Final Project – Prediction of Boston Housing Price

1.Introduction.

1.1 Overview.

A house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. This project is going to be focused on solving the problem of predicting house prices for house buyers and house sellers.

In the project, we will evaluate the performance and predictive power of models that has been trained and tested on data collected from houses in suburbs of Boston, Massachusetts. Models trained on this data that is seen as a good fit could then be used to make certain predictions about a house – in particular, its monetary value. The models would prove to be invaluable for someone like a real estate agent who could make use of such information on a daily basis, or someone like you and me who would like to find their dream home with a reasonable price tag.

1.2 Outline of the shared work.

a. Data Preprocessing.

b. Find the best model that is suitable for the dataset.

c. Build linear regression model.

2. Description of Dataset

2.1 Overview of the dataset.

There are total 14 columns in the dataset. The column "MEDV" is the median of house price of self-occupied house. Other columns are features that are related to the house price.

Table2- 1 The overview of the dataset

#	Column	Description	Non-Null Count	Dtype
0	CRIM	Crime rate per capita in towns	486 non-null	float64
1	ZN	Proportion of residential land	486 non-null	float64
2	INDUS	Proportion of non-commercial land in urban areas	486 non-null	float64
3	CHAS	Charles River Dummy Variable	486 non-null	float64
4	NOX	Environmental protection index	506 non-null	float64
5	RM	Number of rooms per house	506 non-null	float64
6	AGE	Proportion of self-occupied units built before 1940	486 non-null	float64
7	DIS	Weighted distance from Boston's five employment centers	506 non-null	float64
8	RAD	Convenience Index to Highway	506 non-null	int64
9	TAX	Real estate tax rate per US$10,100	506 non-null	int64
10	PTRATIO	Teacher-student ratio in towns	506 non-null	float64
11	B	Proportion of blacks in towns	506 non-null	float64

12	LSTAT	Proportion of landlords belonging to the lower income class	486 non-null	float64
13	MEDV	Median of house price of self-occupied house	506 non-null	float64

2.2 The statistics information.

"CHAS" is a dummy variable, other variables are all continuous variables. According to the minimum, median and maximum, "CRIM", "ZN", "AGE", "B" and "LSTAT" are skewed distributed. According to the count, some columns exist missing values.

Table2- 2 The statistics information of the dataset

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
count	486	486	486	486	506	506	486
mean	3.612	11.212	11.084	0.070	0.555	6.285	68.519
std	8.720	23.389	6.836	0.255	0.116	0.703	28.000
min	0.006	0.000	0.460	0.000	0.385	3.561	2.900
median	0.254	0.000	9.690	0.000	0.538	6.209	76.800
max	88.976	100.000	27.740	1.000	0.871	8.780	100.000

	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
count	506	506	506	506	506	486	506
mean	3.795	9.549	408.237	18.456	356.674	12.715	22.533
std	2.106	8.707	168.537	2.165	91.295	7.1561	9.197
min	1.130	1.000	187.000	12.600	0.320	1.730	5.000
median	3.207	5.000	330.000	19.050	391.440	11.430	21.200
max	12.127	24.000	711.000	22.000	396.900	37.970	50.000

2.3 Missing values.

The columns that exist missing values are "CRIM", "ZN", "INDUS", "CHAS", "AGE" and "LSTAT". Then, we summarize the amount of missing values for each column. They are shown as follows:

Table2- 3 The summary of missing values

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE
20	20	20	20	0	0	20

DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
0	0	0	0	0	20	0

3. Models and Algorithm

3.1 Feature engineering -transformation.

a. Box-Cox transformation.

Box-Cox transformation is a generalized power transformation method which is commonly used in statistical modeling. It is used when continues variables do not satisfy the normal distribution. After the Box-Cox transformation, the correlation between unobservable errors and predictors can be reduced to some extent. The main feature of the Box-Cox transformation is the introduction of a parameter, which is estimated by the data itself to determine the form of data transformation that should be adopted. The Box-Cox transformation can significantly improve the normality, symmetry and variance equality of the data. The characteristic of the Box-Cox transformation is that it can only be applied to samples containing positive values. The transfer function is shown as follows:

$$y^{(\lambda)} = \begin{cases} \dfrac{y^{\lambda} - 1}{\lambda}, & \lambda \neq 0 \\ \log y, & \lambda = 0 \end{cases}$$

We need to determine the transformation parameter λ so that $y^{(\lambda)}$ satisfies

$$y^{(\lambda)} = X\beta + e, e \sim N(0, \sigma^2 I)$$

We use the maximum likelihood method to determine the transformer parameter λ. For fixed λ, β and σ^2, the likelihood function is

$$L(\beta, \sigma^2) = \frac{1}{(\sqrt{2\pi\sigma})^n} \exp\left\{-\frac{1}{2\sigma^2}\left(y^{(\lambda)} - X\beta\right)'\left(y^{(\lambda)} - X\beta\right)\right\} J$$

$$J = \prod_{i=1}^{n} \left|\frac{dy_i^{(\lambda)}}{dy_i}\right| = \prod_{i=1}^{n} y_i^{\lambda - 1}$$

Then we can obtain the maximum likelihood estimation of β and σ^2, the residual sum of squares and the responding maximum of likelihood function. It is a unary function for λ, so we can easily get the value of λ.

b. Yeo-Johnson transformation.

Yeo-Johnson transformation is also one of the power transformation methods by constructing a set of monotone functions to transform data of random variables. It can reduce the heteroscedasticity of random variables and amplify its normality, making its probability density function form closer to normal distribution. The characteristic of the Yeo-Johnson transformation is that it can be applied to samples containing 0 and negative values, so it is also considered to be the extension of the Box-Cox transform in the real filed. The transfer function is shown as follows:

$$y^{(\lambda)} = \begin{cases} \log(X+1) & if\ \lambda = 0, X > 0 \\ \dfrac{(X+1)^{\lambda} - 1}{\lambda} & if\ \lambda \neq 0, X > 0 \\ -\log(-X+1) & if\ \lambda = 2, X \leq 0 \\ \dfrac{-\left[(-\mathrm{X}+1)^{(2-\lambda)} - 1\right]}{2-\lambda} & if\ \lambda \neq 2, X \leq 0 \end{cases}$$

As for the determination of λ, it is the same as the Box-Cox transformation. It also uses the maximum likelihood estimation to determine the value of λ.

Yeo-Johnson transformation is one of the data standardization methods used in the data preprocessing stage of data mining and machine learning. When the normality of the random variables is poor, it is processed using the Yeo-Johnson transformation, which is beneficial to the modeling.

3.2 Standardization.

Standardization is also called zero-mean normalization. After standardization, the mean value of the data is 0 and the standard deviation is 1. The conversion formula is:

$$x_i' = \frac{x_i - \bar{x}}{s}$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

The z-score standardization method is suitable for cases when there is outlier data that exceeds the value range. It is currently the most commonly used data standardization method.

3.3 Multiple linear regression.

3.3.1 Definition.

Multiple linear regression is a machine learning algorithm based on supervised learning. It is a statistical approach to modeling the relationship between a dependent variable and a given set of independent variables.

3.3.2 Expression.

$$y = b_0 + b_1 x_1 + \cdots + b_n x_n$$

y is the dependent variable, $x_1 \ldots x_n$ are multiple independent variables.

Also, we can vectorize the above expression as follows:

$$y = Xb$$

3.3.3 Cost function.

We use the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation as the cost function of the linear regression model.

$$S = \|Xb - y\|^2$$

By differentiating and minimizing S, we can get

$$X^T Xb = X^T y$$

If the matrix $X^T X$ is not singular, then there is a unique solution for b:

$$b = (X^T X)^{-1} X^T y$$

3.6 Evaluation index of models.

3.6.1 Mean squared error.

Mean squared error (MSE) of an estimator measures the average of the squares of the errors – that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss. It is always non-negative, and values closer zero are better.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

3.6.2 R-squared.

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. It is the percentage of the response variable variation that is explained by a linear model. In general, the higher the r-squared, the better the model fits the data. The formula is:

$$R^2 = 1 - \frac{unexplained\ variation}{total\ variation}$$

4. Experimental Setup

4.1 Data preprocessing.

4.1.1 Filling the missing values.

a. The distribution of features that exist missing values.

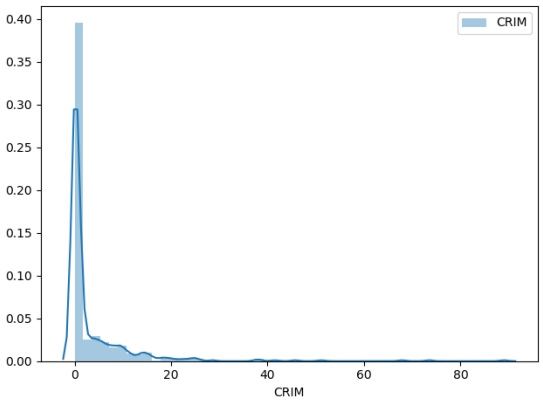

Figure2- 1 The distribution of "CRIM"

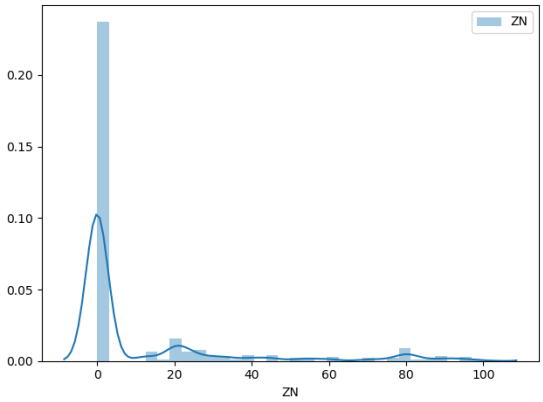

Figure2- 2 The distribution of "ZN"

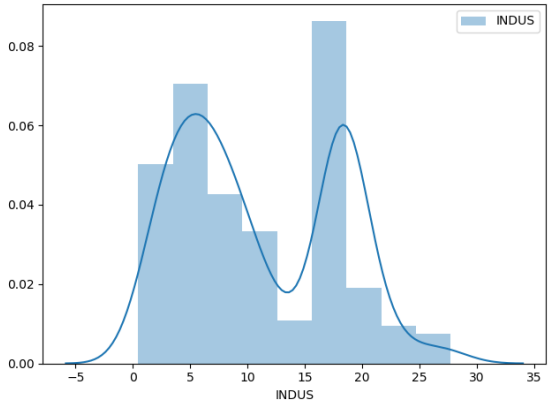

Figure2- 3 The distribution of "INDUS"

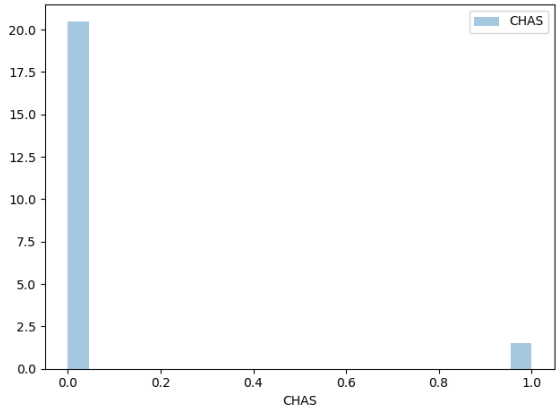

Figure2- 4 The distribution of "CHAS"

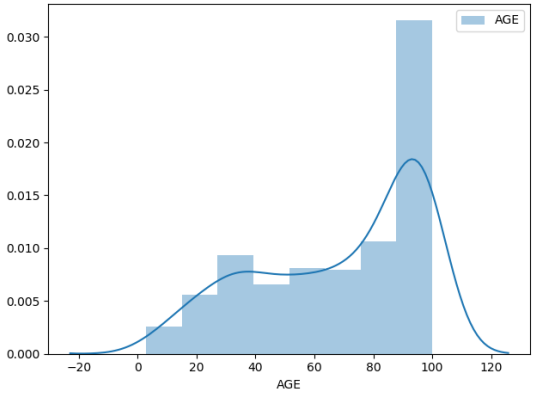

Figure2- 5 The distribution of "AGE"

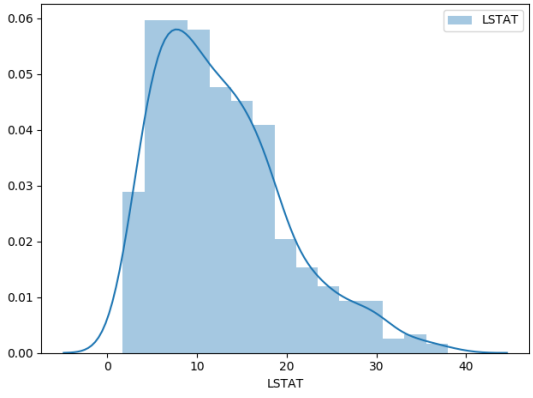

Figure2- 6 The distribution of "LSTAT"

b. Filling methods.

For the “INDUS” feature, its distribution does not show obvious skewness, so we choose the “mean” method to fill the missing values. For the “CRIM”, “ZN”, “AGE” and “LSTAT” features, their distributions show obvious skewness, so we choose the “median” method to fill the missing values. For the “CHAS” feature, it is categorical data, so we choose the “mode” method to fill the missing values.

In the Python, we use the Imputer package to fill the missing values with different strategies.

4.1.2 Non-linear Transformation and standardization.

For the dataset, it has both positive and negative samples, so we use Yeo-Johnson transformation as our transform function. Because it can be applied to samples containing 0 and negative values. After the process, the dataset became more Gaussian-like.

In the Python, we use the Power Transformer to achieve the non-linear transformation and standardization.

4.1.3 Significance of Features.

We use univariate feature selection to select significant features. It works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator. The method based on F-test estimate the degree of linear dependency between two random variables. On the other hand, mutual information methods can capture any kind of statistical dependency, but being nonparametric, they require more samples for accurate estimation.

In the Python, SelectKBest removes all but the k highest scoring features and return the p_value for every feature. For regression problems, we would better use f_regression and mutual_info_regression as our score function. The p_values for each column is shown as follows:

Table4- 1 The significance of features

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
8.59E-24	2.08E-18	5.06E-34	3.17E-05	3.79E-24	4.33E-69	4.92E-18

DIS	RAD	TAX	PTRATIO	B	LSTAT
9.35E-12	2.35E-14	1.32E-29	4.91E-35	7.22E-14	1.75E-107

We find that all features are significant because that all p-values are smaller than 0.05.

Then we plot the relationship between the target price and every feature separately and calculate the f-test score and mutual info score. Also, it indicates that all features are significant.

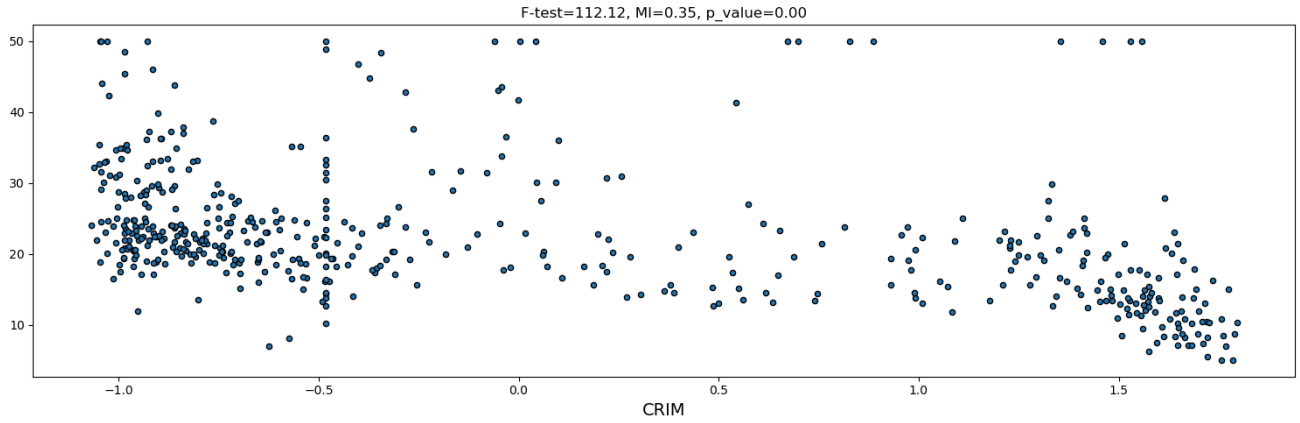

Figure4- 1 The significance of "CRIM"

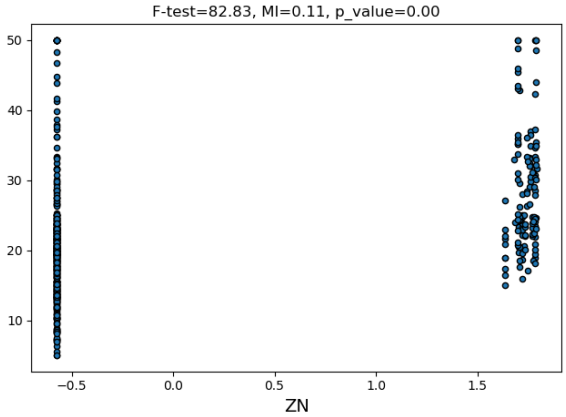

Figure4- 2 The significance of "ZN"

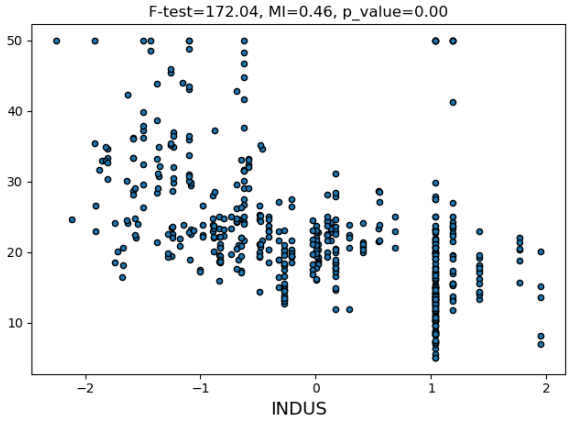

Figure4- 3 The significance of "INDUS"

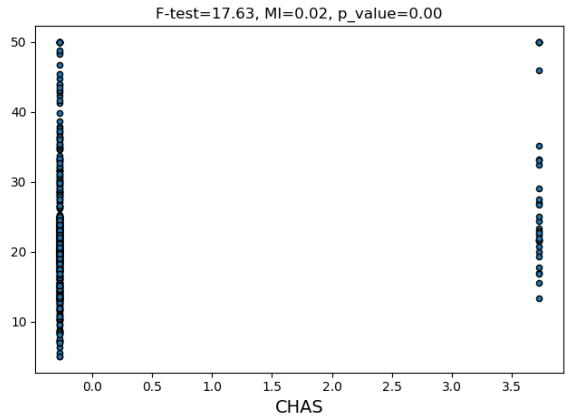

Figure4- 4 The significance of "CHAS"

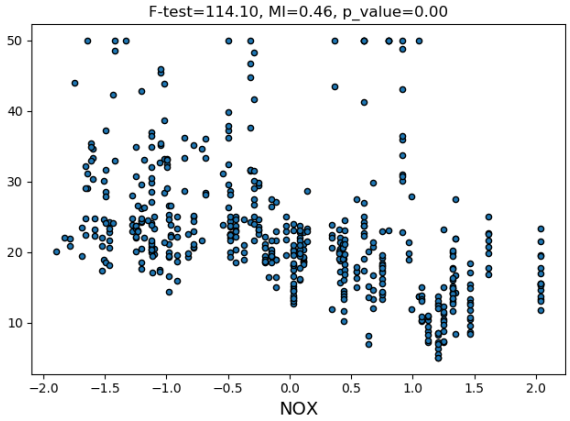

Figure4- 5 The significance of "NOX"

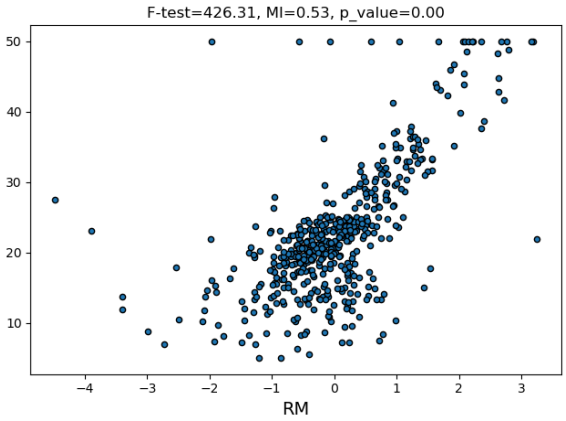

Figure4- 6 The significance of "RM"

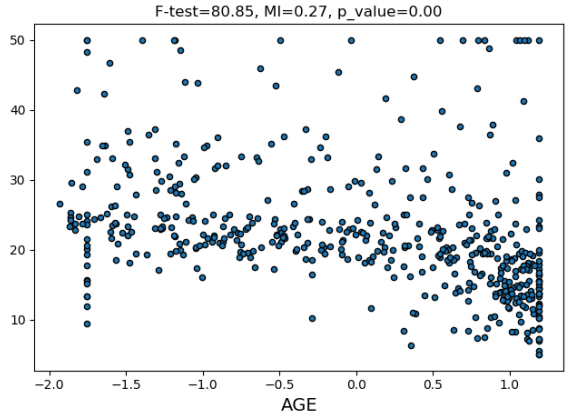

Figure4- 7 The significance of "AGE"

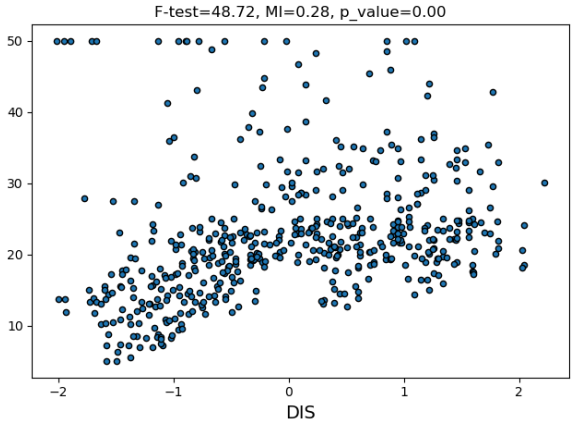

Figure4- 8 The significance of "DIS"

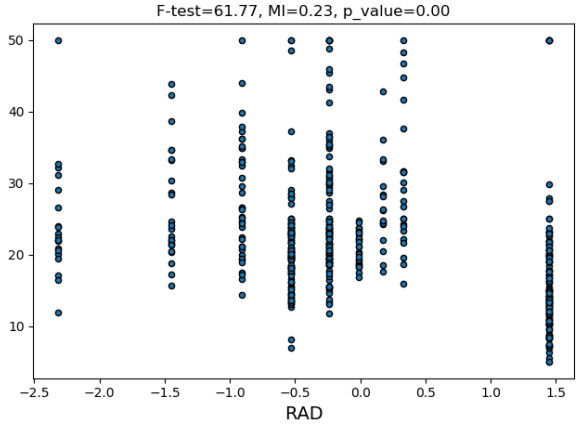

Figure4- 9 The significance of "RAD"

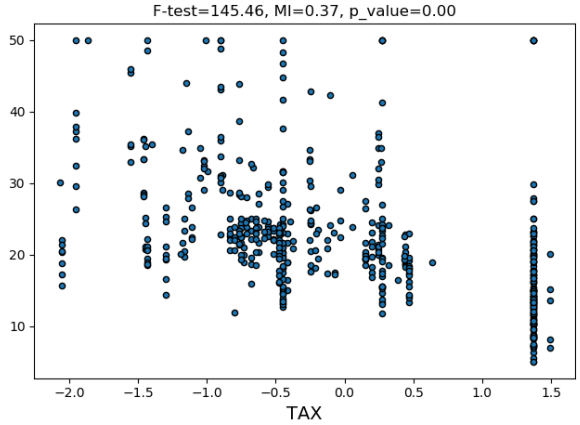

Figure4- 10 The significance of "TAX"

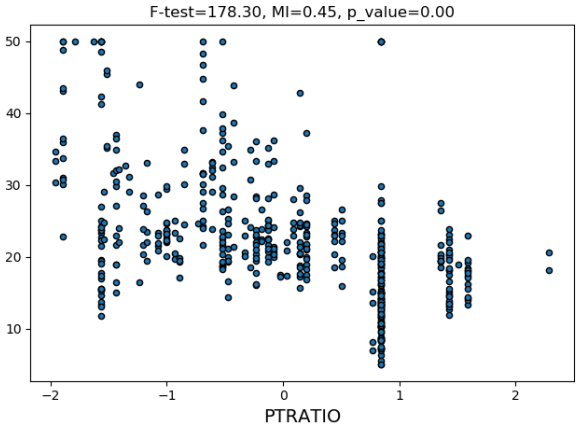

Figure4- 11 The significance of "PTRATIO"

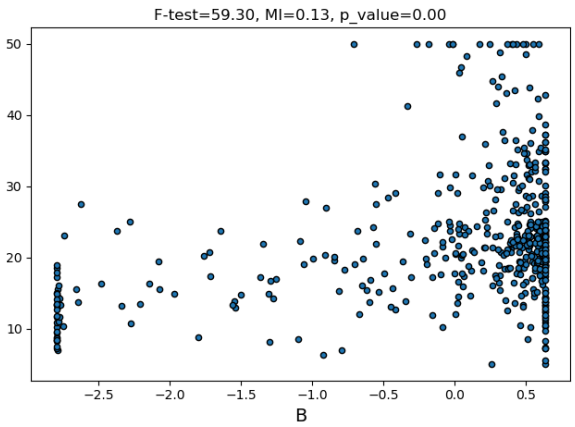

Figure4- 12 The significance of "B"

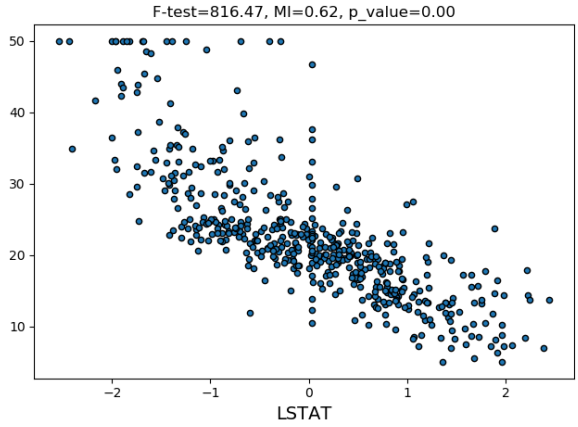

Figure4- 13 The significance of "LSTAT"

4.2 Linear Regression.

First, we separate the dataset to train data and test data. Then, we use the Linear Regression to achieve the ordinary least squares. In the python, we set all parameters as default and get the coefficients for features and the intercept. At the same time, we can calculate the mean square error and R square as the index to judge our model's performance. It is also the basis for judging other machine learning models.

5. Results

5.1 Linear Regression.

5.1.1 The coefficients of the linear regression model.

Table4- 2 The coefficients of the linear regression model

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
0.1286	0.1582	-0.9226	0.8054	-2.2907	1.6932	-0.0723

DIS	RAD	TAX	PTRATIO	B	LSTAT	Intercept
-3.8692	1.0019	-1.2698	-1.3697	0.1950	-5.2169	22.5933

5.1.2 The performance of the model.

The mean square errors is 20.2388 and R-square is 0.7957, which means that almost 80% change of house price can be explained by the linear regression model.

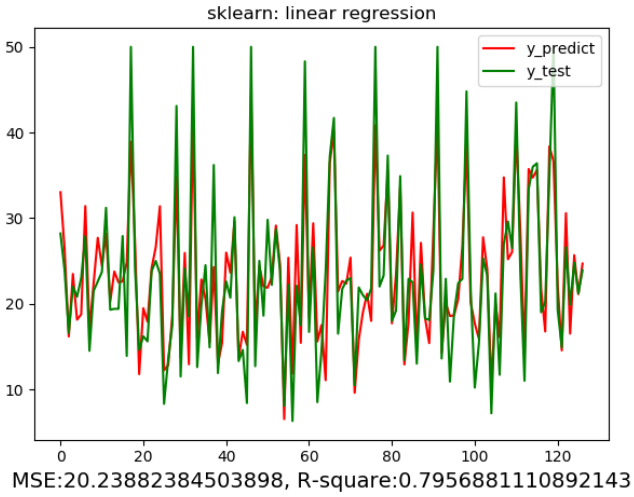

Figure4- 14 The comparison between the predicted value and true value

6. Percentage of the code from the internet.

(3 / (3+105)) * 100 = 2.78

7. Reference

[1] scikit-learn Machine learning in Python. (n.d.). Retrieved from https://scikit-learn.org/stable/

[2] Martin T.H., Howard B.D., Mark H.B., Orlando D.J.(n.d.). Neuron Network Design.

[3] Sebastian R., Vahid M. (2017). Python Machine Learning Second Edition. Packt Publishing Ltd.