

RESIDENTIAL PROPERTY PRICE PREDICTION USING SML TECHNIQUES

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

by

Neeraja Sanjay (38110361)

Sindhura V (38110533)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAA

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600 119

APRIL - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Neeraja Sanjay (38110361), Sindhura V (38110533)** who carried out the project entitled as "**Residential Property Price Prediction using SML Techniques**" my supervision from November 2022 to April 2022.

Internal Guide

Mr. VIGNESH. R, M.E., (Ph.D.,)

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

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

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

Neeraja Sanjay (38110361) and Sindhura. V (38110533) hereby declare that the Project Report entitled “**Residential Property Price Prediction using SML Techniques**” done by us under the guidance of **Mr. Vignesh. R, M.E., (Ph.D.,)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

SIGNATURE OF THE CANDIDATE

PLACE:

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. Vignesh. R, M.E., (Ph.D.)** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

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

ABSTRACT

Investing in real estate is a huge asset, which brings back huge returns in the future. Hence planning the finances well ahead for this investment is crucial and beneficial. This paper will provide an overview of a model which is built to predict property prices. The model takes into account some features for the property like the availability of car parking spaces, type of drainage present, name of the area etc and uses supervised machine learning techniques to predict the price of the property. A couple of machine learning algorithms are compared to find that algorithm which shows the least error and highest accuracy. Calculations like Mean square error, Mean absolute error, R2 error etc is done on all the algorithms in order to find the most efficient one. This report examines in detail the construction and working of this model.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	V
	LIST OF FIGURES	VIII
	LIST OF ABBREVIATIONS	IX
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
2	LITERATURE SURVEY	2
3	METHODOLOGY AND IMPLEMENTATION	4
	3.1 PROPOSED WORK	4
	3.2 DATA SCIENCE	4
	3.3 MACHINE LEARNING	5
	3.4 HTML AND CSS	5
	3.5 DATASET	6
	3.6 PROGRAMMING LANGUAGES AND PACKAGES	6
	3.6.1 SKLEARN	7
	3.6.2 NUMPY	7
	3.6.3 PANDAS	7
	3.6.4 MATPLOTLIB	7
	3.6.5 SEABORN	7
	3.7 DATA PREPROCESSING	7
	3.8 EXPLORATORY DATA ANALYSIS /VISUALIZATION	8
	3.9 COMPARING ALGORITHMS	10
	3.9.1 LOGISCTIC REGRESSION	10
	3.9.2 LINEAR REGRESSION	11
	3.9.3 RIDGE REGRESSION	11
	3.9.4 RANDOM FOREST REGRESSION	11

	3.9.5 SUPPORT VECTOR REGRESSION	12
	3.9.6 LASSO REGRESSION	12
	3.9.7 KNN CLASSIFICATION	12
	3.9.8 STOCHASTIC GRADIENT DESCENT REGRESSION	12
	3.9.9 DECISION TREE	13
4	RESULTS AND DISCUSSION	14
	4.1 RESULT	14
	4.2 PERFORMANCE ANALYSIS	14
	4.2.1 ANALYSIS OF LINEAR REGRESSION	15
	4.2.2 ANALYSIS OF LOGISCTIC REGRESSION	15
	4.2.3 ANALYSIS OF STOCHASTIC GRADIENT DESCENT	15
	4.2.4 ANALYSIS OF RIDGE REGRESSION	15
	4.2.5 ANALYSIS OF KNN CLASSIFICATION	16
	4.2.6 ANALYSIS OF SUPPORT VECTOR REGRESSION	16
	4.2.7 ANALYSIS OF LASSO REGRESSION	16
	4.2.8 ANALYSIS OF DECISION TREE	16
	4.2.9 ANALYSIS OF RANDOM FOREST REGRESSION	17
5	SUMMARY AND CONCLUSIONS	18
	5.1 FUTURE SCOPE	20
	REFERENCES	21
	APPENDIX	22
	A. SOURCE CODE	22
	B. SCREENSHOTS	43

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
FIG 3.1	SYSTEM ARCHITECTURE	5
FIG 3.2	KAGGLE DATA SET	6
FIG 3.3	UNPROCESSED DATA	8
FIG 3.4	PROCESSED DATA	8
FIG 3.5	HEATMAP DEPICTING THE CORRELATION OF ALL THE ATTRIBUTES	9
FIG 3.6	HISTOGRAM DEPICTING THE FREQUENCY OF THE VARIOUS SQUARE FEET AREAS IN THE DATA SET	9
FIG 3.7	PIE CHART DEPICTING THE AVAILABILITY OF PARKING FACILITY	10
FIG 5.1	REAL ESTATE PRICE PREDICTION - ANNA NAGAR	19
FIG 5.2	REAL ESTATE PRICE PREDICTION – T.NAGAR	20

LIST OF ABBREVIATIONS

ML	Machine Learning
AI	Artificial Engineering
SGD	Stochastic Gradient Descent
SVR	Support Vector Regression
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets

CHAPTER – 1

INTRODUCTION

Real Estate is a huge economic indicator of business. Investing in real estate has innumerable benefits and gives huge returns in the future. Moreover, different investors/buyers have different specifications about the property that they want to purchase. Real estate is a huge asset to the country's economy. Its value is different in every area and depends on square footage, population and popularity. Predicting residential property prices are expected to help people who plan to buy a house so that they can know the price range in the future, which helps them plan their finances well. In the proposed model, we will be using the concept of supervised machine learning techniques and Data science to accurately predict the residential house prices.

1.1 OBJECTIVE

Predicting residential property prices are expected to help people who plan to buy a house so that they can know the price range in the future, in order to plan their finances. In addition, house price predictions are also beneficial for property investors to know the trend of housing prices in a certain location. The goal is to develop a machine learning model for residential property price prediction, to potentially replace the updatable supervised machine learning regression models by predicting results in the form of best accuracy by comparing supervised algorithms.

CHAPTER 2

LITERATURE SURVEY

P.Durganjali, M. Vani Pujitha [4]. In their paper, it picks up the most recent prediction analysis to assist economic predictors to use it. It provides a summary of the prediction markets and, together, the present markets that build it easier to predict the market but Accuracy is low since classification can be implemented only for certain expected set of outputs.

Mahmood Husain Abdulaziz, Ahmed M. Zeki [10]. Their objective of this research paper is to use data mining to help the real estate advisors and consumers to estimate the prices of real estate, and to cope with any changes in the land prices according to the market trends. They've minimized error by dividing the dataset into several sub datasets based on the location. This model needs to have a real time data, to cope with the market change and demand for a better prediction analysis.

Mansi Jain, Himani Rajput, Neha Garg, Pronika Chawla [7]. They've provided an overview about how to predict house costs utilizing different regression methods with the assistance of python libraries. This paper provides a brief explanation about various graphical and numerical techniques which will be required to predict the price of a house, but it lacks Basic facilities like nearby shops are not considered, rendering the model incompetent.

Danh Phan [2]. This paper has machine learning techniques that are applied to analyze historical property transactions in Australia to discover useful models for house buyers and sellers. This paper explores useful models for real estate price prediction. It also provides insights into the Melbourne Housing Market.

Manasa J, Radha Gupta, Narahari N S [8]. In this research work, an analytical study has been carried out by considering the data set that remains open to the public by illustrating the available housing properties in machine hackathon platform. Modelling explorations apply some regression techniques such as multiple linear regression (Least Squares), Lasso and Ridge regression models,

boosting algorithms such as Extreme Gradient Boost Regression. The dataset contains less and limited features for an urban city like Bangalore.

Maida Ahtesham, NarmeenZakariaBawany, Kiran Fatima [9]. In their paper, Gradient Boosting Model XGBoost is utilized to predict housing prices. Dataset containing 38,960 records of Karachi, Pakistan is obtained from an Open Real Estate Portal. The paper considers many insights and has lot of features.

Nehal N Ghosalkar, Sudhir N Dhage [3]. Their objective of the paper is prediction of residential prices for the customers considering their financial plans and needs. By breaking down, past market patterns and value ranges, and coming advancements future costs will be anticipated. The prediction error is very less that is 0.3713, considered as a very minimum error while considering a huge city dataset like Mumbai. This paper mainly focuses on predicting the prices based on the financial plans of the customer without considering the market prices and cost increment which is a major con as the cost changes frequently.

Yiyang Luo [8]. This paper contains a simple regression model which contains a few characteristics of a residential asset, and we are able to reach a fairly good result. Some machine learning algorithms such as random forest and support vector machine are also implemented to predict asset pricing. The project has advanced regression models to Basic regression model which both Behave in a similar manner. The dataset has very limited features for a city like Bangalore.

CHAPTER 3

METHODOLOGY AND IMPLEMENTATION

3.1 PROPOSED WORK

Real Estate is a huge economic indicator of business. Investing in real estate has innumerable benefits and gives huge returns in the future. Due to the various factors like, varying price based on the timeline and other influencing factors, building such a model is quite challenging. To address these difficulties, in this model, we study an effective and fine-grained model for metropolitan sub region housing price predictions. Compared to existing works, our proposal is much more efficient as it takes into consideration a lot of unique features that other models do not. We utilize a feature selection mechanism to select more relevant features like Parking space, type of drainage, year of the building, condition and much more. A fine-grained forecasting model is used for sub region spatiotemporal housing price prediction. We use Random forest regression algorithm to develop our model.

3.2. DATA SCIENCE:

Data science is a combined field that utilizes logical strategies and scientific methods, processes, algorithms to draw insights from structured and unstructured data, and apply knowledgeable insights from data across a broad range of application domains. This field helps to understand the data and convert the raw data into more structured manner. Data scientists track down different examples and connections among these data and give better comprehension through different visualized diagrams and graphs.

3.3 MACHINE LEARNING

(ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data.

3.3.1 SUPERVISED MACHINE LEARNING:

Supervised Machine Learning is the process of training the dataset and based on those trained data set, the machine predicts the output. Usually we separate the data into 70-80% for the training and 20-30% for testing the data to check if the machine gives the correct result. In supervised machine learning, we provide the input data as well as the correct output data to the learning model; the aim is to map the input variable to the output variable. Supervised machine learning techniques are used in the real world for various purposes.

3.4 HTML AND CSS

HyperText Markup Language abbreviated as HTML, used for creating interactive web pages and applications. Webpage is written using html which is a backbone for a webpage. There are lot of tags used in html to highlight every single content of a webpage more uniquely.

Cascading Style sheet shortly known as CSS is used for formatting the look of the webpage into providing more additional features and highlights it, starting from providing various fonts to different styles. CSS makes the webpage look more attractive and interactive. There are two ways we can insert CSS with HTML, Internal CSS which is writing the code inside a HTML page. In External CSS, the coding part is done in a separate file and attached to the HTML page externally.

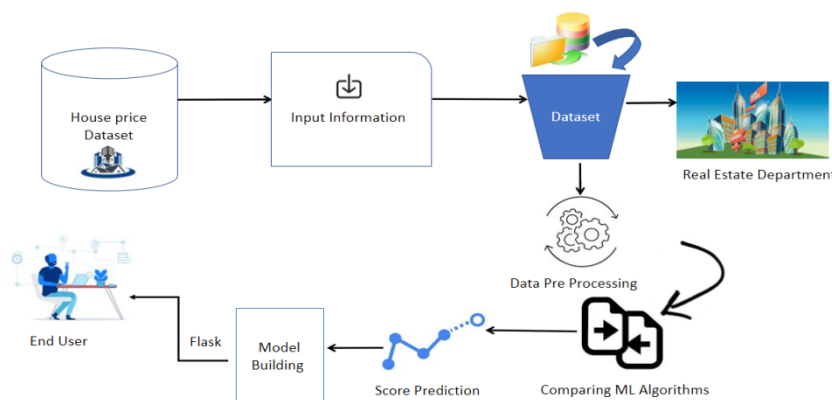


Fig 3.1 System Architecture

3.5 DATA SET

Multiple datasets from different sources would be combined to form a generalized dataset, and then different machine learning algorithms would be applied to extract patterns and to obtain results. The data set collected is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The data set for this project has been taken from Kaggle, which is an extraordinary spot for sharing, downloading and exploring huge amount of various datasets for different purposes. The Dataset “Chennai House Price Prediction” is publicly available with 7110 records and 22 attributes.

HomeInsertPage LayoutFormulasDataReviewViewDeveloper																															
CutCopyFormat PainterClipboard		Calibri11Font					Wrap TextAlignment					GeneralNumber					Conditional Formatting as Table Cell Styles					InsertDeleteFormatCells					AutoSumFillClearSort & FilterFind & SelectEditing				
A1PRT_ID																															
PRT_ID	AREA	INT_SQFT	DATE_SALE	DIST	MAIT	N_BEDROOM	N_BATHROOM	N_ROOM	SALE_CON	PARK_FAC	DATE_BUILT	BUILDTYPE	UTILITY	A1	STREET	MZZONE	QS_ROOM	QS_BATH	QS_BEDROOM	QS_OVER	REG_FEE	COMMIS	SALES_PRICE								
1	P03210	Karapakk	1004	131	1	3	AbNormal	Yes	#####	Commercial	AllPub	Paved	A	4.33	380000	144400	7600000														
2	P09411	Anna Nag	1986	26	2	1	5	AbNormal	No	#####	Commercial	AllPub	Gravel	RH	4.9	4.2	2.5	3.765	760122	304049	2.2E+07										
3	P01812	Adyar	909	70	1	1	3	AbNormal	Yes	#####	Commercial	ELO	Gravel	RL	4.1	3.8	2.2	3.09	421094	92114	1.3E+07										
4	P05346	Velachery	1855	14	3	2	5	Family	No	#####	Others	NoSewr	Paved	I	4.7	3.9	3.6	4.01	356321	77042	9630290										
5	P06210	Karapakk	1226	84	1	1	3	AbNormal	Yes	#####	Others	AllPub	Gravel	C	3	2.5	4.1	3.29	237000	74063	7406250										
6	P00219	Chrompet	1220	36	2	1	4	Partial	No	#####	Commercial	NoSewr	No Access	RH	4.5	2.6	3.1	3.32	409027	198316	1.2E+07										
7	P09105	Chrompet	1167	137	1	1	3	Partial	No	#####	Other	AllPub	No Access	RL	3.6	2.1	2.5	2.67	263152	33955	8488790										
8	P09679	Velachery	1847	176	3	2	5	Family	No	#####	Commercial	AllPub	Gravel	RM	2.4	4.5	2.1	3.26	604809	235204	1.7E+07										
9	P03377	Chrompet	771	175	1	1	2	AdjLand	No	#####	Others	NoSewr	Paved	RM	2.9	3.7	4	3.55	257578	33236	8308970										
10	P09623	Velachery	1635	74	2	1	4	AbNormal	No	#####	Others	ELO	No Access	I	3.1	3.1	3.3	3.16	323346	121255	8083650										
11	P09540	Chrompet	1203	78	2	1	4	AdjLand	Yes	#####	Commercial	AllPub	No Access	RM	4	3.2	4.5	3.83	403326	119504	1.5E+07										
12	P07121	Chrompet	1054	143	1	1	3	Partial	No	#####	Others	NoSewr	Gravel	RM	2.2	3.1	3.3	2.89	245693	141746	9449730										
13	P05512	Chrompet	1196	137	1	1	3	Family	No	#####	Others	NoSewr	No Access	RL	2.1	2.5	2.1	2.26	289554	35097	8774370										
14	P09370	Adyar	1056	83	1	1	3	Partial	Yes	#####	Others	NoSewr	Paved	RH	2.2	3.4	3.7	3.19	319141	37546	9386490										
15	P04085	Velachery	1865	157	3	2	5	Family	No	#####	Commercial	NoSewr	Paved	RM	4.8	2.2	4.9	3.66	619987	294494	1.5E+07										
16	P06328	Velachery	1868	148	3	2	5	Family	No	#####	Commercial	NoSewr	Gravel	RH	3.8	3.8	3.1	597135	188569	1.6E+07											
17	P06039	Karapakk	1639	175	2	2	4	Normal	Se	Yes	#####	House	ELO	Gravel	I	2.3	3.2	4.8	3.57	285648	77904	8656000									
18	P02016	Chrompet	796	134	1	1	2	AdjLand	Yes	#####	Commercial	AllPub	No Access	RL	3.5	4.6	4.7	4.3	316464	130951	1.1E+07										
19	P08160	Adyar	1136	69	1	1	3	Normal	Se	Yes	#####	Others	AllPub	Paved	A	4.6	2.6	2.9	3.35	323214	49725	8287550									
20	P01372	Anna Nag	1902	168	2	1	5	Family	Yes	#####	Commercial	ELO	Paved	RL	4.5	4.9	4.9	4.76	742113	212032	2.1E+07										
21	P00936	Chrompet	1069	53	1	1	3	AbNormal	Yes	#####	Others	NoSewr	Paved	RM	2.8	2.7	4.9	3.39	283544	43622	1.1E+07										
22	P07346	Chrompet	931	96	1	1	3	AbNormal	Yes	#####	Others	ELO	Paved	RL	4.2	3.1	3.1	3.43	248021	27558	9185970										
23	P06851	KK Nagar	2010	114	3	2	5	Partial	No	#####	Others	NoSewr	Gravel	RH	3.4	4.5	3.4	3.73	327762	262210	1.1E+07										
24	P01655	Chrompet	1074	100	1	1	3	AbNormal	No	#####	Others	NoSewr	Gravel	DI	3	3.4	4.0	3.73	328811	266146	9633880										

Fig 3.2 Kaggle Data Set

3.6 PROGRAMMING LANGUAGE AND PACKAGES

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

The packages used in this model include:

- Sklearn
- NumPy
- Pandas
- Matplotlib
- Seaborn

3.6.1 SKLEARN

In python, sklearn is a machine learning package which include a lot of ML algorithms. Here, we are using some of its modules like train_test_split, accuracy_score etc.

3.6.2 NUMPY

It is a numeric python module which provides fast maths functions for calculations. It is used to read data in numpy arrays and for manipulation purpose.

3.6.3 PANDAS:

Used to read and write different files. Data manipulation can be done easily with data frames.

3.6.4 MATPLOTLIB:

Data visualization is a useful way to help identify the patterns from given dataset. Data manipulation can be done easily with data frames.

3.6.5 SEABORN (SNS):

Seaborn (sns) is an amazing data visualization library for statistical graphics plotting in Python. All of the data visualization have amazing colour palette for even better visuals.

3.7 DATA PRE-PROCESSING

Pre-processing refers to the transformations that are applied to the data before feeding it to the desired algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. When the data is collected from different sources it is gathered in a raw format which is not suitable for the analysis. Data collection, data analysis, and the process of reviewing data content, quality, and structure, can be a time consuming task. During the process of data identification, it helps in understanding the data and its properties; this knowledge will help in choosing the right algorithm to use to build the model. The steps/techniques for data cleaning will vary with each dataset. The primary goal of

data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

	PRT_ID	AREA	INT_SQFT	DATE_SALE	DIST_MAINROAD	N_BEDROOM	N_BATHROOM	N_ROOM	SALE_COND	PARK_FACIL	...	UTILITY_AVAIL	STR
0	P03210	Karapakkam	1004	04-05-2011	131	1.0	1.0	3	AbNormal	Yes	...	AllPub	P
1	P09411	Anna Nagar	1986	19-12-2006	26	2.0	1.0	5	AbNormal	No	...	AllPub	G
2	P01812	Adyar	909	04-02-2012	70	1.0	1.0	3	AbNormal	Yes	...	ELO	G
3	P05346	Velachery	1855	13-03-2010	14	3.0	2.0	5	Family	No	...	NoSewr	P
4	P06210	Karapakkam	1226	05-10-2009	84	1.0	1.0	3	AbNormal	Yes	...	AllPub	G

5 rows × 22 columns

Fig 3.3 Unprocessed Data

3.7.1 DATA CLEANING

All this present reality dataset are not really great; subsequently there is a presence of missing qualities created during the information section process. Accordingly, missing qualities are filled, information are sifted as required and improprie information are eliminated. This pre-processing of data helps in rendering highly accurate values and power of analysis also increased.

	AREA	INT_SQFT	DIST_MAINROAD	N_BEDROOM	N_BATHROOM	N_ROOM	PARK_FACIL	DATE_BUILD	BUILDTYPE	UTILITY_AVAIL	SALES_PRICE
0	2.0	1004	131	1.0	1.0	3	1	1967	2	1	7600000
1	1.0	1986	26	2.0	1.0	5	0	1995	2	1	21717770
2	5.0	909	70	1.0	1.0	3	1	1992	2	2	13159200
3	4.0	1855	14	3.0	2.0	5	0	1988	0	0	9630290
4	2.0	1226	84	1.0	1.0	3	1	1979	0	1	7406250

Fig 3.4 Processed Data

3.8 EXPLORATORY DATA ANALYSIS/ VISUALIZATION:

Sometimes data does not make sense until it is in a visual form, such as with charts and plots. Rapidly visualizing data samples is an important skill in both applied machine learning and applied statistics. Data visualization provides helps in gaining a qualitative understanding. This can be helpful when exploring and

getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more.

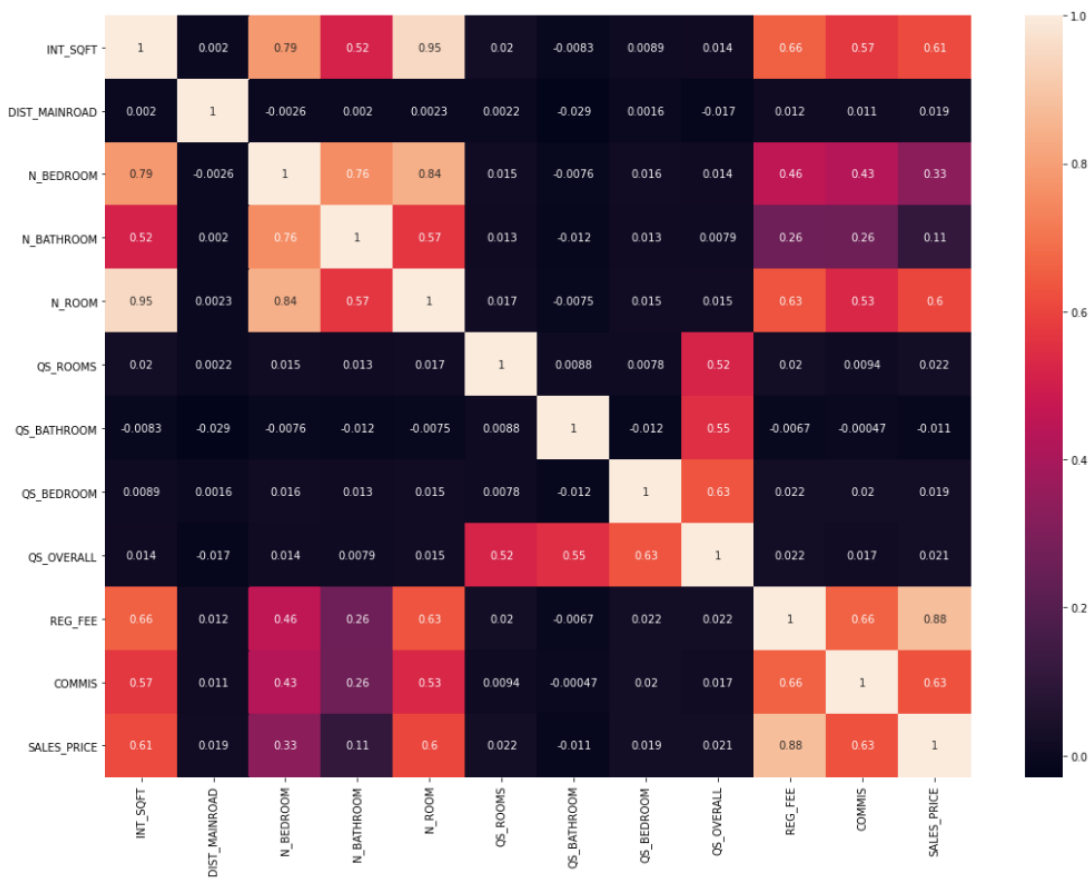


Fig 3.5 HeatMap depicting the correlation of all the attributes

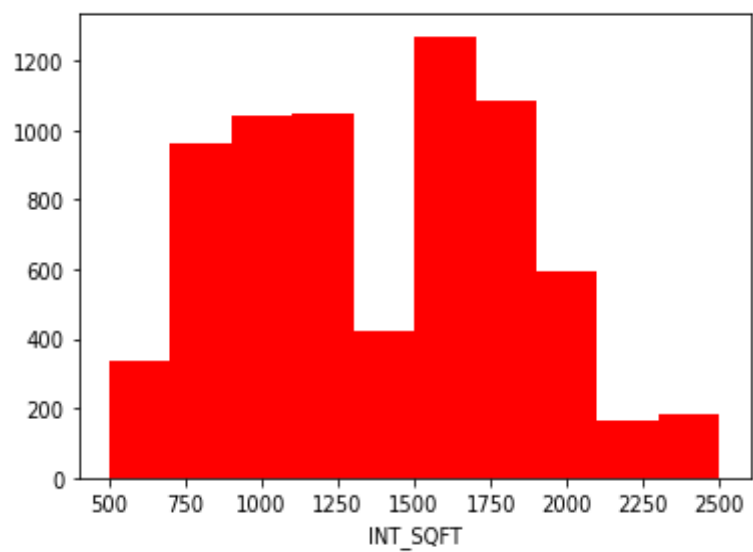


Fig 3.6 Histogram depicting the frequency of the various Square Feet Areas in the data set

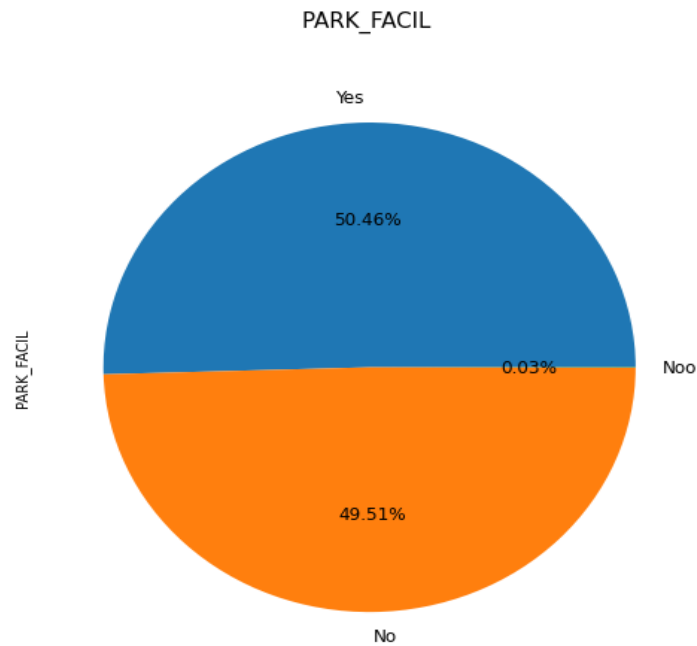


Fig 3.7 Pie Chart depicting the availability of parking facility

3.9 COMPARING ALGORITHMS:

It is important to compare the performance of different machine learning algorithms consistently. A test harness can be created to compare multiple different machine learning algorithms in Python with scikit-learn library. This test harness is used as a template on the machine learning problem to add more and different algorithms to compare. By using re-sampling methods like cross validation, an estimate for how accurate each model may be on unseen data is received. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In this model 7 different algorithms were compared:

3.9.1 LOGISTIC REGRESSION:

It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are

only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

3.9.2 LINEAR REGRESSION

Linear Regression is a machine learning algorithm based on supervised learning. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). It is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables.

3.9.3 RIDGE REGRESSION

Ridge regression is a model tuning strategy that is utilized to break down any information that experiences multicollinearity. This strategy performs L2 regularization. At the point when the issue of multicollinearity happens, least-squares are fair, and differences are huge, this results in anticipated values being far away from the real qualities.

3.9.4 RANDOM FOREST REGRESSION

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

3.9.5 SUPPORT VECTOR REGRESSION

Support Vector Machines (SVM) are popularly and widely used for classification and regression problems in machine learning. SVMs solve binary classification problems by formulating them as convex optimization problems. The optimization problem entails finding the maximum margin separating the hyper plane, while correctly classifying as many training points as possible.

3.9.6 LASSO REGRESSION

Lasso regression is a regularization technique SVMs solve binary classification problems by formulating them as convex optimization problems. The optimization problem entails finding the maximum margin separating the hyper plane, while correctly classifying as many training points as possible.

3.9.7 KNN REGRESSION

K-Nearest Neighbor is one of the most straightforward Machine Learning calculations in view of Supervised Learning strategy. It expects the likeness between the new case/information and accessible cases and put the new case into the class that is similar to the accessible classifications. The calculation stores every one of the accessible information and arranges another information point in view of the similitude. This implies when new information shows up then it tends to be effortlessly grouped into a well suited classification by utilizing K-NN calculation.

3.9.8 STOCHASTIC GRADIENT DESCENT REGRESSION

Stochastic Gradient Descent is a simple and efficient approach to fitting regressors and linear classifiers under convex loss functions such as (linear) Logistic Regression and Support Vector Machines.

3.9.9 DECISION TREE

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

For predicting residential house prices, Random forest regression algorithm is effective as it depicts the least error rate and highest accuracy rate.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 RESULT

Seven different algorithms were considered, before choosing the one which had the highest accuracy and least error rate. The model asks the user to enter the following details before it predicts the value of the property.

- Area Name
- Desired Square feet
- Distance from the main road
- Number of bedrooms
- Number of Bathrooms
- Parking facility
- No of years since the property was built
- Utility and building type

Once these details are filled in, the model predicts an accurate estimation of the value of the property. Users can use this estimation to plan their finances accordingly.

4.2 PERFORMANCE ANALYSIS

In order to determine the most suitable algorithm, certain model evaluating metrics are calculated for each algorithm. The metrics used are:

- **MEAN ABSOLUTE ERROR VALUE:** The mean of the absolute values of the individual prediction errors on over all instances in the test set.
- **MEAN SQUARED ERROR VALUE:** The average squared difference between the estimated values and the actual value
- **R2_SCORE:** The proportion of the variation in the dependent variable that is predictable from the independent variable.
- **ACCURACY RESULT**

4.2.1 ANALYSIS OF LINEAR REGRESSION

MEAN ABSOLUTE ERROR VALUE IS: **1720332.8442812087**

MEAN SQUARED ERROR VALUE IS: **4393730731989.679**

MEDIAN ABSOLUTE ERROR VALUE IS: **1534952.3736709952**

ACCURACY RESULT OF LINEAR REGRESSION IS:

70.33076403387113

R2_SCORE VALUE IS: **0.7031218945674971**

4.2.2 ANALYSIS OF LOGISTIC REGRESSION

MEAN ABSOLUTE ERROR VALUE IS : **3135463.1929990537**

MEAN SQUARED ERROR VALUE IS : **16751683964505.914**

MEDIAN ABSOLUTE ERROR VALUE IS : **2432730.0**

ACCURACY RESULT OF LOGISTIC REGRESSION IS : -

13.176167672609008

R2_SCORE VALUE IS : **-0.13188734165656624**

4.2.3 ANALYSIS OF SGD REGRESSION

MEAN ABSOLUTE ERROR VALUE IS : **883160526974816.2**

MEAN SQUARED ERROR VALUE IS : **7.955263863586042e+29**

MEDIAN ABSOLUTE ERROR VALUE IS : **881087571246447.0**

ACCURACY RESULT OF SGD REGRESSION IS : -

1.0509527609245117e+17

R2_SCORE VALUE IS : **0.7031218945674971**

4.2.4 ANALYSIS OF RIDGE REGRESSION

MEAN ABSOLUTE ERROR VALUE IS : **1680356.2272259383**

MEAN SQUARED ERROR VALUE IS : **4264862093385.803**

MEDIAN ABSOLUTE ERROR VALUE IS : **1463165.1527787223**

ACCURACY RESULT OF RIDGE REGRESSION IS :

70.9287562144931

R2_SCORE VALUE IS : **0.7091885255697601**

4.2.5 ANALYSIS OF KNN CLASSIFIER

MEAN ABSOLUTE ERROR VALUE IS : **4061316.3150425735**

MEAN SQUARED ERROR VALUE IS : **25219120198818.477**

MEDIAN ABSOLUTE ERROR VALUE IS : **3733290.0**

ACCURACY RESULT OF KNN IS : **8.301679228048986**

r2_score : **-0.7196357978903594**

4.2.6 ANALYSIS OF SUPPORT VECTOR REGRESSION

MEAN ABSOLUTE ERROR VALUE IS : **2934565.973596595**

MEAN SQUARED ERROR VALUE IS : **14994716728891.254**

MEDIAN ABSOLUTE ERROR VALUE IS : **2266095.476487669**

ACCURACY RESULT OF SVR IS : **0.006482638892613135**

R2_SCORE VALUE IS : **-0.020496162144285845**

4.2.7 ANALYSIS OF LASSO REGRESSION

MEAN ABSOLUTE ERROR VALUE IS : **1661892.4600778464**

MEAN SQUARED ERROR VALUE IS : **4177537836129.0835**

MEDIAN ABSOLUTE ERROR VALUE IS : **1450632.1542254686**

ACCURACY RESULT OF LASSO REGRESSION IS :

71.60224204301163

R2_SCORE VALUE IS : **0.7156891052987899**

4.2.8 ANALYSIS OF DECISION TREE REGRESSION

MEAN ABSOLUTE ERROR VALUE IS : **1066702.197256386**

MEAN SQUARED ERROR VALUE IS : **1854587190707.3674**

MEDIAN ABSOLUTE ERROR VALUE IS : **862330.0**

ACCURACY RESULT OF DECISION TREE REGRESSOR IS :

86.00426349098841

R2_SCORE VALUE IS : **0.8599890924433804**

4.2.9 ANALYSIS OF RANDOM FOREST REGRESSION

MEAN ABSOLUTE ERROR VALUE IS : **815543.7858798486**

MEAN SQUARED ERROR VALUE IS : **1062277450138.7345**

MEDIAN ABSOLUTE ERROR VALUE IS : **676469.7999999998**

ACCURACY RESULT OF RANDOM FOREST REGRESSOR IS:
91.98358228527732

R2_SCORE VALUE IS : **0.9198040239811384**

As Random Forest Regression showed the Highest R2_score, accuracy result and lowest mean absolute error value, mean squared error value, it is the algorithm that is used to construct this model.

CHAPTER 5

SUMMARY AND CONCLUSIONS

The process started from data cleaning and processing, exploratory analysis, algorithm comparison and finally model building and deploying. Although random forest regression was found to be best suited for this model, different machine learning algorithms like support vector regression, lasso regression etc, were considered. Random forest regression algorithm depicted the least amount of error (Mean Squared error; mean absolute error, root mean square error etc). This model is deployed using python flask server, which is also the back end portion of this project. HTML and CSS is used to develop the front end in order to make it more user friendly. The buyer is asked to fill in basic details such as the area in which he/she wants to purchase the house, their ideal sq-foot area, no of bathrooms and bedrooms , whether they would like to have a car parking, the type of drainage system, and a few other specifics. Based on the buyer's entry, the developed model predicts the accurate price of the property they desire to purchase. With this estimate, the buyer can prepare his finances accordingly.

HOUSE PRICE PREDICTION

AREA
ANNA NAGAR

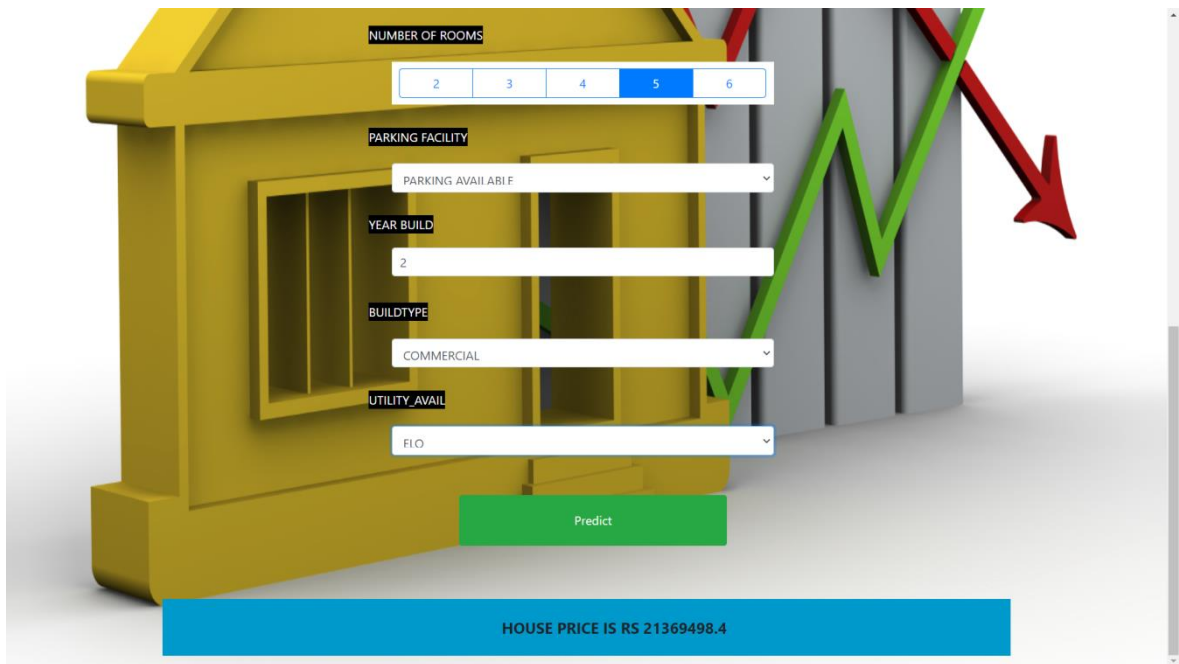
SQUARE FEET
5000

DISTANCE FROM MAIN ROAD
5

NUMBER OF BEDROOM
1 2 3 4

NUMBER OF BATHROOM
1 2

NUMBER OF ROOMS



NUMBER OF ROOMS

2 3 4 5 6

PARKING FACILITY

PARKING AVAILABLE

YEAR BUILD

2

BUILDTYPE

COMMERCIAL

UTILITY_AVAIL

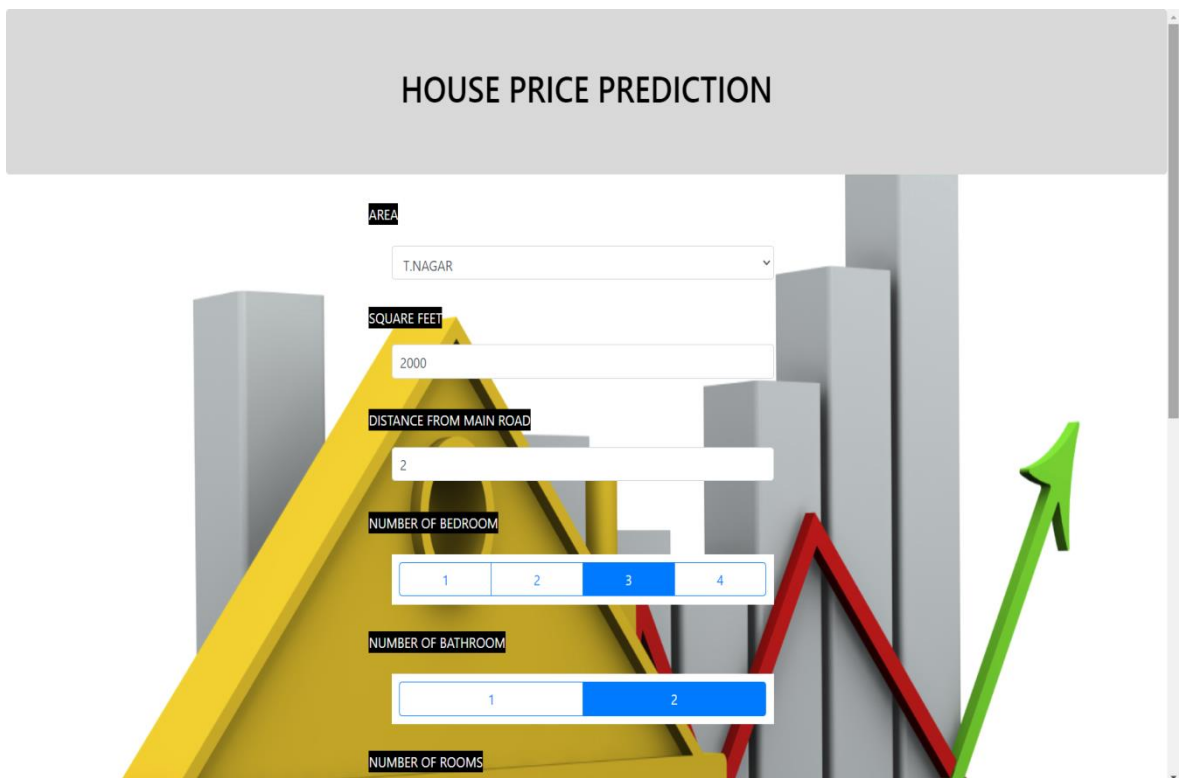
FLO

Predict

HOUSE PRICE IS RS 21369498.4

Fig 5.1 Real Estate Price Prediction - Anna Nagar

HOUSE PRICE PREDICTION



AREA

T.NAGAR

SQUARE FEET

2000

DISTANCE FROM MAIN ROAD

2

NUMBER OF BEDROOM

1 2 3 4

NUMBER OF BATHROOM

1 2

NUMBER OF ROOMS

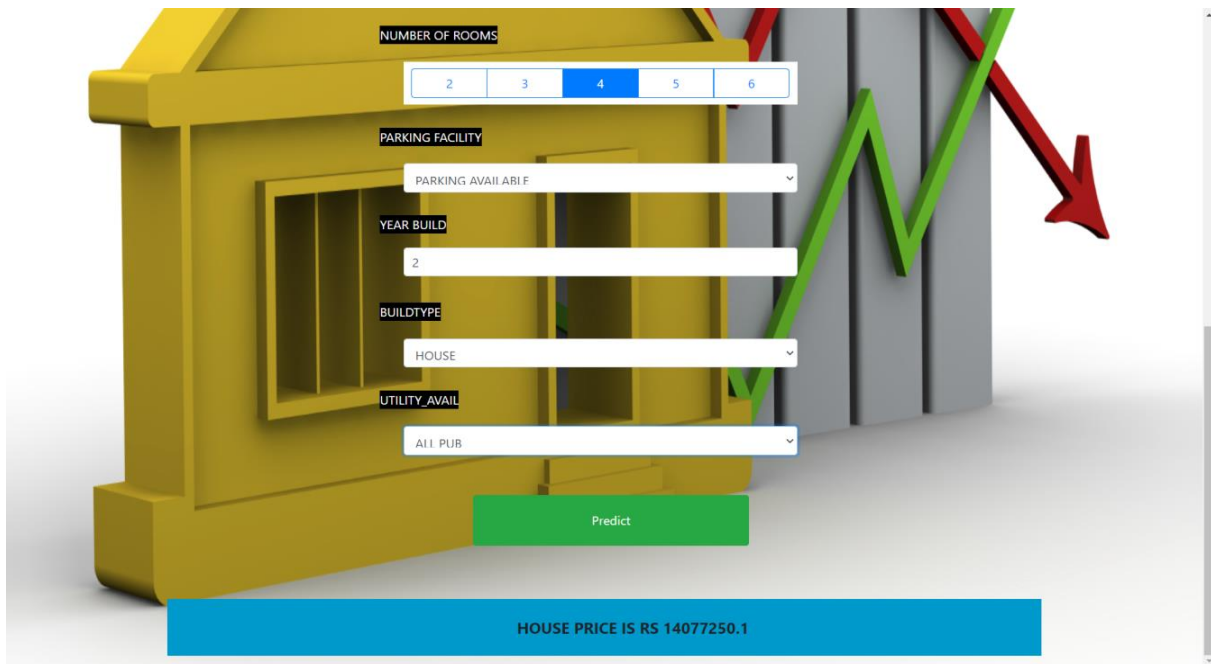


Fig 5.2 Real estate price prediction – T.Nagar

5.1 FUTURE SCOPE

In order to enhance this model, pipe lines can be incorporated into it. As time progresses this helps in keeping the data updated. Another enhancement could be to automate the model so that the data can be saved on a cloud. This way one model can predict the house prices for multiple cities as a cloud can hold bigger amount of data. Deploying this model in an artificial environment can be implemented as well.

REFERENCES

- [1] RushabSawant, Saurabh Jain, YashwantJangid, Ankita and Tushar Tiwari, "Comprehensive Analysis of Housing Price Prediction in Pune using Multi-Featured Random Forest Approach", 978-1-5386-52572/18/IEEE, 2018.
- [2] Danh Phan, "Housing Price Prediction using Machine Learning Algorithms: The Case of Melbourne City, Australia", 978-1-7281-0404- 1/19/IEEE, 2018.
- [3] Nehal N Ghosalkar and Sudhir N Dhage, "Real Estate Value Prediction Using Linear Regression", 978-15386-5257-2/18/IEEE, 2018.
- [4] P. Durganjali and M. Vani Pujitha, "House Resale Price Prediction Using Classification Algorithms", 978-17281-0027-2/19/IEEE, 2019.
- [5] Yong Piao, Ansheng Chen and Zhendong Shang, "Housing Price Prediction Based on CNN", 978-1-72812106-2/19/IEEE, 2019.
- [6] Yiyang Luo, " Residential Asset Pricing Prediction using Machine Learning", 978-1-7281- 5588-3/19/IEEE, 2019.
- [7] Mansi Jain, Himani Rajput, Neha Garg and Pronika Chawla, " Prediction of House Pricing Using Machine Learning with Python", 978-1- 7281-4108-4/20/IEEE, 2020.
- [8] Manasa J and Radha Gupta, "Machine Learning based Predicting House Prices using Regression Techniques", 978-1-7281-4167- 1/20/IEEE, 2020.
- [9] Maida Ahtesham, NarmeenZakariaBawany and Kiran Fatima, "House Price Prediction using Machine Learning Algorithm - The Case of Karachi City, Pakistan",978-1-7281- 8855-3/20/IEEE, 2020.
- [10] Mahmood Husain Abdulaziz and Ahmed M. Zeki, " Prediction of Real Estate Land Prices in the Kingdom of Bahrain", 978-1-7281-9677- 0/20/IEEE, 2020.

APPENDIX

A. SOURCE CODE

MODULE 1:

#Importing Libraries

```
import pandas as pd
```

```
import numpy as np
```

#Reading DataSet

```
data = pd.read_csv(r"C:\Users\SANJAY\Desktop\Project\train.csv")
```

```
data.head()
```

```
data.shape
```

```
df = data.dropna()
```

```
df.head()
```

```
data.shape
```

```
df.columns
```

```
df.info()
```

```
df.describe()
```

```
df.isnull()
```

```
df.isna().sum()
```

```
df.DIST_MAINROAD.unique()
```

```
df.AREA.unique()
```

```
df.SALE_COND.unique()
```

```
df.corr()
```

#Calculating the Mean

```
x = df.DIST_MAINROAD.mean()
print('DIST_MAINROAD mean value:',x)

x = df.QS_ROOMS.mean()
print('QS_ROOMS mean value:',x)

x = df.REG_FEE.mean()
print('REG_FEE mean value:',x)

df.duplicated().sum()
```

MODULE 2:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

```
warnings.filterwarnings('ignore')

df= pd.read_csv(r"C:\Users\SANJAY\Desktop\Project\train.csv")

df.head()

df.shape

df.columns
```

***# Histogram depicting the frequency of the various Square Feet Areas,
Distance from main road in the data set***

```
plt.hist(df['INT_SQFT'],color="red")
plt.xlabel("INT_SQFT")
plt.show()

plt.boxplot(df['INT_SQFT'])
plt.show()
```

```
plt.hist(df['DIST_MAINROAD'],color="green")
plt.xlabel("DIST_MAINROAD")
plt.show()
```

```
plt.figure(figsize=(10,6))
plt.plot(df['REG_FEE'])
plt.show()
```

Heatmap plot diagram

```
fig, ax = plt.subplots(figsize=(18,13))
sns.heatmap(df.corr(), ax=ax, annot=True)
```

#PieChart

```
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(8,11), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' \n', fontsize = 15)
    plt.show()
    return np.round(dataframe_pie/df.shape[0]*100,2)
PropByVar(df,"SALE_COND")
plt.show()
```

#PieChart - Available Parking

```
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(8,12), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' \n', fontsize = 15)
    plt.show()
    return np.round(dataframe_pie/df.shape[0]*100,2)
```

```
PropByVar(df,"PARK_FACIL")
```

```
plt.show()
```

```
data1=df.iloc[:,9]
```

```
data2=df.iloc[:,9:-1]
```

#Scatter Matrix

```
a=pd.plotting.scatter_matrix(data1, alpha=0.05, figsize=(18,10), diagonal='hist')
```

```
axis = 'off'
```

```
plt.show()
```

MODULE 3:

```
import pandas as pd
```

```
import numpy as np
```

```
df = pd.read_csv(r"C:\Users\SANJAY\Desktop\Project\train.csv")
```

```
df.isnull().sum()
```

```
data = df.dropna()
```

```
data.isnull().sum()
```

```
data.head()
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

```
data.head()
```

```
data.columns
```

#Analysing Unique values

```
data.AREA.unique()
```

```
data.SALE_COND.unique()
```

data.PARK_FACIL.unique()

data.BUILDTYPE.unique()

data.UTILITY_AVAIL.unique()

data.STREET.unique()

data.MZZONE.unique()

data.N_BEDROOM.unique()

data.N_ROOM.unique()

data.N_BATHROOM.unique()

data.N_BATHROOM.unique()

data['AREA'] = data["AREA"].map({'Chrompet':0, 'Anna Nagar':1, 'Karapakkam':2,
'KK Nagar':3, 'Velachery':4,
'Adyar':5, 'T Nagar':6, 'Karapakkam':2, 'Chrompet':0, 'Velchery':4,
'Chrompt':0, 'Ann Nagar':1, 'Chormpet':0})

data['SALE_COND'] = data["SALE_COND"].map({'AbNormal':0, 'AdjLand':1,
'Family':2, 'Partial':3, 'Normal Sale':4,
'Partial':5, 'PartialL':5, 'Adj Land':1})

data['BUILDTYPE'] = data["BUILDTYPE"].map({'Others':0, 'House':1,
'Commercial':2, 'Other':0, 'Commercil':2, 'Comercial':2})

data['UTILITY_AVAIL'] = data["UTILITY_AVAIL"].map({'NoSewr ':0, 'NoSeWa':0,
'AllPub':1, 'ELO':2, 'All Pub':1})

data['STREET'] = data["STREET"].map({'Paved':0, 'No Access':1, 'Gravel':2,
'Pavd':0})

data['PARK_FACIL'] = data["PARK_FACIL"].map({'No':0, 'Yes':1, 'Noo':0})

```
data['MZZONE'] = data["MZZONE"].map({'RH':0,'RM':1,'I':3,'RL':4,'A':5,'C':6})
```

```
data['DATE_BUILD'] = pd.DatetimeIndex(data['DATE_BUILD']).year
```

```
data.columns
```

#removing irrelevant data

```
del data['PRT_ID']
```

```
del data['DATE_SALE']
```

```
del data['SALE_COND']
```

```
del data['STREET']
```

```
del data['MZZONE']
```

```
del data['QS_ROOMS']
```

```
del data['QS_BATHROOM']
```

```
del data['QS_BEDROOM']
```

```
del data['QS_OVERALL']
```

```
del data['COMMIS']
```

```
del data['REG_FEE']
```

```
data=data.dropna()
```

```
data.head()
```

```
data.isnull().sum()
```

```
data.SALES_PRICE.unique()
```

```
X = data.drop(labels='SALES_PRICE', axis=1)
```

```
#Response variable
```

```
y = data.loc[:, 'SALES_PRICE']
```

```
from sklearn.metrics import
```

```
mean_absolute_error, mean_squared_error, r2_score, explained_variance_score, m
```

```
edian_absolute_error
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

#Logistic Regression

```
from sklearn.linear_model import LogisticRegression
logR= LogisticRegression()
logR.fit(X_train,y_train)
predictR = logR.predict(X_test)
MAE= (mean_absolute_error(y_test,predictR))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")
```

```
MSE=(mean_squared_error(y_test,predictR))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")
```

```
MedianAE=(median_absolute_error(y_test,predictR))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")
```

```
EVS=(explained_variance_score(y_test,predictR)*100)
print('ACCURACY RESULT OF LOGISTIC REGRESSION IS :',EVS)
print(" ")
```

```
R2=(r2_score(y_test,predictR))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

#Linear Regression

```
from sklearn.linear_model import LinearRegression
linR= LinearRegression()
linR.fit(X_train,y_train)
```

```

predictL = linR.predict(X_test)
MAE= (mean_absolute_error(y_test,predictL))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")

MSE=(mean_squared_error(y_test,predictL))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")

MedianAE=(median_absolute_error(y_test,predictL))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")

EVS=(explained_variance_score(y_test,predictL)*100)
print('ACCURACY RESULT OF LINEAR REGRESSION IS :',EVS)
print(" ")

R2=(r2_score(y_test,predictL))
print('R2_SCORE VALUE IS :',R2)
print(" ")

linR.coef_
linR.intercept_

```

#SGD Regressor

```

from sklearn.linear_model import SGDRegressor
SGD= SGDRegressor()
SGD.fit(X_train,y_train)
predictGD = SGD.predict(X_test)

MAE= (mean_absolute_error(y_test,predictGD))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)

```

```
print(" ")
```

```
MSE=(mean_squared_error(y_test,predictGD))  
print('MEAN SQUARED ERROR VALUE IS :',MSE)  
print(" ")
```

```
MedianAE=(median_absolute_error(y_test,predictGD))  
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)  
print(" ")
```

```
EVS=(explained_variance_score(y_test,predictGD)*100)  
print('ACCURACY RESULT OF SGD REGRESSION IS :',EVS)  
print(" ")
```

```
R2=(r2_score(y_test,predictL))  
print('R2_SCORE VALUE IS :',R2)  
print(" ")
```

MODULE 4:

```
import pandas as pd  
import numpy as np
```

```
df = pd.read_csv(r"C:\Users\SANJAY\Desktop\Project\train.csv")  
data = df.dropna()  
data.head()
```

```
import warnings  
warnings.filterwarnings('ignore')  
data.columns
```

```
data['AREA'] = data["AREA"].map({'Chrompet':0, 'Anna Nagar':1, 'Karapakkam':2,
'KK Nagar':3, 'Velachery':4,
'Adyar':5, 'T Nagar':6, 'Karapakkam':2, 'Chrompet':0, 'Velchery':4,
'Chrompt':0, 'Ann Nagar':1, 'Chormpet':0})
```

```
data['SALE_COND'] = data["SALE_COND"].map({'AbNormal':0, 'AdjLand':1,
'Family':2, 'Partial':3, 'Normal Sale':4,
'Partiall':5, 'Partiall':5, 'Adj Land':1})
```

```
data['BUILDTYPE'] = data["BUILDTYPE"].map({'Others':0, 'House':1,
'Commercial':2, 'Other':0, 'Commercil':2, 'Comercial':2})
```

```
data['UTILITY_AVAIL'] = data["UTILITY_AVAIL"].map({'NoSewr ':0, 'NoSeWa':0,
'AllPub':1, 'ELO':2, 'All Pub':1})
```

```
data['STREET'] = data["STREET"].map({'Paved':0, 'No Access':1, 'Gravel':2,
'Pavd':0})
```

```
data['PARK_FACIL'] = data["PARK_FACIL"].map({'No':0, 'Yes':1, 'Noo':0})
```

```
data['MZZONE'] = data["MZZONE"].map({'RH':0, 'RM':1, 'I':3, 'RL':4, 'A':5, 'C':6})
```

```
data['DATE_BUILD'] = pd.DatetimeIndex(data['DATE_BUILD']).year
```

```
del data['PRT_ID']
del data['DATE_SALE']
del data['SALE_COND']
del data['STREET']
del data['MZZONE']
del data['QS_ROOMS']
```

```

del data['QS_BATHROOM']
del data['QS_BEDROOM']
del data['QS_OVERALL']
del data['COMMIS']
del data['REG_FEE']

data = data.rename(columns={'REG_FEE': 'PRICE'})

data=data.dropna()

X = data.drop(labels='SALES_PRICE', axis=1)
#Response variable
y = data.loc[:, 'SALES_PRICE']

data.head()


#Random Forest Regression

from sklearn.metrics import
mean_absolute_error, mean_squared_error, r2_score, explained_variance_score, median_absolute_error

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)


from sklearn.ensemble import RandomForestRegressor
RF= RandomForestRegressor()
RF.fit(X_train,y_train)

predictR = RF.predict(X_test)

MAE= (mean_absolute_error(y_test,predictR))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")

MSE=(mean_squared_error(y_test,predictR))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")

```

```
MedianAE=(median_absolute_error(y_test,predictR))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")
```

```
EVS=(explained_variance_score(y_test,predictR)*100)
print('ACCURACY RESULT OF RANDOM FOREST REGRESSOR IS :',EVS)
print(" ")
```

```
R2=(r2_score(y_test,predictR))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

#Decision Tree Regression

```
from sklearn.tree import DecisionTreeRegressor
DT= DecisionTreeRegressor()
DT.fit(X_train,y_train)
```

```
predictD = DT.predict(X_test)
```

```
MAE= (mean_absolute_error(y_test,predictD))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")
```

```
MSE=(mean_squared_error(y_test,predictD))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")
```

```
MedianAE=(median_absolute_error(y_test,predictD))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")
```

```
EVS=(explained_variance_score(y_test,predictD)*100)
```

```
print('ACCURACY RESULT OF DECISION TREE REGRESSOR IS :',EVS)
print(" ")
```

```
R2=(r2_score(y_test,predictD))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

#Creating a Pickle file

```
import joblib
joblib.dump(RF,'Home1.pkl')
```

MODULE 5:

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv(r"C:\Users\SANJAY\Desktop\Project\train.csv")
data = df.dropna()
data.head()
```

```
import warnings
warnings.filterwarnings('ignore')
data.columns
data['AREA'] = data["AREA"].map({'Chrompet':0, 'Anna Nagar':1, 'Karapakkam':2,
'KK Nagar':3, 'Velachery':4,
'Adyar':5, 'T Nagar':6, 'Karapakkam':2, 'Chrompet':0, 'Velchery':4,
'Chrompt':0, 'Ann Nagar':1, 'Chormpet':0})
```

```
data['SALE_COND'] = data["SALE_COND"].map({'AbNormal':0, 'AdjLand':1,
'Family':2, 'Partial':3, 'Normal Sale':4,
```

```
'Partial':5, 'PartialI':5, 'Adj Land':1})
```

```
data['BUILDTYPE'] = data["BUILDTYPE"].map({'Others':0, 'House':1,  
'Commercial':2, 'Other':0, 'Commercil':2, 'Comercial':2})
```

```
data['UTILITY_AVAIL'] = data["UTILITY_AVAIL"].map({'NoSewr ':0, 'NoSeWa':0,  
'AllPub':1, 'ELO':2, 'All Pub':1})
```

```
data['STREET'] = data["STREET"].map({'Paved':0, 'No Access':1, 'Gravel':2,  
'Pavd':0})
```

```
data['PARK_FACIL'] = data["PARK_FACIL"].map({'No':0, 'Yes':1, 'Noo':0})
```

```
data['MZZONE'] = data["MZZONE"].map({'RH':0, 'RM':1, 'I':3, 'RL':4, 'A':5, 'C':6})
```

```
data['DATE_BUILD'] = pd.DatetimeIndex(data['DATE_BUILD']).year
```

```
del data['PRT_ID']
```

```
del data['DATE_SALE']
```

```
del data['SALE_COND']
```

```
del data['STREET']
```

```
del data['MZZONE']
```

```
del data['QS_ROOMS']
```

```
del data['QS_BATHROOM']
```

```
del data['QS_BEDROOM']
```

```
del data['QS_OVERALL']
```

```
del data['COMMIS']
```

```
del data['REG_FEE']
```

```
data=data.dropna()
```

```
X = data.drop(labels='SALES_PRICE', axis=1)
```

#Response variable

```
y = data.loc[:, 'SALES_PRICE']
```

```
from sklearn.metrics import
```

```
mean_absolute_error, mean_squared_error, r2_score, explained_variance_score, median_absolute_error
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

#Support Vector Regression

```
from sklearn.svm import SVR
```

```
svmA = SVR()
```

```
svmA.fit(X_train, y_train)
```

```
predictS = svmA.predict(X_test)
```

```
MAE = (mean_absolute_error(y_test, predictS))
```

```
print('MEAN ABSOLUTE ERROR VALUE IS :', MAE)
```

```
print(" ")
```

```
MSE = (mean_squared_error(y_test, predictS))
```

```
print('MEAN SQUARED ERROR VALUE IS :', MSE)
```

```
print(" ")
```

```
MedianAE = (median_absolute_error(y_test, predictS))
```

```
print('MEDIAN ABSOLUTE ERROR VALUE IS :', MedianAE)
```

```
print(" ")
```

```
EVS = (explained_variance_score(y_test, predictS) * 100)
```

```
print('ACCURACY RESULT OF SVR IS :', EVS)
```

```
print(" ")
```

```
R2=(r2_score(y_test,predictS))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

#Lasso Regression

```
from sklearn.linear_model import Lasso
```

```
la= Lasso()
```

```
la.fit(X_train,y_train)
```

```
predictLo = la.predict(X_test)
```

```
MAE= (mean_absolute_error(y_test,predictLo))
```

```
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
```

```
print(" ")
```

```
MSE=(mean_squared_error(y_test,predictLo))
```

```
print('MEAN SQUARED ERROR VALUE IS :',MSE)
```

```
print(" ")
```

```
MedianAE=(median_absolute_error(y_test,predictLo))
```

```
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
```

```
print(" ")
```

```
EVS=(explained_variance_score(y_test,predictLo)*100)
```

```
print('ACCURACY RESULT OF LASSO REGRESSION IS :',EVS)
```

```
print(" ")
```

```
R2=(r2_score(y_test,predictLo))
```

```
print('R2_SCORE VALUE IS :',R2)
```

```
print(" ")
```

MODULE 6:

```
import pandas as pd
```

```
import numpy as np
```

```
df = pd.read_csv(r"C:\Users\SANJAY\Desktop\Project\train.csv")
```

```
data = df.dropna()
```

```
data.head()
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

```
data.columns
```

```
data['AREA'] = data["AREA"].map({'Chrompet':0, 'Anna Nagar':1, 'Karapakkam':2,  
                                'KK Nagar':3, 'Velachery':4,  
                                'Adyar':5, 'T Nagar':6, 'Karapakkam':2, 'Chrompet':0, 'Velchery':4,  
                                'Chrompt':0, 'Ann Nagar':1, 'Chormpet':0})
```

```
data['SALE_COND'] = data["SALE_COND"].map({'AbNormal':0, 'AdjLand':1,  
                                           'Family':2, 'Partial':3, 'Normal Sale':4,  
                                           'Partiall':5, 'Partiall':5, 'Adj Land':1})
```

```
data['BUILDTYPE'] = data["BUILDTYPE"].map({'Others':0, 'House':1,  
                                           'Commercial':2, 'Other':0, 'Commercil':2, 'Comercial':2})
```

```
data['UTILITY_AVAIL'] = data["UTILITY_AVAIL"].map({'NoSewr ':0, 'NoSeWa':0,  
                                                    'AllPub':1, 'ELO':2, 'All Pub':1})
```

```
data['STREET'] = data["STREET"].map({'Paved':0, 'No Access':1, 'Gravel':2,  
                                     'Pavd':0})
```

```
data['PARK_FACIL'] = data["PARK_FACIL"].map({'No':0, 'Yes':1, 'Noo':0})
```

```
data['MZZONE'] = data["MZZONE"].map({'RH':0,'RM':1,'I':3,'RL':4,'A':5,'C':6})
```

```
data['DATE_BUILD'] = pd.DatetimeIndex(data['DATE_BUILD']).year
```

```
del data['PRT_ID']
```

```
del data['DATE_SALE']
```

```
del data['SALE_COND']
```

```
del data['STREET']
```

```
del data['MZZONE']
```

```
del data['QS_ROOMS']
```

```
del data['QS_BATHROOM']
```

```
del data['QS_BEDROOM']
```

```
del data['QS_OVERALL']
```

```
del data['COMMIS']
```

```
del data['REG_FEE']
```

```
data=data.dropna()
```

```
X = data.drop(labels='SALES_PRICE', axis=1)
```

```
#Response variable
```

```
y = data.loc[:, 'SALES_PRICE']
```

```
from sklearn.metrics import
```

```
mean_absolute_error, mean_squared_error, r2_score, explained_variance_score, m
```

```
edian_absolute_error
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
#Ridge Regression
```

```
from sklearn.linear_model import Ridge
```

```
rid= Ridge()
```

```
rid.fit(X_train,y_train)
```

```
predictri = rid.predict(X_test)
```

```
MAE=(mean_absolute_error(y_test,predictri))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")
```

```
MSE=(mean_squared_error(y_test,predictri))
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")
```

```
MedianAE=(median_absolute_error(y_test,predictri))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")
```

```
EVS=(explained_variance_score(y_test,predictri)*100)
print('ACCURACY RESULT OF RIDGE REGRESSION IS :',EVS)
print(" ")
```

```
R2=(r2_score(y_test,predictri))
print('R2_SCORE VALUE IS :',R2)
print(" ")
```

KNeighborsClassifier

```
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier()
knn.fit(X_train,y_train)
```

```
predictk= knn.predict(X_test)
```

```
MAE=(mean_absolute_error(y_test,predictk))
print('MEAN ABSOLUTE ERROR VALUE IS :',MAE)
print(" ")
```

```
MSE=(mean_squared_error(y_test,predictk))
```

```
print('MEAN SQUARED ERROR VALUE IS :',MSE)
print(" ")
```

```
MedianAE=(median_absolute_error(y_test,predictk))
print('MEDIAN ABSOLUTE ERROR VALUE IS :',MedianAE)
print(" ")
```

```
EVS=(explained_variance_score(y_test,predictk)*100)
print('ACCURACY RESULT OF KNN IS :',EVS)
print(" ")
```

```
R2=(r2_score(y_test,predictk))
print('r2_score :',R2)
print(" ")
```

PYTHON FLASK - DEPLOYMENT

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import joblib
```

```
app = Flask(__name__)
model = joblib.load('Home1.pkl')
```

```
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
```

```

'''
For rendering results on HTML GUI
'''

int_features = [(x) for x in request.form.values()]
final_features = [np.array(int_features)]
print(final_features)

prediction = model.predict(final_features)

output = prediction[0]

print(output)

return render_template('index.html', prediction_text='HOUSE PRICE IS RS
{}'.format(output))

if __name__ == "__main__":
    app.run(host="localhost", port=8000)

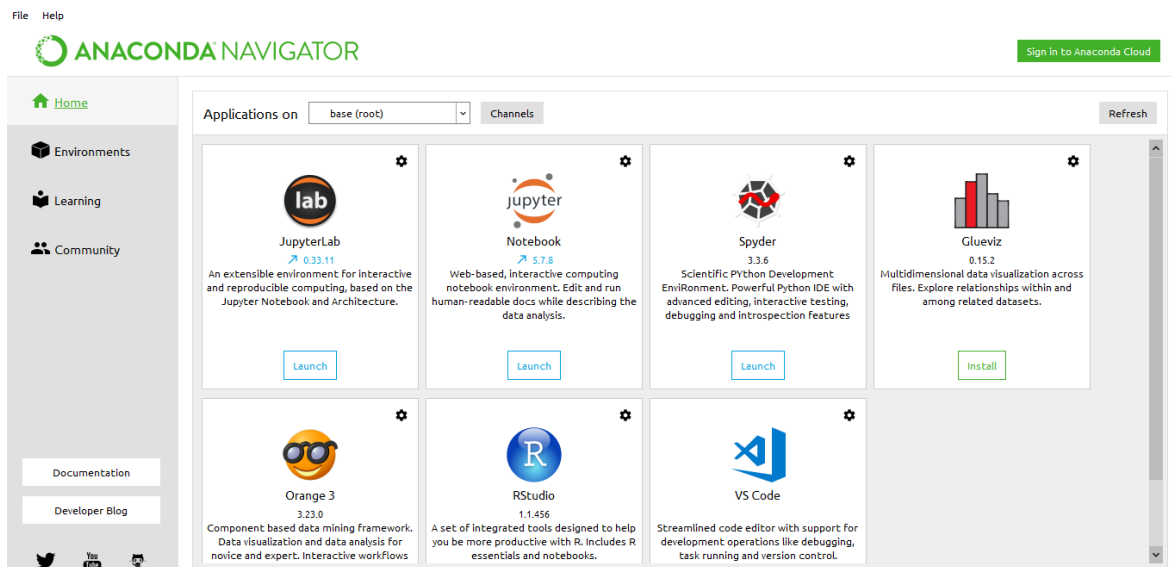
```

B. SCREENSHOTS

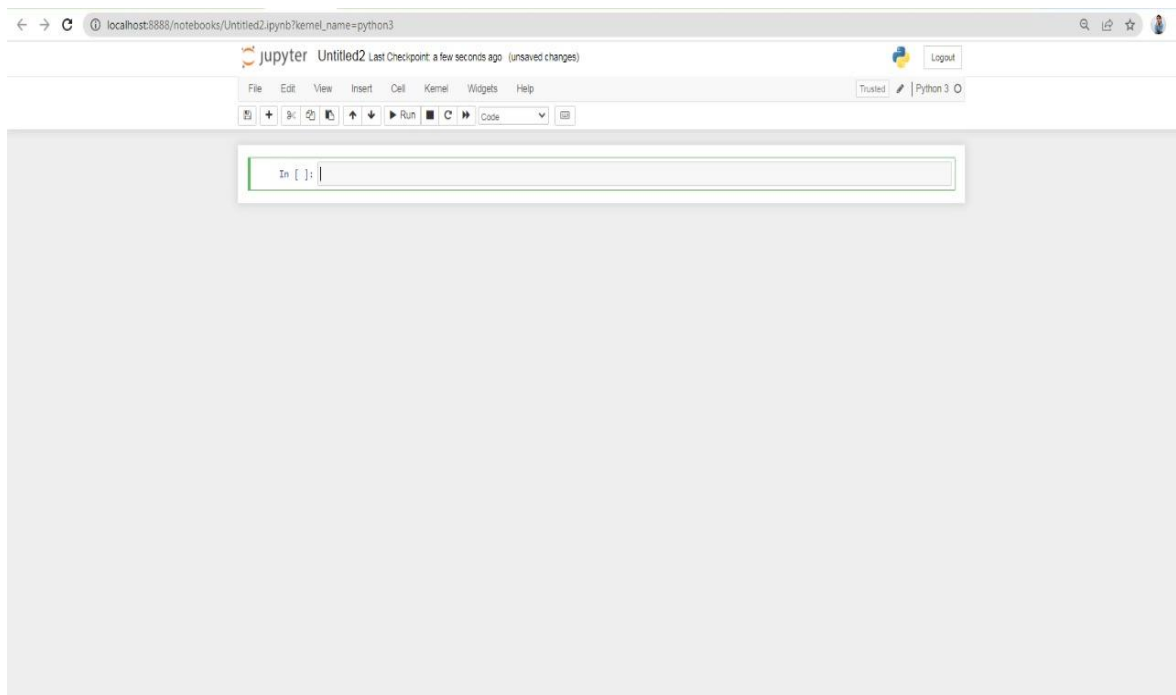
train - Microsoft Excel

train - Microsoft Excel																					
Home Insert Page Layout Formulas Data Review View Developer																					
Normal Page Layout Page Break Preview Custom Views Full Screen ☒ Rules ☒ Formula Bar ☐ Gridlines ☒ Headings ☐ Message Bar Zoom 100% Zoom to Selection New Window Arrange All Freeze Panes Split Hide Unhide View Side by Side Synchronous Scrolling Reset Window Position Save Workspace Switch Windows Macros																					

Data Set from Kaggle



Anaconda Navigator



Jupyter Notebook

HOUSE PRICE PREDICTION

AREA

CHROME PET

SQUARE FEET

1000

DISTANCE FROM MAIN ROAD

5

NUMBER OF BEDROOM

1

2

3

4

NUMBER OF BATHROOM

1

2

NUMBER OF ROOMS

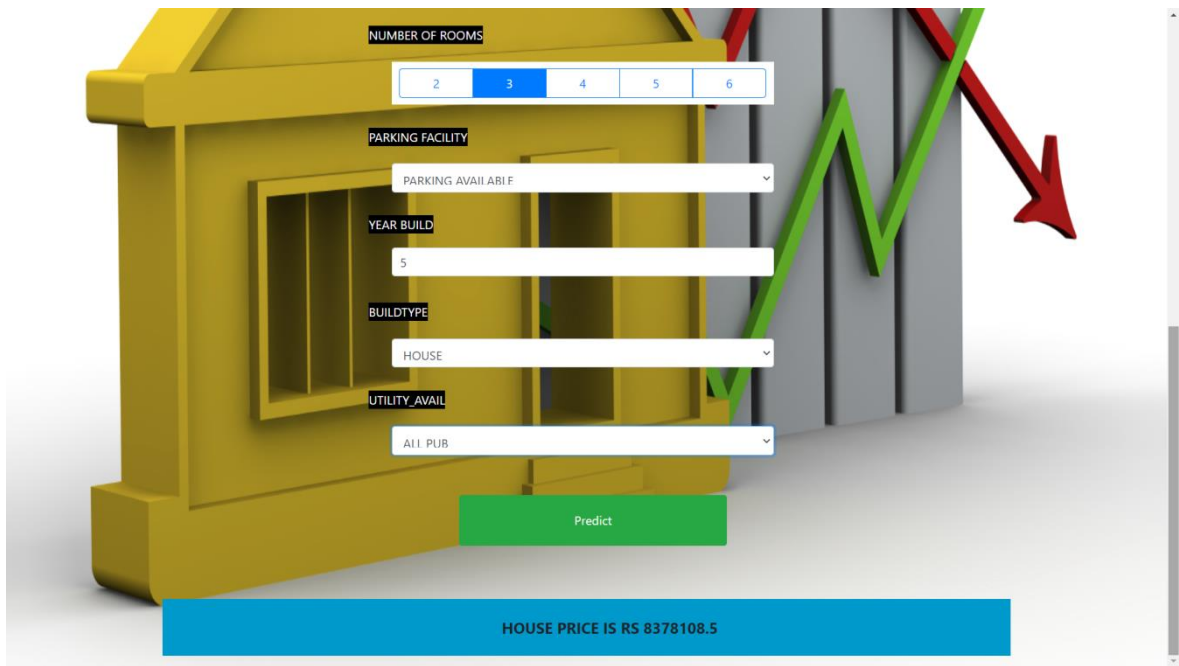
2

3

4

5

6



A 3D-rendered web application interface for real estate price prediction. The background features a yellow house model on the left and a grey bar chart with a green line graph and a red downward arrow on the right. The form is overlaid on the house model. It includes a 'NUMBER OF ROOMS' section with a row of buttons (2, 3, 4, 5, 6), where '3' is selected. Below this is a 'PARKING FACILITY' dropdown menu with 'PARKING AVAILABLE' selected. The 'YEAR BUILD' section has a text input field containing '5'. The 'BUILDTYPE' dropdown menu has 'HOUSE' selected. The 'UTILITY_AVAIL' dropdown menu has 'ALL PUB' selected. A green 'Predict' button is located below the form fields. At the bottom, a blue banner displays the prediction result: 'HOUSE PRICE IS RS 8378108.5'.

NUMBER OF ROOMS

2 3 4 5 6

PARKING FACILITY

PARKING AVAILABLE

YEAR BUILD

5

BUILDTYPE

HOUSE

UTILITY_AVAIL

ALL PUB

Predict

HOUSE PRICE IS RS 8378108.5

Real Estate Price Prediction - Chrompet