A Hybrid Regression Technique for House Price Prediction

Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh

Smita Nannaware
ITCS-5156 Spring 2022 - Lee

February 28, 2022

Abstract

This report is presented as a replication of a previous work [5]. Any assertions made within are subjective and do not represent those of the original author.

Shelter is basic need of human being. House price prediction is important for every individual. House price index usually represent the price changes of residential housing. However single family house price depends on various other factors such as location, house type, size, local amenities and many more. This paper seeks to explore and re-implement one of the current implementations to achieve house price as closely as possible by examining creative feature engineering and hybrid Lasso and Gradient boost regression technique model to predict individual house price.

1 Introduction

Machine learning develops algorithms and builds models from data and uses them to make prediction on the new data. Supervised learning uses data where outcome is known and unsupervised learning uses data with unknown outcome.

How to use machine learning algorithms to predict the house prices? It is a challenging task to predict the house prices as closely as possible to real values based on the model built. Traditionally House price index is used for summarizing the house price prediction and often other factors such as location, size, house type, city, county are ignored which affects the demands and supply.

Kaggle organized a house price prediction competition [1] which provides data with 79 explanatory variables. For local house price prediction there are many regression techniques to use. Lu et al. [5] has participated in the competition and experimented with Lasso regression, Ridge regression and Gradient Boosting regression along with hybrid combinations between them. Data quality is important so normal distribution test for each feature is done to explore possible transformation to a normal distribution. Many transformation is applied to enhance the linearity of input features.

2 Related work

There are numerous research going on around the House price prediction using non traditional factors affecting the house price.

2.1 Effect of Street-based Local Area on house price

Stephen Law [3] proposed the concept of Street-based Local Area (SLA) where it finds strong links between SLA with house price and it shows that using Street-based local area is better than region-based local area. Study used Modularity optimisation algorithm to detect the Street-based local area as shown in Fig.1

This study used the multi-level hedonic regression model where observed variable is a function of two components, a fixed part and a random part.This is the based model on top which other 5 models were built.

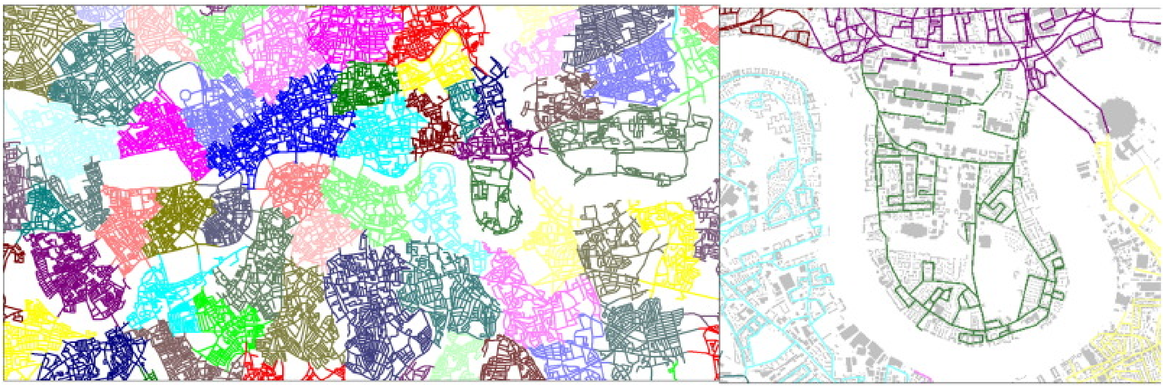


Figure 1: A visualization of SLA for Greater London Area [3]

Eq.(1) multi level regression model [3]

$$\underbrace{Y}_{Observed} = \underbrace{BX_i + \mu}_{Fixed} + \underbrace{\mu_i + \varepsilon_{ijk}}_{Random} \tag{1}$$

Where Y is the observed,
B is the coefficient for predictors,
X_i is the predictor,
μ is mean,
μ_i is the local area random effect

Features such as space syntax integration, floor size, dwelling, number of shops and schools within 800m were considered while building model.

The regression results of this study showed the significant effect of Street-based local area on house price. However research ignored the other factors such as house type, utilities, facilities, etc. affecting house prices. This study gives the Lu et. al [5] a perspective about how non traditional factors can be considered for house price prediction.

2.2 Explaining House Price Dynamics: Isolating the Role of Nonfundamentals

Another non traditional approach is used by David et al. [4] where reduced-form vector autoregression (VAR) models are used to examine dynamic short-run relationship between house prices and sentiments of buyers, home builders, and residential mortgage lenders. The relation between house market sentiments, house prices and market liquidity are expressed in linear functions of their own and each other's lagged values.

First, they sum the estimated lagged coefficients and test joint significance to understand the cumulative effects of lags of endogenous variables as RETURN, SENT and TURN.

After understanding that three sentiment indices are correlated they performed principal component analysis to extract the common sentiment factor.

Next to examine robustness of their results they modified the orthoganalization regressions used to obtain three sentiment proxies followed by use of US Home Price Index to measure real house price appreciation and repeated the VAR analysis using real price appreciation rates.

At last, long-run effect of sentiment is tested using a framework involving regressing future k-period quarterly real housing returns on a vector of control variables and composite sentiment index.

This study reveals the long-run effects of sentiments using overlapping price change regression. This study is based on house market sentiments only whereas there are several other factors such as location, facilities, type of house, etc highly affecting the house prices. This study helps Lu et. al [5] to understand the consideration of different factors and their processing techniques to perform house price regression.

3 Methodology

In this section I will discuss the current approach that is replicated and describe how to apply multiple regression algorithms. At last I will explain the coupling effect of Lasso and Gradient boosting algorithm.

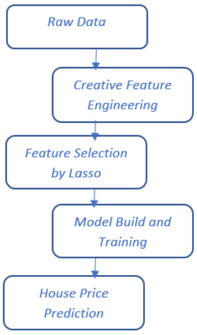


Figure 2: Process Flow

Fig.2 shows the process flow diagram. Kaggle [1] provided the data for this problem. Data is processed using creative feature engineering. From many feature variable some of them selected using Lasso regression for building the regression models. Selected features are then considered for model building and training. Once model is build and tested on train data house prediction is done on test data. The details of the process is explained in next sections.

3.1 Creative Feature Engineering: My version

Investigation of value distribution and correlation of SalePrice with each variable is done and many new variables are introduced. Lu et. al [5] have generated 460 number of features. Following steps are taken as part of feature engineering replication of original source and I was able to generate 323 features.

- Filling missing values with No Value string, median and mode. If there are too many missing values then those features are updated with No Value string. Features as LotFrontage, MasVnrArea are updated with median value and rest of the features are updated with mode value. Fig. 3 shows the null value distribution over features.
- Log transformation is applied to quantitative features such as SalePrice, LotArea, LotFrontage, etc to get Normalized distribution. Fig. 4 shows the log transformation of variables and Fig. 5 their respective distribution before transformation.
- String value features as ExterQual, ExterCond, etc are transformed to discrete values.
- One-hot encoding of remaining qualitative variable is done.
- New features added by multiplying BsmtQual, OverallQual, GarageQual, etc with TotalBsmtSF, LotArea, GarageArea, etc.
- Top correlated features are then transformed by square, cube and square root to add new features.

3.2 Regression Algorithms

There are many regression algorithms which can be used to predict the house price prediction. Lu et. al [5] found that Ridge, Lasso and Gradient boosting are most useful. I have used the same algorithms for this project.

Ridge and Lasso regression models are used when there are high number of input features [5]. To prevent overfitting Ridge regression performs L2 regularization and Lasso regression performs L1 regularization. Decision trees are used as Weak learner in Gradient boosting algorithm and for overfitting prevention subsample parameter is used.

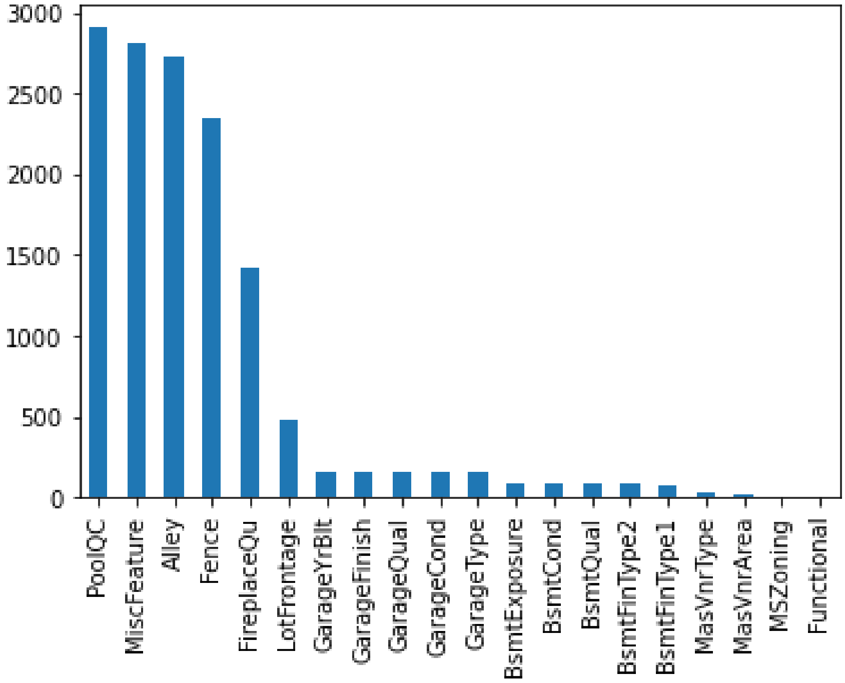


Figure 3: Missing values in Features

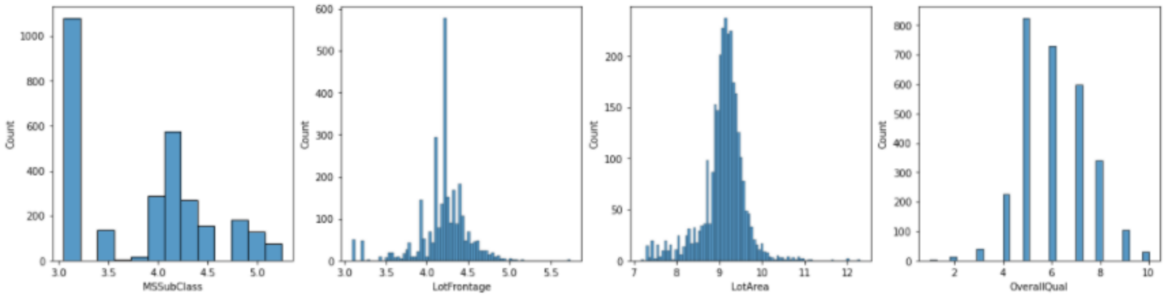


Figure 4: Distribution of features after log transformation

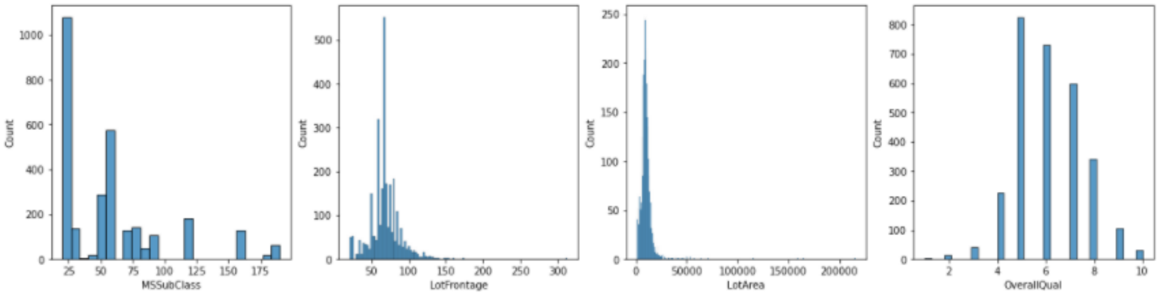


Figure 5: Distribution of features before log transformation

3.2.1 Ridge Regression

Hyperparameter alpha is chosen for Ridge regression. Based on mean square error score, Lu et. al [5] have chosen the alpha values to get maximum score. Fig. 6 shows regression plot between Ridge prediction on X and SalePrice on Y for training data.

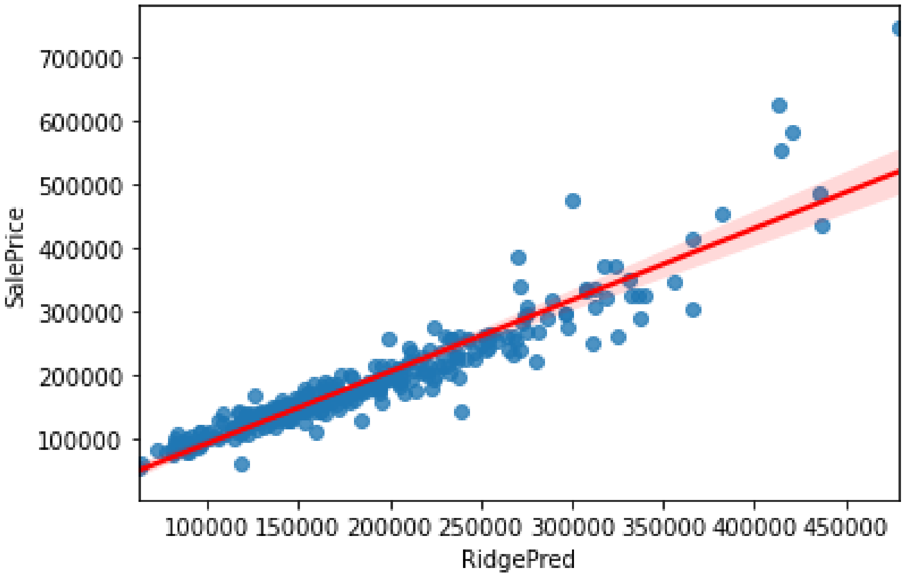


Figure 6: Ridge prediction for Training Data [My result]

3.2.2 Lasso Regression

Here as well hyperparameter alpha is chosen for Lasso regression. To perform feature selection Lasso regression is used which gives meaningful features. Fig. 7 shows regression plot between Lasso prediction on X and SalePrice on Y for training data.

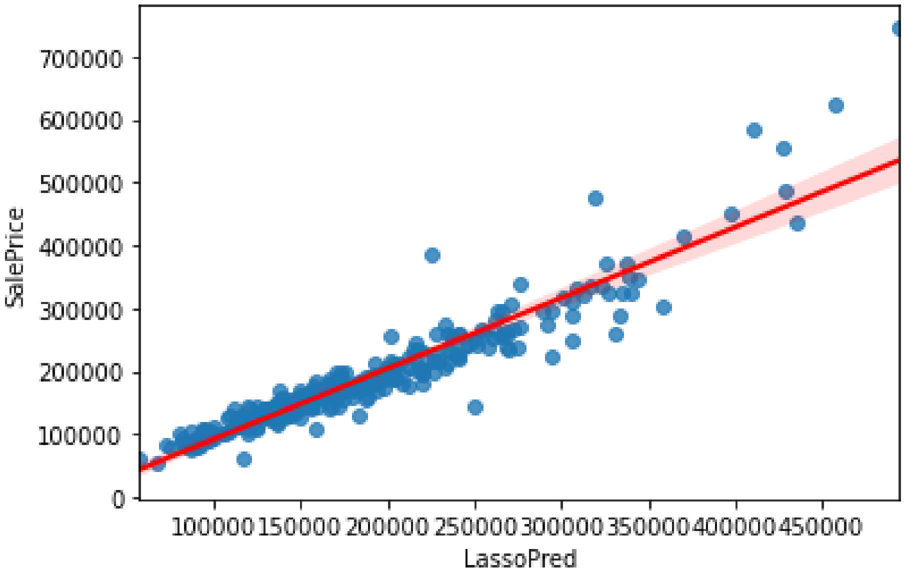


Figure 7: Lasso prediction for Training Data [My result]

3.2.3 Gradient Boosting

There are more than one parameters to choose in case of gradient boosting. Lu et. al [5] have followed below steps to choose parameters.

They fixed the learning rate = 0.1 and number of estimators = 1000 and determined the Decision tree specific parameters as max depth, subsample, gamma, etc for deciding learning rate and number of trees. I have chosen the subsample values experimented by author [5]. Fig. 8 shows Gradient boosting prediction on X and SalePrice on Y for training data.

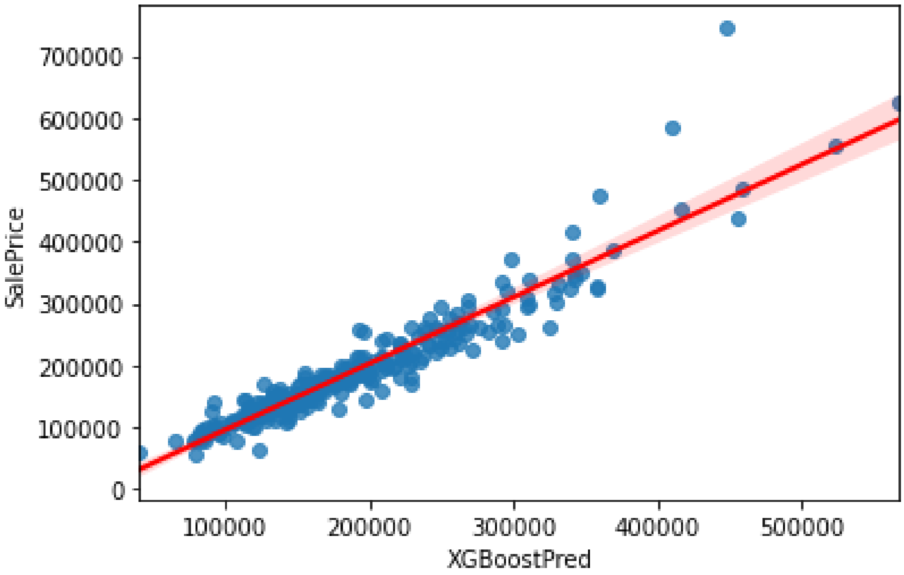


Figure 8: Gradient Boosting prediction for Training Data [My result]

3.3 Hybrid Regression: My version

Lu et. al [5] has tried several hybrid regression combination and found that hybrid regression models perform better than single regression technique. They have not specified which technique they used to combine two or more models, so after research I chose to use Voting regressor from sklearn's ensemble module to combine models as hybrid regression model. Fig. 9 shows the Lasso and Gradient boost hybrid regression model prediction on X and SalePrice on Y for training data.

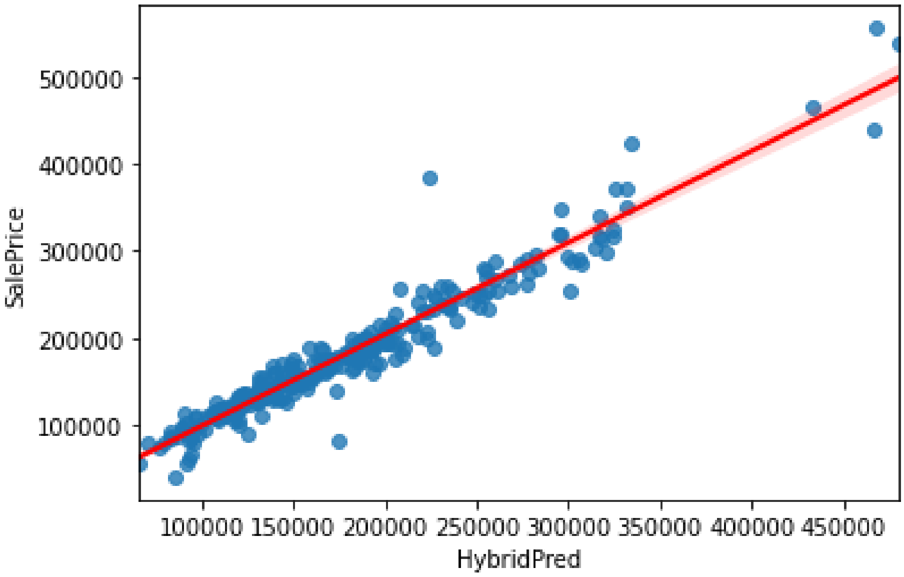


Figure 9: Lasso and Gradient boosting prediction for Training Data [My result]

3.4 Prediction Submission and Valuation

Kaggle house price prediction competition evaluation is Root mean square error (RMSE) (Eq.2) between logarithm of predicted value and logarithm of actual sale price of house.

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

where y_i is log($SalePrice_i$) and $\hat{(}y)_i$ is log(predicted $SalePrice_i$).

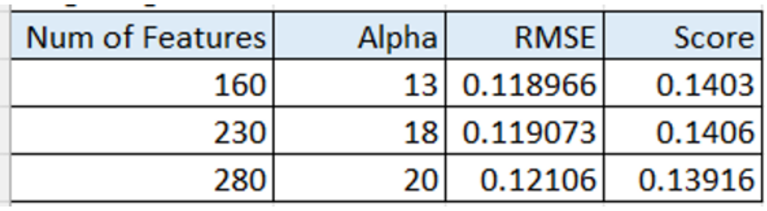
Lower the RMSE value better the result score.

4 Experiments

Lu et. al [5] have published their research based on Kaggle competition on House Prices - Advanced regression techniques. Data for this project is taken from Kaggle [1] website. To replicate the research, I have followed the steps given by authors in their research paper. There are certain steps which are unknown such as detailed feature addition and hybrid regression modelling technique. I performed such steps based on the understanding of relevant topic as detailed in previous section. The hyperparameter values are take from author's paper. The score values are different than that of authors' results as I have generated only 323 feature variables while author has generated 460 features. Also, features that I have generated might not be the same as authors.

4.1 Results: My version

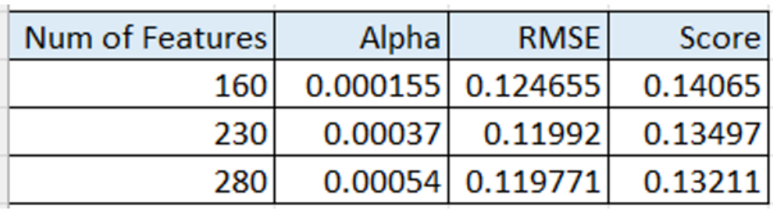
After selection of features using Lasso as mentioned in section 3.2.2 and experimenting with Ridge, Lasso and Gradient Boost regression model I found that the best score is achieved using 280 number of features for Ridge and Lasso regression and 230 number of features for Gradient Boost. Fig.10 shows my results of Ridge regression. RMSE column shows the Root mean square error values for train data and score column shows the score generated by Kaggle for predicted house price submission for test data.



Num of Features	Alpha	RMSE	Score
160	13	0.118966	0.1403
230	18	0.119073	0.1406
280	20	0.12106	0.13916

Figure 10: Ridge regression results [My result]

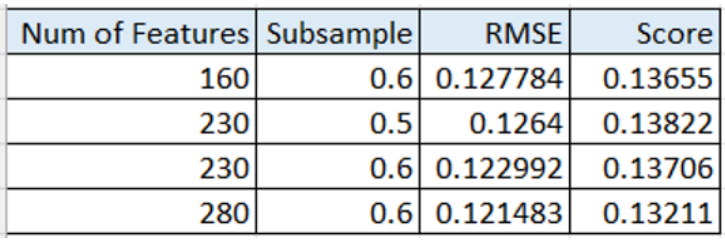
Using Lasso regression I found similar trend where better score is generated for 280 number of features. Fig.11 shows minimum RMSE score is generated for training as well as for test data from Kaggle using 280 features.



Num of Features	Alpha	RMSE	Score
160	0.000155	0.124655	0.14065
230	0.00037	0.11992	0.13497
280	0.00054	0.119771	0.13211

Figure 11: Lasso regression results [My result]

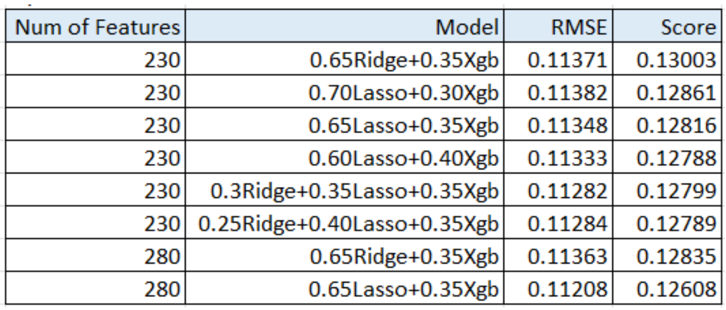
In case of Gradient boost, slightly different results are seen. Fig. 12 shows minimum RMSE score is generated for training data using 280 number of features with subsample as 0.6 whereas minimum score is generated for test data from Kaggle using 230 features.



Num of Features	Subsample	RMSE	Score
160	0.6	0.127784	0.13655
230	0.5	0.1264	0.13822
230	0.6	0.122992	0.13706
280	0.6	0.121483	0.13211

Figure 12: Gradient boost regression results [My result]

Lu et. al [5] tried various combinations of hybrid models for test data predictions. However, I have tried combinations which are listed in the author's paper. In my case, I found that prediction results are better for 280 number of features whereas authors found better results with 230 number of features. Lasso and gradient boost regression achieves best score where the combination is 65% of Lasso with 35% of Gradient Boost. Fig. 13 shows the hybrid results for various combinations.



Num of Features	Model	RMSE	Score
230	0.65Ridge+0.35Xgb	0.11371	0.13003
230	0.70Lasso+0.30Xgb	0.11382	0.12861
230	0.65Lasso+0.35Xgb	0.11348	0.12816
230	0.60Lasso+0.40Xgb	0.11333	0.12788
230	0.3Ridge+0.35Lasso+0.35Xgb	0.11282	0.12799
230	0.25Ridge+0.40Lasso+0.35Xgb	0.11284	0.12789
280	0.65Ridge+0.35Xgb	0.11363	0.12835
280	0.65Lasso+0.35Xgb	0.11208	0.12608

Figure 13: Hybrid regression results [My result]

5 Discussion and Conclusion

In section 4.1 the test results shows that it is useful to create more new features. For skewed features log transformation is very useful and multiplying existing features helps to create new features and thus gives better results. With the help of this project, I learned to perform data cleaning and transformation at best level. I learn to use feature addition and Lasso regression for feature selection which was completely new concept for me.

In section 4.1 the results proves the theory of Lu et. al [5] that the coupling effect of multiple regression works best for house price prediction. Based on results, it is also proved that hybrid regression are better than one of Ridge, Lasso and Gradient Boost algorithm. For test data my best result as 0.11208 is achieved by 65% Lasso with 35% Gradient Boost combination. The effect of hyperparameter tuning and feature selection is understood in this project.

It was challenging to implement code based on information given by author [5] in research paper. My implementation is solely based on how well I understood the research paper [5]. Authors have given enough details of the complete process that they followed. Most difficult task for me was to create new features and I have implemented based on my understanding from data and source paper. Another struggling point was to implement hybrid model where author [5] has not given enough details of actual implementation. After researching about hybrid modelling and ensemble techniques I chose to use VotingRegressor from sklearn. I got to learn many new concepts as data transformation, feature selection, hybrid modelling, how Kaggle competition submission works, and got exposure to end to end machine leaning model development.

As mentioned in section 3.1 there were 460 features used by Lu et. al [5] and I was able to create 323 features only. There is scope to understand the data better and create more significant new features. I used hyperparameters given by author, hyperparameter tuning can be done in future. Also there are more regression algorithms such as Random Forest, Neural Network, etc which have not experimented in this project, it may help to increase prediction accuracy. Also there is a need to verify implementation of hybrid model.

6 My Contributions

I have solely used Lu et. al [5] work as described in their paper. I implemented the code based on my understanding, there is no source from where I have copied the code. However I have used internet search [2] to understand the concept of writing machine learning related code. All the models, hyperparameters and creative feature engineering steps are selected from source paper [5]. Implementation of hybrid model using VotingRegressor is completely based on my understanding to make hybrid models work which needs to be verified.

7 Acknowledgement and Share agreement

Thank you Professor Jake Lee for this term project. I got a chance to learn much more in this project. All the labs and instructions were clear to understand the practical approach of Machine Learning. I would be happy to share my work with students as an example in the following semesters. Thank you.

References

[1] https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data.

[2] https://www.google.com/.

[3] Stephen Law. "Defining Street-based Local Area and measuring its effect on house price using a hedonic price approach: The case study of Metropolitan London". In: *Cities* 60 (2017), pp. 166–179. ISSN: 0264-2751. DOI: https://doi.org/10.1016/j.cities.2016.08.008. URL: https://www.sciencedirect.com/science/article/pii/S0264275116304383.

[4] DAVID C. LING, JOSEPH T.L. OOI, and THAO T.T. LE. "Explaining House Price Dynamics: Isolating the Role of Nonfundamentals". In: *Journal of Money, Credit and Banking* 47.S1 (2015), pp. 87–125. ISSN: 00222879, 15384616. URL: https://www.jstor.org/stable/26614925.

[5] Sifei Lu et al. "A hybrid regression technique for house prices prediction". In: *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2017, pp. 319–323. DOI: 10.1109/IEEM.2017.8289904.