JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Comparison of different ML algorithms for prediction of House Rental Prices

[1]Tarushi Vasishth, [2]Shruti Jain, [3]Anuradha Bhasin, [4]Arunima Kumari

[1]Department of Artificial Intelligence and Data Science,
[1]Bhagwan Parshuram Institute of Technology, Sector-17,Rohini, G.G.S.I.P.University, Delhi,India

Abstract : The ever-changing nature of real estate rental prices makes it a prime candidate for applying machine learning in price prediction. In this paper, a novel prediction model, Voting Regressor is proposed to forecast rental values based on current market conditions. It combines the strengths of various algorithms like Ridge Regression, Decision Trees, SVM Regressor, and Random Forest. The proposed prediction model is tested and validated on a housing dataset having 4746 records and 12 features, including physical characteristics and geographical information. This data is sourced from a leading real estate website, www.magicbricks. com, through Kaggle. Further, the model benefits from a rigorous data preprocessing pipeline encompassing data cleaning, outlier removal, feature engineering, scaling, and hyperparameter tuning. We have evaluated and compared the performance of different models employed to predict rental prices.

IndexTerms **- Ridge Regression, SVR, Decision Tree, Gradient Descent**

I.INTRODUCTION

Housing in India is a complex and diverse topic, reflecting the country's vast size and population. Housing in India ranges from luxurious high-rise apartments in megacities like Mumbai and Delhi to sprawling bungalows in suburban areas. In rural areas, traditional housing materials like mud, wood, and thatched roofs are still common. People in rural areas are migrating to urban areas in search of jobs which has resulted in an increased demand for rented houses. Despite a significant increase in housing units in recent decades, India still faces a housing shortage of nearly 20 million units. This is largely due to rapid urbanization and a growing population. This increased demand has led to an increase in the prices of rented houses. The location of a house largely determines its prices. It refers to the Area type, locality, and city where the house is located. Today, there are many House Rental Prediction Systems (HRPS) that are being used to predict the price of a house. The house renting system in India is evolving at a faster rate. Concepts include how close a house is to the hospital, market, educational institution, metro station, airport, etc. The prices of a rented house depend upon a range of variables including physical conditions, concept, and location of the house. Physical conditions refer to the number of bedrooms, size of the house, size of the hall, size of the kitchen, number of bathrooms, availability of backyard, age of the house, etc. We have analyzed different price forecasting strategies based on various machine learning algorithms such as Linear Regression, Ridge Re- gression, Decision Trees. This research paper is divided into 9 sections – Section 2 discusses related work, Section 3 discusses proposed work, Section 4 discusses methodology and implementation, Section 5 discusses Results and Discussions, Section 6 discusses the conclusion, and Section 7 discusses future work.

II. LITERATURE REVIEW

Real estate rental prices are highly dynamic, making their prediction a valuable application for machine learning.In this section, we have discussed the earlier work done in this domain. Dr. M. Thamarai,Dr. S P. Malarvizhi[1](2019) inves- tigated the use of decision trees for predicting aspects of the housing market in Tadepalligudem, India. The study focused on two key areas: availability and price. They explored whether decision trees, trained on factors like the num- ber of bedrooms, transport facilities, availability of schools, shopping facilities, and medical facilities, could predict if a house was for sale and how much it might cost. Pre-pruning and post-pruning were used to reduce the size of trees and avoid overfitting. Multiple Linear Regression was also used to predict the house prices. Performance metrics such as Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error were incorporated to evaluate the per- formance of both the classifier and regressor. Multiple Linear Regressor per- formed better than Decision Tree Regressor with a mean absolute error of just 1.95. Jian-qiang GUO, Shu-hen CHIANG, Min LIU, Chi-Chun YANG,Kai-yi GOU[2]2020) conducted a study to predict housing prices in Shanghai, China. They used three models, including the traditional linear regression model and two machine learning algorithms, Elastic Net Regression and Random Forest. Abigail Bola Adetunji, Oluwatobi Noah Akande, Funmilola Alaba Ajala, Olo- lade Oyewo, Yetunde Faith Akande, Gbenle Oluwadara[3](2021) This research paper probed the use of the Random Forest machine learning technique for house price prediction. The issue of overfitting was resolved by taking n_estimator pa- rameters as 500 and the efficiency of the model was increased by iterating the model several times. The Mean Absolute Error (MAE), R^2, and the Root Mean Square Error (RMSE)

were used as the performance metrics with an MSE value of 6.702. The model performed exceptionally well due to two factors: the use of K-fold cross-validation and the combined effect of multiple regressions. Winky K.O. Ho, Bo-Sin Tang & Siu Wai Wong [4](2021) investigated mainly three ma- chine learning algorithms, which include support vector machine (SVM), grid search in the random forest (RF), and gradient boosting machine (GBM) to evaluate the property prices. k-fold cross-validation technique was employed to evaluate the performance of SVM. Grid Search was used to enhance the accuracy of the model by hyperparameter tuning. Random Forest and Gradi- ent Boosting Machine were found to have more predictive power as compared to SVM when evaluated with three performance metrics: mean squared error (MSE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). Weldensie T. Embaye,Yacob Abrehe Zereyesus,Yacob Abrehe Zereye- sus [5](2021)investigated various machine learning techniques, including Ridge Regression, Lasso Regression, bagging, boosting, tree regression, and random forests, to predict rental values of houses. This paper has conducted a research to show that ML models provide superior prediction than the hedonic approach using OLS models(Ordinary Least Squares).The study found that Random For- est, Boosting and Bagging regression outperformed OLS approaches on basis of Mean Squared Error (MSE) results and Tree regression is the least performer in the group. Basysyar and Dwilestari [6](2022)compared the performance of Ridge Regression, Lasso Regression, and Elastic Net Regression with feature selection in their study.In this research paper,the irrelevant features were removed using RFE feature elimination tehcnique and in this way, the training and testing data were improvised.Following RFE,ridge regression and Lasso regression was applied.It has been shown that Lasso has outperformed Ridge for the unseen data.Then, Elastic net was used as a regularization technique,blending the L1 and L2 penalties of Lasso and ridge regressions.

Most of the research work was based on prediction of rental prices using ML traditional algorithms.In this paper,we have evaluated and compared different prediction models such as Linear Regression,Linear Regression with Gradient Descent, Ridge Regression, SVR, Decision Trees and Random Forests to predict optimal house rental prices.

III. PROPOSED WORK

We have conducted a comparative analysis of different regression techniques: Linear Regression, Ridge Regression, Decision Trees, SVR, and Random For- est. Linear Regression is a traditional machine learning algorithm that predicts house rent based on a straight line that best fits the data points. We have also employed SGDRegressor, short for Stochastic Gradient Descent Regressor, for linear regression to optimize the results further and Ridge Regression, which is a form of regularization to reduce overfitting and variance in the data and im- prove the overall result of Linear Regression. Decision Tree is utilized to predict continuous target variables and avoid overfitting by pruning. Support Vector Regressor helped us handle outliers and non-linear relationships, while Random Forest Regressor used the collective power of multiple decision trees. We per- formed hyperparameter tuning to adjust the values of hyperparameters that control the learning process in Decision Trees, SVR, and Random Forest. For Decision Trees and Random Forest, we adjusted the maximum depth of the tree, minimum samples per split, and minimum samples per leaf. Additionally, we used a maximum features hyperparameter for Random Forest. For SVR, we used gamma which controls the influence of training points on the decision func- tion and kernel(such as RBF and linear) as hyperparameters. Hyperparameter tuning was performed using GridSeacrhCV which leverages a Grid to define the values of hyperparameters and then tries different combinations of these hyper- parameters to get optimized results. Also, it splits the data into folds, trains the model on different folds with each combination of hyperparameters, and evalu- ates its performance on the remaining folds. This helps in avoiding the chances of overfitting.

IV. RESEARCH METHODOLOGY

4.1 Data Collection

To develop our model, we leveraged a rich dataset consisting of 4746 rows and 12 columns from kaggle.com. The dataset encompasses properties across 6 di- verse cities and their localities. Figure 1 provides a glimpse of our dataset which showcases that the dataset used is rich with various attributes, including the number of bedrooms (BHK), rental rates, property size, number of floors, area type, locality, city, furnishing status, tenant preferences, bathroom count, and contact information for the respective point of contact.

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
0	2022-05-18	2	10000	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2	Contact Owner
1	2022-05-13	2	20000	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1	Contact Owner
2	2022-05-16	2	17000	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1	Contact Owner
3	2022-07-04	2	10000	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1	Contact Owner
4	2022-05-09	2	7500	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1	Contact Owner
...	...	...	...	...	...	...	...	...	...	...	...	...
4741	2022-05-18	2	15000	1000	3 out of 5	Carpet Area	Bandam Kommu	Hyderabad	Semi-Furnished	Bachelors/Family	2	Contact Owner
4742	2022-05-15	3	29000	2000	1 out of 4	Super Area	Manikonda, Hyderabad	Hyderabad	Semi-Furnished	Bachelors/Family	3	Contact Owner
4743	2022-07-10	3	35000	1750	3 out of 5	Carpet Area	Himayath Nagar, NH 7	Hyderabad	Semi-Furnished	Bachelors/Family	3	Contact Agent
4744	2022-07-06	3	45000	1500	23 out of 34	Carpet Area	Gachibowli	Hyderabad	Semi-Furnished	Family	2	Contact Agent
4745	2022-05-04	2	15000	1000	4 out of 5	Carpet Area	Suchitra Circle	Hyderabad	Unfurnished	Bachelors	2	Contact Owner

4746 rows × 12 columns

Figure1. Dataset

4.2 Data Exploration and Preprocessing

Data exploration and Preprocessing are performed to extract meaningful in- sights and build robust model from data. An efficient analysis has helped to comprehend the underlying patterns more quickly. So, in the first step, we have employed statistical analysis in order to gain information about the key statiscal measures such as mean,standard deviation,count,etc. which further enriched our understanding of features of the dataset.We have ensured the data quality by checking for missing values but fortunately, the model didn't contain any missing values. It is evident from Figure 2, that there is an outlier in the given dataset. The outlier has been removed as visualised from Figure 3.

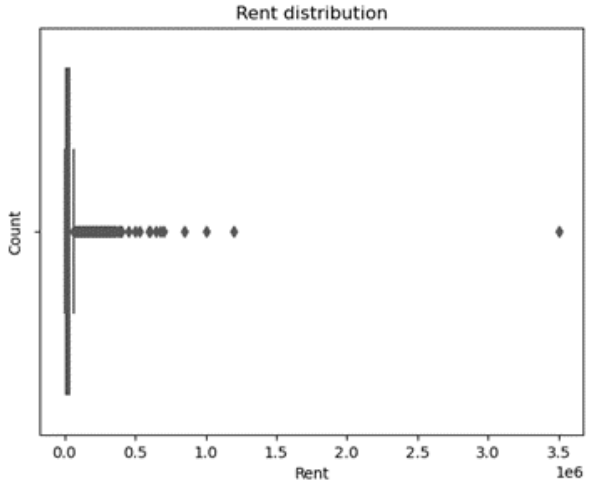


Figure 2. Before Outlier Removal

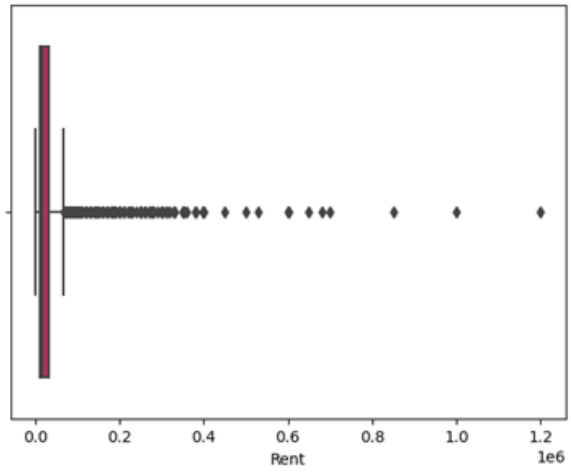
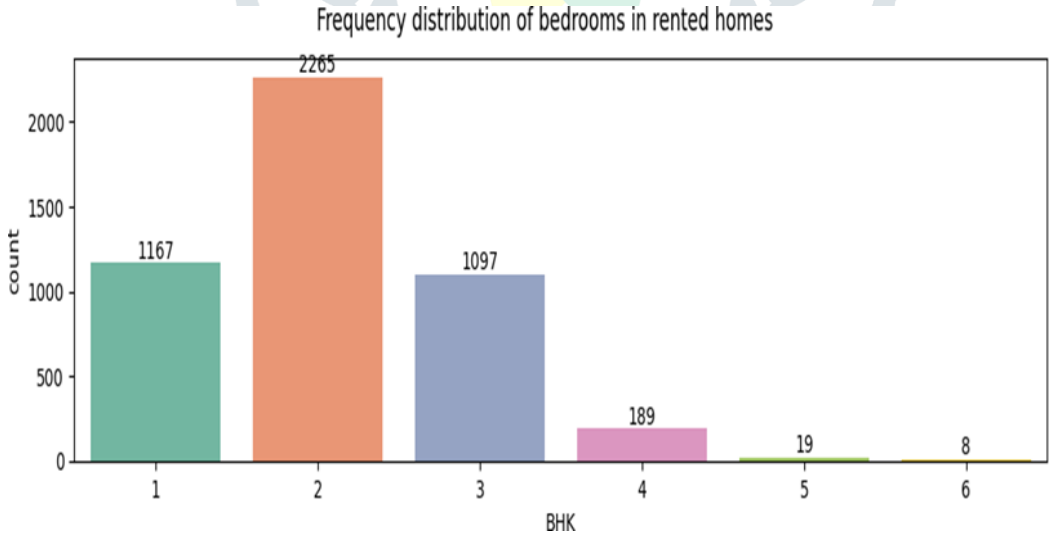
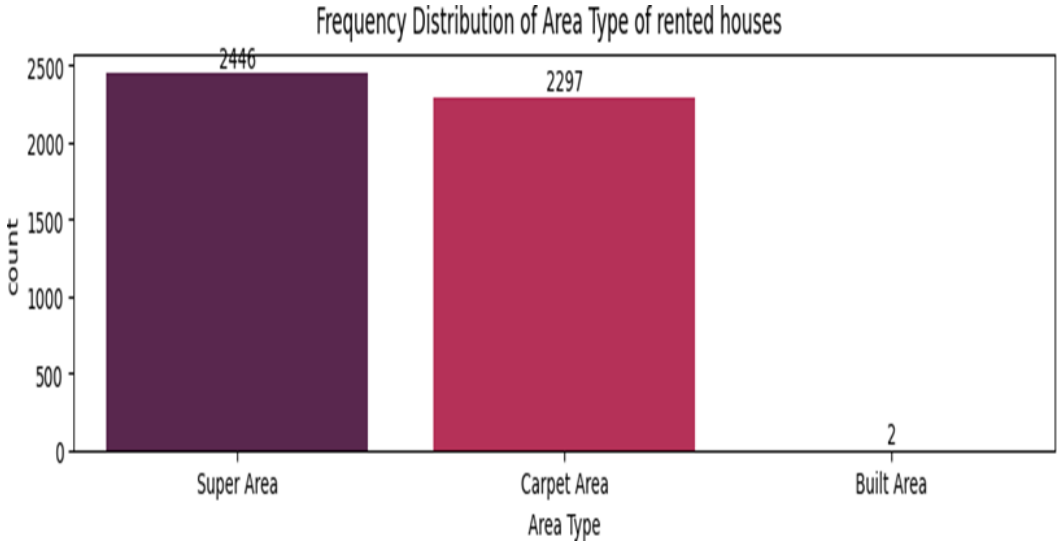


Figure 3. After Outlier Removal

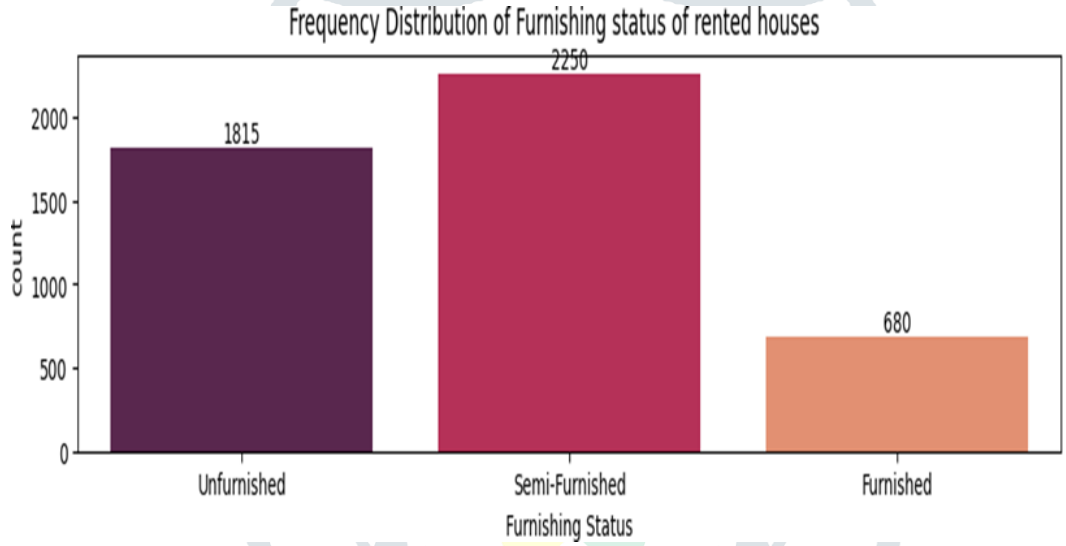
Then, we dropped the irrelevant features like 'Posted On,' 'Floor,' and 'Area Locality' that had minimal influence on rental prices and thereby, focussing only on the relevant features. We then conducted a univariate and bivariate analysis of the remaining features in relation to rent. This analysis utilized count plots, pie charts, and box plots to explore the distribution of each feature visually. The figure 4 represents the frequency distribution of different features. The frequency distribution is represented using count plots.



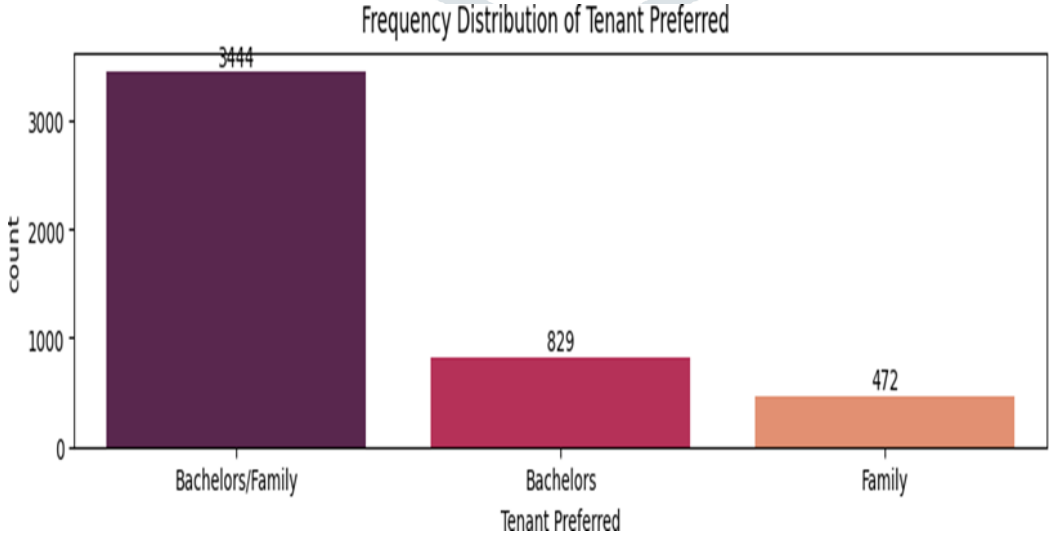
a)


Frequency Distribution of Area Type of rented houses
2446
2297
2
2500
2000
1500
1000
500
0
count
Super Area
Carpet Area
Built Area
Area Type

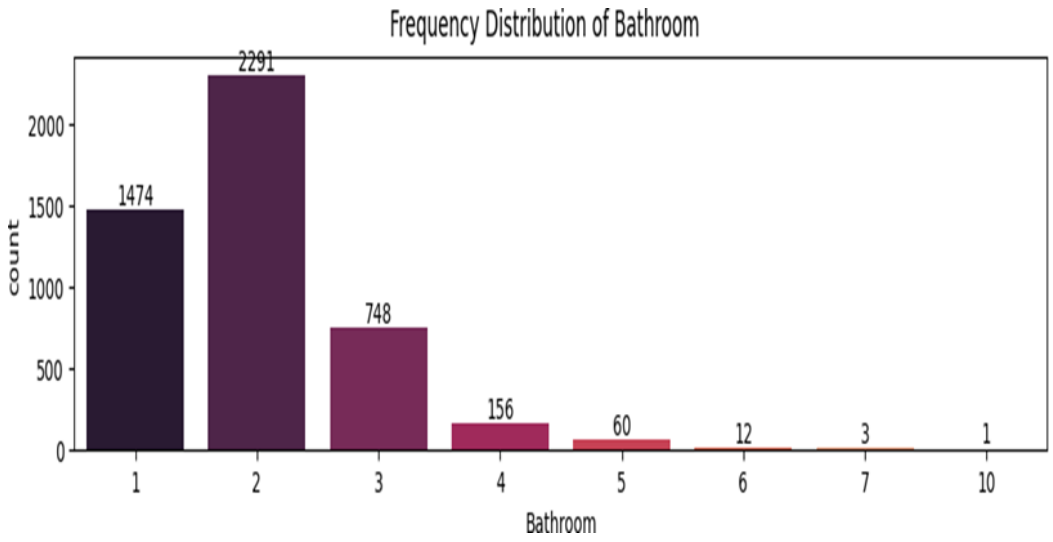
(b)


Frequency Distribution of Furnishing status of rented houses
1815
2250
680
2000
1500
1000
500
0
count
Unfurnished
Semi-Furnished
Furnished
Furnishing Status

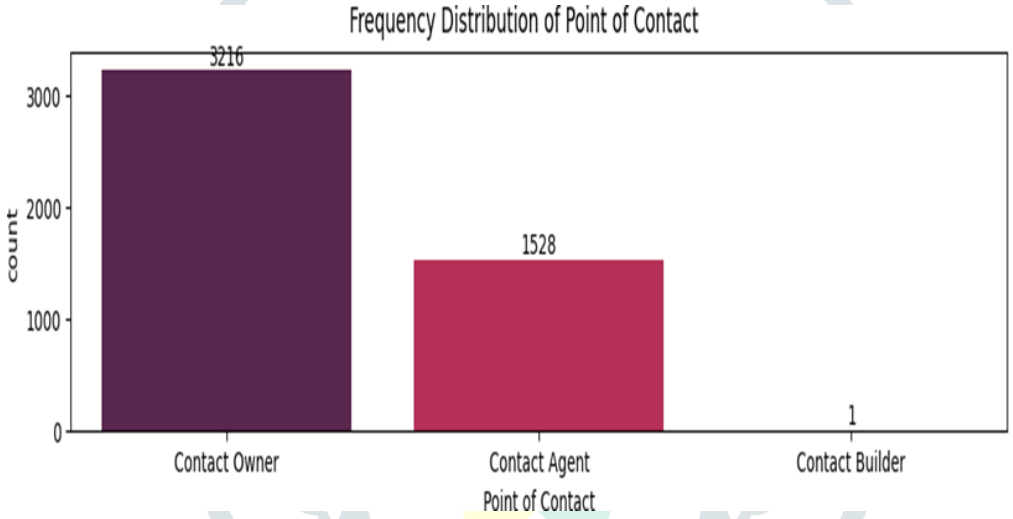
(c)


Frequency Distribution of Tenant Preferred
3444
829
472
3000
2000
1000
0
count
Bachelors/Family
Bachelors
Family
Tenant Preferred

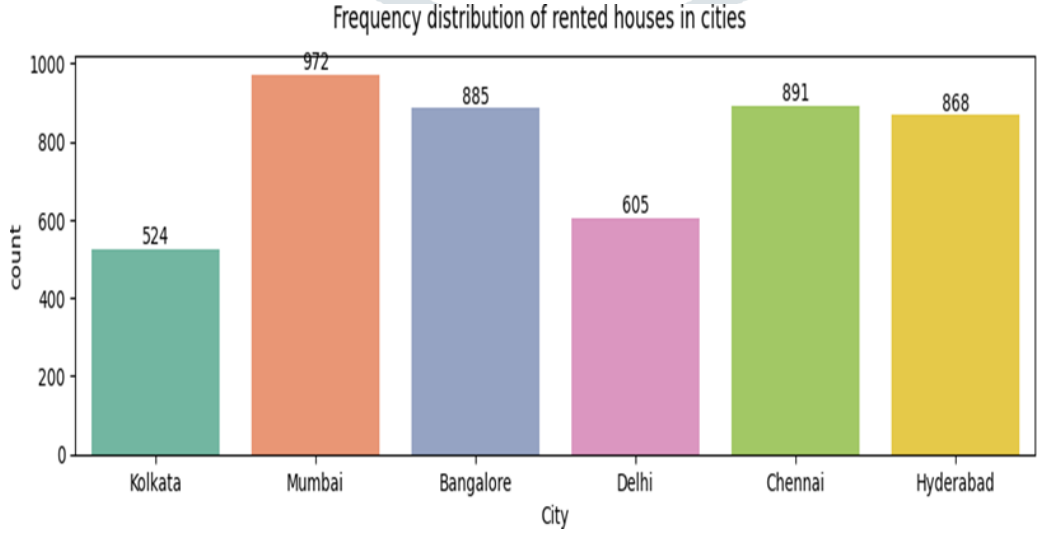
(d)



(e)



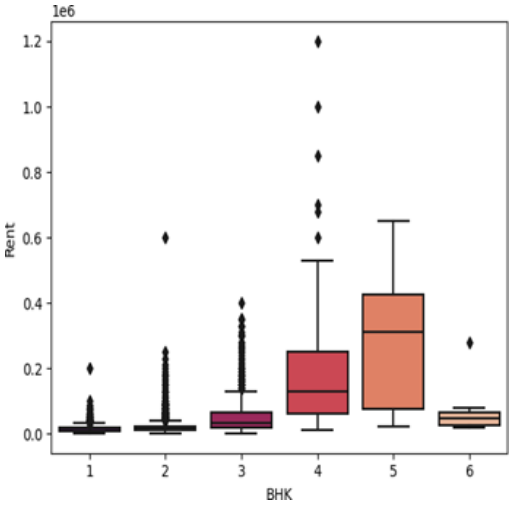
(f)



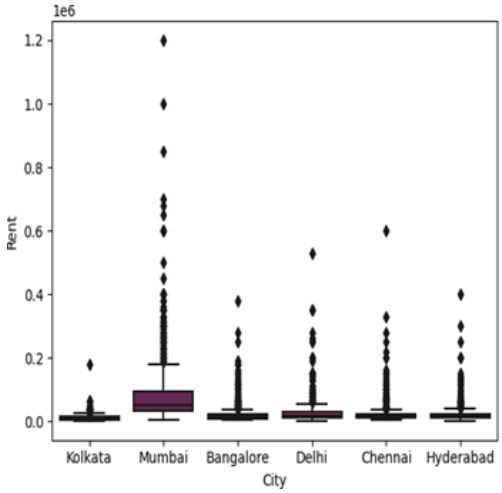
(g)

Figure 4. Count Plots showing Univariate Analysis

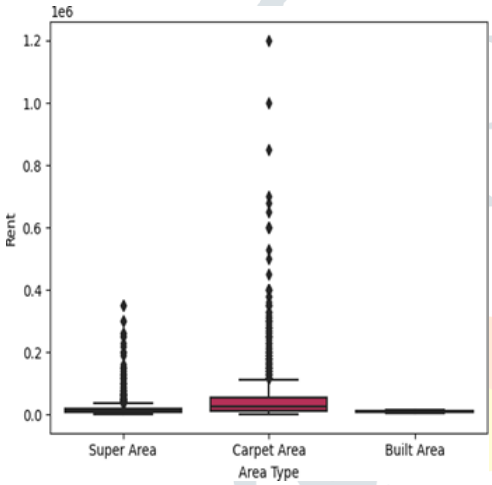
The above part consisted of univariate analysis. We then conducted a bivariate analysis to explore the relationship between individual features and rent. Figure 6 can be used to represent this.



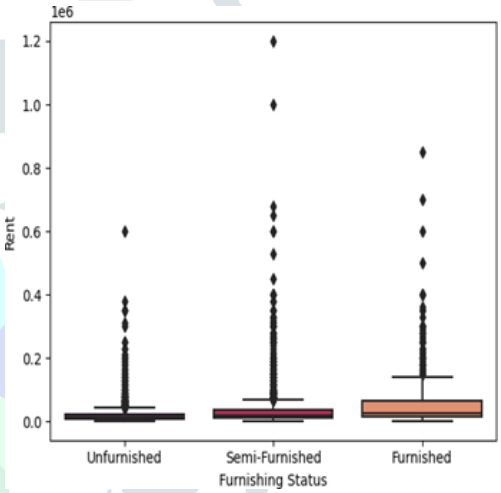
(a)



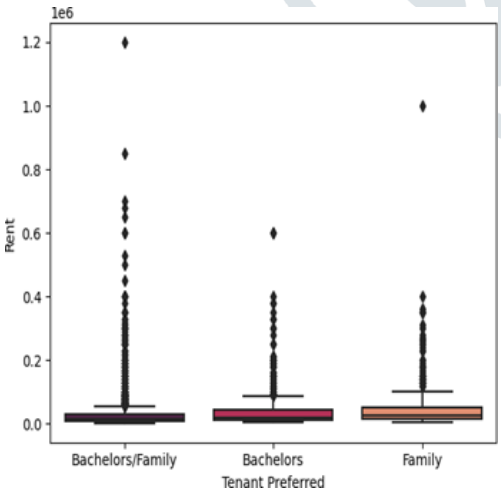
(b)



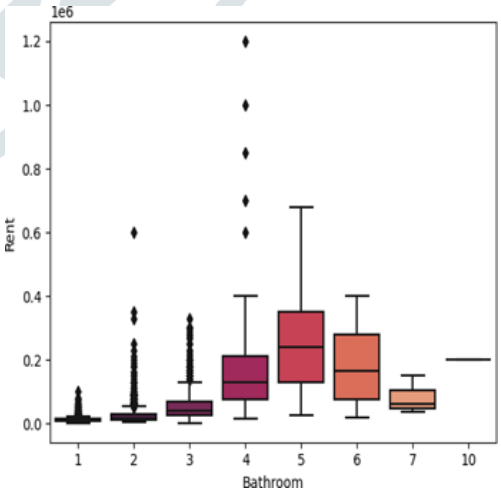
(c)



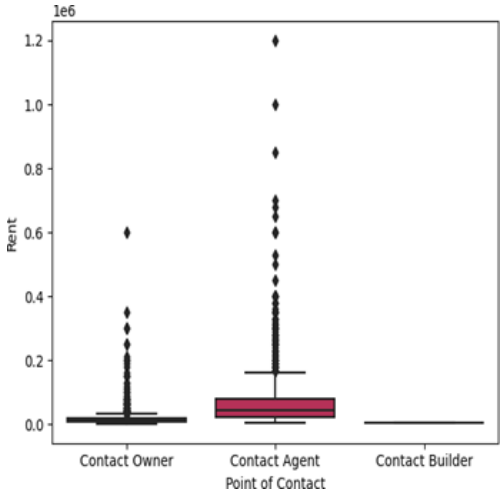
(d)



(e)



(f)



(g)

Figure 5.Box Plots showing Bivariate Analysis

We also performed correlation analysis using a heatmap to understand the potential correlation between all the features which can be seen in Figure 7.

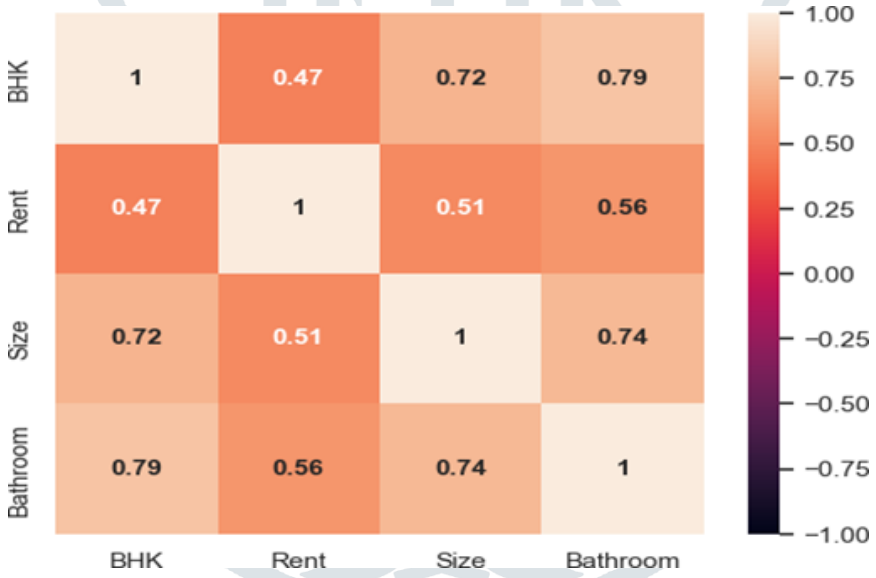
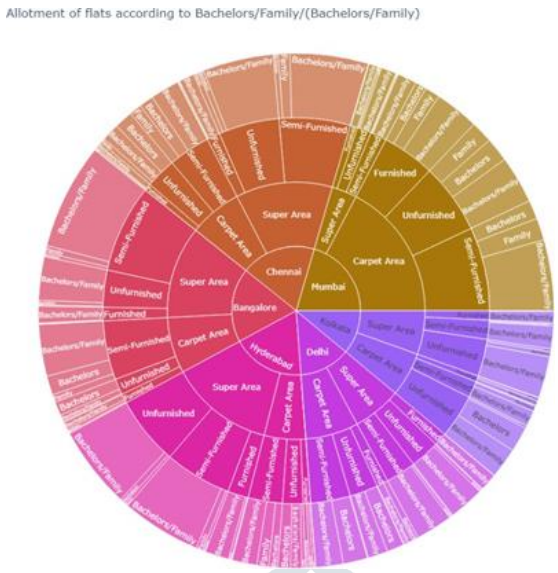


Figure 6. Heatmap

We made a sunburst chart for better visualization of the analyzed dataset.

Figure 7. Sunburst Chart



In the final stage of data preprocessing, we addressed three key tasks to ensure our machine learning model could function effectively. First, we tackled cate- gorical features, which represent qualities that fall into distinct categories (e.g., property type: apartment, house). These were encoded using a label encoder. This process assigns numerical labels to each category, allowing the model to understand the relationships between these features. Next, we split the dataset into two parts: training (75%) and testing (remaining 25%). The training set is used to train the model, essentially teaching it patterns within the data. The testing set, unseen by the model during training, is used to evaluate the model's generalizability and performance on new data. Finally, we scaled the data using a Standard Scaler. This process utilizes the Z-score normalization technique that normalizes the data by transforming each feature to have a mean of zero and a standard deviation of one. This ensures that all features contribute equally dur- ing training, preventing features with larger scales from dominating the model's learning process. By performing these steps, we create a well-prepared dataset that facilitates the training of a robust and accurate rent prediction model.

4.3 Model Development

In this step, we have incorporated the various ML algorithms that we have talked about in the previous sections.

Linear Regression: Linear regression that considers multiple variables is known as multiple linear regression or multivariate linear regression. Multiple linear regression builds upon the foundation of simple linear regression, allowing for the analysis of multiple independent variables simultaneously. The key difference lies in the equation of the two where a linear regression model consists of a single variable and coefficient whereas Multiple Linear Regression consists of multiple variables and coefficients. The equation for multiple linear regression is:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon \quad (1)$$

Here,

- Y represents the predicted outcome (house price in your example).
- X_1 to X_n are the various input features (number of bedrooms, age, etc.).
- β_0 is the intercept.
- β_1 to β_n are the regression coefficients for each corresponding feature, indicating the strength and direction of their influence on the outcome.
- ϵ represents the error term.

Gradient Descent: Training machine learning models involves an iterative pro- cess called gradient descent, which helps them fine-tune their parameters for optimal results.Here we are applying gradient descent on Linear Regression model. It is used in machine learning projects for updating the parameters of a model in order to minimize a cost function. In gradient descent, a key concept is the 'gradient,' which indicates the direction of steepest ascent for a function at a specific point. By iteratively moving in the opposite direction, the algorithm navigates towards lower function values, ultimately reaching the minimum.

Equations for Gradient Descent

cost function

$$J(\theta_0, \theta_1) = \quad 1/2m \sum_{i=1}^{m} \quad [h_\theta(x_i) - (y_i)] \quad ^2 \qquad (2)$$

gradient descent

$$\theta_j \ = \ \theta_j \ - \ \alpha \ \partial/\partial\theta_j \ J(\theta_0, \theta_1) \qquad (3)$$

From (2) and (3), we get

$$\partial J_\theta/\partial\theta \ = \ \partial/\partial\theta \ 1/2m \ \sum_{i=1}^{m} \quad [h_\theta(x_i) \ - \ (y_i)] \quad ^2 \qquad (4)$$

where,

- θj:weights of the hypothesis
- hθ(xi):predicted y value for ith input
- i: Feature index number(0,1,2,...,m)
- α: Learning Rate of gradient descent

Ridge Regression: Ridge regression tackles overfitting in linear regression by introducing a penalty term, ultimately leading to a model that generalizes bet- ter. We created a machine learning pipeline for fitting a regularized polynomial regression model and evaluating its performance using 5-fold cross validation. The cross-validation process provided a more robust estimate of the model's generalization performance on unseen data by training and testing on different splits of the data. We performed ridge regression with a regularization parameter (alpha) set to 10000 and the solver set to "lsqr" (least-squares solver). High alpha values in ridge regression enforce stricter control on coefficient magnitudes. This ultimately leads to a model with a more simplified structure.

Decision Trees: Machine learning algorithms known as decision trees leverage a branching structure, similar to a tree, to perform classification and regression tasks. Decision has a structure similar to that of the tree data structure.A key challenge in building decision trees lies in identifying the most informative at- tribute to split data at each node, including the root.To solve this problem there are two popular techniques. Decision Trees Machine learning algorithms known as decision trees leverage a branching structure, similar to a tree, to perform classification and regression tasks. Decision has a structure similar to that of the tree data structure.A key challenge in building decision trees lies in identifying the most informative at- tribute to split data at each node, including the root.To solve this problem there are two popular techniques.

Information Gain

- IG = Entropy(S) - [(Weighted Avg)*Entropy(each feature)]
- Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no
- Gini index

Gini index=1 − Σj P_J 2 Here, we employed a cost complexity pruning technique to selectively trim the tree using an optimized value of alpha, in order to prevent overfitting. An optimized value of alpha was selected from an array of alphas using a line plot showing the variation of test and training data with alpha.We also applied hyperparameter tuning to improve the results further.For hy- perparameter tuning, we used cross validation.The parameters that we adjusted are maximum depth of the tree, minimum number of samples split and minimum number of leaf nodes.

Support Vector Machine: SVM is mostly used for classification tasks but it can also be utilized for regression tasks (support vector regressor, SVR) where input x can be mapped onto an m-dimensional feature space employing non- linear mapping, to construct a linear model in the feature space. SVR aims to find a hyperplane (a straight line in 2D or a higher-dimensional plane in higher dimensions) that best fits the training data while maintaining a certain margin of error. This margin of error is defined by a tolerance parameter ϵ.Again hyperparameter tuning was employed to reduce overfitting. We implemented grid search for hyperparameter tuning of a support vector regression model. Grid Search consists of a grid that utilizes various hyperparameter combinations to yield best results.In our model, we used the combination of 2 different kernels of SVM i.e., linear and radial basis function. Also, we employed cross validation consisting of 5-folds to further tune the hyperparameters. The hyperparameter that we tried to tune is gamma that is specific for the RBF kernel and controls the influence of distant data points. It's only relevant if the 'rbf' kernel is chosen.

The equations used in SVM algorithm for regression are: The ϵ-insensitive loss function for a data point is:

$$\square\square(\square\square, \square(\square\square)) = \square\square\square(0, |\square\square - \square(\square\square)| - \epsilon) \quad (6)$$

where,

- yi: The actual target value for the i-th data point.
- f (xi: The predicted value for the i-th data point by the SVR model (linear function in its most basic form).
- ϵ: The tolerance parameter defining the margin of error,

minimize: $1/2||w||^2 + c\sum (max(0, |y_i - f(x_i)| - \epsilon)) \quad (7)$

- w^2: Represents the squared norm of the weight vector (w) defining the hyperplane. This term penalizes complex models (large w) to avoid overfit- ting.
- c : A hyperparameter controlling the trade-off between maximizing the mar- gin and minimizing the ϵ-insensitive loss. A higher C prioritizes a larger margin, potentially leading to less tolerance for errors.

Random Forest: Random forest regression is a powerful machine learning technique used for predicting continuous values (regression tasks). It combines the predictions of multiple decision trees to generate a more robust and accurate final prediction. The chances of overfitting are also less. Hyperparameter tuning using Grid Search and cross validation has also been incorporated. The hyperparameters that we have used are maximum depth, minimum number of samples split, minimum number of leaf nodes, and maximum number of features.

V. RESULT AND DISCUSSION

The evaluation is based on two performance metrics, Mean Absolute Error (MAE) and R2 squared error. A model that has a lower value of MAE and a higher value of R2 is considered to be a good fit. A detailed comparison among already existing ML techniques used to predict house rental prices has also been performed.

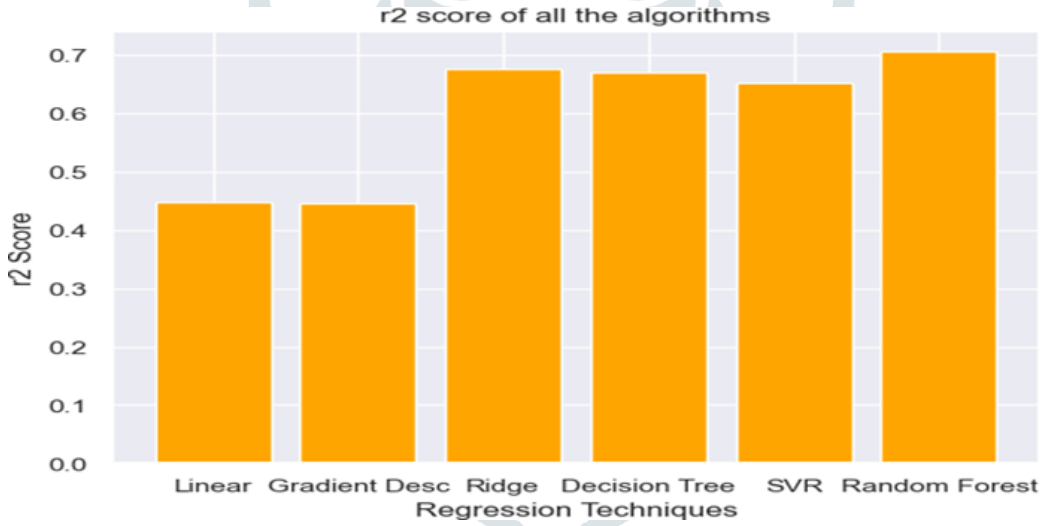


Figure 8. r2 score

As seen from Figure 8, it can be concluded that the Random Forest algorithm outperforms other algorithms with an MAE of 0.198 and an R2 score of 0.704.

VI. CONCLUSION

The study evaluated six different machine learning algorithms for predicting property prices using housing data. These algorithms were Linear Regression, Stochastic Gradient Descent (SGD)-based Linear Regression, Ridge Regression, Decision Trees, Support Vector Regressor (SVR), and Random Forest. It has been observed that the proposed novel approach i.e. Random Forest outperformed the others in generating precise price estimations with significantly lower prediction errors.

VII.FUTURE WORK

Employing deep learning techniques like neural networks could be highly beneficial. Neural networks, with their ability to capture complex relationships in data, might uncover hidden patterns in the data, potentially leading to more accurate rent predictions, especially for complex or time-dependent factors. Exploring LSTMs for historical trends is particularly promising.

REFERENCES

[1] Thamarai, M. and Malarvizhi, S.P., 2020. House Price Prediction Modeling Using Machine Learning. International Journal of Information Engineering & Electronic Business, 12(2).

[2] Guo, J.Q., Chiang, S.H., Liu, M., Yang, C.C. and Guo, K.Y., 2020. Can ma- chine learning algorithms associated with text mining from internet data improve housing price prediction performance?. International Journal of Strategic Property Management, 24(5), pp.300-312.

[3] Adetunji, A.B., Akande, O.N., Ajala, F.A., Oyewo, O., Akande, Y.F. and Oluwadara, G., 2022. House price prediction using random forest machine learning technique. Procedia Computer Science, 199, pp.806-813.

[4] Ho, W.K., Tang, B.S. and Wong, S.W., 2021. Predicting property prices with machine learning algorithms. Journal of Property Research, 38(1), pp.48-70.

[5] Embaye, W.T., Zereyesus, Y.A. and Chen, B., 2021. Predicting the rental value of houses in household surveys in Tanzania, Uganda and Malawi: Evaluations of hedonic pricing and machine learning approaches. Plos one, 16(2), p.e0244953.

[6] Basysyar, F.M. and Dwilestari, G., 2022. House price prediction using exploratory data analysis and machine learning with feature selection. Acadlore Trans. AI Mach. Learn, 1(1), pp.11-21.

[7] Hu, L., He, S., Han, Z., Xiao, H., Su, S., Weng, M. and Cai, Z., 2019. Monitoring housing rental prices based on social media: An integrated approach of machine- learning algorithms and hedonic modeling to inform equitable housing policies. Land use policy, 82, pp.657-673.

[8] https://www.magicbricks.com/

[9] https://www.kaggle.com/

[10] https://www.geeksforgeeks.org/

[11] https://www.analyticsvidhya.com/

[12] https://scikit-learn.org/

[13] https://www.javatpoint.com/