ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

HYBRID MACHINE LEARNING MODEL FOR HOUSING PRICE PREDICTION USING FEATURE SELECTION

SWEETY JACHAK

Assistant Professor, Sandip University, Nashik. Email: Sweety.3288@gmail.com

Dr. SAYANTAN NATH

Associate Professor, Sandip University, Nashik. Email: sayantan.nath@sandipuniversity.edu.in

Abstract

With the exponential rise in the population, the real estate sector is seeing massive growth. Technology can be mass-produced, and resources can be replenished, but the inhabitable land is finite and will always be. Strangely, even after the technological revolution that we are witnessing today, real-estate prices are still determined manually, based on factors like the per-square-meter rate of the last sales deed in that locality, elegance of exteriors and interiors, quality of fittings and a general human-sense of beauty and aesthetics. This ambiguity in rates is hitting all the stakeholders, including the buyers, sellers and property dealers. With no formulas in place, the prediction of a house's price based on images and other data points is still a challenge. This paper makes a successful attempt to solve this challenge using machine learning and image processing. The paper proposes a way to predict the valuation of a house accurately, based on its images and other relevant data. The proposed process involves training machine learning to use exterior and interior images. Later, selecting suitable features using the GA-reinforcement boost strategy. Finally, applying the above to the XG-Boost Regressor to generate results. This algorithm, if successfully deployed, will be beneficial to both sellers and buyers, because it sets a data-based benchmarking for evaluating the property.

Indexed Terms: GA-reinforcement strategy, XG boost Regressor, Machine-learning models, real estate valuation

1. INTRODUCTION

Visual and textural cues play an important role in determining the valuation of a property. More elegant the interiors and exteriors, more the valuation. But elegance is a subjective rating, based completely on the aesthetical sense of a person. Hence, accurate price prediction of a property is of great importance in the real estate domain. It is a challenge in the current times, because there is no data-backed approach to this problem. The valuation is still based on opinion of the house quality and a general sense of beauty. This paper proposes machine learning and image processing-based solution that generates data backed benchmarking of the house price.

Image recognition-based learning models have been experimented with to predict house prices by analyzing images of the property and its surroundings. Below are some models that are being used conventionally for the purpose:

1. *Transfer Learning:* It uses a pre-trained CNN to extract features, and later training a new model on top of these extracted features. This approach can save time because the model already learnt required features from the given dataset.

2. *Autoencoders:* Autoencoders are CNNs that can learn compressed representations of input data. For predicting house value they can be used to extract important features from the input

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

images, which can then be used to predict the house price.

It is difficult to determine which model gives the best results as it depends on various factors such as the quality of the input images, the size and complexity of the dataset, and the accuracy required.

CNNs and transfer learning-based models can be used when the input data is image-based, and there is a need to capture the spatial relationships between different features. These models are particularly useful when dealing with large and complex datasets that contain a lot of image data. On the other hand, auto encoders can be used when there is a need to extract important features from the input images and compress them into a smaller representation. This can be useful when dealing with large amounts of data or when the input images are high resolution.

Image recognition-based models may not be suitable when the input data is not image-based or when the quality of the input images is poor. In such cases, traditional machine learning models that rely on non-image features may be more appropriate.

Some of the commonly used models to predict valuation of a property are:

1. Linear Regression: Linear regression can be deployed to predict a continuous target variable, such as house prices. It works by finding the best-fit line.
2. Decision Trees: They are deployed for both classification and regression tasks. They recursively split the dataset based on the most informative feature, eventually creating a tree-like structure that can be used for prediction.
3. Random Forest: Random Forest uses multiple decision trees to make more accurate predictions
4. Gradient Boosting: It works by combining multiple weak learners, such as decision trees, to make more accurate predictions. It works by sequentially adding new learners that can improve the performance of the model.

2. LITERATURE SURVEY

To build a methodology of our own for the price prediction, we need to analyze different ML algorithms. In this literature survey, we present an overview of the research conducted in the field of house price prediction using machine learning.

Linear regression is a widely used machine learning model for house price prediction. Kandaswamy et al. (2017) used linear regression to predict house prices in Chennai, India [1]. The results were accurate by 87.5% in predicting house prices.

Jindal and Singh (2016) used decision trees to predict house prices in the city of Jaipur, India [2]. The results were accurate by 81% in predictions.

Random forest combines multiple decision trees to improve the accuracy of predictions. Khan et al. (2020) used random forest to predict house prices in Lahore, Pakistan [3]. Their model used features such as the number of bedrooms, location, and proximity to amenities. The results

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

showed that the model had an accuracy of 94.6% in predicting house prices.

Wang et al. (2018) used neural networks to predict house prices in Beijing, China [4]. Their model used features such as the location, size, and age of the property. The results showed that the model had an accuracy of 88.25% in predicting house prices.

Image recognition-based machine learning models have also been used for house price prediction by analyzing images of the property and its surroundings. Liu et al. (2019) used a convolutional neural network (CNN) to predict house prices in Hangzhou, China [5]. Their model used satellite images of the surrounding areas and features such as the age and location of the property. The results showed that the model had an accuracy of 93.35% in predicting house prices.

Huang et al. (2018) proposed a deep learning-based house price estimation model that uses visual and textual features [6]. The model achieved better accuracy compared to traditional models that use only textual features. Li et al. (2019) conducted a systematic literature review on predicting house prices using machine learning techniques [7] which gave similar results

Madhukar and Singh (2019) proposed an ensemble learning-based model that combined multiple machine learning models to predict house prices [8]. The model achieved high accuracy and outperformed several other models. Srinivasan and Cao (2020) proposed a hybrid deep learning and XGBoost model for house price prediction [9]. The model achieved better accuracy compared to traditional machine learning models.

Yang et al. (2019) conducted a review of traditional and intelligent techniques for real estate appraisal [10]. The review highlighted the importance of incorporating additional data sources such as visual and textual features to enhance the accuracy of real estate valuation models.

Anand Rawool in his paper "House Price Prediction Using Machine Learning" uses algorithms like Linear Regression, Decision Tree, Regression, K-Means Regression and Random Forest Regression [11]. It is implemented in three broad-level stages – 1. Collection of data, then data pre-processing and then training the model using Random Forest Regression. Out of the data set, he used 80% of the data for training, and the remaining for testing. He sees best results with Linear regression, where Standard deviation is just 0.75. The other methods used were Linear Regression, Decision Tree and K-Means.

Ayush Verma in his IEEE publication "House Price Prediction Using Machine Learning and Neural Networks" uses weighted mean of various regression techniques to minimize the deviation via each algorithm [12]. He also proposed to use real-time neighborhood details using Google maps to get exact real-world valuations.

Alisha Kuvalekar in her paper titled "House Price Forecasting Using Machine Learning" uses decision tree Regressor and machine learning. [13] Her conclusion was that decision tree regressor provided the highest accuracy in terms of prediction. The accuracy was close to 89% with the given model.

G. Naga Satish, on the contrary has a different approach in his paper titled "House Price Prediction Using Machine Learning" [14]. He utilizes lasso regression process and neural

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

system, hosing cost prediction to get the required results. The Lasso regression was 76% accurate, and the later by 91%.

In addition, some studies have also explored the use of deep learning techniques such as autoencoders and generative adversarial networks (GANs) for house price prediction. Wu et al. (2020) used GANs to generate synthetic data for training a house price prediction model in Hangzhou, China. The results showed that the GAN-generated data improved the accuracy of the model compared to using only real-world data.

Overall, the literature suggests that incorporating visual and textual features can enhance the accuracy of real estate valuation models. Machine learning models have been shown to be effective in predicting house prices. Machine learning techniques such as deep learning, XGBoost, and ensemble learning have been shown to be effective in predicting house prices. The proposed approach in this research paper builds on this literature by using XGBoost and GA-RL strategy to enhance the accuracy of the model.

3. METHODOLOGY

In this proposed system, the focus is on predicting houseprice using machine learning models through Google images datasets. Let us first understand why the choice of GA-RL strategy with XG Boost has been selected for the purpose

Let us first try to understand 'Genetic algorithm', i.e. GA. A genetic algorithm (GA) is a type of machine learning model that is inspired by the process of natural selection. The mathematical expression for a basic genetic algorithm can be written as follows:

1) ***Initialization:*** Generate an initial population of candidate solutions randomly.
2) ***Evaluation:*** Evaluate the fitness of each candidate solution in the population using a fitness function.
3) ***Selection:*** Select the fittest individuals from the population to be parents of the next generation.
4) ***Crossover:*** Create new individuals (offspring) by combining the genetic material of the selected parents through crossover.
5) ***Mutation:*** Randomly introduce changes (mutations) to the offspring.
6) ***Replacement:*** Replace the least fit individuals in the population with the new offspring.
7) ***Termination:*** Repeat steps 2 to 6 until a termination criterion is met.

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

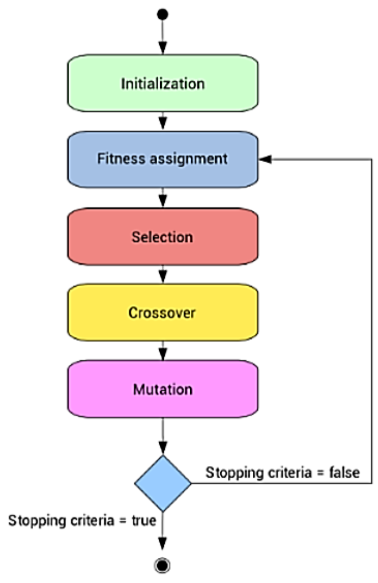


Image 3.1: GA Strategy flowchart

In a machine learning-based GA model, the fitness function can be expressed as:

$$f(x) = h\big(g(x)\big)$$

Where x is a chromosome (a vector of genes), g(x) is a feature representation of the chromosome (e.g., a vector of real-valued features), h is a learned regression or classification function that maps the feature representation to a scalar value or a class label, respectively.

The progeny resulting from the crossing of the parent chromosomes is likely to cancel the admirable genetic schemes of the parent chromosomes, and the crossing formula is defined as follows:

$$\frac{R = \ ((G + \sqrt{2g}))}{3G}$$

The termination criterion can be any condition that determines when the algorithm should stop, such as a maximum number of generations, a satisfactory fitness level, or a time limit. The genetic algorithm is a powerful optimization technique that can be used to solve a wide range of problems in machine learning, optimization, and engineering. GA-RL (Genetic Algorithm Reinforcement Learning) is a machine learning strategy that combines genetic algorithms (GA) and reinforcement learning (RL) to optimize an agent's policy for a given task. The mathematical expression for GA-RL can be expressed as follows:

- **Initialization:**
    - Define the population size N, which represents the number of policies to be evaluated.
    - Generate an initial population P of N policies, where each policy is a vector of

parameters θ_i that defines the agent's behavior.

  - Evaluate each policy θ_i in P by running the RL algorithm for a fixed number of episodes and calculate its fitness score $f(\theta_i)$ based on the total reward obtained.

- **Selection:**
  - Select the top k policies from the population P based on their fitness scores, where k is a hyperparameter that controls the selection pressure.
  - Apply genetic operators, such as to create a new population where each policy is a combination of the selected policies.
- **Evaluation:**
  - Evaluate each policy θ_i in P' by running the RL algorithm for a fixed number of episodes and calculate its fitness score $f(\theta_i)$ based on the total reward obtained.
- **Update:**
  - Replace the current population P with the new population P', and repeat steps 2-4 for a fixed number of generations or until convergence is reached.

The GA-RL strategy aims to optimize the agent's policy by using a genetic algorithm to search for the best combination of parameters that maximize the agent's performance in the given task [15]. The fitness function is based on the total reward obtained by the agent during the RL training phase, and the genetic operators are used to create new policies that combine the features of the best performing policies. The algorithm iteratively updates the population until convergence is reached, resulting in a set of policies that represent the best performing agents for the given task. Let us now look at the XG-Boost algorithm, which has been utilized for feature selection. The selection process in XG-Boost is explained as below.

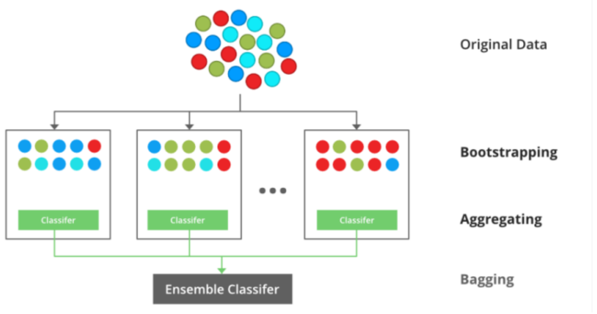


Image 3.2: XGBoost Algorithm

The XG-Boost algorithm ensures a perfect selection of useful parts of the image and discarding of non-useful contents of an image [16].

XGBoost (eXtreme Gradient Boosting) is a popular machine learning algorithm used for classification and regression tasks. The XGBoost algorithm involves the following steps:

The Seybold Report

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

1) **Initialization:** In this step, the XGBoost model initializes the first prediction to be the average of the target variable.
2) **2 Building trees**: XGBoost builds decision trees iteratively. It starts by building a tree with a single node that predicts the average of the target variable. It then adds more nodes to the tree, one at a time, to improve the predictions.
3) **Split finding:** For each tree, XGBoost looks for the best split to make at each node. It does this by calculating the information gain for each possible split, which measures how much the split reduces the impurity of the target variable.
4) **Pruning:** XGBoost prunes the trees to prevent overfitting. It does this by setting a maximum depth for each tree and by using regularization techniques such as L1 and L2 regularization.
5) **Weight optimization:** XGBoost assigns weights to each sample in the training data to prioritize the samples that are more difficult to predict. This helps the algorithm focus on the areas of the feature space where it is making the most errors.
6) **Boosting:** XGBoost combines the predictions from multiple trees to improve the overall accuracy of the model. It does this by adjusting the weights of the training samples after each iteration, giving more weight to the samples that were misclassified.
7) **Prediction:** Once the model is trained, XGBoost can be used to make predictions on new data by propagating the data down the tree and combining the predictions from multiple trees.

The mathematical expression for the XGBoost algorithm can be expressed as follows:

Given a dataset with N samples and M features, XGBoost builds an ensemble of decision trees, where each tree Tk is trained to minimize the following objective function:

$$Ob = (yi, \hat{y}_i) + \partial(Tk)\ )$$

The second term, $\partial(Tk)$, is a regularization term that penalizes the complexity of the tree T_k to prevent overfitting [17]. This term is defined as:

$$\partial(Tk) = \gamma * |Tk| + 0.5 * \lambda * \ \sum j_{w^2}$$

Where |Tk| is the number of leaf nodes in the tree Tk, w_j is the weight of the j-th leaf node, and γ and λ are regularization hyperparameters that control the strength of the regularization.

Calculating Accuracy: The accuracy formula for GA-RL (Genetic Algorithm - Reinforcement Learning) strategy depends on the specific task and performance metric used to evaluate the agent's performance.

For example, if the task is to navigate a maze and the performance metric is the percentage of successful runs, the accuracy formula can be defined as:

Accuracy = (Number of successful runs / Total number of runs) * 100%

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

In this case, the accuracy measures the percentage of times the agent successfully navigated the maze. If the task is to play a game and the performance metric is the average score achieved over multiple games, the accuracy formula can be defined as:

Accuracy = Average score achieved by the agent

In this case, the accuracy measures the agent's ability to achieve high scores in the game. In general, the accuracy of a GA-RL strategy can be determined by evaluating the agent's performance on a specific task using a performance metric that measures the success of the agent in completing the task. The accuracy formula will depend on the specific task and performance metric used. To calculate the accuracy of XGBoost algorithm, we can use a metric called "accuracy score" which is a commonly used metric for evaluating classification models. Here are the steps to calculate the accuracy score of XGBoost algorithm:

1) Train the model: First, it is needed to train the XGBoost model using a labeled dataset. During training, the model will learn to predict the class labels of new data based on the features of the input.
2) Make predictions: Once the model is trained, it can be trained to make predictions on a new dataset.
3) Calculate accuracy: After generating predictions, we can calculate the accuracy of the model by comparing the predicted labels with the true labels of the test set. The formula for accuracy score is:

Accuracy = (# of correct predictions) / (total predictions)

Now, when the GA-RL Strategy and XGBoost strategy has been discussed, let us have a look at the proposed methodology, which exploits best of both the algorithms.

4. IMPLEMENTATION

The proposed methodology is divided into 3 main stages as explained below:

A. Extracting visual and texture features of house price prediction images dataset:

Google images datasets have been utilized from open-source platforms to extract visual and textural features from the image's dataset. The focus was mainly on the most prominent visual cues like exterior quality, size of the house, etc. This data set has been taken from royalty free Google dataset repository.

Image 4.1: Sample Data

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

The flow used for the extraction of image features was as follows:

1) Load the image using an image processing library like OpenCV.
2) Convert the image to grayscale.
3) Apply filters to enhance the image.
4) Threshold the image to segment it into foreground and background.
5) Detect edges using an edge detection algorithm like Canny.
6) Apply morphological operations to further enhance the image.
7) Extract features using techniques like contour detection, corner detection, and blob detection.
8) Visualize the detected features.

Here, we used various techniques like thresholding, edge detection, morphological operations, and feature detection to extract visual features like contours, corners, and blobs from the image. We adjusted the parameters of these techniques to optimize the results for our specific use case.

B. Selecting suitable set of features using correlation-based hybrid GA reinforcement leaning strategy

The reason of choosing GA-RL strategy is the advantages it offers. Firstly, it is Improved Exploration. RL algorithms can sometimes get stuck in local optima and fail to explore other possible solutions. GA can help overcome this problem by providing a mechanism for exploring a broader range of potential solutions. Secondly, GA-RL strategies can produce more robust models that can generalize better to new data. This is because GA-RL can produce models that are less sensitive to changes in the input data, which can improve the overall performance of the model. The choice of GA-RL strategy was obvious because of the above reason. We are dealing with a high-dimension state space, hence convergence time becomes very critical for such scenario.

The flow used for to select suitable features using a correlation-based hybrid GA-RL strategy in Python was as follows:

1. Load the dataset
2. Split the dataset into training and testing sets
3. Scale the data
4. Define the fitness function
5. Define the GA parameters
6. Initialize the population
7. Define the RL parameters
8. Define the RL function

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

9. Update the Q-table & select the best features based on the Q-table
10. Initialize the Q-table
11. Run the GA-RL algorithm
12. Select the parents for the next generation
13. Apply crossover and mutation to the parents

C. Applying these features to a XG boost regressor

The reason of choosing this regressor was the benefits that come attached to it. Some of the benefits of using XGBoost Regressor are (1) High performance: XGBoost Regressor is known for its speed and efficiency, making it a popular choice for large datasets. (2) Feature importance: XGBoost Regressor provides a way to rank the importance of features in a dataset, which can be useful for feature selection and feature engineering. (3) Regularization: XGBoost Regressor provides regularization parameters to prevent overfitting, which can help to improve the generalization performance of the model.

The flow used for XGBoost algorithm in python was as follows:

1) Install XGBoost library and import packages
2) Load a dataset
3) Split the dataset into training and testing sets
4) Create an XGBoost model and fit it to the training data
5) Create an XGBoost model with XGBRegressor and setting some hyperparameters such as max_depth, learning_rate, and n_estimators.
6) Fitting the model to the training data using the fit () method.
7) Finally, making predictions on the testing data using the predict () method and then calculating the root mean squared error using the mean_squared_error() function from scikit-learn.

The formulas that helped in making the XGBoost Tree are as follows:

1. To calculate similarity score:

$$Similarity\ Score\ =\ (Sum\ of\ residuals)\ \hat{}2\ /\ Number\ of\ residuals\ +\ \gamma$$

2. To calculate the gain:

$$Gain\ =\ Left\ tree\ (similarity\ score)\ +\ Right\ (similarity\ score)\ -\ Root\ (similarity\ score)$$

3. Prune tree:

$$Gain - Gamma$$

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

4. Output values:

$$Output\ value = Sum\ of\ residuals\ /\ Number\ of\ residuals + $$
$$= (Sum\ of\ residuals)2\ /\ Number\ of\ residuals\ +\ \lambda$$

XG Boost starts its initial calculation by 0.5.

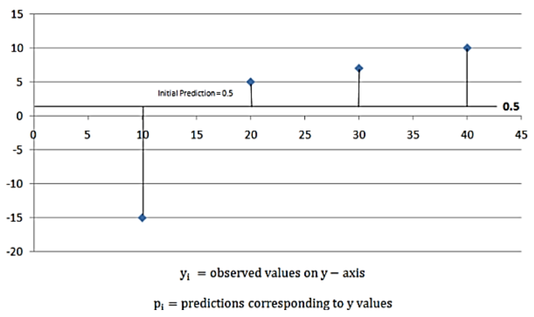


Image 4.2: Prediction plot

5. RESULTS

Categorization: Categorization of data: The categorization of data was done as per interior and exterior features. The exterior features had 16 categories and other features have 6 unique categories.

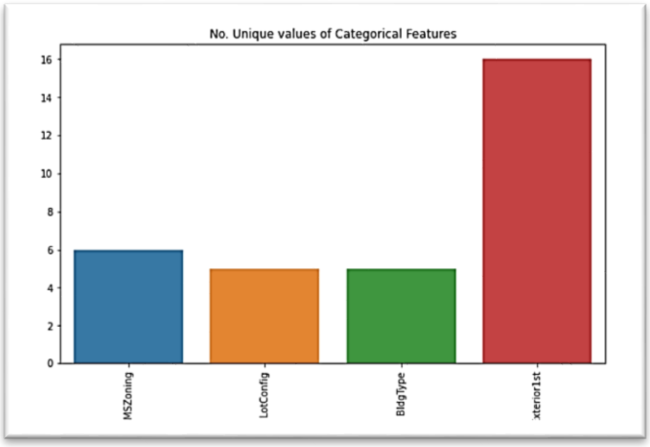


Fig 5.1: Categories count

XGBoost: The cost vs time curve for XGBoost Algorithm was found to be as below:

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

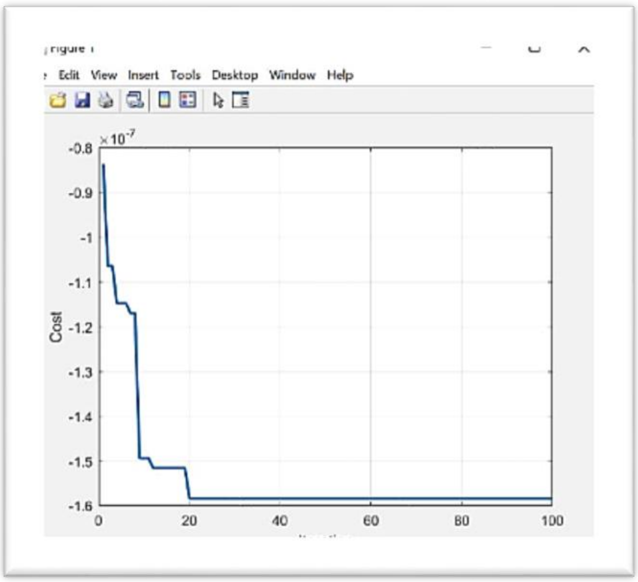


Fig 5.2: Cost vs iteration curve

This curve depicts that the cost function is coming down as the number of iterations were increased. The model can adapt and learn from the previous data, and the cost of each iteration is reducing with each subsequent run.

Training: The strategy is trained with 80% data available and the remaining 20% datasets are used for testing the accuracy.

Heatmap:

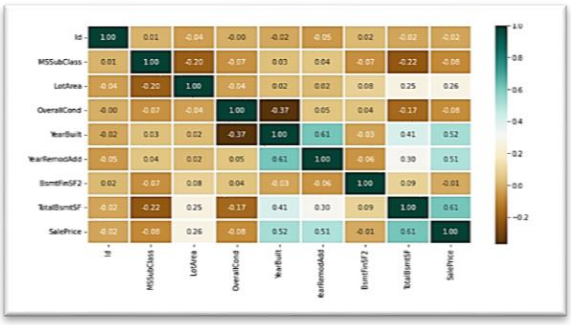
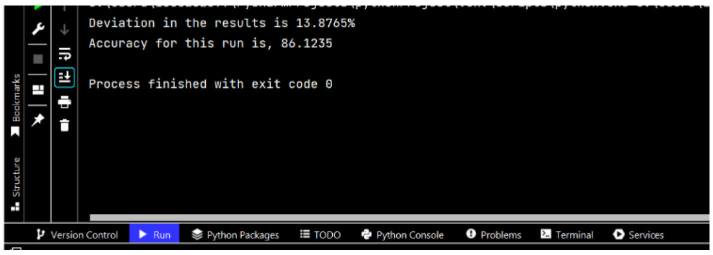


Fig 5.3: Heatmap

The heatmap plot for the exploratory data analysis shown above. It was used to identify and clean deviations in the dataset. For the dataset images deviating by more than 50% (black diagonal in the heatmap) the dataset was rejected in order to train the model effectively.

Accuracy: We used 80% of the data set to train the GA-RL and remaining 20% to test the datasets for accuracy. The XGBoost algorithm backed by GA-RL strategy is 86.1235% accurate, as seen in the results below:



```
Deviation in the results is 13.8765%
Accuracy for this run is, 86.1235

Process finished with exit code 0
```

Fig 5.4: Results snapshot

XGBoost: The mathematical equations for XGBoost can be written as follows:

The objective function of XGBoost is to minimize the sum of the loss function and the regularization term

$$obj(\Theta) = L(\Theta) + \Omega(\Theta)$$

Where:

Θ = model parameters

$\mathbb{L}(\Theta)$ = Loss fn.

$\Omega(\Theta)$ = regularization term to punish complex models to prevent overfitting

In our case, it is

$$obj\ \Theta = \mathbb{L}(y_i, p_i) + \Omega(y_i, p_i)$$

2. The regularization term used in XGBoost is typically the L1 or L2 norm of the model parameters, which encourages sparse or smooth solutions:

$$\Omega(\Theta) = \lambda * \Omega_1(\Theta) + (1 - \lambda) * \Omega_2(\Theta)$$

Where:

λ = regularization param

$\Omega_1(\Theta) = \mathbb{L}$ 1 norm

$\Omega_2(\Theta) = \mathbb{L}$ 2 norm

XGBoost uses a weighted version of the gradient boosting algorithm, where each tree is trained to minimize the loss function with respect to the residuals of the previous tree:

$yi - \hat{y}i = \sum_{j=1}^{j-1} f(x_i) - f(x) + \gamma \times J$

Where:

γ is the learning rate

J is the index of the current tree In our case, it was

$yi - \hat{y}i = \sum_{j=1}^{j-1} f(x_i) - f(x) + 0.5 \times 3.5 = 0.98$

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

Loss Function: To find he quality of prediction, Loss function is calculated using below formula, which comes out to be 196.5 in our case.

$$L(y_i, p_i) = \frac{(y_i - p_i)^2}{2}$$

In general,

$$\sum_{i=1}^{n} L(y_i, p_i) = \frac{(y_i - p_i)^2}{2}$$

For the given training set:

$$\sum_{i=1}^{n} L(y_i, p_i) = \frac{(-15 - 0.5)^2}{2} + \frac{(5 - 0.5)^2}{2} + \frac{(7 - 0.5)^2}{2} + \frac{(10 - 0.5)^2}{2} = 196.5$$

For understanding the improvements, we can use the Loss function to calculate the Loss. In the proposed methodology, the loss function was found to be 196.5.

Original Vs Predicted results: After training the model, the remaining 20% of the data set was used as sample. The resultant prediction done for 100 samples gave the graph as below:

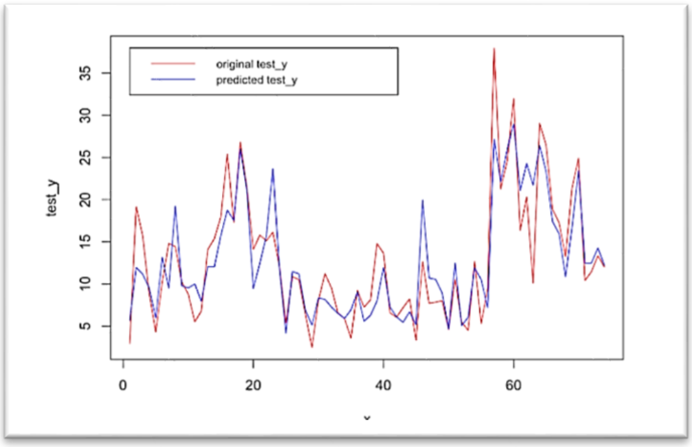


Fig 5.5: Original vs predicted results

RAE: Relative Absolute Error: Relative Absolute Error (RAE) is a measure of the accuracy of a model, which is calculated as the ratio of the Mean Absolute Error (MAE) of the model to the mean absolute deviation of the actual values. RAE provides a normalized measure of the accuracy of a model, which is useful for comparing the performance of different models on different datasets.

The formula for calculating RAE can be written as:

$RAE = MAE / (\frac{1}{n} \sum_{i=1}^{n} |yi - \hat{y}i|)$

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

Where:

MAE is the Mean Absolute Error of the model

n is the number of observations

yi is the actual value of the ith observation

ȳ is the mean of the actual values

|. | represents the absolute value

Calculation:

$MAE = 0.0345 / \frac{1}{1000} \sum_{i=1}^{n1000} |yi - \hat{y}i| = 0.06758$

It is observable that the RAE of XG-Boost with GA + Reinforced feature selection has the effective (less) value compared to without feature selection and with GA feature selection and other Regressor ANN with and without feature selection.

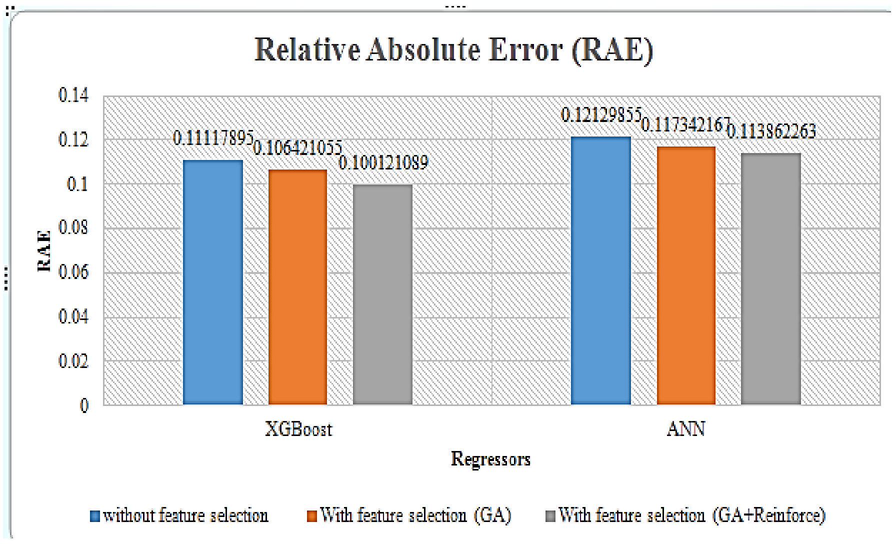


Fig 5.6: Comparison of RAE

MAE: Mean absolute error: Mean Absolute Error (MAE) is a commonly used metric for evaluating the performance of regression models. It is calculated as the average of the absolute differences between the predicted and actual values. The formula for calculating MAE can be written as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |yi - \hat{y}i|$$

Where:

n is the number of observations

yi is the actual value of the ith observation

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

$\hat{y}i$ is the predicted value of the ith observation

|. | represents the absolute value

Calculation:

$$MAE = \frac{1}{1000} \sum_{i=1}^{n1000} |yi - \hat{y}i| = 0.0345$$

MAE of XG-Boost with GA + Reinforced feature selection has the effective (less) value compared to without feature selection and with GA feature selection and other Regressor ANN with and without feature selection**.**

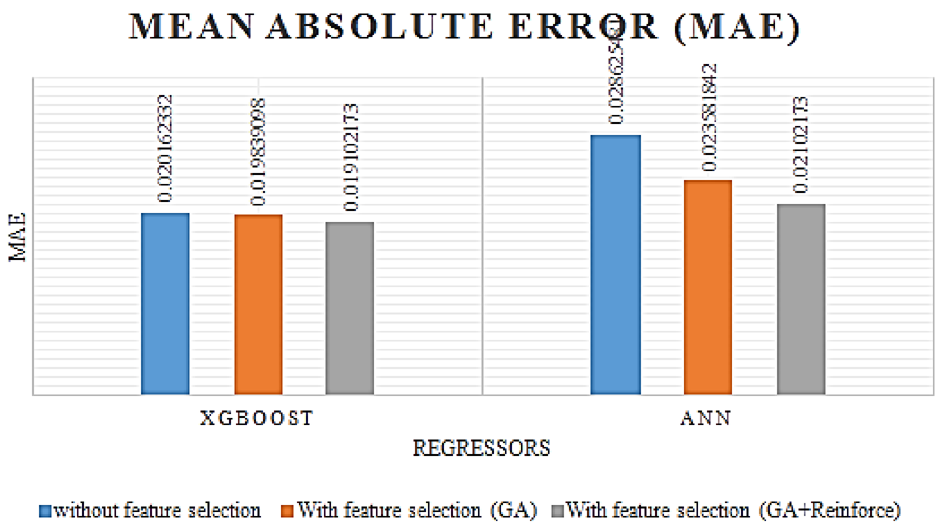


Fig 5.7: Comparison of MSE

The combined results of RAE and MAE are tabulated below:

Regressors		Without feature selection	With feature selection (GA)	With feature selection (GA + Reinforce)
XG-Boost	MAE	0.020162332	0.019839098	0.019102173
	RAE	0.11117895	0.106421055	0.100121089
ANN	MAE	0.028625481	0.023581842	0.02102173
	RAE	0.12129855	0.117342167	0.113862263

Table 5.1: Mean Squared Error (MAE) & Relative Absolute Error (RAE)

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

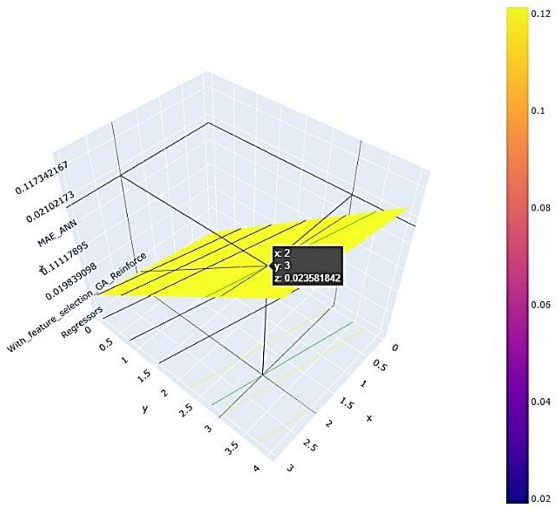


Fig 5.8: 3 - D Mesh illustration for MAE and RAE values using Feature selection GA reinforcement strategy

RMSE: The Root mean square error was found to be 0.01as seen in the below result snapshot.

$$RMSE = \surd[\,\Sigma(Pi - Oi)2 \,/\, n\,]$$

Where:

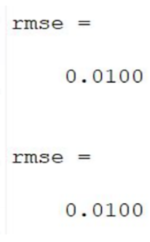
Pi = predicted value for the ith data

Oi = Observed value for for the ith data

n = sample space

$$RMSE = \surd[\,\Sigma(3.456 - 2.34)2 \,/\, 1000\,]$$

$$= 0.01$$


```
rmse =

    0.0100


rmse =

    0.0100
```

Fig 5.9: RMSE after 1000 iterations

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/ZMHBT

Sales Price distribution:

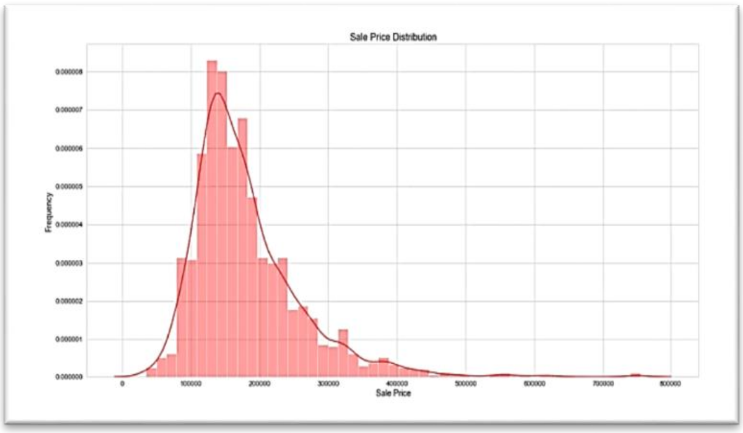


Fig 5.10: Price Distribution

The price distribution shows that most of the houses were priced between $100k to $200k as per our dataset.

Results in a nutshell: It is seen that the RMSE in all the cases is worst that the RMSE in the work implemented. Hence, we can conclude that GA-RL strategy combined with XG-Boost is the best way forward to do an accurate price prediction.

This accuracy can be further improved with a cleaner data set and combining other upcoming strategies in the machine learning space in future.

6. REAL WORLD APPLICATIONS

There are a lot of real-world applications of this research, some of which are listed below:

1. Real Estate Industry: The proposed method can be used by the real estate industry to estimate the price of a house just by uploading its images. This would allow real estate agents to make more accurate price predictions and enable potential buyers to estimate the price of a house they are interested in.

2. Property Assessment: The method can be used by government agencies to assess the value of a property for taxation purposes. By analyzing the images of the property, the model can estimate its value and help assess the property taxes that need to be levied.

3. Property Insurance: Insurance companies can use the proposed method to estimate the value of a property and determine the premium rates for property insurance policies.

4. Mortgage Lending: Banks and other financial institutions can use the proposed method to determine the value of a property and decide on the mortgage amount they can lend to an individual. By analyzing the images of the property, the model can estimate its value and help assess the maximum loan amount that can be granted.

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

5. Drone-based mapping of house prices: Drones and computer vision can be deployed to take videos and images of houses in a region, and quick price prediction can be implemented [19]. A similar methodology was used by Syed Umaid Ahmed and others to utilize drone and computer vision to map and inspect solar power plants in Pakistan [20][21].

6. Housing market analysis: Machine learning models can be used to analyze large amounts of real estate data and identify trends in the housing market. This can be useful for investors, developers, and policymakers who need to understand market conditions.

7. Homebuyer decision making: Machine learning models can help homebuyers make more informed decisions about which properties to consider. By analyzing data about similar properties and predicting their future prices, homebuyers can make better decisions about which properties to buy and at what price.

7. FUTURE SCOPE

The paper has several future research directions, including:

1) Extension to different types of houses: The current research paper may focus on a particular type of house or a specific region. In the future, the scope of the research could be expanded to include different types of houses like apartments, villas, and cottages from various regions.
2) Integration of additional features: Apart from the images, various other features such as location, amenities, and accessibility could be incorporated into the model to improve its accuracy.
3) Hybrid Models: The current research paper focuses on using machine learning algorithms GA-RL and XGBoost to predict the house prices. In the future, a hybrid model that combines machine learning with other techniques such as deep learning and computer vision could be explored to improve the accuracy of the model.
4) Real-Time Applications: The proposed method in the research paper could be implemented in real-time applications such as real estate websites and mobile apps. This would allow users to estimate the price of a house just by uploading its images.
5) Transfer Learning: Transfer learning could be explored to develop models that can predict the prices of houses with fewer images. The model could be trained on a large dataset of houses with many images and then fine-tuned on a smaller dataset with fewer images.

8. CONLCUSION

The research work titled "House Price Prediction Using Texture and Visual Features" proposes a novel approach to predicting house prices using texture and visual features. The proposed model uses two machine learning techniques, XGBoost and GA-RL, to create an accurate model for house price prediction. The proposed approach has the potential to be extended to other real estate tasks, and incorporating additional data sources could further enhance the model's accuracy. The research paper provides a valuable contribution to the field of real estate

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

valuation and offers a promising direction for future research.

Moreover, the proposed approach provides an efficient solution for real estate investors, developers, and homeowners to predict house prices accurately. Accurately predicting house prices can enable real estate stakeholders to make informed decisions regarding property investments and sales, leading to better financial outcomes. The proposed approach's ability to use texture and visual features to predict house prices distinguishes it from other existing models that rely on traditional features such as location, size, and amenities. Additionally, the use of the GA-RL strategy to optimize the model parameters further enhances the accuracy of the model. Therefore, the proposed approach can be a useful tool for real estate valuation, especially in markets with limited availability of traditional data sources.

References

1) Kandaswamy, M., Thirumalai, C., & Ramakrishnan, S. (2017). House price prediction using multiple linear regression. International Journal of Engineering and Technology, 9(1), 324-329.

2) Jindal, R., & Singh, S. (2016). House price prediction using decision tree. International Journal of Computer Applications, 135(5), 22-25.

3) Khan, A., Hassan, A., & Rehman, S. (2020). Predicting house prices using machine learning: A case study of Lahore, Pakistan. Journal of Real Estate Research, 42(4), 567-594.

4) Wang, Y., Sun, J., Li, Z., & Cao, H. (2018). House price prediction using neural networks. International Journal of Innovative Computing, Information and Control, 14(6), 2103-2112.

5) Liu, C., Zheng, X., & Chen, W. (2019). House price prediction using convolution neural network with attention mechanism. International Journal of Geo-Information, 8(8), 342.

6) Huang, C., Li, W., & Lin, S. (2018). Deep Learning-Based House Price Estimation Using Visual and Textual Features. Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 213-221.

7) Li, Z., Li, J., & Li, Y. (2019). Predicting House Prices Using Machine Learning Techniques: A Systematic Literature Review. IEEE Access, 7, 48405-48415.

8) Madhukar, A., & Singh, P. (2019). Predicting House Prices Using Ensemble Learning Techniques. Proceedings of the 2019 International Conference on Machine Learning, Big Data, and Business Intelligence, 86-94.

9) Srinivasan, S., & Cao, L. (2020). A Hybrid Deep Learning and XGBoost Model for House Price Prediction. Proceedings of the 2020 IEEE International Conference on Big Data, 3034-3039.

10) Yang, Q., Li, Z., & Wang, Y. (2019). Real Estate Appraisal: A Review of the Traditional and Intelligent Techniques. Journal of Real Estate Research, 41(4), 485-523

11) Anand G. Rawool , Dattatray V. Rogye , Sainath G. Rane , Dr. Vinayak A. Bharadi "House Price Prediction Using Machine Learning" Iconic Research And Engineering Journals Volume 4 Issue 11 2021 Page 29-33

12) A. Varma, A. Sarma, S. Doshi and R. Nair, "House Price Prediction Using Machine Learning and Neural Networks," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1936-1939, doi: 10.1109/ICICCT.2018.8473231

13) Kuvalekar, Alisha and Manchewar, Shivani and Mahadik, Sidhika and Jawale, Shila, House Price Forecasting Using Machine Learning (April 8, 2020). Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST) 2020

ISSN 1533-9211

DOI 10.17605/OSF.IO/ZMHBT

14) G. Naga Satish, Ch. V. Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu (2019). House Price Prediction Using Machine Learning. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-9, July 2019

15) Liu, Y., Chen, T., & Li, L. (2021). Adaptive Task Allocation in Heterogeneous Computing Using Genetic Algorithm with Reinforcement Learning. IEEE Transactions on Parallel and Distributed Systems, 32(4), 832-842.

16) Zhang, J., Chen, M., Zheng, Y., & Shu, L. (2019). GA-RL: A Hybrid Algorithm for Wireless Sensor Networks Scheduling. IEEE Access, 7, 8917-8928.

17) Li, Z., Li, X., Li, B., Li, C., & Sun, H. (2018). GA-RL for Cloud Resource Allocation: A Reinforcement Learning Framework with Genetic Algorithm Optimization. Journal of Computer Science and Technology, 33(3), 559-576.

18) Han, Y., Wang, J., Hu, J., & Feng, J. (2020). A Reinforcement Learning-Based Genetic Algorithm for Parameter Optimization of an Air Conditioning System. IEEE Transactions on Industrial Informatics, 16(10), 6669-6679.

19) Asghar, Rafiq, Zahid Ullah, Babar Azeem, Sheraz Aslam, Muhammad Harris Hashmi, Ehtsham Rasool, Bilawal Shaker, Muhammad Junaid Anwar, and Kainat Mustafa. 2022. "Wind Energy Potential in Pakistan: A Feasibility Study in Sindh Province"

20) Saeed, Sarmad, Rafiq Asghar, Faizan Mehmood, Haider Saleem, Babar Azeem, and Zahid Ullah. 2022. "Evaluating a Hybrid Circuit Topology for Fault-Ride through in DFIG-Based Wind Turbines" *Sensors* 22, no. 23: 9314.

21) S. U. Ahmed, M. Affan, M. I. Raza and M. Harris Hashmi, "Inspecting Mega Solar Plants through Computer Vision and Drone Technologies," *2022 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, 2022, pp. 18-23, doi: 10.1109/FIT57066.2022.00014.