House Price Prediction using Machine Learning in Python

Neha Kalra[1]*, Nidhi Uppal[2], Prerna Pathak[3], Muskan Nandkani[4], Garima Sharma[5]

Department of Computer Science & Engineering, The NorthCap University, Gurugram, India

*Corresponding Author [1]E-mail: *nehakalra991.nk@gmail.com*
[2]E-mail: *nidhiuppal25@gmail.com;* [3]E-mail: *prernapathak1220@gmail.com*
[4]E-mail: *muskannandkani99@gmail.com;* [5]E-mail: *garimasharma@ncuindia.edu*

***Abstract*:** The business of buying and selling of house continues to grow every year due to population growth and migration to other cities for their financial purposes. Real estate is a very emerging field in everyone's day to day life. The prices of houses are regularly changing on daily basis and are sometimes fired rather than based on actual estimates. Foreseeing property costs by actual components is a main criterion of this research paper. Our basic aim is to take all the actual and primary features to determine the result of our system. We have used regression models like decision tree classifier, random forest, and multiple linear regression classifier for prediction to get better results and for upgraded accuracy. This paper will give information that how we will predict the home price with the help of different features and python with its libraries. The main objective of this research paper is the estimation of the market worth of a land, house, property which will help customers to buy and sell property without moving to a specialist.

Keywords: House Price, Machine Learning Models, Linear Regression, Decision Tree Regression Regressor, Random Forest Regressor

I. INTRODUCTION

The main aim of this project was to make a house price prediction system with different machine learning algorithms so that it will give more accurate results and finally to decide that which is best with low rate of error. This helps us, as house price value examiners, to get to know more about the real estate and assists with settling on more options. This paper proposes a system that predicts house prices using regression machine learning algorithms. This regression model will not only tell the predicted price of the house which is ready for sale but also about the houses which are under construction. This Regression models will help us in finding the connections between dependent attributes and many independent attributes. The dependent feature in this prediction system is the price of the houses and properties and the independent features are number of bedrooms, bathrooms balconies, square feet, address, latitude and longitude of the property location [1-2]. The full implementation is done by writing code in python which is a programming language and for prediction the models used were linear/multiple regression, decision tree and random forest. The libraries were imported in Python like scikit learn for implementing machine learning algos. After completion of the prediction models, we then combined it with flask which is a python framework and helps to connect with user interface [3-5].

II. DESIGNING AND ARCHITECTURE OF SYSTEM

STEP 1: Data Gathering

In this step we collected the data for India's houses and properties from many different websites and then combined them as per our requirements, we gathered many details like number of bedrooms, bathrooms, balconies, square feet, address, latitude and longitude of the property location etc. We should collect and organize data into an ordered and structured manner. For doing AI research the prerequisite is to collect all the data. Dataset validity is an absolute important requirement in any case and the data should be justifiable [2]. Figure 1 shows the generic flow of development.

STEP 2: Preprocessing of data

After collecting the data, we will now perform the cleaning of the dataset. There can be some impurities like missing values, null values, or outliers in our dataset. All these impurities are removed while preprocessing of the data These can be handled by data cleaning. If there will be some features without some record values, then we will replace it with average values like mean of that feature column.

STEP 3: Model training step

As we have divided our dataset in two parts one is training and the other is testing. So, we will train our model with the training dataset by applying the algorithm on it further we will go for testing of our model with testing dataset.

STEP 4: Testing of model and combining the user interface

The test data will go to the trained model and then it will predict the house price. The model is trained and then it is integrated with the frontend using Flask framework in python.

III. METHODOLOGY

A. ALGORITHMS USED

i) Linear Regression (multiple)

It uses various independent variables (features) to predict a dependent variable (target) i.e. house price.

ii) Decision Tree (regressor)

It observes features of an attribute and trains a model in the form of a tree to predict data in the future to produce meaningful output.

iii) Random Forest (regressor)

It uses the technique which is known as bagging. The main idea here is to train various decision trees with different features and then we average the predictions made by all the trees to reduce the variance and increase the accuracy.

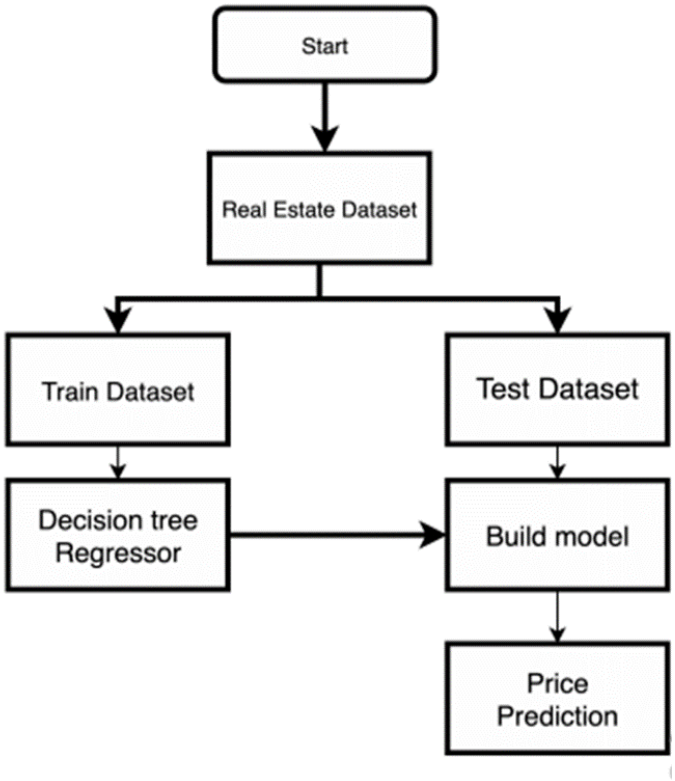

Figure 1. The generic flow of development.

MODEL DUMPING- Dumping the models using pickle library. After that, we measure the accuracy of every machine learning algorithm used and we chose the algorithm with the highest accuracy giving us best results. As random forest gives us the highest accuracy leading to better results and output value prediction, we decide to use it. So, we use the random forest pickle file and import it in our backend that we will be doing on VS Code.

B. FLASK INTEGRATION

After the model building and successful result, the next step is to do the integration with the UI, for this purpose flask is used. Flask is a web framework [2-4].

Flask is used here because of its provision of tools, libraries and many technologies for our web application. Flask is easy to put away routes together and this framework is mainly used for integrating python models.

As we did pickle earlier, it will be used here to directly import the model. Model will be integrated in our util.py file using pickle library and a global variable model which will be taking the input from Artifact's folder. We will be making an Artifacts folder under our Server file which will be consisting of pickle model that we dumped earlier ad a json file.

Our data is in csv format which will be converted first to json format in machine learning file, this data in json format will be directly imported in util.py, from where GET and POST request can fetch data and display in GUI.

We have used the flask server as our backend part to host our application locally. In the respective server folder, we have set two files. The server.py file is responsible for managing routes and for downloading local names and prediction of house prices. It retrieves form data from the previous version and feeds data to the use.py. These routes will be checked using the Postman app. The file.py file is the main part of brain after the backend. It includes the function of uploading JSON file and pickle. This file will take form data from the server.py and uses model to predict the estimated values.

C. DEPLOYMENT

Using HEROKU platform:

A cloud platform as a service (PAAS). Its main use is to deploy, manage and scale new designed apps. The Heroku provides simplicity and adjustability to the user. It gives developers the facility to focus on their core product without distraction of infrastructure maintenance.

In this project we deploy using heroku git and because of this we need to install git and heroku CLI in our system. Now, visit Heroku and create an account. Initially we need to download the gunicorn to our visual A venv site. We can use the pipe to fetch it.

pip install gunicorn - Gunicorn handles applications and takes care of complex issues such as easy installation and the server we use to use Flask in an area where we are developing our application is not good at handling real applications, so we use gunicorn.

We will create a .txt file as we have to let know Heroku about all the files which we have on our local machine so that it can be familiar with all the important files containing flask framework, SK Learn libraries, etc.

This is done by using the pip freeze > requirements.txt syntax.

Procfile is a text file in your app's directory, which clearly states what command should be done to start your application.

For Heroku, Procfile is necessary. Procfile reveals to Heroku that we want to utilize the web interaction with the command named GUNICORN and name.of application. Also, there are a lot of hidden which are unnecessary and which are not to be sent to Heroku. In order to exclude it we created a gitignore file. Our project is now ready for Heroku deployment.

IV. IMPLEMENTATION

A. Data preprocessing:

Missing values are dealt by scaling features using standard scalar and replaced with mean of values of a particular column. To boost the accuracy, we have converted values between 0 to 1 with the help of normalization. Pandas' library of python is used for this. Statistics and visualizations of the dataset such as the minimum, maximum, standard deviation and mean were found out. Dataset is divided into training set which includes 80% of Data from dataset and testing set consists of remaining 20% of Data on which predictions are performed.

EDA: We have used Exploratory Data Analysis techniques to understand different aspects of our dataset and derive useful insights. This step is done after data cleaning and pre-processing. The graphs shown below are created using Tableau which is a data analytics tool. The figure 2 shows the treemap having the average target price(in lakhs) for a particular area type. The pie chart in figure 3, displays average target price(in lakhs) of the house property advertisements published by owner, dealer and builder.

Figure 4 (double bar graph) is known as double bar chart. It shows distance (in km) of every house advertisement posted from hospital and school.

DH (yellow) represents distance from hospital and DS (green) represents distance from school.

We have filtered this graph for top 10 houses.

Correlation matrix: We have to select features that are highly correlated with output i.e. the target price and not the other input features. We have represented the correlation between input features through a heatmap using seaborn library in python.

-1- stronger negative correlation

0- no correlation (best)

1- stronger positive correlation

In figure 5, heatmap showing correlation between features.

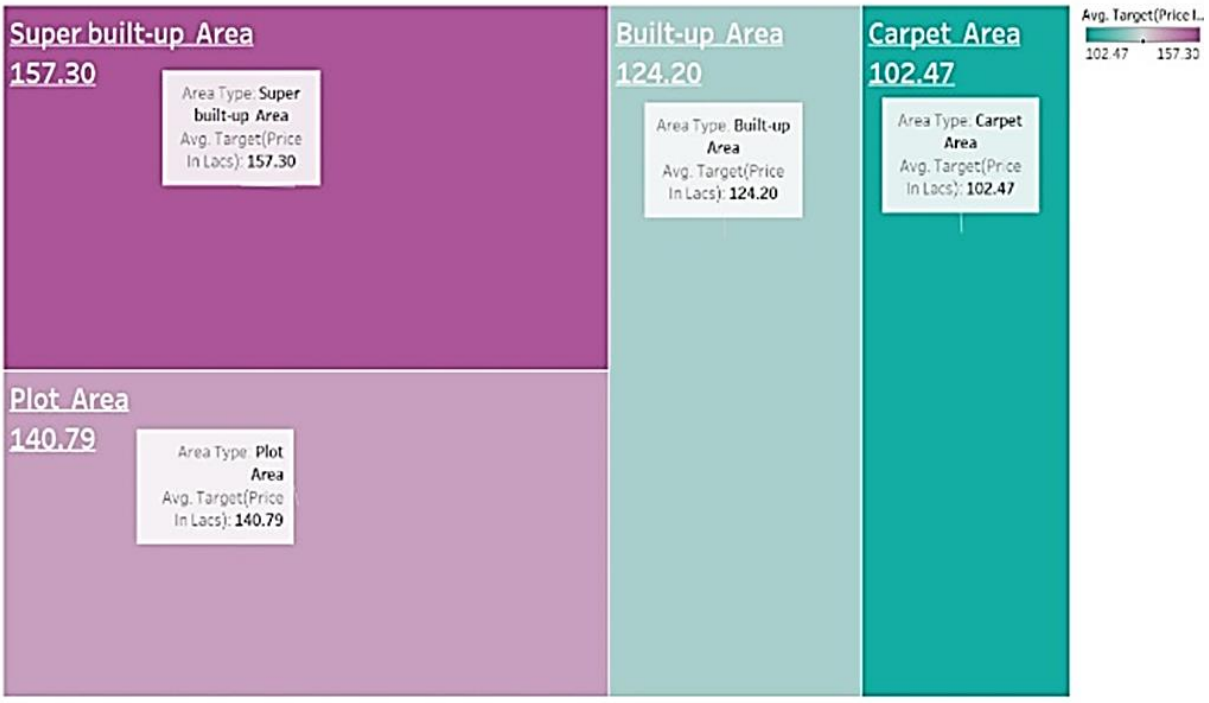

Figure 2. Treemap

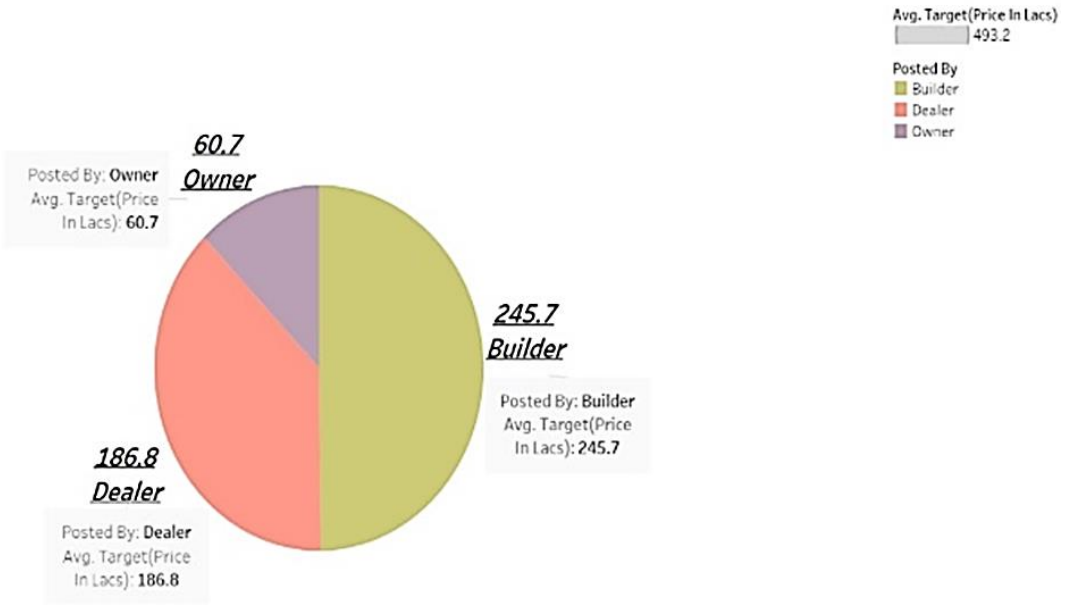

Figure 3. Pie chart

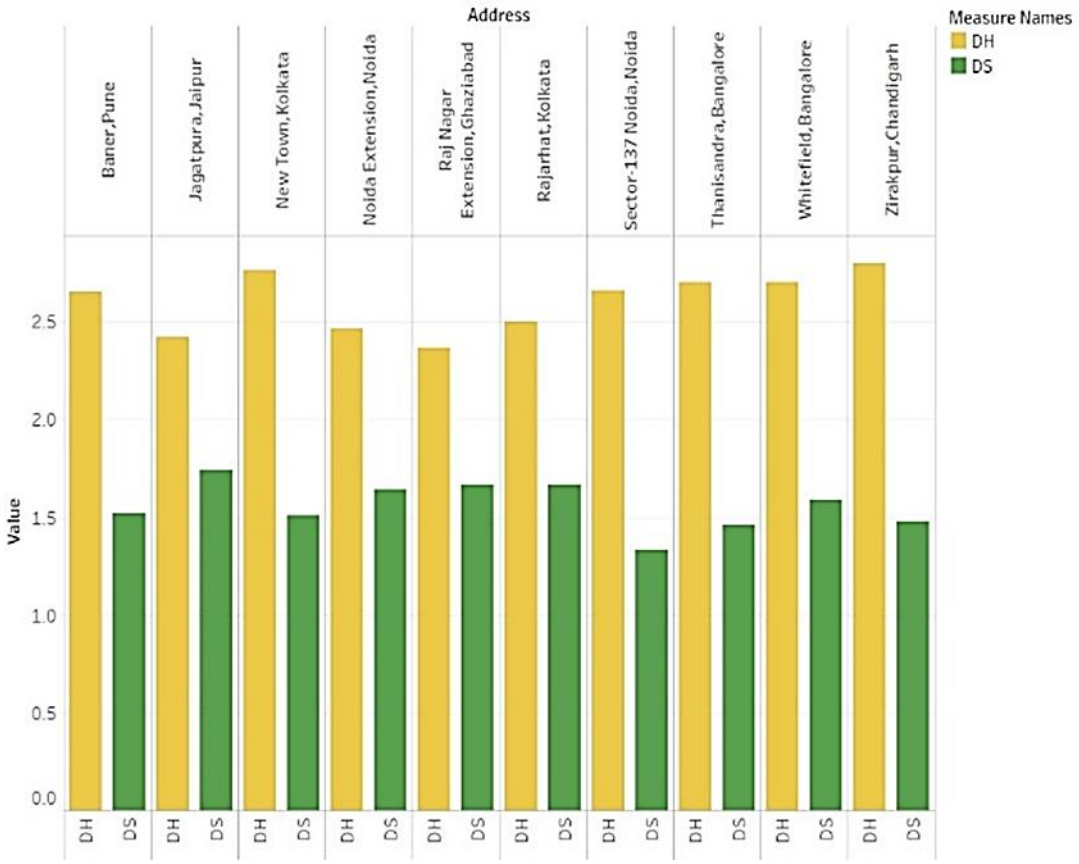

Figure 4. Double bar graph

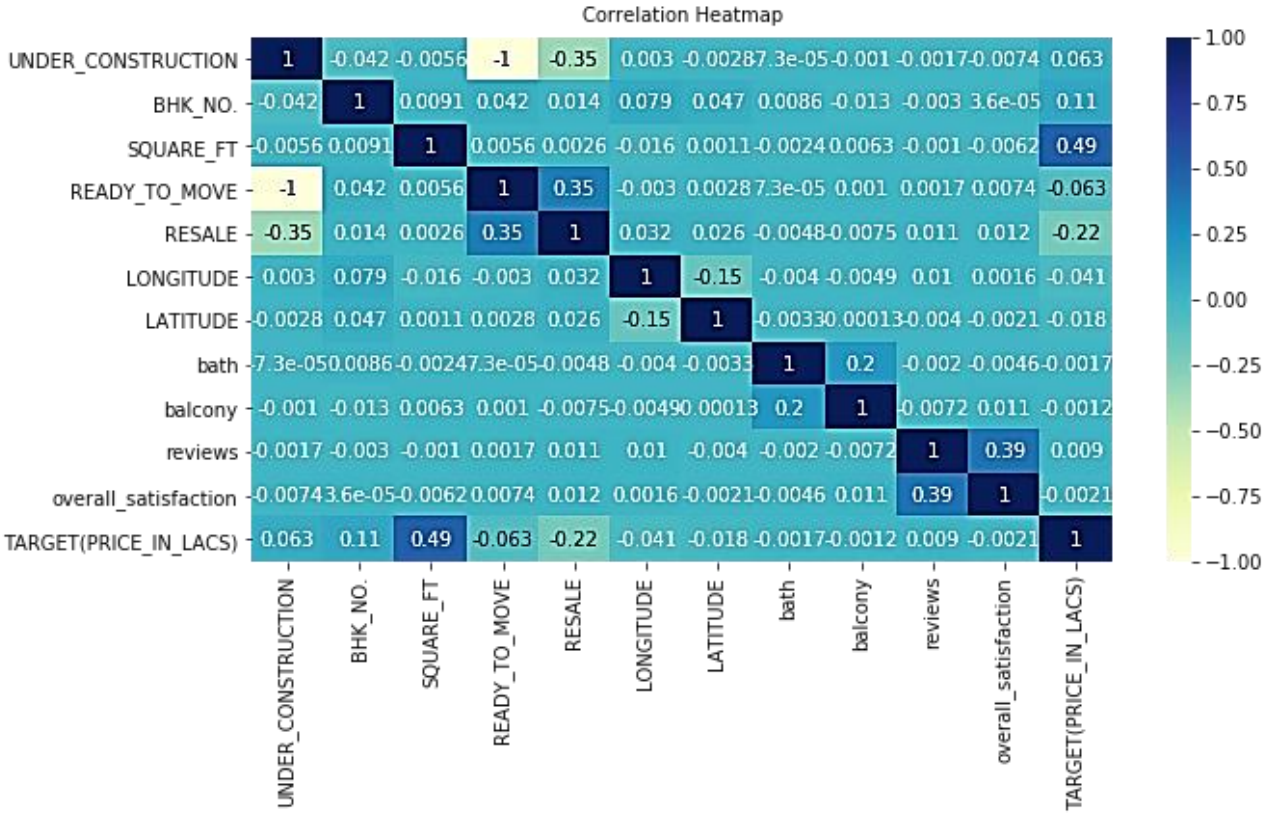

Figure 5. Heatmap showing correlation between features

*B. **Model fitting:***

From the Sklearn library of python, regression models are used for training. Test set results are predicted using the predict function.

*C. **Predicting accuracy:***

Predicting accuracy of each model and then selecting best algorithm for the application.

V. RESULTS

Using Multiple Linear Regression, we obtain the accuracy of 31.0383%, using Decision tree regressor we obtain the accuracy of 83.321% and Random Forest gives the accuracy of 84.248%. Hence, the predicted results that we obtain using Random Forest are more accurate than the other two Regression algorithms. Figure 6 shows bar graph comparing accuracy of each algorithm.

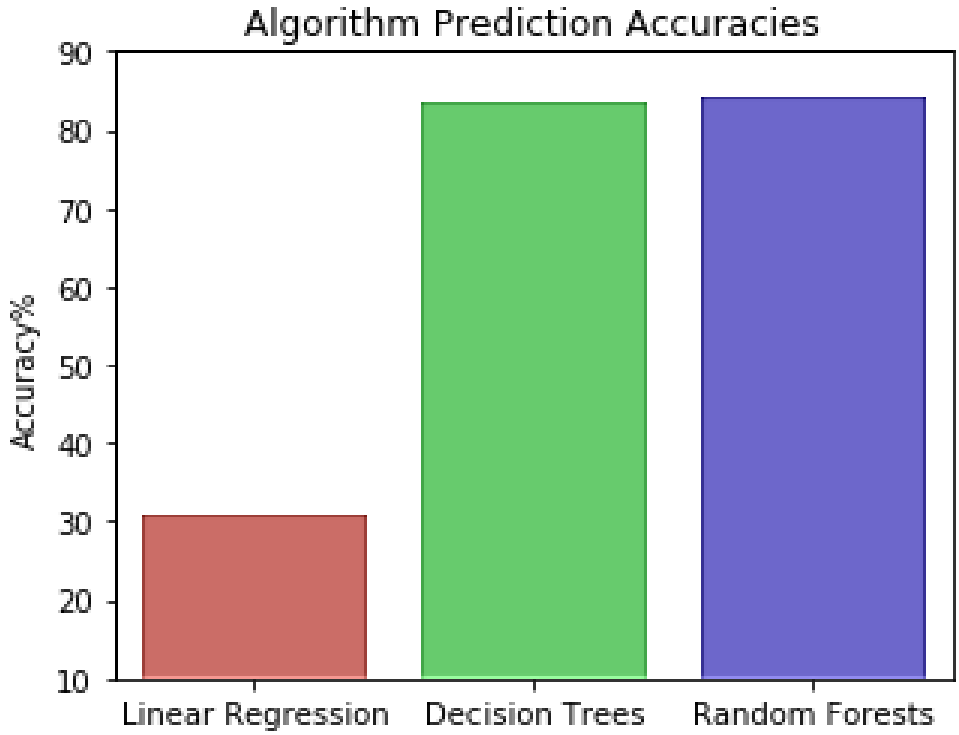

Figure 6. Bar graph comparing accuracy of each algorithm

VI. CONCLUSION

ML algorithms like linear regression, random forest regressor and decision tree regressor are used to build the prediction system so that it can predict the prices of real estate properties. Extra highlights like air quality and crime percentage were remembered for the dataset to help anticipate the costs far better. These features are not mostly included in the datasets of other prediction systems, which makes this system different. These features influence people's decision while purchasing a property, so why not include it in predicting house prices. This framework will fulfill needs of clients by giving them precise prices and prevent the danger of investing into any unacceptable property. The trained model is integrated with the User Interface using the Flask Framework. This application is 84% accurate for the dream home price prediction. It also includes visualizations to achieve better results and attract users. The prices have been calculated with precisely which would be of an incredible assistance for users.

In future, we will present a comparison of the app's prediction and the price from real estate websites such as Housing.com for the same user input. To make the system even more informative and user-friendly, we will be including Gmap. This will show the neighborhood amenities such as hospitals, schools surrounding a region of 1 km from the given location. A review system can also be developed which will be able to gather user's feedback so that the application can display the best suitable results to the user according to his choices and needs.

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical statement: The authors declare that they have followed ethical responsibilities.

REFERENCES

[1] Adair, A.S., Berry, J.N. and McGreal, W.S., 1996. Hedonic modelling, housing submarkets and residential valuation. Journal of property Research, 13(1), pp.67-83.

[2] Shiller RJ, Shiller J. Understanding Recent Trends in House Prices and Home Ownership", NBER Working Papers, 13553. InNational Bureau of Economic Research, Inc.. 98 economics review, issue no. 52 Schwarcz S.(2009)"Understanding the 'Subprime'Financial Crisis", South Carolina Law Review 2007.

[3] Lakshmi, B. N., and G. H. Raghunandhan. "A conceptual overview of data mining." In 2011 National Conference on Innovations in Emerging Technology, pp. 27-32. IEEE, 2011.

[4] Bhuju, G., Phaijoo, G.R. and Gurung, D.B., 2020. Sensitivity analysis of COVID-19 transmission dynamics. International Journal of Advanced Engineering Research and Application (IJAERA), 6(4), pp.72-82.

[5] Lakshmi, J. V. N. "Stochastic gradient descent using linear regression with python." International Journal on Advanced Engineering Research and Applications, Volume 2, Issue No. 7 (2016), pp. 519-524.

