HOUSE PRICE PREDICTION

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering in Information Technology

By

SUSHANT KUKARNI (Roll No. 17101B0001)

SHEFIN SHAJIT (Roll No. 17101B0002)

AKSHAY MOHITE (Roll No. 17101B0012)

Under the Guidance of

Dr. SWATI SINHA

Department of Information Technology Engineering

Vidyalankar Institute of Technology
Wadala(E), Mumbai-400437

University of Mumbai

2020-21

CERTIFICATE OF APPROVAL

This is to certify that the project entitled

"HOUSE PRICE PREDICTION"

is a bonafide work of

SUSHANT KULKARNI (Roll No. 17101B0001)

SHEFIN SHAJIT (Roll No. 17101B0002)

AKSHAY MOHITE (Roll No. 17101B0012)

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of

Undergraduate in **INFORMATION TECHNOLOGY**.

Guide	Head of Department	Principal
(Dr. Swati Sinha)	(Dr Deepali Vora)	(Dr S.A. Patekar)

Project Report Approval for B. E.

This project report entitled ***HOUSE PRICE PREDICTION*** by

1. ***SUSHANT KULKARNI (Roll No. 17101B0001)***
2. ***SHEFIN SHAJIT (Roll No. 17101B0002)***
3. ***AKSHAY MOHITE (Roll No. 17101B0012)***

is approved for the degree of ***Bachelor of Engineering in Information Technology.***

Examiners

1.----------

2.----------

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student	Roll No.	Signature
1. SUSHANT KULKARNI	17101B0001	
2. SHEFIN SHAJIT	17101B0002	
3. AKSHAY MOHITE	17101B0012	

Date:

ACKNOWLEDGEMENT

We are pleased to present **“House Price Prediction”** as our project and take this opportunity to express our profound gratitude to all those people who helped us in completion of this project.

We thank our college for providing us with excellent facilities that helped us to complete and presentthis project. We would also like to thank the staff members and lab assistants for permitting us to usecomputers in the lab as and when required.

We express our deepest gratitude towards our project guide **Dr. Swati Sinha** for her valuable and timely advice during the various phases in our project. We would also like to thank him for providingus with all proper facilities and support as the project co-coordinator. We would like to thank him forsupport, patience and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules.

Finally, we would like to thank everyone who has helped us directly or indirectly in our project.

Sushant Kulkarni

Shefin Shajit

Akshay Mohite

ABSTRACT

With the ever-rising traffic to own a place, house or in general real estate, it is important to have the best tools at disposal which would guide the buyers about where to put their money into. As home-buying season is shifting into high gear, buyers prefer to make smart investments and purchases which in turn provide profits or prove to a steal for the price. Selling and purchasing a house is a section which invites lot of statistics and data. We will be using these numbers to make a prediction of the price of a house in accordance with the existence of amenities such as parking, pool, school, etc.; or not.

Our work is based on a set of data containing house prices of places in Mumbai along with the major parameters affecting the price such as area, location, swimming pool, etc obtained from open web source Kaggle Inc. and predict the price under the parameters. The model implemented incorporates ensemble learning i.e., a combination of machine learning algorithms instead of relying on a single algorithm for improved predictions. The ensemble model incorporated in our system (weighted average of Decision Tree, Linear Regression, and K-Nearest Neighbour) brings an added advantage over using solo algorithms in the process of obtaining minimum error and trying to get the predictions as accurate as possible.

TABLE OF CONTENTS

SR.NO	TITLE	PAGE NO.
1.	**Introduction**	1
1.1	Concept	2
1.2	Aim	3
1.3	Objectives	3
1.4	Problem Definition	3
1.5	Problem Solution	4
1.6	Scope	4
2.	**Literature Survey**	5
3.	**System Design**	12
3.1	Proposed System	13
3.2	Process Model	18
3.3	Feasibility Study	19
3.4	Dataflow Diagram	20
3.5	Sequence Diagram	21
4.	**Methodology**	22
4.1	Data Analysis	24
4.2	Machine Learning	35
5.	**Planning and Scheduling**	44
5.1	Gantt Chart	45
5.2	PERT	46
6.	**Testing and Maintenance**	48
6.1	Testing	49
6.2	Maintenance	54
7.	**System Implementation**	57
7.1	Working	58
7.2	Results	63
8.	**Conclusion and Future Scope**	66
8.1	Conclusion	67
8.2	Future Scope	67

9.	**References**	68
9.1	Research Papers	69
9.2	Online References	70

LIST OF FIGURES AND TABLES

FIGURES

Sr No.	Content	Page No.
3.1	Block Diagram	13
3.2	Pair Plot	15
3.3	Heatmap	15
3.4	Incremental/Iterative Model	18
3.5	Dataflow Diagram	20
3.6	Sequence Diagram	21
4.1	Dataset	25
4.2	dataframe.head() function	26
4.3	dataframe.info()	26
4.4	dataframe.describe()	27
4.5	Seaborn.pairplot()	28
4.6	Seaborn.distplot()	29
4.7	Seaborn.heatmap()	30
4.8	Location_ID and Avg_Price_Area columns	32
4.9	Removing Outliers	33
4.10	Scatterplot before removing Outliers	34

4.11	Scatterplot after removing Outliers	35
4.12	Linear Regression Implementation	37
4.13	K-Nearest Neighbors Algorithm Implementation	39
4.14	Decision Tree Algorithm Implementation	41
4.15	Ensemble Learning Implementation	43
5.1	Gantt Chart	45
5.2	PERT Chart	47
6.1	V Model	51
6.2	Types of Maintenance	55
7.1	Creation of Pickle Files	58
7.2	Creating Pickle Files of ML Models	59
7.3	HTML Form code snippet	60
7.4	JavaScript code for handling Checkbox Input	60
7.5	Webpage	61
7.6	Flask app.py code snippet	62
7.7	Webpage hosted on Flask Server	62
7.8	Mean Absolute Percentage Error	63
7.9	Result 1	64
7.10	Result 2	64
7.11	Result 3	65
7.12	Result 4	65

TABLES

Sr No.	Content	Page No.
1	Literature Survey	11
2	Linear Regression Error Metrics	37
3	KNN Error Metrics	39
4	Decision Tree Error Metrics	41
5	Comparison of Error Metrics	43
6	PERT Table	46

Chapter 1

Introduction

1. INTRODUCTION

1.1 CONCEPT

Housing being one of the basic needs of human, accounts to high percentages of national transactions per year. The real estate sector is a major sector influencing India's economy. In India, about 15 percent of the total jobs are generated by the real estate sector. Since property prices rarely decrease rapidly, it is a major contender for investment. The property prices depend on various intrinsic and extrinsic factors which directly or indirectly affect the long-term price values.

A common issue that normally emerges is the measurement of asset values for investment purposes. The heterogeneity exhibited by the real property values is given due weightage which derives a pattern of variation in the values of properties over a period. Careful attention must be given to the dynamics of various factors affecting the housing prices for full understanding on our research on predicting housing prices.

In 2021, the sector must adopt innovative ways of dealing with the requirements. While houses will continue to be sold, they will now be done with creative disruption. The reinvention will include technology playing a lead role in meeting altered norms being considered by home buyers. There are three factors that influence the price of a house which include physical conditions, concept and location. Physical condition are properties possessed by a house that can be observed by human senses, including the size of the house, the number of bedrooms, the availability of kitchen and garage, the availability of garden, the area of land and buildings, distance from school and stations, the age of the house. Location is an important factor in shaping the price of a house. This is because the location determines the ease of access to public facilities, such as schools, campus, hospitals and health centers as well as family recreation facilities such as malls, culinary tours, or even beautiful scenery.

1.2 AIM

A fair share of India's economic condition affects property prices in the long run. This scenario calls for technology to bring out the best ways to guide the customer's investment decisions. It is important to have the best tools at disposal which would guide the buyers about where to put their money into.The main aim of the project is to predict the house pricing for buyers with respect to various factors and priorities affecting the prices. By analyzing current market trends and price ranges, prices will be predicted.

1.3 OBJECTIVES

1. To study the current literature available for House price prediction.
2. To identify the ML models suitable for accurately predicting House price.
3. Design a website which accepts customers specifications and then combines the application with the trained model.
4. To provide guiding information to the companies and customers in the sector and to contribute to the literature.

1.4 PROBLEM DEINITION

To develop a system that would help the user to estimate the price for a particular house considering the real-life factors such as location of the house and the connectivity, leisure services, schools etc. that affect the price of the house along with the features of house and the amenities available.

1.5 PROBLEM SOLUTION

The solution proposed is by using location of property and ensemble learning model. The ensemble model incorporated in our system (weighted average of Decision Tree, Linear Regression, and K-Nearest Neighbour) brings an added advantage over using solo algorithms in the process of obtaining minimum error in prediction.

1.6 SCOPE

The aim of the system is to help the user to make smart investment decisions while buying a house or find the accurate price of the house considering the factors which increase the price in the vicinity of the house. The price is predicted by machine learning models trained using a data set which contains various attributes affecting the price of the house. The model will be integrated with a Web User Interface (UI) where the user can interact with the system and input the desired value for given parameters.

Chapter 2

Literature Survey

2. LITERATURE SURVEYED

Over the years there have been numerous approaches in predicting house prices per all the intrinsic and extrinsic factors affecting the price without any fluctuations. Although finding the best possible prediction model depends on the data available. The price of a property can differs based on its location, area, amenities, etc., and finding the best predictive model to predict that price has been a concern for researchers over the past decade. In our literature survey, we found various such approaches to find the house price using various models and a combination of models.

One of the methods proposed in the paper by Neelam Shinde and Kiran Gawande [1] includes testing the dataset with four different regression algorithms namely Lasso Regression, Logistic Regression, Decision Tree, and Support Vector Regression. On comparing the error metrics such as R-Squared Value, Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error, Decision Tree turned out to be the best algorithm giving a higher accuracy level of 86.4% and low error values whereas Lasso Regression performed the worst giving an accuracy level of 60.32%.

The majority of our work around ensemble learning for our predictive model has been widely accepted as a success in improving predictions [2]. Where ensemble models can vary depending on the needs of the data and the manner of predictions where even a small amount of improvement can have a big impact. The integration of two or more ensemble members depends on the type of integration the developer would see fit for the data i.e. Constant Weighting Functions and Non-Constant Weighting Functions.

A different approach has been taken in the paper [10]. The paper focused on linking various research related to housing market prices analysis. Substantial focus is provided to hedonic price modeling and its application on the house price market and the possible submarket existence.

Decision Tree is used to make predictions in paper [5] after giving the highest accuracy in terms of prediction values among other algorithms tested namely Linear Regression, Multiple Linear Regression, Decision Tree Regressor, and KNN. Apart from parameters like no. of bedrooms, carpet area, built-up area, age of the property, zip code, no. of bathrooms, latitude, and longitude of the property, they have also included two other features – air quality and crime rate to better the prediction.

Another proposed system used Lasso and Random Forest regression techniques and picked the best model for the data depending on error values [3]. The data was passed onto 6 stages including data pre-processing, test-train 50:50 split, training the data with Lasso and Random Forest models and testing with the test data, and picking the best model.

In the paper by Prof. Pradnya Patil, Darshil Shah, Harshad Rajput, and Jay Chheda [4], the proposed system utilises the UiPath Studio Platform to develop the RPA Flowchart. The UiPath Studio provides data scraping capabilities with the assistance of scraping wizards. A bunch of machine learning algorithms are compared and implemented on the dataset. A comparison between boosting algorithms is done namely XGBoost, Light BGM, and CatBoost. Random Forest was found to do well with small amounts of data and doesn't improve accuracy with more samples. CatBoost was termed the clear winner in comparisons. RPA provided a major improvement in efficiency in terms of fast extraction and less prone to errors.

A further step has been taken in the paper by P. Durganjali and M. Vani Pujitha [8]. It analyses different classification algorithms such as Decision Tree, Logistic Regression, Random Forest, AdaBoost, Naïve Bayes with an accuracy of 92%, 81.5%, 86.5%, 96%, and 88% respectively. AdaBoost and Decision Tree using C 5.0 were selected to predict values of the house and using rules, they predicted profit or loss.

Detailed study of different machine learning algorithms namely Multiple Linear Regression, Elastic Net Regression, Ridge Regression, Ada Boosting Regression, LASSO Regression, and Gradient Boosting has been done on a public output dataset of a specified region in the USA [9]. The attained scores of the algorithms were 0.73, 0.66, 0.73, 0.78, 0.73 and 0.91 respectively. Gradient Boosting turned out to be the best algorithm as it gave low error .

Following is a tabular representation of the Research papers that have been surveyed and the observations drawn from them:

Sr.No.	**Authors**	**Title**	**Observations**
1.	Ayush Varma, Sagar Doshi, Abhijit Sarma, Rohini Nair	House Price Prediction Using Machine Learning And Neural Networks.	Instead of an individual algorithm a series of algorithm yields better results.
2.	P. Durganjali ; M. Vani Pujitha	House Resale Price Prediction Using Classification Algorithms	Ada Boost algorithm has highest accuracy.
3.	CH. Raga Madhuri; G. Anuradha ; M. Vani Pujitha.	House Price Prediction Using Regression Techniques: A Comparative Study	Gradient Boosting regression has highest accuracy.
4.	A. Adair, J. Berry, W. McGreal,	Hedonic modelling, housing submarkets and residential valuation.	Identification of variables having the most significant influence on value and the combination of variables entering into the final models.

5.	O. Bin	A prediction comparison of housing sales prices by parametric versus semi- parametric regressions.	The results show that the semi- parametric regression outperforms the parametric counterparts in both in-sample and out-of-sample price predictions.
6.	T. Kauko, P. Hooimeijer, J. Hakfoort	Capturing housing market segmentation: An alternative approach based on neural network modelling.	The classification abilities of two neural network techniques: the self-organising map (SOM) and the learning vector quantisation (LVQ).
7.	Li Li , Kai-Hsuan Chu	Prediction of Real Estate Price Variation Based on Economic Parameters.	Prediction of Real Estate Price Variation Based on Economic Parameters.
8.	G. Naga Satish, Ch.V.Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu	House Price Prediction Using Machine Learning.	Lasso regression algorithm, in view of accuracy, reliably outperforms alternate models in the execution of housing cost prediction.
9.	Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, Wayan Firdaus Mahmudy	Modelling House Price Prediction using Regression Analysis and Particle Swarm Optimization.	The result from this research proved combination regression and PSO is suitable and get the minimum prediction error obtained

10.	Ayşe SOY TEMÜR1*, Melek AKGÜN2, Günay TEMÜR	Predicting housing sales in Turkey using ARIMA, LSTM and Hybrid models	ARIMA, LSTM and HYBRID model formed from these two models have been used. The HYBRID model produced the best performance among these three models.
11.	Gaikwad Purva Chandrakant, Ganjave Pratiksha Namdev, Gorade Pooja Subhash, S. S. Gore	Implementation of House Price Prediction Model Using Image Processing and Machine Learning	Proposed system focused on predicting the house price according to the area for that image processing and machine learning methods are used. The experimental results showed that this technique that are used while developing system will give accurate prediction of house price.
12.	T. Kauko, P. Hooimeijer, J. Hakfoort.	Capturing housing market segmentation: An alternative approach based on neural network modelling.	The classification abilities of two neural network techniques: the self-organizing map (SOM) and the learning vector quantization (LVQ).
13.	Li Li, Kai-Hsuan Chu.	Prediction of Real Estate Price Variation Based on Economic Parameters.	Prediction of Real Estate Price Variation Based on Economic Parameters.

14.	G. Naga Satish, Ch.V.Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu.	House Price Prediction Using Machine Learning.	Lasso regression algorithm, in view of accuracy, reliably outperforms alternate models in the execution of housing cost prediction.
15.	Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, Wayan Firdaus Mahmudy.	Modelling House Price Prediction using Regression Analysis and Particle Swarm Optimization.	The result from this research proved combination regression and PSO is suitable and get the minimum prediction error obtained
16.	Ayşe SOY TEMÜR1*, Melek AKGÜN2, Günay TEMÜR.	Predicting housing sales in Turkey using ARIMA, LSTM and Hybrid models	ARIMA, LSTM and HYBRID model formed from these two models have been used. The HYBRID model produced the best performance among these three models.

Table 1: Literature Survey

Chapter 3

System Design

3. SYSTEM DESIGN

3.1 PROPOSED SYSTEM

Proposed system means explaining what you are going to do in this project. What is your project and what is new in your project. And what approach you are going to use. In short proposed system is explaining the steps of implementing your project. Following diagram shows the basic block diagram of the proposed system:

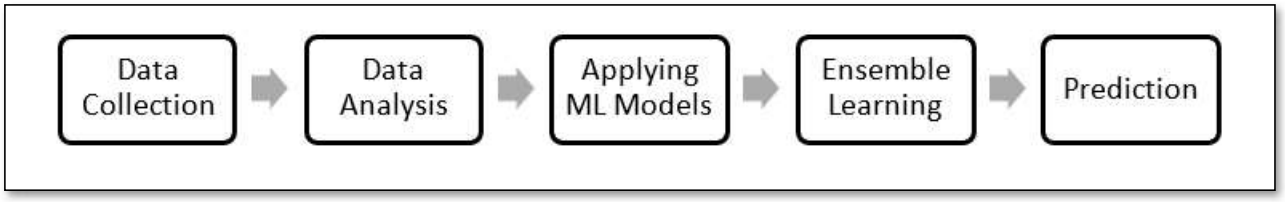


Fig 3.1: Block Diagram

3.1.1 DATA COLLECTION

Data Collection is the process of forming a suitable dataset by extracting data from various credible sources. The dataset formed should be in accordance to your prescribed machine learning model so that data should fit within the boundaries of the algorithm. The data collection process involved the following steps:

Dataset from kaggle.com:

The dataset incorporated in the system is taken from a public dataset source Kaggle Inc. It has data for house prices and features from 413 unique locations of Mumbai, India. It consists of 6347 records with 17 parameters that have the possibility of affecting the property prices. However, out of these

17 parameters, only 7 were chosen (Area, No. of Bedrooms, New/Resale, Gymnasium, Lift Available, Car Parking, Swimming Pool) along with 2 added parameters (Location Id and Price Area) which are bound to have a major effect on housing prices. The area is the total built-up area in square feet. New/Resale specifies if the property is a resale property or a new property. Gymnasium, Lift, Car Parking, and Swimming Pool mention if the property happens to provide these amenities (Binary value i.e., 1s and 0s). Location id is a unique id to all the locations present in the dataset in ascending order of Price Area. Price Area is the average price per area of a location.

Data Customization

The acquired data from websites needs to be optimized so as to fit in a Regression model. The parameters such as "Swimming Pool" and "Lift" were in the format Yes and NO they needed to be converted into 1s and 0s. Likewise there were a total of 11 columns that needed data customization.

3.1.2 DATA ANALYSIS

Data analysis is defined as a process transforming, and modeling data to discover useful information for decision-making. Data analysis is the step where visualizing the data from the datasets helps in making useful insights. The following functions were used in data analysis of our dataset:

.info(), .describe():

These functions gave an overall idea about the dataset including the data type and total entries in each column, The max, min average values in each column. They helped in visualizing which columns need to be compared against one another for logic building.

sns.pairplot():

The seaborn 'pair plots' compare columns against each other .In this instant a comparison of Price against No of bedrooms gave us the insight of how a unit change in number of bedrooms affect the price of property.

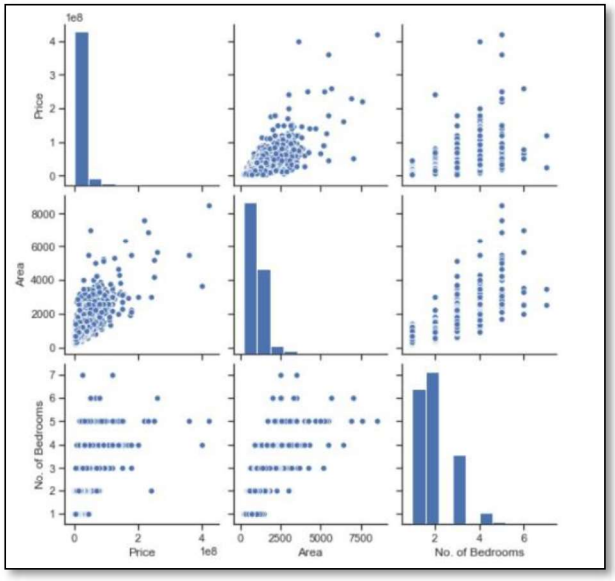


Fig 3.2 Pairplot

sns.heatmap()

We are predicting" Price" of a house and hence we need an insight into how each column effect the price. A heat map provides us with that input. It gives us the coefficient of change of each column with respect to price. It tells us how a unit each column affects the change in price column.

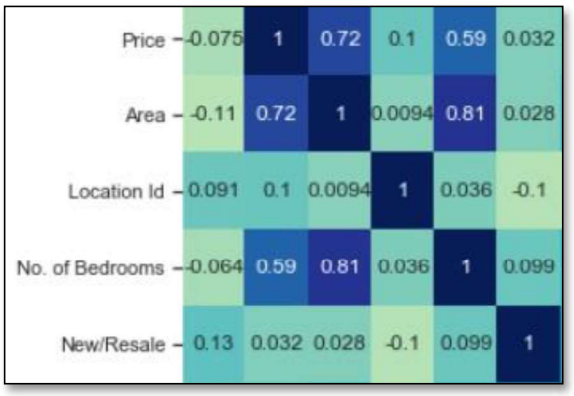


Fig 3.3 Heatmap

3.1.3 APPLYING MACHINE LEARNING MODELS

After analyzing and visualizing the data the next step is to process the data to help us in predicting the house prices. The next step involves splitting the data into training and testing sets. Here we have narrowed down to 3 regression models based on our literature survey and still in the process of finalizing the particular regression models. All models were applied using the particular TensorFlow libraries. The following regression models are applied:

Linear Regression:

Linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.

K-Nearest Neighbor (KNN):

K-Nearest Neighbor (KNN) is a machine learning algorithm that does regressive as well as classification predictive analysis. It is also called a lazy learner algorithm since it does not analyze the data it is trained with instead the algorithm only classifies the new data it is tested with by its similarity.

Decision Tree:

A decision tree algorithm builds the model in a tree structure with decision nodes and leaf nodes. It breaks down the dataset into smaller sets with similar values and the highest node is known as the root node. The tree is made of only conditional control statements with each decision node testing an attribute.

3.1.4 ENSEMBLE LEARNING

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance (bagging), bias (boosting), or improve predictions (stacking). We reached a conclusion from our regression model results and literature survey that a hybrid model using ensemble learning needed to be deployed to gain superior predictive analysis and to minimize the error margin.

3.1.5 PREDICTION

The final product of our project is predicting the price of property. This is done by providing various parameters (Location, carpet area, floor, parking, etc.) as input and the predicted price as output. We have developed a Website for the same. This UI helps user seamlessly input the parameter values and get a prediction for the same.

3.2 PROCESS MODEL

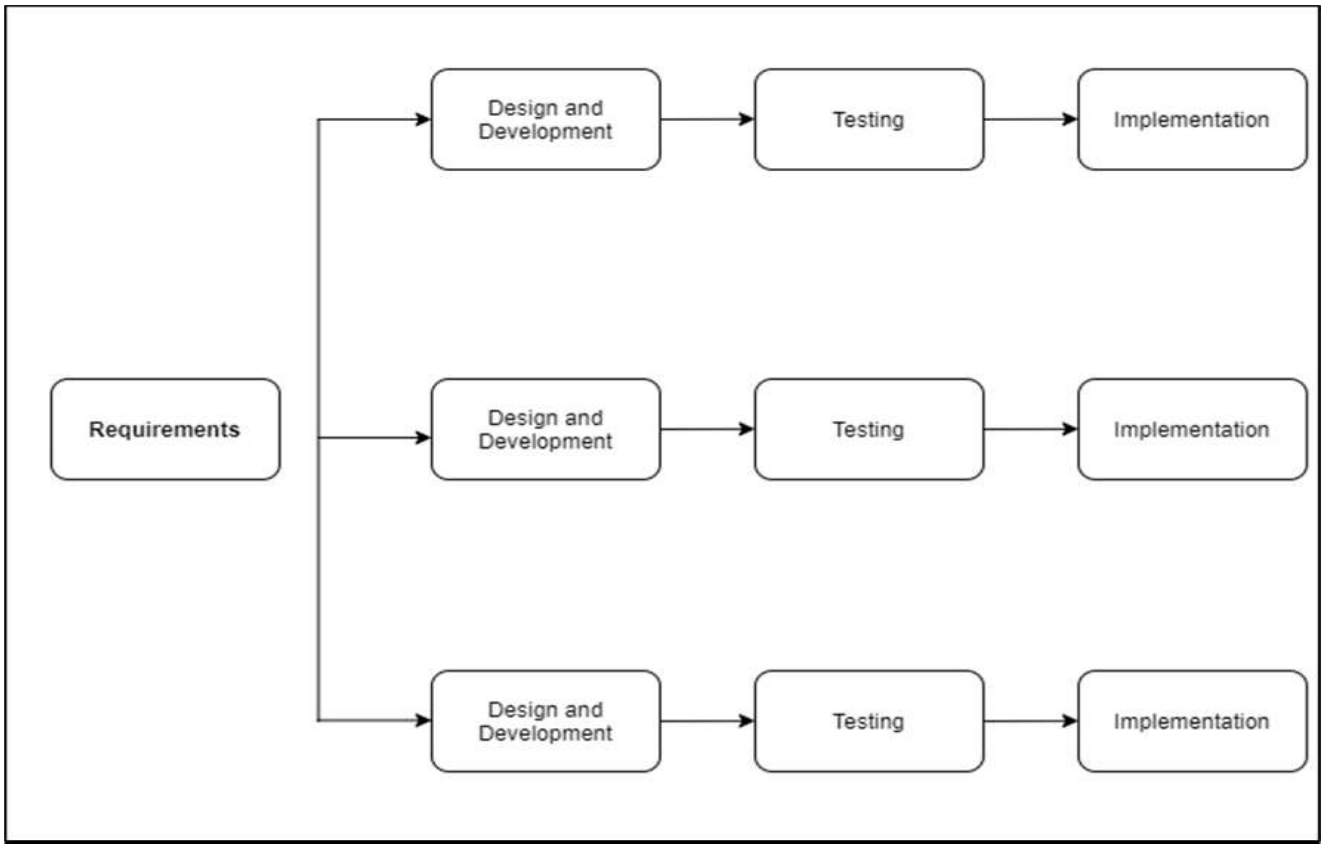


Fig. 3.4 Incremental/Iterative Model

We are using Incremental Model in our project development. The following being the main reasons for the same:

- The iterative model focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete.
- Iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements and hence it is best suited for the development of this project.

- Every Increment is followed by a review and feedback system which helps us understand the scope for further development in a better way.
- In this model an increment has its phases which do not overlap, and each phase is independent so even two different phases do not overlap.

3.3 FEASIBILITY STUDY

- Technical Feasibility: Technical feasibility is the process of validating the technology assumptions, architecture and design of a product or project. The software required for our project are open-source python libraries and APIs.

- Economic Feasibility: Economic feasibility is the cost and logistical outlook for a project or endeavor. The cost estimation for this project is minimal as no hardware components are involved and most software used are free. The main cost involved is of the paid APIs and cost of hosting the application on servers.

- Legal Feasibility: This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts, or social media laws. The project does not involve any legal concerns since all the licenses and laws will be respected and included in the project.

- Operational Feasibility: Operational feasibility is the measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The main objective of this project is to predict the property price and give user an overall idea by future price prediction based on previous trends.

3.4 DATAFLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into detail, which can later be elaborated.

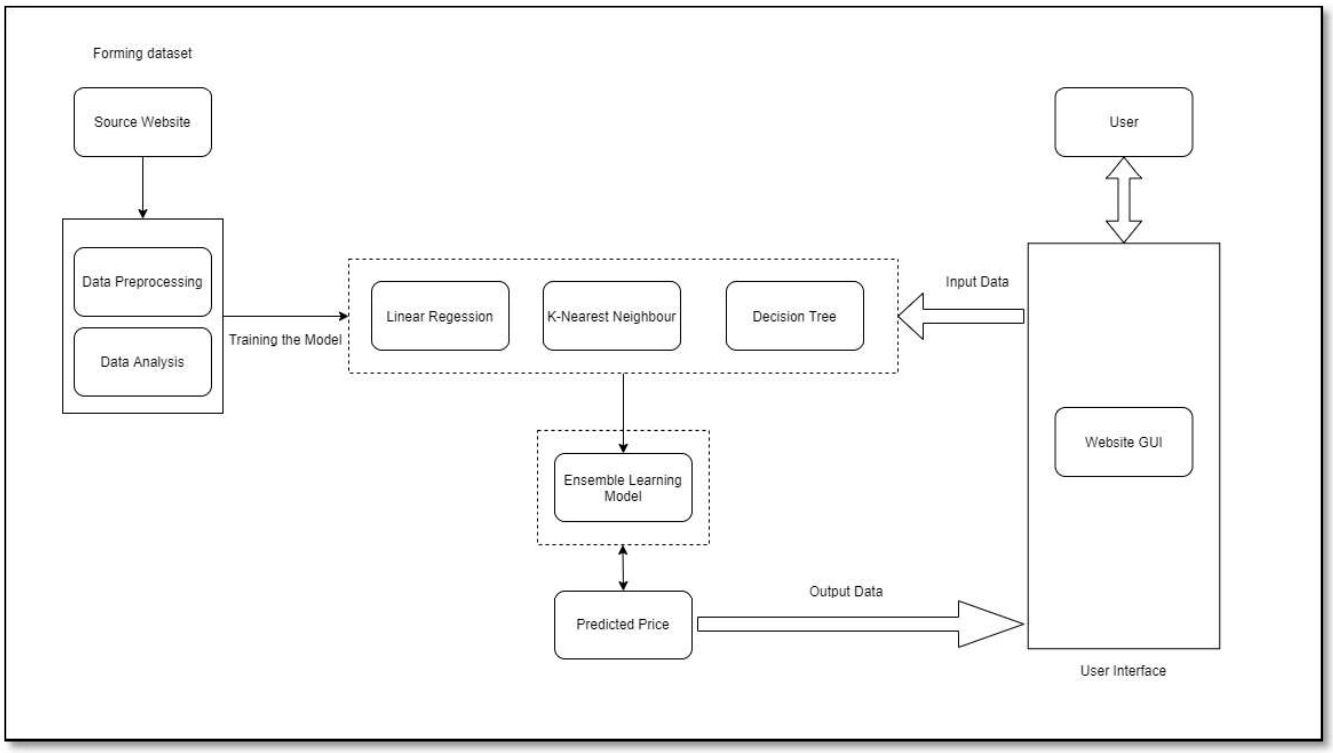


Fig 3.5 Dataflow Diagram

3.5 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. Sequence diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram is a structured representation of behavior as a series of sequential steps over time. It is used primarily to show the interactions between objects in the sequential order. The sequence diagram is also called as Message Sequence Chart.

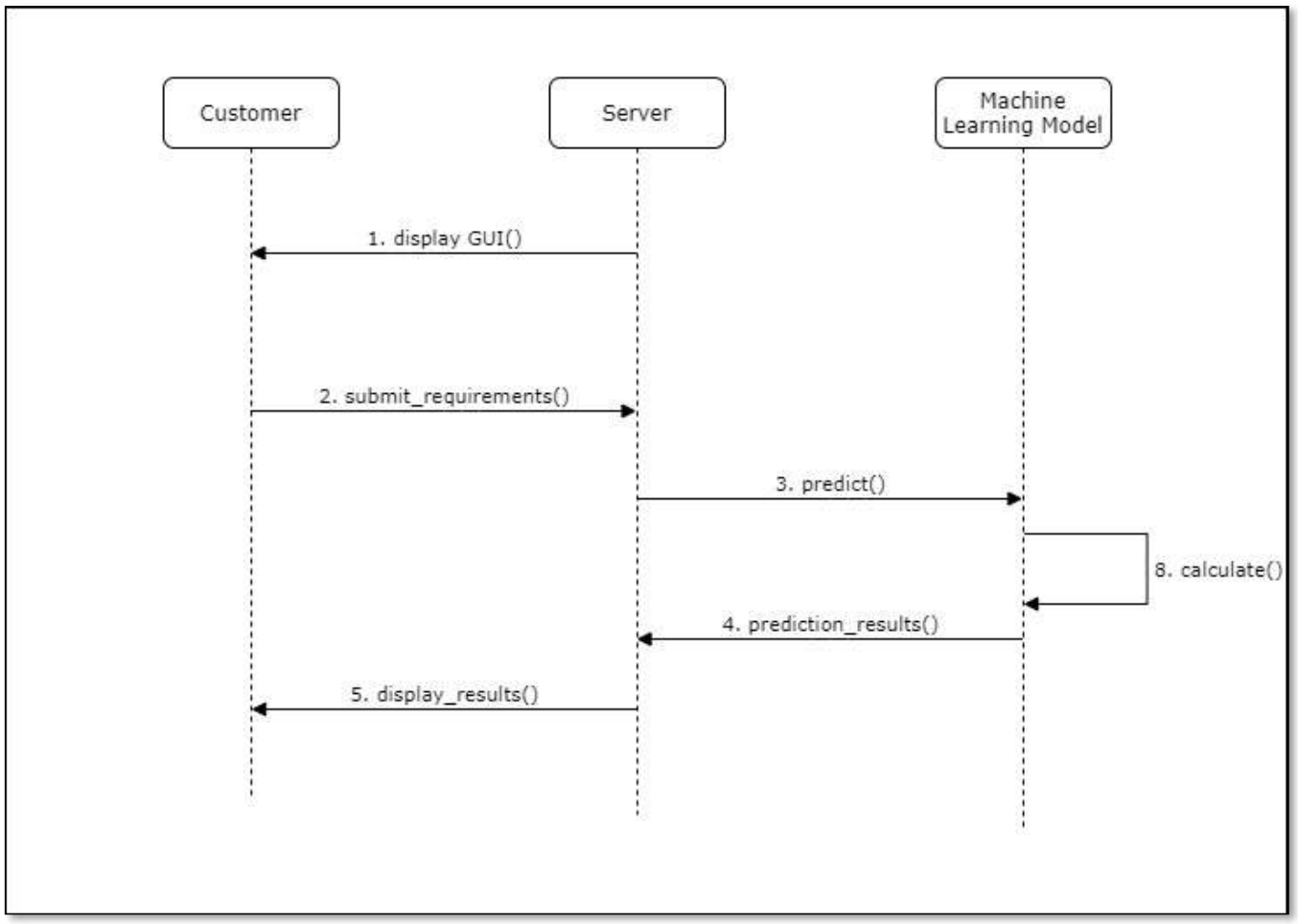


Fig 3.6 Sequence Diagram

Chapter 4

Methodology

4. METHODOLOGY

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.A methodology does not set out to provide solutions—it is therefore, not the same as a method. Instead, a methodology offers the theoretical underpinning for understanding which method, set of methods, or best practices can be applied to a specific case.

Predicting housing prices with real factors is the main crux of our research project. We aim to make our evaluations based on every basic parameter that is considered while determining the price. Our model analyses a set of parameters selected by the customer, to find an ideal price according to their requirements and interest.

For our research project, we have considered Mumbai as our primary location. We have used parameters like ‘area’, ‘no. of bathrooms’, ‘parking’ ‘avg. price per area’ of location, etc. We have considered a verified dataset with diversity to give accurate results for all conditions. It comprises of various essential parameters with appropriate data analysis. We use various regression techniques in this pathway, and our results are not sole determination of one technique rather it is the weighted mean of various techniques to give most accurate results. The method we applied, mainly has two components:

1. Data Analysis
2. Machine Learning

4.1 DATA ANALYSIS

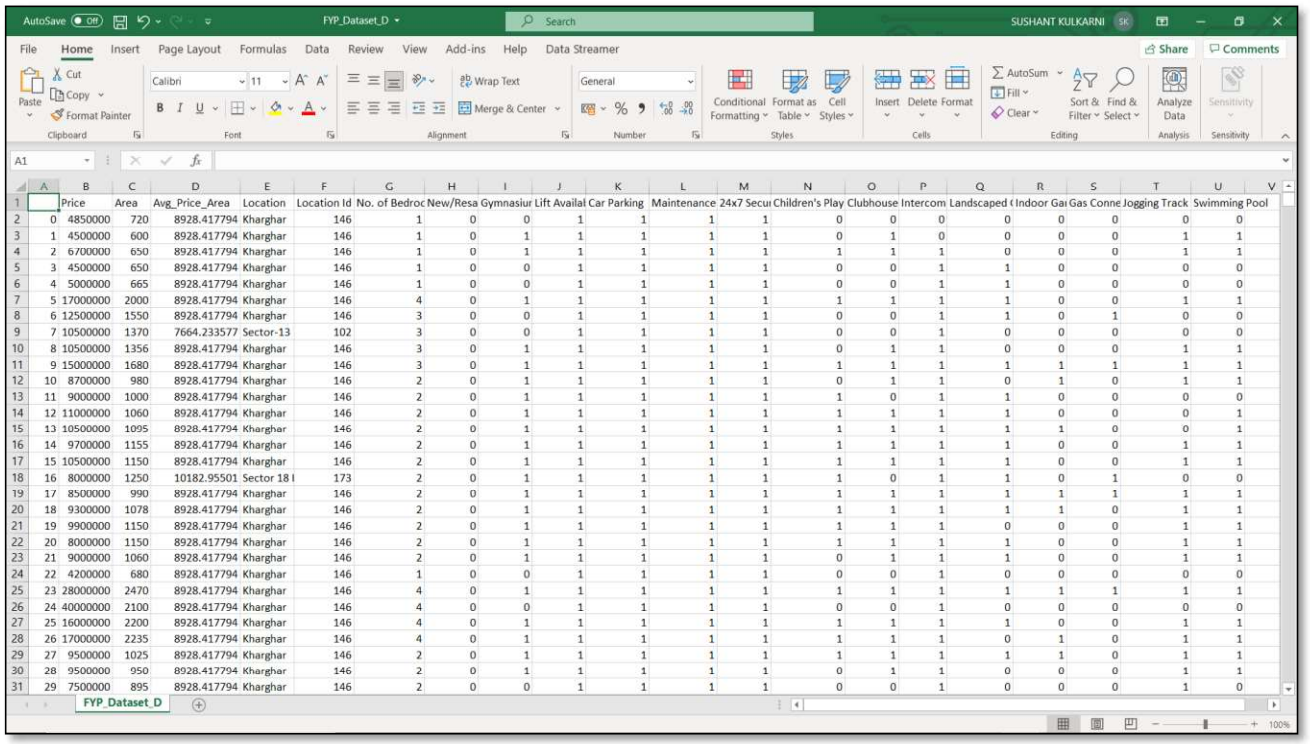
Data analysis is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis. A simple example of Data analysis is whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that decision. This is nothing but analyzing our past or future and making decisions based on it.

Our first step towards creating the model was trying to understand the type, quantity and quality of data we had gathered. We followed a very comprehensive and standard approach in our Data Analysis. The correct form and type of dataset was thoroughly searched. After narrowing down on the dataset it went through the process of cleaning and editing. Any dataset should be structured according to the Machine Learning model's input and output. Data Visualization helped in deciding parameters that should and should not be included in the training set. We followed the following steps in our Data Analysis:

1. Data Collection
2. Understanding the dataset
3. Data Visualization
4. Data Cleaning
5. Data Processing

4.1.1 DATA COLLECTION

We considered two approaches for building or acquiring the dataset. The first being finding an existing dataset on 'Kaggle.com' which is an open-source repository for communities for sharing data and projects. Another attempt involved data scraping. The only reliable sources for genuine property value data were property sites like 'Magicbricks.com', '99acers.com', 'Makaan.com', 'Housing.com'. After rounds of discussion and research we narrowed down to a dataset from 'Kaggle.com' which included favorable parameters.



	Price	Area	Avg_Price_Area	Location	Location Id	No. of Bedroc	New/Resa	Gymnasiur	Lift Availa	Car Parking	Maintenance	24x7 Secu	Children's Play	Clubhouse	Intercom	Landscaped	Indoor Gai	Gas Conne	Jogging Track	Swimming Pool
0	4850000	720	8928.417794	Kharghar	146	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0
1	4500000	600	8928.417794	Kharghar	146	1	0	1	1	1	1	1	0	1	0	0	0	0	1	1
2	6700000	650	8928.417794	Kharghar	146	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1
3	4500000	650	8928.417794	Kharghar	146	1	0	0	1	1	1	1	0	0	1	1	0	0	0	0
4	5000000	665	8928.417794	Kharghar	146	1	0	0	1	1	1	1	0	0	1	1	0	0	0	0
5	17000000	2000	8928.417794	Kharghar	146	4	0	1	1	1	1	1	1	1	1	1	0	0	1	1
6	12500000	1550	8928.417794	Kharghar	146	3	0	0	1	1	1	1	0	0	1	1	0	1	0	0
7	10500000	1370	7664.233577	Sector-13	102	3	0	0	1	1	1	1	0	0	1	0	0	0	0	0
8	10500000	1356	8928.417794	Kharghar	146	3	0	1	1	1	1	1	0	1	1	0	0	0	1	1
9	15000000	1680	8928.417794	Kharghar	146	3	0	1	1	1	1	1	1	1	1	1	1	1	1	1
10	8700000	980	8928.417794	Kharghar	146	2	0	1	1	1	1	1	0	1	1	0	1	0	1	1
11	9000000	1000	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	0	1	1	0	0	0	0
12	11000000	1060	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	0	0	0	1
13	10500000	1095	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	1	0	0	1
14	9700000	1155	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	0	0	1	1
15	10500000	1150	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	0	0	1	1
16	8000000	1250	10182.95501	Sector 18 I	173	2	0	1	1	1	1	1	1	0	1	1	0	1	0	0
17	8500000	990	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1
18	9300000	1078	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	1	0	1	1
19	9900000	1150	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	0	0	0	1	1
20	8000000	1150	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	0	0	1	1
21	9000000	1060	8928.417794	Kharghar	146	2	0	1	1	1	1	1	0	1	1	1	0	0	1	1
22	4200000	680	8928.417794	Kharghar	146	1	0	0	1	1	1	1	0	0	1	0	0	0	0	0
23	28000000	2470	8928.417794	Kharghar	146	4	0	1	1	1	1	1	1	1	1	1	1	1	1	1
24	40000000	2100	8928.417794	Kharghar	146	4	0	0	1	1	1	1	0	0	1	0	0	0	0	0
25	16000000	2200	8928.417794	Kharghar	146	4	0	1	1	1	1	1	1	1	1	1	0	0	1	1
26	17000000	2235	8928.417794	Kharghar	146	4	0	1	1	1	1	1	1	1	1	0	1	0	1	1
27	9500000	1025	8928.417794	Kharghar	146	2	0	1	1	1	1	1	1	1	1	1	1	0	1	1
28	9500000	950	8928.417794	Kharghar	146	2	0	1	1	1	1	1	0	1	1	0	0	0	1	1
29	7500000	895	8928.417794	Kharghar	146	2	0	0	1	1	1	1	0	0	1	0	0	0	1	0

Fig 4.1 Dataset

4.1.2 STUDYING THE DATASET

The Dataset has 6345 data points and 20 columns which include Price, Location, Location ID, No of bedrooms, Built up area and also 15 columns covering other parameters that might affect the property price like Lift availability, gymnasium, swimming pool, etc. Every other column except the Location column is numeric. The location column consists of 413 different locations across the district of Mumbai ,Thane and Navi Mumbai. Pandas have various functions which help in understanding the dataset even further. These functions include:

.head()

It includes a brief look at the top 5 data points(tuples) of the dataframe. It is helpful to get a quick idea about column structure and orientation. The number of tuples shown can be changed by argument inside the function.

```
In [4]: 1 df.head(5)
```

Out[4]:

	Unnamed: 0	Price	Area	Location	Location Id	No. of Bedrooms	New/Resale	Gymnasium	Lift Available	Car Parking	Maintenance Staff	24x7 Security	Children's Play Area	Clubhouse	Inter
0	0	4850000	720	Kharghar	1	1	0	0	1	1	1	1	0	0	
1	1	4500000	600	Kharghar	1	1	0	1	1	1	1	1	0	1	
2	2	6700000	650	Kharghar	1	1	0	1	1	1	1	1	1	1	
3	3	4500000	650	Kharghar	1	1	0	0	1	1	1	1	0	0	
4	4	5000000	665	Kharghar	1	1	0	0	1	1	1	1	0	0	

Fig 4.2 dataframe.head() function

.info()

Pandas dataframe.info() function is used to get a concise summary of the dataframe. It comes really handy when doing exploratory analysis of the data. To get a quick overview of the dataset we use the dataframe.info() function. It gives us idea of the data type of data in each column. It also provides us total entries in each column and tells us total null values in each column. We need to aim to minimize the null values.

```
In [6]: 1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6347 entries, 0 to 6346
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Unnamed: 0            6347 non-null   int64
 1   Price                 6347 non-null   int64
 2   Area                  6347 non-null   int64
 3   Location              6347 non-null   object
 4   Location Id           6347 non-null   int64
 5   No. of Bedrooms       6347 non-null   int64
 6   New/Resale            6347 non-null   int64
 7   Gymnasium             6347 non-null   int64
 8   Lift Available        6347 non-null   int64
 9   Car Parking           6347 non-null   int64
 10  Maintenance Staff     6347 non-null   int64
 11  24x7 Security         6347 non-null   int64
 12  Children's Play Area  6347 non-null   int64
 13  Clubhouse             6347 non-null   int64
 14  Intercom              6347 non-null   int64
 15  Landscaped Gardens    6347 non-null   int64
 16  Indoor Games          6347 non-null   int64
 17  Gas Connection        6347 non-null   int64
 18  Jogging Track         6347 non-null   int64
 19  Swimming Pool         6347 non-null   int64
dtypes: int64(19), object(1)
memory usage: 991.8+ KB
```

Fig 4.3 dataframe.info() function

.describe()

The Pandas dataframe.describe() function gives descriptive demographics of each column of the dataset .It provides us with basic statistics like count, mean, minimum value, maximum value , standard deviation and values of various percentages of data. This data helps us in building logic and

to understand the range of our data for particular columns. In our case this function helped in getting the basic idea of price column in out dataset. The range of prices from minimum value to maximum.

```
In [7]: 1 df.describe()
```

Out[7]:

	Unnamed: 0	Price	Area	Location Id	No. of Bedrooms	New/Resale	Gymnasium	Lift Available	Car Parking	Maintenance Staff	24x7 Security	Cl P
count	6347.000000	6.347000e+03	6347.000000	6347.000000	6347.000000	6347.000000	6347.000000	6347.000000	6347.000000	6347.000000	6347.000000	634
mean	3173.000000	1.515401e+07	1004.327084	99.059083	1.910036	0.341736	0.581377	0.801481	0.562943	0.281393	0.562943	(
std	1832.365411	2.015943e+07	556.375703	105.180645	0.863304	0.474329	0.493372	0.398916	0.496061	0.449714	0.496061	(
min	0.000000	2.000000e+06	200.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	(
25%	1586.500000	5.300000e+06	650.000000	18.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	(
50%	3173.000000	9.500000e+06	905.000000	50.000000	2.000000	0.000000	1.000000	1.000000	1.000000	0.000000	1.000000	
75%	4759.500000	1.750000e+07	1182.000000	149.000000	2.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	
max	6346.000000	4.200000e+08	8511.000000	412.000000	7.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	

Fig 4.4 dataframe.describe() function

4.1.3 DATA VISUALIZATION

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. It is an efficient way of comprehending data when it is numerous in nature. Since the graphic design of the mapping can adversely affect the readability of a chart,[2] mapping is a core competency of Data visualization. The mapping determines how the attributes of elements vary according to the data.

There are 6347 tuples in our dataset for understanding this data better we need help of few data visualization charts and graphs so as to build proper logic for models and help in data cleaning and data processing procedures. Data visualization relies on the cognitive understanding capabilities of the reader and hence only those techniques are useful which give logical knowledge to the user after deployment. We have used the Seaborn Data Visualization library for the pre-defined plots and graphs. These are the following visualizations that we deployed.

Seaborn.pairplot()

By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column. The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each column. It is also possible to show a subset of variables or plot different variables on the rows and columns. We used a subset of the columns of the dataset for columns and rows of the pairplot. We plotted Price, Area, No. of bedrooms against each other. These columns are numerical and do not contain binary nature of data like other columns hence plotting them against one another could draw up logical inferences.

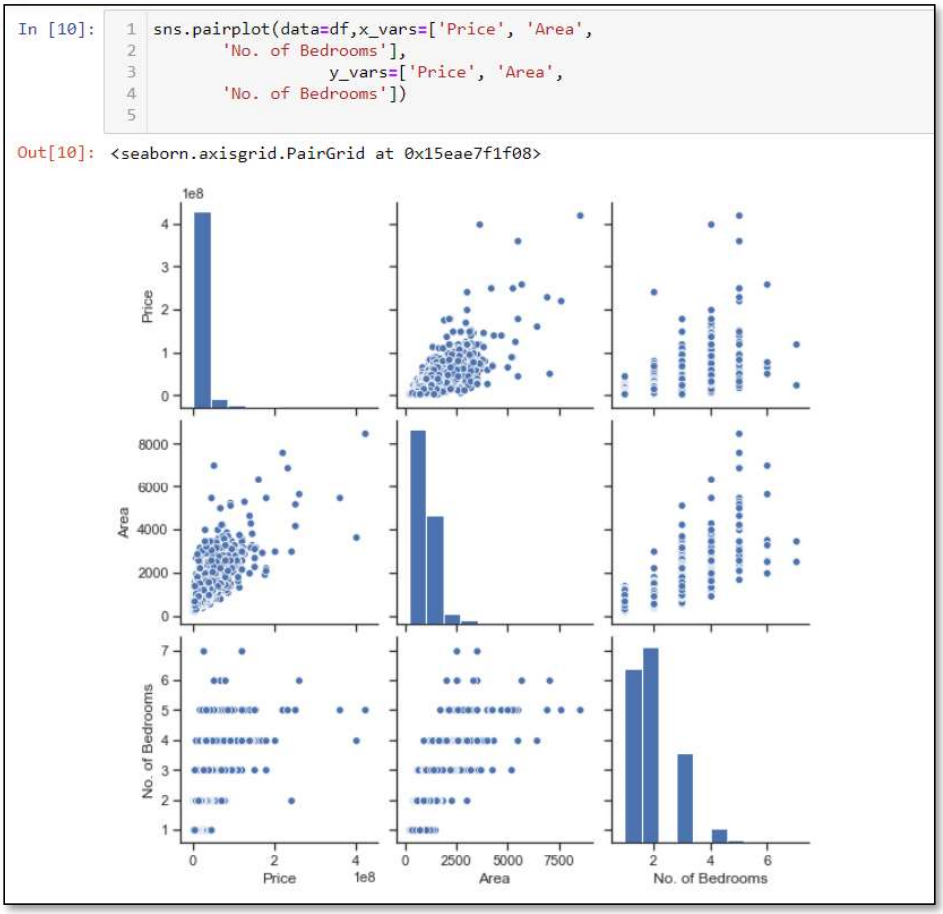


Fig 4.5 Seaborn.pairplot()

This pairplot gives us a rough idea about a parameter's variance with respect to each column. We can see that in the Price vs Area graph as the area increases there are more outliers present in the dataset

and the variance is linear. Also, in the Price vs No. of Bedrooms graph we can see there is an systematic increase in pricing as the number of bedrooms increase. A proper look at the pairplot gives us a fair idea about the presence and quantity of outliers.

Seaborn.distplot()

This function combines the matplotlib hist function (with automatic calculation of a good default bin size) with the seaborn kdeplot() and rugplot() functions. It draws the distribution of a particular parameter with respect to its values. It gives us the fair idea of the concentration of data in a particular range of values.

We have to predict the value in the 'Price' column so it would be helpful to understand the distribution of values in the Price column. That value can help us understand the extent of values of the outliers.

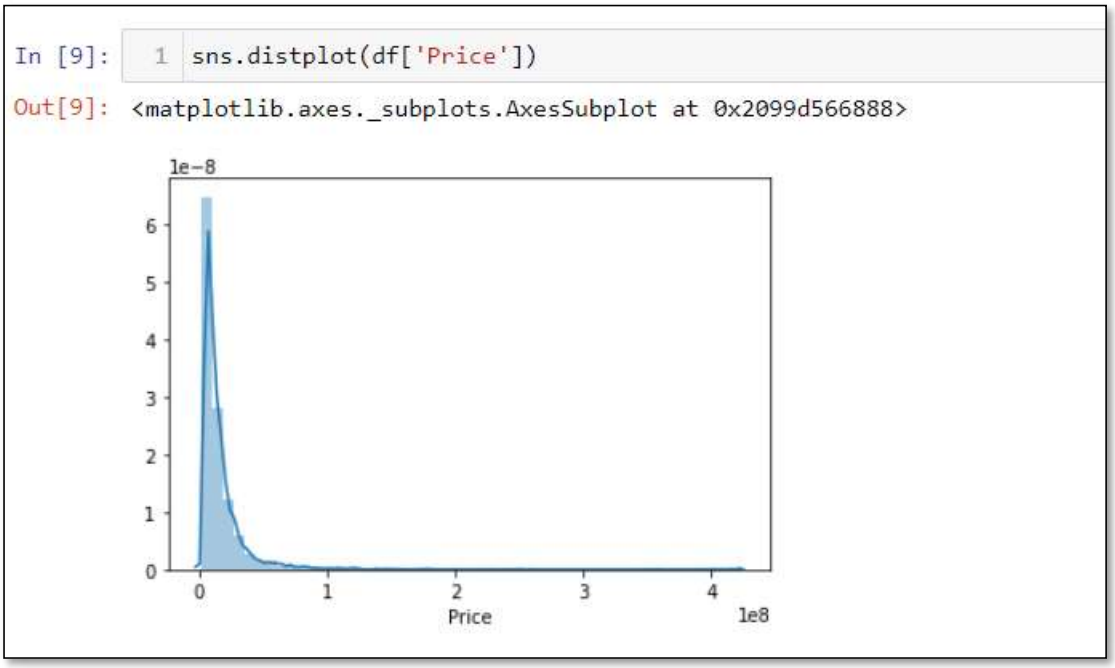


Fig 4.6 Seaborn.distplot()

Here we can see that the majority of values are 0 to 2 crores while the maximum value is around 40 crores. The peak of the distribution is around 1.5 crore. This gives us a rough idea about the variance of Price column values.

Seaborn.heatmap()

It plots rectangular data as a color-encoded matrix. This is an Axes-level function and will draw the heatmap into the currently active Axes if none is provided to the ax argument. Part of this Axes space will be taken and used to plot a colormap, unless cbar is False or a separate Axes is provided to cbar_ax. It color codes the values passed in the function and plots them side by side for us to compare. Generally a correlation function is passed so that the heatmap shows the general correlation between columns this helps us in deciding which factor/parameter affects the intended column ('Price') more.

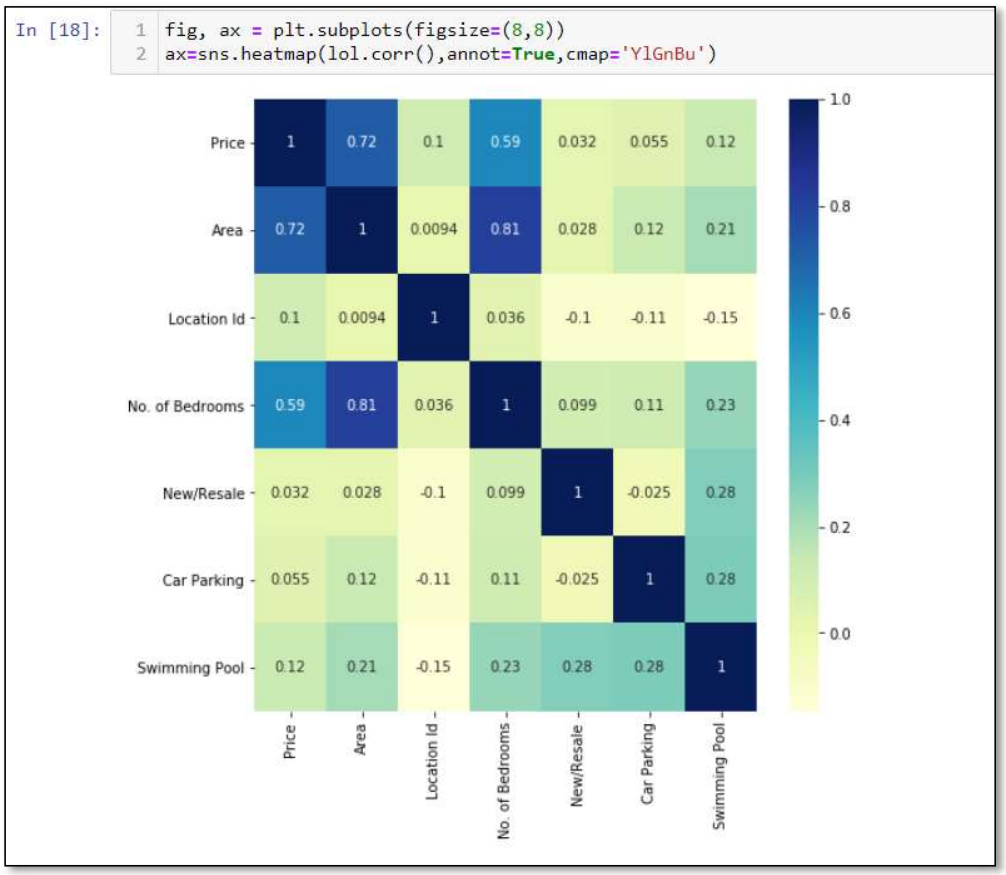


Fig 4.7 Seaborn.heatmap()

The heatmap clearly shows that it is affected the most by 'Area' column values at a correlation of 0.72. followed by 'No of bedrooms' column values at a correlation of 0.59. More the value of correlation nearer to 100 more is its influence on the Price column. Therefore other columns have an insignificant correlation values to be able to influence the Price column more.

4.1.4 DATA CLEANING

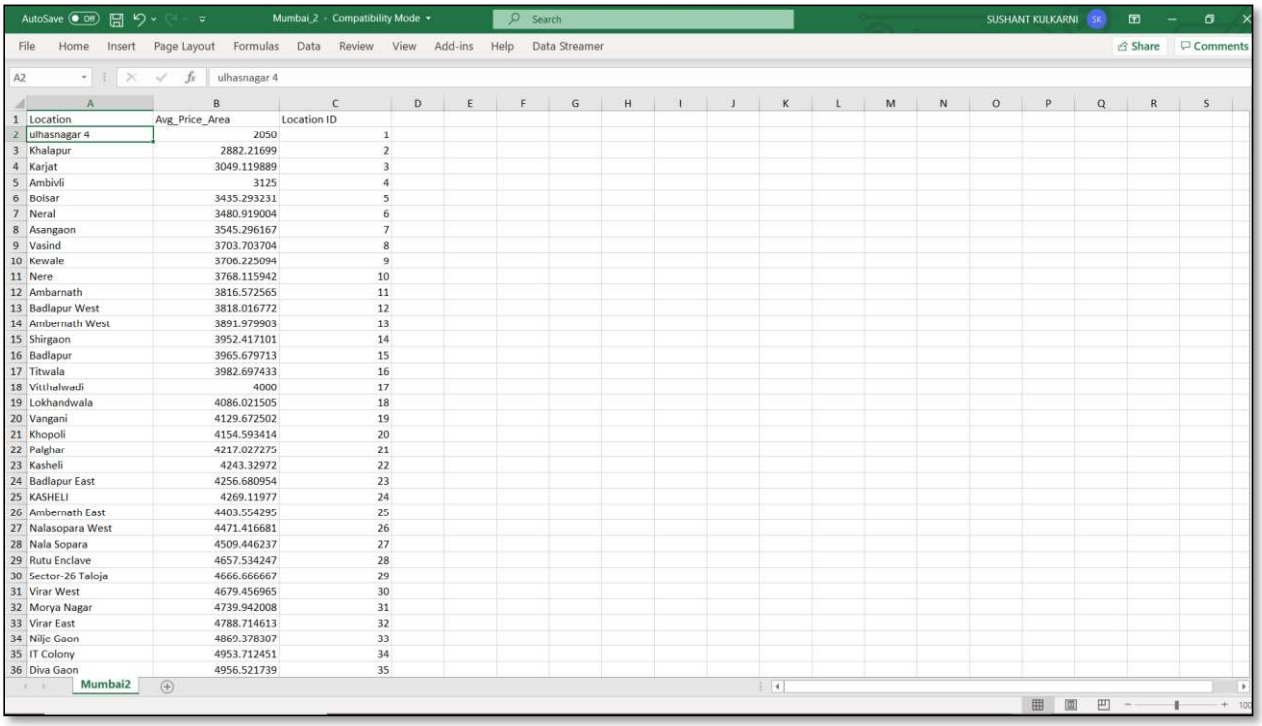
The dataset that was adopted from Kaggle.com had to be processed according to our needs. This involved handling Null values. We got a rough idea about all null values from the info() function. These null values either had to be replaced or the tuples with null values had to be removed. The tuples having null values in the Location, Price, Area, No of Bedrooms column had to be removed as those were columns of high correlation value with Price column. Null values in other binary columns could be easily replaced with either 1 or 0 based on averaging method.

4.1.5 DATA PROCESSING

There were two steps involved in the Data Processing procedure:

Dealing with Location column and introducing Avg_Price_Area column.
Unlike all other column data the Location data is in String format and hence unreadable or unrelatable for machine learning model for connecting it to the Price data. The Location data is very important for Price values and hence should be converted to machine readable format. Hence in another excel sheet('Mumbai2') all the location data was grouped by individual location and there were 413 unique locations. Each location was given a unique location id. This column was then translated back to the main dataset and each location was given a value based on its location Id in its column.

After testing this method, it was found that the error rate was still high, and we couldn't get a better correlation between location_id and Price. To solve this problem another column was created in the Mumbai2 excel sheet for Average Price per Built up area. Here while we grouped using individual locations an average of Price per Area values was taken and stored for every unique 413 value. Now using these values, the data was sorted in ascending order. The location ids were reassigned to the locations now, in ascending order of average price per area column.
Following is a screenshot of the 'Mumbai2' excel sheet:



Location	Avg_Price_Area	Location ID
ulhasnagar 4	2050	1
Khalapur	2882.21699	2
Karjat	3049.119889	3
Ambivli	3125	4
Boisar	3435.293231	5
Neral	3480.919004	6
Asangaon	3545.296167	7
Vasind	3703.703704	8
Kewale	3706.225094	9
Nere	3768.115942	10
Ambarnath	3816.572565	11
Badlapur West	3818.016772	12
Ambernath West	3891.979903	13
Shirgaon	3952.417101	14
Badlapur	3965.679713	15
Titwala	3982.697433	16
Vitthalwadi	4000	17
Lokhandwala	4086.021505	18
Vangani	4129.672502	19
Khopoli	4154.593414	20
Palghar	4217.027275	21
Kasheli	4243.32972	22
Badlapur East	4256.680954	23
KASHELI	4269.11977	24
Ambernath East	4403.554295	25
Nalasopara West	4471.416681	26
Nala Sopara	4509.446237	27
Rutu Enclave	4657.534247	28
Sector-26 Taloja	4666.666667	29
Virar West	4679.456965	30
Morya Nagar	4739.942008	31
Virar East	4788.714613	32
Nilje Gaon	4869.378307	33
IT Colony	4953.712451	34
Diva Gaon	4956.521739	35

Fig 4.8 Location ID and Avg_Price_Area columns.

Dealing with outliers

In statistics, an outlier is a data point that differs significantly from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses. Outliers can occur by chance in any distribution, but they often indicate either measurement error or that the population has a heavy-tailed distribution. In the former case one wishes to discard them or use statistics that are robust to outliers, while in the latter case they indicate that the distribution has high skewness and that one should be very cautious in using tools or intuitions that assume a normal distribution. A frequent cause of outliers is a mixture of two distributions, which may be two distinct sub-populations, or may indicate 'correct trial' versus 'measurement error'; this is modelled by a mixture model. Outliers, being the most extreme observations, may include the sample maximum or sample minimum, or both, depending on whether they are extremely high or low. However, the sample maximum and minimum are not always outliers because they may not be unusually far from other observations. Estimators capable of coping with outliers are said to be robust: the median is a robust statistic of central tendency, while the mean is not.

We need to focus on removing outliers based on the Price column values. We need to deal with the over bloated Price values and under stated price values as they may harm the model's prediction capabilities. For this we have used the Quantile and IQR method. The data is arranged in ascending order and divided into 4 parts by 3 intersections Q1, Q2, Q3. Now we need to find and IQR value using which values below Q1-IQR*1.5 and values above Q3+IQR*1.5 will be deemed outliers and removed. The df_out dataframe will have data without the outliers present in df as following:

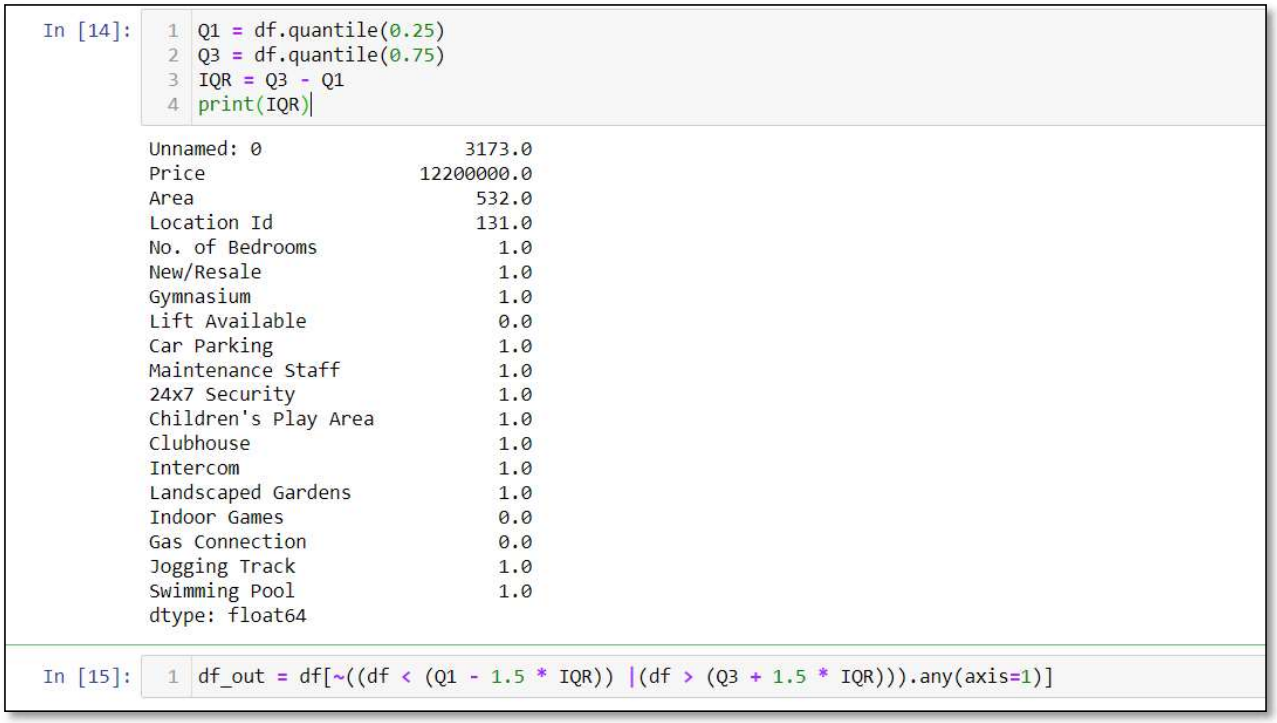


Fig 4.9 Removing Outliers.

The Scatterplot gives us a clear idea of data with and without the outliers. Fig 4.9 shows data which has points scattered and the data doesn't look nuclear because of the existence of outliers. On the other hand, Fig 4.10 shows a more nuclear picture of data without outliers.

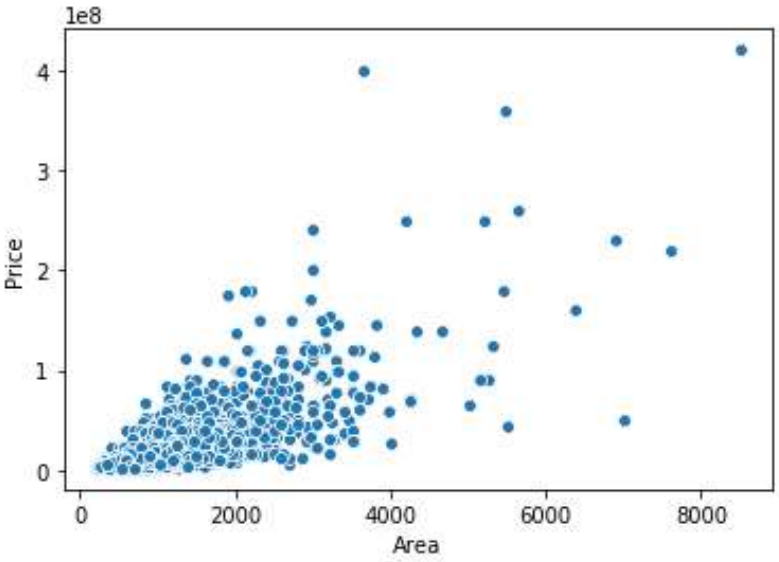


Fig 4.10 Scatterplot before removing outliers.

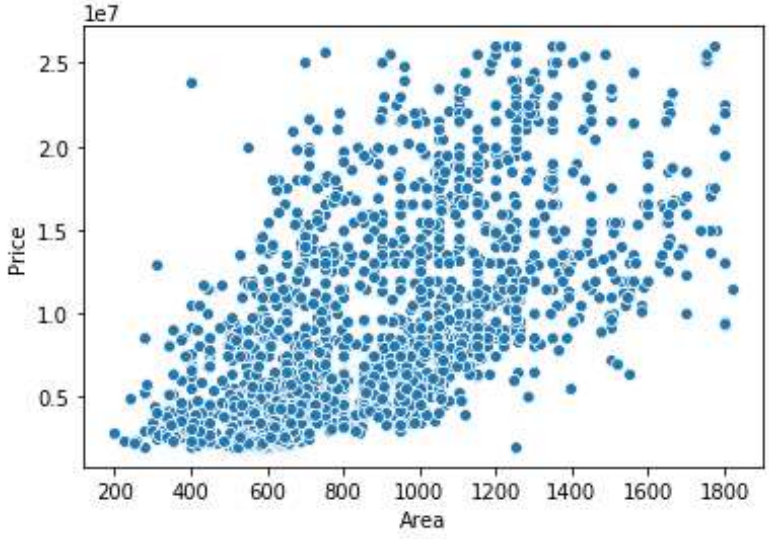


Fig 4.11 Scatterplot after removing outliers.

4.2 MACHINE LEARNING

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect. It is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step. The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers.

In our system Machine Learning plays an integral part in ‘prediction’ of the house prices. It relies on the method of feature learning to understand all features of the input and map its prediction algorithm according to that learning. Machine Learning process involves deciding on a Machine Learning model/algorithm, training of the model, fitting the model and then finally getting prediction as output for a given set of input. All the models are trained on training set of data and then tested on the testing data which is unseen by the model. This testing helps in determining the efficiency or the accuracy of that particular model.

In our case we deployed various models, tested them and noted the error metrics for each model. These metrics helped us determine the 3 best algorithms matching our requirements and giving minimal error. These 3 algorithms/models are:

- Linear Regression
- K-Nearest Neighbors
- Decision Tree

All the models and algorithms were implemented using the TensorFlow library.

4.2.1 LINEAR REGRESSION

Linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Simple linear regression algorithm analyses the relation between two entities where one is dependent and the other is independent. Change in the independent entity reflects the change in the dependent entity. This algorithm does not calculate the dependency instead only the association between the two entities or variables. The equation of the line of linear regression is as follows:

$$y = A + Bx \tag{1}$$

Here X is an independent variable and Y is a dependent variable. A is the intercept whereas B is the slope of the line. In linear regression, the observations are assumed to be the result of random deviations from an underlying relationship between a dependent variable (y) and an independent variable (x). Here, we have trained the linear regression model using the training dataset and then tested it on the testing dataset to make predictions.

Below given Fig.4.11 has a scatter plot between the original house price value and the predicted house price value by the model.

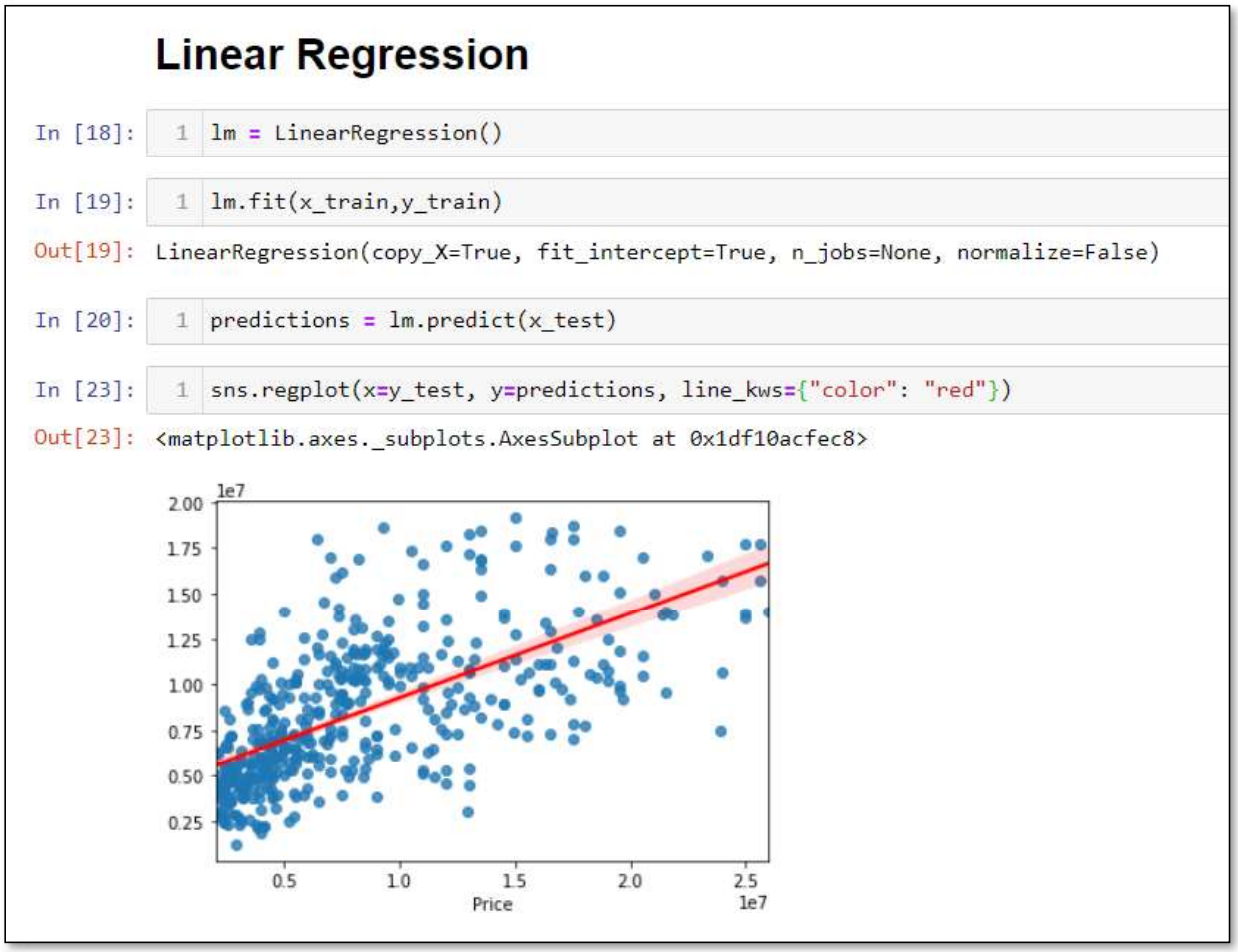


Fig 4.12 Linear Regression Implementation

Below given Table 1 mention the error metrics of the model: Mean Squared Error (MSE), Mean Absolute Error (MAE), R2 Score, and Root Mean Squared Error (RMSE).

MAE	MSE	RMSE	R2
2229371.20	9247442528422.93	3040960.80	0.83

Table 2. Linear Regression Error Metrics

4.2.2 K-NEAREST NEIGHBORS

K-Nearest Neighbor (KNN) is a machine learning algorithm that does regressive as well as classification predictive analysis. It is also called a lazy learner algorithm since it does not analyze the data it is trained with instead the algorithm only classifies the new data it is tested with by its similarity. Here, KNN uses feature similarity to predict the house price values, i.e., it assigns a value to the new data based on how closely it relates (using distance functions for continuous variables such as Euclidean (2) and Manhattan (3) distance functions) to the points in the training set.

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \qquad (2)$$

$$\sum_{i=1}^{k}|x_i - y_i| \qquad (3)$$

Where X refers to the new point, Y refers to the existing point and K is the K-Factor (no. of neighbors the algorithm looks at before assigning a value).

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). In statistics, the k-nearest neighbors algorithm (k-NN) is a non-parametric classification method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover. It is used for classification and regression. In both cases, the input consists of the k closest training examples in data set. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph.

Below given Fig.4.12 has a scatter plot between the original house price value and the predicted house price value by the model.

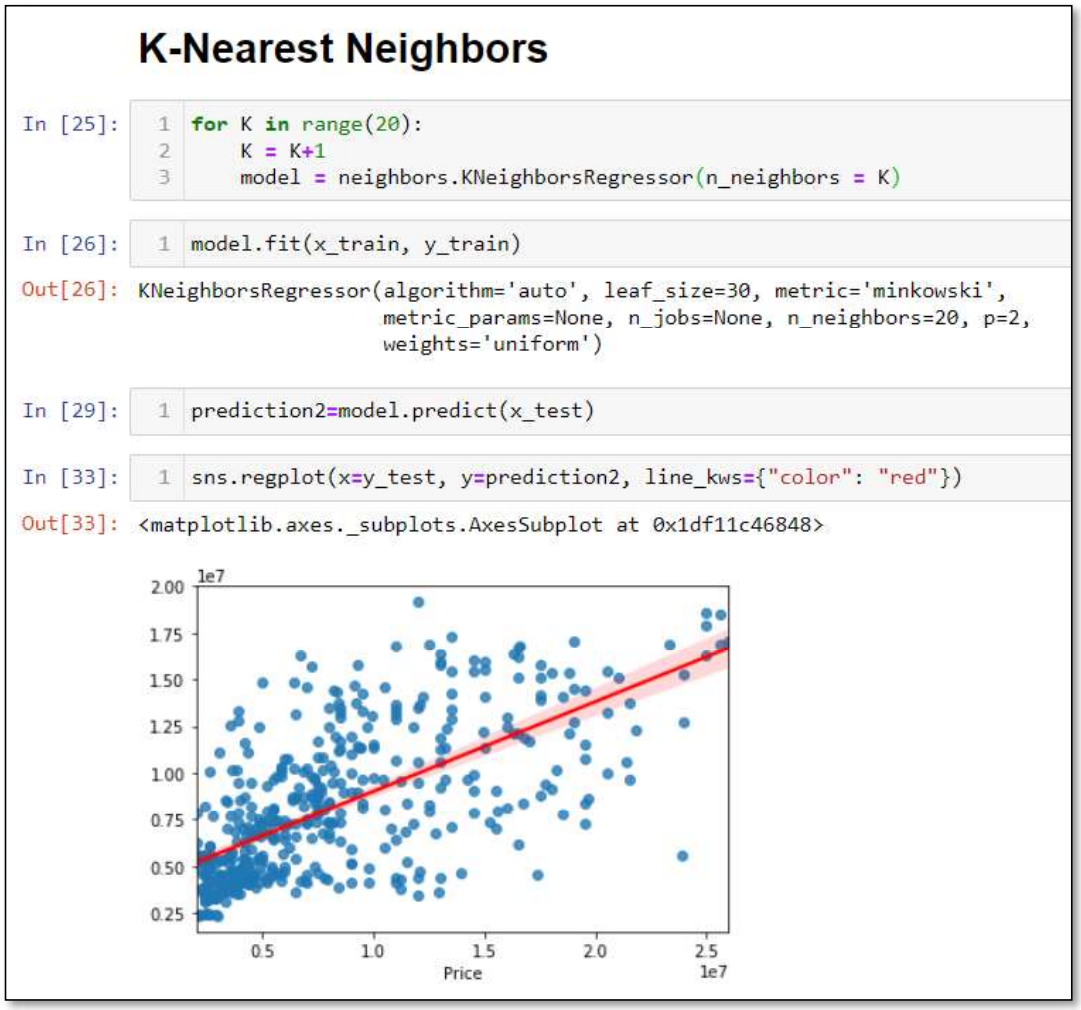


Fig 4.13 K-Nearest Neighbors Algorithm Implementation

Below given Table. II mentions the error metrics of the model: Mean Squared Error (MSE), Mean Absolute Error (MAE), R2 Score, and Root Mean Squared Error (RMSE).

MAE	MSE	RMSE	R2
1948884.96	8927720068103	2987929.06	0.84

Table 3. KNN Error Metrics

4.2.3 DECISION TREE

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks. It is a tree-structured classifier with three types of nodes. The Root Node is the initial node which represents the entire sample and may get split further into further nodes. The Interior Nodes represent the features of a data set and the branches represent the decision rules. Finally, the Leaf Nodes represent the outcome. This algorithm is very useful for solving decision-related problems. With a particular data point, it is run completely through the entirely tree by answering True/False questions till it reaches the leaf node. The final prediction is the average of the value of the dependent variable in that particular leaf node. Through multiple iterations, the Tree is able to predict a proper value for the data point.

A decision tree algorithm builds the model in a tree structure with decision nodes and leaf nodes. It breaks down the dataset into smaller sets with similar values and the highest node is known as the root node. The tree is made of only conditional control statements with each decision node testing an attribute. Starting from the root, the data is split on the feature that results in the largest Information Gain **(**IG**)**. In an iterative process, we then repeat this splitting procedure at each child node until the leaves are pure, i.e. samples at each node all belong to the same class. In order to split the nodes at the most informative features, we need to define an objective function that we want to optimize via the tree learning algorithm. Here, our objective function is to maximize the information gain at each split, which we define as follows:

$$IG(D_p, f) = I(D_p) - \left(\frac{N_{left}}{N_p} I(D_{left}) + \frac{N_{right}}{N_p} I(D_{right})\right)$$

Here, *f* is the feature to perform the split, Dp, Dleft, and Dright are the datasets of the parent and child nodes, *I* is the impurity measure, Np is the total number of samples at the parent node, and Nleft and Nright are the number of samples in the child nodes.

Below given Fig.4.13 is a scatter plot between the original house price value and the predicted house price value by the model.

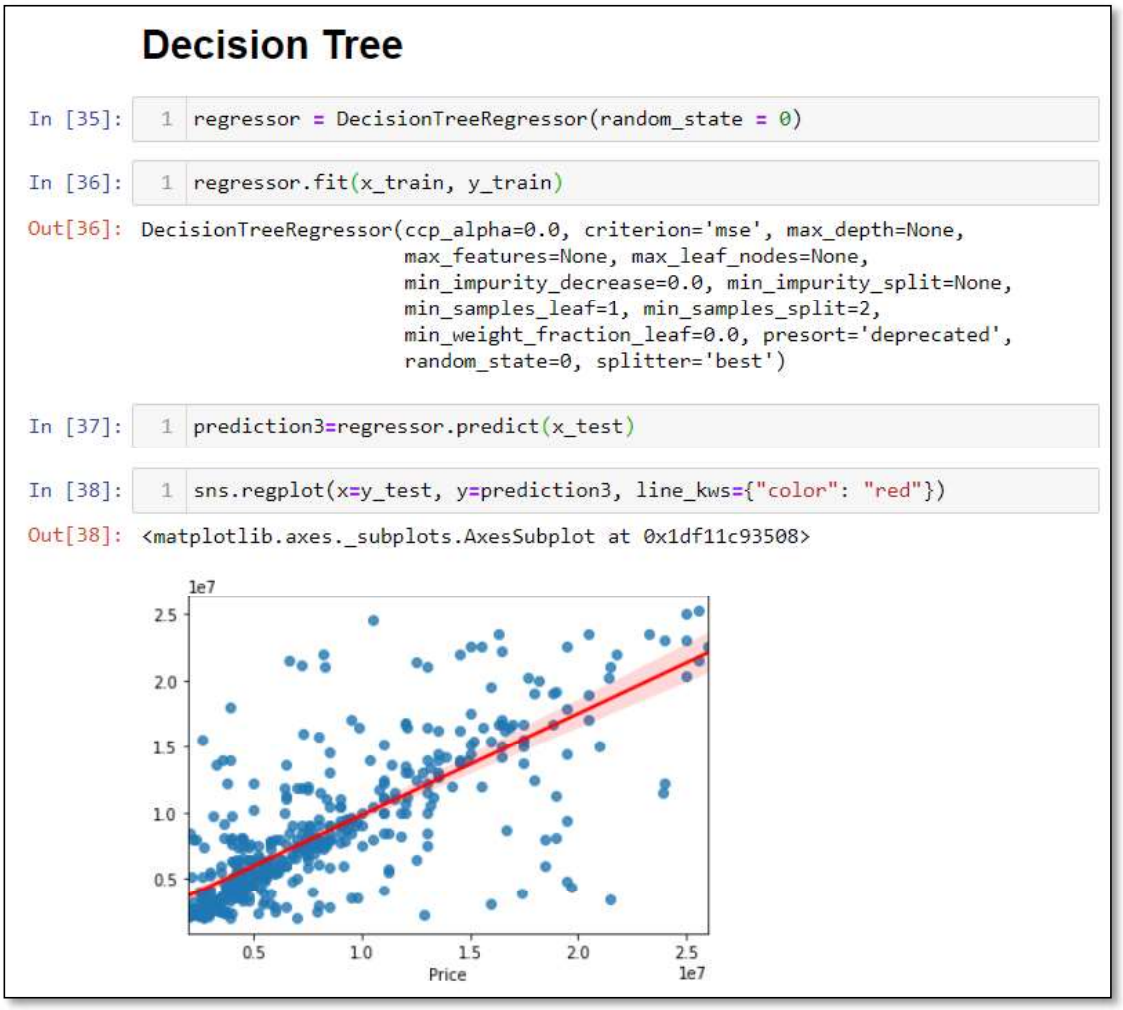


Fig 4.14 Decision Tree Algorithm Implementation

Below given Table. III mentions the error metrics of the model: Mean Squared Error (MSE), Mean Absolute Error (MAE), R2 Score, and Root Mean Squared Error (RMSE).

MAE	MSE	RMSE	R2
1851282.60	10135014746386	3183553.80	0.81

Table 4. Decision Tree Error Metrics

4.2.4 ENSEMBLE LEARNING

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

Evaluating the prediction of an ensemble typically requires more computation than evaluating the prediction of a single model. In one sense, ensemble learning may be thought of as a way to compensate for poor learning algorithms by performing a lot of extra computation. On the other hand, the alternative is to do a lot more learning on one non-ensemble system. An ensemble system may be more efficient at improving overall accuracy for the same increase in compute, storage, or communication resources by using that increase on two or more methods, than would have been improved by increasing resource use for a single method. Fast algorithms such as decision trees are commonly used in ensemble methods (for example, random forests), although slower algorithms can benefit from ensemble techniques as well.

A weighted ensemble is an extension of a model averaging ensemble where the contribution of each member to the final prediction is weighted by the performance of the model. The model weights are small positive values, and the sum of all weights equals one, allowing the weights to indicate the percentage of trust or expected performance from each model. In our model we have given 10% weightage to the Linear Regression algorithm, 30% weightage to KNN algorithm and 60% weightage to Decision Tree algorithm. These weights were finalized after considering the overall errors of all the 3 algorithms individually and then tested for optimal error value of the ensemble model for different weights.

The ensemble learning model is a combination of algorithms or models which help to improve predictions depending on the features of the dataset. We could conclude from our experimental results that a weighted average of predictions from Linear Regression, KNN, and Decision Tree models provided the lowest error values compared to predictions from individual algorithms from Fig. 4.14

and Table. 5 below. The weights assigned to the predictions of models are based on its performance and features of our dataset exclusively.

```
In [13]:
mean_predictions=[]
temp=0
for i in range(len(x_test)):
    temp = predictions[i]*0.1+prediction2[i]*0.3 + prediction3[i]*0.6
    mean_predictions.append(temp)
    temp = 0

In [14]:
sns.regplot(x=y_test, y=mean_predictions, line_kws={"color": "red"})

Out[14]:
<matplotlib.axes._subplots.AxesSubplot at 0x25fc2c22908>
```

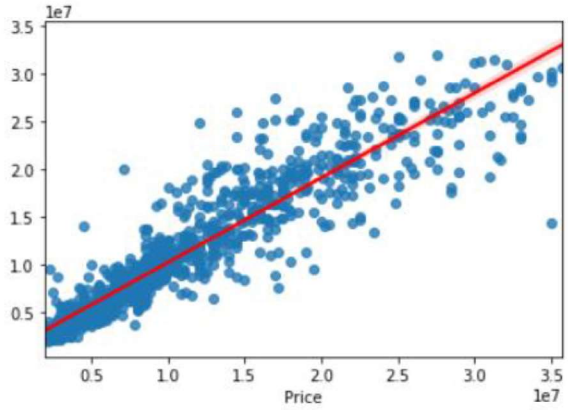


Fig 4.15 Ensemble Learning Implementation

Below given Table. V mentions the error metrics of the model: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R2 Score, and Weight Assigned (W).

MODEL	MAE	MSE	RMSE	R2	W
Linear Reg.	2229371.20	9.2E+10	3040960.80	0.83	0.1
KNN	1948884.96	8.9E+!0	2987929.06	0.84	0.3
Decision Tree	1851282.60	1.0E+09	3183553.80	0.81	0.6
Ensemble	1658701.38	7.3E+10	2694486.53	0.87	-

Table 5. Comparison of Error Metrics

Chapter 5

Planning and Scheduling

5. PLANNING AND SCHEDULING

5.1 GANTT CHART

A Gantt chart is a type of bar chart that illustrates a project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. The width of the horizontal bars in the graph shows the duration of each activity. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements constitute the work breakdown structure of the project.

The following Gantt chart was prepared after taking into consideration the time frames of each task and its complexity. Tasks such as Requirement Gathering, and Testing ML models required standalone preferences while other tasks could be handled simultaneously with others. Literature survey, Data Collection, Processing, Algorithm Design, System Testing and Final documentation were tasks which spanned over multiple months and had to be mapped accordingly.

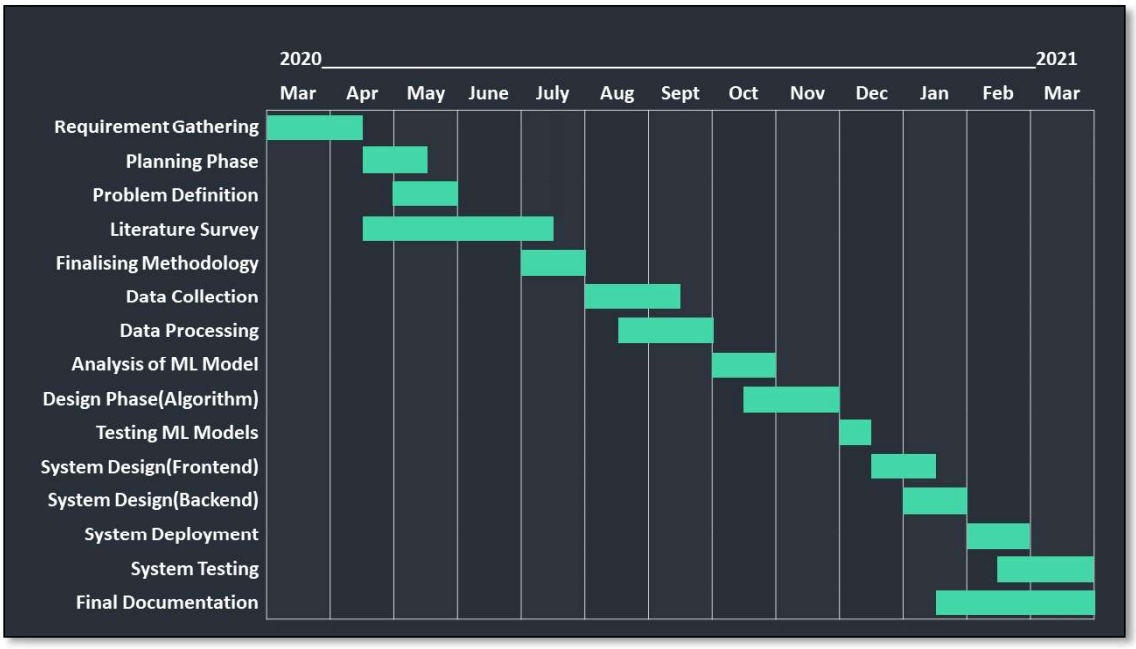


Fig 5.1 Gantt Chart

5.2 PERT

A PERT chart is a network diagram used in the Program Evaluation Review Technique (PERT) to represent a project's timeline. It allows project managers to estimate the duration of projects based on the analysis of task sequences. PERT charts are used by project managers to create realistic schedules by coordinating activities and estimating their duration by assigning three time estimates for each (optimistic, most likely, and pessimistic). This makes PERT charts useful when planning projects where the duration of activities is uncertain. A PERT chart network diagram includes numbered nodes, directional arrows and divergent arrows that illustrate the minimum time and duration of activities. Directional arrows represent the activities, while nodes are milestones.

This PERT table has 8 components Requirement Gathering, Literature survey, Planning, Implementation, Designing Interface, Testing, Deployment, Documentation. Each component is mutually exhaustive and do not overlap in functionality. Having said that they are dependable on one another for commencement and completion. These standalone activities are spread over a period of 13 months or 396 days.

Activities	Duration (days)	Immediate Predecessors
A. Requirement Gathering	45	-
B. Literature Survey	60	A
C. Planning	75	B
D. Implementation	120	C
E. Designing Interface	45	D
F. Testing	60	D,E
G. Deployment	30	F
H. Documentation	75	D,F,G

Table 6. PERT Table

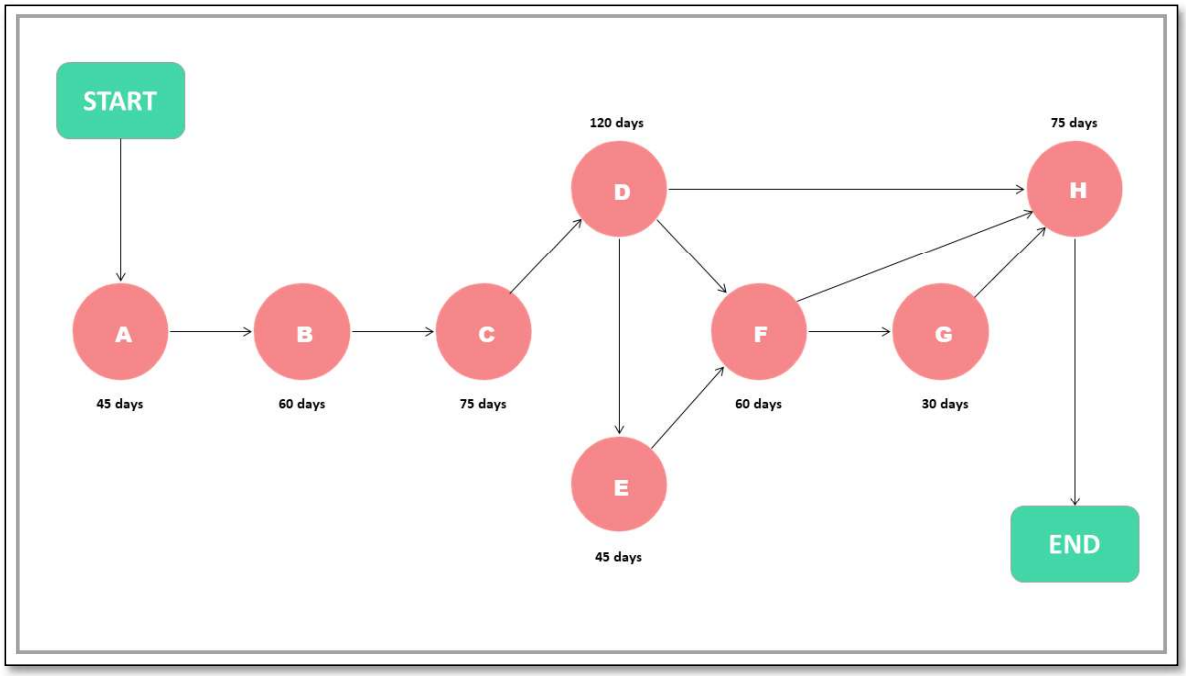


Fig 5.2 PERT Chart

Chapter 6

Testing and Maintenance

6. TESTING AND MAINTENANCE

6.1 TESTING

Software testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software testing is a process that detects important bugs with the objective of having better quality software.

6.1.1 TESTING TECHNIQUES

Basically, there are 3 testing methodologies which are used for testing. They are White Box Testing, Black Box Testing, and Grey Box Testing. These are also called as Testing Techniques. Each of the testing technique is briefed below:

White Box Testing

White box testing technique is used to examine the program structure and business logic, it validates the code or program of an application. It is also called as Clear Box Testing, Glass Box Testing or Open Box Testing.

White Box Testing Techniques include:

- Statement Coverage: Examines all the programming statements.
- Branch Coverage: Series of running tests to ensure if all the branches are tested.
- Path Coverage: Tests all the possible paths to cover each statement and branch.

Black Box Testing

Black Box testing method is used to test the functionality of an application based on the requirement specification. Unlike White Box Testing it does not focus on internal structure/code of the application.

Black Box Techniques include:

- Boundary Value analysis
- Equivalence Partitioning (Equivalence Class Partitioning)
- Decision Tables
- Domain Tests
- State Models
- Exploratory Testing (Requires less preparation and also helps to find the defects quickly).

Grey Box Testing

This method of testing is performed with less information about the internal structure of an application. Generally, this is performed like Black Box Testing only but for some critical areas of application, White Box Testing is used.

6.1.2 TESTING METHODOLOGY

Methodologies can be considered as the set of testing mechanisms used in software development lifecycle from Unit Testing to System Testing. Selecting an appropriate testing methodology is considered to be the core of the testing process.

V Model

V Model is an extension of Waterfall Model where the process execution takes place in a sequential style in V-Shape and is also known as Verification and Validation Model. In this approach, there exists a directly associated testing phase in every single phase of the development cycle. It has been proven

beneficial and cost-efficient than the waterfall model as the testing is performed at each development phase rather than at the end of the development cycle.

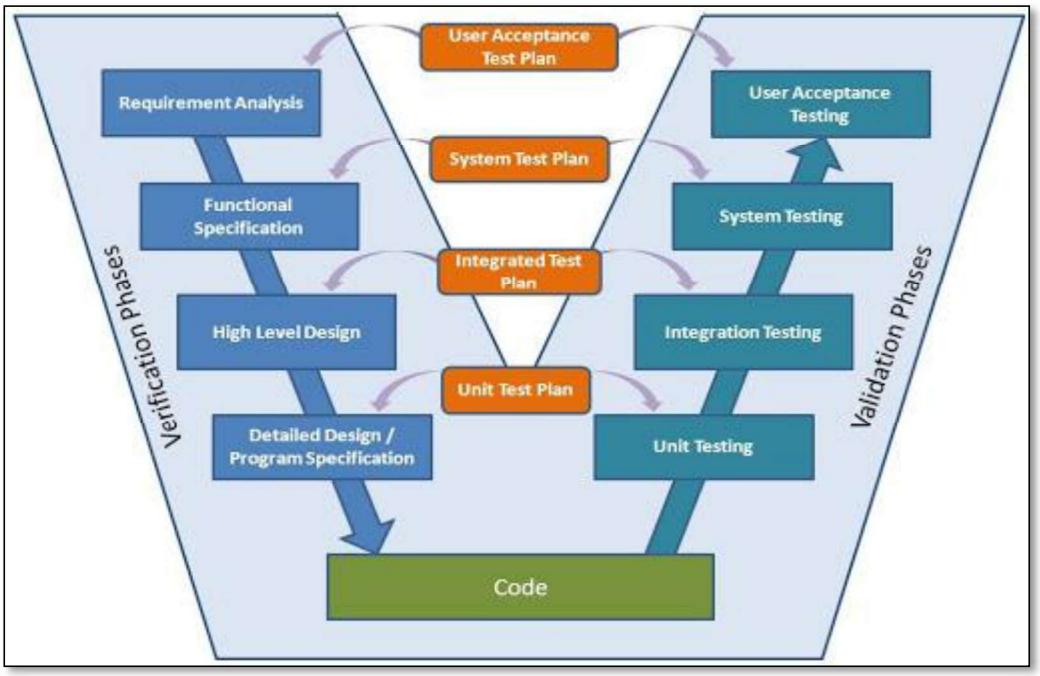


Fig 6.1 V Model

V Model is classified into 3 Phases:

Verification Phase

- Business Requirement Analysis: Communicate with the customer to understand their expectations and requirements.
- System Design: Design complete system and its components along with the hardware and software requirements.
- Architectural Design: In this phase architectural specifications are captured. This also known as high-level Design.
- Module Design: This is also known as Low-Level Design, Detailed internal design for all the specified system modules.

Coding Phase

This phase contains actual coding phase in the development lifecycle. Programming languages should be chosen based on the system and architectural design specified in the previous phase technology platform. Coding is performed according to the standards and guidelines that are pre-defined.

Validation Phase

- Unit Testing: Performed on an individual module to eliminate the bugs at the early stage.
- Integration Testing: Performed to test the communication between different modules in the system.
- System Testing: System Testing is performed on a system as a whole.
- Acceptance Testing: This is associated with the business requirements. It is performed in a user environment from the user's point of view.

Advantages of V model

- Simple, easy to use and understand.
- Overlapping is avoided as phases are executed one at a time.
- Easy to manage and suitable for small projects.

6.1.3 TESTING OF WEB BASED SYSTEM

Web-based systems have a different nature as compared to traditional systems. We have seen the key differences between them. Due to the environment difference and challenges of dynamic behavior, complexity and diversity alsomakes the testing of these systems a challenge. These systems need to be tested not only to check whether it does what it is designed to do but also to evaluate how well it appears on the (different) web browsers. Moreover, they need to be tested for various quality

parameters which are a must for these systems like security, usability, etc. Hence, a lot of effort is required for test planning and test designing.

Interface Testing

Interface is a major requirement in any web application. More importantly, the user interface with web application must be proper and flexible. Therefore, as a part of verification, present model and web scenarios model must be checked to ensure all interfaces. The interfaces between the concerned client and servers should also be considered. There are two main interfaces on the server side: web server and application server interface and application server and database server interface.

Usability Testing

The presentation design emphasizing the interface between user and web application gives rise to usability testing. The actual user of application should feel good while using the application and understand everything visible to him on it. Usability testing is not a functionality testing, but the web application is reviewed and tested from a user's viewpoint.

Navigation Testing

To ensure the functioning of correct sequence of those navigations, navigation testing is performed on various possible paths in the web application. Design the test cases such that the following navigations are correctly executing:

- Internal links
- External links
- Redirected links
- Navigation for searching inside the web application.

Configuration Testing

Diversity in configuration for web applications makes the testing of these systems very difficult. Therefore, configuration testing becomes important so that there is compatibility between various available resources and application software.

Security Testing

Through security testing, we try to ensure that data on the web applications remain confidential, i.e. there is no unauthorized access. Security testing also ensures that users can perform only those tasks that they are authorized to perform.

Performance Testing

Performance testing helps the developer to identify the bottlenecks in the system and can be rectified. Bottlenecks for web applications can be code, database, network, peripheral devices, etc.

6.2 MAINTENANCE

The Institute of Electrical and Electronics Engineers (IEEE) describes software maintenance as the modification of software after delivery to the user. The reasons for these changes include correcting faults, improving performance, and adapting the software to changes in requirements. All software requires maintenance, even when the software, its operating environment, and its requirements are completely stable. Minimizing maintenance costs becomes more important as the software's complexity increases since they often exceed the initial cost of developing the software.

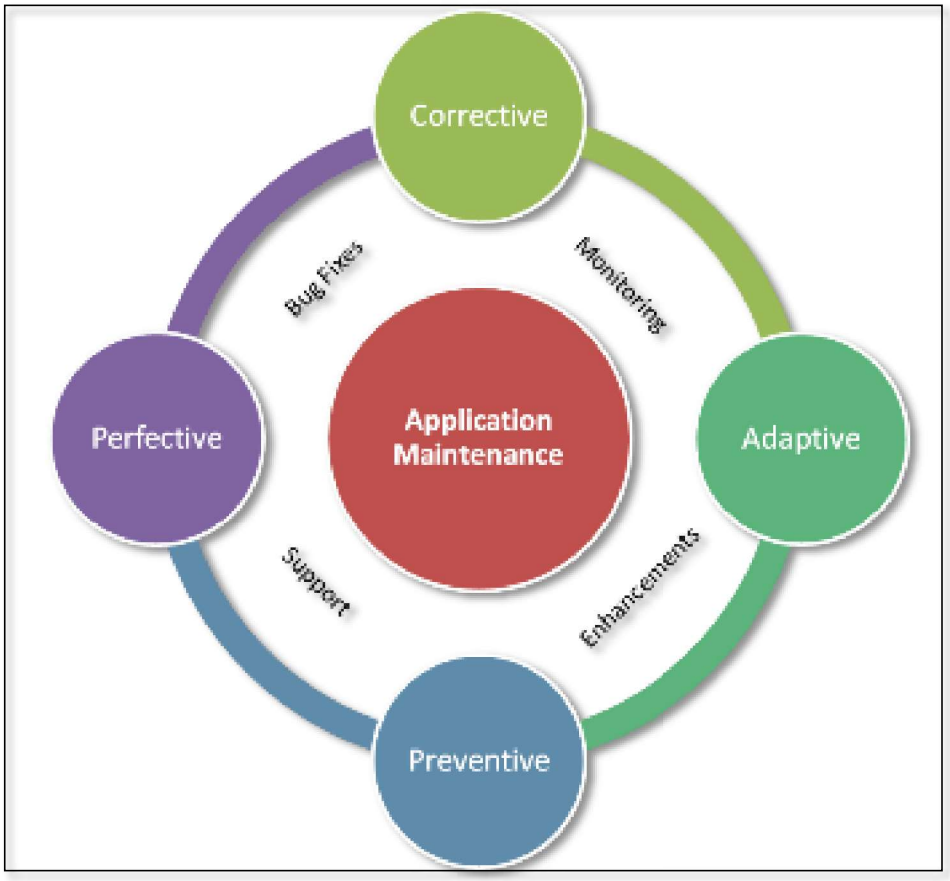


Fig 6.2 Types of Maintenance

Corrective Maintenance

Corrective maintenance is a reactive modification of a software product to correct a known problem. This type of maintenance fixes defects in software, which often takes the form of quick updates performed on a recurring basis. Corrective maintenance is unlikely to have a negative impact on users, rarely complain about getting bugs fixed. It also provides a rapid return on investment (ROI) in improving user experience.

Adaptive Maintenance

Adaptive maintenance is the modification of software to keep it usable after a change to its operating environment. Many factors can change an application's environment, including new technical knowledge, hardware and security threats. These changes occur with greater frequency in most

environments, so software that does not receive regular adaptive maintenance quickly becomes outdated. Adaptive changes primarily affect the software's ability to run on a particular platform, which includes the operating system (OS), hardware and network. These changes tend to have a low impact on users since they focus on the software's internal functioning. Integrating an existing application with new technology may result in a slight improvement in performance areas such as scalability and speed, but its overall functioning generally remains unaffected.

Preventive Maintenance

Preventive maintenance is the modification of software to detect and correct software errors before they take effect. This type of maintenance is also commonly known as future proofing. It includes making the software easier to scale more easily in response to increased demand and fixing latent faults before they become operational faults. Preventative maintenance is almost always completely transparent to the user, as it involves preparation rather than major changes. However, it can have a great impact later by facilitating highly visible changes in addition to increasing the software's overall stability.

Perfective Maintenance

Perfective maintenance improves the software's functionality and usability. It includes refining and deleting existing features as well as adding new features, easily making it the largest category of software maintenance. In addition to changing an application's functionality, perfective maintenance can also affect the way it looks. Changes to the software's interface and user journey are thus part of perfective maintenance. The scope and nature of perfective maintenance also makes this category the most likely to elicit protests from users. Changes to the backend code are noticeable by users, but changes to the front end are highly visible. Managing perfective maintenance therefore requires greater communication with users to mitigate this negative sentiment.

Chapter 7

System Implementation

7. SYSTEM IMPLEMENTATION

7.1 WORKING

After finalizing the Machine Learning Algorithms and after testing the models it is time to implement them to predict house prices. We have developed a website interface for getting user inputs and displaying the predicted price output. Therefore, we need to integrate the python script containing models and the ensemble learner, with the website backend. This can be done using various ways. We are using Flask Library in python for this purpose. Following is a stepwise explanation of the building process of the system.

7.1.1 CREATING PICKLE FILES OF ML MODELS

Pickle is a library in python used for serializing and deserializing data. We have a single dataset which we will split once and train and fir out model once. Now every time the user enters a different set of inputs, we do not want the program to train and fit he model again as it will increase the time complexity and might lead to overfitting issues. Instead, we can run the training and fitting code for the models once and store it inside a .pkl file (pickle file). A pickle file stores a serial form of the model. Now every time the user enters a set of input values, we need to access this .pkl file and then predict our output value. Following is a folder snippet showing creation of pickle files.

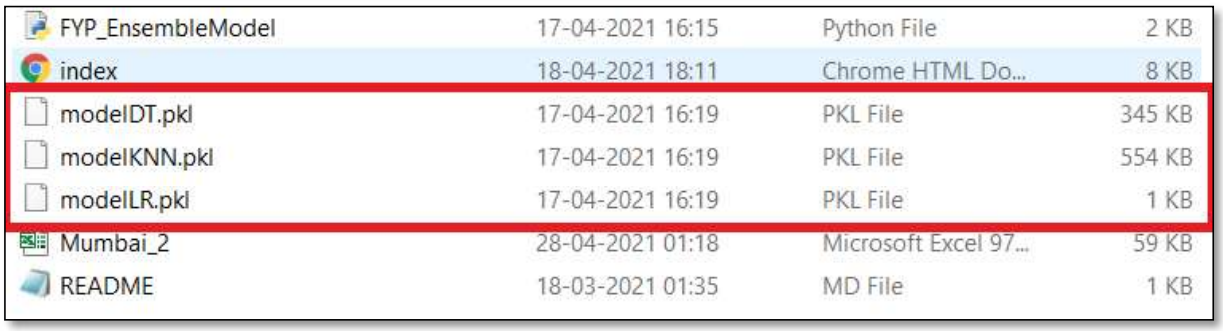


Fig 7.1 Creation of pickle files

Shown below is a code snippet showing training an fitting of the models, followed by initiating creation of pickle files for each ML model.

```
39
40  x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=101)
41
42  #Linear Regression
43
44  lm = LinearRegression()
45  lm.fit(x_train,y_train)
46  pickle.dump(lm,open('modelLR.pkl','wb'))
47  modelLR=pickle.load(open('modelLR.pkl','rb'))
48
49  #KNN
50
51  from sklearn import neighbors
52  for K in range(20):
53      K = K+1
54      knn = neighbors.KNeighborsRegressor(n_neighbors = K)
55
56      knn.fit(x_train, y_train)
57  pickle.dump(knn,open('modelKNN.pkl','wb'))
58  modelKNN=pickle.load(open('modelKNN.pkl','rb'))
59
60  #Decision Tree
61
62  from sklearn.tree import DecisionTreeRegressor
63  dt = DecisionTreeRegressor(random_state = 0)
64  dt.fit(x_train, y_train)
65  pickle.dump(dt,open('modelDT.pkl','wb'))
66  modelDT=pickle.load(open('modelDT.pkl','rb'))
67
```

Fig 7.2 Creating Pickle Files of ML Models

7.1.2 CREATING A WEB INTERFACE

The web interface will the mean of communication between the user and the Ensemble model. The web portal is created taking into consideration ease of use, functionality, and design thinking. It is simple and hardcoded in HTML and designed using Bootstrap CSS. The web page has a HTML form containing all the input fields required by the ML model as input. It has text boxes for Location, Area and No. of Bedrooms input and has checkboxes for Parking, Swimming pool, Lift and Resale/New inputs, as these have binary values. The predicted price is displayed in block numbers besides this form. The checkboxes take Boolean input and convert them to 0/1 while sending it to the model. Following is the snippet for HTML and JavaScript code for the web page:

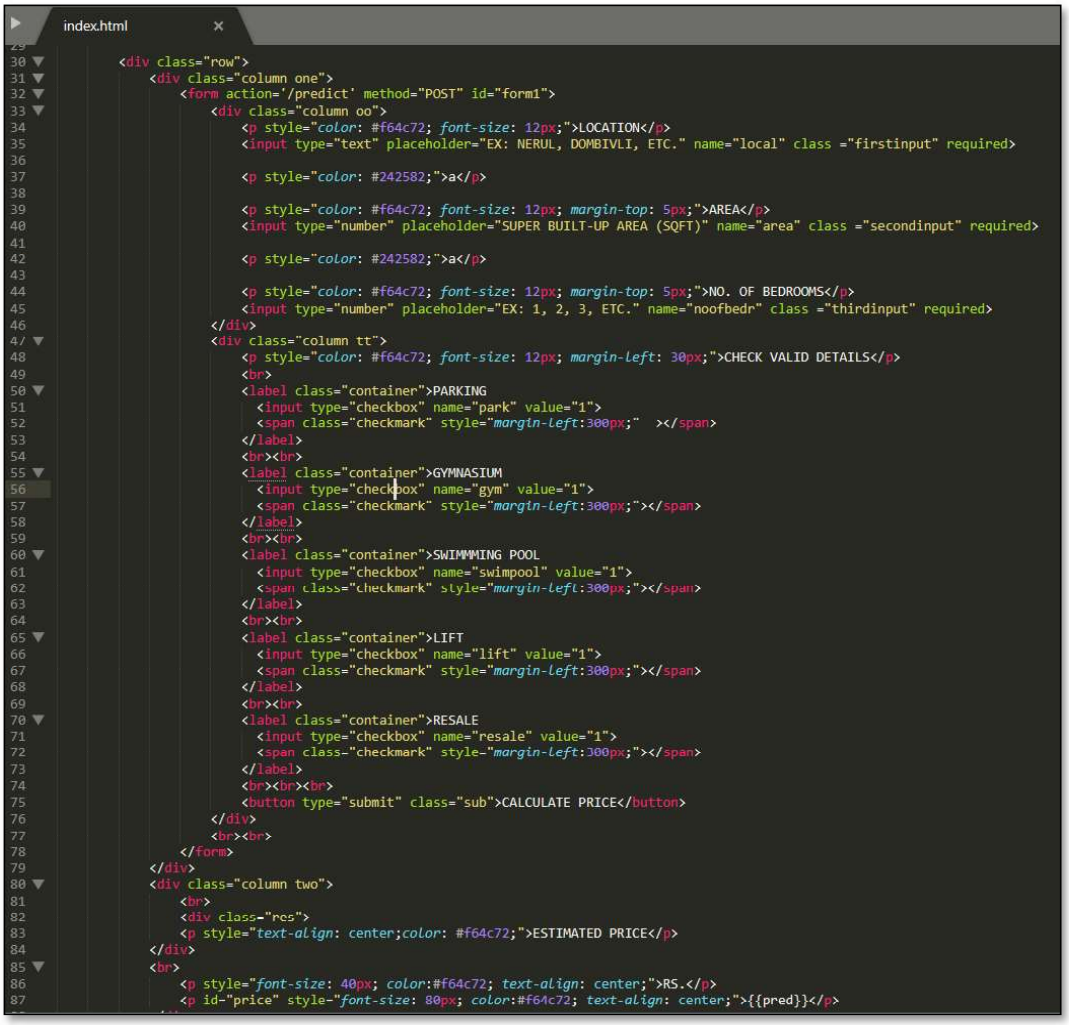


Fig 7.3 HTML Form code snippet

Following is the code snippet of JavaScript code used for handling checkbox inputs:

Fig 7.4 JavaScript code for handling checkbox input

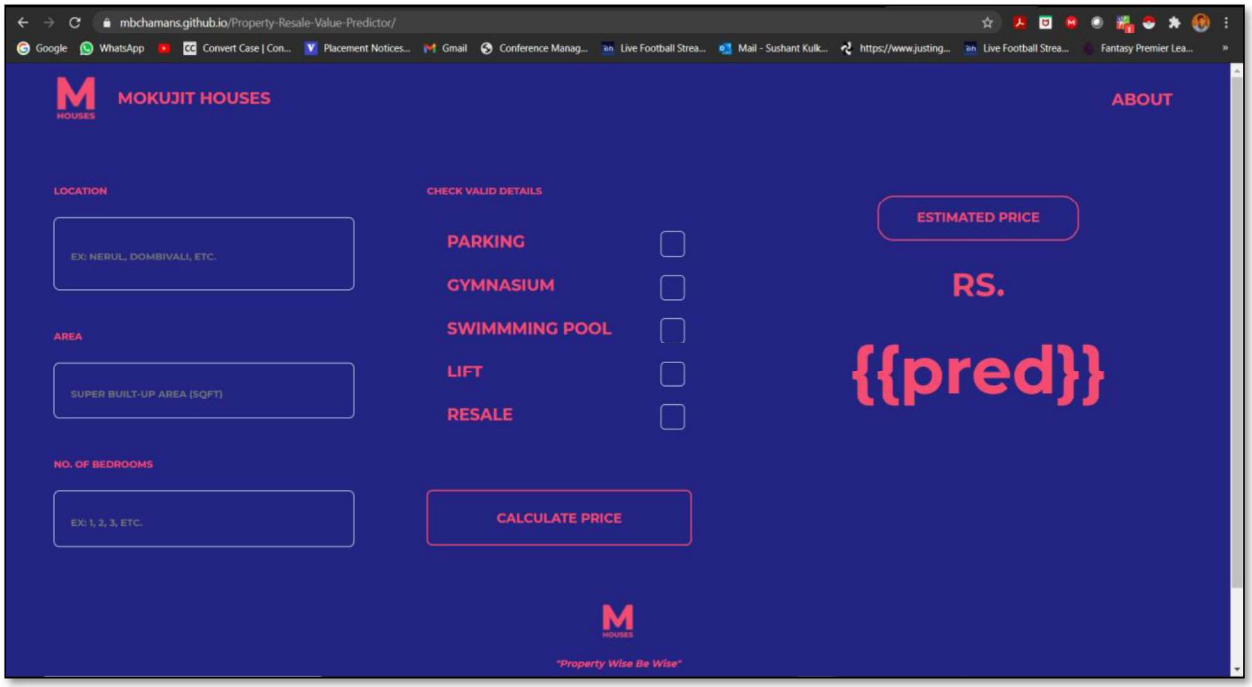


Fig 7.5 Webpage

7.1.3 CREATING AND DEPLOYING FLASK APPLICATION

Flask is a micro web framework written in Python. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Since we have implemented a Machine Learning code written in python into a webpage, we have used the Flask framework. We create the root file of app.py and load all the pickle files of our models. Then we write functions for getting the values from the HTML form. Those values are stored in an array and fed to models as input the ensemble model then produces the output value which is then pushed to the HTML page from the same function. The flask application hosts the webpage on a dedicated server. There is a file structure to be followed while using Flask framework. Following are the code snippets of the Flask code and server link where webpage is hosted.

FYP_EnsembleModel.py | app.py

```python
from flask import Flask,request, url_for, redirect, render_template
import pickle
import numpy as np

app = Flask(__name__)

modelLR=pickle.load(open('modelLR.pkl','rb'))
modelKNN=pickle.load(open('modelKNN.pkl','rb'))
modelDT=pickle.load(open('modelDT.pkl','rb'))

@app.route('/')
def hello_world():
    return render_template("index.html")

@app.route('/')
def hello_world():
    return render_template("index.html")

@app.route('/predict',methods=['POST','GET'])
def predict():
    a=[]
    locality=request.form.get("local")
    carparea=int(request.form.get("area"))
    bed=int(request.form.get("noofbedr"))
    parking=int(request.form.get("park"))
    gym=int(request.form.get("gym"))
    pool=int(request.form.get("swimpool"))
    lift=int(request.form.get("lift"))
    resale=int(request.form.get("resale"))
    for i in range (len(dict)):
        if(dict[i]['Location']==locality):
            locid=dict[i]['Location ID']
            appa=dict[i]['Avg_Price_Area']
    final=[np.array([carparea,appa,locid,bed,resale,gym,lift,parking,pool])]
    prediction1=modelLR.predict(final)
    prediction2=modelKNN.predict(final)
    prediction3=modelDT.predict(final)
    output = prediction1*0.1 + prediction2*0.3 + prediction3*0.6
    predict="{:,}".format(int(output[0][0]))
    return render_template('index.html',pred=predict)

if __name__ == '__main__':
    app.run()
```

Fig 7.6 Flask app.py code snippet

```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [29/Apr/2021 02:12:47] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2021 02:12:47] "GET /static/style.css HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2021 02:12:47] "GET /static/1.png HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2021 02:13:37] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2021 02:13:53] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2021 02:14:03] "POST /predict HTTP/1.1" 200 -
```

Fig 7.7 Webpage hosted on Flask Server

7.2 RESULTS

The outcome of our Ensemble model with the website interface was a complete system which could determine house prices from a set of given input parameters. The methods were abstracted from the user, who could only view the predicted price after hitting the Calculate Price button. The Ensemble model gave a MAPE (Mean Absolute Percentage Error) of 16.09%. Which means the system and the model gave an accuracy of around 84%.

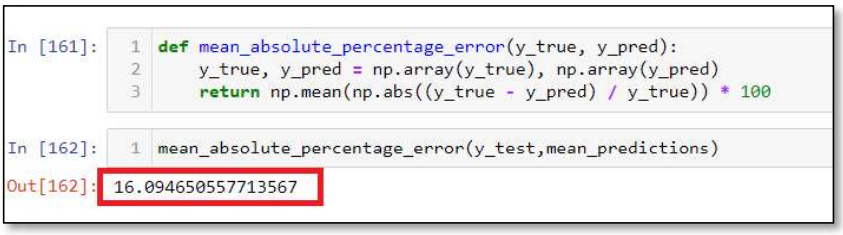


Fig 7.8 Mean Absolute Percentage Error

Following are the screenshots of UI of the final system that has been implemented. There are 4 different scenarios of combinations of Location, Area, No. of Bedrooms and other parameters shown:

7.2.1 (Location: Nerul; Area: 600; No. of Bedrooms: 2; [Parking, Lift, Resale]: checked)

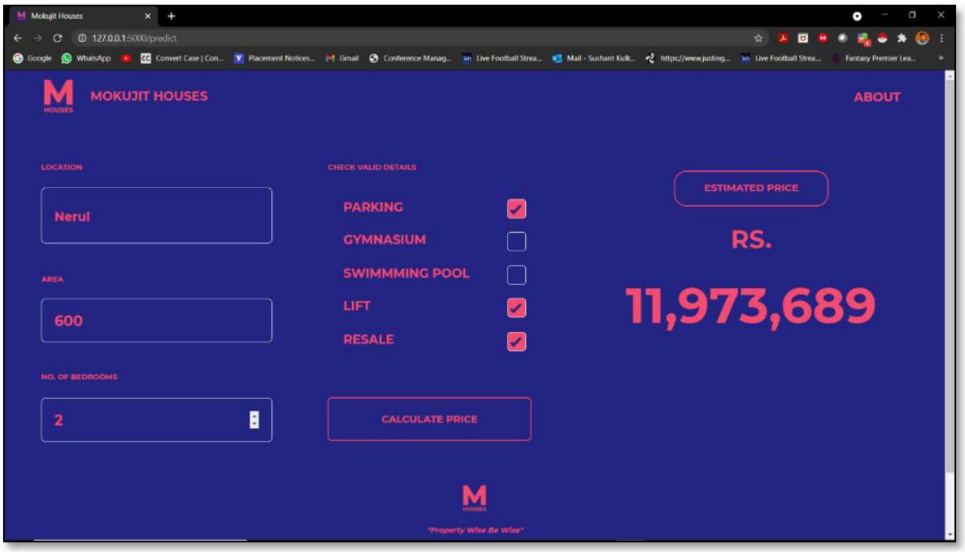


Fig 7.9 Result 1

7.2.2 (Location: Kharghar; Area: 600; No. of Bedrooms: 2; [Parking, Lift, Resale]: checked)

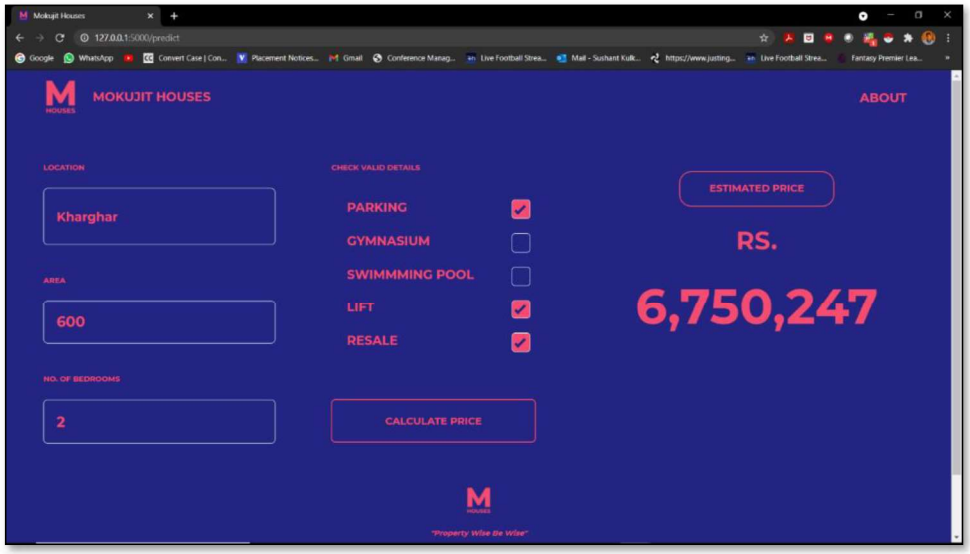


Fig 7.10 Result 2

7.2.3 (Location: Dadar West; Area: 500; No. of Bedrooms: 1; [Parking, Gymnasium, Lift]: checked)

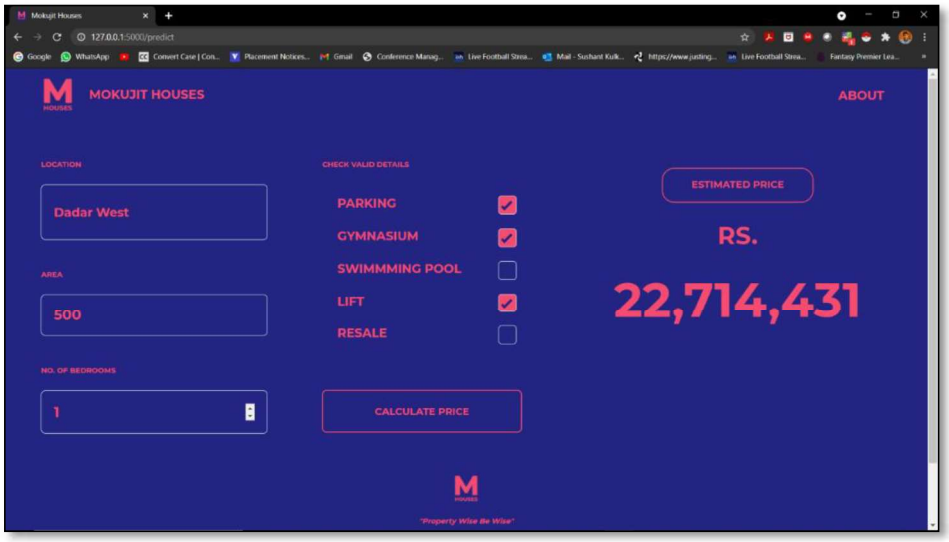


Fig 7.11 Result 3

7.2.4 (Location: Dadar West; Area: 500; No. of Bedrooms: 1; [Parking, Gymnasium, Lift]: checked)

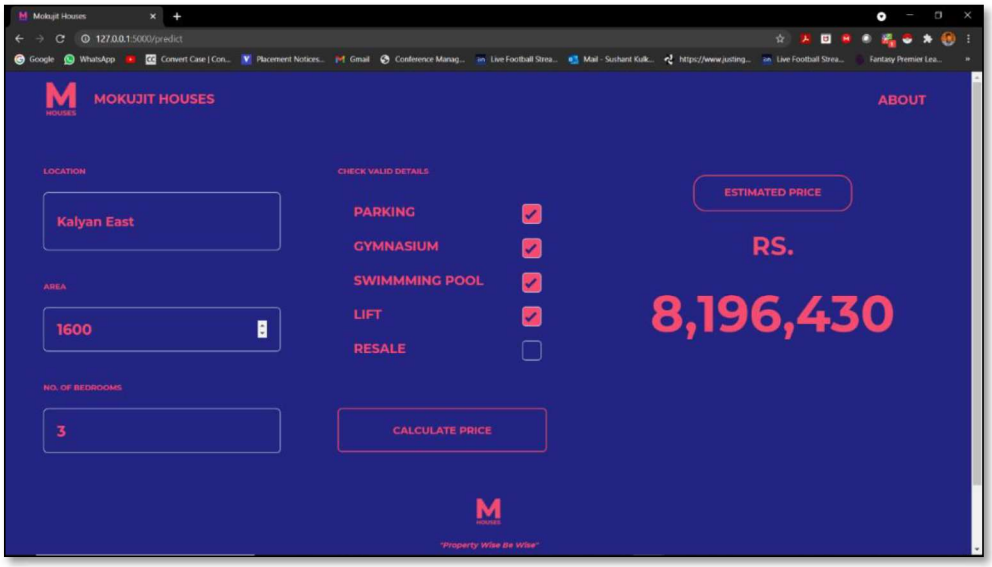


Fig 7.12 Result 4

Chapter 8

Conclusion & Future Scope

8.1 CONCLUSION

The price of housing units depends on a large number of factors. Therefore, price estimation strategy must consider multiple intrinsic and indicative parameters. The efficacy checking of the algorithms is generally done by testing it on certain data sets. Sample space of the data sets also affects the prediction performance. Work on many prediction methodologies is available in the literature. Most of them adapt the idea of comparing individual algorithms and selecting the best performing algorithms as the model for prediction.

We present a novel method of housing price prediction based on ensemble learning instead of using individual algorithms. The performance of the method was tested on 6347 records from the dataset incorporated from Kaggle Inc. It showed improved performance of 84% accuracy and MAPE of 16.09%. The innovation presented in this uses the weighted average ensemble model of Linear Regression, KNN, and Decision Tree.

8.2 FUTURE SCOPE

The performance can further be improved by optimizing the parameters. The key takeaway from this method would be the significant improvement in the house price predictions using the ensemble learning model. Different combinations to the build the ensemble learner could be tried to observe the variations to the results obtained and if available an improvement in the model could be done. If we could include the real-time development status and the on-going development plans to the parameters, the accuracy of the system can be further improved.

Chapter 9

References

9.1 RESEARCH PAPERS

[1] Neelam Shinde, Kiran Gawande, “Valuation of House Prices using Predictive Techniques”, International Journal of Advances in Electronics and Computer Science – 2018.

[2] Joao Mendes Moreira, Alipio Mario Jorge, Carlos Soares, Jorge Freire de Sousa, “Ensemble Approaches for Regression: A Survey”, ACM Computing Surveys – 2012.

[3] Yashraj Garud, Hemanshu Vispute, Nayan Bisai, and Prof. Madhu Nashipudimath, “Housing Price Prediction using Machine Learning”, International Research Journal of Engineering and Technology (IRJET) – 2020.

[4] Prof. Pradnya Patil, Darshil Shah, Harshad Rajput, Jay Chheda, “House Price Prediction Using Machine Learning and RPA”, International Research Journal of Engineering and Technology (IRJET) – 2020.K. Elissa, “Title of paper if known,” unpublished.

[5] Alisha Kuvalekar, Sidhika Mahadik, Shivani Manchewar, Shila Jawale, “House Price Forecasting using Machine Learning”, 3rd International Conference on Advances in Science & Technology (ICAST) – 2020.

[6] Zhongyuan Han, Jiaming Gao, Huilin Sun, Ruifeng Liu, Chengzhe Huang, Leilei Kong, Haoliang Qi, “An Ensemble Learning-based model for Classification of Insincere Question”, FIRE – 2019.

[7] Ayush Varma, Sagar Doshi, Abhijit Sarma, Rohini Nair, “House Price Prediction Using Machine Learning and Neural Networks “, Second International Conference on Inventive Communication and Computational Technologies (ICICCT) – 2018.

[8] P. Durganjali; M. Vani Pujitha, “House Resale Price Prediction Using Classification Algorithms”, International Conference on Smart Structures and Systems (ICSSS) – 2019.

[9] CH. Raga Madhuri; G. Anuradha; M. Vani Pujitha, “House Price Prediction Using Regression Techniques: A Comparative Study”, International Conference on Smart Structures and Systems (ICSSS) – 2019.

[10] A. Adair, J. Berry, W. McGreal, “Hedonic modeling, housing submarkets and residential valuation”, Journal of Property Research – 1996.

[11] O. Bin,” A prediction comparison of housing sales prices by parametric versus semi-parametric regressions”, Journal of Housing Economics – 2004.

[12] T. Kauko, P. Hooimeijer, J. Hakfoort, “Capturing housing market segmentation: An alternative approach based on neural network modeling”, Housing Studies – 2002.

[13] Li Li, Kai-Hsuan Chu, “Prediction of Real Estate Price Variation Based on Economic Parameters “, IEEE International Conference on Applied System Innovation – 2017.

[14] G. Naga Satish, Ch. V. Raghavendran, M.D.Sugnana Rao, Ch. Srinivasulu, “House Price Prediction Using Machine Learning “, International Journal of Innovative Technology and Exploring Engineering (IJITEE) – 2019.

[15] Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, Wayan Firdaus Mahmudy, “Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization”, (IJACSA) International Journal of Advanced Computer Science and Applications.

[16] Ayşe SOY TEMÜR, Melek AKGÜN2, Günay TEMÜR, “Predicting housing sales in Turkey using ARIMA, LSTM, and Hybrid models”, Journal of Business Economics and Management ISSN – 2019.

9.2 ONLINE REFERENCES

- https://www.analyticsvidhya.com/blog/2020/09/integrating-machine-learning-into-web-applications-with-flask/
- https://www.youtube.com/watch?v=Pc8WdnIdXZg
- https://www.toptal.com/machine-learning/ensemble-methods-machine-learning
- https://blog.statsbot.co/ensemble-learning-d1dcd548e936
- https://github.com/vsmolyakov/experiments_with_python/blob/master/chp01/ensemble_methods.ipynb

Papers
Published

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

House Price Prediction Using Ensemble Learning

[1]Sushant Kulkarni, [2]Shefin Shajit, [3]Akshay Mohite, [4]Dr.Swati Sinha

[1]Student, [2]Student, [3]Student, [4]Professor
Department of Information Technology,
Vidyalankar Institute of Technology, Mumbai, India

Abstract: This paper presents a system that works on a set of data containing house prices of places in Mumbai along with the major parameters affecting the price such as area, location, swimming pool, etc. obtained from open web source Kaggle Inc. and predict the resale price under the parameters. The model implemented incorporates ensemble learning i.e. a combination of machine learning algorithms instead of relying on a single algorithm for improved predictions. The ensemble model incorporated in our system (weighted average of Decision Tree, Linear Regression, and K-Nearest Neighbor) brings an added advantage over using solo algorithms in the process of obtaining minimum error in prediction. The trained model demonstrated a Mean Absolute Percentage Error (MAPE) of 16.09%.

Index Terms **- House Price Prediction, Linear Regression, Decision Tree, KNN, Ensemble Learning**

I. INTRODUCTION

The real estate sector is a major sector influencing India's economy. In India, about 15 percent of the total jobs are generated by the real estate sector. In 2021, the sector must adopt innovative ways of dealing with the requirements. While houses will continue to be sold, they will now be done with creative disruption. The reinvention will include technology playing a lead role in meeting altered norms being considered by home buyers. It is important to have the best tools at disposal which would guide the buyers about where to put their money into. Since property prices rarely decrease rapidly, it is a major contender for investment. The property prices depend on various intrinsic and extrinsic factors which directly or indirectly affect the long-term price values. A fair share of India's economic condition affects property prices in the long run. This scenario calls for technology to bring out the best ways to help out the customer's investment decisions. A smart property investment decision about where to put the money depends mainly on three factors conditions, concept, and location. House price prediction can help the customer make a calculated risk in investment. The price point can vary per the square foot area, location, availability of swimming pool, lifts, etc. This paper explains a system that incorporates a dataset containing certain parameters that affect property prices in Mumbai, India. Further details about the dataset and the parameters incorporated are explained in detail in the dataset section.

II. LITERATURE SURVEY

Over the years there have been numerous approaches in predicting house prices per all the intrinsic and extrinsic factors affecting the price without any fluctuations. Although finding the best possible prediction model depends on the data available. The price of a property can differ based on its location, area, amenities, etc., and finding the best predictive model to predict that price has been a concern for researchers over the past decade. In our literature survey, we found various such approaches to find the house price using various models and a combination of models.

One of the methods proposed in the paper by Neelam Shinde and Kiran Gawande [1] includes testing the dataset with four different regression algorithms namely Lasso Regression, Logistic Regression, Decision Tree, and Support Vector Regression. On comparing the error metrics such as R-Squared Value, Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error, Decision Tree turned out to be the best algorithm giving a higher accuracy level of 86.4% and low error values whereas Lasso Regression performed the worst giving an accuracy level of 60.32%.

The majority of our work around ensemble learning for our predictive model has been widely accepted as a success in improving predictions [2]. Where ensemble models can vary depending on the needs of the data and the manner of predictions where even a small amount of improvement can have a big impact. The integration of two or more ensemble members depends on the type of integration the developer would see fit for the data i.e. Constant Weighting Functions and Non- Constant Weighting Functions.

A different approach has been taken in the paper [10]. The paper focused on linking various researches related to housing market prices analysis. Substantial focus is provided to hedonic price modeling and its application on the house price market and the possible submarket existence.

Decision Tree is used to make predictions in paper [5] after giving the highest accuracy in terms of prediction values among other algorithms tested namely Linear Regression, Multiple Linear Regression, Decision Tree Regressor, and KNN. Apart from parameters like no. of bedrooms, carpet area, built-up area, age of the property, zip code, no. of bathrooms, latitude, and longitude of the property, they have also included two other features – air quality and crime rate to better the prediction.

Another proposed system used Lasso and Random Forest regression techniques and picked the best model for the data depending on error values [3]. The data was passed onto 6 stages including data pre-processing, test-train 50:50 split, training the data with Lasso and Random Forest models and testing with the test data, and picking the best model.

In the paper by Prof. Pradnya Patil, Darshil Shah, Harshad Rajput, and Jay Chheda [4], the proposed system incorporates the UiPath Studio Platform to develop the RPA Flowchart. The UiPath Studio provides data scraping capabilities with the assistance of scraping wizards. A bunch of machine learning algorithms are compared and implemented on the dataset. A comparison between boosting algorithms is done namely XGBoost, Light BGM, and CatBoost. Random Forest was found to do well with small amounts of data and doesn't improve accuracy with more samples. CatBoost was termed the clear winner in comparisons. RPA provided a major improvement in efficiency in terms of fast extraction and less prone to errors.

A further step has been taken in the paper by P. Durganjali and M. Vani Pujitha [8]. It analyses different classification algorithms such as Decision Tree, Logistic Regression, Random Forest, AdaBoost, Naïve Bayes with an accuracy of 92%, 81.5%, 86.5%, 96%, and 88% respectively. AdaBoost and Decision Tree using C 5.0 were selected to predict values of the house and using rules, they predicted profit or loss.

Detailed study of different machine learning algorithms namely Multiple Linear Regression, Elastic Net Regression, Ridge Regression, Ada Boosting Regression, LASSO Regression, and Gradient Boosting has been done on a public output dataset of a specified region in the USA [9]. The attained scores of the algorithms were 0.73, 0.66, 0.73, 0.78, 0.73 and 0.97 respectively. Gradient Boosting turned out to be the best algorithm as it gave low error values.

III. DATASET

The dataset incorporated in the system is taken from a public dataset source Kaggle Inc. It has data for house prices and features from 413 unique locations of Mumbai, India. It consists of 6347 records with 17 parameters that have the possibility of affecting the property prices. However, out of these 17 parameters, only 7 were chosen (Area, No. of Bedrooms, New/Resale, Gymnasium, Lift Available, Car Parking, Swimming Pool) along with 2 added parameters (Location Id and Price Area) which are bound to have a major effect on housing prices. The area is the total built-up area in square feet. New/Resale specifies if the property is a resale property or a new property. Gymnasium, Lift, Car Parking, and Swimming Pool mention if the property happens to provide these amenities (Binary value i.e. 1s and 0s). Location id is a unique id to all the locations present in the dataset in ascending order of Price Area. Price Area is the average price per area of a location.

3.1 Data Pre-processing

Data Pre-processing is a major step in transforming the dataset into an efficient format. This includes removing the NaN values (Missing Values) and improper noisy data in the dataset to narrow down the training of the model to the appropriate level. Records containing NaN values were deleted in our dataset to make it fit for training the ensemble learning model.

3.2 Data Analysis

Every single parameter in the dataset is analyzed with every other parameter to check the dependence and correlation using an SNS heatmap. The correlation is measured in the range of -1 to +1 where a higher absolute score shows better correlation and the lower absolute score worse the relation. Below Fig.1 depicts the correlations among the 17 parameters which affect the house prices. We removed parameters that gave bad scores and kept in a total of 7 parameters that majorly affected house prices namely, Area, No. of Bedrooms, New/Resale, Gymnasium, Lift Available, Car Parking, Swimming Pool, Location Id, and Price Area.

We removed outliers from the dataset using Interquartile Rule. On multiplying 1.5 (a constant to discern outliers) to the Interquartile Range (IQR) we set a limit to values outside Q1 and Q3 (Quartile 1 and Quartile 3 respectively). Practically any value that exists outside the fence i.e., values greater than 1.5 * (IQR) below Q1 or greater than 1.5 * (IQR) above Q3 are termed as outliers. Removing outliers helps to make the predictions statistically significant.

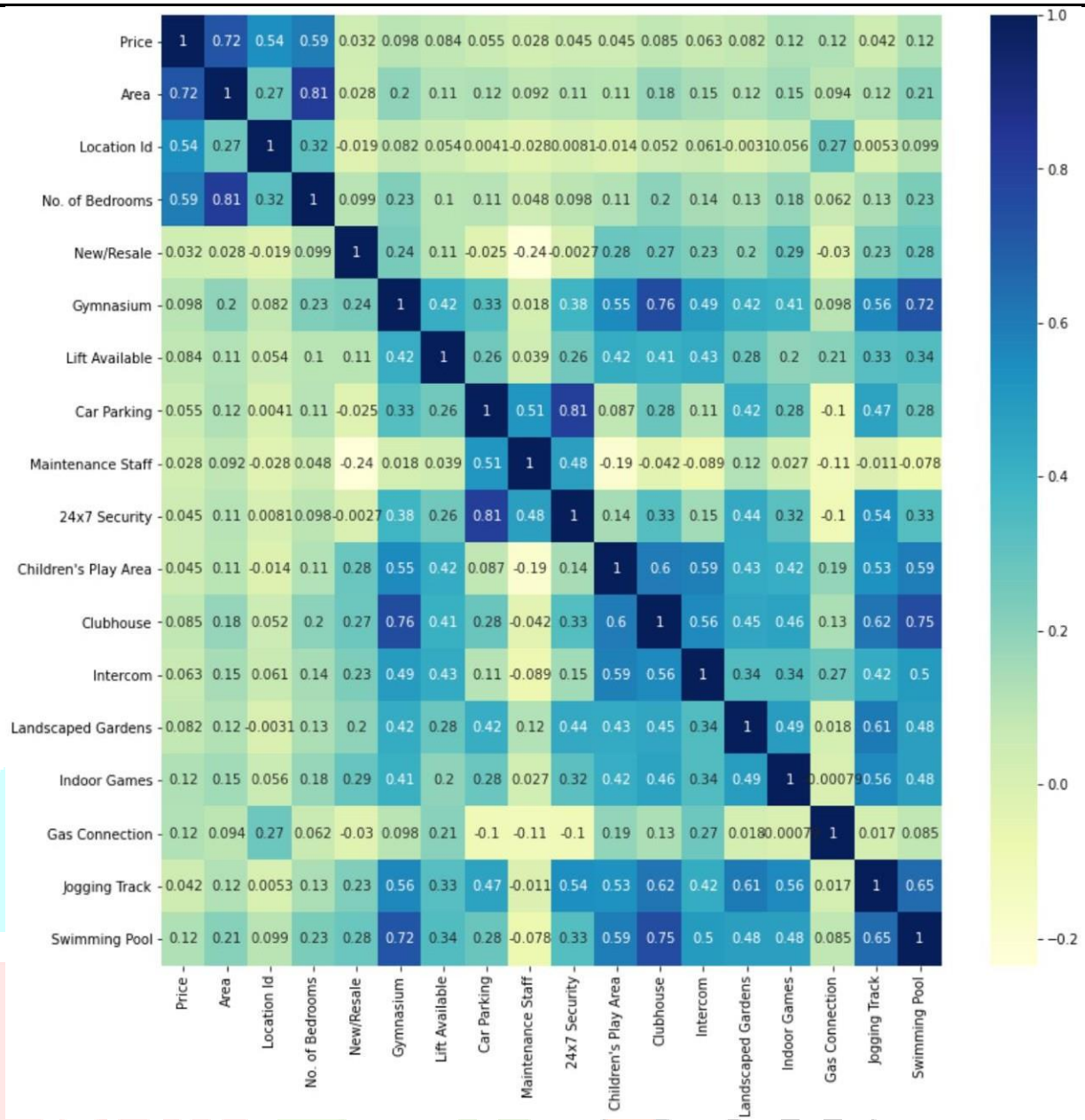


Fig.1. Correlation Matrix Heatmap

IV. METHODOLOGY

After analyzing and visualizing the data the next step is to train the model to help us in predicting the house prices. It involves dividing the dataset into training and testing sets. In the process of developing the model for training the data, we found from our experimental results that a combination of algorithms or models worked better than a single algorithm on the data i.e., gave lower error values. Various regression algorithms were tested including Linear Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), K- Nearest Neighbors (KNN), Bagging, and Boosting where Bagging, Boosting, and Random Forest were ensemble models in themselves. Ensemble Learning includes a combination of algorithms or models to provide results.

The ways of combining models depend exclusively on the features of the dataset. Simple averaging of results of ensemble members (models used in the combination) implies equal contribution to the final prediction. Weighted averaging is an extension to simple averaging which deals with the limitation of simple averaging when some models are known to perform better or much worse than the others. A weighted ensemble is a model where the weight contribution of each model participating (ensemble members) is computed according to the efficiency of predictions. Below given Fig.2 provides an insight into the flow of the system.

We did a 1:3 split on the dataset to make the test and train sets. After training and building the ensemble model with the training set of data, it was tested with the test set of data to make the final predictions. The model was then implemented with a web UI to input parameter values from the user and predict house prices based on those values. After working on various models as a combination for ensemble learning, we found the best fit of three algorithms namely Linear Regression, K-Nearest Neighbor (KNN), and Decision Tree. A weighted average of predictions from these three algorithms gave us the least error range compared to other combinations.

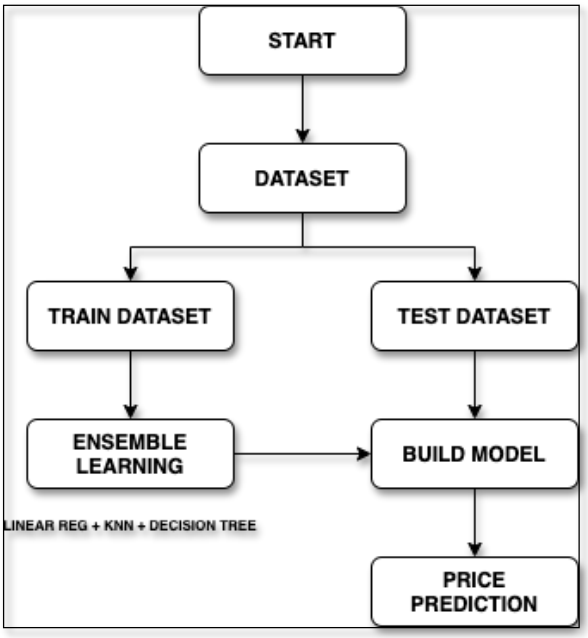


Fig.2. Methodology Flow Chart

4.1 Linear Regression

Simple linear regression algorithm analyses the relation between two entities where one is dependent and the other is independent. Change in the independent entity reflects the change in the dependent entity. This algorithm does not calculate the dependency instead only the association between the two entities or variables. The equation of the line of linear regression is as follows:

$$y = A + Bx \quad (4.1)$$

Where X is the independent variable and Y is the dependent variable. A refers to the intercept whereas B refers to the slope of the line. Here, we have trained the linear regression model using the training dataset and then tested it on the testing dataset to make predictions. Below given Fig.3 is a scatter plot between the original house price value and the predicted house price value by the model.

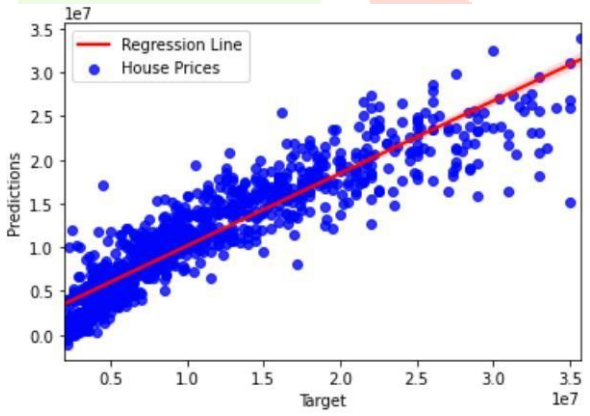


Fig.3. Linear Regression Scatter Plot

Below given Table.1 mention the error metrics of the model: Mean Squared Error (MSE), Mean Absolute Error (MAE), R2 Score, and Root Mean Squared Error (RMSE).

Table 1. Linear Regression Error Metrics

MAE	MSE	RMSE	R2
2229371.20	9247442528422.93	3040960.80	0.83

4.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a machine learning algorithm that does regressive as well as classification predictive analysis. It is also called a lazy learner algorithm since it does not analyze the data it is trained with instead the algorithm only classifies the new data it is tested with by its similarity. Here, KNN uses feature similarity to predict the house price values, i.e., it assigns a value to the new data based on how closely it relates (using distance functions for continuous variables such as Euclidean (2) and Manhattan (3) distance functions) to the points in the training set.

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{4.2}$$

$$\sum_{i=1}^{k}|x_i - y_i| \tag{4.3}$$

Where X refers to the new point, Y refers to the existing point and K is the K-Factor (no. of neighbors the algorithm looks at before assigning a value). Below given Fig.4 is a scatter plot between the original house price value and the predicted house price value by the model.

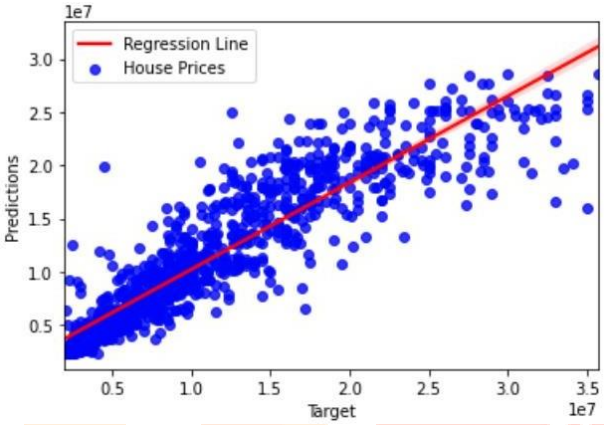


Fig.4. KNN Scatter Plot

Below given Table.2 mentions the error metrics of the model: Mean Squared Error (MSE), Mean Absolute Error (MAE), R2 Score, and Root Mean Squared Error (RMSE).

Table 2. KNN Error Metrics

MAE	MSE	RMSE	R2
1948884.96	8927720068103	2987929.06	0.84

4.3 Decision Tree

A decision tree algorithm builds the model in a tree structure with decision nodes and leaf nodes. It breaks down the dataset into smaller sets with similar values and the highest node is known as the root node. The tree is made of only conditional control statements with each decision node testing an attribute. Below given Fig.5 is a scatter plot between the original house price value and the predicted house price value by the model.

Below given Table.3 mentions the error metrics of the model: Mean Squared Error (MSE), Mean Absolute Error (MAE), R2 Score, and Root Mean Squared Error (RMSE).

Table 3. Decision Tree Error Metrics

MAE	MSE	RMSE	R2
1851282.60	10135014746386	3183553.80	0.81

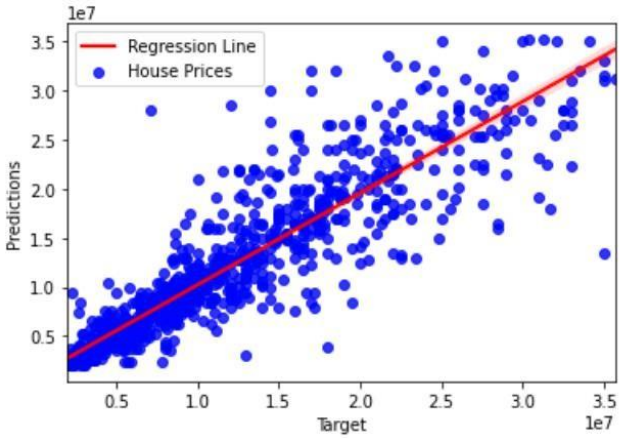


Fig 5. KNN Scatter Plot

4.4 Ensemble Learning Model

As discussed earlier, the ensemble learning model is a combination of algorithms or models which help to improve predictions depending on the features of the dataset. We could conclude from our experimental results that a weighted average of predictions from Linear Regression, KNN, and Decision Tree models provided the lowest error values compared to predictions from individual algorithms from Fig.6 and Table.4 below. The weights assigned to the predictions of models are based on its performance and features of our dataset exclusively.

Below given Table.4 mentions the error metrics of the model: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R2 Score, and Weight Assigned (W).

Table 4. Error Metrics Comparison

MODEL	MAE	MSE	RMSE	R2	W
Linear Reg.	2229371.20	9.2E+10	3040960.80	0.83	0.1
KNN	1948884.96	8.9E+!0	2987929.06	0.84	0.3
Decision Tree	1851282.60	1.0E+09	3183553.80	0.81	0.6
Ensemble	1658701.38	7.3E+10	2694486.53	0.87	-

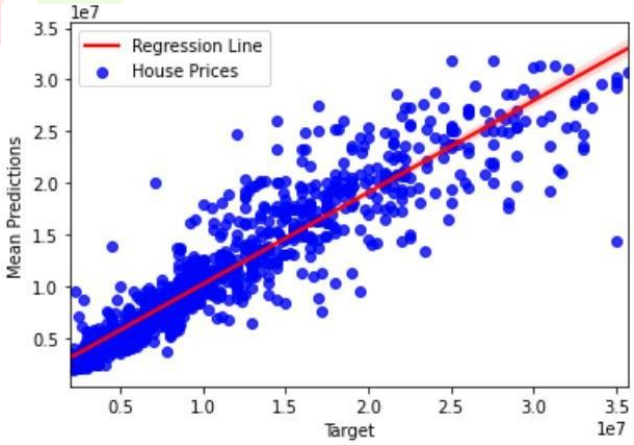


Fig 6. Ensemble Model Scatter Plot

V. EXPERIMENTAL RESULTS

The weighted average ensemble model performs substantially better than the individual models. The trained model attains an accuracy of 84%. The comparison of Mean Absolute Percentage Errors (MAPE) of the models used and the ensemble model is given below in the form of a bar chart Fig. 7. On comparing the various models, we find that the ensemble model works best with the lowest Mean Absolute Percentage Error (MAPE).

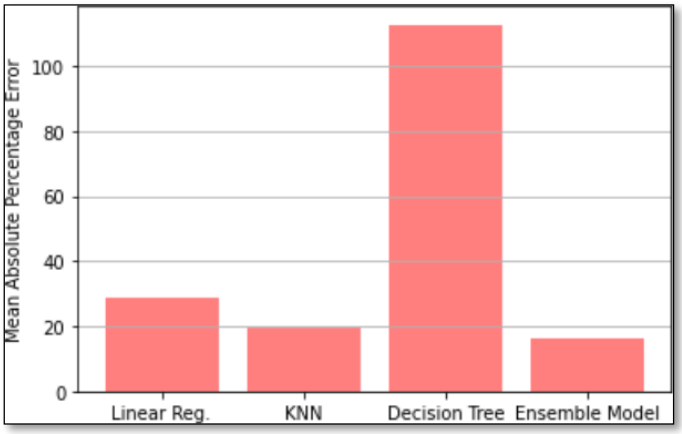


Fig 7. MAPE Comparison

VI. CONCLUSION

The price of housing units depends on a large number of factors. Therefore, price estimation strategy must consider multiple intrinsic and indicative parameters. The efficacy checking of the algorithms is generally done by testing it on certain data sets. Sample space of the data sets also affects the prediction performance. Work on many prediction methodologies is available in the literature. Most of them adapt the idea of comparing individual algorithms and selecting the best performing algorithms as the model for prediction. This paper presents a novel method of housing price prediction based on ensemble learning instead of using individual algorithms. The performance of the method was tested on 6347 records from the dataset incorporated from Kaggle Inc. It showed improved performance of 84% accuracy and MAPE of 16.09%. The innovation presented in this uses the weighted average ensemble model of Linear Regression, KNN, and Decision Tree. The performance can further be improved by optimizing the parameters. The key takeaway from this paper would be the significant improvement in the house price predictions using the ensemble learning model.

VII. ACKNOWLEDGMENT

A sincere thanks to Vidyalankar Institute of Technology for providing a platform and necessary resources to develop this project. Our deepest gratitude to each and every one who contributed substantially towards the completion of the project work.

VIII. REFERENCES

[1] Neelam Shinde, Kiran Gawande, “Valuation of House Prices using Predictive Techniques”, International Journal of Advances in Electronics and Computer Science – 2018.

[2] Joao Mendes Moreira, Alipio Mario Jorge, Carlos Soares, Jorge Freire de Sousa, “Ensemble Approaches for Regression: A Survey”, ACM Computing Surveys – 2012.

[3] Yashraj Garud, Hemanshu Vispute, Nayan Bisai, and Prof. Madhu Nashipudimath, “Housing Price Prediction using Machine Learning”, International Research Journal of Engineering and Technology (IRJET) – 2020.

[4] Prof. Pradnya Patil, Darshil Shah, Harshad Rajput, Jay Chheda, “House Price Prediction Using Machine Learning and RPA”, International Research Journal of Engineering and Technology (IRJET) – 2020.K. Elissa, “Title of paper if known,” unpublished.

[5] Alisha Kuvalekar, Sidhika Mahadik, Shivani Manchewar, Shila Jawale, “House Price Forecasting using Machine Learning”, 3rd International Conference on Advances in Science & Technology (ICAST) – 2020.

[6] Zhongyuan Han, Jiaming Gao, Huilin Sun, Ruifeng Liu, Chengzhe Huang, Leilei Kong, Haoliang Qi, “An Ensemble Learning-based model for Classification of Insincere Question”, FIRE – 2019.

[7] Ayush Varma, Sagar Doshi, Abhijit Sarma, Rohini Nair, “House Price Prediction Using Machine Learning and Neural Networks “, Second International Conference on Inventive Communication and Computational Technologies (ICICCT) – 2018.

[8] P. Durganjali; M. Vani Pujitha, “House Resale Price Prediction Using Classification Algorithms”, International Conference on Smart Structures and Systems (ICSSS) – 2019.

[9] CH. Raga Madhuri; G. Anuradha; M. Vani Pujitha, “House Price Prediction Using Regression Techniques: A Comparative Study”, International Conference on Smart Structures and Systems (ICSSS) – 2019.

[10] A. Adair, J. Berry, W. McGreal, “Hedonic modeling, housing submarkets and residential valuation”, Journal of Property Research – 1996.

[11] O. Bin,” A prediction comparison of housing sales prices by parametric versus semi-parametric regressions”, Journal of Housing Economics – 2004.

[12] T. Kauko, P. Hooimeijer, J. Hakfoort, "Capturing housing market segmentation: An alternative approach based on neural network modeling", Housing Studies – 2002.

[13] Li Li, Kai-Hsuan Chu, "Prediction of Real Estate Price Variation Based on Economic Parameters ", IEEE International Conference on Applied System Innovation – 2017.

[14] G. Naga Satish, Ch. V. Raghavendran, M.D.Sugnana Rao, Ch. Srinivasulu, "House Price Prediction Using Machine Learning ", International Journal of Innovative Technology and Exploring Engineering (IJITEE) – 2019.

[15] Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, Wayan Firdaus Mahmudy, "Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization", (IJACSA) International Journal of Advanced Computer Science and Applications.

[16] Ayşe SOY TEMÜR, Melek AKGÜN2, Günay TEMÜR, "Predicting housing sales in Turkey using ARIMA, LSTM, and Hybrid models", Journal of Business Economics and Management ISSN – 2019.

Certificates

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Sushant Kulkarni

In recognition of the publication of the paper entitled

HOUSEPRICEPREDICTIONUSINGENSEMBLELEARNING

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 9 Issue 5 , Date of Publication:May 2021 2021-05-11 11:26:19

PAPER ID : IJCRT2105379

Registration ID : 207073

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Shefin Shajit

In recognition of the publication of the paper entitled

HOUSEPRICEPREDICTIONUSINGENSEMBLELEARNING

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 9 Issue 5 , Date of Publication:May 2021 2021-05-11 11:26:19

PAPER ID : IJCRT2105379

Registration ID : 207073

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Akshay Mohite

In recognition of the publication of the paper entitled

HOUSEPRICEPREDICTIONUSINGENSEMBLELEARNING

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 9 Issue 5 , Date of Publication:May 2021 2021-05-11 11:26:19

PAPER ID : IJCRT2105379

Registration ID : 207073

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Dr. Swati Sinha

In recognition of the publication of the paper entitled

HOUSEPRICEPREDICTIONUSINGENSEMBLELEARNING

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 9 Issue 5 , Date of Publication:May 2021 2021-05-11 11:26:19

PAPER ID : IJCRT2105379

Registration ID : 207073

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013