DOI: http://dx.doi.org/10.26483/ijarcs.v11i6.6668

Volume 11, No. 6, November-December 2020

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

THE STUDY USING ENSEMBLE LEARNING FOR RECOMMENDING BETTER FUTURE INVESTMENTS

Kajal Bholashankar Jaiswal
Master of Computer Engineering
Thakur College of Engineering and Technology
Mumbai, India

Dr. Harshali Patil
Associate Professor, Computer Engineering
Thakur College of Engineering and Technology
Mumbai, India

Abstract: Generally, House estimation record addresses the summarized esteem changes of private housing. While at a single-family house cost desire, it needs more exact procedure reliant on the spot, house type, size, structure year, close by improvements, and some various parts which could impact house demand and deftly. With limited dataset and data incorporates, a sensible and composite data pre-taking care of, creative component planning methodology is assessed in this paper. People are careful when they are endeavouring to buy another house with their money related plans and market strategies. The objective of the paper is to measure the sensible house costs for non-house holders reliant on their financial courses of action and their desires. By analysing the earlier item, entry ranges and besides alerts enhancements, guessed costs will be evaluated. The paper includes expectations utilizing diverse Regression procedures like Ridge, LASSO, Random Forest, SVM (support-vector machine), KNN (k-nearest neighbours), Ada Boost Regression, Stacking (decision tree, lasso and random forest), Decision Tree. House estimation figure on an instructive file has been done by using all the recently referenced systems to find the best among them. The reason of this paper is to help the vendor with assessing the selling cost of a house perfectly and to assist people with foreseeing the time slap to store up a house. A part of the related segments that influence the cost were furthermore taken into examinations, for instance, states of being, thought, area and territory, etc.

Keywords: Regression, investments, house price, estate agent, property, stacking

I. INTRODUCTION

As we probably are aware 'property' has gotten perhaps the most splendid thing with regards to speculation. I remember, when I got my first job in Airoli, it was exceedingly difficult for me to travel all the way from Western line to Harbor Line, thus changing 3 trains. At long last I chose to move to Airoli close by my office area. As that part of Mumbai is newly developed, there were hardly 2 agents available in the entire Sectors of Airoli. When asked about the rents, the prices for paying guest was sky high. Although I eventually dropped the idea of shifting out there. But later I came to know that the agents there used to double up the rent which was terrifying. This venture can help individuals who are purchasing or leasing a level by knowing the right and rough cost, specifically zone.

As demonstrated by the 2017 type of Upgoing Trends in property Asia Pacific, Mumbai and Bangalore are the foremost significant level metropolitan networks for hypothesis and advancement. These cities have supplanted Tokyo and Sydney. The house costs of twenty-two urban communities out of 26 dropped within the quarter from April to June as compared to the quarter January to March per National Housing Bank's Reside (residential index). With the presentation of realty RERA (Regulation Development Act) and BP (Benami Property) Act during the state India, a more prominent number of speculators are pulled in to put into land in India. An attractive investment that are made in India, have made the Indian economy strong and modern. Notwithstanding, past downturns show that realty costs cannot really develop. Costs of the significant bequest property are identified with the monetary states of the state [1]. Regardless of this, we are not having legitimate normalized approaches to quantify the significant home property estimations.

Overall, the property assessments rise with respect to time and its assessed regard ought to be resolved. This assessed regard is required during the proposal of property or while applying for the development and for the appeal of the property. These evaluated qualities are controlled by the expert appraisers. Nonetheless, downside of this training is that these appraisers could be uneven due to give interests from purchasers, dealers, or home loans. In this manner, we require a motorized desire model that can help with anticipating the property assessments with no inclination. This robotized model can help the first run through purchasers and less experienced clients to comprehend whether the property rates are misrepresented or underestimated. Presently, Property costs rely upon different boundaries in the economy and society. House costs are unequivocally subject to the size of the house and its geological area according to the past investigation [2], [3]. We have likewise viewed as different natural boundaries, (for example, number of rooms, living region, stopping, utilities and development material) and furthermore outside boundaries, (for example, area, nearness, forthcoming activities, and so on.) [4], [5]. At that point we have applied these boundary esteems to two diverse AI calculations.

This article suggests alongside latest Forecast on Research desires thinking about examples to moreover plan their budgetary issues. The guideline motivation of the adventure FORECASTING VARIATIONS ON HOUSE. Value was to make the best desire for house costs by using

legitimate computations and finding which among them is best fitting for foreseeing the expense with low error rate. There is a since quite a while ago run among people for buying and selling of house, which is an interesting issue. This issue licenses us, as house estimation specialists, to get acquainted with the housing industry area and helps with making more instructed decisions. The assessment that were done in this paper is essentially established on the datasets of California, United States. considering unexpected changes in cost of houses.

In this paper, Regression strategies which are reasonable to our concern, we attempt to exhibit all the conceivable. The short diagram of all the reference taken are as per the following:

RR [Ridge] and LR [LASSO] Regressions are utilized in which Ridge relapse regularizes the [rc] relapse coefficient by representing an enthusiasm on the size.

LR [LASSO] Regression is additionally same to Ridge however with a little distinction, it utilizes the L1 penalty. In the grouping calculation Naive Bayes is utilized., GBR [Gradient Boosting Regression] is utilized as most favorable Regression technique. In the ANN [Artificial Neural Network] hypothesis is utilized. Gluttonous Pricing hypothesis is utilized in which expect the property estimation as the whole of its quality qualities. DR [Direct Regression] model is utilized in. In GWR [Geographically Weighted Regression] is used which licenses neighborhood assortments in rate. In Bayesian LR [Linear Regression] is used. The DFD [Data Flow Diagram] delineates the request for steps i.e., the stream in which the examination had completed. We attempt to display expectations by performing preparing and parting of information and foresee it utilizing different AI procedures like various types of Regression. We have broken down every relapse strategy and determined its score.

Presently we actualize strategies, for example, SVM [Support - Vector Machine], RR [Ridge Regression], [Lasso Regression] LR, DT [Decision Tree], AR [Ada-boost Regression] and RF [Random Forest] utilizing instruments like python programming, Ipython jupyter Notebook Graph Lab, Sframes. This work is actualized utilizing Python IDLE, Below Figure 1 is utilized here to speak to the progression of information and its handling associated with various relapse procedures.

II. LITERATURE REVIEW

In most recent twenty years imagining the property evaluation has become a basic field. Rise in the enthusiasm for property and fanciful direct of economy power pros to find a way that foresee the land costs without any inclinations. As such, it is a test for scientists to find all the second factors that can impact the cost of property and make a perceptive model by considering all the components.

Building a prescient model for land value valuation requires a comprehensive data with respect to the issue. Various investigators have gone after this issue and passed on their assessment work. This examination work is enlivened from [6] by far most of authors. The creator has scratched the lodging informational collection from Centris.ca and duProprio.com. Their dataset comprises of around 25,000 models and 130 components. Around 70 highlights were scratched from the above sites and land organizations, for example, Century 21, RE/MAX, and Sutton, and so forth. Other 60 highlights were sociodemographic dependent on where the property is found. Afterward, creator executed Principal Component Analysis to diminish the dimensionality. The maker used four backslide systems to anticipate the worth assessment of the property. LR (Linear Regression, Support Vector Machine, KNN (K Nearest Neighbors) and RF (Random Forest Regression) and a gathering approach by merging KNN and RF technique are the four strategies been utilized. The gathering approach foreseen the expenses with least mix-up of 0.0985. In any case, applying PCA did not improve the figure botch. A lot of assessment has been done on ANN (Artificial Neural Networks). This has helped different agents centering a ground issue to deal with utilizing neural structures. In [7], the maker has contemplated liberal worth model and ANN model that predict the house costs. Any ware that are subject to intramural qualities just as outside attributes, Hedonic value models are essentially used to ascertain their cost. The luxurious model fundamentally fuses backslide framework that thinks about different limits, for instance, zone of the property, age, number of rooms, and so on. The Neural Network is set up from the outset, and the heaps and inclinations of the edges and center points independently are using experimentation technique. Discovery technique is only preparing of Neural Network Model.

Notwithstanding, the R-Squared an incentive for Neural Network model was more noteworthy contrasted with indulgent model and the RMSE (Root mean square mistake) estimation of Neural Network model was generally lower. Consequently, it is reasoned that ANN (Artificial Neural Network) performs superior to Hedonic model. Two or three specialists like that in [8] have utilized classifiers to anticipate the property appraisals. The writer in research article [8] has collected the information from Multiple Listing Service (MLS), recorded home credits rates and government financed school assessments. The creator utilized MRIS [Metropolitan Regional Information Systems] informational collection for Land. Nearly around 15,000 records were eliminated by the maker from these three sources which included 76 components. Thus, t-test was used to pick 49 factors as a starter screening. Their examination question was to choose if the end cost was lofty or beneath the posting cost [8]. As needs be, to address this order issue, the creator utilized four AI. C4.5, RIPPER, Naive Bayesian, and Ada-Boost are the four computations used by creator.

Regardless, they found that RIPPER beats other house figure models. The disadvantage is that presentation assessment depends just on classifiers. Execution correlation of other AI calculations ought to likewise be thought of. In article [9], the writers have anticipated the financial exchange costs utilizing direct relapse method. They have gathered financial exchange information from TCS stock Database. The creator has additionally utilized RBF and polynomial relapse strategy alongside straight relapse and found that last is better than the rest of the strategies. In [11], the creator has considered the most macroeconomic boundaries that influence the house costs variety. BPN (Back engendering neural organization) is utilized by creator in this and RBF (spiral premise work neural organization) to set up the non-straight model for land's value variety forecast. The dataset has been taken from Taipei, Taiwan dependent on driving and synchronous monetary records. The creator has thought about 11 boundaries. The expectation results got from them

are contrasted with public Cathay House Price Index or the Sinyi Home Price Index.

The two-mistake measurements utilized were MAE and RMSE. At the point when the forecast outcomes were contrasted with Cathay House Price Index, RBF NN (Neural Network) indicated preferable expectation results over BPN NN. Additionally, for Sinyi Home Price Index BPN NN demonstrated preferable forecast results over RBF NN. Some examination articles portray the through and through techniques and philosophy to assemble the land data and their pre-handling methodology. The writer in article [12] portrays programming that is used in land esteem value assessment. The item assesses diverse land laborers and pages of land associations and records their present interfaces with land purchase or rental into their item data base. He has amassed data from Czech Republic. The data is collected every month to record the progressions occurring in land. The product aggregates 110,000 sections reliably consistently. These passages join different writings, ads, and pictures of the property. The creator has accumulated data from the year 2007 to 2015.This unstructured data that is assembled is exchanged into an organized structure.

Assorted property types have different limits. Hence, it makes the informational collection more lucid. This informational index is then assessed. As such it makes the educational file more fathomable. This enlightening list is then surveyed. New entries made each month are stood out from the more settled segments and checked for their zenith. In the last period of the product, this accessible clean informational index is then assessed and creates different representations agreeing co to the necessity of the client. Thusly, the yield gained may be used as explanation behind legitimate hypotheses or conceivably dwelling decisions for both typical individuals and associations. A couple of researchers have focused in on incorporate assurance and feature extraction strategy. The essayist in article [14] uses an open source instructive assortment of the housing bargains in King County, USA. There are around 20 informative variables. Different component choice and extraction calculations joined with SVR has been utilized by creator in this. The creator has gathered roughly 21,000 perceptions in a timeframe of one year.

The paper shows different information investigation performed on the informational collection. Highlight Selection is the path toward picking a subset of factors from a given arrangement of boundaries either dependent on their significance or their recurrence. Nonetheless, highlight extraction is the route toward lessening the range of the data. Beginning game plan of data is changed into decided characteristics which are comparably helpful and non-overabundance. The four component determination calculations utilized are RFE (Recursive Feature Elimination), Lasso and Ridge and RF (Random Forest) Selector and the mean from every calculation is figured. Utilizing highlight choice, the creator chooses fifteen highlights out of twenty. The calculation for highlight extraction utilized is PCA (Principal Component Analysis) and the boundaries are decreased from twenty to sixteen. Both the strategies work similarly well with the R squared estimation of 0.86, is closed by the creator.

III. METHODOLOGY

A short pointer that can help us understand the flow of project. Below steps wise procedure that is been carried out while implementing the project.

Step 1 – Collect the data from testing and training file based on used parameters

Step 2 – Find out the parameters that have null values

Step 3 – Remove the parameters with null value so that mean (average) is not affected

Step 4 – Show the cleaned data with the help of a chart

Step 5 – Plot a co-related chart to show the correlation between parameters used

Step 6 – Apply Algorithms one by one and plot charts to show R2 score. Compare the results

A. Data Collection

The dataset utilized in this undertaking was from Kaggle Inc [21] an open source site. It includes 3000 records with 80 limits that get the opportunity of impacting the property costs. Anyway, out of these 80 limits only 37 were picked which will without a doubt impact the housing costs. Limits, for instance, Area in square meters, Overall quality which rates the overall condition and finishing of the house, Location, Year in which house was collected, Numbers of Bedrooms and washrooms, Garage zone and number of vehicles that can fit in parking space, pool an area, selling year of the house and Price at which house is sold. Selling cost is a penniless variable on a couple of other free factors. A couple of limits had numerical characteristics, and some were assessments. These examinations were changed over to numerical characteristics. Following Table 1 address a short portrayal about most huge limits that impact the selling cost of the house.

Parameters	Description	Data Type
LotArea	Lot size in square feet	Integer
OverallQual	Rates the Overall material and finish of the house	Integer
OverallCond	Rates the overall condition of the house	Integer
YearBuilt	Original Construction Date	Integer
YearRemodAdd	Remodel date (same as construction date if no remodeling or additior	Integer
TotalBsmtSF	Total square feet of basement area	Integer
1stFlrSF	First Floor square feet	Integer
GrLivArea	Above grade (ground) living area square feet	Integer
FullBath	Basement full bathrooms	Integer
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)	Integer
GarageCars	Size of garage in car capacity	Integer
GarageArea	Size of garage in square feet	Integer
YrSold	Year Sold (YYYY)	Integer
SalePrice	House sellng price	Integer

Table 1- Parameters Used

B. Data Pre-processing

It is a pattern of changing the unrefined, complex data into intentional reasonable data. It incorporates the path toward finding missing and overabundance data in the dataset. Entire dataset is checked for NaN and whichever

recognition involves NaN will be deleted. Consequently, this gets consistency the dataset. In any case, in our dataset, there was no missing characteristics found inferring that each record was built up its contrasting feature regards. Data Pre-planning is a procedure that is used to change over the rough data into a perfect enlightening record. Continuously end, at whatever point the information is collected from various sources it is amassed in harsh arrangement which is not doable for the appraisal.

Need of Data Pre-processing

- For accomplishing better outcomes from the applied model in Machine Learning ventures the arrangement of the information must be in a legitimate way. Some predetermined Machine Learning model needs data in a predefined design, for instance, Random Forest calculation doesn't uphold invalid qualities, subsequently, to execute arbitrary backwoods calculation invalid qualities must be overseen from the first crude informational collection.

- Another angle is that informational index ought to be designed so that more than one Machine Learning and Deep Learning estimations are executed in one educational assortment, and best out of them is picked

C. Feature Engineering

AI fits mathematical documentations to the data to construe a couple of pieces of information. The models acknowledge features as information. A segment is regularly a numeric depiction of a piece of authentic miracles or data. Just the course there are stalemates in a maze, the method of data is stacked up with upheaval and missing pieces. Our action as a Data Scientist is to find a clear a path to a definitive goal of encounters.

Mathematical plans go after numerical sums, and unrefined data isn't really numerical. Feature Engineering is the strategy for eliminating features from data and transforming them into plans that are fitting for Machine Learning counts.

It is divided into 3 broad categories: -

Feature Selection: All highlights are not equivalent. It is tied in with choosing a little subset of highlights from a huge pool of highlights. We select those characteristics which best clarify the relationship of an autonomous variable with the objective variable. There are sure highlights which have priority higher than different highlights to the exactness of the model. It is quite different as dimensionality decrease on the grounds that the dimensionality decrease technique does as such by consolidating existing credits, while the element choice strategy incorporates or prohibits those highlights. The techniques for Feature Selection are Chi-squared test, connection coefficient scores, LASSO, Ridge regression and so forth.

Feature Transformation: It gathers changing our exceptional part to the segments of unique highlights. Scaling, discretization, binning and filling missing information respects are the most extensively seen kinds of information change. To decrease skewness from right of the data, we use log.

Feature Extraction: Exactly when the data to be dealt with through a computation is exorbitantly huge, it's regularly seen as monotonous. Examination with a huge number of elements uses a huge amount of count power and memory, subsequently we should diminish the dimensionality of such factors. It is a term for building mixes of the segments. For even information, we use PCA to reduce highlights. For picture, we can utilize line or edge affirmation. Highlight extraction fuses lessening the measure of points of interest expected to portray a monstrous strategy of information. Feature extraction incorporates reducing the amount of advantages expected to depict a colossal plan of data. When performing assessment of complex data one of the difficult issues originates from the number of variables included. Evaluation with unlimited factors commonly needs an arrangement extraordinary of memory and assessment power, moreover it might make a solicitation calculation overfit to preparing tests and sum up inadequately to new models. Feature extraction is a general term for systems for creating blends of the variables to get around these issues while so far depicting the data with satisfactory exactness. Numerous AI experts acknowledge that properly progressed component extraction is the best approach to suitable model turn of events.

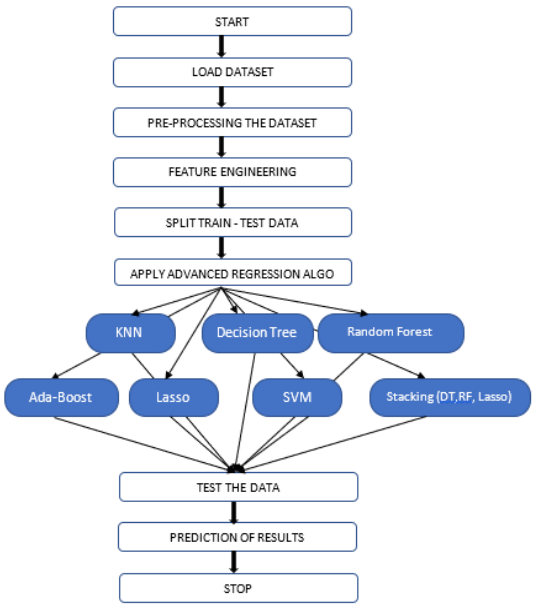

Figure 1- Block Diagram

D. ALGORITHM USED

i. KNN Algorithm

KNN can be utilized for both grouping and regression reasonable issues. In any case, it is much more completely

utilized all together issues in the business. To evaluate any strategy, we generally look at 3 imperative perspectives:

- Ease to decipher yield
- Computation time
- Predictive Power

KNN calculation fairs over all boundaries of contemplations. It is usually utilized for its simple of translation and low count time. The main burden of KNN calculation is that the expectation time is extremely high as it finds the separation between each information point.

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Euclidean Distance Formula

```
kn = KNeighborsRegressor()
kn.fit(X_train,y_train)
print('Train set')
pred=kn.predict(X_train)
print('KNeighborsRegressor Mean Squared error :{}'.format(mean_squared_error(y_train,pred,squared=False)))
print('KNeighborsRegressor r2_score :{}'.format(r2_score(y_train,pred)))
print('Tesst set')
pred=kn.predict(X_test)
mae=mean_absolute_error(y_test,pred)
print('KNeighborsRegressor Mean Absolute error :',mae)
print('KNeighborsRegressor Mean Squared error :{}'.format(mean_squared_error(y_test,pred,squared=False)))
print('KNeighborsRegressor r2_score :{}'.format(r2_score(y_test,pred)))

plt.scatter(y_test,pred,color='gray')
plt.xlabel('Actual values')
plt.ylabel('Predicted values')
plt.plot(np.unique(y_test), np.polyld(np.polyfit(y_test, pred, 1))(np.unique(y_test)))
plt.text(0.6, 0.5, 'R-squared = %0.2f' % r2_score(y_test,pred))
plt.show()
df = pd.DataFrame({'Actual': y_test, 'Predicted': pred})
df1 = df.head(30)
df1.plot(kind='bar',figsize=(10,8))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
visualizer = ResidualsPlot(kn)
visualizer.fit(X_train, y_train)
visualizer.score(X_test, y_test)
visualizer.poof()
```

Figure 2 - KNN Algorithm

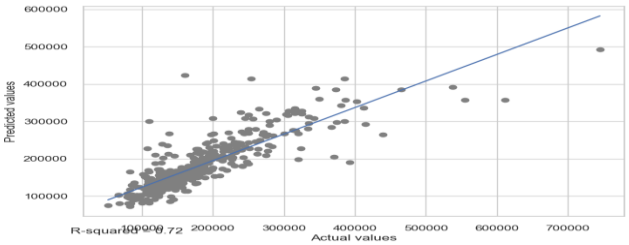

Figure 3 - KNN R-Squared graph

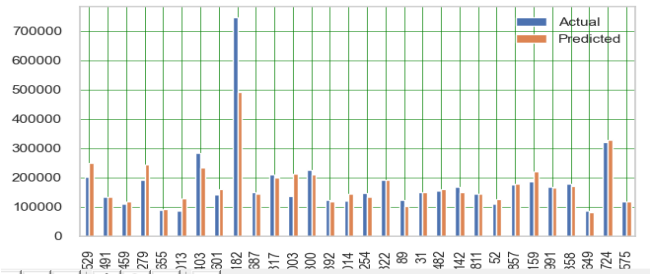

Figure 4 - KNN Actual vs Predicted Chart

The graphical representation of all the different regression techniques listed above are clearly represented below using Python IDLE.

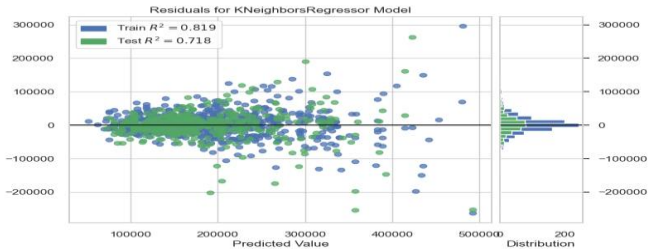

Figure 5 - KNN Test and Train Residuals

ii. Decision Tree Algorithm

Decision Tree calculation has a place with the group of regulated learning calculations. In contrast to other supervised learning computation, the decision tree computation can be utilized for tackling regression and arrangement issues as well. The target of using a DT (Decision Tree) is to make a group model that can use to envision the class or evaluation of the target variable by taking in direct choice standards amassed from before information (getting ready data). In Decision Trees, for anticipating a mean for a record, we get ready to start from the establishment of the tree. We consider the appraisals of the root property with the record's brand name. Considering relationship, we follow the branch identifying with that worth and ricochet to the assessments of the root property with the record's trademark. Based on correlation, we follow the branch relating to that worth and bounce to the following hub. Choice trees order the models by arranging them down the tree from the root to some leaf/terminal hub, with the leaf/terminal hub (node) giving the arrangement of the model.

Each hub(node) in the tree goes about as an investigation for some attribute, and each edge sliding from the centre analyses to the expected reactions to the test. This cycle is repetitive and is repeated for each subtree set up at the new hub node

```
def DT(X_train,y_train,X_test,y_test):
  dt = DecisionTreeRegressor()
  dt.fit(X_train,y_train)
  print('Train set')
  pred=dt.predict(X_train)
  print('DecisionTree Regressor Mean Squared error :{}'.format(mean_squared_error(y_train,pred,squared=False)))
  print('DecisionTree Regressor r2_score :{}'.format(r2_score(y_train,pred)))
  print('Tesst set')
  pred=dt.predict(X_test)
  mae=mean_absolute_error(y_test,pred)
  print('DecisionTree Regressor Mean Absolute error :',mae)
  print('DecisionTree Regressor Mean Squared error :{}'.format(mean_squared_error(y_test,pred,squared=False)))
  print('DecisionTree Regressor r2_score :{}'.format(r2_score(y_test,pred)))

  plt.scatter(y_test,pred,color='gray')
  plt.xlabel('Actual values')
  plt.ylabel('Predicted values')
  plt.plot(np.unique(y_test), np.polyld(np.polyfit(y_test, pred, 1))(np.unique(y_test)))
  plt.text(0.6, 0.5, 'R-squared = %0.2f' % r2_score(y_test,pred))
  plt.show()
  df = pd.DataFrame({'Actual': y_test, 'Predicted': pred})
  df1 = df.head(30)
  df1.plot(kind='bar',figsize=(10,8))
  plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
  plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
  plt.show()
  visualizer = ResidualsPlot(dt)
  visualizer.fit(X_train, y_train)
  visualizer.score(X_test, y_test)
  visualizer.poof()

DT(X_train[top_feature],y_train,X_test[top_feature],y_test)
```

Figure 6 - Decision Tree Algorithm

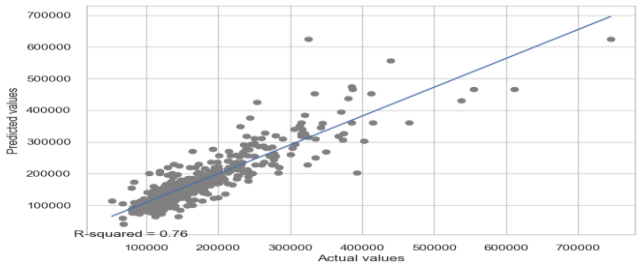

Figure 7 - Decision Tree R-Squared graph

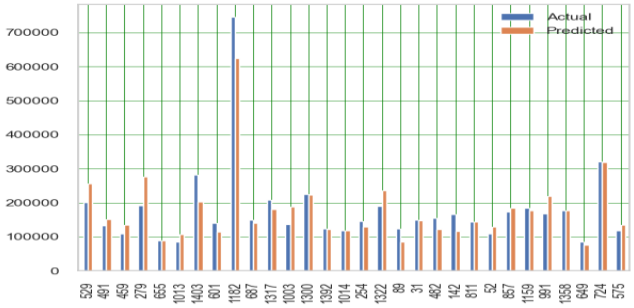

Figure 8 - Decision Tree Actual vs Predicted Chart

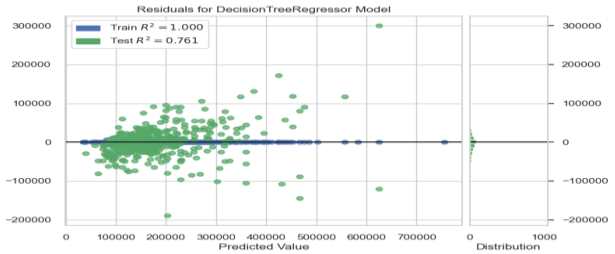

Figure 9 - DT Test and Train Residuals

iii. Support Vector Machine Algorithm

"Support Vector Machine" (SVM) is a Machine Learning Technology that can be used for the two i.e. plan (Classification) and Relapsing (Regression) problems or issues. SVM is known and popular estimations in ML.

It's an algorithm or theory that every data analyst or AI analyst should know. Mathematical formula for SVM:

$$x \cdot y = ||x||\,||y|| \frac{x_1y_1 + x_2y_2}{||x||\,||y||} = x_1y_1 + x_2y_2$$

Code for SVM -

```
from sklearn.svm import SVR
def svm(X_train,y_train,X_test,y_test):
  svm = SVR()
  svm_model = svm.fit(X_train,y_train)
  print('Train set')
  pred=svm_model.predict(X_train)
  print('SVM Mean Squared error :{}'.format(mean_squared_error(y_train,pred,squared=False)))
  print('SVM r2_score :{}'.format(r2_score(y_train,pred)))
  print('Tesst set')
  pred=svm_model.predict(X_test)
  mae=mean_absolute_error(y_test,pred)
  print('SVM Mean Absolute error :',mae)
  print('SVM Mean Squared error :{}'.format(mean_squared_error(y_test,pred,squared=False)))
  print('SVM r2_score :{}'.format(r2_score(y_test,pred)))

  plt.scatter(y_test,pred,color='gray')
  plt.xlabel('Actual values')
  plt.ylabel('Predicted values')
  plt.plot(np.unique(y_test), np.polyld(np.polyfit(y_test, pred, 1))(np.unique(y_test)))
  plt.text(0.6, 0.5, 'R-squared = %0.2f' % r2_score(y_test,pred))
  plt.show()
  df = pd.DataFrame({'Actual': y_test, 'Predicted': pred})
  df1 = df.head(30)
  df1.plot(kind='bar',figsize=(10,8))
  plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
  plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
  plt.show()
  visualizer = ResidualsPlot(svm_model)
  visualizer.fit(X_train, y_train)
  visualizer.score(X_test, y_test)
  visualizer.poof()
```

Figure 10 – Support Vector Machine Algorithm

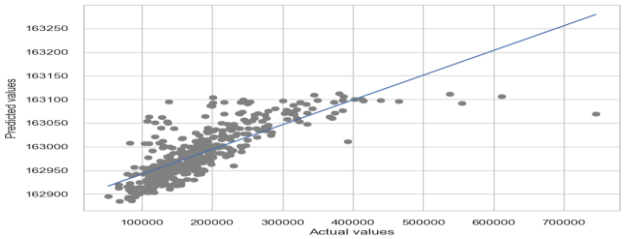

Figure 11 – SVM R-Squared graph

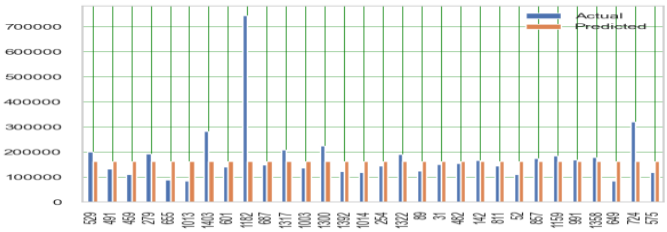

Figure 12 – SVM Actual vs Predicted Chart

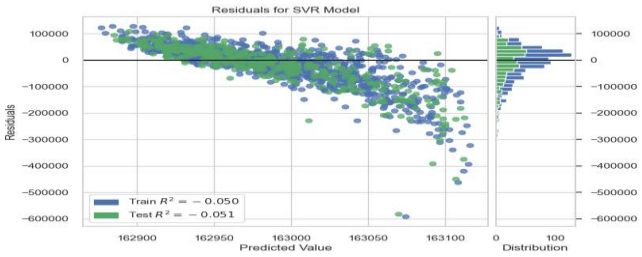

Figure 13 – SVM Test and Train Residuals

iv. LASSO Regression

Lasso Regression is a one type of regression that is called as LR (linear regression) that utilizes shrinkage. Shrinkage is the place information esteems are contracted towards an essential issue, like the mean. The Lasso method empowers basic, scanty models (for example models with less boundaries). This specific sort of relapse is appropriate for models indicating significant levels of multi-collinearity or when you need to mechanize certain pieces of model choice, like variable choice/boundary disposal.

The full form for "Lasso" is (Least Absolute Shrinkage and Selection Operator).

Lasso answers or solutions are programming issues that are quadratic, which are best comprehended with programming (like MATLAB). The objective of the calculation is to minimize below equation:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Which is the same as minimizing the sum of squares with constraint Σ |Bj≤ s. Some of the βs are shrunk to exactly zero, resulting in a regression model that's easier to interpret.

Code for Lasso Algorithm:

```
def lassoCV(X_train,y_train,X_test,y_test):
  lasso = LassoCV(alphas=np.arange(0.5,5,0.5), n_alphas=0,normalize=True,max_iter=100000)
  lasso_model = lasso.fit(X_train,y_train)
  print('Train set')
  pred=lasso_model.predict(X_train)
  print('LassoCV Mean Squared error :{}'.format(mean_squared_error(y_train,pred)))
  print('LassoCV r2_score :{}'.format(r2_score(y_train,pred)))
  print('Tesst set')
  pred=lasso_model.predict(X_test)
  mae=mean_absolute_error(y_test,pred)
  print('LassoCV Mean Absolute error :',mae)
  print('LassoCV Mean Squared error :{}'.format(mean_squared_error(y_test,pred)))
  print('LassoCV r2_score :{}'.format(r2_score(y_test,pred)))
  plt.scatter(y_test,pred,color='gray')
  plt.xlabel('Actual values')
  plt.ylabel('Predicted values')
  plt.plot(np.unique(y_test), np.polyld(np.polyfit(y_test, pred, 1))(np.unique(y_test)))
  plt.text(0.6, 0.5, 'R-squared = %0.2f' % r2_score(y_test,pred))
  plt.show()
  df = pd.DataFrame({'Actual': y_test, 'Predicted': pred})
  df1 = df.head(30)
  df1.plot(kind='bar',figsize=(10,8))
  plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
  plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
  plt.show()
  visualizer = ResidualsPlot(lasso_model)
  visualizer.fit(X_train, y_train)
  visualizer.score(X_test, y_test)
  visualizer.poof()
```

Figure 14 – LASSO Regression Algorithm

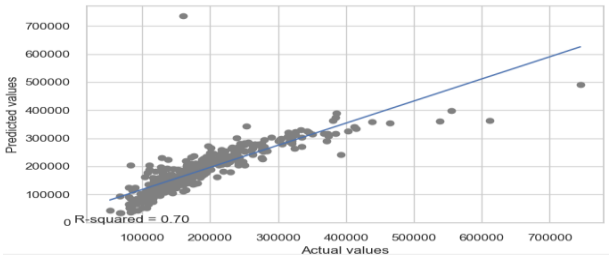

Figure 15 - Lasso R-Squared graph

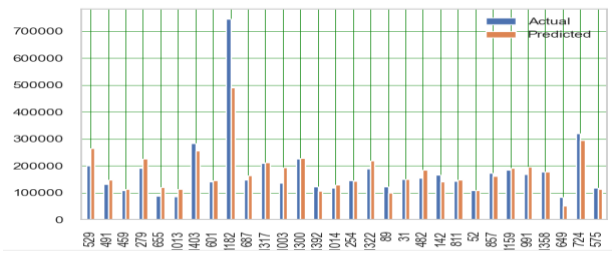

Figure 16 – Lasso Actual vs Predicted Chart

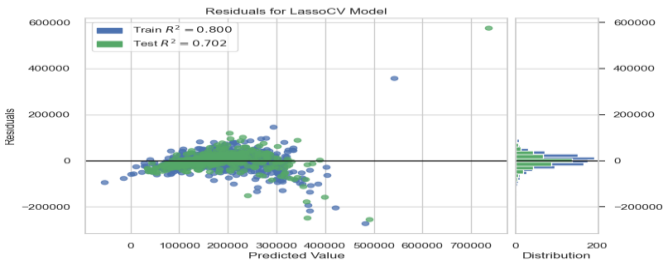

Figure 17 – Lasso's Test and Train Residuals

v. Ada-Boost Algorithm

When nothing works, Boosting does. These days numerous individuals use either XGBoost or LightGBM or Cat Boost to win rivalries at Kaggle or Hackathons. AdaBoost is the initial used algorithm in the Boosting world. In solving boosting problems, AdaBoost is the boosting algorithms used earlier. Adaboost helps you concatenate multiple classifiers i.e. a "strong classifier" is built using multiple "weak classifiers". AdaBoost works by placing additional weight on hard to order occurrences and less on those effectively taken care of well. It very well may be utilized for both characterization and relapse issue. The last condition for characterization can be spoken to as:

$$F(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x))$$

where f_m represents the m_th (weaker)powerless classifier and theta_m is the relating weight. It is actually the weighted mix of M feeble(weak) classifiers.

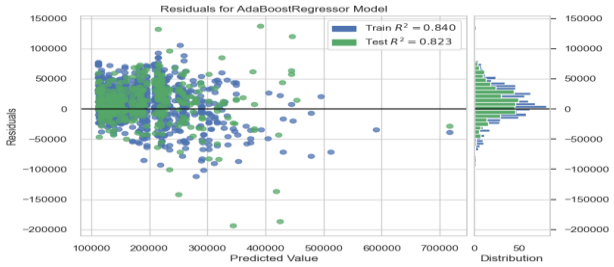

Figure 18 – Ada-Boost's Test and Train Residuals

```
def AdaBoost(X_train,y_train,X_test,y_test):
  ada = AdaBoostRegressor()
  ada.fit(X_train,y_train)
  print('Train set')
  pred=ada.predict(X_train)
  print('Adaboost Mean Squared error :{}'.format(mean_squared_error(y_train,pred)))
  print('Adaboost r2_score :{}'.format(r2_score(y_train,pred)))
  print('Tesst set')
  pred=ada.predict(X_test)
  mae=mean_absolute_error(y_test,pred)
  print('AdaBoost Mean Absolute error :',mae)
  print('Adaboost Mean Squared error :{}'.format(mean_squared_error(y_test,pred)))
  print('Adaboost r2_score :{}'.format(r2_score(y_test,pred)))
  plt.scatter(y_test,pred,color='gray')
  plt.xlabel('Actual values')
  plt.ylabel('Predicted values')
  plt.plot(np.unique(y_test), np.polyld(np.polyfit(y_test, pred, 1))(np.unique(y_test)))
  plt.text(0.6, 0.5, 'R-squared = %0.2f' % r2_score(y_test,pred))
  plt.show()
  df = pd.DataFrame({'Actual': y_test, 'Predicted': pred})
  df1 = df.head(30)
  df1.plot(kind='bar',figsize=(10,8))
  plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
  plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
  plt.show()
  visualizer = ResidualsPlot(ada)
  visualizer.fit(X_train, y_train)
  visualizer.score(X_test, y_test)
  visualizer.poof()
```

Figure 19 – Ada-Boost Regression Algorithm

vi. Random Forest Algorithm

Random Forest is an adaptable, simple to utilize AI calculation that produces, even without hyper-boundary tuning, an extraordinary outcome not than more often. On the record of its straightforwardness and decent variety, it is likewise one of the most utilized calculations, (for both order(classification) and relapse(regression) errands). In this post we will figure out the calculation functions for random forest algorithm, how it varies from different calculations and how to utilize it.

Random Forest is an administered learning calculation. The "forest" it constructs, is a gathering of choice trees, generally prepared with the "bagging" technique. The overall thought of the bagging strategy is that a mix of learning models expands the general outcome.

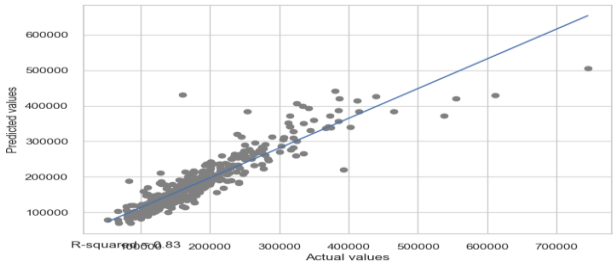

Figure 20 – Random Forest R-Squared graph

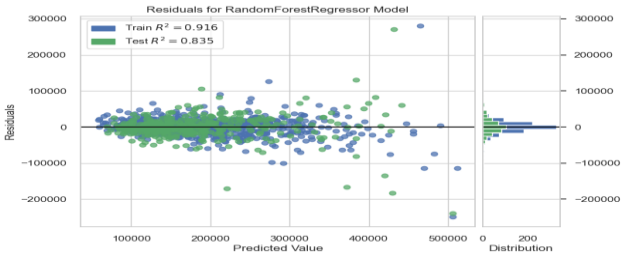

Figure 21 – Random Forest's Test and Train Residuals

vii. Stacking Algorithm

Stacking is a group learning method that consolidates various characterization or relapse models by means of a meta-classifier or a meta-regressor. The base level models are prepared dependent on a total preparing set, at that point the meta-model is prepared on the yields of the base level model as highlights.

The base level frequently comprises of various learning calculations and hence stacking outfits are regularly heterogeneous. The calculation utilized in this paper for stacking are Decision Tree, Lasso and Random Forest.

Stacking is a normally utilized strategy for winning the Kaggle information science rivalry. For instance, the primary spot for the Otto Group Product Classification challenge was won by a stacking outfit of more than 30 models whose yield was utilized for three meta-classifiers the features are: 1. Adaboost, 2. XGBoost and 3. Neural Network.

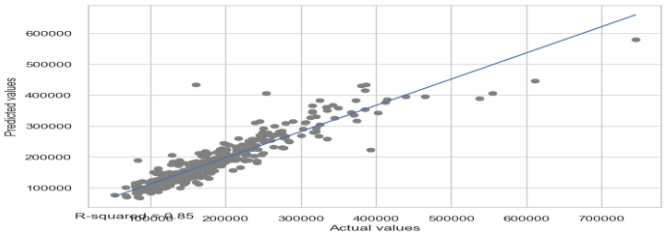

Figure 22 - Stacking Algorithm (Combination of Decision tree, RF, Lasso) R-Squared graph

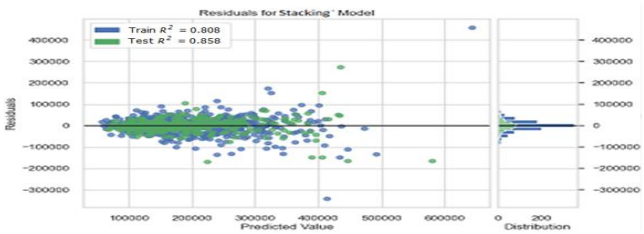

Figure 23 – Stacking Test and Train Residuals

When compared all the used algorithm's in the paper, the results from Stacking algorithm were found to be the best in terms of all the performance metrics.

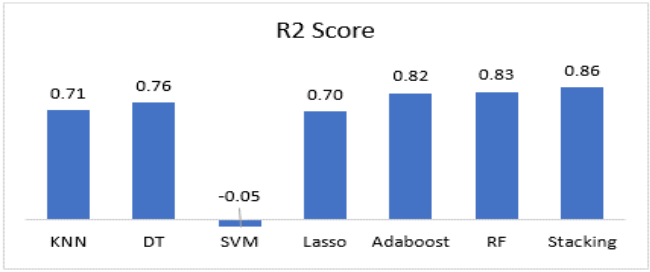

Figure 24 - Comparison of various Regressions

IV. CONCLUSION

In this examination paper, we have utilized AI calculations to anticipate the house costs. We have referred to the one small step at a time procedure to separate the dataset and finding the connection between the limits. Hence, we can choose the boundaries which are not related to one another and are autonomous in nature. We have referred to the one small step at a time approach to separate the dataset and finding the connection between the limits. Thus, we determined the exhibition of each model utilizing distinctive execution measurements and looked at them dependent on these measurements.

Algorithm	MAE	MSE	R2 Score	Time (sec)
KNN	26071.01	43780.72	71%	0.36
DT	26381.53	40249.08	76%	0.21
SVM	56432.15	84452.73	-5%	0.95
Lasso	23926.98	2021381081	70%	0.29
AB	24374.28	1203303333	82%	0.72
RF	19127.73	951774101.7	83%	6.67
Stacking	19097.68	32101.6	86%	0.17

Table 2 -Comparison of Algorithm's Based on parameters

Algorithm	D-1500	D - 1200	D- 1000	D- 800	D - 500	D - 400	D - 300	D - 200	D -100
KNN	0.71	0.71	0.68	0.68	0.69	0.73	0.72	0.69	0.44
DT	0.76	0.63	0.68	0.6	0.72	0.69	0.45	0.69	0.61
SVM	-0.05	-0.04	-0.039	-0.04	-0.012	-0.01	-0.04	-0.01	-0.08
lasso	0.70	0.75	0.75	0.81	0.82	0.84	0.79	0.79	0.68
Adaboost	0.82	0.66	0.65	0.73	0.79	0.82	0.81	0.79	0.73
RF	0.83	0.81	0.82	0.81	0.82	0.84	0.83	0.81	0.67
Stacking	0.86	0.78	0.8	0.8	0.82	0.85	0.84	0.84	0.69

Table 3 - Comparison of R2 score based on different dataset size

This article mostly focuses on the examination between various AI calculations (Random Forest, Ridge Regression, LASSO Regression, Stacking Regression, Ada Boosting Regression, KNN Algorithm, Decision Tree) about House value expectation Analysis. From the above test results, stacking algorithm has high exactness esteem when contrasted with the various calculations with respect to house value expectations. Here the [MSE] Mean Square Error and Mean Absolute Error are utilized to ascertain the exactness estimation of the calculation on the Boston City of United States Dataset which was gathered from public dataset. The paper can be reached out by applying the above said calculations to anticipate House resale esteem.

For future work, we suggest that taking a shot at huge dataset would yield a superior and genuine picture about the model. We have attempted just barely any Machine Learning calculations that are really relapse calculations however we have to prepare numerous other information and comprehend their anticipating conduct for constant qualities as well. By improving the mistake esteems this examination work can be helpful for advancement of uses for different individual urban communities.

V. ABBREVIATION AND ACRONYM

1. DT – DECISION TREE
2. RF – RANDOM FOREST
3. AB – ADABOOST
4. DFD – DATA FLOW DIAGRAM
5. KNN – K NEAREST NEIGHBOR
6. SVM – SUPPORT VECTOR MACHINE
7. LASSO - LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR

VI. REFERENCE

[1] R. J. Shiller, “Understanding recent trends in house prices and home ownership,” National Bureau of Economic Research, Working Paper 13553, Oct. 2007.

[2] D. Belsley, E. Kuh, and R. Welsch, Regression Diagnostics: Identifying Influential Data and Source of Collinearity. New York: John Wiley, 1980.

[3] J. R. Quinlan, “Combining instance-based and modelbased learning,” Morgan Kaufmann, 1993, pp. 236–243.

[4] S. C. Bourassa, E. Cantoni, and M. Hoesli, “Predicting house prices with spatial dependence: a comparison of alternative methods,” Journal of Real Estate Research, vol. 32, no. 2, pp. 139–160, 2010. [Online] Available: http://EconPapers.repec.org/RePEc:jre:issued:v:3 2: n: 2:2010: p:139-160.

[5] S. C. Bourassa, E. Cantoni, and M. E. Hoesli, “Spatial dependence, housing submarkets and house price prediction,” eng, 330; 332/658, 2007, ID: unige:5737. [Online]. Available: http:// archive - ouverte. unige. ch/unige:5737.

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

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

[8] Park, Byeonghwa, and Jae Kwon Bae. "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data." Expert Systems with Applications 42.6 (2015): 2928-2934.

[9] Bhuriya, Dinesh, et al. "Stock market predication using a linear regression." Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of.Vol. 2. IEEE, 2017.

[10] Majumder, Manna, and MD Anwar Hussain. "Forecasting of Indian stock market index using artificial neural network. "Information Science (2007): 98-105.

[11] Li, Li, and Kai-Hsuan Chu. "Prediction of real estate price variation based on economic parameters." Applied System Innovation (ICASI), 2017 International Conference on. IEEE, 2017.

[12] Hromada, Eduard. "Mapping of real estate prices using data mining techniques." Procedia Engineering 123 (2015): 233- 240.

[13] Y. P. Anggodo, A. K. Ariyani, M. K. Ardi, and W. F. Mahmudy, ―Optimation of Multi-Trip Vehicle Routing Problem with Time Windows using Genetic Algorithm, ‖ J. Environ. Eng. Sustain. Technol., vol. 3, no. 2, pp. 92–97, 2017.

[14] R. A. Rahadi, S. K. Wiryono, D. P. Koesrindartotoor, and I. B. Syamwil, Factors influencing the price of housing in Indonesia, ‖ Int. J. Hous.Mark. Anal., vol. 8, no. 2, pp. 169–188, 2015.

[15] V. Limsombunchai, ―House price prediction: Hedonic price model vs. artificial neural network, ‖ Am. J. …, 2004.

[16] D. X. Zhu and K. L. Wei, ―The Land Prices and Housing Prices —— Empirical Research Based on Panel Data of 11 Provinces and Municipalities in Eastern China, ‖ Int. Conf. Manag. Sci. Eng., no. 2009, pp. 2118–2123, 2013.

[17] S. Kisilevich, D. Keim, and L. Rokach, ―A GIS-based decision support system for hotel room rate estimation and temporal price prediction: The hotel brokers’ context, ‖ Decis. Support Syst., vol. 54, no. 2, pp. 1119–1133, 2013.

[18] C. Y. Jim and W. Y. Chen, ―Value of scenic views: Hedonic assessment of private housing in Hong Kong, ‖ Landsc. Urban Plan., vol. 91, no. 4, pp. 226–234, 2009.

[19] L. Bryant, ―Housing affordability in Australia: an empirical study of the impact of infrastructure charges, ‖ J. House. Built Environ., 2016.

[20] S. Rosen, ―Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition, ‖ J. Polit. Econ., vol. 82, no. 1, pp. 34–55, 1974.

[21] https://www.kaggle.com/alphaepsilon/housing-prices-dataset.