Domain: Machine Learning

Technology: Python

HOUSE PRICE PREDICTION USING MACHINE LEARNING

Domain: Machine Learning

Technology: Python

OBJECTIVE:

The main objective of our project is to reduce the problems faced by the customer. In the present situation, the customer visits a real estate agent so that he/she can suggest suitable showplaces for his investments.

ABSTRACT:

Generally, House prices increases every year so this makes difficult for customers to evaluate house price based on existing values at the time of buying. This motivated us to propose a system that predicts the price of the houses using various Machine Learning algorithms such as AdaBoost, XGBoost, LightGBM&LogisticRegressionalgorithm s.The prediction is done using the features of the previously sold houses over the years.Using the interior and exterior features of the house, the appropriate price can be learned by the algorithms.The algorithm is trained and the price of the house in the near future can be predicted.The prediction would be very much useful in real estate fields where the customers can check the appropriate price of the house before even visiting the site of the house.

Keywords:

- House price prediction
- Machine learning
- Linear regression
- Supervised learning
- Model

Domain: Machine Learning

Technology: Python

INTRODUCTION

Data is at the heart of technical innovations, achieving any result is now possible using predictive models. Machine learning is extensively used in this approach. Machine learning means providing valid dataset and further on predictions are based on that, the machine itself learns how much importance a particular event may have on the entire system based on its pre-loaded data and accordingly predicts the result. Various modern applications of this technique include predicting stock prices, predicting the possibility of an earthquake, predicting company sales and the list has endless possibilities . For our research project, we have considered Bengaluru as our primary location and are predicting real-time house prices for various localities in and around Bengaluru. We have used parameters like 'square feet area', 'no. of Bedrooms', 'No of Bathrooms', etc. We have taken into account a verified dataset with diversity so as give accurate results for all conditions and develop a real estate valuation model which predicts the value of a property using the domain of Machine Learning. The algorithmic approach involves usage ridge regression on top of linear regression approach (Supervised Learning). We use various regression techniques in this pathway, and our results are not sole determination of one technique rather it is the weighted mean of various techniques to give most accurate results. The results proved that this approach yields minimum error and maximum accuracy than individual algorithms applied.

LITERATURE SURVEY

Housing prices indicate the current economic situation and also are a concern to the buyers and sellers. There are many factors that have an impact on house prices, such as the number of bedrooms and bathrooms, House price depends upon its location as well. Predicting house pricesmanually is a difficult task and generally not very accurate, hence there aremany systems developed for house price prediction.

SYSTEM ANALYSIS :-

EXISTING METHOD :-

In the existing system, all the work is done manually. People's need to submit their documents on time and must be submitted through consultors or managers by which they sometimes not able to find for particular price on time. All working personel within department involved just for doing the same task which is document verification and there may be a chance in which the best one may be left behind house.

PROPOSED SYSTEM :-

In our Proposed system, we make use of machine learning algorithms such as XGBoost, LightGBM in order to carry out the various phases of house price

prediction such as Data preprocessing, Training the model and testing the model. Here, the dataset is initially preprocessed by removing the unwanted data, null values and labeling textual values. The preprocessed data is split into two for the purpose of testing and training, which is then sent through the applied machine learning algorithms for training the model with the data and then testing with the test data. Finally, the real time data is provided to the algorithm for predicting the price of thehouse.

DRAWBACKS :-

- It doesn't predict future prices of the houses mentioned by the customer **.**
- Due to this, the risk in investment in an apartment or an area increases considerably.
- To minimize this error, customers tend to hire an agent which again increases the cost of the process.

METHODOLOGY AND ALGORITHMS:- Methodology :-

The selling price is estimates using by considering various parameters such as population rate in particular area, distance to roadways, property age etc. The dataset collection is taken from a standard source such that 80 parameters along with 1000's of test and training data are considered for property valuation and separate dataset is considered for testing and training a model. For further improvement of accuracy, Ridge regularization is applied on top of linear regression so that data are regularized with increase in model accuracy. Users who are going to sell the property can get the accurate values based on this regression prediction. Users requires no intermediate person (broker) to sell in the entity. The python language with its standard libraries are utilized for model expectations dependent on dataset esteem. Since end- user can't run this model each and every time by utilizing python idle there comes the usability lab. To overcome this as well as for powerful utilization of this model by end-users a separate site page is structured with the goal that clients can legitimately pass esteems from site to python code and get the exact value for the entity .

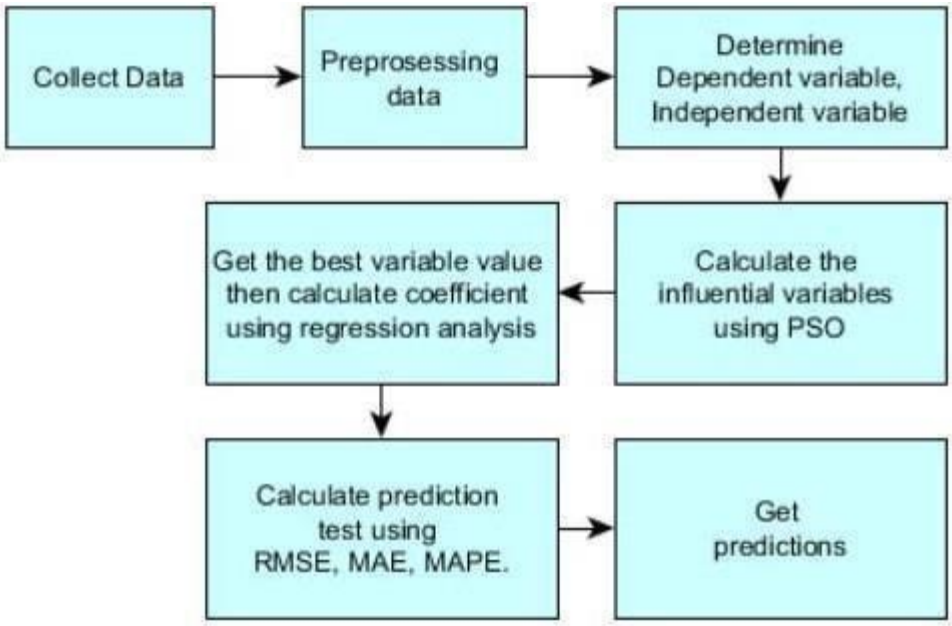
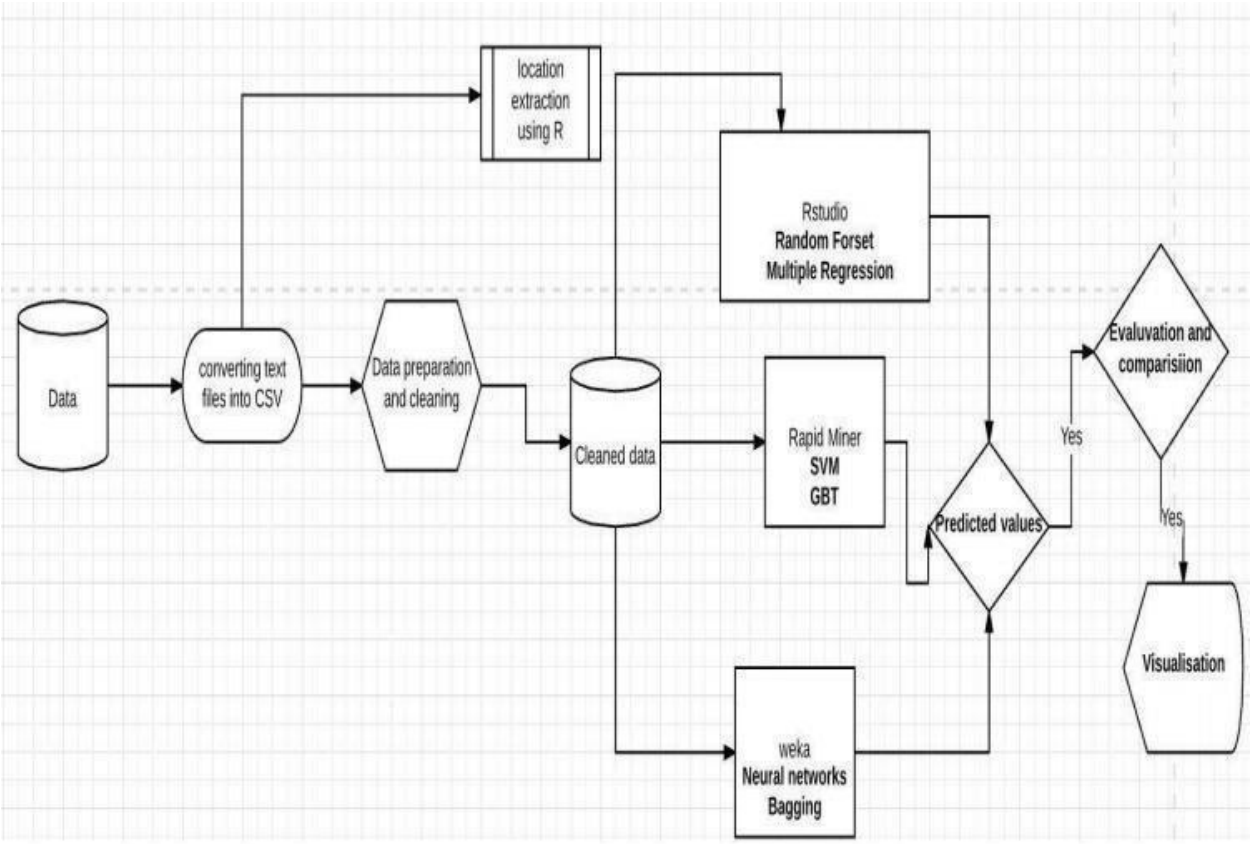


Fig. 1. Diagram flow research.

Architecture :-

This new model will help the new purchasers and less experienced clients tocomprehend the pace of the property that are over-appraised or under-evaluated. Presently, the cost of theproperty rely upon parameters of the land in the monetary framework and the public.We have thought about different basic parameters, (for example, number of rooms, living zone and so forth). At that point these parameter esteems are applied in Linear Regressor model calculations. We have estimated direct linear regression is applied to anticipate the selling pace of an entity.



In this methodology we are foreseeing house value esteems utilizing Linear relapse with edge regularization way to deal with decline the blunder inactivity and furthermore for examination dependent on different mistake measurements, for example, Mean Absolute Error (MAE), Mean Squared Error (MSE), R- Squared worth and Root Mean Squared Error (RMSE).

Algorithm's :-

1. Support Vector Regression
2. Random Forest Regression
3. Linear Regression

Support Vector Machines (SVMs) for Regression :-

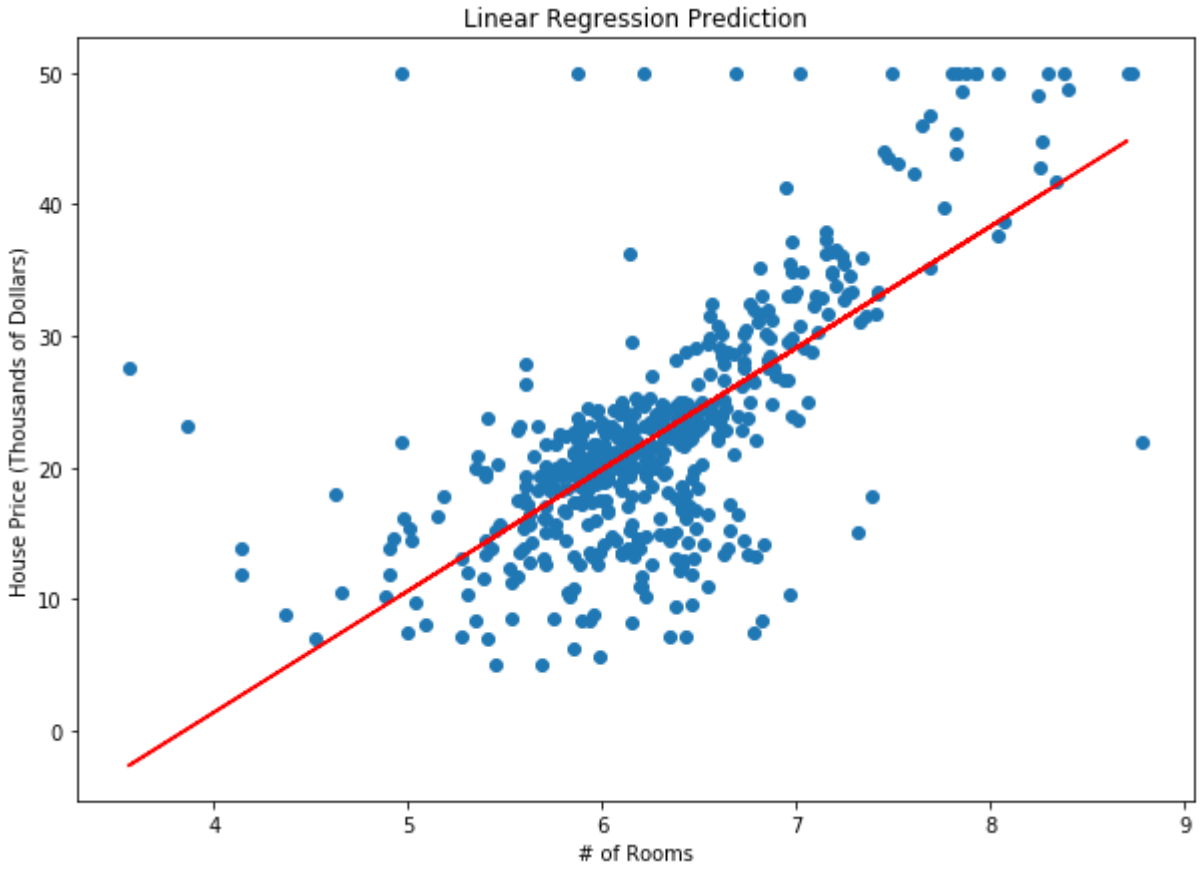
Support Vector Machines (SVMs) are well known in classification problems. The use of SVMs in regression is not as well documented, however. These types of models are known as Support Vector Regression (SVR).

In this, I will walk through the usefulness of SVR compared to other regression models, do a deep-dive into the math behind the algorithm, and provide an example using the Boston Housing Price dataset.

Simple Linear Regression :-

In most linear regression models, the objective is to minimize the sum of squared errors. Take Ordinary Least Squares (OLS) for example. The objective function for OLS with one predictor (feature) is as follows:where y_i is the target, w_i is the coefficient, and x_i is the predictor (feature).

Lasso, Ridge, and ElasticNet are all extensions of this simple equation, with an additional penalty parameter that aims to minimize complexity and/or reduce the number of features used in the final model.



OLS Prediction of Boston Housing Prices

Regardless, the aim — as with many models — is to reduce the error of the test set.

In Python, scikit-learn is a widely used library for implementing Machine Learning Algorithms.

SVM is also available in the scikit-learn library.

```
from sklearn import svm
import pandas as pd
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')
```

Using this libraries we could implement SVM model to predict our ratings using support vectors.

RANDOM FOREST ALGORITHM:-

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

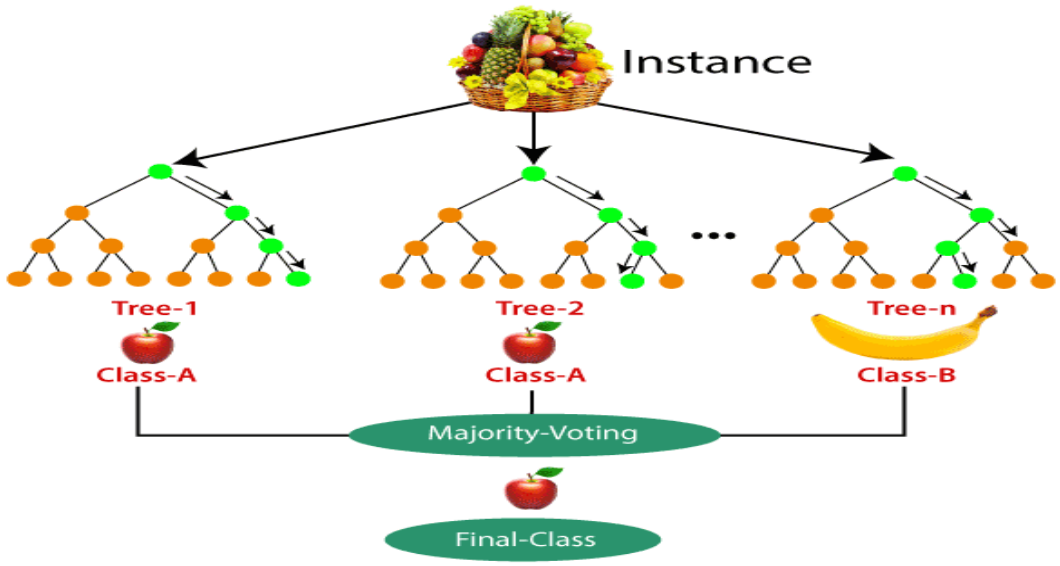


Fig 3.41 Random Forest Classifier Instance

Linear Regression :.

Linear Regression: Linear regression is the most simple method for prediction. It uses two things as variables which are the predictor variable and the variable which is the most crucial one first whether the predictor variable These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables. The equation of the regression equation with one dependent and one independent variable is defined by the formula [8]. b = y + x*a where, b = estimated dependent variable score, y = constant, x = regression coefficient, and a = score on the independent variable.

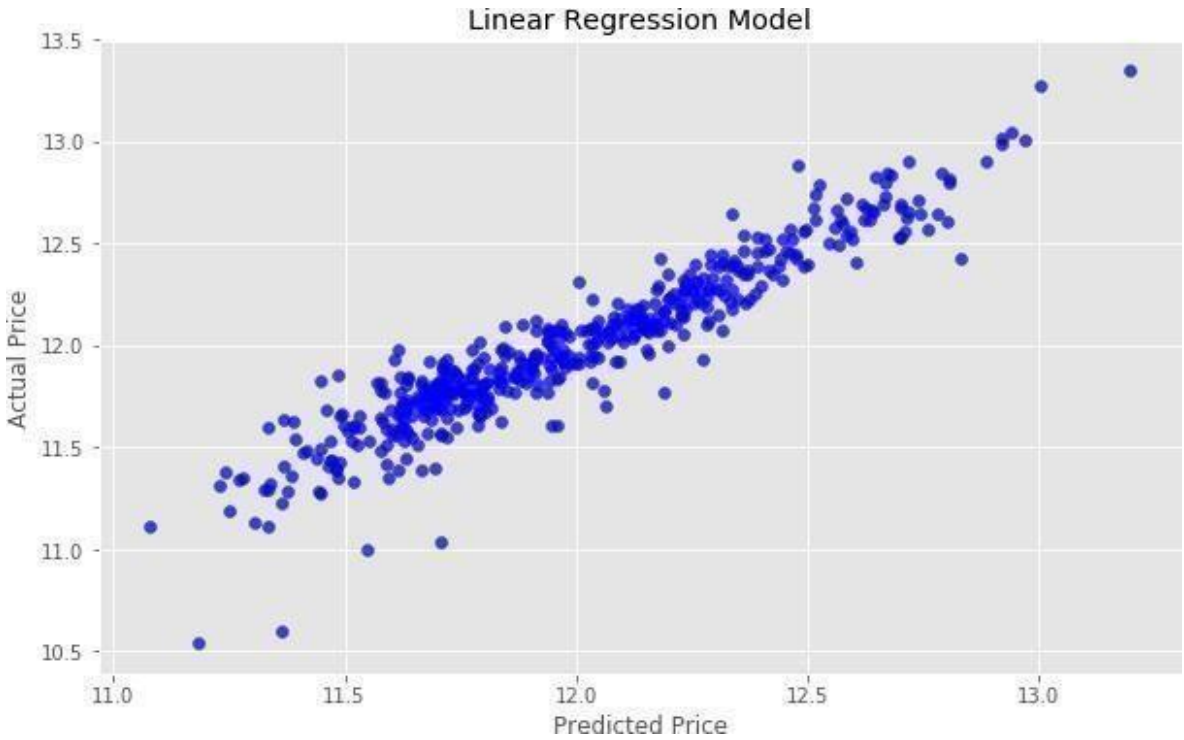


Fig:- Linear Regression Model

SYSTEM REQUIREMENTS SPECIFICATION :-

Hardware System Configuration:-

- Processor - I3/Intel Processor
- RAM - 4GB (min)
- Hard Disk - 160GB

Software System Configuration:-

- • Input device : Keyboard or touch screen Display Display Resolution :1024 X 768 or higher.
- Storage :330 MB (Initial Download)
- Other requirements : Support for Graphical User interface

SYSTEM DESIGN:

PYTHON LANGUAGE: -

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is used to create web and desktop applications, and some of the most popular web applications like Instagram, YouTube, Spotify all have been developed in Python. You can also develop the next big thing by using Python.

Domain: Machine Learning

Technology: Python

Python is open-source, which means that anyone can download it freely from www.python.org and use it to develop programs. Its source code can be accessed and modified as required in the project.

DATASET:-

The real estate markets, like those in Sydney and Melbourne, present an interesting opportunity for data analysts to analyze and predict where property prices are moving towards. Prediction of property prices is becoming increasingly important and beneficial. Property prices are a good indicator of both the overall market condition and the economic health of a country. Considering the data provided, we are wrangling a large set of property sales records stored in an unknown format and with unknown data quality issue .

Implementation :-

Our project HOUSE PRICE PREDICTION is used to recommend movies when a user is interested to watch a particular movie. Our project is divided into 3 modules. Before that we perform 3 major components, these are

1. Data Collection 2. Data Cleaning 3. Data Transfer.

First we collected data from www.kaggle.com in .csv format . As its in raw format the data contains outliers ,noise and missing values. So we need to pre-process the data using various techniques . We mainly replaced null values with either mean or median of the particular column of data and in some cases we replaced null values with 0 as well. Now as the data is in processed as well as useful manner we performed visualization which makes us understand the distribution of data among columns of the dataset. The

system recommend some movies which are highly correlated to the movie that user is interested to watch. This is mainly based on unsupervised K Nearest Neighbours algorithm which is used to calculate the distance between tuples. This algorithm is also called as lazy learner algorithm as it doesn't take training and testing tuple from the dataset. It returns the k nearest or with less distance tuples to a current tuple

The dataset is divided into training dataset and test dataset which contains the classes like Hit, Flop, likes and votes and predicting variables like actor, actress, composer, genre, director producer. We used SVM i.e. Support Vector Machines and Random Forest to predict output variables and considered the algorithm which is more accurate.

Machine learning has also been used for predicting movie success by using algorithms like RF and SVM . Although the use of RF and SVM within the movie domain seems to be fairly limited, the two algorithms have been applied and evaluated in many applications for the purpose of regression as well as classification.

INPUT DESIGN:-

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding − o What are the inputs needed for the system? o How end users respond to different elements of forms and screens.

OBJECTIVES FOR INPUT DESIGN:

The objectives of input design are −

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

OUTPUT DESIGN:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

OBJECTIVES OF OUTPUT DESIGN:

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

OUTPUT SCREEN SHOTS WITH DESCRIPTION :-

we used machine learning algorithms to predict rating of a house price prediction by splitting data into test set and train dataset in 70:30 ratio. We used Random Forest and Support vector machine algorithms to predict ratings and calculated accuracy.

Domain: Machine Learning

Technology: Python

intern.ipynb - Colaboratory

colab.research.google.com/drive/1wFlbJBupKlnjwYK_svrlmei8EOW52BY9

intern.ipynb

File Edit View Insert Runtime Tools Help Last saved at 11:45 AM

Comment Share

+ Code + Text

Connect Editing

```
import numpy as np
```

```
[ ] import pandas as pd
```

```
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor
from sklearn.preprocessing import PolynomialFeatures
```

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

intern.ipynb - Colaboratory

colab.research.google.com/drive/1wFlbJBupKlnjwYK_svrlmei8EOW52BY9

intern.ipynb

File Edit View Insert Runtime Tools Help Last saved at 11:45 AM

Comment Share

+ Code + Text

Connect Editing

```
[ ] drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ] data= pd.read_csv('//content/drive/MyDrive/DATA SET/data.csv')
```

```
data
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	De
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	Bl
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	14
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	85
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	17

Domain: Machine Learning

Technology: Python

intern.ipynb - Colaboratory

colab.research.google.com/drive/1wFlbJBupKlnjwYK_svrlmei8EOW52BY9

intern.ipynb

File Edit View Insert Runtime Tools Help Last saved at 11:45 AM

Comment Share

+ Code + Text

Connect Editing

```
[ ] 4600 rows × 18 columns
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   date           4600 non-null   object
 1   price          4600 non-null   float64
 2   bedrooms       4600 non-null   float64
 3   bathrooms      4600 non-null   float64
 4   sqft_living    4600 non-null   int64
 5   sqft_lot       4600 non-null   int64
 6   floors         4600 non-null   float64
 7   waterfront     4600 non-null   int64
 8   view           4600 non-null   int64
 9   condition      4600 non-null   int64
 10  sqft_above     4600 non-null   int64
 11  sqft_basement  4600 non-null   int64
 12  yr_built       4600 non-null   int64
 13  yr_renovated   4600 non-null   int64
 14  street         4600 non-null   object
 15  city           4600 non-null   object
 16  statezip       4600 non-null   object
 17  country        4600 non-null   object
dtypes: float64(4), int64(9), object(5)
memory usage: 647.0+ KB
```

intern.ipynb - Colaboratory

colab.research.google.com/drive/1wFlbJBupKlnjwYK_svrlmei8EOW52BY9

intern.ipynb

File Edit View Insert Runtime Tools Help Last saved at 11:45 AM

Comment Share

+ Code + Text

Connect Editing

```
[ ]  17  country        4600 non-null   object
     dtypes: float64(4), int64(9), object(5)
     memory usage: 647.0+ KB
```

```
[ ] data.dropna(inplace = True)
```

```
data.isna().sum()
```

```
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
street           0
city             0
statezip         0
country          0
dtype: int64
```

```
[ ] data.drop_duplicates()
```

Domain: Machine Learning

Technology: Python

intern.ipynb - Colaboratory

colab.research.google.com/drive/1wFlbJBupKlnjwYK_svrlmei8EOW52BY9

intern.ipynb

File Edit View Insert Runtime Tools Help Last saved at 11:45 AM

Comment Share

+ Code + Text

Connect Editing

```
data.drop_duplicates()
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	De
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	Bl
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	14
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	85
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	17
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4595	2014-07-09 00:00:00	3.081667e+05	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954	1979	1

intern.ipynb - Colaboratory

colab.research.google.com/drive/1wFlbJBupKlnjwYK_svrlmei8EOW52BY9

intern.ipynb

File Edit View Insert Runtime Tools Help Last saved at 11:45 AM

Comment Share

+ Code + Text

Connect Editing

```
[ ]
```

4600 rows × 18 columns

```
data.describe()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built
count	4.600000e+03	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000
mean	5.519630e+05	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	1827.265435	312.081522	1970.786304
std	5.638347e+05	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	862.168977	464.137228	29.731848
min	0.000000e+00	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	370.000000	0.000000	1900.000000
25%	3.228750e+05	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	1190.000000	0.000000	1951.000000
50%	4.609435e+05	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	1590.000000	0.000000	1976.000000
75%	6.549625e+05	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	2300.000000	610.000000	1997.000000
max	2.659000e+07	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	9410.000000	4820.000000	2014.000000

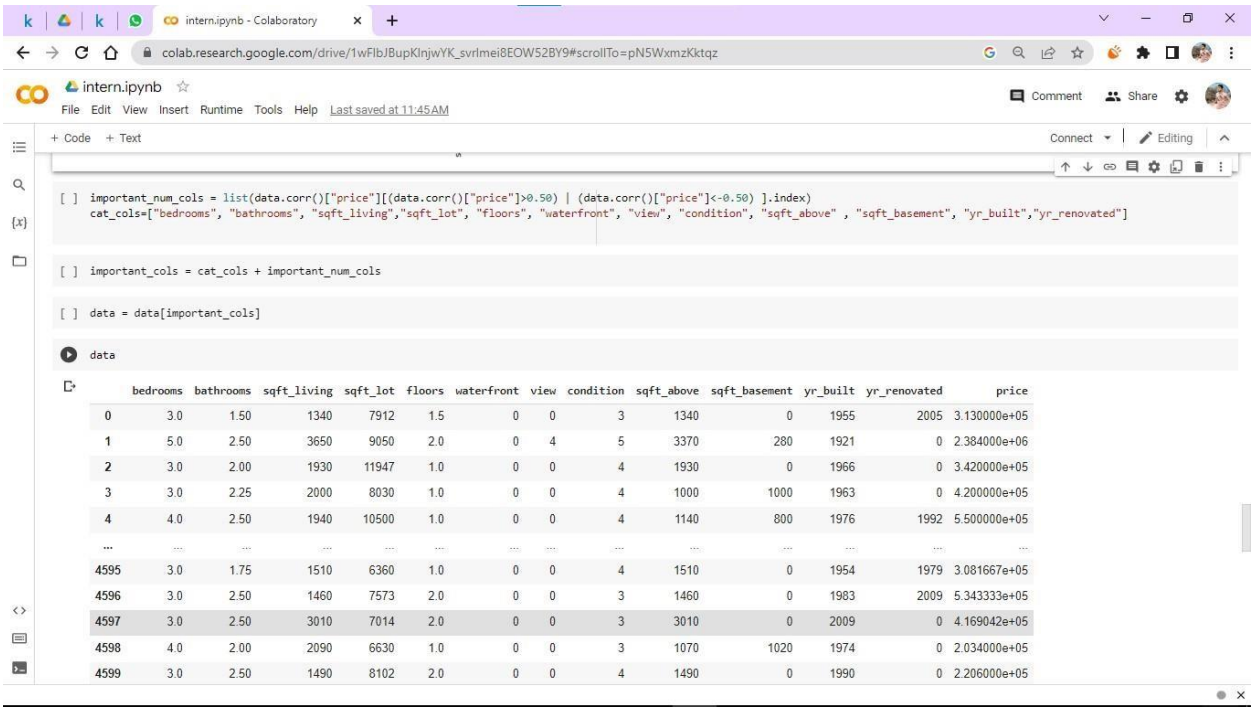
+ Code + Text

```
[ ] data.head()
```

date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street

Domain: Machine Learning

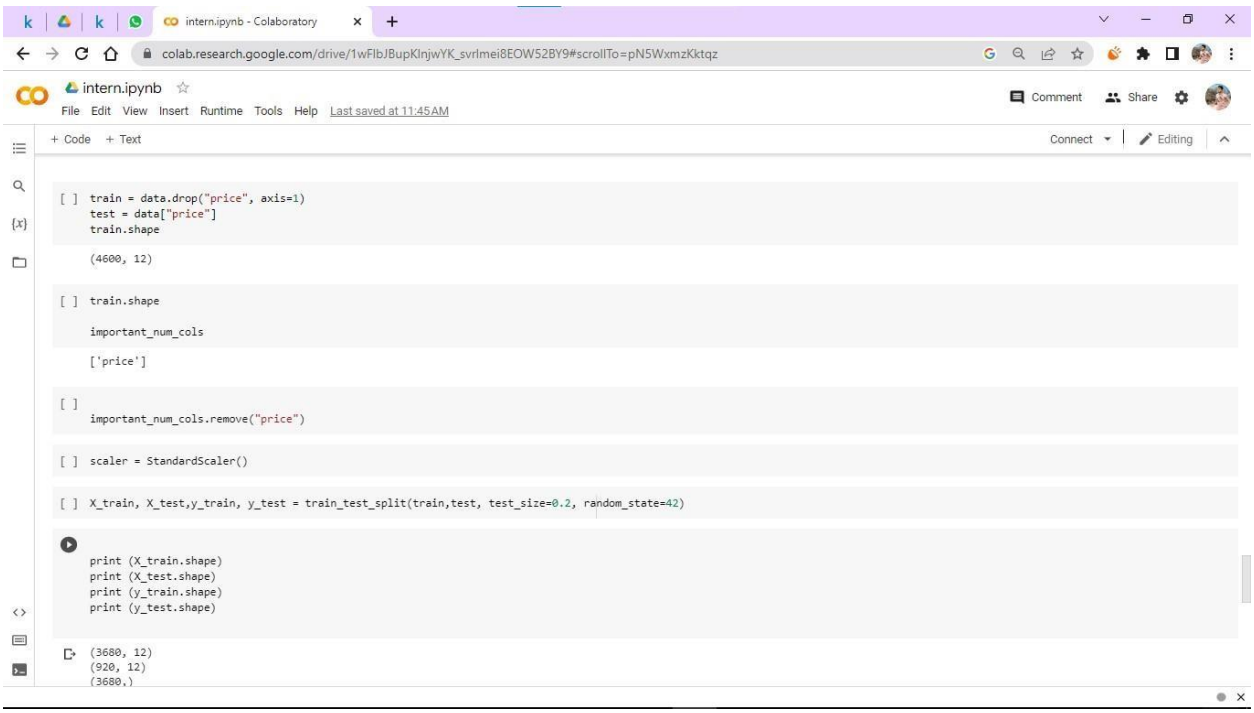
Technology: Python



	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	price
0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	3.130000e+05
1	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	2.384000e+06
2	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	3.420000e+05
3	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	4.200000e+05
4	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	5.500000e+05
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4595	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954	1979	3.081667e+05
4596	3.0	2.50	1460	7573	2.0	0	0	3	1460	0	1983	2009	5.343333e+05
4597	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	2009	0	4.169042e+05
4598	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	1974	0	2.034000e+05
4599	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	1990	0	2.206000e+05

Domain: Machine Learning

Technology: Python



```
train = data.drop("price", axis=1)
test = data["price"]
train.shape
```

```
(4600, 12)
```

```
train.shape

important_num_cols
```

```
['price']
```

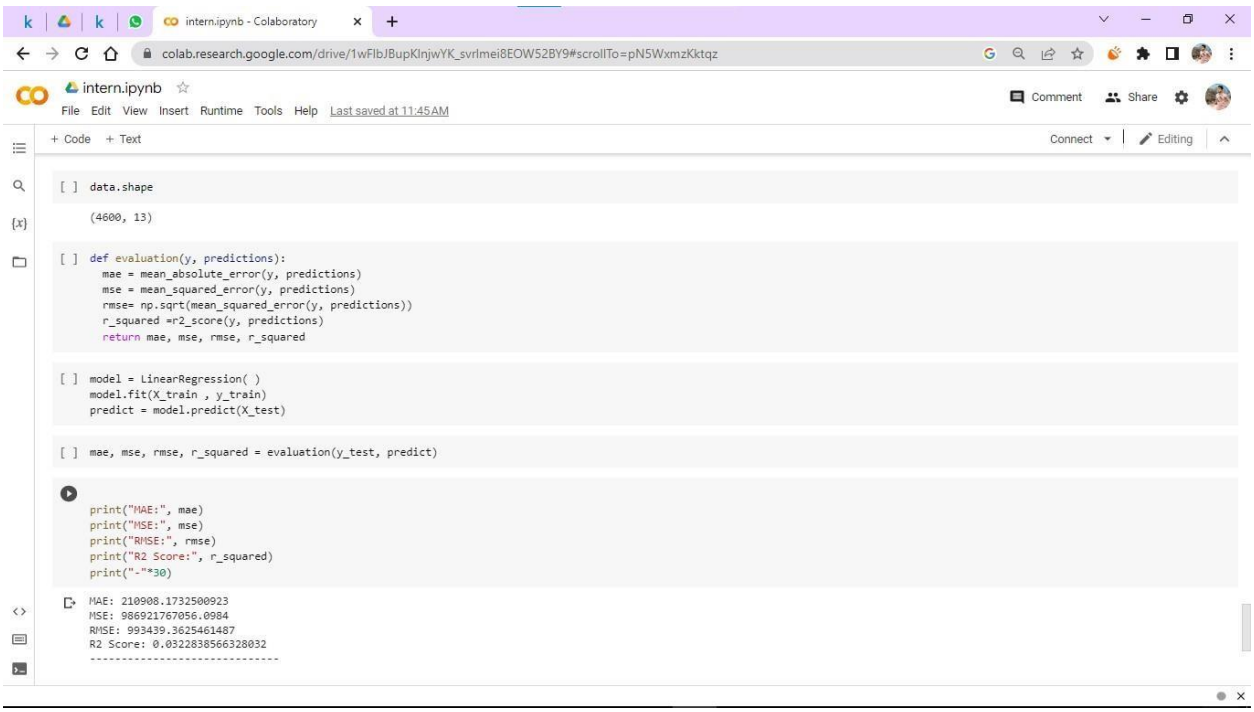
```
important_num_cols.remove("price")
```

```
scaler = StandardScaler()
```

```
X_train, X_test,y_train, y_test = train_test_split(train,test, test_size=0.2, random_state=42)
```

```
print (X_train.shape)
print (X_test.shape)
print (y_train.shape)
print (y_test.shape)
```

```
(3680, 12)
(920, 12)
(3680,)
```



```
data.shape
```

```
(4600, 13)
```

```
def evaluation(y, predictions):
  mae = mean_absolute_error(y, predictions)
  mse = mean_squared_error(y, predictions)
  rmse= np.sqrt(mean_squared_error(y, predictions))
  r_squared =r2_score(y, predictions)
  return mae, mse, rmse, r_squared
```

```
model = LinearRegression( )
model.fit(X_train , y_train)
predict = model.predict(X_test)
```

```
mae, mse, rmse, r_squared = evaluation(y_test, predict)
```

```
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
```

```
MAE: 210908.1732500923
MSE: 986921767056.0984
RMSE: 993439.36254461487
R2 Score: 0.0322838566328032
------------------------------
```

View results: In this page bidder can view the status of their tender

FUTURE SCOPE :-

Future work on this study could be separated into four fundamental regions to further develop the outcome even further. This should be possible by:-

- The utilized pre-handling strategies truly do help in the forecast exactness. Nonetheless, exploring different avenues regarding various blends of pre-handling strategies to accomplish better expectation exactness.

- Utilize the accessible elements and assuming they could be joined as binning highlights has shown that the information gotmoved along.

- Preparing the datasets with various relapse strategies, for example, Elastic net relapse that consolidates both L1 and L2 standards. To grow the examination and check the execution.

CONCLUSION :-

A system that aims to provide a reliable prediction of housing prices based on test data has been developed. The system makes use of both Linear Regression and Ridge Regularization. The system will get the user parameter values directly from webpage and projects the output based on the trained data. The system will satisfy customers by providing accurate output and preventing the risk of investing in the wrong house. Additional features for the customer's benefit can also be added to the system without disturbing it's core functionality. A major future update could be the addition of larger cities to the database, which will allow our users to explore more houses and which will permit the users to investigate more house datasets, commercial places and to get more precision and consequently go to an appropriate choice