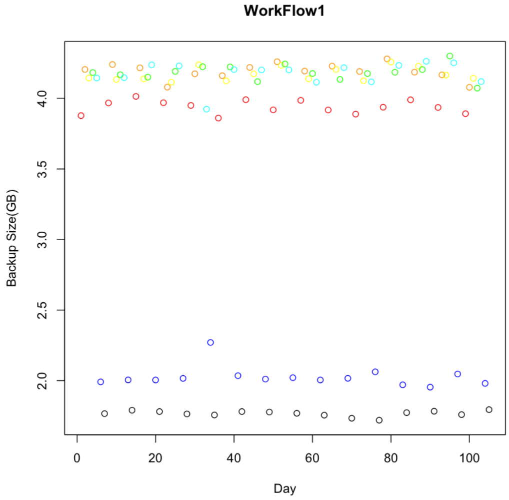
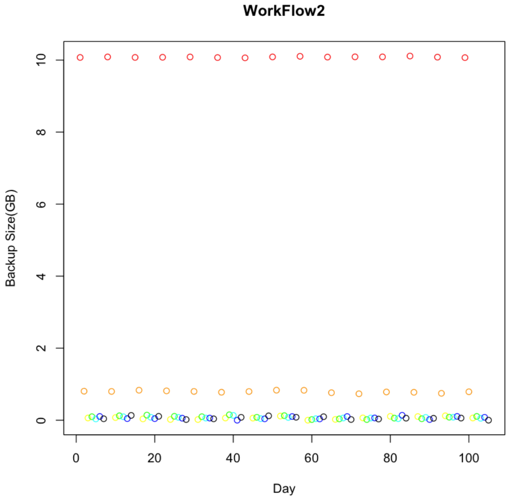
EE 219 Project 1

Network backup Dataset

Data Preprocessing & Observations

1. Transform the categorical features into numeric values
   a. `Day.of.Week` : Monday -> 1, Tuesday -> 2, ..., Sunday -> 7
   b. `Work.Flow.ID`: work_flow_0 -> 1, ..., work_flow_4 -> 5
   c. `File.Name`: File_0 -> 1, File_2 -> 2, ..., File_29 -> 30
2. By rearranging the data, we found that each workflow is in charge of 6 files
   a. workflow1: files 1, 2, 13, 24, 25, 26
   b. workflow2: files 3, 4, 27, 28, 29, 30
   c. workflow3: files 5, 6, 7, 8, 9, 10
   d. workflow4: files 11, 12, 14, 15, 16, 17
   e. workflow5: files 18, 19, 20, 21, 22, 23
3. By plotting the data in terms of day and total backup size for each workflow as below,

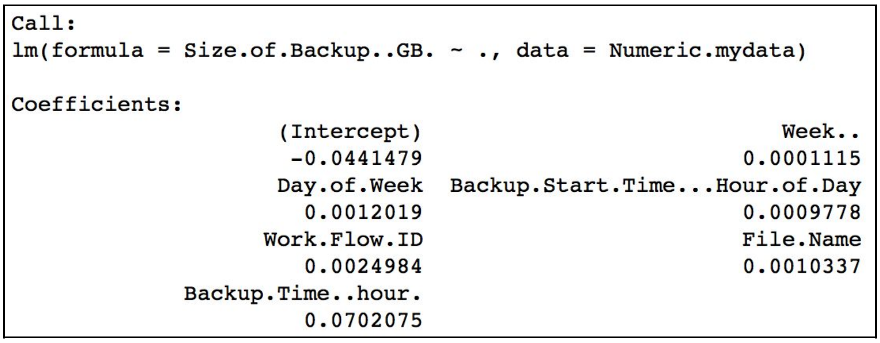




   we can see that for workflow1, backup sizes on weekend are usually small(blue indicates Saturday, black indicates Sunday); for workflow2, backup sizes on Monday (red indicates Monday) are much larger than other days Within each week, the backup sizes decrease greatly.
4. By plotting the data in terms of hour and total backup size for each workflow as below, we can see that the overall backup size within each hour is decreasing.

Modeling and Prediction

1. **Linear Regression**: by using a simple linear regression algorithm without cross validation on the whole dataset predicting backup sizes, we obtain the following statistics and RMSE = 0.0790403. The statistics show that the attribute `Backup.Time..hour.` is the most significant since it has the largest coefficient. This is actually intuitive that larger size requires more time to backup. The second important attribute is `Work.Flow.ID` since every workflow is responsible for different files as shown in the figures above in observation.

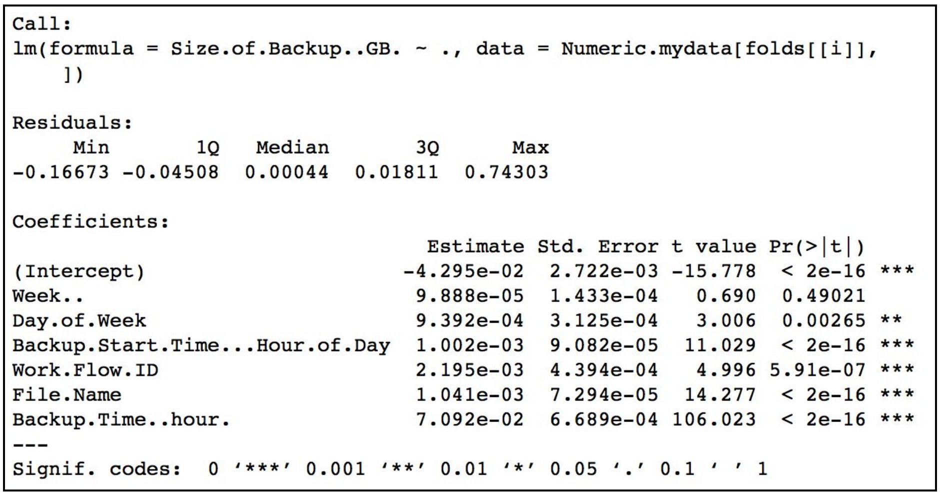


```
Call:
lm(formula = Size.of.Backup..GB. ~ ., data = Numeric.mydata)

Coefficients:
                (Intercept)                            Week..
                 -0.0441479                         0.0001115
                Day.of.Week  Backup.Start.Time...Hour.of.Day
                  0.0012019                         0.0009778
               Work.Flow.ID                         File.Name
                  0.0024984                         0.0010337
         Backup.Time..hour.
                  0.0702075
```

To prevent overfitting, 10-fold cross validation is performed. We obtain the validation RMSE of each fold as follows:

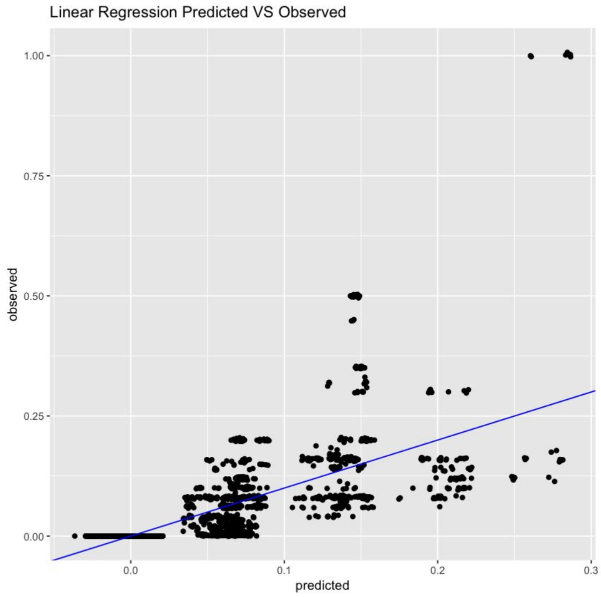
```
 1. 0.0896429066059419
 2. 0.0868426863599083
 3. 0.0818615399530692
 4. 0.0822183120176312
 5. 0.0725316289182609
 6. 0.0852231793215002
 7. 0.0683279113201512
 8. 0.0690066601510758
 9. 0.0777150419290645
10.  0.0744410181464035
```

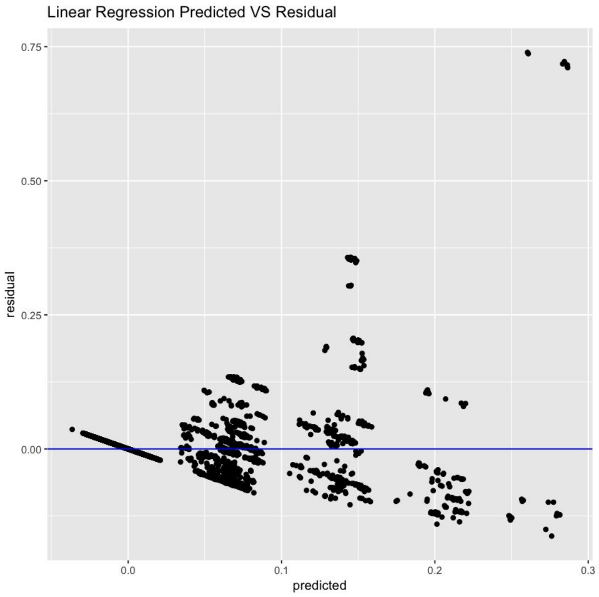
The smallest RMSE is 0.0683279. As we can see in the statistics, `Backup.Time..hour.` still has the largest weight among all others. From the significant codes we know that if the p-value of an attribute is smaller than 0.05, then it's considered statistically significant. `Week..` is the number of week, which has the greatest p-value, so it's the least significant.



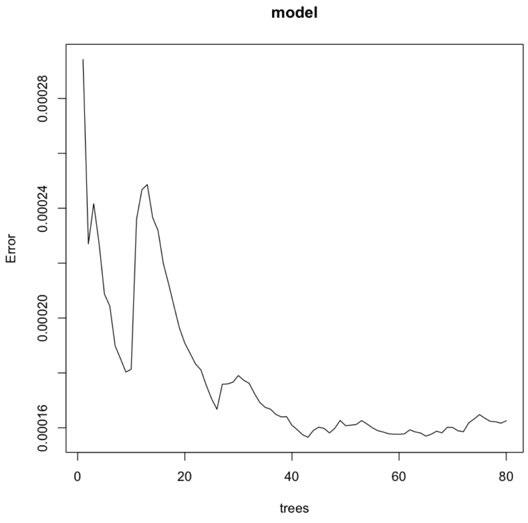
```
Call:
lm(formula = Size.of.Backup..GB. ~ ., data = Numeric.mydata[folds[[i]],
    ])

Residuals:
     Min       1Q   Median       3Q      Max 
-0.16673 -0.04508  0.00044  0.01811  0.74303 

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)                     -4.295e-02  2.722e-03 -15.778  < 2e-16 ***
Week..                           9.888e-05  1.433e-04   0.690  0.49021    
Day.of.Week                      9.392e-04  3.125e-04   3.006  0.00265 ** 
Backup.Start.Time...Hour.of.Day  1.002e-03  9.082e-05  11.029  < 2e-16 ***
Work.Flow.ID                     2.195e-03  4.394e-04   4.996 5.91e-07 ***
File.Name                        1.041e-03  7.294e-05  14.277  < 2e-16 ***
Backup.Time..hour.               7.092e-02  6.689e-04 106.023  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

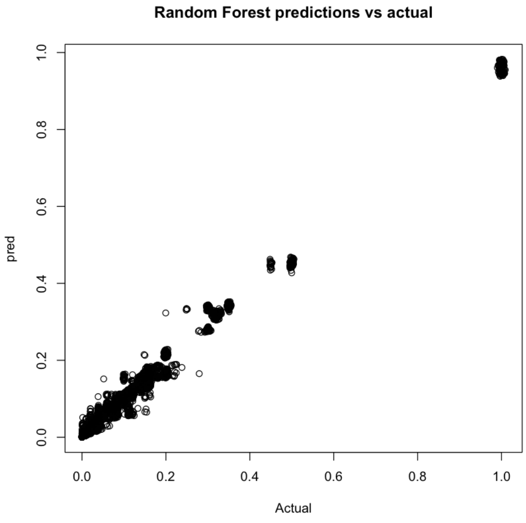
By applying the model with the best validation RMSE on the test data, we can plot the following figures. It’s obvious that the model somehow captures the trend of the training data, but it’s not very accurate, showing that linear regression might not be enough to fit the data.



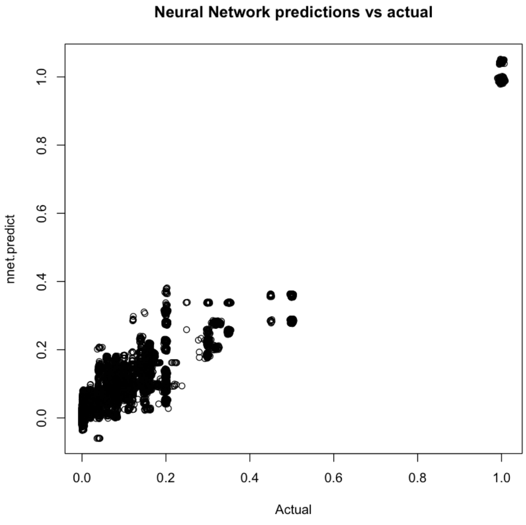


2. **Random Forest Regression**: Random forest takes samples multiple times and build several tree models for ensembling, so we need to find a rather stable model. With the number of trees set to 40 and 80, random forest gives a more stable result with RMSE's of 0.0111011 and 0.0110535 respectively. From the error plot in the left, we can see that the error decreases and becomes more stable approximately after 20 trees. The right figure shows the relationship between prediction and ground truths, which almost end up linear. After performing cross validation on random forest, 40 and 80 trees still remain more robust than other settings.

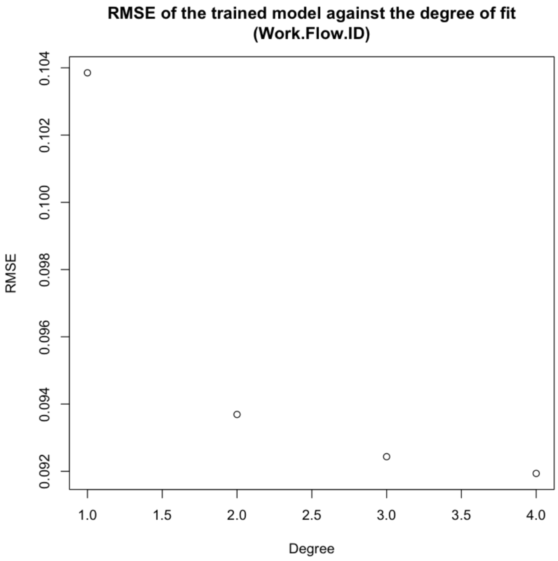


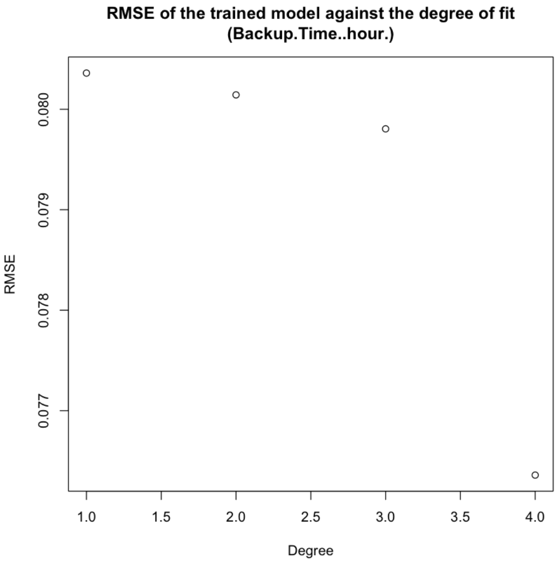


3. **Neural Network Regression**: The main parameter in our implementation is the number of estimators in the hidden layer. The estimators take inputs and evaluate the weights of each, and pass down the weighted effects. We started from 5 estimators to 100 with a step of 5, and when there are 10 estimators, the RMSE is reduced to 0.0371378, while other settings result in RMSE over 0.04.



4. **Polynomial Fitting**: We first divide the data into 80% of training set and 20% of testing set, and we train the model on each attribute separately with different degrees. We choose the two attributes that have large coefficients from linear regression and do polynomial fitting. When fitting `Work.Flow.ID`, RMSE has an abrupt drop at degree of 2; when fitting `Backup.Time..hour.`, RMSE drops at degree of 4.





Now we perform 10-fold cross validation on fitting `Backup.Time..hour.` since it gives better RMSE. The RMSE's of each fold is as follows:

1. 0.0849477755333286
2. 0.0827696126511417
3. 0.0769168008064376
4. 0.0770522545752197
5. 0.071290816669811
6. 0.0815652709062936
7. 0.0687224156184904
8. 0.0684046716755573
9. 0.0744483908190771
10. 0.0720844037073897

The smallest RMSE is 0.0684046, which is pretty close to linear regression after cross validation. We also tested on fitting `Work.Flow.ID` and `File.Name`, but the results are of no value.

In general, doing training without cross validation can lead to a low training error because it predicts what it learns from. However, this might result in high test error, or generalization error, because the model fits the training data too well that it captures the details of the training data instead of the overall trend. In this case, it usually might not perform well when encountering new data. Cross validation takes away a part of the training data as validation data, making the model more general. So if the model can achieve low validation error, it is more likely to perform better for predicting new data. In other word, cross validation prevents the model from becoming too complex, but in a degree that learns well from training data and perform well on new data.

5. Now we are predicting the backup sizes for each workflow. Since the workflow ID becomes trivial, we drop the workflow columns.

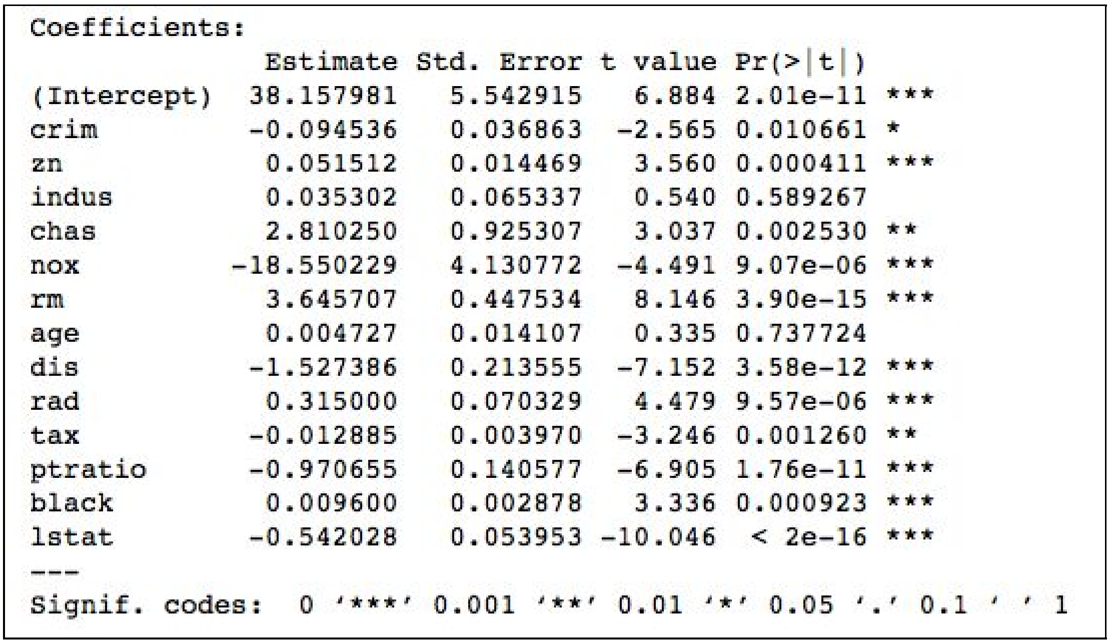
Linear Regression	workflow1	workflow2	workflow3	workflow4	workflow5
Training RMSE	0.0294035	0.1036223	0.0254950	0.0058991	0.0841548
Validation RMSE (min)	0.0278761	0.0936496	0.0227952	0.0052634	0.0781235
Generalization RMSE	0.0298813	0.0987535	0.0281457	0.0056347	0.0862810

From the results above, we can see that workflow1, workflow3, and workflow5 have training RMSE’s less than generalization RMSE’s, which means that the original model directly trained from training data might be slightly overfitting. For workflow2 and workflow4, the training RMSE’s are greater than generalization RMSE’s, which mean that the original model is not complex enough, or underfitting, and through cross validation, the complexity is improved and the model is more likely to fit unseen data better.

Boston Housing Dataset

4. **Linear Regression**

First, we try to fit the data with a linear regression model with “medv” as the target variable and all other attributes as the features. By performing a 10 fold cross validation, we could obtain the statistics of the trained model as shown below.



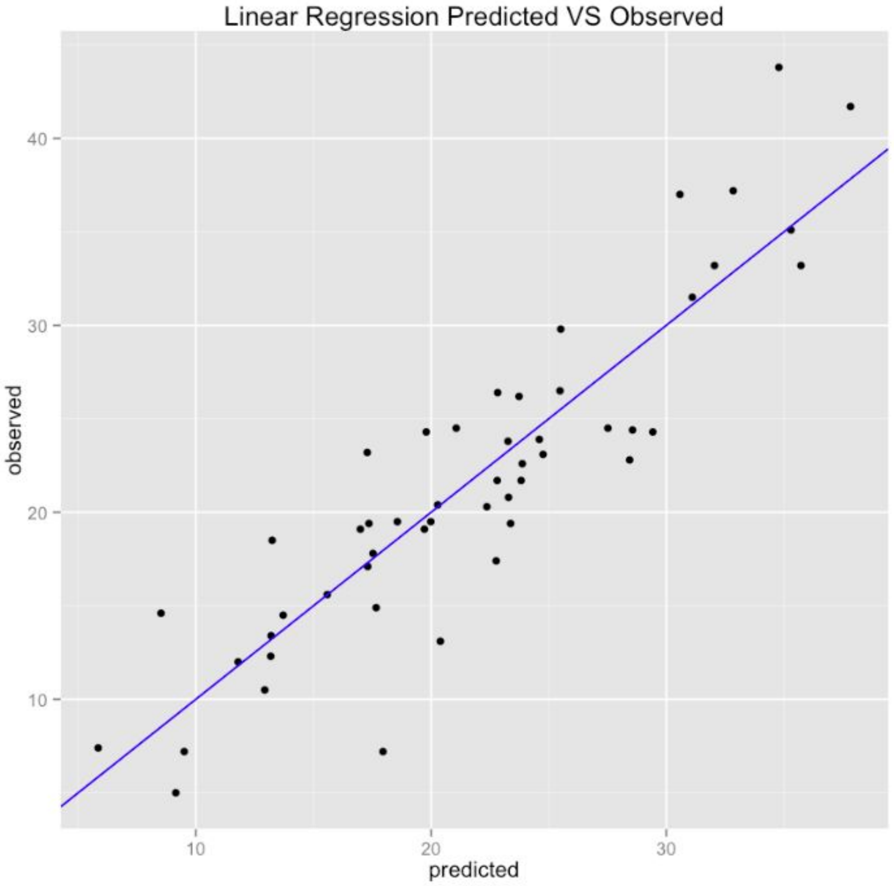
```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  38.157981   5.542915   6.884 2.01e-11 ***
crim         -0.094536   0.036863  -2.565 0.010661 *
zn            0.051512   0.014469   3.560 0.000411 ***
indus         0.035302   0.065337   0.540 0.589267
chas          2.810250   0.925307   3.037 0.002530 **
nox         -18.550229   4.130772  -4.491 9.07e-06 ***
rm            3.645707   0.447534   8.146 3.90e-15 ***
age           0.004727   0.014107   0.335 0.737724
dis          -1.527386   0.213555  -7.152 3.58e-12 ***
rad           0.315000   0.070329   4.479 9.57e-06 ***
tax          -0.012885   0.003970  -3.246 0.001260 **
ptratio      -0.970655   0.140577  -6.905 1.76e-11 ***
black         0.009600   0.002878   3.336 0.000923 ***
lstat        -0.542028   0.053953 -10.046  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the p-value of the statistics, we can see that most of the p-values are significant indicating strong relationship of these variables with the outcome. However, the p-values for “indus” and “age” is larger than 0.05, which indicates that they are statistically insignificant.

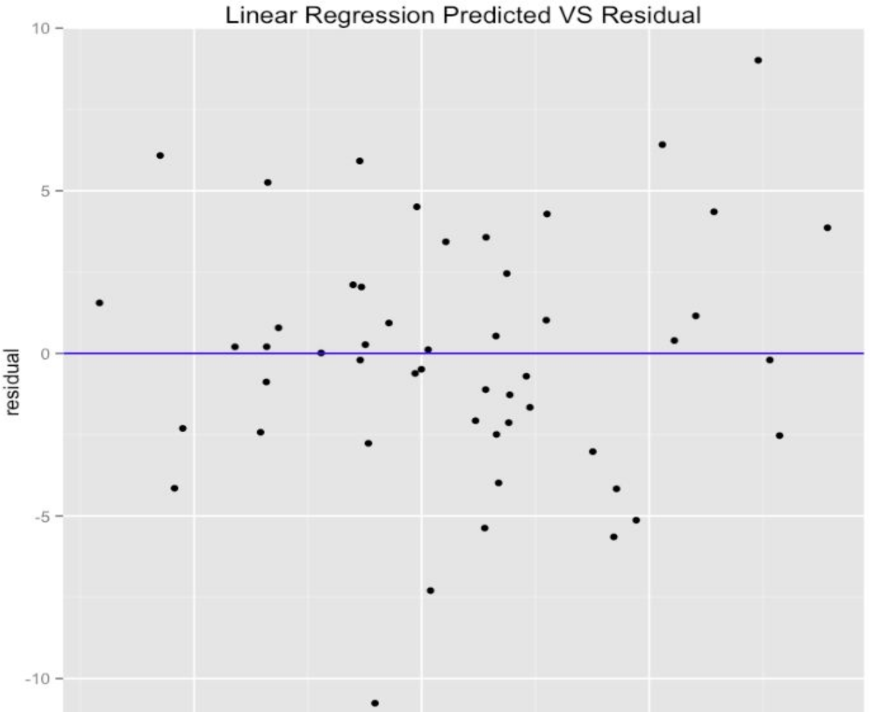
We computed the root mean squared error for each fold, and take an average over all the errors to get the averaged RMSE = 4.84186

Plot the “Fitted values and actual values scattered plot over time”.

From the visualizations below, the plots in the graph appears to fit linearly. If we sketch a straight line among the data plots, we can see that most of the plots are close to this line. Thus, this observation shows that this linear regression model fits the data pretty well.

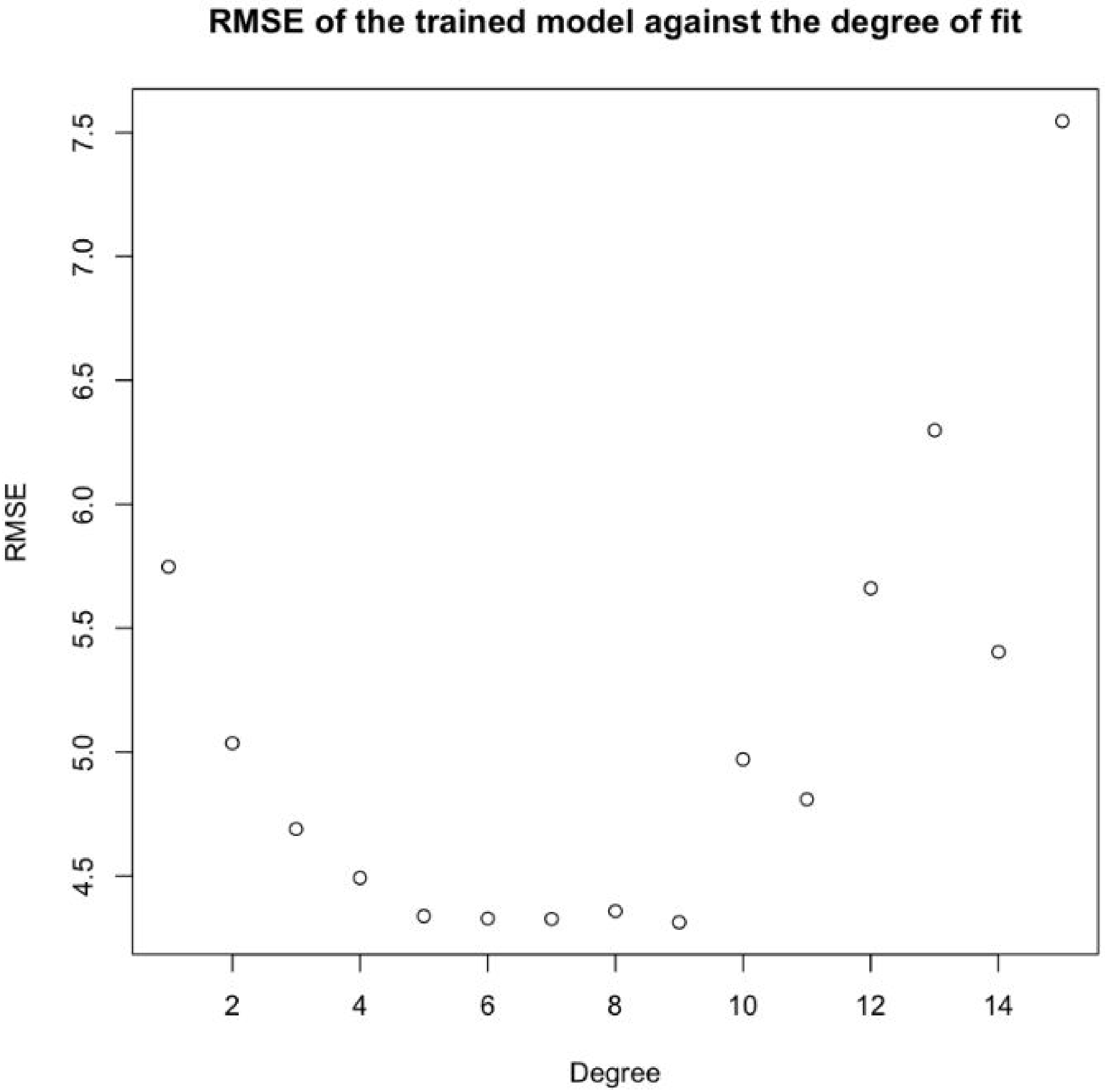


Residuals versus fitted values plot

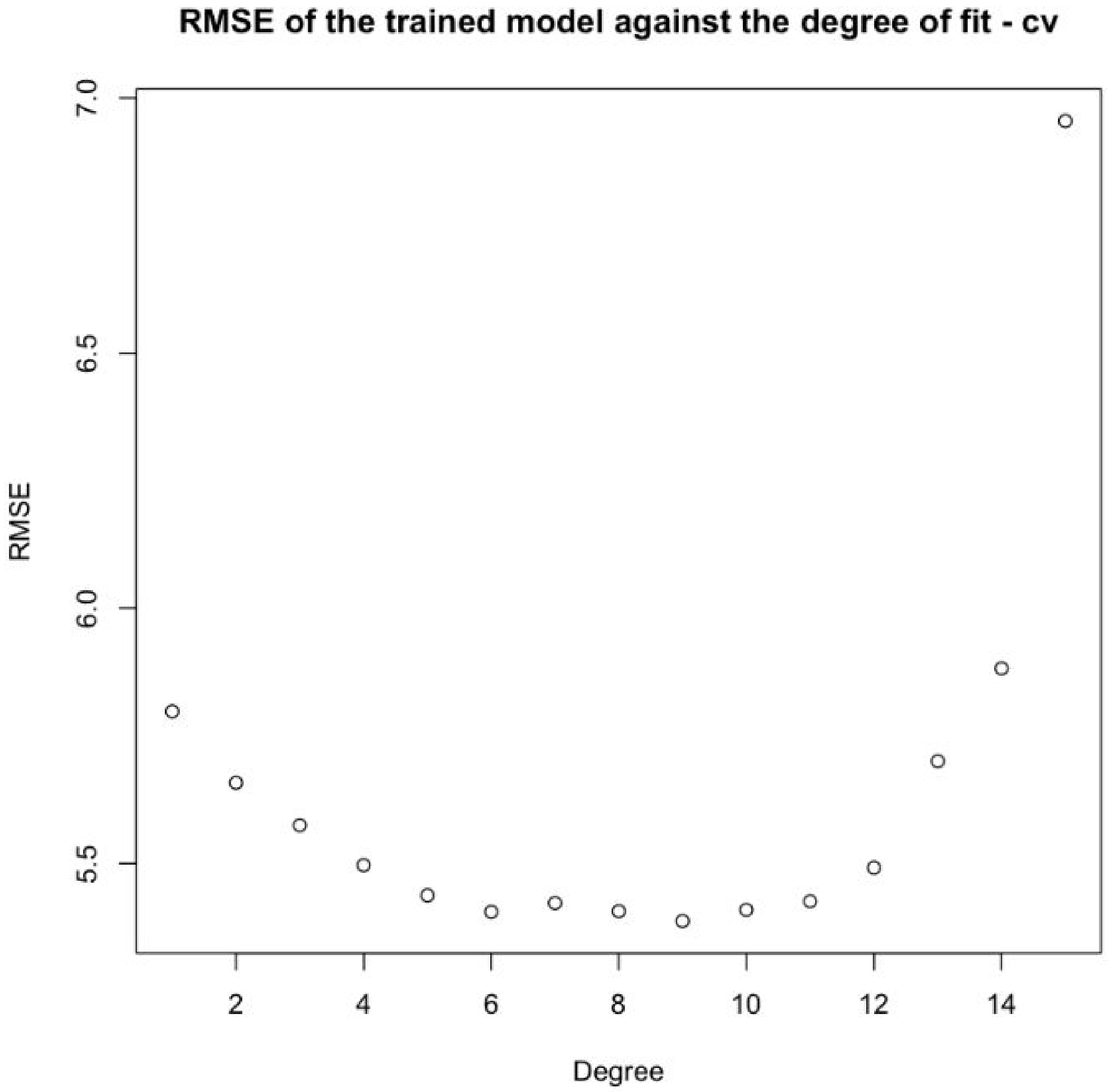


Repeat the same steps for a polynomial regression function and find the optimal degree of fit

(1) We first split the data into 80/20 with 80% being the training set and 20% being the testing set. For this fixed set of training and testing data, the threshold for the degree of polynomial fitted is 9. The generalization error gets worse significantly starting degree = 10 as shown in the graph.



(2) When performing 10 fold cross validation on the original dataset, the threshold on the degree of the fitted polynomial is also around degree = 9.



5.

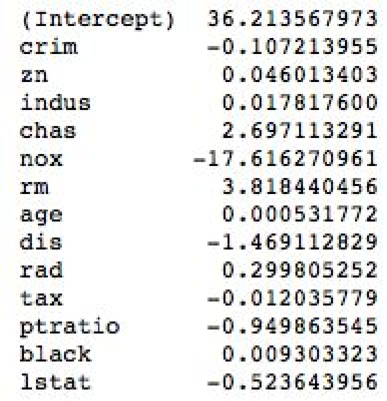
a) Ridge Regression

By tuning the complexity parameter α in the range of {1. 0.1, 0.01, 0.001}, the best RMSE obtained via 10-fold cross validation is 4.839102. The complexity parameter α for this model is 0.01.

b) Lasso Regression

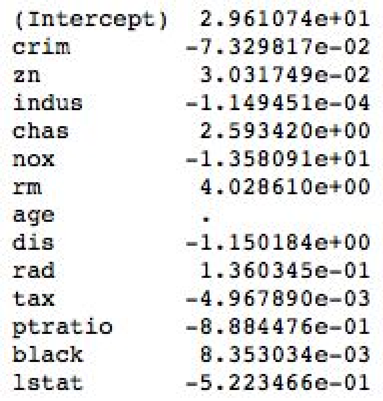
The best RMSE obtained via 10-fold cross validation is 4.829873. The complexity parameter α for this model is 0.9. We are tuning α in the range of {0.5, 1}.

By compare the value of the optimal coefficients obtained via regularization with the un-regularized linear regression model, we can see that the positive coefficients are positive across all the models, this is true for the negative coefficients as well.



```
(Intercept)  36.213567973
crim         -0.107213955
zn            0.046013403
indus         0.017817600
chas          2.697113291
nox         -17.616270961
rm            3.818440456
age           0.000531772
dis          -1.469112829
rad           0.299805252
tax          -0.012035779
ptratio      -0.949863545
black         0.009303323
lstat        -0.523643956
```

Ridge Regression



```
(Intercept)  2.961074e+01
crim        -7.329817e-02
zn           3.031749e-02
indus       -1.149451e-04
chas         2.593420e+00
nox         -1.358091e+01
rm           4.028610e+00
age          .
dis         -1.150184e+00
rad          1.360345e-01
tax         -4.967890e-03
ptratio     -8.884476e-01
black        8.353034e-03
lstat       -5.223466e-01
```

Lasso Regularization