Real Time Prediction of Real Estate Market

A Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

Borra Richa Reddy	**19881A0568**
Nagandla Karthikeya	**19881A0597**
Remala Mahesh Reddy	**20885A0511**

SUPERVISOR
Mr. A.Ramesh
Assistant Professor

Department of Computer Science Engineering

VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

April, 2022

VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

Department of Computer Science Engineering

CERTIFICATE

This is to certify that the project titled **Real Time Prediction of Real Estate Market** is carried out by

Borra Richa Reddy	**19881A0568**
Nagandla Karthikeya	**19881A0597**
Remala Mahesh Reddy	**20885A0511**

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering** during the year 2021-22.

Signature of the Supervisor	**Signature of the HOD**
Mr. A.Ramesh	**Dr. Ramesh Karnati**
Assistant Professor	**Professor and Head, CSE**

Project Viva-Voce held on ________________________

Examiner

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Mr. A.Ramesh**, Assistant Professor and Project Supervisor, Department of Computer Science Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **Dr. Ramesh Karnati**, the Head of the Department, Department of Computer Science Engineering, his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science Engineering department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Borra Richa Reddy

Nagandla Karthikeya

Remala Mahesh Reddy

Abstract

The literature on machine learning prediction of property values in India is quite restricted to this day. The real estate market is one of the most price-sensitive field in the world, and it is always changing. It is one of the important domains in which machine learning techniques may be applied to improve and predict costs with great accuracy. Our project's major focus is on predicting house values using real-world factors. Our goal is to base our evaluations on each and every basic parameter that is taken into account when establishing the pricing. In this process, we apply a variety of regression techniques, and the output is based on a weighted average of the numerous techniques to produce the most accurate results, providing us with the least amount of error. The data shows that different algorithms can have a significant impact on accuracy.

Keywords: Prediction; Machine learning; Regression; Decision Tree; Random Forest; Real Estate.

Table of Contents

List of Figures

Abbreviations

Abbreviation	Description
VCE	Vardhaman College of Engineering
ML	Machine learning
LR	Linear Regression
RF	Random Forest
KNN	K Nearest Neighbour
MLP	Multilayer Perceptron
RR	Ridge Regression
LMLR	Linear model Lasso Regression
DT	Decision Tree
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets

CHAPTER 1

Introduction

1.1 Real Estate Market in India

The Real Estate market in India has made some amazing progress. Being the biggest vote based system and the second-biggest crowded country, India has encountered a significant blast in the Real Estate area post-freedom principally after the 1990s. India is on the road to rapid urbanisation and a high-standard of living. [1]It's still a growing economy, but we'll figure it out sooner rather than later. With more people moving to towns and cities in search of better quality of life, the housing industry will become more important. The real estate market is one of the most well-known in the world. Housing, retail, hospitality, and business are the four sub-sectors. After agriculture, the real estate sector creates the second-highest number of jobs in India. This sector is also likely to attract greater non-resident Indian (NRI) investment in the short and long term. The Indian real estate sector has experienced rapid growth in recent years, with growing demands for both office and residential premises. Increased urbanisation and rising household income have spurred demand for residential homes. India is one of the world's top ten most rapidly rising housing markets. By 2030, India's real estate market is predicted to be worth 1 trillion dollars and by 2050 it will contribute 13 Percent to the country's GDP. Fast forward to when supply mechanics began to play a significant part in the industry. As the concept of the nuclear family became more popular, so did the demand for residential space. The first mall, the Spencer Plaza in Chennai, introduced the mall concept to the rest of the country in the early 2000s. Following the introduction of technology and the Internet, IT companies were established in numerous metropolitan, Tier II, and Tier III cities. After 2006, the Indian government approved the modernisation of Brownfield airports such as Mumbai and Delhi, as well as Greenfield airports such as Bengaluru, resulting in a surge in real estate development near the airports. In 2014, Real Estate Investment Trusts (REITs) were formed, allowing investors with modest funds to make secure investments. The government passed the Real Estate (Regulation Growth) Act, or RERA, on May 1st, 2017, to encourage the development of cities, which ultimately benefited home buyers. The real estate business has had a considerable impact on the market in recent years. It contributes the second most to the Indian economy. It also ranks second in terms of employment creation and workforce in India.

1.2 Limitations of Prevailing Methodologies

There has been a significant amount of research in the field of housing price prediction, but none of it has resulted in any practical answers. There is virtually little evidence of a corporation setting up a working house price prediction. [2]For the time being, there are very few digital solutions available for such a large market, and the majority of people and businesses employ the following methods:
Buyers/Customers:
1.When consumers first consider buying a home or real estate, they often go online to research trends and other information. People do this in order to find a home that has all they require. People keep track of the prices associated with these residences while performing these tasks. The common person, on the other hand, lacks specific knowledge and reliable information about the true price. This can lead to misinformation since people believe the prices on the internet are genuine.
2.The second thing that comes to mind is contacting numerous estate agents. The issue is that these agents should only be paid a fraction of the money only to look for a house and establish a price for you. People, in most circumstances, believe this price tag because they have no other options. There may be instances where agents and sellers have a hidden deal and the customer is sold an overpriced home without his or her awareness.

Seller/Agencies:

1. When a person considers selling his or her home, they compare it against hundreds of thousands of other homes that have been listed all over the world. Calculating the price by comparing it to multiple estates takes a long time and can result in inaccurate pricing.

2.Large real estate organisations must sell a variety of items and must assign staff to manage each of these things. Again, the price tag forecast is based on a person, thus there is opportunity for human error. These allocated individuals must also be compensated. However, having a machine crunch the figures for you can save you a lot of time and money while also providing precision that a human cannot.

1.3 Objectives

1.The model's primary goal is to estimate property prices.

2.The proposed design is a browser based web application.

3.The system aids real estate in making quicker decisions.

1.4 Proposed Work

The goal of this system is to determine the price of a house based on the numerous attributes that the user provides as input. The ML model is given these features, and it makes a prediction based on how these features affect the label. This will be accomplished by first looking for an adequate dataset that meets both the developer's and the user's needs. Furthermore, after the dataset has been finalised, it will go through a data cleaning procedure in which all unnecessary data will be removed and the raw data will be converted to a csv file. This will also go through data transformation,[3] where it will be turned into a NumPy array before being delivered to the model for training. Various machine learning techniques will be employed to train the model, and their error rate will be retrieved, resulting in a final algorithm and model that can make correct predictions. Users and businesses will be able to log in and then fill out a form with information about various aspects of their property for which they wish to estimate the price. The form will also be submitted after a comprehensive selection of qualities. The user's data will be sent to the model, and the user will be able to see the estimated price of the property they input in seconds.

It is proposed to construct an algorithm that can predict house prices based on certain input features using machine learning techniques. [4]This algorithm's business application is that classified websites can use it to predict prices of new properties that are about to be listed by taking some input variables and predicting the correct and justified price, avoiding taking price inputs from customers and thus preventing any errors from creeping into the system. We used Google Colab/Jupiter IDE. Jupiter IDE is an open-source web application that allows us to share and create documents using LiveCode, visualisations, equations, and narrative prose. It includes data cleaning, data transformation, numerical value simulation, statistical modelling, data visualisation, and machine learning technologies. And created a method that will allow consumers to get a good idea of the price of real estate. The user can enter their specifications and receive prices for the residences they want. The user can also acquire an example house plan as a reference for houses. In Housing value of the Bangalore is analyzed and forecast by MLP, KNN Regressor, Linear model Lasso Regression,Ridge regression, Decision Tree, Random Forest methods and the corresponding characteristics. After getting rid of the missing samples from the original data set, 80 percent samples are treated as training data and 20 percent samples are treated as test data.

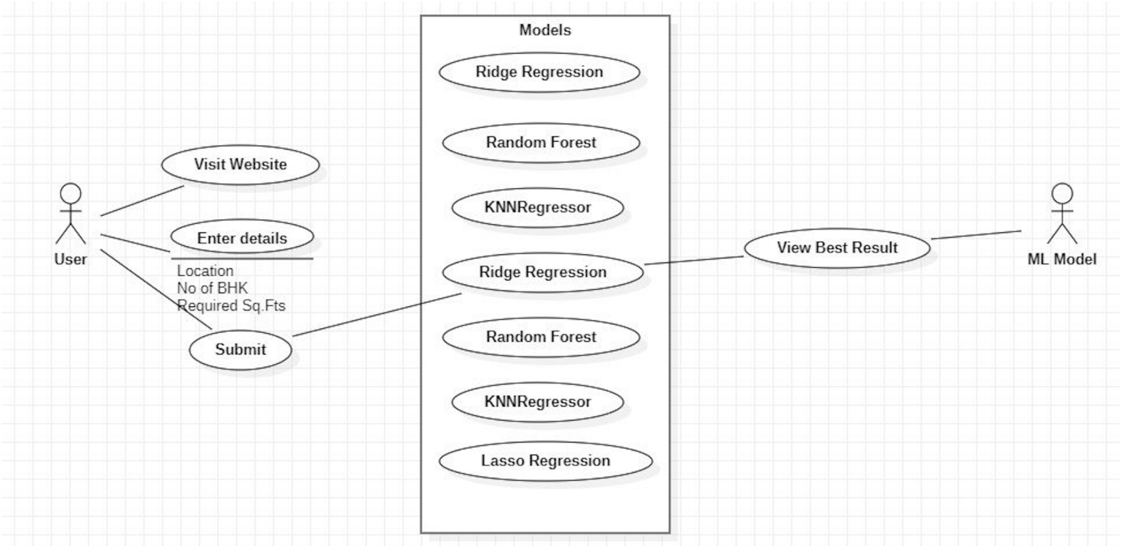


Figure 1.1: use case diagram

CHAPTER 2

Literature Survey

2.1 Machine learning approach

Machine learning is a type of artificial intelligence that consists of readily available computers that may be trained without having to be meticulously designed. Machine learning is interested in the expansions of computer programmes that are capable of changing when exposed to fresh data. Supervised learning, Unsupervised learning, and Reinforcement learning are the three main categories of machine learning algorithms.

Supervised learning is when we teach or train a computer using well-labeled data, which means that part of the data has already been tagged with the correct answer.[5] Following that, the computer is given a fresh collection of examples so that the supervised learning algorithm may analyse the training data and produce a correct result from the labelled data. **Unsupervised learning** is the process of teaching a computer to operate on data that hasn't been classed or labelled and letting the algorithm to act on it without supervision.[6] The machine's objective here is to sort unsorted data into groups based on similarities, patterns, and differences without any prior data training. Unlike supervised learning, there is no teacher present, which implies the computer will not be trained. As a result, the machine is limited in its ability to discover the hidden structure in unlabeled data on its own. **Reinforcement learning** is an area of Machine Learning.. It's all about taking the right steps to maximise your reward in a given situation. It is used by a variety of software and computers to determine the best feasible behaviour or path in a given situation. Reinforcement learning differs from supervised learning in that the solution key is included in the training data, [7]allowing the model to be trained with the correct answer, whereas in reinforcement learning, there is no answer and the reinforcement agent decides what to do to complete the task. It is obligated to learn from its experience in the absence of a training dataset.

Machine[8] learning may be used to bring different data sources together and analyse hidden value, and real estate buyers and investors can use the actionable insights to decide where to invest next. This enables individuals to make better short- and long-term judgments, predicting outcomes with greater accuracy rather than relying just on instinct and hoping for the best. Machine learning has a wide range of applications, one of which is real estate prediction. The main goal of this work is to take these Machine Learning Techniques and turn them into ML models that can be used to service people. A Buyer's first goal is to find their ideal home with all of the facilities they require. Furthermore, people search for these houses/real estates with a budget in mind,[9] and there is no guarantee that they will find a product at a fair price. Similarly, a seller seeks a specific number to use as a price tag for the property; this cannot be a wild guess; extensive study is required to arrive at a house value. The real estate market is one of the most competitive in terms of pricing, and the same tends to vary significantly based on a variety of factors.[10] Predicting future property prices is an important module in decision making for both buyers and investors in supporting budget allocation, property finding stratagems, and determining appropriate policies, so it becomes one of the prime fields to apply machine learning concepts to optimise and predict prices with high accuracy. The analysis of land price trends is deemed necessary to support urban planning decisions.[11]The real estate market is a stochastic process that is inherently unstable. Investors make judgments based on market movements in order to maximise their profits. For their decision-making, developers want to know about future trends. A considerable amount of data that determines land price is required for analysis, modelling, and forecasting in order to effectively estimate property prices and future trends. The elements that influence land prices must be investigated, and their impact on price must be calculated.[12] A review of previous data should be considered. It can be inferred that developing a simple linear mathematical relationship for these time-series data is not feasible for predicting.As a result, it became critical to develop a non-linear model that can accurately fit the data characteristics in order to analyse and forecast future trends. Because the real estate industry is rapidly evolving, the analysis and forecasting of land values using mathematical modelling and other scientific methodologies is a pressing requirement for all parties involved. [13]Land demand began to rise, and housing and real estate activities began to boom. To make space for multistory and high-rise buildings, the barren lands and paddy fields were demolished.Investments began to stream into the real estate industry, and the price of land did not follow a consistent trend over time. The need to forecast land price trends was felt by everyone in the business, including the government, regulatory organisations, financial institutions, developers, and investors. As a result, in this study, we give a number of key features to consider when projecting house prices with high accuracy. To reduce the Residual Sum of Squares error, we can employ regression models with multiple features.

CHAPTER 3

Methodology

3.1 Packages

1.NumPy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python. It is the fundamental package for scientific computing in Python.It provides a high-performance multidimensional array object, and tools for working with these arrays.
2.Pandas: pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with ”relational” or ”labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. It is used for data cleaning and analysis.
3.Sklearn: Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.scikit-learn is built on NumPy, SciPy and matplotlib provides tools for data analysis and data mining. It provides classification and clustering algorithms built in and some datasets for practice like iris dataset, Boston house prices dataset, diabetes dataset etc.
4.Matplotlib: Matplotlib is a graphing package for Python with NumPy, the Python numerical mathematics extension. It provides an object-oriented API for embedding charts into applications utilising GUI toolkits such as Tkinter, wxPython, Qt, or GTK. It is one of the most popular Python packages used for data visualization. It is used along with NumPy to provide an environment

that is an effective open source alternative for MatLab.

5.Keras: Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

6.Subprocess: The subprocess module provides a function named call. This function allows you to call another program, wait for the command to complete and then return the return code. Allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions: os. system os.

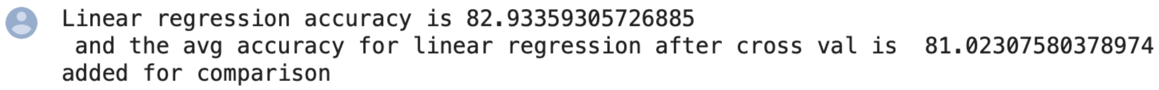
3.2 Algorithms

1.Linear Regression: A statistical tool for undertaking predictive analysis is linear regression. Linear regression makes predictions for continuous, real or numeric variables. The name comes from the fact that the linear regression procedure uncovers a linear relationship between a dependent (y) variable and one or more independent (y) variables. Because linear regression indicates a linear relationship, it shows how the dependent variable's value changes as the independent variable's value changes.

Advantage: More than one predictor can be included in a linear model as long as the predictors are additive. The best fit line is the one that has the least amount of error across all points; it has a high efficiency, but it can also cause problems.

```
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
```

```
cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state
    =0)
LR= LinearRegression()
LR.fit(X_train, y_train)
LRA=LR.score(X_test,y_test)
LRA=LRA*100
CvLR=cross_val_score(LinearRegression(),X,Y, cv=cv)
LRAvg=(sum(CvLR)/len(CvLR))*100
print("Linear regression accuracy is",LRA,"\n and the avg
    accuracy for linear regression after cross val is ",
    LRAvg )
All_Algos.append([LRAvg,'LinearRegression'])
print("added for comparison")
```



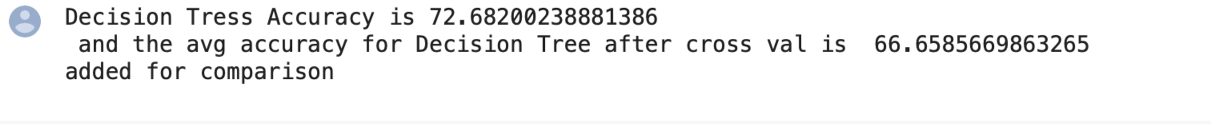
```
Linear regression accuracy is 82.93359305726885
 and the avg accuracy for linear regression after cross val is  81.02307580378974
added for comparison
```

Figure 3.1: Accuracy of Linear Regression

2.Decision Tree: Decision Tree is a supervised learning technique that can be used to solve classification and regression problems, but it is most often used to solve classification problems. In this tree-structured classifier, internal nodes represent dataset properties, branches represent decision rules, and each leaf node delivers the conclusion.
Advantage: During pre-processing, decision trees need less effort for data preparation than other algorithms. A decision tree model is simple to understand and communicate to technical teams and stakeholders.

```
from sklearn.tree import DecisionTreeRegressor
DT = DecisionTreeRegressor(random_state=0)
DT.fit(X_train, y_train)
DTA = DT.score(X_test,y_test)
DTA = DTA*100
```

```
CvDT = cross_val_score(DecisionTreeRegressor(),X,Y, cv=cv)
DTAvg = (sum(CvDT)/len(CvDT))*100
print("Decision Tress Accuracy is",DTA,"\n and the avg accuracy for Decision Tree after cross val is ",DTAvg )
print("added for comparison ")
```

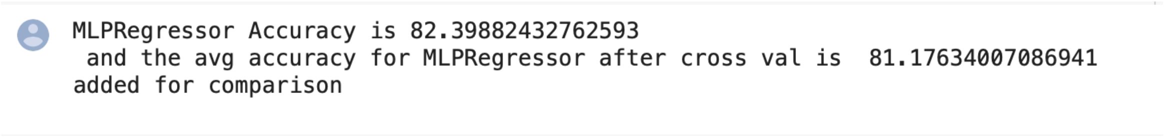


```
Decision Tress Accuracy is 72.68200238881386
 and the avg accuracy for Decision Tree after cross val is  66.6585669863265
added for comparison
```

Figure 3.2: Accuracy of Decision tree

3.Ridge regression: The method of ridge regression is used to analyse multicollinearity in multiple regression data. It works best when the number of predictor variables in a data collection exceeds the number of observations. When multicollinearity occurs in a set, the second-best scenario occurs. Advantage: Ridge Regression eliminates overfitting, which occurs when regular squared error regression fails to distinguish less essential features and uses them all, resulting in overfitting. Ridge regression adds a small bias to the model in order to fit it to the data's true values.

```
from sklearn.linear_model import Ridge
RR= Ridge(alpha=1.0)
RR.fit(X_train, y_train)
RRA=RR.score(X_test,y_test)
RRA=RRA*100
CvRR=cross_val_score(Ridge(alpha=1.0),X,Y, cv=cv)
RRAvg=(sum(CvRR)/len(CvRR))*100
print("Ridge regression Accuracy is",RRA,"\n and the avg accuracy for Ridge regression after cross val is ",RRAvg )
All_Algos.append([RRAvg,"RRegressor"])
print("added for comparison ")
```



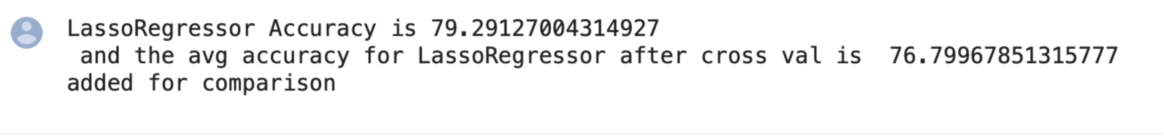
```
MLPRegressor Accuracy is 82.39882432762593
 and the avg accuracy for MLPRegressor after cross val is  81.17634007086941
added for comparison
```

Figure 3.3: Accuracy of Ridge Regression

4.Linear model Lasso Regression:Penalized regression is another name for Lasso regression. In machine learning, this strategy is commonly used to choose a subset of variables. When compared to other regression models, it has a higher prediction accuracy. The Lasso Regularization technique aids in model understanding. The lasso regression penalises the dataset’s less essential attributes.

Advantage:The advantage of lasso is that it eliminates any unneeded parameters from the model.

```
from sklearn import linear_model
LMLR=linear_model.Lasso(alpha=0.1)
LMLR.fit(X_train, y_train)
LMLRA=LMLR.score(X_test,y_test)
LMLRA=LMLRA*100
CvLMLR=cross_val_score(linear_model.Lasso(alpha=0.1),X,Y,
    cv=cv)
LMLRAvg=(sum(CvLMLR)/len(CvLMLR))*100
print(”LassoRegressor Accuracy is”,LMLRA,”\n and the avg
    accuracy for LassoRegressor after cross val is ”,LMLRAvg
    )
All_Algos.append([LMLRAvg,”LassoRegressor”])
print(”added for comparison ”)\\
```



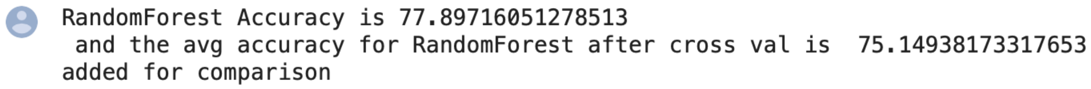
```
LassoRegressor Accuracy is 79.29127004314927
 and the avg accuracy for LassoRegressor after cross val is  76.79967851315777
added for comparison
```

Figure 3.4: Accuracy of Linear model Lasso Regression

5.Random Forest: The Random Forest Regression (RFR) is a multi-regression tree ensemble technique. The output is the average of the individual RTs, which are trained using a random subset of the features.

Advantages: Feature normalisation is not required. The training of individual decision trees can be done in concurrently. Random woods are a popular choice. They help to prevent overfitting.

```
from sklearn.ensemble import RandomForestRegressor
RF= RandomForestRegressor(max_depth=10, random_state=0)
RF.fit(X_train, y_train)
RFA=RF.score(X_test, y_test)
RFA=RFA*100
CvRF=cross_val_score(RandomForestRegressor(max_depth=10,
    random_state=0),X,Y, cv=cv)
RFAvg=(sum(CvRF)/len(CvRF))*100
print("RandomForest Accuracy is",RFA,"\n and the avg
    accuracy for RandomForest after cross val is ",RFAvg )
All_Algos.append([RFAvg,"RandomForest"])
print("added for comparison ")\\
```



```
RandomForest Accuracy is 77.89716051278513
 and the avg accuracy for RandomForest after cross val is  75.14938173317653
added for comparison
```

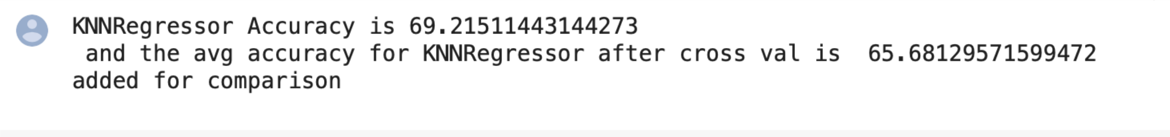
Figure 3.5: Accuracy of Random Forest

6.KNN Regressor:The k-Nearest-Neighbor (KNN) approach is a non-parametric instance-based learning technique. Training is not required in this scenario. Fix Hodges submitted the first work on KNN in 1951 for the United States Air Force. The procedure starts by storing all of our training set's input feature vectors and outputs. We find the k closest neighbours from our training set for each unlabeled input feature vector.

Advantages: The K-Nearest Neighbour(KNN) Classifier is a straightforward classifier that excels at basic recognition tasks.

```
from sklearn.neighbors import KNeighborsRegressor
KNN= KNeighborsRegressor(n_neighbors=4)
```

```
KNN.fit(X_train, y_train)
KNNA=KNN.score(X_test,y_test)
KNNA=KNNA*100
CvKNN=cross_val_score(KNeighborsRegressor(n_neighbors=4),X,
    Y, cv=cv)
KNNAvg=(sum(CvKNN)/len(CvKNN))*100
print("KNNRegressor Accuracy is",KNNA,"\n and the avg
    accuracy for KNNRegressor after cross val is ",KNNAvg )
All_Algos.append([KNNAvg,"KNNRegressor"])
print("Added for comparison ")\\
```



```
KNNRegressor Accuracy is 69.21511443144273
 and the avg accuracy for KNNRegressor after cross val is  65.68129571599472
added for comparison
```

Figure 3.6: Accuracy of KNN Regressor

7.MLP regressor: A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation). Multilayer perceptrons are sometimes colloquially referred to as “vanilla” neural networks, especially when they have a single hidden layer.

Advantages of MLP are:

1.Capability to learn non-linear models.

2.Capability to learn models in real-time (on-line learning).

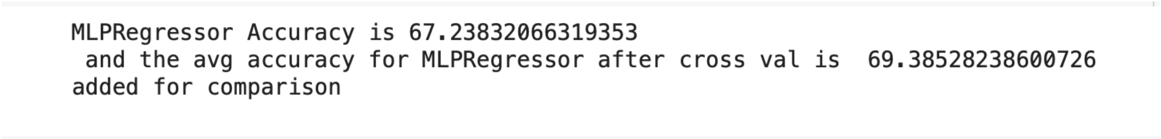
Disadvantages of MLP include:

1.MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore different random weight initializations can lead to different validation accuracy.

2.MLP requires tuning a number of hyperparameters such as the number of hidden neurons, layers, and iterations.

3.MLP is sensitive to feature scaling

```
from sklearn.neural_network import MLPRegressor
MLPR= MLPRegressor(random_state=1, max_iter=500)
MLPR.fit(X_train, y_train)
MLPRA=MLPR.score(X_test,y_test)
MLPRA=MLPRA*100
CvMLPR=cross_val_score(MLPRegressor(random_state=1, max_iter=500),X,Y, cv=cv)
MLPRAvg=(sum(CvMLPR)/len(CvMLPR))*100
print("MLPRegressor Accuracy is",MLPRA,"\n and the avg accuracy for MLPRegressor after cross val is ",MLPRAvg )
All_Algos.append([MLPRAvg,"MLPRegressor"])
print("added for comparison ")
```

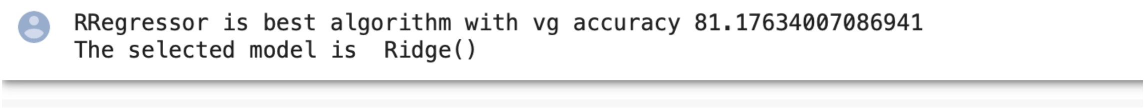


```
MLPRegressor Accuracy is 67.23832066319353
 and the avg accuracy for MLPRegressor after cross val is  69.38528238600726
added for comparison
```

Figure 3.7: Accuracy of MLP Regressor

After determining the accuracy of each algorithm separately. We averaged all of the algorithms and found that Ridge regression had the best accuracy.

```
#To find out the best Accuracy algorithm
Best=sorted(All_Algos)[::-1][0]
All_Names=['LinearRegression',"DecisionTree","MLPRegressor","RRegressor",'LassoRegressor',"RandomForest","KNNRegressor"]
All_models=[LR,DT,MLPR,RR,LMLR,RF,KNN]
BA=All_models[All_Names.index(Best[1])]
print(Best[1],"is best algorithm with vg accuracy",Best[0])
print("The selected model is ",BA)
```



```
RRegressor is best algorithm with vg accuracy 81.17634007086941
The selected model is  Ridge()
```

Figure 3.8: selected Algorithm

3.3 Data Preprocessing

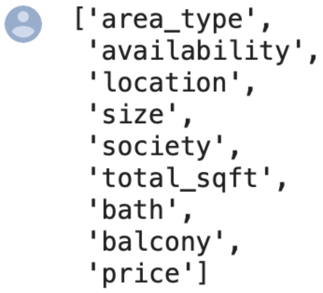
Data preprocessing is the procedure for preparing raw data for use in a machine learning model. It's the first and most important stage in building a machine learning model. It is not always the case that we come across clean and prepared data when working on a machine learning project. It is also necessary to clean and format data before doing any operation with it. As a result, we employ the data preprocessing task. Real-world data sometimes contains noise, missing values, and is in an unsuitable format that cannot be used directly in machine learning models. Data preprocessing is a necessary task for cleaning data and making it suitable for a machine learning model, which improves the model's accuracy and efficiency.

```
data.head() #To show first 5 rows of data
data.tail() #To show last 5 rows of data
data.shape #To get no of rows and columns of data
cols=list(data.columns) #To get list of columns in data
cols
newdata = data.drop(['area_type','society','balcony','availability'],axis='columns') #Drop the columns not required
Rdata=Rdata.dropna(subset=['size','total_sqft'])
newdata.isnull().sum() #Checking if there are null values
newdata.shape #shape of new data
NNdata= newdata.dropna() #Dropping rows with null values
NNdata.isnull().sum()  #Checking if there are null values
NNdata.shape #shape of new data
NNdata.dtypes #To check data types of all elements
```

```
NNdata['size'].unique() #Unique values in Non NUll data size column
NNdata['bhk'] = NNdata['size'].apply(lambda x: int(x.split(' ')[0]))
Rdata['bhk'] = Rdata['size'].apply(lambda x: int(x.split(' ')[0]))
NNdata.head()
Rdata.head() #checking Retrieval data
NNdata['total_sqft'].unique()
#All are string values and some even have a range which is to be corrected
#lets define a function so that it can be reused for every row
def convert_sqft_to_float(x):
tokens = x.split('-')
if len(tokens) == 2:
return(float(tokens[0])+float(tokens[1]))/2
try:
return float(x)
except:
return None
NNdata['total_sqft'] = NNdata['total_sqft'].apply(convert_sqft_to_float)
Rdata['total_sqft'] = Rdata['total_sqft'].apply(convert_sqft_to_float)
def is_float(x):
try:
float(x)
except:
return False
return True
# Creating a new column to find the price per sq-ft
```

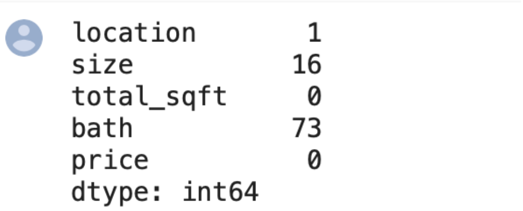
```
#We multiply the price here with 100000 since its in lakhs
NNdata[~NNdata['total_sqft'].apply(is_float)].head(12)
NNdata['Price_per_sqft'] = NNdata['price']*100000/NNdata['total_sqft']
NNdata.head()
NNdata['location'].unique()
NNdata['location'] = NNdata['location'].apply(lambda x: x.strip())
locations = NNdata.groupby('location')['location'].agg('count').sort_values(ascending=False)
locations
NNdata['location'] = NNdata.location.apply(lambda x: 'other' if x in locations[locations<10] else x)
NNdata=NNdata[~(NNdata['total_sqft']/NNdata['bhk']<300)]
NNdata.Price_per_sqft.describe()
def remove_pps_outliers(df):
df_out = pd.DataFrame()
for key, subdf in df.groupby('location'):
m = np.mean(subdf.Price_per_sqft)
st = np.std(subdf.Price_per_sqft)
reduced_df = subdf[(subdf.Price_per_sqft>(m-st)) & (subdf.Price_per_sqft<=(m+st))]
df_out = pd.concat([df_out, reduced_df],ignore_index=True)
return df_out
NNdata= remove_pps_outliers(NNdata)
NNdata.head()
NNdata =NNdata.drop(['size','Price_per_sqft'],axis='columns')
NNdata.head()
Cdata=NNdata.copy()
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
```

```
Corelationlist=Cdata.corr()['price'].abs().tolist()
Cdata.corr()['price'].abs().sort_values(ascending=False)
dummies = pd.get_dummies(NNdata.location)
dummies.head()
NNdata= pd.concat([NNdata, dummies],axis="columns")
NNdata.head()
NNdata=NNdata.drop(['location'],axis='columns')
NNdata.head()
NNdata.shape
X =NNdata.drop(['price'], axis='columns')
X
Y =NNdata['price']
Y
#splitting into training and testing data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,Y,
    test_size=0.2, random_state=10)
```



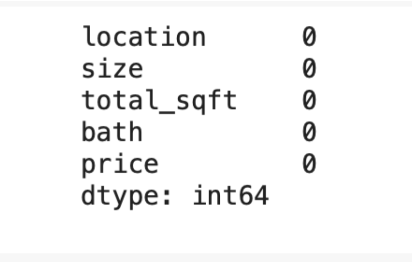
```
['area_type',
 'availability',
 'location',
 'size',
 'society',
 'total_sqft',
 'bath',
 'balcony',
 'price']
```

Figure 3.9: list of columns in data



```
location       1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

Figure 3.10: null values in data



```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

Figure 3.11: Dropping rows with null values

CHAPTER 4

Website Creation

4.1 Web Technologies

1.HTML: HTML stands for Hyper Text Markup Language, which is a programming language used to create online pages and apps. HyperText simply means ”Text within Text”. A hypertext is a text that has a link. A hypertext link is one that takes you to a new webpage when you click on it. HyperText is a method of connecting two or more web pages (HTML documents). A markup [14]language is a programming language for applying layout and formatting principles to text documents. The markup language enhances the interactivity and dynamic nature of text. It can convert text into graphics, tables, and links, among other things.

2.CSS: Cascading Style Sheets(CSS) is a simple design language created to make the process of making web pages appealing. [15]The style and feel of a web page is handled by CSS. You can use CSS to manage the colour of the text, font style, paragraph spacing, how columns are scaled and laid out, [4]what background pictures or colours are used, layout designs, display variants for different devices and screen sizes, and a multitude of other effects. CSS is simple to learn and understand, [16]but it gives you a lot of power over how an HTML document looks. CSS is frequently used in conjunction with the markup languages HTML or X-HTML.

3.JAVA SCRIPT: JavaScript is an interpreted, lightweight programming language. It is intended for the development of network-centric applications. [17]It works in combination with and complements Java. Because JavaScript is interwoven with HTML, it is incredibly simple to use. It’s free to use and cross-platform. JavaScript is now used in game creation, mobile app development, and desktop app development. As a Javascript programmer, this opens up a lot of possibilities for you.

4.Boot-Strap: Bootstrap is the most widely used HTML, CSS, and JavaScript framework for creating mobile-friendly and responsive websites. [18]It provides design templates for typography, forms, buttons, tables, navigation, modals, picture carousels, and more, all built with HTML and CSS. JavaScript plug-ins can also be used. It makes it easier for people to create a responsive website. It works with the majority of browsers, including Chrome, Firefox, Internet Explorer, Safari, and Opera.

4.2 Backend Connection

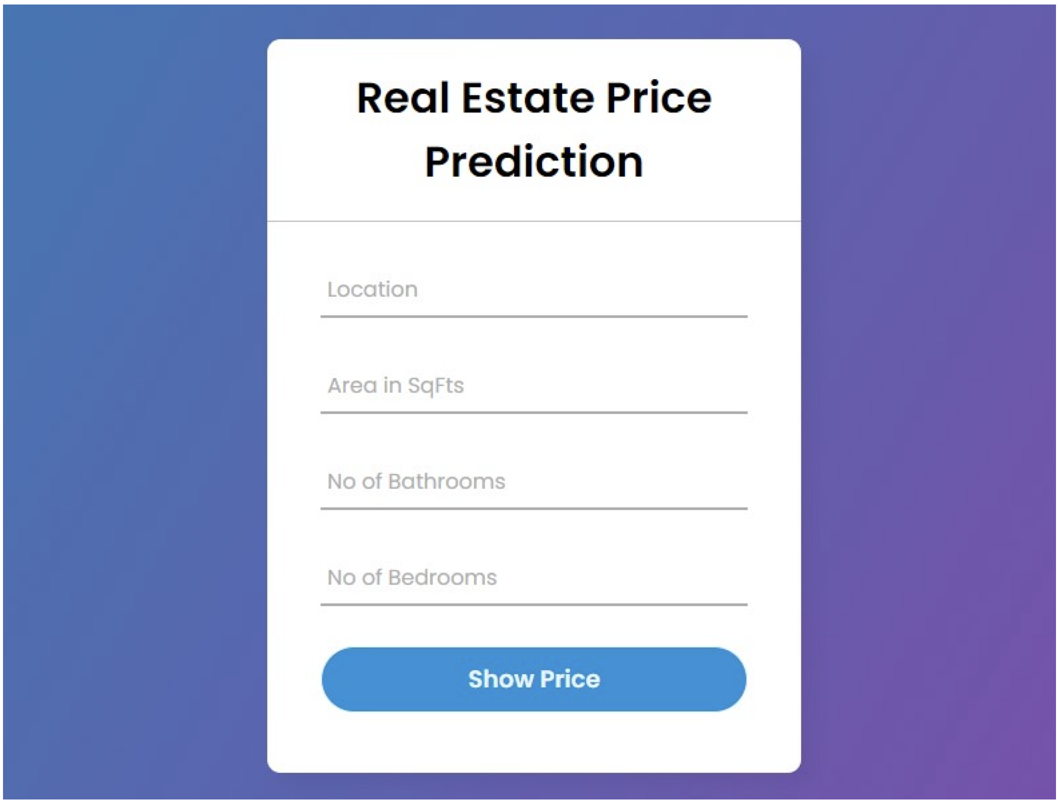
Flask is a back-end framework,[19] which means that it provides the technologies, tools, and modules that can be used to build the actual functionalities of the web app rather than the design or look of it. It is a little and lightweight Python web system that gives helpful devices and elements that make making web applications in Python simpler. It gives designers adaptability and is a more available system for new engineers since you can fabricate a web application rapidly utilizing just a solitary Python document.

Input:

1. Area type-describes the area.
2. Size- in BHK or Bedroom (1-10 or more).
3. Total Square Feet - size of the property in square feet
4. Bathrooms -Number of bathrooms.
5. Balcony- Number of balconies.
6. Price
7. Society

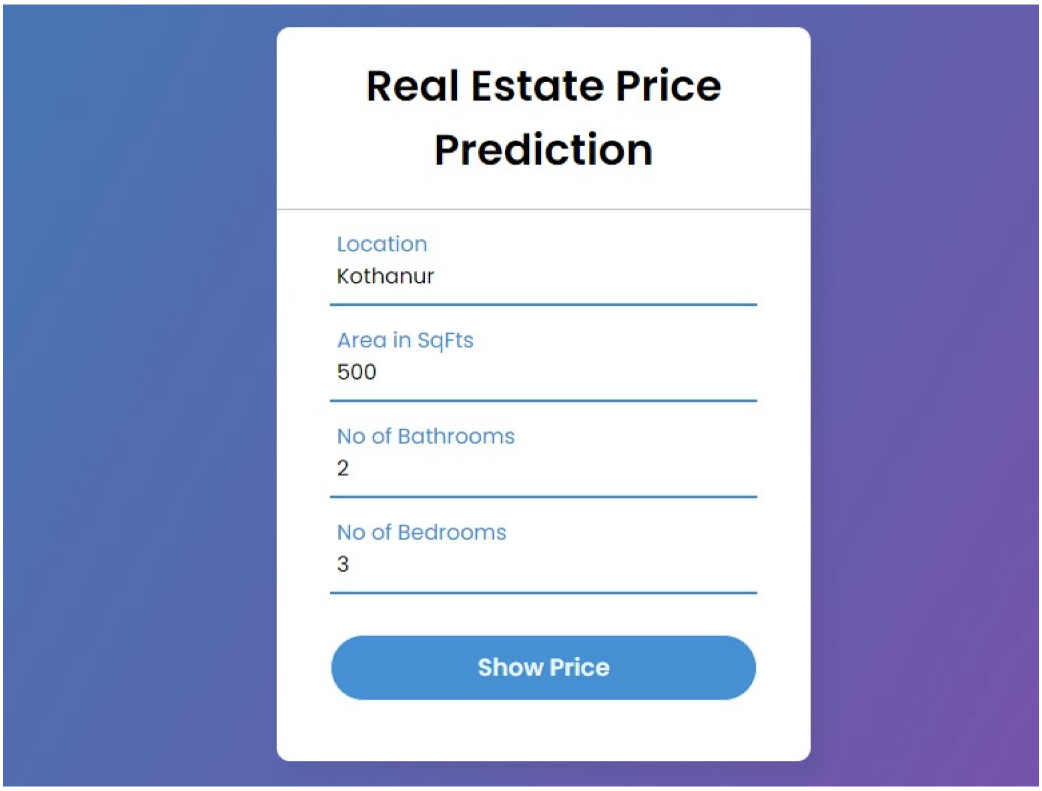
Output:

House price prediction using ML technique.



Real Estate Price Prediction

Figure 4.1: webpage without details entered



Real Estate Price Prediction

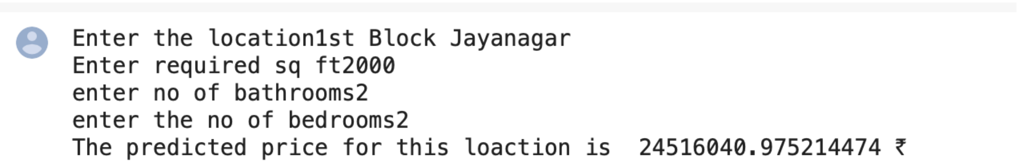
Location
Kothanur

Area in SqFts
500

No of Bathrooms
2

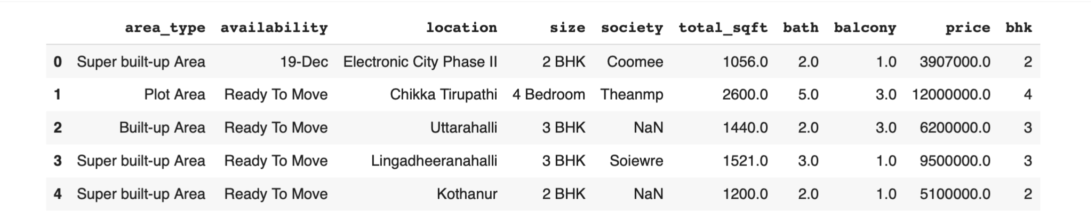
No of Bedrooms
3

Figure 4.2: webpage with details entered



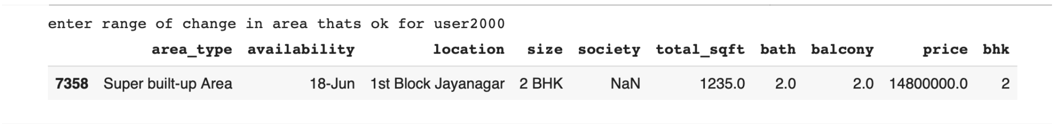
```
Enter the location1st Block Jayanagar
Enter required sq ft2000
enter no of bathrooms2
enter the no of bedrooms2
The predicted price for this loaction is  24516040.975214474 ₹
```

Figure 4.3: Taking input values and predictig price



	area_type	availability	location	size	society	total_sqft	bath	balcony	price	bhk
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056.0	2.0	1.0	3907000.0	2
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600.0	5.0	3.0	12000000.0	4
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440.0	2.0	3.0	6200000.0	3
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521.0	3.0	1.0	9500000.0	3
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200.0	2.0	1.0	5100000.0	2

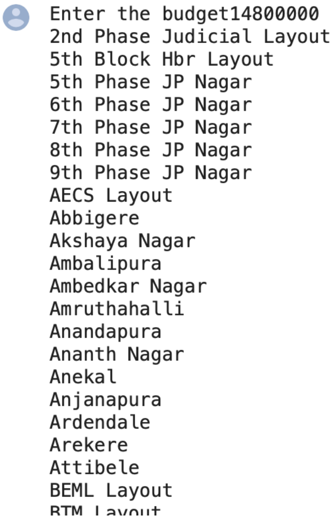
Figure 4.4: retrieving and showing data to the user based on the given data set



```
enter range of change in area thats ok for user2000
```

	area_type	availability	location	size	society	total_sqft	bath	balcony	price	bhk
7358	Super built-up Area	18-Jun	1st Block Jayanagar	2 BHK	NaN	1235.0	2.0	2.0	14800000.0	2

Figure 4.5: list of properties close to given size and in specified place



```
Enter the budget14800000
2nd Phase Judicial Layout
5th Block Hbr Layout
5th Phase JP Nagar
6th Phase JP Nagar
7th Phase JP Nagar
8th Phase JP Nagar
9th Phase JP Nagar
AECS Layout
Abbigere
Akshaya Nagar
Ambalipura
Ambedkar Nagar
Amruthahalli
Anandapura
Ananth Nagar
Anekal
Anjanapura
Ardendale
Arekere
Attibele
BEML Layout
BTM Layout
```

Figure 4.6: user inputs the budget and areas he can afford are shown

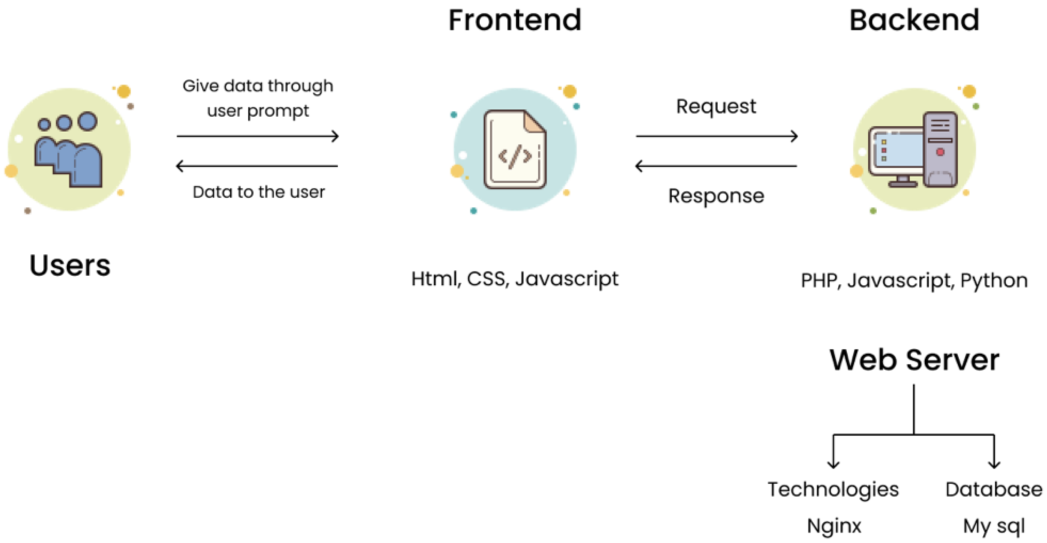


Figure 4.7: process with user interaction

CHAPTER 5

Software model

5.1 Waterfall Model

The first Process Model to be introduced was the Waterfall Model. A linear-sequential life cycle model is another name for it.[20] It is really easy to comprehend and use. Each phase must be finished before the next can begin in a waterfall model, and the phases do not overlap. The Waterfall model is the most basic SDLC approach for software development.

The waterfall model depicts the software development process as a sequential flow of events. This indicates that any step of the development process can start only after the previous one has finished.[21] The phases in this waterfall model do not overlap.

The Waterfall methodology is most appropriate in the following situations:
The requirements are well-documented, explicit, and well-defined.
The product definition is consistent.
Technology is well-understood and unchanging.
There are no prerequisites that are unclear.
The product is supported by a large number of resources with the necessary experience.
The task is short.

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be created with deadlines for each step of development, and a product can be guided through the various phases of the development process one by one.

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

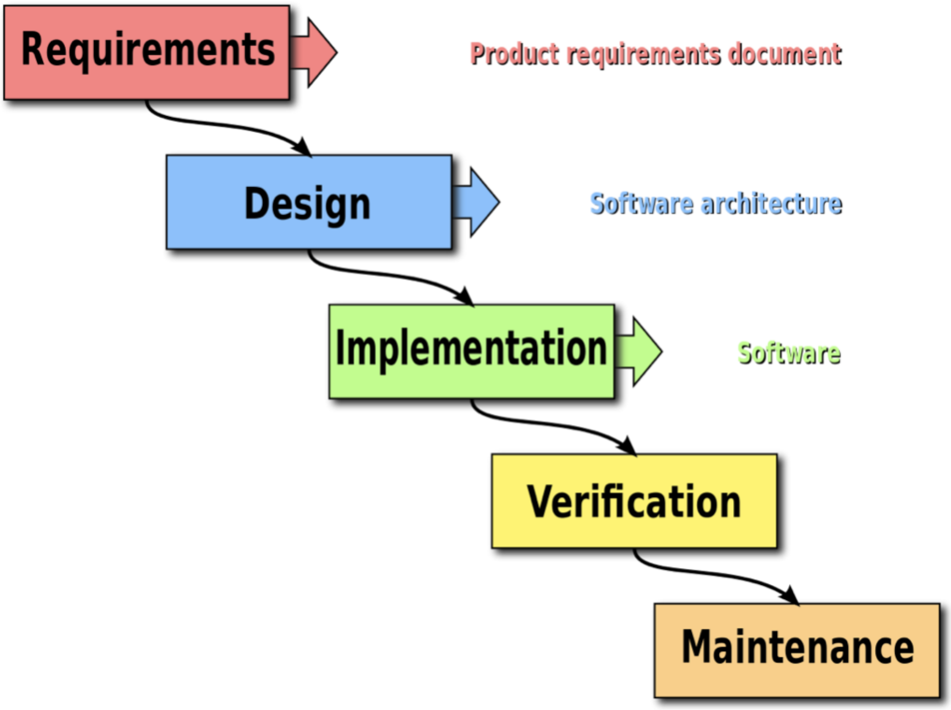


Figure 5.1: Process model- Waterfall

CHAPTER 6

Conclusions and Future Scope

6.1 Conclusion

Every person's dream is to own their own home. We want individuals to acquire houses and real estate at their rightful pricing using this proposed approach, and we don't want them to be duped by shady agents looking for a quick buck. Furthermore, this model will assist large corporations in setting pricing by providing precise predictions, saving them time and money. The essence of the market is correct real estate values, which we seek to secure with this approach. The system is capable of self-training and price prediction using the raw data presented to it. After reviewing various research papers as well as numerous blogs and articles, a set of algorithms was chosen that could be used on both of the model's datasets. After extensive testing and training, it was established that the Ridge regre Algorithm produced the greatest results of all the algorithms. The system was capable of handling massive amounts of data and was powerful enough to predict the prices of numerous houses with diverse features. The method is simple to use and saves time.

6.1.1 Future scope

The additional feature that can be added to our proposed system is to provide users with a full-featured user interface with many features for users to utilise with the ML model in multiple locations. Additionally, an Amazon EC2 connection will enhance the system's functionality and convenience of use. Finally, the project will be completed by creating a well-integrated online application that can estimate prices whenever users want it to.

The used pre-processing methods do help in the prediction accuracy. However, experimenting with different combinations of pre-processing methods to achieve better prediction accuracy.

The factors that have been studied in this study has a weak correlation with the sale price. Hence, by adding more factors to the local dataset that affect the house price,such as GDP, average income, and the population, mortgage value, nearby amenities, and neighborhood growth. In order to increase the number of factors that have an impact on house prices. This could also lead to a better finding.

REFERENCES

[1] Alisha Kuvalekar, Shivani Manchewar, Sidhika Mahadik, and Shila Jawale. "House Price Forecasting Using Machine Learning". In: *Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST)*. 2020.

[2] M Thamarai and SP Malarvizhi. "House Price Prediction Modeling Using Machine Learning." In: *International Journal of Information Engineering & Electronic Business* 12.2 (2020).

[3] Krzysztof Chrostek, Katarzyna Kopczewska, et al. "Spatial prediction models for real estate market analysis". In: *Ekonomia* 35.0 (2013).

[4] Da-Ying Li, Wei Xu, Hong Zhao, and Rong-Qiu Chen. "A SVR based forecasting approach for real estate price prediction". In: *2009 International conference on machine learning and cybernetics*. Vol. 2. IEEE. 2009, pp. 970–974.

[5] Thuraiya Mohd, Suraya Masrom, and Noraini Johari. "Machine learning housing price prediction in Petaling Jaya, Selangor, Malaysia". In: *Int. J. Recent Technol. Eng* 8.2 (2019), pp. 542–546.

[6] Robert N Lussier. "A success versus failure prediction model for the real estate industry". In: *American Journal of Business* (2005).

[7] Raja Manjula, Shubham Jain, Sharad Srivastava, and Pranav Rajiv Kher. "Real estate value prediction using multivariate regression models". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 263. 4. IOP Publishing. 2017, p. 042098.

[8] Anurag Sinha. "Utilization Of Machine Learning Models In Real Estate House Price Prediction". In: ().

[9] Jian Guan, Donghui Shi, Jozef M Zurada, and Alan S Levitan. "Analyzing massive data sets: an adaptive fuzzy neural approach for prediction, with a real estate illustration". In: *Journal of organizational computing and electronic commerce* 24.1 (2014), pp. 94–112.

[10] Neelam Shinde and Kiran Gawande. "Valuation of house prices using predictive techniques". In: *Journal of Advances in Electronics Computer Science* 5.6 (2018), pp. 34–40.

[11] Jingyi Mu, Fang Wu, and Aihua Zhang. "Housing value forecasting based on machine learning methods". In: *Abstract and Applied Analysis*. Vol. 2014. Hindawi. 2014.

[12] Sayan Putatunda. "PropTech for Proactive Pricing of Houses in Classified Advertisements in the Indian Real Estate Market". In: *arXiv preprint arXiv:1904.05328* (2019).

[13] Chad Cowden, Frank J Fabozzi, and Abdolreza Nazemi. "Default prediction of commercial real estate properties using machine learning techniques". In: *The Journal of Portfolio Management* 45.7 (2019), pp. 55–67.

[14] Jian-Guo Liu, Xiao-Li Zhang, and Wei-Ping Wu. "Application of fuzzy neural network for real estate prediction". In: *International Symposium on Neural Networks*. Springer. 2006, pp. 1187–1191.

[15] M Vignesh, V Vijay, S Krishna, and K Sathyamoorthy. "HOUSE PRICE PREDICTION USING MACHINE LEARNING BY RANDOM FOREST ALGORITHM". In: ().

[16] Wei-Xing Zhou and Didier Sornette. "Analysis of the real estate market in Las Vegas: Bubble, seasonal patterns, and prediction of the CSW indices". In: *Physica A: Statistical Mechanics and its Applications* 387.1 (2008), pp. 243–260.

[17] Bindu Sivasankar, Arun P Ashok, and Gouri Madhu. "Fousiya S," House Price Prediction". In: *International Journal of Computer Science and Engineering (IJCSE)* 8.7 (2020).

[18] Aswin Sivam Ravikumar. "Real estate price prediction using machine learning". PhD thesis. Dublin, National College of Ireland, 2017.

[19] Mr Rushikesh Naikare, Mr Girish Gahandule, Mr Akash Dumbre, and Mr Kaushal Agrawal. "Prof. Chaitanya Manka," House Planning and Price Prediction System using Machine Learning". In: *International Engineering Research Journal* 3.3 (2019).

[20] Ioannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. "Reconstructing the house from the ad: Structured prediction on real estate classifieds". In: *EACL2017, the 15th Conference on the European Chapter of the Association for Computational Linguistics*. 2017, pp. 1–6.

[21] Daoyuan Sun, Ce Zhang, Wei Xu, Mei Yun Zuo, Junjie Zhou, and Yudie Du. "Does web news media have opinions? Evidence from Real Estate Market Prediction". In: (2014).