

A Project report

on

HOUSE PRICE PREDICTION

*Submitted in partial fulfillment of the requirements
for the award of the degree of*

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

A. PRATHIBHA (184G1A0559)

P. BHAVYA SRI (184G1A0508)

A. HARSHITHA (184G1A0522)

H. JAVEED (184G1A0526)

Under the Guidance of

Dr. B. Hari Chandana M.Tech., Ph.D.

Assistant Professor



Department Of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram village, B K Samudram Mandal, Ananthapuramu-515701.

2021-2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram village, B K Samudram Mandal, Ananthapuramu-515701.



Certificate

This is to certify that the project report entitled **House Price Prediction** is the bonafide work carried out by **A. Prathibha** bearing Roll Number 184G1A0559, **P. Bhavya Sri** bearing Roll Number 184G1A0508, **A. Harshitha** bearing Roll Number 184G1A0522, **H. Javeed** bearing Roll Number 184G1A0526 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2021-2022.

Guide

Dr. B. Hari Chandana M. Tech., Ph. D.
Assistant Professor

Head of the Department

Mr. P. Veera Prakash M. Tech.,(Ph. D)
Assistant Professor & HOD

Date:

EXTERNAL EXAMINER

Place: Rotarypuram

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Dr. B. Hari Chandana**, M. Tech., Ph. D, **Assistant professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. we thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Mr. K. Venkatesh**, M. Tech., **Assistant Professor**, project coordinator for valuable guidance and unstinting encouragement that enable us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash**, M. Tech., (Ph. D), **Assistant Professor & Head of the Department, Computer Science & Engineering**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna**, M. Tech., Ph. D **Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the management for providing excellent facilities.

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

DECLARATION

We, Ms. A. Prathibha bearing reg no: 184G1A0559, Ms. P. Bhavya Sri bearing reg no: 184G1A0508, Ms. A. Harshitha bearing reg no: 184G1A0522, Mr. H. Javeed bearing reg no:184G1A0526 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, here by declare that the dissertation entitled “HOUSE PRICE PREDICTION” embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Dr. B. Hari Chandana, M.Tech.,Ph.D, Assistant Professor, Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other Universities of Institute for the award of any Degree or Diploma.

A. PRATHIBHA	184G1A0559
P. BHAVYA SRI	184G1A0508
A. HARSHITHA	184G1A0522
H. JAVEED	184G1A0526

CONTENTS	Page No.
List of figures	VIII
List of screens	IX
List of abbreviations	X
Abstract	XI
Chapter 1: Introduction	1
1.1 Objective of the project	2
1.2 Machine Learning	2
1.2.1 Supervised Learning	3
1.2.1.1 Regression	4
1.2.1.2 Classification	4
1.2.1.3 Classification Algorithm	4
1.2.1.3.1 Decision Tree	4
1.2.1.4 Steps involved in supervised learning	6
1.3 Linear Regression	6
1.4 XGBoost Regression	7
1.5 Random Forest	7
1.6 Decision Tree Classifiers	8
1.7 Dimensionality Reduction	8
1.7.1 Feature selection	9
1.7.2 Feature extraction	9
1.7.3 Benefits of applying Dimensionality reduction	9
Chapter 2: Literature Survey	10
Chapter 3: Analysis	12
3.1 Introduction	12
3.2 Software Requirements specification	12
3.3 Hardware Requirements	13
3.4 Software Requirements	13
3.5 Jupyter notebook	13
3.5.1 Introduction	13
3.5.2 Installation	14
3.5.3 Project jupyter overview	16
Chapter 4: Design	17
4.1 UML Introduction	17

4.1.2 Usage of UML in project	17
4.2 Use Case diagram	17
4.3 Architecture of project	18
4.4 Steps involved in design	19
4.4.1 Data collection	20
4.4.2 Data Preprocessing	20
4.4.2.1 Explanatory Data Analysis	20
4.4.2.2 Filling Missing Data & data Encoding	21
4.4.3 Training the model	22
4.4.3.1 Linear Regression	23
4.4.3.2 Decision Tree Regression	23
4.4.3.3 XGBoost Regression	23
4.4.3.4 Random Forest Regression	24
4.4.4 Model evaluation	24
Chapter 5: Implementation	26
5.1 Libraries used	26
5.2 Implementation	30
5.2.1 Importing modules and libraries	31
5.2.2 Data visualization	31
5.2.2.1 Descriptive analysis	31
5.2.2.2 Univariate analysis	34
5.2.2.3 Correlation matrix	36
5.3 Feature engineering	36
5.4 Filling missing values and label encoding	37
5.4.1 Filling missing values with mean and median	37
5.4.2 Outlier Analysis	38
5.4.3 Label Encoding	39
5.4.4 Splitting the dataset	40
5.5 Model prediction	41
5.5.1 Modelling with Linear regression	41
5.5.2 Modelling with Decision Tree regression	42
5.5.3 Modelling with XGBoost Regression	42
5.6 Evaluation of models	43
5.6.1 Prediction of outputs	43

5.7 Dimensionality reduction	43
Chapter 6: Testing	45
6.1 Introduction	45
6.2 Black box testing	45
6.3 White box testing	46
6.4 Performance evaluation	46
Conclusion	48
References	49

LIST OF FIGURES

Fig. No.	Title	Page. No.
1.1	Types of Machine Learning	2
1.2	Process of ML Algorithm	3
1.3	Types of Supervised Learning	4
1.4	Classification Algorithm	6
1.5	Linear Regression	7
1.6	Dimensionality Reduction	9
4.1	Use case diagram for house price prediction	18
4.2	Architecture of house price prediction	19
5.1	Training and test dataset	29

LIST OF SCREENS

Screen No.	Title	Page No.
3.1	Jupyter Notebook	15
3.2	Notebook of jupyter notebook	16
5.2	Dataset with 9 different attributes	30
5.3	Importing of modules & libraries	31
5.4	Pandas head method	32
5.5	Pandas info method	33
5.6	Pandas describe method	33
5.7	Analysis of Area_type	34
5.8	Corelation Values	36
5.9	Filling the missing values	37
5.10	Label encoding of attributes	40
5.11	Train and Test split of data	40
5.12	Fitting model from linear regression	41

LIST OF ABBREVIATIONS

CSV	Comma-separated values
SRS	Software Requirement Specification
UML	Unified modelling language
NumPy	Numerical Python
ML	Machine Learning
XGBoost	Extreme gradient boost
SVM	Support Vector Machine
ROC	Receiver Operating Characteristics
PR	Precision Recall
EDA	Explanatory Data Analysis
API	Application Program Interface

ABSTRACT

Housing prices are a crucial reflection of the economy, and property values are of great interest for consumers as well as sellers. Real Estate is the one of the least transparent industries in our ecosystem. Predicting house prices with real time factors is the main aim of this research project. This paper aims to make valuations based on some basic parameters which are considered while determining the price of a house. In order to carry out the real time research, real time housing data of from one City has been collected manually.

The project tends to use Regression technique for Machine learning as we are dealing with continuous outcome variable. We have carried out a research by implementing different regression models to compare and determine the most effective model to resolve given problem statement. The goal of this research project is to create an effective machine learning model that is able to accurately estimate the price of the house based on given features and deploy the machine learning model in the form of a website to reach out individuals.

Keywords:

Machine learning, House Prices Prediction, Linear Regression, XG Boost, Decision Tree Regression, Random Forest Regression.

CHAPTER 1

INTRODUCTION

In the past years, Machine learning has proven to be able to solve real world problems using various algorithms. It plays a major role in advances of medical imaging, spam and fraud detection, enhancements in automobile industry, security alerts and Business Analysis. Here, we have used machine learning algorithms to perform predictive analysis of house prices to provide an overview of real estate businesses and property. It provides the information in a detailed format which is able to be understood by machines. Real estate prices keep changing frequently based on certain parameters. For a Real Estate Business, data is the most important source for analysis and predictions. It is always a perk to know about the predictions of variations of an entity which will be happening near future and business managers can act accordingly to avoid future loss. And for this we need a most accurate predicting Model for analysis. Similarly, we need a proper prediction on the real estate and the houses in the housing market to provide appropriate estimation of prices to help real estate managers know about prophecies. Buying a house will be a life time goal for most of the individuals but there are a lot of people who make huge mistakes while buying the properties. One of the common mistakes is buying properties that are too expensive but it's not worth it. Various methods have been used in the price prediction.

This project aims to predict the real estate price using the machine learning techniques with the help of the Real-Time Data of houses in Bangalore, India. The goal of this statistical analysis is to help us understand the relationship between house features and how these variables are used to predict house price. It uses comparison of Regression algorithms to find out best fitting model to predict the house price. So, it would be helpful for the people to avoid them from making mistakes. The results proven that this approach yields minimum error and most accuracy than individual algorithms applied. The goal of this project is to make a machine learning model that is able to accurately estimate the worth of the house given the options.

1.1 Objective of the Project

The objective is to perform techniques for house price predictions. On the basis of a performance evaluation, a best suited predictive model is suggested for the company sale. The results are summarized in terms of accuracy of machine learning techniques taken for prediction. The main objective of the system is to analyze the future prices of a particular house and to predict whether a particular house price will increase or decrease by using a different machine learning algorithm.

1.2 Machine Learning

Machine Learning is the area of study which enables machines to learn without being explicitly programmed. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data". A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it.

Machine learning gives a system the ability to learn automatically and improve its recommendations using data alone, with no additional programming needed. Because retailers generate enormous amounts of data, machine learning technology quickly proves its value. When a machine learning system is fed data—the more, the better—it searches for patterns. Going forward, it can use the patterns it identifies within the data to make better decisions. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

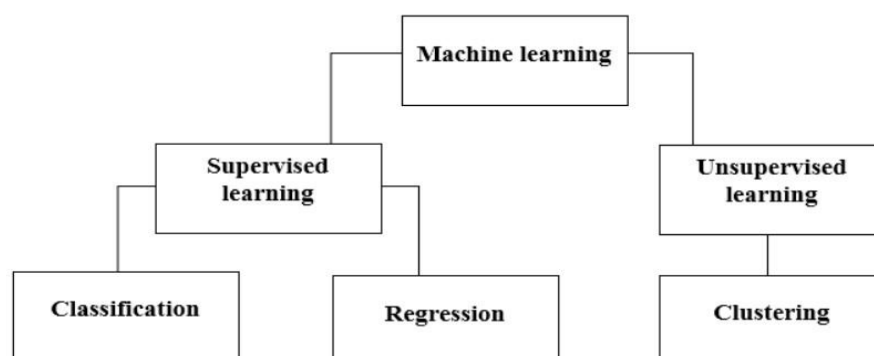


Fig.1.1. Types of Machine Learning

1.2.1. Supervised Learning

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y). Supervised learning is the type of machine learning in which machines are trained using well "labeled" trained data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output. The working of Supervised learning can be easily understood by the below example and diagram:

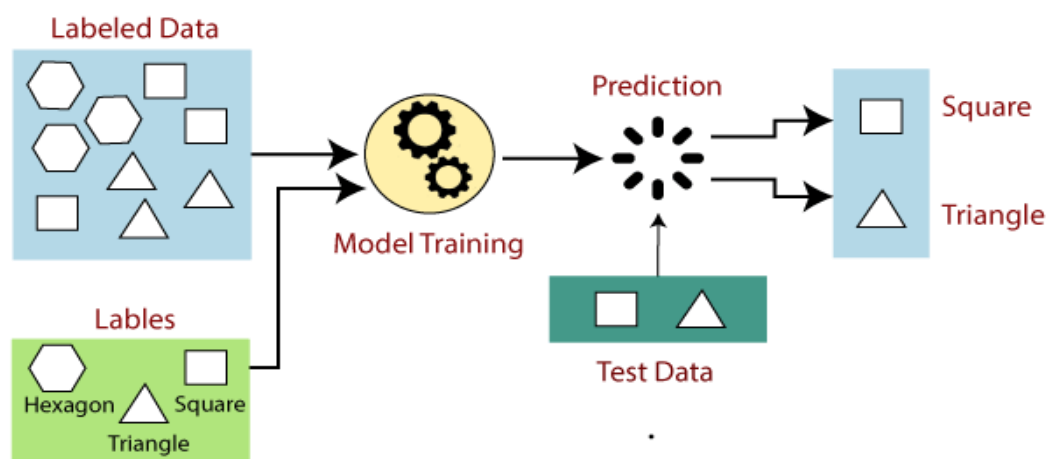


Fig.1.2. Process of any ML algorithm

Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

- If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square.
- If the given shape has three sides, then it will be labelled as a triangle.
- If the given shape has six equal sides, then it will be labelled as hexagon.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape. The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output. Supervised learning can be further divided into two types:

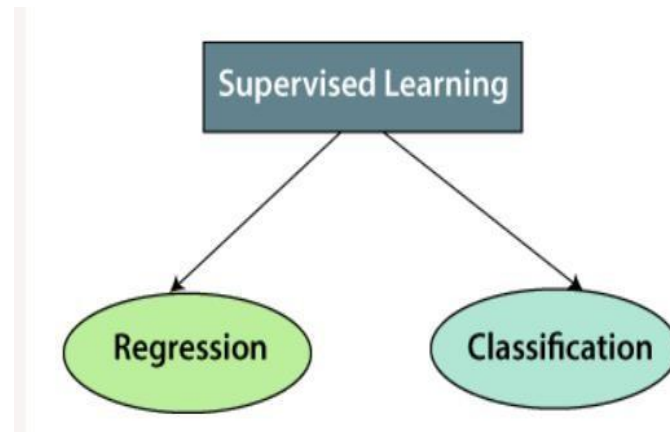


Fig.1.3. Types of supervised learning

1.2.1.1 Regression

Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables. It is mainly used for prediction. Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as price, etc. Some real-world examples for regression are predicting the sales based on input parameters etc.

1.2.1.2 Classification

Classification is supervised learning. It can be performed on both structured and unstructured data. Classification is the process of finding a model that helps to separate the data into different categorical classes. In this process, data is categorized under different labels according to some parameters given in input and then the labels are predicted for the data.

1.2.1.3 Classification Algorithm

1.2.1.3.1 Decision Tree

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. An example is shown in the diagram at right. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

A decision tree is a simple representation for classifying examples. For this section, assume that all of the input features have finite discrete domains, and there is a single target feature called the "classification". Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target or output feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. See the examples illustrated in the figure for spaces that have and have not been partitioned using recursive partitioning, or recursive binary splitting. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data.

The dependent variable, Y , is the target variable that we are trying to understand, classify or generalize. The vector X is composed of the features, x_1 , x_2 , x_3 etc., that are used for that task.

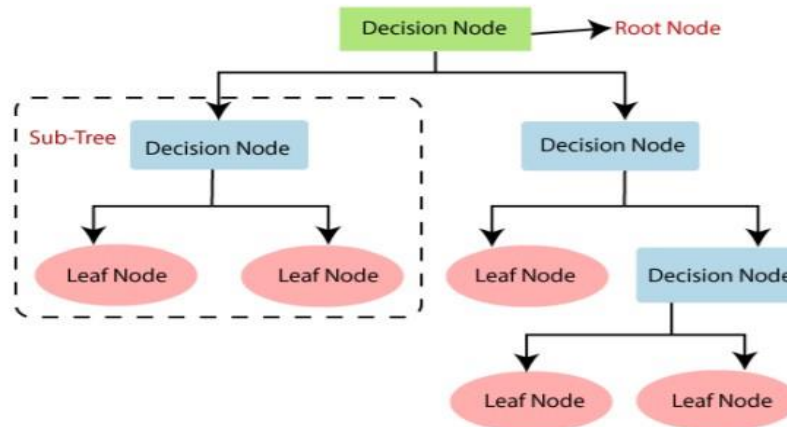


Fig.1.4. Classification algorithm

1.2.1.4 Steps Involved in Supervised Learning

- 1) First Determine the type of training dataset
- 2) Collect/Gather the labelled training data.
- 3) The training dataset into training dataset, test dataset, and validation dataset.
- 4) Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- 5) Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- 6) Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- 7) Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

1.3. Linear Regression

Linear regression is one of the most basic types of regression in machine learning. The linear regression model consists of a predictor variable and a dependent variable related linearly to each other. In case the data involves more than one independent variable, then linear regression is called multiple linear regression models. The below-given equation is used to denote the linear regression model:

$$y = mx + c + e$$

where, m is the slope of the line, c is an intercept, and e represents the error in the model.

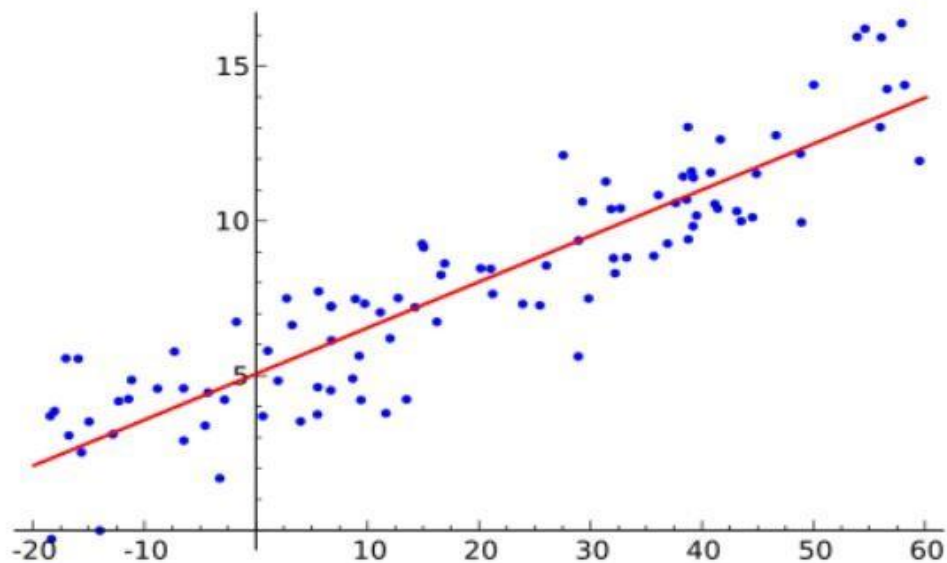


Fig.1.5. Linear regression

The best fit line is determined by varying the values of m and c . The predictor error is the difference between the observed values and the predicted value. The values of m and c get selected in such a way that it gives the minimum predictor error. It is important to note that a simple linear regression model is susceptible to outliers. Therefore, it should not be used in case of big size data.

1.4. XG Boost

XGBoost stands for eXtreme Gradient Boosting. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. It uses a gradient boosting framework. XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data. The implementation of the algorithm was engineered for the efficiency of computing time and memory resources. Boosting is a sequential technique which works on the principle of an ensemble. It combines a set of weak learners and improves prediction accuracy. At any instant t , the model outcomes are weighed based on the outcomes of previous instant $t-1$. The outcomes predicted correctly are given a lower weight and the ones misclassified are weighted higher.

1.5. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

1.6. Decision Tree Classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision-making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C_1, C_2, \dots, C_k is as follows:

Step 1. If all the objects in S belong to the same class, for example C_i , the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O_1, O_2, \dots, O_n . Each object in S has one outcome for T so the test partitions S into subsets S_1, S_2, \dots, S_n where each object in S_i has outcome O_i for T . T becomes the root of the decision tree and for each outcome O_i we build a subsidiary decision tree by invoking the same procedure recursively on the set S_i .

1.7. Dimensionality Reduction

Dimensionality reduction technique can be defined as, "It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information." These techniques are widely used in machine learning for obtaining a better fit predictive model while solving the classification and regression problems. It is commonly used in the fields that deal with high dimensional data. It can also be used for data visualization, etc.

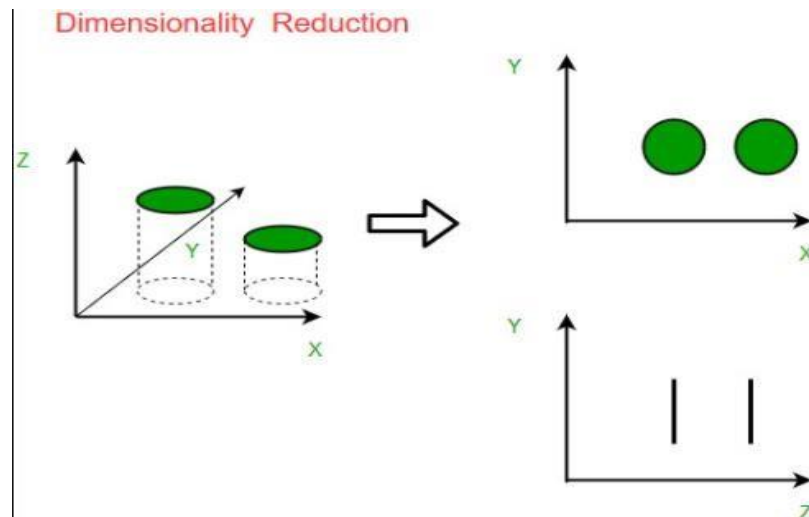


Fig.1.7. Dimensionality reduction

1.7.1. Feature Selection

Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy. In other words, it is a way of selecting the optimal features from the input dataset.

1.7.2. Feature Extraction

Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions. This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

1.7.3. Benefits of applying Dimensionality Reduction

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.
- Less Computation training time is required for reduced dimension of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.

CHAPTER 2

LITERATURE SURVEY

[Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair - 1]: Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation. Predicting housing prices with real factors is the main crux of our research project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price. 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 results proved that this approach yields minimum error and maximum accuracy than individual algorithms applied. We also propose to use real-time neighborhood details using Google maps to get exact real-world valuations.

[G. Naga Satish, Ch. V. Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu -2]: Predictive models for deciding the sale price of houses in metropolitan cities is still remaining as more challenging and trickier task. The sale price of properties in cities like Hyderabad depends on a variety of interdependent factors. Key factors which may affect the house price include area of the property, location of the property and its amenities. In this system, an attempt has been made to construct a predictive model for evaluating the price based on the factors that affect the price. Modelling study apply some supervised learning techniques such as Bayesian classifier or KNN algorithms. Such models are used to build a predictive model, and to pick the best performing model by performing a comparative analysis on the predictive errors obtained between these models. Here, the attempt is to construct a predictive model for evaluating the price based on factors that affects the price. We build this concept as real time application useful for real estate business and also buyer and sellers.

[CH. Raga Madhuri, G. Anuradha, M. Vani Pujitha - 3]: This paper provides an overview about how to predict house costs utilizing different regression methods with the assistance of python libraries. The proposed technique considered the more refined aspects used for the calculation of house price and provide the more accurate

prediction. It also provides a brief about various graphical and numerical techniques which will be required to predict the price of a house. This paper contains what and how the house pricing model works with the help of machine learning and which dataset is used in our proposed model.

[4] A SVR based forecasting approach for real estate price prediction: The support vector machine (SVM) has been successfully applied to classification, cluster, and forecast. This study proposes support vector regression (SVR) to forecast real estate prices in China. The aim of this paper was to examine the feasibility of SVR in real estate price prediction. The experimental results were calculated based on the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root mean squared error (RMSE) and the SVR based approach was an efficient tool to forecast real estate prices. [Hong Zhao, Rong-Qiu Chen, Wei Xu, Da-Ying Li Published in: 2009 International Conference on Machine Learning and Cybernetics].

[5] Using machine learning algorithms for housing price prediction: This study used machine learning to develop housing price prediction models. This study analyzes the housing data of 5359 townhouses in Fairfax County, VA. The 10-fold cross-validation was applied to C4.5, RIPPER, Bayesian, and AdaBoost. [The case of Fairfax County, Virginia housing data, Byeonghwa Parka, Jae Kwon Baeb, Department of Business Statistics, Hannam University, 70 Hannam-ro, Daedeok-gu, Republic of Korea].

CHAPTER 3

ANALYSIS

3.1 Introduction

The Analysis Phase is where the project life cycle begins. This is the phase where you break down the deliverables in the high-level Project Charter into the more detailed business requirements. Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps.

1. **Elicitation** – I ask questions, you talk, I listen
2. **Validation** – I analyze, I ask follow-up questions
3. **Specification** – I document, I ask follow-up questions
4. **Verification** – We all agree

Most of the work in the Analysis Phase is performed by the role of analyst.

3.2 Software Requirement Specification

SRS is a document created by a system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from clients are written in natural language. It is the responsibility of system analysts to document the requirements in technical language so that they can be comprehended and useful by the software development team.

3.3 Hardware Requirements

Any Contemporary PC.

3.4 Software Requirements

Operating system: Windows 7 Or Windows 10

Tools: Jupyter Notebook

Dataset: CSV file

Languages Used: Python

Frameworks Used: Flask

3.5 Jupyter notebook

3.5.1 INTRODUCTION

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python and R. Jupyter ships with the IPython kernel, which allows you to write your programs in python, but there are currently over 100 other kernels that you can also use.

IPython notebook was developed by Fernando Perez as a web based front end to IPython kernel. As an effort to make an integrated interactive computing environment

for multiple languages, the Notebook project was shifted under Project Jupyter providing front end for programming environments Julia and R in addition to python.

A notebook document consists of rich text elements with HTML, formatted text, figures, mathematical equations etc. The notebook is also an executable document consisting of code blocks in python or other supporting languages.

Getting Up and Running with Jupyter Notebook

The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

There are many distributions of the Python language. This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website.

It is also assumed that you are using Python 3.

3.5.2. Installation

If so, then you can use a handy tool that comes with Python called pip to install Jupyter Notebook like this:

```
$pip install jupyter
```

This will start up Jupyter and your default browser should start (or open a new tab) to the respective URL.

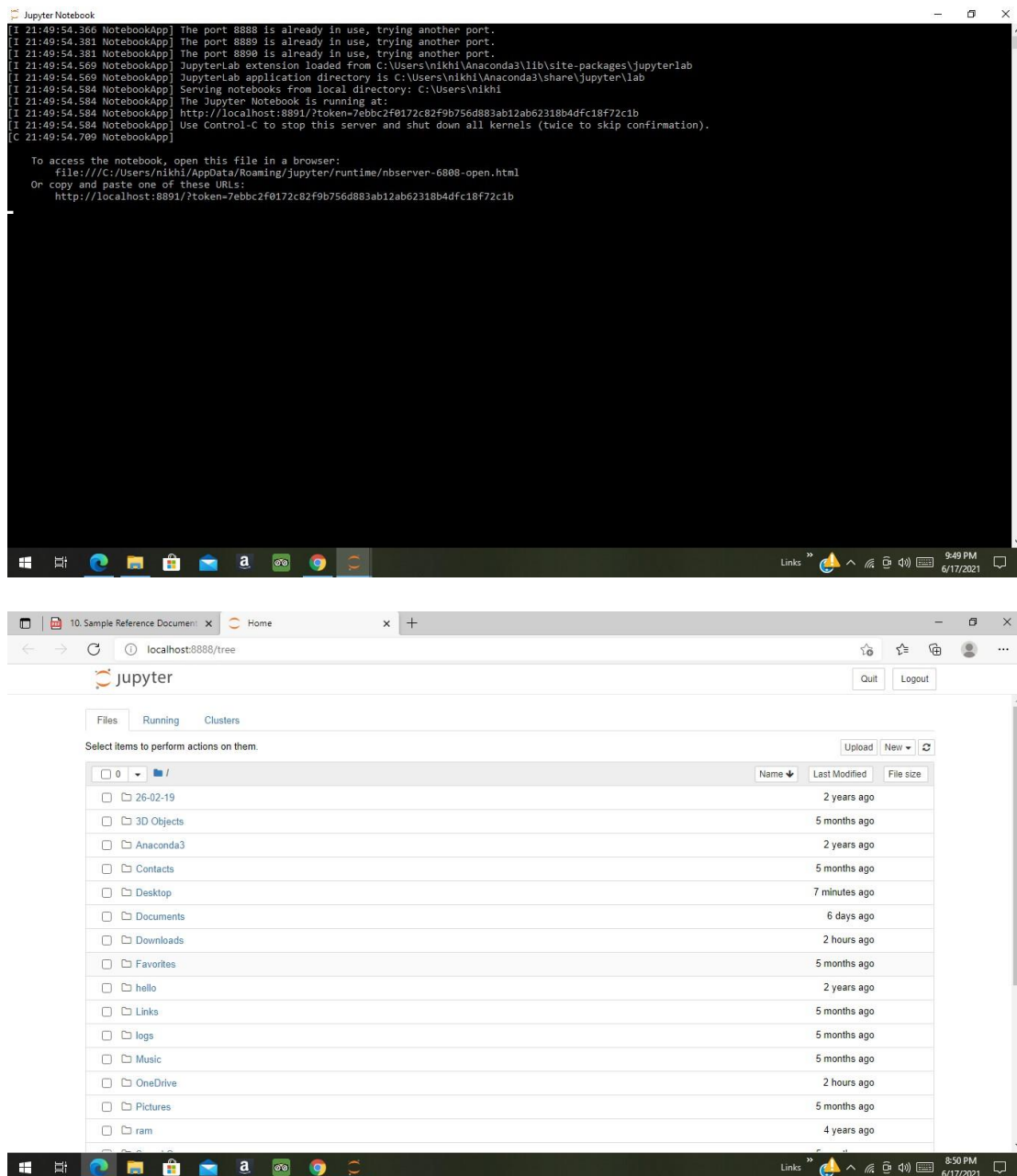


Fig.3.1 Jupyter notebook

Creating a Notebook

Now that you know how to start a Notebook server, you should probably learn how to create an actual Notebook document.

All you need to do is click on the new button (upper right), and it will open up a list of choices.

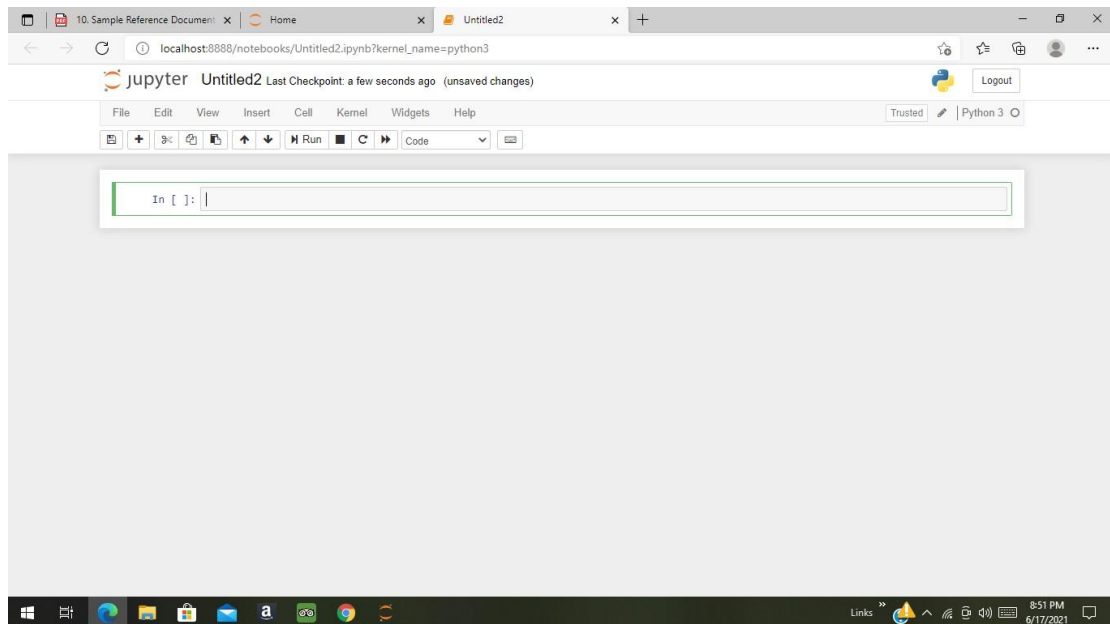


Fig.3.2- Notebook of Jupyter notebook

3.5.3. Project Jupyter Overview

Project Jupyter started as a spin-off from the IPython project in 2014. IPython's language-agnostic features were moved under the name – Jupyter. The name is a reference to core programming languages supported by Jupyter which are Julia, Python and R. Products under the Jupyter project are intended to support interactive data science and scientific computing.

The project Jupyter consists of various products described as under –

- **IPykernel** – This is a package that provides the IPython kernel to Jupyter.
- **Jupyter client** – This package contains the reference implementation of the Jupyter protocol. It is also a client library for starting, managing and communicating with Jupyter kernels.
- **Jupyter notebook** – This was earlier known as IPython notebook. This is a web based interface to IPython kernel and kernels of many other programming languages.
- **Jupyter kernels** – Kernel is the execution environment of a programming language for Jupyter products.
- **Qtconsole** – A rich Qt-based console for working with Jupyter kernels.

CHAPTER 4

DESIGN

4.1 UML Introduction

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from a distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

4.1.2 Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams

4.2 Use Case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

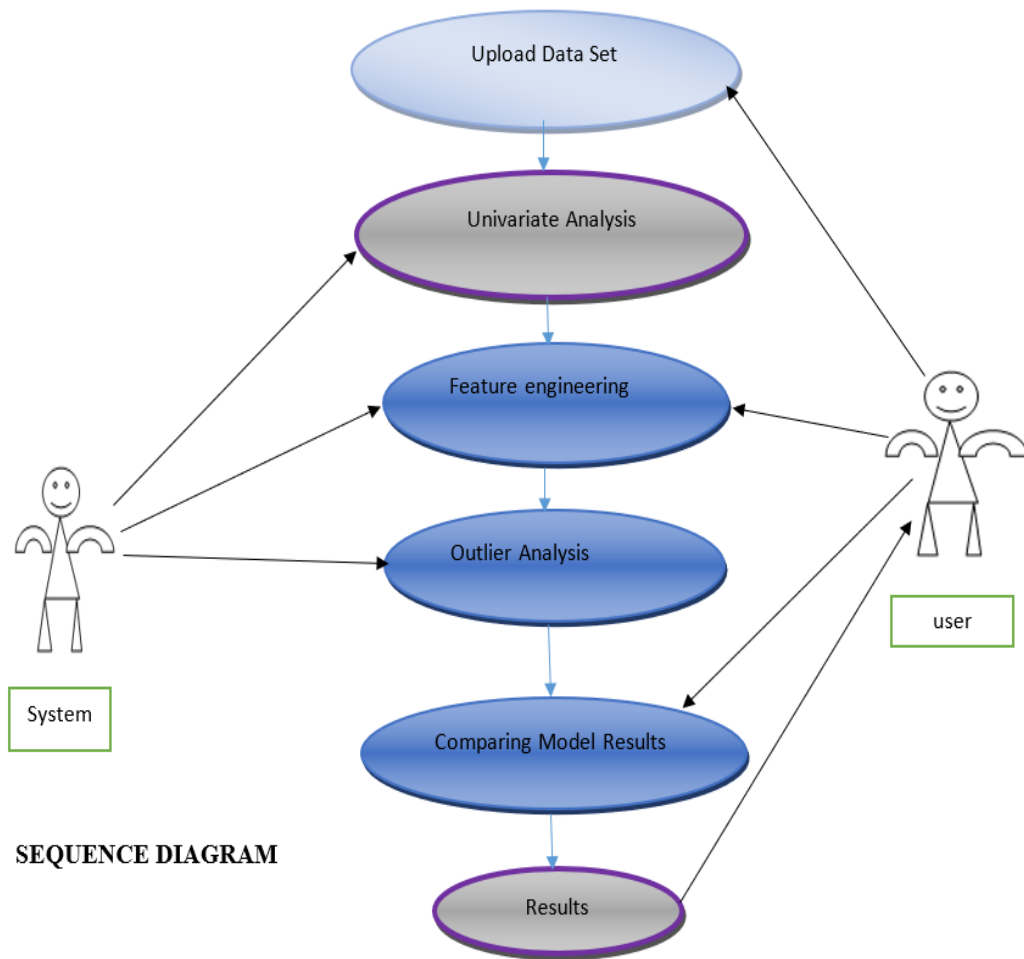


Fig.4.1. Use case diagram for house price prediction

4.3 Architecture of project

An architecture is a way of representing the flow of data of a process or a system (usually an information system). This also provides information about the outputs and inputs of each entity and the process itself. Machine learning architecture defines the various layers involved in the machine learning cycle and involves the major steps being carried out in the transformation of raw data into training data sets capable for enabling the decision making of a system.

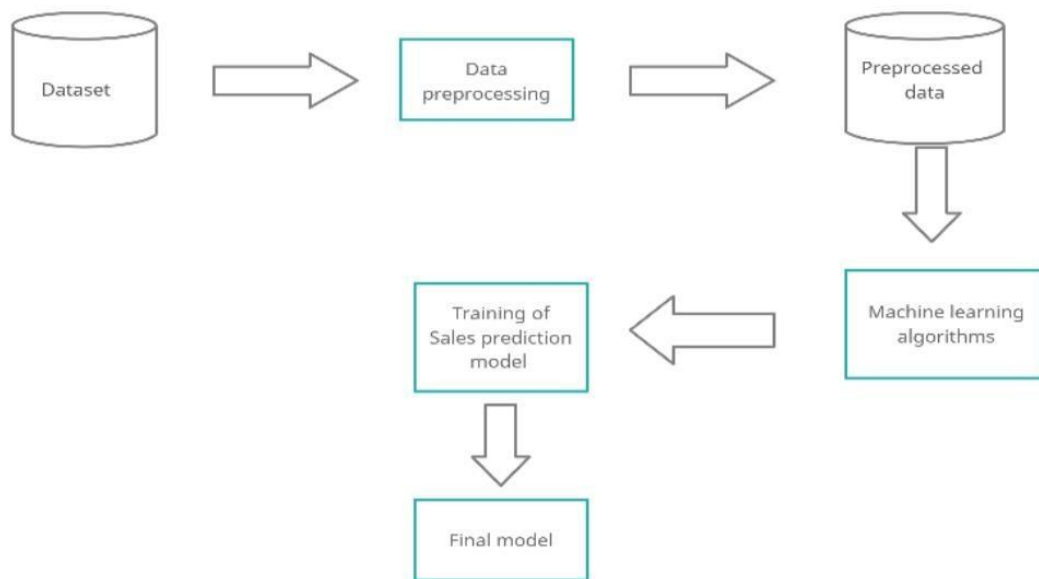


Fig.4.2. Architecture of house price prediction

The above architecture describes how house price prediction can be done. Initially obtain the datasets from kaggle website. After obtaining the datasets, perform data transformation to it in such a way that there shouldn't be any integration problem or any redundancy issue.

Now, apply feature selection techniques to the bangalore house price dataset which has 9 features in it. Then a subset of features which are most important in the prediction of future house prices. choose the algorithm that gives the best possible accuracy with the subset of features obtained after feature selection. Applying the algorithms to the dataset actually means that it needs to train the model with the algorithms and test the data so that the model will be fit.

4.4 Steps involved in Design

- Data Collection
- Data Pre-processing
- Model Training
- Model Evaluation

4.4.1 Data Collection

- Data is an important asset for developing any kind of Machine learning model. Data collection is the process of gathering and measuring information from different kinds of sources.
- This is an initial step that has to be performed to carry out a Machine learning project. In the present internet world these datasets are available in different websites (Ex: Kaggle, Google public datasets, Data.gov etc)
- The dataset used in our project is downloaded from the Kaggle website and it contains nearly 13321 records and 9 different attributes (area_type, availability, location, size, society, total_sqft, price etc).
- The dataset consists of 8 independent attributes and one dependent attribute (House Price).
- So, the aim of the project is to predict the dependent variables using independent variables.

4.4.2 Data Pre-Processing

- Data Preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.
- When creating a machine learning project, it is not always the case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So, for this, we use data Preprocessing tasks.
- Preprocessing of the data consists of different kinds of steps in which analysis of the data, Data cleaning, Data encoding are part of this.

4.4.2.1 Explanatory Data Analysis

- Exploratory data analysis is an approach of analyzing datasets to summarize their main characteristics, often using statistical graphics and other data visualization methods.
- The main purpose of EDA is to help look at data before making any assumptions.

- It can help identify obvious errors, as well as better understanding patterns within the data, detect outliers or anomalous events, and find interesting relations among the variables.
- Specific statistical functions and techniques you can perform with EDA tools include:
 - ➔ Clustering and dimension reduction techniques, which help create graphical display of high dimensional data containing many variables.
 - ➔ Univariate visualization of each field in the raw dataset, with summary statistics.
 - ➔ Bivariate visualizations and summary statistics that allows you to assess the relationship between each variable in the dataset and the target variable in the dataset and the target variable you're looking at.
 - ➔ Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
 - ➔ Predictive models, such as linear regression, use statistics and data to predict outcomes.
- This data analysis is of two types:
 - a. Univariate analysis
 - b. Bivariate analysis

Univariate analysis is the simplest form of data analysis where the data being analysed contains only one variable. Since it's a single variable it doesn't deal with causes or relationships.

Bivariate data is data that involves two different variables whose values can change.

Bivariate data deals with relationships between these two variables.

4.4.2.2 Filling Missing Data & Data Encoding

- The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

- By calculating the mean and Mode: In this way, we will calculate the mean or Mode of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc.
- Data encoding: Since the machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.
- Our dataset also consists of different categorical data in which they are encoded in this step.

4.4.3 Training the Model

In this step the model is trained using the algorithms that are suitable for house price prediction is a kind of problem in which One variable has to be determined using some independent variables. Regression model is suitable for this kind of scenario.

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output.

Our project implements four algorithms Linear regression, XG boost regression, Decision Tree regression and Random Forest Regression where Linear regression is a normal regression algorithm and Xgboost is a Gradient descent algorithm.

4.4.3.1 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).

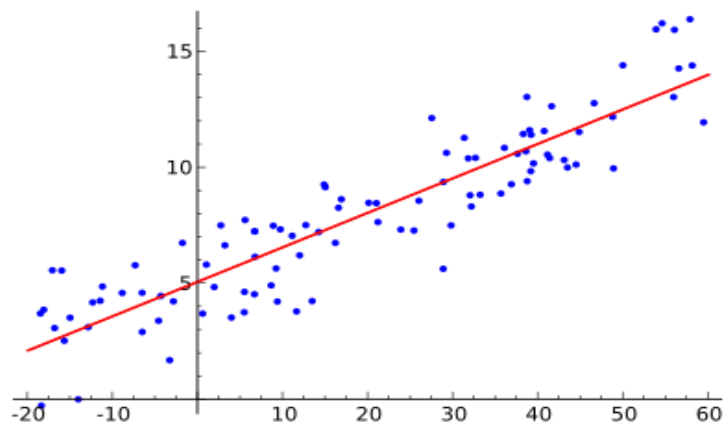


Fig.4.5.1. Linear regression

4.4.3.2 Decision Tree Regression

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

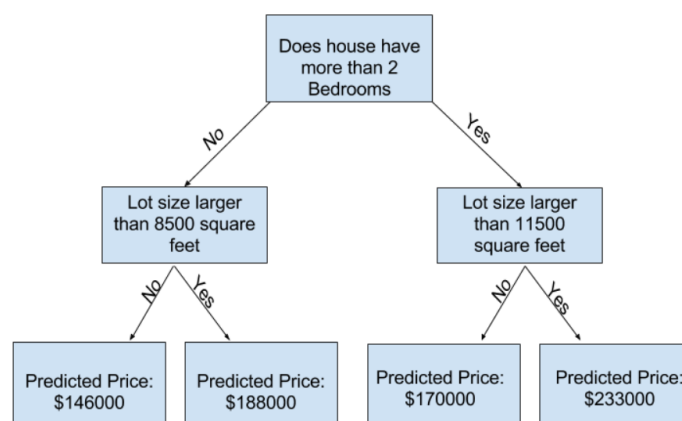


Fig.4.5.2. Decision Tree Regression

4.4.3.3 XGBoost Regression

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.). A wide range of applications: Can be used to solve regression, classification, ranking, and user-defined prediction problems.

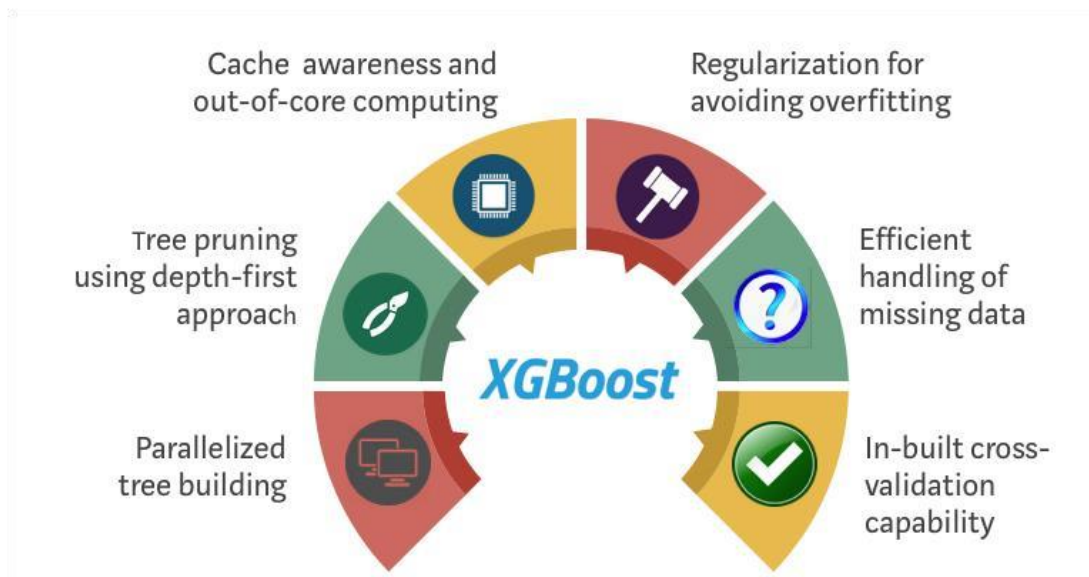


Fig.4.5.3. XGBoost Regressor

4.4.3.4 Random Forest Regression

Random Forest Regression is a supervised learning algorithm that uses **ensemble learning** method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

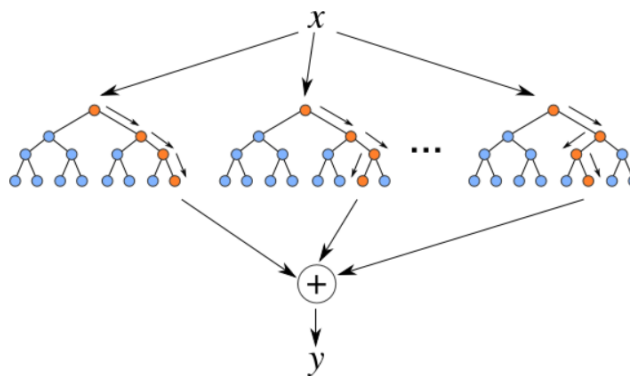


Fig.4.5.2. Random Forest Regression

4.4.4 Model Evaluation

In this step the trained model is evaluated by determining the accuracy of the model against the test data.

various ways to check the performance of our machine learning or deep learning model and why to use one in place of the other. We will discuss terms like:

1. Accuracy
2. Precision
3. Recall
4. Specificity
5. F1 score
6. Precision-Recall or PR curve
7. ROC (Receiver Operating Characteristics) curve
8. PR vs ROC curve.

Out of these we used Accuracy for evaluating our model. Accuracy is the most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worst happens when classes are imbalanced.

CHAPTER-5

IMPLEMENTATION

Implementation part is made using CSV file containing 9 different attributes with nearly 13321 records. House Prices are predicted using the data collected with Machine Learning algorithms like Linear regression, Decision Tree regression, Random Forest Regression and XGBoost Regressor. All these algorithms helps to predict the house prices. House Prices are predicted by implementing all three algorithms separately and are compared one with another.

5.1. Libraries Used

Python is increasingly being used as a scientific language. Matrix and vector manipulation are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high performance matrix computation capabilities.

Pip:

The pip command is a tool for installing and managing Python packages, such as those found in the Python Package Index. It's a replacement for easy installation. The easiest way to install the nfl* python modules and keep them up-to-date is with a Python-based package manager called pip.

pip install (module name)

NumPy:

NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open-source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas,

Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem. NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation. NumPy can be imported into the notebook using

import numpy as np.

Pandas:

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Pandas provides an in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels. Hence, with 2d tables, pandas are capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs. Pandas can be imported into Python using:

import pandas as pd.

Matplotlib:

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. Matplotlib comes with a wide variety of plots. Plots help to understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Matplotlib can be imported into Python using:

import matplotlib.pyplot as plt

Seaborn:

Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

Seaborn offers the functionalities like Dataset oriented API to determine relationship between variables, Automatic estimation and plotting of linear regression

plots. It supports high-level abstractions for multi-plot grids and Visualizing univariate and bivariate distribution and bivariate distribution. It can be imported into Python as

import seaborn as sns

Sklearn:

Scikit-learn is a free software machine library for Python programming language. It features various classification , regression and clustering algorithms including Linear regression, Decision Tree Regression, Random Forest Regression and XGBoost Regression. In our project we have used different features of sklearn library like:

from sklearn.preprocessing import LabelEncoder

In machine learning, we usually deal with datasets which contain multiple labels in one or more than one column. These labels can be in the form of words or numbers. To make the data understandable or in human readable form, the training data is often labeled in words.

Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important preprocessing step for the structured dataset in supervised learning.

Label encoding converts the data in machine readable form, but it assigns a unique number (starting from 0) to each class of data. This may lead to the generation of priority issues in training of data sets. A label with high value may be considered to have high priority than a label having lower value.

from sklearn.model_selection import train_test_split

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure

should not be used, such as when you have a small dataset and situations where additional configuration is required, such as when it is used for classification and the dataset is not balanced.

- **Train Dataset:** Used to fit the machine learning model.
- **Test Dataset:** Used to evaluate the fit machine learning model.



Fig.5.1. Training and test data set

from xgboost import XGBRegressor

XGBoost is an open-source library providing a high-performance implementation of gradient boosted decision trees. An underlying C++ codebase combined with a Python interface sitting on top makes for an extremely powerful yet easy to implement package. It becomes the go-to library for winning many Kaggle competitions. Its gradient boosting implementation is second to none and there's only more to come as the library continues to garner praise.

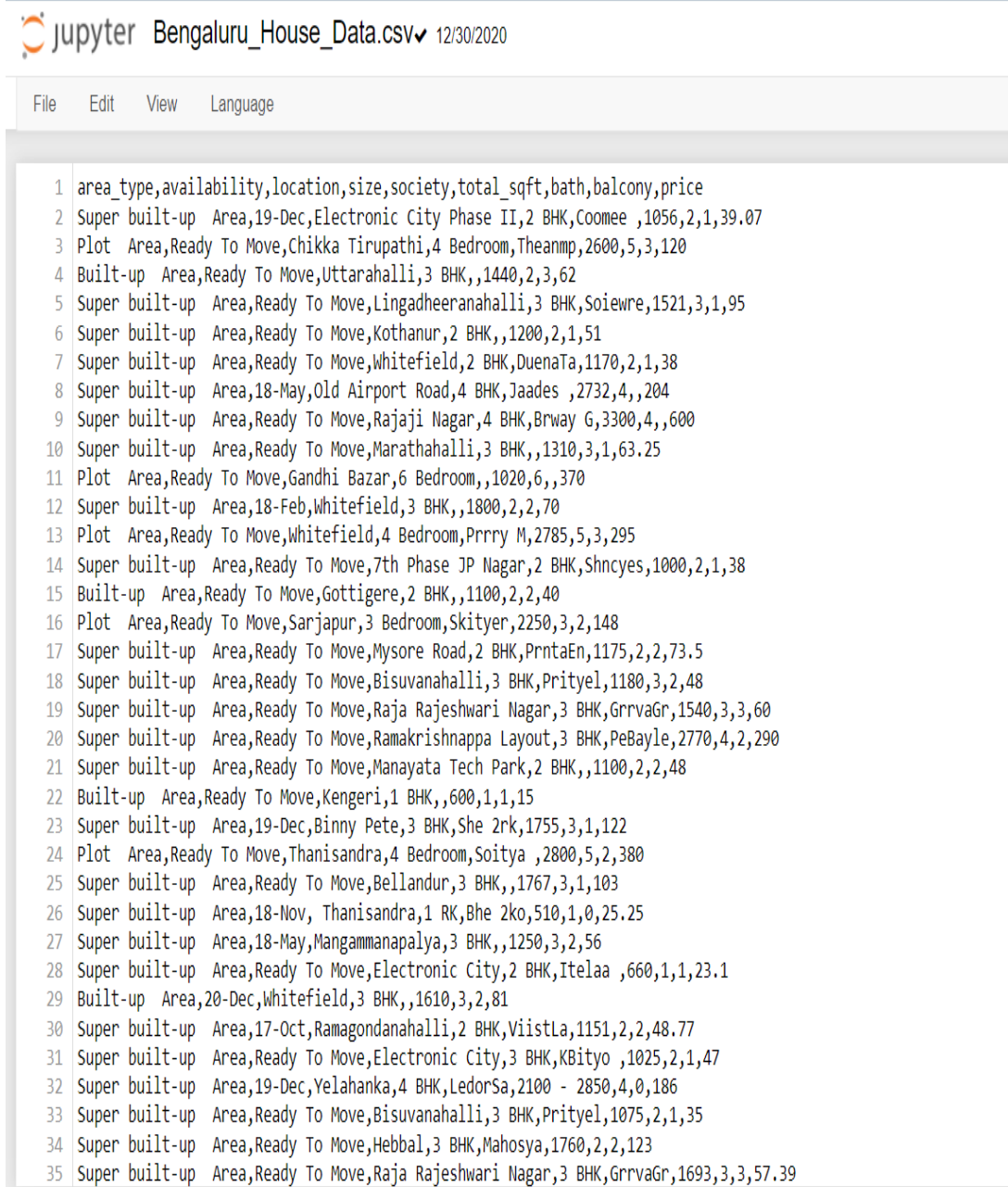
Boosting on the other hand, takes a more iterative approach. It's still technically an ensemble technique in that many models are combined together to perform the final one, but takes a more clever approach.

CSV file:

The dataset used in this project is a .CSV file.

In computing, a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the

name for this file format. CSV is a simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOffice Calc. Its data fields are most often separated, or delimited, by a comma.



```

1 area_type,availability,location,size,society,total_sqft,bath,balcony,price
2 Super built-up Area,19-Dec,Electronic City Phase II,2 BHK,Coomee ,1056,2,1,39.07
3 Plot Area,Ready To Move,Chikka Tirupathi,4 Bedroom,Theanmp,2600,5,3,120
4 Built-up Area,Ready To Move,Uttarahalli,3 BHK,,1440,2,3,62
5 Super built-up Area,Ready To Move,Lingadheeranahalli,3 BHK,Soiewre,1521,3,1,95
6 Super built-up Area,Ready To Move,Kothanur,2 BHK,,1200,2,1,51
7 Super built-up Area,Ready To Move,Whitefield,2 BHK,DuenaTa,1170,2,1,38
8 Super built-up Area,18-May,Old Airport Road,4 BHK,Jaades ,2732,4,,204
9 Super built-up Area,Ready To Move,Rajaji Nagar,4 BHK,Brway G,3300,4,,600
10 Super built-up Area,Ready To Move,Marathahalli,3 BHK,,1310,3,1,63.25
11 Plot Area,Ready To Move,Gandhi Bazar,6 Bedroom,,1020,6,,370
12 Super built-up Area,18-Feb,Whitefield,3 BHK,,1800,2,2,70
13 Plot Area,Ready To Move,Whitefield,4 Bedroom,Prrry M,2785,5,3,295
14 Super built-up Area,Ready To Move,7th Phase JP Nagar,2 BHK,Shncyes,1000,2,1,38
15 Built-up Area,Ready To Move,Gottigere,2 BHK,,1100,2,2,40
16 Plot Area,Ready To Move,Sarjapur,3 Bedroom,Skityer,2250,3,2,148
17 Super built-up Area,Ready To Move,Mysore Road,2 BHK,PrrtaEn,1175,2,2,73.5
18 Super built-up Area,Ready To Move,Bisuvanahalli,3 BHK,Prityel,1180,3,2,48
19 Super built-up Area,Ready To Move,Raja Rajeshwari Nagar,3 BHK,GrrvaGr,1540,3,3,60
20 Super built-up Area,Ready To Move,Ramakrishnappa Layout,3 BHK,PeBayle,2770,4,2,290
21 Super built-up Area,Ready To Move,Manayata Tech Park,2 BHK,,1100,2,2,48
22 Built-up Area,Ready To Move,Kengeri,1 BHK,,600,1,1,15
23 Super built-up Area,19-Dec,Binny Pete,3 BHK,She 2rk,1755,3,1,122
24 Plot Area,Ready To Move,Thanisandra,4 Bedroom,Soitya ,2800,5,2,380
25 Super built-up Area,Ready To Move,Bellandur,3 BHK,,1767,3,1,103
26 Super built-up Area,18-Nov, Thanisandra,1 RK,Bhe 2ko,510,1,0,25.25
27 Super built-up Area,18-May,Mangammanapalya,3 BHK,,1250,3,2,56
28 Super built-up Area,Ready To Move,Electronic City,2 BHK,Itelaa ,660,1,1,23.1
29 Built-up Area,20-Dec,Whitefield,3 BHK,,1610,3,2,81
30 Super built-up Area,17-Oct,Ramagondanahalli,2 BHK,ViistLa,1151,2,2,48.77
31 Super built-up Area,Ready To Move,Electronic City,3 BHK,KBityo ,1025,2,1,47
32 Super built-up Area,19-Dec,Yelahanka,4 BHK,LedorSa,2100 - 2850,4,0,186
33 Super built-up Area,Ready To Move,Bisuvanahalli,3 BHK,Prityel,1075,2,1,35
34 Super built-up Area,Ready To Move,Hebbal,3 BHK,Mahosya,1760,2,2,123
35 Super built-up Area,Ready To Move,Raja Rajeshwari Nagar,3 BHK,GrrvaGr,1693,3,3,57.39

```

Fig.5.2. Dataset with 9 different attributes

5.2.Implementation

5.2.1. Importing all the required Modules and Libraries

All the required libraries like Sklearn and Modules like Numpy, Pandas, Matplotlib, Seaborn are imported into the Jupyter notebook initially into the file created in the notebook.

After importing all the modules and libraries into the notebook, A csv file has to be loaded using Pandas into the notebook. The implementation of these will be as follows:

```
# Importing the necessary libraries

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

pd.options.display.max_columns = None
pd.set_option('display.max_rows', 500)
%matplotlib inline
matplotlib.rcParams["figure.figsize"] = (10,5)

from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.pipeline import FeatureUnion
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split

#Reading the data

housing = pd.read_csv("Bengaluru_House_Data.csv")
housing.head()
```

Fig.5.3. Importing of modules and libraries

5.2.2. Data Visualization

As it contains large amounts of data it is not possible to analyse with the human eye normally so the feature of Data Visualization helps to analyse the entire data. The relation between any two features can be only analysed with the Data Visualization technique.

This Visualization can be made in different forms by representing the data in the pictorial forms like graph, bar chart and many other forms.

Some Visualizations are made for the dataset that is collected for the Prediction of sales. They are as follows:

5.2.2.1. Descriptive Analyzation

It is important to know about the information of each and every attribute, such information can be easily predicted and is analysed as follows:

Pandas **head()** method is used to return top n (5 by default) rows of a data frame or series.

```
housing.head()
```

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

Fig.5.4. Pandas head method

The `info()` function is used to print a concise summary of a Data Frame. This method prints information about a Data Frame including the index dtype and column dtypes, non-null values and memory usage.

This method helps to provide a very small and important summary of the entire dataset so that it is easy to have an idea on the entire dataset that is present. It provides information about Count of the attribute, Type of the attribute along with the attribute name etc.

```
In [4]: #Checking the features in the dataset

housing.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13320 entries, 0 to 13319
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   area_type        13320 non-null  object
1   availability      13320 non-null  object
2   location         13319 non-null  object
3   size             13304 non-null  object
4   society          7818 non-null   object
5   total_sqft       13320 non-null  object
6   bath             13247 non-null  float64
7   balcony          12711 non-null  float64
8   price            13320 non-null  float64
dtypes: float64(3), object(6)
memory usage: 936.7+ KB
```

There are 13320 samples and 9 features. There are few features with missing values.

Fig.5.5. Pandas info() method

Pandas **describe()** is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. When this method is applied to any kind of dataset the output will be as follows:

```
In [29]: housing_clean["sqft"].describe()

Out[29]: count      1.330400e+04
         mean      1.911209e+03
         std       1.728725e+04
         min       1.000000e+00
         25%       1.100000e+03
         50%       1.276000e+03
         75%       1.680000e+03
         max       1.306800e+06
         Name: sqft, dtype: float64
```

Fig.5.6. pandas describe() method

Mean: Mean value of the entire column and in the dataset.

Min & Max: Minimum and Maximum values of the entire column.

5.2.2.2. Univariate Analysis

This kind of analysis which is Univariate helps to find and analyse all the information about a single attribute and Variable in the dataset. So that we can determine the uniformity and any kind of uneven nature of the variable can also be predetermined.

```
In [5]: sns.countplot(x="area_type", data = housing)
Out[5]: <AxesSubplot:xlabel='area_type', ylabel='count'>
```

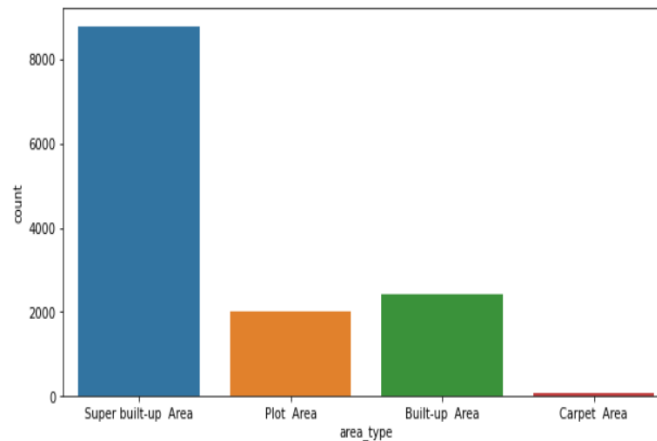


Fig.5.7. Visual Analysis of area_type

From the above analysis we can find the count of various area types in the distribution of the area_type. so that we can be made normalized to acquire more Uniformity which in order gives the more accuracy of the Model of any algorithm that is trained.

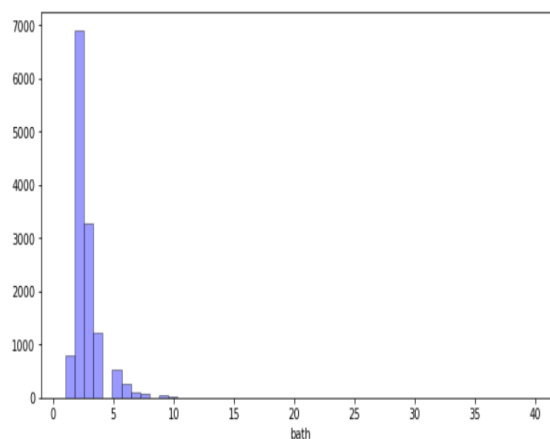
```
In [10]: housing["total_sqft"].value_counts()
```

```
Out[10]: 1200      843
          1100      221
          1500      205
          2400      196
          600       180
          ...
          3580        1
          2461        1
          1437        1
          2155        1
          4689        1
          Name: total_sqft, Length: 2117, dtype: int64
```

```
In [11]: sns.distplot(housing["bath"], hist=True, kde=False,
                      bins=50, color = 'blue',
                      hist_kws={'edgecolor':'black'})
```

c:\users\prathibha\pycharmprojects\hpm1\venv\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

```
Out[11]: <AxesSubplot:xlabel='bath'>
```



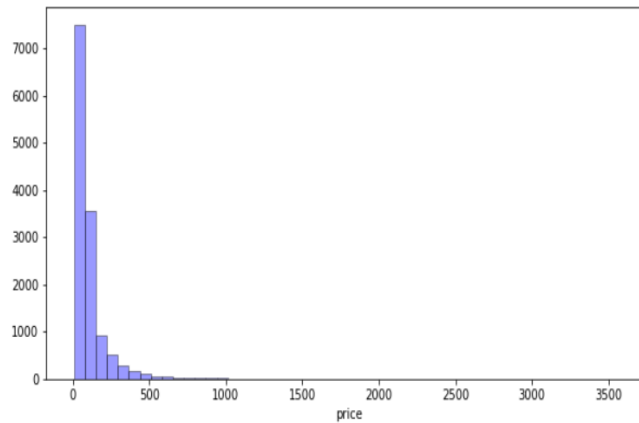
```
In [12]: housing["balcony"].value_counts()
```

```
Out[12]: 2.0      5113
          1.0      4897
          3.0      1672
          0.0      1029
          Name: balcony, dtype: int64
```

```
In [13]: sns.distplot(housing["price"], hist=True, kde=False,
                    bins=50, color = 'blue',
                    hist_kws={'edgecolor':'black'})
```

c:\users\prathibha\pycharmprojects\hpm1\venv\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

```
Out[13]: <AxesSubplot:xlabel='price'>
```



5.2.2.3. Corelation matrix

Correlation is an indication about the changes between two variables. We can plot correlation matrix to show which variable is having a high or low correlation in respect to another variable.

```
In [65]: housing_clean.corr()["price"].sort_values(ascending=False)
```

```
Out[65]: price          1.000000
sqft          0.820807
price_per_sqft 0.782903
bath          0.650978
bhk           0.634278
sqft_per_bhk  0.470972
balcony        0.295589
Name: price, dtype: float64
```

Fig.5.8. Correlation values

5.3. Feature Engineering

What is a feature and why do we need the engineering of it? Basically, all machine learning algorithms use some input data to create outputs. This input data comprise features, which are usually in the form of structured columns. Algorithms require

features with some specific characteristics to work properly. Here, the need for feature engineering arises. I think feature engineering efforts mainly have two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the performance of machine learning models.

one of the features of the dataset in which maximum number of values in that row are Zeros so they has to be normalised and are set to some value. So, mean of the entire column of data_visibilty is calculated and the value is set to that mean value.

This make all the Zeros of the column to particular value and accuracy of the model can be made more efficient.

5.4. Filling missing values and Label encoding

5.4.1. Filling missing values with Median and Mean

```
In [14]: housing.isnull().sum()
Out[14]: area_type      0
availability    0
location        1
size            16
society         5502
total_sqft      0
bath            73
balcony         609
price           0
dtype: int64

In [15]: housing_clean = housing.copy()

In [16]: housing_clean[pd.isnull(housing_clean["size"])]
Out[16]:
```

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
579	Plot Area	Immediate Possession	Sarjapur Road	NaN	Asiss B	1200 - 2400	NaN	NaN	34.185
1775	Plot Area	Immediate Possession	IVC Road	NaN	Orana N	2000 - 5634	NaN	NaN	124.000
2264	Plot Area	Immediate Possession	Banashankari	NaN	NaN	2400	NaN	NaN	460.000
2809	Plot Area	Immediate Possession	Sarjapur Road	NaN	AsdiaAr	1200 - 2400	NaN	NaN	28.785
2862	Plot Area	Immediate Possession	Devanahalli	NaN	Ajleyor	1500 - 2400	NaN	NaN	46.800

Fig.5.9. Filling missing values

If the column is removed, this may adversely affect the accuracy. So, this can be done by filling the columns of numeric values with Mean of the column and Column with categorical values are filled with the Mode of the column.

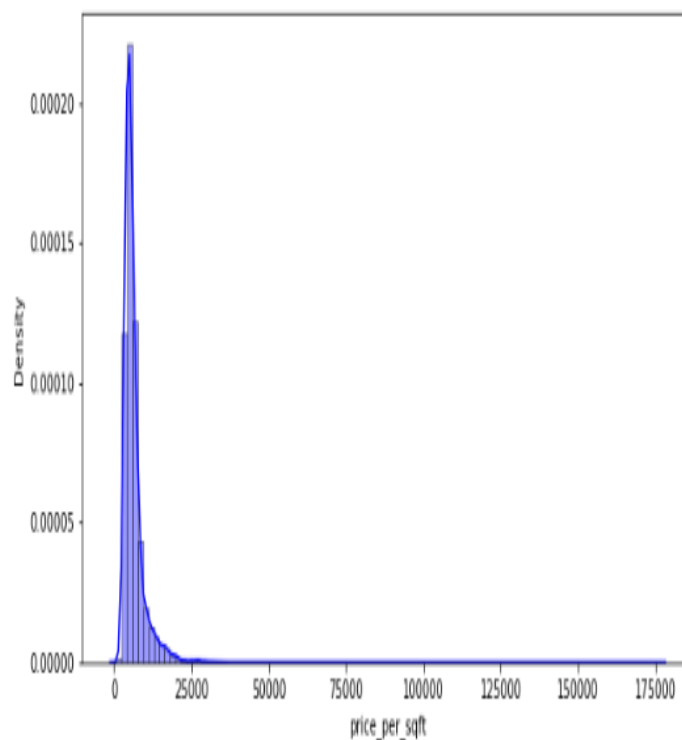
5.4.2 Outlier Analysis

Outliers can skew results, and anomalies in training data can impact overall model effectiveness. Outlier detection is a key tool in safeguarding data quality, as anomalous data and errors can be removed and analysed once identified. Outlier detection is an important part of each stage of the machine learning process.

```
In [152]: sns.distplot(housing_clean["price_per_sqft"], hist=True, kde=True,  
                      bins=100, color = 'blue',  
                      hist_kws={'edgecolor':'black'})
```

```
c:\users\prathibha\pycharmprojects\hpm1\venv\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a  
deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)
```

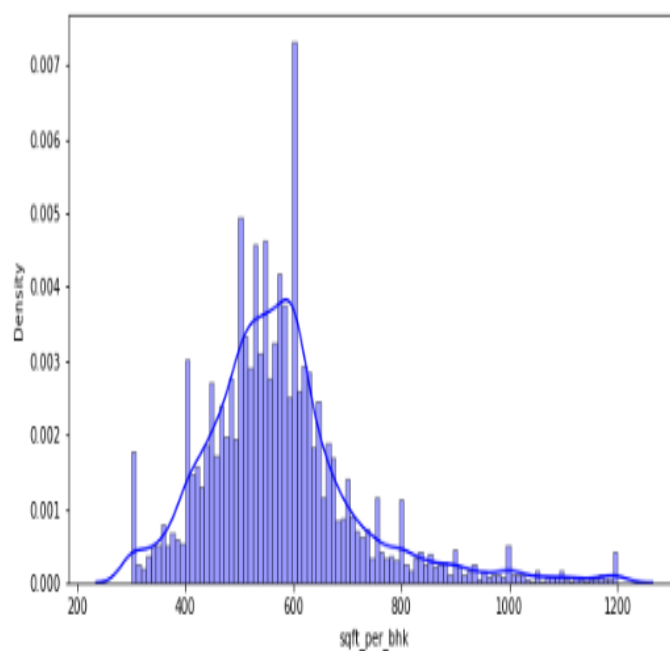
```
Out[152]: <AxesSubplot:xlabel='price_per_sqft', ylabel='Density'>
```



```
In [149]: sns.distplot(housing_clean["sqft_per_bhk"], hist=True, kde=True,  
                      bins=100, color = 'blue',  
                      hist_kws={'edgecolor':'black'})
```

```
c:\users\prathibha\pycharmprojects\hpm1\venv\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a  
deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)
```

```
Out[149]: <AxesSubplot:xlabel='sqft_per_bhk', ylabel='Density'>
```



5.4.3. Label encoding

Label encoding is done for all the columns with the categorical variables where all they are strings so they have to be converted into the numbers for all the columns.

```

In [69]: le1 = LabelEncoder()
housing_availability = le1.fit_transform(housing_clean.iloc[:,3])

In [70]: le2 = LabelEncoder()
housing_area_type = le2.fit_transform(housing_clean.iloc[:,4])

In [71]: le3 = LabelEncoder()
housing_location = le3.fit_transform(housing_clean.iloc[:,5])

In [72]: ohe1 = OneHotEncoder()
housing_availability = ohe1.fit_transform(housing_availability.reshape(-1,1))
housing_availability = pd.DataFrame(housing_availability.toarray(), columns=le1.classes_)

In [73]: ohe2 = OneHotEncoder()
housing_area_type = ohe2.fit_transform(housing_area_type.reshape(-1,1))
housing_area_type = pd.DataFrame(housing_area_type.toarray(), columns=le2.classes_)

In [74]: ohe3 = OneHotEncoder()
housing_location = ohe3.fit_transform(housing_location.reshape(-1,1))
housing_location = pd.DataFrame(housing_location.toarray(), columns=le3.classes_)

```

Fig.5.10. Label encoding of attributes

In the dataset collected for this project there are some attributes or columns which are having categorical attributes so they can be encoded with the labels by performing the process of Label Encoding.

5.4.4. Splitting the dataset

```

In [82]: lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
lin_reg.score(X_test, y_test)

```

Fig.5.11. Train and Test split of data

For all the Machine learning models to train with any algorithm of their choice the dataset has to be divided into two parts called Training dataset and Testing dataset.

Generally, the training dataset will be 80% of the entire dataset and 20% of the data as the Testing dataset.

Training dataset will be used to train the model and testing dataset will be used to find the accuracy of our predicted model. Performance evaluation can be made with the accuracy of the trained model with required algorithm.

Those splitting of the dataset to train and test splitting can be made using the command from the sklearn library as shown above.

X_train-Represents train dataset.

X_test-Represents test dataset.

5.5. Model prediction

In this prediction the entire data is trained with three different models in which each model provides different output values of different accuracies. All the models are compared and the conclusions are made.

5.5.1. Modelling with Linear Regression

Algorithm:

Step1: Load the dataset

Step2: Divide the dataset

Step3: Assign the linear regression algorithm to a variable.

Step4: Fit the training dataset using Linear regression algorithm

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.

```
In [82]:  lin_reg = LinearRegression()
          lin_reg.fit(X_train, y_train)
          lin_reg.score(X_test, y_test)
```

Fig.5.12. Fitting model with Linear regression.

reg-Variable of Linear regression algorithm

.fit()-fit method of sklearn

.predict()-predict method for predicting the values

X_train, y_train -Training data

X_test, y_test -Testing data

5.5.2. Modelling with Decision Tree Regression

Algorithm:

Step1: Load the dataset

Step2: Divide the dataset

Step3: Assign the Decision Tree regression algorithm to a variable.

Step4: Fit the training dataset using decision tree regression algorithm of the required degree. For this project the polynomial degree used is 4.

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.

5.5.3. Modelling with XGBoost Regression

Algorithm:

Step1: Load the dataset

Step2: Divide the dataset

Step3: Assign the XGBoost regressor to a variable.

Step4: Fit the training dataset using XGBoost Regression algorithm.

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.

5.6. Evaluation of models

5.6.1. Prediction of outputs

After training all the models the output values are loaded into the Excel files and can be viewed according to our requirements. The outputs of each algorithm will be as follows:

```
In [153]: housing_clean.sort_values(["price"], ascending=False)
```

Out[153]:

	area_type	availability	location	bath	balcony	price	bhk	sqft	price_per_sqft	sqft_per_bhk
12443	Plot Area	Ready To Move	Other	8	4	2800.0	4	4350.0	59770.114943	1087.5
6421	Plot Area	Soon to be Vacated	Bommenahalli	3	2	2250.0	4	2940.0	76530.612245	735.0
8398	Super built-up Area	Ready To Move	Bannerghatta Road	4	5	1400.0	5	2500.0	56000.000000	500.0
9535	Plot Area	Ready To Move	Indira Nagar	5	4	1250.0	4	2400.0	52083.333333	600.0
1299	Plot Area	Ready To Move	Chamrajpet	7	1	1200.0	9	4050.0	29629.629630	450.0
...
5410	Super built-up Area	Ready To Move	Attibele	1	1	10.0	1	400.0	2500.000000	400.0
11091	Built-up Area	Ready To Move	Attibele	1	1	10.0	1	410.0	2439.024390	410.0
7482	Super built-up Area	Ready To Move	Other	2	1	10.0	1	470.0	2127.659574	470.0
12579	Super built-up Area	Ready To Move	Chandapura	1	1	10.0	1	410.0	2439.024390	410.0
8594	Built-up Area	Ready To Move	Chandapura	1	1	9.0	1	450.0	2000.000000	450.0

5.7. Dimensionality Reduction

Dataset consists of 9 different attributes where 8 are independent attributes in which Dimensionality reduction is the process of removing few attributes from those which improve the performance of the model.

By removing two attributes the performance of the model is evaluated and the process of dimensionality reduction is performed and then we perform the accuracy of the model using score method for various algorithms.

```
In [82]: lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
lin_reg.score(X_test, y_test)
```

Out[82]: 0.7902192001533112

```
In [85]: dt_reg = DecisionTreeRegressor()
dt_reg.fit(X_train, y_train)
dt_reg.score(X_test, y_test)
```

Out[85]: 0.7391232985331604

```
In [86]: rf_reg = RandomForestRegressor()
rf_reg.fit(X_train, y_train)
rf_reg.score(X_test, y_test)
```

Out[86]: 0.8141836883698912

```
In [89]: xgb_reg = XGBRegressor()
xgb_reg.fit(X_train, y_train)
xgb_reg.score(X_test, y_test)
```

c:\users\prathibha\pycharmprojects\hpm1\venv\lib\site-packages\xgboost\data.py:262: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.
elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):

Out[89]: 0.8291254192651423

```
In [90]: def find_best_model_using_gridsearchcv(X, y):
    algos = {
        'linear_regression': {
            'model': LinearRegression(),
            'params': {
                'normalize': [True, False]
            }
        },
        'lasso': {
            'model': Lasso(),
            'params': {
                'alpha': [0.1, 0.3, 0.5, 0.7, 0.9],
                'selection': ['random', 'cyclic']
            }
        },
        'ridge': {
            'model': Ridge(),
            'params': {
                'alpha': [0.1, 0.3, 0.5, 0.7, 0.9]
            }
        },
        'random_forest': {
            'model': RandomForestRegressor(),
            'params': {
                'n_estimators': [10, 20, 500], 'max_depth': [2, 4, 6, 8],
            }
        },
        'xgboost': {
            'model': XGBRegressor(),
            'params': {
                'n_estimators': [100, 200, 300],
            }
        }
    }
    scores = []
    cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
    for algo_name, config in algos.items():
        gs = GridSearchCV(config['model'], config['params'], cv=cv, return_train_score=False)
        gs.fit(X, y)
        scores.append({
            'model': algo_name,
            'best_score': gs.best_score_,
            'best_params': gs.best_params_
        })
    return pd.DataFrame(scores, columns=['model', 'best_score', 'best_params'])
#find_best_model_using_gridsearchcv(X, y)

In [91]: xgb_reg = XGBRegressor()
xgb_reg.fit(X, y)
xgb_reg.score(X, y)

Out[91]: 0.9131660566803912
```

Accuracy of Linear Regression:0.7902192001533112

Accuracy of Decision Tree Regression:0.7391232985331604

Accuracy of Random Forest Regression:0.8141836883698912

Accuracy of XGBoostRegressor:0.9131660566803912

CHAPTER-6

TESTING

6.1. INTRODUCTION

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

Testing is a process of executing a program with the intent of finding an error

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.

The first approach is what is known as Black box testing and the second approach is White box testing. We apply white box testing techniques to ascertain the functionalities top-down and then we use black box testing techniques to demonstrate that everything runs as expected.

6.2. Black-Box Testing

This technique of testing is done without any knowledge of the interior workings of the application. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining the outputs without knowing how and where the inputs are worked upon.

- Well suited and efficient for large code segments
- Code access is not required
- Clearly separates user's perspectives from the developer's perspective through visibly defined roles.

6.3. White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. It is also called “glass testing” or “open-box testing”. In order to perform white box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to look inside the source code and find out which part of the code is working inappropriately.

In this, the test cases are generated on the logic of each module. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent modules have been executed.
- Execute all logical decisions and loops.
- Execute through proper plots and curves.

6.4. Performance Evaluation

Score method: It is a kind of method used to evaluate the performance of the model. Performance evaluation is made for this project using Score method of the sklearn library of Python. The score method is applied for all three algorithms as follows:

```
lin_reg = LinearRegression()  
lin_reg.fit(X_train, y_train)  
lin_reg.score(X_test, y_test)
```

0.7902192001533112

```
ridge_reg = Ridge(alpha = 0.1)  
ridge_reg.fit(X_train, y_train)  
ridge_reg.score(X_test, y_test)
```

0.7901996166492571

```
lasso_reg = Lasso(alpha = 0.1)  
lasso_reg.fit(X_train, y_train)  
lasso_reg.score(X_test, y_test)
```

0.7621611341683487

```
dt_reg = DecisionTreeRegressor()  
dt_reg.fit(X_train, y_train)  
dt_reg.score(X_test, y_test)
```

0.7586794042976315

```
rf_reg = RandomForestRegressor()  
rf_reg.fit(X_train, y_train)  
rf_reg.score(X_test, y_test)
```

0.80708490532443

```
xgb_reg = XGBRegressor()  
xgb_reg.fit(X_train, y_train)  
xgb_reg.score(X_test, y_test)
```

0.8291254192651423

```
xgb_reg = XGBRegressor()  
xgb_reg.fit(X, y)  
xgb_reg.score(X, y)
```

0.9131660566803912

Accuracy of the models of the algorithms are as follows:

Linear regression accuracy: 79.02%

Decision Tree regression accuracy: 75.86%

Random Forest regression accuracy: 80.70%

XGBoost regression accuracy: 91.31%

CONCLUSION

In this project, we have used machine learning algorithms to predict the house prices. We have performed step by step procedure to analyse the dataset and found the correlation between the parameters. The manually collected Real-time Dataset has been collected which contains 13321 entries and independent variables. We analyse and pre-process this dataset before performing Exploratory Data Analysis. This analysed feature set was given as an input to machine learning algorithms and calculated the performance of each model to compare based on Accuracy score.

We found that XGBOOST Regressor fits our dataset and gives the highest accuracy of 91.31%. Decision Tree gives the least accuracy of 56.02%. Support Vector Regression gives an accuracy of 62.81%. Thus, we conclude that we implemented regression techniques to check how well an algorithm fits to given problem statement of House price prediction.

REFERENCES

Journals:

- [1] Maharshi Modi, Ayush Sharma, Dr. P. Madhavan “Applied Research on House Price Prediction Using Diverse Machine Learning Techniques”, International Journal of Scientific & Technology Research Volume 9, Issue 04, April 2020.
- [2] 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) ISSN: 2278-3075, Volume-8 Issue-9, July 2019.
- [3] Dr. M. Thamarai, Dr. S P. Malarvizhi “House Price Prediction Modeling Using Machine Learning”, I.J. Information Engineering and Electronic Business, 2020, 2, 15-20.
- [4] Neelam Shinde, Kiran Gawande “Valuation of House Prices Using Predictive Technique”, International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835 Volume- 5, Issue-6, Jun.-2018.
- [5] Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair, Computer Engineering Department, KJ Somaiya College of Engineering, Mumbai “House Price Prediction Using Machine Learning and Neural Networks”, 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT).
- [6] Hong Zhao, Rong-Qiu Chen, Wei Xu, Da-Ying Li “A SVR based forecasting approach for real estate price prediction”, 2009 International Conference on Machine Learning and Cybernetics.
- [7] Uysal, İ., Güvenir, H. A. “An overview of regression techniques for knowledge discovery”, The Knowledge Engineering Review, Vol. 14:4, 1999, 319±340 (KER 14404) Printed in the United Kingdom.
- [8] Bhuriya, Dinesh, et al. “Stock market predication using a linear regression.”, Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of. Vol. 2. IEEE 2017.

- [9] Limsombunchai “House price prediction: hedonic price model vs. artificial neural network.”, New Zealand Agricultural and Resource Economics Society Conference 2004.
- [10] S. C. Bourassa, E. Cantoni, and M. Hoesli, “Predicting house prices with spatial dependence: a comparison of alternative methods,” *Journal of Real Estate Research*, vol. 32, no. 2, pp. 139–160, 2010.
- [11] Li, Li, and Kai-Hsuan Chu “Prediction of real estate price variation based on economic parameters”, *Applied System Innovation (ICASI)*, 2017 International Conference in IEEE, 2017.
- [12] Pedregosa, Fabian, et al. “Scikit-learn: Machine learning in Python”, *Journal of machine learning research* 12. oct (2011): 2825-2830.