IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

Housing Price Prediction via Improved Machine Learning Techniques

Arjun Patil[1] and Vaishnavi A Patil[2]
Assistant Professor and Head Department of IT[1]
Student P.G. Department of IT[2]
Veer Wajekar ASC College, Phunde, Uran

Abstract: *House Price Index (HPI) is commonly used to estimate the changes in housing price. Since housing price is strongly correlated to other factors such as location, area, population, it requires other information apart from HPI to predict individual housing price. There has been a considerably large number of papers adopting traditional machine learning approaches to predict housing prices accurately, but they rarely concern about the performance of individual models and neglect the less popular yet complex models. As a result, to explore various impacts of features on prediction methods, this paper will apply both traditional and advanced machine learning approaches to investigate the difference among several advanced models. This paper will also comprehensively validate multiple techniques in model implementation on regression and provide an optimistic result for housing price prediction.*

Keywords: *House Price Index*

I. INTRODUCTION

Commonly, House Price Index (HPI) is used to measure price changes of residential housing in many countries, such as the US Federal Housing Finance Agency HPI, S&P/Case-Shiller price index, UK National Statistics HPI, UK Land Registry's HPI, UK Halifax HPI, UK Rightmove HPI and Singapore URA HPI. The HPI is a weighted, repeat- sales index, meaning that it measures average price changes in repeat sales or refinancings on the same properties. This information is obtained by reviewing repeat mortgage transactions on single-family properties whose mortgages have been purchased or securitized by Fannie Mae or Freddie Mac since January 1975. With some analytical tools, it allows
housing economists to estimate changes in the rates of mortgage defaults, prepayments, and housing affordability in specific geographic areas [1]. Because HPI is a rough indicator calculated from all transactions, it is inefficient to predict the price of a specific house. Many features such as district, age, and the number of floors also need to be considered instead of just the repeat sales in previous decades.

In recent years, due to the growing trend towards Big Data, machine learning has become a vital prediction ap- proach because it can predict house prices more accurately based on their attributes, regardless of the data from previous years. Several studies explored this problem and proved the capability of the machine learning approach [2],[3],[4]; however, most of them compared the models' performance but did not consider the combination of dif- ferent machine learning models. S. Lu et al. did conducted an experiment using a hybrid regression technique on forecasting house price data, but it requires intensive parameter tuning to find the optimal solution [5].

Due to the importance of model combination, this paper adopted the Stacked Generalization approach [6],[7], a machine learning ensemble technique, to optimize the predicted values. We used the "Housing Price in Beijing" dataset, fetched and uploaded to Kaggle by Q. Qiu [8]. By applying several methods on this dataset, we could validate the performance of each individual approach. The lowest Root Mean Squared Logarithmic Error (RMSLE) is 0.16350 on the test set, which belongs to the Stack Generalization method. The paper is structured as follows: Section 2 illustrates the details of the methodology; Section 3 compares the results; and Section 4 discusses the results, draws a conclusion, as well as proposes further potential directions to study the problem.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

II. METHODOLOGY

Data Preprocessing

"Housing Price in Beijing" is a dataset containing more than 300, 000 data with 26 variables representing housing prices traded between 2009 and 2018. These variables, which served as features of the dataset, were then used to predict the average price per square meter of each house.

The next step was to investigate missing data. Variables with more than 50% missing data would be removed from the dataset. The variable "Day on market" was removed because of 157, 977 missing data. Any observation which had missing values were also removed from the dataset. Below are a few feature engineering processes which were done to cleanse the dataset:

Remove attributes indicating the number of kitchens, bathrooms, and drawing rooms due to their ambiguity. Set the number of living rooms (bedrooms were mistranslated to living rooms) in a range from 1 to 4. Add attribute "distance" indicating the distance of the house from the center of Beijing. Replace attribute "constructionTime" with attribute "age" by deducting the year that the house constructed from the current year (2019). Set minimum values for attributes "price" and "area". Split the attribute "floor" into attributes "floorType" and "floorHeight".

After feature engineering, the dataset was checked for outliers. Through Inter-Quartile Range (IQR), an outlier x can be detected if:

$$x < Q_1 - 1.5 \cdot IQR \quad \text{OR} \quad Q_3 + 1.5 \cdot IQR < x \qquad (1)$$

where:

Q_1 = 25th percentiles Q_3 = 75th percentiles $IQR = Q_3 - Q_1$

After applying Equation (1) to every column of the dataset, the final dataset contained 231962 data with 19 features, 9 of which were numerical values and 10 of which were categorical values. Table 1 includes details of each attribute.

Table 1. List of Attributes

Attribute Name	Data Type	Description
Lng	float64	Longitude of the house
Lat	float64	Latitude of the house
district	int64	District (District 1- District 13)
distance	float64	Distance to the center of Beijing
age	int64	Age of the house
square	float64	Area of the house
communityAverage	float64	Average housing price of the community
followers	int64	Number of followers
tradeTime	int64	Trade Time (2002-2018)
livingRoom	int64	Number of bedrooms
floorType	object	Floor height relative to the building
floorHeight	int64	Floor height
buildingType	int64	Building Type
renovationCondition	int64	Renovation Condition
buildingStructure	int64	Building Structure
ladderRatio	float64	Ratio between population and number of elevators of the floor
elevator	int64	Whether the house has any elevator
fiveYearsProperty	int64	Whether the house is a five-year property
subway	int64	Whether the house is near any subways

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

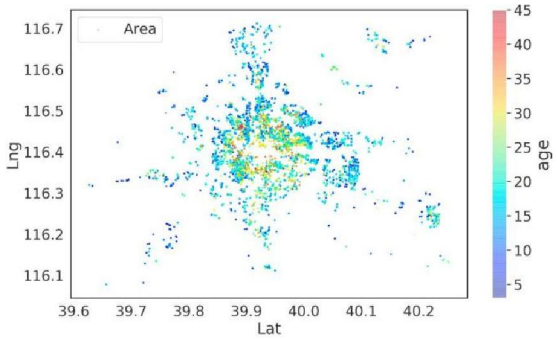

Fig. 1. Age Distribution

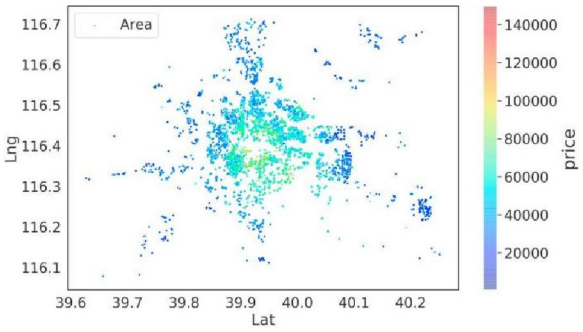

Fig. 2. Price Distribution

Data Analysis

Exploratory data analysis is an essential step before building a regression model. In this way, researchers can discover the implicit patterns of the data, which in turn helps choose appropriate machine learning approaches.

Fig. 1 features houses in Beijing as data points on the map of Beijing. In Fig. 1, the oldest houses are concentrated densely in the center of Beijing, while the newest ones are spread sparsely in the suburban areas. In Fig. 2, the most expensive houses centralizes close to the center of Beijing, while the cheapest ones spreads in the suburban periphery. Since the patterns in Fig. 1 and Fig. 2 are similar, a strong correlation between location, age, and price can be observed. There are also noticeable differences in housing prices across 13 districts, which are summarized in Fig. 3.

Besides the location features, other features of the house also significantly contribute to the models' performance. In Fig, 4, the difference in price among several building types can be clearly illustrated. Since bungalow is an old building type and is more likely to be built close to the center of Beijing, its price is costly regardless of the small area of a house (Fig. 5).

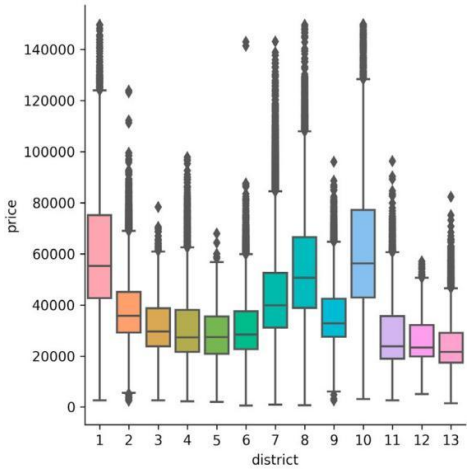

Fig. 3. Correlation between District and Price

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

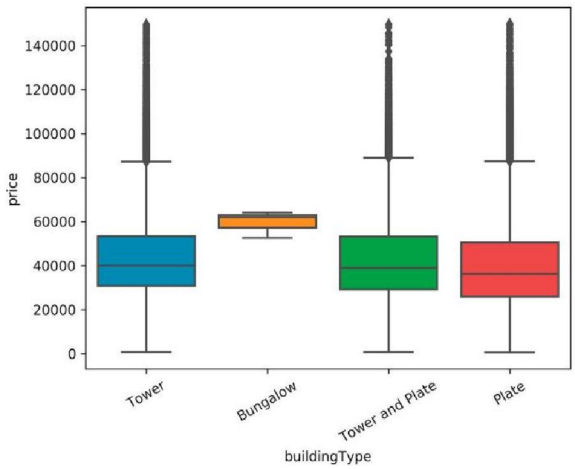

Fig. 4. Correlation between Building Type and Price

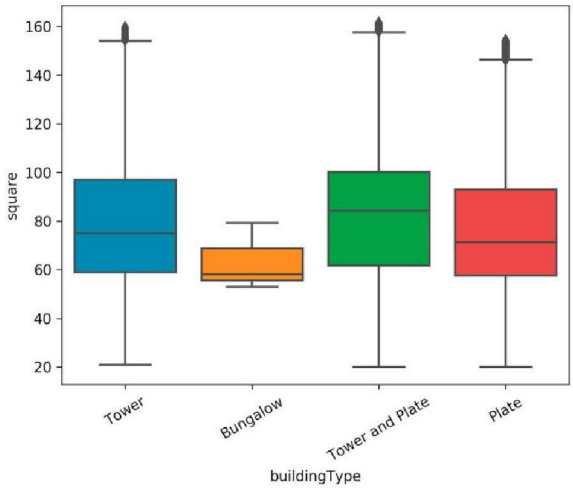

Fig. 5. Correlation between Building Type and Area

Model Selection

Before building models, the data should be processed accordingly so that the models could learn the patterns more efficiently. Specifically, numerical values were standardized, while categorical values were one-hot-encoded. After being processed, the dataset included 58 features. Fig. 6 illustrates the cumulative explained variance. Since most of the features are categorical, it is evident that the variance almost converges at the 30th component. Then, the dataset was split into training set and test set with a ratio of 4 : 1 by utilizing the scikit-learn package [9].

Evaluation function used in this paper is Root Mean Squared Logarithmic Error (RMSLE). This function is illus- trated as follow:

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

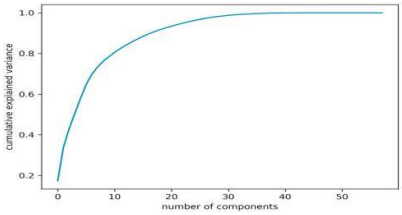

Fig. 6. Cumulative Explained Variance

Random Forest

Random Forest is a kind of ensemble models that combines the prediction of multiple decision trees to create a more accurate final prediction. Random Forest is a verified powerful tool based on previous studies [10]. The random forest algorithm can be summarized in the following steps by S. Raschka and V. Mirjalili in their book "Python Machine Learning" [11]:

Draw a random bootstrap sample of size *n* (randomly choose *n* samples from the training set with replacement).

Grow a decision tree from the bootstrap sample, at each node:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

Randomly select *d* features without replacement.

Split the node using the feature that provides the best split according to the objective function, for instance, maximizing the information gain.

Repeat the steps 1-2 *k* times.

Aggregate the prediction by each tree to assign the class label by majority vote.

In this paper, we used the RandomForestClassifier class provided by sklearn. The RandomForestClassifier has a n_estimators parameter that allows to indicate how many trees to build, which we set at 900. While adding more trees to the random forest normally improves accuracy, it also increases the overall training time of the model. The class also includes the bootstrap parameter which we set to True.

Regarding random forest subsets, however, only a constrained set of features will be used to introduce varia- tion into the trees. By iterating the model multiple times, we also added a few parameters when we initialized the RandomForestClassifier to further enhance the performance:

Set max_depth = 20, which restricts these trees can only go to 20. Set min_samples_split = 10, which only allows a node that should have at most ten rows before it can be split.

The result of this model on training data is exceptional, 0.12980. The RMSLE of the training set is the lowest among other methods. Fig. 7 illustrates the performance of this model on the training set where the X-axis is the prediction result and the Y-axis is the actual housing price.

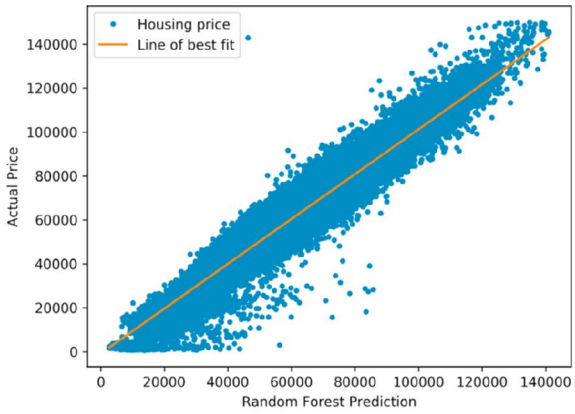

Fig. 7. Random Forest for Training Data

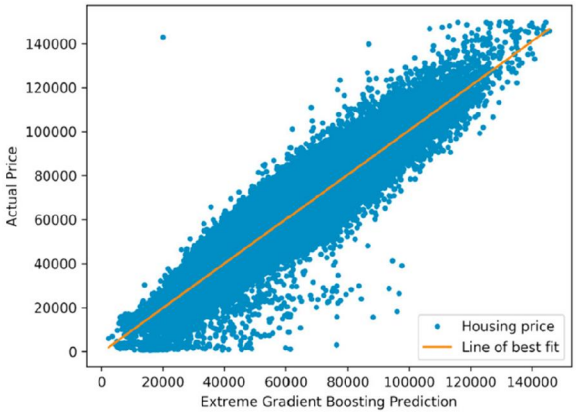

Fig. 8. Extreme Gradient Boosting for Training Data

Extreme Gradient Boosting (XGBoost)

XGBoost is a scalable machine learning system for tree boosting. The system is available as an open-source pack- age. The system has generated a significant impact and been widely recognized in various machine learning and data mining challenges [12].

The most crucial reason why XGBoost succeeds is its scalability in all scenarios. The system runs more than ten times faster than existing popular solutions on a single machine and scales to billions of examples in distributed or memory-limited settings. The scalability of XGBoost is due to several major systems and algorithmic optimizations including a novel tree learning algorithm for handling sparse data and a theoretically justified weighted quantile sketch procedure enabling instance weight handling in approximate tree learning. Parallel and distributed computing make learning faster, which allows quicker model exploration. More importantly, the model exploits out-of-core computa- tion and enables data scientists to process a hundred millions of examples on a desktop. Finally, after combining these techniques to make an end-to-end system, it can scale to even more extensive data with the least amount of cluster resources [12]. In this paper, we utilized the XGBRegressor from xgboost open-source package [13]. After tweaking the XGBoost model multiple times, we set our parameter to the following:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

Set learning_rate = 0.1 Set n_estimators = 200 Determined the optimal tree specific parameters min_child_weight = 2, subsample = 1, colsample_bytre
= 0.8 Set requlatization parameter: reg_lambda = 0.45, reg_alpha = 0, gamma = 0.5
The model performed with a high accuracy where the RMSLE of the training set is around 0.16118. Fig. 8 shows the Extreme Gradient Boosting prediction in X-axis, and the actual price in Y-axis for training data.

Light Gradient Boosting Machine (LightGBM)

LightGBM is a gradient boosting framework that uses a tree-based learning algorithm. LightGBM has faster train- ing speed with lower memory usage compare to XGBoost [14]. Moreover, it can also support parallel and GPU learning or handle large scale of data. Even though both of the aforementioned boosting models follow Gradient Boosting Algorithm, XGBoost grows tree level-wise and LightGBM grows tree leaf-wise. There are also detailed dif- ferences in modelling making them outstanding among different gradient boosting models. In this paper, we utilized the LGBMRegressor from lightgbm open-source package [15]. The optimal parameters for the model are listed as follows: Set learning_rate = 0.15

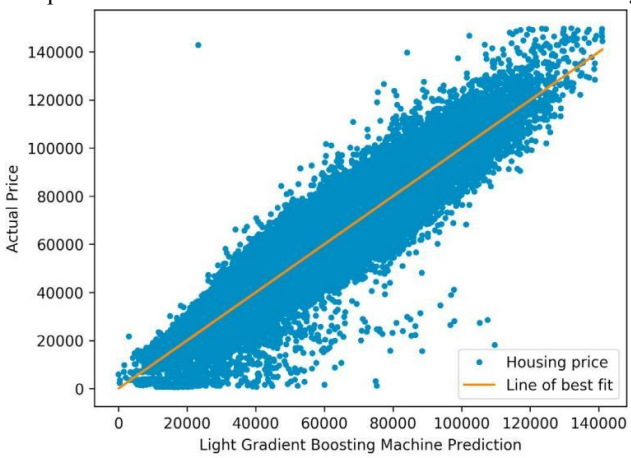

Fig. 9. Light Gradient Boosting Machine for Training Data

Set n_estimators = 64
Set optimal tree specific parameters: min_child_weight = 2, num_leaves = 36, colsample_bytree = 0.8
Set requlatization parameter: reg_lambda = 0.40
When applying the LightGBM method, the model achieved a decent accuracy with the loss of 0.16687 on the training set. Even though the performance was not as good as the other methods, LightGBM is the fastest among all models researched on this paper. Fig. 9 illustrates the performance of this model on the training set where the X-axis is the prediction result and the Y-axis is the actual housing price.

Hybrid Regression

Hybrid Regression Model is a model that includes two or more different regression models. The coupling effect of multiple regression models is proven by S. Lu et al., in which a combination of 65% Lasso and 35% XGBoost achieves a RMSLE of 0.11260 on test set of Kaggle [5]. This result is significantly better than results of Lasso and XGBoost in that paper, which are 0.11449 and 0.11843 respectively [5].
A similar approach was also tested on the "Housing Price of Beijing" dataset. The model consisted of 33.33% Random Forest, 33.33% XGBoost, and 33.33% LightGBM from the previous models. The hybrid regression model achieved 0.14969 on the training set. This method averages the individual predictions to form a final prediction. Therefore, it is a fair approach to balance between bias and variance in the composite models. This technique also supports weight assignment to each component model, but it may lead to bias over one model, losing the benefits of generalization. Fig. 10 illustrates the performance of this model on the training set where the X-axis is the prediction result and the Y-axis is the actual housing price.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

Stacked Generalization

Stacked Generalization is a machine learning ensembling technique introduced by D. Wolpert [7]. A Python pack- age for stacking named "Vecstack" was built and uploaded to GitHub by I. Ivanov [6]. The main idea of this method is to use the predictions of previous models as features for another model. This approach also utilizes the k-fold cross-validation technique to avoid overfitting. This paper used the most common architecture, 2-level stacking architecture, to predict the housing price, where the first stacking level consisted of Random Forest and LightGBM and the second stacking level was XGBoost. The stacking model also used 5-fold cross-validation because the dataset is relatively large. The result of the model on the training set, which is 0.16404, is not as impressive as the Hybrid Regression approach. Fig. 11 shows the prediction of the Stacked Generalization Model on the X-axis and the actual price on the Y-axis.

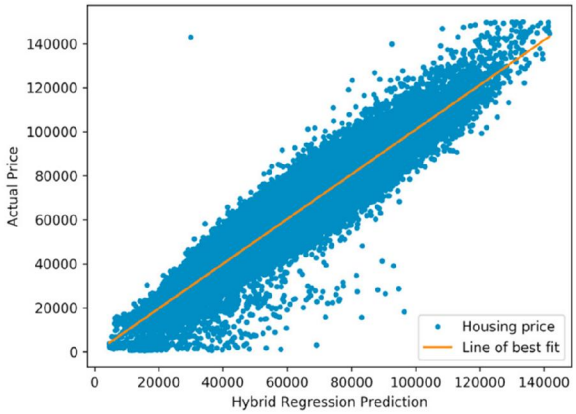

Fig. 10. Hybrid Regression for Training Data

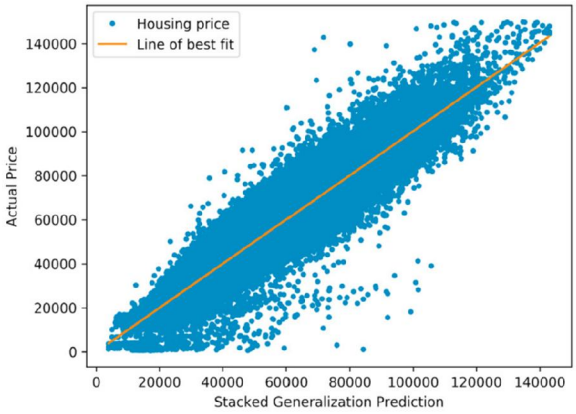

Fig. 11. Stacked Generalization Regression for Training Data

Results

Many iterations of performance tuning were done to find the optimal solution of each model. Random Forest Re- gression, XGBoost, and LightGBM were intensively tuned by function GridSearchCV provided by scikit-learn [9] to achieve the results listed in Table 2. For Hybrid Regression and Stacking methods, performance tuning was not required since both methods were combinations of the best regressions. Instead, architecture implementation could be considered to further enhance the prediction.

Table 2. Prediction Results

Model	***Train Set***	***Test Set***
Random Forest	**0.12980**	0.16568
Extreme Gradient Boosting	0.16118	0.16603
Light Gradient Boosting Machine	0.16687	0.16944
Hybrid Regression	**0.14969**	0.16372
Stacked Generalization Regression	0.16404	**0.16350**

As listed in the Table 2, the best results belong to Random Forest for the training set and Stacked Generalization Regression for the test set. Since 49 of 58 features of the one-hot-encoded dataset were boolean values, it is reasonable that the Random Forest worked well on this dataset. However, Random Forest was prone to overfitting, which led to a decent performance on unseen data. Both XGBoost and LightGBM were not subject to overfitting, but the accuracy of their predictions was not as good as Random Forest on both training and test data.

Unlike the three traditional machine learning methods, Hybrid Regression and Stacked Generalization Regression were neither tuned nor implemented sophisticatedly but delivered promising results on the training set and test set. Since Random Forest was proven to be overfitting, Hybrid Regression could be considered as the best model on training set where the RSMLE is

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

0.14969. Surprisingly, Stacked Generalization Regression did not work well on the training set as Hybrid Regression, but this model did exceptionally on the test set. This is probably because of the two reasons:

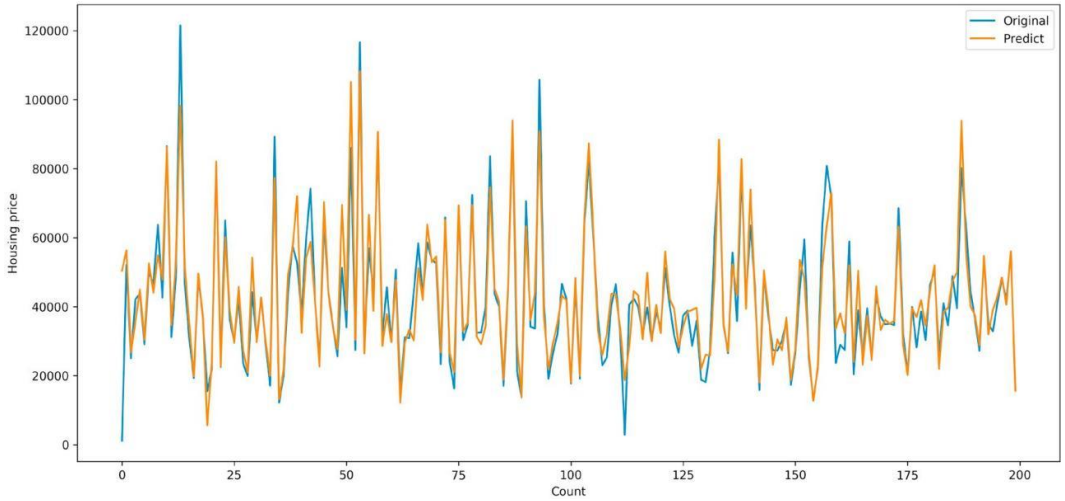

Fig. 12. Comparison of Hybrid Regression's predicted results and original test set

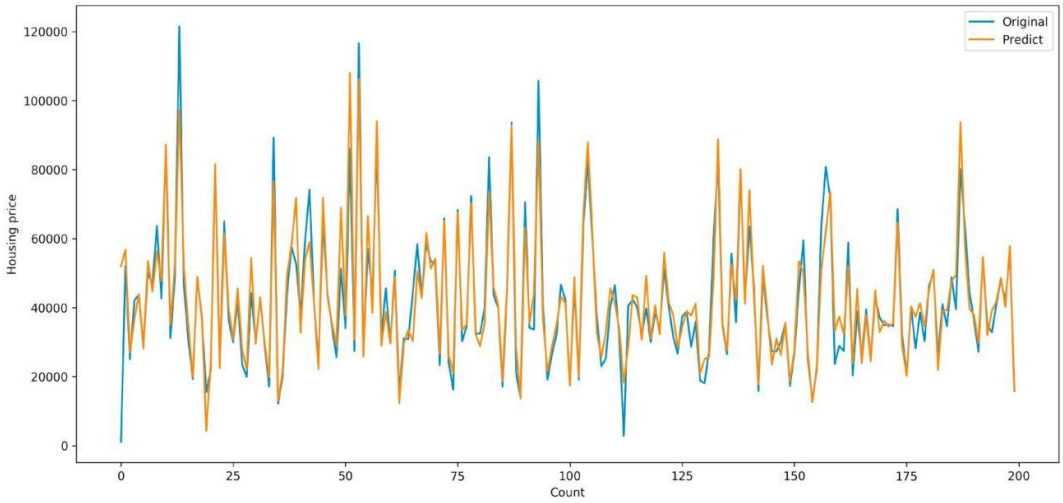

Fig. 13. Comparison of Stacked Generalization Regression's predicted results and original test set

K-fold cross-validation: k-fold cross-validation is a suitable method to find an acceptable bias-variance trade- off. Stacking Regression utilizes this technique to obtain the generalization performance for each component model.

Coupling effect of multiple regressions: different regression methods may support each other. The second stack- ing level can learn again and predict the housing prices accurately based on the pre-estimated prices from the first stacking level. Even though Hybrid Regression shares the coupling effect of multiple regressions with Stacked Generalization Re- gression, Hybrid Regression is less competitive because its learning mechanism is only averaging every prediction. Based on Fig. 12 and Fig. 13 which illustrate the performance of both models on the first 200 observations of the test set, the difference between the two models is not considerably large. On one hand, Hybrid Regression predicts more accurately on outliers (extremely high or low housing prices). On the other hand, Stacked Generalization Re-gression performs better on common observations (standard housing prices). This proves that Stacked Generalization Regression is slightly better than Hybrid Regression in generalization.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 6, February 2025
Impact Factor: 7.67

III. DISCUSSION AND CONCLUSION

This paper investigates different models for housing price prediction. Three different types of Machine Learning methods including Random Forest, XGBoost, and LightGBM and two techniques in machine learning including Hybrid Regression and Stacked Generalization Regression are compared and analyzed for optimal solutions. Even though all of those methods achieved desirable results, different models have their own pros and cons. The Random Forest method has the lowest error on the training set but is prone to be overfitting. Its time complexity is high since the dataset has to be fit multiple times. The XGBoost and LightGBM are decent methods when comparing accuracy, but their time complexities are the best, especially LightGBM. The Hybrid Regression method is simple but performs a lot better than the three previous methods due to the generalization. Finally, the Stacked Generalization Regression method has a complicated architecture, but it is the best choice when accuracy is the top priority. Even though Hybrid Regression and Stacked Generalization Regression deliver satisfactory results, time complexity must be taken into consideration since both of them contain Random Forest, a high time complexity model. Stacked Generalization Regression also has K-fold cross-validation in its mechanism so it has the worst time complexity. Further research about the following topics should be conducted to further investigate these models, especially the combinations of different models:

The coupling effect of multiple regression models. The "re-learn" ability of machine learning models. The combination of Machine Learning and Deep Learning methods. The driven factors for the good performance of tree-based models. The faster ways to fit complex models.

REFERENCES

[1]. House Price Index. Federal Housing Finance Agency. https://www.fhfa.gov/ (accessed September 1, 2019).

[2]. Fan C, Cui Z, Zhong X. House Prices Prediction with Machine Learning Algorithms. Proceedings of the 2018 10th International Conference on Machine Learning and Computing - ICMLC 2018. doi:10.1145/3195106.3195133.

[3]. Phan TD. Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia. 2018 International Confer- ence on Machine Learning and Data Engineering (ICMLDE) 2018. doi:10.1109/icmlde.2018.00017.

[4]. Mu J, Wu F, Zhang A. Housing Value Forecasting Based on Machine Learning Methods. Abstract and Applied Analysis 2014;2014:1–7. doi:10.1155/2014/648047.

[5]. Lu S, Li Z, Qin Z, Yang X, Goh RSM. A hybrid regression technique for house prices prediction. 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) 2017. doi:10.1109/ieem.2017.8289904.

[6]. Ivanov I. vecstack. GitHub 2016. https://github.com/vecxoz/vecstack (accessed June 1, 2019). [Accessed: 01-June-2019].

[7]. Wolpert DH. Stacked generalization. Neural Networks 1992;5:241–59. doi:10.1016/s0893-6080(05)80023-1.

[8]. Qiu Q. Housing price in Beijing. Kaggle 2018. https://www.kaggle.com/ruiqurm/lianjia/ (accessed June 1, 2019).

[9]. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. The Journal of Machine Learning Research 2011;12:2825–30.

[10]. Breiman L. Random Forests. SpringerLink. https://doi.org/10.1023/A:1010933404324 (accessed September 11, 2019).

[11]. Raschka S, Mirjalili V. Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow. 2nd ed. Birmingham: Packt Publishing; 2017.

- Chen T, Guestrin C. XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 16 2016. doi:10.1145/2939672.2939785.

[12]. DMLC. xgboost. GitHub. https://github.com/dmlc/xgboost (accessed June 1, 2019).

[13]. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30 2017:3146–54.

[14]. Microsoft. LightGBM. GitHub. https://github.com/microsoft/LightGBM (accessed June 1, 2019)