4500 Project Report: House Prices Prediction

Min Fu
mf3200@columbia.edu

Yuqin Xu
yx2478@columbia.edu

Abstract

This project contains two parts including data analysis and house price prediction. Firstly, the distribution of house price was studied and hypothesis testing was proposed to verify the distribution. Then the project visualized data to explore the relationship between attributes of houses. Furthermore, 9 regression models were put forward to predict the SalePrice of house at first. Based on this, this report deigned an weight averaged model containing support vector regression and kernel ridge regression to get a result which is better than any individual model. To improve accuracy more, stacking strategy was applied to train models and improved cross validation accuracy up to 90%. Eventually, this model was used to predict the sale price of house in test data which realized root mean squared error 9.45%.

keywords: house price prediction, data analysis, machine learning

1 INTRODUCTION

The real estate industry is an important industry of world economy, and housing price is one of the most critical factors. Traditional prediction methods are difficult to achieve accurate prediction of house prices. How to mine market information and build more effective indicators and models to improve the predict accuracy becomes an urgent problem. Nowadays, the rapid development of big data and machine learning technology makes it possible to analyze house prices based on web search data, which can make up for the subjectivity of factor selection in traditional predictive model construction and improve the accuracy of prediction. Our project will focus on the analysis of house property and prediction of house price with those important attributes and try to find something interesting from the data.

2 SYSTEM OVERVIEW

We use House Prices dataset containing every aspect of residential homes in Ames, Iowa from Kaggle. Since only train data of this dataset contains SalePrice, we will separate train data as train data and test data in this project. As shown in Fig.1, the train dataset has 1460 observations and 81 attributes which are a mix of numerical and categorical data, such as Type of roof, Height of the basement, Swimming pool area and quality and so on. The last column is the SalePrice of house and other 80 features are factors that might affect the SalePrice. Our purpose is to figure out which features have strong correlation with the SalePrice and predict the SalePrice using the other 80 features.

```
df_train.head(5)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoS
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

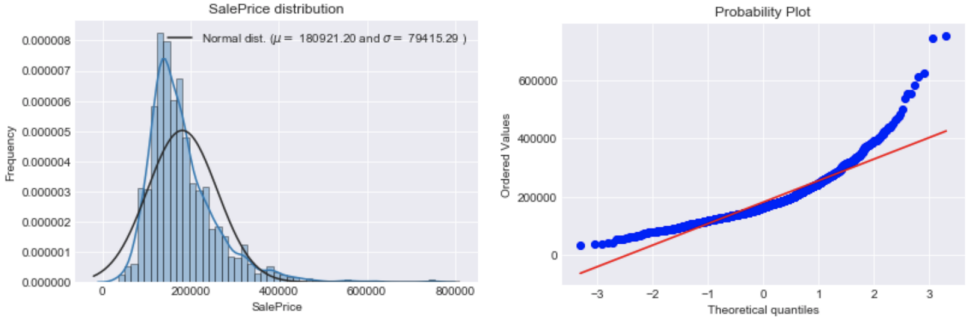
5 rows × 81 columns

Fig. 1. Dataset

3 DATA ANALYSIS

3.1 Question 1: What is the distribution of SalePrice?

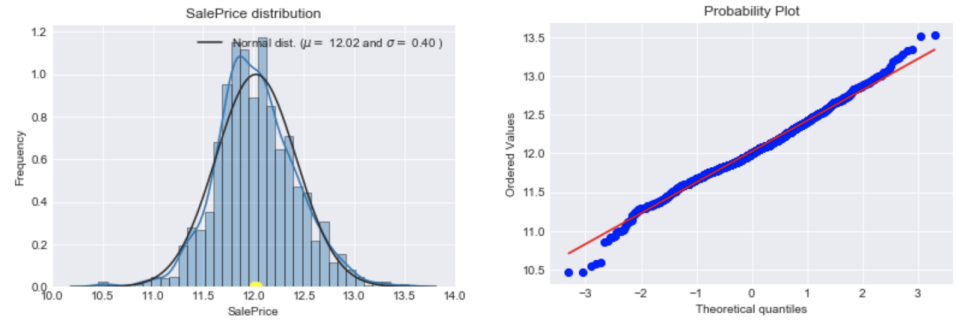
Our first question is what is the distribution of SalePrice and our null hypothesis is its a normal distribution. Normal distribution is important for the model fitting and can reduce the impact from outliers. We used library seaborn, which can plot an univariate distribution of observations. It can also fit distributions and plot the estimated probability density function over the data. We also drew a Q-Q plot, which is a plot of the quantiles of two distributions against each other, to compare the shapes of normal distribution and data distribution.



(i) Fit distribution of original data (ii) Q-Q plot of original data

Fig. 2. Visualization of original data

From Fig.2, we can see the distribution of SalePrice is not a perfect normal distribution, it looks more like a log normal distribution. So, we applied log function to the SalePrice of house and re-ploted.



(i) Fit distribution of log-transferred data (ii) Q-Q plot of log-transferred data

Fig. 3. Visualization of log-transferred data

Now We got a picture (Fig.3) looks like normal distribution with the mean of 12.02 and standard deviation of 0.40. We also used bootstrap to check the mean value. However, when we drew the Q-Q plot and calculated the p-value, its very small. So actually, we can reject the null hypothesis. But comparing with the p-value of original data, the log transformation does improve the p-value a lot.

3.2 Question 2: Relationships between features

Question 2 is which features will influence the SalePrice of house more and what is the relationship between each feature. Its common sense that the overall quality of house, original construction date, living area would be more important for people who want to live in the house. And features like Size of garage in car capacity and in square feet might have strong relationship. We drew a correlation matrix (Fig.4) of 37 numerical attributes. The lighter color means stronger correlation. We chose some correlations and try to explain them.

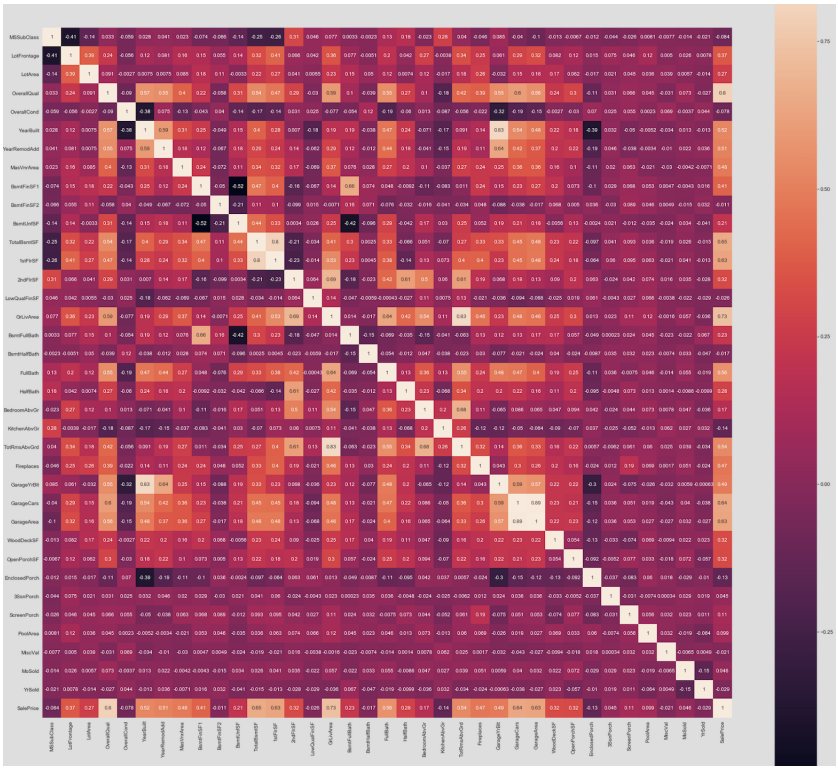
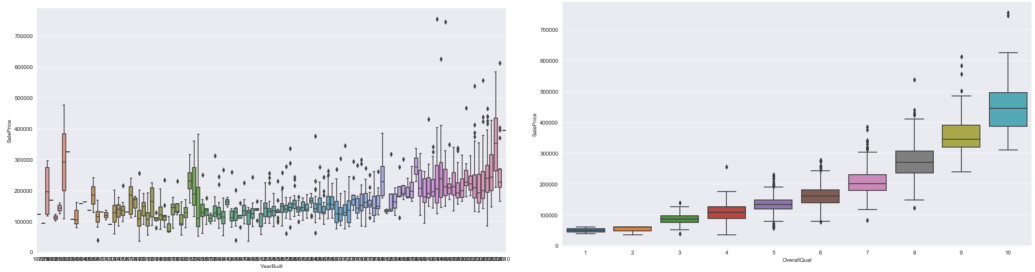


Fig. 4. Correlation matrix

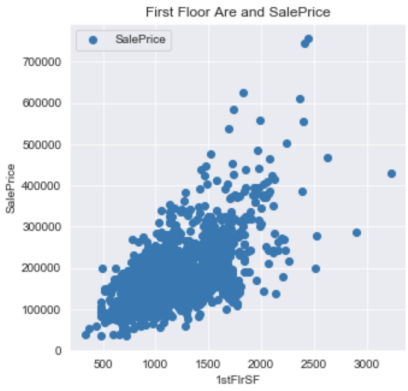


(i) YearBuilt vs. SalePrice (ii) OverallQual vs. SalePrice

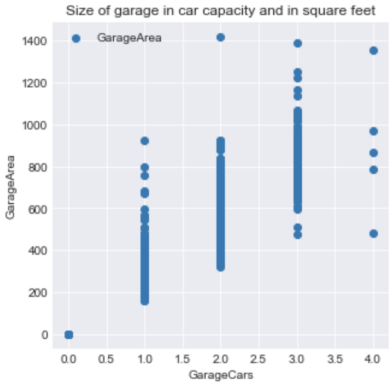
Fig. 5. Boxplots of relationships between features

The left plot shows the original construction date doesnt have strong relationship with the SalePrice as we predict. That might due to the Housing renovation. We can see the house built in 1900 has similar price distribution with house built in 2000. We guess the house is too old so it had some reconstructions to become a totally new house. And The right plot shows The relationship between the overall quality of house and the SalePrice is definitely strong.

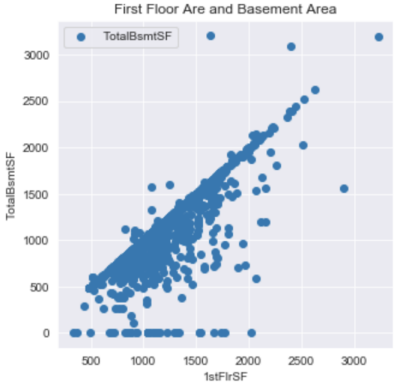
(i) LivingArea vs. SalePrice



(ii) 1stFlOOrArea vs. SalePrice



(iii) GarageArea vs. GarageCars



(iv) 1stFlOOrArea vs. BasementArea

Fig. 6. Scatter plots of relationships between features

And it is also easy to understand the positive correlation between living area, the area of first floor and the sale price. The correlation between the area of first floor and the sale price is not that strong is because the house might have other floors and basement. To better build the model, we need to do decorrelation.

4 MODEL CONSTRUCTION

Question 3: Predict the SalePrice using machine learning.

4.1 Feature Engineering

To improve the performance of machine learning algorithms, Feature Engineering is highly useful and beneficial. As we know, Feature Engineering is a data preparation process. One modifies the data such that Machine Learning algorithms have the ability to identify more patterns. This is done by combining and transforming existing features into new features.

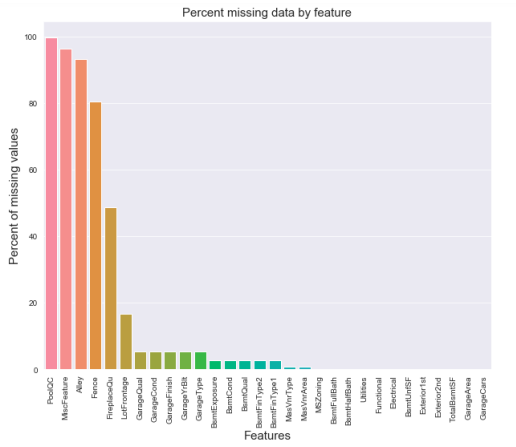
Before building a model, we need to do feature engineering at first. In this part, we mainly deal with missing data, categorical features and do principal component analysis.

4.1.1 Missing Data

From the following figures, we can know the percentage of missing values for each house attribute. It is obvious that the missing ratio of some features are especially higher than other attributes. For example, pool quality, type of alley access to property, fence quality, fireplace quality and other miscellaneous features not covered in other categories.

	Missing Ratio
PoolQC	99.691
MiscFeature	96.400
Alley	93.212
Fence	80.425
FireplaceQu	48.680
LotFrontage	16.661
GarageQual	5.451
GarageCond	5.451
GarageFinish	5.451
GarageYrBlt	5.451
GarageType	5.382
BsmtExposure	2.811
BsmtCond	2.811
BsmtQual	2.777
BsmtFinType2	2.743
BsmtFinType1	2.708
MasVnrType	0.823
MasVnrArea	0.788
MSZoning	0.137
BsmtFullBath	0.069

(i) Missing Data Percentage



(ii) Missing Ratio of each Feature

Fig. 7. Missing Data

After reading data description, we know the actual meaning that each feature represents and then fill in the missing area with four kinds of values. For features like pool quality, the data description says ”NA” means ”No Pool”. Considering the fact that the majority of houses have no Pool at all, the huge missing ratio up to 99% of pool quality makes sense. So we just fill in them with None. For features like the area of masonry veneer, missing value most likely means no masonry veneer for these houses. That means we can use ”0” to fill in them. For features like linear feet of street connected to property, since the area of each street connected to the house may be similar to that of other houses in its neighborhood, we can replace missing values using the median value of houses in its neighborhood. For features like Kitchen quality, there is only one missing value. Hence, it is acceptable that we just set it using the most frequent value TA meaning typical.

4.1.2 Categorical Features

Categorical data are common in many Data Science and Machine Learning problems. In general, it is more challenging to deal with them than numerical data. In particular, many machine learning algorithms require that their input is numerical and therefore categorical features must be transformed into numerical features before we can use any of these algorithms. One of the most common ways to make this transformation is to one-hot encode the categorical features, especially when there does not exist a natural ordering between the categories.

Based on this, when dealing with categorical features including ”MSSubClass”, ”BsmtFullBath”, ”BsmtHalfBath”, ”HalfBath” and so on, we just use LabelEncoder to map them into numerical values as the following figure shows.

```
# using labelencoder to map categorical values to numerical values
le = LabelEncoder()
for c in cols:
    new_col = le.fit_transform(all_data[[c]])
    all_data["o" + c] = new_col
```

Fig. 8. Categorical Features

4.1.3 Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure used to emphasize variation and bring out strong patterns in a dataset. It is often used to make data easy to explore and visualize. PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. PCA rotates the original data space into the directions of highest variance of the data. As we know, low variance can often be assumed to represent undesired background noise. The dimensionality of the data can therefore be reduced, without loss of relevant information, by extracting a lower dimensional component space covering the highest variance.

Therefore, we use PCA to simplify the complexity of our high-dimensional data without loss of some important information.

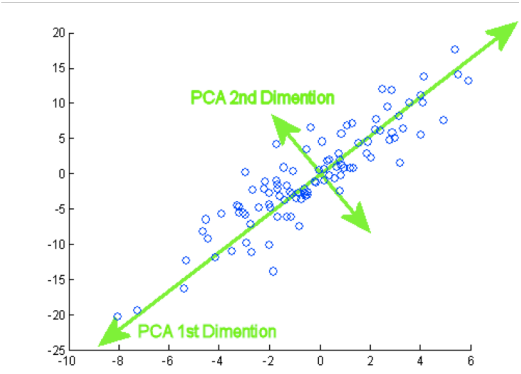


Fig. 9. Principal Component Analysis

4.2 Model Construction

To predict the house price which is a continuous output, we need to use machine learning algorithm like regression to build a set of models. Meanwhile, we use 5-folds cross-validation to evaluate these models and use root mean squared error (RMSE) to measure the accuracy of models.

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(predicted_t - label)^2}$$

Fig. 10. Root Mean Squared Error

4.2.1 Simple Models

After doing feature engineering, we built 9 models including linear regression, lasso regression, random forest regression, gradient boosting regression, support vector regression, elastic-net regression, Bayesian ridge regression and kernel ridge regression.

```
# models
models = {
          "LR": LinearRegression(),
          "Ridge" : Ridge(alpha=60),
          "Lasso": Lasso(alpha=0.0005,max_iter=10000),
          "RF" : RandomForestRegressor(),
          "GBR" : GradientBoostingRegressor(),
          "SVR": SVR(gamma= 0.0004, kernel='rbf', C=13, epsilon=0.009),
          "Ela": ElasticNet(alpha=0.005, l1_ratio=0.08, max_iter=10000),
          "Bay": BayesianRidge(),
          "Ker": KernelRidge(alpha=0.2, kernel='polynomial', degree=3, coef0=0.8)
         }
```

Fig. 11. Single Model Result

4.2.2 Average Weight Model

As illustrated in the following figure, the idea of Model averaging is to generate a meta-model (and meta-predictions) using a weighted average of the models. From the result of single models, we came up with a idea to design an weight averaged model using more than one regression method. That means for each data point, every single model has its own prediction. Then we multiply the prediction result of each model by its corresponding weight and add them together to get the final house price prediction result.

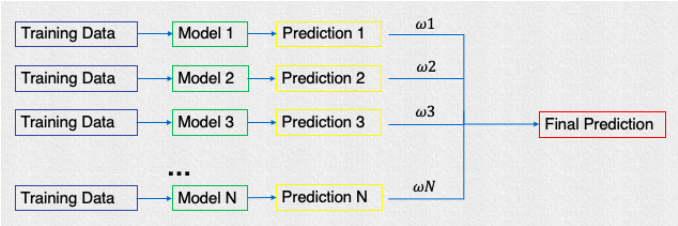


Fig. 12. Average Weight Model

4.2.3 Stacking Model

Stacking is a model ensembling technique used to combine information from multiple predictive models to generate a new model. In general, the stacked model will outperform each of the individual models because of its smoothing nature and ability to highlight each base model where it performs best and discredit each base model where it performs poorly. For this reason, stacking is most effective when the base models are significantly different. In brief, stacking is another kind of ensemble model, where a new model is trained from the combined predictions of two or more previous models. The predictions from the models are used as inputs for each sequential layer, and combined to form a new set of predictions which can achieve a better performance.

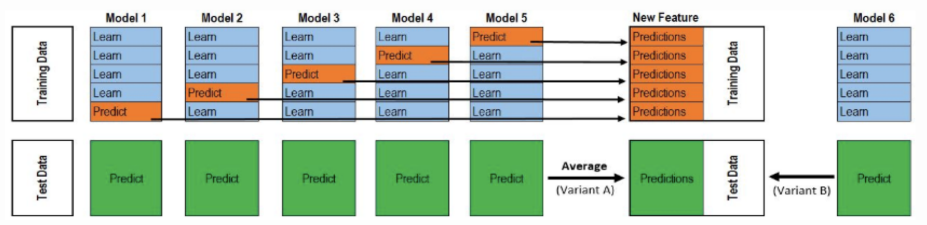


Fig. 13. Stacking Model

5 EXPERIMENT RESULTS

5.1 Simple Models

In this part, the 9 models we chose contains linear regression, lasso regression, random forest regression, gradient boosting regression, support vector regression, elastic-net regression, Bayesian ridge regression and kernel ridge regression.

After tuning parameters, from the following result, we can know that the root mean squared error of Support vector regression and kernel ridge regression is smaller than any other models.

```
for k, v in models.items():
    err = rmse_cv(v, train_X_scaled, y_log)
    print("{}: {:.4f}, {:.4f}".format(k, err.mean(), err.std()))
```

```
LR: 877120656.9807, 531453219.4491
Ridge: 0.1103, 0.0054
Lasso: 0.1117, 0.0064
RF: 0.1391, 0.0041
GBR: 0.1236, 0.0056
SVR: 0.1078, 0.0074
Ela: 0.1113, 0.0060
Bay: 0.1107, 0.0059
Ker: 0.1082, 0.0055
```

Fig. 14. Single Model Result

5.2 Average Weight Model

From the above result of single models, we came up with a idea to design an weight averaged model in the following figure using these two regression, support vector regression and kernel ridge regression. We multiply the prediction result of each model by its corresponding weight and add them together to get the final house price. This average-weight model achieves cross validation accuracy up to 89% which is better than any individual model.

```
print("The cross-validation score of average model kel and svr: {}".
      format(rmse_cv(avg_model, train_X_scaled, y_log).mean()))
```

```
The cross-validation score of average model kel and svr: 0.10655573186062914
```

Fig. 15. Average Weight Model Result

5.3 Stacking Model

To improve accuracy more, for the first layer, we build a stacking model including ridge regression, lasso regression, support vector regression, elastic-net regression, Bayesian ridge regression and kernel ridge regression. Then we extract the features generated from the first stacking layer and combine them with original features as the input of the second model, kernel ridge regression. As expected, the stacking model improves cross validation accuracy to 90%.

```
print("The cross-validation score of stack model after adding new features: {}".
      format(rmse_cv(stack_model, X_train_final, new_y).mean()))
```

```
The cross-validation score of stack model after adding new features: 0.10111936156382176
```

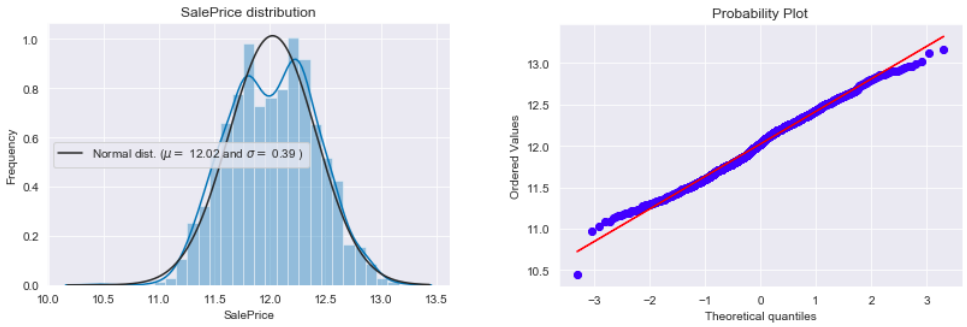
Fig. 16. Stacking Model Result

5.4 Test Result

Eventually, we use the stacking model to predict the sale price of house in test data. The result is as follows.

	Id	SalePrice
0	1461	235908.955
1	1462	169020.644
2	1463	137595.561
3	1464	147219.154
4	1465	125584.469
5	1466	143028.436
6	1467	151610.075
7	1468	140976.115
8	1469	140872.300
9	1470	225776.825

Fig. 17. Single Model Result



(i) Fit distribution of log-transferred test result (ii) Q-Q plot of log-transferred test result

Fig. 18. Visualization of log-transferred test result

After the submission of final prediction result, the root mean squared error is as follows:

Fig. 19. Final Test Result

6 CONCLUSION

In this project, we mainly completed two parts of job including data analysis and machine learning model construction.

For the the first part, we studied the distribution of house prices and the relationship between each attribute of house. We applied log function to the house SalePrice and used library seaborn to fit it to some distribution and plot the estimated probability density function. We also drew a Q-Q plot to compare the shapes of normal distribution and house price distribution. With hypothesis testing, it seems that sale price does not fit into normal distribution. In addition, we visualize the

data to explore the relationships between features and get some interesting results. For example, the overall quality of house, original construction date, living area would be some important factors which influence people's decisions a lot. Features like size of garage in car capacity and in square feet might have strong relationship with each other.

For the second part, we used 9 regression models to predict the SalePrice at first. We found that the root mean squared error of Support vector regression and kernel ridge regression is smaller than other models after tuning parameters. Based on this, we designed an weight averaged model using these two regression. We multiply the prediction result of each model by its corresponding weight and then add them together to get the final house price. This average-weight model achieves cross validation accuracy up to 89% which is better than any individual model. To improve accuracy more, we used stacking strategy where a new model is trained from the combined predictions of two or more previous models to train our models. As expected, the stacking model improves cross validation accuracy up to 90%. Eventually, we use this model to predict the sale price of house in test data which realized root mean squared error 9.45%.

References

[1] Smola AJ, Schlkopf B. A tutorial on support vector regression. Statistics and computing. 2004 Aug 1;14(3):199-222.

[2] An S, Liu W, Venkatesh S. Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression. Pattern Recognition. 2007 Aug 1;40(8):2154-62.

[3] Liaw A, Wiener M. Classification and regression by randomForest. R news. 2002 Dec 3;2(3):18-22.

[4] Elith J, Leathwick JR, Hastie T. A working guide to boosted regression trees. Journal of Animal Ecology. 2008 Jul;77(4):802-13.

[5] Ogutu JO, Schulz-Streeck T, Piepho HP. Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. InBMC proceedings 2012 Dec (Vol. 6, No. 2, p. S10). BioMed Central.

[6] Park T, Casella G. The bayesian lasso. Journal of the American Statistical Association. 2008 Jun 1;103(482):681-6.

[7] Wu D, Ngai G, Carpuat M. A stacked, voted, stacked model for named entity recognition. InProceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 2003 May 31 (pp. 200-203). Association for Computational Linguistics.

[8] https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard

[9] https://github.com/massquantity/Kaggle-HousePrices