A

Mini-project Report

on

“Housing Prices Predictor”

by

Mihir Ranade (33162)

Akhilesh Murugkar (33151)

Prathamesh Paraswar (33155)

Sneha Patil (33158)

Submitted to the respective faculty

In partial fulfillment of Laboratory Practice – I Submission

Under the guidance of

Prof. Shreesudha Kembhavi

Department Of Information Technology

Pune Institute of Computer Technology College of Engineering

Sr. No 27, Pune-Satara Road, Dhankawadi, Pune - 411 043.

A. Y. 2021-2022

INTRODUCTION

- It is imperative to find the correct price for a housing residence if anyone is planning to shift to a particular locality in a city with different brokers quoting different prices for similar houses in the same locality in a particular city.
- This project aims to reduce the burden of the common man who intends to find the probable range of prices for housing residences for a given parameters with minimal error.
- This project will encourage all the stake-holders in the real estate sector like brokers and the buyers as well as sellers to be more realistic when quoting housing prices and will give a base for understanding the market regarding the same.

Relavant Literature Survey

1. https://github.com/supreethtgowda/House_Price_Prediction_Project.git : This is another project which predicts the Housing Prices with many different attributes like

- dayhours: Date house was sold
- price: Price is prediction target
- room_bed: Number of Bedrooms/House
- room_bath: Number of bathrooms/bedrooms
- living_measure: square footage of the home
- lot_measure: quare footage of the lot
- ceil: Total floors (levels) in house
- coast: House which has a view to a waterfront
- sight: Has been viewed
- condition: How good the condition is (Overall)
- quality: grade given to the housing unit, based on grading system
- ceil_measure: square footage of house apart from basement
- basement_measure: square footage of the basement
- yr_built: Built Year
- yr_renovated: Year when house was renovated
- zipcode: zip
- lat: Latitude coordinate
- long: Longitude coordinate
- living_measure15: Living room area in 2015(implies-- some renovations) This might or might not have affected the lotsize area
- lot_measure15: lotSize area in 2015(implies-- some renovations)

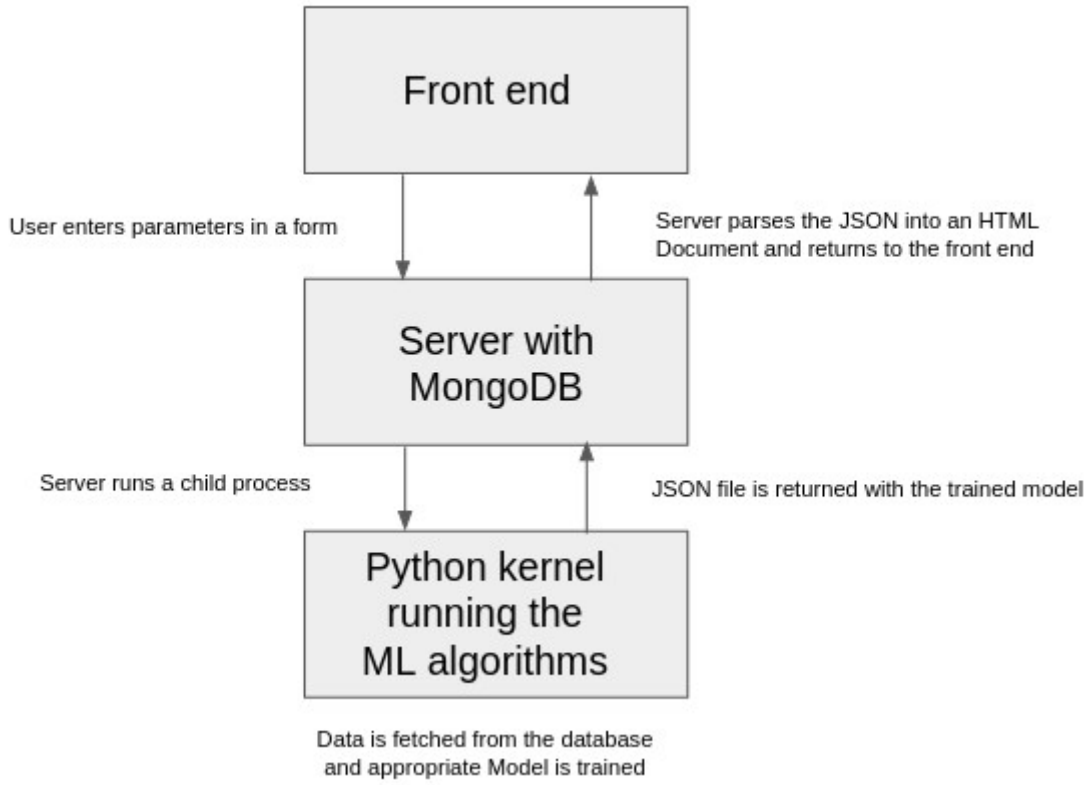
- furnished: Based on the quality of room 23: total_area: Measure of both living and lot

2. https://towardsdatascience.com/machine-learning-project-predicting-boston-house-prices-with-regression-b4e47493633d : Machine Learning Project: Predicting Boston House Prices With Regression. In this project, we will develop and evaluate the performance and the predictive power of a model trained and tested on data collected from houses in Boston's suburbs.

List of Software/Hardware Requirements

In order to access the project and use it, one needs an internet connection, a device which can access Internet through web-browser, like PC or Mobile. We have made this project accessible as a web-apps, so general web-app dependencies apply like Browser version, etc. But no extra dependencies are required.

Proposed Architecture



- We have used HTML, CSS and Javascript for frontend. For Backend part, we have used Node.js and MongoDB.
- For editing the static pages we have used express which will make the pages dynamic.
- We have used MongoDB because it is a NOSQL database and data retrieval is faster.
- We have connected ML Model, MongoDB Database and run them with the help of Node.js.
- We run the Linear Regression from within a child process which is spawned from Node.js and runs python scripts and returns JSON files for the particular input given which are passed on to the front-end for viewing purpose.

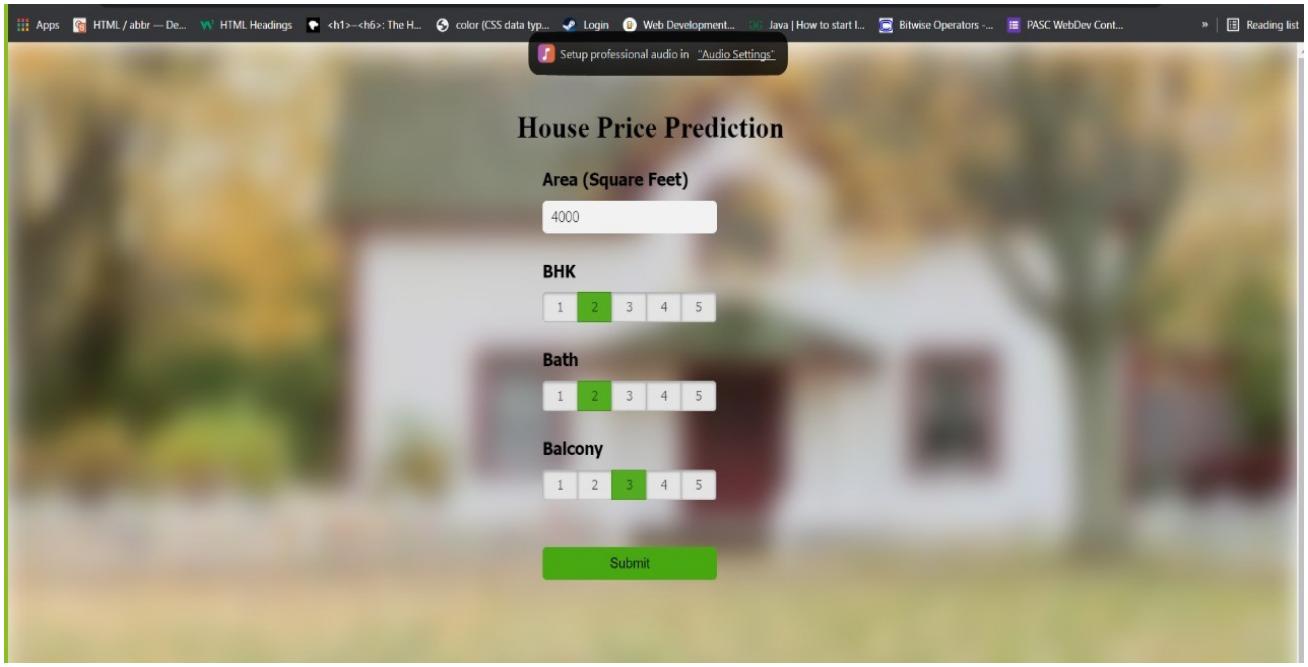
Conclusion

- This project helped us understand the CRUD operations behind any database
- This project helped us to understand how to work in MongoDB and to connect MongoDB with Node.js using mongoose library.
- This project also helped us to understand linear regression in Machine Learning.

References

- GeeksForGeeks for Pymongo library.
- TutorialsPoint for basic MongoDB operations.
- GeeksforGeeks for Linear Regression.
- TowardsDataScience.com for data manipulation using Pandas library.
- MongoDB official documentation for predictive data models in MongoDB.

Annexure



entries

	_id ObjectId	total_sqft Int32	bath Int32	balcony Int32	bhk Int32	price Mixed
1	619f039799e7c14251466107	2850	4	1	4	428
2	619f039799e7c14251466108	1630	3	2	3	194
3	619f039799e7c14251466109	1875	2	3	3	235
4	619f039799e7c1425146610a	1200	2	0	3	130
5	619f039799e7c1425146610b	1235	2	2	2	148
6	619f039799e7c1425146610c	2750	4	0	4	413
7	619f039799e7c1425146610d	2450	4	2	4	368
8	619f039799e7c1425146610e	1875	3	1	3	167
9	619f039799e7c1425146610f	2065	4	1	3	210
10	619f039799e7c14251466110	2059	3	2	3	225
11	619f039799e7c14251466111	1394	2	1	2	100
12	619f039799e7c14251466112	1077	2	2	2	93
13	619f039799e7c14251466113	1566	2	0	2	180
14	619f039799e7c14251466114	840	2	2	1	50
15	619f039799e7c14251466115	1590	3	3	3	131

hello.py > ...

```python
import pymongo
import sys
import pandas as pd
import numpy as np
from pymongo import MongoClient
client = MongoClient('localhost', 27017)
db = client.MINIPRO
collection = db.DATABASE
data = pd.DataFrame(list(collection.find()))
df=data[['total_sqft','bath','balcony','bhk','price']]
X = df.drop(df.columns[4],axis='columns')
Y=df[df.columns[4]]
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test =  train_test_split(X,Y,test_size=0.2)
print(X_train)
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
print(X_train.shape)
print(Y_train.shape)
Y_train=pd.DataFrame(Y_train)
lr.fit(X_train,Y_train)
l=[[sys.argv[1],sys.argv[2],sys.argv[3],sys.argv[4]]]
print(l)
print(lr.predict(sys.argv[1],sys.argv[2],sys.argv[3],sys.argv[4]))
l=[[sys.argv[1],sys.argv[2],sys.argv[3],sys.argv[4]]]
print(l)

```

```javascript
const express = require("express");
const bodyParser = require("body-parser");
const mongoose=require("mongoose");
const app = express();

app.use(bodyParser.urlencoded({ extended: false }))
app.use(express.static("public"));
app.set('view engine', 'ejs');

mongoose.connect("mongodb://localhost:27017/miniproject")
const data=mongoose.Schema({
    square_foot:{
        type:Number,
    },
    bath:{
        type:Number,
    },
    balcony:{
        type:Number,
    },
    bhk:{
        type:Number,
    },price:{
        type:Number,
    }
})

const entry=mongoose.model("entry",data);
app.get("/",function(req,res)
{
    res.render("submit");
})
app.post("/",function(req,res)
{
    const sq_foot=req.body.Squareft;
    const bath=req.body.uiBathrooms;
    const balcony=req.body.uiBalcony;
    const bhk=req.body.uiBHK;
```