

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

## Jnana Sangama, Belagavi – 590 018.

***An Internship Report On***

## “HOUSE PRICE PREDICTION” 12.09.2022 to 10.10.2022

*Submitted in partial fulfilment of the for the award of the degree of*

Bachelor of Engineering

*In*

Computer Science and Engineering

*Submitted by*

|  |  |
| --- | --- |
| **SABA HIFZA** | **1VI19CS087** |

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING VEMANA INSTITUTE OF TECHNOLOGY

**BENGALURU – 560034**

### 2022-2023



Karnataka Reddy Jana Sangha**®**

VEMANA INSTITUTE OF TECHNOLOGY

**(Affiliated to Visvesvaraya Technological University, Belagavi)**

Koramangala, Bengaluru-560034.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# CERTIFICATE

This is to certify that the Internship work entitled “**HOUSE PRICE PREDICTION**” is a bonafide work carried out by **Ms. SABA HIFZA(1VI19CS087)** during the academic year 2022-23 in partial fulfilment of the requirement for the award of **Bachelor of Engineering in Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi.** It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The internship report has been approved as it satisfies the academic requirements in respect of the Internship prescribed for the said degree.

|  |  |  |
| --- | --- | --- |
| **Guide** | **HOD** | **Principal** |
| Mr. P Gopala Krishna | Dr. M. Ramakrishna | Dr. Vijayasimha Reddy B.G |

## External Viva

Name of the Examiner Signature with date 1.

2.

# ACKNOWLEDGEMENT

I sincerely thank Visvesvaraya Technology University for providing a platform to do the internship work.

I express my sincere thanks to **Dr. Vijayasimha Reddy B G**, Principal, Vemana Institute of Technology, Bengaluru, for providing necessary facilities and motivation to carry out internship work successfully.

I express heartfelt gratitude and humble thanks to **Dr. M. Ramakrishna**, Professor and Head, Computer Science and Engineering, Vemana Institute of Technology, for his constant encouragement, inspiration and help to carryout internship work successfully.

I am very thankful to my external guide, **Santosh,** full stack data scientist at Cranes Varsity who has given in-time valuable instructions and put me in contact with experts in the field, with extensive guidance regarding practical issues.

I would like to express my sincere gratitude towards my internal guide, Ms. P Gopala Krishna, Assistant Professor for providing encouragement and inspiration throughout the internship.

I thank internship coordinators Prof. Naveen H S and Prof. Kavitha Bai A.S for their cooperation and support during the internship work.

I am thankful to all the teaching and non-teaching staff members of Computer Science and Engineering department for their help and much needed support throughout the internship.

SABA HIFZA 1VI19CS087



# ABSTRACT

The real estate market is a standout amongst the most focused regarding pricing and keeps fluctuating. It is one of the prime fields to apply the ideas of machine learning on how to enhance and foresee the costs with high accuracy. The price for a property is based on the datasets of the houses in a particular range. The proposed technique considered the more refined aspects used for the calculation of house prices and to find out more accurate prediction utilizing different regression methods with the help pf python libraries. It will help clients to put resources without moving towards a broker. This study employs a variety of machine learning algorithms. This research gives a comparison of fundamental and technical analysis based on many characteristics. This research compares and contrasts several prediction algorithms used to predict house prices. With the use of visualization, several techniques are evaluated based on methodologies, datasets, and efficiency.

Keywords: Machine Learning, Regression Technique, XGBRegression model, Classification Technique, Graphical and Numerical representation Decision Tree, Precision, Recall, Accuracy.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE.NO** |
| **1** | **INTRODUCTION** | **1** |
| **2** | **ORGANIZATION OVERVIEW** | **2** |
| **3** | **SYSTEM REQUIRMENTS** | **4** |
|  | 3.1 Software Requirements | 4 |
|  | 3.2 Hardware Requirements | 4 |
|  | 3.3 Tools and languages used | 4 |
| **4** | **BASIC CONCEPTS OF PYTHON** | **8** |
|  | 4.1 Variable | 8 |
|  | 4.2 String | 8 |
|  | 4.3 Files | 10 |
|  | 4.4 Lists | 11 |
| **5** | **MODULES USED IN THE DATASET** | **12** |
| **6** | **IMPLEMENTATION** | **15** |
|  | 6.1 Importing the dependency | 19 |
|  | 6.2 Study of dataset | 19 |
|  | 7.1 Source code | 21 |
| **8** | **CONCLUSION** | **22** |
| **9** | **REFERENCE** | **23** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG.NO** | **NAME** | **PAGE.NO** |
| 3.1 | Jupyter Notebook | 5 |
| 3.2 | New Notebook | 5 |
| 3.3 | Setting Notebook name | 5 |
| 3.4 | Executing code | 6 |
| 3.5 | Output display | 6 |
| 3.6 | Code and text button | 6 |
| 3.7 | Run all | 7 |
| 3.8 | Deleting cell | 7 |
| 5.1 | Data flow diagram | 10 |
| 6.2 | First 5 rows of dataset | 15 |
| 6.3 | No missing values | 15 |
| 6.4 | Statistical measures | 16 |
| 6.5 | Heat map | 16 |
| 6.6 | Training Models | 17 |
|  |  |  |

**LIST OF TABLES**

**TABLE.NO NAME PAGE.NO**

* 1. Variable Types 8
  2. String operation 9
  3. Different modes of file opening 10
  4. Lists operation 11
  5. Function of list 11

**CHAPTER 1**

# Introduction

The real estate sector is an important industry with many stakeholders ranging from regulatory bodies to private companies and investors. Among these stakeholders, there is a high demand for a better understanding of the industry operational mechanism and driving factors. [1-3] Today there is a large amount of data available on relevant statistics as well as on additional contextual factors, and it is natural to try to make use of these in order to improve our understanding of the industry. As a house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. For example, When any person/business wants to sell or buy a house, they always face this kind of issue as they don't know the price which they should offer. Due to this they might be offering too low or high for the property. Therefore, the available data given for the properties in the area can be used and analyzed which will help in predicting the price. How these attributes influence the house prices needs to be found. Right pricing is a very important aspect to sell a house. It is very important to understand what are the factors and how they influence the house price. The Objective is to predict the right price of the house based on the attributes. [4-6] This project aims to provide an overview of how to predict house prices using different regression methods and python libraries. In this study, Python programming language with a number of Python packages will be used.

[7-8] Having selected the most important features an random forest regression model is trained for working on changes in house price prediction. The proposed technique considers refined aspects for calculating house prices and provides more accurate predictions. It also covers various graphical and numerical techniques required for predicting house prices and explains how the model works with the help of machine learning and dataset is used in the proposed model.

### 

### Objectives

The objectives of this project are as follows:

* To apply data pre-processing and preparation techniques in order to obtain clean data
* To build machine learning models able to predict house price based on house features
* To analyse and compare models performance in order to choose the best model

**CHAPTER 2**

# Organization Overview

The passion of Cranes Varsity towards work defines the core values. Cranes Varsity is a pioneer Technical Training institute turned EdTech Platform offering Technology educational services for over 24 years. Cranes Varsity carries a legacy of being the Authorized-training partner for Texas Instruments, MathWorks, Wind River & ARM. Cranes Varsity has training leadership in EMBEDDED, MATLAB & DSP, extending training domains to emerging industry trends like Automotive, IoT, VLSI, Java full-stack, Data Science & Business Analytics.

Cranes is designed with two wings:

### Training and Placement

### Being a trusted partner of over 5000+ reputed Academia, Corporate & Defense Organizations they had successfully trained 1 Lakh+ engineers via its network of 2000+ Universities & colleges and placed 70,000+ engineers at major Indian Corporate & MNCs. Over 50,000+ Alumnae testified their legacy and are the great ambassadors of Cranes Varsity Brand through their jobs worldwide. The in-house placement team further ensures that these students get placed in leading corporate firms – with whom Cranes Varsity has decades-old relationships.

### 2. Transformation Centre for graduates, students and corporates

It was established with an ambitious vision of bridging the gap between the technology academia and the industry. The team continuously strives to be an organization that brings together technology and education, empowering aspiring professionals to seek assured placements and a lucrative career path.

**Mission of the organization**

Empower every Person and every company to achieve more with the commitments to employee development and client partnerships.

## Vision of the organization

Setting a benchmark in software Application and product development and transforming individuals to build their passionate dream career.

The following are the key highlights in redefining their learning process:

* Transformation of Learning/Thought process to challenge Interviews
* Capability built through Hands-on Projects & Periodic Assessment to measure Learning.
* Project Implementation using Industry Methodology.
* Cost Effective.
* Free Career Counselling.
* Placement Assurance for focused work.
* Guidance Open Interactive session with corporate experts.
* for Funding/ Innovative / R & D Projects.
* Workshops/ Internships/ Courses/ Team Projects on Technical Niche skill.

Cranes Varsity’s high-impact hands-on technology training catapults engineering students, graduates, and working professionals to be quickly employable in Niche high-end engineering fields. Being a trusted recruitment & training partner with Corporate, they engage with them for the “Hire, Train & Deploy” Model. They offer an array of high-end technology training in Embedded & Automotive Systems, C, C++, MATLAB, RTOS, Linux, LDD, BSP, Embedded Testing, IoT Architecture, Protocols – Edge node Computing, Gateway & Security with industrial IoT, DSP & MATLAB, VLSI design, Java technologies, Cloud Computing, Azure, Python, Data Science & Analytics, Tableau, Artificial Intelligence with Machine Learning, Deep Learning, NLP, Business Intelligence and more. Their Learning Approach Model is EEE – Educate, Evolve, Employment through our Pedagogical practices that integrate Learning Management Systems (LMSS).

**Logo of the company**:



**CHAPTER 3**

**System Requirements**

## Software Requirements

* Jupyter Notebook
* Numpy (Library function)
* Pandas (Library function)
* Sk Learn Linear Model

## Hardware Requirements

* RAM-4gb or more
* Cache-4mb or more
* GPU-2gb or more
* Monitor-LED, LCD display recommended with a refresh rate of 40Hz minimum.
* OS-Any compatible operating systems like windows, ubuntu, linux is advisable.

## Tools and language used

## Jupyter Notebook

The Jupyter Notebook is an unbelievably incredible asset for intelligently creating and introducing information science ventures. This article will show how to set Jupyter Notebooks on your machine and how to use for data science projects. A note pad coordinates code and its yield into a solitary archive that consolidates representations, story text, numerical conditions, and other media. This natural work process advances iterative and fast turn of events, settling on note pad an inexorably well-known decision at the core of contemporary data science, examination, and progressively science at large.

As a component of the open source Project Jupyter, they are totally free. The Jupyter venture is the replacement to the prior I Python Notebook, which was first distributed as a model in 2010. Despite the fact that it is conceivable to utilize various programming dialects inside Jupyter Notebooks, The focus is on Python, as it is the most widely-used example. To take full advantage of this resource, familiarity with programming is required. Explicitly Python and pandas explicitly.

All things considered, in the event that you have involvement in another dialect, the Python in this article shouldn't be excessively obscure, will in any case assist you with getting Jupyter Notebooks set up locally.

**CHAPTER 4**

* 1. **Variables**

# Basic Concepts of Python

A Python variable is a reserved memory location to store values. In other words, a variable in python program gives data to the computer for processing. Every value in Python has a datatype. Different data types in Python are Numbers, List, Tuple, Strings, Dictionary, etc. Variables can be declared by any name or even alphabets like a, aa, abc, etc**.**

Python supports four different numerical types: below table 4.1 briefs about it. int (signed integers)

long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values

complex (complex numbers)

**Table 4.1 Variable types**

|  |  |  |  |
| --- | --- | --- | --- |
| Int | Long | Float | Complex |
| 10 | 51924361L | 0.0 | 3.14j |
| 100 | -0x19323L | 15.2 | 45.j |
| -786 | 0xDEFABCECBDAECBFBAEI | -21.9 | 9.322e-36j |
| 080 | 535633629843L | 32.3+e18 | .876j |
| -0490 | -052318172735L | -90 | -.6545+0j |
| -0x69 | -4721885298529L | -32.54e100 | 3e+26j |

* 1. **String**

String is the most popular types in Python. It can be created by enclosing characters in quotes. Python treats single quotes as same as the double quotes. Creating strings is as simple as assigning a value to a variable.

### Accessing values in strings

Python does not support a character type. These are treated as strings of length one and also considered a substring. To access substrings, use the square brackets for slicing along with the index or indices to obtain the substring. For example

**var = 'Hello World!'**

**print "var[0]: ", var[0]**

When the above code is executed, it produces the following result

**Updated String : Hello Python**

### Updating string

To add or delete an existing string use “Update” by (re)assigning a variable to another string. The new value can be related to its previous value or to a completely different string altogether. For example

**var = 'Hello World!'**

**print "Updated String: - ", var[:6] + 'Python'**

### String operation

The string consists of characters, where various operations can be performed on a string and the below table 4.2 gives the list of string operations.

**Table 4.2 String operation**

|  |  |
| --- | --- |
| Operator | Description |
| + | Adds values on either side of the operator |
| \* | Creates new strings, concatenating multiple copies of the same |
| [] | Slice - Gives the character from the given index |
| [:] | Range Slice - Gives the characters from the given range |
| In | Membership - Returns true if a character exists in the given string |
| % | Format - Performs String formatting |

## Files

When the file has to be read or write, first open the file. Opening the file communicates with the operating system, which knows where the data for each file is stored. When the file is opened. The operating system to find’s the file by name and make sure the file exists. In this example, opening the file mbox.txt, which should be stored in the folder.

### Open a file

Before read or write a file, use Python’s built-in open () function to open the file. Syntax: fileObject = open(file\_name[access\_mode][buffering])

The parameter details are:

* File\_name − The file\_name argument is a string value that contains the name of the file which can be accessed.
* Access\_mode − The access\_mode determines the mode in which the file has to be opened, i.e., read, write, append, etc. A complete list of possible values is given below in the table 4.3.
* Buffering, If the buffering value is set to 0, no buffering takes place. If the buffering valueis 1, line buffering is performed while accessing a file. If the buffering value is specified asan integer greater than 1, then buffering action is performed with the indicated buffer size. If negative, the buffer size is the system default (default behavior).

### Table 4.3 modes of opening a file

|  |  |  |
| --- | --- | --- |
| Sl.No | Modes | Description |
| 1 | r | Opens a file for reading only. |
| 2 | r+ | Opens a file for both reading and writing. |
| 3 | w | Opens a file for writing only. |
| 4 | w+ | Opens a file for both writing and reading. |
| 5 | a | Opens a file for appending. |
| 6 | a+ | Opens a file for both appending and reading. |

**Close a file**

The close () method of a file object flushes any unwritten information and closes the file object, after which no more writing can be done. Python automatically closes a file when the reference object of a file is reassigned to another file. It is a good practice to use the close () method to close a file. Syntax: fileObject.close()

### Reading a file

The read() method reads a string from an open file. It is important to note that Python strings can have binary data. apart from text data.

Syntax: fileObject.read([count])

### Writing a file

The write () method writes any string to an open file. It is important to note that Python strings can have binary data and not just text. The write() method does not add a newline character ('\n') to the end of the string.

Syntax: fileObject.write(string)

## Lists

A string, a list is a sequence of values. In a string, the values are characters and, in a list, they can be any type. The values in list are called elements or sometimes items. The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

### List Operations

Lists respond to the + and \* operators much like strings, they mean concatenation and repetition and except that the result is a new list, not a string. The below table 4.4 shows the list operations.

### Table 4.4 list operation

|  |  |  |
| --- | --- | --- |
| Expression | Description | Result |
| len([1,2,3]) | Length | 3 |
| [1,2,3]+[4,5,6] | Concatenation | [1,2,3,4,5,6] |
| [“Hi!”]\*3 | Repetition | [“Hi!”,”Hi!”,”Hi!] |

**List functions:** Python has some built in functions which can be used on list to perform the functions. The sum () function only works when the list elements are numbers. The other functions (max (), len(), etc.) work with lists of strings and other types that can be comparable.

**Table 4.5 Functions of lists**

|  |  |  |
| --- | --- | --- |
| Sr.No | Function | description |
| 1 | cmp(list) | Compares elements of both lists. |
| 2 | len(list) | Gives the total length of the list. |
| 3 | max(list) | Returns item from the list with max value. |
| 4 | min(list) | Returns item from the list with min value. |

**CHAPTER 5**

**Modules used in the Dataset**

## 5.1 NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrixes. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and freely available. NumPy stands for Numerical Python. In Python, the lists serves the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important. NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in Cor C++.

### Installation of NumPy

If [Python](https://www.w3schools.com/python/default.asp) and [PIP](https://www.w3schools.com/python/python_pip.asp) is already installed on a system, then installation of NumPy is very easy. Install it using this command: *C:\Users\Your Name>pip install numpy*

Once NumPy is installed, import it in the applications by adding the import keyword. Example:

import numpy

arr = numpy.array([1,2,3,4,5])print(arr) #[1 2 3 4 5] NumPy is usually imported under the np alias.

### 5.2 Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. relevant data is very important in data science.

### 

### Installation of Pandas

If [Python](https://www.w3schools.com/python/default.asp) and [PIP](https://www.w3schools.com/python/python_pip.asp) already installed on a system, then installation of Pandas is very easy. Install it using this command: *C:\Users\Your Name>pip install pandas*

Once Pandas is installed, import it in the applications by adding the import keyword.

### 5.3 Seaborn

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions. Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

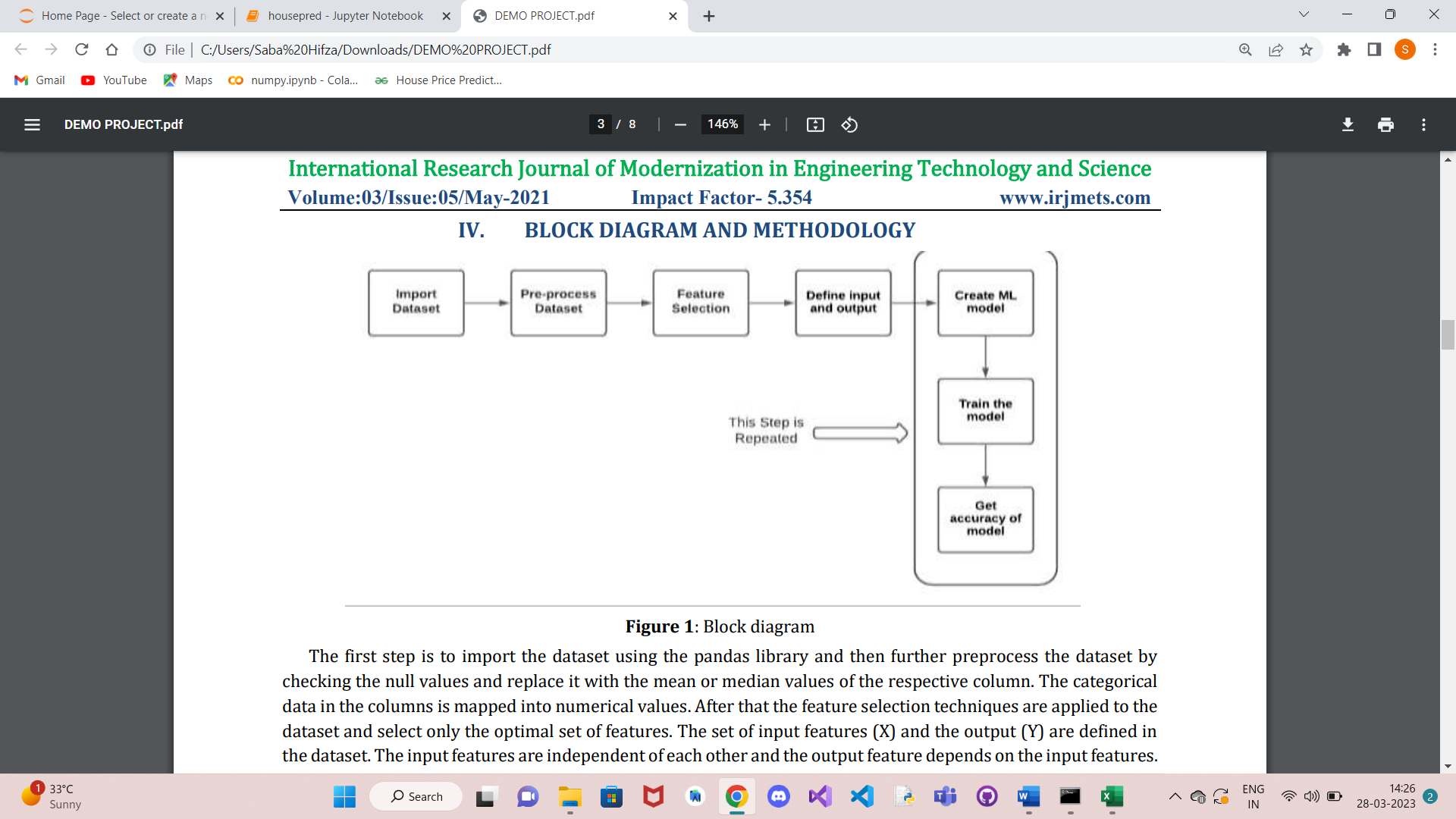
### Installation of Seaborn

If [Python](https://www.w3schools.com/python/default.asp) and [PIP](https://www.w3schools.com/python/python_pip.asp) already installed on a system,

install it using this command: *C:\Users\Your Name>pip install seaborn*

**5.4** **Matplotlib**

Matplotlib is a low-level library of Python which is used for data visualization. It is easy to use and emulates MATLAB like graphs and visualization. This library is built on the top of the NumPy arrays and consist of several plots like line chart, bar chart, histogram, etc.

**5.5 DATA FLOW DIAGRAM**

The first step is to import the dataset using the pandas library and then further preprocess the dataset by checking the null values and replace it with the mean or median values of the respective column. The categorical data in the columns is mapped into numerical values. After that the feature selection techniques are applied to the dataset and select only the optimal set of features. The set of input features (X) and the output (Y) are defined in the dataset. The input features are independent of each other and the output feature depends on the input features. Then the library is imported, and the ML Model is created, and the train-split-test method is used to separate the data into training data and testing data and then train the ML Model with training data and predict using test data. Then the accuracy of the model is calculated by simply taking the ratio of the predicted testing data and the actual testing data. This method is repeated with each ML Algorithm and the accuracy of each algorithm is calculated. Finally, the accuracy of the algorithms is compared and then which of the algorithms is the best for this dataset is determined.

**CHAPTER 6**

# RESULTS

## Importing the dependency

import pandas as pd import numpy as np

import matplotlib.pyplot as plt import seaborn as sns

import sklearn.datasets

from sklearn.model\_selection import train\_test\_split

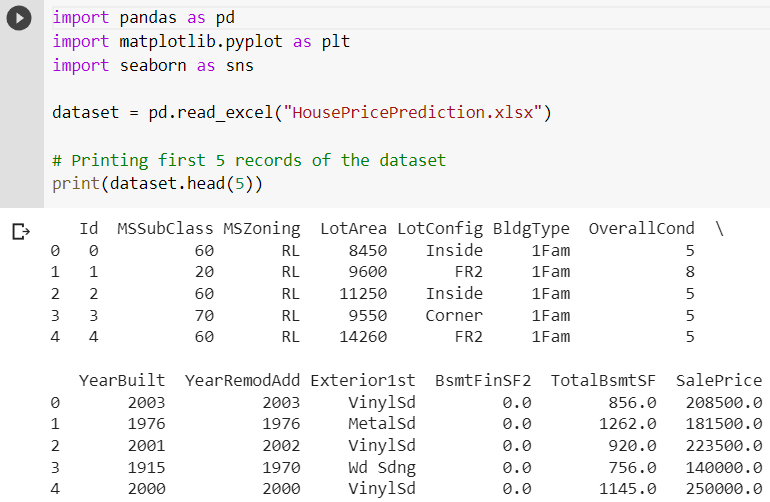
from sklearn.metrics import metrics

## Study of dataset

The next step is checking the technical information in the data

print(dataset.head(5))

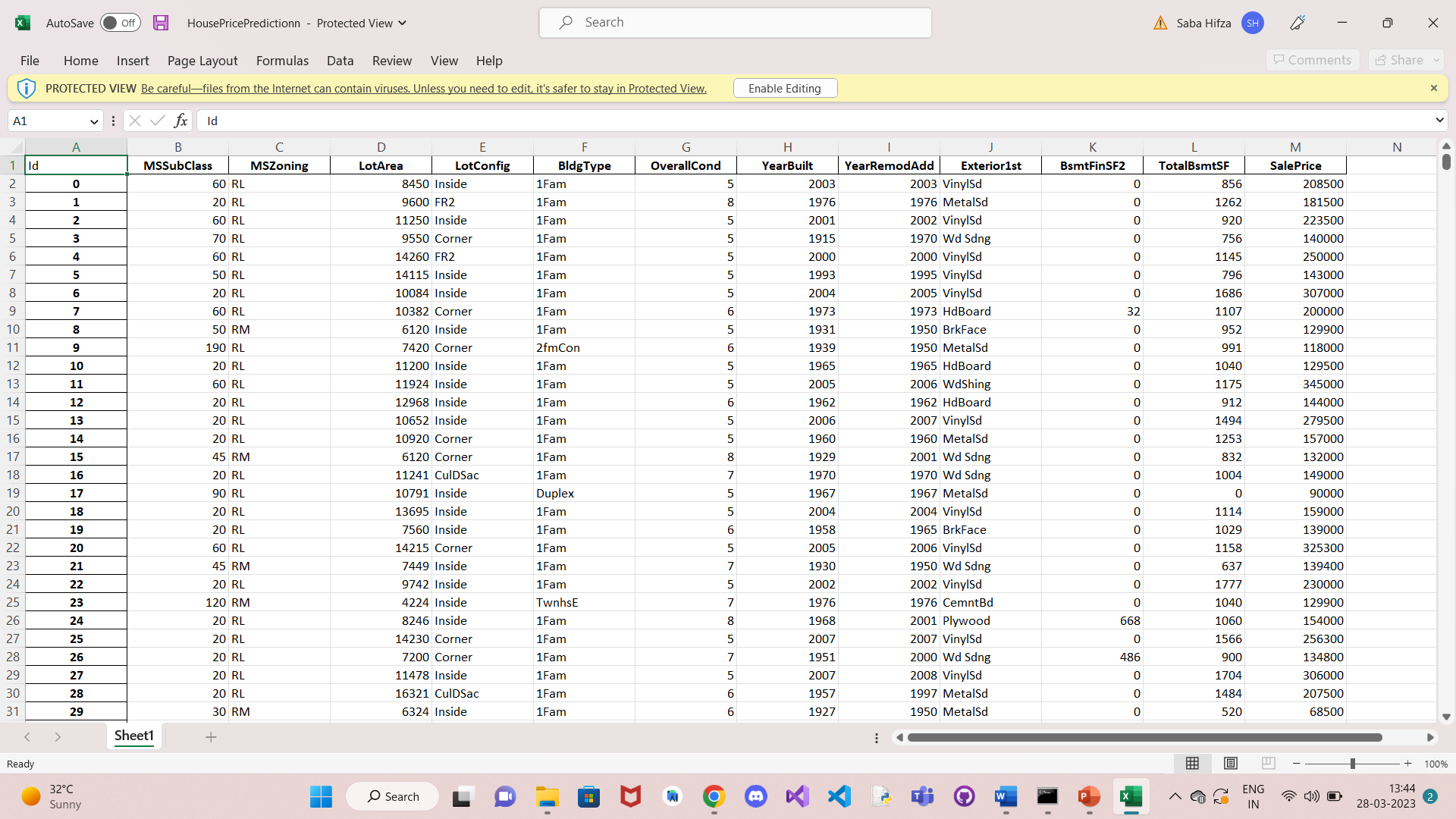
**Output:** Figure 6.1 displays the first 5 rows of the dataset



### Fig 6.1 First 5 rows of the dataset

### Statistical Measures of the Dataframe

housing\_dataframe.describe()

****

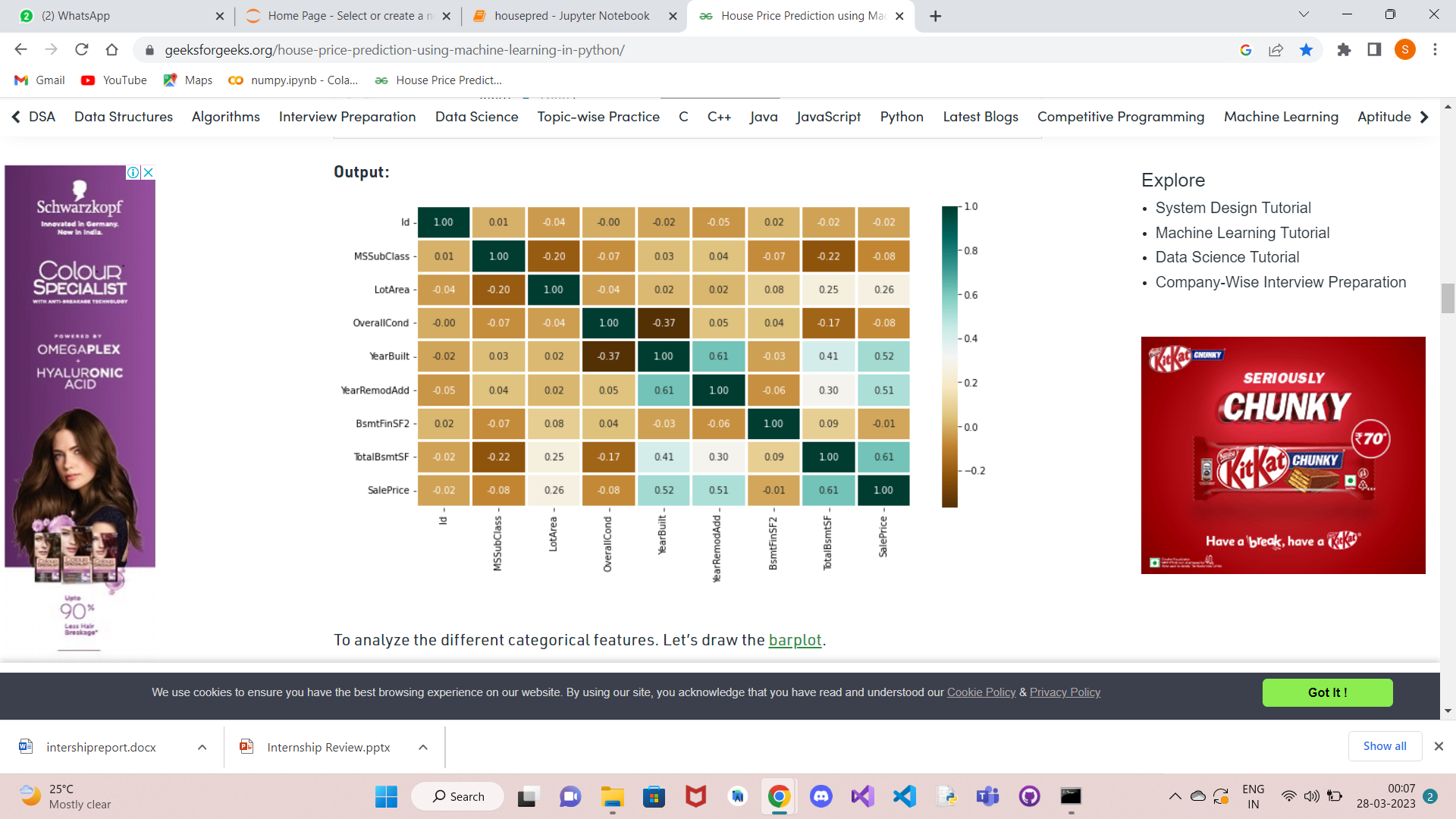
**Fig 6.3 Statistical Measures of the Dataframe**

**Correlation**

plt.figure(figsize=(10,10))

sns.heatmap(correlation, cbar=True, square=True, fmt = '1f' , annot = True, annot\_kws={'size':8}, cmap = 'Blues')

**Output:** Figure 6.4 shows the positive and negative correlation in the dataset



### Fig 6.4 Heatmap

### To analyze the different categorical feature

### Fig 3: No. Unique values of Categorical Features

### 

### The plot shows that Exterior1st has around 16 unique categories and other features have around  6 unique categories. To find out the actual count of each category we can plot the bar graph of each four features separately.

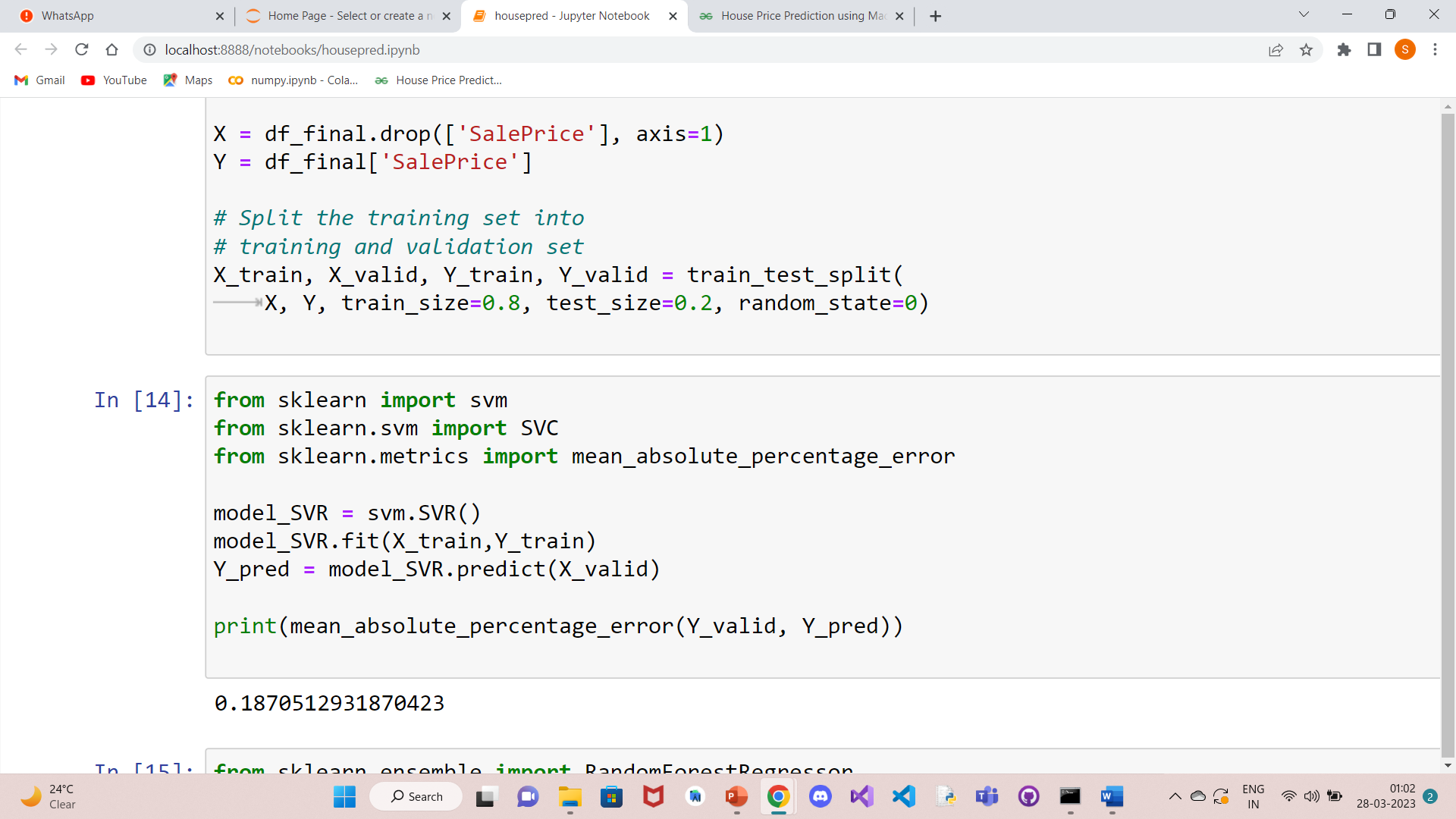
### 

### Checking features which have null values in the new dataframe (if there are still any).

### 

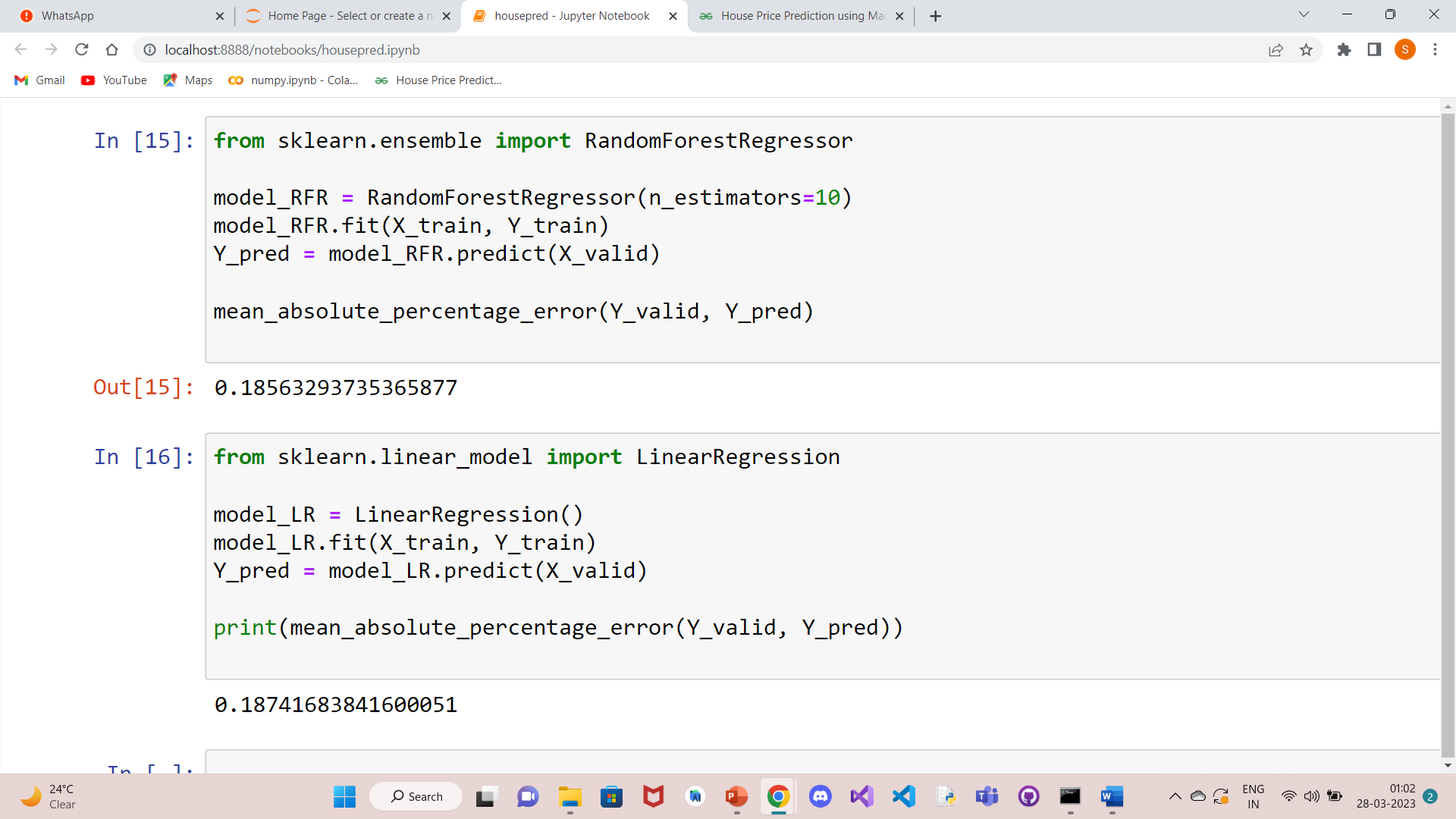
### Fig:Null values

**Model Training**

**SUPPORT VECTOR MACHINE**

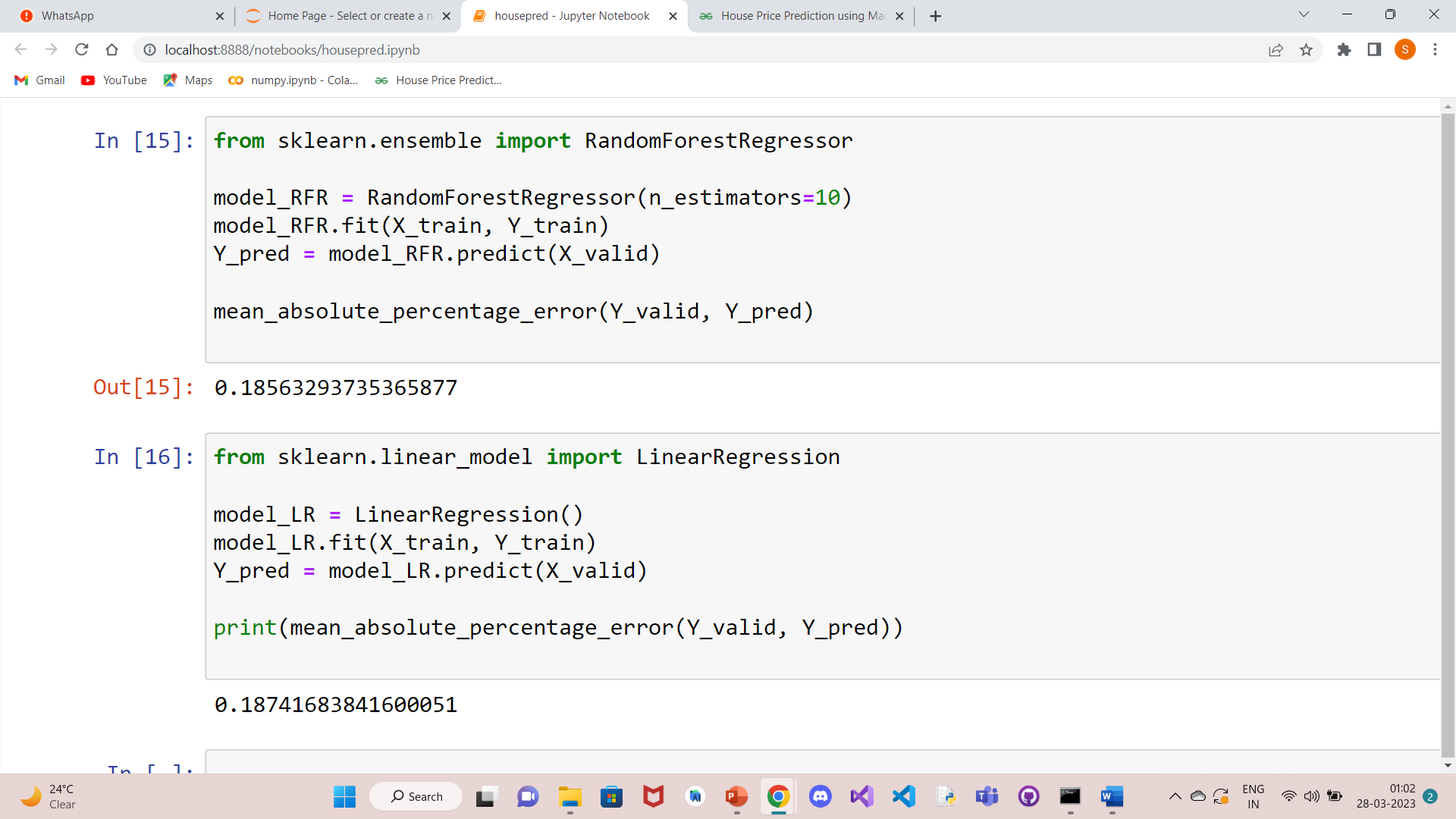
**Fig:6.5 Support vector machine**

**RANDOM FOREST REGRESSOR**

****

**Fig:6.6 Random forest regressor**

**LINEAR REGRESSION**

****

**Fig:6.6 Linear Regress****ion**

**7.1 Source Code**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

dataset = pd.read\_excel("HousePricePredictionn.xlsx")

# Printing first 5 records of the dataset

print(dataset.head(5))

dataset.shape

obj = (dataset.dtypes == 'object')

object\_cols = list(obj[obj].index)

print("Categorical variables:",len(object\_cols))

int\_ = (dataset.dtypes == 'int')

num\_cols = list(int\_[int\_].index)

print("Integer variables:",len(num\_cols))

fl = (dataset.dtypes == 'float')

fl\_cols = list(fl[fl].index)

print("Float variables:",len(fl\_cols))

plt.figure(figsize=(12, 6))

sns.heatmap(dataset.corr(),

cmap = 'BrBG',

fmt = '.2f',

linewidths = 2,

annot = True)

unique\_values = []

for col in object\_cols:

unique\_values.append(dataset[col].unique().size)

plt.figure(figsize=(10,6))

plt.title('No. Unique values of Categorical Features')

plt.xticks(rotation=90)

sns.barplot(x=object\_cols,y=unique\_values)

unique\_values = []

for col in object\_cols:

unique\_values.append(dataset[col].unique().size)

plt.figure(figsize=(10,6))

plt.title('No. Unique values of Categorical Features')

plt.xticks(rotation=90)

sns.barplot(x=object\_cols,y=unique\_values)

dataset.drop(['Id'],

axis=1,

inplace=True)

dataset['SalePrice'] = dataset['SalePrice'].fillna(

dataset['SalePrice'].mean())

new\_dataset = dataset.dropna()

new\_dataset.isnull().sum()

from sklearn.preprocessing import OneHotEncoder

\_encoder = OneHotEncoder(sparse=False)

OH\_cols = pd.DataFrame(OH\_encoder.fit\_transform(new\_dataset[object\_cols]))

OH\_cols.index = new\_dataset.index

OH\_cols.columns = OH\_encoder.get\_feature\_names\_out()

df\_final = new\_dataset.drop(object\_cols, axis=1)

df\_final = pd.concat([df\_final, OH\_cols], axis=1)

from sklearn.metrics import mean\_absolute\_error

from sklearn.model\_selection import train\_test\_split

X = df\_final.drop(['SalePrice'], axis=1)

Y = df\_final['SalePrice']

# Split the training set into

# training and validation set

X\_train, X\_valid, Y\_train, Y\_valid = train\_test\_split(

X, Y, train\_size=0.8, test\_size=0.2, random\_state=0)

**SUPPORT VECTOR MACHINE**

from sklearn import svm

from sklearn.svm import SVC

from sklearn.metrics import mean\_absolute\_percentage\_error

model\_SVR = svm.SVR()

model\_SVR.fit(X\_train,Y\_train)

Y\_pred = model\_SVR.predict(X\_valid)

print(mean\_absolute\_percentage\_error(Y\_valid, Y\_pred))

**RANDOM FOREST REGRESSOR**

from sklearn.ensemble import RandomForestRegressor

model\_RFR = RandomForestRegressor(n\_estimators=10)

model\_RFR.fit(X\_train, Y\_train)

Y\_pred = model\_RFR.predict(X\_valid)

mean\_absolute\_percentage\_error(Y\_valid, Y\_pred))

**LINEAR REGRESSION**

from sklearn.linear\_model import LinearRegression

model\_LR = LinearRegression()

model\_LR.fit(X\_train, Y\_train)

Y\_pred = model\_LR.predict(X\_valid)

print(mean\_absolute\_percentage\_error(Y\_valid, Y\_pred))

**CHAPTER 8**

# Conclusion

The project is essential for making strategic decisions. The database maintained is updated on every prediction and the system works efficiently with massive dataset of thousands of rows. It is able to create a model that can help industry producers, distributors and sellers predict the house prices and have better understanding of each critical and up-to-date feature. It’s observed that model feature sets performed better than others. In general, model is the best model for prediction and it works very well on small sized datasets. First, the main problem came from the fact that the data set was unstructured. Loading the dataset to pandas dataframe structured the data. Statistical measures of the dataframe were found which gave the overview of the dataset. From the heat map it was found out that the model is working perfectly fine as the error values of testing and training data were very small when compared. Finally, data analysis and visualization was done to see how close the testing and training data values were and they were pretty close with minimal error values. In the future, the user can try other performance measures and other machine learning techniques for better performance and comparison of results. This analysis will help businesses predict the house prices based on various features to help in buying and selling the houses.

# References

[1] Hamizah Zulkifley, Shuzlina Abdul Rahman,Hasbiah Ubaidullah, Ismail Ibrahim,House Price Prediction using a Machine Learning Model: A Survey of Literature I.J. Modern Education and Computer Science, 2020, 6, 46-54, Published Online December 2020 in MECS, DOI: 10.5815/ijmecs.2020.06.04.

[2] Smith Dabreo, Shaleel Rodrigues, Valiant Rodrigues, Parshvi Shah, Real Estate Price Prediction, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 10 Issue 04, April-2021.

[3] ALISHA KUVALEKAR, SHIVANI MANCHEWAR, SIDHIKA MAHADIK, SHILA JAWALE,House Price Forecasting Using Machine Learning, Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST) 2020.

[4] Byeonghwa Park, Jae Kwon Bae “Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data”. 2017.

[5] Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh - “A hybrid regression technique for house prices prediction” 2017, IEEE.

[6] Kuvalekar, Alisha and Manchewar, Shivani and Mahadik, Sidhika and Jawale, Shila, House Price Forecasting Using Machine Learning (April 8, 2020). Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST) 2020.

[7] Sayan Putatunda,"PropTech for Proactive Pricing of Houses in Classified Advertisements in the Indian Real Estate Market".

[8] Thuraiya Mohd, Suraya Masrom, Noraini Johari, "Machine Learning Housing Price Prediction in Petaling Jaya, Selangor, Malaysia ", International Journal of Recent Technology and Engineering (IJRTE), Volume-8, Issue-2S11, 2019.