House Price Prediction with Regression Methods

Lucen Zhao
MATH4432 Monthly-Project 2
Student ID: 20256435
lzhaoaj@connect.ust.hk

Abstract

In this project, I applied regression models we learned in this course to predict the house price and evaluated the results. Among all the models, GAM, boosting and LASSO achieved the best results. The results are evaluated with R^2 and rooted MSE.

1 Introduction

House price prediction is a interesting topic that can be explored with regression methods. In this project, I applied regression models on a dataset with over 1400 entries and over 80 distinct features to predict the housing price. The methods I used include polynomial Ridge regression, polynomial LASSO regression, k-nearest neighbour regression, principle component regression (PCR), generalized additive model (GAM), random forest and boosting. The features are selected and modified with principle component analysis, recurrent feature elimination and forward stepwise selection with f-regression. The results are submitted in the Kaggle contest `https://www.kaggle.com/c/house-prices-advanced-regression-techniques/`.

2 Data Preprocessing

Because there are some missing contents in the dataset, and there are some categorical values that cannot be directly used for training with scikit-learn package, first the data need to be pre-processed. Initially, there are 1460 entries and 81 columns (80 features + 1 target). The composition of features is shown in Table 1:

Table 1: Types of features

Feature Type	Number
Object (categorical)	43
Discrete Value (integer)	34
Continuous Value (float)	3
Total	80

For some columns, most of the values are unknown (NaN). In these cases, the columns cannot be very helpful since the feature does not exist for a large portion of data. Hence, I directly dropped the columns with over 1000 NaN values. For those columns with less NaN, if they are categorical values, these are substituted with "Others", and will be regarded as a new category. If they are integer or float values, they will be substituted with the median value of that column. After this preprocess, there are 76 columns (75 features) left. Moreover, we used the log value of House Price for training the model.

Then the categorical values are encoded with integers to represent different categories, and the integer and float values are normalized as values with mean 0 and variance 1. The scatter point plots between

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

some features (before normalization) and the log-value targeting HousePrice images are shown in Figure 1.

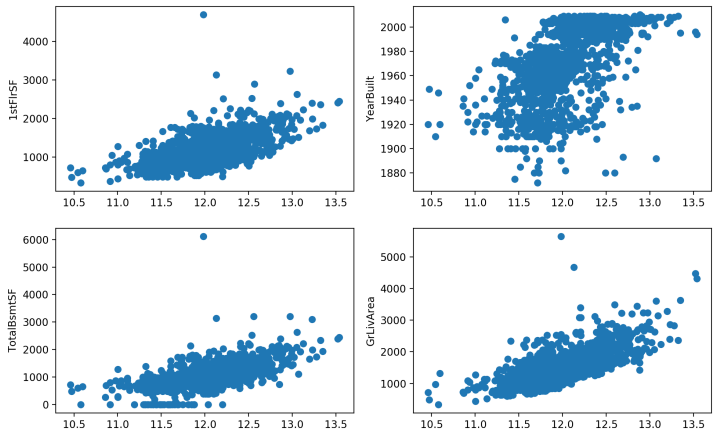

Figure 1: Samples from MNIST database

3 Methodology

3.1 Feature Engineering

Except the normalization, encoding and log-transform mentioned above, several other feature engineering methods are applied, including feature selection and feature construction.

3.1.1 Feature Selection

I tried several approaches to select the best subset of features to be used in training models.

Forward Stepwise Selection I applied a forward stepwise feature selection which is different from the one we learned in lectures. Instead of using R^2 as the criteria, here the F-statistics and the mutual information (based on k-nearest neighbours) are respectively used to judge which are the best features to select. The resulting features are greedily selected in each step. Both approaches are tried in feature selection.

Recurrent Feature Elimination The recurrent feature elimination method is similar to backward stepwise selection. In each step, the importance of features are calculated, and the one with smallest importance will be eliminated, until the expected number of feature is reached, or the best subset is selected (decided with cross-validation).

Principle Component Analysis PCA is also used for feature dimensionality reduction.

3.1.2 Feature Construction

Polynomials of Predictors To apply polynomial regression models, the degree-2 and degree-3 polynomial terms of predictors with non-categorical values are calculated. These polynomial terms are only applied to Ridge and LASSO regression models. For GAM model, no manually created polynomial terms are needed, and for tree-structured methods, there is no point to use these terms.

Feature Combination The correlation between features are measured first. For these features with strong correlations are multiplied together to create new features. These features are applied to all models we used.

3.1.3 Relationship between Feature Selection and Construction

There are 2 choices in the order of using these 2 methods: (a) construct features before selection; (b) select some features to construct new ones, then select again.

In this project, for the construction of polynomial terms, I tried both methods, however, considering the subset of features finally selected, there is no big difference between these 2 orders.

3.2 Regression Models

Polynomial Regression Models The linear regression model without polynomial terms is regarded as the baseline. Both Ridge and LASSO are applied with polynomial terms. The parameter of these 2 models are selected with cross-validation.

K-Nearest Neighbour Regression Instead of using the classification model of KNN, here the KNN regressor is used.

Principle Component Regression Principle component regression (PCR) is actually combined with other regression methods. After principle component analysis, the selected components are used as predictors of different regression models.

Generalized Additive Model The GAM model is also used. Because of its powerfulness, as the number of splines in the GAM model is large enough, there is no need to use other kinds of spline models.

Random Forest With the random forest (RF) model, there is no need to use a single regression tree model. Several different number of trees and number of features used are tested with RF model, and the optimal one is decided with cross-validation.

Gradient Boosting The number of splits is fixed as 1, and 0.01, 0.001 are tried as the shrinkage parameter. The number of trees in boosting model is decided with cross-validation.

3.3 Validation and Evaluation

To select best parameters and best set of features, 3-fold cross-validation is used.

Bootstrapping is used to evaluate the model.

4 Experiment

In the experiment, firstly, the set of features to be used is decided for each model. Then, best tuning parameters are selected.

4.1 Feature Selection

For each model, we use a default setting of tuning parameters to select its best set of features. Firstly, we construct polynomial terms and combination of features, then we select them using our methods of selection. The candidates for best subset of features include:

- Original Features (Baseline)
- Construction (Combination and degree-2, degree-3 polynomials)
- Construction + F-regression forward stepwise selection (80%)
- Construction + mutual information forward stepwise selection (80%)
- Construction + PCA (80%)
- Construction + recurrent feature elimination (cross-validated)

The results of experiment are listed below. The score here refers to the square-rooted mean-square error between log-transformed values. Due to the limitation of computational power, the recurrent feature elimination is only performed on Ridge and LASSO regression models. Moreover, the construction for GAM model only include combination of 2 features. The polynomial of a single feature are not included.

Table 2: Feature Selection via CV

Model	Features	Score
Ridge ($\alpha = 0.0001$)	Original (Baseline)	0.1526
Ridge ($\alpha = 0.0001$)	With constructed features	0.3268
Ridge ($\alpha = 0.0001$)	Construction + F-regression	0.1636
Ridge ($\alpha = 0.0001$)	Construction + mutual information	0.1655
Ridge ($\alpha = 1$)	Construction + PCA	0.7661
Ridge ($\alpha = 0.001$)	Construction + recurrent feature elimination	0.1399
LASSO ($\alpha = 0.00001$)	Original (Baseline)	0.1527
LASSO ($\alpha = 0.00001$)	With constructed features	0.2559
LASSO ($\alpha = 0.00001$)	Construction + F-regression	0.1589
LASSO ($\alpha = 0.00001$)	Construction + mutual information	0.1526
LASSO ($\alpha = 0.00001$)	Construction + PCA	0.4442
LASSO ($\alpha = 0.00001$)	Construction + recurrent feature elimination	0.1321
K-Nearest Neighbour (k=5)	Original (Baseline)	0.2417
K-Nearest Neighbour (k=5)	With constructed features	0.2889
K-Nearest Neighbour (k=5)	Construction + F-regression	0.1853
K-Nearest Neighbour (k=5)	Construction + mutual information	0.1839
K-Nearest Neighbour (k=5)	Construction + PCA	0.2890
Random Forest (ntrees=500, max_depth=4)	Original (Baseline)	0.1752
Random Forest (ntrees=500, max_depth=4)	With constructed features	0.1639
Random Forest (ntrees=500, max_depth=4)	Construction + F-regression	0.1675
Random Forest (ntrees=500, max_depth=4)	Construction + mutual information	0.1667
Random Forest (ntrees=500, max_depth=4)	Construction + PCA	0.2511
Random Forest (ntrees=500, max_depth=4)	Construction + recurrent feature elimination	0.2511
Gradient Boosting (ntrees=500, shrinkage=0.01)	Original (Baseline)	0.1807
Gradient Boosting (ntrees=500, shrinkage=0.01)	With constructed features	0.1725
Gradient Boosting (ntrees=500, shrinkage=0.01)	Construction + F-regression	0.1742
Gradient Boosting (ntrees=500, shrinkage=0.01)	Construction + mutual information	0.1734
Gradient Boosting (ntrees=500, shrinkage=0.01)	Construction + PCA	0.2702
Gradient Boosting (ntrees=500, shrinkage=0.01)	Construction + recurrent feature elimination	0.2702
Generalized Additive Model (splines=10)	Original (Baseline)	0.1120
Generalized Additive Model (splines=10)	With constructed features	0.1184
Generalized Additive Model (splines=10)	Construction + F-regression	0.1263
Generalized Additive Model (splines=10)	Construction + mutual information	0.1215
Generalized Additive Model (splines=10)	Construction +PCA	0.1088

From the table above, we can see that for most models, the results using forward stepwise selection with mutual information regression has achieved a good score for KNN, RF and Boosting. Hence it is chosen as my feature for these models. For Ridge and LASSO, I use recurrent elimination for feature selection. For GAM model, surprisingly, the score for PCA is the best, and the one for original features is also high. This is possibly because the GAM model is very flexible. If we add too many additional dimensions of features, overfitting is likely to happen here. Hence, PCA features are chosen for GAM model.

4.2 Model Selection

For different models, we need to fix different tuning parameters.

Ridge and LASSO Regression The α value of both regression models.

K-Nearest Neighbour k, the number of neighbours to be considered.

Random Forest The number of estimators (trees), and number of features (square-root, log or no limitation).

Gradient Boosting The number of estimators (trees), and shrinkage (only consider 0.01 and 0.001).

Generalized Additive Model The number of splines.

The experimental results of parameter selection are listed below:

Table 3: Model Selection via CV

Model	Score
Ridge regression ($\alpha = 0.001$)	0.1336
Ridge regression ($\alpha = 0.01$)	0.1329
Ridge regression ($\alpha = 0.1$)	0.1324
Ridge regression ($\alpha = 1$)	0.1306
Ridge regression ($\alpha = 3$)	0.1305
LASSO regression ($\alpha = 0.0001$)	0.1288
LASSO regression ($\alpha = 0.001$)	0.1284
LASSO regression ($\alpha = 0.01$)	0.1702
LASSO regression ($\alpha = 0.05$)	0.2070
LASSO regression ($\alpha = 0.1$)	0.2575
K-nearest neighbour (k=3)	0.1907
K-nearest neighbour (k=5)	0.1829
K-nearest neighbour (k=10)	0.1831
Random forest (ntree=500, m=p)	0.1643
Random forest (ntree=500, m=$\sqrt{p}$)	0.1659
Random forest (ntree=500, m=$\log p$)	0.1670
Random forest (ntree=1000, m=p)	0.1647
Random forest (ntree=1000, m=$\sqrt{p}$)	0.1656
Random forest (ntree=1000, m=$\log p$)	0.1671
Gradient boosting (ntree=500, $\lambda = 0.01$)	0.1734
Gradient boosting (ntree=2000, $\lambda = 0.01$)	0.1376
Gradient boosting (ntree=5000, $\lambda = 0.01$)	0.1325
Gradient boosting (ntree=2000, $\lambda = 0.001$)	0.2292
Gradient boosting (ntree=5000, $\lambda = 0.001$)	0.1736
Generalized Additive Model (nspline=5)	0.1103
Generalized Additive Model (nspline=8)	0.1086
Generalized Additive Model (nspline=10)	0.1088

According to the table above, it can be seen that LASSO model, Boosting model and GAM model have achieved good results. Hence, finally, 3 models are selected.

- LASSO regression model ($\alpha = 0.001$) with RFE features.
- Gradient boosting model (ntree=5000, $\lambda = 0.01$) with mutual information selected features.
- Generalized Additive Model (nspline=8) with PCA selected features.

4.3 Testing Results

The testing results of the three models are obtained by submitting the test scores to the Kaggle contest.

The models and the results will be analyzed in detail in the following section.

Table 4: Final results

Model	Feature	Score
Generalized Additive Model (nspline=8)	PCA	0.13097
Gradient boosting (ntree=5000, $\lambda = 0.01$)	mutual information	0.12990
LASSO regression ($\alpha = 0.001$)	RFE	0.13448

5 Evaluation

In this section, I will mainly evaluate the three models which achieved the best performance, namely, LASSO, GAM and boosting models. For other models, I will evaluate possible reasons that they did not achieve a good performance.

5.1 LASSO Regression

5.1.1 Feature

For LASSO regression, there are only 31 dimension of features selected by recurrent feature elimination. Compared with the 236 features obtained from original feature + combinations + polynomials, this is really a small value. The features used are listed below:

'MSZoning', 'LotArea', 'LotShape', 'Condition2', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtExposure', 'BsmtFinSF1', 'HeatingQC', 'CentralAir', 'KitchenQual', 'Functional', 'Fireplaces', 'GarageFinish', 'PavedDrive', 'SaleCondition', '1stFlrSF*OverallQual', '1stFlrSF*YearBuilt', 'GarageArea*OverallQual', 'GarageCars*YearBuilt', 'GrLivArea*OverallQual', 'GrLivArea*YearBuilt', 'YearBuilt*TotalBsmtSF', 'OverallQual$\hat{2}$', 'OverallCond$\hat{2}$', '2ndFlrSF$\hat{2}$', 'FullBath$\hat{2}$', 'KitchenAbvGr$\hat{2}$', '1stFlrSF$\hat{3}$', 'GrLivArea$\hat{3}$'

Despite the original features, there are some polynomials and combinations chosen for this model. Features such as 1stFlrSF, OverallQual, and YearBuilt appeared multiple times in this set, suggesting that these could be some of the key features for our model.

5.1.2 Result

Using 100 data points for validation, we obtained the evaluation scores for the model.

The R^2 score of the LASSO model is 0.8772, meaning that almost 90% of data can be explained by this model. Hence, the result is quite satisfying.

The explained variance score is the same 0.8772, meaning that more than 87% of the variance are explainable.

5.2 GAM Model

The features of GAM model are selected with PCA, so unlike the LASSO model, the name of features cannot be shown directly. Hence, here we selected 200 data points for validation and analysis.

5.2.1 Feature

The features of GAM model are selected with PCA, so unlike the LASSO model, the name of features cannot be shown directly. Hence, here I focus on the relationship between the number of components from PCA and the resulted validation error. Noted that the dimension of feature is 164 before PCA analysis, including original features and combination of features. The validation result is shown in Figure 2.

In the plot above, I chose the component number from 10 to 90. It is clear that with 80 components, the optimal error rate is achieved.

5.2.2 Result

The R^2 score of the GAM model is 0.9237, meaning that more than 90% of data can be explained by this model. This result is even better than the result achieved from LASSO. However, because there are only 200 points for validation, the scores could have some variations from the actual scores.

The explained variance score is the same 0.9243, meaning that more than 90% of the variance are explainable.

The resulted points are shown in Figure 2.

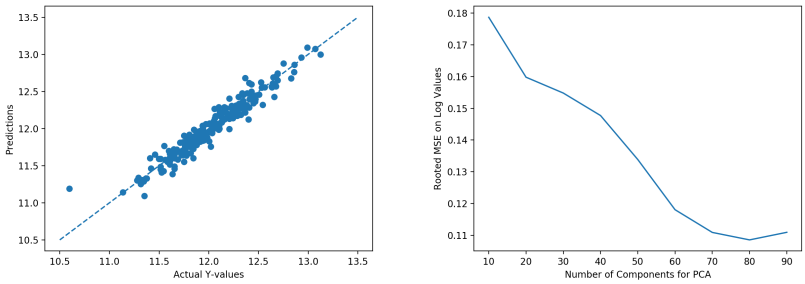

Figure 2: GAM Validation Results

5.3 Tree Models

In this section, I will evaluate random forest and gradient boosting model.

5.3.1 Feature

Before the mutual information backward stepwise selection, there are 236 features in all. The figure showing the relationship between number of feature ($236 \times percentage$) and the validation error is shown in Figure 3. Surprisingly, the MSE decrease sharply as the percentage of feature increases. It tells us that our number of feature for that model might not be enough for the best prediction results. That is to say, we can construct more features from the existing ones, such as try other kinds of combination, to improve the result of the model.

5.3.2 Results

The results with the best model (90% features selected) is shown below.

The R^2 score of the Boosting model is 0.8948, meaning that almost 90% of data can be explained by this model.

The explained variance score is the same 0.8948, meaning that almost 90% of the variance are explainable.

The resulted points are shown in Figure 3.

5.3.3 Random Forest

The results of random forest models are not as good as expected. According to our evaluation above, it is likely to because the our features are not efficient enough for regression. If some more powerful combinations of features can be created, the result might improve. Furthermore, one of the biggest difference in my setting for random forest and the boosting model is that the boosting model only contain trees with single split, while the max number of iteration in random forest was set as 4. in that case, the number of trees in RF is much less than boosting. Whether this really has significant impact on the performance need to be proved further.

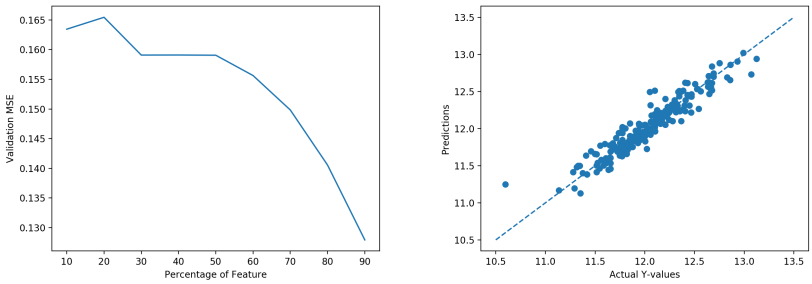

Figure 3: Backward Stepwise Selection

6 Conclusion

To conclude, this project tried Polynomial regression models, KNN regression, GAM, Random Forest and Boosting for regression to predict the house price. The features are processed with feature construction and feature selection based on forward stepwise selection, PCA and recurrent feature elimination. The results are satisfying. Further possible improvements including using more methods for feature extraction, and do more detailed evaluation considering the performance of different models.