Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/46/20241300

Prediction of second-hand house prices in Ames city based on data mining

Chucheng Zhou

Xiamen University Malaysia- Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, Sepang, Selangor 43900 Malaysia

SWE2209553@xmu.edu.my

Abstract. As the secondary housing market continues to play a vital role in the global real estate industry, there is an increasing demand for accurate and efficient prediction of secondary housing prices. In this study, transaction data from the second-hand housing market in Ames, Iowa, USA, available on the Kaggle website, was utilized. A total of 79 influential feature variables, including Lot Area, Overall Quality, and Year Built, were selected to construct seven different models for predicting secondary housing prices: Linear Regression, Lasso Regression, Ridge Regression, LassoCV, RidgeCV, Random Forest, and Neural Network models. To assess the effectiveness of these models, we utilized a range of evaluation metrics including the R2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results indicated that the RidgeCV model stood out among the others, displaying the most robust predictive performance with a goodness-of-fit of 90.3%. This discovery holds significant implications for improving the accuracy of secondary housing valuation.

Keywords: House price prediction, Lasso regression, Ridge regression, Random forest, Neural network.

1. Introduction

The ongoing and swift growth of the global economy has led to a rising influx of pre-owned homes into the real estate sector. This has resulted in the continual expansion of transactions within the secondary housing market, establishing it as a crucial and essential segment of the real estate industry. Therefore, accurate prediction of secondary housing prices holds significant importance in promoting sustained and stable development in the real estate sector. There are two key reasons for this importance: Firstly, precise price predictions empower individuals to make well-informed decisions when it comes to acquiring their dream homes, thereby enhancing their sense of security and belonging within their chosen houses. Secondly, housing prices are subject to rapid fluctuations and significant volatility. So price predictions can assist businesses in making efficient decisions and help governments in preventing economic bubbles. In recent times, researchers have thoroughly investigated the forecasting of secondary housing prices, yielding a substantial body of research accomplishments. Some scholars have approached the analysis of secondary housing prices from the perspective of feature price theory. For example, Jia Shenghua et al. developed the theoretical framework for real estate feature price models and their applications [1]. Limsombunchao, V. House price prediction: hedonic price model vs. artificial neural network [2]. H. Selim investigated the determinants of house prices in Turkey using hedonic

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/46/20241300

regression and artificial neural networks [3]. C. Bitter et al. explored the incorporation of spatial variation in housing attribute prices, comparing geographically weighted regression and the spatial expansion method [4]. With the advancement of machine learning, some scholars have applied machine learning algorithms to predict secondary housing prices [5]. For instance, Varma, A., Sarma et al. House price prediction using machine learning and neural networks. Xu et al. constructed machine learning models such as regression trees, boosting, bagging, and random forests [6]. Through model selection and parameter tuning, they found that random forests yielded excellent fitting results. In practice, numerous real estate companies have developed a variety of algorithms to tackle real-world challenges in predicting house prices. One of the well-known algorithms used for predicting house prices is the Zestimate, which was developed by Zillow.

Due to the comparative analyses between different methods are not comprehensive or in-depth enough. So this paper takes the secondary housing market in Ames, Iowa, as an example and utilizes three classes of models: linear regression, random forest, and neural networks, to predict the prices of secondary houses in this locale. By evaluating the models using three metrics, we aim to select the optimal predictive model. This endeavor is intended to provide a reasonable reference for government agencies, intermediary institutions, and prospective homebuyers.

2. Models

For this paper, we are using three base models: linear regression, random forest, and neural networks. For linear regression, we added lasso and ridge constraints.

2.1. Linear Regression

The first relationship between house price and various features that comes to mind is linearity. Therefore, we first use the most basic model, linear regression, to try to extract a linear relationship. We start by initializing 78 random gaussian weights for each of the features excluding sales price. Then through backpropagation and gradient descent, we find proper weights and then carry them onto test datasets.

However, a disadvantage of linear regression is that it might overfit due to various reasons: either too many iterations or the bias between training and testing data. Therefore, it's often beneficial to add constraints on the magnitude of our weights. In the conventional approach, we have the option to employ either the L1 norm, commonly referred to as lasso regression, or the L2 norm, which is also recognized as ridge regression.

2.1.1. Lasso Regression. Lasso, also known as a regularization sparse model, achieves sparsity by introducing L1 regularization, which compresses some coefficients while setting others to zero. This results in a more refined model, retaining the advantages of subset shrinkage. It is a biased estimator for dealing with data that has complex multicollinearity [7].

The optimal weights for a lasso regression are defined as below:

$$\min_{\beta \epsilon R^p} \left\{ \frac{1}{N} ||y - X\beta||_2^2 \right\} subject\ to ||\beta||_1 \leqslant t. \tag{1}$$

It's very similar to the basic linear regression model, only with an added L1 constraint. Its Lagrange form is written as below:

$$\min_{\beta \epsilon R^p} \left\{ \frac{1}{N} ||y - X\beta||_2^2 + \lambda ||\beta||_1 \right\} \tag{2}$$

2.1.2. Ridge Regression. Ridge regression analysis is a biased estimation method specifically used for analyzing multicollinearity in data. It is, in fact, an improved version of the least squares method, which sacrifices the unbiasedness of least squares in favor of giving up some accuracy to seek a regression process that is slightly less effective but more in line with reality [8]. It performs better in fitting ill-

DOI: 10.54254/2755-2721/46/20241300

conditioned data compared to the least squares method. If we replace the L1 norm constraint to L2 norm constraint, we would obtain ridge regression instead:

$$\min_{\beta \epsilon R^p} \left\{ \frac{1}{N} ||y - X\beta||_2^2 \right\} subject\ to ||\beta||_2 \leqslant t. \tag{3}$$

and its Lagrange form is written as below:

$$\min_{\beta \epsilon R^p} \left\{ \frac{1}{N} ||y - X\beta||_2^2 + \lambda ||\beta||_2 \right\} \tag{4}$$

The key difference between Lasso Regression and Ridge Regression is the use of norms on the constraint conditions. It also raises another significant difference.

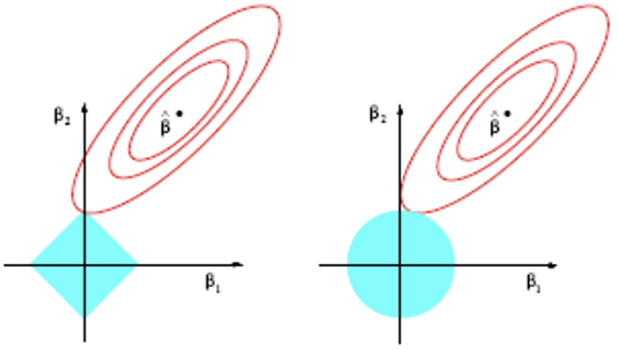


Figure 1. Lasso vs Ridge regression [9]

As indicated in Figure 1, the red lines are the contours of our constrained functions. While the blue areas centered at the origin are the constraint regions. On the left we have the L1 norm constraint, the constraint regions will intersect the contours at corners, thus setting some of our variables to 0. While for L2 norm constraint, unless we started on the axis, it won't meet on the axis, thus none of our variables will be set to 0. So lasso regression can completely ignore some of the variables while ridge regression cannot.

2.2. Random Forest

Random forest is a great non-parametric model in machine learning. Random forest is a collection of many decision trees. A decision tree splits the data into different groups by certain features each time. Features are often chosen by orders of importance with respect to the influence on our dependent variable. It can also be seen as a flow-chart, which represents the nature of human thinking. Figure 2 serves as an illustration of a decision tree. It Second-Hand House splits all prices by house size and then by their number of bedrooms.

DOI: 10.54254/2755-2721/46/20241300

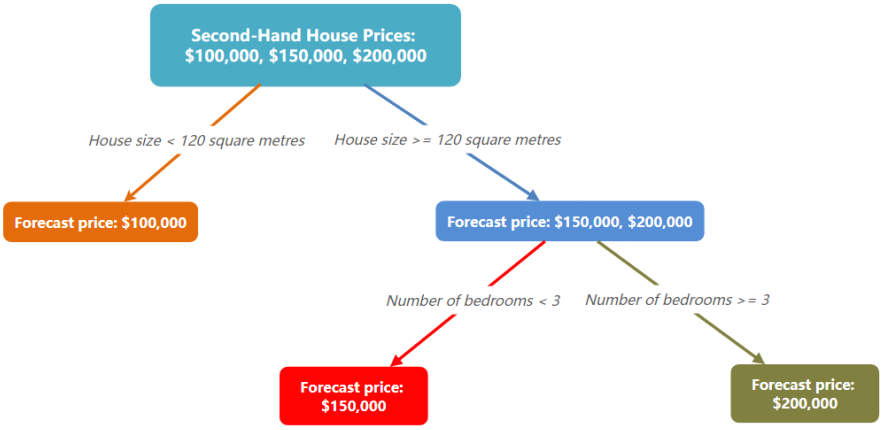


Figure 2. Decision Tree example (Original)

Mathematically, a decision tree can be expressed as follows: Consider a dataset D, a set of features denoted as F, and a dependent variable Y. The objective is to discover a set of rules denoted as R that effectively partitions D using features from F in order to make predictions for Y.

$$\begin{aligned} D &= \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \\ R &= Splitting\ rules\ based\ on\ features\ from\ F \\ Y &= Dependent\ variable\ to\ predict \end{aligned} \tag{5}$$

A decision tree recursively splits the dataset based on features, such that:

$$D_t = D - D_t - D_r \tag{6}$$

Where: D_t is the current dataset at node t. D_l is the left child dataset after a split. D_r is the right child dataset after a split.

It can represent linear relationships as well as non-linear relationships as it breaks complex datasets into manageable parts. Random forest works by collecting results from a number of different decision trees and takes either the average or the majority of these results to generate the final output. The output of a random forest ensemble can be computed as:

For regression tasks (average of predictions from individual trees:

$$\hat{Y}_{RF}(x) = \frac{1}{N}\sum_{i=1}^{N}\hat{Y}_i(x) \tag{7}$$

For classification tasks (majority vote of predictions from individual trees):

$$\hat{Y}_{RF}(x) = mode\left(\hat{Y}_1(x), \hat{Y}_2(x), \ldots, \hat{Y}_N(x)\right) \tag{8}$$

Where: $\hat{Y}_{\mathrm{RF}}(x)$ represents the prediction made by the random forest for the input data point x. $\hat{Y}_i(x)$ stands for the prediction generated by the i-th decision tree within the forest, and N signifies the total number of decision trees in the random forest. For classification, "mode" is the function that calculates the majority class.

2.3. Neural Network

The design of the neural network draws inspiration from the physiological workings of the human brain. It consists of multiple hidden layers with non-linear activation functions to extract non-linear relationships. Each hidden unit represents some non-linear or linear combinations of the inputs. Through

DOI: 10.54254/2755-2721/46/20241300

back-propagation and gradient descent, we can converge the weights of each hidden layer to near-optimal values. Typically, having more hidden units and hidden layers leads to better results, but it also increases the training time. Modern deep neural networks often require days or even weeks to train. For our experiment, we simplified the model to have only one hidden layer with 512 hidden units. The Figure 3 is a simplified neural network diagram. And the Figure 4 is an algorithm flowchart describing the training process of a neural network:

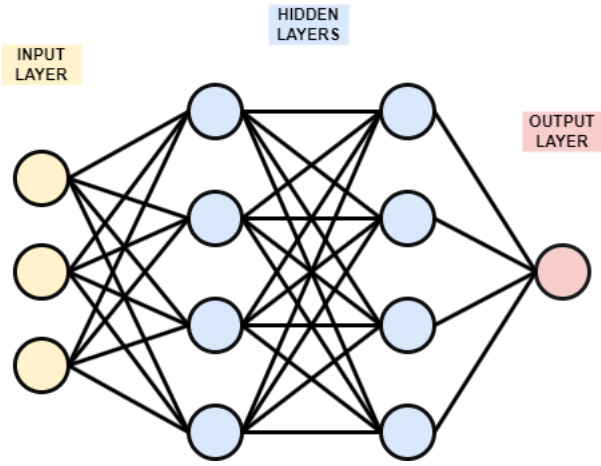


Figure 3. A simplified neural network (Original)

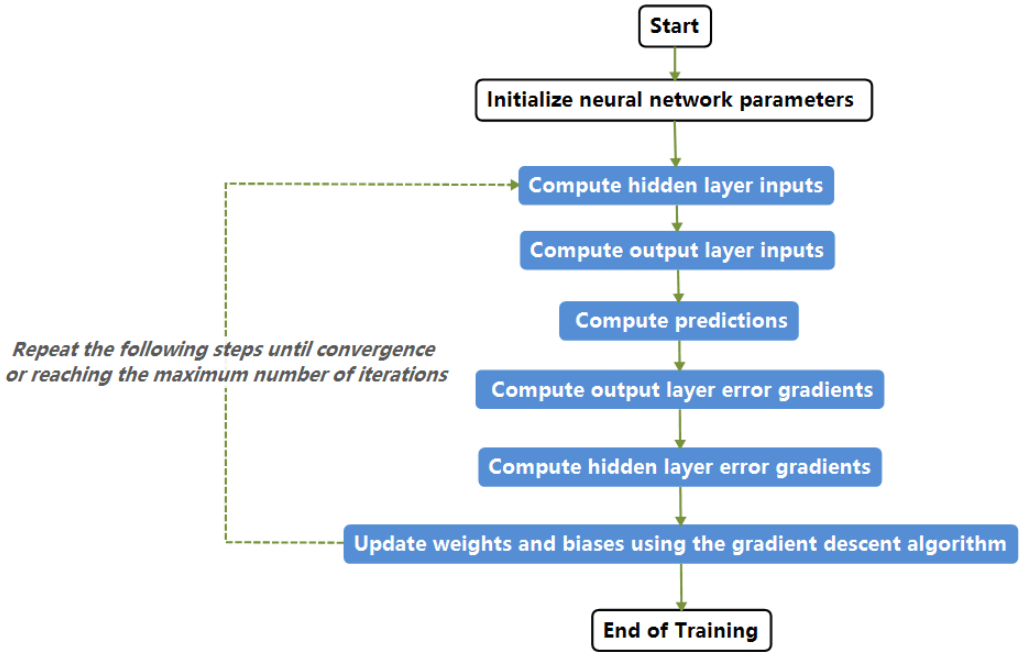


Figure 4. An algorithm flowchart (Original)

In Figure 4, we start by initializing the neural network parameters, then proceed with forward propagation to calculate predictions, compute the loss function, and perform backpropagation to update the network's weights and biases using the gradient descent algorithm. This iterative process continues until certain convergence criteria are satisfied or the maximum allowable number of iterations is reached. This describes the fundamental training procedure of a neural network.

DOI: 10.54254/2755-2721/46/20241300

3. Experiment Setup

3.1. Dataset

The dataset comes from the Kaggle website [10] and comprises secondary housing transaction data for Ames, Iowa, spanning from the year 1872 to 2010. It consists of a total of 2,919 records, including 1,460 training data and 1,459 testing data. Each instance contains 79 feature variables such as Lot Area, Overall Quality, and Year Built.

3.2. Feature Engineering

3.2.1. Analysis of Feature Variable. We applied cross validation on lasso and ridge regression for better test accuracies. Using Python, we obtained the names of 79 feature variables, as shown in the Table 1.

Table 1. Feature Variable Names

MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMatl, Exteriorlst, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, ExterCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating, HeatingQc, CentralAir, Electrical, 1stFlRSF, 2ndFlRsf, LowQua1FinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, KitchenQual, TotRmsAbvGrd, Functional, Fireplaces, FireplaceQu, GarageType, GarageYrBlt, GarageFinish, GarageCars, GarageArea, GarageQual, GarageCond, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SnPorch, ScreenPorch, PoolArea, PoolQC, Fence, MiscFeature, MiscVal, MoSold,YrSold, SaleType, SaleCondition, SalePrice

3.2.2. Analysis of Feature Correlations. Next, we generate a correlation matrix for the features using Python and visualize this matrix with a heatmap to explore linear relationships within the dataset. Figure 5 presents the correlation matrix.

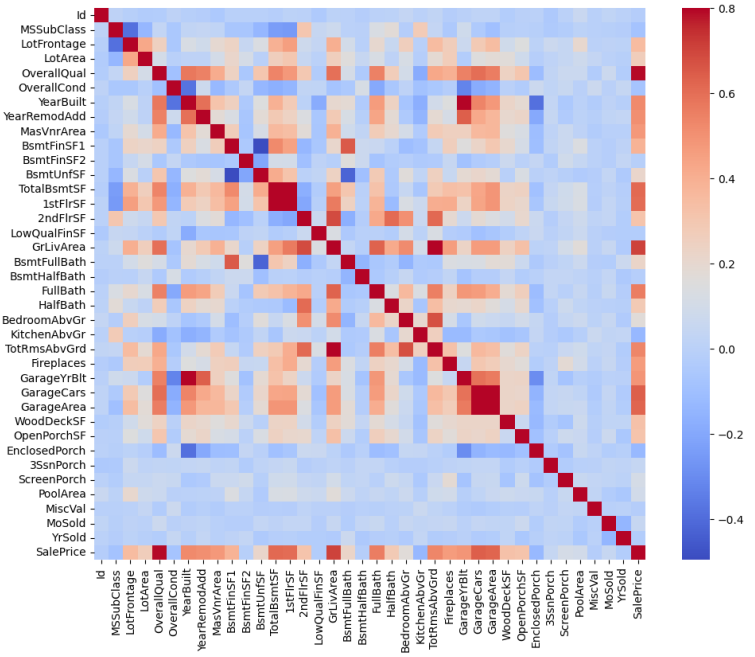


Figure 5. Correlation Matrix Among Features (Original)

DOI: 10.54254/2755-2721/46/20241300

In the figure, we observe a strong correlation between the "TotalBsmtSF" and "1stFlrSF" variables, as well as the "GarageX" variables, indicating the presence of multicollinearity. Additionally, we find that the variables "OverallQual," "GrLivArea," "GarageCars," "GarageArea," "TotalBsmtSF," "1stFIrSF," "FullBath," "TotRmsAbvGrd," and "YearBuilt" exhibit a strong correlation with "SalePrice." Therefore, we have generated a heatmap (Figure 6) to visualize the relationships between these variables and "SalePrice."

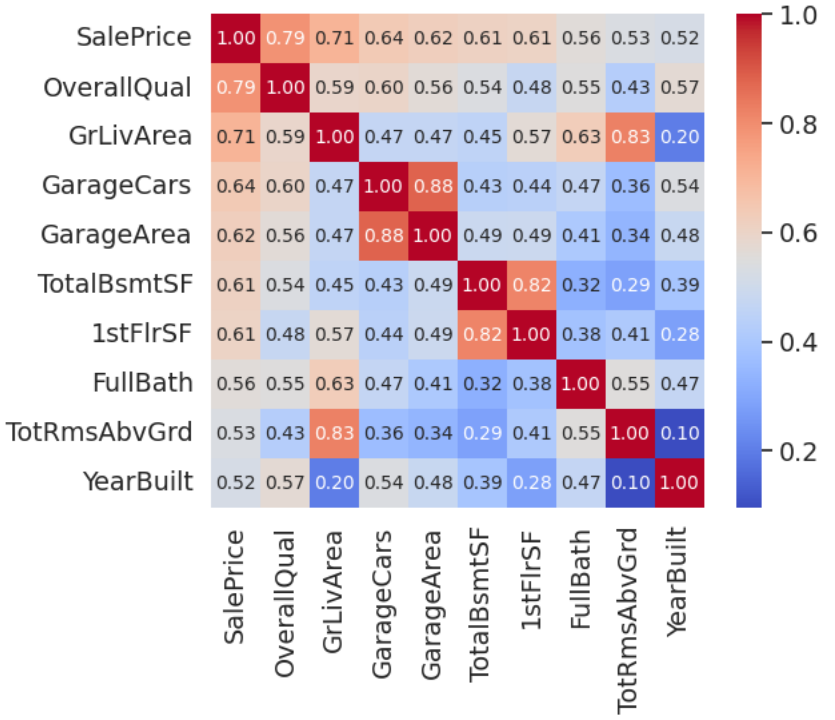


Figure 6. Correlation matrix Among some features (Original)

We observed a strong correlation between "Overall Quality," "Ground Living Area," and "Total Basement Square Foot" with "SalePrice." Furthermore, we identified "Garage Cars" and "Garage Area" as highly correlated variables. However, the number of cars that can fit into a garage depends on the garage's area. Therefore, we only need one of these variables for our analysis. This paper opted to retain "Garage Cars" due to its stronger correlation with "SalePrice." Similarly, we addressed two pairs of strongly correlated variables: "Total Basement Square Foot" and "1st Floor," as well as "Total Rooms Above Grade" and "Ground Living Area."

3.2.3. Missing Values and Outliers. In the dataset, some data information was missing for various reasons, and missing data may also contain valuable information [11]. To ensure data quality and usability, we employed common methods for handling missing data, which included deleting missing values and filling missing values.

For missing values, we considered two main approaches:

1. Deletion: If the number of missing values was relatively small and information loss was not a concern, we directly deleted rows or columns containing missing values.
2. Filling: Missing values were replaced with specific values, such as the mean, median, mode, or zero, depending on the type of data.

DOI: 10.54254/2755-2721/46/20241300

Table 2. Missing Data Status for Each Feature Variable

Feature	Total	Percent
PoolQC	1453	0.995
MiscFeature	1406	0.963
Alley	1369	0.938
Fence	1179	0.808
FireplaceQu	690	0.473
LotFrontage	259	0.177
GarageYrBlt	81	0.055
GarageCond	81	0.055
GarageType	81	0.055
GarageFinish	81	0.055
GarageQual	81	0.055
BsmtFinType2	38	0.026
BsmtExposure	38	0.026
BsmtQual	37	0.025
BsmtCond	37	0.025
BsmtFinType1	37	0.025
MasVnrArea	8	0.005
MasVnrType	8	0.005
Electrical	1	0.001

In Table 2, we provided an overview of the number of missing values and their proportions for each feature variable. Variables like PoolQC, MiscFeature, Alley, Fence, FireplaceQu, and LotFrontage, which had a missing rate exceeding 15%, were removed from the dataset. Similarly, we observed that variables with names like "GarageX" and "BsmtX" shared the same number of missing data and were removed as well. For "MasVnrArea" and "MasVnrType," we considered them unimportant and decided to remove them. Lastly, there was a single missing observation in "Electrical," which we deleted while retaining the variable.

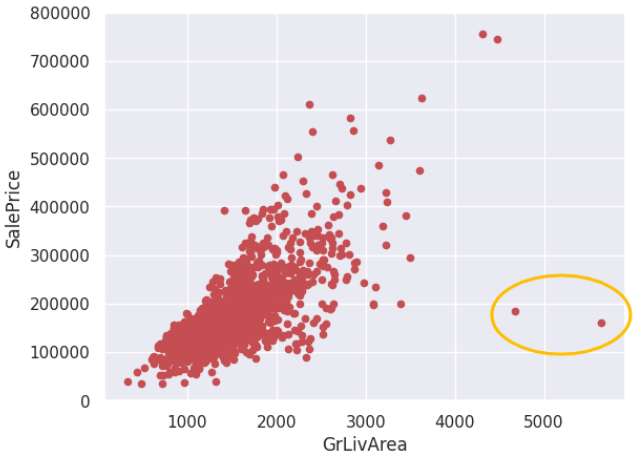


Figure 7. Scatterplot of "GrLivArea" and "SalePrice." (Original)

Moving on to outliers, we identified them using scatterplot analysis. An example of this was seen in Figure 7, where we analyzed "GrLivArea" and "SalePrice." Two high-value data points in "GrLivArea" were deemed as outliers and subsequently removed to ensure data integrity.

DOI: 10.54254/2755-2721/46/20241300

3.3. Evaluation metrics

For the evaluation metrics, We provide the R2 score, along with the mean absolute error (MAE) and mean squared error (MSE) as part of our reporting. A higher R^2 score means better performance, while smaller MAE and MSE values indicate better performance. Their definitions are as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{9}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{10}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{11}$$

3.4. Implementation Details

We implemented both linear regression and random forest models using the Python library scikit-learn, and constructed a neural network using PyTorch. Since SalePrice was not provided in the testing data, we split the original training data into a training set and a testing set using a 7-to-3 ratio. For linear regression, Lasso and Ridge, we used the default parameters. For LassoCV and RidgeCV. we set the parameter cv to 5. For the neural network, we added two hidden layers and applied a dropout of 0.2 for them. We employed the AdamW optimizer with a carefully adjusted learning rate of approximately 0.001, while setting the weight decay to 0.001. To facilitate single-GPU training, we configured a batch size of 16. Furthermore, the training epoch was established at 5000.

4. Experimental Results

Table 3 contains the performance results for the training set, while Table 4 presents the results for the testing set. Notably, the random forest and cross-validated ridge regression models exhibited the strongest performance. It's notable that random forest achieved an impressive R^2 score of 0.984 on the training set and maintained a score of 0.900 on the testing set without overfitting.

We also conducted a comparison among various linear regression versions. It is evident that ordinary linear regression presented the highest score of 0.930 on the training set, while models with regularization exhibited slightly lower scores. However, on the testing set, models with regularization outperformed ordinary linear regression. This underscores the effectiveness of regularization in mitigating overfitting. For the neural network, we initially encountered overfitting issues, The results are not satisfactory. To address this, we applied techniques such as batch normalization and dropout, and simplified the model architecture, resulting in improved performance.

Table 3. Results on the training set.

Model	R^2	MAE(10^4)	MSE(10^8)
Linear Regression	0.930	1.411	4.637
Lasso Regression	0.930	1.413	4.638
Ridge Regression	0.928	1.439	4.730
LassoCV	0.929	1.427	4.685
RidgeCV	0.922	1.496	5.168
Random Forest	0.984	0.653	1.025
Neural Network	0.964	1.077	2.392

DOI: 10.54254/2755-2721/46/20241300

Table 4. Results on the testing set.

Model	R^2	MAE(10^4)	MSE(10^8)
Linear Regression	0.887	1.796	6.411
Lasso Regression	0.889	1.782	6.282
Ridge Regression	0.901	1.734	5.600
LassoCV	0.897	1.737	5.841
RidgeCV	0.903	1.701	5.504
Random Forest	0.900	1.623	5.654
Neural Network	0.896	1.725	5.882

The analysis of the above experimental results is as follows:

- **Random Forest's Robustness:** The Random Forest model exhibited outstanding performance, with an impressive R2 score of 0.984 on the training set. It maintained a high score of 0.900 on the testing set without overfitting. This robustness can be attributed to the ensemble nature of Random Forest, which combines multiple decision trees, thereby reducing overfitting and achieving high generalization.
- **Effectiveness of Regularization:** In the linear regression models, ordinary linear regression achieved the highest R2 score of 0.930 on the training set. However, it suffered from overfitting issues on the testing set, where models with regularization, such as Lasso and Ridge Regression, outperformed ordinary linear regression. This demonstrates the effectiveness of regularization techniques in mitigating overfitting and improving model generalization.
- **Neural Network Overfitting and Improvements:** Neural networks initially show signs of overfitting and don't work well. To address this issue, techniques like batch normalization and dropout were applied, and the model architecture was simplified. These adjustments led to improved performance, highlighting the importance of proper model tuning and regularization in neural networks.

5. Feature Importance

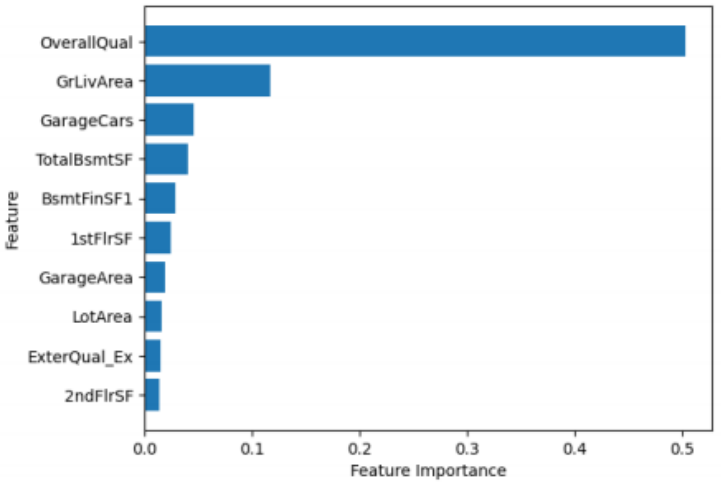


Figure 8. The top 10 most important features (Original)

Figure 8 displays the results of a feature importance analysis. In this analysis, we assessed the significance of various features in solving the problem at hand. It is important to note that we did not normalize the features in our linear regression models, which means that the coefficients obtained from the models do not accurately represent the true importance of the features.

DOI: 10.54254/2755-2721/46/20241300

Despite the absence of feature normalization, we managed to identify the top 10 significant features. Among these, OverallQual and GrLivArea stood out as the most crucial factors in addressing the problem. This suggests that the overall quality of the property (OverallQual) and the living area (GrLivArea) have a substantial influence on the outcome we are trying to predict. The reason for their high importance could be attributed to the fact that these features often encapsulate a wide range of information about a property. OverallQual, for instance, may encompass aspects like the condition of the house, materials used, and various subjective factors that collectively contribute to its overall quality. GrLivArea, on the other hand, reflects the size and layout of the living area, which can greatly impact the property's value.

6. Conclusion

In this paper, we have addressed the crucial task of predicting house prices, a matter of significant practical importance within the real estate industry. Our experiments, conducted using the Kaggle dataset, involved rigorous preprocessing, including outlier handling. We explored multiple iterations of linear regression, random forest, and neural networks, all of which yielded encouraging results. A notable discovery was the identification of OverallQual and GrLivArea as the primary predictors of house prices. This insight underscores the pivotal role played by these two features in shaping housing valuations.

Looking ahead, there are several avenues for further research and improvement. Firstly, the incorporation of more advanced feature engineering techniques and the consideration of domain-specific knowledge could potentially enhance predictive accuracy. Secondly, the exploration of ensemble methods, harnessing the strengths of various models, holds the potential for even more robust predictions. In conclusion, continuously monitoring the model's performance and adjusting it to match the ever-changing dynamics of the property market is vital for ensuring the accuracy of house price predictions in real-world applications. This project provides a certain solution for future data mining efforts in the field of house price prediction, and we look forward to further exploration and refinement to meet the ever-growing demands of the real estate market.

References

[1] Jiang, S. (2004). Housing features and their price—analysis based on feature pricing. *Journal of Zhejiang University.*

[2] Limsombunchao, V. (2004). House price prediction: hedonic price model vs. artificial neural network.

[3] Selim, H. (2009). Determinants of house prices in Turkey: Hedonic regression versus artificial neural network. *Expert Systems with Applications, 36*(2), 2843-2852.

[4] Bitter, C., Mulligan, G. F., & Dall'erba, S. (2007). Incorporating spatial variation in housing attribute prices: A comparison of geographically weighted regression and the spatial expansion method. *Journal of Geographical Systems, 9,* 7-27.

[5] Varma, A., Sarma, A., Doshi, S., & Nair, R. (2018, April). House price prediction using machine learning and neural networks. In 2018 second international conference on inventive communication and computational technologies (pp. 1936-1939). IEEE.

[6] Xu, K. (2022). Predicting housing prices and analyzing the real estate market in the Chicago suburbs using Machine Learning. *arXiv:2210.06261*.

[7] Liu, J., Cui, L., Liu, Z., & Luo, X. (2015). Regularization Sparse Models. *Journal of Computer Research and Development, 38*(07), 1307-1325.

[8] Yang, N. (2004). The Unique Role of Ridge Regression Analysis in Resolving Multicollinearity Issues. *Statistics and Decision, 000*(003), 14-15.

[9] Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression (with discussion). Annals of Statistics, 32(2), 407-499.Figure 3.11

[10] "House Price – Advanced Regression Techniques." Kaggle. Retrieved from https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques

DOI: 10.54254/2755-2721/46/20241300

[11] Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2006). Multivariate data analysis 6th Edition.