International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 11, ISSN No. 2455-2143, Pages 247-254
Published Online March 2021 in IJEAST (http://www.ijeast.com)

ANALYSIS OF DIFFERENT REGRESSION MODELS FOR REAL ESTATE PRICE PREDICTION

Pranav Kangane, Aadesh Mallya, Aayush Gawane, Vivek Joshi, Shivam Gulve
Department of Computer Science and Engineering,
NMIMS Mukesh Patel School of Technology Management & Engineering, Mumbai, India

Abstract— **The housing market is a standout amongst the most engaged with respect to estimating the price and continues to vary. Individuals are cautious when they are endeavoring to purchase another house with their financial plan and market strategies. Consequently, making the housing market one of the incredible fields to apply the ideas of machine learning on how to enhance and anticipate the house prices with precision. The objective of the paper is the prediction of the market value of a real estate property and present a performance comparison between various regression models applied. Nine algorithms were selected to predict the dependent variable in our dataset and then their performance was compared using R2 score, mean absolute error, mean squared error and root mean squared error. Moreover, this study attempts to analyze the correlation between variables to determine the most important factors that are bound to affect the prices of house.**

Keywords— **Boosting Regression, Machine Learning, Metrics, Regression, Supervised Learning**

I. INTRODUCTION

Land Property isn't just the essential need of a man yet today it likewise speaks to the wealth and esteem 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 decrease quickly. Changes in the land cost can influence different family speculators, investors, policymakers, and many. Interest in the land area is by all accounts an alluring decision for ventures. In this way, foreseeing the land esteem is a significant financial file. For the most part, the property estimations ascend concerning time and its assessed esteem should be determined. This evaluated esteem is needed during the offer of property or while applying for the credit and for the attractiveness of the property. These evaluated qualities are dictated by proficient appraisers. In any case, the downside of this training is that these appraisers could be one-sided due to presented interests from purchasers, dealers, or home loans. In this way, we require a robotized forecast model that can assist with foreseeing the property estimations with no predisposition. This mechanized model can help first-time purchasers and less experienced clients to comprehend whether the property rates are misrepresented or misjudged.

In this research paper, we have made an effort to apply nine of those modern algorithms and compare their efficacy in predicting the price of houses in the city of Ames, Iowa. We have made an effort to determine which algorithm performs the best, when performing the peculiar process of predicting the price of a house based on the parameters given.

The remaining article is structured as follows. We describe the literature survey in Section II. Methodology and Machine Learning algorithms applied in Section III. Results and Observation in Section IV. Conclusion in Section V and Future Work in Section VI. Finally, Section VII concludes the References.

II. LITERATURE SURVEY

Building Over the most recent decades estimating assets estimation has become a large field. The ascent in the interest for property and peculiar behavior of the economy has urged analysts to design a model that can forecast the house prices based on various significant factors. Building a prescient model for house prices calls for exhaustive information of all the factors that could affect the cost of the house. To design this model numerous experts have taken a shot at this problem and conveyed their work.

Manjula [1] used various significant attributes for predicting house prices using a regression model for achieving good accuracy. Shinde et al., [2] used various machine learning algorithms like lasso, Logistic regression, decision trees for predicting house prices and compared the accuracy. Alfiyatin [3] has designed a model for house price prediction using Regression and Particle Swarm Optimisation (PSO). In which he proved that accuracy can be improved using PSO with regression. A. Varma [4] designed a model that was able to extract real-time neighbourhood information to achieve precise world evaluation using Google maps. Fan [5] Used decision tree approach for predicting the resale house prices based on important features. For this hedonic regression method was applied that could identify the relationship between the features and house price. Hujia Yu [6] utilized classification and regression algorithms for predicting house prices. It was observed that living area square feet, roof content, and neighbourhood have the highest statistical importance in predicting house prices and could be further improved using PCA technique. Kuvalekar [7] used a decision tree for predicting the house prices in Mumbai city and made use of geographical variables to find the starting prices.

III. METHODOLOGY

The Methodology represents a description about the framework that is undertaken. It comprises various milestones that should be accomplished in order to satisfy the objective. We have undertaken different data mining and machine learning concepts. The accompanying figure, Fig.1 represents step-wise undertakings that should be finished.

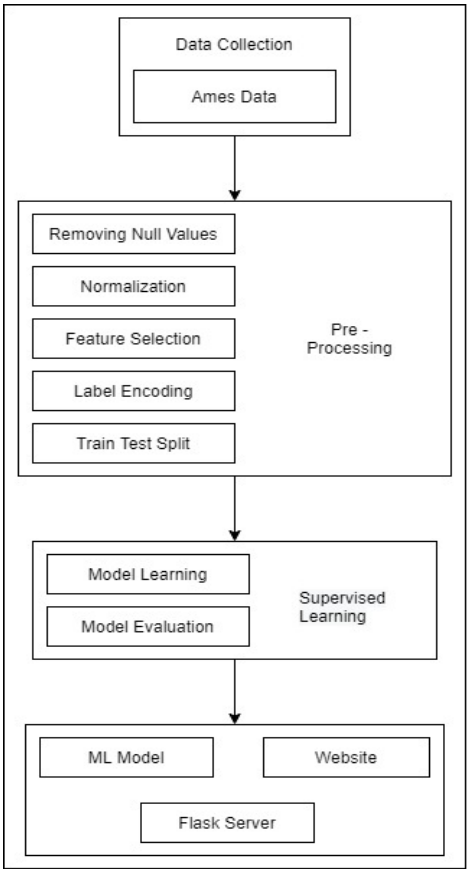


Figure 1: Project Methodology

3.1 Data Collection

The dataset used in this project is an open-source dataset from Kaggle. We have used Ames Housing Dataset. It comprises data based on all residential home sales in Ames, Iowa between 2006 and 2010. The dataset consists of 1460 records with 80 explanatory variables on the quality and quantity of physical attributes of residential homes in Iowa that were sold between 2006 and 2010. It comprises all the data of various factors that a buyer might want to think about the property. Parameters such as Area in square meters, Overall quality which rates the overall condition and finishing of the house, Location, Year in which house was built, Numbers of Bedrooms and bathrooms, Garage area and number of cars that can fit in garage, swimming pool area, selling year of the house and other factors which will undoubtedly influence the housing price. The outcome is the selling price which depends on all these independent variables.

3.2 Data Visualization

To understand the data, visualizations are important because they provide a visual summary of information and make it easier to identify patterns and trends. Charts and graphs make communicating data findings easier and help us to gain valuable insights.

The histogram Fig. 2 shows the number of houses built in every decade. According to the chart, there is a steady increase in the number of houses as we move from 1870 to 2010. However, the number reached its peak in the decade of 2000-2010.

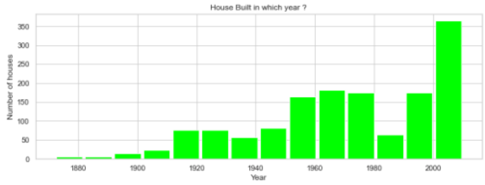


Figure 2: Houses Built vs Decade Histogram

The bar chart Fig. 3 shows the number of houses that were sold in every month of every year. A pattern could be observed in the chart that most houses for every year were sold in June. The graph also depicts that there is a steady increase in the number of houses as the month passes by. However, after reaching the peak value in June, the number of houses sold tend to decrease for the rest of the months.

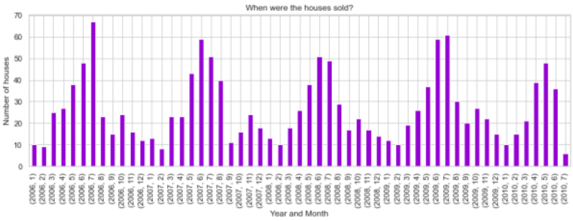


Figure 3: Houses Sold vs Year and Month

The horizontal bar chart Fig. 4 depicts the number of houses in every neighbourhood. It is seen that NAmes has greatest number of houses followed by CollgCr and Old Town. While Blueste comprises the least number of houses.

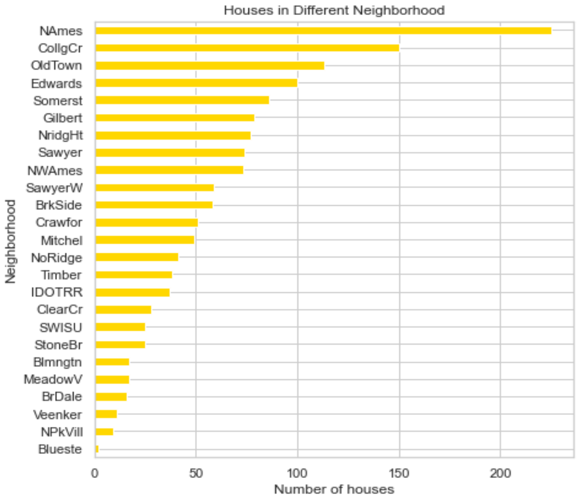


Figure 4: Houses in Different Neighbourhood

A correlation matrix is important because it helps us in discovering the relationship that exists amongst various parameters. One such correlation matrix is plotted in Fig. 6 for 10 selected parameters having correlation more than 50 percent. The highest correlation of 88 percent exists between garage area and garage cars. It is followed by garage year built and year built with a correlation of 83 percent.

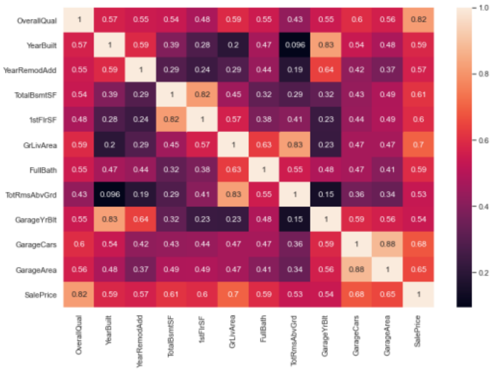


Figure 6: Attributes having correlation more than 50%

A bar plot as shown in Fig. 7 shows the correlation between all the attributes concerning the target variable which is the Sale Price. It is observed that Overall Quality has the most influences on the value of Sale Price followed by Ground Living Area and Garage Cars. It can also be seen that some of the variables have negative correlation which means that with the increase in attribute value, the Sale Price decreases. Enclosed Porch followed by Kitchen Above Ground have the highest negative correlation.

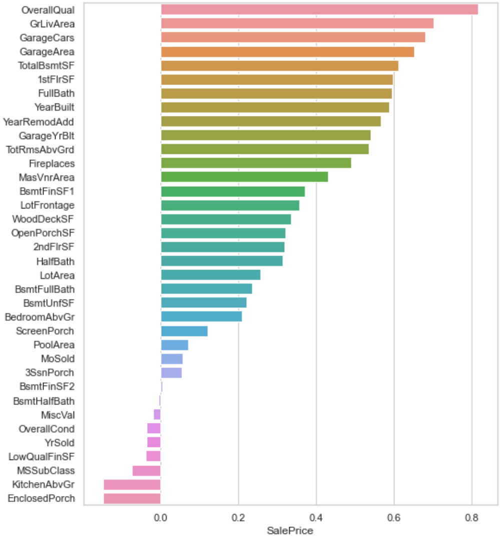


Figure 7: Correlation between Attributes and Sale Price

Since Overall Quality has the highest influence on Sale Price, a box plot is constructed to understand the distribution of Sale Price with respect to Overall Quality. As shown in the Fig. 8 it can observed that the Sale Price has a median value of almost $ 450000 for houses having Overall Quality 10. For houses with Overall Quality 1, the median price is almost $ 50000.

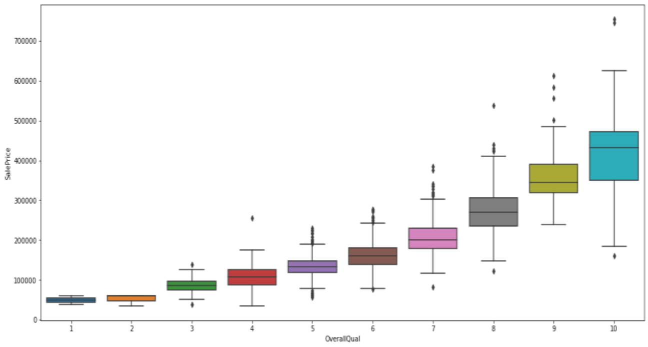


Figure 8: Overall Quality Boxplot

3.3 Data Pre-processing

Handling missing values is crucial because most of the machine learning algorithms do not support data comprising null values. These missing values need to be handled properly or else might lead to inaccurate results. A bar plot is constructed in Fig. 8 that shows the count of missing values in every attribute. It is found that 19 attributes consist of missing values in the dataset.

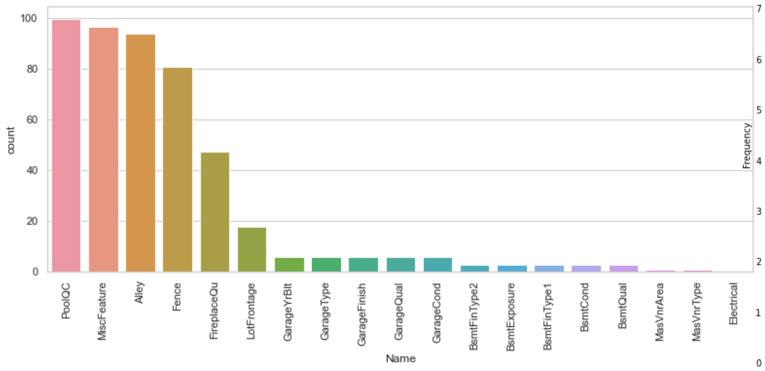
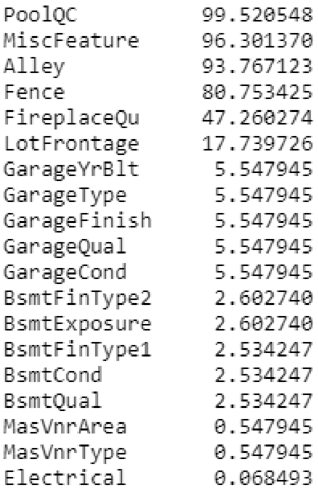


Figure 9: Attributes with missing values

Figure 9 shows the exact percentage of missing values in the dataset. It is found that 99.5 % of missing values are found for Pool Quality attribute followed by 96. 3 % for Misc Feature while the least value is 0.06 % for Electrical attribute. Since most of the features having missing values were having a low correlation value with respect to Sale Price, it was decided to drop the attributes to keep the data accurate.



```
PoolQC          99.520548
MiscFeature     96.301370
Alley           93.767123
Fence           80.753425
FireplaceQu     47.260274
LotFrontage     17.739726
GarageYrBlt      5.547945
GarageType       5.547945
GarageFinish     5.547945
GarageQual       5.547945
GarageCond       5.547945
BsmtFinType2     2.602740
BsmtExposure     2.602740
BsmtFinType1     2.534247
BsmtCond         2.534247
BsmtQual         2.534247
MasVnrArea       0.547945
MasVnrType       0.547945
Electrical       0.068493
```

Figure 10: Percentage of Missing Values

After removing the null values, we need to check the distribution of the target variable i.e. Sale Price. In Fig.10 it can be seen that the distribution of Sale Price is not normal. The target variable is right-skewed which means that the majority of the data has low-priced houses and a limited number of records for high-priced houses. Hence, to change the target variable from right sewed to a normal distribution, a logarithmic function was applied as shown in Fig. 11.

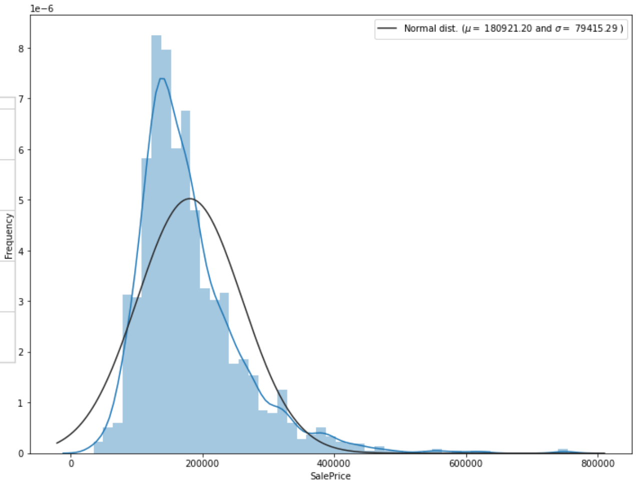


Figure 10: Right Skewed Distribution

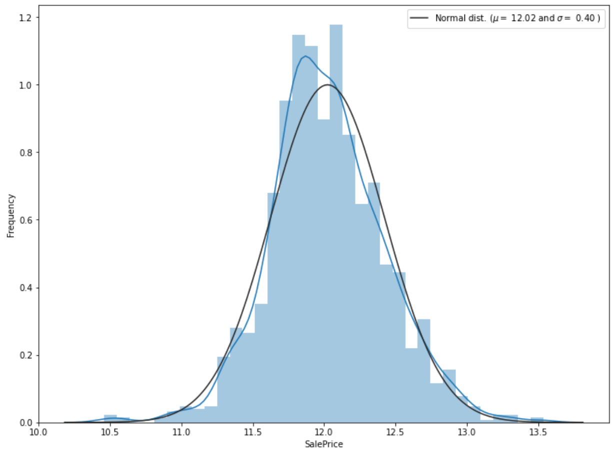


Figure 11: Normal Distribution

By removing the features consisting of null values, our dataset was left with 61 attributes for predicting the Sale Price. Since our dataset contains multiple labels in one or more columns, it is important to convert the labels into the numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. Hence, all the categorical variables were converted into numeric form using the Label Encoder from Sklearn library. Before applying the machine learning models, the dataset is split into train and test data in the ratio of 70:30 with a random state of 42.

3.4 Algorithms Applied

We have selected nine algorithms to predict the dependent variable in our dataset. The algorithms that we have selected are basically Multiple Linear Regression, Linear SVM, Decision Tree, Ridge, Lasso, Gradient Boosting, XGBoosting, CATBoost and Random Forest Regression. These algorithms were implemented with the help of python's SciKit-learn Library.

3.4.1 Linear Support Vector Machine

The Linear function is used when the dataset consists of data that is linearly separable, that is, it can be separated using a single line. It is one of the most common kernels. It is mostly used when there are a large number of features but a relatively low number of records in a particular dataset. One such example is that of text classification, as every single alphabet is a new input feature. Hence, linear kernel-based support vector machines are most commonly used for text classification. Linear kernel-based SVM are easily modelled and easily trained. Also, training of an SWM with a linear function, only requires the optimization of the C regularization parameter. On other hand, training of SVM of other kernels – polynomial kernels, RBF, etc. requires the optimization of the y parameter that requires a grid search to be performed, which only ends up increasing the execution time.

3.4.2 Multiple Linear Regression

Multiple linear regression (MLR) is a statistical technique which uses several explanatory variables to predict the outcome of a response variable. Among a number of random variables, it is used to determine a mathematical relationship. This means that the MLR examines how multiple independent variables are related to one dependent variable. The advantages of MLR are that it works well irrespective of the size of the dataset and it is easier to implement and interpret and also very efficient to train. MLR faces difficulties where it has to take the assumption between the dependent variable and the independent variables. Also, if the number of observations is lesser than the number of features, MLR should not be used as it may lead to overfitting.

3.4.3 Ridge Regression

The regularized form of linear regression is Ridge regression. The term multicollinearity alludes to the collinearity concept in statistics. In multicollinearity, one predicted value in multiple regression models is linearly predicted with others to achieve a certain level of accuracy. It occurs when there are high correlations between more than two predicted variables. The type of model tuning strategy that's used to analyse any data that suffers from multicollinearity is Rigid Regression.

3.4.4 Lasso Regression

The LASSO stands for Least Absolute Shrinkage and Selection Operator. Lasso regression is a regularization technique which is used over regression methods for a more accurate prediction. The Lasso model uses shrinkage where shrinkage alludes to data values which are contracted towards a central point as the mean. The lasso procedure also encourages simple, sparse models i.e., models with fewer parameters. Lasso regression is ordinarily used for models showing high levels of multicollinearity or when we need to automate certain parts of model selection, like variable selection/parameter elimination.

3.4.5 Decision Tree Regression

Decision tree regression comprises a tree like structure and consists of 3 types of nodes. The Root Node is the initial node used to represent the entire dataset and splits into further nodes. The internal nodes denote the attributes and decision rules are represented by branches. The outcome is represented by the leaf nodes. The model observes the features and trains the model in the structure of the tree to forecast data in future to produce meaningful continuous data. This approach is easy to understand, requires less data cleaning. However, it is prone to overfitting and the problem can be solved using the Random Forest algorithm.

3.4.6 Random Forest

Random Forest regression uses an ensemble learning approach for regression. It is a method in which it combines the predictions from multiple machines learning models to make an accurate prediction. The algorithms consist of multiple trees that run in parallel and do not interact with each other. During training multiple decision trees are constructed and the output is taken as the mean of the classes as the prediction of all trees. This algorithm efficiently runs on large databases and has an effective method for estimating missing data. However, it requires high computational power as well as resources as it builds numerous trees to combine their outputs.

3.4.7 Gradient Boosting

Gradient Boosting is a robust technique which is used for solving regression and classification problems. It is based on sequential ensemble learning where the model is created in stage-wise fashion. It infers the model by enabling the optimization of an absolute differentiable loss function. As we tend to add every weak learner, a new model is formed that provides a more precise estimation of the response variable. The algorithm requires three components i.e., loss function, weak learner and additive model. The loss function needs to be optimized for reducing the error in prediction. Decision trees are used as weak learners and are required for making predictions. To reduce the loss, decision trees can be added and do not change the existing tree. The model can be used to resolve multicollinearity problems where the correlations among the predictor variables are high.

3.4.8 XGBoost Regression

XGBoost algorithm is an extension of gradient boosting algorithm. In case of gradient boosting, the computations are performed in a sequential manner due to which we get output at a slower rate. In order to enhance the performance, XGBoost performs parallel computations on decision trees. The model uses cache optimization to manage and utilize resources. The model can handle missing values by finding out the trends in them and apprehends them.

3.4.9 Cat Boost Regression

CatBoost is an algorithm for gradient boosting on decision trees. In addition to regression and classification, Cat Boost is widely used for ranking tasks, forecasting and making recommendations. It is usually used with data which that high variability of data types and formats. Often datasets contain categorical features and CatBoost automatically handles categorical features by grouping in categories by target statistics. Ordered target statistics are used to calculate the target statistics in CatBoost. CatBoost has an effective usage with default parameters thus, reducing the time needed for parameter tuning.

IV. RESULT AND OBSERVATION

The performance and efficiency of a regression model can be measured on several factors. Unlike classification/clustering problems, regression efficiency of models cannot be evaluated using measures such as F1 score, Precision, Recall, and Accuracy scores, etc. This is because, unlike classification problems, it is impossible to estimate the exact future value of any data element. The measures we will be using to evaluate the various regression algorithms that we have tested on the given dataset are as follows -

1. R2 Score
The R2 Score or R-Squared Score is a measure of how accurately the line has been fit on the given dataset i.e., how close the data points are to the fitted regression/prediction line. Since we are using multiple linear regression here (since it is a multi-dimensional, multi-attribute dataset), the R2 score is called the coefficient of multiple determination here. The definition is fairly straightforward. It is the percentage of the total variance explained by the model. In simpler terms, it is a measure of how close data points lie from the predicted data value. Understandably, higher the value of the R2-Score for the regression model, higher is its accuracy. The R2-Score varies from 0-100%, with 0 standing absolutely 0 correlation between the data points and the corresponding fitted regression value, and 100 standing for a complete correlation between the two, which practically is impossible to achieve. In our experiment, we have evaluated the different regression models using the R2-Score as one of the prime parameters.

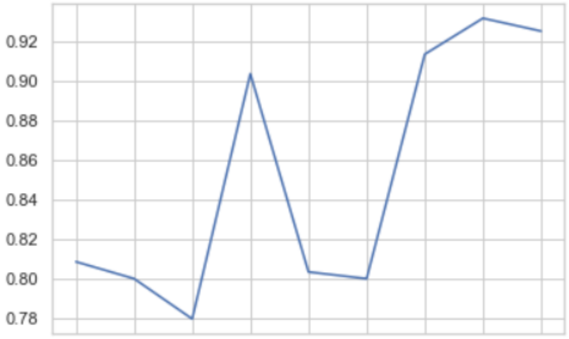


Figure 12: Comparison of R2-Scores

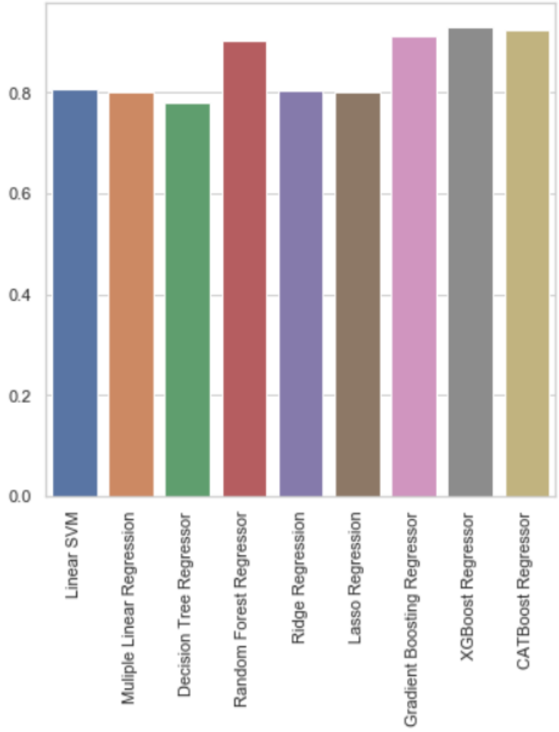


Figure 13: Comparison of R2-Scores

2. Mean Absolute Error (MAE)
Absolute error, as the name suggests, is simply the amount of prediction error the model makes. The prediction error is simply the difference between the actual value in the validation set and the value predicted by the model. The error value can be either positive or negative, suggesting that the predicted value can be

greater than or less than the actual value. However, the sign of the error doesn't matter, since an error is an error, be it positive or negative. Hence, we take an absolute of the error and use it for further calculation. When we take the mean of all recorded absolute error, the value obtained is called the mean absolute error (MAE). Hence, given any dataset, the MAE is the mean of all prediction errors on every single instance of the validation dataset. In our experiment, we have used Mean Absolute Error as one of the parameters/metrics to evaluate the efficiency of the different regression models. The mathematical expression for MAE is as follows –

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

3. Mean Squared Error (MSE)
The mean squared error, as the name suggests, is the average of all squared errors and is usually used as a loss function so as to help the model re-adjust it's weights and fit a line that has a lower prediction error. Consider a set of points and a regression line fit among them that best tries to predict the future values. Now consider a value of X and the corresponding value of Y for that value of X on the regression line (Yr) and the actual value of Y for that value of X in the validation set (Ya). Squared error would be the difference [(Ya - Yr) ^2]. A mean of all these errors is called the mean squared error. The mean squared error like MAE is always positive. The idea is to have as low a value of MSE as possible, as that would indicate a line that minimizes the squared error. In our experiment, we have used Mean Squared Error as one of the parameters/metrics to evaluate the efficiency of the different regression models. The mathematical expression for MSE is as follows –

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

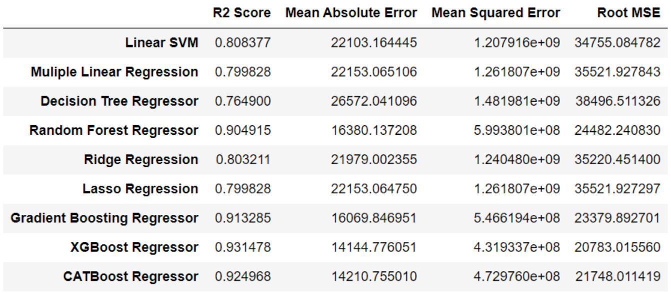
4. Root Mean Squared Error (RMSE)
The root mean squared error is simply the square root of the mean squared error value. Like the MSE, it is used to estimate the accuracy of the regression line fit over the data points. It is used to measure the difference between the predicted value on the regression line and the observed/actual data point in the validation set. Both MSE and RMSE are negatively oriented in the sense that lower values of these scores indicate better/higher accuracy of the model. Then why is there a need to make a separate metric taking the square root of MSE? This is because, taking the root of the averaged error values in the mean squared errors has some interesting implications for the root MSE. Since the errors are also squared before the average is taken, RMSE becomes useful when large values of errors are undesirable. Thus, RMSE becomes more appropriate than MAE or even MSE, when taking absolute values becomes problematic during mathematical calculations and when large error values are undesirable. In our experiment, we have used Root Mean Squared Error/Deviation as one of the parameters/metrics to evaluate the efficiency of the different regression models. The mathematical expression for RMSE is as follows –

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}{N}}$$

From the explanation given above, it is clear that the criteria for a model to be accurate and appropriate to be used for house price prediction, it should meet the following criteria -

1. Highest R-Squared Score
2. Lowest Mean Absolute Error value
3. Lowest Mean Squared Error value
4. Lowest Root MSE value

We will now see the evaluation of the different models on these parameters. Given below is a table containing different parameter scores run on 9 different regression models.



	R2 Score	Mean Absolute Error	Mean Squared Error	Root MSE
Linear SVM	0.808377	22103.164445	1.207916e+09	34755.084782
Muliple Linear Regression	0.799828	22153.065106	1.261807e+09	35521.927843
Decision Tree Regressor	0.764900	26572.041096	1.481981e+09	38496.511326
Random Forest Regressor	0.904915	16380.137208	5.993801e+08	24482.240830
Ridge Regression	0.803211	21979.002355	1.240480e+09	35220.451400
Lasso Regression	0.799828	22153.064750	1.261807e+09	35521.927297
Gradient Boosting Regressor	0.913285	16069.846951	5.466194e+08	23379.892701
XGBoost Regressor	0.931478	14144.776051	4.319337e+08	20783.015560
CATBoost Regressor	0.924968	14210.755010	4.729760e+08	21748.011419

Table 1: Performance of Algorithms

Evidently, the four best performing regression algorithms are -

1. Random Forest Regression
2. Gradient Boost Regression
3. XGBoost Regression
4. CATBoost Regression

With XGBoost Regression having the highest model score of 93.14%, it becomes the best performing regression model. From our experiment we can observe that the XGBoost regression outperformed all other regression models. It -

- Has the highest R2 Score - 93.14%
- Has the lowest MAE value - 14144.77
- Has the lowest MSE value - 4.32 x 10^8
- Has the lowest RMSE value- 20738.02

Since XGBoost is basically an extreme gradient boosting algorithm, we see that that gradient boost algorithm also has an R2 Score (91.3%) very close to that of XGBoost. The XGBoost owing to its second order gradients and advanced regularization performs slightly better than traditional Gradient Boosting algorithm.

We can also see that the decision tree regressor has the worst performance with the lowest R2 score and highest error values. This can be attributed to the large number of attributes in the dataset and the complexity of entropy gain calculation to generate the decision tree. While the decision tree regression by itself performs the worst, the random forest regressor, which is basically a combination of outputs of multiple decision trees, performs fairly well with a reasonable r2 score of 90.49%.

V. CONCLUSION

This article aims to provide an insight into the efficiency of different regression models that can be used to predict house prices. The models compared here include – linear svm, multiple linear regression, lasso regression, ridge regression, decision trees, random forest, gradient boost, XGBoost and catboost regressors. The motive behind evaluating so many regression models was to get the best regression model which will provide an accurate output and prevent a potential user from making wrong investments in real estates. Our experiments were carried out on a dataset consisting of nearly 1480 records with up to 80 different input attributes and we have reached the conclusion that the extreme gradient boost regressor (XGBoost) has the maximum efficiency and outperforms all other regression techniques, with a score of 93.4%. While the referred database is already large, we can make the model more accurate by adding the house price records of various other cities, as well as rural areas, so as to introduce variations in the model while simultaneously increasing its accuracy to predict house prices under given conditions. The paper can be extended to various other arenas of real estate by applying said regression model to house resale databases etc., which will benefit the people.

VI. REFERENCE

[1] Manjula, R. (2017). Real Estate Value Prediction Using Multivariate Regression Models. *Materials Science And Engineering Conference Series*, 4.

[2] Shinde, N., & Gawande, K. (2018). Valuation Of House Prices Using Predictive Techniques. *International Journal Of Advances In Electronics And Computer Science And Applications*, 34-40.

[3] Alfiyatin, A., Taufiq, H., Febrita, R., & Mahmudy, W. (2017). Modeling House Price Prediction Using Regression Analysis And Particle Swarm Optimization. *(Ijacsa) International Journal Of Advanced Computer Science And Applications*, 323-326.

[4] Varma, A. (1936). House Price Prediction Using Machine Learning And Neural Networks. *2018 Second International Conference On Inventive Communications And Computational Technologies*, 1936-1939.

[5] Zhi Fan, G., Ong, S., & Koh, H. (2006). Determinants Of House Price: A Decision Tree Approach. *Urban Studies*, 12.

[6] Yu, H., & Wu, J. (2016). Real Estate Price Prediction With Regression And Classification. *Cs 229 Autumn 2016 Project Final Report*, 1-5.

[7] Kuvalekar, A., Manchewar, S., Mahadik, S., & Jawale, S. (2020). House Price Forecasting Using Machine Learning. *3rd International Conference On Advances In Science And Technology*.