# Real Estate Price Prediction

**Vijay Gaikwad[1], Zaid Shaikh [2], Shabbir Asgar[3], Prathamesh Shenavi[4], Anushka Shinde[5], Girish Sarwade[6]**

*[2-6]Students, Dept. of Computer Engineering, Vishwakarma Institute of Technology Pune, Maharashtra, India*
*[1]Dean Quality Assurance, Vishwakarma Institute of Technology Pune, Maharashtra, India*

---

**Abstract -** *We are foreseeing the arrangement cost of the houses utilizing different AI calculations. Lodging deals cost are dictated by various factors like space of the property, area of the house, material utilized for development, age of the property, number of rooms and carports, etc. This paper utilizes AI calculations to fabricate the forecast model for houses. Here, AI calculations, for example, strategic regression and backing vector regression, Lasso Regression strategy and Decision Tree are utilized to assemble a prescient model. We have thought about lodging information of about 13000 properties. Logistic Regression*

***Key Words***: Real Estate, Prediction, Linear Regression, K Fold Cross Validation, Grid Search CV, Website Interface.

## 1. INTRODUCTION.

Land Property isn't just the essential need of a man yet today it additionally addresses the wealth and distinction of an individual. Interest in land by and large is by all accounts productive in light of the fact that their property estimations don't decay quickly. Changes in the land cost can influence different family financial backers, brokers, strategy creators and many. Interest in land area is by all accounts an appealing decision for the ventures. Venture is a business movement that the vast majority are keen on this globalization time. There are a few articles that are regularly utilized for speculation, for instance, gold, stocks and property. Specifically, property speculation has expanded essentially since 2011, both on request and property selling.

## 2. LITERATURE SURVEY

First we have explored different papers and conversation on AI for house value prediction[1].The title of the papers is house value expectation is on AI and neural organizations, the depiction of the paper is least blunder and greatest accuracy[2].Next title of the paper is Libertine models dependent on value information from Belfast construe that submarkets and private valuation this model is utilized to recognized over a more extensive spatial scale and suggestions for the assessment measure identified with the choice of practically identical proof and the nature of factors that the qualities may needed.

[3] The title of the paper is getting later patterns in house costs and house purchasing in this paper they utilized input component or social plague that empowers a perspective on lodging as a significant interest in the market

## 3. METHODOLOGY AND APPROACH

### A. Linear regression:

Straightforward direct regression measurable technique permits us to sum up and study the connection between two consistent quantative factors.

- One variable, indicated x, is viewed as the indicator, illustrative, or autonomous variable.
- The other variable, demonstrated y, is viewed as the reaction, result, or ward variable.

### B. Multiple Regression Analysis

Multiple regression analysis is utilized to check whether there is a genuinely essential affiliation the center of sets of factors. It's utilized to find designs in theindividuals sets of data. Various backslide Investigation will be practically a similar Likewise essential straight backslide. The primary differentiation the center of direct straight backslide Also various backslide is in the number for indicators ("x" factors) used inside those backslide. Clear backslide assessment jobs An outright x variable to each subordinate "y" variable. A valid example: (x1, Y1).

Various backslide usage various "x" factors for each free factor: (x1)1, (x2)1,(x3)1, Y1). In one-variable straight regression, you may data specific case subordinate variable (I. E. "deals") against a self-ruling variable (I. E. "benefit"). In any case you could make charmed by what assorted sorts from guaranteeing offers mean for the backslide. You Might set your X1 as specific case kind from asserting deals, your X2 Similarly as thus sort about bargains and so on

### C. Decision Tree

Decision tree assembles regression or characterization models as a tree structure. It separates a dataset into more modest and more modest subsets while simultaneously a related choice tree is steadily evolved. The end-product is a tree with choice hubs and leaf hubs.

A choice hub (e.g., Outlook) has at least two branches (e.g., Sunny, Overcast and Rainy), each addressing esteems for the characteristic tried. Leaf hub (e.g., Hours Played) addresses a choice on the mathematical objective. The highest choice hub in a tree which compares to the best indicator called root hub. Choice trees can deal with both clear cut and mathematical information.

### D. Lasso Regression

LASSO Regression which might be a champion among those backslide models that would open will look at the data. Further, the regression model might be exhibited for an example and the recipe is Additionally recorded to reference.

Tether represents Least Absolute Shrinkage and Selection Operator.

LASSO Regression is a champion among the regularization schedules that makes closefisted models nearby for tremendous number for highlights, the spot sweeping suggests whichever of the accompanying two things :

- Vast enough to improve those tendency of the model on over-fit. Least ten factors can establishment over fitting.
- Huge enough will cause computational tests. The present condition could arise in the occasion from guaranteeing an enormous number or billions about Characteristics.

Tie backslide performs L1 regularization that is it incorporates those discipline equivalent of the preeminent regard of the degree of the coefficients. Here the minimization objective will concern representation copied.

Minimization objective = LS Obj + λ (whole about by and large regard of coefficients). The spot LS Obj stays for least squares target which will be nothing yet the straight backslide focus without regularization Furthermore λ might be those turning figure that controls the action for regularization. The tendency will work with those growing nature of λ and the distinction will lessen Concerning outline the action for shrinkage (λ) increases.

The rope regression gauge is characterized as :

Here the turning part λ controls those quality for punishment, that is. When λ = 0: we get same coefficients Similarly as essential straight backslide. At λ = ∞: continually on coefficients are zero. The moment that 0 < λ < ∞: we get coefficients between 0 What's more that for essential straight backslide subsequently At λ is in the midst of the two limits, we would changing those under two plans.

- Fitting A straight model for y once X.
- Contracting those coefficients.

### E. Grid Search CV

It is the way toward performing hyperparameter tuning to decide the ideal qualities for a given model. As referenced over, the presentation of a model fundamentally relies upon the worth of hyperparameters. Note that it is extremely unlikely to know ahead of time the best qualities for hyperparameters so in a perfect world, we need to attempt all potential qualities to know the ideal qualities. Doing this physically could take a lot of time and assets and along these lines we use GridSearchCV to computerize the tuning of hyperparameters.

GridSearchCV is a capacity that comes in Scikit-learn's(or SK-learn) model_selection package.So a significant point here to note is that we need to have Scikit-learn library introduced on the PC. This capacity assists with circling through predefined hyperparameters and fit your assessor (model) on your preparation set. Along these lines, eventually, we can choose the best boundaries from the recorded hyperparameters.

## 4. IMPLEMENTATION

| area_type | availability | location | size | society | total_sqft | bath | balcony | price |
|---|---|---|---|---|---|---|---|---|
| Super built | 19-Dec | Electronic | 2 BHK | Coomee | 1056 | 2 | 1 | 39.07 |
| Plot Area | Ready To I | Chikka Tir | 4 Bedroom | Theanmp | 2600 | 5 | 3 | 120 |
| Built-up A | Ready To I | Uttarahalli | 3 BHK | | 1440 | 2 | 3 | 62 |
| Super built | Ready To I | Lingadhee | 3 BHK | Soiewre | 1521 | 3 | 1 | 95 |
| Super built | Ready To I | Kothanur | 2 BHK | | 1200 | 2 | 1 | 51 |
| Super built | Ready To I | Whitefield | 2 BHK | DuenaTa | 1170 | 2 | 1 | 38 |
| Super built | 18-May | Old Airpor | 4 BHK | Jaades | 2732 | 4 | | 204 |
| Super built | Ready To I | Rajaji Nag | 4 BHK | Brway G | 3300 | 4 | | 600 |
| Super built | Ready To I | Marathahe | 3 BHK | | 1310 | 3 | 1 | 63.25 |
| Plot Area | Ready To I | Gandhi Ba | 6 Bedroom | | 1020 | 6 | | 370 |
| Super built | 18-Feb | Whitefield | 3 BHK | | 1800 | 2 | 2 | 70 |
| Plot Area | Ready To I | Whitefield | 4 Bedroom | Prrry M | 2785 | 5 | 3 | 295 |
| Super built | Ready To I | 7th Phase | 2 BHK | Shncyes | 1000 | 2 | 1 | 38 |
| Built-up A | Ready To I | Gottigere | 2 BHK | | 1100 | 2 | 2 | 40 |
| Plot Area | Ready To I | Sarjapur | 3 Bedroom | Skityer | 2250 | 3 | 2 | 148 |
| Super built | Ready To I | Mysore Rc | 2 BHK | PrntaEn | 1175 | 2 | 2 | 73.5 |
| Super built | Ready To I | Bisuvanah | 3 BHK | Prityel | 1180 | 3 | 2 | 48 |
| Super built | Ready To I | Raja Rajes | 3 BHK | GrrvaGr | 1540 | 3 | 3 | 60 |
| Super built | Ready To I | Ramakrish | 3 BHK | PeBayle | 2770 | 4 | 2 | 290 |
| Super built | Ready To I | Manayata | 2 BHK | | 1100 | 2 | 2 | 48 |
| Built-up A | Ready To I | Kengeri | 1 BHK | | 600 | 1 | 1 | 15 |
| Super built | 19-Dec | Binny Pete | 3 BHK | She 2rk | 1755 | 3 | 1 | 122 |
| Plot Area | Ready To I | Thanisand | 4 Bedroom | Soitya | 2800 | 5 | 2 | 380 |
| Super built | Ready To I | Bellandur | 3 BHK | | 1767 | 3 | 1 | 103 |
| Super built | 18-Nov | Thanisand | 1 RK | Bhe 2ko | 510 | 1 | 0 | 25.25 |
| Super built | 18-May | Mangamm | 3 BHK | | 1250 | 3 | 2 | 56 |
| Super built | Ready To I | Electronic | 2 BHK | Itelaa | 660 | 1 | 1 | 23.1 |

**Fig-1 :** Dataset

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20, 10)
```

**Fig-2 :** Importing the Libraries

```
df1.groupby('area_type')['area_type'].agg('count')
```

```
area_type
Built-up  Area          2418
Carpet  Area            87
Plot  Area              2025
Super built-up  Area    8790
Name: area_type, dtype: int64
```

**Fig-3 :** Grouping by Area type

```
df2 = df1.drop(['area_type', 'society', 'balcony', 'availability'], axis='columns')
df2.head()
```

**Fig-4 :** Dropping the irrelevant variables

```
df3 = df2.dropna()
df3.isnull().sum()
```

```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

**Fig-5 :** Dropping the Null Values

```
df3['size'].unique()
```

```
array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',
       '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',
       '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',
       '9 BHK', '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',
       '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
       '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

```
df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
```

**Fig-6 :** Adding a New BHK Feature

```
def convert_sqft_to_num(x):
    tokens = x.split('-')
    if len(tokens) == 2:
        return (float(tokens[0])+ float(tokens[1]))/2
    try:
        return float(x)
    except:
        return None
```

**Fig-7 :** Convert Sqft Ranges to Float Value

```
df5 = df4.copy()
df5['price_per_sqft'] = df5['price']*100000/df5['total_sqft']
df5.head()
```

| | location | size | total_sqft | bath | price | bhk | price_per_sqft |
|---|---|---|---|---|---|---|---|
| 0 | Electronic City Phase II | 2 BHK | 1056.0 | 2.0 | 39.07 | 2 | 3699.810606 |
| 1 | Chikka Tirupathi | 4 Bedroom | 2600.0 | 5.0 | 120.00 | 4 | 4615.384615 |
| 2 | Uttarahalli | 3 BHK | 1440.0 | 2.0 | 62.00 | 3 | 4305.555556 |
| 3 | Lingadheeranahalli | 3 BHK | 1521.0 | 3.0 | 95.00 | 3 | 6245.890861 |
| 4 | Kothanur | 2 BHK | 1200.0 | 2.0 | 51.00 | 2 | 4250.000000 |

**Fig-8 :** Adding Price Per Sqft Variable

```
df5.location = df5.location.apply(lambda x: 'other' if x in location_stats_less_than_10 else x)
len(df5.location.unique())
```

**Fig-9 :** Dimensionality Reduction

```
df5[(df5.total_sqft/df5.bhk)<300].head()
```

| | location | size | total_sqft | bath | price | bhk | price_per_sqft |
|---|---|---|---|---|---|---|---|
| 9 | other | 6 Bedroom | 1020.0 | 6.0 | 370.0 | 6 | 36274.509804 |
| 45 | HSR Layout | 8 Bedroom | 600.0 | 9.0 | 200.0 | 8 | 33333.333333 |
| 58 | Murugeshpalya | 6 Bedroom | 1407.0 | 4.0 | 150.0 | 6 | 10660.980810 |
| 68 | Devarachikkanahalli | 8 Bedroom | 1350.0 | 7.0 | 85.0 | 8 | 6296.296296 |
| 70 | other | 3 Bedroom | 500.0 | 3.0 | 100.0 | 3 | 20000.000000 |

```
df5.shape
```

```
(13246, 7)
```

```
df6 = df5[~((df5.total_sqft/df5.bhk)<300)]
df6.shape
```

```
(12502, 7)
```

**Fig-10 :** Price Per Sqft Outlier Removal

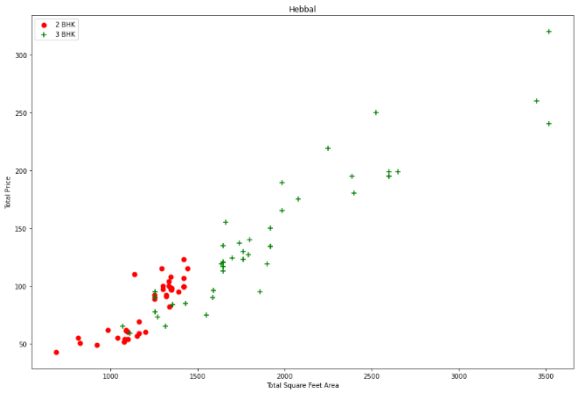


**Fig-11 :** Hebbal Area Outliers

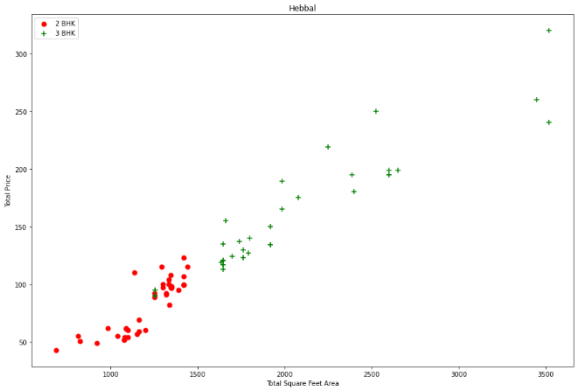


**Fig-12 :** Hebbal Outliers Removed

```
df8[df8.bath>df8.bhk+2]
```

| | location | size | total_sqft | bath | price | bhk | price_per_sqft |
|---|---|---|---|---|---|---|---|
| **1626** | Chikkabanavar | 4 Bedroom | 2460.0 | 7.0 | 80.0 | 4 | 3252.032520 |
| **5238** | Nagasandra | 4 Bedroom | 7000.0 | 8.0 | 450.0 | 4 | 6428.571429 |
| **6711** | Thanisandra | 3 BHK | 1806.0 | 6.0 | 116.0 | 3 | 6423.034330 |
| **8411** | other | 6 BHK | 11338.0 | 9.0 | 1000.0 | 6 | 8819.897689 |

```
df9 = df8[df8.bath<df8.bhk+2]
df9.shape
```

```
(7251, 7)
```

**Fig-12 :** Bathroom Outliers Removed

```
dummies = pd.get_dummies(df10.location)
dummies.head(3)
```

| | 1st Block Jayanagar | 1st Phase JP Nagar | 2nd Phase Judicial Layout | 2nd Stage Nagarbhavi | 5th Block Hbr Layout | 5th Phase JP Nagar | 6th Phase JP Nagar | 7th Phase JP Nagar | 8th Phase JP Nagar | 9th Phase JP Nagar |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

3 rows × 242 columns

```
df11 = pd.concat([df10, dummies.drop('other', axis='columns')], axis='columns')
df11.head()
```

**Fig-13 :** One Hot Encoding for Locations

```
df12 = df11.drop('location', axis='columns')
df12.head()
```

| | total_sqft | bath | price | bhk | 1st Block Jayanagar | 1st Phase JP Nagar | 2nd Phase Judicial Layout | 2nd Stage Nagarbhavi | 5th Block Hbr Layout | 5th Phase JP Nagar |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2850.0 | 4.0 | 428.0 | 4 | 1 | 0 | 0 | 0 | 0 | 0 |
| **1** | 1630.0 | 3.0 | 194.0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 |
| **2** | 1875.0 | 2.0 | 235.0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 |
| **3** | 1200.0 | 2.0 | 130.0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 |
| **4** | 1235.0 | 2.0 | 148.0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |

5 rows × 245 columns

**Fig-14 :** Dropped Size, Price Per Sqft and Locations

```
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, random_state=10)
```

```
from sklearn.linear_model import LinearRegression
lr_clf = LinearRegression()
lr_clf.fit(X_train, y_train)
lr_clf.score(X_test, y_test)
```

```
0.8475442839658576
```

Using K Fold Cross Validation For Measuring Accuracy of Linear Regression Model

```
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
cross_val_score(LinearRegression(), X, y, cv=cv)
```

```
array([0.82522814, 0.77204574, 0.85146771, 0.80498844, 0.84034118])
```

**Fig-15 :** Model Building

```
from sklearn.model_selection import GridSearchCV

from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeRegressor

def find_best_model_using_gridsearchcv(X,y):
    algos = {
        'linear_regression' : {
            'model': LinearRegression(),
            'params': {
                'normalize': [True, False]
            }
        },
        'lasso': {
            'model': Lasso(),
            'params': {
                'alpha': [1,2],
                'selection': ['random', 'cyclic']
            }
        },
        'decision_tree': {
            'model': DecisionTreeRegressor(),
            'params': {
                'criterion' : ['mse','friedman_mse'],
                'splitter': ['best','random']
            }
        }
    }
    scores = []
    cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
    for algo_name, config in algos.items():
        gs =  GridSearchCV(config['model'], config['params'], cv=cv, return_train_score=False)
        gs.fit(X,y)
        scores.append({
            'model': algo_name,
            'best_score': gs.best_score_,
            'best_params': gs.best_params_
        })

    return pd.DataFrame(scores,columns=['model','best_score','best_params'])

find_best_model_using_gridsearchcv(X,y)
```

**Fig-16 :** Findind Best Model Using Grid Search CV

```
def predict_price(location, sqft, bath, bhk):
    loc_index = np.where(X.columns == location)[0][0]

    x = np.zeros(len(X.columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

    return round(lr_clf.predict([x])[0], 2)
```

```
predict_price('1st Phase JP Nagar', 1000, 3, 3)
```

```
86.73
```

```
predict_price('Indira Nagar', 1000, 3, 3)
```

```
184.57
```

**Fig-17 :** Testing Model

```
import pickle
with open('banglore_home_prices_model.pickle', 'wb') as f:
    pickle.dump(lr_clf, f)
```

```
import json
columns = {
    'data_columns' : [col.lower() for col in X.columns]
}

with open("columns.json", "w") as f:
    f.write(json.dumps(columns))
```

**Fig-18 :** Exporting the Model

## 4. EXECUTION AND OUTPUTS

At the point when the code gets executed first we get yields plots and afterward expectation happens. These plots assist us with understanding the connection between's target variable (cost) and diverse indicator factors. This plot gives a reference diagram for rooms and number of houses. It is seen from dataset the check of 3 room houses are more prominent in number and 7 room houses are least in number.

```
@app.route('/get_location_names')
def get_location_names():
    response = jsonify({
        'locations' : util.get_location_names()
    })

    response.headers.add('Access-Control-Allow-Origin', '*')
    return response

@app.route('/predict_home_price', methods=['POST'])
def predict_home_price():
    total_sqft = float(request.form['total_sqft'])
    location = request.form['location']
    bhk = int(request.form['bhk'])
    bath = int(request.form['bath'])

    response = jsonify({
        'estimated_price': util.get_estimated_price(location,total_sqft,bhk,bath)
    })
    response.headers.add('Access-Control-Allow-Origin', '*')
    return response
```

**Fig-19 :** Server.py

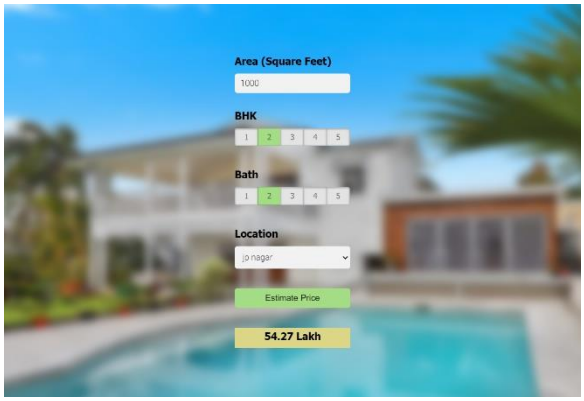


**Fig-20** : Website Interface

## 5. CONCLUSION

We have overseen out how to set up a model that gives clients for a novel best methodology with look at future housing esteem forecasts. A couple of backslide techniques have been researched Furthermore analyzed, while showing up during an expectation system considering best support. Straight previous suggest works bring been used inside our model, something to that effect that future worth expectations will have a propensity towards even more reasonable qualities. We devised a methodology with use also as extensively data as time grants for our expectation framework, by embracing those thoughts from guaranteeing slope boosting. Inspite of Hosting created all the endeavoring arrangement that met our early on prerequisites, there are Different updates that could be delivered later on. These fuse overhauls we didn't choose in light of compelled length of the time. A genuine concern for the forecast system might be the stacking time frame. Besides, our informational index requires over one day ought to get ready. As gone against playing out the calculations consecutively, we may use different processors and equal the calculations in question, which may conceivably diminish the planning time Furthermore expectation period. Incorporate even more functionalities under the model, we can give decisions for customer with select a region then again area should deliver those high temperature maps, rather than entering in the rundown.

## ACKNOWLEDGEMENT

This major project would not have been possible without the valuable assistance of many people to whom we are indebted, in particular, our project guide Shri. Vijay Gaikwad sir. We would also like to thank Our College for providing us with the necessary components for our project. Our thank also goes to all the teachers of the Department who helped us in many difficult situations regarding the project and provided with the necessary advice. A special word of thanks is to our class mates and our families for providing us the moral support.

## REFERENCES

[1]David E. Rapach , Jack K. Strauss " Forecasting genuine lodging value development in the Eighth District states"

[2]Vasilios Plakandaras+ and Theophilos, Rangan Gupta, Periklis Gogas "Determining the U.S. Genuine House Price Index"

[3]Calhoun C. A., (2003), "Property Valuation Models and House Price Indexes for The Provinces of Thailand: 1992 2000", Housing Finance International, 17(3): 31 – 41.

[4]Frew J., Jud G.D., (2003), "Assessing the Value of Apartment Buildings", Journal of Real Estate Research Vol. 25, No. 1, 2003, 77 – 86.

[5] Kanojia Anita (2016), "Valuation of Residential Properties by Hedonic Pricing Method-A State of Art" International Journal of Recent Advances in Engineering and Technology(IJRAET).

[6] Visit Limsombunchai, Christopher Gan and Minso Lee,"House Price Prediction: Hedonic Price Model versus Fake Neural Network", American Journal of Applied Sciences 1 (3):193-201, 2004, ISSN 1546-9239, 193-201.

[7]Housing Price Prediction A Nguyen March 20, 2018.

## BIOGRAPHIES

**Zaid Shaikh :**
Machine Learning and Data Science Enthusiast and Programmer. Second Year Computer Science Student in Vishwakarma Institute of Technology, Pune, INDIA – 411037.

**Vijay Gaikwad :**
Ph.D in Electronics and Telecommunication Engineering. Dean Quality Assurance, Vishwakarma Institute of Technology, Pune, INDIA – 411037.

**Shabbir Asgar :**
Data Science Enthusiast and Coder. Second Year Student at Vishwakarma Institute of Technology, Pune, INDIA – 411037.

**Prathamesh Shenavi :**
Second Year Student at Vishwakarma Institute of Technology, Pune, INDIA – 411037.

**Girish Sarwade :**
Second Year Student at Vishwakarma Institute of Technology, Pune, INDIA – 411037.

**Anushka Shinde :**
Second Year Student at Vishwakarma Institute of Technology, Pune, INDIA – 411037.