Real Estate Investment Advising Using Machine Learning

Dr. Swapna Borde, Department of Computer Engineering, VCET, Mumbai University, India

Aniket Rane, Department of Computer Engineering, VCET, Mumbai University, India

Gautam Shende, Department of Computer Engineering, VCET, Mumbai University, India

Sampath Shetty, Department of Computer Engineering, VCET, Mumbai University, India

***Abstract*— The project makes a comparative study of various Machine Learning algorithms namely Linear Regression using gradient descent, K nearest neighbor regression and Random forest regression for prediction of real estate price trends. The aim of this paper is to examine the feasibility of these machine learning algorithms and select the most feasible one. To achieve the aim, parameters like Living Area, Number of rooms, Distance from airport/highway/station/major landmarks, Proximity to hospitals, Shopping options, Number of theaters, geographical location(harbor/central/western) are used as the input to the model and real estate price in the next quarters is the output variable. The quarterly data during 2005-2016 is employed as the data set to construct the model and the data has been obtained using Web Scraping from websites like 99acres.com, Magicbricks.com, Google.com. The experimental results on the training data set are used to compare the various algorithms based on error calculation using Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE).**

***Keywords*— Linear Regression, K Nearest Neighbor (KNN), Random Forest algorithm, forecast**

I. INTRODUCTION

Prices of real estate properties are critically linked with our economy. Accurately predicting real estate prices is not possible. However, prediction of real estate trends is a realistic prospect. Yet, we do not have accurate measures of housing price trends based on the vast amount of data available. In Mumbai alone there are around 10,000 current listings of 350 areas or more at 99acres.com. This rich dataset should be sufficient to establish a regression model to accurately predict the real estate prices in Mumbai.

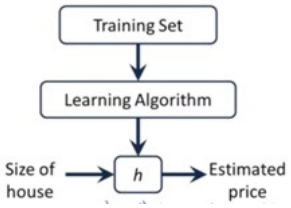
A property's appraised value is important in many real estate transactions such as sales, loans, and its marketability. Traditionally, estimates of property prices are often determined by professional appraisers. The disadvantage of this method is that the appraiser is likely to be biased due to vested interest from the lender, mortgage broker, buyer and seller. Therefore, an automated prediction system can serve as an important third party source which is not biased.

For the buyers of real estate properties, an automated price prediction system can be useful to find under/overpriced properties currently on the market. This can be useful for first time buyers with relatively little experience, and suggest purchasing strategies for buying properties.

This paper makes a comparative study of the three mentioned algorithms viz. Linear Regression using gradient descent, KNN Regression, Random forest regression for analyzing the trends. The data set extracted is split into training set and testing set in a ratio of 80:20 or 4:1. Finally, based on the conclusions drawn from the comparative study of the algorithms are used to develop a front end that suggests the user areas that would be most profitable for investment. On the front end, the user is asked for parameters such as Total budget and Total Area and based on the prediction of the prediction model which implements the most feasible algorithm for our data set, the user is provided with suggestions for investment

II. PREDICTION BASED ON LINEAR REGRESSION USING GRADIENT DESCENT

Linear Regression principle is basically used for prediction and forecasting. Being the simpler and basic technique as compared to other machine learning algorithms, the implementation is less complex but the degree of error is slightly above the average. In this paper, we focus on Liner Regression using Gradient Decent for regression and forecast. Linear Regression using Gradient Descent is a learning machine which estimates a function according to a given data set $G=\{(x_i, y_i)\}^n$, where x_i denotes the input vector, y_i denotes the output value and n denotes the total number of the data. If the learning machine is working on multi-parameters, x_i denotes the input matrix and y_i denotes the output vector.



Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

So basically in linear regression learning machine, we need to fit the training data set with the best fit line which we call the hypothesis in our paper.

For hypothesis, we need to know the values of θ_0 and θ_1 so that for each value of x we can predict the best possible value of corresponding y.

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

We need to evaluate the cost function (J) for set of data pairs to find the coefficients value.

For the error to be minimum, we need to minimize the cost function (J), such that the coefficients which give the minimum cost function are selected. We used gradient descent algorithm to minimize the cost function. So the Gradient Descent algorithm will give values of coefficients for the hypothesis.

Gradient descent algorithm:

Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

Gradient descent algorithm

repeat until convergence {

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ (for $j = 0$ and $j = 1$)

}

Figure 1.1 demonstrates training set for a single parameter (size in feet) and the output (price). In Figure 1.2 we plot the training set such that parameter (size in feet) is plotted on the X axis and the prices are plotted on Y axis. The blue line is the prediction line(hypothesis).

Size in feet^2 (x)	Price(Rupees) in 1000's(y)
150	**165**
180	**195**
280	**300**
350	**300**
…..	**.....**

Figure 1.1

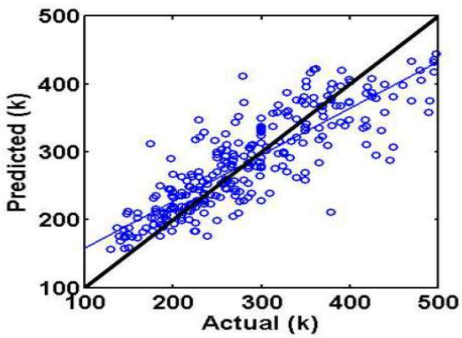


Figure 1.2

III. PREDICTION BASED ON KNN ALGORITHM

K nearest neighbours is a simple algorithm that stores all available cases and predicts the numerical target based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique. A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the K nearest neighbors. KNN regression uses the same distance functions as KNN classification. [9]

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k} |x_i - y_i|$

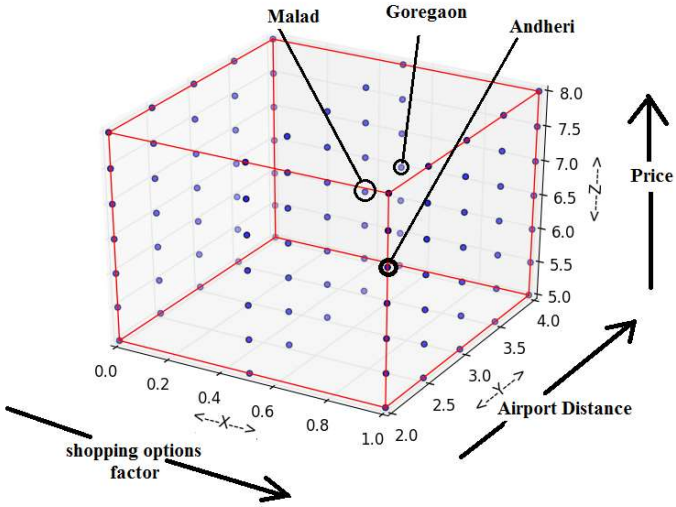
Minkowski $\left(\sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{1/q}$

Overview of the Algorithm:

1) Fix K. Here K is the number of nearest neighbours that will be used for prediction.
2) Select method for distance calculation (Hamming Distance / Euclidean Distance / Minkowski / etc)
3) Calculate weights of all vectors from the test vector.
4) Find the k closest vectors in the multidimensional system based upon the distances calculated.
5) Voting by the k nodes and estimation using a function like median or inverse weighted distance average.

In our model we use the most common distance function i.e the Euclidean distance function. We also experimented with various values of K as it has been established that beyond a threshold value of K, there is no significance increase in the accuracy and the problem of over fitting and over-bias comes into the picture. An instance of this was also found in the study of the paper [1]

The kNN model in our system can be visualized as follows



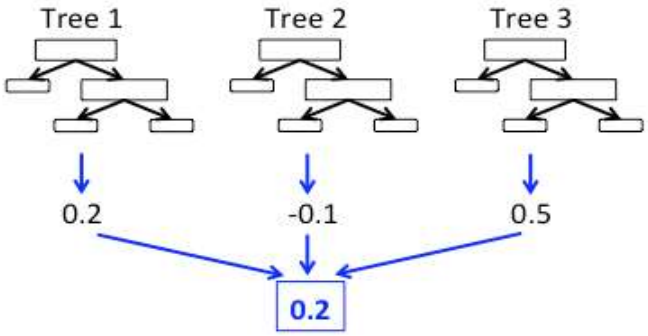
This visualization can be extended to an n dimensional plot and the training set populates the n dimensional space. Later on, for prediction, we find the votes of the k closest neighbours in this n dimensional plot.

IV. Prediction based on Random Forest Algorithm

Random forest algorithm is an ensemble learning method for regression and other tasks that makes use of decision tree learning and finding the mean prediction of the output of individual trees.

A decision tree is built using the top-down approach starting from a root node and at each stage the data is partitioned into subsets that contain instance with similar values. Each internal (non-leaf) node in the decision tree is labeled with an input feature. Based on the best split for an input feature the decisions are made and accordingly the tree is grown. The branches coming from a node labeled with an input feature are labeled with each of the possible values of the target or output feature or the branch leads to a subordinate decision node on a different input feature. The leaf node gives us the output based on the decisions made. In our case it would be the value of a property.

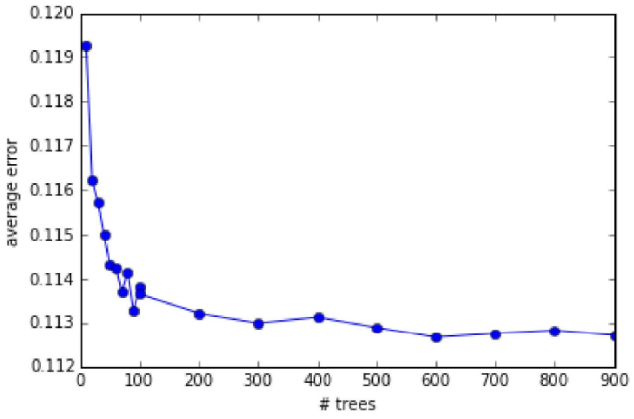
Random forest algorithm takes the decision tree concept further by producing a large number of decision trees. The approach selects the training sample as well as a set of features randomly to grow each decision tree. The output of Random forest model is the average of the output of individual decision tree.



The basic algorithm is as follows:

1) Select a set of training samples randomly from the complete training set.
2) Select a set of parameters at random out of all the possible parameters.
(The above steps would make sure that all the parameters are given importance. Thus avoiding over-fitting the data)
3) Construct a Decision tree
- Find the best split on the selected parameter
- Make decision based on the split and grow the tree accordingly. The leaf node would give the predicted value.

4) Repeat the above steps for a number of decision trees (until the threshold value is reached after which the error value remains same).
5) Finally, find the mean of the output of individual decision tree which would give us the required prediction.



It has been established that there is a threshold beyond which there is no significant gain in accuracy as the number of trees increases as was also the case in the paper [1]. This helps us in deciding the number of trees to be generated by experimenting with the number of trees.

V. EVALUATION

Linear Regression:
Features- Year/Quarter, Distance to Airport, Distance to Altamount road, Distance to Vashi, Distance to Virar, Shopping mall count, Hospital Count

Coefficients obtained –

1) Using gradient descent -
293.539,-42.823,-472.687, 409.558, 219.353, 93.805, -127.294
Bias = 124.885794

2) Using least squares -
264.143, -40.785, -472.843, 373.372, 181.876, 64.645, -202.342
Bias = 4317.805

As expected, the features like distance to Altamount Road and Distance to Airport have negative coefficients as the price is expected to drop as we move away from these locations. It should be noted that Altamount is most valued real estate area in Mumbai. On the flip side, it can be seen that the Year/ Quarter and Distance to Vashi, Distance to Virar and number of shopping malls have positive coefficients because more the number of malls, more likely is the area to be developed, however it can be seen that the mall coefficient is not given too much of an importance because of the conflict that a developed area will have higher real estate pricing so it would also not be likely to have too many malls. It is obvious that increase in distance from places such as Virar and Vashi would most likely mean migration toward the Suburbs and Southern Mumbai which consequently results in the positive co-efficient. It is intuitive to think that increase in the number of hospitals would mean an increase in the price, however this is true only to some extent beyond which if the number of hospitals is more, then it's an indication that price is rather low in that area so as to construct so many hospitals. Since Linear Regression cannot distinguish this distinction between too many and fairly enough due to its linearity, it considers the overall effect which turns out to be that increase in number of hospitals leads to a net decrease in price.

K-Nearest Neighbors:

Error table for k = 2, 3,4,5,6

	2	3	4	5	6
RMSE	0.016	0.019	0.020	0.023	0.024
MAE	0.079	0.092	0.097	0.105	0.110
MAPE	7.528	8.691	9.599	10.302	11.051

It can be seen that as the number of neighbours' increases, the accuracy decreases and as the number of neighbours' decreases, there is a gradual increase in the accuracy. As we increase the number of neighbours, we get a more generalized prediction however in our training set, it means the loss of accuracy. Hence, we preferred keeping k=4 which we found to be a good point for balancing the factor of generalization and accuracy.

Random Forest:

The random forest regression was found to have a very low error and fit the data very well. However, unexpectedly, it was not able to incorporate accurately the time/quarter feature. There is not a huge difference between predictions for a quarter in 2017 and a quarter in 2020. Thus even though it is able to predict near future trends with good accuracy, the predictions aren't coherent if we consider a prediction for (say) the next 6 quarters. Here, since we are only interested in predicting the trends for the next 2 or 3 quarters due to the dynamicity of the Real Estate pricing, this limitation specific to our data set is not a matter of concern.

Final table:

	Linear Regression using least squares	Linear Regression using gradient descent	K Nearest Neighbor (K=4)	Random Forest
RMSE	0.118	0.122	0.020	0.007
MAE	0.301	0.319	0.097	0.062
MAPE	40.089	45.925	9.599	6.328

VI. CONCLUSION

This paper proposes forecasting of Real Estate trends in Mumbai in near future by implementing regression using Random Forest algorithm as it was found to have the less error than the other algorithms that were experimented with, namely Linear Regression and K-nearest neighbors. The features used were Year/Quarter, Distance to Airport, Distance to Altamount road, Distance to Vashi, Distance to Virar, Shopping mall count, Hospital Count. The data which consisted of over 5000 samples was split into a 4:1 ratio for training and testing sets respectively. The prices of most recent quarters were used in the testing set. In future, besides adding more features such as traffic conditions and population density, we also intend to develop a user friendly GUI that can serve as an unbiased virtual broker for potential investors. We also plan to extend this model to other major cities in India like Bangalore, Chennai, Delhi and Kolkata with some changes in features that are specific to the considered city.

REFERENCES

[1] Pow, Emil Janulewicz and Liu Dave (Mc Gill University),"Applied Machine Learning : Prediction of real estate properties in Montreal, Canada".

[2] DA-YING LI1, WEI XU2, HONG ZHAO3 and RONG-QIU CHEN1 ,"A SVR based forecasting approach for real estate price prediction", Proceedings of the Eighth

International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009.
[3] Aaron NG and Dr Mark Deisenroth (Imperial College, London) ,“Machine Learning for a London housing Price Prediction Mobile Application”.
[4] R. J. Shiller, “Understanding recent trends in house prices and home ownership”, National Bureau of Economic Research.
[5] S. C. Bourassa, E. Cantoni, and M. E. Hoesli, “Spatial dependence, housing submarkets and house price prediction”, 2007.
[6] Course on “Machine Learning”, Andrew Ng(Linear Regression). Available:https://www.coursera.org/learn/machine-learning
[7] The elements of statistical learning, Trevor Hastie - Random Forest Generation
[8] K nearest neighbours, Sckitlearn.org
[9] K nearest neighbours regression, http://www.saedsayad.com/k_nearest_neighbors_reg.htm