**PROJECT REPORT**

(Project Term Aug-Nov 2021)

## (House Price Prediction by Machine Learning)

Submitted by

**Ankit Mehta Registration Number :11903842**

**Course Code: INT246 – Soft Computing Techniques**

Under the Guidance of

**(Dr. Sagar Pande Sir)**

# School of Computer Science and Engineering

A picture containing text

Description automatically generated

**DECLARATION**

We hereby declare that the project work entitled (“House Price Prediction by ML”) is an authentic record of our own work carried out as requirements of Project for the award of B. Tech degree in \_Computer Science and Engineering\_ from Lovely Professional University, Phagwara, under the guidance of (Dr. Sagar Pande Sir), during August to November 2020. All the information furnished in this project report is based on our own intensive work and is genuine.

Name of Student: Ankit Mehta

Registration Number: 11903842

Roll no - 01

Ankit Mehta

(Signature of Student)

Date: 20-11-2021

**CERTIFICATE**

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Dr. Sagar Pande Sir

**Signature and Name of the Mentor**

**School of Computer Science and Engineering,**

Lovely Professional University,

Phagwara, Punjab.

Date: 20-11-2021

**ACKNOWLEDGEMENT**

I would like to express my gratitude towards my university as well as Dr. Sagar Pande Sir providing me the golden opportunity to do this wonderful Project on **House Price Prediction** which also helped me in doing a lot of homework and learning. As a result, I came to know about so many new things. So, I am really thanking full to them.

Moreover, I would like to thank my friends who helped me a lot whenever I got stuck in some problem related to my course. I am thankful to have such a good support of them as they always have my back whenever I need.

Also, I would like to mention the support system and consideration of my parents who have always been there in my life to make me choose right thing and oppose the wrong. Without them I could never **had learned and became a person who I am now.**

**>→→→→→→→→→→→**

**TABLE OF CONTENTS**

Title Page……………………………………………………………………………(i)

Declaration…...……………………………………………………………………. (ii)

Certificate……...…….………………………………………………………………(iii)

Acknowledgement…………………...………………………………………….......(iv)

Table of Contents……………………...……………………………………………(v)

1. Introduction

2. Profile of the Problem. Rationale/Scope of the Project (Problem Statement)

3. Existing System

* Introduction
* Existing Software
* DFD for present system
* What’s new in the system to be developed

4. Problem Analysis

* Product definition
* Feasibility Analysis
* Project Plan

5. Software Requirement Analysis

* Introduction
* General Description
* Specific Requirements

6. Design

* System Design
* Design Notations
* Detailed Design
* Flowcharts
* Pseudo code

7. Testing

* Functional testing
* Structural testing
* Levels of testing
* Testing the project

8. Implementation

* Implementation of the project
* Conversion Plan
* Post-Implementation and Software Maintenance

9. Project Legacy

* Current Status of the project
* Remaining Areas of concern
* Technical and Managerial lessons learnt

10. User Manual: A complete document (Help Guide) of the software developed***.***

11. Source Code (wherever applicable) or System Snapshots

12. Bibliography

**1. INTRODUCTION**

Machine learning is a branch of Artificial Intelligence which is used to analyze the data more smartly. It automates the process using certain algorithms to minimize human intervention in the process.

Machine learning is the practice of building systems, known as models, that can be trained using data to find patterns which can then be used to make predictions on new data.

Unlike a lot of other programming, a machine learning model is not a rules-based system where a series of ‘if/then’ statements are used to determine outcomes (e.g., ‘If a student miss more than 50% of classes then automatically fail them’).

Instead, machine learning model examines the statistical relationships between data points in a data set with defined outcomes, and then applies what it has learned about those relationships to analyze and predict outcomes for a new data set.

In this machine learning project, we are going to predict the house price using python. This project will help the sellers and buyers to have an overview of the situation so that they can act accordingly. Predicting house prices can help to determine the selling price of a house of a particular region and can help people to find the right time to buy a home.

2. **Profile of the Problem. Rationale/Scope of the Project**

The problem is the tradition ways of predicting data and analyzing the price of the house. So, machine Learning helps us to predict better price.

In this day and age, it is hard to predict the price of a house. There are a lot of variables that affects its price. People are having a hard time knowing the housing price when each house have different conditions. Generally, people want to buy houses that is worth the price. But they do not know how much a specific condition of the house affects its price. For the sellers, they do not want to sell their houses in a lower-than-average cost. So, they need to be able to accurately predict housing prices so that they can get or give the best price.

Thousands of houses are sold every day. There are some questions every buyer asks himself like: What is the actual price that this house deserves? To solve this problem, we can use Python to analyze available data. These data can be used to predict housing price given certain variables or factors.  Before getting the data, we need to understand the problem we are trying to solve. If you know the domain, think of which factors could play an epic role in solving the problem. If you don't know the domain, read about it.

3.**Existing System**

* **Introduction**

Tradition ways of predicting data and analyzing the price of the house. So, machine Learning helps us to predict better price.

In traditional programming, we would feed the input data and a well written and tested program into a machine to generate output. When it comes to machine learning, input data along with the output is fed into the machine during the learning phase, and it works out a program for itself. To understand this better, refer the below diagram or exiting software:

Diagram

Description automatically generated

* DFD for present system

## There are Seven Steps of Machine Learning

1. Gathering Data - Now, we download the data and look at it. Determine which features are available and which aren't, how many features we generated in hypothesis generation hit the mark, and which ones could be created. Answering these questions will set us on the right track. The Quality and quantity of our data dictate how accurate our model is and this data we will use for training.
2. Preparing that data - We can't determine everything by just looking at the data. We need to dig deeper. This step helps us understand the nature of variables (skewed, missing, zero variance feature) so that they can be treated properly. It involves creating charts, graphs (univariate and bivariate analysis), and cross-tables to understand the behavior of features.
3. Choosing a model - Using a suitable algorithm, we train the model on the given data set. Once the model is trained, we evaluate the model's performance using a suitable error metric. Here, we also look for variable importance, i.e., which variables have proved to be significant in determining the target variable. And, accordingly we can shortlist the best variables and train the model again.
4. Training - Finally, we test the model on the unseen data (test data) set. The goal of training is to predict correctly as often as possible.
5. Evaluation - sees some matric or combination of matrices to measure objective performance of model. Test the model against previously unseen data.

**Selecting a performance measure:** A typical performance measure for regression problem is the Root Mean Square Error (RMSE)

1. Hyperparameter Tuning - Hyperparameter tuning is the problem if choosing a set of optimal parameters for a learning algorithm.
2. Prediction - For predict the data first we need to save our model and launch that model.

Chart

Description automatically generated

* What’s new in the system to be developed
* To apply data preprocessing and preparation techniques in order to obtain clean data
* To build machine learning models able to predict house price based on house features
* To analyze and compare models’ performance in order to choose the best model

**4. Problem Analysis**

We are given dataset of house price with some feature like number of bedrooms, crime rate in area, etc. our task is to create a model which will predict the price for any new house by looking at the features.

* Product definition - In this task on House Price Prediction using machine learning, our task is to use data from the Delhi census to create a machine learning model to predict house prices in the State. The data includes features such as population, median income, and median house prices for each block group in Delhi.
* Feasibility Analysis – a machine learning model is proposed to predict a house price based on data related to the house (its size, the year it was built in, etc.). We can use Python to analyze available data. These data can be used to predict housing price given certain variables or factors.
* Project Plan –
* To apply data preprocessing and preparation techniques in order to obtain clean data
* To build machine learning models able to predict house price based on house features
* To analyze and compare models’ performance in order to choose the best model

**5. Software Requirement Analysis –**

* Introduction

Using a suitable algorithm, we train the model on the given data set. Once the model is trained, we evaluate the model's performance using a suitable error metric. Here, we also look for variable importance, i.e., which variables have proved to be significant in determining the target variable. And, accordingly we can shortlist the best variables and train the model again. Finally, we test the model on the unseen data (test data) set.

* **General Description & Specific Requirements**

Before choosing the model, we need to be clear about what type of model we need and to find out the right type of model we need to focus on requirements and observe the dataset.

There are **three steps** to choose right type of model:

**1. Supervised, Unsupervised or reinforcement learning?**

1. Supervised: Data and label are given
2. Unsupervised: Data given but not label
3. Reinforcement: It is a decision process and learn series of action.

As we choose dataset which include label so we will use **Supervised** Learning.

**2. Classification or Regression?**

1. Classification: It is basically classify the data
2. Regression: It is use to predict or extract value.

In this project we need to predict House price so we will use **Regression** Task.

**3. Batch Learning or online learning techniques?**

1. Batch Learning: In this learning we have our data, and we build our model. This learning can be use when we want to create one time model and we can also rebuild it but not significantly.
2. Online Learning: in this data we are receiving data continuously. This learning can be use when data is significantly change.

As we have our data, and our data is not significantly changes so we will use **Batch learning** in this project.

For this project we will try different Regression model to find out which model will be better, and which will give minimum error.

We will try 3 different Regression model:

1. Decision Tree regression
2. Linear Regression
3. Random Forest Regression

The output for this model is given in below picture.

Text, letter

Description automatically generated

As we can see that Random Forest Regression is giving minimum error,so we will use this model for this project.

**6. Design –**

**1.Gathering Data**

The Quality and quantity of our data dictate how accurate our model is and this data we will use for training.

1. **Preparing that data**

Prepare the data for training, which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)

Our Dataset contains total 506 Data (Rows) and 14 Attributes (Columns). The Attributes name of the dataset:

Text, letter

Description automatically generated

In this project we start preparing the data by splitting the data in training set (80%) and testing set (20%).

Graphical user interface, text, application

Description automatically generated

As you can see that our dataset is divided into train and test set. From 506 data train set contains 404 data and test set contains 102 data.

In our Dataset we have one attribute which is categorical (0 and 1).

Graphical user interface, application

Description automatically generated

There might be possible that in training set there is no data which contain 1 as “CHAS” Attribute and it only contains attributes which have 0 value. So, problem can be arisen when we try to test data which has value 1. To solve this problem, we must split the data with the same ratio.

Graphical user interface, text, application

Description automatically generated

As you can see we divide the data and the ratio of data is almost same [ (95/7) =13.5714 and (376/28) =13.4286].

Also, we can tune the parameter to increase a performance. As a data scientist we should be able to observe and find out which parameters can be tune and give better performance. From all the attributes there are two attributes “TAX” and “RM” if we divide TAX by RM so it will give us tax per room.

Graphical user interface, text, application

Description automatically generated

From above image we can see that by tuning the parameter we get very strong negative correlation (TAXRM) which we affect the performance of model.

**Missing Attributes**

To take care of missing attributes, you have three options:

1. Get rid of the missing data points (It can be used when missing data is less)
2. Get rid of the whole attribute (When the attribute is not important, or it will not affect the performance)
3. Set the Value to some value (0, mean or median)

In this project we will use 3rd option and we fill median to the missing attributes using Simple Imputer Method.

Graphical user interface, text, application, email

Description automatically generated

**SciKit-learn Design**

Primarily, three types of objects

1. Estimators: It estimates some parameter based on a dataset. Example. imputer. It has a fit method and transform method. Fit method (Fits the dataset and calculates internal parameters)
2. Transformers: transform method takes input and returns output based on the learning from fit (). It also has a convenience function called fit transform () which fits and then transforms.
3. Predictors: Linear Regression model is an example of predictor. fit () and predict () are two common functions. It also gives score () function which will evaluate the predictions.

**Feature Scaling**

Primarily, two types of feature scaling methods:

1. Min-max scaling (Normalization): (value — min)/(max — min) [Sklearn provides a class called “MinMaxScaler” for this]
2. Standardization: (value — mean)/std [Sklearn provides a class called “StandardScaler” for this]

**Creating a Pipeline**

A machine learning pipeline is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome, whether positive or negative. The process of tying together different pieces of the ML process is known as a pipeline.

Graphical user interface, text, application, email

Description automatically generated

**3. Choosing a Model**

Before choosing the model, we need to be clear about what type of model we need and to find out the right type of model we need to focus on requirements and observe the dataset.

There are **three steps** to choose right type of model:

**1. Supervised, Unsupervised or reinforcement learning?**

1. Supervised: Data and label are given
2. Unsupervised: Data given but not label
3. Reinforcement: It is a decision process and learn series of action.

As we choose dataset which include label so we will use **Supervised** Learning.

**2. Classification or Regression?**

1. Classification: It is basically classifying the data
2. Regression: It is used to predict or extract value.

In this project we need to predict House price so we will use **Regression** Task.

**3. Batch Learning or online learning techniques?**

1. Batch Learning: In this learning we have our data, and we build our model. This learning can be use when we want to create one time model and we can also rebuild it but not significantly.
2. Online Learning: in this data we are receiving data continuously. This learning can be use when data is significantly change.

As we have our data, and our data is not significantly changes so we will use **Batch learning** in this project.

For this project we will try different Regression model to find out which model will be better, and which will give minimum error.

We will try 3 different Regression model:

1. Decision Tree regression
2. Linear Regression
3. Random Forest Regression

The output for this model is given in below picture.

Text, letter

Description automatically generated

As we can see that Random Forest Regression is giving minimum error, so we will use this model for this project.

* 1. **Testing -**

The goal of testing is to predict correctly as often as possible.

Implementation of testing:

Text

Description automatically generated

**Evaluation**

Uses some matric or combination of matrices to measure objective performance of model. Test the model against previously unseen data.

**Selecting a performance measure:**

A typical performance measure for regression problem is the Root Mean Square Error (RMSE)

Graphical user interface, text

Description automatically generated

For Example, if the actual price of house is 5cr and model predict 10cr then there will be error of +5 and if the actual price of house is 10cr and model predict 5cr then there will be error of -5. if we consider both value +5 and -5 then we must choose performance measure which will give absolute value and RMSE can do that.

RMSE is generally the preferred performance measure for regression task, so we choose it for this project. We can also use other measures like Mean Absolute Error, Manhattan normed but we will choose RMSE for this project.

Graphical user interface, text, application, email

Description automatically generated

**Hyperparameter Tuning**

Hyperparameter tuning is the problem if choosing a set of optimal parameters for a learning algorithm. Intrinsic configuration to the model and whose value is defined by the data is called Parameter. By contrast, a configuration that is external to the model and whose value cannot be estimated from the data is called Hyperparameter.

When performing Machine Learning tasks, we generally divide our dataset in training and test sets. This is done so that to test our model after having trained it (in this way we can check its performances when working with unseen data). When using Cross-Validation, we divide our training set into N other partitions to make sure our model is not overfitting our data-Fold, we divide our training set into N partitions and then iteratively train our model using N-1 partitions and test it with the left-over partition (at each iteration we change the left-over partition).We used N=10

Graphical user interface, application, Word

Description automatically generated

Above picture shows the implementation of k-fold cross validation and rmse scores for random forest regression technique.

1. **Implementation**

For predict the data first we need to save our model and launch that model.

Code for Saving the model:

Graphical user interface, text, application

Description automatically generated

Implementation of Testing the test data on model:

Text

Description automatically generated

Implementation of Using the model with data

Text

Description automatically generated

1. **Project Legacy –**

Now our model is ready so we can launch this model but before launching the model we need to take care of a few things like checking the assumptions and, we need to check weather model is giving correct output or not. For example, in this project we are predicting the price so if requirement is predicting the price and not categorize as expensive or cheap but from our assumption, we build a model which categorize price as cheap or expensive then it will create a huge problem. So before launching the model we need to check all the requirement and verify first to the client. What are the things that a potential home buyer considers before purchasing a house? The location, the size of the property, vicinity to offices, schools, parks, restaurants, hospitals or the stereotypical white picket fence? What about the most important factor — the price?

Now with the lingering impact of demonetization, the enforcement of the Real Estate (Regulation and Development) Act (RERA), and the lack of trust in property developers in the city, housing units sold across India in 2017 dropped by 7 percent. In fact, the property prices in Delhi fell by almost 5 percent in the second half of 2017, said a study published by property consultancy Knight Frank.  
For example, for a potential homeowner, over 9,000 apartment projects and flats for sale are available in the range of ₹42-52 lakh, followed by over 7,100 apartments that are in the ₹52-62 lakh budget segment, says a report by property website Makaan. According to the study, there are over 5,000 projects in the ₹15-25 lakh budget segment followed by those in the ₹34-43 lakh budget category.

Buying a home, especially in a city like Delhi, is a tricky choice. While the major factors are usually the same for all metros, there are others to be considered for the Silicon Valley of India. With its help millennial crowd, vibrant culture, great climate and a slew of job opportunities, it is difficult to ascertain the price of a house in Bengaluru.

1. **User Manual: A complete document (Help Guide) of the software developed*.***

Machine learning is a branch of Artificial Intelligence which is used to analyze the data more smartly. It automates the process using certain algorithms to minimize human intervention in the process.

In this machine learning project, we are going to predict the house price using python. This project will help the sellers and buyers to have an overview of the situation so that they can act accordingly.

The project has been created to help people understand the complete process of machine learning / data science modeling. These steps ensure that you won't miss out any information in the data set and would also help another person understand your work.

Predict the price of a house by its features. If you are a buyer or seller of the house but you don’t know the exact price of the house, so supervised machine learning regression algorithms can help you to predict the price of the house just providing features of the target house.

**11. Source Code (wherever applicable) or System Snapshots**

import pandas as pd

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

housing.head()

housing.info()

housing['CHAS'].value\_counts()

housing.describe()

%matplotlib inline

*# For learning purpose*

import numpy as np

def split\_train\_test(data, test\_ratio):

    np.random.seed(42)

    shuffled = np.random.permutation(len(data))

    print(shuffled)

    test\_set\_size = int(len(data) \* test\_ratio)

    test\_indices = shuffled[:test\_set\_size]

    train\_indices = shuffled[test\_set\_size:]

    return data.iloc[train\_indices], data.iloc[test\_indices]

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

train\_set, test\_set  = train\_test\_split(housing, test\_size=0.2, random\_state=42)

print(f"Rows in train set: {len(train\_set)}\nRows in test set: {len(test\_set)}\n")

from sklearn.model\_selection import StratifiedShuffleSplit

split = StratifiedShuffleSplit(n\_splits=1, test\_size=0.2, random\_state=42)

for train\_index, test\_index in split.split(housing, housing['CHAS']):

    strat\_train\_set = housing.loc[train\_index]

    strat\_test\_set = housing.loc[test\_index]

strat\_test\_set['CHAS'].value\_counts()

strat\_train\_set['CHAS'].value\_counts()

housing = strat\_train\_set.copy()

corr\_matrix = housing.corr()

corr\_matrix['MEDV'].sort\_values(ascending=False)

housing.plot(kind="scatter", x="RM", y="MEDV", alpha=0.8)

housing["TAXRM"] = housing['TAX']/housing['RM']

housing.head()

corr\_matrix = housing.corr()

corr\_matrix['MEDV'].sort\_values(ascending=False)

housing.plot(kind="scatter", x="TAXRM", y="MEDV", alpha=0.8)

housing = strat\_train\_set.drop("MEDV", axis=1)

housing\_labels = strat\_train\_set["MEDV"].copy()

a = housing.dropna(subset=["RM"]) *#Option 1*

a.shape

housing.drop("RM", axis=1).shape *# Option 2*

median = housing["RM"].median() *# Compute median for Option 3*

housing["RM"].fillna(median) *# Option 3*

housing.shape

housing.describe() *# before we started filling missing attributes*

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy="median")

imputer.fit(housing)

imputer.statistics\_

X = imputer.transform(housing)

housing\_tr = pd.DataFrame(X, columns=housing.columns)

housing\_tr.describe()

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import StandardScaler

my\_pipeline = Pipeline([

    ('imputer', SimpleImputer(strategy="median")),

*#     ..... add as many as you want in your pipeline*

    ('std\_scaler', StandardScaler()),

])

housing\_num\_tr = my\_pipeline.fit\_transform(housing)

housing\_num\_tr.shape

from sklearn.linear\_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

*# model = LinearRegression()*

*# model = DecisionTreeRegressor()*

model = RandomForestRegressor()

model.fit(housing\_num\_tr, housing\_labels)

some\_data = housing.iloc[:5]

some\_labels = housing\_labels.iloc[:5]

prepared\_data = my\_pipeline.transform(some\_data)

model.predict(prepared\_data)

list(some\_labels)

from sklearn.metrics import mean\_squared\_error

housing\_predictions = model.predict(housing\_num\_tr)

mse = mean\_squared\_error(housing\_labels, housing\_predictions)

rmse = np.sqrt(mse)

rmse

from sklearn.model\_selection import cross\_val\_score

scores = cross\_val\_score(model, housing\_num\_tr, housing\_labels, scoring="neg\_mean\_squared\_error", cv=10)

rmse\_scores = np.sqrt(-scores)

rmse\_scores

def print\_scores(scores):

    print("Scores:", scores)

    print("Mean: ", scores.mean())

    print("Standard deviation: ", scores.std())

print\_scores(rmse\_scores)

from joblib import dump, load

dump(model, 'Delhi.joblib')

X\_test = strat\_test\_set.drop("MEDV", axis=1)

Y\_test = strat\_test\_set["MEDV"].copy()

X\_test\_prepared = my\_pipeline.transform(X\_test)

final\_predictions = model.predict(X\_test\_prepared)

final\_mse = mean\_squared\_error(Y\_test, final\_predictions)

final\_rmse = np.sqrt(final\_mse)

*# print(final\_predictions, list(Y\_test))*

1. **Bibliography**

* GeeksForGeeks - [GeeksforGeeks | A computer science portal for geeks](https://www.geeksforgeeks.org/)
* Kaggle - [Bengaluru House price data | Kaggle](https://www.kaggle.com/amitabhajoy/bengaluru-house-price-data)
* Hacker Earth - [Practical Machine Learning Project in Python on House Prices Data Tutorials & Notes | Machine Learning | Hacker Earth](https://www.hackerearth.com/practice/machine-learning/machine-learning-projects/python-project/tutorial/)

* Code Chef - [Competitive Programming | Participate & Learn | Code Chef](https://www.codechef.com/)
* Git & GitHub - <https://github.com/>

* Stack Overflow - <https://stackoverflow.com/>
* Quora - <https://www.quora.com/>

* Wikipedia - <https://en.wikipedia.org/wiki/Coding>
* Google - <https://edu.google.com/code-with-google/>
* Code - <https://code.org/>