Price Prediction of Ames Housing Through Advanced Regression Techniques

Yueting Han[1, *]

[1]Department of Statistics and Data Science, University of California, Santa Barbara, California, USA

*Corresponding author: yuetinghan@ucsb.edu

Abstract. House price fluctuates at all times, and it is necessary to predict the future house prices for the reason that it is crucial for buyers, property investors and the economy as a whole. This article reveals the important features relating to house price and demonstrates the method of predicting house prices in Ames, Iowa through advanced regression techniques, such as correlation, feature engineering, and model building. The prediction is performed through machine learning methods based on the dataset containing almost all the features of residential houses in Ames, Iowa. The dataset was originally splitted into the 'train set' and the 'test set'. Six models were chosen, LASSO Regression, Elastic Net Regression, Gradient Boosting Regression, XBoost, LightGBM, and Stacked Model, for evaluation under the five-fold cross-validation to check the performance of the algorithms and report the root-mean-square logarithmic error. Eventually, Gradient Boosting Regression Model is chosen by the lowest root-mean-square logarithmic error of 0.04615 and a low cross-validated value of 0.1174 to predict the house price to benefit buyers, investors, and the whole society.

Keywords: Price prediction; feature engineering; LASSO Regression; Elastic Net Regression; Gradient Boosting Regression; XGBoost; LightGBM.

1. Introduction

With the fast formation and development of cities and the increasing immigrants in the U.S, the process of urbanization is fastened, yet causing a series of new obstacles. One of the significant obstacles is the constant rise of house prices due to the growing population density. According to Chenhao Zhou, the average price of houses was roughly $27,000 in the 1970s, yet the same amount of money cannot even get a double bedroom in some less developed states today [1]. In consequence, it is markedly vital to predict house prices.

When people make a house transaction, to make it either a residential house or an investment property, this is referred to as the housing market. The trend of house prices has considerable effects on economy as a whole. In other words, the housing market is closely related to consumer spending. The increasing trend generally improves confidence and vitalizes greater consumer spending. The house owners would be more willing to consume more money on daily goods, housing services, renovation, and exterior or interior decorations, which paves the way for stimulating gross domestic product (GDP) and economic growth. Nevertheless, the decreasing trend can hurt consumers' confidence, since homeowners are afraid that the property will be less worthy than its outstanding mortgage. Therefore, people would be far more likely to cut down on daily spending and resist making additional investments, which can cause a possible economic recession.

House price prediction is also crucial for both buyers and investors in many aspects. It provides buyers with more information about the current housing market in order to benefit their future finance plan, and general concepts about what should be the important features of a dream house buyers should not neglect. Most importantly, it tells buyers the right time to purchase a house. Meanwhile, it gives suggestions to the investors to determine the sale price of a property and offers property investors the tendency of housing prices in a certain location to prevent risky investments. Being such an important property in people's lives, not everyone is an expert in choosing their dream house or a potential investment.

A home buyer's acknowledgement of a residential house can be limited. The buyers can easily conclude a pool, number of bedrooms, and a large living room as important features of a potential dream house. A buyer might not be able to describe their dream house with miscellaneous but essential features like the proximity to a south-west railroad, the exterior covering on the house or the quality of the garage. Thus, this article proves that there are much more crucial features that highly influence the sale price.

There are several approaches for predicting the house price, and one of them is the time-series analysis approach. Similar to the regression analysis, time series analysis is a unique method to forecast the future based on a function of previous data points. However, the recorded data points are collected over a consistent period instead of the data points that are recorded randomly or inconsistently. This prerequisite excludes many house pricing datasets, since time is not always well-included. Time series approach is used when previous data points are available, and when it is practicable to assume that some of the previous pattern shown in the past can be maintained in the future. This leads to another important problem that, as one of the most valuable properties, houses' sale prices are changing at all times. Different from other goods and services, houses do not have a stable unit price. Thus, houses are not only priced based on intervals like month or quarter, but also based on many other related features. Because of the specialty of the chosen dataset and house prices with so many influential features, the time series model cannot be sufficient to predict house price.

Instead of heavily relying on one variable, like time series relies on time, regression techniques are an approach which looks for the relationship between the target variable and its several potential-related independent variables. It is a reliable statistical method of identifying which variables have impacts on the target variable. This process helps people better determine which variables are more important among the dataset, which factors can be mostly ignored, and how these variables can be affecting each other. The best subset of variables outputted from the regression techniques may create the best fit model to make the final prediction with a small root-mean-square logarithmic error. Regression is one of the most basic and easiest concepts in machine learning, but it is powerful in model building and prediction.

The dataset is originally from a Kaggle competition, "House Prices: Advanced Regression Techniques", and is splitted into two sets, the train set and the test set for further analysis and evaluation. The goal of this project can be generally summarized into two points, the first goal is to find out the truly related features that might cause fluctuations of the house price; secondly, the goal is to build models for evaluation and to predict the value of the sales price for each house under each 'Id' in the test set.

2. Literature Review

With the aim of finding the best model, analysts devoted themselves into looking for the most suitable subset of features. There are quite a few features that affect house prices. In the research by Rahidi.et al, the group of researchers categorized these potential features into three major groups, physical condition, concept, and location [2]. The category of Physical conditions are features possessed by a property which can be observed by people, such as the number of bedrooms, the area of the living room, and the year when the house is built. The category of concept contains the ideas provided by developers which are used to attract potential buyers such as stylish home and healthy environments [3]. Lastly, the features included in the category of location are regarded as one of the most significant factors in affecting the house price. The reason is that the location decides the land price [4]. Additionally, the location also includes hedonic factors like the access to hospitals, schools, malls, railway stations [5, 6].

After reviewing others' work, the chosen features for this project can also be easily categorized into three types: 'building', which includes the variables that shows the physical features of the architecture (e.g. 'OverallQual'); 'space', which represents the features that demonstrates the space properties of the house (e.g. 'TotalBsmtSF'); 'location', which means the features that impliesthe

position and environment of where the house is located (e.g. 'Neighborhood'). There are many guesses on whether 'location' features might be the top factor that affect house prices. This article will look for the strong features for further predictions.

3. Procedure

This house price prediction project will be using Python notebook with these main packages: 'NumPy' for working with arrays, 'pandas' for data processing, 'matplotlib' and 'seaborn' for exploratory data analysis and data visualizations, 'SciPy' for statistics like normality and skewness, and 'scikit-learn' for building and evaluating the machine learning models. The basic procedures are shown in Figure.1 below.

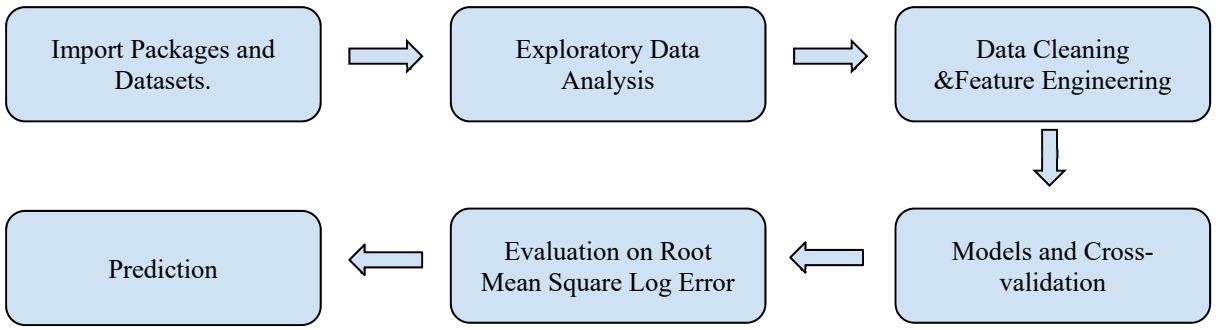

Figure 1. Flow chart of procedures

3.1 Dataset

The Ames House Price dataset is chosen from Kaggle and already categorized into two sets. The train set contains 1460 houses under different 'Id's, and 81 features including the target variable 'SalePrice' for the machine to train and learn, in order to predict the 'SalePrice' of the 1459 houses in the test set with 80 other features. There are two types of variables: 'numerical' and 'categorical'. The 'Id' column is set as the index, since it is helpful for recording but not a feature for house prices.

3.2 Exploratory Data Analysis

In Order to understand more about the dataset, the features' meanings and their relevance to the target variable 'SalePrice' should be paid attention to. A couple of questions should be kept in mind when examining the features, 'What are the features people usually look at when buying a house?', 'which ones are more important?', and 'Is this feature already described in any other variables?'. After that, the hypothesis is created that 'OverallQual', 'YearBuilt', 'TotalBsmtSF', 'GrLivArea' might play a significant role in predicting house prices. These four features are two 'space' variables and two 'building' variables as categorized above, which is kind of unusual since 'location' variables are usually more expected. Thus, scatter plots and boxplots, shown in Figure.2, are used to better visualize the relevance between these variables and 'SalePrice'.

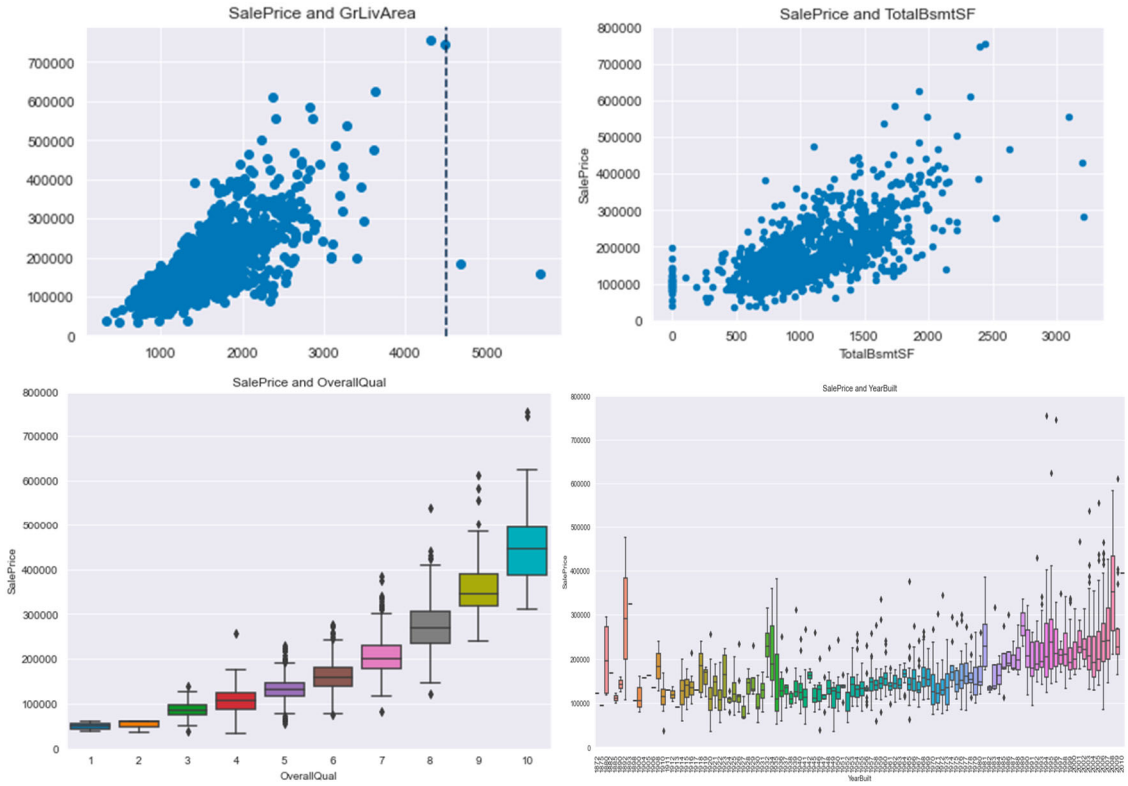

Figure 2. Part of visualization between variables and ‘SalePrice’

As a result, 'GrLivArea' and 'TotalBsmtSF' each show a linear relationship with 'SalePrice'. Both are positively related which implies that when 'GrLivArea' or 'TotalBsmtSF’ goes up, the ‘SalePrice’ also increases. 'OverallQual' and 'YearBuilt' also demonstrate relationships with 'SalePrice'. The relationship is obvious in the boxplotwith 'OverallQual', which shows how sales prices goes up with the increasing level of overall quality.

Visualizing is also a great way to find outliers, which can significantly affect the models. In the case of 'GrLivArea', the first scatter plot in Figure.2, shows two values with high 'GrLivArea' have low ‘SalePrice’ compared to the majority of the data points. They are considered as the outliers since they cannot represent the case well. As a result, they are dropped for further analysis.

After examining the important features, the next step is to take a look at the target variables ‘SalePrice’. Demonstrated by the first distribution plot and the probability plot presented in Figure.3, as a right skewed feature is shown, the ‘SalePrice’ is not normally distributed, which can disrupt the models by outliers. The undesirable skewness will be discussed more later in the data cleaning section. One of the methods demonstrated in equation (1), the log transformation, simply applies log on the variable and is usually used when the variable has a positive skewness and a few existences of high values which suits this case well.

$$Y(s) = \ln\big(Z(s)\big), Z(s) > 0 (1)$$

Thus, a log transformation is applied as to make the ‘SalePrice’ normally distributed, as the last two plots shown in Figure.3.

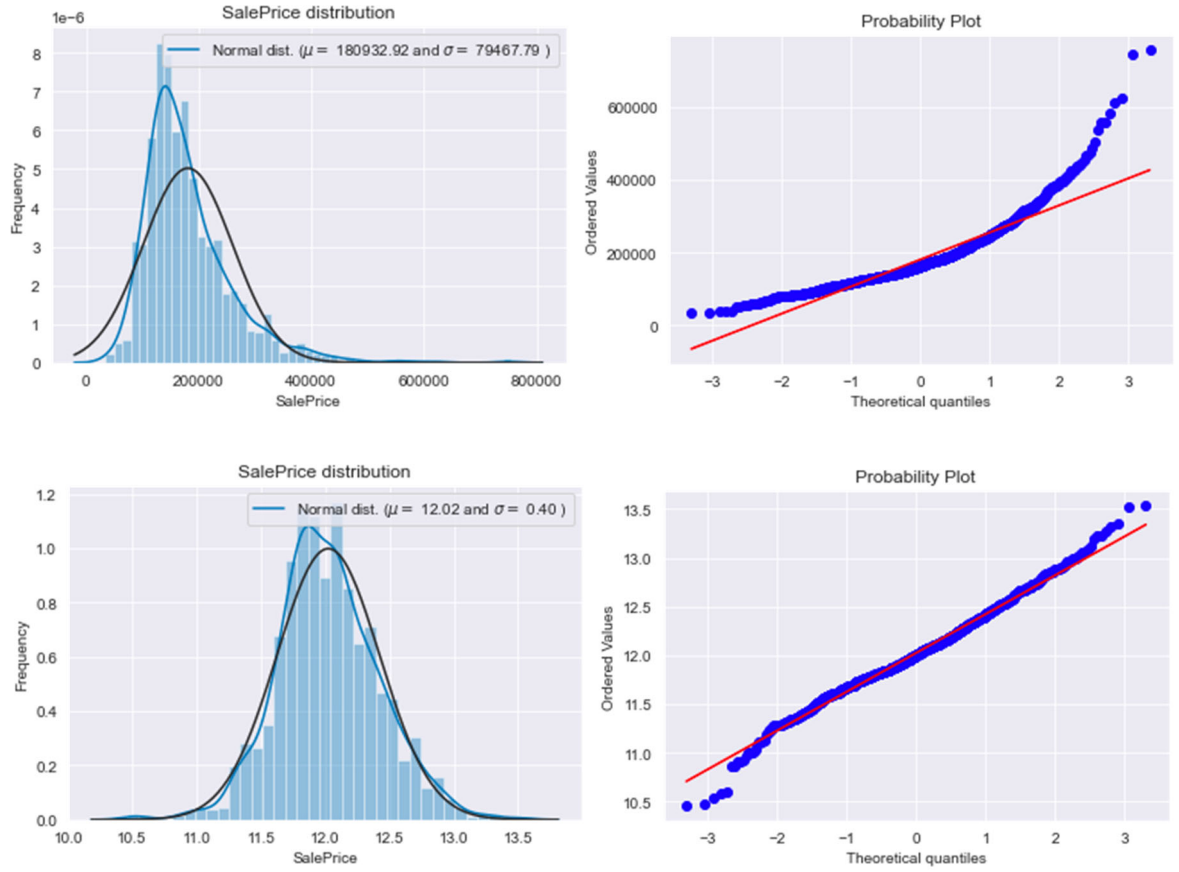

Figure 3. Plots of 'SalePrice' before and after log transformation

3.3 Data cleaning andfeature engineering:

The train set and the test set are concatenated as 'all_data' with the dimension of 2917 houses with different 'Id' and 79 features with the drop of 'Id' and the target variable 'SalePrice', and the 'SalePrice' is separately recorded as 'y_train' for further data cleaning. Now, it is much more convenient to clean the data in both sets by 'all_data'.

The first step is to look for the Pearson correlations between variables as a whole and between variables and the 'SalePrice'. Pearson correlation, shown in equation (2), is the most widely used statistical tool that represents the degree of how variables are related without a statement of cause and effect through a heatmap.

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{[n\sum x_i^2 - (\sum x_i)^2][n\sum y_i^2 - (\sum y_i)^2]}} \quad (2)$$

The top 10 correlated variables to 'SalePrice' is demonstrated in Figure.4. The features visualized above are all included, which ensures their importance to the 'SalePrice'. Additionally, multicollinearity is a big problem to cope with. It is the situation that indicates two variables are highly correlated which can cause inaccuracy in prediction since variables are no longer independent and can undermine the statistical significance of an independent variable. In order to reduce multicollinearity by dropping the variables that are highly correlated to another, heatmap is used again to detect multicollinearity through finding the correlations between variables. As shown in Figure.5, 'GarageCars' and 'GarageArea' are strongly correlated to each other. This is understandable because the number of cars that fit into the garage depends highly on the garage area, and vice versa. Therefore, only 'GarageCars' will be kept for further analysis, since it has a comparatively higher correlation with 'SalePrice'. 'TotalBsmtSF' and '1stFlrSF', 'TotRmsAbvGrd' and 'GrLivArea' are also groups of

high correlations. Thus, including the useless features, 'MasVnrType', 'MasVnrArea', and 'Utilities', a total of six variables are deleted from 'all_data'.

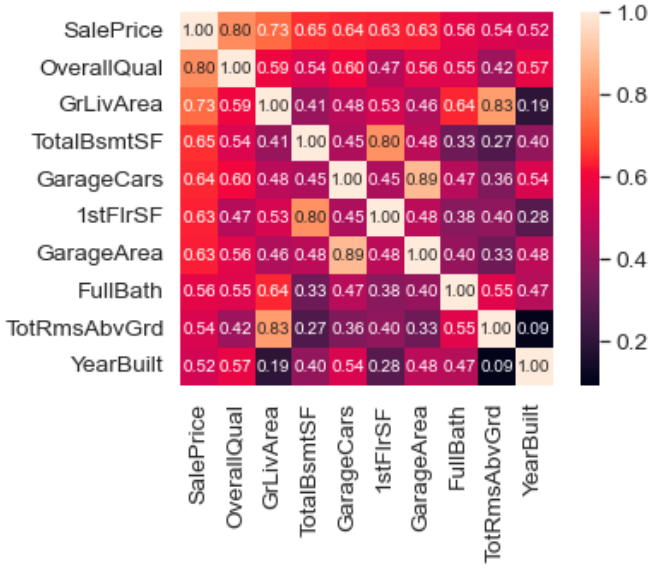

Figure 4. Heatmap of the top 10 most correlated variablesto 'SalePrice'

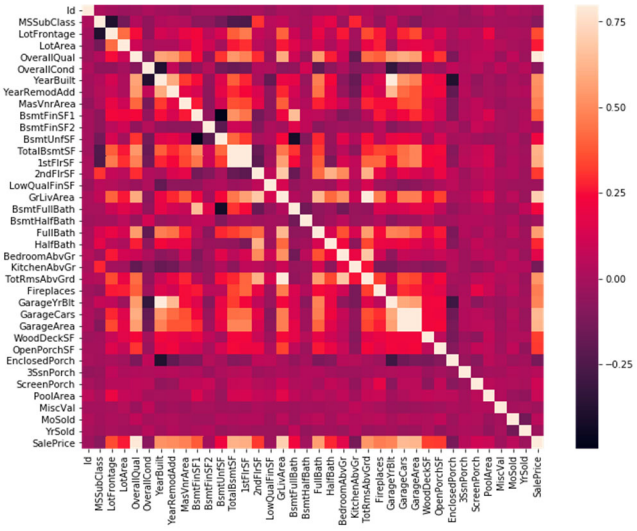

Figure 5. Heatmap of Correlations Between Variables

Next, the missing values are found and filled with appropriate values. There are many missing data in quite a few variables. The five features, 'PoolQC', 'MiscFeature', 'Alley', 'Fence', 'FireplaceQu', all have missing values percentages more than 40%, which should be dropped not only because the missing value percentage is too high, but also because they seem to be the features that are not usually considered when people buy a house. Moreover, by digging more into these variables, 'PoolQC', 'MiscFeature' and 'FireplaceQu' could be potential candidates for outliers with few data recorded [7]. The rest of the features with missing values are each filled with appropriate values.For example the 'LotFrontage' as a numeric variable is filled with the median 'lotFrontage' by each 'Neighborhood' [7]; and the missing values of some categorical features related to basement properties are filled with 'None', since they can be interpreted as no basement. After all these, there is no missing value remaining in the 'all_data' with 68 features. After dealing with the missing values, categorical features are separated out intending to be converted into numeric form through label encoding, for machines to read and operate.

After that, all features are numerical and ready to deal with skewness, which is a measure of how a variable is distorted from a normal distribution. As stated above when transforming the target variable, skewness is undesirable most of the time, since it leads to excessively large variance in estimates. The box-cox transformation, shown in equation (3), is a powerful transformation that makes the variances more constant, and the variable more normally distributed as the 'SalePrice' do. Thus, it is applied to the features that have a skewness higher than 0.75 [8].

$$y_i^{(\lambda)} = \frac{y_i^{\lambda}-1}{\lambda}, \text{if } \lambda \neq 0 \quad (3)$$

3.4 Models and Cross-Validation

After the data cleaning, it is time to separate the train set and the test set again for modeling. However, overfitting is possible to happen when a model learns the random fluctuation and noise, so that it fits so closely against the train set to the extent that there can be a negative influence on the accuracy of the prediction. To avoid this, Cross-Validation is the best way, which randomly splits the original train set into train folds and the validation fold and iterates several times to find all the results. In this project, the train set is split into 5 folds and root-mean-square logarithmic error is calculated for further evaluations. The models chose to cooperate with the Cross-Validation are LASSO Regression, Elastic Net Regression, Gradient Boosting Regression, XGBoost, and LightGBM.

LASSO Regression and Elastic Net Regression are two types of regularization models which minimize the residuals. Lasso, as shown in equation (4), uses shrinkage technique where data points are shrunk towards a central point as the mean, and puts a penalty equivalent to the sum of absolute values of the weights to obtain less model complexity, avoid overfitting, and shrink the weights of features that cause multicollinearity by adding a shrinkage penalty to the residual sum of squares(RSS).

$$\text{RSS} + \text{Lasso} = \sum_{i=1}^{n}(y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_j))^2 + \lambda_1 \sum_{j=1}^{p}|\beta_j| \quad (4)$$

Elastic Net Regression is similar to LASSO Regression but with one more penalty equivalent to the sum of squared weights, represented in equation (5).

$$\text{RSS} + \text{Ridge} + \text{Lasso} = \sum_{i=1}^{n}(y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_j))^2 + \lambda_1 \sum_{j=1}^{p}|\beta_j| + \lambda_2 \sum_{j=1}^{p}\beta_j^2 \quad (5)$$

Gradient Boosting is an ensemble algorithm which multiple new models are added to correct errors from previous models and combined to get better results. It is also called gradient tree boosting, since ensembles are composed of decision tree models. For the intention to reduce the residuals, one treeis added each time to the ensembled tree. Then, equation (6) is used to calculate the output for each leaf that minimizes the summation, noting that the previous predictions are taken into account.

$$\gamma_{jm} = \text{argmin} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \text{ for } j = 1, \ldots, J_m \quad (6)$$

Lastly, a new prediction, Fm(x), is made for each sample, which is composed by the last prediction and the output for each leaf applied by a learning rate, shown in equation (7).

$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} 1(x \in R_{jm}) \quad (7)$$

XGBoosting and LightGBM both follow the principle of Gradient Boosting. XGBoosting is a more regularized variant of Gradient Boosting. As it was named, it reaches the limit of computations for boosted tree algorithms, which utilizes advanced regularization to improve model generalization

capabilities and control overfitting. Similar toXGBoosting, LightGBM also uses decision trees for regression. In comparison to the level-wise tree growth in XGBoost, LightGBM uses leaf-wise growth which results in more loss reduction and higher accuracy, in the meanwhile, faster speed. However, overfitting might happen so that XGBoost may build a more robust model [9].

Stacked model is a technique in ensemble learning that combines the skills of various methods for a more accurate and stable prediction. It is important because by combining the best of many algorithms there will be less variance than a single regressor. In this project, the function 'StackingCVRegressor' is used to find the stacked model and it is the final model for prediction.

3.5 Evaluation on the root-mean-squarelogarithmicerror(rmlse)

The root-mean-square logarithmic error is one of the most commonly used measurements for evaluating models. It is the log transformation of the root-mean-square error; thus, it can be regarded as the root-mean-square error of the log-transformed predicted and log-transformed actual values [10]. One is added to both sides of values for natural logarithm to prevent getting the log of potential zero values.

The final evaluations for the models are shown in table 1, with both the error mean and standard deviation of the root-mean-square logarithmic errors with Cross-Validation(rmlse_cv) and the root-mean-square logarithmic errors(rmlse).

Table 1.Final Evaluations of the Six Models

Models/evaluation	rmlse_cv(mean, std)	rmlse
LASSO Regression	0.1189(0.0055)	0.11294
Elastic Net Regression	0.1188(0.0055)	0.11275
Gradient Boosting Regression	0.1174(0.0067)	0.04615
XBoost	0.1164(0.0078)	0.08093
LightGBM	0.1192(0.0066)	0.07568
Stacked Model	0.1119(0.0074)	0.66917

4. Results Analysis

The top rows of the final prediction are shown in Table.2, however, there is no right answer provided. The only way to check the result is the score given by the Kaggle competition. After several tries, the best score is 0.12444, ranked top 14%. This result is eventually contributed by the Stacked Model as the final predicting model, even though 0.00028 was not a huge change in score compared to the result when Gradient Boosting Regression was the final predicting model. Gradient Boosting Regression was chosen first due to its comparatively low root-mean-square logarithmic error mean with Cross-Validation and the low standard deviation. It also has the smallest root-mean-square logarithmic error among all the models, which is clearly depicted by Figure.6, whereas the Stacked Model performed worst. The small sample size might be a potential factor that causes this unexpected situation. The Stacked Model also takes too long to run, which is a condition that is always not recommended. Thus, Gradient Boosting Regression can be a great model to use for this project.

Table 2. Top rows of the Final Prediction Using Stacked Model

Id	SalePrice
1461	121439.42
1462	160816.69
1463	187468.2

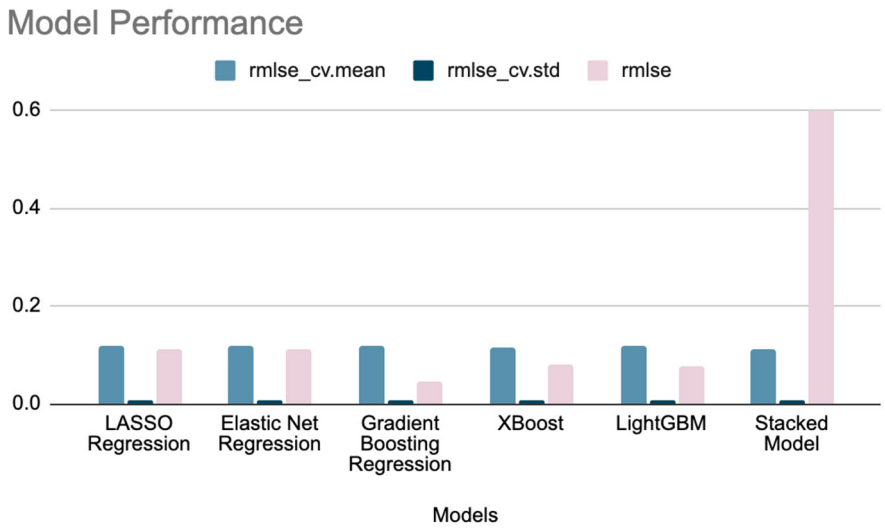

Figure 6. Model Performance

5. Conclusions

A house can be the most valuable property a person would ever own. Its price is undoubtedly influential in many aspects. Thus, predicting house prices is essential. In this article, the house prices of Ames are predicted by advanced regression techniques, like correlation, feature engineering, regularization models, gradient boosting models, and so forth. Through correlation, the hypothesis of the four features that are significantly related to the target variable is not rejected. 'OverallQual' has a correlation of 0.8, 'YearBuilt' has a correlation of 0.52, 'TotalBsmtSF' has a correlation of 0.65, and 'GrLivArea' has a correlation of 0.73. Additionally, multicollinearity is detected by the heatmap of correlation between variables. Through feature engineering, the target variable, 'SalePrice' becomes more normally distributed.The missing values are filled with appropriate values.Numerical features' skewness is reduced by box-cox transformation, and categorical features are label encoded for further machine learning. Through modeling, Gradient Boosting Regression Model is chosen by the lowest root-mean-square logarithmic error of 0.04615, a comparatively low root-mean-square logarithmic error mean with Cross-Validation, and a less time consumption.

Everything seems appropriate, yet there are still problems to pay attention to for further research. Features are categorized manually, which means they can be categorized differently by each different researcher. This works in making hypotheses but cannot be widely used. Moreover, the 'location' features were unexpectedly less correlated, this might be a consequence when the amount of 'location' features is small. This can always be a problem before the features can be categorized validly. Another possible problem is the small sample size of the chosen dataset, which could be the reason that the Stacked Model has an extremely high root-mean-square log error. With small sample size, manually ensembling the models with different weights could be a better alternative. Lastly, the prediction is made solely depending on the features of the house, however, the market uncertainty caused by the society and political factors could be ignored. New problems are always appearing in machine learning analysis for researchers to make astonishing improvements at all times.

References

[1] Zhou, C. 2021. House price prediction using polynomial regression with particle swarm optimization. Journal of Physics: Conference Series, 18(2).https://iopscience.iop.org/article/10.1088/1742-6596/1802/3/032034/pdf

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

[3] Alfiyatin, A. N., Taufiq, H., Febrita, R. E., Mahmudy, W. F. 2017. Modeling house price prediction using regression analysis and particle swarm optimization. International Journal of Advanced Computer Science and Applications, 8(10),323-326.https://thesai.org/Downloads/Volume8No10/Paper_42-Modeling_House_Price_Prediction_using_Linear_Regression.pdf

[4] D. X. Zhu and K. L. Wei, The Land Prices and Housing Prices —— Empirical Research Based on Panel Data of 11 Provinces and Municipalities in Eastern China, Int. Conf. Manag. Sci. Eng., 2013, pp. 2118–2123.

[5] S. Kisilevich, D. Keim, and L. Rokach, A GIS-based decision support system for hotel room rate estimation and temporal price prediction: The hotel brokers' context, Decis. Support Syst., vol. 54, 2013, pp. 1119– 1133.

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

[7] Marcelino, P.2017. Comprehensive Data Exploration with Python.https://www.kaggle.com/code/pmarcelino/comprehensive-data-exploration-with-python/notebook

[8] Serigne. 2017. Stacked Regressions: Top 4% on LeaderBoard. https://www.kaggle.com/code/serigne/stacked-regressions-top-4-on-leaderboard/notebook#Modelling

[9] Saha, Sumit.2022. XGBoost vs LightGBM: How Are They Different.https://neptune.ai/blog/xgboost-vs-lightgbm

[10] Gok, H. 2019. Metrics. https://hrngok.github.io/posts/metrics/