House Price Prediction

Revanth Reddy Kontham
student Id: 1107178
Masters of Applied Computer Science
Lakehead University
Thunder Bay, Ontario, canada
email: rkontham@lakeheadu.ca

Akhilesh Kumar Kondoju
student Id: 1107334
Masters of Applied Computer Science
Lakehead University
Thunder Bay, Ontario, canada
email: akondoju@lakeheadu.ca

Rajesh Aytha
student id:1105343
Masters of Applied Computer Science
Lakehead University
Thunder Bay, Ontario, canada
email: raytha@lakeheadu.ca

Abstract:

Contrary to the widespread belief that house prices are dependent on generic factors like several bedrooms and square area of the house, etc. Generally, house buyers neglect this information. As a result, their price estimation is very different from the actual prices according to their area. In this project, we developed a house cost prediction system that is used to predict the house prices by taking different features into considerations like the number of bedrooms, bathrooms, square feet living, etc. For this purpose, we will use different Machine Learning techniques such as linear regression, Polynomial regression, Adaboost Regression, decision Tree and many more, where we predict and compare the accuracies produced by them.

I. Introduction

According to the US Census Bureau, 560,000 houses were sold in the United States in 2016 [1]. Also, 65% of all American families owned houses in 2016 [2]. For the Americans who sold and bought these houses, a good housing price prediction would better prepare them for what to expect before they make one of the most important financial decisions in their lives. A recent report from the Zillow Group, a popular housing database website, indicates that house sellers and buyers are increasingly turning to online research to estimate house prices before contacting real estate agents [4]. Researching how much the house you are interested in is worth it on your own can be difficult for multiple reasons. One particular reason is that there many factors that influence the potential price of a house, making it more complicated for an individual to decide how much a house is worth on their own without external help. This can lead to people making poorly informed decisions about whether to buy or sell their houses and which prices are reasonable. Because houses are long term investments, people must make their decisions with the most accurate information possible. Therefore, housing websites such as Zillow, Trulia and Redfin 1, exist to provide estimations of housing valuations based on the houses characteristics, at no cost. Buying a house is a stressful thing, one has to pay huge sums of money and invest many hours and even then, there is a persisting concern whether its a good deal or not. Buyers are generally not aware of factors that influence house prices. Almost all the houses are described by the total area in square foot, the neighborhood and the number of bedrooms. This creates an illusion that house prices are dependent almost solely on the above-stated factors. Therefore, the houses are overpriced and a buyer should have a better idea of the actual value of the houses. Moreover, many Machine Learning techniques give different price estimations which leads to creating confusion to the buyer, which price to follow. So here in our paper, we will implement different algorithms where the buyer can give requirements of his interests like number of bedrooms, number of bathrooms, sqft-living, zip code, year in which house is built, etc and the accuracies of those algorithms are calculated and compared with each other and best is given to buyer. As a result, buyers can obtain the best price which mainly resolves the problem of estimating the price of the house.

II. Literature Survey

Our main objective is to provide the user with the best suitable house according to his/her wish and desired features by using pattern recognition techniques. The system we choose for our model must be scalable to run against a large database with thousands of data. Summarizing with all references getting output with a certain pattern of functions. Pow states that Real Estate property prices are linked with the economy (Pow, Janulewicz, Liu, 2014). He analyzes and predicts the real estate property prices in Montreal based on 130 features such as geographical location and room numbers[3]. He compares different machine learning methods such as linear regression, Support Vector Regression, knearest Neighbour, and Random Forest. He wraps up that Random Forest outperforms other algorithms (Pow, Janulewicz, Liu, 2014). Pow uses dataset comprises of 130 features from Centris.ca and deProprio.com (Pow, Janulewicz, Liu, 2014). He first pre-processes the data. He determines the outlier by looking at the distribution of values. Then he applies feature engineering on the dataset by mitigating the dimensionality with Principal Component Analysis (PCA).

Parks paper analyzes the housing data on 5359 townhouses in Fairfax County, Virginia based on different machine learning algorithms such as RIPPER (Repeated Incremental Pruning to Produce Error Reduction), Nave Bayes, AdaBoost. He proposes an enhance prediction model to help sellers make their decisions on the house price valuations[4]. He concludes the RIPPER algorithm outperforms other models on predicting house prices (Park Bae, 2015). Kumar tries to find a machine learning approach to predict house prices around Bangalore based on features such as house size and bedroom number. He extracts data from real estate website and analyzes using the dataset with WEKA[5]. Kumar experiments with different machine learning algorithms such as linear regression, Decision Tree, and Nearest Neighbour (Kumar et al., 2015). He concludes that Nave Bayes is consistent for unequal distribution frequency and Decision Tree is the most consistent classifier for equal frequency distributions. Khamis compares the performance of predict house prices between the Multiple Linear Regression model and Neural Network model in New York[6]. The dataset is a sample of randomly chosen 1047 houses with features such as lot size and house ages from the Math10 website. (Kamarudin Khamis, 2014). The experimental results show that the R square value in the Neural Network model is higher than the Multiple Linear Regression model by approximately 27% (Kamarudin Khamis, 2014). Khamis concludes that Neural Network Page 13 of 56 Model has an overall better performance and is preferred over the Multiple Linear Regression model (Kamarudin Khamis, 2014). Bahia collects the data from the UCI Machine Learning Repository. He pre-processes the data, follows by feature selection and transformation. The dataset has 506 samples. Bahia selects 13 variables to use for Artificial Neural Network predictions. He compares the results between Feed Forward Back Propagation Artificial Neural Network with Cascade Forward Back Propagation Neural Network. The input layer is a 13 x 506 matrix, and the output layer is a 1 x 506 matrix of the median value of owner-occupied homes (Bahia, 2013)[7]. He uses mean square error (MSE) from the output in training, validation, and test as the evaluation metrics. He divides the dataset to 80% training and 20% for testing and trains up to 100 epochs. Bahia concludes that Cascade Forward Neural Network outperforms Feed Forward Neural Network because the MSE for Cascade Feedforward Back Propagation is less than Feed Forward Back Propagation neural network (Bahia, 2013). Sirmans, Macpherson, and Zietz (2005) provide a study of 125 papers that use a hedonic pricing model to estimate house prices in the past decade [8].

The paper provides a list of 20 attributes that are frequently used to specify hedonic pricing models. This dataset contains 12 attributes on this list. Moreover, Sirmans, Macpherson, and Zietz (2005) also discuss the effects of some variables on housing prices. For example, the number of bathrooms is usually positively correlated to the final sale price. Out of 40 times appearing in housing price studies, this attribute has a positive effect 34 times and is statistically significant 35 times. On average, keeping other variables unchanged, an increase of 1 bathroom leads to a 10% to 12% increase in the propertys value. Similarly, my paper shows that, based on the dataset of sold houses in five counties, the number of the bathroom has a statistically significant and positive effect on sold price. On average, an increase of 1 bathroom could increase a houses price by 15, 787. The dataset consists of nearly 20,000 samples which can be assisted with machine learning models. we can train our machine models for sentiment analysis and we can also analyze our customer positive reviews and negative reviews[11]. In the literature review, we have seen different results of different algorithms on house price predictions. They all focus on the results of different algorithm's efficiency and accuracy. There have not been comparisons between the performances of feature selections. It is important to know about all the algorithms and their process and the way of functioning the project.

III. Data Description

Dataset was taken from kagle opensource platform. This dataset contains house deal costs for King County, which incorporates Seattle. It contains 21000 samples and the below are some features of the dataset.

*1) **id** :* It is the special numeric number appointed to each house being sold.

*2) **date** :* It is the date on which the house was sold out.

*3) **price** :* It is the cost of the house that we need to anticipate so this is our objective variable and separated from it are our highlights.

*4) **bedrooms** :* It decides the number of rooms in a house.

*5) **bathrooms** :* It decides the number of washrooms in the room of a house.

*6) **sqft_living** :* It is the estimation variable that decides the estimation of the house in square foot.

*7) **sqft_lot** :* It is likewise the estimation variable that decides the square foot of the part.

*8) **floors** :* It decides the all-out floors' methods levels of the house.

*9) **waterfront** :* This feature determines whether a house has a view to waterfront 0 means no 1 means yes.

*10) **view** :* This element decides if a house has been seen or not 0 methods no 1 methods yes.

*11) **condition** :* It decides the general state of a house on a size of 1 to 5.

*12) **grade** :* It decides the general evaluation given to the lodging unit, in light of the King County reviewing framework on a size of 1 to 11.

*13) **sqft_above** :* It decides the area of the house separated from the storm cellar.

*14) **sqft_basement** :* It decides the area of the storm cellar of the house.

*15) **yr_built** :* It decides the date of the structure of the house.

*16) **yr_renovated** :* It decides the time of the remodels of the house.

*17) **zipcode** :* It decides the postal district of the area of the house.

*18) **lat** :* It decides the scope of the area of the house.

*19) **long** :* It decides the longitude of the area of the house.

*20) **sqft_living15** :* Lounge room territory in 2015(implies- - a few redesigns).

*21) **sqft_lot15** :* lotSize region in 2015(implies- - a few redesigns).

IV. METHODOLOGY AND JUSTIFICATION

We have used 4 machine learning algorithms:

A. Linear Regression:

In general, there are two types of supervised machine learning algorithms which are Regression and classification. The Regression algorithms predict continuous value outputs while the classification algorithms predict discrete outputs. For example, predicting the price of a house in dollars is a regression problem whereas predicting whether a tumor is benign in body or not is a classification problem.

Linear Regression Theory: The term linearity in algebra refers to a linear relationship between more than two variables. If we draw a relationship between these two variables in a two-dimensional space, we get a straight line. The main task of Linear regression is to predict a dependent variable(y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x and y, were x is input and y is output. Therefore, the name is Linear Regression. Linear regression gives us a straight line that best fits the data points when we plot the independent variable (x) on the x-axis and dependent variable (y) on the y-axis. We know equation of straight line is:

y=mx+c

Where c is the intercept and m is the slope of the line and is called as regression coefficient. Linear regression algorithm gives us the optimal value for the intercept and the slope. The y and x variables remain the same because they are the features. Depending upon the values of intercept and slope there can be some multiple lines so in that situation linear regression algorithm fits all the lines on data points and returns the line that gives the least error. In the project we have used ScikitLearn library to implement the regression function.

Steps to perform Linear Regression:

At first sight, we obtain the data which consists of various features. So, the major task is to predict the house rate using the features in data set by getting the highest accuracy rate. Then next step is to divide the data into two parts that is attributes and labels. Here, attributes are the independent variables and labels are dependent variables where those values have to be predicted. In addition to it, in the next step we split the data to the training set and test set. After splitting the data into training and testing sets, finally the last process is the training our algorithm. For this, we need to import Linear Regression class from the Scikit-Learn library, instantiate it, and call the fit() method along with our training data. The main use of fit() method is to fit our model to the training data set. In the end, linear regression model results in a line that fits the data and finds the best value for the intercept and slope. In addition to the work, we can evaluate the performance of the algorithm. Which can determine how well our algorithm works on a given particular dataset when compared to the other algorithms. For regression algorithms, three evaluation metrics are commonly used: Mean Absolute Error which is the mean of the absolute value of the errors. Mean Squared Error which is the mean of the squared errors. Root Mean Squared Error which is the square root of the mean of the squared errors We dont have to perform these calculations manually. The Scikit-Learn library comes with pre-built functions which can be used to find out these values [12].

Justification:

The linear regression technique is used to examine mainly the following things. Firstly, it examines a set of predictor variables to do a good job in predicting an outcome of dependent variable and secondly, which variables in particular are significant predictors of the outcome variable. And these linear regression estimates are used to explain the relationship between one dependent variable and one or more independent variables more than other techniques and the regression analysis helps us to understand to what extent the dependent variable changes with a change in independent variables.

B. Decision tree regression:

A Decision tree manufactures relapse or grouping models as a tree structure. It separates a dataset into littler and littler subsets while simultaneously a related choice tree is steadily created. The conclusive outcome is a tree with choice hubs and leaf hubs. A choice hub has at least two branches, each speaking to values for the quality tried.[13] The Leaf hub speaks to a choice on the numerical objective. The highest choice hub in a tree which compares to the best indicator called root hub. Choice trees can deal with both unmitigated and numerical information. Decision tree algorithm: The center calculation for building choice trees called ID3 by J. R. Quinlan which utilizes a top-down, eager pursuit through the space of potential branches with no backtracking. The ID3 calculation can be utilized to build a choice tree for relapse by supplanting Information Gain with Standard Deviation Reduction.

Standard deviation: A decision tree is fabricated top-down from a root hub and includes apportioning the information into subsets that contain occasions with comparative qualities (homogenous). We utilize standard deviation to ascertain the homogeneity of a numerical example. On the off chance that the numerical example is totally homogeneous its standard deviation is zero.

a) The Standard deviation for one trait:

- Standard Deviation (S) is for tree building (fanning).
- The coefficient of Deviation (CV) is utilized to choose when to quit stretching. We can utilize Count (n) also.
- Normal (Avg) is the incentive in the leaf hubs.

b) The standard deviation for two characteristics. Standard Deviation Reduction :

The standard deviation decrease depends on the lessening in standard deviation after a dataset is part on a quality. Developing a choice tree is tied in with finding a characteristic that profits the best quality deviation decrease (i.e., the most homogeneous branches).

Stage 1: The standard deviation of the objective is determined.

Stage 2: The dataset is then part on the various qualities. The standard deviation for each branch is determined. The subsequent standard deviation is subtracted from the standard deviation before the split. The outcome is the standard deviation decrease.

Stage 3: The characteristic with the biggest standard deviation decrease is picked for the choice hub.

Step 4a: The dataset is separated dependent on the estimations of the chose characteristic. This procedure is run recursively on the non-leaf branches until all information is handled.

By and by, we need some end criteria. For instance, when the coefficient of deviation (CV) for a branch decreases than a specific limit or potentially when too barely any occurrences (n) stay in the branch.

Step 4b: ”Cloudy” subset needn't bother with any further parting since its CV is not exactly the edge. The related leaf hub gets the normal of the ”Cloudy” subset.

Step 4c: However, the ”Radiant” branch has a CV more than the limit which needs further parting. We select ”Breezy” as the best hub after ”Standpoint” since it has the biggest SDR. Since the quantity of information focuses for the two branches (FALSE and TRUE) is equivalent or under 3 we stop further fanning and allot the normal of each branch to the related leaf hub.

Step 4d: Moreover, the ”blustery” branch has a CV which is more than the limit.

This branch needs further parting. We select ”Blustery” as the best hub since it has the biggest SDR.[14] Since the quantity of information focuses for every one of the three branches (Cool, Hot and Mild) is equivalent or under 3 we stop further stretching and relegate the normal of each branch to the related leaf hub. At the point when the quantity of cases is more than one at a leaf hub we ascertain the normal as the last an incentive for the objective.

Justification:

Decision tree regression contrasted with different calculations choice trees require less exertion for information arrangement during pre-handling. A choice tree doesn't require a standardization of information. A choice tree doesn't require scaling of information also. Missing esteems in the information additionally doesn't influence the way toward building a choice tree to any significant degree. A Decision tree model is extremely instinctive and simple to disclose to specialized groups just as partners.

C. Random Forest Regression:

Random forest is a Supervised Learning calculation that utilizes gathering learning strategies for classification and regression.

Random forest is a stowing system and not a boosting method. The trees in Random forest are run in parallel. There is no cooperation between these trees while building the trees. It works by developing a large number of choice trees at preparing time and yielding the class that is the method of the classes (classification) or mean expectation (regression) of the individual trees. A random forest is a meta-estimator which totals numerous decision trees, with some supportive adjustments:

1. The number of highlights that can be part of at every hub is restricted to some level of the aggregate (which is known as the hyperparameter). This guarantees the troupe model doesn't depend too intensely on any individual element and utilizes all conceivably prescient highlights.

2. Each tree draws an arbitrary example from the first informational index while creating its parts, including a further component of haphazardness that counteracts overfitting. The Random Forest is one of the best AI models for prescient investigation, making it a mechanical workhorse for AI. The random forest model is a kind of added substance model that settles on forecasts by consolidating choices from a grouping of base models.

Various types of models have various focal points. The random forest model is truly adept at taking care of forbidden information with numerical highlights, or all-out highlights with less than many classifications. In contrast to direct models, a random forest can catch a non-straight association between the highlights and the objective.

One significant note is that tree-based models are not intended to work with scanty highlights. When managing meager info information (for example all out highlights with huge measurement), we can either pre-process the inadequate highlights to produce numerical insights or change to a straight model, which is more qualified for such situations.

Justification:

It runs productively on huge databases. It can deal with a large number of information factors without variable cancellation. It gives appraisals of what factors are significant in the grouping. It creates an inside fair gauge of the speculation mistake as the woods building advances. It has a viable strategy for evaluating missing information and keeps up precision when a huge extent of the information is absent.

D. AdaBoost Regressor:

The main idea behind the boosting methods is to train predictors sequentially, each trying to correct its predecessor. The two most popular and commonly used boosting algorithms are AdaBoost and Gradient Boosting. In this project, we have used the AdaBoost algorithm only. AdaBoost can be used as both a classifier and a regressor.

The following are the steps that are included in the algorithm:

The center guideline of AdaBoost is to fit an arrangement of frail students (i.e., models that are just somewhat superior to arbitrary guessings, for example, little choice trees) on over and again adjusted adaptations of the information

- The expectations from every one of them are then consolidated through a weighted greater part vote (or whole) to create the last forecast

- The information changes at each purported boosting emphasis comprise of applying loads w_1, w_2, ..., w_N to every one of the preparation tests

- At first, those loads are good to go to w_i = 1/N, so the initial step basically prepares a feeble student on the first information
- For each progressive cycle, the example loads are independently changed and the learning calculation is reapplied to the reweighted information
- At a given advance, those preparation models that were mistakenly anticipated by the helped model instigated at the past advance have their loads expanded, though the loads are diminished for those that were anticipated accurately

As emphasized continue, models that are hard to anticipate get regularly expanding the impact. Each ensuing frail student is along these lines compelled to focus on the models that are missed by the past ones in the arrangement.

The AdaBoost model predicts the example by having each tree in the woodland. At that point, we split the trees into bunches as indicated by their choices. The last characterization is made by the woods all in all and is chosen by the gathering with the biggest total.

In executing the code, at whatever point we are working with the all-out component, we should encode it as numbers. We split our information into preparing and test sets to assess our model's exhibition.

Justification: The principle thought behind the boosting techniques is to prepare indicators successively, each attempting to address its antecedent. The center standard of AdaBoost is to fit a succession of powerless students (i.e., models that are just marginally superior to arbitrary speculating, for example, little choice trees) on more than once changed forms of the information. The forecasts from every one of them are then joined through a weighted larger part vote (or aggregate) to deliver the last expectation.

E. Final Thoughts:

Like Random Forest, AdaBoost settles on forecasts by applying numerous choice trees to each example and consolidating the expectations made by singular trees. In the AdaBoost calculation, every choice tree contributes some value to the last forecast. To a degree, AdaBoost is like Random Forest in light of the fact that the two systems confirm the expectations settled on by every choice tree to choose the last.

V. Functioning process

Whenever this model is executed an input window occurs on the screen where we can give input values. As soon as the input values are given it performs four methodologies like Linear Regression, Decision Tree Regression, Random Forest Regression, AdaBoost Regressor.

In Linear Regression, Firstly, it examines a set of predictor variables to do a good job in predicting an outcome or dependent variable and secondly, which variables, in particular, are significant predictors of the outcome variable. And these linear regression estimates are used to explain the relationship between one dependent variable and one or more independent variables more than other techniques and the regression analysis helps us to understand to what extent the dependent variable changes with a change in independent variables.

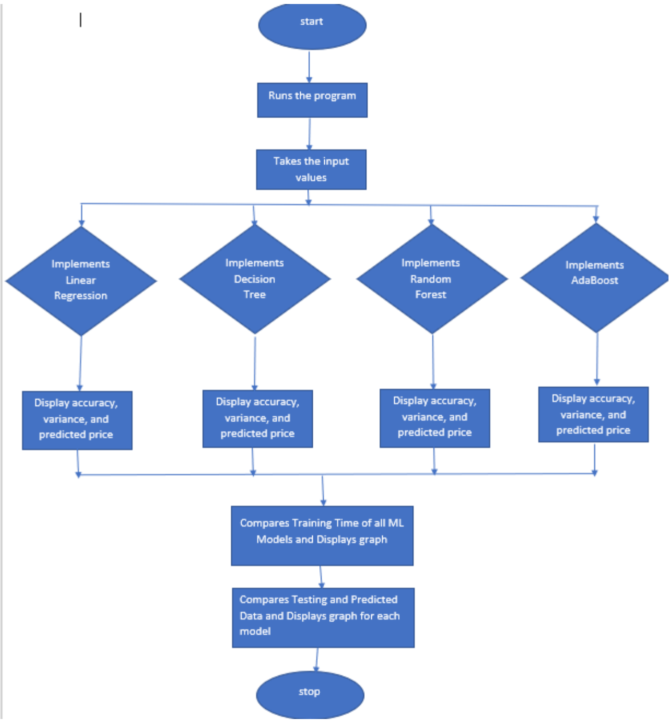

start
Runs the program
Takes the input values
Implements Linear Regression
Implements Decision Tree
Implements Random Forest
Implements AdaBoost
Display accuracy, variance, and predicted price
Display accuracy, variance, and predicted price
Display accuracy, variance, and predicted price
Display accuracy, variance, and predicted price
Compares Training Time of all ML Models and Displays graph
Compares Testing and Predicted Data and Displays graph for each model
stop

Fig. 1. Block Diagram of this Model

In Decision Tree Regression, Firstly, based on the given input value the standard deviation of the object is determined. Secondly, the dataset is then part of the various qualities. The standard deviation for each branch is determined. The subsequent standard deviation is subtracted from the standard deviation before the split. The outcome is the standard deviation decrease. Thirdly, The characteristic with the biggest standard deviation decrease is picked for the choice hub. Finally, the dataset is separated dependent on the estimations of the chose characteristic. The related leaf hub gets the normal of the ”Cloudy” subset. Based on the leaf hub a certain price is estimated.

In Random Forest Regression, It runs productively on huge databases. It can deal with a large number of information factors without variable cancellation. It gives appraisals of what factors are significant in the grouping. It creates an inside fair gauge of the speculation mistake as the woods building advances. It has a viable strategy for evaluating missing information and keeps up precision when a huge extent of the information is absent. Based on the process certain price is estimated.

In AdaBoost Regressor, The expectations from every one of them are then consolidated through a weighted greater part vote to create the last forecast. The information changes at each purported boosting emphasis comprise of applying loads to every one of the preparation tests. At first, those loads are good, so the initial step prepares a feeble student on the first information. For each progressive cycle, the example loads are independently changed and the learning calculation is reapplied to the reweighted information. At a given advance, those preparation models that were mistakenly anticipated by the helped model instigated at the past advance have their loads expanded, though the loads are diminished for those that were anticipated accurately.

After these processes, it compares all the training times of these ML models and based on its testing and predicted data it displays certain graphs. Comparing these graphs, the price will be predicted for four methodologies that are used which will be displayed on the output window.

VI. Results and Discussions

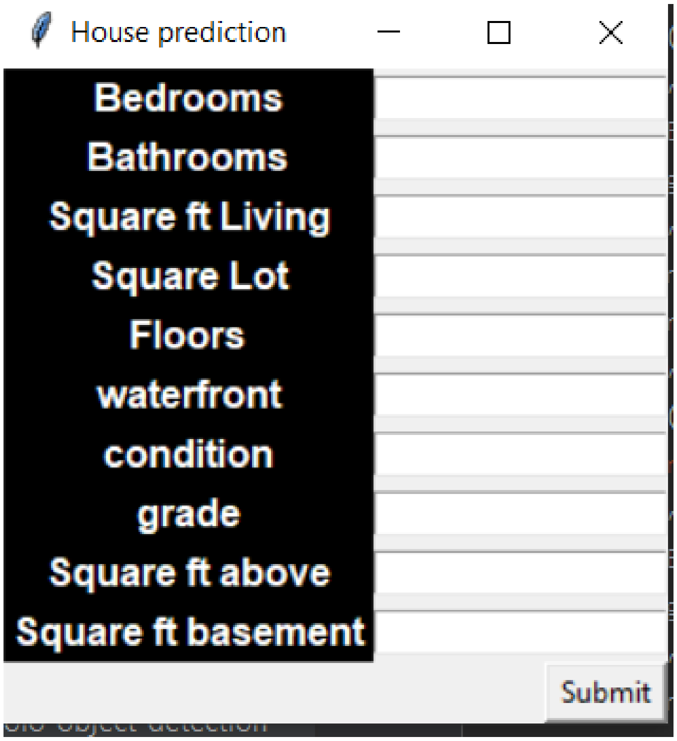
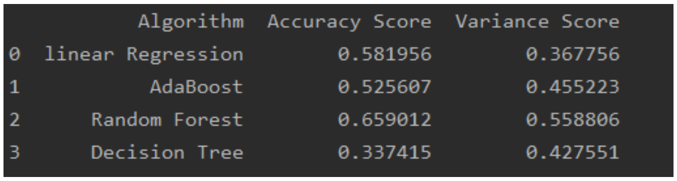


Fig. 2. Input Window

In this window, user is allowed to enter values and click on submit button.



```
             Algorithm  Accuracy Score  Variance Score
0  linear Regression        0.581956        0.367756
1           AdaBoost        0.525607        0.455223
2      Random Forest        0.659012        0.558806
3      Decision Tree        0.337415        0.427551
```

Fig. 3. Accuracy and Variance

This window displays accuracy and variance score of all algorithms which are used.

From the below fig 4 it is derived that decision tree has set aside immaterial measure of effort to prepare though Random forest has taken most extreme time and it is yet evident on the grounds that as we increment the quantity of tree 400 for this situation preparing time will increment so we should pay special mind to ideal model which has more noteworthy exactness and less preparing time in contrast with other. Along these lines, for this situation, despite the fact that it is requiring some investment Random forest is best appropriate for the model.

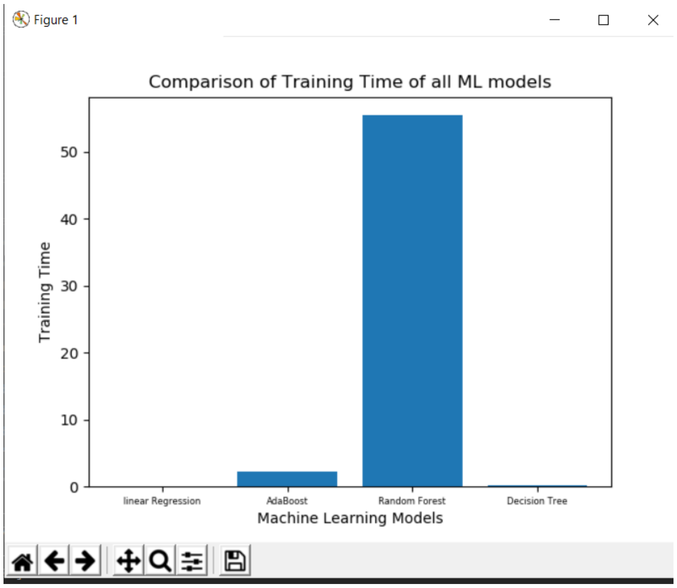
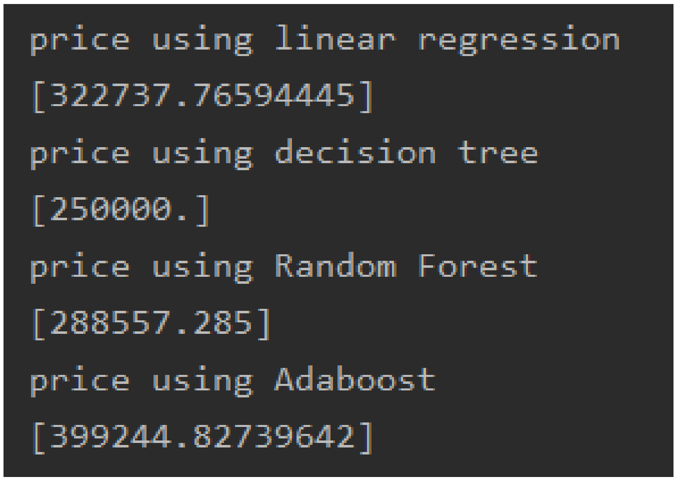


Fig. 4. Time complexity of Algorithms

In this window, predicted price of all algorithms are displayed.



```
price using linear regression
[322737.76594445]
price using decision tree
[250000.]
price using Random Forest
[288557.285]
price using Adaboost
[399244.82739642]
```

Fig. 5. Predicted prices

Comparing all the 21000 samples of the dataset graphs are formed. As we are using a huge dataset some points on the graph are overlapped. The following fig 5, fig 6, fig 7 and fig 8 are the graph forms of algorithms used.

Figure 5 is about Linear Regression.

Figure 6 is about Decision Tree Regression.

Figure 7 is about Random Forest Regression.

Figure 8 is about AdaBoost Regressor.

Comparison between Testing data and Predicted data.

In this graph:
Red dots are represented as Testing data.
Green dots are represented as Predicted data.

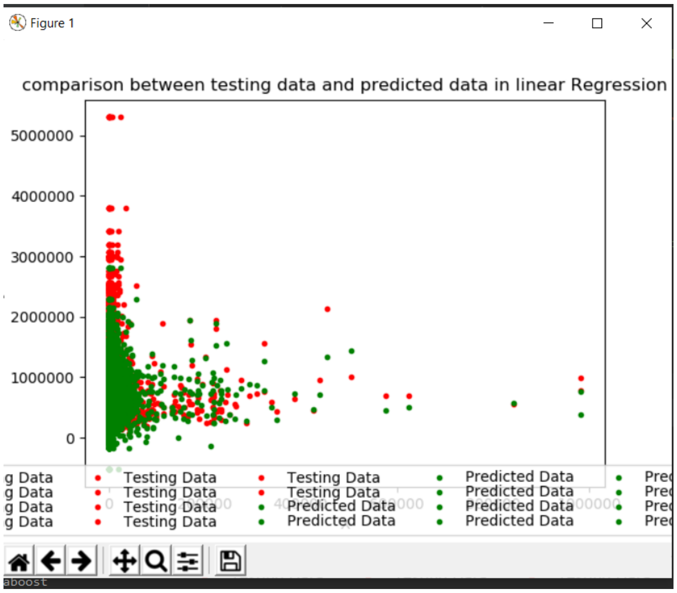


Fig. 6. Linear Regression graph

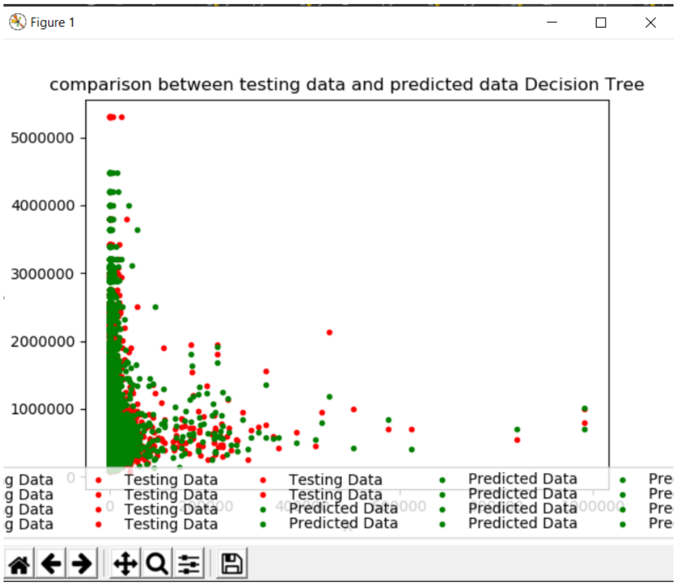


Fig. 7. Decision Tree Regression graph

VII. Conclusion

In the project, we are comparing the above-mentioned regression techniques and trying to predict the best house price for the user. Analyzing all the algorithms with there accuracy and variance scores random forest shows balance scores compared to other algorithms. Even though Linear Regression has high accuracy compared to a random forest but couldn't show balance in accuracy and variance, the Random forest Algorithm is termed as the best algorithm compared to those three algorithms.

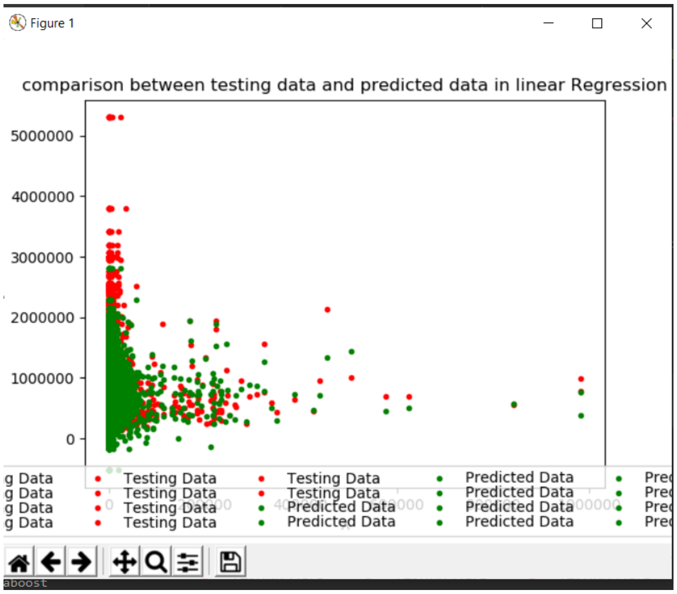


Fig. 8. Random Forest Regression graph

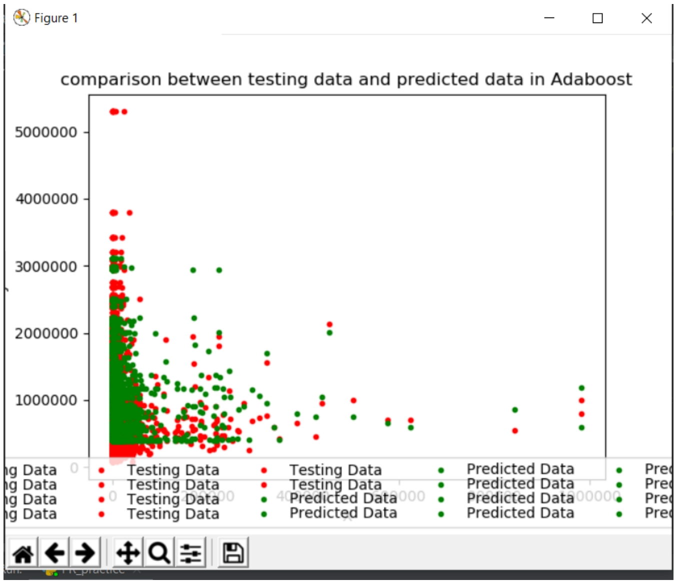


Fig. 9. AdaBoost Regressor graph

VIII. Future Work

Later on, various AI models, for example, XGBoost can be utilized to do the investigation. Additionally, since the quantities of highlights are little, more element building, for example, includes total, should be possible later on. Furthermore, an examination between polynomial relapse and direct relapse should be possible for improving exactness.

IX. References

[1] Number of houses sold in the United States from 1995 to 2016. www.statista.com.

[2] Quick Facts: Resident Demographics. National Multifamily Housing Council. Accessed: 11/11/2017. 2017.

[3] Pow, Janulewicz and Liu ”Applied Machine Learning Project 4 Prediction of real estate property prices in Montreal”, 2014.

[4] Park and Bae ”Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data”, 2015.

[5] Kumar et al, ”Cultural similarities and housing market linkage: evidence from OECD countries”, 2015.

[6] Kamarudin Khamis, ”Comparative Study On Estimate House Price Using Statistical And Neural Network Model”, 2014.

[7] Bahia, ”Housing Price prediction Using Support Vector Regression”, 2013.

[8] G. Stacy Sirmans, David A. Macpherson, and Emily N. Zietz. The Composition of Hedonic Pricing Models. In: Journal of Real Estate Literature 13.1 (2005).

[9] Sherwin Rosen. Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. In: The Journal of Political Economy 82.1 (1974).

[10] Hasan Selim. Determinants of house prices in Turkey: Hedonic regression versus artificial neural network. In: Expert Systems with Applications 36 (2009).

[11] Manu siddhartha, ”House price prediction”, kaggle datasets, 2016.