Trent University

Comparing the accuracy of Decision Tree, Random Forest, and Gradient Boosting model in predicting housing prices

Doan Phan and Truc Pham

AMOD-5610

Fall 2022

December 09, 2022

Table of Contents

Abstract

The study examines the accuracy of three common machine learning algorithms that were used to estimate the value of real estate. Specifically, the analysis was carried out on the Ames Housing Dataset. The Decision Tree, Random Forest, and Gradient Boosting Regression were used for analysis and the comparisons were made between the three models. Data preprocessing techniques were conducted, and the selected models were trained on preprocessed data. Tuning of hyperparameters including max depth, number of features, and estimators were applied on all three models. Model run time and absolute mean square error were used as evaluation metrics. The results revealed that Gradient Boosting Regression model achieved highest accuracy.

Introduction

The real estate market is considered as one of the many markets with price fluctuations. Predicting the price of real estate as accurate as the market price is important because housing prices increase every year (Monika et al., 2021). Accordingly, this market has been drawing attention from investors, economists, and homeowners (Jha et al., 2020). Understanding the price increase allows investors and homeowners to budget their investments accordingly.

However, understanding the price of real estate can be challenging because it relies on multiple factors that are difficult or impossible to predict (Iwai et al., 2022). Different factors can affect the price of real estate. Some factors are measurable, and some are unmeasurable. Economic factors such as supply, and demand can influence the price. Indeed, Pai et al. (2020) categorized these factors as quantitative and qualitative factors. Examples of quantitative factors are the number of bedrooms, bathrooms, lot size, year built, and zip code (postal code). Qualitative factors such as preferences or lifestyles are more difficult to measure. There are not many measurements available to determine how qualitative factors affect real estate prices (Pai et al., 2020). Thus, it is often difficult for real estate owners to determine when the best time is to buy or sell (Iwai et al., 2022).

Property valuation or usually known as appraisal can be done using different methods. However, it is often done using traditional valuation methods. Traditional methods involve sale price comparison, income rate, and cost context. Nevertheless, parameters used to predict housing prices are changing rapidly or sometimes lack transparency, which may affect the appraisal process (Zaki et al., 2022).

Data availability and the improvement of machine learning algorithms has made property valuation in real estate an easier task (Singh et al., 2020). Indeed, machine learning allows analysts to extract valuable information from large datasets to make predictions. In real estate, machine learning has been used to predict the future prices of a property (Iban, 2022). Machine learning models have been changing the property evaluation process with their accuracy and convenience (Singh et al., 2020). Many machine learning models are utilized in property evaluation due to their high performance and trustworthiness but choosing which model to use is often a debatable topic. Therefore, this work will present the top three common algorithms for predicting housing prices and comparing their performance in terms of accuracy.

The rest of this paper is as follows: (2) An overview of existing literature; (3) Methods used for analysis which includes dataset description and brief theoretical foundation of three machine learning algorithms; (4) Results from analysis; (5) Discussion; (6) Conclusion.

Literature Review

Algorithm selection to predict prices were addressed in many studies. There have been many tools and models developed to predict real estate prices, specifically residential housing prices. Machine learning has been widely used to detect patterns of complicated or large datasets for analysis and predictions. It consists of building a model based on a trained dataset to predict the outcome of new data (Pai et al. (2020).

Home valuation models can be divided into two main categories: traditional methods and modern methods. Traditional models are models that use regression to predict house prices. Examples of these models are the hedonic price model (multiple linear regression) and support vector regression. Modern models include artificial neural networks and gradient boost (Zulkifley et al., 2020).

Traditional models are popular and have a long history in price prediction, but their performance is debatable. Zaki et al. (2022), in his study, revealed hedonic pricing model did not work well with nonlinear relationships. Specifically, multicollinearity between variables and outliers can affect the performance of this model. The hedonic pricing model was believed to be too simple that can produce biased results and underestimated predictions. (Iban, 2022). However, some studies have shown that Multiple Linear Regression (MLR) and Support Vector Regression (SVR) yield good results. MLR can determine which variables have more weight in explaining the dependent variable. Variables such as land size and the number of rooms can be considered as important independent variables for predicting the dependent variable, which in this case is house price (Zulkifley et al., 2020). SVR, which is based on Support Vector Machine, however, can process non-linear results. The model uses a subset of training data which allows it to process non-linear results. Thus, it can overcome small sample sampling errors and avoid over-fitting. In predicting real estate prices, SVR can be used to collect important information on attributes including neighborhood, structural, and locational (Zulkifley et al., 2020).

Conversely, tree-based methods were recommended for the real estate valuation system. A study by Iban (2022) provides a better understanding of Explainable Artificial Intelligence (XAI) in real estate appraisal systems. Nevertheless, the author stated that although artificial intelligence is efficient in evaluating real estate prices, tree-based algorithms are indispensable in the field. In addition, Singh et al. (2020) believed that machine learning models such as linear regression, random forest, and gradient boosting models are suitable for determining the price of a house. These models are efficient for feature selections due to the high number of explanatory variables of housing datasets. In which the authors believed that the Gradient Boosting Models returned the most significant results (Singh et al., 2020).

Tree-based algorithms are more explainable than other modern techniques such as artificial neural networks or kernel techniques (Iban, 2022). Yet these methods are often neglected because of their simplicity although they offer good performance. As evidence by a limited

number of studies that focus on the performance of common tree-based methods of machine learning such as Decision Tree and Random Forest. Based on existing literature, the most common machine learning algorithms that are currently in use to predict housing prices are Decision Tree, Random Forest, and Gradient Boosting Model. The goal of this project is to compare the performance of Decision Tree, Random Forest, and Gradient Boost by applying these models to the same housing dataset.

Methods

This section briefly discussed the chosen dataset, the basics of Decision Tree, Random Forest, and Gradient Boost machine learning models.

Dataset and the proposed analysis

The Ames Housing Dataset, compiled by Dean De Cock, was retrieved from Kaggle. It consists of a total of 81 explanatory variables describing almost all features of residential homes in the city of Ames in Iowa State. The description of all features is illustrated in Appendix 1.

Data preprocessing is essential prior to applying the models. The preprocessing process includes visualizing, finding missing values, changing variable types, encoding categorical variables, and selecting features. Feature selections involve removing variables with a low correlation with the 'SalePrice' variable to reduce the time to fit the model.

Machine learning models entail training and testing datasets. The training dataset is used to train the model whereas testing data is used to make a prediction based on patterns learned from the dataset provided. For this project, 80% of the data was used for training and 20% of the data was used for testing. Specifically, the models will identify patterns and relationships of variables using historical housing data to predict the sale price. To compare the three models, the performance and absolute mean square error of each model will need to be evaluated.

Decision Tree

Decision Tree is a common supervised machine learning algorithm that is suitable for either regression or classification problems. The Decision Tree does not require any assumptions regarding the distribution of data. The algorithm involves two parts: constructing a top-down tree-like structure during the learning process and pruning to reduce the complexity of trees (Kumar et al., 2021).

This entire process is often called a "recursive partitioning process". The partition process starts from the root node, which is the training dataset. Then it continues to split into smaller branches which are called nodes. These nodes are sub-groups that show statistical differences in explanatory variables. Once there is no other significant difference found, the node is considered as a leaf or terminal node. Hence, Decision Tree can identify the most significant dependent variables needed to predict the target variable (Fan et al., 2006).

Random Forest

Random Forest is a supervised machine learning algorithm that utilizes the use of several trees to provide one result. The algorithm is suitable for both classification and regression cases (Singh et al., 2020). Random forest is a type of simple regression tree ensemble that uses the bagging method. At each split of the tree, random predictors are chosen at random from the available predictors (Antipov et al., 2012). Bagging methods can reduce bias in high-variance data (Hegelich, 2016).

However, the inter-tree correlation can reduce the accuracy of the model. Thus, random forest is more successful when there is no or minimal correlation of decision trees (Hegelich, 2016). If similar trees are used in the model, the final random forest result will not be much different from a single decision tree model.

Gradient Boosting

Similar to Decision Tree and Random Forest, Gradient Boosting works well with both classification and regression tasks. It is known for its high accuracy, stability, and flexibility (Iban, 2022). Gradient Boosting is considered as a variant of ensemble methods, where weak prediction models are "boosted" to achieve better performance. In other words, it combines weak learners by adding one at a time and transforming them into one single strong learner (Nasiboglu et al., 2022).

Gradient Boosting Regressor (GBR) is similar to the regression model. It encompasses the concept of residuals, which is the difference between the predicted value and the actual value. GBR improves a model prediction by mapping features to the residual and repeating this step multiple times. Decision Trees are often used as weak learners in GBR (Nasiboglu et al., 2022).

Hyperparameter Tuning

Hyperparameter tuning is a decision often made by machine learning users to choose a set of values that optimize results. Identifying the appropriate hyperparameters that influence the performance of a machine learning algorithm, as well as determining the appropriate values for these parameters is a challenging task (Kumar et al., 2021).

Tuning a Decision Tree model means identifying the minimum number of samples required at a leaf node. This can be achieved by setting the maximum depth of the tree or identifying the splitting criterion (Kumar et al., 2021). Max depth (max_depth) means how deep the tree can be or the number of levels of the tree from the root node to the leaf node. Max features (max_features) are the number of features requires to have the best split (Kumar et al., 2021).

For the random forest model, it is important to have uncorrelated decision trees in random forests. The random forest can achieve this by utilizing bootstrapping to create the randomness of features (Hegelich, 2016). Feature randomness can be achieved by setting the size of random subsets off features (max_features) and the number of trees in the forest (n_estimators). Increasing the number of max_features can reduce variance. Increasing n_estimators can improve the results. However, increasing the number of trees does not always improve the model.

In Gradient Boosting Regressor, as the number of iterations increases, the model achieves higher accuracy. However, when the number of iterations is too large, the model can become overfitted (Nasiboglu et al., 2022). Boosting can be controlled using sets of parameters, n_estimators and max_depth similar to Random Forest and Decision tree. However, learning_rate need to be considered to control shrinkage. Lower learning_rate values create a well generalized model as they make the model more robust to the specific characteristics of trees (Ho et al., 2021).

Results

Data Preprocessing

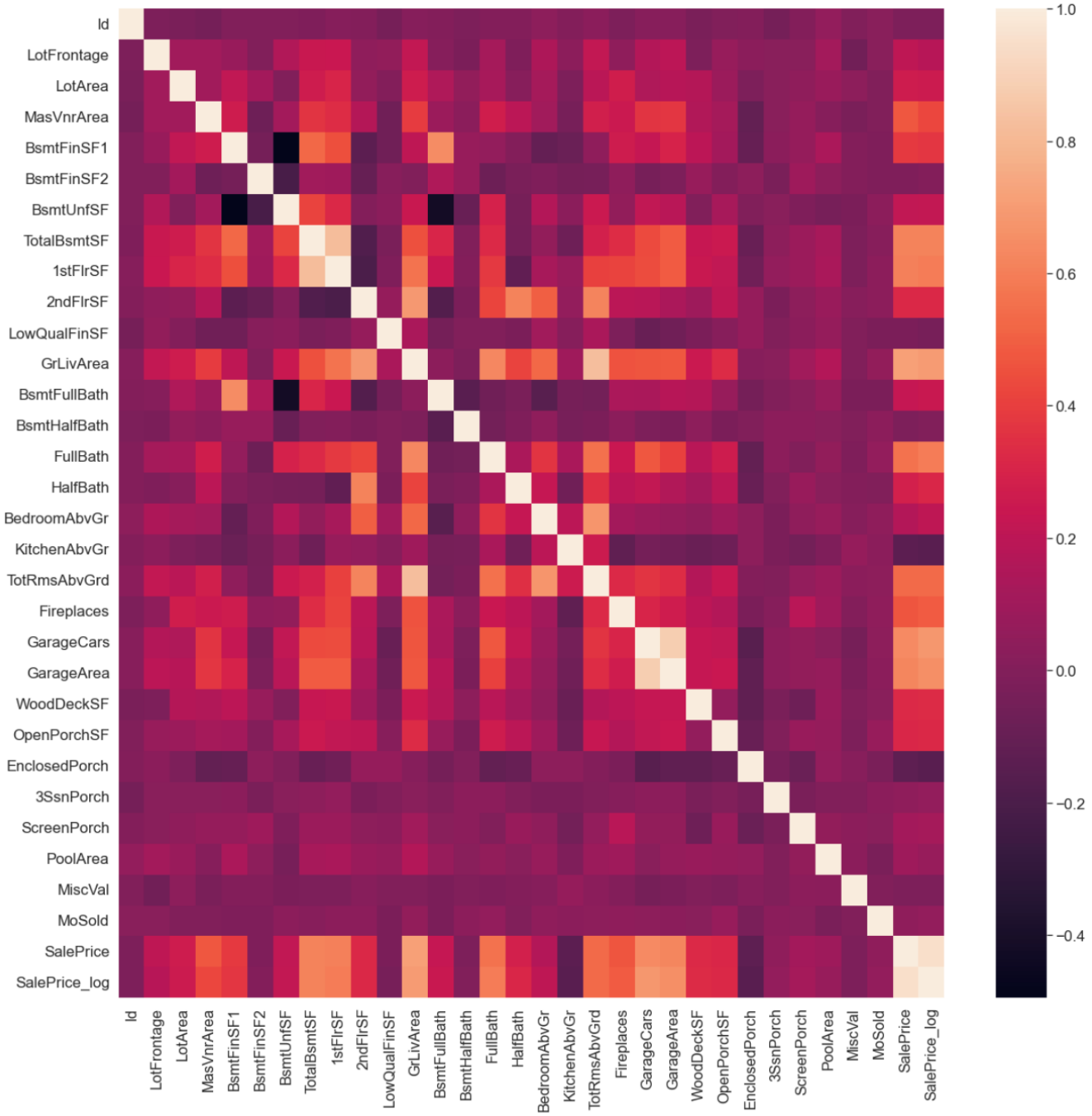
As far as missing values are concerned, attributes with missing values were carefully examined and replaced with related values. Specifically, attributes related to features that do not always come with a home including a pool, garage, alley access, fence, fireplace, and basement were assumed as not included in the property. Hence, these attributes were given a value such as "No Pool", "No Feature", "No alley access", "No Fireplace", "No Garage", "No Basement", or "Not Applicable". In addition, attributes containing measurements of those values were assigned a value of zero. Specifically, some basement values were NANs where they should have been zeros. Thus, NANs values were replaced with 0s.

With respect to variable types, some variable types were changed to meet the purpose of the prediction. Particularly, the year built (original construction date), year sold, year garage was built, and remodel date (or construction date if there is no removal date) need to be treated as categorical variables. Consequently, they were converted from integer to string variable type.

In addition, values of some categorical variables were decoded to avoid values mix up between variables. Variable 'MSSUbCLass', which identifies the type of dwelling in the sale was assigned a specific number. For instance, 90 indicates Duplex properties and 60 indicates two-story properties built in 1946 or newer. Furthermore, the variable 'Overallcond' indicates the overall condition of a property. The overall condition of property was ranked from one to ten, in which one indicates a very poor condition and 10 indicates a very excellent condition. Similarly, 'OverallQual' ranks the overall material and finish of the house from one to ten. The three variables mentioned above were decoded and replaced with associated string values.

On the other hand, the correlation of variables was also examined using the Heat Map. To reduce the fitting time, variables correlated with 'Saleprice' less than the threshold of 0.15 were omitted. As a result, the following 11 variables were dropped from further analysis: Id, BsmtFinSF2, LowQualFinSF, BsmtHalfBath, KitchenAbvGr, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, MiscVal, MoSold. These variables were related to basement measurements, porch area measurements, pool area, the value of miscellaneous, and month sold. The following figure (Figure 1) shows the correlations of independent variables.

Figure 1. Correlation map



Model Fitting and Tuning

Hyperparameter tuning with Grid search with 5 CV was employed for each model to minimize model error.

Decision Tree

Tuning of Decision Tree included setting parameters on max_depth with increment of 5 from 5 to 30, and max_features with increment of 0.1 from 0.1 to 0.7. Altogether there were 42 combinations available for fitting random search. The total runtime for tuning was 7 seconds. With 42 combinations, the average fitting time for each was 0.16 seconds. Best performance was found with score of -0.15214090866615362 when hyperparameters were {'max_depth': 15, 'max_features': 0.7}.

Random Forest

Tuning of Random Forest included setting hyperparameters n_estimators with increment of 100 from 100 to 600 and max_features with increment of 0.1 from 0.1 to 0.7. The total combinations of settings were 42 which resulted in the total tuning time of 145 seconds. The average fitting time was 3.45 seconds. Best performance was found with score of -0.09966299436802066 when the hyperparameters were {'max_features': 0.2, 'n_estimators': 300}.

Gradient Boosting Regression

Tuning of Gradient Boosting Regression involved three parameters n_estimators, max_depth, and learning_rate. The parameters settings were as follow: n_estimators with increment of 100 from 100 to 600, max_depth with increment of 5 from 5 to 30, learning_rate of 0.01, 0.05, and 0.1, which resulted in 108 possible combinations. The total runtime for tuning was 605 seconds. Accordingly, the average fitting time was 5.60 seconds. Best parameters were found when the combination of parameters were {'learning_rate': 0.05, 'max_depth': 5, 'n_estimators': 600}, in which the best score was -0.08930796342548727.

Table 2 below summarizes the results obtained by each model after tuning:

Table 2. Performance comparison of each model

Models	Average Runtime (seconds)	Best Score
Decision Tree	0.16	-0.15214090866615362
Random Forest	3.45	-0.09966299436802066
Gradient Boosting	5.60	0.08930796342548727

Discussion

Based on the results, Decision Tree had the shorter fitting time compared to Random Forest, although they both had the same number of combinations. In contrast, Gradient Boosting Model had the longest runtime. Indeed, Random Forest took longer time to run than Decision Tree due to the number of trees in random forest. As the number of trees increases, the training time also increases. Similarly, Gradient Boosting is supposed to be an improved model Decision Tree and Random Forest, therefore it took much longer time than the other two algorithms.

Negative Mean Absolute Error in machine learning refers to the magnitude difference between the predicted value of an observation and the real value of that observation. The best score of each model refers to the accuracy score of the best combination of hyperparameter of each model. Decision Tree had the lowest best score, whereas Gradient Boosting had the highest score. Accordingly, Gradient Boosting Model outperformed the Decision Tree and Random Forest models in predicting house prices.

Although the three models revealed good results, there were some factors that could have affected the results. First, the size of the dataset was small. The number of observations in the dataset could affect the training and testing process. The result could have been better if the model were trained with more observations. Second, different data preprocessing techniques such as Principal Component Analysis (PCA) can be used. The dimension of data can significantly change the fitting time of the model. Lastly, the parameters chosen for tuning each model can yield different results. The number of hyperparameter chosen for tuning or which hyperparameter need to be selected does affect the results.

As mentioned in the previous section, several existing studies on this topic have shown results that Gradient Boosting produces a high-accuracy model. However, there are many other regressions and tree-based models that would need to be examined. For example, Support Vector Machine (SVM) or Support Vector Regressions (SVR) also have been known for their comparable predictive power. Therefore, it is difficult to conclude the best model because model performance can be affected by different factors. For future work, it is recommended to consider other machine learning models such as SVM or SVR.

Conclusion

Machine learning has provided a powerful tool to discover hidden patterns of large and complex datasets to predict future outcomes. With machine learning, time-consuming and challenging tasks become feasible. In the field of real estate, data is often complex and involve many explanatory variables. Thus, machine learning has become a common tool used in the real estate field. Specifically, machine learning algorithms are being used to predict housing prices. However, which model is best to use for house price prediction is a debatable topic.

This project attempted to use three machine learning algorithms: Decision Tree, Random Forest, and Gradient Boost Regression to estimate housing prices using the Ames Housing Dataset. The performance of each model was compared. Each model was tuned using appropriate hyperparameters. Tuning Decision Tree includes setting parameters max_depth and

max_features, whereas n_estimators and max_features was set for Random Forest. Hyperparameters, max_depth, n_estimators, and learning_rate, were tuned for Gradient Boosting Regressor. The study revealed that Gradient Boosting Regressor yielded the highest accuracy of all three models. However, it is difficult to conclude the best model to predict housing price as there are many factors that can affect the results. Future works on this topic are recommended to consider different models.

References

Antipov, Evgeny A., and Elena B. Pokryshevskaya. “Mass Appraisal of Residential Apartments: An Application of Random Forest for Valuation and a Cart-Based Approach for Model Diagnostics.” *Expert Systems with Applications*, vol. 39, no. 2, 2012, pp. 1772–1778., https://doi.org/10.1016/j.eswa.2011.08.077.

Fan, Gang-Zhi, et al. “Determinants of House Price: A Decision Tree Approach.” *Urban Studies*, vol. 43, no. 12, 2006, pp. 2301–2315., https://doi.org/10.1080/00420980600990928.

Hegelich, Simon. “Decision Trees and Random Forests: Machine Learning Techniques to Classify Rare Events.” *European Policy Analysis*, vol. 2, no. 1, 6 Mar. 2016, pp. 98–120., https://doi.org/10.18278/epa.2.1.7.

Ho, Tang, B.-S., & Wong, S. W. (2021). “Predicting property prices with machine learning algorithms”. *Journal of Property Research*, *38*(1), 48–70. https://doi.org/10.1080/09599916.2020.1832558

Iban, Muzaffer Can. “An Explainable Model for the Mass Appraisal of Residences: The Application of Tree-Based Machine Learning Algorithms and Interpretation of Value Determinants.” *Habitat International*, vol. 128, Oct. 2022, p. 102660., https://doi.org/10.1016/j.habitatint.2022.102660.

Iwai, Koichi, and Tomoki Hamagami. “A New Xgboost Inference with Boundary Conditions in Real Estate Price Prediction.” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 17, no. 11, 2022, pp. 1613–1619., https://doi.org/10.1002/tee.23668.

Jha, Shashi Bhushan, et al. “Housing Market Prediction Problem Using Different Machine Learning Algorithms: A Case Study.” *ArXiv.org*, 17 June 2020, https://arxiv.org/abs/2006.10092.

Kumar, Sunil, et al. “Hyper Heuristic Evolutionary Approach for Constructing Decision Tree Classifiers.” *Journal of Information and Communication Technology*, vol. 20, no. Number 2, 2021, pp. 249–276., https://doi.org/10.32890/jict2021.20.2.5.

Nasiboglu, Resmiye, and Efendi Nasibov. “FyzzyGBR—a Gradient Boosting Regression Software with Fuzzy Target Values.” *Software Impacts*, vol. 14, 2022, https://doi.org/10.1016/j.simpa.2022.100430.

Pai, & Wang, W.-C. (2020). “Using Machine Learning Models and Actual Transaction Data for Predicting Real Estate Prices”. *Applied Sciences*, *10*(17), 5832–. https://doi.org/10.3390/app10175832

Singh, Sharma, A., & Dubey, G. (2020). “Big data analytics predicting real estate prices”. *International Journal of System Assurance Engineering and Management*, *11*(Suppl 2), 208–219. https://doi.org/10.1007/s13198-020-00946-3

Zaki, John, et al. “House Price Prediction Using Hedonic Pricing Model and Machine Learning Techniques.” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 27, 2022, https://doi.org/10.1002/cpe.7342.

Zulkifley, Nor Hamizah, et al. “House Price Prediction Using a Machine Learning Model: A Survey of Literature.” *International Journal of Modern Education and Computer Science*, vol. 12, no. 6, 2020, pp. 46–54., https://doi.org/10.5815/ijmecs.2020.06.04.

Appendix

Appendix 1. Feature description

Feature	Description
MSSubClass	Identifies the type of dwelling involved in the sale
MSZoning	Identifies the general zoning classification of the sale
LotFrontage	Linear feet of street connected to property
LotArea	Lot size in square feet
Street	Type of road access to property
Alley	Type of alley access to property
LotShape	General shape of property
LandContour	Flatness of the property
Utilities	Type of utilities available
LotConfig	Lot configuration
LandSlope	Slope of property
Neighborhood	Physical locations within Ames city limits
Condition1	Proximity to various conditions
Condition2	Proximity to various conditions (if more than one is present)
BldgType	Type of dwelling
HouseStyle	Style of dwelling
OverallQual	Rates the overall material and finish of the house
YearBuilt	Original construction date
YearRemodAdd	Remodel date (same as construction date if no remodeling or additions)
RoofMatl	Roof material
Exterior1st	Exterior covering on house
Exterior2nd	Exterior covering on house (if more than one material)
MasVnrType	Masonry veneer type
MasVnrArea	Masonry veneer area in square feet
ExterQual	Evaluates the quality of the material on the exterior
ExterCond	Evaluates the present condition of the material on the exterior
Foundation	Type of foundation
BsmtQual	Evaluates the height of the basement
BsmtCond	Evaluates the general condition of the basement
BsmtExposure	Refers to walkout or garden level walls
BsmtFinType1	Rating of basement finished area
BsmtFinSF1	Type 1 finished square feet
BsmtFinType2	Rating of basement finished area (if multiple types)
BsmtFinSF2	Type 2 finished square feet
BsmtUnfSF	Unfinished square feet of basement area
TotalBsmtSF	Total square feet of basement area
Heating	Type of heating
HeatingQC	Heating quality and condition
CentralAir	Central air conditioning
Electrical	Electrical system
1stFlrSF	First Floor square feet
2ndFlrSF	Second floor square feet
LowQualFinSF	Low quality finished square feet (all floors)

GrLivArea	Above grade (ground) living area square feet
BsmtFullBath	Basement full bathrooms
BsmtHalfBath	Basement half bathrooms
FullBath	Full bathrooms above grade
HalfBath	Half baths above grade
Bedroom	Bedrooms above grade (does NOT include basement bedrooms)
Kitchen	Kitchens above grade
KitchenQual	Kitchen quality
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
Functional	Home functionality (Assume typical unless deductions are warranted)
Fireplaces	Number of fireplaces
FireplaceQu	Fireplace quality
GarageType	Garage location
GarageYrBlt	Year garage was built
GarageFinish	Interior finish of the garage
GarageCars	Size of garage in car capacity
GarageArea	Size of garage in square feet
GarageQual	Garage quality
GarageCond	Garage condition
PavedDrive	Paved driveway
WoodDeckSF	Wood deck area in square feet
OpenPorchSF	Open porch area in square feet
EnclosedPorch	Enclosed porch area in square feet
3SsnPorch	Three season porch area in square feet
ScreenPorch	Screen porch area in square feet
PoolArea	Pool area in square feet
PoolQC	Pool quality
Fence	Fence quality
MiscFeature	Miscellaneous feature not covered in other categories
MiscVal	$Value of miscellaneous feature
MoSold	Month Sold (MM)
YrSold	Year Sold (YYYY)
SaleType	Type of sale
SaleCondition	Condition of sale