International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhancing House Price Forecasting with Stacking Regression and Multiple Machine Learning Approaches

Kelvin Leomitro[1], Alexander Agung Santoso Gunawan[2]

Submitted: 13/11/2022 **Revised:** 12/01/2023 **Accepted:** 09/02/2023

Abstract: This paper introduces a comprehensive algorithmic model for predicting house prices, addressing the absence of a standard reference for property valuation. To counter this, we utilize a set of machine learning techniques that consider various house attributes and features, thereby providing a more standardized approach to house pricing. The dataset used in this study is obtained from Kaggle. A range of algorithms, including Gradient Boost, Support Vector Regression, Decision Tree, Random Forest, Bagging Tree, Ridge, Lasso, Elastic Net, and Stacking Regression, are applied to improve prediction accuracy. Stacking Regression demonstrates the potential for achieving superior prediction scores compared to conventional algorithms. Our experimental results reveal that a Stacking Regression model incorporating Gradient Boosting, Bagging Tree, and Ridge as input algorithms outperforms the other methods, yielding a lower RMSE score. After parameter tuning, the best RMSE score attained with general algorithms was 0.11157 using the Ridge algorithm. In contrast, the Stacking Regression model delivered the best RMSE score of 0.10954, highlighting its enhanced predictive capabilities for house prices.

Keywords: *House Price Prediction, Machine Learning, Stacking Regression, Supervised Learning*.

1. Introduction

Property serves as a fundamental human requirement, not only as a residential asset but also as an investment opportunity. However, the advent of the COVID-19 pandemic has brought about a slowdown, even a halt, in the growth of numerous business sectors. This impact extends to Indonesia's financial landscape, including the property industry. As the demand for properties decreases, the prices of most properties are correspondingly declining. This decline can be attributed to the heightened financial prudence exercised by individuals during a pandemic situation. Even amidst unfavorable circumstances, certain positive opportunities can emerge. Given the declining demand and property prices, potential buyers stand to benefit, as they can purchase properties at lower than usual costs. The determination of property prices typically depends on a plethora of factors including surface area, building area, the number of bathrooms and bedrooms, location, materials used, interior design, availability of a garage, among others [1]. This variety of features and advantages inherently contribute to the wide range of property prices, each uniquely impacting the property's value.

As the landscape of technology continues to evolve at an accelerated pace, Machine Learning comes into the spotlight. This instrument, capable of enhancing numerous daily tasks including decision-making processes, is widely applicable across diverse scenarios. As illustrated by Subudhi et al., machine learning has been deployed to assist medical decision-making amidst the COVID-19 pandemic [2]. This suggests the potential of machine learning in other sectors, such as aiding in predicting property prices. Machine learning, as a technological instrument, develops algorithms that learn and reason, replicating human cognitive processes. This discipline amalgamates aspects of various fields including statistics, artificial intelligence, probability, and computer science among others. The effectiveness of machine learning is largely predicated on the quality of training data; superior data quality tends to deliver more accurate results.

This study aims to construct an algorithmic model to predict property prices, employing a supervised learning approach. Supervised learning utilizes algorithms that learn from datasets where the target features are already labelled. A selection of algorithms, commonly used for regression tasks, have been employed. These include Gradient Boost, Support Vector Regression, Decision Tree, Random Forest, Bagging Tree, Ridge, Lasso, Elastic Net, and Stacking Regression. Regression techniques are classified into three categories: tree-based, linear-based, and distance-based.

In our study, we incorporate several tree-based methods including Gradient Boosting, Decision Tree, Bagging Tree, and Random Forest. Each model presents unique characteristics. Gradient Boosting is a method that generates base models in sequence, focusing on challenging training instances to enhance prediction accuracy [3]. The Random Forest technique is an ensemble method that generates multiple decision trees, employing randomly selected subsets of training samples and variables [4]. A Decision Tree is constructed using a top-down approach, initiating from a root node and subsequently partitioning the data into subsets that consist of instances with similar values at each stage [5]. Lastly, the Bagging Tree method is an algorithm that merges the principles of bagging and decision trees. Here, the dataset is partitioned into several subsets, with a decision tree executed for each subset [6].

Our study incorporates linear-based techniques such as Ridge, Lasso, and ElasticNet, and a distance-based technique represented by Support Vector Regression. Ridge, Lasso, and ElasticNet are enhanced versions of the linear regression algorithm, each

[1] Computer Science Department - Bina Nusantara Graduate Program - Master of Computer Science, Jakarta, Indonesia, 11480
Email: kelvin.leomitro@binus.ac.id
ORCID ID : 0009-0002-3186-5368
[2] Computer Science Department - School of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480
Email: aagung@binus.edu
ORCID ID : 0000-0002-1097-5173

exhibiting distinct characteristics. Lasso technique helps eliminate some features and reduces overfitting in the model, whereas Ridge technique suppresses features that exert minimal influence on the target attribute. ElasticNet is a hybrid approach, amalgamating principles from both Ridge and Lasso to construct the model [7]. On the other hand, Support Vector Regression, a distance-based algorithm, formulates the function approximation problem as an optimization challenge. It strives to find the narrowest tube centered around the surface, while minimizing the prediction error - the distance between the predicted and the actual output [8].

This study aims to introduce the application of Stacking Regression, a multi-level algorithm, as an ensemble learning approach to predict house prices. Stacking Regression, often used in Kaggle competitions for its enhanced accuracy [9], leverages pre-built model algorithms as inputs for the Stacking method. The paper is structured as follows: the first section presents the challenges associated with house price prediction and discusses how Stacking Regression can offer a solution. The second section provides a literature review concerning house price prediction. The third section delineates the research methodology implemented in this study. The fourth section introduces the proposed model, and the fifth offers concluding remarks. The study findings illustrate the superior predictive capabilities of the proposed Stacking Regression model, which delivered the best RMSE score of 0.10954 in house price predictions.

2. Literature Review

Several studies have explored home price prediction using machine learning, often utilizing the same dataset as in our study. Different machine learning models such as Ridge, Lasso, Gradient Boost, and Hybrid Regression were applied. Hybrid Regression is a technique that merges multiple algorithms into a single model. The dataset is divided for each algorithm used, then the results are integrated for the final Hybrid Regression model. One study combined 65% Lasso and 35% Extreme Gradient Boosting, yielding an RMSE score of 0.1149 [10]. In another study, Abbasi et al. investigated property price prediction, adding four new features ("Remodeled", "Seasonality", "New House", and "Total Area") during the feature engineering process and creating dummy data for categorical-type features. They used algorithms such as Random Forest, Elastic Net, Lasso, and Ridge, achieving the best RMSE score of 0.1113 with the Ridge algorithm [11]. Parasich et al. also conducted research on property price prediction using machine learning with the same dataset. They omitted some features, namely "Street", "Fence Type", "Roof Type", and "Roof Material". Their algorithmic approach incorporated Lasso, Extreme Gradient Boosting, Elastic Net, Neural Network, Residual Regressor, and several variations on these algorithms. They achieved the best RMSE score using an ensemble of Lasso, XGBoost, Random Forest, and Neural Network, combined with a Residual Regressor, resulting in an RMSE score of 0.11093 [12].

3. Material and Methods

The initial step in developing an algorithmic model involves data preparation and optimization. Enhanced data optimization can yield more efficient algorithms and produce improved prediction scores. The dataset utilized in this study is "House Price Prediction – Advanced Technique Regression" from Kaggle.com. This study discusses various methods for data optimization, including handling of null values, outlier management, and feature engineering. The dataset comprises 79 features and 1459 rows, with 6965 null values spread across 18 features.

Table 1. Features on Dataset.

Feature	Description
1stFlrSF	First Floor Square Feet
2ndFlrSF	Second Floor Square Feet
3SsnPorch	Three Season Porch Square Feet
Alley	Property Alley
BsmtFullBath	Basement Full Bathroom
BasementHalfBath	Basement Half Bathroom
Bedroom	Numbers of Bedroom
BldgType	Building Type
BsmtCond	Rate Basement Condition
BsmtExposure	Refers to walkout or garden level walls
BsmtFinSF1	Type 1 finished square feet
BsmtFinSF2	Type 2 finished square feet
BsmtFinType1	Rating of basement finished area
BsmtFinType2	Rating of basement finished area (if multiple types)
BsmtQual	Evaluates the height of the basement
BsmtUnfSF	Unfinished square feet of basement
CentralAir	Central Air Conditioning
Condition1	Property Condition
Condition2	Property Condition (if more than one)
Electrical	Electrical Type
EnclosesdPorch	Porch Square Feet
ExterCond	Exterior Condition
Exterior1st	Exterior covering on house
Exterior2nd	Exterior covering on house (if more than one)
ExterQual	Rate Exterior Condition
Fence	Fence Quality
Fireplace	Numbers of Fireplace
FireplaceQu	Fireplace Quality
Foundation	Property Foundation
FullBath	Full bathrooms above grade
Functionality	Home functionality (Assume typical unless deductions are warranted)
GarageArea	Garage Square Feet
GarageCars	Numbers car can park on Garage
GarageCond	Garage Condition
GarageFinish	Rate Garage Interior
GarageQual	Rate Garage Quality
GarageType	Garage Type
GarageYrBlt	Garage Year Built
GrLivArea	Above grade (ground) living area square feet
HalfBath	Half baths above grade
Heating	Property Heating
HeatingQC	Heating Quality
HouseStyle	Property Style
Kitchen	Numbers of Kitchen
KitchenQual	Rate Kitchen Quality
LandContour	Property Land Contour
LandSlope	Property Land Slope
LotArea	Property Surface Area
LotConfig	Property Configuration
LotFrontage	Property Frontage
LotShape	Property Shape
LowQualFinSF	Low quality finished square feet (all floors)
MasVnrArea	Masonry Veneer Area
MasVnrType	Masonry Veneer Type
MiscFeature	Miscellaneous Feature
MiscVal	Miscellaneous Value
MoSold	Month Property Sold
MsSubClass	Identifies the type of dwelling involved in the sale.
MsZoning	Identifies the general zoning classification of the sale.
OpenPorchSF	Porch Square Feet

Neighborhood	Physical locations within Ames city limits
OverallCond	Overall Condition Rate
OverallQual	Overall Quality Rate
PavedDrive	Paved driveway
PoolArea	Pool Area
PoolQC	Pool Quality
RoofMat1	Property Roof Material
RoofStyle	Property Roof Style
SaleCondition	Property Condition When Sold
SaleType	Property Sale Type
ScreenPorch	Screen porch area in square feet
Street	Property Street Type
TotalBsmtSF	Total square feet of basement area
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
Utilities	Property Utilities Availability
WoodDeckSF	Wood Deck Square Feet
YearBuilt	Property Year Built
YearRemodAdd	Property Year Remodelled
YrSold	Property Year Sold

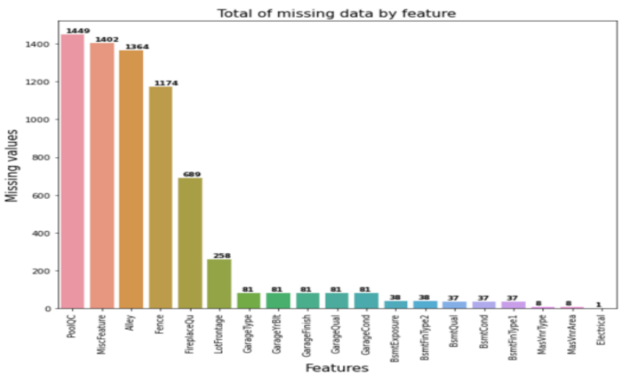

Fig 1. Missing Values on Data

Null values in a dataset do not necessarily indicate missing data. In some instances, these null values signify the absence of a certain feature in a property. For example, a null value in the PoolQC column could imply that the property does not have a pool. Conversely, there are situations where null values indeed represent missing information. A null value in LotFrontAge could be an instance of this, as it would be unusual for a house to lack a frontage. Therefore, null value handling varies depending on the feature in question.

Table 2. Null Value and Handling Method.

Number Of Null	Feature	Handling Method	Description
259	LotFrontAge	Average based on Neighborhood category	Neighborhood usually same
1369	Alley	No Alley	Null value means doesn't have alley
8	MasVnrType	No Masonry	Null value means doesn't have masonry veneer
8	MasVnrArea	0	Null value means doesn't have masonry veneer
37	BsmtQual	No Basement	Null value means doesn't have Basement
37	BsmtCond	No Basement	Null value means doesn't have Basement
38	BsmtExposure	No Basement	Null value means doesn't have Basement
37	BsmtFinType1	No Basement	Null value means doesn't have Basement
38	BsmtFinType2	No Basement	Null value means doesn't have Basement
1	Electrical	Median of Electrical Feature	Can filled by which electrical most used
690	FireplaceQu	No Fireplace	Null value means doesn't have fireplace
81	GarageType	No Garage	Null value means doesn't have a garage
81	GarageFinish	No Garage	Null value means doesn't have a garage
81	GarageQual	No Garage	Null value means doesn't have a garage
81	GarageCond	No Garage	Null value means doesn't have a garage
1453	PoolQC	No Pool	Null value means doesn't have a Pool
1179	Fence	No Fence	Null value means doesn't have a fence
1406	MiscFeature	No Misc Feature	Null value means doesn't have a misc feature

Once null values are addressed, the next step involves handling outliers, which can be considered anomalous data points. These outliers can potentially lead to inaccuracies in the model as they contribute abnormal or erroneous data to the algorithm [13]. Various methods can be employed to remove outliers, one such approach being the use of Z-score. The Z-score quantifies the deviation of a data point from the mean of the entire dataset. A data point is considered an outlier if it deviates significantly from the norm [14]. Typically, a Z-score within the range of -3 to 3 is deemed acceptable. In this study, the Z-score analysis revealed that 265 rows contained outliers. However, as some rows contained more than one outlier, only 227 rows needed to be removed to eliminate all outliers. Consequently, the total data left for analysis after outlier removal stood at 1233 rows.

After addressing null values and handling outliers, the next crucial step is feature engineering, a method that aims to optimize the dataset. Feature engineering encompasses various techniques, and one approach is feature filtering. Feature filtering involves the removal of non-important features from the dataset. To implement feature filtering, the KBest method is one of several approaches that can be utilized to identify and retain the most relevant features for analysis.

Table 3. KBest Result

Trial	Numbers of Feature	RMSE Score
1	10 Feature	0.15692
2	15 Feature	0.13993
3	20 Feature	0.14001
4	25 Feature	0.1366
5	79 Feature (Without Filtering)	0.12336

During the feature engineering process, the KBest score test revealed that the RMSE score is superior when no feature filtering is applied compared to when some features are eliminated. This suggests that all features play a significant role in predicting the target feature, indicating that it is better to retain all features rather than eliminating any. The next step in feature engineering involves eliminating irrelevant features while incorporating potentially important ones. Upon examination, two features were identified as not directly influencing house prices: "YrSold" (Year Sold) and "MoSold" (Month Sold). These features primarily provide information rather than directly impacting prices; thus, they were eliminated. Additionally, the features "YearBuild" and "YearRemodAdd" were found to be similar. "YearBuild" represents the year the property was first built, while "YearRemodAdd" indicates the year of remodeling (or is the same as "YearBuild" if no remodeling has occurred). As the year of remodeling is generally deemed more pertinent to buyers, "YearBuild" was removed, focusing solely on "YearRemodAdd". Lastly, new features were introduced, starting with "HouseAge". This feature denotes the age of the property at the time of sale and is calculated by taking the difference between "YearRemodAdd" and "YrSold".

The final step in feature engineering is feature transformation. Some algorithms require features to follow a normal distribution, particularly distance-based algorithms. To meet this requirement, a transformation was applied to the target feature. In this study, a log transformation was used to transform the target feature.

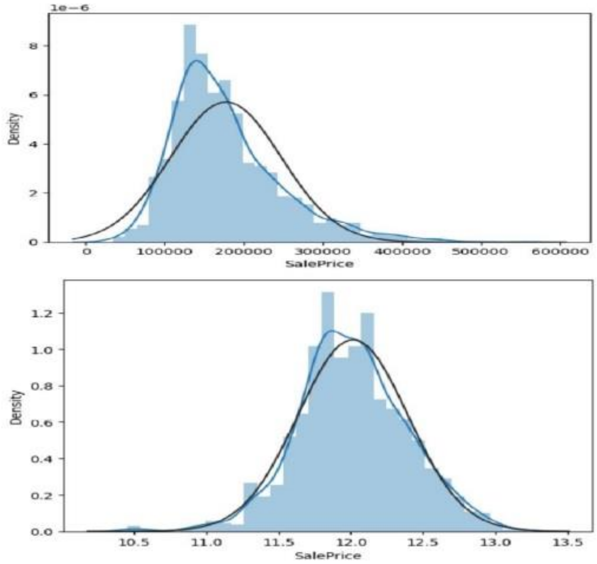

Fig 2. Log Transformation Result

As seen in the figure before it was transformed, the data is skewed to the left, which means the data is not normally distributed. After the log-transformation, the feature data is distributed centered, so we can assume that the feature data is normally distributed.

After transforming the target feature, another feature needs to be transformed too because it is not distributed normally. Boxcox is another method we can use to check the normality of the feature data. Boxcox can produce skewness score, which means that if the skewness score is between -0.5 and 0.5, the feature is normally distributed. The result of the boxcox test on another numeric feature is 21 features not normally distributed, that is PoolArea, 3SsnPorch, LowQualFinSF, MiscVal, KitchenAbvGr, BsmtFinSF2, BsmtHalfBath, ScreenPorch, EnclosedPorch, MasVnrArea, OpenPorchSF, LotArea, WoodDeckSF, BsmtUnfSF, 2ndFlrSF, BsmtFinSF1, 1stFlrSF, Fireplaces, HalfBath, GrLivArea, HouseAge (the feature explanation is in table 1).

After doing all the feature engineering, the dataset has only 1233 rows and a total of 78 features that will be used to model the data. For the modelling data process, the order is to test basic algorithms with default parameters, then try to use hyper parameter tuning to find best parameter for each algorithm, and re-test the algorithm model using hyper parameters. The last one is testing the stacking regression, which in this step will find the best algorithm combination as input and regressor for stacking regression.

For evaluating models, the RMSE score will be used. Mostly from previous researchers using RMSE scores to evaluate the models. KFold will also be used in the modeling of the data process. KFold is a process that splits a dataset into k numbers and leaves one split for evaluate another split [15]. For the example numbers fold we declare is 5, the dataset will split into 5, 4 split data will used as training and 1 split data will used as testing. This process is repeated k times with another combination training and test set. For the numbers of fold, there are no standard numbers, so some tests will do.

Table 4. Best Fold Numbers

Number of Fold	RMSE Score
5	0.12646
10	0.12493
15	0.12424
20	0.12311
25	0.12314

As the results of the findings best folds by RMSE score show, 20 and 25 folds show a similar score, but 20 folds is a better score than 25 folds. This test is limited to 25 folds because the processing time is too long when testing more than 25 folds. After everything is set, algorithm models are ready to be built. For the basic algorithm test, parameters used are parameters used are those that default to that algorithm.

Table 5. Basic Algorithm Modelling Score

Algorithm	RMSE Score	MAE Score
Gradient Boosting	0.12311	0.0863
Support Vector Regression	0.23744	0.17853
Random Forest	0.13546	0.09304
Decision Tree	0.20556	0.14296
Bagging Tree	0.14437	**0.10086**
Ridge	**0.11866**	0.11157
Lasso	0.24805	0.15858
Elastic Net	0.24860	0.11607

After conducting algorithm tests, it was determined that the Ridge algorithm achieved the best RMSE score, while the Bagging Tree algorithm yielded a superior MAE score. Notably, Gradient Boosting, Random Forest, and Bagging Tree exhibited similar scores to the Ridge algorithm. Hyperparameter tuning can be performed by manually assigning customized values to the parameters [16]. An alternative approach is to utilize GridSearchCV, which aids in identifying the optimal parameter values. GridSearchCV works by providing a range of parameter choices, allowing it to determine the best parameter values based on the assigned range.

Table 6. Grid Search CV Hyperparameter and Result

Algorithm	Parameter
Gradient Boosting	• Learning Rate = 0.1 • Estimators = 1000 • Max Depth = 1 • Min Sample Split = 2
Support Vector Regression	• C = 500 • Kernel = rbf
Random Forest	• Estimatos = 200 • Min Sample Split = 2 • Min Sample Leaf = 2 • Max Depth = 20
Decision Tree	• Splitter = random • Max Depth = 10 • Min Sampe Leaf = 2 • Max Depth = 20
Bagging Tree	• Estimators = 400 • Max Samples = 0.1 • Max Features = 0.1
Ridge	• Alpha = 11
Lasso	• Alpha = 0.01
Elastic Net	• Alpha = 0.01

After testing some numbers for parameters, the results show that the best numbers of parameters are based on Table 6. After doing a hyper-parameter test, model algorithms will be re-tested and see how they differ between before and after hyper-parameter tuning.

Table 7. Basic Algorithm Compared After Hyperparameter

Algorithm	RMSE Score Before	RMSE Score After Hyperparameter
Gradient Boosting	0.12311	0.0863
Support Vector Regression	0.23744	0.17853
Random Forest	0.13546	0.09304
Decision Tree	0.20556	0.14296
Bagging Tree	0.14437	0.10086
Ridge	**0.11866**	**0.11157**
Lasso	0.24805	0.15858
Elastic Net	0.24860	0.11607

Based on hyper-parameter applications, RMSE scores are increasing significantly for some algorithms like Lasso, Elastic Net, and Support Vector Regression. After hyper-parameter testing, the result is Ridge algorithms can still produce a better RMSE, followed by Gradient Boosting and Elastic Net. For the general algorithms, it can be assumed that the best score in RMSE with several tests is based on Table 7. After that, go to stacking regression. For the process of stacking, the first step is to find the best meta regressor for stacking, then start to find the best algorithms input for stacking regression.

Table 8. Finding Meta Regressor

Algorithm	RMSE Score
Gradient Boosting	0.1288
Support Vector Regression	0.14025
Random Forest	0.13344
Decision Tree	0.13195
Bagging Tree	0.13209
Ridge	0.11397
Lasso	0.24805
Elastic Net	0.26809

As shown in Table 8, ridge algorithm can produce a better RMSE score for basic stacking. Algorithms input are all algorithms used in this study, which are Gradient Boosting, Support Vector Regression, Random Forest, Decision Tree, Bagging Tree, Ridge, Lasso, Elastic Net. For the next stacking trial, the ridge algorithm will be used for meta regressor.

For the stacking trial, some method will be used to find the best combination algorithms to input for stacking regression. The first try to classify an algorithm based on its method. There's are tree-based, linear-based, and distance-based. For the tree-based methods, including Random Forest, Decision Tree, Bagging Tree, and Gradient Boost. For the linear based, including Lasso, Ridge, Elastic Net. The Last one for distance-based only including Support Vector Regression.

Table 9. Stacking Regression First Trial

Combination Algorithm	RMSE Score
Tree Based	0.11538
Linear Based	0.11155
Tree Based + Linear Based + Support Vector Regression	0.11349

For the first trial of stacking, the results showed that linear-based algorithms as input can produce a better RMSE. The second trial of stacking regression involves using all algorithms and eliminating them one by one to find the best combination.

Table 10. Stacking Regression Second Trial

Combination Algorithm	RMSE Score
Gradient Boosting, Decision Tree, Bagging Tree, Random Forest, Support Vector Regression, Ridge, Lasso, Elastic Net	0.11358
Gradient Boosting, Decision Tree, Bagging Tree, Random Forest, Support Vector Regression, Ridge, Lasso	0.11536
Gradient Boosting, Bagging Rree, Random Forest, Support Vector Regression, Ridge, Lasso	0.11246
Gradient Boosting, Bagging Tree, Support Vector Regression, Ridge, Lasso	0.11231
Gradient Boosting, Bagging Tree, Support Vector Regression, Ridge	0.10983
Gradient Boosting, Bagging Tree, Ridge	0.10954
Gradient Boosting, Ridge	011077

As on second trial of stacking regression, the best result is 0,10954 with input algorithms combination are Gradient Boosting, Bagging Tree, and Ridge.

4. Result and Discussion

In our experiments, it was found that stacking regression outperformed general algorithms in terms of RMSE score. After parameter tuning, the best RMSE score achieved using general algorithms was 0.11157 with the Ridge algorithm. However, in the stacking test, various combinations were tested, resulting in an improved RMSE score of 0.10954, which is the best score achieved in this study.

Previous research by Lu et al., using the same dataset and research goals, employed Ridge, Lasso, Gradient Boost, and Hybrid Regression for modeling. Their best RMSE score was 0.1126 [11]. Abbasi also utilized the same dataset, employing Random Forest, Elastic Net, Lasso, and Ridge algorithms. Their best RMSE score was 0.1113 [12]. Lastly, Parasich et al., in their article using the same dataset as ours, employed Lasso, XGBoost, Elastic Net, Neural Network, and Residual Regressor as model algorithms, achieving a best RMSE score of 0.11093 [13].

However, there are limitations to our study. Firstly, we used a freely available dataset from Kaggle, which may not fully reflect real-world conditions. It would be preferable to use a real dataset that closely mirrors the actual situation to obtain more accurate prediction scores. Additionally, the limitations of our computational resources impact our study. Machine learning is dependent on computational power, and better computing capabilities would enable more extensive experimentation, such as processing more algorithms and conducting more comprehensive hyperparameter tuning. Moreover, with increased computational resources and more innovative researchers, the optimization potential of machine learning algorithms can be further explored and harnessed.

5. Conclusion

Based on our experimental results, the Stacking Regression model, utilizing Gradient Boosting, Bagging Tree, and Ridge as input algorithms, outperformed other methods, demonstrating a lower RMSE score. In comparison to existing literature, our study successfully optimized the RMSE score through the implementation of a stacking ensemble learning approach. This approach leverages the strengths of various algorithms, leading to improved performance.

Future research directions for this study involve exploring the adaptation of the proposed algorithm with different components. Although our current algorithm is derived from conventional machine learning techniques, there is potential to include neural networks as stacking components. Incorporating neural networks into the stacking framework can provide additional flexibility and further enhance the predictive capabilities of the model.

References

[1] Ahtesham, M., Bawany, N. Z., & Fatima, K., "House Price Prediction using Machine Learning Algorithm - The Case of Karachi City, Pakistan. Proceedings – 2020," 21st International Arab Conference on Information Technology, ACIT 2020, 7-11.

[2] Subudhi, S., Verma, A., & Patel, A. B., "Prognostic machine learning models for COVID-19 to facilitate decision making," International Journal of Clinical Practice, 74(12), 4–7.

[3] Zhang, Y., & Haghani, A., "A gradient boosting method to improve travel time prediction," Transportation Research Part C: Emerging Technologies, 58, 308–324.

[4] Belgiu, M., & Drăgu, L., "Random Forest in remote sensing: A review of applications and future directions," ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24–31.

[5] *Borde, S., Rane, A., Shende, G., & Shetty, S., "Real Estate Investment Advising Using Machine Learning," International Research Journal of Engineering and Technology (IRJET), 4(3), 1821–1825.*

[6] *Zhang, L., Xiong, J., Zhao, H., Hong, H., Zhu, X., & Li, C., "Sleep stages classification by CW Doppler radar using bagged trees algorithm," 2017 IEEE Radar Conference, RadarConf 2017, 0788–0791.*

[7] Melkumova, L. E., and S. Ya Shatskikh. "Comparing Ridge and LASSO estimators for data analysis." Procedia engineering 201 (2017): 746-755.

[8] Awad, M., & Khanna, R., "Support Vector Regression," 2011 1st International EConference on Computer and Knowledge Engineering, ICCKE 2011, Vc, 28–33.

[9] PROCESS Corporation, Boston, MA, USA. Intranets: Internet technologies deployed behind the firewall for corporate productivity. Presented at INET96 Annual Meeting. [Online]. Available: http://home.process.com/Intranets/wp2.htp

[10] Pavlyshenko, B., "Using Stacking Approaches for Machine Learning Models," Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, 255–258.

[11] Lu, S., Li, Z., Qin, Z., Yang, X., & Goh, R. S. M., "A hybrid regression technique for house prices prediction," IEEE International Conference on Industrial Engineering and Engineering Management, 2017-Decem, 319–323.

[12] Abbasi, S., "Advanced Regression Techniques Based Housing Price Prediction Model," 13th International Conference of Iranian Operations Research Society, February, 2020, 1–10.

[13] Parasich, A. V., Parasich, V. A., Kaftannikov, I. L., & Parasich, I. V., "Predicting sales prices of the houses using regression methods of machine learning," RPC 2018 - Proceedings of the 3rd Russian-Pacific Conference on Computer Technology and Applications, 3–7.

[14] Aggarwal, C. C., "Outlier Analysis. In Outlier Analysis," Motorola Semiconductor Data Manual, Motorola Semiconductor Products Inc., Phoenix, AZ, USA, 1989.

[15] Aggarwal, V., Gupta, V., Singh, P., Sharma, K., & Sharma, N., "Detection of spatial outlier by using improved Z-score test," Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019, 2019-April(Icoei), 788–790.

[16] Yang, L., & Shami, A., "On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing," 415, 295–316.