varunr1

by Varunr1 Varunr1

Submission date: 15-May-2021 09:48AM (UTC+0530)
Submission ID: 1586493972
File name: FINAL_REPORT.pdf (4.38M)
Word count: 10023
Character count: 50181

PREDICTION OF PROPERTY PRICES

18
A PROJECT REPORT

Submitted by

VARUN NIGAM [Reg No: RA1711003030357]
PRAJWAL ATHREY [Reg No: RA1711003030346]

6
Under the guidance of

Dr. MANIKANDAN R.
(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2021

SRM UNIVERSITY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**PREDICTION OF PROPERTY PRICES**" is the bonafide work of " **VARUN NIGAM [Reg No: RA1711003030357], PRAJWAL ATHREY [Reg No: RA1711003030346]**", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. MANIKANDAN R.
GUIDE
Assistant Professor
Dept. of Computer Science & Engineering

SIGNATURE

Dr. R. P. MAHAPATRA, Ph.D
HEAD OF THE DEPARTMENT
Dept. of COMPUTER SCIENCE & ENGINEERING

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Various predicting models for evaluating the price of any property in cities like Bengaluru, which is India's IT hub, is a challenging task. The price of housing is not only related with buyer and seller, but also gives us an idea of the economic situation of the state. In this project we are going to create a website where the user would input certain details as per the requirements and get the predicted price of the property. The price of a property in any city depends on a number of inter-related factors. Some of the key factors which affect the price of any property include the area, location, number of bedrooms, bathrooms, balcony and the amenities around. In this research an analytical study has been done in consideration to the data set used which has nine features. In this research, an attempt has been made to develop a prediction model for calculating the price of any property based on the factors affecting it. Some regression techniques such as Linear Regression, Lasso Regression and Decision Tree Regression would be used to develop a prediction model. The best performing model would be taken into account based on the analysis of the errors and the scored obtained by these models. Our attempt would be to create a predictive structure for calculating the price of the property based on various factors.

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to our guide, Dr. MANIKANDAN R. for his valuable guidance, consistent encouragement, personal caring, timely help and providing us with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to us for completing this research work.

VARUN NIGAM

PRAJWAL ATHREY

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ANN	Artificial Neural Network
BHK	Bedroom Hall Kitchen
CV	Cross Validation
CSV	Comma Seperated Values
CSI	Complete Structural Identification
pr	Probability
LASSO	Least Absolute Shrinkage and Selection
MSE	Mean Square Error
PCA	Principal Component Analysis
SVM	Support Vector Machine
SI	Structural Identification

LIST OF SYMBOLS

$b0$	y-intercept
bp	Slope of each predictor
xi	Predictor Variable
yi	Dependent factor
λ	Penalty Term
θ	Coefficient Shrinkage
α	Evaluation Factor
$\{(1-pr)\}$	probability of false event

CHAPTER 1

INTRODUCTION

1.1 Project Overview

The Project describes the process of creating a Real Estate Prediction Model. A model is built using Sklearn and Linear Regression techniques and the dataset to be used is bangalore housing price dataset. In the second component of the project, we will write a Python Flask Server program that uses saved model to serve http requests. In the third component,a website will be built using Html, CSS and Javascript that will allow the user to enter his/her details like square feet area , number of bedrooms , number of balconies, the locality etc and a call is made to the the python flask Server which will retrieve the predicted price. During the process of model creation,a wide variety of data science concepts have been used such as Data Loading, Data Cleaning, Outlier Detection and Removal, GridSearch cv for hyperparameter tuning, Feature Engineering, Dimensionality Reduction, K Fold Cross Validation etc.

1.2 Real Estate Market

Investment in property has become more popular in recent times. The Real Estate (Putatunda (2019)) is one of the fastest growing industries but at the same time there are various factors which buyers are not aware of, making it less transparent. There are various parameters, on which the property price depends, like the area, locality, available amenities, number of bedrooms, balcony etc.Other factors also include accessibility to public transport like metro, connectivity to national highways, schools, expressways, and health facilities around. Prediction of property price might become tricky when done manually. Also, the price of any property should not be considered based on national trends because the value tends to change from state to state and even in neighbouring cities of the same state. Sometimes, the Market values tend to rise or fall based

on a particular situation, for example in case of natural disaster, the real estate prices may increase to 3 folds the original price and home buyers may find it tough to get a home of their choice. Hence many different types of prediction structures are developed for property prices. The aim of our system is to develop a website where the user can get the price of any property based on the inputs made as per their requirements. To build a model which predicts the price of any property based on several factors, various regression techniques are used. Various regression models are taken in account like Linear Regression, Lasso and Ridge regression,Decision tree regression and Random Forest regression. Based on the accuracy of the models and the percentage error, a comparative study is done and the best performing model is taken for further evaluation. After getting the best performing model, we can use it for estimating the property prices. Our data set consists of various features like availability, area, number of bedrooms, balcony, society, area type and price.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Real Estate is one of the fastest growing industry and at the same time,it is the least transparent one.The market and demand for housing is growing every year due to the increase in population and wide migration of people from villages to cities or from one city to other city for various purpose.The growing unaffordability in real estate is one of the major problems in metropolitan cities around the globe.Property value tend to change over time but factors like recession and natural disasters can affect the price.Technology has become more dependent and we can get accurate predictions by using various techniques, for these applications the researchers have proposed various machine learning techniques.

2.2 Literature Review

There are various methods and techniques available for the prediction of property prices. Some researchers have used Random Forest method for the prediction. The Random Forest method gave a good accuracy with low percentage error.The researchers also used Logistic Regression,Support Vector Regression, Lasso Regression and Decision Tree Regression. When these algorithms were used together then the decision tree gave good results while lasso regression was not successful.A lot of researchers have used Artificial Neural Network (ANN). In one of the Research works, the author compared the ANN model and hedonic model (Limsombunchai (2004)). Hedonic model calculates price that are based on internal and external characteristics. The hedonic model makes use of regression techniques and includes various parameters such as area, number of bedrooms etc. Black Box method is used for neural network model

training.Apart from these, researchers have also implemented Gradient Boosting Regression and Elastic Net Regression ,where the Gradient boosting Regression showed the best performance score while elastic net showed the lowest performance. Many other regression techniques like Principle Component Analysis,Polynomial Regression and Support Vector Machine were used.The Regression tree delivered a good result while Polynomial regression resulted in comparatively lower errors. Simultaneously Principal Component Analysis (PCA) and support Vector machine gave good accuracy results.

2.3 Research Gap

Though many regression techniques gave good prediction results but there were also drawbacks associated with these techniques.In one of the research works, the researchers used a very small dataset and this led to poor accuracy, hence, the accuracy could have been increased if large datasets were implemented.There was also problem of overfitting, (Shinde and Gawande (2018)). The problem of overfitting could have been reduced if trained with Support Vector Machine (SVM) with higher accuracy.In Some of the research works, the final predicted graph had presence of outliers due to noise in the dataset, The outliers could have been detected and removed using Outlier detection algorithms which were not implemented. Some researchers used one factor square feet area to estimate their prediction while the property price of any property depends on other factors like location, number of bedrooms, proximity to school, nearest metro etc. There were also researchers who used neural networks in their prediction but the neural network used in the implementation did not provide satisfactory results and this resulted in poor performance. In business context, the application can involve direct use of the appropriate algorithm for the prediction purpose by taking some inputs from the user and giving most justified value without taking price input from user and preventing any exceptions from occurring in the system.

To sum up some of the given gaps that can be used for further development of the Real Estate price prediction:

- Real estate market depends on various features and we cannot get proper idea of the value of any property by considering one or two features. Other features also need to be taken care of.

- If the model is prone to over-fitting then some techniques like outlier removal can be used to remove the unwanted noise from the data which could interfere in the predicted value.

- Systems should have an easy, smooth and intractable interface for the users to ease the task of understanding the predictions.

Table 2.1: Overview of Survey

PAPER	AUTHOR	DESCRIPTION	FINDINGS
ELECTRONIC GOVERNANCE OF HOUSING (2019) Lydia et al. (2019)	E.Laxmi Lydia, Gogineni Hima Bindu.	Only Linear Regression for Boston Dataset.Mean Square Error calculated to 30.4187%.	Presence of Outliers,due to presence of noise in dataset.Functions in few situations to estimate Boston City information.
HOUSING PRICE PREDICTION (2019) Madhuri et al. (2019)	CH.Raga Madhuri, Anuradha G, M.Vani Pujitha	Multiple-Linear model,Lasso,Ridge, Gradient Boosting model.Gradient Boosting[Highest score 0.9177],Least Of Elastic Net.	Mainly comparison of various algorithms, price calculated using one factor only-Square feet area.
HOUSE PRICE PREDICTION, MELBOURNE CITY (2018) Phan (2018)	Danh Phan	Polynomial Regression,Support Vector Machine, Regression Tree and Neural Networks used.Regression Tree delivers as good result as Linear Regression.	Neural Networks seems not to work effectively with this dataset and there is overfitting issue.

CHAPTER 3

METHODOLOGY

The methodology consists of detailed description of the framework that is used in the project. It consists of a checklist that needs to be covered in order to achieve the objectives. We have taken various data mining, machine learning and web based concepts for achieving the goal. The best performing model would be used for further evaluation and it would be connected to the next component which would deal with the interface part.

3.1 PREDICTING MODEL

In this chapter the methods and the various models and predicting techniques used will be discussed. The first component of the system is a predicting model which would make use of the various inputs from the user and using data mining and machine learning models, a predicted value of the property would be obtained.

3.1.1 Data Collection And Understanding

The quality and quantity of the data which is extracted from different sources play an important role in depicting how accurately the model will work. The data collected should be analysed and if needed the values should be converted to the form in which we want to feed the data to our models. The dataset used in the project is obtained from the open source repository named Kaggle.This dataset of Bangalore house price consists of around 13,500 records and 9 features.

These features are inclusive of:

- Area Type: It tells whether the area is “Built-up”, “Super built-up”, “Carpet Area” or the “Plot Area”.
- Location: It tells where the property is located in Bangalore.

- Size: This is mentioned in Bedroom Hall Kitchen (BHK).
- Availability: By when the property will be ready.
- Bath: Number of bathrooms present in the property.
- Balcony: Number of balconies present in the property.
- Total Sqft: Area of property in square feet.
- Society: Name of society which the property is a part.
- Price: Value of the property.

The cost price of the property is dependent on these independent variables or features. Some parameters having numerical values in floating form where later converted to appropriate values which were used in the prediction.

3.1.2 Data Preparation

In this stage the data is evaluated into the form as required for the predictions. In this process the raw data is taken and then filtered into the understandable form. It involves finding out the redundant values, missing values and values which are not in appropriate form. Randomization of the data is done which eliminates the effects of the order in which our data is collected or in which the data is prepared. The entire dataset is checked for the NaN values and these values are either removed or replaced with mean, median or mode of the data. This helps in making the data uniform. The help of visualization can also be taken to understand relationship between the various factors and performing exploratory analysis.
The steps carried out in this preparation or pre-processing stage are stated below:

- First, convert the categorical values to numerical form to fit in regression models.
- Replacing null values with suitable alternatives values like mean or median.
- Scaling of the data is done.
- Data is split into training-testing set.

The pre-processing performed for each feature is depicted below:

- The feature of society has been dropped as it does not have much influence in model and also it has a large percentage of null values present.

- There are around 1287 unique locations and one location is missing in the records. These locations have been brought under one category as others as these unique values did not have much impact.
- There are around 73 null values present in the number of bathroom. These values have been replaced with the mode which came out to be 2.

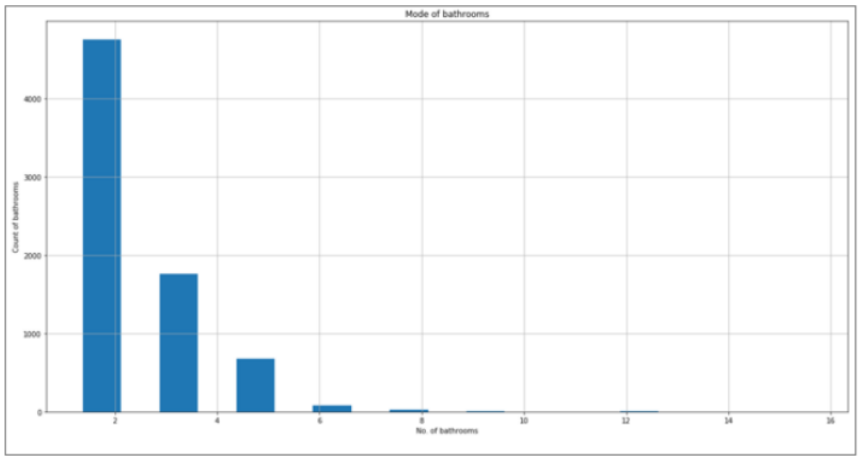


Figure 3.1: Null Removal with Mode

- In the total square feet record, values were present in acres, square yards, perch, etc. These were all converted in square feet and the values given in range were replaced by the mean value of upper and lower limit.
- The feature of size is given in bedroom, BHK and RK. A new column is created named BHK using the numerical part of the value in size column so that size can be excluded.
- Outlier detection was also carried out to remove noise from the data for every location. Outliers can be explained as some experimental defects or errors which are present in the data. The training data consists of some observations which are different from others in some or the other way. In simple terms, an outlier is a point which deviates from an overall series or pattern in a given sample. A graph has been plotted between total square feet and price. Z-score method has been used which is signed standard deviation by which mean of a data point is above or below the mean of what is observed.

From the below scatter-plot we can clearly see the presence of outliers in the data which are treated as noise and eliminated from the data. The green symbols represent 3 BHK property and the blue symbols represents the 2 BHK property.

In the graph, we can see the presence of 3 BHK properties in the area range of two BHK. We have treated such properties as outliers because practically we do not find properties with these numbers of rooms in small area. The number of rooms also depends on the area of the property. The properties which were present outside the range of the z-score method used were treated as the outliers.

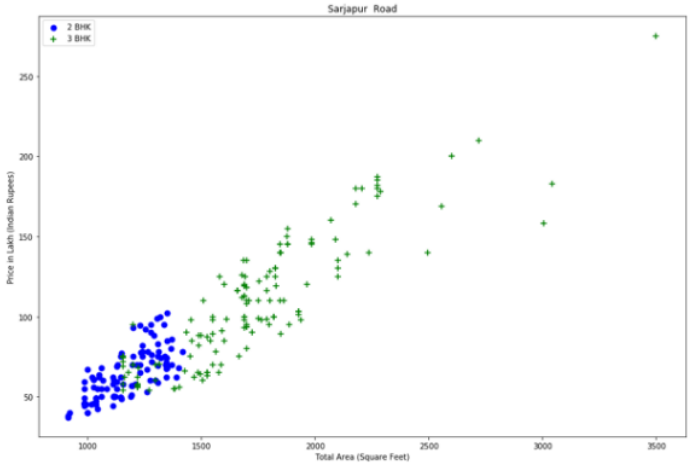


Figure 3.2: Before Outlier Removal

In the above scatter plot we can see that some properties of 3 BHK are removed from the 2 BHK cluster which were considered as the outliers. Some of the properties are still in 2 BHK cluster because their price range was in the limit which can be accepted.

3.1.3 Data Analysis

In this stage, we analyse the data try to find out the relation between the features before applying any predicting model. For plotting the graphs we make use of the Matplotlib module of python. The characteristic of the data is determined and the study of the features is carried out. Sometimes the outlier detection and removal process can be carried out in this stage as well. When the manipulation of the data takes place, we need to make some alterations in the data as per the use. The data analysis tools make the task easy to analyse the relation and manipulate the data accordingly. We have made use of Python for the analysis portion. These tools help to identify the trends and the patterns in the data which is used in interpretation.

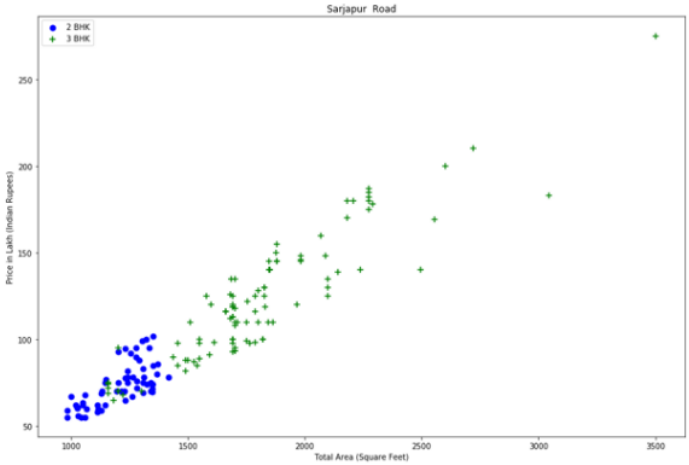


Figure 3.3: After Outlier Removal

In the above graph we have plotted the frequency of properties based on the square feet

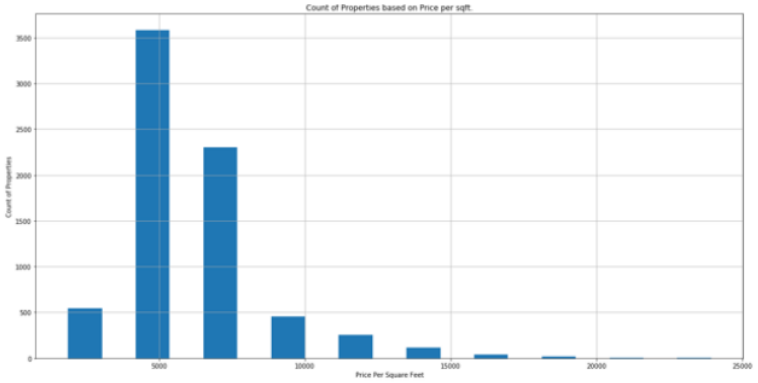


Figure 3.4: Cost of Properties based on sqft price

area. The properties with price per square feet around 5000 were having the highest frequency. There were some properties which were having very high values like 20000 or even 25000 which are practically not possible and are very rare to be observed. Hence these values can be removed treating them like outliers. With the help of these analysing tools it becomes manageable to determine some ambiguity in the data and hence work to remove such occurrences.

Similar methodology is being carried out of the number of bathrooms. The mode is calculated and where ever the value is missing or is ambiguous is replaced by this mode value. In some cases, the value of bathrooms came out to be greater than 10 or even 15.

In reality it is not possible to have such a large number of bathrooms in any property or it is very rare. Hence we have considered that the number of bathrooms can be maximum 2 greater than the number of bedrooms. Therefore, any value exceeding this range would be considered as an outlier. This is because the number of bathrooms being greater than the bedrooms, by a very large margin is very rarely observed.

3.1.4 Choosing a Model

Different types of algorithms are used for different operations. In the process of model selection, we choose one model from the candidate list of many models based on their performance or adaptability according to the data. The behaviour of different models is different based on the data provided. We usually don't know which model will be fit for the data beforehand. Hence we evaluate and fit different models on our problem. Generally the data is split into the train, test and validation set. Then we fit our various models in the train set and then evaluate them on the validation set and choose accordingly. Finally the performance is evaluated of the final model on the test data set. The model which is selected is used for further evaluation and is connected with the UI part of the system.

3.1.5 Applications of Models

One we get the insights and understanding of data, which has been cleaned and analysed, we proceed with applying various algorithms which fits our data and the best performing model is taken for further consideration. These algorithms are implemented with the help of the Sklearn or the SciKit-Learn library in python. The use of Pandas module is also done. It helps in reading and manipulating the data. Most of the data is in the Comma Seperated Values (CSV) form which is less understandable hence making it difficult to directly use that data in the model. It converts the data into human readable table format. This makes it easy to interpret the numerical values which can be used for the computational purpose. Numpy, which is a library in python, is also used for the manipulation of the tables which contains the numerical data. It has various functions which help in working with matrices and algebra.It includes multidimensional array

and a group of routines with which these arrays are processed. Using all these tools we would be implementing the algorithms for prediction of the property prices. Following algorithms are implemented:

1- Linear Regression: Linear Regression, (Ghosalkar and Dhage (2018)) is one of the conventional algorithms of machine learning. Here relationship is established between independent and dependent variable. The model is called Simple Linear Regression if there is one feature and Multiple Linear Regression when there are more predictor features. The main idea is to obtain the best fitting line or the line of best fit for our data in which the error margin is as low as possible.

In Simple Linear Regression we have two continuous variables. One is predictor or independent variable and other is dependent variable. It has an equation of the from: y=v+rx, where y is the dependent variable, x is explanatory variable, r is the slope of the line and v is the intercept. Simple Linear Regression can be used to determine how strong the relationship exists between two features. It is also used to determine the value of any dependent feature based on any independent feature at a given point. This algorithm assumes certain factors for our dataset. The margin of error for the predicted outcome does not change to a large extent which respect to the independent features. This means that the variance is homogeneous in nature.

The observations are collected with the help of some valid statistical methods by preventing any underlying relationship between features.

Normal distribution is usually followed by the data. As the name suggest, the relation between the dependent and independent features is linear which means the line of best fit is a straight line instead of a curve which passes through as may points of data following a given pattern. Linear regression makes use of the MSE, mean square error value to determine the error margin of the model. In this we measure the distance of observed value and predicted value from each of the independent feature. Then we square each of these distance values. After this step, mean of the values of distance is calculated.

In Multiple Linear Regression , (Manasa et al. (2020)) there are several explanatory features to depict the result of any outcome. This means that it is a type of linear regression where two or more independent features establish a relationship with a dependent feature. The relationship can be linear or non-linear based on the relationship. Here the line of best fit passes through a multi-dimensional area of the given data points. This is widely used in financial and economical related features hence suitable for our property prediction. It can also be implemented to forecast or predict the impact of variations or changes. The equation is in the form of: yi=b0+b1.xi1+b2.xi2+. . . .+ bp where xi is predictor variable, yi is dependent variable b0=y-intercept, bp =slope of each predictor. This technique helps us to determine the variation in our model and how each independent feature contributed in the evaluation of the variance. There should not be a high correlation between the independent features. High correlation means one feature can be used to determine other feature.

When multicollinearity is present in the independent features then some problems might arise to determine which variable contributed to the value of variance in dependent feature. This model assumes that the margin of error is constant at each point of linear system. When the analysis is done then the residuals should be plotted against the predicted result to make sure that the points are evenly distributed across all independent features. The model also demands that the observations should not depend on other observed values. When the residuals are distributed normally then a condition of multivariate normality arises.

The linear and non-linear models determine the outcome using two or more features. The non-linear one is complicated to execute as it is based on the assumptions derived from the errors obtained.

2- Lasso Regression: Lasso regression, (Lu et al. (2017)) also known as the Least Absolute Shrinkage and Selection OperatorLeast Absolute Shrinkage and Selection (LASSO) is a form of LR technique including the regularization function. In this model the absolute value summation of the magnitude of coefficients is taken into the consideration. It is basically a form of Linear Regression which makes implementation of the shrinkage technique. The concept of shrinkage implies that the value of any data point is shrunk to any central node which is generally the mean value.

This regression technique performs the L1 regularization method which involves a penalty term which is equivalent to the absolute value of coefficients' magnitude. When we implement the regularization then we can overcome the problem of over fitting. The accuracy of the model is compromised if we overlook the over fitting of the data in which the training data is more trained and this results in model giving poor results when served with data other than the training set.

A loss function is taken into account while fitting the model which is known as the sum of squares. The aim is to reduce the loss function to the maximum extent by choosing such coefficients in the equation. If the coefficients are not chosen properly then some unwanted data might also get involved in the training data set. When cases where there wrongly chosen coefficients arise then we try to shrink or regularize these values as close to zero. This type of algorithm is mainly used when we observe huge multicollinearity in our features, which means there is large correlation between the predictor features where one factor can be used to estimate the value of other. This value of correlation is estimated with the help of the correlation coefficient.

When this coefficient has value equal to +1 then we say that the features have a strong and positive relation. When it has a value equal to -1 then we say that the predictors have a strong but negative relation. When this value is equal to zero then we can say that the predictors have no relation between them.

Lasso Regression has the advantage that it has a very strong in-built capability of the selection of features which becomes very helpful in several situations. At the same time if the relationship between the predictor and the target feature is not linear in nature then it might become complicated to implement this model in a non-linear relationship. Also this type of regularization can cause the data getting sparse or spread out which might also lead to loss of some coefficients. This technique works if the data has been scaled beforehand with the help of various scaling and standardizing techniques.
The equation for this model is given below:

Residual Sum Of Squares + λ * (Total Sum of the magnitude of coefficients absolute value). The λ is the value of shrinkage. When this value is equal to 0 then it indicates that it is almost equal to the linear regression in which all factors have been considered and residual sum of squares would be used to develop the predictive system. When this value comes out to be ∞ then it means that some no feature is left and all of them have been discarded. As this value reaches the proximity of ∞ then we can assume that some features are being discarded by the model.With the increase in the shrinkage value λ the bias also increases. The decrease in the value of λ leads to increase in the variance.
Bias can be defined as the amount by which the prediction of our model differs from the target feature when we compare it to the training set. Model selection can be used to introduce the bias. In order to learn fast, linear model have high value of bias. On the other hand variance indicates how much the value of the target will deviate if the training set is altered. Variance can also lead to the over fitting of data where minute variations in the training data can get highlighted. Increasing the variance will decrease the bias and vice versa. The correct balance has to be determined while evaluating both these values. In supervised algorithms of machine learning we aim to achieve low variance and bias value in order to get good prediction results.

3-Decision Tree Regression : The Decision tree regression, (Navlani (2018)) comes under one of the supervised machine learning models. It is usually used for categorical values and continuous variables of output. As clear from the name of the model, it uses a tree form of structure for the development of classification and regression based models. It divides the given dataset further into smaller subsets and hence simultaneously through association method, decision tree is developed incrementally. It trains the

model in the structure of a tree like model and yields a continuous meaningful value as an outcome.

Decision tree can be considered as the model based on predictions which depict the target feature with the help of some binary rules. It forms a simple model which is comprised of leaves, nodes and branches. Root nodes represent the entire structure of the data which gets further split into smaller subsets which are homogenous in nature. Splitting is the process of breaking down the given node into two or more subsets.

Decision node is the one which can be defined as a sub-node which further gets divided into sub-nodes. Leaf nodes or terminal nodes are the one which cannot be divided further. Pruning is defined as reverse splitting process in which we remove the branches of any tree. It can be done using few ways like limiting the height of our tree, ignoring or removing the leaves which have few branches or by setting a limit on the leaf nodes number. The nodes which get divided into sub nodes are called the parent nodes and the ones which we get as the result of the splitting is known as the child node.

To determine the purity of the split we can make use of various features. Gini Impurity is defined as a measure to depict the purity of the split. It is a value which lies in the range of 0 to 0.5 where the value of 0 determines that the split is pure which means 100% lies in the same set or class. A value of 0.5 means impure split where 50% value lies in one class and 50% in other class. This makes the split difficult.

$$\text{Gini value} = 2 \times \text{pr} \times (1\text{-pr}).$$

Here "pr" is the probability or percentage of true events and (1-pr) is the probability of false events. Another measure of the purity of split is Entropy. The value of entropy lies in the range of 0 to 1. Here the value of 0 signifies that the split is pure and the data belongs to the same class whereas the value of 1 defines that the split is impure and 50% data is present in one class and remaining half is present in the other. Entropy value = $-[\text{pr} \log_2 \text{pr} + (1\text{-pr}) \log_2 (1\text{-pr})]$, where Probability (pr) is the probability of positive event and (1-pr) is the probability of negative event.

The parameter of "criterion" is used to depict how the impurity of the splitting procedure. Generally Ginni value is used for the criterion part but it works effectively in case of categorical data. Our data is continuous in nature hence we cannot use the ginni impurity value for this purpose. We have made use of the Mean Square Error (MSE) value for this implementation.

$$\text{MSE} = 1/N \sum_{i=1}^{N} (v_i - v_i^-)^2.$$

In the "splitter" parameter the decision tree determines and search for the features which can be considered for the split. We have set the default value as best which implies that the model will make a choice from all the features and then decide the feature which would be appropriate for the purpose of splitting.

Out[93]:

	model_type	best_score	best_params
0	linear_regression	0.847796	{'normalize': False}
1	decision_tree_regressor	0.716033	{'criterion': 'friedman_mse', 'splitter': 'best'}
2	lasso_regression	0.726802	{'alpha': 2, 'selection': 'random'}

Figure 3.5: Scores of the Models

4-Ridge Regression: This algorithm is used to tune our model and check the presence of multicollinearity. When this problem occurs then we get large value of variance. This might interfere in the predicted values being different than the actual ones to a large extent. L2 regularization is used in the implementation of the model.The cost function in ridge regression is depicted as:

$$\text{MIN}(\|(\text{v-y})\ (\theta)\ \|^2 + \lambda\ \|(\theta)\|^2)$$

Here lambda represents the term of penalty. It is known as the alpha function in the ridge model. As the value of this this increases, the error margin also increase and thus the coefficients magnitude is lowered. The parameters get shrink which also eliminate the multicollinearity. The coefficient shrinkage also reduces the complexity of the model.

The usual regression equation for most of the models in machine learning is in the form of V = ZB + e where V is the dependent factor and Z is the independent factor. The coefficient of regression which is to be depicted is represented as B. The residuals and error is given as e. After adding the function of lambda the variance which is not taken into account by general models are approved. In the L2 regularization we take the summation of the square values of the coefficients as a factor for the optimization purpose. Residual sum square + α * (summation of square value of coefficient).

The value of alpha plays a major role in evaluation of the model. If the value is 0 then it can be treated like the linear regression method and the coefficients are also obtained in the linear manner. If the value is ∞ then the coefficients are eliminated because if the weightage on square value of coefficients become infinite then any value less than 0 will make infinite objective. If the value is greater than 0 but less than ∞ then the value of alpha will determine the distribution of the weightage of different coefficients. For simple linear regression the value is close to around 0 and 1.

The main aim of the regularization is to make the set of hypothesis small. If this set is large in size then it could lead to complex evaluations. The validation error is lowered which sometimes come at the expense of the error in training. Regularization is very productive when we look at the performance of prediction with lesser variance value at the price of the value of bias, which can be expensive in terms of computation. If a model is trained on lower dimensional dataset then it can be more productive computationally. The output generated from this model is not unbiased in nature.

The ridge model had an accuracy of 80.816 and the value of Cross Validation (CV) came out to be 59.798. In the cross validation method we basically split or divide our data into number of partitions or folds and then we perform the model evaluation. After getting the score of each analysis we calculate the average of this score.

5-Random Forest Model: (Hong et al. (2020)) This is a supervised model in machine learning. It has a very wide application in fields of medicines, stock market, real estate, banking, etc. As the name suggests it develops a forest with the help of decision trees by the means of ensemble. Every individual tree in the forest split into the prediction of the class.

The class which secures more number of votes is considered for the prediction purpose. Low correlation should be ensured while working with this model. Every tree in the system also tries to lower the errors which they generate individually. It uses the bagging concept where the collection of models leads to increase in the result. To sate in simple words, this model make a collection of decision trees and put them in a forest to get more stable and precise predicted outcomes.

This model can be used for the classification and regression based problems which add to its advantages.It can also be implemented for Structural Identification (SI) and Complete Structural Identification (CSI). The hyper-parameters are similar in nature in this model when compared with the decision tree. We can also add some randomness to our data using this model while developing this model. It searches the best factor among the set of random features subsets instead of determining the best feature among the random set of the factors for splitting. Hence in random forest model only a subset of factor which is random, is taken into account for the splitting of the given node. Trees can be developed more randomly by implementing additional thresholds instead of searching of the appropriate feature for the split. Also in this model it is relatively more convenient to determine the value and importance of the features. We use Sklearn to determine that which feature can be used to reduce the factor of impurity in the forest which is developed using the decision trees. It evaluates the score for every factor and scales the outcomes obtained after the training.

This model also has an added advantage that in most cases the hyper-parameters that are set to default generally give good prediction results. The issue of over-fitting can also be resolved using the random forest method if there is sufficient number of trees in the forest. But the main problem which arises in this case is that the more number of trees the model has, the complexity and the duration of execution increases which makes the model slow in real time execution of the system. Generally these models take less time while training but when it comes to the real time execution then the model might slow down to some extent. In our model, the accuracy score came out to be 78.541 and the cross-validation score was 58.724. In order to increase the performance of the model we need to make sure that some predictive ability should be present in the features for the predicting purpose else it would create problem in the evaluation of the model.

The prediction of the trees in the forest and the trees themselves should not be correlated and if they are, the value of correlation should be as low as possible. The factors and the hyper-parameters which have been chosen also impact the correlation.

3.1.6 Evaluating the Model

After selecting the model for the evaluation, we need to test that model against the values of the unseen data or the test data. This unseen dataset is basically used to depict the performance of the model when it is put into application in the real world.
The training data is the one on which the model is first evaluated and this set is considered as the seen data as the model has the experience of working with this data. In case of over-training, the model develops a great understanding of the train data but when some real world data or the unseen data or the test data is given to the model then it fails to give good accuracy. This is because the model develops the extra capacity to predict only from the training data and anything beyond that set would not be able to get handled by the model.
Usually the training set and the testing set is split in the ratio of 7:3 or 70% training data set and remaining 30% for the testing data set. This ratio can be increased or decrease as per use but keeping a very small set of data for the testing part may leave the model as under-trained and when the test data would be exposed to the model then it might fail to give optimum results. The training set should also not be very large as the problem of over-fitting may arise. There should be sufficient data that can be used in order to test that how the model performs when it is exposed to the unseen data.

3.2 User Interface and Connectivity to Web page

The website is connected using the python flask framework to the backend. The flask server provides a local IP address on which the website runs. The IP address provided by the flask is used to transfer the details which the user has input to the flask. Flask is one of the frameworks used in the web. There is usually no in built interaction with the database. Flask-sqlalchemy is a package which is used to connect to the database.

A wide variety of database management system can be implemented using this package. The use of environment variables is also important because it ensures that no matter on which system the code will run; it will always indicate the correct information.

The server consists of two files. Server.py file is used for getting the locations by handling the paths or routes and predict the property price. It also transfers the inputs from frontend to the util file. Such paths are tested using Postman application. The util file acts as the main brain in backend. It loads the pickle and JSON file. This file takes inputs from server file and uses the machine learning model to predict property price.

The flask acts as an interface which interacts with the model and the website which

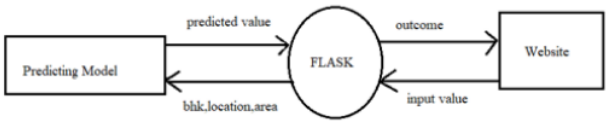


Figure 3.6: Operation of the Flask System

seems to behave like the brain of the website. The user has to submit the input values like the number of rooms, bathrooms, area, location in the city, etc. After entering this information on the front end, which is the website, this information is carried to the machine learning predicting model which generates the price of the property based on these factors. This value is carried by the flask model to the front end of the system where the user can view the output as the predicted cost of the property in Lakhs. The predicting model remains hidden for the user and only the frontend of the system is displayed where the flask acts as an intermediate layer between the front end and the prediction system of the machine learning.

CHAPTER 4

SYSTEM DESIGN

The system consists of mainly two sections. First section is of the machine learning model and the other section is of the user interface part which would host the website and the result will be displayed on the system. These sections combined together will form the resultant system as a whole.

4.1 Machine Learning Model

In this section we have developed a predicting model which would estimate the price of the property based on various factors. Different kinds of algorithms have been used and the one with good score will be taken into evaluating the further system.

First step is to collect the data which we have collected from the Kaggle repository.

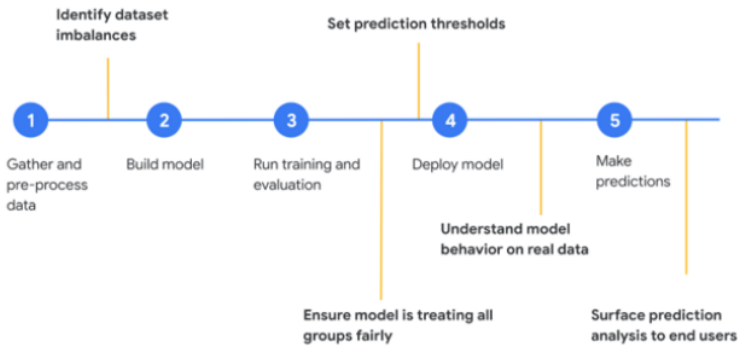


Figure 4.1: Machine Learning Systems

Then we have eliminated the imbalances like missing data or invalid form of data that was present in the data set. Outliers present in the data can also be eliminated at this stage. The mode and mean values were taken into account while dealing with the imbalances. For outliers, certain statistical methods were taken into account. Randomization

of the data is also taken into account as it eliminates the consequences of the order in which data is collected. After removing these problems we have to build the predicting model with the pure data obtained. We check the score of the accuracy and then select the appropriate model into account. Training and testing of the data is done and the model is evaluated on the given parameters. After the training part the model is tested on the unseen or the test data. When this stage is completed then we deploy the model and start making the predictions for the price.

In this first component we have made use of various libraries like NumPy and Pandas for the data preparation and cleaning. Mtplotlib was also taken into account which was used for the visualization part. Sklearn was used for the building of the predictive model.

4.2 User Interface Section

After obtaining the predicted price from the machine learning predictor model, we need to surface that value to the user in the interface section. The concept of python flask server has been used to implement the interface part. It acts as an interface between the predicting model and the frontend.

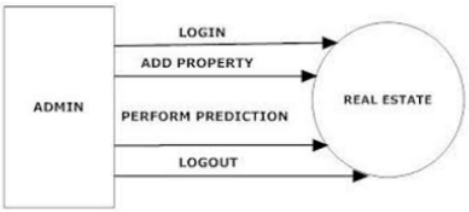


Figure 4.2: UI Section

The frontend is made of HTML, CSS and basic Javascript. In this part the user needs to input the values as per the requirement like number of rooms, area, location, etc. After this, the inputs are sent to the predicting model which gives the price based on these inputs and this value is transmitted via flask server.

CHAPTER 5

EXPERIMENTAL OUTCOMES

5.1 Jupyter Notebook Code

The project consists of coding which is done in various sections like Jupyter Notebook and Pycharm. The various sections will be discussed in this chapter.

Jupyter Notebook code is presented below:

```
import pandas as vpd
import numpy as vnp
from matplotlib import pyplot as vplt
%matplotlib inline
import matplotlib
matplotlib.rParams["figure.figsize"] = (20,10)
vdf1 = pd.read_csv(r"C:\Users\VARUN NIGAM\Downloads\
    Bengaluru_House_Data.csv")
vdf1.head()
vdf1.shape
vdf1.columns
vdf2 = df1.drop(['area_type_val','society area type','balcony number'
    ,'available'],axis='columns')
vdf2.shape
vdf2.isnull().sum()
vdf2.shape
vdf3 = vdf2.dropna()
vdf3.isnull().sum()
vdf3['bhk'] = df3['size'].apply(lambda vx: int(x.split(' ')[1]))
vdf3.bhk.unique_val()
def is_float(vx):
    try:
        float(vx)
    except:
        return False
```

```
    return True
vdf3[~vdf3['total_sqft'].apply()].head(11)
def convert_sqft_to_num(x):
    tokens = x.split('-')
    if len(tokens) == 1:
        return (float(tokens[0])+float(tokens[1]))/2
    try:
        return float(x)
    except:
        return None
vdf4 = vdf3.copy()
vdf4.total_sqft = vdf4.total_sqft.apply(convert_sqft_to_num)
vdf4 = vdf4[vdf4.total_sqft.notnull()]
vdf4.head(3)
vdf4.loc[31]
vdf5 = vdf4.copy()
vdf5['price_per_sqft_area'] = vdf5['price']*100000/vdf5['total_sqft']
df5.head()
vdf5_stats = vdf5['cost_per_sqft_area'].describe()
vdf5_stats
vdf5.to_csv("bhp.csv",index=False)
vdf5.location = vdf5.location.apply(lambda x: x.strip())
location_stats = vdf5['location'].value_counts()
location_stats
location_stats.sum()
len(location_stats[location_statsVAL>10])
len(location_stats[location_statsVAL<=10])
location_stats_less_than_11 = location_stats[location_statsVAL<=11]
location_stats_less_than_10
len(vdf5.location.unique_val())
vdf5.location = vdf5.location.apply(lambda h: 'other' if h in
    location_stats_less_than_11 else h)
len(df5.location.unique())
vdf5[vdf5.total_sqft/vdf5.bhk<310].head()
df6 = vdf5[~(df5.total_sqft/df5.bhk<310)]
df6.shape
df6.price_per_sqft.describe()
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
```

```
    for key, subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per_sqft<=(m+st))]
        df_out = pd.concat([df_out,reduced],ignore_index=T)
    return df_outv
vdf7 = remove_pps_outliers(df6)
vdf7.shape
def plot_scatter_chart(df,location):
    bhk_2 = df[(df.location==location) & (df.bhk==2)]
    bhk_3 = df[(df.location==location) & (df.bhk==3)]
    matplotlib.rcParams['figure.figsize'] = (16,11)
    plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2_BHK', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='+', color='green', label='3_BHK', s=51)
    plt.xlabel("Total Area (Square Feet)")
    plt.ylabel("Price in Lakh (Indian Rupees)")
    plt.title(location)
    plt.legend()

plot_scatter(vdf7," Area Rajaji Nagar")
plot_scatter(vdf7,"Sarjapur  Road Area")
def remove_bhkv_outliers_fn(df):
    exclude_indices = np.array([])
    for location, loc_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in loc_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'meanval': np.mean(bhk_df.price_per_sqft_area),
                'stdval': np.std(bhk_df.price_per_sqft_area),
                'countval': bhk_df.shape[1]
            }
        for bhkv, bhk_df in location_df.groupby('bhkv'):
            stats = bhk_stats.get(bhk-1)
            if stats ['count_val']>6:
                excl_indices = np.append(exclude_indices, bhk_df[bhk_df.price_per_sqft_area<(stats['means'])].index.values)
```

```
    return df.dropval(excl_indices,axis='indexes')
hdf8 = remove_bhk_outliers(vdf7)
hdf8.shape
plot_scatter_chart(hdf8," Area Rajaji Nagar")
plot_scatter_chart(hdf8," Area Sarjapur  Road")
import matplotlib
matplotlib.rParams["figure.figsizeval"] = (25,15)
plt.hist(vdf8.price_per_sqft_area,rwidth=0.6)
plt.grid(True)
plt.title("Count of Properties based on Price per sqft.")
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count of Properties")
df8.bath.unique()
plt.hist(df8.bath,rwidth=0.5)
plt.grid(True)
plt.title('Mode of bath')
plt.xlabel("No. of bath")
plt.ylabel("Count of bath")
hdf8[hdf8.bath>9]
hdf8[hdf8.bath>hdf8.bhk+3]
gdf9 = hhdf8[hdf8.bath<hdf8.bhk+3]
gdf9.shape
gdf9.head(5)
df10 = gdf9.drop(['size','price_per_sqft_area'],axis='columns')
df10.head(3)
dummy = pd.get_dummies(df10.location)
dummy.head(5)
df11 = pd.concat([df10,dummies.drop()],axis='column')
df11.shape
df12 = df11.drop('locationval',axis='columns')
df12.head(5)
X = df12.drop(['priceval'],axis='')
X.head(3)
X.shape
y = df12.price
y.head(3)
len(y)
5
from sklearn.model_selection library import train_test_split model
```

```
vX_train, vX_test, vy_train, vy_test = train_test_split(vX,vy,
    test_size=0.2,random_state=15)
from sklearn.linear_model import LinearRegression
lr_clfval = LinearRegression()
lr_clfval.fit(vX_train,vy_train)
lr_clf.score(vX_test,vy_test)
from sklearn.model_selection library import ShuffleSplit module
from sklearn.model_selection import cross_val_score
cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
cross_val_score(LinearRegression(), X, y, cv=cv)
from sklearn.model_selection import GridSearchCV
from sklearn. library linear_model import Lasso model
from sklearn.linear_model import Ridge_model
from sklearn.tree import DecisionTreeRegressor_model
def find_best_model_using_gridsearchcv(X,y):
    algorth = {
        'linear_regression' : {
            'model_type': LinearRegression(),
            'params': {
                'normalize': [True, False]
            }
        },
        'decision_tree_regressor': {
            'model_type': DecisionTreeRegressor(),
            'params': {
                'criterion' : ['mseval','friedman_mse'],
                'splitter': ['best','randomstate']
            }
        },
        'lasso_regression_model': {
            'model_type': Lasso(),
            'params_list': {
                'alpha_val': [1,3],
                'selection_criterion': ['random', 'cyclicval']
            }
        }


    }
```

```
    scores = []
    cv = ShuffleSplit(n_splits=6, test_size=0.3, random_state=5)
    for algo_name, config in algos.list():
        gs =  GridSearchCV(config['model_type'], config['params'],
    return_train_score=TRUE)
        gs.fit(X,y)
        scores.appendval({
            'model_types': algo_name,
            'best_scoreval': gs.best_score,
            'best_param': gs.best_params_
        })

    return pd.DataFrame(scores,columns=['model_types','best_scoreval'
    ,'best_param'])

find_best_model_gridsearchcv(vX,vy)
from sklearn.model_selection import cross_val_score_model
def classify(model,x,y):
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size
    =0.30,random_state=40)
    model.fit(x_train,y_train)
    print("Accuracy:",model.score(x_test,y_test)*100)
    score=cross_val_score(model,x,y,cv=5)
    print("Cross validation is",np.mean(score)*100)
from sklearn.ensemble library import RandomForestRegressor_model
modl=RandomForestRegressormodel()
classify(model,X,y)
from sklearn. library linear_model import Ridge_model
modl=Ridgereg()
classify(modl,vX,vy)
```

5.2 Pycharm Code

The server code is depicted below which is implemented on PyCharm IDE.

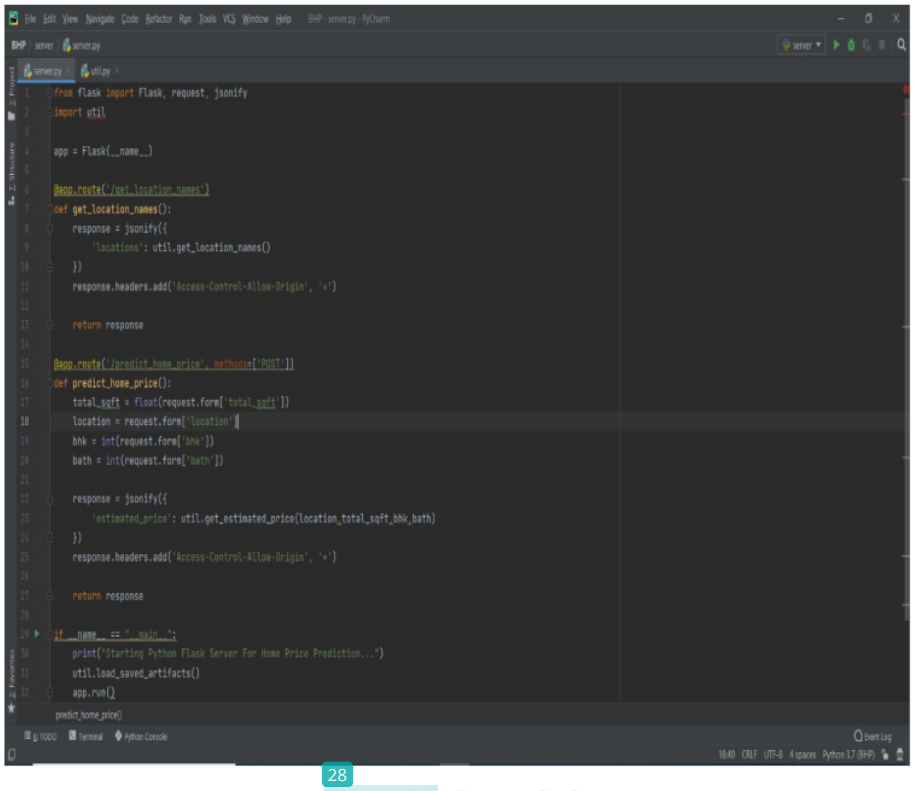
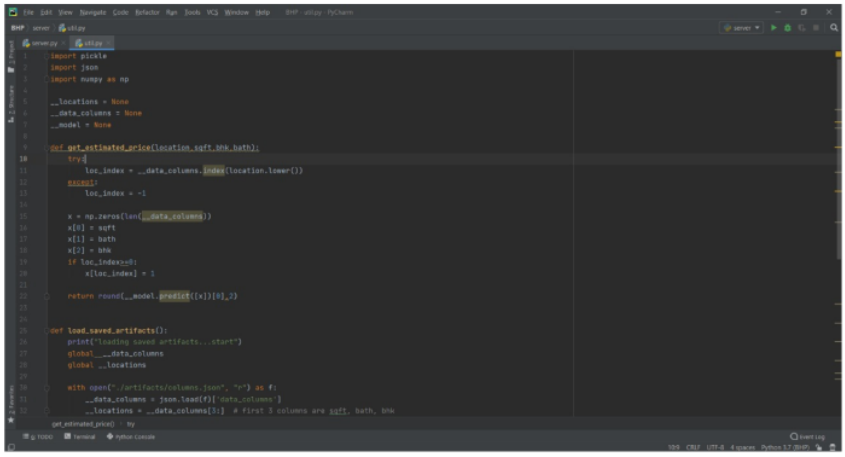
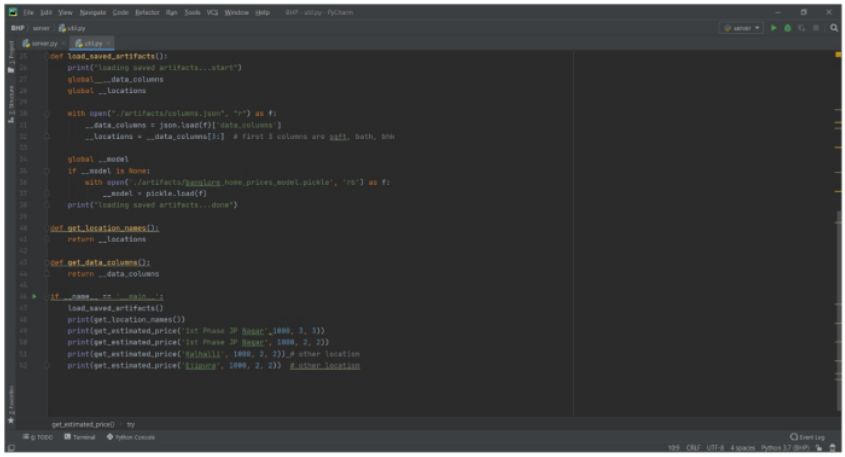


Figure 5.1: Server Code



```
import pickle
import json
import numpy as np

__locations = None
__data_columns = None
__model = None

def get_estimated_price(location,sqft,bhk,bath):
    try:
        loc_index = __data_columns.index(location.lower())
    except:
        loc_index = -1

    x = np.zeros(len(__data_columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index>=0:
        x[loc_index] = 1

    return round(__model.predict([x])[0],2)


def load_saved_artifacts():
    print("loading saved artifacts...start")
    global __data_columns
    global __locations

    with open("./artifacts/columns.json", "r") as f:
        __data_columns = json.load(f)['data_columns']
        __locations = __data_columns[3:]  # first 3 columns are sqft, bath, bhk
```

Figure 5.2: Util Code 1



```
def load_saved_artifacts():
    print("loading saved artifacts...start")
    global __data_columns
    global __locations

    with open("./artifacts/columns.json", "r") as f:
        __data_columns = json.load(f)['data_columns']
        __locations = __data_columns[3:]  # first 3 columns are sqft, bath, bhk

    global __model
    if __model is None:
        with open('./artifacts/bangalore_home_prices_model.pickle', 'rb') as f:
            __model = pickle.load(f)
    print("loading saved artifacts...done")

def get_location_names():
    return __locations

def get_data_columns():
    return __data_columns

if __name__ == '__main__':
    load_saved_artifacts()
    print(get_location_names())
    print(get_estimated_price('1st Phase JP Nagar',1000, 3, 3))
    print(get_estimated_price('1st Phase JP Nagar', 1000, 2, 2))
    print(get_estimated_price('Kalhalli', 1000, 2, 2))# other location
    print(get_estimated_price('Ejipura', 1000, 2, 2))  # other location
```

Figure 5.3: Util Code 2

5.3 Results

```
In [62]: predict_price('1st Phase JP Nagar',1000, 2, 2)
Out[62]: 83.86570258312173

In [63]: predict_price('1st Phase JP Nagar',1000, 3, 3)
Out[63]: 86.0806228498695

In [64]: predict_price('Indira Nagar',1000, 2, 2)
Out[64]: 193.3119773317986

In [65]: predict_price('Indira Nagar',1000, 3, 3)
Out[65]: 195.52689759854636
```

Figure 5.4: Experimental Result

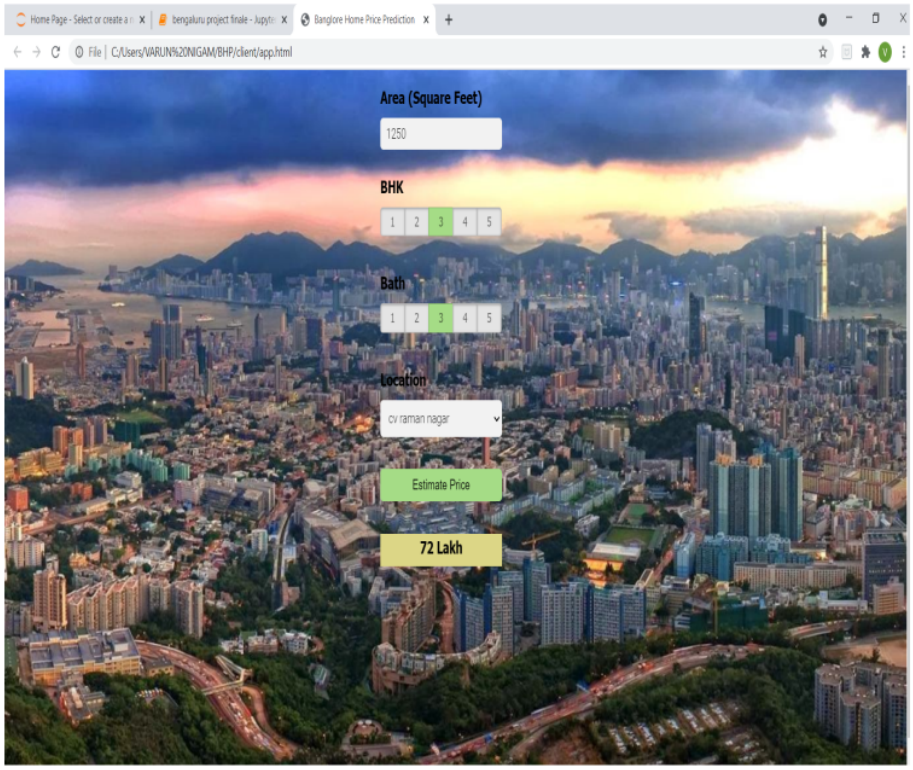


Figure 5.5: Working System

CHAPTER 6

CONCLUSION

The rapid increase in the demand in real estate and at the same time the increase in the hidden factors has made this field one of the least transparent one. With the increase in the number of people who are migrating from one place to other, from towns to cities or from one city to other, has made the real estate machinery an important aspect. Real-estate also depicts the current scenario of the state/country and many find it as a good investing option.

The aim of the project was to develop an interactive machinery or system which can depict the price of any property based on several factors. The user makes input, as per his requirement, on the frontend or the webpage and gets the predicted price. A model, however optimal it might be, can always be made more robust. Various machine learning algorithms were implemented in the system and best performing model amongst them was used for the further evaluation. The use of various advanced techniques like neural networks and hybrid algorithms can be done to improve the accuracy further.

CHAPTER 7

SCOPE FOR FUTURE

The given model is implemented successfully but there were some factors which could have been taken into account. Bengaluru is a rapidly growing city hence more features can be included to depict the property price. Characteristics like availability of parking for vehicles, swimming pool, availability of schools and colleges, connectivity with highways and metro stations and availability of other public transport also affect the price of any property. For instance the price of any property will be more if there is connectivity with public transport or there is availability of schools and other educational institutes around the area. If the property does not have proper availability of the amenities around it then the price would naturally decrease.

Also the with the help of the price of the city we cannot estimate the price of properties of the villages or towns surrounding the city as the prices in metropolitan cities are not comparable to the prices in small towns. The use of various deep learning algorithms and hybrid algorithms can be done in order to improve the prediction. A recommendation system can also be added to make the task of the user little easier by estimating and recommending the property of the choice of the user at the same time.

REFERENCES

1. Ghosalkar, N. N. and Dhage, S. N. (2018). "Real estate value prediction using linear regression." *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. 1–5.
2. Hong, J., Choi, H., and Kim, W.-s. (2020). "A house price valuation based on the random forest approach: the mass appraisal of residential property in south korea." *International Journal of Strategic Property Management*, 24(3), 140–152.
3. Limsombunchai, V. (2004). "House price prediction: hedonic price model vs. artificial neural network." *New Zealand agricultural and resource economics society conference*. 25–26.
4. Lu, S., Li, Z., Qin, Z., Yang, X., and Goh, R. S. M. (2017). "A hybrid regression technique for house prices prediction." *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, IEEE. 319–323.
5. Lydia, E. L., Bindu, G. H., Sirisham, A., and Kiran, P. P. (2019). "Electronic governance of housing price using boston dataset implementing through deep learning mechanism." *International Journal of Recent Technology and Engineering (IJRTE) ISSN*, 2277–3878.
6. Madhuri, C. R., Anuradha, G., and Pujitha, M. V. (2019). "House price prediction using regression techniques: A comparative study." *2019 International Conference on Smart Structures and Systems (ICSSS)*, IEEE. 1–5.
7. Manasa, J., Gupta, R., and Narahari, N. S. (2020). "Machine learning based predicting house prices using regression techniques." *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. 624–630.
8. Navlani, A. (2018). "Decision tree classification in python." *Data Camp*.
9. Phan, T. D. (2018). "Housing price prediction using machine learning algorithms: The case of melbourne city, australia." *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, IEEE. 35–42.
10. Putatunda, S. (2019). "Proptech for proactive pricing of houses in classified advertisements in the indian real estate market." *arXiv preprint arXiv:1904.05328*.
11. Shinde, N. and Gawande, K. (2018). "Valuation of house prices using predictive techniques." *International Journal of Advances in Electronics and Computer Science, ISSN*, 2393–2835.

varunr1

ORIGINALITY REPORT

9%	8%	4%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	indianaiproduction.com Internet Source	2%
2	mafiadoc.com Internet Source	1%
3	www.scribd.com Internet Source	1%
4	ijesc.org Internet Source	1%
5	Vaibhav Verdhan. "Supervised Learning with Python", Springer Science and Business Media LLC, 2020 Publication	1%
6	documents.mx Internet Source	1%
7	www.iraj.in Internet Source	<1%
8	Sandali Khare, Mahendra Kumar Gourisaria, GM Harshvardhan, Subhankar Joardar, Vijander Singh. "Real Estate Cost Estimation	<1%

Through Data Mining Techniques", IOP Conference Series: Materials Science and Engineering, 2021
Publication

9	www.pluralsight.com Internet Source	<1%
10	Vivek Singh Rana, Jayanto Mondal, Annu Sharma, Indu Kashyap. "House Price Prediction Using Optimal Regression Techniques", 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020 Publication	<1%
11	Berna EROL, Recep Fatih CANTEKIN, Seda Karadeniz KARTAL, Rifat HACIOGLU et al. "Improvement of Filter Estimates Based on Data from Unmanned Underwater Vehicle with Machine Learning", 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020 Publication	<1%
12	aisel.aisnet.org Internet Source	<1%
13	www.citv.ac.in Internet Source	<1%
14	machinelearningmastery.com Internet Source	

		<1%
15	journals.vgtu.lt Internet Source	<1%
16	Romano Scozzafava. "Chapter 23 Weak Implication and Fuzzy Inclusion", Springer Science and Business Media LLC, 2010 Publication	<1%
17	V Koktashev, V Makeev, E Shchepin, P Peresunko, V V Tynchenko. "Pricing modeling in the housing market with urban infrastructure effect", Journal of Physics: Conference Series, 2019 Publication	<1%
18	www.coursehero.com Internet Source	<1%
19	app.trdizin.gov.tr Internet Source	<1%
20	researcharchive.lincoln.ac.nz Internet Source	<1%
21	I-ling Yen. "A Unified Framework for Defect Data Analysis Using the MBR Technique", 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 06), 11/2006 Publication	<1%

22	researchprofiles.canberra.edu.au Internet Source	<1%
23	www.analyticsvidhya.com Internet Source	<1%
24	Régis Gras. "Reduction of Redundant Rules in Statistical Implicative Analysis", Studies in Classification Data Analysis and Knowledge Organization, 2007 Publication	<1%
25	U Bansal, A Narang, A Sachdeva, I Kashyap, S P Panda. "Empirical analysis of regression techniques by house price and salary prediction", IOP Conference Series: Materials Science and Engineering, 2021 Publication	<1%
26	fiercest.ru Internet Source	<1%
27	Marcel A J van Gerven. "On the decoding of intracranial data using sparse orthonormalized partial least squares", Journal of Neural Engineering, 04/01/2012 Publication	<1%
28	dl.lib.mrt.ac.lk Internet Source	<1%
29	doku.pub Internet Source	<1%

30	Bodunde Akinyemi, Oluwakemi Adewusi, Adedoyin Oyebade. "An Improved Classification Model for Fake News Detection in Social Media", International Journal of Information Technology and Computer Science, 2020 Publication	<1 %
31	Gan Srirutchataboon, Saranpat Prasertthum, Ekapol Chuangsuwanich, Ploy N. Pratanwanich, Chotirat Ratanamahatana. "Stacking Ensemble Learning for Housing Price Prediction: a Case Study in Thailand", 2021 13th International Conference on Knowledge and Smart Technology (KST), 2021 Publication	<1 %
32	eprints.kfupm.edu.sa Internet Source	<1 %
33	journalofcloudcomputing.springeropen.com Internet Source	<1 %
34	web-tools.uts.edu.au Internet Source	<1 %
35	www.tandfonline.com Internet Source	<1 %

Exclude quotes On Exclude matches Off

Exclude bibliography Off