

HOUSE VALUE PREDICTION USING MACHINE LEARNING

INTERDISCIPLINARY PROJECT

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

THOTA SAHITHI (Reg. No – 41111259)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY - 1 UNIVERSITY BY UGC

Accredited "A++" by NAAC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL – 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **THOTA SAHITHI (41111259)** who carried out the Project entitled "**HOUSE VALUE PREDICTION USING MACHINE LEARNING**" under my supervision from January 2024 to April 2024.

Internal Guide

Dr. C. HEMALATH MAM

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Interdisciplinary Viva Voce Examination held on __

**Internal Examiner
Examiner**

External

DECLARATION

I, **THOTA SAHITHI (Reg. No- 41111259)**, hereby declare that the Project Report entitled “**HOUSE VALUE PREDICTION USING MACHINE LEARNING**” done by me under the guidance of **DR. C. HEMALATHA MAM**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph. D., Dean**, School of Computing, and **Dr. L. Lakshmanan, M.E., Ph.D., Head of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **DR.C. HEMALATHA MAM**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

CERTIFICATE



ABSTRACT

Now-a-days everyone wish to live in the large cities but the competition in the market related to all the resources is increasing day by day. A middle-class family can't afford the price of rent, food, water and electricity while surviving his family. The price of the flats in the city is increasing and there is so much of risk to predict the actual price of the house. Our research paper [1] will helps you to predict the price of the house to a good accuracy. The main motive of our research paper is to predict the price [2] of the house by analysing the customer needs and their financial income. As we see when a client wants to purchase the house in the city, he used to see three things within the city location, area and available resources around the society. Our research paper will help the clients to know the actual price of the house and it will also help the builders to know about the selling price that will fits the client needs. House price forecasting is an important topic of real estate. The literature attempts to derive useful knowledge from historical data of property markets. Machine learning techniques are applied to analyse historical property transactions in India to discover useful models for house buyers and sellers. Revealed is the high discrepancy between house prices in the most expensive and most affordable suburbs in the city of Mumbai. Moreover, experiments demonstrate that the Multiple Linear Regression that is based on mean squared error measurement is a competitive approach.

TABLE OF CONTENTS

CHAPTER NO	TITTLE	PAGENO
	ABSTRACT	vi
	LIST OF FIGURES	vii
	LIST OF TABLES	viii
1	INTRODUCTION	9
2	LITERATURE SURVEY	
	2.1 REVIEW ON EXIXTING SYSTEM	10
	2.2 CHALLAENGES ON EXISTING SYSTEM	11
3	ANALYSIS AND DESIGN OF PROPOSED SYSTEM	
	3.1 NECESSITY FOR PROPOSED SYSTEM	12
	3.2 HARDWARE AND SOFTWARE REQUIREMENTS	13
	3.3 ARCHITECTURE DESIGN	14
4	IMPLEMENTATIONS OF PROPOSED SYSTEM	
	4.1 DESCRIPTION OF DATASET	15
	4.2 EMPLOYED MODULES	16
	4.3 DETAILED DESCRIPTION OF PROPOSED SYSTEM	17
	4.4 ALOGORITHM USED	19-21
	4.5 CODING	22-25
5	RESULTS AND DISCUSSION	
	5.1SCREENSHOTS	26-33
	5.2 RESULT	
6	CONCLUSION AND FUTURE ENHANCEMENT	
	REFERENCES	

LIST OF FIGURES

FIGURENO NO	FIGURENAME	PAGE
3.3	ARCHITECTURE DESIGN	14
4.3	DETAILED DESCRIPTION OF PROPOSED SYSTEM	18
4.4	ALGORITHM USED	20
4.4	ALGORITHM USED	21
5.2	RESULT	33

CHAPTER 1

INTRODUCTION

Machine learning is a subfield of artificial intelligence (AI). The objective of AI by and large is to comprehend the construction of information and fit that information into models that can be perceived and used by individuals. In spite of the fact that AI is a field inside software engineering, it contrasts from conventional computational methodologies. In conventional registering, calculations are sets of unequivocally customized guidelines utilized by PCs to figure or issue settle. AI calculations rather take into account PCs to prepare on information data sources and utilize measurable examination to yield esteems that fall inside a particular reach. Along these lines, AI encourages PCs in building models from test information to robotize dynamic cycles dependent on information inputs. Demonstrating utilizes AI calculations, where machine gains from the information and utilizations them to anticipate another information. The most every now and again utilized model for prescient investigation is relapse. AI in software engineering endeavours to tackle issues algorithmically as opposed to absolutely numerically. Hence, it depends on making calculations that grant the machine to learn. In any case, there are two general gatherings in AI which are regulated and solo. Administered is the place where the program gets prepared on pre-decided set to have the option to foresee when another information is given. Solo is the place where the program attempts to discover the relationship and the secret example between the information. A near report was completed with assessment measurements also. When we get a solid match, we can utilize the model to figure financial estimation of that specific lodging property in Bengaluru. The exhibition will be estimated after foreseeing house costs since the forecast in numerous relapse calculations depends on a particular element as well as on an obscure number of properties that bring about the worth to be anticipated. House costs rely upon an individual house detail. Houses have a variation number of highlights that might not have a similar expense because of its area. For example, a major house may have a greater cost on the off chance that it is situated in attractive rich territory than being put in a helpless area. The information utilized in the trial will be dealt with by utilizing a blend of pre-preparing strategies to improve the forecast

precision. Be that as it may, there are two general gatherings in AI which are administered and unaided. Regulated is the place where the program gets prepared on pre-decided set to have the option to anticipate when another information is given. Unaided is the place where the program attempts to discover the relationship and the secret example between the information.

CHAPTER 2

LITERATURE SURVEY

Sarip and Hafez [1] Fuzzy logic is a structured model proposed for house price - prediction, offering a unique approach to handling uncertainty and imprecision in data in real estate markets. It allows for a more flexible and nuanced analysis of housing price trends. Fuzzy logic can handle vague and ambiguous data effectively, capturing the subjective nature of human decision-making processes. It can adapt well to changing market conditions and accommodate a wide range of input variables, making it versatile for predicting house prices. However, prior research has limitations such as complexity, black-box nature, and the need for expertise in real estate economics and fuzzy logic. Yuqing He's [2] research paper presents a Polynomial Linear Prediction Model for US Housing Prices, focusing on approximating house prices without individual prices and identifying key determinants of housing values. The model uses multiple linear regression and polynomial modelling to provide accurate predictions based on various features and attributes. The study offers a novel approach to predicting house prices and aims to understand key determinants of housing values. However, it faces limitations such as limited comparison with other models and potential challenges in interpretability. Further exploration of model transparency could improve its practical applicability.

2.1 REVIEW ON EXISITING SYSTEM:

Multi Linear Regression Multiple Linear Regression. It shows the relationship between two or more explanatory variables and scalar response variable.

Independent variable value is associated with dependent variable value Limitations
The dependent variable y must be continuous. The independent variables can be of any type. The dependent variable is usually dependent on independent variables.

2.2 CHALLENGENS ON EXISITING SYSTEM:

1. **Data Quality:** One of the primary challenges is ensuring the quality and reliability of the data used for training the predictive models. Inaccurate or incomplete data can lead to biased or inaccurate predictions.
2. **Feature Selection:** Identifying the most relevant features that influence house prices is crucial. However, selecting the right set of features from a vast pool of potential variables can be challenging and often requires domain expertise.
3. **Model Complexity:** Balancing model complexity with interpretability is essential. Complex models may capture intricate patterns in the data but might be difficult to interpret and prone to overfitting, especially with limited data.
4. **Spatial Variability:** Housing markets can vary significantly based on location. Models trained on data from one region may not generalize well to other regions due to differences in factors such as demographics, economic conditions, and local regulations.
5. **Temporal Dynamics:** Housing markets are dynamic, and factors influencing house prices can change over time. Models need to account for these temporal dynamics to make accurate predictions, which may require continuous retraining and updating.
6. **Outliers and Anomalies:** Outliers and anomalies in the data can significantly impact model performance if not handled appropriately. Robust preprocessing techniques are needed to identify and handle such instances effectively.
7. **Model Evaluation:** Assessing the performance of predictive models for house value prediction requires careful consideration of evaluation metrics. Metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE) need to be chosen wisely based on the

specific requirements of the problem.

8. **Ethical Considerations:** Predictive models in real estate can inadvertently perpetuate or exacerbate existing biases, such as discrimination in housing. Ensuring fairness and equity in model predictions is essential to mitigate these ethical concerns.
9. **Data Privacy and Security:** Housing data often contains sensitive information about individuals. Ensuring data privacy and security throughout the data lifecycle, from collection to storage to analysis, is crucial to maintain trust and compliance with regulations such as GDPR or CCPA.

CHAPTER 3

ANALYSIS AND DESIGN OF PROPOSED SYSTEM

3.1 PROPOSED SYSTEM:

Linear Regression is a technique that helps to identify the relationship between a dependent variable and independent variable. The regression technique that we used here is linear regression.

3.1.1 Advantages:

- Space complexity is very low it just needs to save the weights at the end of training. hence, it's a high latency algorithm.
- It's very simple to understand
- Good interpretability
- Feature importance is generated at the time model
- building. With the help of hyperparameter Lambda, you can handle features selection hence we can achieve dimensionality reduction

3.1.2 Proposed System Design:

In this project work, I used five modules and each module has own functions, such as:

1. Dataset
2. Exploratory Data Analysis
3. Data Cleaning
4. Train Test Split
5. Algorithm

3.2 HARDWARE AND SOFTWARE REQUIREMENTS:

3.2.1 Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list, especially in case of operating systems. The minimal hardware requirements are as follows,

1. PROCESSOR: PENTIUM IV
2. RAM: 8 GB
3. PROCESSOR: 2.4 GHZ
4. MAIN MEMORY: 8GB RAM
5. PROCESSING SPEED: 600 MHZ
6. HARD DISK DRIVE: 1TB
7. KEYBOARD :104 KEYS

3.2.2 Software requirements:

deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. These requirements are need to be installed separately. The minimal software requirements are as follows,

- FRONT END: PYTHON
- IDE: ANACONDA
- OPERATING SYSTEM: WINDOWS 10

3.3 ARCHITECTURE DESIGN:

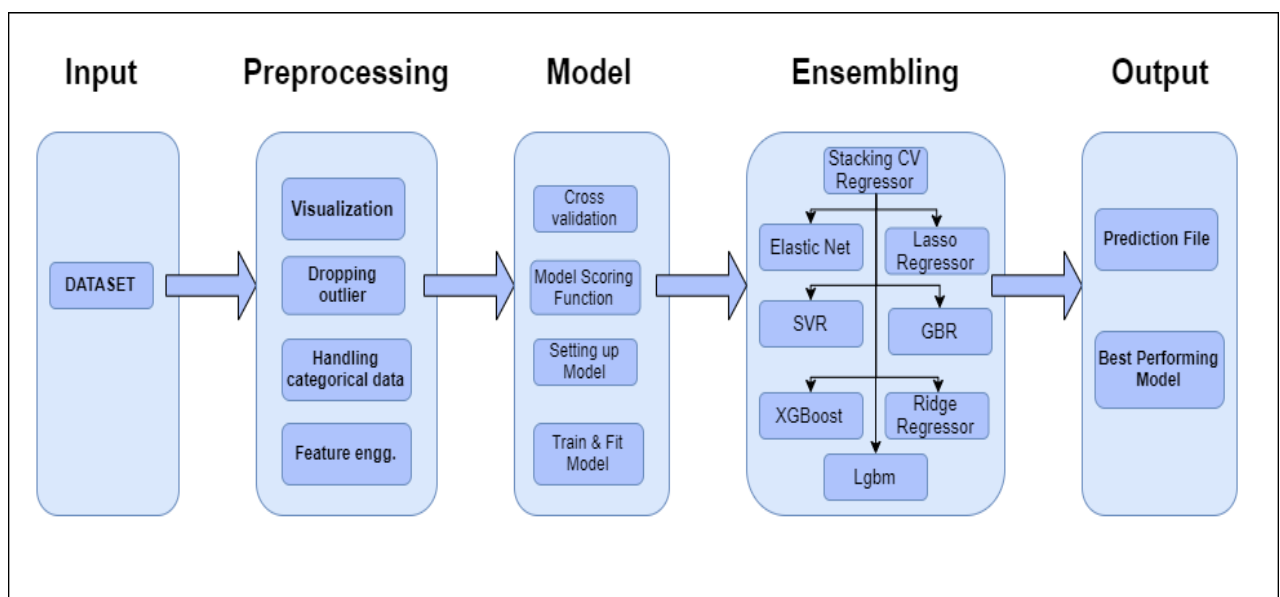


Figure 1

3.3.1 Description of The Architecture:

- Sample dataset is collected
- Sample Data is loaded.
- It determines the dependent value it is nothing but the value that is being

dependant on other values here the dependent value is price

- It also determines the independent values it is nothing but the value that does not depend on other value here square feet, area, no of bedrooms bathrooms are independent value
- Using linear regression it will calculate the variables.
- When a user determines the requirements, it will predict and shows the results.

CHAPTER 4

IMPLEMENTATION OF PROPOSED SYSTEM

4.1 DESCRIPTION OF THE DATASET:

When it comes to making one of the most significant investments of your life, buying a house, the stakes are high. Factors like location, size, and price play a pivotal role in making the right decision. In this data science project, I aimed to predict house prices in Bangalore using a dataset containing essential features such as location, size, total square footage, bedrooms, bathrooms, and, of course, the target variable — price. Along the way, I encountered the challenge of outliers and utilized feature engineering to enhance the model's performance. Let's explore this exciting journey.

Objective: To develop a machine learning model for predicting home prices based on input features such as location, no of bedrooms, no of bathrooms, square area feet and surge pricing.

Tools and Technologies: Python, Pandas, Scikit-Learn, Matplotlib

Understanding the Dataset

Dataset Name: banglore_home_price.csv

Dataset Source: Kaggle

Our journey begins with understanding the dataset. The data comprises several columns, each representing a crucial aspect of a house:

Location: The neighbourhood or area where the house is located.

Total Sq. Ft: The total square footage of the property.

Bedrooms: The number of bedrooms in the house in Terms of BHK

Bathrooms: The number of bathrooms in the house.

Price: The target variable, which is the price of the house.

4.2 EMPLOYED MODULES:

1.Collection of Dataset:

The dataset used in this project was Parameters such as Area in square meters, Location, no of bedrooms and no of bathrooms in that particular property. Selling price is a dependent variable on several other independent variables.

2 Data Preprocessing:

It is a process of transforming the raw, complex data into systematic understandable knowledge. It will find out missing and redundant data in the dataset. Thus, this brings uniformity in the dataset. But in our dataset, there was no missing values.

3. Import Libraries:

A library is a collection of modules the first step is to import the libraries that we require in our system. There are functions for them, which can be invoked without writing the required code. This is a list for most popular Python libraries for Data Science. We have imported panda's library and named it as pd.

4 Import the Dataset:

A lot of datasets come in CSV formats. At first, we have to locate directory of csv file and read it using a method called `read_csv` which may be found in the library called `pandas`.

5 Encoding categorical data:

Sometimes we have texts as our data. We can find categories in text form. Now it gets tougher for machines to know texts and process them, hence we are changing them to numbers. Therefore, we have to encode the categorical data.

6 Split Dataset into Training and Test:

Set Now we should split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our loaded data training set, i.e. our machine learning models will understand the relationships in our training set and then we will test the models on our test set to check how it predicts. In general, we need to allocate 80% of the dataset to training set and the remaining 20% to test set.

7 Dependent and independent variables in regression:

Regression analysis is used to describe the relationship between dependent variables and independent variables. It predicts value of dependent variable by analysing the value of independent variables.

8 Prediction:

Prediction is nothing but the output of an algorithm after being trained on a dataset and applied to new data and predicts the output. Finally, our model will predict the house price based on user inputs.

4.3 DETAILED DESCRIPTION OF THE PROPOSED SYSTEM:

- Linear Regression is a supervised machine learning model that attempts to

model a linear relationship between dependent variables (Y) and independent variables (X). Every evaluated observation with a model, the target (Y)'s actual value is compared to the target (Y)'s predicted value, and the major differences in these values are called residuals. The Linear Regression model aims to minimize the sum of all squared residuals. Here is the mathematical representation of the linear regression:

$$Y = a_0 + a_1X + \varepsilon$$

- The values of X and Y variables are training datasets for the model representation of linear regression. When a user implements a linear regression, algorithms start to find the best fit line using a_0 and a_1 . In such a way, it becomes more accurate to actual data points; since we recognize the value of a_0 and a_1 , we can use a model for predicting the response.

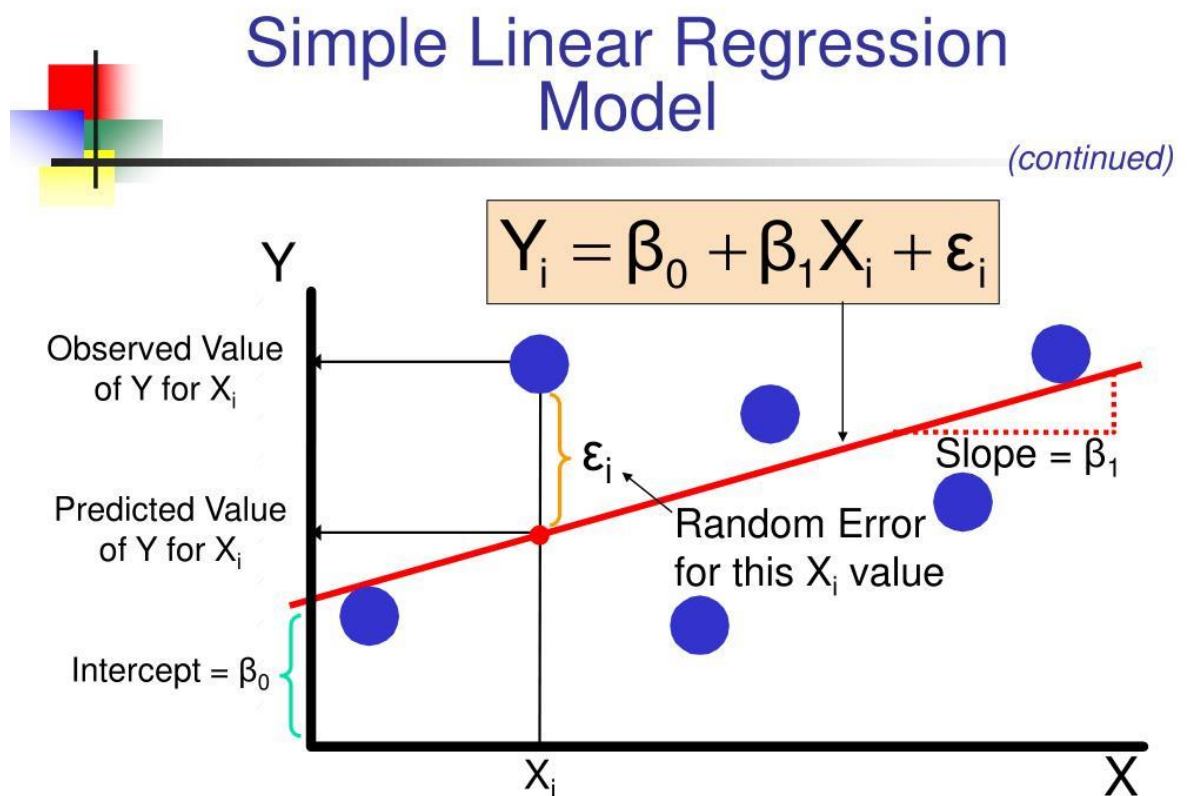


Figure 2

- As you can see in the above diagram, the red dots are observed values for both X and Y.
- The black line, which is called a line of best fit, minimizes a sum of a squared error.
- The blue lines represent the errors; it is a distance between the line of best fit and observed values.
- The value of the a_1 is the slope of the black line.

4.4 ALGORITHMS USED:

1. BASIC LINEAR MODEL:

The formulation for multiple regression model is $Y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n$
the assumptions in the model are:

- The error terms are normally distributed.
- The error terms have constant variance.
- The model carries out a linear relationship between the target variable and the functions.

Here the different relapse models are created by the most un-square methodology (Ordinary Least Squares/OLS). The precision of the planned model is hard to gauge without assessing its yield on both train and test informational collections. This can be accomplished utilizing effectiveness metric or some likeness thereof. It could be by estimating some sort of blunder, fit's integrity, or some other valuable computation. For this investigation, we assessed model's presentation utilizing measurements: the coefficient of assurance, R^2 and RMSE (Root Means Square Error). (Root Means Square Error), RMLSE (Root Mean Squared Logarithmic Error)

RMSE: It can be characterized as the standard example deviation between the anticipated qualities and the noticed ones. It is to be noticed that unit of RMSE is same as reliant variable y. The lower RMSE esteems are characteristic of a superior fit model. On the off chance that the model's essential target is forecast, RMSE is a more grounded measure. 2. R-squared and Adjusted R-squared: The

R-square worth gives a proportion of how much the model recreates the real outcomes, in light of the proportion of all out variety of results as clarified in the model.

$$RMSE = \sqrt{\frac{\sum_i^N (x_i - \hat{x}_i)^2}{N}}$$

2.RIDGE REGRESSION:

Ridge regression model is a regularization model, where an extra variable (tuning parameter) is added and optimized to address the effect of multiple variables in linear regression which is usually referred as noise in statistical context. In mathematical form, the model can be expressed as

$$y = bx + e$$

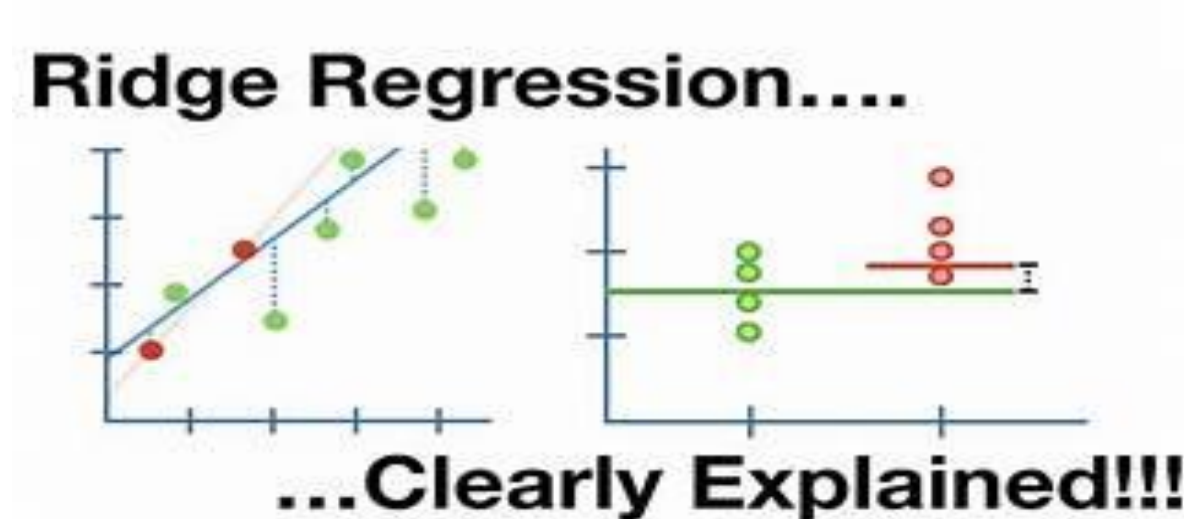


Figure 3

Here, y is the needy variable x alludes to highlights in network structure and b alludes to relapse coefficients and e addresses residuals. In light of this, the factors are normalized by taking away the particular factors and separating them by their

standard deviations. The tuning capacity indicated as λ is then appeared as part of regularization in the edge relapse model. In the event that λ 's worth is enormous, the squares ' leftover total gives off an impression of being zero. In the event that it is not exactly the arrangements adjust to least square strategy. λ is discovered utilizing a strategy called cross - approval. Edge relapse diminishes the coefficients to self-assertively low qualities however not to nothing. We will likewise perform matrix search cross-approval to tune the regularization hyper boundary λ . We have picked wide reach for hyper boundary and discovered 0.001

3. LASSO REGRESSION:

LASSO implies least total shrinkage, and the choice administrator is a LR method that likewise regularizes usefulness. It is indistinguishable from edge relapse, then again, actually it differs in the estimations of regularization. The outright estimations of the amount of relapse coefficients are contemplated. It even sets the coefficients to nothing so it totally decreases the mistakes. So determination of highlights are come about by rope relapse. In the recently referenced edge condition, the part ' e ' has outright qualities rather than squared qualities. It is to be noticed that computationally Lasso relapse strategy is definitely more concentrated than Ridge relapse procedure. We have performed lattice search cross-approval to tune the regularization hyper boundary λ . We have picked wide reach for hyper-boundaries and discovered 0.001 as best worth. The model got has a variety of coefficients: Cluster ([2.26872602e-02, 3.84815301e-05, 1.88127460e-01, 6.35927974e-02, - 0.00000000e+00, 4.65240926e-01, - 5.40347504e-02, 2.12552087e-01, 1.13374140e-01, 3.49669654e-07])

Lasso Regression....



...Clearly Explained!!!

Figure 4

4.5 CODING:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams['figure.figsize'] = (15,7)
df = pd.read_csv("Bengaluru_House_Data.csv")
df.head()
list(df.columns)
df['area_type'].value_counts()
f = df.drop(['area_type','society','balcony','availability'],axis='columns')
df.head()
df.isnull().sum()
df = df.dropna()
df.isnull().sum() # check there is no null value present
df['size'].unique()
#Creating a BHK column from size
extract_num = lambda x: int(x.split(' ')[0])
df['BHK'] = df['size'].apply(extract_num)
df.head(5)
df[df['BHK']>15]
df['total_sqft'].unique()
def is_float(x)
    try:
        float(x)
    except:
        return False
    return True
```

```

def convert_str_value_to_num(x):
    value = x.split('-')
    if len(value) == 2: return (float(value[0])+float(value[1]))/2
    try:
        return float(x)
    except:
        return None

df[~df['total_sqft'].apply(is_float)].head()
df.total_sqft = df.total_sqft.apply(convert_str_value_to_num)
df = df[df.total_sqft.notnull()]
df.head(5)
df['price_per_sqft'] = df['price'] * 100000 / df['total_sqft']
df.head()
df['price_per_sqft'].describe()
df.to_csv("BHP_cleaned.csv",index=False)
df.location = df.location.apply(lambda x: x.strip())
df_location_stat = df['location'].value_counts(ascending=False)
df['location'].unique()
location_stats_less_10 = df_location_stat[df_location_stat<=10]
location_stats_less_10
df[df.total_sqft/df.BHK<300].head()
df = df[~(df.total_sqft/df.BHK<300)]
df.price_per_sqft.describe()
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) &
(subdf.price_per_sqft<=(m+st))]
        df_out = pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df = remove_pps_outliers(df)

```

```

df.head()

def plot_rates_for_BHK(df,location):
    bhk2 = df[(df.location==location) & (df.BHK==2)]
    bhk3 = df[(df.location==location) & (df.BHK==3)]
    bhk4 = df[(df.location==location) & (df.BHK==4)]
    plt.scatter(bhk2.total_sqft,bhk2.price , color='blue',label='2 BHK', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price , marker='+', color='green',label='3 BHK',
s=50)
    plt.scatter(bhk4.total_sqft,bhk4.price , color='red' , marker='x',label='4 BHK',
s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()

list(df.location.unique())
plot_rates_for_BHK(df,"Hebbal")
plot_rates_for_BHK(df,"Whitefield")
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('BHK'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('BHK'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices,
bhk_df[bhk_df.price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')

```



```

f = remove_bhk_outliers(df)
plot_rates_for_BHK(df,"Rajaji Nagar")
plot_rates_for_BHK(df,"Hebbal")
plt.hist(df.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
df = df[df.bath<df.BHK+2]
df.head()
dummies = pd.get_dummies(df.location)
dummies.head(5)
df = pd.concat([df,dummies.drop('other',axis='columns')],axis='columns')
df.head()
df = df.drop(['location','size','price_per_sqft'],axis='columns')
df.head(5)
df.price
f.to_csv("ModeifiedCleanedData.csv",index=False)
feature = df.drop(['price'],axis='columns')
label = df.price
from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(feature , label , test_size = 0.2
, random_state = 10)
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,y_train)
def predict_price(location,sqft,bath,bhk):
    loc_index = np.where(feature.columns==location)[0][0]
    x = np.zeros(len(feature.columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

```

```

        return model.predict([x])[0]
predict_price('Indira Nagar',1000, 3, 3)
import pickle
with open('model.pickle','wb') as f:
    pickle.dump(model,f)
import json
columns = {
    'data_columns' : [col.lower() for col in feature.columns]
}
with open("columns.json","w") as f:
    f.write(json.dumps(columns))

```

CHAPTER 5

RESULTS AND DISCUSSION

First, I will import the conditions, that will make this program somewhat simpler to compose. I'm bringing in the AI library sklearn, numpy, and pandas. I will stack the Housing Data Set from learn. datasets and print it subsequent to putting away it into the variable dataset by utilizing a capacity/technique called load data (). however, first I will import the library sklearn. datasets. I will part the information into autonomous factors (X's) and ward variable (Y)data sets. I will feel free to instate the Linear Regression model, split the information into 67% preparing and 33% testing information, and afterward train the model with the preparation informational collection that contains the autonomous

5.1 SCREENSHOTS:

1. Import Libraries:

```
In [2]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
```

```
In [3]: import matplotlib
matplotlib.rcParams['figure.figsize'] = (15,7)
```

Read Data from CSV file

```
In [4]: df = pd.read_csv("Bengaluru_House_Data.csv")
```

```
In [5]: df.head()
```

```
Out[5]:
```

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

2. Data Preprocessing:

Data Processing

```
In [6]: list(df.columns)
```

```
Out[6]: ['area_type',
'availability',
'location',
'size',
'society',
'total_sqft',
'bath',
'balcony',
'price']
```

Count each AreaType

```
In [7]: df['area_type'].value_counts()
```

```
Out[7]: Super built-up Area    8790
Built-up Area                2418
Plot Area                    2025
Carpet Area                   87
Name: area_type, dtype: int64
```

Drop features that are not required to build our model

```
In [8]: df = df.drop(['area_type', 'society', 'balcony', 'availability'], axis='columns')
```

```
In [9]: df.head()
```

3. Data Cleaning:

```
Out[9]:
```

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00

Data Cleaning: Handle null values

```
In [10]: df.isnull().sum()

Out[10]: location      1
         size         16
         total_sqft    0
         bath         73
         price         0
         dtype: int64

In [11]: df = df.dropna()
         df.isnull().sum() # check there is no null value present

Out[11]: location      0
         size         0
         total_sqft    0
         bath         0
         price         0
         dtype: int64
```

4. EDA & Feature Engineering:

EDA & Feature Engineering

```
In [12]: df['size'].unique()

Out[12]: array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',
                '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',
                '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',
                '9 BHK', '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',
                '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
                '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

Splitting Text and values in different columns

```
In [13]: #Creating a BHK column from size
         extract_num = lambda x: int(x.split(' ')[0])
         df['BHK'] = df['size'].apply(extract_num)
```

```
In [14]: df.head(5)
```

```
Out[14]:
```

	location	size	total_sqft	bath	price	BHK
0	Electronic City Phase II	2 BHK	1056	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00	4
2	Uttarahalli	3 BHK	1440	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00	3
4	Kothanur	2 BHK	1200	2.0	51.00	2

```
In [15]: df[df['BHK']>15]
```

```
Out[15]:
```

	location	size	total_sqft	bath	price	BHK
1718	2Electronic City Phase II	27 BHK	8000	27.0	230.0	27
3379	1Hanuman Nagar	19 BHK	2000	16.0	490.0	19
3609	Koramangala Industrial Layout	16 BHK	10000	16.0	550.0	16
4684	Munnekollal	43 Bedroom	2400	40.0	660.0	43
11559	1Kasavanhalli	18 Bedroom	1200	18.0	200.0	18

```
In [16]: # there was problem in 4684 if BHF is 43 and size is 43 quite different so we drop the column
df = df.drop(4684)
```

Perform EDA in total_sqft column

```
In [17]: df['total_sqft'].unique()
```

```
Out[17]: array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

```
In [18]: def is_float(x):
      try:
          float(x)
      except:
          return False
      return True
```

Calculate Pricee per square feet and add column

```
In [21]: df['price_per_sqft'] = df['price'] * 100000 / df['total_sqft']
df.head()
```

```
Out[21]:
```

	location	size	total_sqft	bath	price	BHK	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000

```
In [22]: df['price_per_sqft'].describe()
```

```
Out[22]: count    1.319900e+04
mean       7.919276e+03
std        1.067311e+05
min        2.678298e+02
25%        4.267620e+03
50%        5.438066e+03
75%        7.317073e+03
max        1.200000e+07
Name: price_per_sqft, dtype: float64
```

```
In [23]: df.to_csv("BHP_cleaned.csv",index=False)
```

Location

```
In [24]: df.location = df.location.apply(lambda x: x.strip())
df_location_stat = df['location'].value_counts(ascending=False)

In [25]: df['location'].unique()

Out[25]: array(['Electronic City Phase II', 'Chikka Tirupathi', 'Uttarahalli', ...,
                '12th cross srinivas nagar banshankari 3rd stage',
                'Havanur extension', 'Abshot Layout'], dtype=object)

In [26]: location_stats_less_10 = df_location_stat[df_location_stat<=10]
location_stats_less_10

Out[26]: BTM 1st Stage          10
Gunjur Palya                  10
Nagappa Reddy Layout          10
Sector 1 HSR Layout           10
Thyagaraja Nagar              10
..
Rajanna Layout                1
Subramanyanagar               1
Lakshmipura Vidyaanyapura      1
Malur Hosur Road              1
Abshot Layout                 1
Name: location, Length: 1047, dtype: int64

In [27]: df.location = df.location.apply(lambda x: 'other' if x in location_stats_less_10 else x)
```

5. Visualize Data:

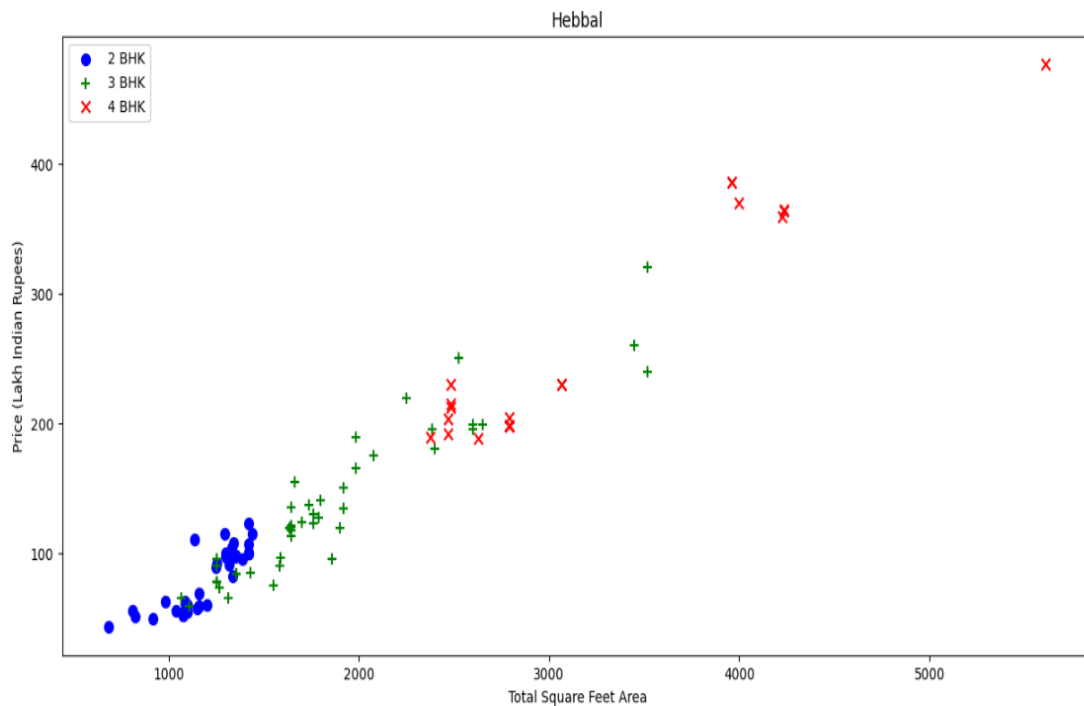
Visualize Data for Different location prize for BHK

```
In [32]: def plot_rates_for_BHK(df,location):
bhk2 = df[(df.location==location) & (df.BHK==2)]
bhk3 = df[(df.location==location) & (df.BHK==3)]
bhk4 = df[(df.location==location) & (df.BHK==4)]
plt.scatter(bhk2.total_sqft,bhk2.price , color='blue',label='2 BHK', s=50)
plt.scatter(bhk3.total_sqft,bhk3.price , marker='+', color='green',label='3 BHK', s=50)
plt.scatter(bhk4.total_sqft,bhk4.price , color='red' , marker='x',label='4 BHK', s=50)
plt.xlabel("Total Square Feet Area")
plt.ylabel("Price (Lakh Indian Rupees)")
plt.title(location)
plt.legend()

In [33]: list(df.location.unique())

Out[33]: ['1st Block Jayanagar',
          '1st Phase JP Nagar',
          '2nd Phase Judicial Layout',
          '2nd Stage Nagarbhavi',
          '5th Block Hbr Layout',
          '5th Phase JP Nagar',
          '6th Phase JP Nagar',
          '7th Phase JP Nagar',
          '8th Phase JP Nagar',
          '9th Phase JP Nagar',
          'AECS Layout',
          'Abbigere',
          'Akshaya Nagar',
          'Ambalipura',
          'Ambedkar Nagar',
          'Amruthahalli',
```

```
In [34]: plot_rates_for_BHK(df,"Hebbal")
```

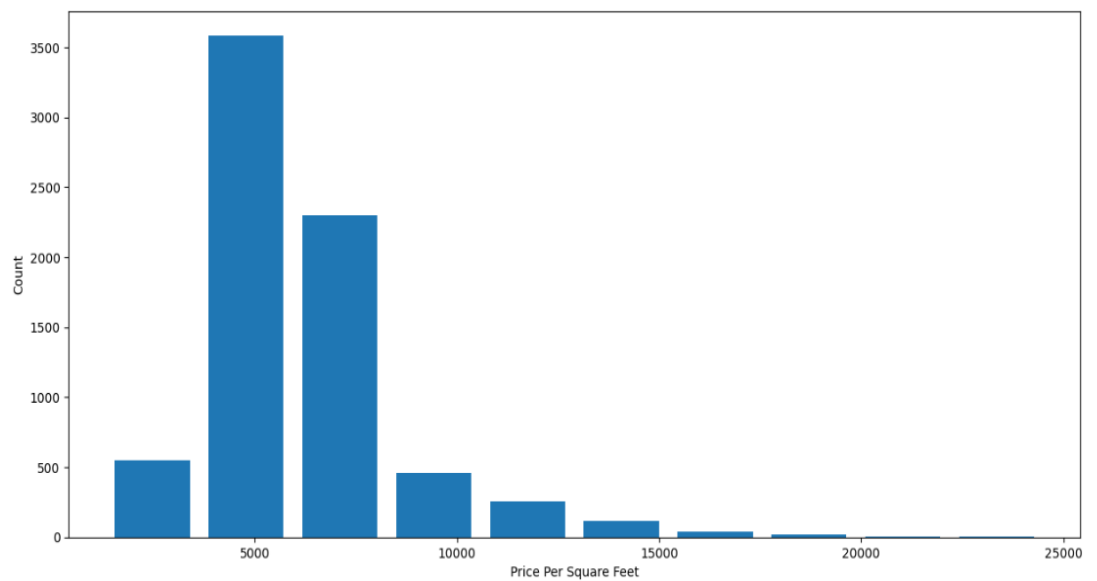


```
In [35]: plot_rates_for_BHK(df,"Whitefield")
```

Before and after outlier removal: Rajaji Nagar

```
In [40]: plt.hist(df.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

```
Out[40]: Text(0, 0.5, 'Count')
```



Modify Dataframe

```
In [44]: df = pd.concat([df,dummies.drop('other',axis='columns')],axis='columns')
df.head()
```

```
Out[44]:
```

	location	size	total_sqft	bath	price	BHK	price_per_sqft	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	...	Vijayanagar	Vishveshwarya Layout	Vishwapa Lay
0	1st Block Jayanagar	4 BHK	2850.0	4.0	428.0	4	15017.543860	1	0	0	...	0	0	
1	1st Block Jayanagar	3 BHK	1630.0	3.0	194.0	3	11901.840491	1	0	0	...	0	0	
2	1st Block Jayanagar	3 BHK	1875.0	2.0	235.0	3	12533.333333	1	0	0	...	0	0	
3	1st Block Jayanagar	3 BHK	1200.0	2.0	130.0	3	10833.333333	1	0	0	...	0	0	
4	1st Block Jayanagar	2 BHK	1235.0	2.0	148.0	2	11983.805668	1	0	0	...	0	0	

5 rows × 247 columns

Build Regression Model

Split Train and Text Data

```
In [49]: from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(feature , label , test_size = 0.2 , random_state = 10)
```

Build Regression Model

```
In [50]: from sklearn.linear_model import LinearRegression
```

```
In [51]: model = LinearRegression()
model.fit(X_train,y_train)
```

```
Out[51]: LinearRegression()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

Model Accuracy

```
In [52]: accuracy = model.score(X_test,y_test)
accuracy = str(accuracy).replace("0.", "")[:2]
print(f"Accuracy : {accuracy}%")
```

Accuracy : 86%

K Fold cross validation

```
In [53]: from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)

cross_val_score(LinearRegression(), feature, label, cv=cv)

Out[53]: array([0.82702546, 0.86027005, 0.85322178, 0.8436466 , 0.85481502])
```


Predict BHK Price

```
In [54]: def predict_price(location,sqft,bath,bhk):
loc_index = np.where(feature.columns==location)[0][0]
x = np.zeros(len(feature.columns))
x[0] = sqft
x[1] = bath
x[2] = bhk
if loc_index >= 0:
    x[loc_index] = 1

return model.predict([x])[0]

In [55]: predict_price('Indira Nagar',1000, 3, 3)
```

5.2 RESULT:



Area (Square Feet)

BHK

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Bath

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Location

Choose a Location ▼

Estimate Price

Figure 5

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

An optimal model does not necessarily represent a robust model. A model that An ideal model doesn't really address a vigorous model. A model that oftentimes utilize a learning calculation that isn't reasonable for the given information structure. Once in a while the actual information may be excessively boisterous or it could contain too couple of tests to empower a model to precisely catch the objective variable which suggests that the model remaining parts fit. At the point when we notice the assessment measurements got for cutting edge relapse models, we can say both act along these lines. We can pick possibly one for house value expectation contrasted with essential model. With the assistance of box plots, we can check for anomalies. In the event that present, we can eliminate exceptions and check the model's presentation for development.

6.2 FUTURE WORK:

We can build models through advanced techniques namely random forests, neural networks, and particle swarm optimization to improve the accuracy of predictions. It is necessary to check before deciding whether the built model should or should not be used in a real-world setting. The data has been collected in 2016 and Bengaluru is growing in size and population rapidly. So, it is very much essential to look into the relevancy of data today. The characteristics present in the data set are not sufficient to describe house prices in Bengaluru. The dataset considered is quite limited and there are a lot of features, like the presence of pool or not, parking lot and others, that remain very relevant when considering a house price. The property has to be categorized either as a flat or villa or independent house. Data collected from a big urban city like Bengaluru would not be applicable in a rural city, as for equal value of feature prices, which will be comparatively higher in the urban

REFERENCES

- ❖ H.L. Harter, Method of Least Squares and some alternatives-Part II. International Static Review. 1972, 43(2), pp. 125-190.
- ❖ J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp. 68-73.
- ❖ Lu. Sifei et al, A hybrid regression technique for house prices prediction. In proceedings of IEEE conference on Industrial Engineering and Engineering Management: 2017.
- ❖ R. Victor, Machine learning project: Predicting Boston house prices with regression in towards datascience.
- ❖ Predicting house price Bengaluru (Machine Hackathon)
<https://www.machinehack.com/course/predicting-house-prices-in-bengaluru/>
- ❖ Pow, Nissan, Emil Janulewicz, and L. Liu (2014). Applied Machine Learning Project 4 Prediction of real estate property prices in Montréal.
- ❖ S. Neelam, G. Kiran, Valuation of house prices using predictive techniques, Internal Journal of Advances in Electronics and Computer Sciences: 2018, vol 5, issue-6
- ❖ S. Abhishek.: Ridge regression vs Lasso, How these two popular ML Regression techniques work. Analytics India magazine, 2018.
- ❖ S. Raheel. Choosing the right encoding method-Label vs One hot encoder. Towards datascience, 2018
- ❖ Wu, Jiao Yang (2017). Housing Price prediction Using Support Vector Regression.

*****THANKYOU*****