

House Price Prediction

Bob Dai

zdai38@wisc.edu

Xiaozhe Cheng

xcheng82@wisc.edu

Zining Tan

ztan46@wisc.edu

Abstract

The house price is an important index for real estate. The house price can be affected by many other factors such as location, size, build year. In this project, we use machine learning algorithms to make the prediction of the house price based on The Ames Housing Dataset from Kaggle. The dataset contains 2930 observations and 79 explanatory variables. In the data processing, we deal with missing values in the dataset and utilize One-Hot Encoding to transform categorical features. We split our dataset into 75% training and 25% testing sets. The models we use include Random Forest algorithm, XGBoost algorithm, combined with k-Fold Cross-Validation to tune the hyperparameters based on accuracy which are evaluated by the coefficient of determination R^2 of the prediction. We use these two models to make prediction and compare their accuracy. The score of the random forest is 0.9998 while the score of the XGBoost is 0.9999. This result turns out that XGBoost is slightly better than random forest on the test set. We also discuss the strength of our models, the question we have met when we did this project and how can we improve our models in the future.

1. Introduction

In the past ten years, the economy has developed rapidly, and housing prices have also grown rapidly, which has become a hot issue of continuous concern to the society, which has made many people who plan or plan to buy a house worry. The factors affecting housing prices are complex, and are related to supply and demand, government regulation, and town planning. Therefore, how to make a scientific and effective forecast of housing prices has become a hot issue in this field. With the advent of the era of big data and the development of machine learning, the ability of data storage, management and analysis has been greatly improved. As an important part of a country's national economy, real estate is not only related to the country's macroeconomic trend, but also related to social harmony and stability. The value of house price prediction is more prominent. Through the prediction of housing prices, we can maximize

the economic benefits. House prices are a hot issue with complex influencing factors, which is difficult to make a comprehensive and accurate prediction. House prices are an important indicator that reflects the performance of the economy. Real estate developers and home buyers pay close attention to housing price fluctuations. Building an effective housing price prediction model is of great significance to the financial market, people's sentiments, and people's livelihood.

For this project, our team aims to predict the house prices based on the data we obtain from Kaggle using the machine learning method. This dataset was compiled by Dean De Cock for use in data science education. It contains the final price of the house price in Ames, Iowa. What we need to do is to use the 79 factors about the house in the dataset to predict the price, but these factors are both discrete and continuous, numerical and character, and there are a large number of missing values, and a certain number of outliers. We divide this data set into training and testing data sets. We use the training dataset to fit the machine learning model. After the optimal fitted model is obtained from the training set, the testing set is used to make model predictions. The accuracy of the models are evaluated by the coefficient of determination R^2 of the prediction.

The model can be used as a cornerstone to further develop a "House Price Prediction System" that can be extended everywhere in the world. This system can be used by real estate agents, house owners, and any individuals, so that they can get more information about the price range of the house price. For the country, accurate prediction of local house prices is conducive to the government to issue relevant policies. It is also conducive to the economic development of the country.

2. Related Work

The real estate industry has become a competitive industry, and has a close relationship with many economic indicators. Many people want to study how to predict house prices through some related variables. In recent years, many methods have been used to explore the relationship between house prices and house characteristics. Chenchen Fan, Zeichen Cui and Xiaofeng Zhong used Lasso linear

regression model, Ridge linear regression model, Random forest model, Linear Kernel, Gaussian Kernel and XGB to build different models. Their accuracy was evaluated by Mean Squared Error. They finally found that Lasso linear regression model, Ridge linear regression model and XGB have lower MSE.[6] Ayush Varma, Abhijit Sarma, Sagar Doshi and Rohini Nair chose Linear Regression, Forest Regression, Boosted Regression and Neural Networks to make predictions. They mentioned that the Neural networks will lead to the largest efficiency of the algorithm. [12] Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang and Rick Siow Mong Goh used both Regression Algorithms and Hybrid Regression and the results were evaluated by RMSE. They used Ridge regression, Lasso regression and Gradient boosting separately, and then compared the coupling effect of multiple regression algorithms. Finally, they found the best hybrid regression result for test data is 65% Lasso and 35% Gradient boosting combination which 0.11260 scores. [8] Danh Phan applied Linear regression, Polynomial regression, Regression tree and Neural Network to build models based on the Melbourne Housing Market dataset. He also used Support Vector Machine, Step-wise and Principal Component Analysis to increase the accuracy which is evaluated by MSE. By comparing with SVM, they found the Regression tree and Neural Network were faster. By comparing the results of different models, they stated that the combination of Step-wise and SVM model is the best model with lowest errors.[10] XibinWang, JunhaoWen, YihaoZhang, Yubiao Wang used particle swarm optimization (PSO) and support vector machine (SVM) to build real estate price prediction and have a good forecasting performance.[13] Their datasets are similar to our datasets. They all contain a factor representing the house price and a lot of other explanatory variables. Most of these studies have different evaluation criteria to ours so the selection of models will be different. We will use some of their models to process and forecast based on our dataset.

3. Proposed Method

3.1. One-hot encoding

In order to mathematically handle categorical variables in machine learning models, we need to convert categorical variables into numerical variables. This is done by creating dummy variables, binary variables that represent the category the sample belongs to. This approach can be achieved by using the one-hot encoding transformer in Python. k-1 dummy variables out of k categorical levels can represent each categorical variable.

3.2. Random Forest

We use random forest as a main approach to make predictions in our project. Random Forest are the models

which can be used for both classification and regression. Bagging of trees is the main technique which is used by Random Forest. The main idea of the random forest is decorrelating several decision trees. We can create a lot of decision trees based on this approach.

In the random forest model, we fit a large amount of decision trees based on several different bootstrap samples and then selected a random subset of features at each node. Compared with other algorithms, it has great advantages and performs well in many current datasets. It can handle data with high dimensions (many features) without feature selection. When creating random forest, unbiased estimation is used for generalization error, and the model has strong generalization ability. In the process of training, the interaction between features can be detected and the implementation of random forest model is relatively simple.

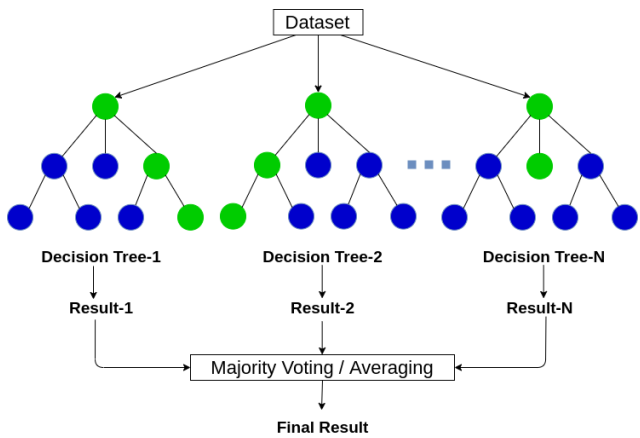


Figure 1. Image source: <https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/>

3.3. XGBoost

XGBoost is one of the fastest ways to perform gradient boosting, a scalable implementation of gradient boosting. It does this by solving one of the main inefficiencies of gradient boosted trees: use the potential loss of all possible splits to create new branches. XGBoost solves this inefficiency problem by observing the distribution of features on all data points in the leaf and adopting this information to reduce the search space. [11]

XGBoost is likely performing gradient boosting in an extreme way. Gradient Boosting fits trees sequentially to improve error of previous trees and boost weak learners to strong learners. [5]

We chose to implement XGBoost in our model because compared to Gradient Boosting, XGBoost includes regularization term for penalizing tree complexity, which improves model generalization and uses second order approximation for optimizing the objective. [7]

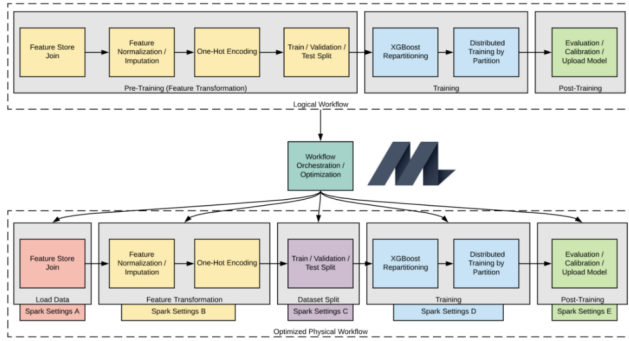


Figure 2. Image source: <https://eng.uber.com/productionizing-distributed-xgboost/>

3.4. K-Fold Cross-Validation

We use K-Fold Cross-Validation to calculate the best accuracy of our model while doing model evaluation in model selection in our project. It is one of the most popular techniques for model evaluation and model selection in machine learning. Cross-validation is a resampling process used to evaluate machine learning models on limited data samples. This process has a parameter k , which represents the number of groups to split a given data sample into. In this way, this process is usually called k -fold cross-validation. When choosing a specific value for k , it can be used in the reference model instead of k . K-fold cross-validation allows non-overlapping validation folds and overlapping training folds. [4]

3.5. Grid Search

We use grid search in our model to get the best hyperparameters for our model while doing model selection. Grid search is a tuning technique adopted to calculate the optimal value of hyperparameters. This search performed exhaustively on specific parameter values of the model. This model is also called an estimator. In this approach we use grid search to test the accuracy for each combination of hyperparameters in our Random Forest model and XGboost model. Grid search allows us to test every combination of the hyperparameters by running with multiple resolutions and we can type in the hyperparameter values we want to test by ourselves. The implementation of grid search would enhance the efficiency of our model selection process. Grid search can also work in poor coverage. [9]

4. Experiments

4.1. Dataset

The dataset of our study is the Ames Housing dataset which was compiled by Dean De Cock for use in data science education and we get this dataset from Kaggle Competition "House Prices - Advanced Regression Tech-

niques." This dataset describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. It contains 2930 observations and a lot of explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous).

There were several variables containing missing values. For example, the variable "Alley" which is the type of alley access to property contained 2732 missing values. The variable "Pool QC" which is the pool quality contained 2917 missing values. The variable "Lot Frontage" which is the linear feet of street connected to property contained 490 missing values. The following is a bar plot of all the variables and showed how many values were missing.

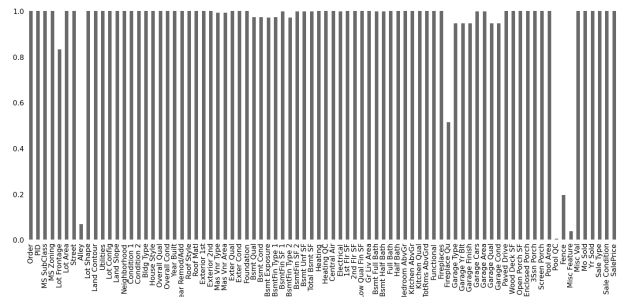


Figure 3. Bar plot of missing values

We apply for the correlations between SalePrice and other variables and we also made a plot for several variables with the highest correlation with SalePrice as follows.

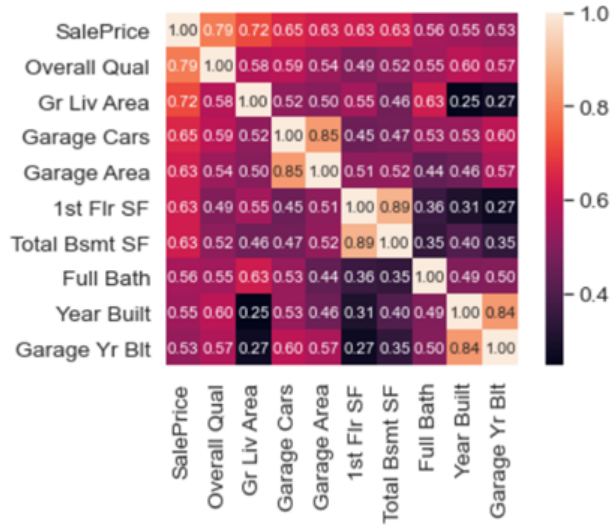


Figure 4. Several variables that have highest correlation with Sale Price

There are the descriptions of the variables have the highest correlation with SalePrice as follows.

Name	Type	Description
Overall Qual	Ordinal	Rates the overall material and finish of the house
Gr Liv Area	Continuous	Above grade (ground) living area square feet
Garage Cars	Discrete	Size of garage in car capacity
Garage Area	Continuous	Size of garage in square feet
1st Flr SF	Continuous	First Floor square feet
Total Bsmt SF	Continuous	Total square feet of basement area
Full Bath	Discrete	Full bathrooms above grade
Year Built	Discrete	Original construction date
Year Remod/Add	Discrete	Remodel date (same as construction date if no remodeling or additions)

Figure 5. The description of nine variables that have correlation with Sale Price

4.2. Software

We perform all experiments on Python, which provides various tools and libraries to conduct our experiments. In order to provide the reference of the version of the software, we list the software that we use in this experiment, include CPython 3.8.5, IPython 7.18.1, numpy 1.19.1, scipy 1.5.0, matplotlib 3.3.1, sklearn 0.23.2, xgboost 1.2.1.

4.3. Hardware

We perform all experiments on Zining’s computer. We list the hardware that Zining’s computer contains, include AMD Ryzen 5 3600 6-core processor 3.95GHz CPU and 16 GB RAM

4.4. Data Preparation

This dataset contains 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa with 2930 samples. There are 27 variables with missing values, but there are only 6 variables with missing values more than 400 and 5 variables with missing values more than 1000. Since the missing values in these 6 variables accounts for a relatively large proportion, it might not reasonable to estimate the value using KNN algorithm or simply use the mode. Thus, we drop the variables containing missing values greater than 400. Except for the missing variables in these 6 variables, there are only 252 samples with missing values in our dataset. Considering our dataset is large enough, we drop 252 samples and there is no missing value in our dataset.

The variables in the dataset are both discrete and continuous, numerical and categorical. We use the commonly used method, one-hot encoding. This method creates a binary column for each category of each categorical variable, which is also called1 dummy variable. We use “pandas.get_dummies”, which expects the input to be an array-like integer or string representing the categorical values. “Drop_first” should be true to remove the first dummy variable. After the data preparation step, our dataset has 241 variables and 2678 samples.

Then we need to split the dataset into train and test sets. We use “train_test_split” in the “sklearn.model_selection”. The train set accounts for 75% of the whole dataset and the test set accounts for 25% of the whole dataset. There are

2008 samples in the train set and 670 samples in the test set. The validation set is not necessary, since we will tune hyperparameters by using cross validation.

4.5. Method Implementation

Implementation of random forest: First, we need to import “RandomForestRegressor” from “sklearn.ensemble”. Second, we set the appropriate values for hyperparameters such as number of estimators and maximum depth. Third, we fit the model using our train set. Fourth, the score method provides the coefficient of determination R^2 of the prediction on our test set.

Implementation of XGboost: First, we need to import “xgboost”. Second, we set the appropriate values for hyperparameters such as number of estimators, maximum depth, and learning rate. Third, we fit the model using the train set. Fourth, the score method provides the coefficient of determination R^2 of the prediction on our test set.

Implementation of grid search cross-validation. First, we need to import “GridSearchCV” from “sklearn.model_selection”. Second, we choose the $k = 10$, because the empirical study shows that it generally has smaller standard deviation and higher accuracy. We set about 3 to 5 values for each hyperparameter each time to make sure that the runtime would not be too long. After we got the best hyperparameter for the first time, we would do grid search again with the smaller or larger values depending on whether the value we get last time is the smallest one or the largest one. After multiple times of grid search, we could get the best hyperparameters for both of our models.

5. Results and Discussion

5.1. Random Forest

The following graph is the graph we made with our model selection results. From the graph, we can see that the best hyperparameter we got for random forest is max_depth = 12 and n_estimators = 50. The best accuracy rate which is evaluated by the coefficient of determination R^2 of the prediction we got for this model is 99.75%.

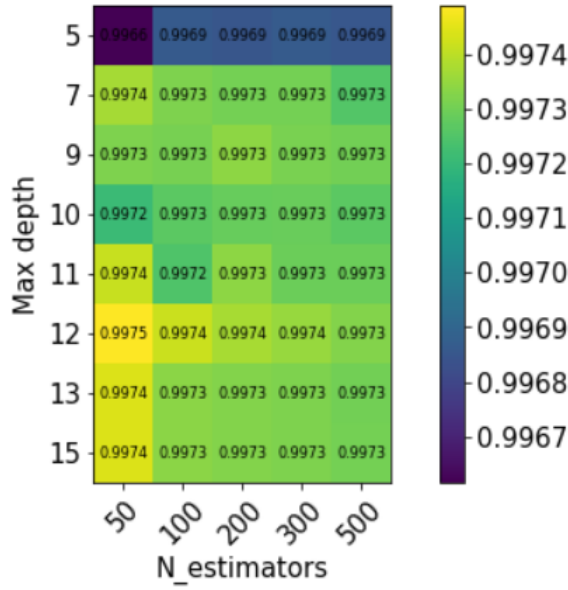


Figure 6. Grid Search Result for Random Forest

Then we use the random forest model with the best hyperparameter which is `max_depth = 12` and `n_estimators = 50` to predict the test set. Finally, we got the coefficient of determination R^2 of the prediction is 0.9998189239770987. We can compare the predicted values and the real values by the following pictures. The following picture shows the predicted value and true values for each observation. The blue dots represent the predicted values of the house price from the Random Forest Model when `max_depth = 12` and `n_estimators = 50`. The red cross represents the true values of the house price. Most of the points and cross overlap, which means the Random Forest Model can lead to accurate prediction values.

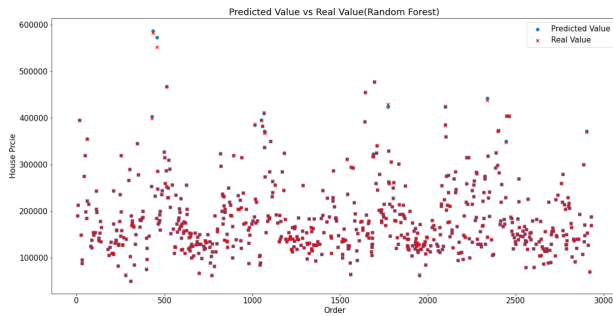


Figure 7. The predicted values and true values for each observation based on Random Forest

5.2. XGBoost

The following graph shows the model selection process for our XGBoost model. From the graph, we can see that the

best hyperparameters for our XGBoost model is `max_depth = 4` while `learning_rate = 0.07`. The `n_estimators` for this model is 300 and we got the best accuracy rate for our XGBoost model is 99.75% as well.

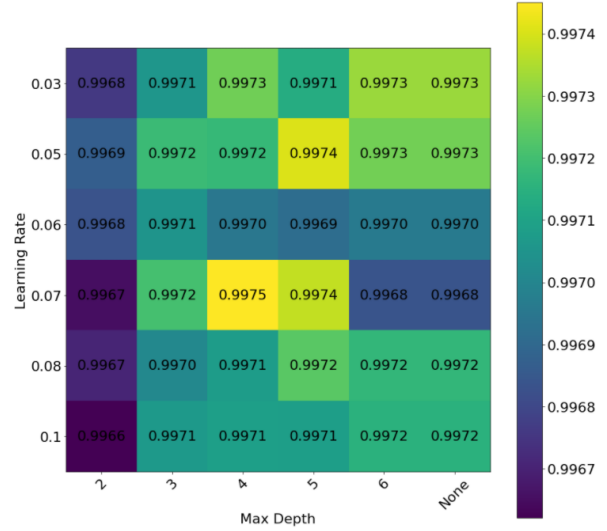


Figure 8. Grid Search Result for XGBoost

Then we used the XGBoost model with the best hyperparameter which is `max_depth = 4` while `learning_rate = 0.07` to predict the test set. Finally, we got the coefficient of determination R^2 of the prediction is 0.999948644199853. The following picture shows the predicted value from XGBoost Model and true values for each observation. The blue dots represent the predicted values of the house price from the XGBoost Model when `max_depth = 4` while `learning_rate = 0.07`. The red cross represents the true values of the house price. Most of the points and cross overlap, which means the XGBoost can also lead to accurate prediction values.

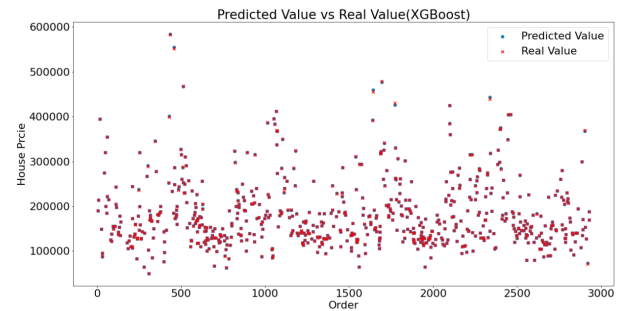


Figure 9. The predicted values and true values for each observation based on XGBoost

5.3. Discussion

Both Random Forest and XGBoost perform well in the house price prediction based on these 79 factors. The coefficient of determination R^2 of the prediction of our Random Forest models is 0.9998189 while the coefficient of determination R^2 of the prediction of our XGBoost is 0.9999486. The score of XGBoost is relatively higher than random forest. Finally, we chose XGBoost as our best model. The XGBoost is better because the XGBoost leverages the patterns in residuals. “XGBoost is normally used to train gradient-boosted decision trees and other gradient boost models. Random forests use the same model representation and inference, as gradient-boosted decision trees, but a different training algorithm.” [3] The biggest advantage of our project is the accuracy of our project. Our project has an accuracy of over 99%. The high accuracy makes our model a useful tool to predict the house prices with given data. Regarding the model’s performance, the model fitting time will increase as the complexity of models increases. Although XGBoost performs relatively better compared to random forest, XGBoost will take longer durations. If the dataset is very large, maybe a random forest might be more appropriate. If there is a lot of noise, it seems like that the XGBoost can result in overfitting and the random forest might be better.

There are two possible evaluation estimators in our project. The root mean squared error (RMSE) between the logarithm of the predicted value and the logarithm of the observed value is another potential estimator for the accuracy of our model, since by taking logarithmic, the errors in predicting expensive house and cheap houses will affect the result equally. This way of estimating prediction accuracy is widely used in Kaggle competition in regression settings. RMSE is the measurement of how far is the predicted value to true value. [2]

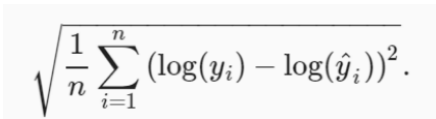

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (\log(y_i) - \log(\hat{y}_i))^2}.$$

Figure 10. The formulation of RMSE

However, the “GridSearchCV” has the method “best_params_”, which provides the best parameter combination based on the average of r^2 on left-out test fold.

$$R^2 = 1 - \frac{RSS}{TSS}$$

, “where RSS is the residual sum of squares $((y_{\text{true}} - y_{\text{pred}}) ** 2).sum()$ and TSS is the total sum of squares $((y_{\text{true}} - y_{\text{true.mean()}}) ** 2).sum()$. It represents the proportion of variance in y that can be explained by x. The

best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.” [1]

6. Conclusions

In the project, we try two different models to predict the house price with obtained data. We adopted Random Forest and XGBoost to train the data. We also used K-fold Cross-validation and Grid Search for model selection. Both of our models result in a high accuracy of prediction. Although Random Forest has a higher accuracy on training dataset, XGBoost gets a better score on the test dataset.

In conclusion, our goal of predicting the house price based on the obtained data is successfully accomplished. Based on the result we got, we can conclude that we should choose XGBoost to predict the data. Nevertheless, there is still space for improvement for our model. We missed some information by dropping variables and observations. We can adopt advanced methods for filling the missing values such as knn. We can normalize the raw data and use the Principal Component Analysis (PCA) to create a new set of features. In the future we would like to work out the importance of each feature in the model since in this project, we failed to tell if each feature has a negative or positive impact on house price and which feature has the largest impact on house price. In the future, we hope our model would be implemented in the housing market to help both the consumers and sellers predicting the house price fluctuations.

7. Acknowledgements

I would like to express my special thanks of gratitude to our professor Sebastian Raschka who gave us the golden opportunity to do this wonderful project. And we have learned a lot of machine learning algorithm from this amazing course, we really appreciate that. Secondly i would also like to thank my classmates and friends who provide us the opportunities to learn from each other. I would also like to thank the Kaggle website which provided us the dataset.

8. Contributions

Xiaozhe Cheng found the original dataset The Ames Housing Dataset from a kaggle competition. He also conducted data preparation and visualization. For the coding part of the project, Bob was responsible for developing the data loader, cleaning the data, and building random forest and XGB machine learning models. Zining was mainly responsible for the experimental part and developing a machine learning model. Xiaozhe was responsible for developing an evaluation model. For the writing task for the final report, Xiaozhe was responsible for introduction, related work, a part of proposed method and a part of experi-

ments ; Bob Dai was responsible for experiments and discussion; Zining was responsible for proposed method, discussion and conclusion.

References

- [1] 3.2.4.3.2. sklearn.ensemble.randomforestregressor¶.
- [2] House prices - advanced regression techniques.
- [3] Random forests in xgboost¶.
- [4] J. Brownlee. A gentle introduction to k-fold cross-validation, Aug 2020.
- [5] J. Brownlee. A gentle introduction to the gradient boosting algorithm for machine learning, Aug 2020.
- [6] C. Fan, Z. Cui, and X. Zhong. House prices prediction with machine learning algorithms. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing, ICMLC 2018*, page 6–10, New York, NY, USA, 2018. Association for Computing Machinery.
- [7] N. Kumar.
- [8] S. Lu, Z. Li, Z. Qin, X. Yang, and R. S. M. Goh. A hybrid regression technique for house prices prediction. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 319–323, 2017.
- [9] F. Malik. What is grid search?, Feb 2020.
- [10] T. D. Phan. Housing price prediction using machine learning algorithms: The case of melbourne city, australia. In *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pages 35–42, 2018.
- [11] G. Tseng. Gradient boosting and xgboost, Nov 2018.
- [12] A. Varma, A. Sarma, S. Doshi, and R. Nair. House price prediction using machine learning and neural networks. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1936–1939, 2018.
- [13] X. Wang, J. Wen, Y. Zhang, and Y. Wang. Real estate price forecasting based on svm optimized by pso. *Optik*, 125(3):1439 – 1443, 2014.