**House Price Prediction**

**Summary of problem statement, data and findings**

**Abstract**

Investment in real estate generally seems to be profitable because their property values do not decline rapidly. Changes in the real estate price can affect various household investors, bankers, policy makers and many. Investment in real estate sector seems to be an attractive choice for the investments. Generally, the property values rise with respect to time and its appraised value need to be calculated. This appraised value is required during the sale of property or while applying for the loan and for the marketability of the property. These appraised values are determined by the professional appraisers. However, drawback of this practice is that these appraisers could be biased due to bestowed interests from buyers, sellers or mortgages. Thus, we require an automated prediction model that can help to predict the property values without any bias. This automated model can help the first-time buyers and less experienced customers to understand whether the property rates are overrated or underrated.

**Introduction**

People are looking to buy a new home tend to be more conservative with their budgets and market strategies. House price prediction can help the developer determine the selling price of a house and to customer to arrange the right time to purchase a house. This system helps to find a starting price for a property based on the geographical variables. By breaking down past market patterns, value ranges and coming advancements future costs will be anticipated. The goal is to predict efficient house pricing for real estate with respect to customers budgets and priorities. It is one of the prime fields to apply the ideas of machine learning on how to enhance and foresee the costs with high accuracy.

**Overview of the final process**

We are predicting the sale price of the houses using various machine learning algorithms and comparing different Models. Housing sales price are determined by numerous factors such as area of the property, location of the house, quality, area of property, number of bedrooms and garages and so on. Machine learning algorithms are used to build the prediction model for houses. Here, machine learning algorithms such as logistic regression, Decision Tree, Ada boosting, Gradient boosting, Random Forest, Lasso Regression, Ridge Regression are employed to build a predictive model. We have considered housing data of 21613 properties. we have compared these algorithms based on parameters such as MAE, MSE and accuracy. This also represents significance of our approach and the methodology

**Step-by-step walk through the solution**

**Methodology**

Methodology represents a description about the framework that is undertaken. It consists of various milestones that need to be achieved in order to fulfil the objective. We have undertaken different data mining and machine learning concepts.

**Data Collection -> Pre-processing -> Data Analysis -> Application of Algorithms -> Model Evaluation**

**1) Data Collection**

The dataset "Innercity" used in this project consists of 21613 records with 23 parameters that have the possibility of affecting the property prices.

Overall quality which rates the overall condition and finishing of the house is room\_bed: Number of Bedrooms/House, room bath: Number of bathrooms/bedrooms, living\_measure: square footage of the home, lot\_measure : quare footage of the lot, ceil: Total floors (levels) in house,coast: House which has a view to a waterfront, sight: Has been viewed, condition: How good the condition is (Overall),quality: grade given to the housing unit, based on grading system,ceil\_measure: square footage of house apart from basement,basement\_measure: square footage of the basement , yr\_built: Built Year,yr\_renovated: Year when house was renovated, zipcode : zip , lat: Latitude coordinate,long: Longitude coordinate,living\_measure15: Living room area in 2015(implies--some renovations) This might or might not have affected the lotsize area,lot\_measure15: lotSizearea in 2015(implies--some renovations),furnished: Based on the quality of room,total\_area: Measure of both living and lot.

**2) Data Pre-processing**

It is a process of transforming the raw, complex data into systematic understandable knowledge. It involves the process of finding out missing and redundant data in the dataset. Entire dataset is checked for NaN and whichever observation consists of NaN will be deleted. Thus, this brings uniformity in the dataset. However in our dataset, there was no missing values found that means each record was constituted its corresponding feature values.

**3) Data Analysis**

Before applying any model to our dataset, we need to find out characteristics of our dataset. Thus, we need to analyse our dataset and find relationship between the parameters.

Data distribution has been found by using skew method.

Dataset has parameter with integer, object and float data type.

Parameter like cid,price,room\_bed,living\_measure,lot\_measure,coast,sight,condition,quality, ceil\_measure,basement, yr\_built,yr\_renovated, zipcode,living\_measure15, lot\_measure15,furnished, total\_area are of integer type where as dayhours is of object typewhere as dayhours is of object type and rest parameters like room\_bath,ceil,lat,long has float data type.

As the parameter dayhours is in object type, it is conveted to integer type in year format with column heading 'sold\_year, and dayhours column is dropped.

As mentioned out of these 23 parameters only 21 were chosen which are bound to affect the housing prices. Parameters such as cid(a notation for a house) and Zipcode were dropped as they have least effect on house pricing.

As latitude and longitude of location is provided , zipcode is not that important in finding prices.

In column 'yr\_renovated,the data is in 0s and 1s.0 denotes house has not been renovated and 1 denotes house is renovated.So, we have 914 houses which are renovated and 20699 houses not renovated.

Correlation number gives the degree of association between two variables. The correlation number exists between +1 to -1. Some of the parameters are highly coorelated with each other like price with living\_measure(square footage of the home),sight,Number of bathrooms/bedrooms,square footage of house apart from basement,furnished(Based on the quality of room) etc. Some parameters show least correlation with each other like price with sold\_year or price with contidion.

Data distribution,visualisation and outliers detection has been done on parameters which are likely to have high influence on target variable that is price.

Box plot, distance plot and histogram and scatter plot has been used to show the relation between parameters and target variable visually.Parameters are room\_bed, room\_bath,living\_measure,lot\_measure,ceil,coast,sight,condition,quality,ceil\_measure, lat,long,living\_measure15,lot\_measure15,furnished,total\_area,basement.

**Box plot**

A box-whisker plot is a univariate plot used to visualize a data distribution.

The ends of whiskers are the maximum and minimum range of data distribution. The central line in the box is the median of the entire data distribution. The right and left edges in the box are the medians of data distribution to the right and left from the central median, respectively.

**Histogram**

A histogram is a graphical representation of a frequency distribution where data points are organized as bins, plotted with values along the x-axis and the count of data points in each bin along the y-axis.

**Distplot**

Distplot is a histogram with a line on it.A distplot plots a univariate distribution of observations.The distplot() function combines the matplotlib hist function with the seaborn kdeplot() and rugplot() functions.

**Scatter plot**

Scatter plot is used to understand relationship between two different attributes in the dataset. Below we have compared Price (target) vs each of the attribute in the dataset.

From the analysis we found there are outliers in out data set by using IQR method and they are treated using Z score and were removed from data set for better model building and higher accuracy.The shape data before outlier removal was (21613, 23) and The shape of data after outlier removal was (18702, 21).

Multivariate analysis and pairplot has been done to show relationship between the parameters.

**4) Application of Algorithim**

Once the data is clean and we have gained insights about the dataset, we can apply an appropriate machine learning model that fits our dataset.

**Why Regression?**

The below shows why Regression is chosen over Classification

* The relationship between the variables is linear.
* The data is homoskedastic, meaning the variance in the residuals (the difference in the real and predicted values) is more or less constant.
* The residuals are independent, meaning the residuals are [distributed randomly](http://gsb420.blogspot.fi/2008/03/lecture-8-residual-analysis-checking_04.html) and not influenced by the residuals in previous observations. If the residuals are not independent of each other, they’re considered to be autocorrelated.
* The residuals are normally distributed. This assumption means the probability density function of the residual values is normally distributed at each x value.

**Model evaluation**

We have selected seven algorithms to predict the dependent variable in our dataset.

By looking at the dataset, we simply can’t suggest the best Regression Model for this problem. So, we will try out different Regression models available in scikit-learn with a k-fold cross validation method.

let's assume k = 10 (k-fold cross validation)

It means we split the training data into train and test data using a test\_size parameter for 10-folds. Each fold will have different samples that are not present in other folds. By this way, we can thoroughly train our model on different samples in the dataset.

Before doing anything, we will split our dataframe df into features X and target Y.

The seven algorithms are Logistic Regression, Decision Tree, Ada boosting, Gradient boosting, Random Forest, Lasso Regression Technique and Ridge Regression. These algorithms were implemented with the help of python’s SciKit-learn Library.The predicted outputs obtained from these algorithms were saved in comma separated value file. This file was generated by the code at run time.

One variable, denoted x, is regarded as the predictor, explanatory, or independent variable. The other variable, denoted y, is regarded as the response, outcome, or dependent variable.

Data has been split into Train and Test data set with 70:30 ratio.

**Linear Regression**

Linear regression models are used to show or predict the relationship between two variables or factors. The factor that is being predicted (the factor that the equation solves for) is called the dependent variable. In order to use linear regression the quarter attribute is assigned on x-axis and the values of rates on y-axis. For each of the attribute linear regression is performed once. The x-axis being independent is the choice available to the user to select from a dropdown list.

Performance on training data using Linear Regression: 0.6813652804824706

Performance on testing data using Linear Regression: 0.6846292202079216

Accuracy LR: 0.6846292202079216

MSE: 17336082488.007713

MAE: 92785.73570294016

**Decision Tree model**

Decision trees are considered to be the best and most widely used supervised learning algorithm. This model has the ability to predict the output with at most accuracy and stability. It is used to predict any kind of problems such as classification or regression. However, in our case we want to predict a continuous target value hence our problem is of regression type. In this model, the available dataset can be continuous or categorical.

**Overfitting** is a term used in statistics that refers to a modelling error that occurs when a function corresponds too closely to a particular set of data. As a result, overfitting may fail to fit additional data, and this may affect the accuracy of predicting future observations. Some of the methods used to prevent overfitting include ensembling, data augmentation, data simplification, and cross-validation.

Detecting overfitting is almost impossible before you test the data. It can help address the inherent characteristic of overfitting, which is the inability to generalize data sets. The data can, therefore, be separated into different subsets to make it easy for training and testing. The data is split into two main parts, i.e., a test set and a training set.

The training set represents a majority of the available data (about 80%), and it trains the model. The test set represents a small portion of the data set (about 20%), and it is used to test the accuracy of the data it never interacted with before. By segmenting the dataset, we can examine the performance of the model on each set of data to spot overfitting when it occurs, as well as see how the training process works. The performance can be measured using the percentage of accuracy observed in both data sets to conclude on the presence of overfitting. If the model performs better on the training set than on the test set, it means that the model is likely overfitting.

Here Performance on training data using Decision tress is 0.686260061416718 and testing data Performance is 0.6872872170495214.

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

Performance on training data using Decision Tree: 0.9996195969820089

Performance on testing data using Decison Tree: 0.7265266655872384

Accuracy DT: 0.7265266655872384

MSE: 15032959891.768757

MAE: 80559.20673676707

Performance on training data using Decision tress is 0.686260061416718 and testing data Performance is 0.6872872170495214 after reducing Overfitting.

KFold cross validation: 0.679%

Performance on training data using Decision Tree model with best parameter 0.8459240607603937

Performance on testing data using Decision Tree model with best parameter: 0.7920570202828918

The best parameter is checked by random cv.

**Random search** is a technique where random combinations of the hyperparameters are used to find the best solution for the built model. It is similar to grid search, and yet it has proven to yield better results comparatively.

**Ada-boosting model**

Boosting in machine learning is a way of combining multiple simple models into a single composite model.

Ada-boosting model combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set.

Ada boosting model is created with below update:

Performance on training data using Ada-boosting: 0.6397341556645111

Performance on testing data using Ada-boosting: 0.6364515517059215

Accuracy AD: 0.6364515517059215

MSE: 19984431950.761406

MAE: 118412.00280279652

K Fold cross validation: 0.622%

Performance on training data using Ada-boosting with best parameter 0.7137897925745126

Performance on testing data using Ada-boosting with best parameter: 0.7032362386304609

The best parameter is checked by random cv.

**Gradient boosting model**

we combine more and more simple models, the complete final model becomes a stronger predictor. The term "gradient" in "gradient boosting" comes from the fact that the algorithm uses gradient descent to minimize the loss. When gradient boost is used to predict a continuous value, we are using gradient boost for regression. This is not the same as using linear regression.

Gradient boostingmodel is created with below update:

Performance on training data using Gradient-boosting: 0.8604559034581047

Performance on testing data using Gradient-boosting: 0.843253689887486

Accuracy: 0.843253689887486

MSE: 8616419579.495983

MAE: 62298.31001070939

KFold cross validation:0.846%

Performance on training data using Gradient boosting with best parameter: 0.9399614061120317

Performance on testing data using Gradient boosting with best parameter: 0.879376553749345

The best parameter is checked by random cv.

**Random Forest**

In Random Forest, the number of features that can be split on at each node is limited to some percentage of the total (which is known as the hyperparameter). This ensures that the ensemble model does not rely too heavily on any individual feature, and makes fair use of all potentially predictive features. Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents overfitting.

Random Forest model is created with below update:

Performance on training data using Random Forest: 0.9813064602004667

Performance on testing data using Random Forest: 0.8715015329896307

Accuracy RF: 0.8715015329896307

MSE: 7063622143.887199

MAE: 55480.96299428419

K Fold cross validation: 0.868%

Performance on training data using Random Forest with best parameter:0.9725719835234887

Performance on testing data using Random Forest with best parameter: 0.8739691242379061

The best parameter is checked by random cv.

**Lasso Regression model**

Lasso is a powerful regression technique. It works by penalizing the magnitude of coefficients of features along with minimizing the error between predicted and actual observations. Lasso is called as L1 Regularization technique. Lasso attempts to minimize the cost function.

Lasso Regression model is created with below update:

Performance on training data using Lasso Regression: 0.6800760820039002

Performance on testing data using Lasso Regression: 0.6824732150071855

Accuracy las: 0.6824732150071855

MSE: 17454599124.29588

MAE: 93164.91119845033

KFold cross validation: 0.679%

**Ridge regression model**

Ridge Regression performs L2 regularization, i.e. adds penalty equivalent to square of the magnitude of coefficients.

Performance on training data using Ridge Regression: 0.6813794948326115

Performance on testing data using Ridge Regression: 0.6846224378308359

Accuracy rid: 0.6846224378308359

MSE: 17336455318.517567

MAE: 92786.1600888913

K Fold cross validation: 0.681%

**Stacking Ensemble Model**

Stacking or Stacked Generalization is an ensemble machine learning algorithm. Stacking is another ensemble model, where a new model is trained from the combined predictions of two (or more) previous model. The predictions from the models are used as inputs for each sequential layer, and combined to form a new set of predictions.   
Steps involved in stacking:

* The train set is split into training and validation sets.
* We train the base models on the training set.
* We make predictions only on the validation set and the test set.
* The validation predictions are used as features to build a new model.
* This model is used to make final predictions on the test set using the prediction values as features.

Accuracy of Stacking Model: 0.8721159284224861

### Choosing the best model

On comparing the various models based on Accuracy we find that Random Forest Random search and Gradient Boost with random search works the best with accuracy of 0.873969 & 0.879377 and Ada-boosting with random search parameter model performs least with an accuracy of 0.636452.

In Stacking model, we have combined Random Forest random search ,Gradient boost random search model, Ada boost random search , Decision Tree & Ridge model with random search. So for better prediction and to achieve a more generalised model we will be selecting Stacking Model which is the combination of all the best regressors as our final model which has the accuracy of 0.872116.

|  |  |  |
| --- | --- | --- |
| **Sl no.** | **Model\_names** | **Accuracy** |
| 1 | ada\_boost | 0.636452 |
| 2 | Lasso\_kfold | 0.679374 |
| 3 | linear Regression\_kfold | 0.680592 |
| 4 | Ridge\_kfold | 0.680632 |
| 5 | Lasso | 0.682473 |
| 6 | Ridge | 0.684622 |
| 7 | linear Regression | 0.684629 |
| 8 | decision\_tree | 0.687287 |
| 9 | ada\_boost\_random\_search | 0.703236 |
| 10 | decision\_tree\_random\_search | 0.792057 |
| 11 | gardient\_boost | 0.843254 |
| 12 | random\_forest | 0.871502 |
| **13** | **Stacking** | **0.872116** |
| 14 | random\_forest\_random\_search | 0.873969 |
| 15 | gardient\_boost\_random\_search | 0.879377 |

**Pipeline build**

A machine learning pipeline describes or models your ML process: writing code, releasing it to production, performing data extractions, creating training models, and tuning the algorithm.

Machine learning (ML) pipelines consist of several steps to train a model. Machine learning pipelines are iterative as every step is repeated to continuously improve the accuracy of the model and achieve a successful algorithm. To build better machine learning models, and get the most value from them, accessible, scalable and durable storage solutions are imperative, paving the way for on-premises object storage.

This pipeline can be used for further modification, future reference and deployment.

Test score: 0.87

**Comparison to benchmark**

Random Forest Random search and Gradient Boost with random search works the best with accuracy of 0.873969 & 0.879377 which minimum accuracy difference.

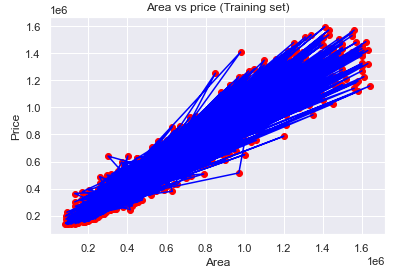
But we choose Stacking model over these with accuracy of 0.872116 as stacked models in real-world big data problems can produce greater prediction accuracy and robustness than do individual models. The model stacking approach is powerful and compelling enough to alter the initial data mining mindset from finding the single best model to finding a collection of really good complementary models. Basically used for better prediction and to achieve a more generalised model.

**Visualizations**

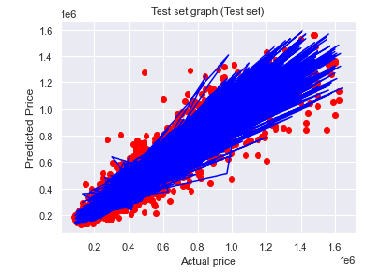
Predicted values of chosen best model that is Ensembled stacked model has been printed and plotted (predicted values on train and test data)

Scatter plots are used here as they show how much one variable is affected by another.The below graphs shows how closely and positivelyare area and price,actual price and predicted value , true value and predicted value are associated with other.

Graph has been plotted between Area vs price (Training set) using scatter plot.



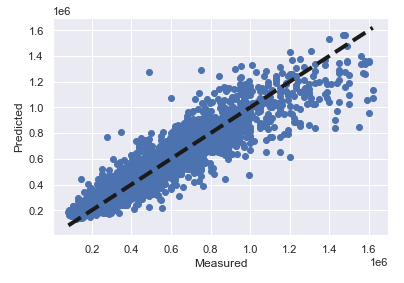
Graph has been plotted between Actual price vs Predicted Price(Test set) using scatter plot.



Graph has been plotted between True value vs predictions using scatter plot.



Graph has been plotted between Predicted vs Measured value using scatter plot.



**Implications**

Property investment has increased significantly since 2011, both on demand and property selling. One of the increasing of property demand is because of high population.Prediction of house prices are expected to help people who plan to buy a house so they can know the price range in the future, then they can plan their finance well. In addition, house price predictions are also beneficial for property investors to know the trend of housing prices in a certain location.However people can prioritize their need as per their budget and their desire.

Moreover middleman and brokerage frauds will be reduced with customer satisfaction.This project is made with a vision to help both investor and customer to find actual value of the property taking into consideration different features and attributes they are looking for with no loss of capital.

**Limitation**

The linear regression algorithm helps to fulfill customers by increasing the accuracy of estate choice and reducing the risk of investing in an estate.A lot’s of features that could be added to make the system more widely acceptable. One of the major future scopes is adding estate database of more cities which will provide the user to explore more estates and reach an accurate decision. More factors like recession that affect the house prices shall be added. In-depth details of every property will be added to provide ample details of a desired estate. This will help the system to run on a larger level.

**Closing Reflections**

The linear regression algorithm used here helps to fulfill customers by increasing the accuracy of estate choice and reducing the risk of investing in an estate.We have learnt how to clean and filter data which can be used for prediction.The approach for summarizing, visualizing, and becoming intimately familiar with the important characteristics of a data set is vital to create best model.

Model performance and efficiency could have been enhanced if more focus has been given to to Exploratory Data Analysis as it provides the context needed to develop an appropriate model – and interpret the results correctly

Submitted By :-

Chris Mathew

Chinmayee Panda

Laxmipriya Sahoo

Mahammad Parwez Alam

Sidharth Sankar Nayak