

**“BOSTON HOUSE PRICE PREDICTION SYSTEM”**

Sixth Semester Mini Project Report as part of Machine Learning (TCE12)

Submitted to Ramaiah Institute of Technology

(Autonomous Institute Affiliated to VTU, Belgaum)

in partial fulfillment of the requirements for the award of

BACHELOR OF ENGINEERING

In

TELECOMMUNICATION ENGINEERING

For the Academic Year 2019-2020

**Submitted By**

PRAGATHI P 1MS17TE036

RAMYA U S 1MS17TE040

SUPRIYA G 1MS17TE057

**Under the guidance of**

**Dr. Shobha K R**

Associate Professor

Dept. of Telecommunication Engg,

MSRIT,Bangalore 560054

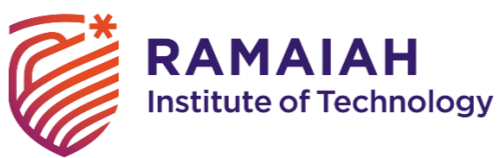
**DEPARTMENT OF TELECOMMUNICATION ENGINEERING,**

**RAMAIAH INSTITUTE OF TECHNOLOGY,**

(Autonomous Institute affiliated to VTU),

**BANGALORE 560054**

**May 2020**



**RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute Affiliated to VTU)**

**Vidya Soudha, Jnana Gangothri MSR Nagar,**

**Bangalore- 560 054, Karnataka**

**Department of Telecommunication Engineering**

**CERTIFICATE**

This is to certify that the Mini Project work entitled **“Boston house price prediction system” was** carried out by Pragathi P (1MS17TE036), Ramya U S (1MS17TE040), Supriya G(1MS17TE057) a bonafide student of Ramaiah Institute of Technology, Bangalore, in partial fulfillment for the award of Bachelor of Engineering in Telecommunication Engineering, of the Visvesvaraya Technological University, Belgaumduring the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report. The Mini Project Report has been approved as it satisfies the academic requirements in respect of Mini Project work.

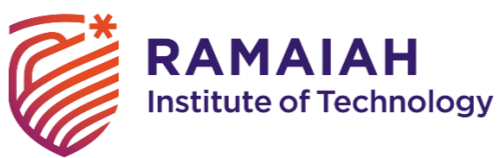
**Internal guide**

**Dr.Shobha K R Dr. B K Sujatha**

Associate Professor Professor and Head

Dept of TC Engg. Dept. of TC Engg,

RIT RIT



**RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute Affiliated to VTU)**

**Vidya Soudha, Jnana Gangothri MSR Nagar,**

**Bangalore- 560 054, Karnataka**

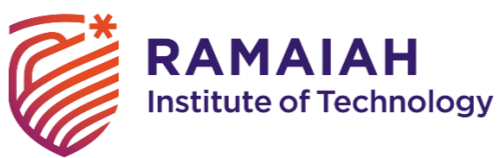
**Department of Telecommunication Engineering**

## **DECLARATION**

We hereby declare that the mini-project entitled “BOSTON HOUSE PRICE PREDICTION SYSTEM” has been carried out independently by us, under the guidance of Dr. SHOBHA K.R, Associate Professor, Telecommunication Engineering, Ramaiah Institute of Technology, Bangalore*.*

Place: Bangalore

Date: 27/04/2020



**RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute Affiliated to VTU)**

**Vidya Soudha, Jnana Gangothri MSR Nagar,**

**Bangalore- 560 054, Karnataka**

**Department of Telecommunication Engineering**

**ACKNOWLEDGEMENT**

It is my profound gratitude that I express my indebtedness to all who have guided me to complete this mini project successfully.

I am grateful to my HOD **Dr. B.K.Sujatha** for allowing me to undertake this mini project work and also providing me with support and sharing his knowledge whenever needed.

I am thankful to my principal **Dr. N.V.R Naidu** for his guidance and support to complete my mini project.

The valuable guidance, the exemplary support and timely suggestions made available to me by my guide **Dr. Shobha K.R.**, Associate Professor, Telecommunication dept, RIT went a long way in completion of the mini project. I sincerely acknowledge her help, guidance and constant support which were ever present throughout the mini-project work.

I also thank my friends and the staff members of Telecommunication dept. and also my family for the help and support provided by them in successful completion of the mini project. I would also like to thank the other members of the lab, workplace and my friends for being there for me during my hardships and creating an amiable atmosphere to work in.

My accomplishments would be incomplete without my beloved parents, for without their support and encouragement I would not have reached up to this level. I owe my achievements to them.

Pragathi P(1MS17TE036)

Ramya U S(1MS17TE040)

Supriya G(1MS17TE057)

**ABSTRACT**

Investment is a business activity on which most people are interested in this globalization era. There are several objects that are often used for investment, for example, gold, stocks and property. In particular, property investment has increased significantly. Housing price trends are not only the concern of buyers and sellers, but it also indicates the current economic situation. There are many factors which has impact on house prices, such as numbers of bedrooms and bathrooms. Even the nearby location, a location with a great accessibility to highways, expressways, schools, shopping malls and local employment opportunities contributes to the rise in house price. Manual house predication becomes difficult, hence there are many systems developed for house price prediction. In this project we have proposed an advanced house price prediction system using linear regression. This system aim is to make a model which can give us a good house pricing prediction based on other variables. We are going to use Linear Regression for this dataset and hence it gives a good accuracy. In this house price prediction project we are collecting a boston housing data from the suburbs of boston. This dataset has 506 samples and 14 features .Since most of the features in this dataset may not affect efficiently for the price of the house so they are considered as non- relavant features ,Now the dataset is reduced using cross correlation. The reduced housing dataset which has 489 samples and 4 features .The dataset is trained using linear regression model for more accuracy. User can give his required values for the features of the house and predict the price of the house.

**TABLE OF CONTENTS**

1. INTRODUCTION

1.1. Background

1.2. Problem statement

1.3. Project Objective

1.4. Motivation

1.5. Applications

1.6. Advantages

2. LITERATURE REVIEW

3. ALGORITHM

4. IMPLEMENTATION DETAILS

5. RESULTS AND DISCUSSION

6. CONCLUSION AND FUTURE ENHANCEMENTS

7. REFERENCES

**1. INTRODUCTION**

**1.1 Background**

Machine learning plays a major role from past years in image detection, spam reorganization, normal speech command, product recommendation and medical diagnosis. Present machine learning algorithm helps us in enhancing security alerts, ensuring public safety and improve medical enhancements. Machine learning system also provides better customer service and safer automobile systems. In the present project we discuss about the prediction of future housing prices that is generated by machine learning algorithm. We utilize linear regression as our model because of its adaptable and probabilistic methodology on model selection. Our result exhibit that our approach of the issue need to be successful, and has the ability to process predictions that would be comparative with other house cost prediction models.We used the Boston housing dataset from Kaggle which provides housing values in Suburbs of Boston. It contains about fourteen features measuring various factors concerning the areas and houses in Boston.We in that point recommend a housing cost prediction model to support a house vender or a real estate agent for better information based on the valuation of house. Those examinations exhibit that linear regression algorithm, in view of accuracy, reliably outperforms alternate models in the execution of housing cost prediction.

**1.2 PROBLEM STATEMENT**

For price prediction of houses, the data has been formulized as a multiple linear regression which attempts to model the relationship between the predictor variables and the response variable thereby predicting the price of houses by fitting a linear equation to the observed data. Formally a multiple linear regression model with several predictor variables X1, X2, ….,Xk and one response variable Y can be written as



**1.3 PROJECT OBJECTIVE**

The goal of this project is to develop a machine learning model that effectively estimates the price of the houses in Boston. We in that point recommend a housing cost prediction model to support a house vender or a real estate agent for better information based on the valuation of house. Those examinations exhibit that linear regression algorithm, in view of accuracy, reliably outperforms alternate models in the execution of housing cost prediction.

**1.4 MOTIVATION:**

We found this real estate domain to be interesting enough to apply various machine learning algorithms and analyse the performance of our model with respect to each of the algorithms. The reason why we focused on pricing is because this a major factor that plays an important role for people to decide on buying a house. We used the Boston housing dataset from Kaggle which provides housing values in Suburbs of Boston. It contains about fourteen features measuring various factors concerning the areas and houses in Boston.

**1.5 APPLICATIONS**

* A model like this would be very useful for a real estate agent who could make use of this on his daily basis to predict the price of the house.

##### 1.6 ADVANTAGES

* Saves time in predicting the price of the house.
* Easy to access the system from anywhere and anytime.
* Minimizes the loss to an real estate agent by predicting the price range of the house.

**2. LITERATURE REVIEW**

Machine learning is a form of artificial intelligence which is composed available computers with the efficiency to be trained without being veraciously programmed. Machine learning interest on the extensions of computer programs which is capable enough to modify when unprotected to new-fangled data. Machine learning algorithms are broadly classified into three divisions, namely; Supervised learning, Unsupervised learning and Reinforcement learning .Supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with correct answer. After that, machine is provided with new set of examples so that supervised learning algorithm analyses the training data and produces a correct outcome from labelled data. Unsupervised learning is the training of machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance.

Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data. Unlike, supervised learning, no teacher is provided that means no training will be given to the machine. Therefore, machine is restricted to find the hidden structure in unlabelled data by our-self.

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience. Machine learning has many application’s out of which one of the applications is prediction of real estate. The real estate market is one of the most competitive in terms of pricing and same tends to be vary significantly based on lots of factor, forecasting property price is an important modules in decision making for both the buyers and investors in supporting budget allocation, finding property finding stratagems and determining suitable policies hence it becomes one of the prime fields to apply the concepts of machine learning to optimize and predict the prices with high accuracy. The study on land price trend is felt important to support the decisions in urban planning. The real estate system is an unstable stochastic process. Investors decisions are based on the market trends to reap maximum returns. Developers are interested to know the future trends for their decision making.(3)

To accurately estimate property prices and future trends, large amount of data that influences land price is required for analysis, modelling and forecasting. The factors that affect the land price have to be studied and their impact on price has also to be modelled. An analysis of the past data is to be considered. It is inferred that establishing a simple linear mathematical relationship for these time-series data is found not viable for forecasting. Hence it became imperative to establish a non-linear model which can well fit the data characteristic to analyse and forecast future trends. As the real estate is fast developing sector, the analysis and forecast of land prices using mathematical modelling and other scientific techniques is an immediate urgent need for decision making by all those concerned. The increase in population as well as the industrial activity is attributed to various factors, the most prominent being the recent spurt in the knowledge sector viz. Information Technology (IT) and Information technology enabled services.

Demand for land started of showing an upward trend and housing and the real estate activity started booming. All barren lands and paddy fields ceased their existence to pave way for multistore and highrise buildings. Investments started pouring in Real estate Industry and there was no uniform pattern in the land price over the years. The need for predicting the trend in land prices was felt by all in the industry viz. the Government, the regulating bodies, lending institutions, the developers and the investors. Therefore, in this paper, we present various important features to use while predicting housing prices with good accuracy. We can use regression models, using various features to have lower Residual Sum of Squares error. While using features in a regression model some feature engineering is required for better prediction. Often a set of features multiple regressions or polynomial regression (applying a various set of powers in the features) is used for making better model fit. For these models are expected to be susceptible towards over fitting ridge regression is used to reduce it. So, it directs to the best application of regression models in addition to other techniques to optimize the result.(5)

**Multiple Regression Technique** - Sampathkumar

Regression analysis is widely used for forecasting. Regression analysis is used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. If more independent variables are added, it is able to determine an estimating equation that describes the relationship with greater accuracy. Multiple regressions look at each independent variable and test whether it contributes significantly to the way the regression describes the data. The general multiple regression equation is



**Advantages:** The estimates of the unknown parameters obtained from linear least squares regression are the optimal. Estimates from a broad class of possible parameter estimates under the usual assumptions are used for process modelling. It uses data very efficiently. Good results can be obtained with relatively small data sets.

**Disadvantages:** The outputs of regression can lie outside of the range [0,1]. It has limitations in the shapes that linear models can assume over long ranges. The extrapolation properties will be possibly poor. It is very sensitive to outliers. It often gives optimal estimates of the unknown parameters.(6)

**Linear Regression** - MansuralBhuiyan, Mohammad Al Hasan

To establish baseline performance with a linear classifier, we used Linear Regression to model the price targets, Y, as a linear function of the data, X



**Advantage:** A linear model can include more than one predictor as long as the predictors are additive. the best fit line is the line with minimum error from all the points, it has high efficiency but sometimes this high efficiency created.

**Disadvantage:** Linear Regression Is Limited to Linear Relationships. Linear Regression Only Looks at the Mean of the Dependent Variable. Linear Regression Is Sensitive to Outliers. Data Must Be Independent.(7)

**k-Nearest Neighbours (KNN)** -E. Fix and J. L. Hodges Jr

k-Nearest-Neighbour (KNN) is a non-parametric instance-based learning method. In this case, training is not required. The first work on KNN was submitted by Fix & Hodges in 1951 for the United States Air-force. The algorithm begins by storing all the input feature vectors and outputs from our training set. For each unlabelled input feature vector, we find the k nearest neighbours from our training set. The notion of nearest uses Euclidean distance in the m-dimensional feature space. For two input vectors x and w, their distance is defined by:



**Advantages:** The K-Nearest Neighbour (KNN) Classifier is a very simple classifier that works well on basic recognition problems.

**Disadvantage:** KNN algorithm is that it is a lazy learner, i.e. it does not learn anything from the training data and simply uses the training data itself for classification. To predict the label of a new instance the KNN algorithm will find the K closest neighbours to the new instance from the training data, the predicted class label will then be set as the most common label among the K closest neighbouring points. The main disadvantage of this approach is that the algorithm must compute the distance and sort all the training data at each prediction, which can be slow if there are a large number of training examples. Another disadvantage of this approach is that the algorithm does not learn anything from the training data, which can result in the algorithm not generalizing well and also not being robust to noisy data. Further, changing K can change the resulting predicted class label.(8)

**3. ALGORITHM**

1.1 COLLECT BOSTON HOUSING DATASET (DATA EXPLORATION)

1.9 MAKE PREDICTIONS FOR THE GIVEN DATA

1.8 EVALUATE MODEL PERFORMANCE

1.7 GET THE BEST MAX\_DEPTH FOR THE MODEL

1.6 ANALYZE THE MODEL PERFORMANCE WITH DIFFERENT MAX\_DEPTH

1.5 FIT THE REGRESSION LINE TO THE TRAIN AND TEST DATASET

1.4 SPLIT THE DATASET IN TO TEST AND TRAIN DATASET

1.3 LOAD THE REDUCED HOUSING DATASET

1.2 REDUCE THE DATASET USING CROSS CORRELATION(DATA PREPROCESSING)

**FIG 1. FLOW CHART**

**EXPLANATION OF FLOW CHART:**

The above FIG 1 is the flow chart for this project “Boston house price prediction system”. The explanation for this is as follows,

1.1. Collect Boston housing dataset (Data exploration):Boston housing dataset is collected which has 506 samples with 14 features.

1.2. Reduce the dataset using cross correlation: Boston housing dataset is reduced using cross correlation in which the non-relevant features are reduced. Now the housing dataset has only 489 samples with 4 essential features like ‘RM’, ‘LSTAT’, ‘PTRATIO’, ‘MEDV’.

1.3. Load the reduced housing dataset: Load the housing dataset which has 489 samples and the 3 features which mainly affect the price of the house.

1.4. Split the dataset in to test and train dataset: The housing dataset is split in to test and train dataset with test size of 0.2.

1.5. Fit the regression line to the train and test dataset: The regression line is fit separately for both test and train dataset.

1.6. Analyze the model performance with different max\_depth: The model is analyzed with different max\_depth using learning curves..

1.7. Get the best max\_depth for the model: The best max\_depth is obtained for the model using validation curve.

1.8. Evaluate model performance: The model is evaluated to reduce the errors.

1.9. Make predictions for the given data: Prediction of price is made for the clients data.

**4. IMPLEMENTATION DETAILS**

The Boston housing data was collected in 1978 and each of the 506 entries represent aggregated data about 14 features for homes from various suburbs in Boston, Massachusetts. The boston housing dataset is given below.

<https://drive.google.com/file/d/16_R_CAZ0SZCqDzoh68GOMO82gfMTygzd/view?usp=sharing>

For the purposes of this project, the following preprocessing steps have been made to the dataset.

* 16 data points have an 'MEDV' value of 50.0. These data points likely contain **missing or censored values** and have been removed.
* 1 data point has an 'RM' value of 8.78. This data point can be considered an **outlier** and has been removed.
* The features 'RM', 'LSTAT', 'PTRATIO', and 'MEDV' are essential. The remaining **non-relevant features** have been excluded.
* The feature 'MEDV' has been **multiplicatively scaled** to account for 35 years of market inflation.

The reduced housing dataset is shown below,

<https://drive.google.com/file/d/1nosSjaPe4UkBnCa6fzFjRIswxd85mmEm/view?usp=sharing>

* The dataset is split in to test and train dataset with test size of 0.2.
* Regression line is fit for both test and train dataset.
* Then analyze the model performance for different values of max\_depth and get the suitable max\_depth.
* Evaluate the model performance to reduce the errors.
* Predict the price of the house for the clients data.

**CODE:**

# Import libraries necessary for this project

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import ShuffleSplit

from sklearn.metrics import r2\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import learning\_curve as curves

from sklearn.model\_selection import validation\_curve as curvesvalidate

from sklearn.tree import DecisionTreeRegressor

from sklearn.model\_selection import ShuffleSplit

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import make\_scorer

import warnings

warnings.filterwarnings("ignore", category = UserWarning, module = "matplotlib")

###########################################

from sklearn.datasets import load\_boston

#The main dataset with 506 samples and 14 features

boston\_dataset = load\_boston()

boston = pd.DataFrame(boston\_dataset.data, columns=boston\_dataset.feature\_names)

boston['MEDV'] = boston\_dataset.target

print(boston)

correlation\_matrix = boston.corr().round(2)

sns.heatmap(data=correlation\_matrix, annot=True)

# Load the Boston housing reduced dataset of 489 samples and 4 features

data = pd.read\_csv('housing.csv')

print(data)

prices = data['MEDV']

features = data.drop('MEDV', axis = 1)

print("Boston housing dataset has %d data points with %d variables each" %(data.shape[0],data.shape[1]))

#description of the dataset

# Minimum price of the data

minimum\_price = np.min(prices)

# Maximum price of the data

maximum\_price = np.max(prices)

# Mean price of the data

mean\_price = np.mean(prices)

# Median price of the data

median\_price = np.median(prices)

# Standard deviation of prices of the data

std\_price = np.std(prices)

# Show the calculated statistics

print("Statistics for Boston housing dataset:\n")

print("Minimum price: $%.2f" %(minimum\_price))

print("Maximum price: $%.2f" %(maximum\_price))

print("Mean price: $%.2f" %(mean\_price))

print("Median price $%.2f" %(median\_price))

print("Standard deviation of prices: $%.2f" %(std\_price))

# Shuffle and split the data into training and testing subsets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, prices, test\_size=0.2, random\_state=10)

print("Training and testing split was successful.")

#create regression line for training and testing dataset

plt.figure(figsize=(20, 5))

for i, col in enumerate(features.columns):

plt.subplot(1, 3, i+1)

x = X\_train[col]

y = y\_train

plt.plot(x, y, 'o')

# Create regression line

plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)))

plt.title(col)

plt.xlabel(col)

plt.ylabel('prices')

plt.show()

plt.figure(figsize=(20, 5))

for i, col in enumerate(features.columns):

# 3 plots here hence 1, 3

plt.subplot(1, 3, i+1)

x = X\_test[col]

y = y\_test

plt.plot(x, y, 'o')

# Create regression line

plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)))

plt.title(col)

plt.xlabel(col)

plt.ylabel('prices')

plt.show()

# Produce learning curves for varying training set sizes and maximum depths

#Analyze the model performance with different max\_depth

def ModelLearning(X, y, clf):

""" Calculates the performance of several models with varying sizes of training data.

The learning and testing scores for each model are then plotted. """

# Create 10 cross-validation sets for training and testing

cv = ShuffleSplit(X.shape[0], test\_size = 0.2, random\_state = 0)

# Generate the training set sizes increasing by 50

train\_sizes = np.rint(np.linspace(1, X.shape[0]\*0.8 - 1, 9)).astype(int)

# Create the figure window

fig = plt.figure(figsize=(10,7))

# Create three different models based on max\_depth

for k, depth in enumerate([1,3,6,10]):

# Create a Decision tree regressor at max\_depth = depth

regressor = clf(max\_depth = depth)

# Calculate the training and testing scores

sizes, train\_scores, test\_scores = curves(regressor, X, y, cv = cv, train\_sizes = train\_sizes)

# Find the mean and standard deviation for smoothing

train\_std = np.std(train\_scores, axis = 1)

train\_mean = np.mean(train\_scores, axis = 1)

test\_std = np.std(test\_scores, axis = 1)

test\_mean = np.mean(test\_scores, axis = 1)

# Subplot the learning curve

ax = fig.add\_subplot(2, 2, k+1)

ax.plot(sizes, train\_mean, 'o-', color = 'r', label = 'Training Score')

ax.plot(sizes, test\_mean, 'o-', color = 'g', label = 'Testing Score')

ax.fill\_between(sizes, train\_mean - train\_std, train\_mean + train\_std, alpha = 0.15, color = 'r')

ax.fill\_between(sizes, test\_mean - test\_std, test\_mean + test\_std, alpha = 0.15, color = 'g')

# Labels

ax.set\_title('max\_depth = %s'%(depth))

ax.set\_xlabel('Number of Training Points')

ax.set\_ylabel('Score')

ax.set\_xlim([0, X.shape[0]\*0.8])

ax.set\_ylim([-0.05, 1.05])

# Visual aesthetics

ax.legend(bbox\_to\_anchor=(1.05, 2.05), loc='lower left', borderaxespad = 0.)

fig.suptitle('Decision Tree Regressor Learning Performances', fontsize = 16, y = 1.03)

fig.tight\_layout()

plt.show()

ModelLearning(features, prices,DecisionTreeRegressor)

#Complexity curve

#get the best max\_depth for the model

def ModelComplexity(X, y, clf):

""" Calculates the performance of the model as model complexity increases.

The learning and testing errors rates are then plotted. """

# Create 10 cross-validation sets for training and testing

cv = ShuffleSplit(X.shape[0], test\_size = 0.2, random\_state = 0)

# Vary the max\_depth parameter from 1 to 10

max\_depth = np.arange(1,11)

# Calculate the training and testing scores

train\_scores, test\_scores = curvesvalidate(clf(), X, y, \

param\_name = "max\_depth", param\_range = max\_depth, cv = cv, scoring = 'r2')

# Find the mean and standard deviation for smoothing

train\_mean = np.mean(train\_scores, axis=1)

train\_std = np.std(train\_scores, axis=1)

test\_mean = np.mean(test\_scores, axis=1)

test\_std = np.std(test\_scores, axis=1)

# Plot the validation curve

plt.figure(figsize=(7, 5))

plt.title('Decision Tree Regressor Complexity Performance')

plt.plot(max\_depth, train\_mean, 'o-', color = 'r', label = 'Training Score')

plt.plot(max\_depth, test\_mean, 'o-', color = 'g', label = 'Validation Score')

plt.fill\_between(max\_depth, train\_mean - train\_std, train\_mean + train\_std, alpha = 0.15, color = 'r')

plt.fill\_between(max\_depth, test\_mean - test\_std, test\_mean + test\_std, alpha = 0.15, color = 'g')

# Visual aesthetics

plt.legend(loc = 'lower right')

plt.xlabel('Maximum Depth')

plt.ylabel('Score')

plt.ylim([-0.05,1.05])

plt.show()

ModelComplexity(X\_train, y\_train, DecisionTreeRegressor)

#Evaluating model performance,and fit it to reduce the error

def performance\_metric(y\_true, y\_predict):

"""Calculates and returns the performance score between

true and predicted values based on the metric chosen. """

#Calculate the performance score between 'y\_true' and 'y\_predict'

score = r2\_score(y\_true, y\_predict)

#Return the score

return score

def fit\_model(X, y):

""" Performs grid search over the 'max\_depth' parameter for a

decision tree regressor trained on the input data [X, y]. """

# Create cross-validation sets from the training data

cv\_sets = ShuffleSplit(X.shape[0], test\_size = 0.20, random\_state = 0)

#Create a decision tree regressor object

regressor = DecisionTreeRegressor()

#Create a dictionary for the parameter 'max\_depth' with a range from 1 to 10

params = {'max\_depth':np.arange(1,11)}

#Transform 'performance\_metric' into a scoring function using 'make\_scorer'

scoring\_fnc = make\_scorer(performance\_metric)

#Create the grid search object

grid = GridSearchCV(regressor, param\_grid = params, scoring = scoring\_fnc, cv = cv\_sets)

# Fit the grid search object to the data to compute the optimal model

grid = grid.fit(X, y)

# Return the optimal model after fitting the data

return grid.best\_estimator\_

# Fit the training data to the model using grid search

reg = fit\_model(X\_train, y\_train)

print(reg)

# Produce the value for 'max\_depth'

print("Parameter 'max\_depth' is {} for the optimal model.".format(reg.get\_params()['max\_depth']))

#make predictions

# Produce a matrix for client data

client\_data = [[5, 34, 15], # Client 1

[4, 55, 22]] # Client 2

# Show predictions

for i, price in enumerate(reg.predict(client\_data)):

print("Predicted selling price for Client %d's home: $%.2f" %(i+1, price))

#Performs trials of fitting and predicting data.

def PredictTrials(X, y, fitter, data):

# Store the predicted prices

prices = []

for k in range(10):

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = k)

# Fit the data

reg = fitter(X\_train, y\_train)

# Make a prediction

pred = reg.predict([data[0]])[0]

prices.append(pred)

# Result

print("Trial %d: $%.2f" %(k+1, pred))

# Display price range

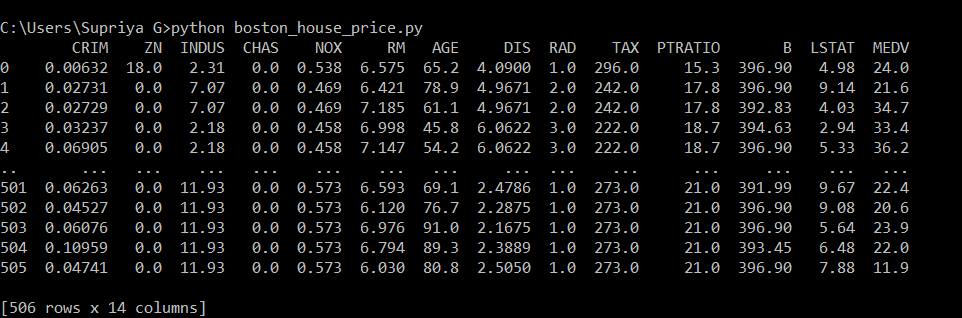
print("\nRange in prices: $%.2f" %(max(prices) - min(prices)))

PredictTrials(features, prices, fit\_model, client\_data)

**5. RESULTS AND DISCUSSION**

The result for the above project “Boston house price prediction system” is discussed below with serious of figures.

1. Initially load the Boston housing dataset:

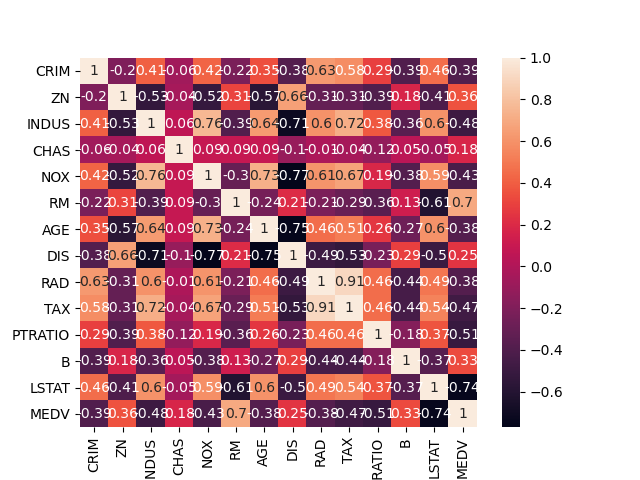


**Fig 5.1: Boston housing dataset**

The above shown figure 5.1 gives us the Boston housing dataset which as 506 samples with 14 features as shown above.

2. Data Preprocessing:

Cross correlation matrix for the above dataset is:

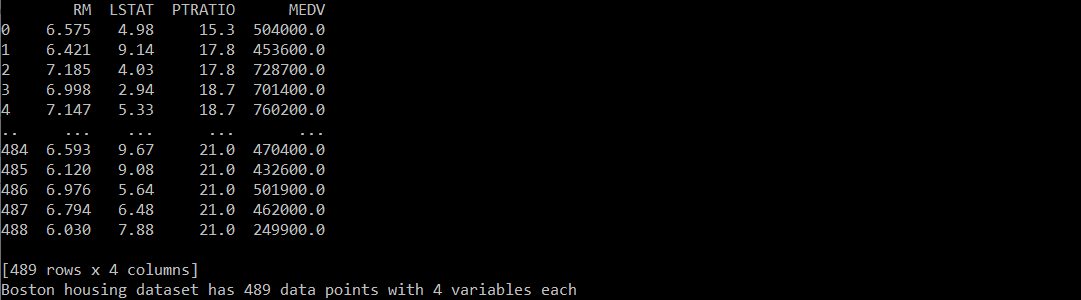


**FIG 5.2: Cross Correlation matrix**

The above figure 5.2 gives us the correlation matrix for the Boston housing dataset. By using the cross correlation matrix we can remove the features which are least dependent on the price of the house.

3. Load the housing dataset:

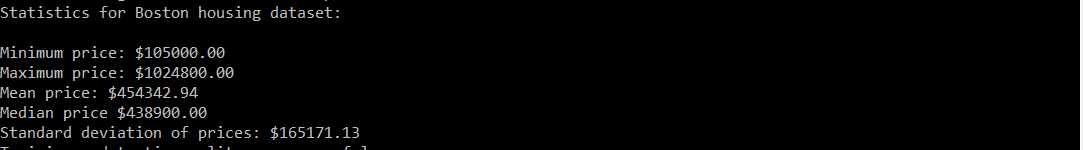
Reduced housing dataset after data preprocessing is..



**Fig 5.3: Housing dataset**

The above fig3 gives us the reduced housing dataset which has only 4 features. Now there are only three features which mainly affect the price of the house.

4. Description of the dataset:



**Fig 5.4: Statistics of dataset**

The above figure 5.4 gives us the statistics of the reduced housing dataset. It gives us the minimum price, maximum price, mean price, median price, standard deviation.

5. Splitting of dataset:

The dataset is split with test size of 0.2.

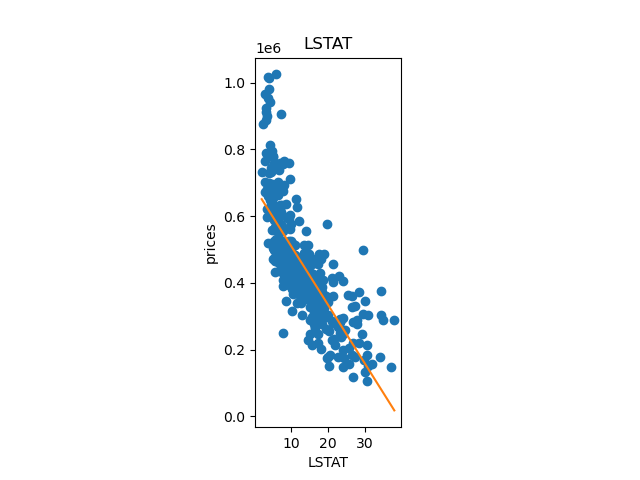
OUTPUT: Training and testing split was successful.

6. Fit the regression line for train and test dataset: The regression line is fit for both test and train dataset.

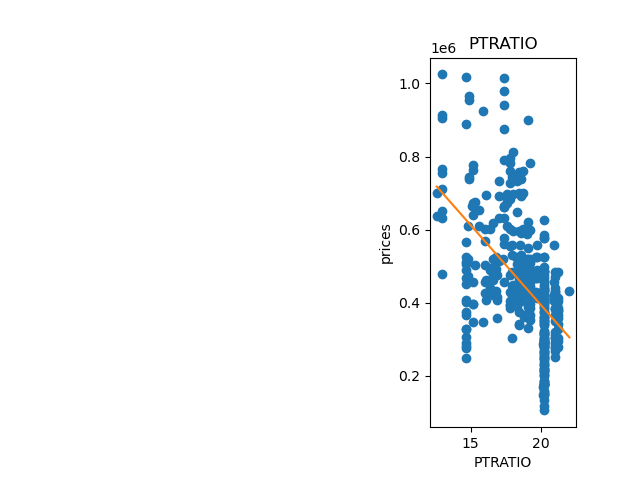
Regression line for train dataset:



**Fig 5.6.1: Train\_RM**

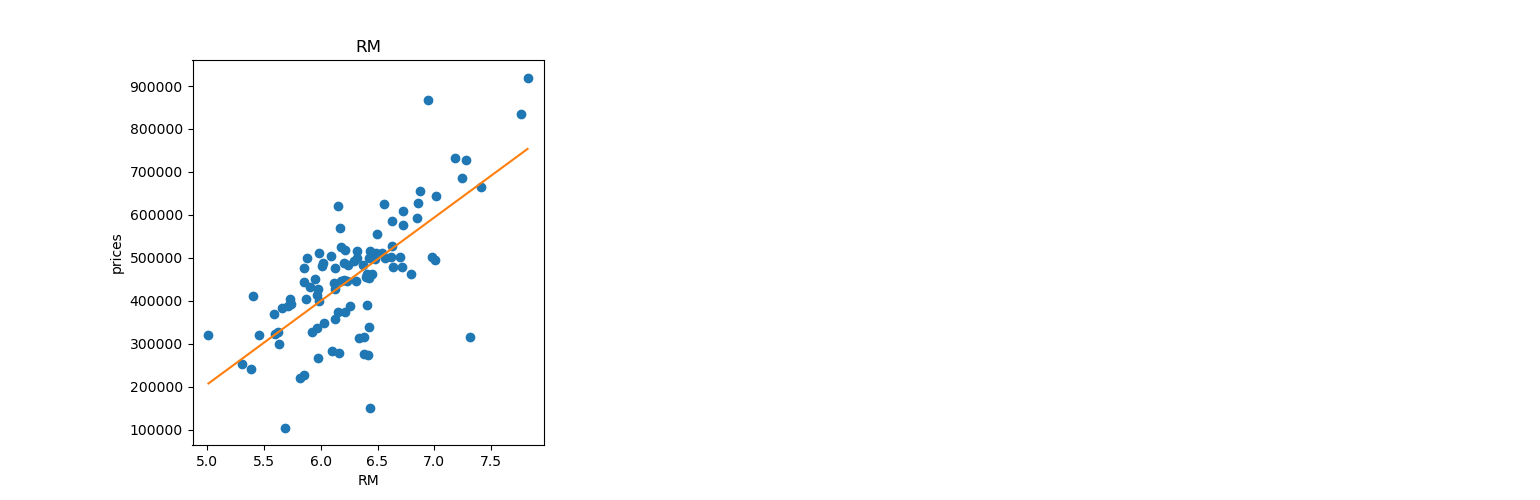


**Fig 5.6.2: Train\_LSTAT**

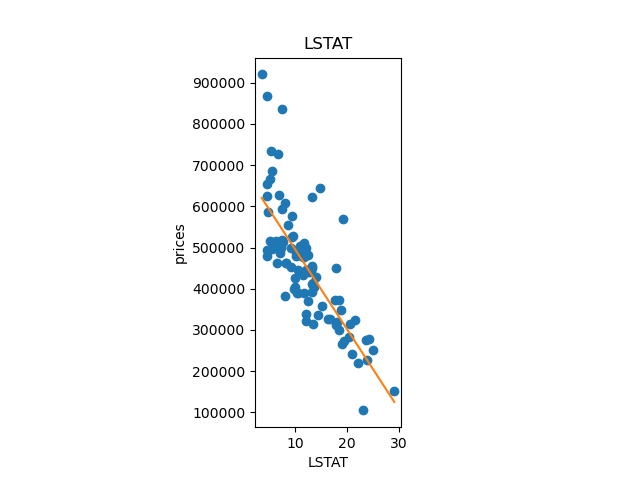


**Fig 5.6.3: Train\_PTRATIO**

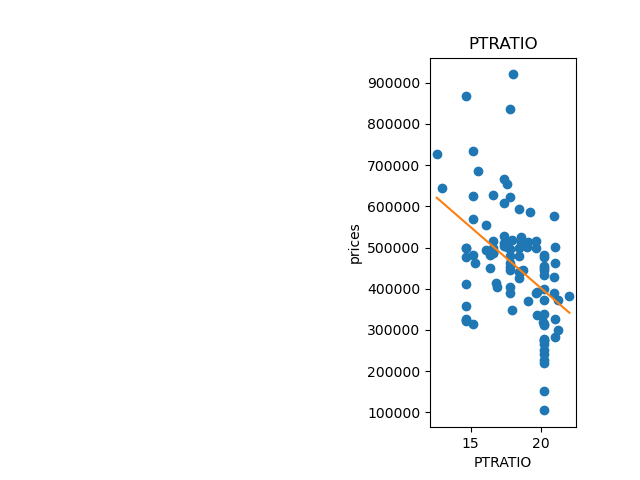
Regression line for test dataset:



**Fig 5.6.4: Test\_RM**



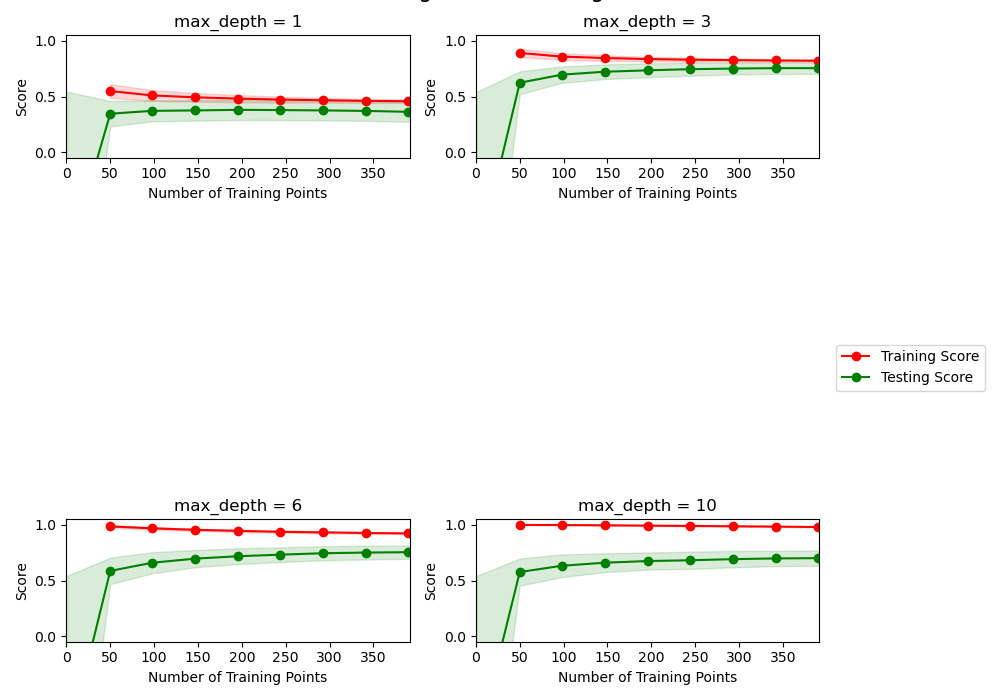
**Fig 5.6.5: Test\_LSTAT**



**Fig 5.6.6: Test\_PTRATIO**

The above figures 5.6.1, 5.6.2, 5.6.3 shows the regression line for the three features with the price for train dataset. And the figure 5.6.4, 5.6.5, 5.6.7 shows the regression line for three features with the price for test dataset. From the above plots we can observe that as RM increases price increases so it linearly varies with price, and as LSTAT and PTRATIO increases price decreases.

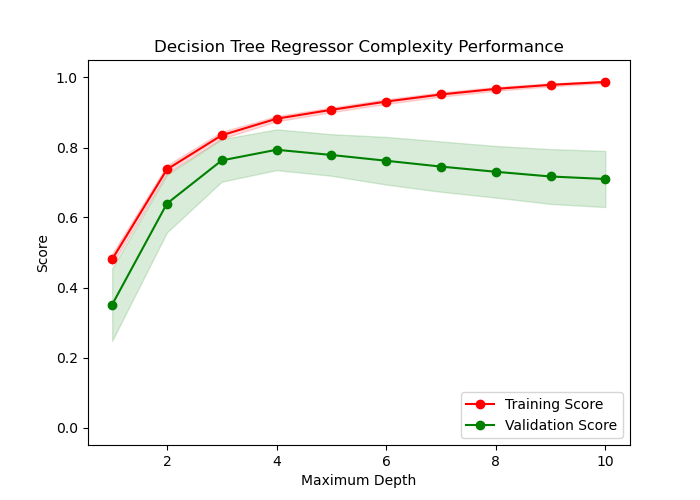
7. Analyzing model performance with different max\_depth:



**Fig 5.7: Model performance with different max\_depth**

The above figure 5.7 shows the plot of score to the number of training points for different max\_depth. We can observe that the model as a better performance around the max\_depth of 3 and 4.

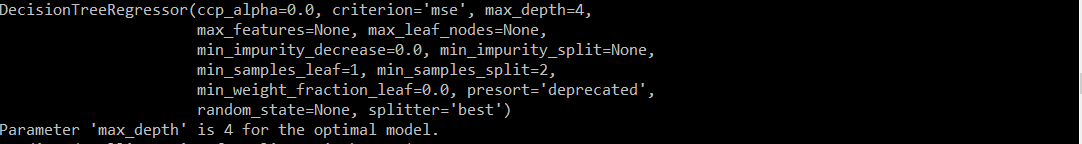
8. Best max\_depth for this model is:



**Fig 5.8:Complexity Curve**

The above figure 5.8 shows the complexity curve which gives the best max\_depth for the model. The best max\_depth for this model is 4.

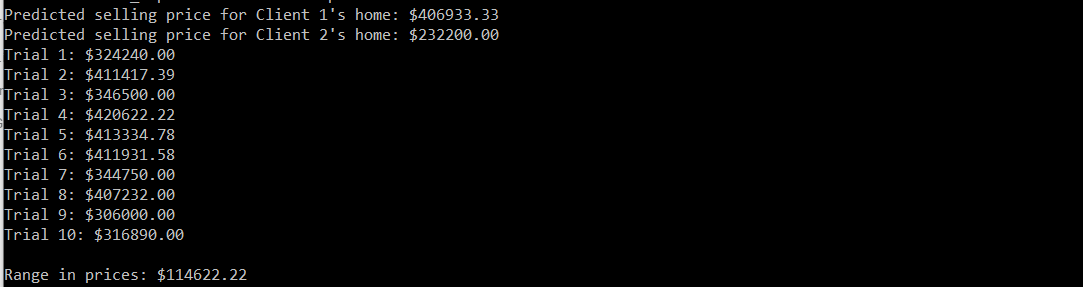
9. Evaluating the model performance:



**Fig 5.9 : Model performance**

The above figure gives the model performance it describes the various parameters as shown in the above figure.

10. Predicting the price for the client’s data:



**Fig 5.10: Predicted price for clients data**

The above figure 5.10 shows the predictions for client’s data. And it also predicts the price for the client’s data in 10 trials so that we can get the range of the price.

**6. CONCLUSION AND FUTURE ENCHANCEMENTS:**

Using more features,

We used the features ‘RM’, ‘PTRATIO’, and LSTAT’ for predicting the prices of house, we believe that more features could have been taken into account. For example, Tax, accessibility to radial highways could have also been included as independent variables to predict the price using linear regression.

Better training and testing data split,

The training size used for this project is 80%. We felt that a training size of 70% could have been used which could have improved the accuracy of our models using linear regression.

**7. REFERENCES:**

1. <https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>
2. <https://www.wired.co.uk/article/machine-learning-ai-explained>
3. David E. Rapach , Jack K. Strauss “ Forecasting real housing price growth in the Eighth District states”
4. <http://lib.stat.cmu.edu/datasets/boston>
5. Vasilios Plakandaras+ and Theophilos, Rangan Gupta\*, PeriklisGogas “Forecasting the U.S. Real House Price Index”
6. Gupta and Das (2010) Forecasting the US Real House Price Index: Structural and Non-Structural Models with and without Fundamentals
7. Wang, J., & Tian , P. Real Estate Price Indices Forecast by Using Wavelet Neural Network, Computer Simulation, 2005:2.
8. Paul K. Asabere and Forrest E. Huffman. “Price Concessions, Time of the Market,and the Actual Sale Price of Homes”. In: Journal of Real Estate Finance and Economics 6(1993), pp. 167–174.

URL: <https://link.springer.com/article/10.1007/BF01097024>