School of Computer Science

THE UNIVERSITY *of* ADELAIDE

Assessment Cover Sheet

Student Name	
Student ID	
Assessment Title	
Course/Program	
Lecturer/Tutor	
Date Submitted	
OFFICE USE ONLY Date Received	

KEEP A COPY

Please be sure to make a copy of your work. If you have submitted assessment work electronically make sure you have a backup copy.

PLAGIARISM AND COLLUSION

Plagiarism: using another person's ideas, designs, words or works without appropriate acknowledgement.
Collusion: another person assisting in the production of an assessment submission without the express requirement, or consent or knowledge of the assessor.

CONSEQUENCES OF PLAGIARISM AND COLLUSION

The penalties associated with plagiarism and collusion are designed to impose sanctions on offenders that reflect the seriousness of the University's commitment to academic integrity. Penalties may include: the requirement to revise and resubmit assessment work, receiving a result of zero for the assessment work, failing the course, expulsion and/or receiving a financial penalty.

I declare that all material in this assessment is my own work except where there is clear acknowledgement and reference to the work of others. I have read the University Policy Statement on Academic Honesty & Assessment Obligations (http://www.adelaide.edu.au/policies/230).
I give permission for my assessment work to be reproduced and submitted to other academic staff for the purposes of assessment and to be copied, submitted to and retained by the University's plagiarism detection software provider for the purposes of electronic checking of plagiarism.

Signed……………………………………………… Date …………………………………………

Zhe Zhang A1775543

Big Data Analysis & Project
Sep 5 2019

House Prices: Advanced Regression Techniques

Abstract

Machine learning is the next big thing that is going to disrupt the real estate market. This paper proposes a machine learning approach for solving the house price prediction problem based on The Ames Housing dataset. I choose Python as the programming language. The main tools I used in this project is Scikit-learn. I apply advanced machine learning algorithms such as lasso, ridge, SVR, Kernel Ridge, Elastic Net, Bayesian Ridge. To get more accurate result, I use ensemble methods to combine these models. The final results I got on Kaggle is 0.12043.

Keywords：Machine learning, Housing Price, Data science.

Introduction

This project aims to predict the sales price of each house based on The Ames Housing dataset.

The Kaggle provided us with the data sets needed for this competition, including the training data set (train.csv) for the training model and the test data set (test.csv) for testing the performance of the model.

Each data record represents the relevant information of each house, in which there are 1460 training data and 1460 test data respectively, and there are 79 characteristic columns of the data, of which 35 are of numerical types and 44 are of category types. A detailed description of the features is described in the competition webpage.

In this project, we are required to use the FIRST (not random) 70% of the training samples to form a "small training set" and the remaining 30% of the training samples to form a "small validation set". Then the algorithm needs to be developed by training the model on the small training set and evaluating on the small validation set. Finally, we are encouraged to run the model on the test set to obtain the estimation.

Results are evaluated on Root-Mean-Squared-Error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price. (Taking logs means that errors in predicting expensive houses and cheap houses will affect the result equally.)

In this project, I choose Python as the programming language. Python has become a common language for many data science applications. It has both the power of a general-purpose programming language and the ease of use of domain-specific scripting languages such as MATLAB or R. Python has a library for data loading, visualization, statistics, natural language processing, image processing and other functions. This large toolbox provides a large number of general and specialized functions for data scientists. One of the main advantages of using Python is the ability to interact directly with code using terminals. Machine learning and data analysis are essentially iterative processes, with data-driven analysis. These processes must have fast iterations and easy-to-intersect tools. As a common programming language, Python can also be used to create complex graphical user interfaces (graphical user interface, GUI) and Web services, or it can be integrated into existing systems.

The main tools I used in this project is Scikit-learn. Scikit-learn is an open-source project that can be used and distributed for free, and anyone can easily get access to its source code to see what's behind it. The Scikit-learn project is constantly being developed and improved, and its user community is very active. It contains many of the most advanced machine learning algorithms available, each with detailed documentation (http://scikit-learn. Org/stable/documentation). Scikit-learn is a very popular tool and the most famous Python machine learning library. It is widely used in industry and academia, and there are a large number of tutorials and code fragments on the Internet. Scikit-learn can also work with a large number of other Python scientific computing tools.

It's important to understand scikit-learn and its usage, but there are other libraries that can also improve the programming experience. Scikit-learn is based on NumPy and SciPy scientific computing libraries. In addition to NumPy and SciPy, we will also use pandas and matplotlib.

Methodology

Data Collection

The quantity & quality of the data dictate how accurate the model is. The outcome of this step is generally a representation of data which we will use for training.Using pre-collected data, by way of datasets from Kaggle, UCI, etc., still fits into this step.

Data Preparation

The first thing is to overview the data and to evaluate which features can be used to do feature engineering, and which features needs to be filtered out. Because of the use of the linear model, a series of problems need to be considered, such as multiple collinearity and whether the data obeys Gaussian distribution and so on.

The case of house price predict uses multiple independent variables to predict the dependent variable SalePrice, and the predicted variable is a continuous variable, so the multiple linear regression models can be the first choice. It is not difficult to establish a regression model, but it is difficult to analyze the model and improve the prediction accuracy. In addition, pre-data preprocessing (such as data filling) is also essential.

For the convenience of data cleaning and feature engineering, I combined the training data set and the test data set into a single data set. Besides, it is also considered that the category variable needs Label Encoder and One-Hot Encoder. This can also prevent the test set contains eigenvalues that the training set does not have, which affects the performance of the model. When training the model, the merged dataset can be re-divided into training sets and test sets according to the index.

Visualize data can help detect relevant relationships between variables or class imbalances, or perform other exploratory analysis.

Choose a Model

In this project, I use ensemble learning to improve machine learning results. Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model.

At first I test 13 models and select following six models:

```
LASSO, RIDGE, SVR, KER, ELA, BAY
```

The weight of them are:

```
W1 = 0.02; W2 = 0.2; W3 = 0.25; W4 = 0.3; W5 = 0.03; W6 = 0.2.
```

Train the Model

The goal of training is to answer a question or make a prediction correctly as often as possible.

Evaluate the Model

Even we tried many different parameters and chose the one with the highest accuracy in the test set, this accuracy may not be extended to the new data. Because we use training data to adjust parameters, we can no longer use it to evaluate the quality of the model. This is also why we need to divide the data into training sets and validation sets in the first place. We need a separate dataset to evaluate, a dataset that was not used when creating the model.

According the requirement of this project, results are evaluated on Root-Mean-Squared-Error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price.

Parameter Tuning

Finding the value of important parameters of a model (one that provides the best generalization performance) is a difficult task, but it is necessary for almost all models and datasets. There are some standard ways to do this in Scikit-learn. The most common method is grid search (grid search), which mainly refers to trying all possible combinations of parameters we care about.

After the turning, parameters I chose for above 6 models are:

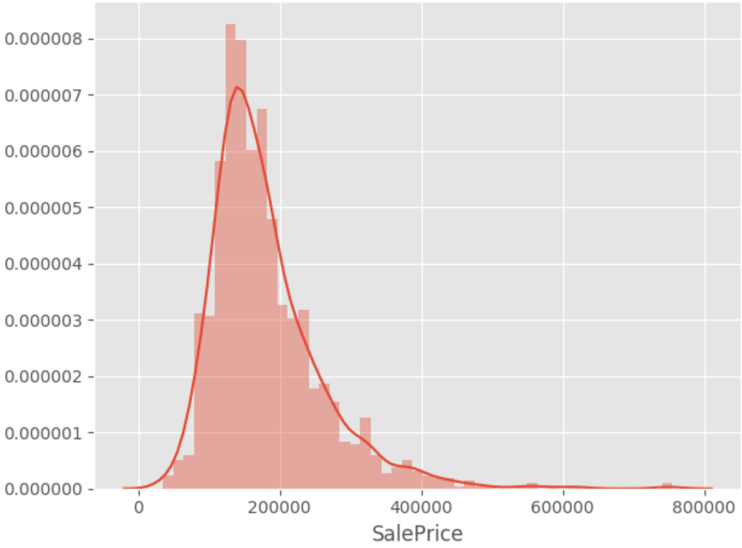
```
LASSO = LASSO(ALPHA=0.0007,MAX_ITER=10000)
RIDGE = RIDGE(ALPHA=60)
SVR = SVR(GAMMA= 0.0004,KERNEL='RBF',C=13,EPSILON=0.009)
KER = KERNELRIDGE(ALPHA=0.2 ,KERNEL='POLYNOMIAL',DEGREE=3 , COEF0=1)
ELA = ELASTICNET(ALPHA=0.005,L1_RATIO=0.1,MAX_ITER=10000)
BAY = BAYESIANRIDGE()
```

Make Predictions

Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model; a better approximation of how the model will perform in the real world.

The final results I got on Kaggle is 0.12043.

Experiment

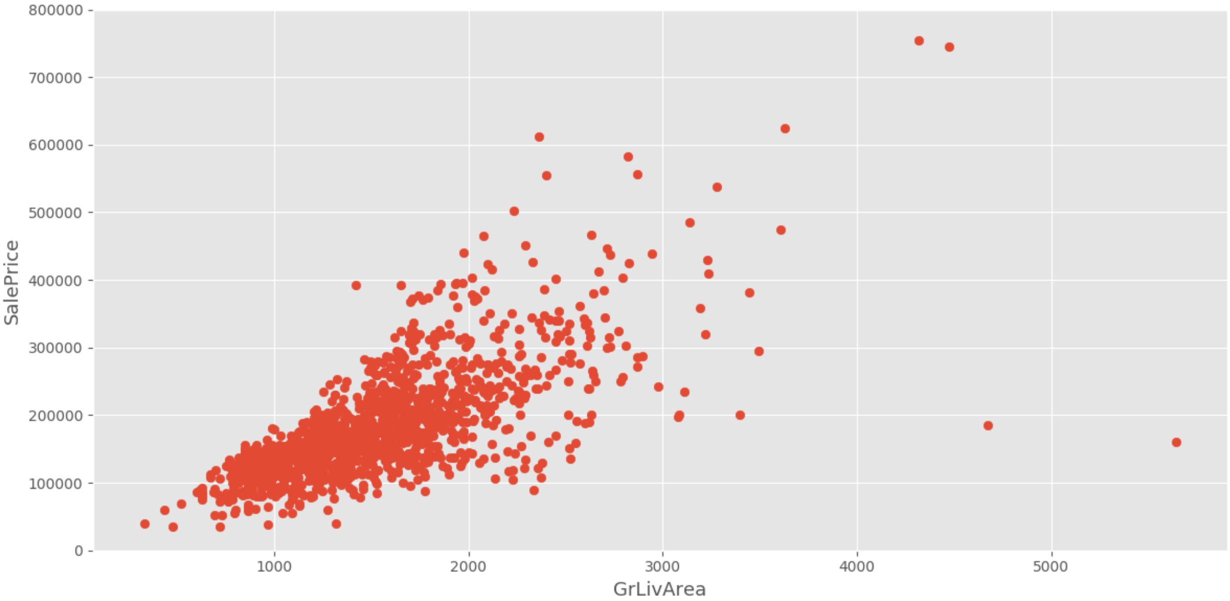


count	1460.00
mean	180921.20
std	79442.50
min	34900.00
25%	129975.00
50%	163000.00
75%	214000.00
max	755000.00

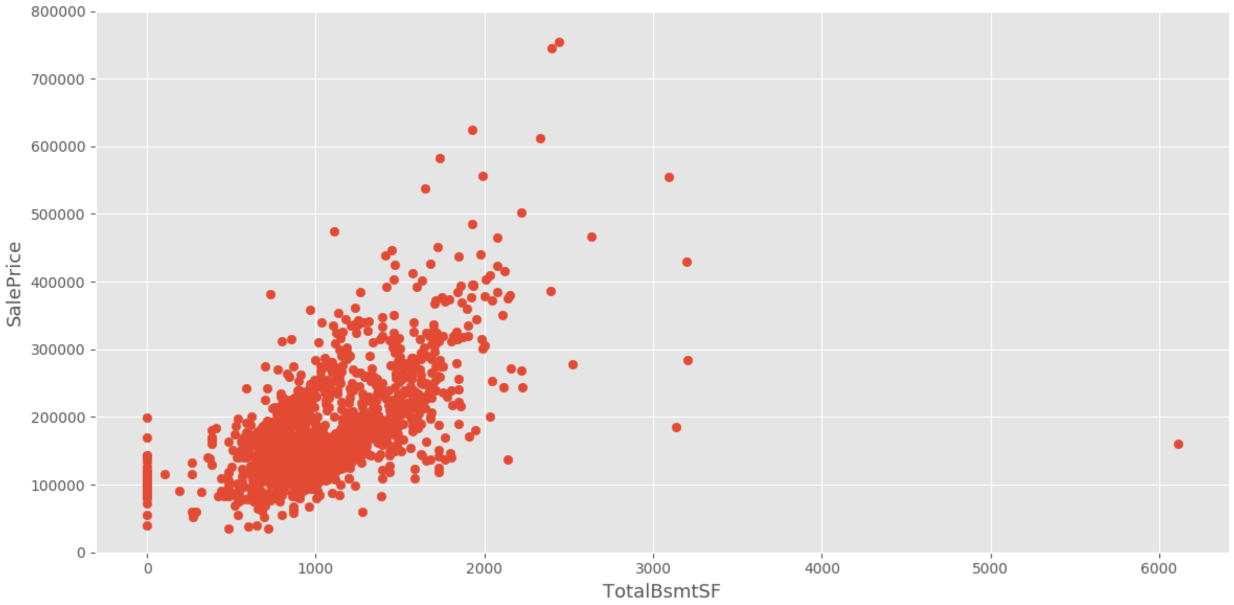
From the histogram we can see that the SalePrice deviates from normal distribution.

The skewness and kurtosis are:

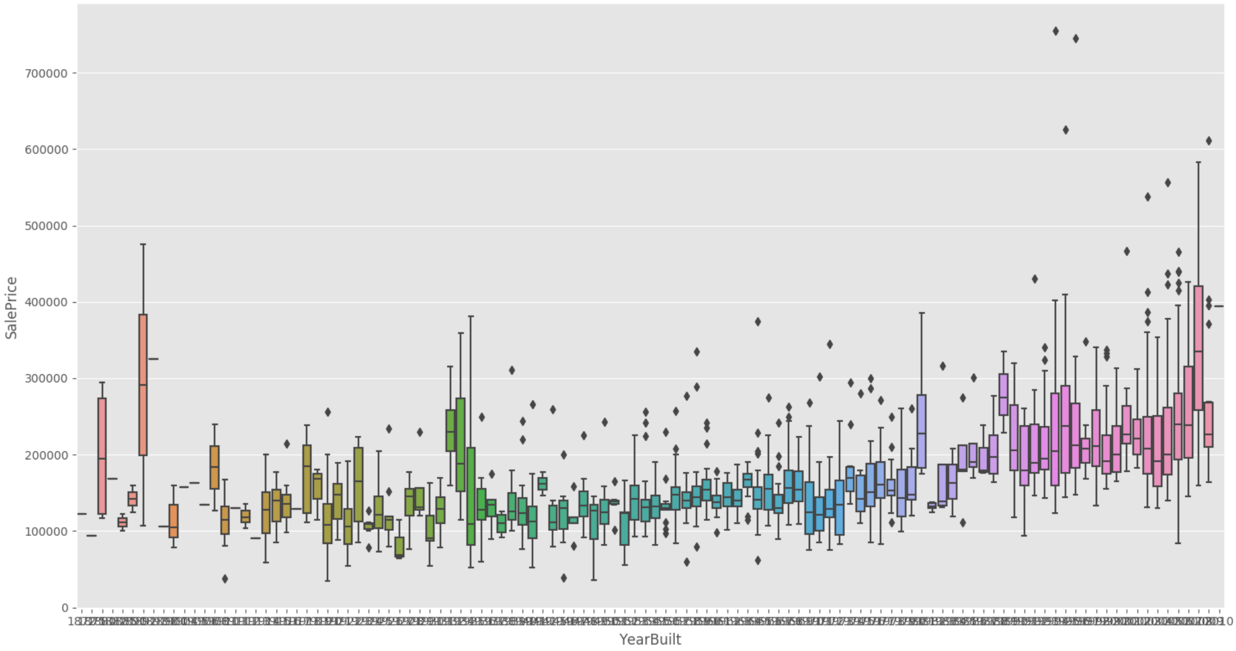
```
SKEWNESS: 1.882876
KURTOSIS: 6.536282
```



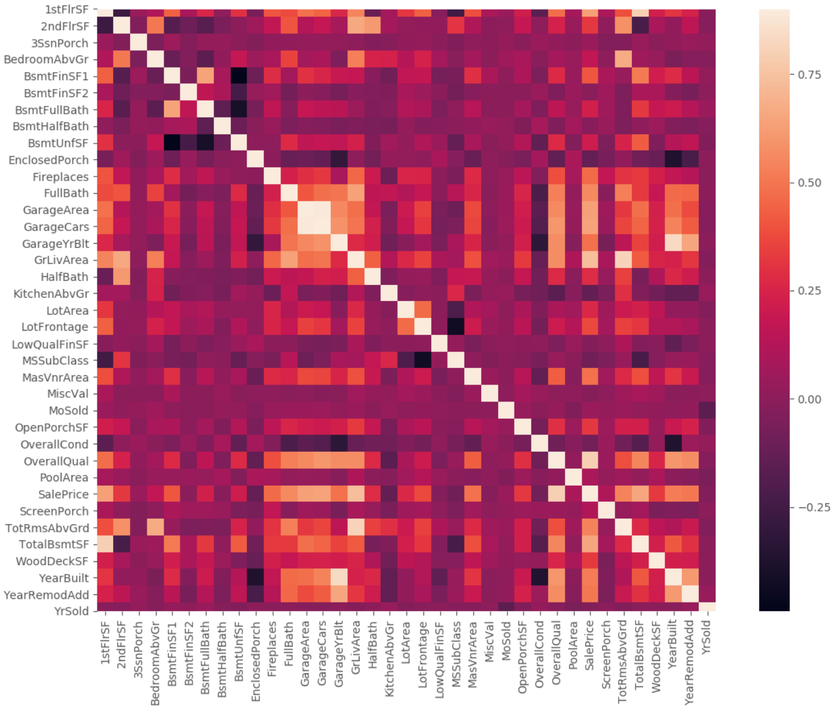
It can be seen that SalePrice and GrLivArea are closely related and basically linear.

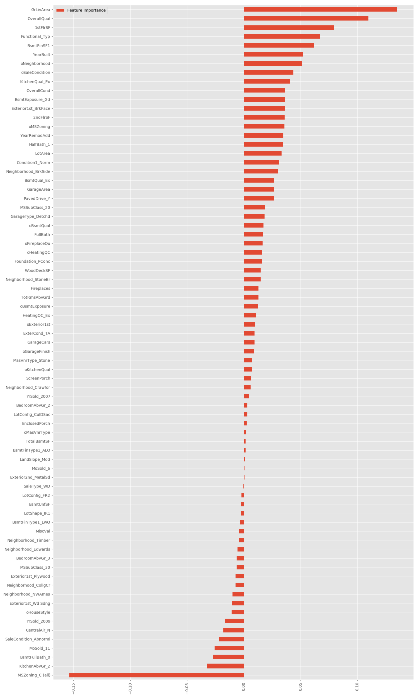


TotalBsmtSF and SalePrice are also closely related, and it can be seen from the diagram that the distribution is basically exponential. As for the leftmost point, we can see that TotalBsmtSF has no effect on SalePrice in certain cases.



There is no strong trend in the relationship between the two variables, but it can be seen that the price of houses with shorter construction time is higher.





The correlation coefficients of TotalBsmtSF and 1stFlrSF variables, and the GarageX variable group both show a strong correlation between these variables. The degree of correlation reaches a case of multiple collinearity. We can conclude that these variables contain almost the same information.

Combining different features is usually a good way, but we have no idea what features should we choose. Luckily there are some models that can provide feature selection, Here I use Lasso.

When more than 15% of the data is missing, we should delete the relevant variable and assume that it does not exist. Finally, we should deal with some missing data, outliers and classified data.

Discussion & Conclusion

In this project, I have learned the basic concepts and applications of machine learning, the most commonly used machine learning algorithms in practice, the advantages and disadvantages of these algorithms, and the importance of the presentation of data to be processed in machine learning. And what aspects of the data should be focused on, the advanced methods of model evaluation and parameter adjustment, etc.

I analyzed many variables, not only the SalePrice alone but also combined with the most relevant variables. I deal with missing data and outliers, verify some basic statistical assumptions, and convert category variables to virtual variables.

Feature engineering plays an important role in data mining, and it determines the upper limit of the accuracy of the predicted result. It's worth to spend time on feature engineering.

When testing models, I found that tree models are generally not as good as linear modules. For a single model, the Elastic Net model behaves the best, which got 0.112025 during the evaluation. Then followed by the Kernel Ridge model, which is 0.112659.

Using only one model for prediction, the accuracy is not satisfied (the final score is around 0.139 when simply using he Kernel Ridge model). In order to improve the prediction accuracy, the ensemble method is necessary.

In addition, there is a Meta-model Stacking way to raise the score to 0.11577. However, it does not seem to be able to manually divide the training set and the validation set, so I switched to the ensemble method.

Reference

Müller, A. and Guido, S. (n.d.). Introduction to machine learning with Python.

Kaggle.com. (2019). *All You Need is PCA (LB: 0.11421, top 4%) | Kaggle*. [online] Available at: https://www.kaggle.com/massquantity/all-you-need-is-pca-lb-0-11421-top-4/data [Accessed 6 Sep. 2019].

Kaggle.com. (2019). *Stacked Regressions : Top 4% on LeaderBoard | Kaggle*. [online] Available at: https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard [Accessed 6 Sep. 2019].

Kaggle.com. (2019). *All You Need is PCA (LB: 0.11421, top 4%) | Kaggle*. [online] Available at: https://www.kaggle.com/massquantity/all-you-need-is-pca-lb-0-11421-top-4/data [Accessed 6 Sep. 2019]