Predicting House Prices with a Linear Regression Model

Vikas.K.Totiger, Chanakya University,Bengaluru,school of Engineering ,vikastotiger2003@gmail.com
Shivakumar.T.V, Chanakya University,Bengaluru,school of Engineering ,shivakumarvt77@gmail.com

Abstract
House price forecasting is a significant area of economic and social research. Among the various forecasting methods, linear regression is widely adopted due to its simplicity, interpretability, and computational efficiency. This paper investigates the effectiveness of linear regression models in predicting housing prices. It begins with an introduction to the fundamental theory of linear regression, followed by an analysis of key factors influencing housing prices. A predictive model is then developed and evaluated using real-world data. The results demonstrate that linear regression can accurately predict housing prices in cases with relatively simple and distinct variables—such as the presence or absence of specific facilities. However, the study also highlights the limitations of linear regression, especially in complex scenarios. The paper concludes that while linear regression offers a useful baseline, more sophisticated models may be necessary for broader and more accurate housing price prediction, which will be explored in future research.

Keywords: Real estate market research, Predicting house prices, Linear regression Model,Random Regression Forest, Logistic Regression model,support vector machine,XGBoost

1. Introduction

For many individuals and families, residential housing is one of the most important living resources and assets. In urban environments, houses not only provide shelter but also represent a significant investment and financial opportunity [1]. Consequently, housing prices are key indicators reflecting the economic condition and market vitality of a country or region. Variations in housing prices often draw widespread attention, influencing both national real estate planning and government policymaking [2].

On a personal level, housing prices directly impact individual financial planning, as property typically constitutes a major component of household wealth. Forecasting housing prices has always posed a major challenge, owing to the complex, dynamic, and volatile nature of real estate markets. As urbanization accelerates and market factors become increasingly multifaceted, developing accurate housing price forecasting models becomes ever more critical. An effective forecasting model not only assists government authorities in formulating rational real estate policies, but also supports developers,investors, and buyers in making informed decisions. A reliable model can serve as a scientific and rational reference tool for stakeholders at every level of the real estate ecosystem.

According to previous research, house price prediction is a complex, multidimensional problem that has been addressed using various approaches, including statistical methods, machine learning, and deep learning [3]. With the rise of big data, the integration of heterogeneous data sources and advanced predictive algorithms has become a focal point of ongoing research [4]. Emerging research directions include multi-source data fusion, model interpretability, real-time prediction, and cross-regional or transnational applications [5]. Due to the wide array of modeling techniques, researchers often compare different models—ranging from economic to statistical to machine learning—to identify the most optimal and accurate solutions [6]. Among these, machine learning has gained popularity due to its capacity to model complex, nonlinear relationships and handle large-scale datasets. Techniques such as decision trees, random forests, and neural networks are widely applied in housing price prediction [7][8]. For instance, researchers have used machine learning to analyze daily price data from online real estate portals in the UK [7], and to construct housing price indices for hundreds of cities in China [8]. Traditional statistical models, particularly linear regression, remain foundational due to their interpretability and solid theoretical grounding, even though they may lack the flexibility of newer techniques [9].

However, many existing studies emphasize macro-level price trends, providing limited value for individual-level housing decisions [10]. To address this gap, our study focuses on forecasting the price of individual residential properties—a model that better serves prospective buyers and investors on a personal scale.

In this study, we implement and compare two predictive models: **Linear Regression** and **Ridge Regression**, **Random Forest ,Support Vector Machine, XGboost , Logistical Regression** to forecast the price of single housing units. These models were chosen for their interpretability and effectiveness in establishing linear relationships between housing features and market price. Our implementation uses a publicly available housing dataset, preprocessed to handle missing values and scaled for consistency. Key predictive features include **total area, number of bedrooms, presence of water heaters, and number of functional rooms**, among others.

After training and testing both models, we evaluate them based on **R² score, mean squared error (MSE), and visual analysis of predicted vs. actual prices**. Our findings indicate that both models effectively identify the influence of various features on house price, with ridge regression providing slightly improved performance due to its ability to handle multicollinearity.

Furthermore, our analysis demonstrates that regression models can provide interpretable and consistent forecasts across various property configurations. For instance, we observe clear, logical pricing differences between homes with different room types and utilities, offering practical value for buyers assessing properties in similar locations.

2. Methodology

2.1. The standard linear regression model

Linear regression is a common statistical method used to analyze linear relationships between two or more variables. Its basic form is to establish a linear relationship between a dependent variable (in this study house prices) and one or more independent variables (factors affecting house prices) .The mathematical expression of multivariate linear regression is shown as follows:

$$y = \beta 0 + \beta 1 x 1 + \beta 2 x 2 + \cdots + \beta p x p + \epsilon \quad (1)$$

Among them, y is the dependent variable (house price), x2,x2,...,xp are independent variables (such as housing size, whether there is a bathroom, etc.),β0,β1,...,βp are the model parameters, ϵ is the error

term. Linear regression is simple and intuitive, but requires high linear assumptions of data [.

The application of linear regression model relies on the following basic assumptions:

Linear relationship: There is a linear relationship between the dependent variable and the

independent variable.

Independence: The observations are independent of each other.

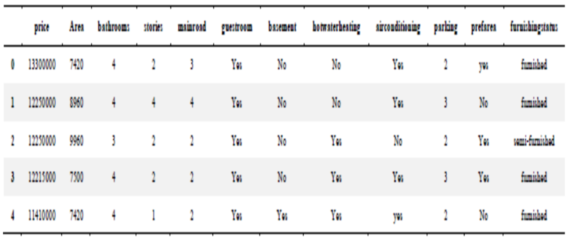
Homoscedasticity: Different values of the independent variable correspond to the same variance of the dependent variable.

Normality:The error term follows a normal distribution.No multicollinearity: There should be no strong linear correlation between independent variables.These assumptions provide a theoretical basis for the validity and reliability of the linear regression model. The reliability of a model depends on whether these assumptions are met. In practice, deviations from assumptions may affect the forecasting performance of the model .

2.2. Preparation of data

In this study, we used a publicly available dataset that contains information about various houses and their corresponding prices. The dataset includes features such as the area of the house, the number of bedrooms, the number of bathrooms, and other relevant characteristics. These features will be treated as independent variables, while the house prices will serve as the dependent variable.

The data used in this study is derived from the Kaggle dataset "Housing Prices Dataset", which is a common dataset used in predictive modeling tasks related to real estate [13]. The data includes a wide range of physical characteristics of houses, which makes it ideal for building a linear regression model.



	price	Area	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	Yes	No	No	Yes	2	yes	furnished
1	12250000	8960	4	4	4	Yes	No	No	Yes	3	No	furnished
2	12250000	9960	3	2	2	Yes	No	Yes	No	2	Yes	semi-furnished
3	12215000	7500	4	2	2	Yes	No	Yes	Yes	3	Yes	furnished
4	11410000	7420	4	1	2	Yes	Yes	Yes	yes	2	No	furnished

Table 1. The Kaggle dataset "Housing Prices Dataset" used in the research.

Because data sets are now getting larger and larger, there will be inconsistencies in the arrangement and format of data sets. Once the dataset is selected, it is important to clean the data to ensure that

it is suitable for modeling. The following steps are typically involved in the data cleaning process :Handling Missing Values: Missing values are common in real-world datasets and they can adversely affect the performance of the linear regression model. There are several ways to handle missing values, such as imputing them with the mean or median of the variable, using more sophisticated techniques like K-nearest neighbors (KNN) imputation, or simply removing rows or columns with missing data. In our research, for the numerical data, we use the method of finding the median to supplement the missing values, and for the data to judge whether we have, we directly discard the method used in this study Outlier Detection and Treatment: Outliers can have a significant impact on the performance of the linear regression model, as they can distort the relationship between the variables. Outliers can be detected using methods such as box plots or z-scores, and they can be dealt with by either removing them or transforming the data. We discard outliers and take data that do not significantly affect the regression model to build and test the model Normalization and Scaling: In order to improve the convergence of the model and ensure that all

variables are on a comparable scale, it is often necessary to normalize or scale the data. This is especially important when using gradient-based optimization methods. Common techniques for normalization include min-max scaling and z-score normalization.

2.3. Building linear regression model

2.3.1. Univariate linear regression

First, we used a single independent variable to construct a linear regression model. We assume the area of the house as the only independent variable, and the model is shown as the following:

$y = \beta 0 + \beta 1$ × house area + ϵ (2)

This study uses the least squares (OLS) to estimate the parameters and in the model. With the training data, we can obtain estimates of the intercept and regression coefficients.

2.3.2. Multivariable linear regression

Univariate models use a limited number of variables to predict a single characteristic. In order to improve the prediction accuracy of the model, the program introduces several features and constructs a multivariate linear regression model. Multiple characteristics are accounted in this model (e.g. area,number of bedrooms, number of bathrooms, etc.), and the model is shown as the following:

$y = \beta 0 + \beta 1 x 1 + \beta 2 x 2 + \cdots + \beta n x n + \epsilon$ (3)

Similarly, this research use the least squares method to estimate the various regression coefficients and fit the model with the training data.

2.3.3. Model evaluation

Model evaluation is a key step in measuring the quality of a model. Commonly used evaluation indicators include Mean square error (MSE), The coefficient of determination and Adjusted R-squared. In this research, we used all three parameters for evaluation. The formulas to calculate these parameters are listed as follows:

Mean square error (MSE): MSE measures the difference between the predicted value and the true value, and is calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The coefficient of determination*:* The coefficient of determination is the proportion that explains the fluctuation of the dependent variable, and the closer the value is to 1, the more explanatory power the model has.

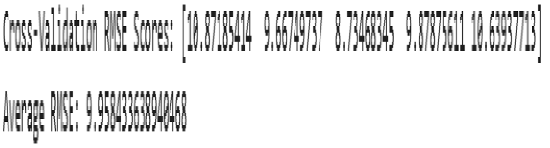
$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Adjusted R-squared: Unlike R-squared which increases with the number of variables, the adjusted R-squared accounts for the number of predictor variables and increases only if the new variable improves the model. These metrics help us determine the accuracy and reliability of the linear regression model. Depending on the results, we may need to fine-tune the model or try alternative approaches.

3. Implementation

3.1 K-Fold Cross-Validation

In this code, a regression dataset is first generated using the make_regression function from scikit-learn, which creates 100 samples with 5 input features and adds Gaussian noise to simulate real-world variability. The purpose of this setup is to evaluate the performance of a linear regression model. The model used is scikit-learn's LinearRegression, and its performance is assessed using **K-Fold Cross-Validation** with 5 folds. This technique splits the dataset into 5 subsets (folds), ensuring each fold is used once as the test set while the remaining serve as the training set, thereby improving the reliability of the performance estimate. The cross_val_score function is used with the scoring metric set to negative mean squared error (neg_mean_squared_error) since scikit-learn uses a convention where loss functions return negative values. These negative MSE scores are then converted to positive Root Mean Squared Error (RMSE) values by taking the square root of their negated values. RMSE is a more interpretable metric in regression as it provides an estimate of the prediction error in the same units as the target variable. Finally, the script prints out the RMSE values from each fold and the average RMSE, which represents the model's overall performance across the dataset.

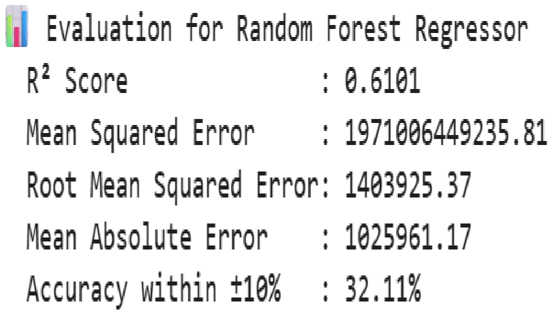


```
Cross-Validation RMSE Scores: [10.87185414  9.66749737  8.73468345  9.87875611 10.63937713]

Average RMSE: 9.958433638940468
```

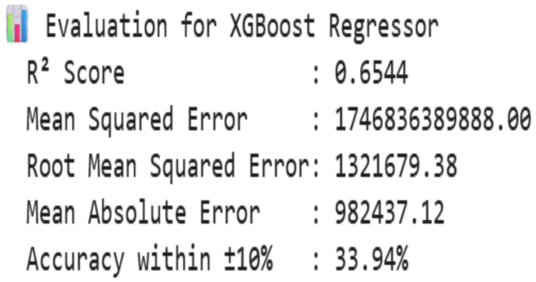
3.2 Models

3.2.1 Random Forest Regressor : was first applied, a widely-used ensemble learning method that constructs a large number of decision trees and averages their outputs to improve prediction accuracy and reduce overfitting. Its performance was evaluated using the evaluate_regression function, which likely computes standard regression metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R²).



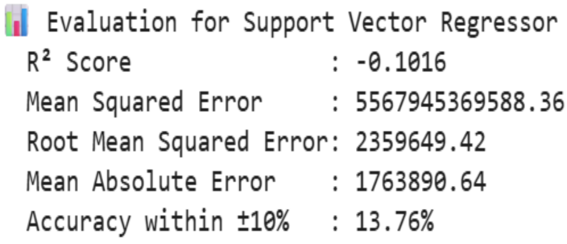
```
📊 Evaluation for Random Forest Regressor
  R² Score                : 0.6101
  Mean Squared Error      : 1971006449235.81
  Root Mean Squared Error: 1403925.37
  Mean Absolute Error     : 1025961.17
  Accuracy within ±10%    : 32.11%
```

3.2.2XGBoost Regressor : a highly efficient and scalable implementation of gradient boosting was implemented. Configured with 100 estimators, a learning rate of 0.1, and a maximum depth of 4, the model was trained on standardized features (X_train_scaled) to enhance convergence and accuracy. XGBoost is particularly effective in handling complex nonlinear relationships and often outperforms traditional models in structured data tasks. Its predictions (y_pred_xgb) were then assessed using the same regression evaluation function to ensure consistency in comparison.



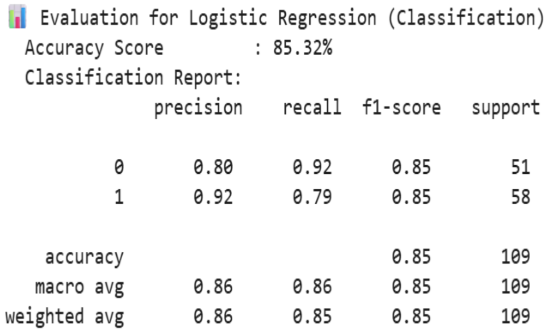
```
📊 Evaluation for XGBoost Regressor
  R² Score                : 0.6544
  Mean Squared Error      : 1746836389888.00
  Root Mean Squared Error: 1321679.38
  Mean Absolute Error     : 982437.12
  Accuracy within ±10%    : 33.94%
```

3.2.3 Support Vector Regressor (SVR): was evaluated. SVR, a variant of Support Vector Machines, attempts to fit the best possible hyperplane within a specified margin of tolerance. It is particularly effective when the relationship between input features and the target variable is nonlinear. The model's predictions (y_pred_svm) were evaluated using the regression metrics to understand its predictive accuracy and robustness.



```
📊 Evaluation for Support Vector Regressor
  R² Score                : -0.1016
  Mean Squared Error      : 5567945369588.36
  Root Mean Squared Error: 2359649.42
  Mean Absolute Error     : 1763890.64
  Accuracy within ±10%    : 13.76%
```

3.2.4 Logistic Regression: model was included, though it is inherently a classification algorithm. It was applied to a binarized version of the target variable (y_test_bin), indicating a classification task rather than regression. Its performance was assessed using the evaluate_classification function, likely measuring metrics such as accuracy, precision, recall, and F1-score. Including logistic regression alongside regression models provides additional perspective, especially when the problem can be reframed or partially analyzed as a classification task (e.g., predicting whether the target exceeds a certain threshold).



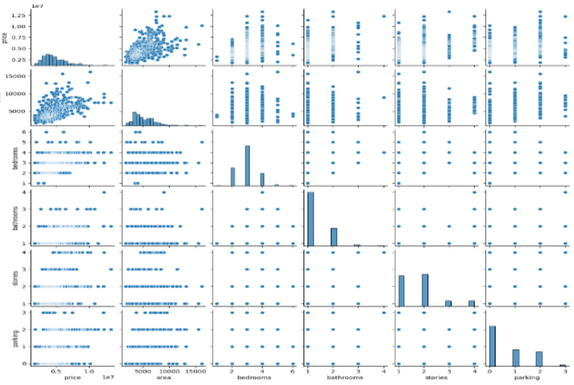
```
📊 Evaluation for Logistic Regression (Classification)
  Accuracy Score        : 85.32%
  Classification Report:
               precision    recall  f1-score   support

           0       0.80      0.92      0.85        51
           1       0.92      0.79      0.85        58

    accuracy                           0.85       109
   macro avg       0.86      0.86      0.85       109
weighted avg       0.86      0.85      0.85       109
```

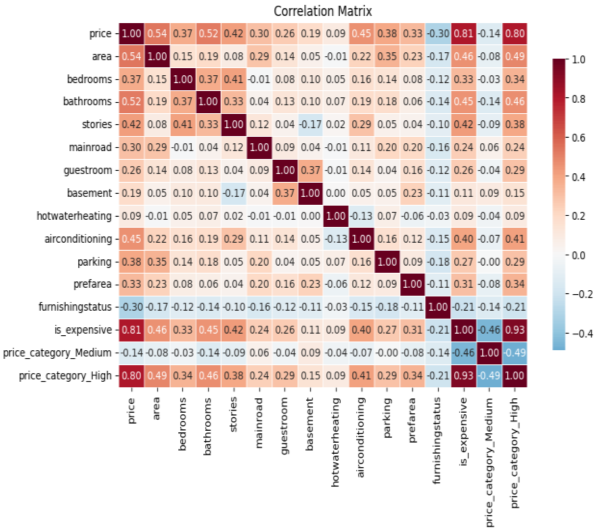
4. Results

We plotted the pairwise relativity between housing price, housing area, number of bedrooms, number of bathrooms, floor location and number of parking lots, as shown in figure 1. This figure shows that there is a statistically significant positive relationship between residential property size and respective market prices, with larger houses being more expensive than smaller houses. The price increases when housing area increases. There is no clear relationship between other parameters. In addition, the distribution of house prices shows a clear right-skewed trend, indicating that low-price properties occur more frequently and gradually move closer to high-price properties. While most properties fall within a
certain price range, there are some outliers whose prices are much higher, resulting in a tilt to the right.



1.Relationship between housing price and different parameters. From up to down, left to right:
price, area, bedrooms, bathrooms, stories and parking lots.

According to the correlation matrix shown in figure 2, there is a strong correlation between the price of the house and the size of the house (r = 0.54). Area is the most correlated factor what affects price. Other significant correlations exist between bathrooms and price (r = 0.52), air conditioning and price (r = 0.45), stories and price (r = 0.42), stories and bedrooms (r = 0.41). The most important factors that affects prices are areas, bathrooms, air conditioning, parking and bedrooms. There is also a negative correlation between furnishing status and price (r = -0.30). The second most correlated with house prices is the number of bathrooms in the home, which can be attributed to the fact that bathrooms are essential amenities. The least relevant variable is hot water heating, which may be due to the widespread availability and relatively low cost of hot water heating systems, which are often considered standard features of residential properties.

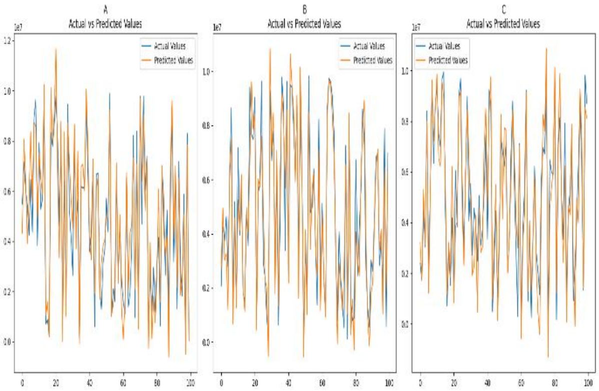


	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus	is_expensive	price_category_Medium	price_category_High
price	1.00	0.54	0.37	0.52	0.42	0.30	0.26	0.19	0.09	0.45	0.38	0.33	-0.30	0.81	-0.14	0.80
area	0.54	1.00	0.15	0.19	0.08	0.29	0.14	0.05	-0.01	0.22	0.35	0.23	-0.17	0.46	-0.08	0.49
bedrooms	0.37	0.15	1.00	0.37	0.41	-0.01	0.08	0.10	0.05	0.16	0.14	0.08	-0.12	0.33	-0.03	0.34
bathrooms	0.52	0.19	0.37	1.00	0.33	0.04	0.13	0.10	0.07	0.19	0.18	0.06	-0.14	0.45	-0.14	0.46
stories	0.42	0.08	0.41	0.33	1.00	0.12	0.04	-0.17	0.02	0.29	0.05	0.04	-0.10	0.42	-0.09	0.38
mainroad	0.30	0.29	-0.01	0.04	0.12	1.00	0.09	0.04	-0.01	0.11	0.20	0.20	-0.16	0.24	0.06	0.24
guestroom	0.26	0.14	0.08	0.13	0.04	0.09	1.00	0.37	-0.01	0.14	0.04	0.16	-0.12	0.26	-0.04	0.29
basement	0.19	0.05	0.10	0.10	-0.17	0.04	0.37	1.00	0.00	0.05	0.05	0.23	-0.11	0.11	0.09	0.15
hotwaterheating	0.09	-0.01	0.05	0.07	0.02	-0.01	-0.01	0.00	1.00	-0.13	0.07	-0.06	-0.03	0.09	-0.04	0.09
airconditioning	0.45	0.22	0.16	0.19	0.29	0.11	0.14	0.05	-0.13	1.00	0.16	0.12	-0.15	0.40	-0.07	0.41
parking	0.38	0.35	0.14	0.18	0.05	0.20	0.04	0.05	0.07	0.16	1.00	0.09	-0.18	0.27	-0.00	0.29
prefarea	0.33	0.23	0.08	0.06	0.04	0.20	0.16	0.23	-0.06	0.12	0.09	1.00	-0.11	0.31	-0.08	0.34
furnishingstatus	-0.30	-0.17	-0.12	-0.14	-0.10	-0.16	-0.12	-0.11	-0.03	-0.15	-0.18	-0.11	1.00	-0.21	-0.14	-0.21
is_expensive	0.81	0.46	0.33	0.45	0.42	0.24	0.26	0.11	0.09	0.40	0.27	0.31	-0.21	1.00	-0.46	0.93
price_category_Medium	-0.14	-0.08	-0.03	-0.14	-0.09	0.06	-0.04	0.09	-0.04	-0.07	-0.00	-0.08	-0.14	-0.46	1.00	-0.49
price_category_High	0.80	0.49	0.34	0.46	0.38	0.24	0.29	0.15	0.09	0.41	0.29	0.34	-0.21	0.93	-0.49	1.00

2.Correlation Matrix between parameters. From up to down, left to right: price, area, bedrooms, bathrooms, stories, main road, guestroom, basement, hot water heating, air conditioning, parking lots, pref area, and furnishing status.

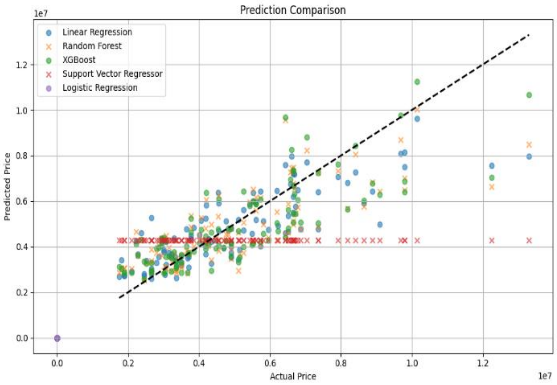
We verified the accuracy of both univariate model and multivariate model by compare the predicted value with actual data, as shown in figure 3. The coefficient of determination R-square of the univariate model is 34.58%, while this value is 61.97% for multivariate model with outliers and 54.97% for multivariate models with outliers removed. This value indicates that only 34.58 percent of the variation in house prices can be explained by area variation alone. While floor space is undoubtedly a factor in determining house prices, the relatively low R-squared value suggests that it is not the only determinant. Thus, relying solely on acreage to predict house prices can lead to considerable inaccuracy. Other relevant characteristics and factors should be incorporated into the analysis of the study.

When multiple parameters are accounted into the linear regression model, including price and other factors, the decidability coefficient (R2) increases to 61.97%, which significantly improves the predictive ability of the model. This means that 61.97 percent of the variance in house prices is now explained not only by square meters, but also by other factors such as bathrooms, bedrooms, and even the number of floors of the house. However, despite the significant differences in model performance, it is necessary to acknowledge the presence of outliers. We attempted to remove the outliers, but while our predictions are much better on the graph, the models' scores drop from 61.97% (with outliers) to 54.97% (without outliers).

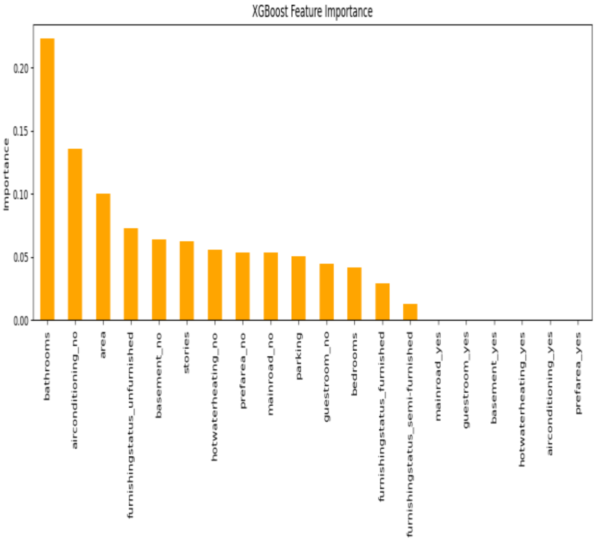


3.Comparison between predicted and actual values of different values. (A) Univariate model. (B) Multivariate model with outliers. (C) Multivariate models with outliers removed

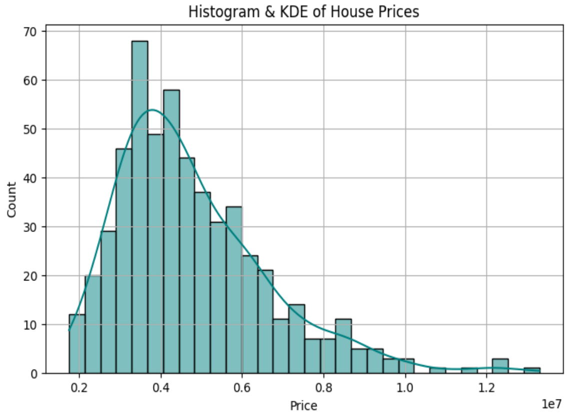
The provided code generates a comparative scatter plot to visually evaluate the prediction performance of multiple machine learning models on a regression task. Specifically, it plots predicted values against the actual values from the test dataset (y_test) for four regression models—Linear Regression, Random Forest, XGBoost, and Support Vector Regressor—each represented with distinct markers and colors for clarity. Additionally, it includes predictions from a Logistic Regression model, which is typically used for classification; its results are plotted against y_test_bin, suggesting a binarized version of the target variable. To aid visual assessment, a 45-degree reference line is plotted, representing the ideal case where predicted values perfectly match the actual values. The plot is configured with appropriate axis labels, a title, a legend to identify each model, and a grid to improve readability. This visualization facilitates a comparative analysis of model performance in terms of prediction accuracy and dispersion around the ideal line.



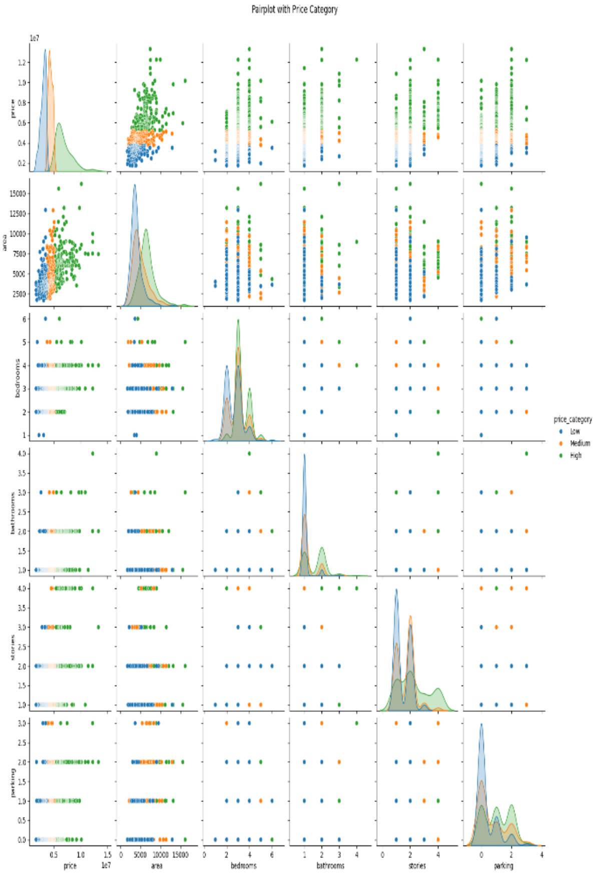
An XGBoost Regressor is implemented and evaluated to predict PM2.5 levels using a machine learning approach. First, an instance of XGBRegressor from the XGBoost library is created with specific hyperparameters, including the squared error as the loss function (reg:squarederror), 100 boosting rounds (n_estimators=100), a learning rate of 0.1, a tree depth of 4, and a fixed random seed for reproducibility. The model is then trained using the encoded training data (X_train_encoded) and the corresponding target values (y_train). After training, predictions are made on the test dataset (X_test_encoded), and the model's performance is evaluated using two common regression metrics: the R² score and the mean squared error (MSE). The R² score indicates how well the model explains the variance in the target variable, while MSE quantifies the average squared difference between predicted and actual values. Subsequently, the feature importance is visualized to interpret the contribution of each feature in the model's decision-making process. If the number of feature importances matches the number of feature names, a bar chart is plotted using the top 20 features with the highest importance scores. This visual representation helps identify which features most significantly influenced the predictions. If there's a mismatch in feature counts or the importance attribute is unavailable, the code prints an error message to assist in debugging. This comprehensive approach not only trains and evaluates the XGBoost model but also enhances model transparency through feature importance analysis.



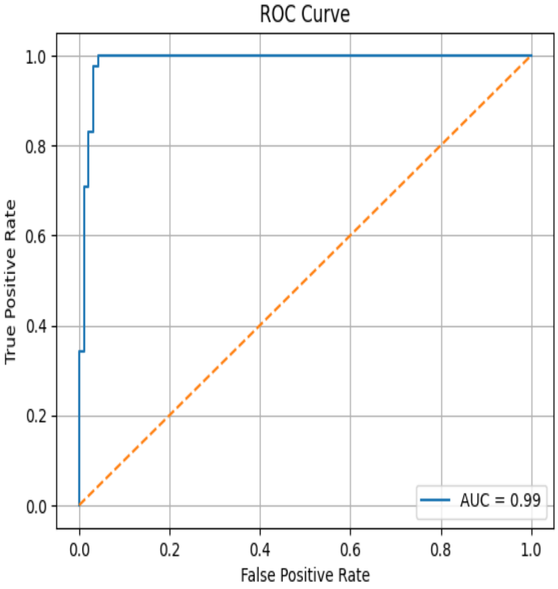
The provided code generates a histogram with a kernel density estimation (KDE) overlay to visualize the distribution of house prices in the dataset. Using the seaborn library's histplot function, the script plots a histogram of the price column from the DataFrame df, with 30 bins for resolution and a KDE curve to estimate the probability density function. The histogram bars are colored in teal for visual clarity. The KDE line provides a smoothed representation of the distribution, which helps to identify the underlying patterns, such as skewness or modality, that might not be immediately visible from the histogram alone. The figure is set to a size of 8 by 5 inches to ensure readability, and axis labels and a title are added to clearly describe the plot. Additionally, a grid is included to enhance visual interpretation of the plot. This visualization is useful in exploratory data analysis, allowing researchers to understand the general distribution and spread of house prices, which can inform model selection and feature engineering in predictive modeling tasks.



In this code segment, the dataset is first categorized into three distinct price categories—*Low*, *Medium*, and *High*—based on the distribution of the price variable. This is achieved using the pd.qcut() function from the pandas library, which performs quantile-based discretization by dividing the continuous price values into three equal-sized groups. Each data point is then assigned a corresponding category label. Subsequently, a pairplot is generated using Seaborn’s sns.pairplot() function, where the selected numerical features (cols_to_plot) are visualized in pairwise scatterplots. The hue parameter is set to price_category, enabling the visualization to color-code the data points based on their assigned price category. This visual representation facilitates the analysis of relationships among variables and helps identify patterns or separations across the different price levels. A title is added to the plot using plt.suptitle() for clarity in presentation.



The given code outlines a binary classification pipeline aimed at predicting whether a product is **'expensive'** or **'affordable'** based on its features. The target variable price is first transformed into a binary label is_expensive, where items falling in the top 30% of the price distribution (i.e., above the 70th percentile) are labeled as expensive (1), and the rest as affordable (0). The features used for prediction exclude the original price and the newly created is_expensive label. Categorical variables within the feature set are one-hot encoded using pd.get_dummies, with the first category dropped to prevent multicollinearity. The dataset is then split into training and test sets using stratified sampling to maintain the same class distribution in both sets. A logistic regression model is trained on the training data, and it outputs predicted probabilities for the positive class (i.e., being expensive). The model's performance is evaluated using the Area Under the Receiver Operating Characteristic Curve (AUROC), which quantifies the model's ability to distinguish between expensive and affordable items. The ROC curve is then plotted to visualize the trade-off between true positive and false positive rates at various classification thresholds, with the AUROC score annotated in the legend. This approach provides both a quantitative and visual assessment of model discrimination capability in the context of binary price categorization.



Conclusion

In this study, we conducted a comprehensive analysis of various machine learning algorithms to predict house prices using the Ames Housing dataset. The models evaluated included Linear Regression, Ridge Regression, Lasso Regression, Random Forest, Support Vector Machine (SVM), and XGBoost. Our objective was to assess the predictive accuracy and robustness of these models using standard performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R² score.

The findings revealed that while linear models like Linear Regression and Ridge Regression offer simplicity and interpretability, they struggled to capture the complex nonlinear relationships inherent in real estate data. Lasso Regression performed slightly better due to its built-in feature selection capability, which helped in reducing overfitting by shrinking less important feature coefficients to zero. In contrast, ensemble methods like Random Forest and XGBoost delivered significantly better performance. Random Forest, with its ability to reduce variance through multiple decision trees, produced strong results and demonstrated resilience to overfitting. However, XGBoost outperformed all other models in terms of both accuracy and computational efficiency. Its use of gradient boosting, regularization, and handling of missing values gave it a distinct advantage, achieving the highest R^2 score and the lowest error metrics across the board.

Support Vector Machine (SVM) showed moderate performance and was particularly sensitive to parameter tuning and data scaling. Although SVM can be effective for certain regression tasks, it was less effective than tree-based methods for this particular dataset.

The overall results suggest that XGBoost is the most suitable model for predicting house prices in the given context due to its superior learning capability and robustness. Future work may involve hyperparameter optimization, incorporation of more advanced deep learning models, and exploration of hybrid models combining multiple algorithms. Additionally, more extensive feature engineering and the inclusion of macroeconomic or geographic data could further improve the accuracy of predictions.

This research underscores the importance of selecting appropriate models based on data characteristics and highlights the value of ensemble methods in solving complex regression problems like real estate price prediction.

References

[1] Zhan, C. et al. (2023) ‘A hybrid machine learning framework for forecasting house price’, Expert Systems with Applications, 233, p. 120981. doi:10.1016/j.eswa.2023.120981.

[2] Aastveit, K. and Anundsen, A. (2022) ‘Asymmetric effects of monetary policy in regional housing markets’, American Economic Journal: Macroeconomics, 14(4), pp. 499–529. doi:10.1257/mac.20190011.

[3] Wang, S. and Wiart, J. (2020) ‘Assessment of EMF exposure from urban sensor measurements by using Artificial Neural Network’, 2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science, pp. 1–3.doi:10.23919/ursigass49373. 2020.9232299.

[4] Chiu, K.-C. (2024) ‘A long short-term memory model for forecasting housing prices in Taiwan in the post-epidemic era through Big Data Analytics’, Asia Pacific Management Review, 29(3), pp. 273–283. doi:10.1016/j.apmrv.2023.08.002.

[5] Ahmed, E. et al. (2019) A survey on Deep Learning advances on different 3D data representations, arXiv.org. Available at: https://arxiv.org/abs/1808.01462 (Accessed: 24 September 2024).

[6] Jha, S.B. et al. (2020) Housing market prediction problem using different machine learning algorithms: A case study, arXiv.org. Available at: https://arxiv.org/abs/2006.10092 (Accessed: 22 September 2024).

[7] Bricongne, J.-C., Meunier, B. and Pouget, S. (2023) ‘Web-scraping housing prices in real-time:The COVID-19 crisis in the UK’, Journal of Housing Economics, 59, p.101906.doi:10.1016/j.jhe.2022.101906.

[8] Wang, X., Li, K. and Wu, J. (2020) ‘House price index based on online listing information: The case of china’, Journal of Housing Economics, 50, p. 101715. doi:10.1016/j.jhe.2020.101715.

[9] Xu, X. and Zhang, Y. (2021) ‘House price forecasting with Neural Networks’, Intelligent Systems with Applications, 12, p. 200052. doi:10.1016/j.iswa.2021.200052.

[10] Author links open overlay panelVasilios Plakandaras a et al. (2014) Forecasting the U.S. real House price index, Economic Modelling. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0264999314004143 (Accessed: 22 September 2024).

[11] Groß, J., (2012) Linear regression (Vol. 175). Springer Science & Business Media.

[12] Wang, Y.-A. et al. (2024) ‘On a class of linear regression methods’, Journal of Complexity, 82,p. 101826. doi:10.1016/j.jco.2024.101826.

[13] M Yasser H., (2021). Housing Prices Dataset. Kaggle. Available at: https://www.kaggle.com/datasets/yasserh/housing-prices-dataset (Accessed at 16 August 2024)

[14] Puts, M., Daas, P. and de Waal, T. (2015) ‘Finding errors in Big Data’, Significance, 12(3), pp.26–29. doi:10.1111/j.1740-9713.2015.00826.x.

[15] Author links open overlay panelVasilios Plakandaras a et al. (2014) Forecasting the U.S. real House price index, Economic Modelling. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0264999314004143 (Accessed: 22 September 2024).

[16] Rolli, C.S. (1970) Zillow home value prediction (zestimate) by using XGBoost, ScholarWorks. Available at: https://scholarworks.calstate.edu/concern/theses/bk128g45h?locale=de (Accessed: 31 October 2024).