**Third Progress Report**

**of**

Project - I

**Subject Code: 4IT31**

**Academic Year 2021-22**

**Group Number : G - 13**

**Name of Students : Kalp Gohil 18IT405**

**Man Desai 18IT404**

**Param Shah 18IT413**

**Jay Patel 17IT419**

**Topic : House Price Prediction**

**Guide Name : Prof. Prachi Shah**

**BACHELOR OF ENGINEERING**

*In*

**INFORMATION TECHNOLOGY**

****

**Birla Vishvakarma Mahavidyalaya**

**Engineering College,**

**Vallabh Vidhyanagar-388120**

**Table of Content**

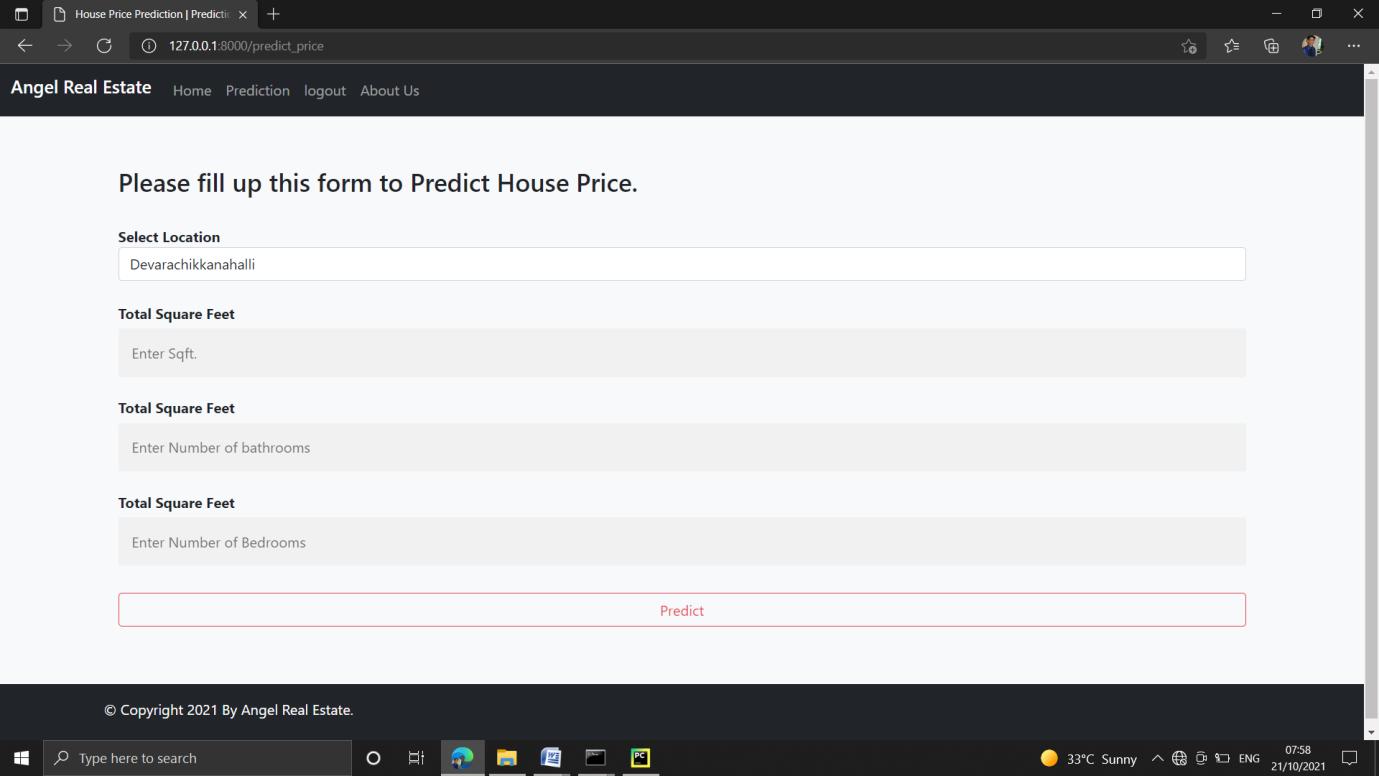
|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topics** | **Page No.** |
| **1** | **Previous Work Done** | **3** |
| **2** | **Further Work Done on Modules** | **4** |
| **3** | **Remaining Work** | **12** |
| **4** | **Conclusion** | **13** |

**1 – Previous Work Done :**

* Sign-Up / Login
* Data Cleaning & Pre-Processing
* UI Design

**2 – Further Work Done on Modules :**

* **Prediction Page:**

****

**Code:**

{% extends 'common.html' %}

{% load static %}

{% block title %} House Price Prediction | Prediction {% endblock %}

{% block content %}

<!-- Form Start -->

<div class="container my-4">

<form action="predict\_price\_actual" method="GET">

<div class="container">

<h3>Please fill up this form to Predict House Price.</h3>

<br>

<label for="location"><b>Select Location</b></label>

<select class="selectpicker form-control" id="location" name="location" required>

{% for location in locations %}

<option value="{{ location }}">{{ location }}</option>

{% endfor %}

</select><br>

<label for="sqft"><b>Total Square Feet</b></label>

<input type="text" placeholder="Enter Sqft." id="sqft" name="sqft" required>

<label for="bath"><b>Total Square Feet</b></label>

<input type="text" placeholder="Enter Number of bathrooms" id="bath" name="bath" required>

<label for="bhk"><b>Total Square Feet</b></label>

<input type="text" placeholder="Enter Number of Bedrooms" id="bhk" name="bhk" required>

<button type="submit" class="btn btn-outline-danger">Predict</button>

</div>

</form>

</div>

<!-- Form End -->

{% endblock %}

* **Data Cleaning & Pre-Processing:**
* **Description:**In this module, first of all, data set has been imported, then data has been cleaned and pre processed in order to predict accurate price.

**Code:**

* **Outlier Removal using Business Logic.**
  + **Check size of each room.**

x = house[house['total\_sqft'] < 1000]['total\_sqft']

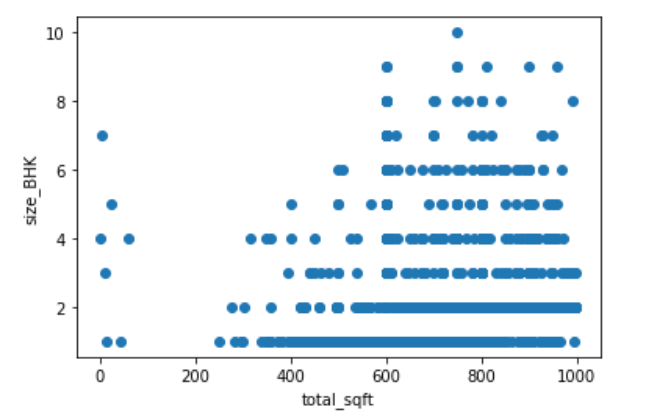
y = house[house['total\_sqft'] < 1000]['size\_BHK']

plt.scatter(x, y)

plt.ylabel('size\_BHK')

plt.xlabel('total\_sqft')

plt.show()



**As a data scientist, when we have a conversation with business manager (who has expertise in real estate domain), he/she will tell that normally sq.ft. per bedroom is 300. If we have 400sq.ft apartment with 2 bhk, then that is like outlier for us. We will remove such outliers by keeping our minimum thresold for sq.ft per bhk to 300.**

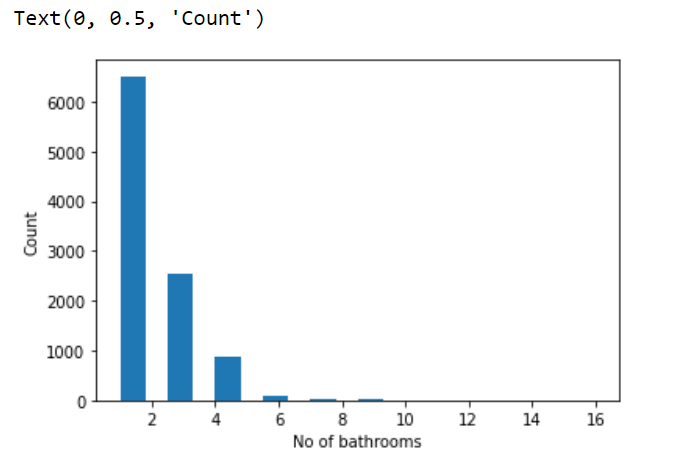
house = house[~((house['total\_sqft'] / house['size\_BHK']) < 300)]

* + **Check no of bathrooms.**

plt.hist(house['bathroom'], width=0.8)

plt.xlabel('No of bathrooms')

plt.ylabel('Count')



**After conversation with business manager (who has expertise in real estate domain), he/she will tell that normally number of bathrooms is equal or one more than number of rooms.**

house = house[~(house['bathroom'] > house['size\_BHK']+2)]

house = house[~(house['bathroom'] >= 9)]

* **Train-Test Split using Sklearn:**

X = house.drop(['price\_lakhs'], axis=1)

Y = house['price\_lakhs']

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

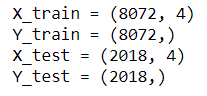
X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2)

print("X\_train =", X\_train.shape)

print("Y\_train =", Y\_train.shape)

print("X\_test =", X\_test.shape)

print("Y\_test =", Y\_test.shape)



* **Handling Categorical Column:**

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import make\_column\_transformer

from sklearn.pipeline import make\_pipeline

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

ohe = OneHotEncoder()

ohe.fit(X[['location']])

ohe.categories\_

column\_trans = make\_column\_transformer((OneHotEncoder(categories=ohe.categories\_), ['location']), remainder='passthrough')

* **Selecting Model:**
* **Description:** We tried many regression model and results of them are as follow.

---- For Linear Regression ----

Accuracy : 82.8091182793346 %

RMSE : 34.908484140321875

---- For Decision Tree Regression ----

Accuracy : 76.65343414199555 %

RMSE : 40.68116843238034

---- For SVR ----

Accuracy : 46.47308668476266 %

RMSE : 61.59819898734227

---- For Lasso Regression ----

Accuracy : 77.48152037294282 %

RMSE : 39.95318874050546

---- For Ridge Regression ----

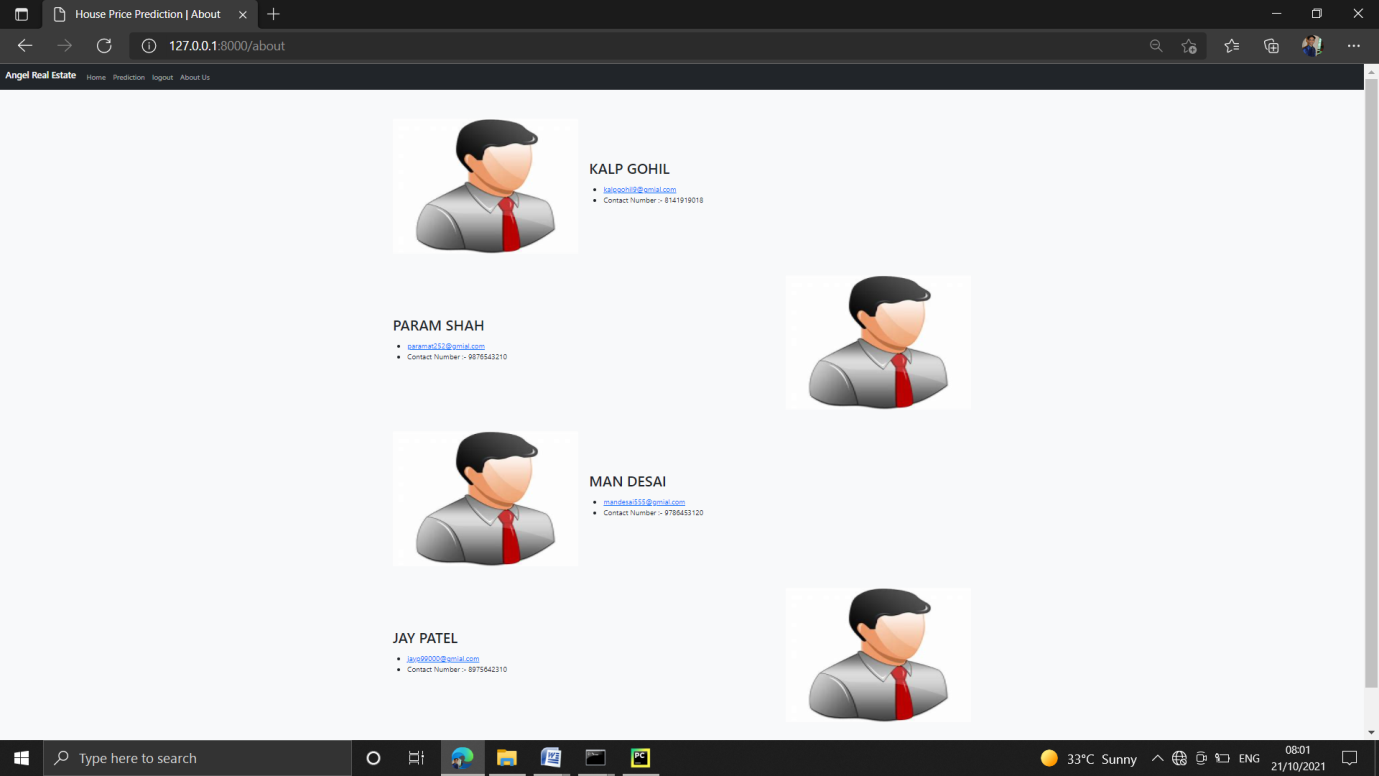
Accuracy : 76.68507224446014 %

RMSE : 40.65359454156182

**So, Here Linear Regression performs well.**

* **UI Design:**
* **Description:** Here we had design about page.

**About Page**

****

**3 - Remaining work and Future Planning :**

As Part of Second mid we have completed around 80% - 90% work and we are remaining with following work module wise:

1. **Model Training:**

* Here, as mentioned Linear Regression performs well. So we will train cleaned data using Linear Regression.

1. **UI Design:**

* As a front end, all the required pages has been designed. In backend, we need to attach our trained model with this website.
* Also so UI part needs improvement, So we will be doing that also in this time.
* So, next we are planning to complete almost 100% of project before the second mid presentation. In that time we will complete our model training part along with its UI integration and as mentioned above.

**4 - Conclusion:**

As part of second mid presentation we have completed our SRS and other three progress report and also, we are done with around 80% - 90% work on actual system implementation. If some changes are required then we will take that into consideration and do changes in completed work.

\*\*\*\*\**The End*\*\*\*\*\*