Proceedings of the Second International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020)
IEEE Xplore Part Number: CFP20K58-ART; ISBN: 978-1-7281-4167-1

Machine Learning based Predicting House Prices using Regression Techniques

Manasa J
Department Of Mathematics
Dayananda College Of Engineering-RC
Visveshwariah Technological University
Bengaluru, Country
mansa.chandan@gmail.com

Radha Gupta
Professor and Head,
Department Of Mathematics
Dayananda College Of Engineering
Bengaluru, Country
radha.gaurav.gupta@gmail.com

Narahari N S
Professor
Department Of IEM
RV College Of Engineering
Bengaluru, Country
naraharins.rvce.edu.in

***Abstract*— Predictive models for determining the sale price of houses in cities like Bengaluru is still remaining as more challenging and tricky task. The sale price of properties in cities like Bengaluru depends on a number of interdependent factors. Key factors that might affect the price include area of the property, location of the property and its amenities. In this research work, an analytical study has been carried out by considering the data set that remains open to the public by illustrating the available housing properties in machine hackathon platform. The data set has nine features. In this study, an attempt has been made to construct a predictive model for evaluating the price based on the factors that affect the price. Modeling explorations apply some regression techniques such as multiple linear regression (Least Squares), Lasso and Ridge regression models, support vector regression, and boosting algorithms such as Extreme Gradient Boost Regression (XG Boost). Such models are used to build a predictive model, and to pick the best performing model by performing a comparative analysis on the predictive errors obtained between these models. Here, the attempt is to construct a predictive model for evaluating the price based on factors that affects the price.**

Keywords—house price, lasso regression, ridge regression, regression methods

I. INTRODUCTION

Modeling uses machine learning algorithms, where machine learns from the data and uses them to predict a new data. The most frequently used model for predictive analysis is regression. As we know, the proposed model for accurately predicting future outcomes has applications in economics, business, banking sector, healthcare industry, e-commerce, entertainment, sports etc. One such method used to forecast house prices are based on multiple factors [7]. In metropolitan cities like Bengaluru, the prospective home buyer considers several factors such as location, size of the land, proximity to parks, schools, hospitals, power generation facilities, and most importantly the house price. Multiple linear regression is one of the statistical techniques for assessing the relationship between the (dependent) target variable and several independent variables. Regression techniques are widely used to build a model based on several factors to predict price. In this study, we have made an attempt to build house price prediction regression model for data set that remains accessible to the public in Machine hackathon platform. We have considered five prediction models, they are ordinary least squares model, Lasso and Ridge regression models, SVR model, and XGBoost regression model. A comparative study was carried out with evaluation metrics as well. Once we get a good fit, we can use the model to forecast monetary value of that particular housing property in Bengaluru. The paper is divided into the following sections: Section 2 addresses previous related work, Section 3 explains the description of the data set used, pre-processing of data and exploratory analysis of data before regression model is built. Section 4 presents a summary of the regression models developed in the comparison study and the evaluation metrics is used. Section 5 sums up the models and concludes with the future scope of the proposed work. Section 6 lists the applicability of the model.

II. PREVIOUS RELATED WORK

Pow, Nissan, Emil Janulewicz, and L. Liu [11] used four regression techniques namely Linear Regression, Support Vector Machine, K-Nearest Neighbors (KNN) and Random Forest Regression and an ensemble approach by combining KNN and Random Forest Technique for predicting the property's price value. The ensemble approach predicted the prices with least error of 0.0985 and applying PCA didn't improve the prediction error.

Several studies have also focused on the collection of features and extraction procedures. Wu, Jiao Yang [12] has compared

various feature selection and feature extraction algorithms combined with Support Vector Regression. Some researchers have developed neural network models to predict house prices. Limsombunchai, compared hedonic pricing structure with artificial neural network model to predict the house prices [13].The R-Squared value obtained for Neural Network model was greater when compared to hedonic model and the RMSE value of Neural Network model was relatively lower. Hence they concluded that Artificial Neural Network performs better when compared with Hedonic model.

Cebula applies the hedonic price model to predict housing prices in the City of Savannah, Georgia. The log price of houses has been shown to be positively and substantially associated with the number of bathrooms, bedrooms, fireplaces, garage spaces, stories and the house's total square feet [14].

Jirong, Mingcang and Liuguangyan apply support vector machine (SVM) regression to predict China's housing prices from 1993 to 2002. They have applied the genetic algorithm to tune the hyper-parameters in the SVM regression model. The error scores obtained for the SVM regression model was less than 4% [15].

Tay and Ho compared the pricing prediction between regression analysis and artificial neural network in predicting apartment's prices. It was concluded that that the neural network model performs better than regression analysis model with a mean absolute error of 3.9% [16].

III. Data Understanding and Pre-Processing

A. Data Description

The two data sets-train set and test data considered in the project is taken from Machine Hackathon platform. It consists of features that describe house-property in Bengaluru. There are 9 features in both the data sets. The features can be explained as follows:

1. Area type-describes the area
2. Availability-when it is possesses or when it is ready.
3. Price- Value of the property in lakhs.
4. Size- in BHK or Bedroom (1-10 or more)
5. Society- to which it belongs.
6. Total_sqft - size of the property in sqft.
7. Bath-No of bathrooms 8. Balcony- No of balcony
9. Location – where it is located in Bengaluru

With 9 features available, we try to build regression models to predict house price. We predicted the price of test data set with the regression models built on train data set [8].

B. Data understanding and basic EDA

The purpose is to create a model that can estimate housing prices. We divide the set of data into functions and target variable. In this section, we will try to understand overview of original data set, with its original features and then we will make an exploratory analysis of the data set and attempt to get useful observations. The train data set consists of 11200 records with 9 explanatory variables. In test data set, there were around 1480 records with 9 variables. While building regression models we are often required to convert the categorical i.e. text features to its numeric representation. The two most common ways to do this is to use label encoder or one hot encoder. Label encoding in python can be achieved by using sklearn library.

Label encoder encodes labels with a value between 0 and n-1. If a label repeats, it attributes the same value as previously assigned [6]. One hot encoding refers to splitting the column that contains numerical categorical data to many columns depending on the number of categories present in that column. Each column contains "0" or "1" corresponding to which column it has been placed [6].

This dataset includes quite a few categorical variables (both train and test data set) for which we will need to create dummy variables or use label encoding to convert into numerical form. These would be fake/dummy variables because they are placeholders for actual variable and are created by ourselves. Also, there are a lot of null values present as well, so we will need to treat them accordingly. The features bath, price, balcony are numerical variables. Features like area_type, total_sqft, location, society, availability, and size appears as categorical variables.

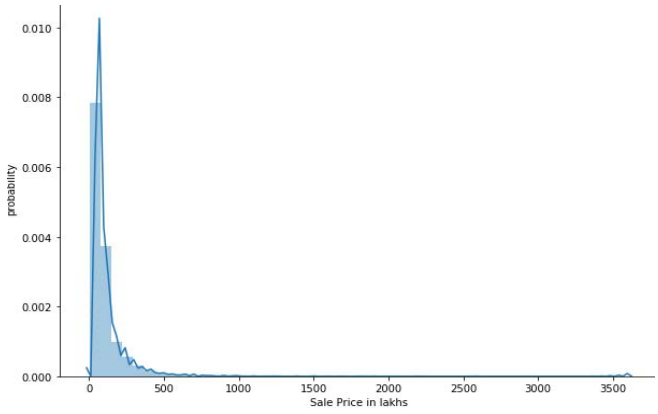

Fig.1. Distribution of price in train data set

It could be seen that price's distribution is highly skewed. The price ranges from 8 lakhs to 3600 lakhs. Most of the price lies below 500 lakhs. Kurtosis is a metric that shows whether the data set is heavy or light tailed compared to a normal distribution. It was observed that the skewness and kurtosis were around 8 and 108 respectively. Since the price has positively skewed distribution; we used log transformation of price for further analysis. After log transformation is applied to price variable, the distribution was as shown in Figure-2 [6].

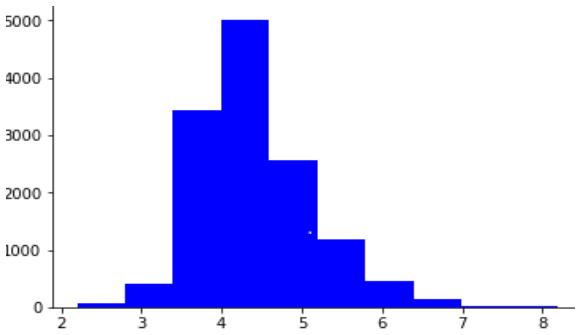

Fig.2. Distribution of transformed price

After applying log transformation to the price variable, we observe that kurtosis and skewness were reduced to 0.85628 and 1.34. We considered pairwise scatterplot that will allow us to visualize the pair-wise relationships and correlations between the different features as in Figure 3 The scatter plot helps us see how dispersed the data points are. It is helpful to get a quick overview of how the data is being distributed and whether it includes outliers or not. Furthermore, we can say from the histogram that the price variable (to be noted we applied log transformation for the original price variable) appears to be distributed normally, but contains several outliers.

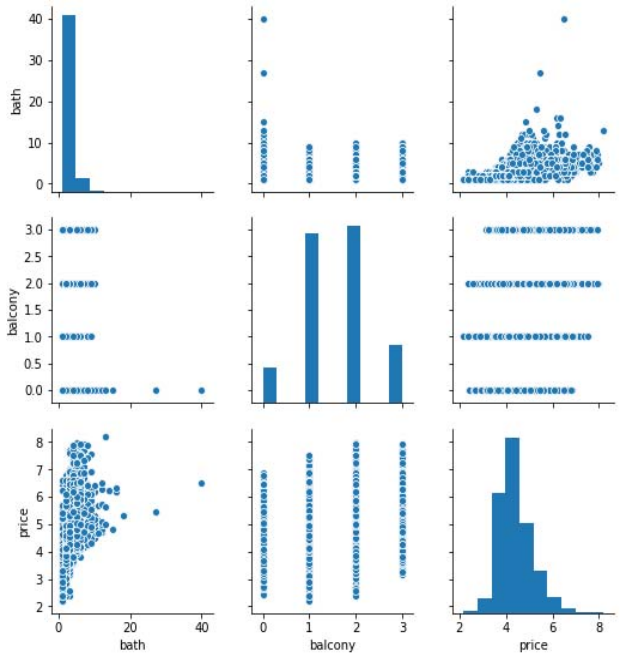

Fig.3. Pairwise scatterplot between quantitative variables

C. *Data pre-processing*

The general steps in data pre-processing are:

- Converting categorical features into numerical variables in order to fit linear regression model.
- Imputing null records with appropriate values.
- Scaling of data
- Split into train –test sets.

The data preprocessing of each feature in train and test data sets is summarized as below:

- Around 41% of society records are missing in the train data set; around 57% of records are missing in test data. So the feature society is dropped from both the data sets as it doesn't add much to the model.
- There are around 1305 different locations. One data point location record is missing .We have imputed the null record with 'others'. Since the feature location is categorical, we use thee Label Encoder to convert categorical into numerical feature.
- The null values present in balcony records around 609 data points were imputed with mode (most occurring value) '2' where null values in test data which were around 69 have been imputed with 2.
- The null values present in bath records have been imputed with mode (most occurring value) '2 BHK' in both sets.
- We observe that, all total-sqft records are not in square-feet in both the data sets. Some of them are in square-yards, acres, perch, guntha and grounds. Every data point with respect to total-sqft has been converted into square-feet by carrying out necessary transformations
- The area_type has four categories: Super built-up area, plot area, carpet area and built-up area. We have converted into dummy variables in both sets.
- The column size has records in BHK, bedroom and RK. The numerical part associated with BHK and Bedroom has been extracted and two separate features BHK and Bedroom have been created. The feature size has been excluded from data.
- We have grouped the availability records into two categories: Ready to Move and Others. Likewise, pre-processing steps in the data test set were performed.

All these data preprocessing steps have been carried out in Jupyter notebook python with necessary packages.

IV. Regression Models and Evaluation Metrics Used

Linear regression is one of the most well- known algorithms in statistics and machine learning. The objective of a linear regression model is to find a relationship between one or more features (independent/explanatory/predictor variables) and a continuous target variable (dependent/response) variable. If there is only one feature, the model is simple linear regression and if there are multiple features, the model is multiple linear regression [8].

A. Basic Linear Model

The formulation for multiple regression model is $Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 + \ldots\ldots\ldots + \alpha_p X_p$ The assumptions in the model are:

- The error terms are normally distributed.
- The error terms have constant variance.
- The model carries out a linear relationship between the target variable and the functions.

Here the multiple regression models are developed by the least square approach (Ordinary Least Squares / OLS). The accuracy of the designed model is difficult to measure without evaluating its output on both train and test data sets. This can be achieved using efficiency metric of some kind. It may be by measuring some type of error, fit's goodness, or some other useful calculation.

For this study, we evaluated model's performance using metrics: the coefficient of determination R^2, adjusted R^2 and RMSE (Root Means Square Error). (Root Means Square Error), RMLSE (Root Mean Squared Logarithmic Error).

1. RMSE: It can be defined as the standard sample deviation between the predicted values and the observed ones. It is to be noted that unit of RMSE is same as dependent variable y .The lower RMSE values are indicative of a better fit model. If the model's primary objective is prediction then RMSE is a stronger measure [7].
2. R-squared and Adjusted R-squared: The R-square value provides a measure of how much the model replicates the actual results, based on the ratio of total variation of outcomes as explained in the model.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_{predicted} - y_{observed})^2}{\sum_{i=1}^{n} (y_{predicted} - \overline{y}_{observed})^2}$$

The higher the R-squared, the better the model fits the data given. The R-squared value ranges from 0 to 1, representing the percentage of a squared correlation between the target variable's expected and real values. But in case of multiple linear regressions, R-squared value may increase with increasing features even though the model is not actually improving. A related, Adjusted R-squared statistic can be used to address this disadvantage. This measures the model's goodness and penalizes the model to use more predictors.

3. RMSLE (Root Mean Squared Logarithmic Error): It is the root mean squared error of the logarithmic-transformed predicted and log transformed actual values. The error term can be exploded to a very high value if outliers are present in case of RMSE, whereas in case of RMLSE the outliers are drastically scaled down therefore so that impact is nullifiedWe have split the training data into subsets of train and test data .We attempt to build linear regression model using OLS in python using necessary packages namely sk-learn and scikit packages after data pre-processing and train-test split of data set. We know that multi-colinearity is the effect of the multiple regression models having related predictors. In simple, when data set has a large number of predictors, it is possible that few of these predictors may be highly correlated. Existence of high correlation between independent variables is called multi-collinearity. Presence of multi-collinearity can destabilize the model. Apart from correlation matrix, to check this sort of relations between variables we use, variation inflation factor (VIF).It measures the magnitude of multi-collinearity. It is defined as follows:

$$VIF = \frac{1}{1 - R^2}$$

There are many approaches to handle multicollinearity .One simple technique is to dropping the variables from the model building that is highly correlated with others with the help of p-statistic and Variation Inflation factor. The threshold value for VIF is 5 .

VIF greater than 5 requires further investigation to assess the impact of multi-collinearity [7]. Taking into account all the above metrics, we attempt to build basic regression model. The model obtained is as follows:

$$\Pr ice = 0.2460 + 1.6971 * bath + 0.0064 * availabili\ ty + 0.0494 * Plot\ \ Area + 0.0488 * balcony + 7.488 * 10^{-6} * location + 6.002 * 10^{-6} * TOTAL\ \ \_ sqft - 0.0046 * Super\ built - up\ \ Area$$

The metrics obtained for this simpler model is summarised in TABLE 1:

TABLE 1: R-square and RMSE values for linear regression model

Metric	Train set	Test set
R-square	0.418	-2.12
RMSE	0.0912	0.2077
RMSLE	0.02755	0.03493

R- Squared value for model developed is 0.418 and adjusted R-squared value is 0.418. The p values and VIF with respect to features were in permissible range. We have got negative value for test set which needs further investigation.

A predictive model has to be simple as simple possible, but no simpler. Regularization is a method used to construct an optimally complex model, that is to say a model that is as simple as possible when performing well on training data. Through this process, we can strike balance between keeping model simpler, yet not making it too naïve to be of any use. The regression only tries to minimize the error and it does not account for model complexity. Some of the effective regularization techniques lower the complexity of the model and prevent overfitting.. In recent years, many researchers have used these advanced models to handle multi-collinearity. We then prepared a series of fits using two regularized linear regression models.-Ridge and Lasso regression models.

B. Ridge Regression

Ridge regression model is a regularization model, where an extra variable (tuning parameter) is added and optimized to address the effect of multiple variables in linear regression which is usually referred as noise in statistical context. In mathematical form, the model can be expressed as

$$y=xb+e$$

Here, y is the dependent variable x refers to features in matrix form and b refers to regression coefficients and e represents residuals. Based on this, the variables are standardised by subtracting the respective factors and dividing them by their standard deviations. The tuning function denoted as ƛ is then shown as aspect of regularization in the ridge regression model. If ƛ's value is large then the squares ' residual sum appears to be zero. If it is less than the solutions conform to least square method. ƛ is found out using a technique called cross -validation. Ridge regression reduces the coefficients to arbitrarily low values though not to zero. We will also perform gridsearch cross-validation to tune the regularisation hyper parameter λ [5].We have chosen wide range for hyper-parameter and found 0.001 as best value. The model obtained has an array of coefficients:

Array ([2.86025752e-02, 4.16108845e-05, 1.83397566e-01, 5.65203565e-02, -3.85346357e-02, 4.89806025e-01, -4.48912340e-02, 2.25223943e-01, 1.24512000e-01, 1.39492670e-06])

The evaluations metrics for the model is summarized in TABLE 2:

TABLE 2: R-square and RMSE values for ridge regression model

Metric	Train set	Test set
R-square	0.4345	0.4358
RMSE	0.5415	0.5224
RMSLE	0.0410	0.040701

We observe RMSE and R- square values are almost same (with slight differences) for both train and test data. Later predictions have made on test data.

C. Lasso Regression

LASSO means least absolute shrinkage, and the selection operator is an LR technique that also regularizes functionality. It is identical to ridge regression, except that it varies in the values of regularisation. The absolute values of the sum of regression coefficients are taken into consideration. It even sets the coefficients to zero so it completely reduces the errors. So selection of features are resulted by lasso regression. In the previously mentioned ridge equation, the component ' e ' has absolute values instead of squared values [5].

It is to be noted that computationally Lasso regression technique is far more intensive than Ridge regression technique. We have performed grid-search cross- validation to tune the regularisation hyper parameter λ. We have chosen wide range for hyper-parameters and found 0.001 as best value. The model obtained has an array of coefficients:

Array ([2.26872602e-02, 3.84815301e-05, 1.88127460e-01, 6.35927974e-02, -0.00000000e+00, 4.65240926e-01, -5.40347504e-02, 2.12552087e-01, 1.13374140e-01, 3.49669654e-07])

We see that one of coefficient of features has been reduced to zero. The metrics obtained is summarized as follows and shown in table 3.:

TABLE 3: R-square and RMSE values for lasso regression model

Metric	Train set	Test set
R-square	0.4341	0.4430
RMSE	0.5416	0.5224
RMSLE	0.04105	0.04069

The model developed can now be used to make predictions on test data where the value of the target variable is unknown.

D. SVR (Support Vector Regression)

In simple linear regression we attempt to minimize the error, whereas in SVR we try to fit he error within a certain threshold. It is a regression algorithm and uses a similar Support Vector Machines (SVM) methodology for regression Analysis [10].Regression data contains continuous real numbers. To fit such type of data, the SVR model approximates the best values with a given margin called ε tube (epsilon-tube, ε identifies a tube width) with considering the model complexity and error rate shown in table 4.

TABLE 4: R-square and RMSLE values for lasso regression model

Metric	Train set	Test set
R-square	0.799	0.6630
RMSLE	0.0256	0.0317

E. XGBoost Regression Model

XGBoost stands for extreme gradient boosting which is most efficient technique for either regression or classification problem. It is decision tree based algorithm that make use of gradient boosting framework. It provides the features that greatly have impact on performance of model.
This technique helps in developing a model that have less variance and more stability. In addition, the execution speed is fast when compared to other algorithms shown in table 5.

TABLE 5: R-square and RMSE values for XGBoost regression model

Metric	Train set	Test set
R-square	0.7868	0.7584
RMSE	0.3309	0.3462
RMSLE	0.0256	0.0317

V. CONCLUSIONS AND FUTURE SCOPE

An optimal model does not necessarily represent a robust model. A model that frequently use a learning algorithm that is not suitable for the given data structure. Sometimes the data itself might be too noisy or it could contain too few samples to enable a model to accurately capture the target variable which implies that the model remains fit.
When we observe the evaluation metrics obtained for advanced regression models, we can say both behave in a similar manner. We can choose either one for house price prediction compared to basic model. With the help of box plots, we can check for outliers. If present, we can remove outliers and check the model's performance for improvement.

We can build models through advanced techniques namely random forests, neural networks, and particle swarm optimization to improve the accuracy of predictions.

VI. MODEL APPLICABILITY

It is necessary to check before deciding whether the built model should or should not be used in a real-world setting .The data has been collected in 2016 and Bengaluru is growing in size and population rapidly. So, it is very much essential to look into the relevancy of data today. The characteristics present in the data set are not sufficient to describe house prices in Bengaluru. The dataset considered is quite limited and there are a lot of features, like the presence of pool or not, parking lot and others, that remain very relevant when considering a house price. The property has to be categorized either as a flat or villa or independent house. Data collected from a big urban city like Bengaluru would not be applicable in a rural city, as for equal value of feature prices, which will be comparatively higher in the urban area.

REFERENCES

[1] H.L. Harter,Method of Least Squares and some alternatives-Part II.International Static Review.1972,43(2) ,pp. 125-190.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] Lu. Sifei et al,A hybrid regression technique for house prices prediction. In proceedings of IEEE conference on Industrial Engineering and Engineering Management: 2017.

[4] R. Victor,Machine learning project:Predicting Boston house prices with regression in towards datascience.

[5] S. Neelam,G. Kiran,Valuation of house prices using predictive techniques, Internal Journal of Advances in Electronics and Computer Sciences:2018,vol 5,issue-6

[6] S. Abhishek.:Ridge regression vs Lasso,How these two popular ML Regression techniques work. Analytics India magazine,2018.

[7] S.Raheel.Choosing the right encoding method-Label vs One hot encoder. Towards datascience,2018.

[8] Raj, J. S., & Ananthi, J. V. (2019). Recurrent Neural Networks and Nonlinear Prediction in Support Vector Machines. *Journal of Soft Computing Paradigm (JSCP),* 1(01), 33-40.

[9] Predicting house prices in Bengaluru(Machine Hackathon) https://www.machinehack.com/course/predicting-house-prices-in-bengaluru/

[10] Raj, J. S., & Ananthi, J. V. (2019). Recurrent neural networks and nonlinear prediction in support vector machines. Journal of Soft Computing Paradigm (JSCP), 1(01), 33-40.

[11] Pow, Nissan, Emil Janulewicz, and L. Liu (2014). Applied Machine Learning Project 4 Prediction of real estate property prices in Montréal.

[12] Wu, Jiao Yang(2017). Housing Price prediction Using Support Vector Regression.

[13] Limsombunchai, Visit. 2004.House price prediction: hedonic price model vs. artificial neural network.New Zealand Agricultural and Resource Economics Society Conference.

[14] Rochard J. Cebula (2009).The Hedonic Pricing Model Applied to the Housing Market of the City of Savannah and Its Savannah Historic Landmark District; The Review of Regional Studies 39.1 (2009), pp. 9–22

[15] Gu Jirong, Zhu Mingcang, and Jiang Liuguangyan. (2011).Housing price based on genetic algorithm and support vector machine". In: Expert Systems with Applications 38 pp. 3383–3386.

[16] Danny P. H. Tay and David K. H. Ho.(1992)Artificial Intelligence and the Mass Appraisal of Residential Apartments. In: Journal of Property Valuation and Investment 10.2 pp. 525–540.