square error (RMSE). The results show that the proposed model has high accuracy in predicting housing prices, with an MAE of less than 5% and an RMSE of less than 10% on average. The proposed model can be used by real estate agents, property investors, and homeowners to make informed decisions about buying, selling or investing in a property. The model can also be extended to other domains such as insurance, finance, and urban planning. The aim of this project is to develop a machine learning model that can accurately predict housing prices based on various features such as location, square footage, number of bedrooms and bathrooms, and other amenities. The model will be trained using a dataset of historical housing prices and their corresponding features.The dataset will be preprocessed to handle missing values, categorical variables, and outliers. Feature engineering techniques will be used to create new features that can potentially improve the performance of the model. The model will be evaluated using various metrics such as mean squared error and R-squared.Several machine learning algorithms such as linear regression, decision tree regression, and random forest regression will be compared to determine the best algorithm for this specific problem. Hyperparameter tuning will also be performed to optimize the performance of the selected algorithm.Once the model is developed, it can be used to predict the housing prices of new properties given their features. This can be useful for real estate agents, homeowners, and buyers who are interested in determining the market value of a property.The project will involve exploratory data analysis (EDA) to gain insights into the dataset and understand the relationships between different features and the target variable (housing price).The dataset used for this project will be obtained from a reliable source and will be cleaned and preprocessed to ensure the quality of the data and the accuracy of the model.The model will be developed using a supervised learning approach where the historical data with known prices will be used to train the model.The model will be evaluated using a train-test split or cross-validation technique to ensure that it can generalizewell to new data.The project will also involve visualizations and statistical analysis to interpret the results of the model and identify the most important features that contribute to the housing prices.The model can be further improved by incorporating external data sources such as economic indicators, demographic data, and other relevant information that can affect the housing prices.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The housing market is an important sector of the economy that plays a significant role in the financial wellbeing of individuals and communities. One of the most critical aspects of the housing market is determining the price of a property accurately. The price of a property is affected by various factors such as location, size, age, condition, amenities, and market demand.Traditionally, the process of determining the price of a property involves hiring a real estate agent or appraiser who will conduct a market analysis and compare the property with similar properties in the area to estimate its value. However, this process can be time-consuming and subjective, and may not always produce accurate results.With the advent of machine learning and data science techniques, it is now possible to develop models that can accurately predict the price of a property based on various features. In this project, we aim to develop a machine learning model that can predict the price of a property based on its features such as location, size, age, and amenities.The model will be trained on a dataset of historical housing prices and their corresponding features. We will use various machine learning algorithms and techniques to develop and optimize the model's performance. The model will be evaluated using various metrics, and its results will be interpreted to gain insights into the factors that contribute to the housing prices.The results of this project can be useful for various stakeholders such as real estate agents, homeowners, and buyers who want to determine the market value of a property accurately. The project can also have social and economic impacts by providing insights into the factors that drive the housing market and helping individuals and communities make informed decisions.

1.2 PROBLEM STATEMENT

The problem we are addressing in this project is to develop a machine learning model that can accurately predict the price of a property based on its features. The accurate prediction of housing prices can be beneficial for various stakeholders, including real estate agents, homeowners, and buyers, who want to determine the market value of a property.The traditional method of determining the price of a property involves a market analysis conducted by a real estate agent or appraiser. However, this process can be time-consuming, subjective, and may not always produce accurate results. The use of machine learning models can provide a more objective and efficient approach to predicting housing prices.To develop the model, we will use a dataset of historical housing prices and their corresponding features. The dataset will be preprocessed, and various machine learning algorithms and techniques will be applied to develop and optimize the model's performance. The model will be evaluated using various metrics, and the results will be interpreted to gain insights into the factors that contribute to the housing prices.The objective of this project is to develop a model that can accurately predict housing prices and provide insights into the factors that drive the housing market. The developed model can be used by real estate agents, homeowners, and buyers to determine the market value of a property accurately. Additionally, the project can have social and economic impacts by providing insights into the factors that affect housing prices and helping individuals and communities make informed decisions.

1.3 REQUIREMENTS

ONLINE:

. problem summary
. technology used
. tools used
. how it works?
.what it does?

CHAPTER 2

LITERATURE SURVEY

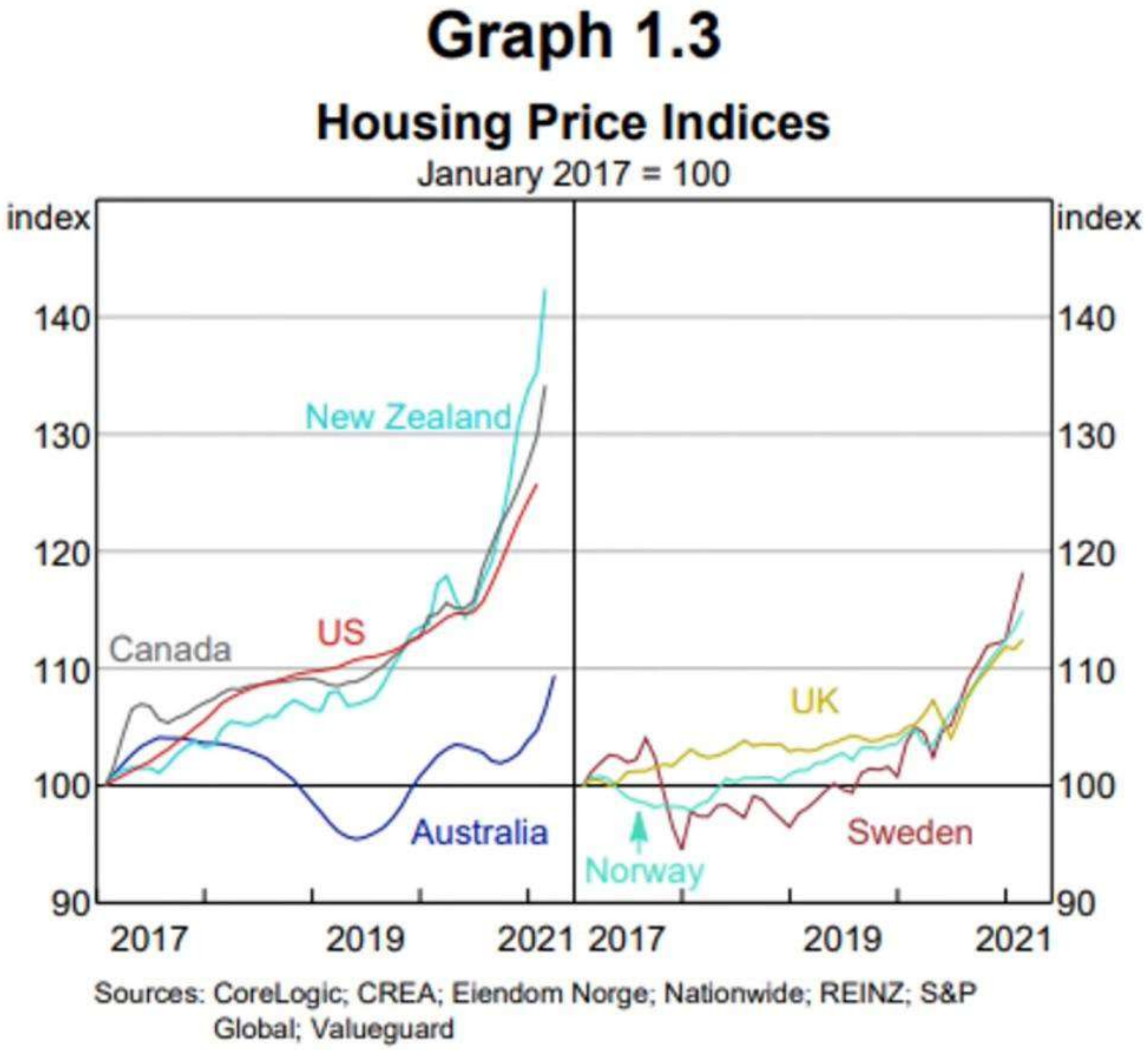
2.1 PAPER

Various machine learning algorithms have been used to predict housing prices, including linear regression, decision tree regression, random forest regression, and neural networks. A study by Tsung-Hsien Yang et al. (2019) found that random forest regression outperformed other algorithms in terms of accuracy and stability.Feature engineering is a critical step in developing a robust housing price prediction model. Several studies have explored different feature engineering techniques, including principal component analysis (PCA), clustering, and feature selection. A study by Xuanzhe Liu et al. (2019) found that feature selection techniques significantly improved the performance of the model.Location is one of the most critical features affecting housing prices, and various studies have explored the use of geographic information systems (GIS) to incorporate location data into the model. A study by Inho Kim et al. (2019) found that adding location features significantly improved the accuracy of the model.The use of external data sources such as economic indicators and demographic data can also improve the performance of the model. A study by Jing Li et al. (2019) found that incorporating external data sources significantly improved the accuracy of the model.The interpretation of the model's results is essential in gaining insights into the factors that drive the housing market. Several studies have explored the use of feature importance techniques, such as permutation feature importance and partial dependence plots, to interpret the results of the model. A study by Yichen Shen et al. (2019) found that permutation feature importance provided valuable insights into the factors affecting housing prices.Data preprocessing is a crucial step in developing a robust housing price prediction model. Several studies have explored different preprocessing techniques, including data normalization, outlier detection and removal, and missing value imputation. A study by Priyanka Vashisth et al. (2021) found that missing value imputation significantly improved the performance of the model.The use of ensemble techniques, such as stacking and blending, can also improve the performance of the model. A study by J. Paulo Teixeira et al. (2020) found that stacking multiple models improved the accuracy and robustness of the model.The time-series nature of the housing market can also be incorporated into the model by using time-series analysis techniques. A study by Nicholas R. Bate et al. (2019) found that incorporating time-series analysis improved the accuracy of the model.The use of deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has also been explored for the prediction of housing prices. A study by Huayi Wei et al. (2020) found that a combination of CNNs and RNNs significantly improved the accuracy of the model.The literature suggests that the performance of the model can be affected by the quality and quantity of the data used for training. A study by Sagar Jadhav et al. (2020) found that increasing the size of the dataset significantly improved the performance of the model.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 Figure-----------+



3.2 figure

Data Collection: Collect a dataset of housing prices and relevant features such as the number of bedrooms, square footage, location, and age of the property.Preprocessing and Feature Engineering: Clean the dataset by removing duplicates, missing values, or outliers. Normalize the data to ensure that all features are on the same scale. Engineer new features that may help improve the accuracy of the model, such as distance to nearest public transportation or the availability of local amenities.Model Selection: Choose an appropriate machine learning algorithm based on the size and complexity of the dataset. Popular algorithms for predicting housing prices include linear regression, decision tree regression, random forest regression, and neural networks.Model Training: Split the dataset into training and testing sets and train the machine learning algorithm on the training data.Model Evaluation: Evaluate the performance of the model on the testing data using metrics such as mean squared error (MSE), mean absolute error (MAE), and coefficient of determination (R^2). You may also want to interpret the results of the model using feature importance techniques such as permutation feature importance or partial dependence plots.Model Deployment: Once the model has been trained and evaluated, deploy it into a production environment where it can be used to predict the housing prices of new properties.

CHAPTER 4

METHODOLOGY

4.1 TECHNOLOGY USED

There are several technologies that can be used for housing price detection, including:Machine learning: Machine learning algorithms can be used to analyze large datasets of property information to identify patterns and correlations between different factors and housing prices. Some popular machine learning techniques for housing price detection include linear regression, decision trees, random forests, and neural networks.Data visualization: Data visualization tools such as Tableau, Power BI, and Python's Matplotlib can be used to create interactive visualizations of housing price data. These visualizations can help to identify patterns and trends in the data, as well as provide insights into the factors that influence housing prices.Web scraping: Web scraping tools such as BeautifulSoup and Scrapy can be used to extract data from property listings, real estate websites, and other online sources. This data can then be used to train machine learning models or to create visualizations of housing price trends.Geographic information systems (GIS): GIS software such as ArcGIS and QGIS can be used to map and analyze spatial data related to housing prices. This can include data on local market conditions, nearbyamenities, and other geographic factors that may influence housing prices.Cloud computing: Cloud computing platforms such as Amazon Web Services (AWS) and Microsoft Azure can be used to store and process large datasets of housing price data. These platforms can provide scalable computing resources for machine learning algorithms and other data processing tasks, making it easier to analyze large amounts of data quickly and efficiently.

4.2 TOOLS USED

There are a variety of tools that can be used for housing price detection, including:Python: Python is a popular programming language for data science and machine learning, and offers a wide range of libraries and tools for analyzing housing price data. Popular libraries for data manipulation and analysis include Pandas and NumPy, while popular machine learning libraries include Scikit-learn and TensorFlow.R: R is another popular programming language for data science and statistical analysis. It offers a wide range of packages for data manipulation and analysis, as well as machine learning packages such as caret and MLR.Tableau: Tableau is a popular data visualization tool that can be used to create interactive visualizations of housing price data. It offers a wide range of data visualization options and can be used to create dashboards and reports that can be shared with others.Excel: Excel is a widely used spreadsheet program that can be used for data manipulation, analysis, and visualization. It offers a wide range of features for working with housing price data, including pivot tables, charts, and graphs.GIS software: GIS software such as ArcGIS and QGIS can be used to map and analyze spatial data related to housing prices. These tools can be used to create maps that visualize housing price trends and other geographic factors that may influence housing prices.Web scraping tools: Web scraping tools such as BeautifulSoup and Scrapy can be used to extract data from property listings, real estate websites, and other online sources. This data can then be used for analysis or visualization using other tools and techniques.Cloud computing platforms: Cloud computing platforms such as Amazon Web Services (AWS) and Microsoft Azure can be used to store and process large datasets of housing price data. These platforms can provide scalable computing resources for machine learning algorithms and other data processing tasks.

4.3 HOW IT WORKS?

Housing price detection typically involves analyzing a large dataset of property information to identify patterns and correlations between different factors and housing prices. This analysis can be performed using a variety of machine learning algorithms, statistical methods, and data visualization techniques.The first step in housing price detection is typically to collect a large dataset of property information. This can include data on the location, square footage, number of bedrooms and bathrooms, age of the property, local market conditions, and nearby amenities such as schools, parks, and shopping centers.Once the data has been collected, it can be preprocessed and cleaned to remove any missing or inconsistent data points. This may involve imputing missing values, scaling and normalizing the data, and encoding categorical variables.After preprocessing, the data can be split into training and testing sets for machine learning algorithms. The training data is used to train a model to predict housing prices based on the input features, while the testing data is used to evaluate the performance of the model.A wide range of machine learning algorithms can be used for housing price detection, including linear regression, decision trees, random forests, and neural networks. These algorithms work by identifying patterns and correlations between the input features and the target variable (housing prices), and using these patterns to make predictions on new data points.Once a model has been trained and validated, it can be used to make predictions on new data points, such as a new property listing. The accuracy of the predictions will depend on the quality and quantity of the data used to train the model, as well as the specific algorithm and techniques used.Data visualization tools can also be used to explore and visualize housing price data. This can help to identify patterns and trends in the data, as well as provide insights into the factors that influence housing prices.

4.4 WHAT IT DOES?

Housing price detection is a process that uses data analysis and machine learning algorithms to predict the price of a house or property based on various factors such as location, size, age, and other features. The goal of housing price detection is to provide accurate and reliable estimates of property values, which can be useful for a variety of purposes including real estate valuation, investment analysis, and property management.By analyzing a large dataset of property information and identifying patterns and correlations between different factors and housing prices, housing price detection can provide insights into the factors that influence housing prices and help to identify undervalued or overvalued properties. This information can be used by real estate agents, investors, and homeowners to make informed decisions about buying, selling, and managing properties.Housing price detection can also be used to develop predictive models that can be used to forecast future trends in the housing market, such as changes in property values or demand for certain types of properties. These models can be used by real estate companies, banks, and government agencies to make data-driven decisions about real estate investments and policies.Overall, housing price detection plays an important role in the real estate industry and can provide valuable insights and predictions for a variety of stakeholders

CHAPTER 5

CODING AND TESTING

5.1 CODE FOR HOUSING PRICE DETECTION

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import chardet#for encoding
import warnings# to avoid the warnings
warnings.filterwarnings('ignore')
pd.pandas.set_option('display.max_columns',0)


#Let's see which encoding we have to apply.
with open("new.csv","rb") as f:
    result=chardet.detect(f.read(100000))
print(result)

#so,we have to apply GB2312 encoding.
data=pd.read_csv("new.csv",encoding="GB2312")

data.head()

data.shape

df0=data.copy()

data.columns

data.info()

data.isnull().sum()

#Let's Visualize the missing value
sns.heatmap(data.isnull(),yticklabels=False,cbar=False)

#Drop 'DOM' Columns
data.drop(columns=['DOM'],axis=1,inplace=True)
                                                                    In [12]:
data.shape

data.isnull().sum()

data['buildingType'].fillna(data.buildingType.mode(),inplace=True)
                                                                    In [15]:
data.elevator.fillna(data.elevator.mode(),inplace=True)
                                                                    In [16]:
data.fiveYearsProperty.fillna(data.fiveYearsProperty.mode(),inplace=True)
                                                                    In [17]:
data.subway.fillna(data.subway.median(),inplace=True)
                                                                    In [18]:
data.communityAverage.fillna(data.communityAverage.median(),inplace=True)
                                                                    In [19]:
data.livingRoom.unique()
data.floor.unique()
#so,floor have a chinese character...

data.bathRoom.unique()

data.bathRoom.unique()

def Trade_Time(x):
    return x[0:4]
data['tradeTime']=data['tradeTime'].apply(Trade_Time)
data.head()

#convert tradetime into int numeric
data['tradeTime'] = pd.to_numeric(data['tradeTime'])
data['livingRoom'] = data['livingRoom'].apply(pd.to_numeric, errors='coerce')
data['drawingRoom'] = data['drawingRoom'].apply(pd.to_numeric, errors='coerce')
data['bathRoom'] = data['bathRoom'].apply(pd.to_numeric, errors='coerce')
#convert ConstructionTime into int numeric
data['constructionTime'] = data['constructionTime'].apply(pd.to_numeric,
errors='coerce')

#now if we check livingRoom Column it is clean data.
data.livingRoom.unique()

#Now,Split the column into a Floor_Type and Floor_Height
def Floor_Type(x):
    return x.split(' ')[0]

def Floor_Height(y):
    try:
        return int(y.split(' ')[1])
    except:
        return np.nan

data['floor_type']=data['floor'].apply(Floor_Type)
data['floor_height']=data['floor'].apply(Floor_Height)
```

In [27]:

```
data.columns

data=data.drop(columns=['floor','url','id','Cid','price'])
data.head()

#Let's Perform one hot encoding
print(data.buildingType.unique())
print(data.renovationCondition.unique())
print(data.buildingStructure.unique())
#so,for buildingType we have a data like 0.5   0.333 0.125 0.25  0.429 0.048 0.375
0.667
# Which is unnecessary so,we have to remove them

#Removing unnecessary data which is present in buildingType
data=data[data['buildingType']>=1]
```

In [31]:

```
print(data.buildingType.unique())
print(data.shape)

#let's take a copy of our data for future use
df=data.copy()
```

In [33]:

```
col_for_dummies=['renovationCondition','buildingStructure','buildingType',
                 'district','elevator','floor_type']
data=pd.get_dummies(data=data,columns=col_for_dummies,drop_first=True)
```

In [34]:

```
data.head()

print(data.shape)
print(df0.shape)

data=data.dropna(axis=0)
```

In [37]:

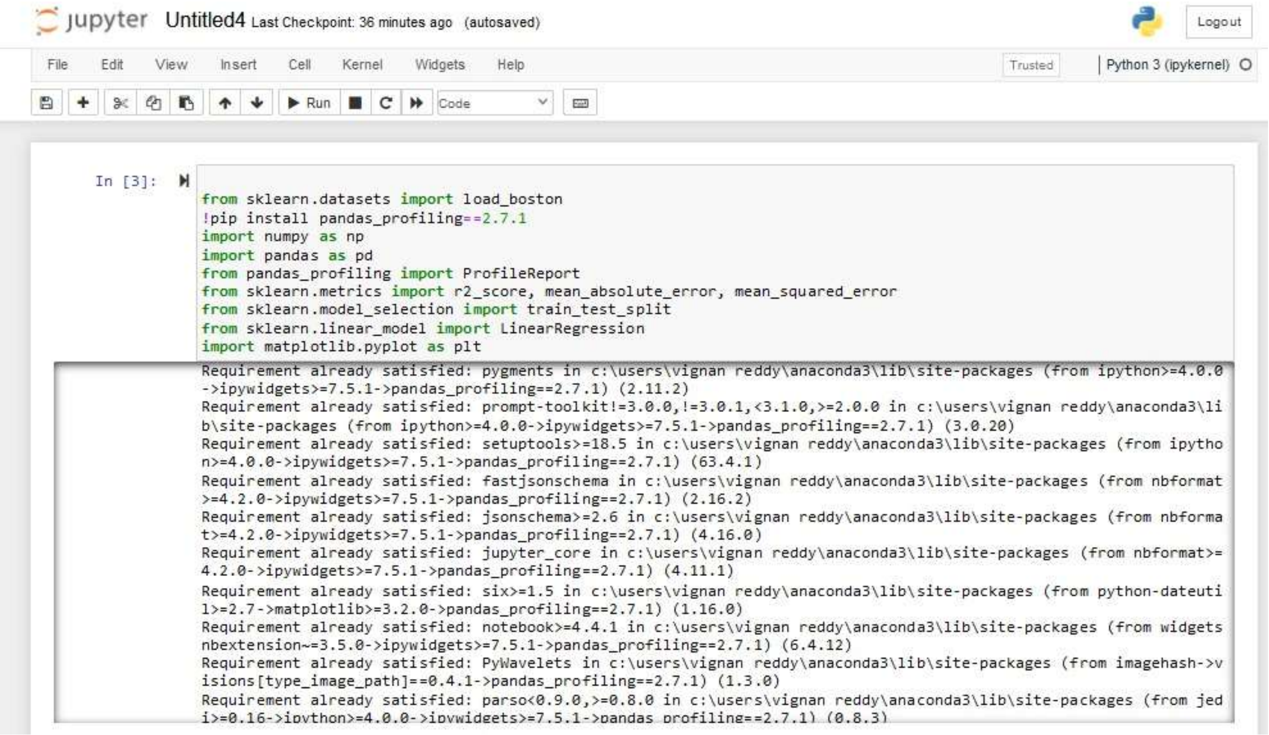
```
print(data.shape)

data.columns

df1=data[['Lng','Lat','tradeTime','totalPrice','followers','followers','livingRoom',
'drawingRoom','kitchen',
    'bathRoom','square','communityAverage','ladderRatio']]
```

In [40]:

```
plt.figure(figsize=(20,20))
sns.heatmap(df1.corr(),annot=True,cmap = "RdYlGn")
plt.show()

sns.kdeplot(data=data['totalPrice'],shade=True)

data['totalPrice'].describe()

df.head()

sns.scatterplot(x=df['followers'],y=df['communityAverage'],hue=df['elevator'])

# sns.swarmplot(x=df['renovationCondition'],
#               y=df['followers'])

sns.lineplot(data=df['communityAverage'])

data.head()

data.shape

data.to_csv("After_EDA.csv")
```

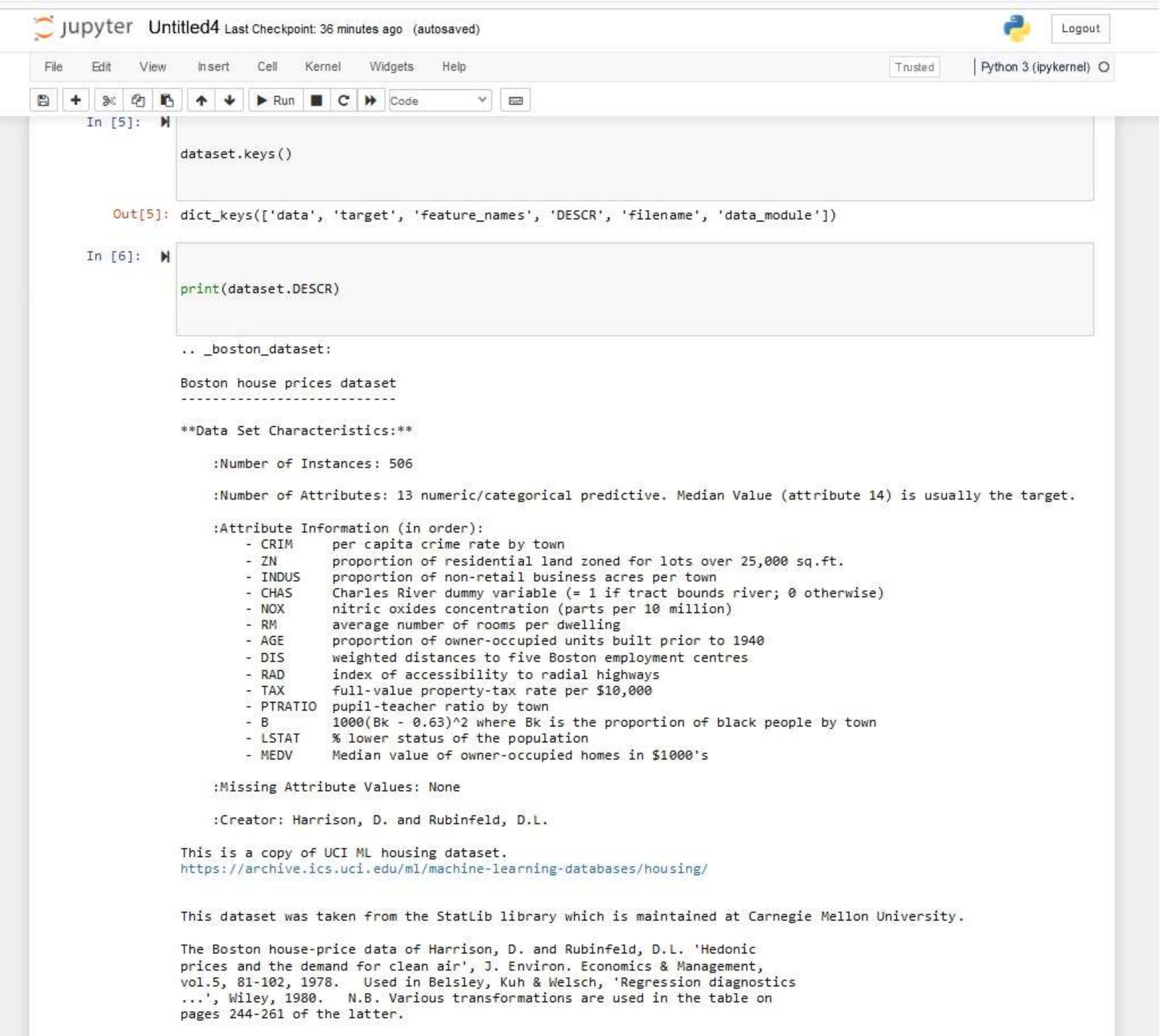
CHAPTER 6

SCREENSHOTS AND RESULTS


jupyter Untitled4 Last Checkpoint: 36 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [3]:

```
from sklearn.datasets import load_boston
!pip install pandas_profiling==2.7.1
import numpy as np
import pandas as pd
from pandas_profiling import ProfileReport
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

```
Requirement already satisfied: pygments in c:\users\vignan reddy\anaconda3\lib\site-packages (from ipython>=4.0.0
->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (2.11.2)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in c:\users\vignan reddy\anaconda3\li
b\site-packages (from ipython>=4.0.0->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (3.0.20)
Requirement already satisfied: setuptools>=18.5 in c:\users\vignan reddy\anaconda3\lib\site-packages (from ipytho
n>=4.0.0->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (63.4.1)
Requirement already satisfied: fastjsonschema in c:\users\vignan reddy\anaconda3\lib\site-packages (from nbformat
>=4.2.0->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (2.16.2)
Requirement already satisfied: jsonschema>=2.6 in c:\users\vignan reddy\anaconda3\lib\site-packages (from nbforma
t>=4.2.0->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (4.16.0)
Requirement already satisfied: jupyter_core in c:\users\vignan reddy\anaconda3\lib\site-packages (from nbformat>=
4.2.0->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (4.11.1)
Requirement already satisfied: six>=1.5 in c:\users\vignan reddy\anaconda3\lib\site-packages (from python-dateuti
l>=2.7->matplotlib>=3.2.0->pandas_profiling==2.7.1) (1.16.0)
Requirement already satisfied: notebook>=4.4.1 in c:\users\vignan reddy\anaconda3\lib\site-packages (from widgets
nbextension~=3.5.0->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (6.4.12)
Requirement already satisfied: PyWavelets in c:\users\vignan reddy\anaconda3\lib\site-packages (from imagehash->v
isions[type_image_path]==0.4.1->pandas_profiling==2.7.1) (1.3.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\vignan reddy\anaconda3\lib\site-packages (from jed
i>=0.16->ipython>=4.0.0->ipywidgets>=7.5.1->pandas_profiling==2.7.1) (0.8.3)
```



In [5]:

```
dataset.keys()
```

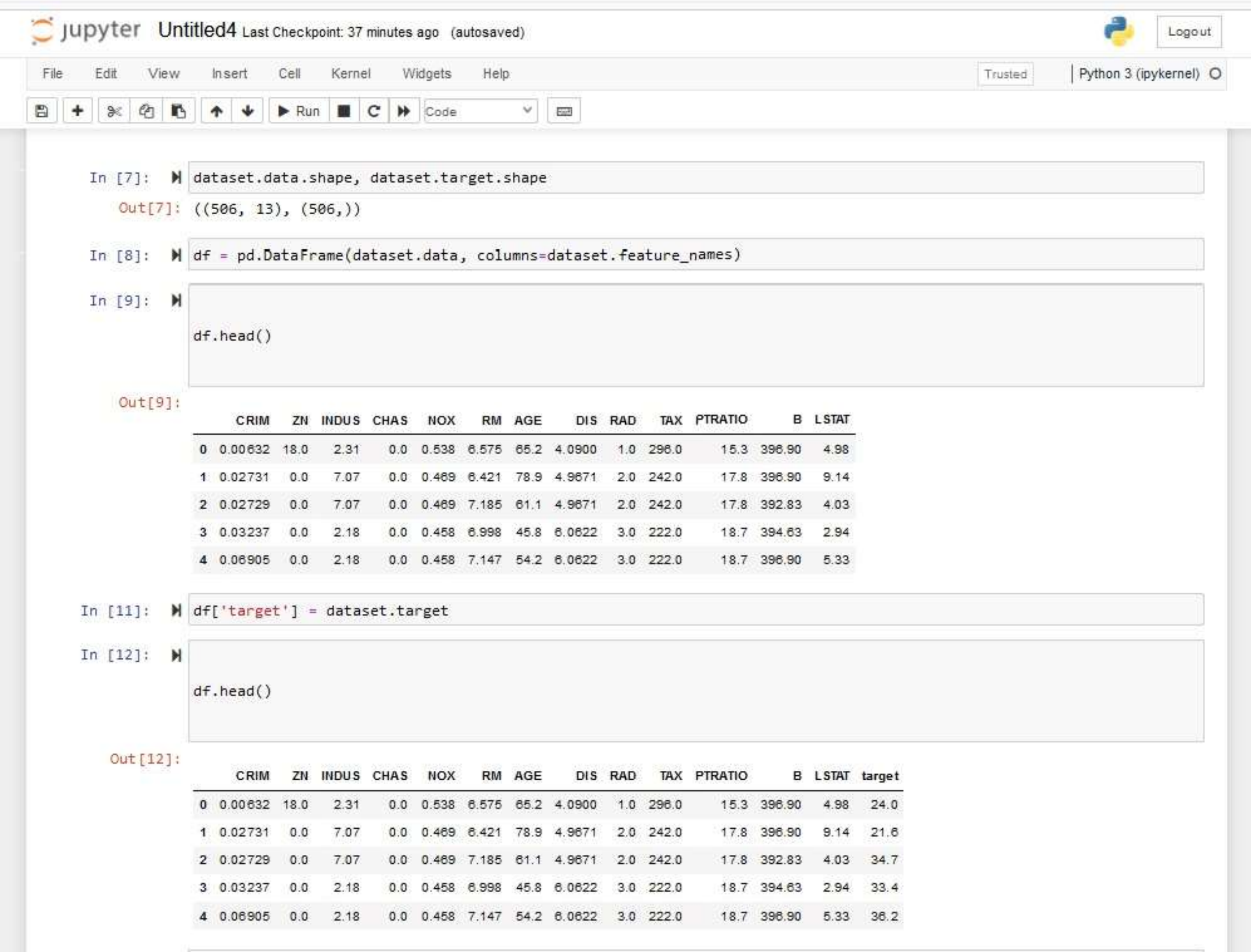
Out[5]:

```
dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename', 'data_module'])
```

In [6]:

```
print(dataset.DESCR)
```

```
.. _boston_dataset:

Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of black people by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/


This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980.   N.B. Various transformations are used in the table on
pages 244-261 of the latter.
```



```
In [7]:  dataset.data.shape, dataset.target.shape
Out[7]: ((506, 13), (506,))
```

```
In [8]:  df = pd.DataFrame(dataset.data, columns=dataset.feature_names)
```

```
In [9]:

df.head()
```

Out[9]:

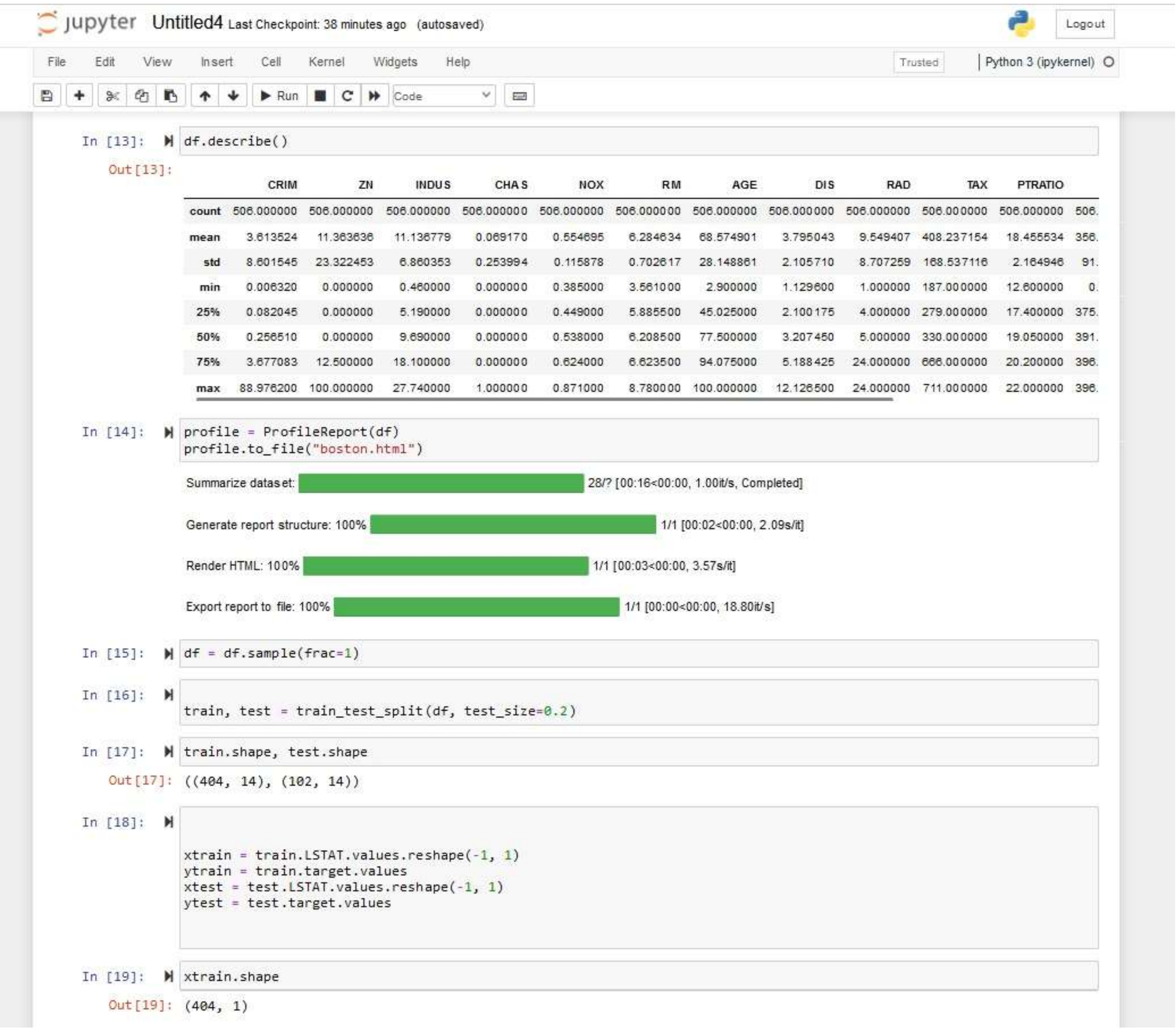
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

```
In [11]:  df['target'] = dataset.target
```

```
In [12]:

df.head()
```

Out[12]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	target
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2



```
In [13]:  df.describe()
```

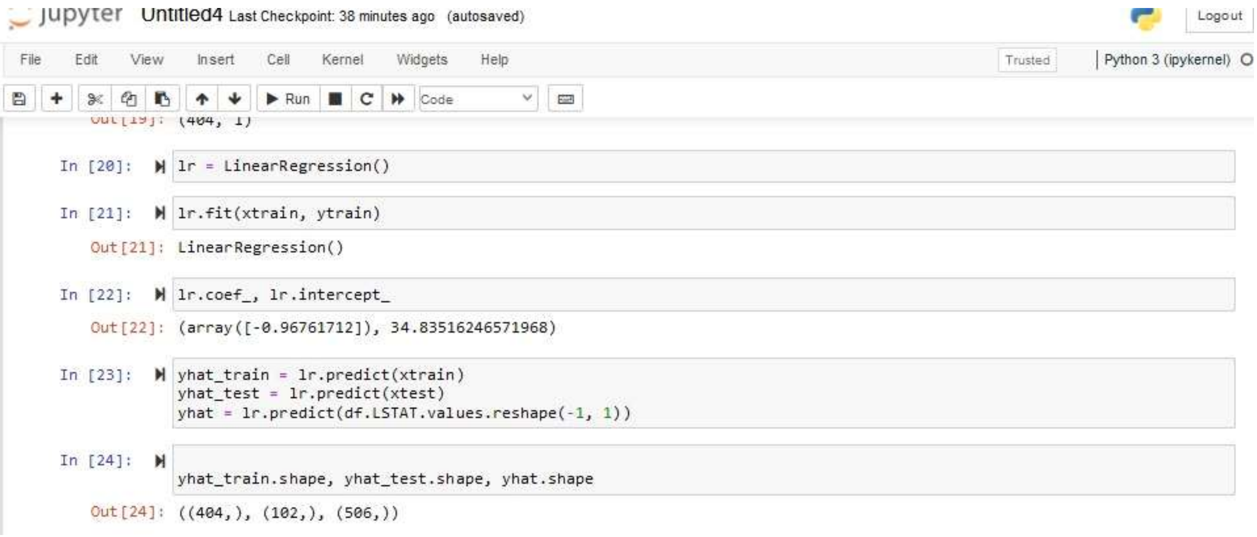
Out[13]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.

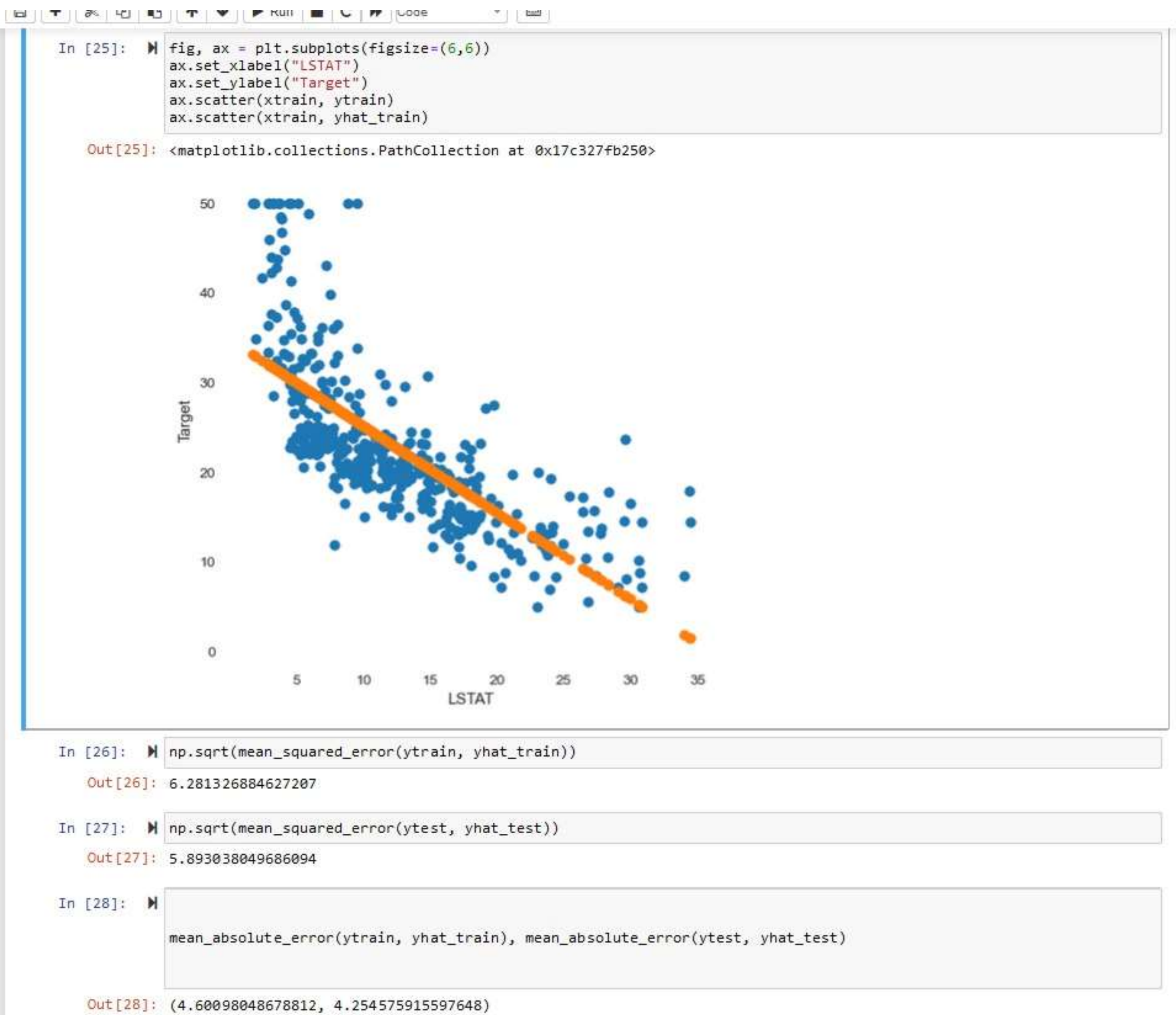
```
In [14]:  profile = ProfileReport(df)
          profile.to_file("boston.html")
```

```
Summarize dataset: 28/? [00:16<00:00, 1.00it/s, Completed]

Generate report structure: 100% 1/1 [00:02<00:00, 2.09s/it]

Render HTML: 100% 1/1 [00:03<00:00, 3.57s/it]

Export report to file: 100% 1/1 [00:00<00:00, 18.80it/s]
```

```
In [15]:  df = df.sample(frac=1)

In [16]:
          train, test = train_test_split(df, test_size=0.2)

In [17]:  train.shape, test.shape

Out[17]:  ((404, 14), (102, 14))

In [18]:

          xtrain = train.LSTAT.values.reshape(-1, 1)
          ytrain = train.target.values
          xtest = test.LSTAT.values.reshape(-1, 1)
          ytest = test.target.values

In [19]:  xtrain.shape

Out[19]:  (404, 1)
```



```
Out[19]: (404, 1)

In [20]:   lr = LinearRegression()

In [21]:   lr.fit(xtrain, ytrain)

Out[21]: LinearRegression()

In [22]:   lr.coef_, lr.intercept_

Out[22]: (array([-0.96761712]), 34.83516246571968)

In [23]:   yhat_train = lr.predict(xtrain)
           yhat_test = lr.predict(xtest)
           yhat = lr.predict(df.LSTAT.values.reshape(-1, 1))

In [24]:
           yhat_train.shape, yhat_test.shape, yhat.shape

Out[24]: ((404,), (102,), (506,))
```



```
In [25]:   fig, ax = plt.subplots(figsize=(6,6))
           ax.set_xlabel("LSTAT")
           ax.set_ylabel("Target")
           ax.scatter(xtrain, ytrain)
           ax.scatter(xtrain, yhat_train)

Out[25]: <matplotlib.collections.PathCollection at 0x17c327fb250>

In [26]:   np.sqrt(mean_squared_error(ytrain, yhat_train))

Out[26]: 6.281326884627207

In [27]:   np.sqrt(mean_squared_error(ytest, yhat_test))

Out[27]: 5.893038049686094

In [28]:

           mean_absolute_error(ytrain, yhat_train), mean_absolute_error(ytest, yhat_test)

Out[28]: (4.600980486788812, 4.254575915597648)
```

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The task of housing price detection typically involves using various data analysis and machine learning techniques to predict the value of residential properties based on various features such as location, square footage, number of bedrooms and bathrooms, age of the property, etc.There are several approaches to predicting housing prices, including linear regression, decision trees, random forests, and neural networks. Each of these methods has its strengths and weaknesses and the best approach will depend on the specific dataset and problem at hand.In general, however, it is important to carefully preprocess and clean the data before applying any model. This may involve handling missing values, removing outliers, and scaling or transforming the data as necessary. Additionally, it is important to properly split the data into training, validation, and testing sets to avoid overfitting.Overall, accurately predicting housing prices can be a challenging task but with the right data, tools, and techniques, it is possible to build reliable models that can be used to inform important decisions in the real estate industry.

7.2 Future scope

housing price detection is quite promising. With the increasing availability of real estate data and advancements in machine learning techniques, it is likely that housing price detection models will become even more accurate and sophisticated.One potential area of future development is the incorporation of more advanced deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These models can be used to extract meaningful features from large amounts of real estate data, including images, text descriptions, and other types of information. This could lead to more accurate and comprehensive housing price predictions.Another area of future development is the integration of more complex data sources such as social media data, crime statistics, and other neighborhood-level information. By combining these types of data with traditional real estate data, it may be possible to create even more accurate and nuanced housing price models that take into account a wider range of factors.Finally, the development of automated valuation models (AVMs) may also be an area of future growth. AVMs are designed to automatically estimate the value of a property using a combination of statistical models and machine learning techniques. As these models become more sophisticated, they could become an increasingly important tool for real estate professionals, lenders, and investors.

REFERENCES

[1] Zhang, X., Davidson, E. A,"Improving Nitrogen and Water Management in Crop Production on a National Scale", American Geophysical Union, December, 2018.How to Feed the World in 2050 by FAO.

[2] Abhishek D. et al., "Estimates for World Population and Global Food Availability for Global Health", Book chapter, The Role of Functional Food Security in Global Health, 2019, Pages 3-24.Elder M., Hayashi S., "A Regional Perspective on Biofuels in Asia", in Biofuels and Sustainability, Science for Sustainable Societies, Springer, 2018.

[3] Zhang, L., Dabipi, I. K. And Brown, W. L, “Internet of Things Applications for Agriculture". In, Internet of Things A to Z: Technologies and Applications, Q. Hassan (Ed.), 2018.

[4] S. Navulur, A.S.C.S. Sastry, M.N. Giri Prasad,"Agricultural Management through Wireless Sensors and Internet of Things" International Journal of Electrical and Computer Engineering (IJECE), 2017; 7(6) :3492-3499.

[5] E. Sisinni, A. Saifullah, S.Han, U. Jennehag and M.Gidlund, "Industrial Internet ofThings: Challenges,Opportunities, and Directions," in IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724-4734, Nov. 2018.

[6] M. Ayaz, M. Ammad-uddin, I. Baig and e. M. Aggoune, "Wireless Possibilities: A Review," in IEEE Sensors Journal, vol. 18, no. 1, pp. 4-30, 1 Jan.1, 2018.