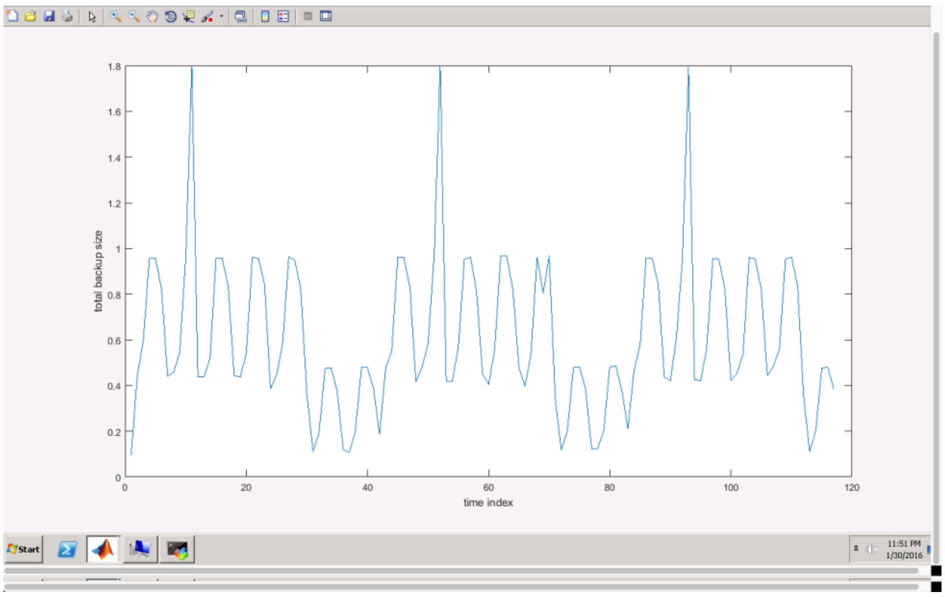
EE239AS Project 1

Network backup Dataset

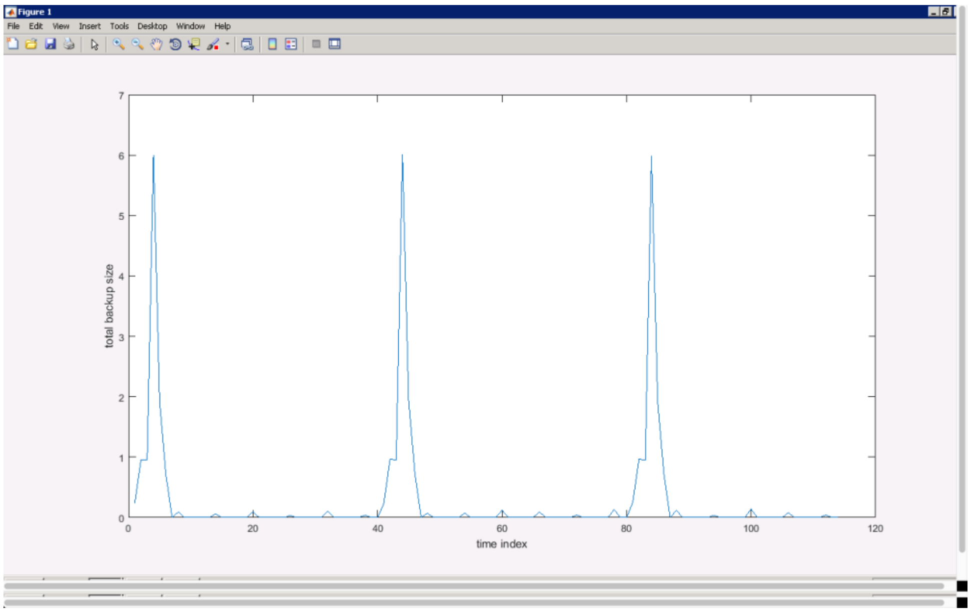
1. Plotting

We plot the actual copy sizes of all the files on day 1-20 and get the following results:

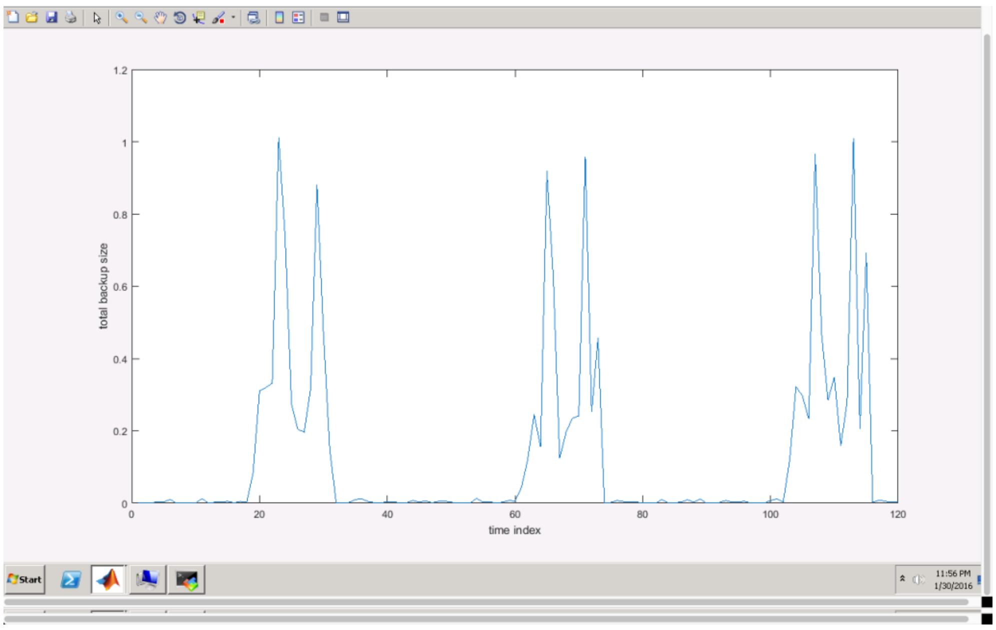
work_flow_0:



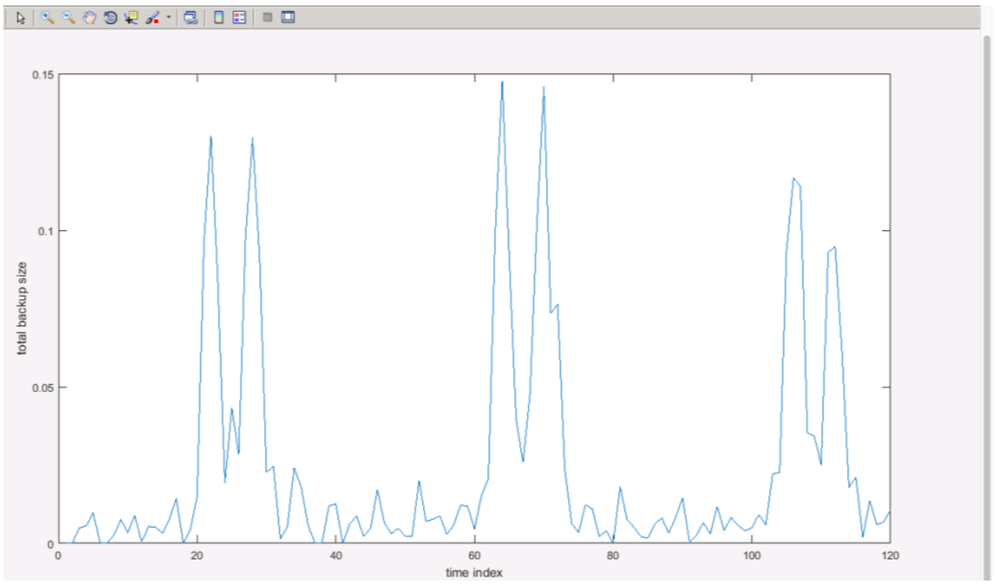
work_flow_1:



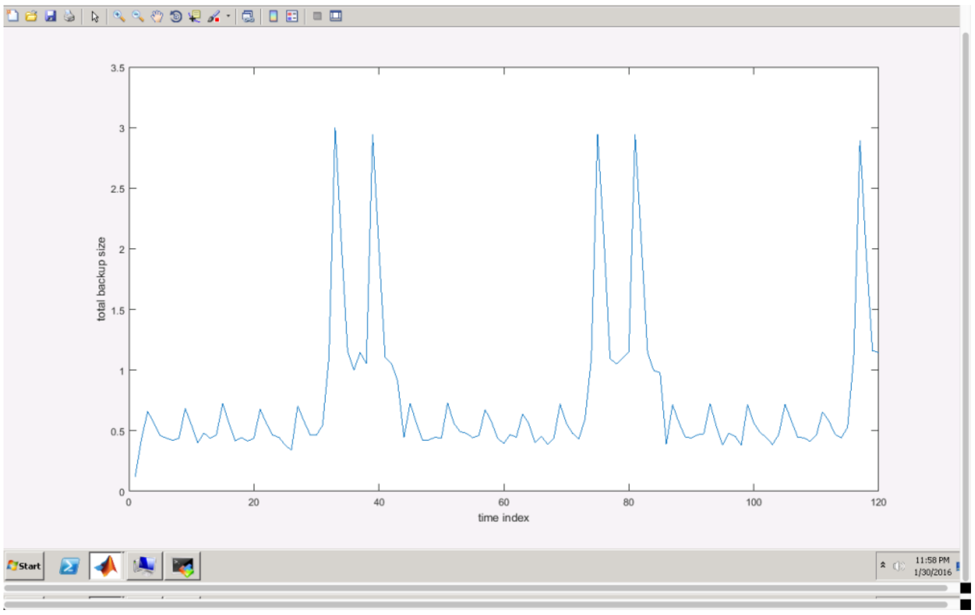
work_flow_2:



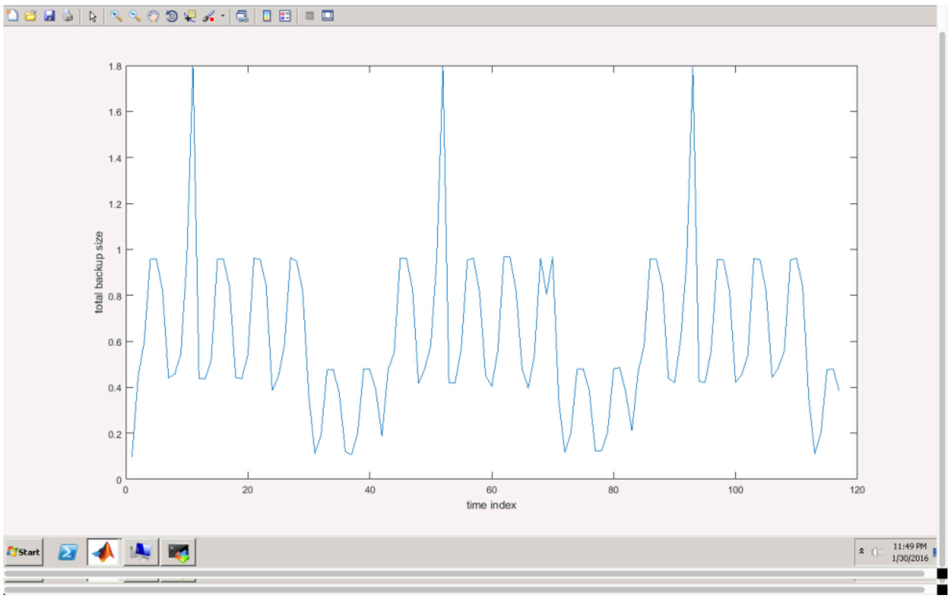
work_flow_3:



work_flow_4:



The 5 workflows together:



Conclusion:

From each workflow of the copy sizes on the period of days 1-20, we can see that the although the pattern for each workflow is different, they all repeat three times over the first 20 days. Copy sizes depend on day of the week.

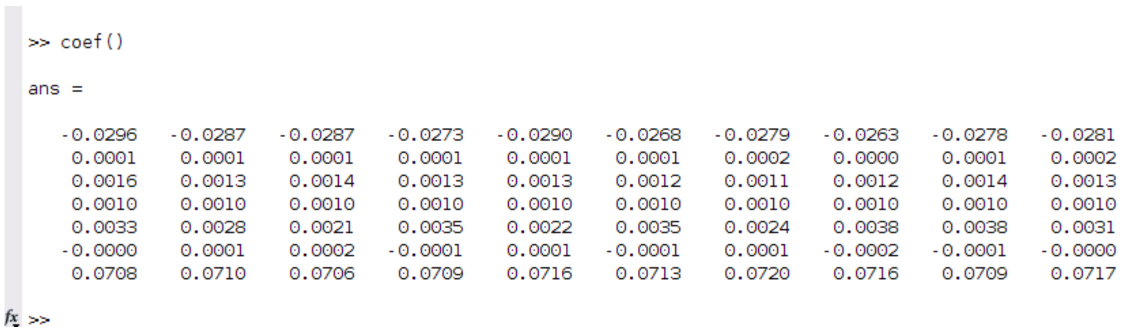
2. Predict the copy size of a file given the other attributes

a. Linear regression

(Note : Computations of RMSE are done without normalization (i.e., dividing N) which doesn`t affect the comparison and conclusion)

Analyze the significance of different variables with the statistics obtained from the model you have trained and report your obtained Root Mean Squared Error (RMSE).

The coefficient matrix is as following:

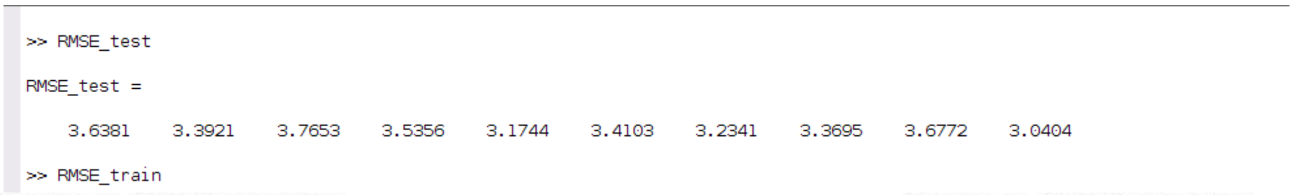


```
>> coef()

ans =

  -0.0296   -0.0287   -0.0287   -0.0273   -0.0290   -0.0268   -0.0279   -0.0263   -0.0278   -0.0281
   0.0001    0.0001    0.0001    0.0001    0.0001    0.0001    0.0002    0.0000    0.0001    0.0002
   0.0016    0.0013    0.0014    0.0013    0.0013    0.0012    0.0011    0.0012    0.0014    0.0013
   0.0010    0.0010    0.0010    0.0010    0.0010    0.0010    0.0010    0.0010    0.0010    0.0010
   0.0033    0.0028    0.0021    0.0035    0.0022    0.0035    0.0024    0.0038    0.0038    0.0031
  -0.0000    0.0001    0.0002   -0.0001    0.0001   -0.0001    0.0001   -0.0002   -0.0001   -0.0000
   0.0708    0.0710    0.0706    0.0709    0.0716    0.0713    0.0720    0.0716    0.0709    0.0717

>>
```

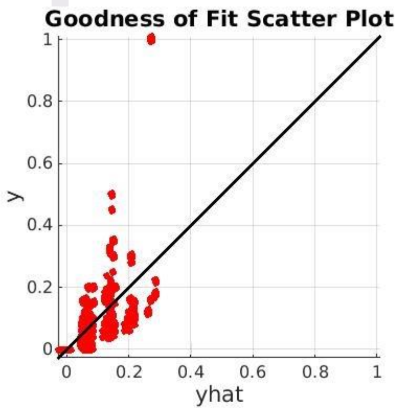
Row 1 indicates the bias of the regression result. Row 2 to 7 corresponds to features week number, day of the week, hour of the day, workflow ID, file number and time respectively. Column 1 to 10 indicates 10 tests during the 10 fold cross validation. From the result we can see that file name has little effect on the copy size (the corresponding coefficients are almost all zeros).

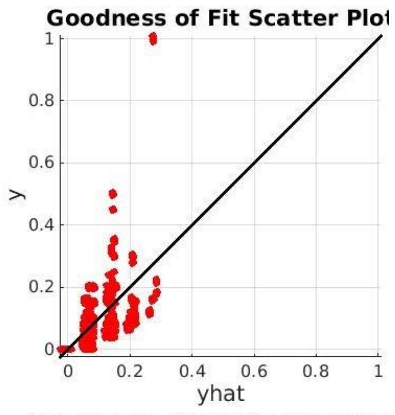
Meanwhile, copy time has the biggest significance on copy size, which is very reasonable result. In addition, second row and third row have some effect on the outcome of copy size as well, which means that time also has some influence. From the result of size versus time from the last problem, it is also very easy to see the influence of time on copy sizes.

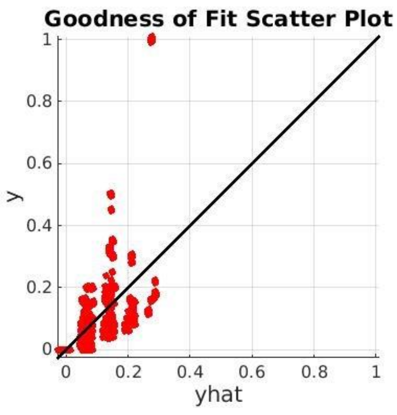
To evaluate how well the model fits the data, RMSE is as following in the 10 trials:

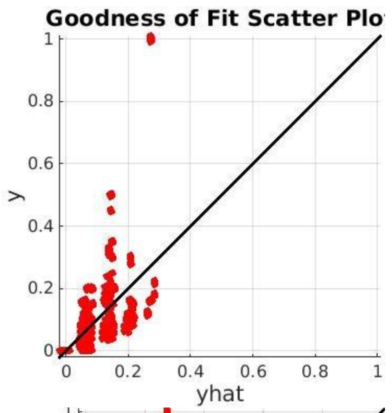


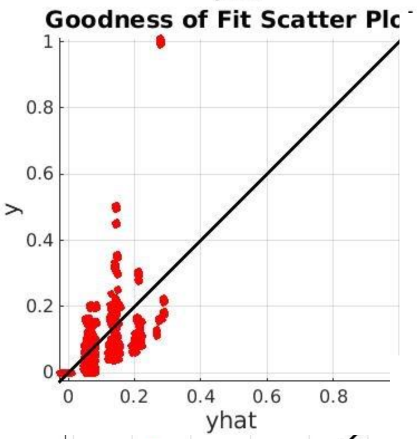
```
>> RMSE_test

RMSE_test =

    3.6381    3.3921    3.7653    3.5356    3.1744    3.4103    3.2341    3.3695    3.6772    3.0404

>> RMSE_train
```

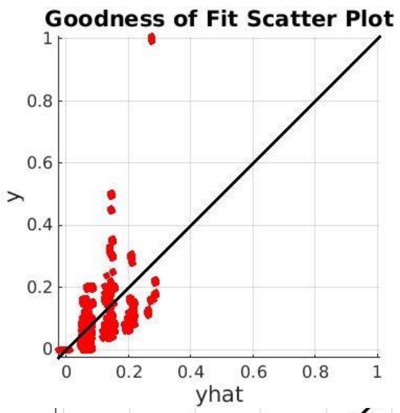


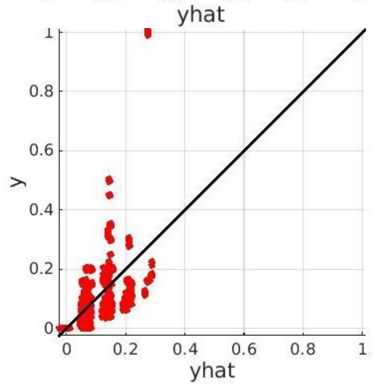


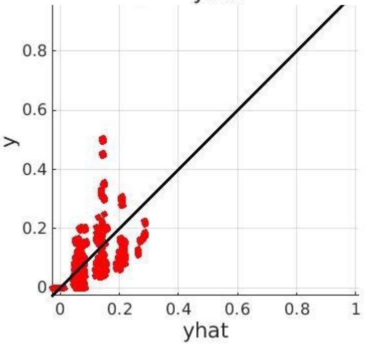


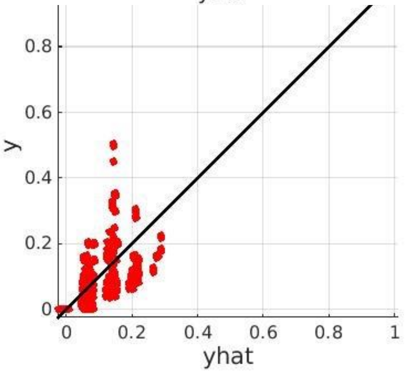


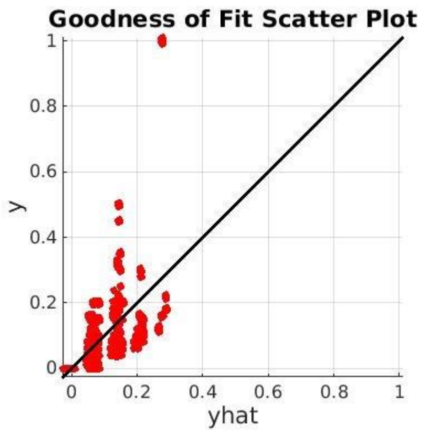


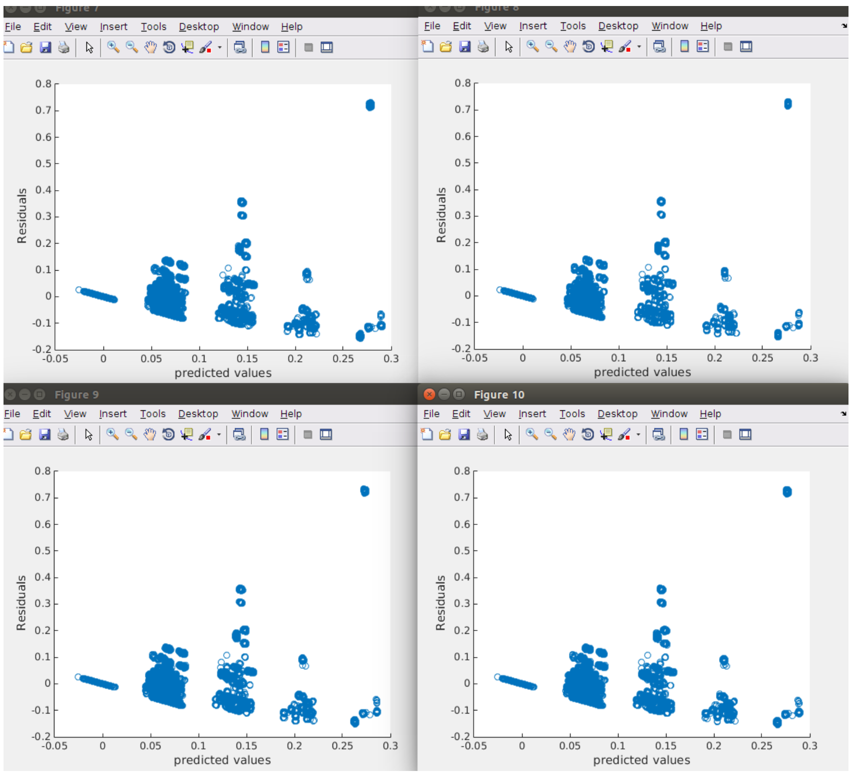


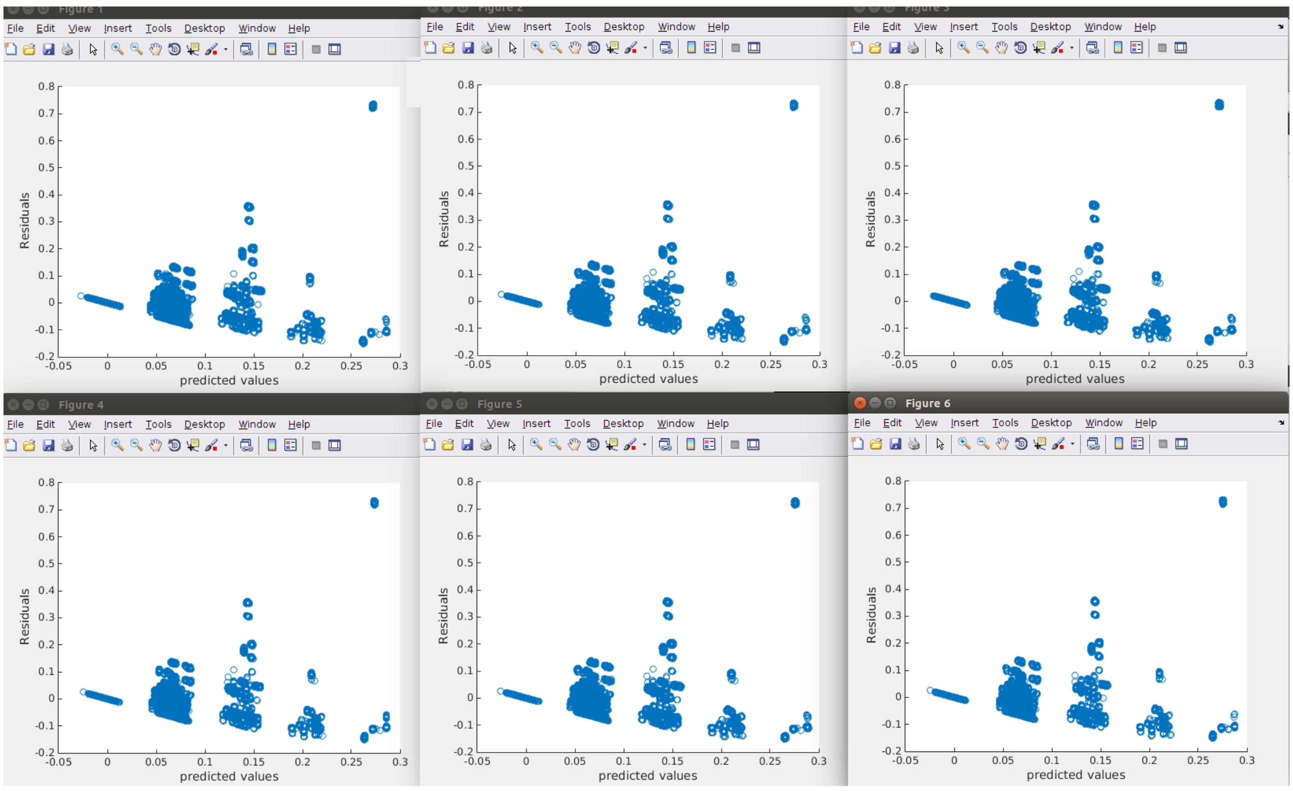










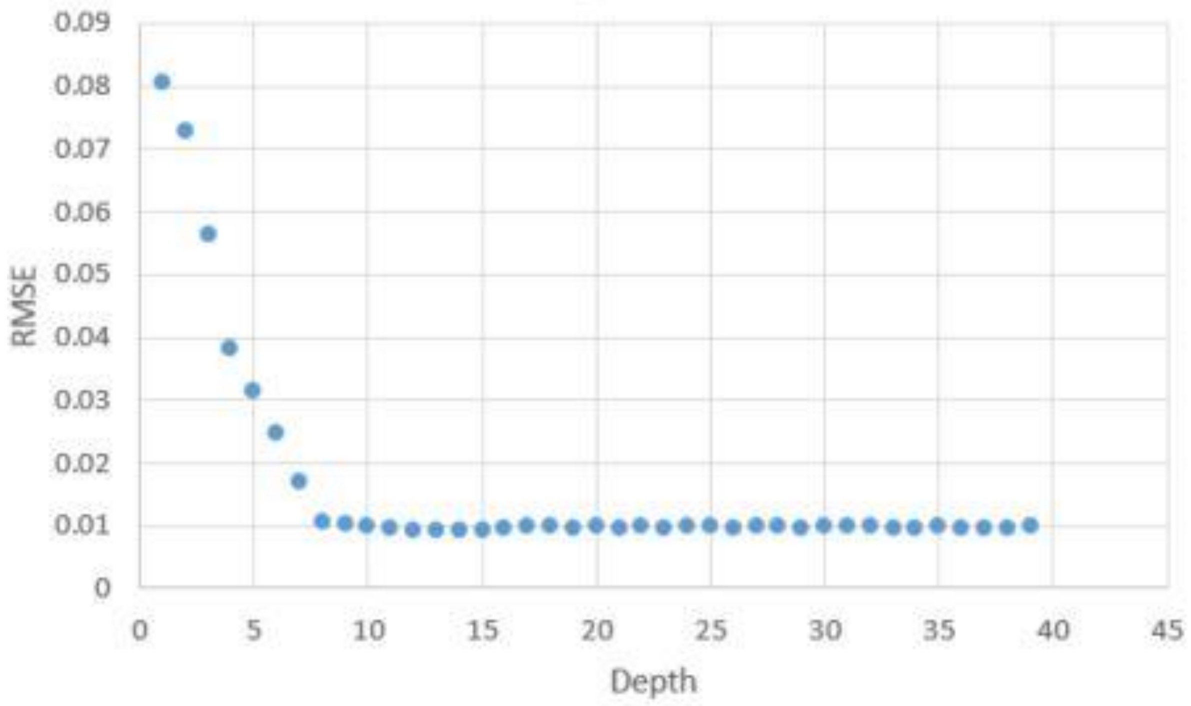


From these results we can see that linear regression model does not fit these data very well. RMSE for from all 10 tests are averaged 3.5, which is considerably big.

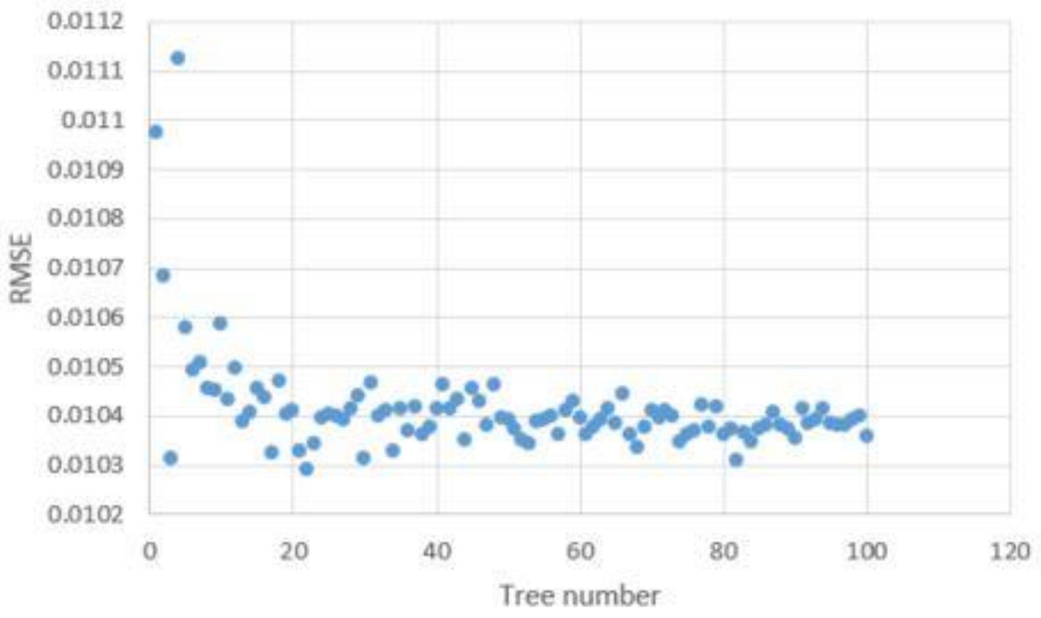
b. Random forest

In the random forest regression model, an ensemble of tree were created to reduce the variance of the predicted results. In our implementation, the sklearn package of Python were utilized to fit the data. The day and workflow parameter is transformed into integer before training the random forest. In the regression process, two important parameters, which are the depth of tree and the number of trees played very important roles. The depth of tree would reduce the bias of the prediction and the number of trees would reduce the variance of the prediction. The two consequences combined would reduce the RMSE.

In the process of tuning the parameter of the random forest, the maximum number of tree depth were optimized with fixed tree number of eight. The dependency of the RMSE on the maximum depth number were illustrated in the figure below. The normalized RMSE decreases with increasing maximum depth number and saturates around 0.01.

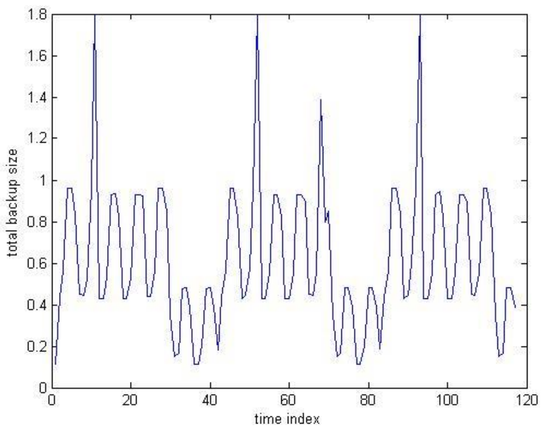


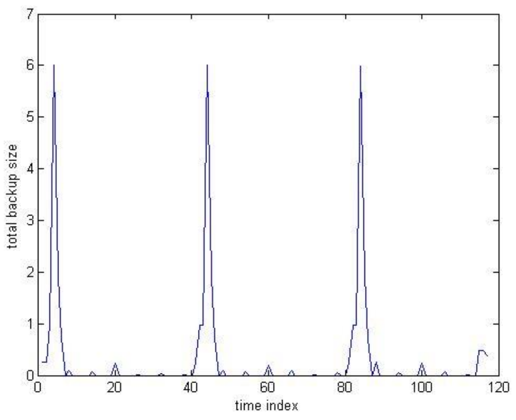
Similarly, the tree number was also optimized with same maximum depth number of nine. The relationship between the tree number and normalized RMSE were shown in the figure below. Different from the maximum depth number, the RMSE saturates more quickly around 0.0104

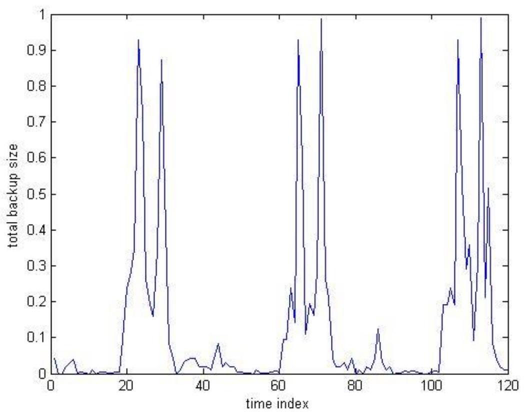


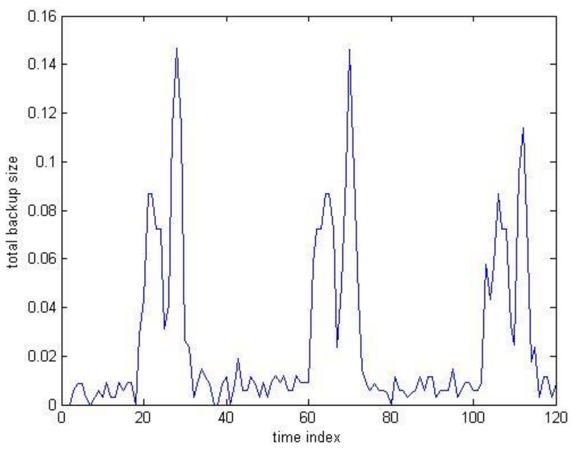
Additionally, the optimized non-normalized RMSE of the test data is 0.654443128253 which is lower than that of linear regression.

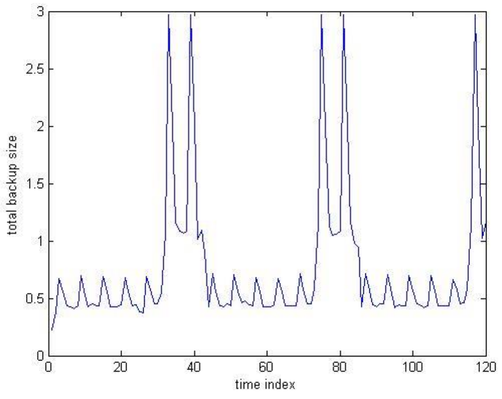
In our final prediction of the data, thirty tree were and depths of eight were utilized. The copy size of different workflow were shown in the figures below.





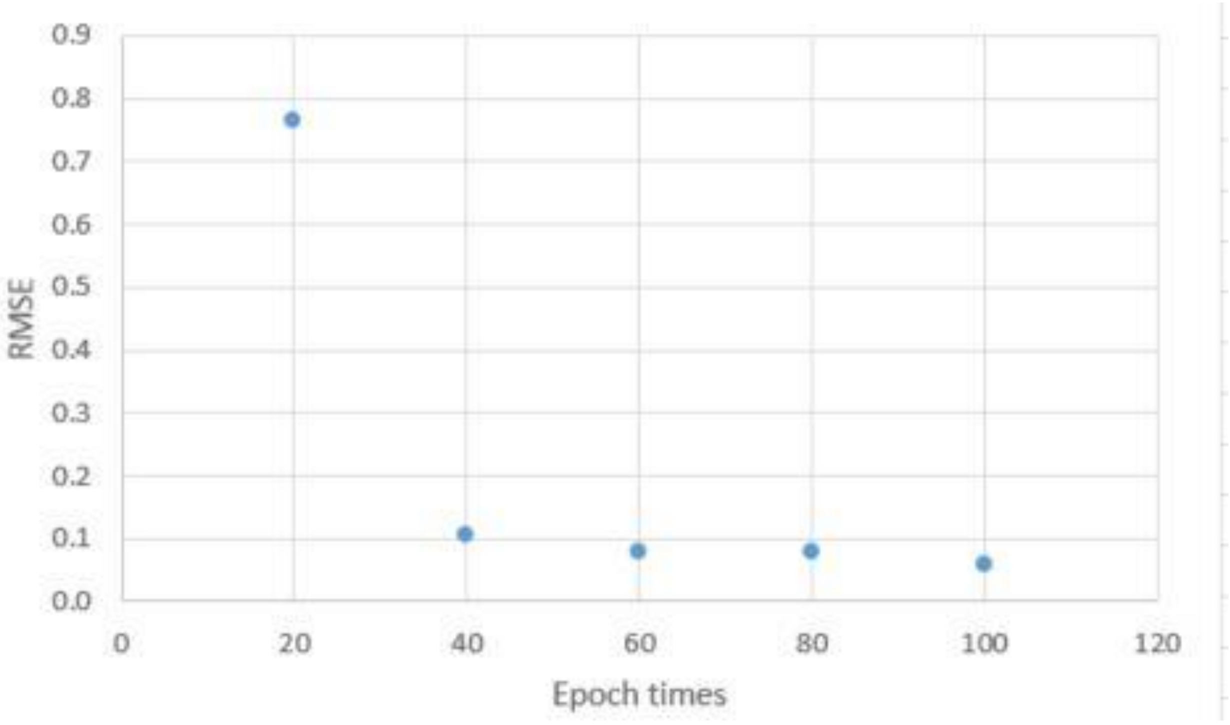






c. Neural network

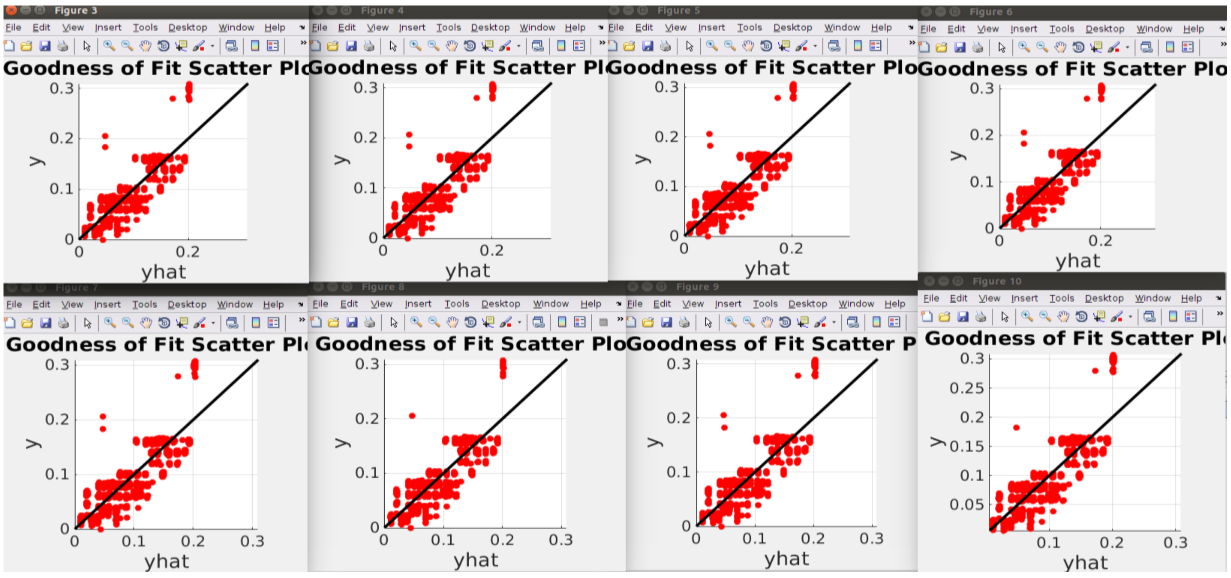
In the neural network regression model, three layer of neurons were used to estimate the back-up size function of the other input parameters. In our model, a neuron network with three layers of neurons were created and trained by the data set randomly picked from the original data set. The training data set accounts for 90% of the complete data set and the remaining 10% is used as estimated data. The three layers of neurons consists of five, three, and one neurons respectively. To obtain a best fitting result, epoch times, were optimized. The epoch-dependent RMSE were plotted in the figure below and it is obvious that RMSE greatly reduces with increasing epoch times and saturates around 0.1 after epoch times exceed 40.

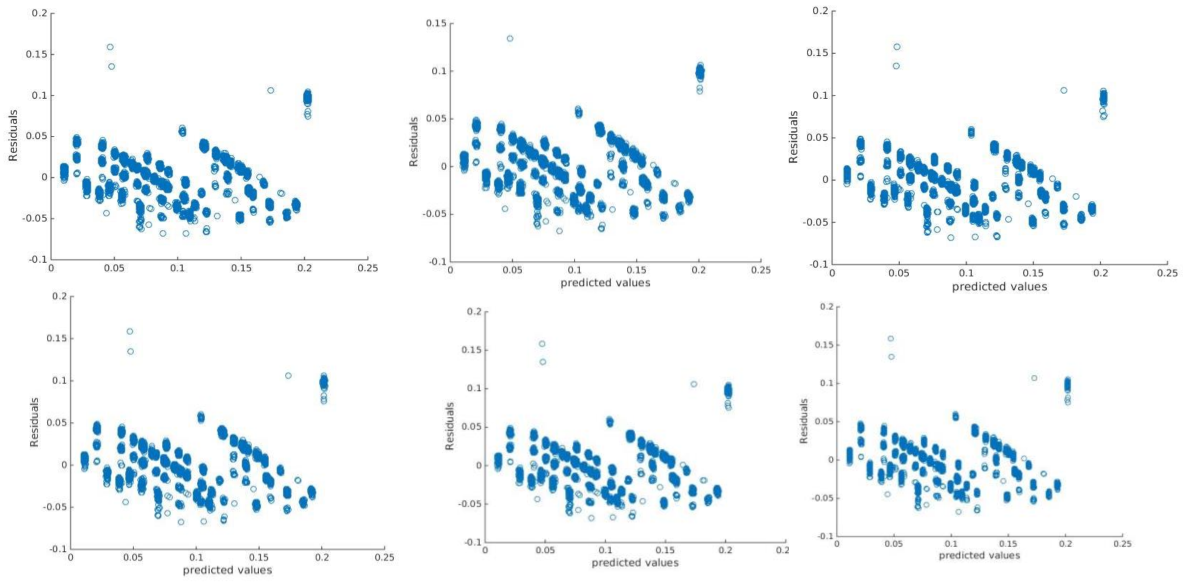


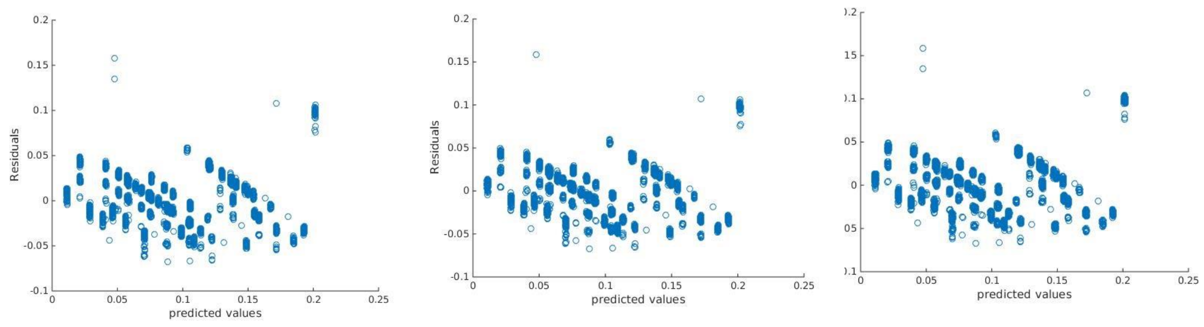
3. Separate the workflows

work_flow_0:

```
RMSE_train =

    1.6891    1.6973    1.6947    1.6955    1.7001    1.6990    1.6965    1.6829    1.7033    1.6773

>> RMSE_test

RMSE_test =

    0.5786    0.5542    0.5624    0.5601    0.5467    0.5491    0.5570    0.5973    0.5356    0.6122
```



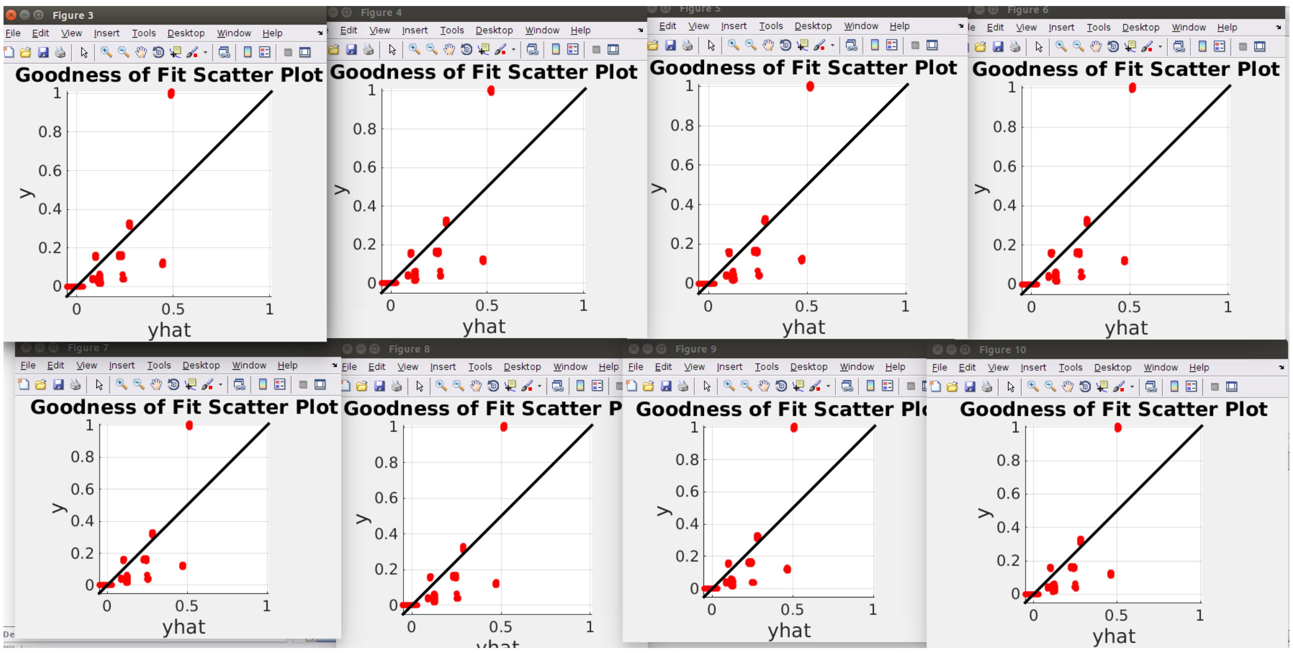




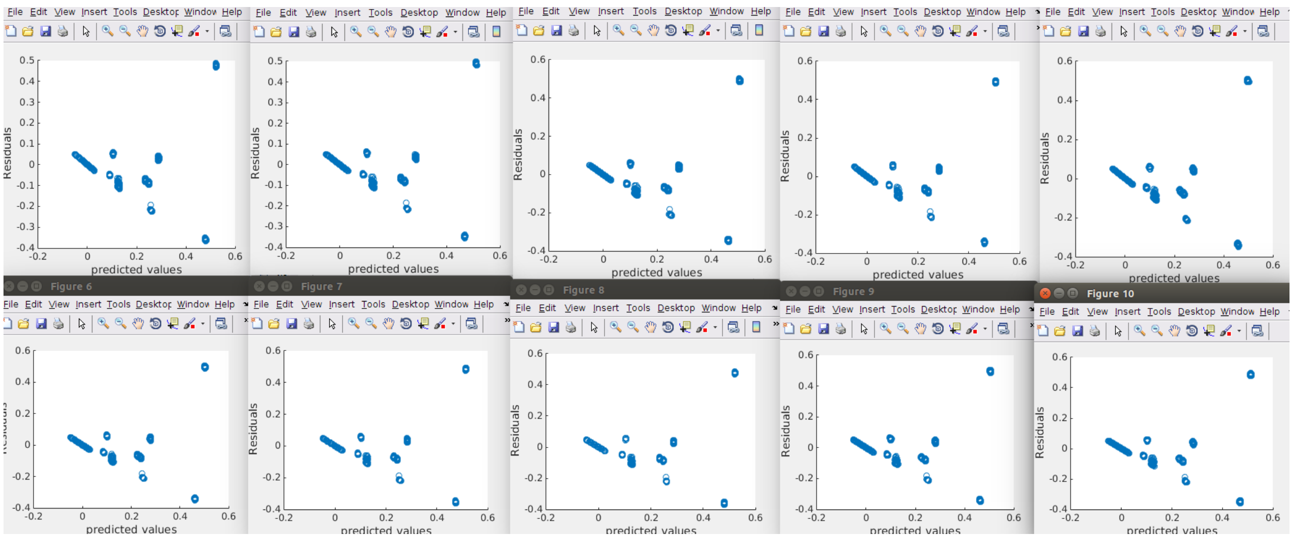
Compared the RMSE of workflow0 and all data, we can see that the fit is greatly improved.

work_flow_1:

```
>> RMSE_test

RMSE_test =

    2.0751    2.1531    2.1083    1.9576    1.8588    2.0593    1.9573    1.5994    1.9327    2.0069

RMSE_train =

    5.8617    5.8336    5.8588    5.9038    5.9337    5.8670    5.9017    6.0088    5.9096    5.8853
```



From the results of workflow1, we can see that the linear fit is better than all data, but still not as good as workflow1.

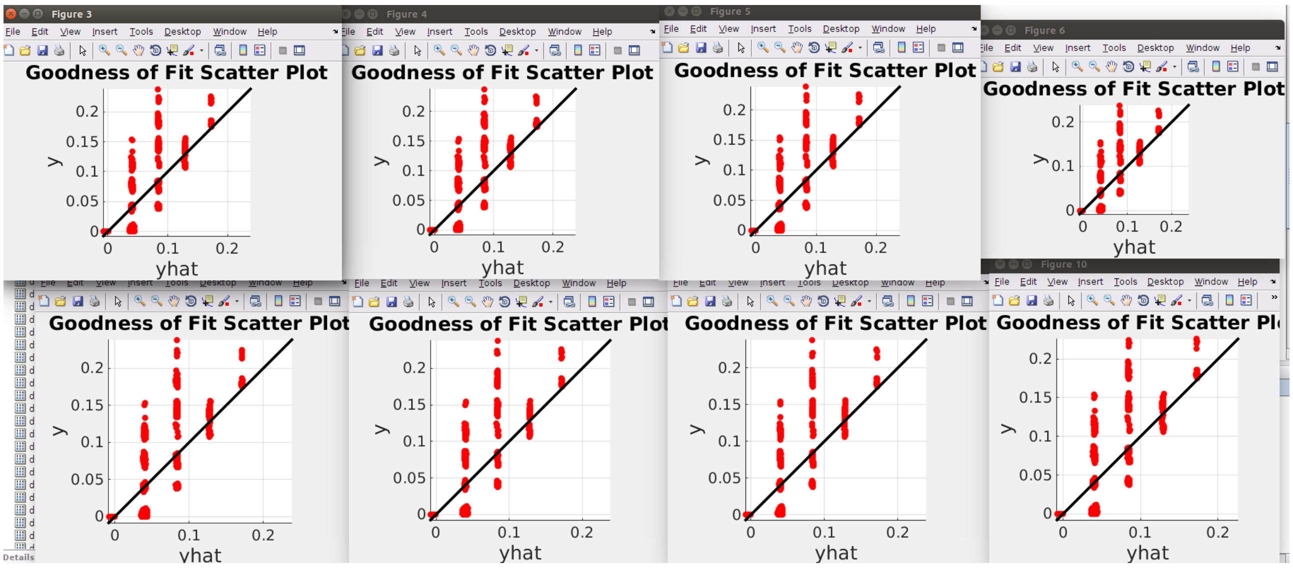


work_flow_2:

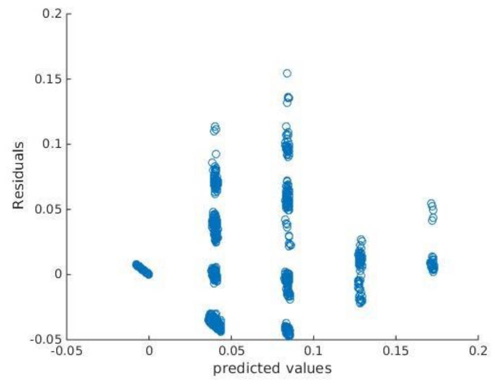
```
>> RMSE_test

RMSE_test =

    0.4716    0.4516    0.3808    0.4579    0.5458    0.5424    0.5438    0.5166    0.4868    0.5157
```

From the results we can see that there the fit is not perfect. The plot of residuals versus

Predicted values are as following:



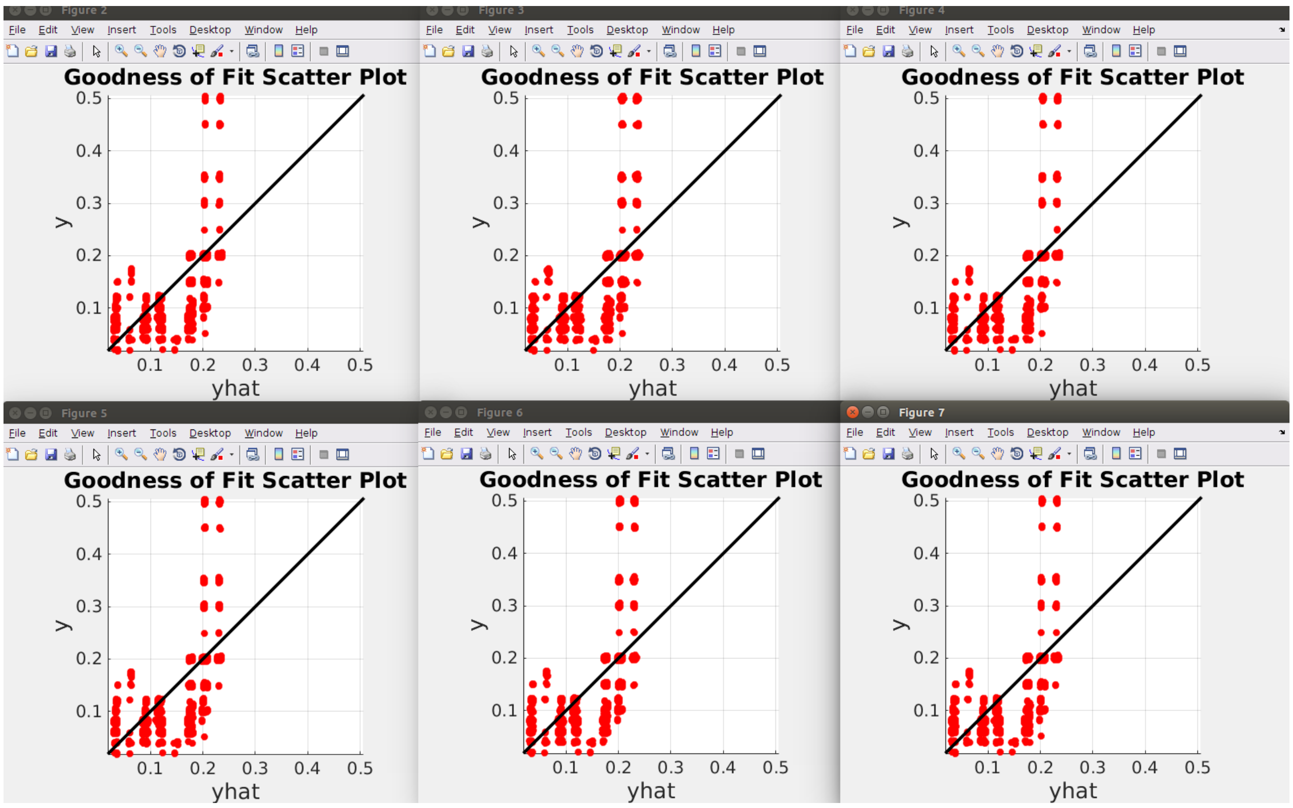
```
RMSE_train =

    1.4864    1.4926    1.5122    1.4908    1.4610    1.4622    1.4616    1.4714    1.4815    1.4717
```

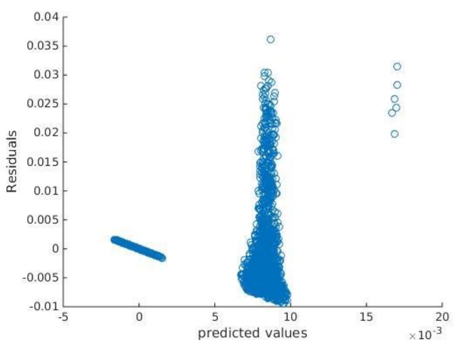


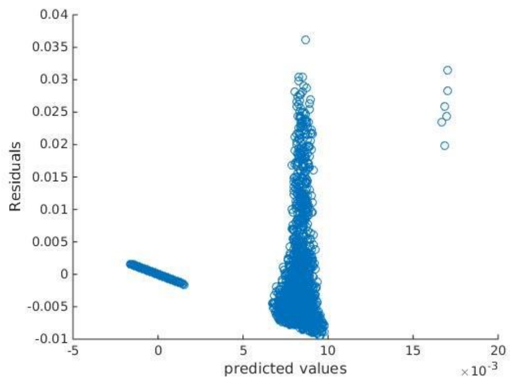
work_flow_3:

```
RMSE_test =

    0.1240    0.1047    0.1015    0.1205    0.1117    0.1128    0.1186    0.1275    0.1138    0.1112
```

```
>> RMSE_test

RMSE_test =

    0.4716    0.4516    0.3808    0.4579    0.5458    0.5424    0.5438    0.5166    0.4868    0.5157
```



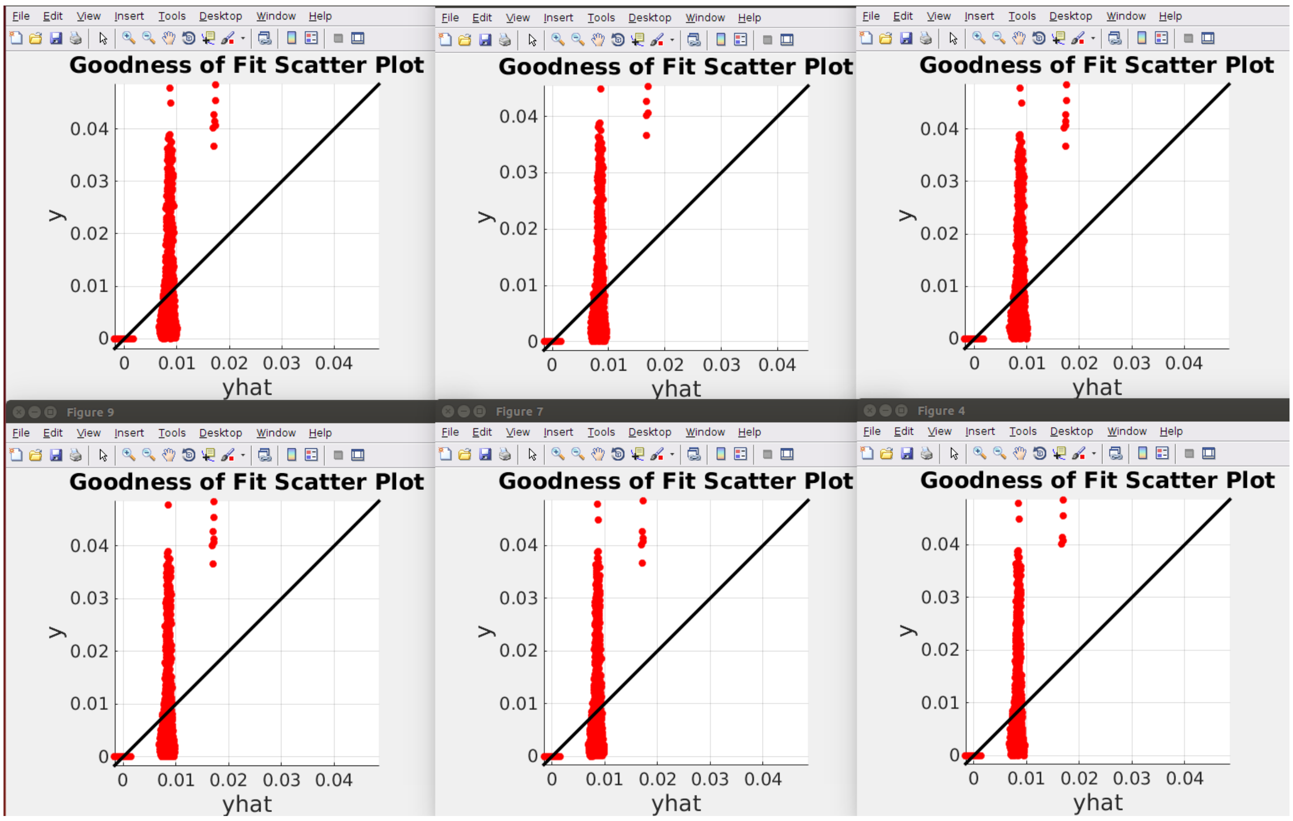




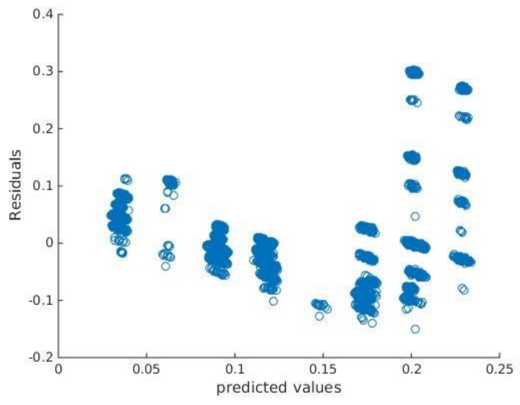
We can see that from the residuals plot that linear model doesn't fit work_flow_3 data very well either.

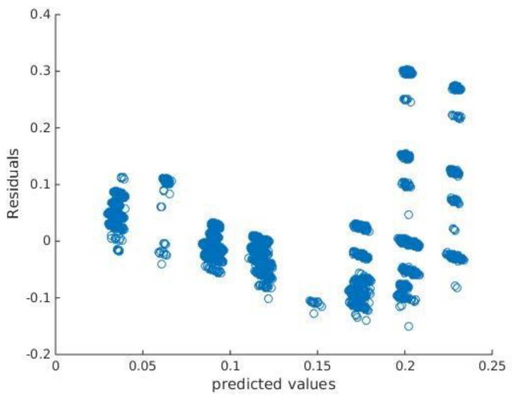
work_flow_4:

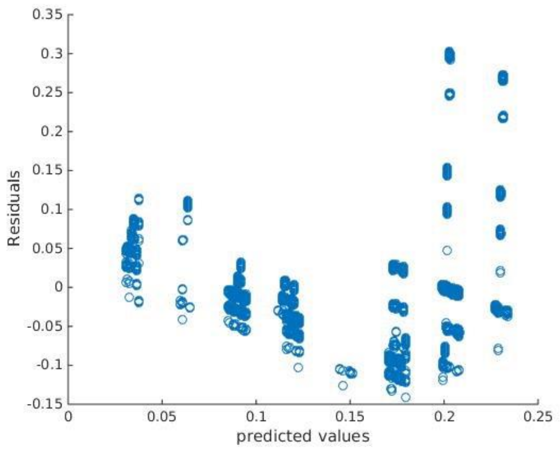
```
>> RMSE_test

RMSE_test =

    1.6342    1.5887    1.6821    1.6708    1.6319    1.6920    1.5038    1.6399    1.7084    1.6176

RMSE_train =

    4.9093    4.9241    4.8931    4.8971    4.9105    4.8899    4.9511    4.9073    4.8840    4.9151
```

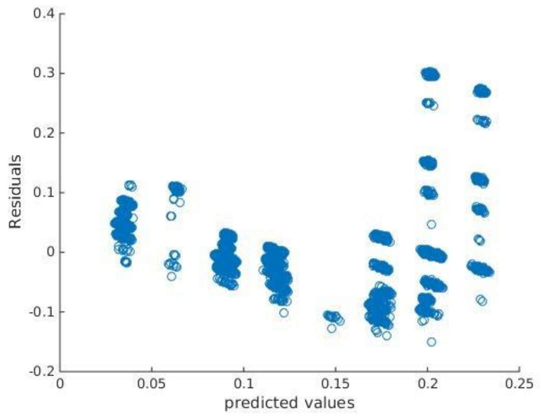


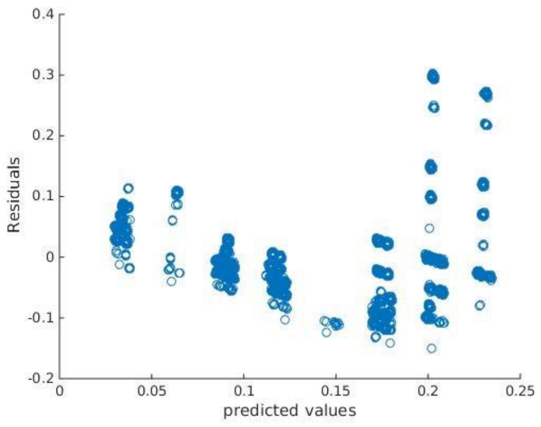
As we can see from the results of these five groups of dataset, every workflow has a different pattern. workflow1 agrees with linear model very well. But other groups don't fit linear model well. As a result, the whole dataset don't fit linear model very well. There may exist other higher order model that fits the data better. (For some of the results, I only list part of the plots out of the 10 plots since the 10 patterns out of cross validation are very similar)





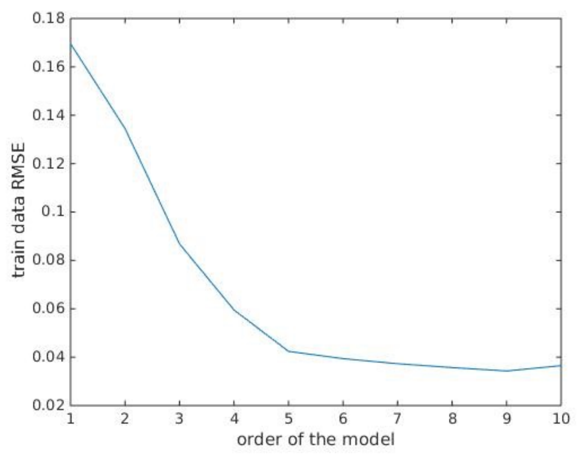


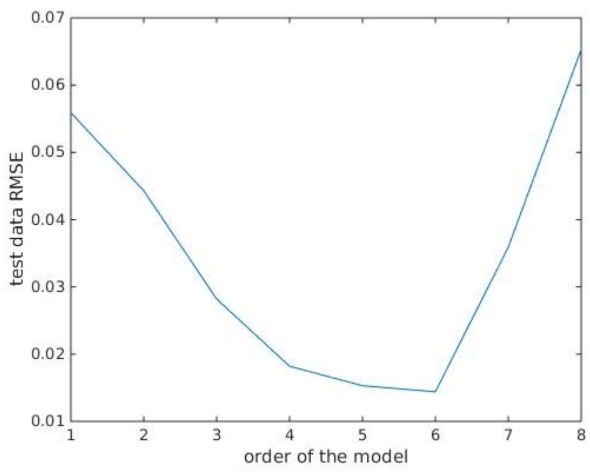




Now we increase the degree of the polynomial to improve fitting and use a 10 fold cross validation to evaluate the results.

The RMSE plot of the trained model against the degree of the polynomial is as following:





The threshold on the degree of the fitted polynomial is 6. When the degree of the polynomial goes higher than 6, the RMSE begins to increase quickly. Also comparing the residuals plot between order 1 to 10. We can find that order 6 has the best distribution, which also proves order 6 fits the best.

Cross validation helps controlling the complexity of the model since now we have both training data and test data. From training the model, sometimes we may overfit the model, but through cross validation we can solve this problem by finding the threshold of the degree of the fitted polynomial.

Hence the complexity of the model is controlled.

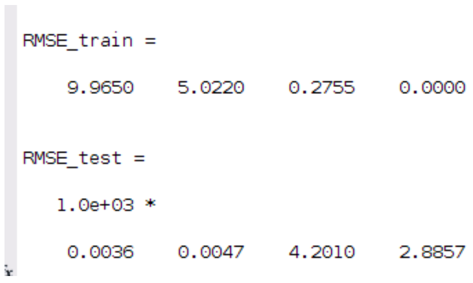
Boston Housing Dataset

4. Linear regression

(Note：Computations of RMSE are done without normalization (i.e., dividing N) which doesn`t affect the comparison and conclusion)

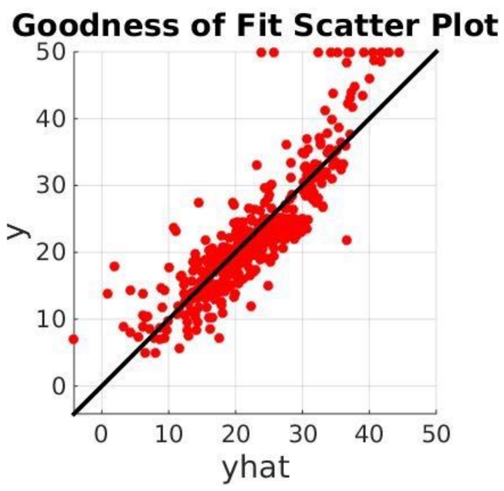
MEDV is the target variable and least square as the penalty function. After performing a 10 fold cross validation. We have the following results:

RMSE:

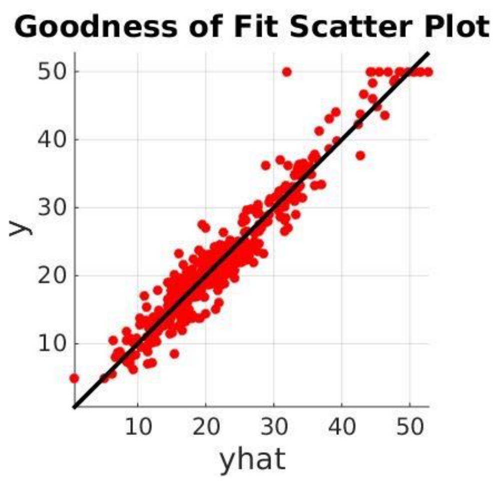


```
RMSE_train =

    9.9650    5.0220    0.2755    0.0000


RMSE_test =

   1.0e+03 *

    0.0036    0.0047    4.2010    2.8857
```

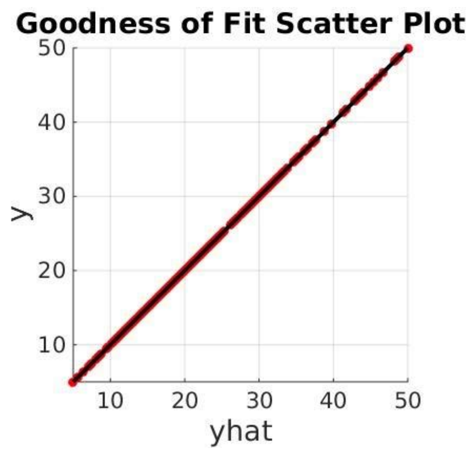
fitting plot(linear model):



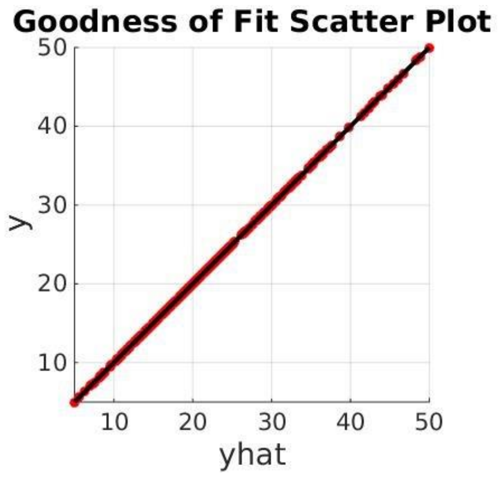
fitting plot(second_order_model):



fitting plot(third_order_model):

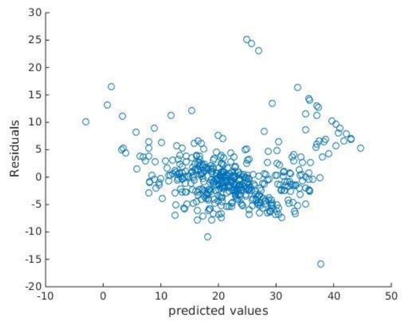


fitting plot(fourth_order_model):

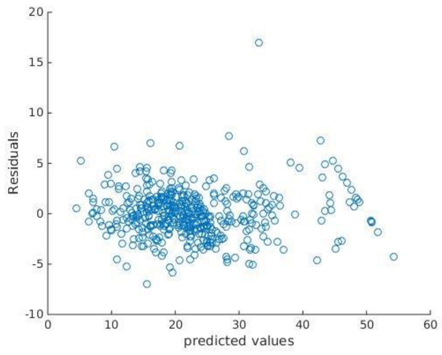


Although the RMSE for training data goes lower and lower as the order goes higher and higher, and the fitting plot of training data seems better and better, the RMSE for test data is increasing. And this means that the problem of overfitting becomes more and more obvious. The plot of residuals versus predicted values also indicate this.

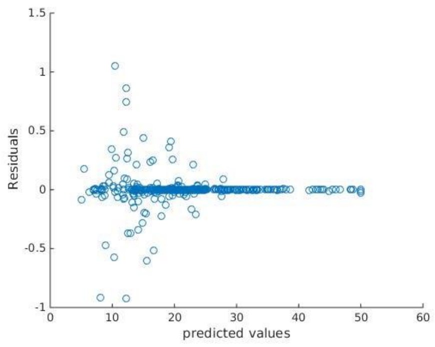
plot of residuals versus predicted values (linear model):



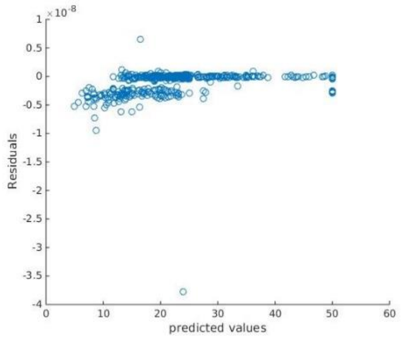
second order model:



third order model:



fourth order model:



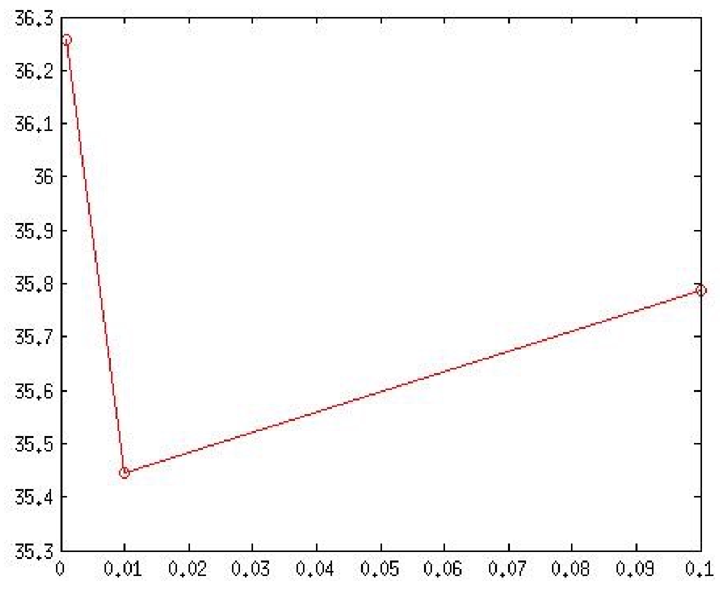
From the results RMSE is the lowest under linear model, we can see that linear model fits the data best from the plot of residuals versus predicted values.

5. Control fitting via regularization

(Note：Computations of RMSE are done without normalization (i.e., dividing N) which doesn`t affect the comparison and conclusion)

In this part, we try to control over fitting via regularization of the parameters.

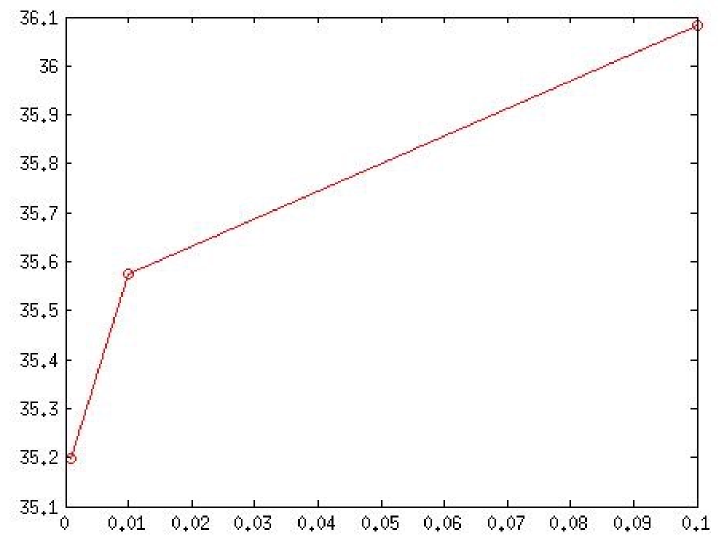
plot of RMSE versus alpha:



Compare alpha=0.1, 0.01,0.001, we can find that alpha=0.01 is the best since RMSE is the lowest then.

b) Repeat the previous part for Lasso regularization.

Plot of RMSE versus beta:



We can find that among {0.1,0.01,0.001}, under 0.001 RMSE is the minimum. Hence beta=0.001 is the best.

Note：all the computations of RMSE except in Problem 2.b and 2.c are done without normalization (i.e., dividing N)