HOUSE PRICE PREDICTION - A COMPARATIVE EVALUATION OF SUPERVISED LEARNING

Abstract

In today's time, any business or company would like to know how to predict the next value based on prediction of any data using algorithms or mathematical models have been the statisticians tool. These tools improvised in time and modern computers programs have helped in more efficient ways to predict. Machine learning (ML) is a branch of computer science that enables computers to learn without having to be explicitly programmed. In other words, without being explicitly instructed, ML algorithms can learn from data and better their performance and execution over time.

Speculating on real estate is worth millions of dollars. People are looking to buy houses within their budgets and by analyzing market strategies. There are numerous aspects that must be considered when forecasting house prices for consumers based on their budget as well as their priorities. House price patterns were calculated using dataset, to make predictions for new houses. Various Machine learning algorithms were run on the house price data and compared. The significant contribution is compared between different regression models and found similar and equivalent in results, but Gradient Boosting had a better outcome when compared to other algorithms of Supervised Learning

Keywords: Machine Learning, Supervised Learning, Linear Regression, Multiple Linear Regression, Gradient Boosting.

Authors

Suhas Shastry H.S.
Department of Electronics and Communication Engineering
The National Institute of Engineering
Mysuru Karnataka, India.
2020ec_suhasshastryhs_b@nie.ac.in

Soumyadeep Roy
Department of Electronics and Communication Engineering
The National Institute of Engineering
Mysuru Karnataka, India.
2020ec_soumyadeeproy_b@nie.ac.in

Shiza Mehek S.K.
Department of Electronics and Communication Engineering
The National Institute of Engineering
Mysuru Karnataka, India.
2020ec_shizameheksk_b@nie.ac.in

Priyanka G. Chalikar
Department of Electronics and Communication Engineering
The National Institute of Engineering
Mysuru Karnataka, India.
2020ec_priyankagchalikar_b@nie.ac.in

Anand Srivatsa
Department of Electronics and Communication Engineering
The National Institute of Engineering (NIE), Mysuru
Karnataka, India.
anand.srivatsa@nie.ac.in

Dr. Ananthapadmanabha T
Director
School of Engineering University of Mysuru
Karnataka,India.
drapn2015@gmail.com

I. INTRODUCTION

In machine learning, predictions are the results of an algorithm after it has been prepared and qualified on a dataset and applied to fresh data. The algorithm will produce likely values for an unknown variable for each record in the new data, allowing the model designed to estimate what that value will most likely be. To learn the link between input and output variables, machine learning models are trained using historical data. Once trained, the model can be used to generate predictions on new data. The new data with independent variables is supplied into the model, and the model learns the relationship between the input and output variables to predict the price [1].

House price prediction plays a vital role in the real estate industry and has significant implications for buyers, sellers, investors, and policymakers. Understanding the importance of house price estimates can help individuals and organizations make informed decisions about real estate transactions, investment strategies, and government initiatives.

Accurate house price predictions help in deciding the fair market value of a property. Buyers and sellers rely on these predictions to negotiate prices, ensuring a fair deal for both parties involved in a transaction. Similarly, real estate professionals, such as appraisers and agents, utilize price predictions to provide reliable valuation services. House price predictions are valuable for individuals and organizations engaged in real estate investment. Investors use these predictions to assess the potential returns and risks associated with buying, selling, or holding properties. Accurate predictions help identify investment opportunities, optimize portfolio management, and inform decisions about property acquisition, development, or divestment.

House price predictions assist in managing financial and market risks. Lenders and financial institutions use these predictions to evaluate the risk associated with mortgage loans and determine appropriate lending terms. Additionally, insurance companies, investors, and regulators rely on house price predictions to assess the exposure to potential market downturns and formulate risk mitigation strategies. House price predictions contribute to understanding housing affordability trends. Policymakers, government agencies, and housing advocacy groups use these predictions to monitor and analyze housing market dynamics. They help identify areas with escalating prices, potential housing bubbles, or affordability challenges. This data is critical for formulating policies and actions aimed at providing individuals and communities with affordable housing options [2-5].

House prices have a major impact on the overall economy. Accurate predictions allow policymakers, economists, and analysts to check the health of the housing market and its influence on broader economic indicators. House price predictions can be used as leading indicators of economic activity, influencing consumer spending, construction sector performance, employment trends, and financial market stability. House price predictions assist in urban planning and development initiatives. Local governments and city planners utilize these predictions to assess the demand for housing, identify areas with potential price appreciation or decline, and guide land-use decisions. Predictions can aid in designing sustainable and inclusive communities by considering the future housing needs and market dynamics [6-10].

Significant contributions that can be considered are:

- Predictions of houses in California can be predicted using this tool.

- The accuracy of the model is accurate to what has already been predicted.
- Different algorithms yield the same accuracy except for Gradient Boosting.

The remaining part of the chapter are set as follows, III Supervised Models, IV Materials and Methods, V Experimental Results, VI Conclusion.

II. SUPERVISED MODELS

Machine learning is a subtopic of artificial intelligence that focuses on the development of algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. It involves creating mathematical models and algorithms that can analyse and interpret complex data, identify patterns, and make predictions or take actions established on the patterns identified. Machine learning can be broadly classified into three types: supervised learning, unsupervised learning, and reinforcement learning.

Truong, Quang, and colleagues [5] investigated the impact of characteristics on prediction approaches and compared the performance of various advanced models. The study emphasizes the full validation of numerous strategies in regression model implementation and delivers an optimistic result for home price prediction. Some existing methods rely on textual data, statistical analysis, and machine learning to estimate house values based on characteristics such as square footage, zip code, and number of rooms.

Varma et al. [6] forecast real-time house values in Mumbai and its neighboring areas. Square feet area, number of bedrooms and bathrooms, style of flooring, lift and parking availability, and furnishing condition are all factors evaluated for forecast to forecast the prices.

Park et al. [7] did a comparison between hedonic-based methods and machine learning algorithms, showing the prospective of the latter for price prediction. The study looked at different machine learning performance compared and developed a more accurate housing price prediction model. The experiment involved merging real estate, public school ratings, and mortgage rate data and using four machine learning classifiers.

In supervised learning [12] the algorithm is trained on a labelled dataset where each example is associated with a known target or output. The algorithm learns to map the input data to the correct output by generalizing patterns from the training data. It can then make predictions on new, unseen data established on the learned patterns. To train models, various alternative methodologies or algorithms can be applied. Here are a few examples of Supervised Machine Learning algorithms that could be employed.

Linear regression is a straightforward and commonly employed supervised learning approach. It fits a linear equation to the observed data to model the connection between a dependent variable and one or more independent variables. It is widely used for regression problems with a continuous target variable [13-14].

Decision trees are supervised learning models that can be used for classification as well as regression. They partition the feature space into areas using an if-then-else set of decision rules. The predicted outcome is based on the majority class or mean value of the cases in that region, and each region corresponds to a leaf node in the tree [15].

Another popular supervised learning model is logistic regression. It is used for binary classification issues with a categorical target variable with two classes D.Banerjee et. al. used such methods [16]. Using a logistic function, logistic regression calculates the odds of the target variable belonging to each class.

Adentunji et.al., [17] discusses about Random forests which combine methods for learning that multiple decision trees. Each tree is trained on a randomly selected subset of data to be trained and its features. The final prediction is made by aggregating the predictions of individual trees, typically through majority voting or averaging.

Mora-Garcia [18] includes Support vector machines which are sophisticated supervised learning models that are utilized for classification and regression problems. They want to determine the best hyperplane for separating the data into multiple classes while maximizing the margin between them. The kernel method can also be used by SVMs to address non-linear decision boundaries.

Tchuente et al. [19] predicted housing values, using several machine learning techniques, such as support vector regression, random forest, and gradient boosting, were compared. The authors used real estate data to evaluate the performance of these models and examine the effectiveness of various strategies.

This study investigates the use of deep learning models, specifically convolutional neural networks (CNNs), to predict property prices. The research [20] makes use of image-based features and examines the performance of CNNs in predicting property prices.

This research studies the application of ensemble learning approaches for housing price prediction, such as stacking and bagging. To increase prediction accuracy, the authors mix different models, including support vector regression and random forest.

To estimate housing values, this study presents a hybrid technique that integrates machine learning algorithms with geographic information system (GIS) data. The authors include spatial information and compare the performance of various models, such as decision trees and artificial neural networks. [21] This research focuses on feature engineering techniques and gradient boosting algorithms for house price prediction. The study explores the effect of feature choice and preprocessing on model performance and discusses the effectiveness of gradient boosting in this context.

Our work looked at different models to find the house prices, the first of models used is Linear regression, used to predict a continuous numeric output variable based on one or more input features. Linear regression seeks to discover the coefficients or weights that characterise the linear connection between the input data and the target variable. By modifying the coefficients, the model attempts to minimise the difference between the expected and real values of the target variable. Linear regression can be modified to deal with more complex scenarios, such as polynomial regression or multiple linear regression.

1. **Linear Regression:** Linear regression is a statistical modelling technique that is used to investigate the relationship between a dependent variable (also known as the target or response variable) and one or more independent variables (also known as predictors or features). It assumes that the variables have a linear relationship, which means that the relationship may be represented by a straight line. In linear regression, the house price data comprises of observations for both the dependent and independent variables. Each observation contains the values of the independent variables as well as the value of the dependent variable, the housing price.

   Linear regression makes several assumptions, including linearity (the relationship between variables is linear), independence (observations are independent of each other), constant variance (homoscedasticity), and absence of multicollinearity (independent variables are not highly correlated). Linear regression seeks the best-fitting line that represents the connection between independent and dependent variables. A simple linear regression model with one independent variable has the equation:

   $$\mathbf{y = b_0 + b_1 * x} \quad \rightarrow \textbf{Equation (1)}$$

   Here, y is the dependent variable, x is the independent variable, b0 is the y-intercept (the value of y when x is 0), and b1 is the slope (the change in y corresponding to a unit change in x). The assumption here is that only one variable is considered to predict what would be the next value. The purpose of linear regression is to estimate the coefficient values (b0 and b1) that minimise the gap between the predicted and actual values of the dependent variable in the training data. This is usually accomplished through the use of a method known as ordinary least squares (OLS), which minimises the sum of the squared discrepancies between the anticipated and actual values. Following the estimation of the coefficients, the accuracy of the linear regression model is evaluated using several metrics such as the coefficient of determination (R-squared), which quantifies the proportion of the variance in the dependent variable that can be explained by the independent variables. Mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE) are some more metrics.

   After the model is coached and assessed, new data can be worked to make predictions. Given the values of the independent variables, the model calculates the corresponding predicted value of the dependent variable using the estimated coefficients. Linear regression can be extended to multiple independent variables (multiple linear regression) by adding more additional terms for each independent variable in the equation. The estimation and evaluation steps remain similar to the simple linear regression case.

2. **Multiple Linear Regression:** Multiple linear regression is a simple linear regression extension that allows for the examination of the connection between a dependent variable and multiple independent variables. It seeks the best-fitting linear equation that best describes the connection between the variables. In a nutshell, the data utilised in multiple linear regression comprises of observations of the dependent variable and two or more independent variables. Each observation contains the independent variable values as well as the dependent variable value.

Representation of a Model: The equation represents the relationship between the dependent variable (y) and the independent variables (x1, x2, x3, etc.) in multiple linear regression.:

$$\mathbf{y = c_0 + c_1x_1 + c_2x_2 + c_3x_3 + \cdots} \quad \rightarrow \textbf{Equation (2)}$$

In this scenario, y is the dependent variable, x1, x2, x3, etc., are the independent variables, and c0, c1, c2, c3, etc., are the coefficients or parameters to be estimated (size, beds, baths, size_units,...). The y-intercept is represented by coefficient c0, while the slopes are represented by coefficients c1, c2, c3, and so on (the change in y corresponding to a unit change in each respective independent variable). Multiple linear regression attempts to estimate the coefficient values (c0, c1, c2, c3, etc.) that minimise the difference between the predicted and actual values of the dependent variable in the training data. This is commonly accomplished using the ordinary least squares (OLS) approach, which minimises the sum of squared discrepancies between predicted and actual values. Following the estimation of the coefficients, the validity of the multiple linear regression model is evaluated using several metrics such as the coefficient of determination (R-squared), mean squared error (MSE), mean absolute error (MAE), or root mean squared error (RMSE). These metrics indicate how well the model fits the data and how much volatility in the dependent variable can be explained by the independent variables.

After training and evaluating the model, it can be used to make predictions on new data. Using the calculated coefficients, the model derives the corresponding predicted value of the dependent variable given the values of the independent variables. Numerous linear regression analyses the relationship between numerous independent variables and a dependent variable, allowing the individual contributions and total effects of the independent variables on the dependent variable to be identified. The estimation and evaluation steps in multiple linear regression are like those in simple linear regression, but the model involves additional terms for each independent variable.

3. **Gradient Boosting*:*** Gradient Boosting is a strong machine learning technique that may be applied to both regression and classification problems. It sequentially constructs an ensemble of weak prediction models, often decision trees, to create a stronger predictive model. Gradient Boosting works by iteratively minimizing the errors made by the previous models, thereby improving the overall prediction accuracy. The process starts with an initial prediction model, which can be a simple model like the mean or a small decision tree. This model serves as the "base model" for subsequent iterations.

In each iteration, a new model is added to the ensemble to rectify faults introduced by earlier models. The new model is trained to anticipate the difference between the target variable and the previous ensemble's predictions. During training, Gradient Boosting uses a technique called gradient descent to optimize the new model's parameters. The gradient descent algorithm computes the loss function's gradient with respect to the expected values. It then adjusts the new model's parameters in the aim of minimizing the loss function, eventually reducing prediction errors.

To control the contribution of each new model, a learning rate (or shrinkage parameter) is introduced. The learning rate decides how greatly the ensemble learns from each new model. A smaller learning rate makes the ensemble converge more slowly but can often lead to better generalization. Each new model is added to the ensemble after it has been trained and optimized, and the predictions of all models in the ensemble are merged to generate the final forecast. Averaging the predictions for regression problems or voting/weighted voting for classification problems can be used to combine them.

The iterative process continues until a predefined stopping criterion is met. This can be a maximum number of iterations, reaching a specific level of accuracy, or when the addition of new models no longer improves the performance on the validation set. Gradient Boosting, particularly implementations like XGBoost, LightGBM, or CatBoost, offer several advantages:

- It handles both numerical and categorical features naturally.
- It can capture complex relationships and interactions between variables.
- It handles missing data effectively.
- It can handle large datasets efficiently.
- It provides feature importance rankings, allowing for better understanding of variable contributions.

However, it is important to consider potential overfitting, as Gradient Boosting models can become too complex and over-learn the training data. Regularization techniques, cross-validation, and appropriate hyperparameter tuning can help mitigate overfitting.

3. **Random Forest:** Forest is a joint learning method for making more accurate and robust predictions by combining the predictions of numerous decision trees. It is commonly used for classification as well as regression tasks. Using distinct subsets of the training data, Random Forest generates several decision trees. Bootstrap sampling is used to train each tree on a random sample of the data.

Bootstrap sampling involves randomly selecting data points from the original dataset with replacement, resulting in slightly different subsets for each tree.

Random Forest incorporates unpredictability into the feature selection process in addition to data sampling. Only a random subset of features are examined for splitting at each node of a decision tree. This randomness helps to decorrelate the trees and reduce overfitting. A recursive binary splitting technique is used to build each decision tree in the Random Forest. The tree is built by recursively splitting the data based on the features chosen and their optimal splitting points. The splitting is done using criteria such as Gini impurity (for classification) or mean squared error (for regression), with the goal of creating homogeneous data subsets at each node.

After training the Random Forest and constructing all of the decision trees, predictions are made by aggregating the predictions of individual trees. For classification tasks, the class with the majority of votes from the trees is chosen as the final prediction. For regression tasks, the average or median of the individual tree predictions is taken as the ultimate prediction. Random Forest provides a measure of feature importance, indicating the influence of each feature to the overall predictive power of the model. Feature importance is calculated based on how much the accuracy or impurity decreases when a particular feature is used for splitting across all the trees.

4. **CatBoost:** Dorogush et al. [11] CatBoost is a gradient boosting algorithm that is well-known for its ability to handle category variables and automatically accommodate missing data. It is a gradient boosting framework addition that includes particular strategies for dealing with categorical features. The CatBoost method works by iteratively creating an ensemble of decision trees.

   CatBoost employs a loss function to compute the difference between the predicted and true values of the target variable. The loss function used depends on the job at hand (regression or classification) and can include functions like mean squared error (MSE) for regression or logarithmic loss (logloss) for binary classification. Each instance in the dataset is given an initial prediction by the algorithm. The initial prediction can be a constant value, such as the mean for regression or the log odds for binary classification. CatBoost computes the loss function gradients with regard to the initial predictions. These gradients show the size and direction of the steepest decline towards the best prediction.

   Lundberg et al. [12] Iteratively, CatBoost constructs decision trees. It selects a sample of the training data (through gradient-based sampling) and fits a decision tree to this subset at each iteration. The decision tree is trained to approximate the negative gradients (the loss function's negative derivatives with respect to the original predictions). CatBoost also handles categorical variables using an algorithm called ordered boosting, which is based on permutations of the instances in the leaves of the tree. To arrive at the final prediction, the decision trees are combined. To get the overall forecast, the predictions from all the trees in the ensemble are averaged (for regression) or combined using weighted voting (for classification).

   CatBoost applies various regularization techniques to prevent overfitting, such as depth regularization, learning rate decay, and feature importance calculation. These techniques help improve the model's generalization ability and prevent it from memorizing noisy or irrelevant patterns. While CatBoost does not have a specific equation that governs the entire algorithm, the core principles involve calculating gradients, constructing decision trees, and merging the estimates of multiple trees. The details of the specific equations used in CatBoost can be found in the CatBoost documentation and research papers related to the algorithm [13].

III. MATERIALS AND MODELS

The dataset has been taken from Kaggle.com and show the following information, beds, baths, size, unit size, lot size zip code and finally price. In this the y is the price and the independent variables are the other columns beds, baths etc. These are independent variables, and these independent variables determine the prices of the house.

Table 1: House Price Data [24]

Beds	baths	Size	size_units	lot_size	lot_size_units	zip_code	price
3	2.5	2590	sqft	6000	sqft	98144	795000
4	2	2240	sqft	0.31	acre	98106	915000
4	3	2040	sqft	3783	sqft	98107	950000

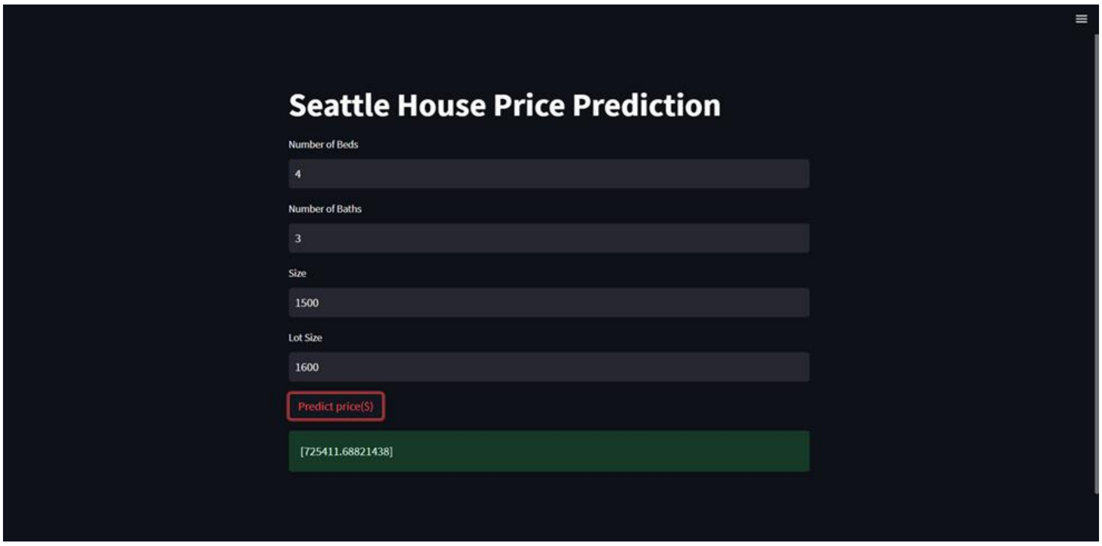
4	3	3800	sqft	5175	sqft	98199	1950000
2	2	1042	sqft			98102	950000
2	2	1190	sqft	1	acre	98107	740000
1	1	670	sqft	6000	sqft	98133	460000
5	3.5	4510	sqft	6000	sqft	98105	3150000
3	2.5	1520	sqft	741	sqft	98108	565000
4	2	2340	sqft	9500	sqft	98178	699000

IV. DEPLOYING THE MODEL

The model used Python 3.x, python libraries pandas, numpy, matplotlib, seaborn, scipy to develop. For specific regression algorithms SciKit, Catboost and Streamlit has been used. Deploying the ML Model can be done in many ways. It can be done using local deployment, cloud-based deployment like Amazon Web Services(AWS), Microsoft Azure, containerization or by using some libraries such as Streamlit. With Streamlit, we create a web application that loads our trained ML interface for users to interact with it. Users can input data, make predictions, and see the results directly in their web browser. It give one to quickly build user interfaces and share your models or data visualizations with others.
To deploy your ML model using Streamlit, you typically follow these steps:

- Install Streamlit: Start by installing Streamlit using pip or conda in the terminal by going to the directory where we are working.
- Run the Streamlit application: Run the Streamlit application script from the command using the streamlit run command:

Before running the streamlit application, we converted the model into a .h5 format which is one of the methods to store large amounts of data. Basically, it converts the model into one file which can be used for model deployments anywhere without the need for any type of code. The .h5 format file now has to be converted into .sav file which is used to make app.py file which holds the logic for writing the streamlit web app through which users can enter their respective choice for number of bathrooms, bedrooms, size and lot size and the web app can predict the house price according to the inputs given by the user.



Seattle House Price Prediction

Number of Beds: 4

Number of Baths: 3

Size: 1500

Lot Size: 1600

[725411.68821438]

Figure 1: House Price Prediction Screen Shot

V. METHODOLOGY

This technique models the relationship between independent variables (features) and a dependent variable (target) by fitting a linear equation to the observed data. Given a data set of n statistical units, a linear regression model assumes that the relationship between the dependent variable y and the vector of regressors x is linear. A Cost Function is used to determine how incorrect the model is in determining a relationship between input and output. It indicates how poorly your model is behaving/predicting. Gradient Descent is a technique used to optimise the cost function or model error. It is used to determine the smallest amount of error in your model. Gradient descent determines the inaccuracy in your model for various input variable values. This is repeated several times until the error numbers get increasingly tiny. Soon, you'll arrive to variable values with the lowest error and the cost function optimised. A straight line is employed to suit the model, and the equation for a straight line is Y = Axe + B.

The cost function for the Linear Regression model will be the model's minimal Root Mean Squared Error, calculated by subtracting predicted values from actual values. The cost function will be the one with the lowest error value.

The first step involved in linear regression is data collection. Gather the dataset that contains the input features and corresponding target variable. Ensure that the dataset is representative and sufficient for training the model. We use the Pandas library for analyzing the data. Pandas is the primary tool data scientists use for exploring and manipulating data.

The next step is Data Preprocessing. Perform data preprocessing steps such as handling missing values, handling outliers, and scaling or normalizing the features. This step aims to clean and prepare the data for further analysis. Here, we eliminate the data which is wrong or duplicated or null or incomplete and convert all the data columns into 1 unit. This step cleans the data.

Splitting the Dataset: Dividing the dataset into training and testing subsets. The training set is used to train the linear regression model, while the testing set is used to evaluate its performance. The average ratio is 80% for training and 20% for testing. The train_test split function divides the dataset into two distinct sets using a random division made in the data set.

VI. EXPERIMENT RESULTS

A correlation matrix is a square matrix that shows the correlation coefficients between pairs of variables. It is a useful tool for understanding the relationships and dependencies between different variables in a dataset. The correlation coefficient, typically denoted as "r," measures the strength and direction of the linear relationship between two variables. It ranges between -1 and 1, where -1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no linear correlation.

A histogram plot, or histplot, is a graphical interpretation of the division of a numerical variable. It displays the frequencies or counts of data points falling into different bins or intervals. They provide info about the shape, central tendency, and spread of a variable, allowing you to understand its distribution characteristics. They are commonly used to visualize and analyse the data before applying machine learning algorithms.

A box plot is a graphical representation of the distribution of a numerical variable or multiple variables across different categories. It identifies any outliers present. Box plots are used to visualize and compare the distribution of variables, detect outliers, and understand the spread and skewness of the data. The boxplot function from seaborn generates the box plot. It visualizes the distribution of the numerical variable(s) across different categories specified by the x parameter. The resulting plot will show the quartiles, median, and any outliers present in each category.

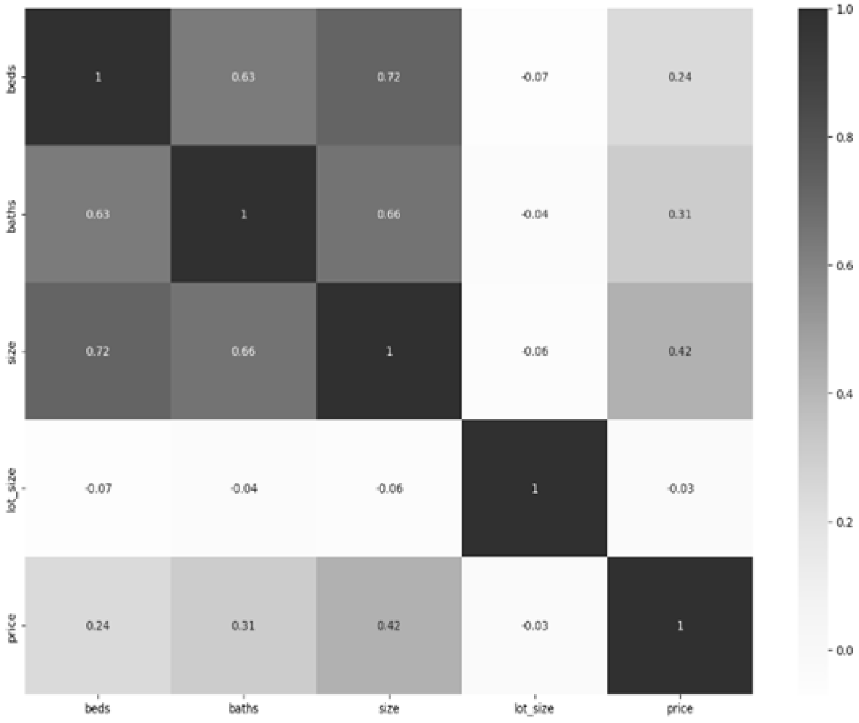


Figure 2: Correlation of House Parameters

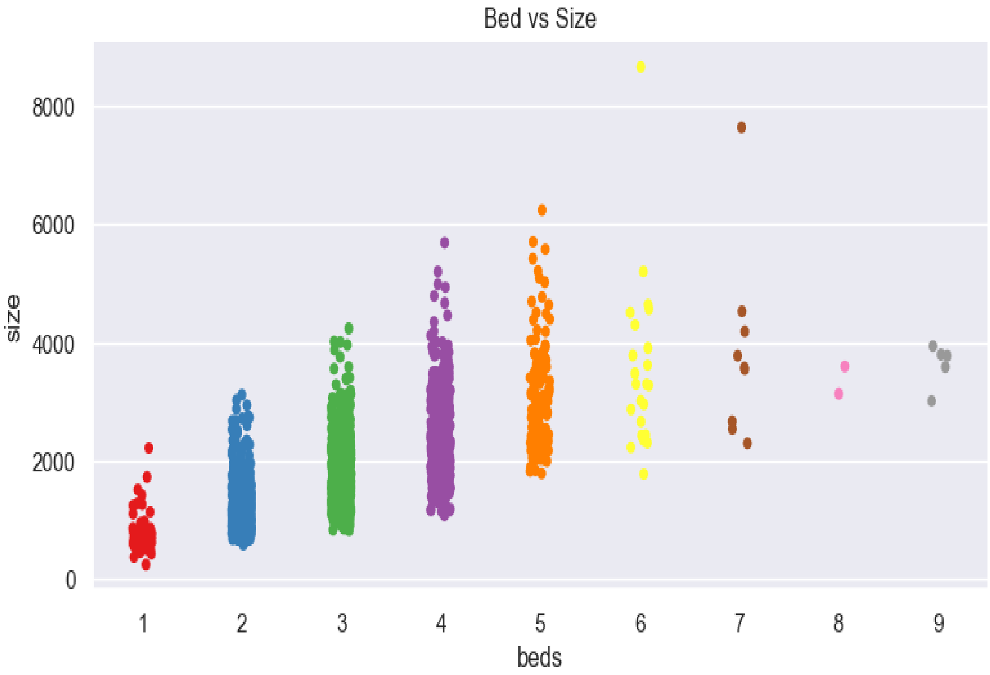


Figure 3: Correlation between Beds and House Size

Scatterplot displays individual data points as dots on a two-dimensional graph, with one variable represented on the x-axis and the other variable on the y-axis. Scatter plots aid in the comprehension of trends, correlations, and data point distribution. Scatter plots are very effective for finding patterns and trends in the connection between two variables.

Gradient descent is then used to train the model. In a machine learning model, a cost function is a mathematical function that measures the inaccuracy or disparity between anticipated and actual values. A cost function's purpose is to measure how well a model performs and to direct the learning process by altering the model's parameters.

$$\boldsymbol{Cost\ function\ J(w,b) = \frac{1}{2m}\sum_{i=0}^{m-1}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)^2} \rightarrow \textbf{Equation (3)}$$

Where $\boldsymbol{f_{w,b}\left(x^{(i)}\right) = wx^{(i)} + b}$

The model seeks to minimise the cost function during training by modifying its parameters using optimisation algorithms such as gradient descent or stochastic gradient descent. The optimization process involves iteratively updating the model's parameters to find the optimal values that minimize the cost function and upgrade the model's performance. With 10000 iterations and a learning rate of 0.001, the model is trained. The coefficient of determination, often known as the R2 score, is used to assess the efficacy of a linear regression model. The extent of fluctuation in the output dependent characteristic that can be predicted by the input independent variable(s). It is used to determine how effectively the model reproduces observed results, based on the ratio of total deviation of results explained by the model. It measures the proportion of the variance in the dependent variable that can be explained by the independent variables. The R2 score ranges from 0 to 1. The R2 Score obtained in this case is 0.59. The meaning of this is 59% of the changeability of the prediction depends on the output attribute can be explained by the model but 41% of the variability cannot be accounted for.

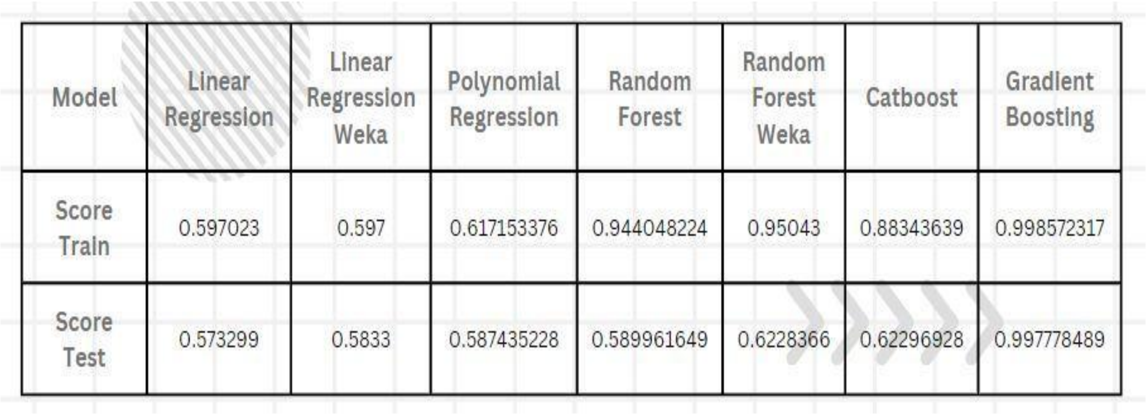
$$\mathbf{R2 = 1 -} \frac{\mathit{Sum\ of\ Squares\ (residual)}}{\mathit{Sum\ of\ Total\ Errors}} \quad \rightarrow \textbf{Equation (4)}$$

The Linear Regression model, as well as a few other approaches such as Polynomial Regression, Random Forest, and Gradient Boosting, were also implemented using a few pre-trained models such as scikit, weka, and catboost, the results of which are analysed and summarised.

VII. CONCLUSION

The Linear Regression model, as well as a few other approaches such as Polynomial Regression, Random Forest, and Gradient Boosting, were also implemented using a few pre-trained models such as scikit, weka, and catboost, the results of which are analysed and summarised.

Table 2: Comparison of Algorithms against Training and Testing the models



Model	Linear Regression	Linear Regression Weka	Polynomial Regression	Random Forest	Random Forest Weka	Catboost	Gradient Boosting
Score Train	0.597023	0.597	0.617153376	0.944048224	0.95043	0.88343639	0.998572317
Score Test	0.573299	0.5833	0.587435228	0.589961649	0.6228366	0.62296928	0.997778489

We ran our model in alternative configurations, such as Weka ® Waikato Environment for Knowledge Analysis, a proprietary software, on this same dataset using multiple Machine Learning models, which gave similar results. The above table shows the Train Score and Test Score. The models evaluated include Linear Regression with a manually implemented version achieving a Train Score of 0.597023 and a Test Score of 0.573299, and a Weka implementation achieving a Train Score of 0.597 and a Test Score of 0.5833, Polynomial Regression with a Train Score of 0.617153 and a Test Score of 0.587435, Random Forest achieving a Train Score of 0.944048 and a Test Score of 0.589962, and a Weka implementation of Random Forest achieving a Train Score of 0.95043 and a Test Score of 0.622837), Catboost with a Train Score of 0.883436 and a Test Score of 0.622969, and Gradient Boosting with a Train Score of 0.998572 and a Test Score of 0.997778. The linear-based models (Linear Regression and Polynomial Regression) showed limited performance, with relatively low scores on both training and testing datasets. In contrast, the ensemble methods (Random Forest, Catboost, and Gradient Boosting) demonstrated significant improvements, with Random Forest performing better than linear models. Among the ensemble methods, Catboost and Gradient Boosting stood out as the top performers, showcasing exceptional generalization capabilities with near-perfect scores on the training dataset and high scores on the testing dataset. Based on these results, it is evident that ensemble methods like Random Forest, Catboost, and Gradient Boosting are more suitable for the dataset compared to linear-based models.

This chapter presents a comparative evaluation of different approaches to the house price forecast challenge using machine learning algorithms. A close to accurate prediction can be made with the help of a trained model like gradient boost. By running the created model prediction, which ever independent variable is given as input the prediction of house price was near to what has been trained and new input was near to the market value.

REFERENCES

[1] Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair - “Housing Price Prediction Using Machine Learning and Neural Networks” 2018, IEEE.

[2] Sifei Lu, Zengxiang Li, Zheng Qin , Xulei Yang , Rick Siow Mong Goh - “A hybrid regression technique for house prices prediction” 2017,IEEE

[3] Park, B., & Bae, J. K. (2015). “Using machine learning algorithms for housing price prediction”. *Expert Systems With Applications*, *42*(6), 2928-2934. Retrieved 7 21, 2023, from https://sciencedirect.com/science/article/pii/s0957417414007325

[4] "A Hybrid Approach for House Price Prediction using Machine Learning and Geographic Information System" by Kumari et al. (2020).

[5] Truong, Quang, et al. "Housing Price Prediction via Improved Machine Learning Techniques." Procedia Comput. Sci., vol. 174, 1 Jan. 2020, pp. 433-42, doi:10.1016/j.procs.2020.06.111.

[6] 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.

[7] Park, Byeonghwa and Jae Kwon Bae. "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data." Expert Syst. Appl., vol. 42, no. 6, 15 Apr. 2015, pp. 2928-34, doi:10.1016/j.eswa.2014.11.040.

[8] Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin, Yandex, “CatBoost: gradient boosting with categorical features support” 14 June 2023, catboost.ai/en/docs/concepts/educational-materials-papers.

[9] Lundberg, Scott and Su-In Lee. "A Unified Approach to Interpreting Model Predictions." arXiv, 22 May. 2017, doi:10.48550/arXiv.1705.07874.

[10] "Machine learning Polynomial Regression - www.javatpoint.com/machine-learning-polynomial-regression. (Accessed on 13 July 2023)

[11] Kaggle Repository https://www.kaggle.com/datasets/samuelcortinhas/house-price-prediction-seattle (Accessed on 14 April, 2023)

[12] Yağmur, Ayten, et al. "House price prediction modeling using machine learning techniques: a comparative study." Aestimum, vol. 81, 2022, doi:10.36253/aestim-13703.

[13] Park, Byeonghwa and Jae Kwon Bae. "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data." Expert Syst. Appl., vol. 42, no. 6, 15 Apr. 2015, pp. 2928-34, doi:10.1016/j.eswa.2014.11.040.

[14] Ho, Winky K. O., et al. "Predicting property prices with machine learning algorithms." Journal of Property Research, vol. 38, no. 1, 2 Jan. 2021, pp. 48-70, doi:10.1080/09599916.2020.1832558.

[15] Truong, Quang, et al. "Housing Price Prediction via Improved Machine Learning Techniques." Procedia Comput. Sci., vol. 174, 1 Jan. 2020, pp. 433-42, doi:10.1016/j.procs.2020.06.111.

[16] D. Banerjee and S. Dutta, "Predicting the housing price direction using machine learning techniques," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 2017, pp. 2998-3000, doi: 10.1109/ICPCSI.2017.8392275.

[17] Adetunji, Abigail Bola, et al. "House Price Prediction using Random Forest Machine Learning Technique." Procedia Comput. Sci., vol. 199, 1 Jan. 2022, pp. 806-13, doi:10.1016/j.procs.2022.01.100.

[18] Mora-Garcia, Raul-Tomas, et al. "Housing Price Prediction Using Machine Learning Algorithms in COVID-19 Times." Land, vol. 11, no. 11, 21 Nov. 2022, p. 2100, doi:10.3390/land11112100.

[19] Tchuente, D., Nyawa, S. Real estate price estimation in French cities using geocoding and machine learning. *Ann Oper Res* **308**, 571–608 (2022). https://doi.org/10.1007/s10479-021-03932-5.

[20] Ho, Winky K. O., et al. "Predicting property prices with machine learning algorithms." Journal of Property Research, vol. 38, no. 1, 2 Jan. 2021, pp. 48-70, doi:10.1080/09599916.2020.1832558.

[21] G. K. Kumar, D. M. Rani, N. Koppula and S. Ashraf, "Prediction of House Price Using Machine Learning Algorithms," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2021, pp. 1268-1271, doi: 10.1109/ICOEI51242.2021.9452820.

[22] Zulkifley, N. H., Rahman, S. A., Ubaidullah, N. H., & Ibrahim, I. (2020). “House Price Prediction using a Machine Learning Model”: A Survey of Literature. *International Journal of Modern Education & Computer Science*, *12*(6).

[23] Monika, R. (2021). “House price forecasting using machine learning methods”. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(11), 3624-3632.

[24] https://www.kaggle.com/datasets/samuelcortinhas/house-price-prediction-seattle/code