Machine Learning Project

1. Data

In Rdata set, there are data of cpi, ppi, X and fake.testing.X. In order to do coding conveniently, I load Rdata in R and output 4 dataframe to csv files and read such csv files in python.

After that, I use cpi and ppi data to compute inflation rate separately, which are the target values we need to predict.



1. Models

To test the model, I use train\_test\_split to split dataset to 80% of training data and 20% of test data. Features that are fitted by linear regression models will be scaled to normalized values in order to fit the model requirement.

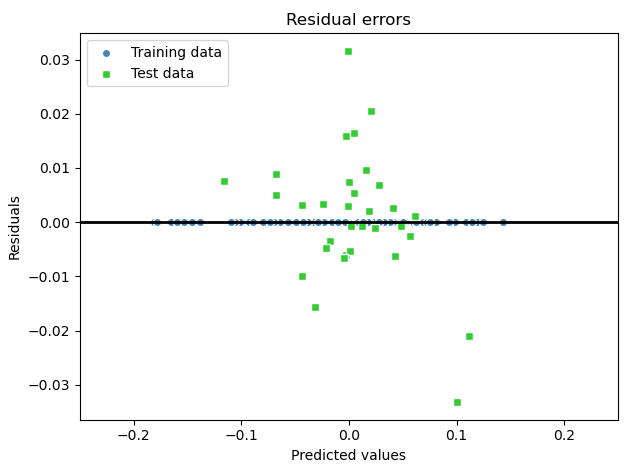
* 1. Inflation rate by cpi

An advantage of the **decision tree algorithm** is that it does not require any transformation of the features if we are dealing with nonlinear data.

The **random forest algorithm** is an ensemble technique that combines multiple decision trees. A random forest usually has a better generalization performance than an individual decision tree due to randomness, which helps to decrease the model's variance. Other advantages of random forests are that they are less sensitive to outliers in the dataset and don't require much parameter tuning. The only parameter in random forests that we typically need to experiment with is the number of trees in the ensemble.

To be more specific, Extra Trees are the improved method to solve a large number of features.

This method adds an extra level of randomization. It not only selects for each tree a different, random subset of features, but also randomly selects the threshold for each decision.



MSE train: 0.000, test: 0.000

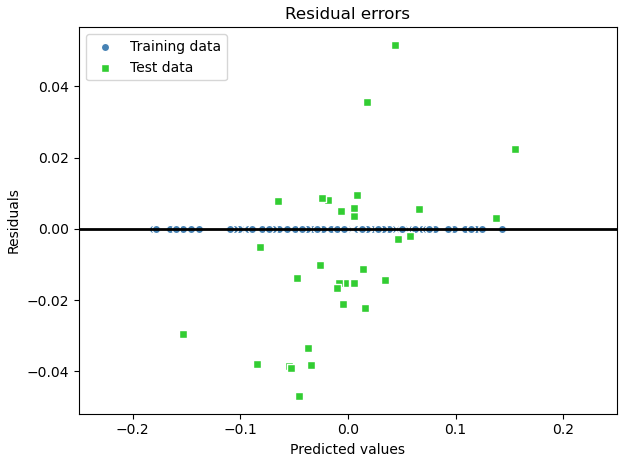
R^2 train: 1.000, test: 0.946

We can see that R^2 of out of sample data is 0.946, which is extremely good.

2.2 Inflation rate by ppi

2.2.1 First try by linear regression

First try OLS linear regression to predict inflation rate by ppi. The cost function is Gradient Descent.



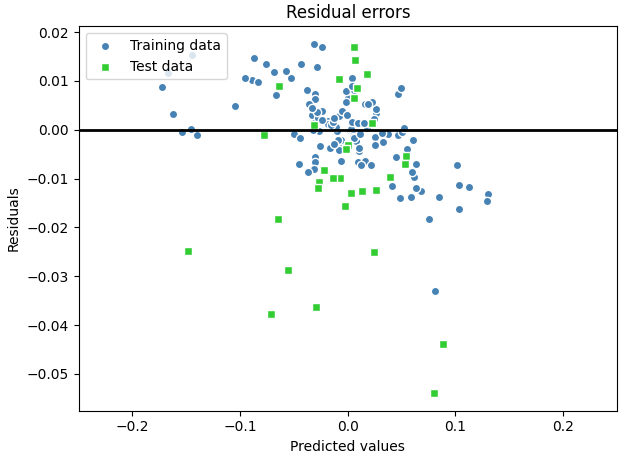
MSE train: 0.000, test: 0.001

R^2 train: 1.000, test: 0.789

We can see that R^2 of training data is very well but of test data is 0.789, which is overfitting. To solve the problem, I try to use regularized methods for regression.

2.2.2 Random Forest algorithm

Random Forest may have a good performance on ppi data.



MSE train: 0.000, test: 0.000

R^2 train: 0.984, test: 0.851

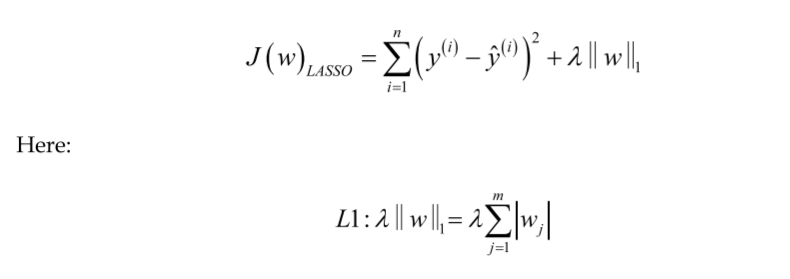
R^2 of out of sample data is 0.851, which is still not good enough.

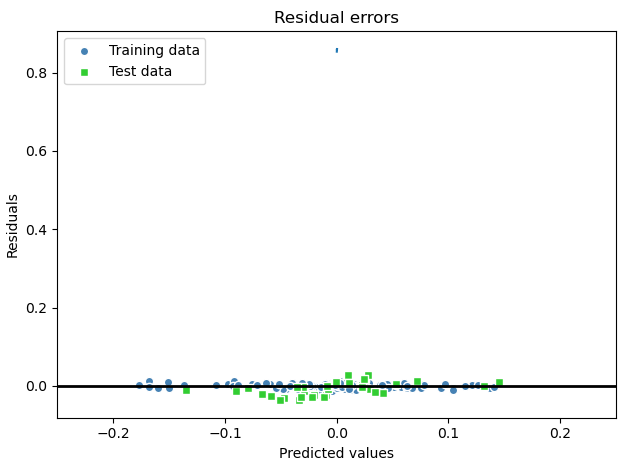
2.2.3 Tackle overfitting by LASSO

Regularization is one approach to tackle the problem of overfitting by adding additional information, and thereby shrinking the parameter values of the model to induce a penalty against complexity. The most popular approaches to regularized linear regression are the so-called Ridge Regression, Least Absolute Shrinkage and Selection Operator (LASSO), and Elastic Net.

Because of up to 167 features in x, LASSO is a better approaches. Depending on the regularization strength, certain weights can become zero, which also makes LASSO

useful as a supervised feature selection technique.





The best alpha for Lasso is 0.0002.

MSE train: 0.000, test: 0.000

R^2 train: 0.992, test: 0.860

We can see that R^2 of out of sample data is 0.86, which is better than linear regression and random forest. The overfitting problem can be improved.

1. Training and Testing

To achieve the ultimate output of the exercise, we can use **TestingFinal** function, which is updated with best machine learning model, with the input fake.testing.X and will return 30 values as its forecast of the inflation rates according to CPI and another 30 values as its forecast of inflation rates according to PPI.

The real testing ans is as following:

