Analyze the Factors Affecting House Prices

Jungang Bu, Jing Guo, Wei Wei, Xinyuan Yang
Group 10

Contents

Abstract

In this report, we will analyze the factors affecting house prices. To solve this problem, we are ready to study how 79 variables influence the house sale price by using different models, such as Linear Regression Based Model, Random Forest, Gradient Boosting Tree and Support Vector Machines (SVM). Besides, by using different error metrics, like Cross Validation Root Mean Square Logarithmic Error (CV RMSLE), Mean Absolute Percentage Error (MAPE), we can deduce which model is the best among these we have chosen.
Keywords: Linear Regression, Random Forest, Gradient Boosting Tree, SVM, Error Merics

1 Introduction

House is so important that most people will think over when buying (or leasing) their own one. When asking a home buyer to describe their dream house, they probably will not begin with the height of the basement ceiling or the proximity to an east-west railroad, instead, there will be a large amount of factors, such as Built Year, Heating, etc., for people to consider.

In this project, under the influence of 79 attributes, we will solve this regression problem based on the Linear Regression with principle component analysis (PCA), Random Forest, Gradient Boosting Tree and SVM. Then, after setting and adjusting various kinds of models, we will use CV RMSLE and MAPE, the error metrics, to judge which model is the most appropriate one among these we have chosen.

2 Data Processing and Feature Engineering

2.1 Data Set Selection

We originally have two data sets from Kaggle, a train set with the target value "**SalePrice**", and a test set without the target value, which is used for the competition host to grade on. All the following analysis use the train set since we need the true value of the target.

The location of the data set is
`https://www.kaggle.com/c/house-prices-advanced-regression-techniques`

2.2 Data Set Description

Initially, we have 79 attributes, in which the SalePrice is the response and others are predictors. We are going to describe every attribute including their dimensions and meanings in the appendix.

2.3 Data Process

For the data process, we primarily use the exploratory data analysis (EDA). EDA is an approach to analyzing data sets to summarize their main characteristics, often with visual methods.

First of all, we overview the whole data set, containing the train one and the test one from Kaggle. For some seemingly numerical attributes, actually they might be the categorical ones. For example, **MSSubClass**, **OverallCond**, which means the building class and the overall condition rating respectively, will be transformed into the categorical ones. Now we can get 34 numerical attributes and 45 categorical ones. For the next step we start dealing with the missing value.

2.3.1 Missing Value

By examining the whole data set, we observe that 34 attributes have missing values, 4 of these which have more than 50% ones. While focusing on the train data set, we just observe that 19 attributes have the missing values since the remaining attributes

have just one missing value in the test data set. The details will be shown in Figure 1.

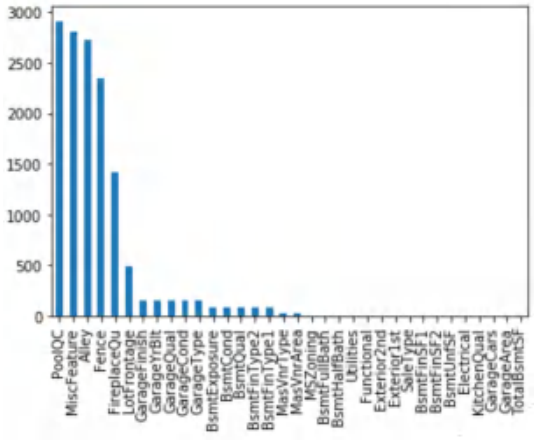


Figure 1: Missing value of 34 attributes

After matching these variables with their explanation, we found that most missing values "NA" for these attributes represent the meaning of lack of subject described by the corresponding attribute, like missing pool, fence, garage or basement. We change most of these categorical attributes into "None". What's more, for some special cases, we will take different changes depending on the their descriptions. For example, the missing values of **LotFrontage**, which means "Linear feet of street connected to property", are replaced by the median if the neighborhood. The missing values of **GarageYrBlt**, **GarageArea**, **GarageCars**, which means the "Year Garage built", "Size of Garage" and "Car Capacity of Garage" respectively, are replaced by 0 since these variables show that there is no garage at all.

2.3.2 Numerical Attributes

Since now there is no missing value in the data set, we can have a glance at the correlation matrix between the numerical attributes in Figure 2.

From the plot, we observe that there are many bright places which indicates the strong correlation between those corresponding attributes. We test the value of the correlated pairs **GarageCars** and **GarageArea** is equal to 0.882475. Though the value is high, it is still somehow acceptable since we usually only deal with those higher than 0.9 or 0.95. Thus we leave both of them remaining in the data set.

Then, by observing the distributions of all numerical attributes, we can have a further exploration and do the next process for the data set. From Figure 3, we could see that some independent variables look like good candidates for log transformation: **TotalBsmtSF**, **KitchenAbvGr**, **LotFrontage**, **LotArea**, etc. While gaining on regression transformation will smooth out some irregularities which could be important like large amount of houses with 0 **2ndFlrSF**. Such irregularities are good candidates for feature construction. We will actually do the log transformation based on the

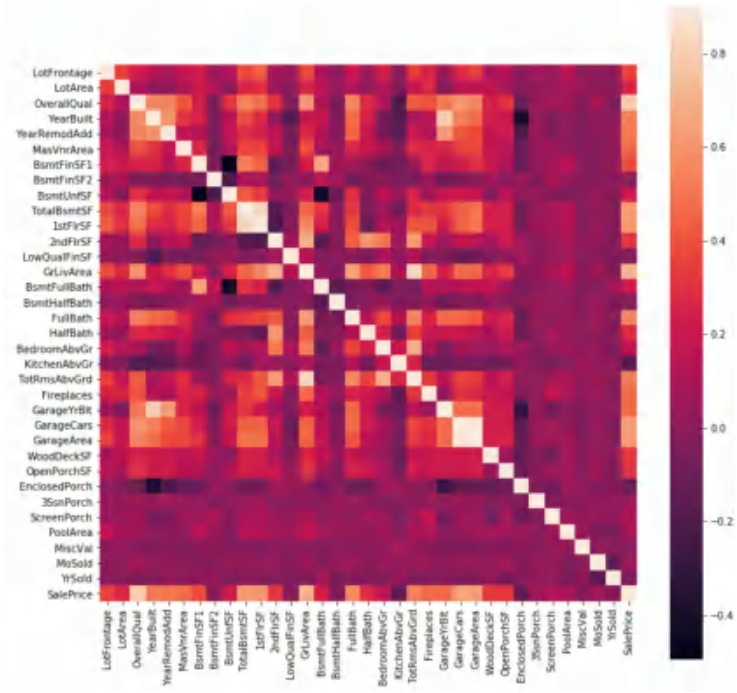


Figure 2: Correlation matrix of numerical attributes

skewness [1]. For those skewness > 1, which is up to 15 attributes, need to be performed the log transformation. The value of skewness for each attribute is shown in Figure 3.

After this transformation, all these numerical attributes should be taken into the consideration for the next step. Thus, we will ready to see the relationships between those variables and the predictor, **SalePrice**.

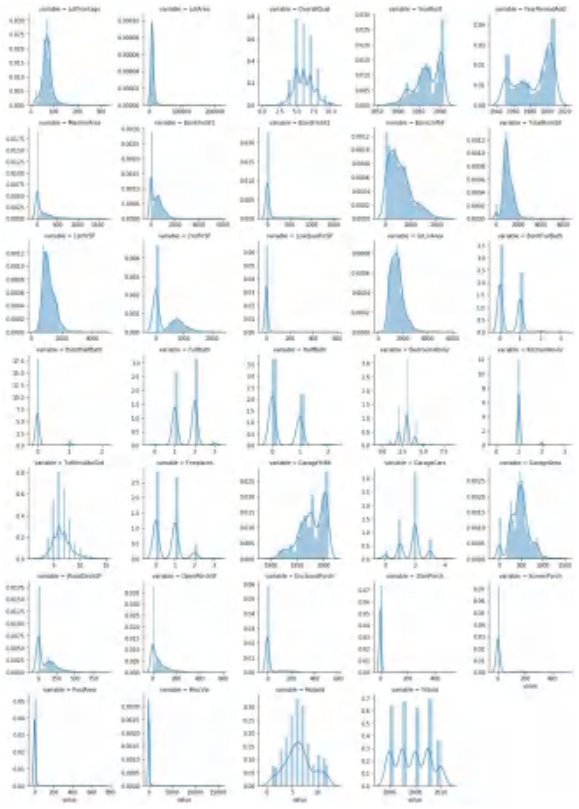
It can be found in Figure 4 that all these attributes have a relationship with the predictor more or less, which implies that when setting models, all of them should be included in the models. Also, there exist some outliers obviously. For instance, the points where $\mathbf{GrLivArea} > 4,000$ and $\mathbf{SalePrice} < 300,000$. However, outliers removal is not always safe, thus we decided to delete these two as they are very huge and really bad (extremely large areas for very low prices).

There are probably other outliers in the training data. However, removing all of them may affect badly on our models if ever there were also outliers in the test data. That's why, instead of removing all, we just manage to make some of our models robust on them. We also mention this in the modelling part.

The last step for processing the numerical attributes is to add one more feature which might be useful during the analysis in models. We introduce a new feature defined as

$$\mathbf{TotalSF} = \mathbf{TotalBsmtSF} + \mathbf{1stFlrSF} + \mathbf{2ndFlrSF}.$$

[1] In probability theory and statistics, skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean.



	Skew
MiscVal	24.434913
PoolArea	15.932532
LotArea	12.560986
3SsnPorch	10.286510
LowQualFinSF	8.995688
KitchenAbvGr	4.480268
BsmtFinSF2	4.247550
ScreenPorch	4.114690
BsmtHalfBath	4.095895
EnclosedPorch	3.083987
MasVnrArea	2.893554
OpenPorchSF	2.337421
LotFrontage	1.546174
WoodDeckSF	1.544214
GrLivArea	1.009951
BsmtUnfSF	0.919855
1stFlrSF	0.886723
2ndFlrSF	0.812121
BsmtFinSF1	0.764002
HalfBath	0.679351
TotRmsAbvGrd	0.659822
Fireplaces	0.631409
BsmtFullBath	0.589751
TotalBsmtSF	0.511177
MoSold	0.215211

Figure 3: Left: The Distribution of numerical attributes, Right: The Skewness of each numerical attribute

2.3.3 Categorical Attributes

Similar to the former procedure, we firstly examine all factors' relationships with the target variable **SalePrice**. Most categories seems to have impact on **SalePrice**. Some categorical variables may contain information in their ordering set, and we will use Label encoding [2] on them. For else, we will use One-hot encoding [3].

[2]Label Encoding refers to converting categorical labels in a data set used for machine learning purposes, into numeric form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for a structured data set in supervised learning.

[3]One-hot encoding: In digital circuits and machine learning, a one-hot is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0). In statistics, dummy variables represent a similar technique for representing categorical data.

Figure 4: Relationships between numerical attributes and SalePrice

2.3.4 Target Variable

Before setting the models, we examine the distribution pattern of the target variable, **SalePrice**.

We can know from the plots that the predictor, **SalePrice** is apparently not normally distributed. Thus we will do a log transformation to make it normally distributed so that we can fit the data with our models more appropriately.

From Figure 7, we can observe that **SalePrice** follows the normal distribution after the logarithm transformation, which might be helpful during setting the models.

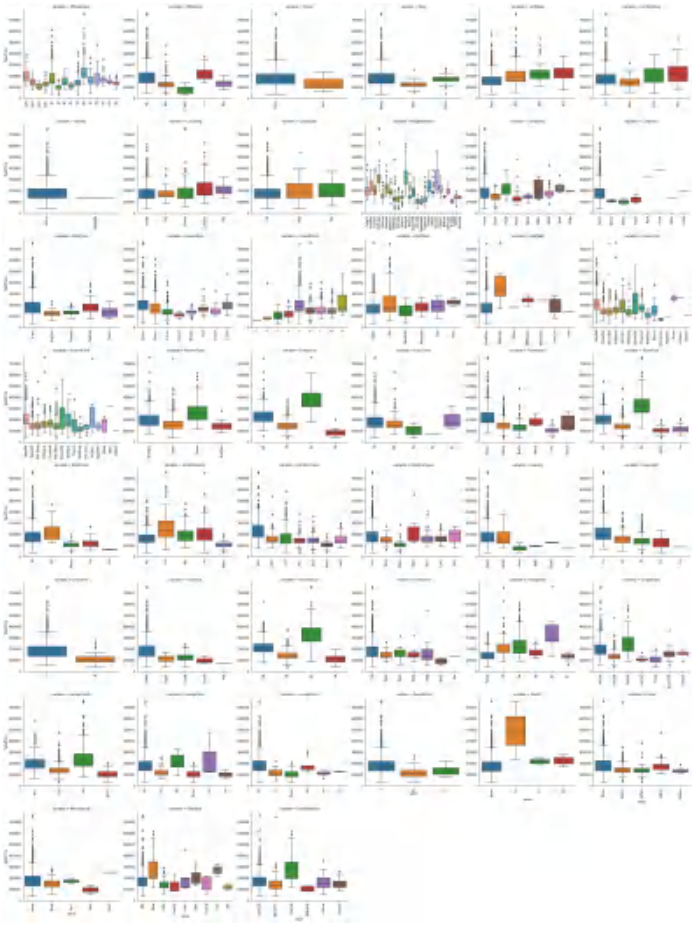


Figure 5: Relationship between categorical attributes and SalePrice

3 Models

3.1 Overview

In this part, we are going to try several different methods and compare their performances. As we need a data set to help us determine the goodness of a model, we further split the train set that Kaggle provides us into a 70% train set, and a 30% percent validation set. We use the splitted train set to fit the model, and calculate the error metrics on the validation set. For some of the models that involve with searching a optimal hyper-parameter, we use 5 fold CV and grid search. In the end, we use the whole train set plus validation set to train the final model, and submit the prediction on the test set on Kaggle to see the ranking.

The error metric we mainly use is CV RMSLE on the train set. Other interesting metrics are all calculated on the validation set, including RMSE, RMSLE, MAE, MAPE.

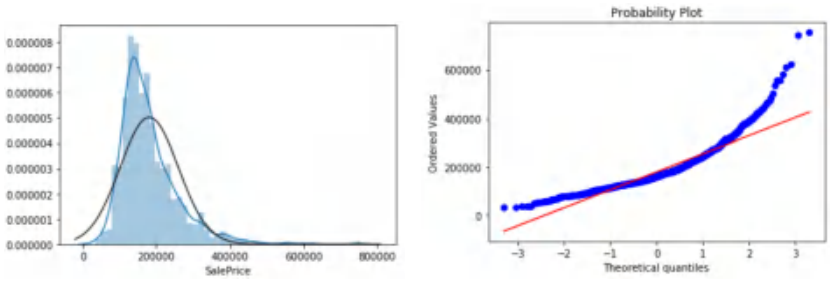


Figure 6: Left: The distribution of SalePrice, Right: The Quantile-Quantile Plot of SalePrice

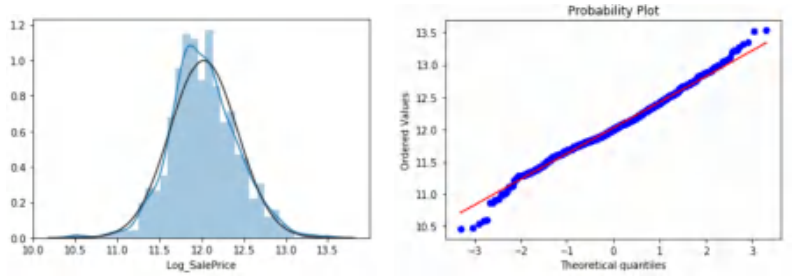


Figure 7: Left: The distribution of logarithm of SalePrice, Right: The Quantile-Quantile Plot of logarithm of SalePrice

3.2 Linear Regression Based Models

3.2.1 Linear Regression

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. In this case, we worked multiple linear regression to predict house price. The multiple linear regression model takes the form $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$. And here is our coefficient matrix and we used RMSE as metrics to estimate the model.

Also, because linear regression based models are sensitive to the extreme values, we use a standardization method called robust scaling. Similar to MinMax scaling which scales the data into $[0, 1]$ between minimum and maximum, robust scaling uses 25 percentile and 75 percentile instead of the minimum and maximum. This method is more robust when there are extreme values in the dataset, while MinMax scaling will scale most numbers very close to 0 or 1 when there is a outlier.

However in this model, we got CV score about 1302822962.7540, which is a pretty huge number. As shown in the coefficient matrix, there are some high coefficients, which might be the reasons for causing this problem. There might be some extreme numbers in our data set which can cause outliers the linear regression sensitive with. But when we trained the following model, this situation does not appear. So we will

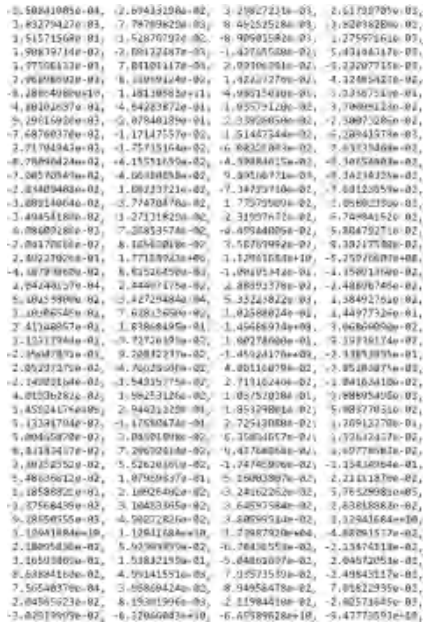


Figure 8: Coefficient Matrix

not drop these huge numbers in data set directly which may cause the model overfitted to the data that is not the extreme value.

3.2.2 Lasso Regression

Lasso (least absolute shrinkage and selection operator; also LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. Lasso was introduced in order to improve the prediction accuracy and interpretability of regression models by altering the model fitting process to select only a subset of the provided covariates for use in the final model rather than using all of them. Similarly, we are using robust scaling in Lasso regression too. Here is the result of Lasso regression.

```
Lasso CV score: 0.1136 (0.0124)
Lasso train rmse score: 17385.5909
Lasso val rmse score: 20621.7483
Lasso train rmsle score: 0.0951
Lasso val rmsle score: 0.1137
Lasso train mae score: 11785.0415
Lasso val mae score: 13800.9500
Lasso train mape score: 6.82%
Lasso val mape score: 7.95%
```

Figure 9: Lasso Regression

As shown in the Lasso Regression, the CV score is 0.1136 which is smaller than

the Linear Regression.

3.2.3 Ridge Regression

Ridge Regression is particularly useful to mitigate the problem of multicollinearity in Linear regression, which commonly occurs in models with large numbers of parameters. In general, the method provides improved efficiency in parameter estimation problems in exchange for a tolerable amount of bias. Also, we are using robust scaling here.

Here is the result of Ridge Regression:

```
Ridge CV score: 0.1154 (0.0110)
Ridge train rmse score: 17574.3133
Ridge val rmse score: 20749.2541
Ridge train rmsle score: 0.0982
Ridge val rmsle score: 0.1125
Ridge train mae score: 11987.7283
Ridge val mae score: 13834.7838
Ridge train mape score: 6.96%
Ridge val mape score: 7.94%
```

Figure 10: Ridge Regression

As Figure 10 shows, the CV score for Ridge Regression is 0.1154, which is slightly bigger than the value of Lasso regression method.

3.2.4 Elastic Net

The Elastic Net is a regularized regression method that linearly combines the L_1 and L_2 penalties of the Lasso and Ridge methods. The Elastic Net method overcomes the limitations of the Lasso. The Elastic Net adds a quadratic part to the penalty ($\|\beta\|^2$), which when used alone is Ridge regression. The estimates from the Elastic Net method are defined by $\hat{\beta} \equiv \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1)$. The quadratic penalty term makes the loss function strongly convex, and it therefore has a unique minimum. The Elastic Net method includes the Lasso and Ridge regression: in other words, each of them is a special case where $\lambda_1 = \lambda$,$\lambda_2 = 0$ or $\lambda_1 = 0$,$\lambda_2 = \lambda$. Here is the result of Elastic Net:

```
alpha:  0.0002496826788042955 3
l1_ratio:  1.0
```

Figure 11: ElasticNet Regression

After doing the parameter searching and we got the L_1 ratio is 1, which means that Elastic Net method is completely the same as the Lasso method.

3.3 Random Forest

Random Forest is an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean

prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Here is the result of Random Forest Method:

```
Random Forest CV score: 0.1444 (0.0128)
Random Forest train rmse score: 11601.6103
Random Forest val rmse score: 30108.4749
Random Forest train rmsle score: 0.0528
Random Forest val rmsle score: 0.1309
Random Forest train mae score: 6381.5916
Random Forest val mae score: 17029.7876
Random Forest train mape score: 3.47%
Random Forest val mape score: 9.03%
```

Figure 12: Random Forest

Figure 12 shows that the CV score for Random Forest is 0.1444, which is obviously worse than Lasso Regression and Ridge Regression. Furthermore, it is obvious that Random Forest performs better in training set than validation set, which may indicate that there exist some overfitting problems. However, we can not work with parameter searching in this method. It might be the problem of the model itself.

3.4 Gradient Boost Regression

Gradient Boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. Here is the result of Gradient Boost Regression:

```
Gradient Boosting Regression CV score: 0.1186 (0.0123)
Gradient Boosting Regression train rmse score: 14566.8813
Gradient Boosting Regression val rmse score: 23332.9635
Gradient Boosting Regression train rmsle score: 0.0836
Gradient Boosting Regression val rmsle score: 0.1105
Gradient Boosting Regression train mae score: 9829.2624
Gradient Boosting Regression val mae score: 14148.8615
Gradient Boosting Regression train mape score: 5.80%
Gradient Boosting Regression val mape score: 7.75%
```

Figure 13: Gradient Boost Regression

In Figure 13, the CV score is 0.1186 after working parameter searching on it, which means that even it is slightly worse than Ridge Regression and Lasso Regression, it is better than than the Random Forest.

3.5 Support Vector Machines

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. After trying parameter searching, we apply Support Vector Machines Approach into the house price data sets and the result is shown in the Figure 14.

```
SVR CV score: 0.1146 (0.0130)
SVR train rmse score: 17002.1481
SVR val rmse score: 20314.0812
SVR train rmsle score: 0.1000
SVR val rmsle score: 0.1061
SVR train mae score: 11144.2195
SVR val mae score: 13313.5640
SVR train mape score: 6.41%
SVR val mape score: 7.34%
```

Figure 14: Support Vector Machines

The SVR CV Score is 0.1146, which means that it is slightly better than the Ridge Regression but worse than the Lasso Regression.

3.6 Averaging Models

When we have all these models, simply taking the average of them may just yield better results. We tried it and here is the result. Note that because the random forest model is significantly worse than the others, we are including all other methods in this averaging model.

```
Averaged models CV score: 0.1119 (0.0119)
Averaged models train rmse score: 16103.8001
Averaged models val rmse score: 20402.5227
Averaged models train rmsle score: 0.0916
Averaged models val rmsle score: 0.1075
Averaged models train mae score: 10764.0760
Averaged models val mae score: 13339.9694
Averaged models train mape score: 6.26%
Averaged models val mape score: 7.50%
```

Figure 15: Averaging model

It indeed increases the performance by CV score of 0.1119.

4 Submission and Conclusion

Now we have the averaging model as our best model, and we are ready to submit. We re-train the model on the whole train set and validation set, and do the prediction on the test set. Up to when we submit the prediction, we got a ranking of 542/4645 (top 11.6%). Not bad! Our model is better than those 88% percent of participants! Note that this is a relatively old competition, and on the leaderboard, there are some submissions that have even 0 RMSLE, meaning they fit the prediction perfectly. They are definitely overfitting the public test set, so our actual rank may be a bit higher. Our methods that used some data processing techniques, feature engineering, tried a various of methods and used the technique of averaging models finally reaches a decent ranking in this Kaggle competition.

A Data Description

The attributes with their dimensions and meanings.

Name	Desctription
SalePrice	the property's sale price in dollars. This is the target variable that we are trying to predict.
MSSubClass	The building class
MSZoning	The general zoning classification
LotFrontage	Linear feet of street connected to property
LotArea	Lot size in square feet
Street	Type of road access
Alley	Type of alley access
LotShape	General shape of property
LandContour	Flatness of the property
Utilities	Type of utilities available
LotConfig	Lot configuration
LandSlope	Slope of property
Neighborhood	Physical locations within Ames city limits
Condition1	Proximity to main road or railroad
Condition2	Proximity to main road or railroad (if a second is present)
BldgType	Type of dwelling
HouseStyle	Style of dwelling
OverallQual	Overall material and finish quality
OverallCond	Overall condition rating
YearBuilt	Original construction date
YearRemodAdd	Remodel date
RoofStyle	Type of roof
RoofMatl	Roof material
Exterior1st	Exterior covering on house
Exterior2nd	Exterior covering on house (if more than one material)
MasVnrType	Masonry veneer type
MasVnrArea	Masonry veneer area in square feet
ExterQual	Exterior material quality
ExterCond	Present condition of the material on the exterior
Foundation	Type of foundation
BsmtQual	Height of the basement
BsmtCond	General condition of the basement
BsmtExposure	Walkout or garden level basement walls
BsmtFinType1	Quality of basement finished area
BsmtFinSF1	Type 1 finished square feet
BsmtFinType2	Quality of second finished area (if present)
BsmtFinSF2	Type 2 finished square feet
BsmtUnfSF	Unfinished square feet of basement area
TotalBsmtSF	Total square feet of basement area
Heating	Type of heating
HeatingQC	Heating quality and condition
CentralAir	Central air conditioning
Electrical	Electrical system
1stFlrSF	First Floor square feet
2ndFlrSF	Second floor square feet
LowQualFinSF	Low quality finished square feet (all floors)
GrLivArea	Above grade (ground) living area square feet
BsmtFullBath	Basement full bathrooms
BsmtHalfBath	Basement half bathrooms
FullBath	Full bathrooms above grade
HalfBath	Half baths above grade

Bedroom	Number of bedrooms above basement level
Kitchen	Number of kitchens
KitchenQual	Kitchen quality
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
Functional	Home functionality rating
Fireplaces	Number of fireplaces
FireplaceQu	Fireplace quality
GarageType	Garage location
GarageYrBlt	Year garage was built
GarageFinish	Interior finish of the garage
GarageCars	Size of garage in car capacity
GarageArea	Size of garage in square feet
GarageQual	Garage quality
GarageCond	Garage condition
PavedDrive	Paved driveway
WoodDeckSF	Wood deck area in square feet
OpenPorchSF	Open porch area in square feet
EnclosedPorch	Enclosed porch area in square feet
3SsnPorch	Three season porch area in square feet
ScreenPorch	Screen porch area in square feet
PoolArea	Pool area in square feet
PoolQC	Pool quality
Fence	Fence quality
MiscFeature	Miscellaneous feature not covered in other categories
MiscVal	Value of miscellaneous feature
MoSold	Month Sold
YrSold	Year Sold
SaleType	Type of sale
SaleCondition	Condition of sale

References

[1] https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard/notebook#Modelling

[2] https://www.kaggle.com/lavanyashukla01/how-i-made-top-0-3-on-a-kaggle-competition#EDA

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

[4] Gareth, James. An introduction to statistical learning: with applications in R. Springer Verlag, 2010.

[5] https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python