EE219 PROJECT 1 REPORT

REGRESSION ANALYSIS

Prepared by-
Twinkle Gupta : 804740325
Shikhar Malhotra : 504741656
Omkar Patil : 904760474

Regression analysis is statistical procedure that aims at finding the relationship between a target variable and various relevant parameters. Various regression models can be used for this purpose. In this project, we experiment with linear regression, random forests, neural networks and polynomial regression on two data sets namely 'Network Backup Dataset' and 'Boston Housing Dataset'.

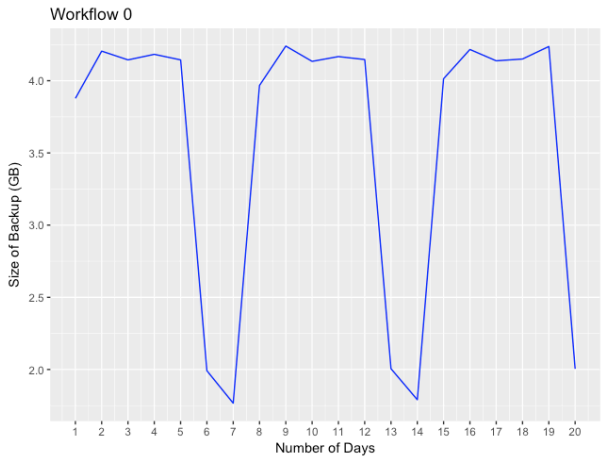
PART 1

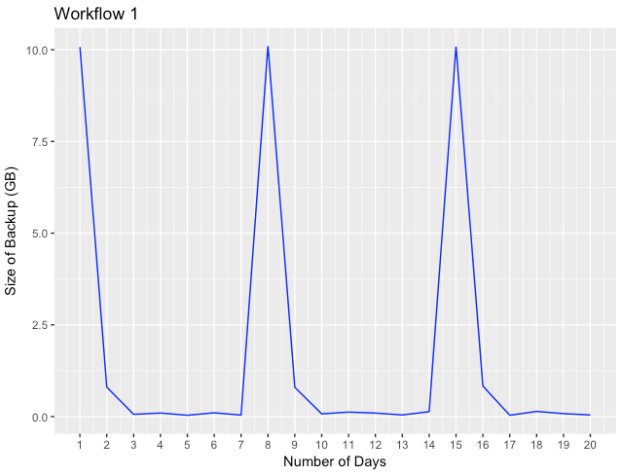
Dataset used : Network Backup Dataset.
Description of dataset :
The dataset is comprised of simulated traffic data on a backup system in a network. The dataset has around 18000 data points with the following columns:

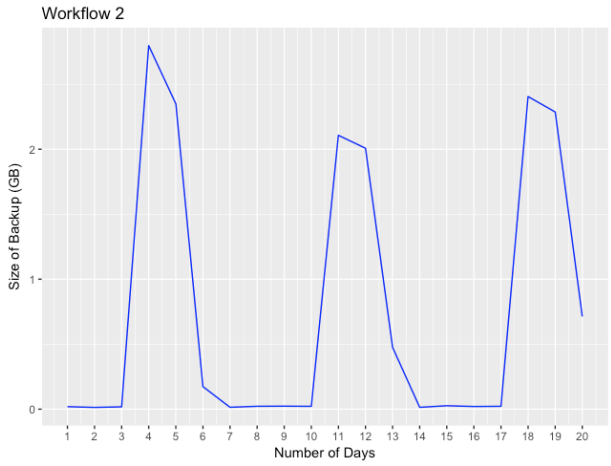
- Week index
- Day of the week at which the file back up has started
- Backup start time-Hour of the day: the exact time that the backup process is complete
- Workflow ID
- File name
- Backup size: the size of the file that is backed up in that cycle in GB
- Backup time: the duration of the backup procedure in hour

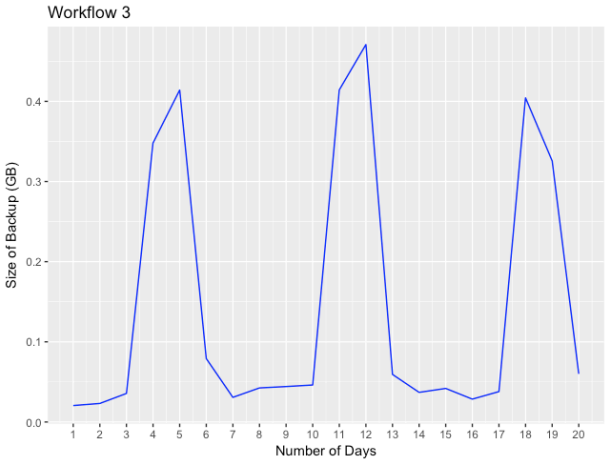
Problem 1

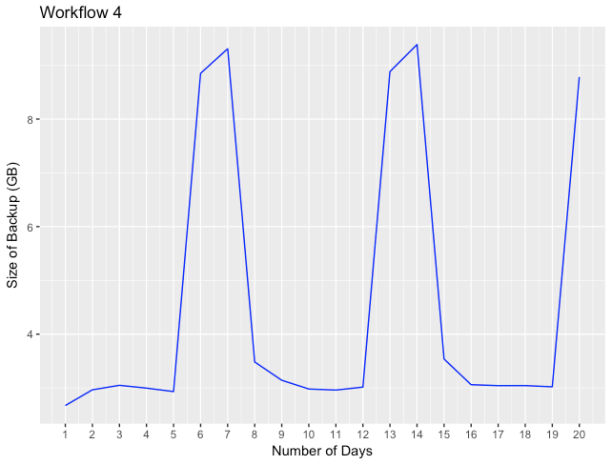
To get insight into the data, we first plot total backup size with time for the first 20 days for all the workflows. These plots are shown below.


Workflow 0
Size of Backup (GB)
Number of Days

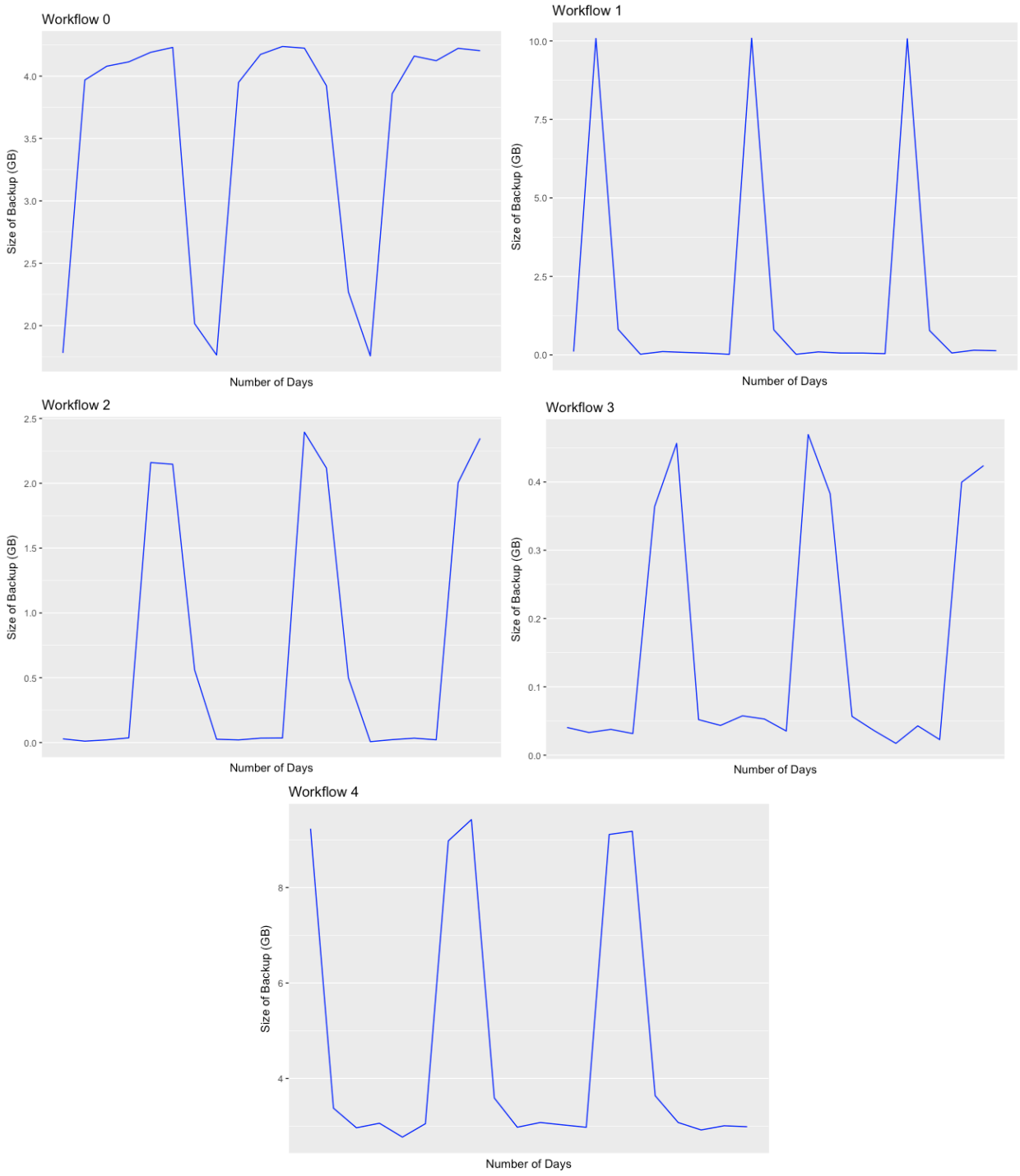

Workflow 1
Size of Backup (GB)
Number of Days


Workflow 2
Size of Backup (GB)
Number of Days


Workflow 3
Size of Backup (GB)
Number of Days


Workflow 4
Size of Backup (GB)
Number of Days

We plotted similar graphs for the next 20 days and obtained the following graphs –



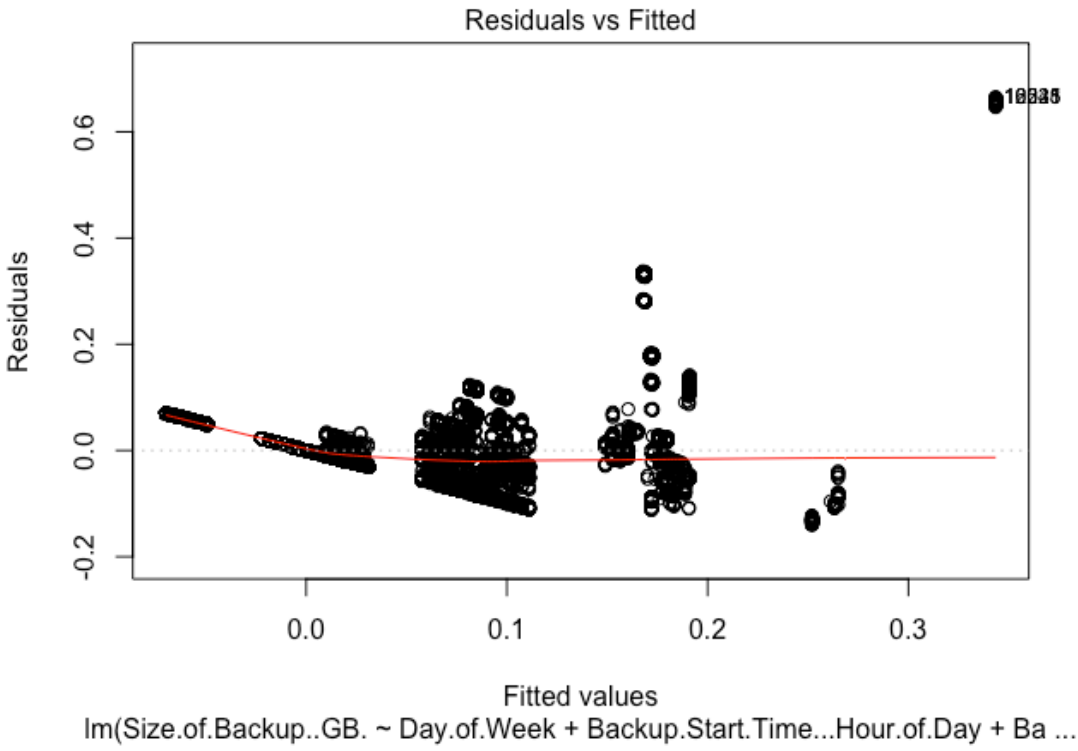
As can be seen from the plots, we observe a periodic property in size of backup data with time, which repeats in an interval of 1 week.

Problem 2

a) Linear Regression Model

For this part, we fit a linear regression model on our data with backup size at the target variable and other attributes as features. We use the function 'lm' in R for this purpose. We get an RMSE of **0.07833304**.

The p-value for each parameter tests the null hypothesis that the coefficient is equal to zero (no effect). A low p-value (< 0.05) indicates that we can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be a meaningful addition to the model because changes in the predictor's value are related to changes in the response variable. Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response. Following this analysis we chose the best parameters which were : Backup time, backup days, start time. Using these parameters, we trained the linear regression model again and obtained an RMSE of **0.0740.** Following is the plot of the model-

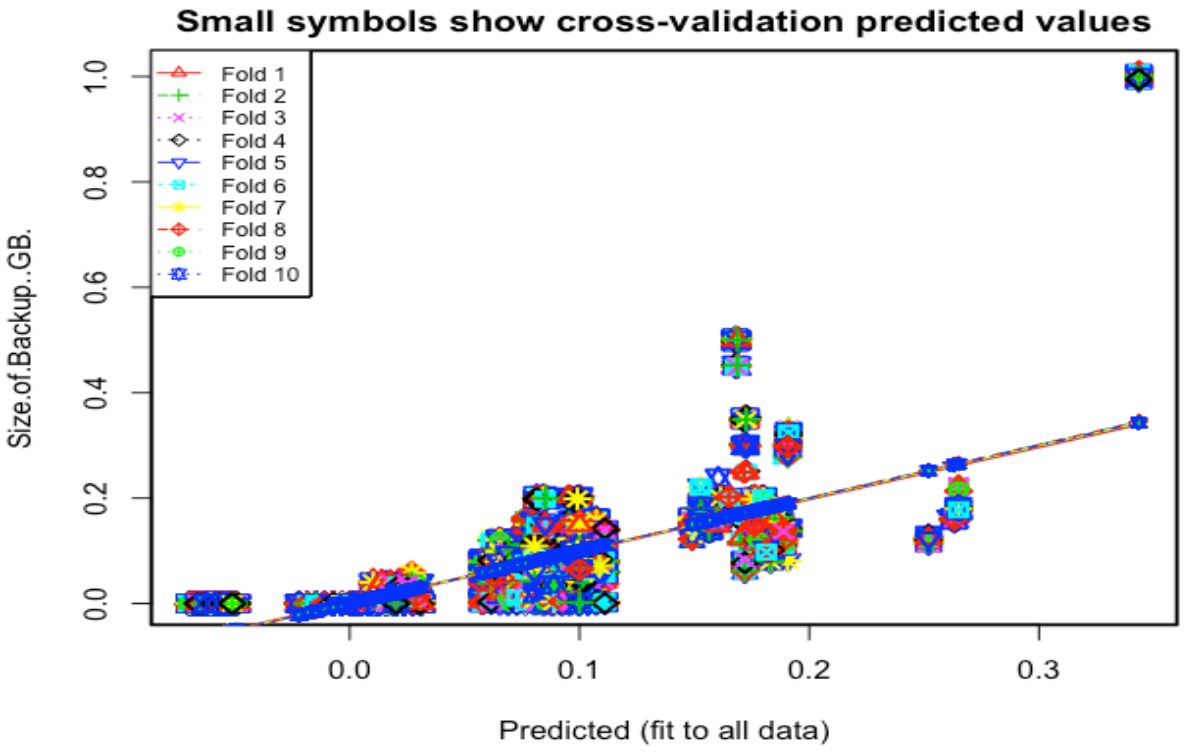


```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                   -7.073e-02  1.882e-03 -37.575   <2e-16 ***
Day.of.WeekMonday              8.224e-02  2.061e-03  39.909   <2e-16 ***
Day.of.WeekSaturday            7.026e-02  2.079e-03  33.788   <2e-16 ***
Day.of.WeekSunday              7.126e-02  2.084e-03  34.188   <2e-16 ***
Day.of.WeekThursday            4.769e-02  2.057e-03  23.184   <2e-16 ***
Day.of.WeekTuesday             1.892e-03  2.061e-03   0.918    0.359
Day.of.WeekWednesday           5.509e-02  2.095e-03  26.297   <2e-16 ***
Backup.Start.Time...Hour.of.Day  9.328e-04  7.957e-05  11.723   <2e-16 ***
Backup.Time..hour.             7.997e-02  6.163e-04 129.758   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07406 on 18579 degrees of freedom
Multiple R-squared:  0.495,     Adjusted R-squared:  0.4948
F-statistic:  2276 on 8 and 18579 DF,  p-value: < 2.2e-16
```

The stars beside each parameter are indicative of the importance of each parameter.

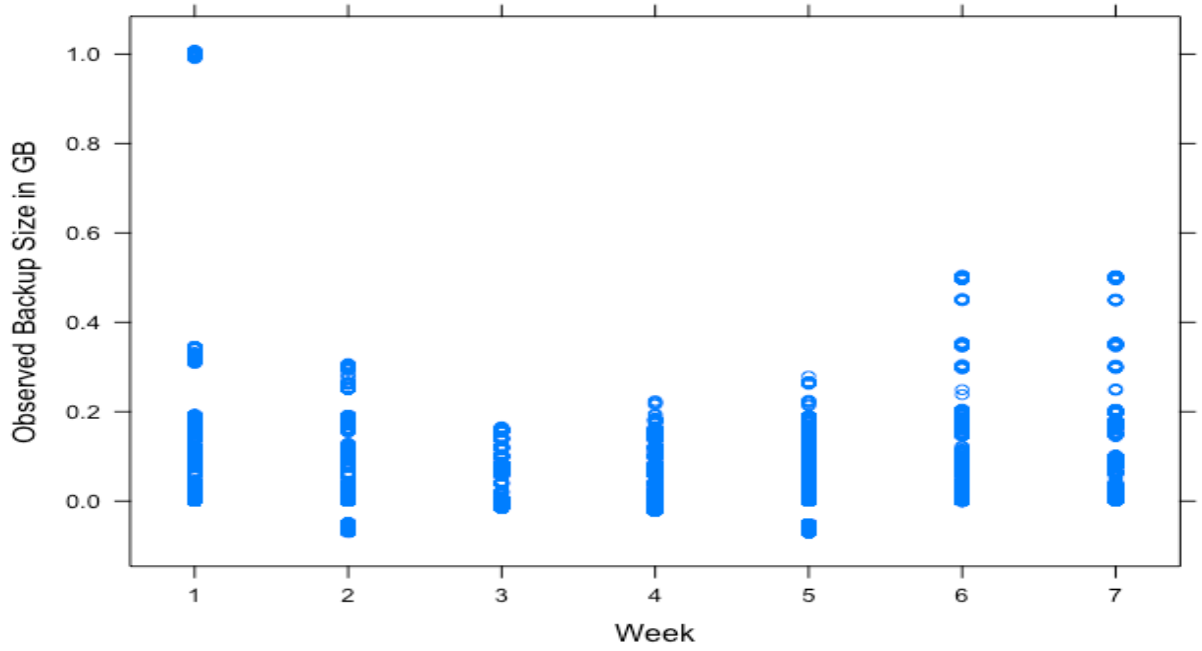
We then use 10-fold cross-validation to test our model. We split the data randomly into 10 parts and each time take 90% of the data for training and regard the other 10% to have an unknown response variable for testing. We perform this process 10 times, each time taking a different set of test data and finally calculate the RMSE values. Doing this gives us an RMSE of **0.07407877,** which was similar to the result obtained without cross-validation.

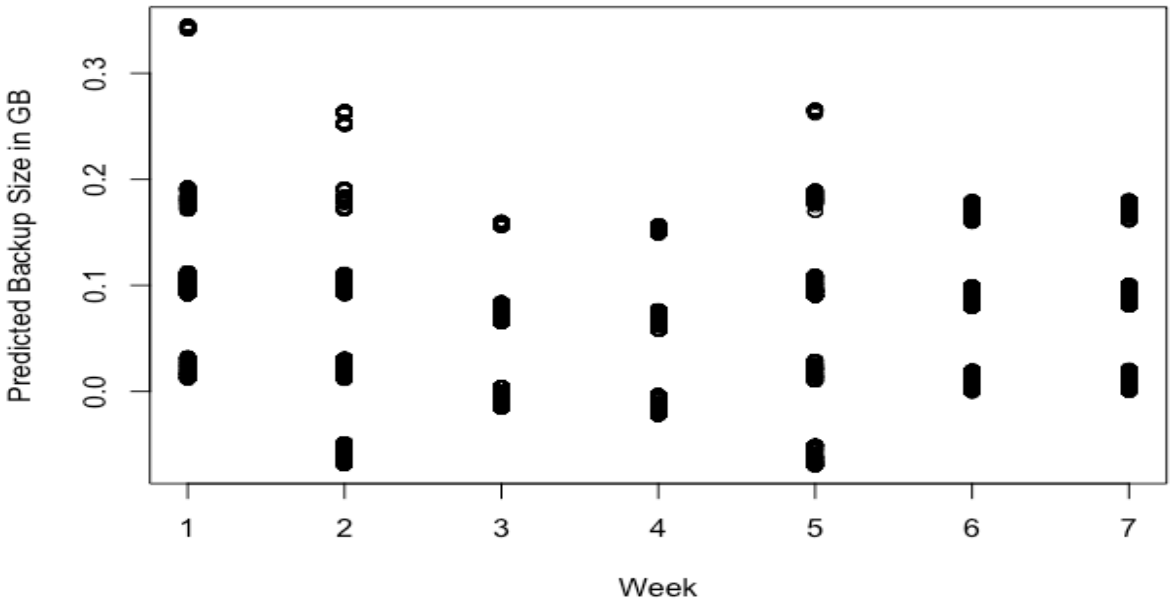
Using an inbuilt function in R, we plot the actual and predicted values for every fold –

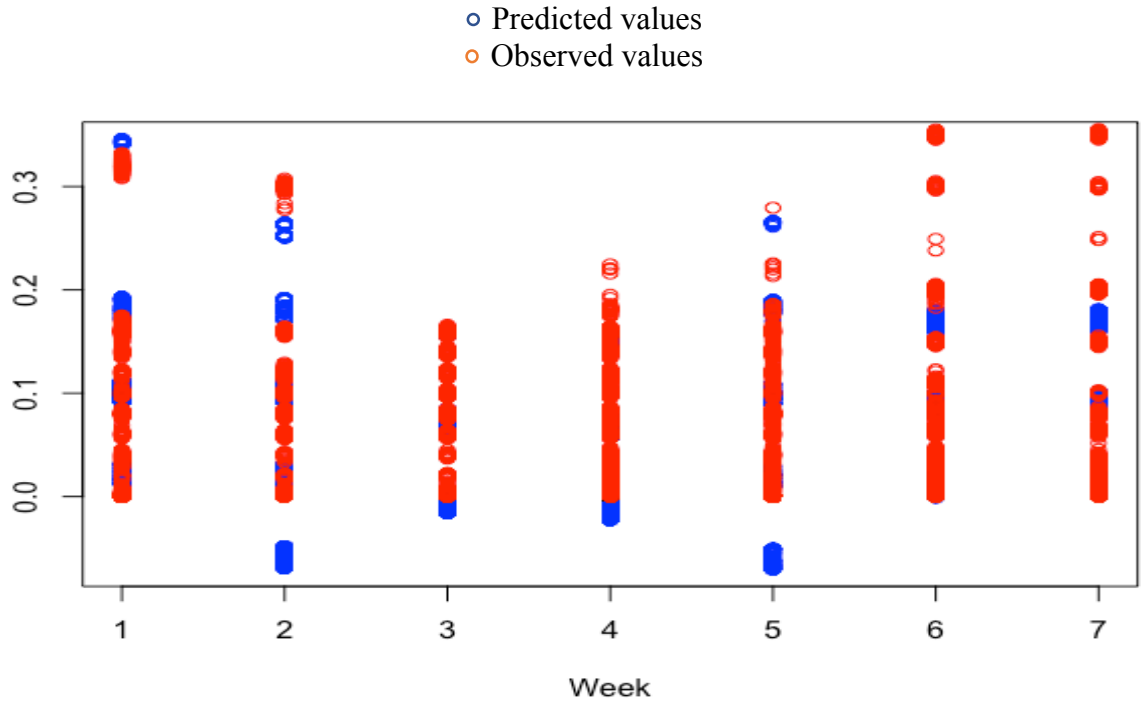


The plot shows that the predicted values were very close to actual values of backup size for most of the folds. But for some folds, this might not be true. Hence doing a cross-validation analysis is necessary. This also the reason why randomizing data helps. In our case, cross-validation does not yield very different result because most of the folds predict values were close to the actual values.

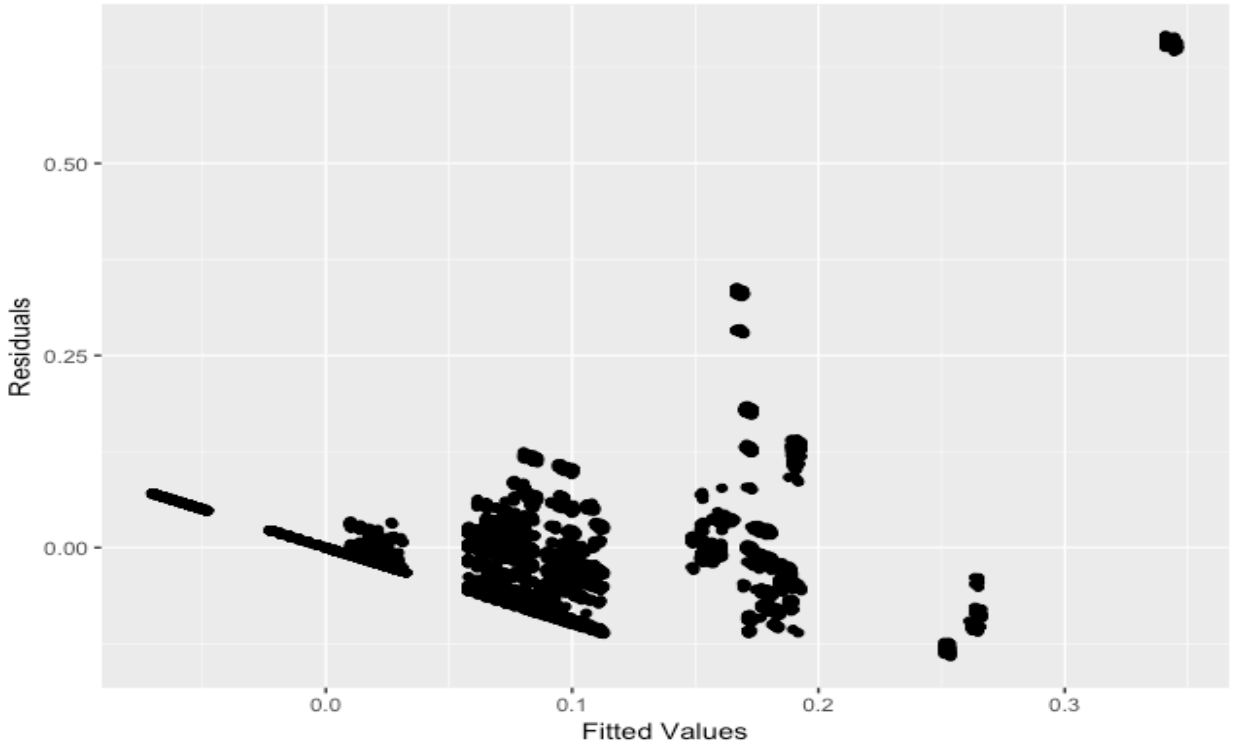
We plot fitted values and actual values over time –







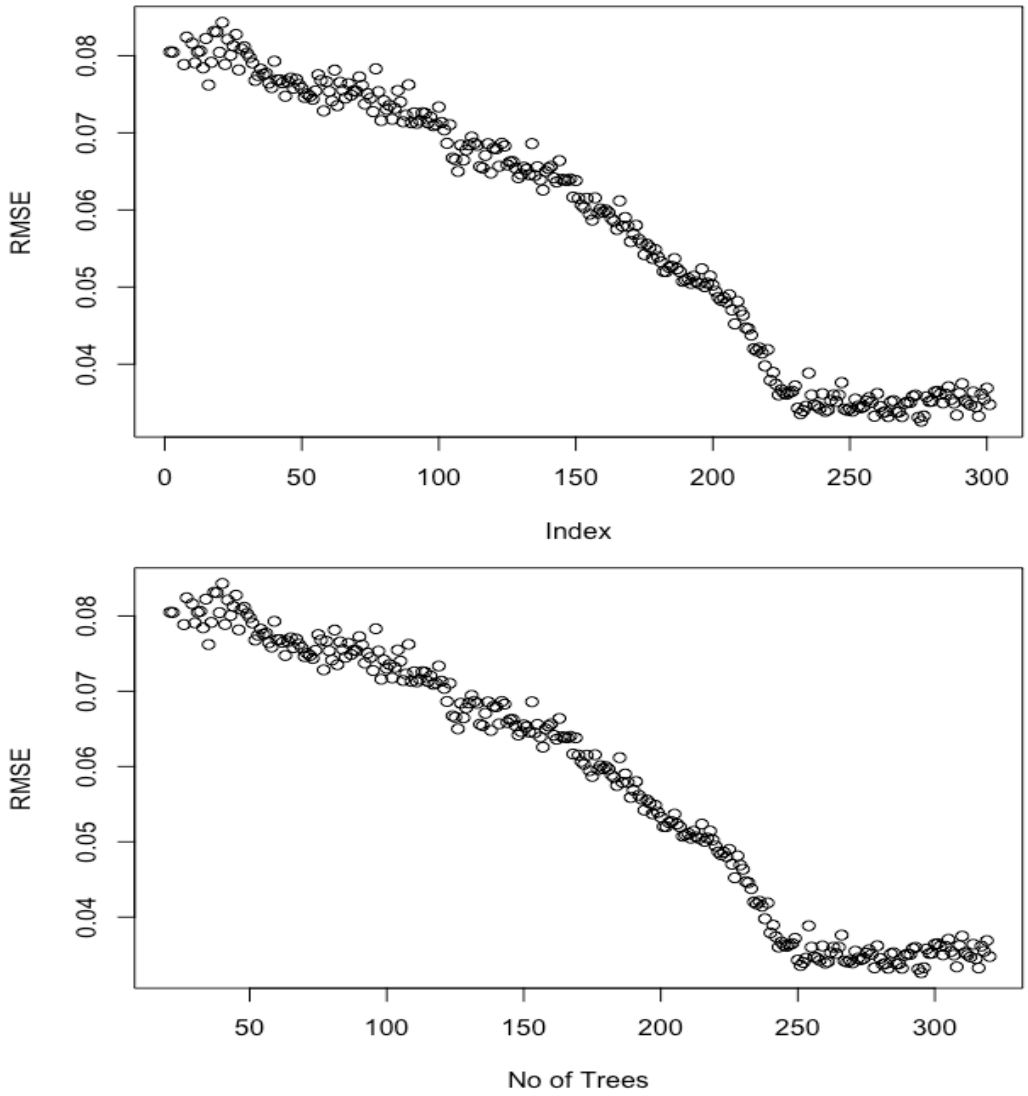
Residual vs Fitted values –

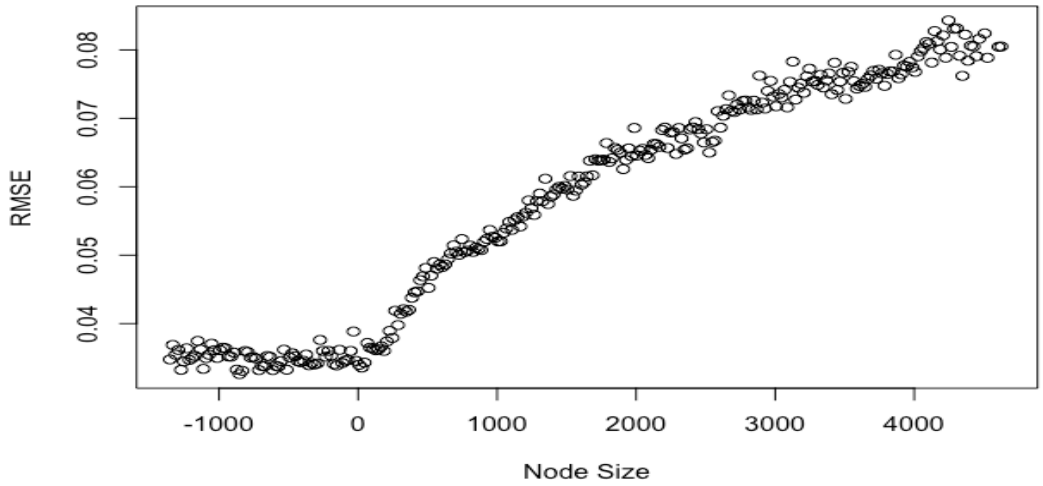


b) Random Forest Regression

We use 'randomForest' library of R. 'randomForest()' function takes as input the data and parameters such as node size and number of trees in the forest. Node size represents the maximum number of observations that can fit in a node. Node size is indicative of the depth and is inversely proportional to it.
To calculate node size from depth, we approximate node size as the number of observations divided by the depth.
We first develop a random forest model with depth = 4 and number of trees = 20 to extract the most important features. These were found to be Day of Week, Start time, backup time and workflow id. We use these features in our subsequent analysis.

We vary the gradually increase the number of trees and decrease the node size (increase depth) and obtain the following results-

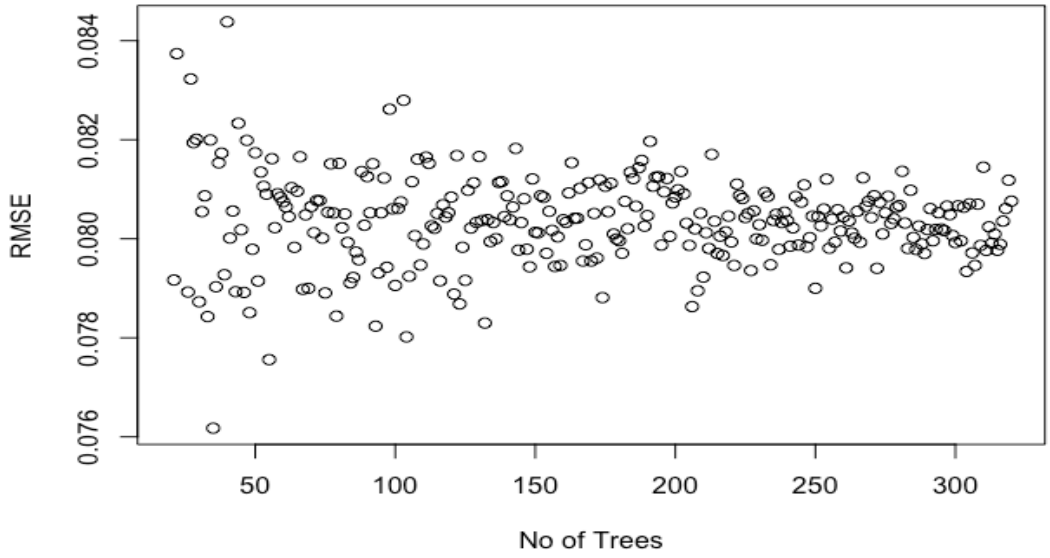




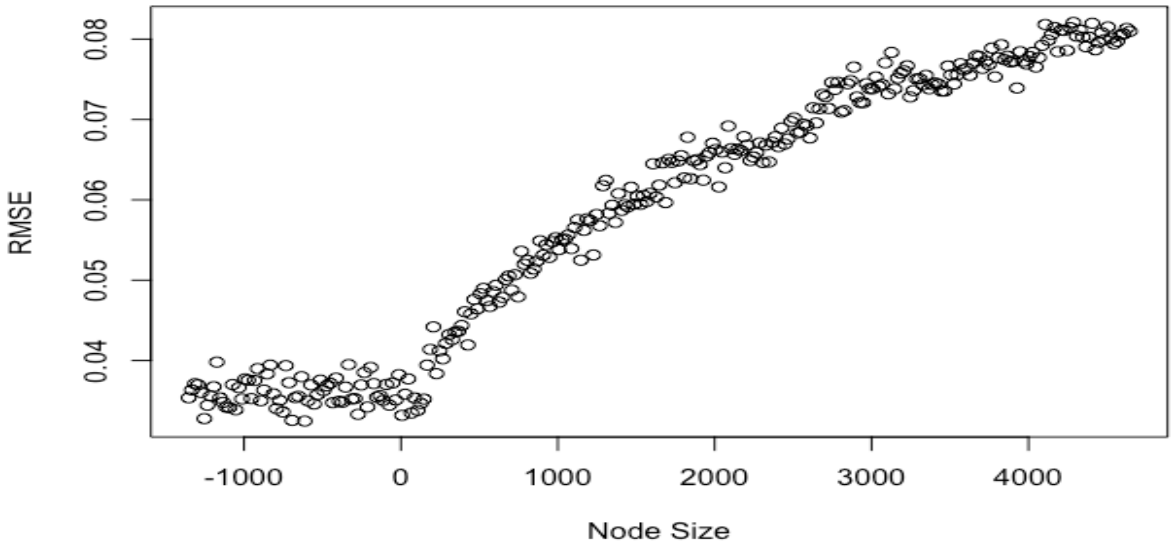
With increasing index, the number of trees increase and the depth also increases. We can see that RMSE reduces with increase in number of trees and depth, but does not decrease any further after a threshold is reached. We obtain the best RMSE of **0.03262175** at **number of trees = 255** and **node size = 60.**

On comparing with Linear Regression, we find Random Forests perform much better, giving an RMSE almost 0.04 less than the one obtained with Linear Regression.

NOTE: During our analysis we also observed that changing only the number of trees does not change the RMSE values much as shown below



Also, to test the effect of changing only the depth, we kept number of trees constant to 80 and observe that depth plays a more important role in RMSE reduction as compared to tree size as shown below-



Therefore, increasing depth has a more profound effect on the accuracy of the model as compared to number of trees.

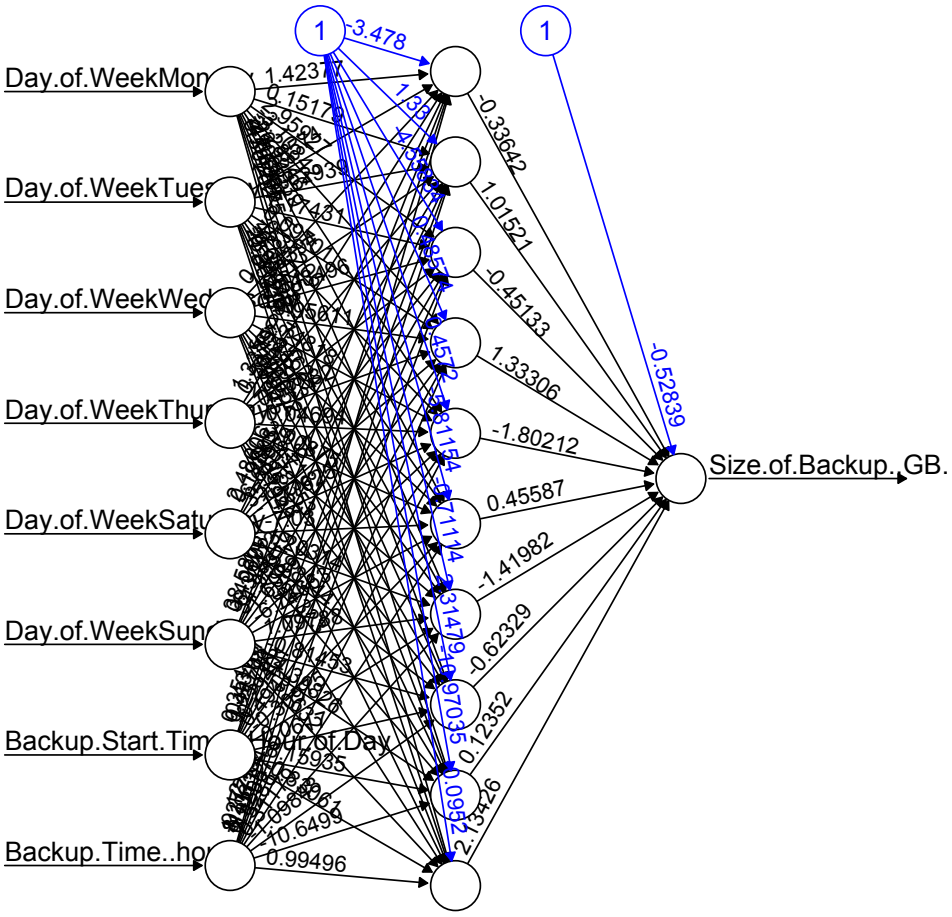
c) Neural Networks

'Day of Week' is a categorical parameter and needs to be converted into numerical form before being fed into the neural network. We use one hot encoding for this purpose. For modeling neural networks, we use the library 'neuralnet' in R. We vary the number of hidden layers and the number of hidden units in each layer.

We first use one hidden layer and vary the number of hidden units in it. (We do not increase the number of hidden units above the number of input parameters)

Number of hidden units	RMSE
2	0.04681281034
4	0.03923748097
6	0.03628660076
8	0.03411434981
10	**0.03263919916**

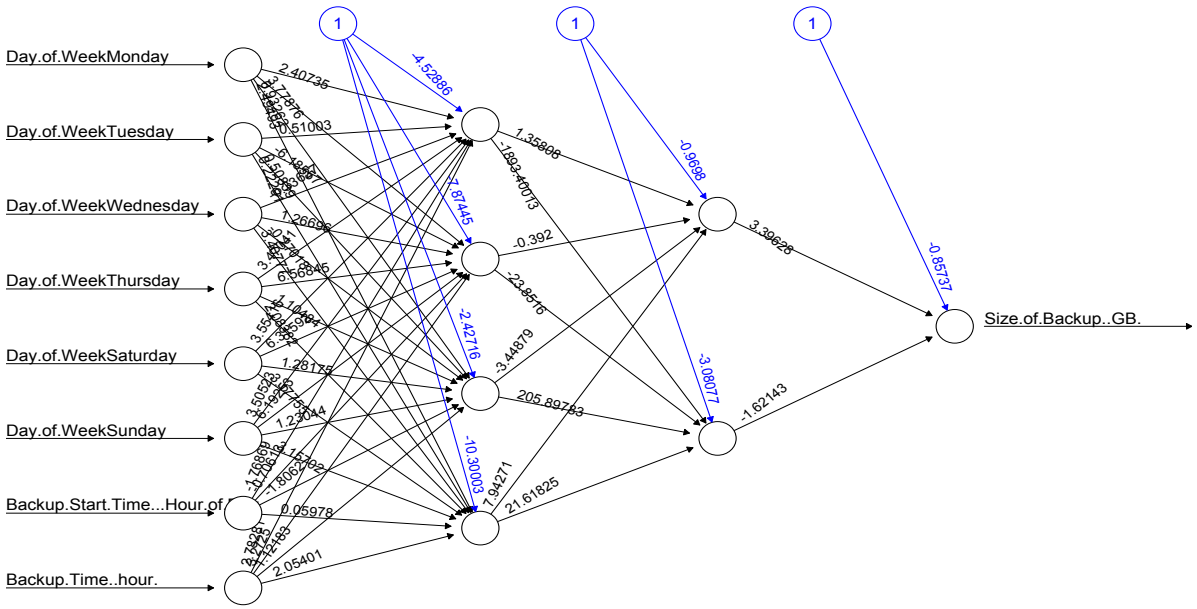
We obtain minimum RMSE in this case at hidden units = 10.



We then increase the number of hidden layers and obtain the following results –

Number of units in first hidden layer	Number of units in second hidden layer	RMSE
2	2	0.03799274183
4	2	**0.03182765621**
4	4	0.03297858879

We observe that minimum RMSE was obtained at 2 hidden layers, with first layer having 4 hidden units and second later having 2 hidden units.



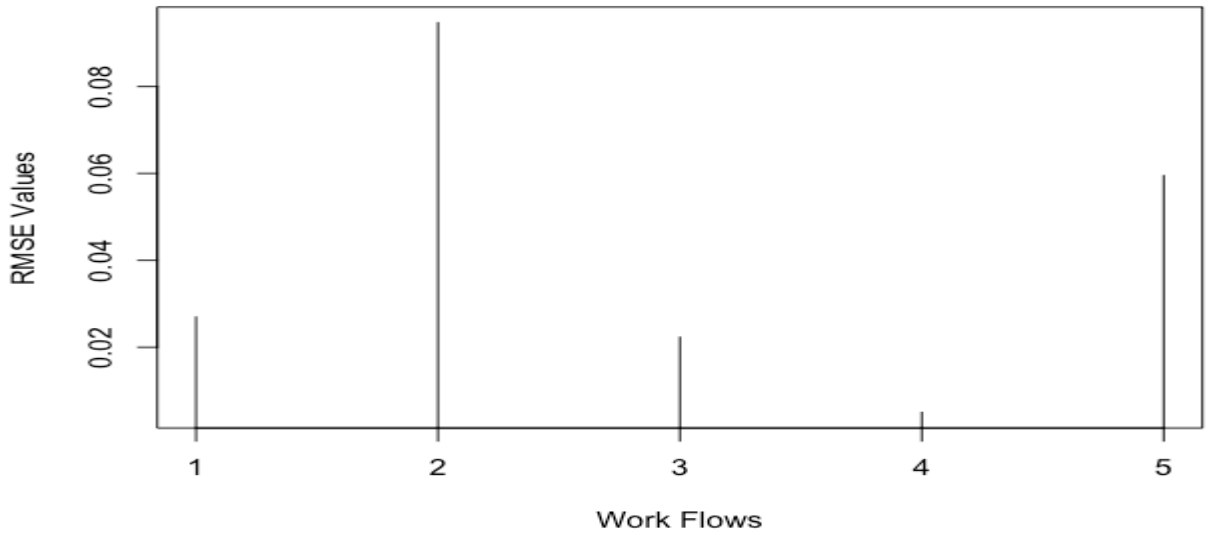
Error: 12.636118 Steps: 22901

Problem 3 – Piecewise Linear Regression and Polynomial Regression

Piecewise Linear Regression

Piecewise Linear Regression is a method in regression analysis in which the independent variable is partitioned into intervals and a separate line segment is fit to each interval. In the network backup dataset, we observe that data is mostly categorical and discontinuous and hence piecewise linear regression might give us better results.

We first separate the data according to workflows and fit linear regression models separately for each workflow. We use the same features as used in part 2a). We obtained a mean RMSE of **0.04169366485.**

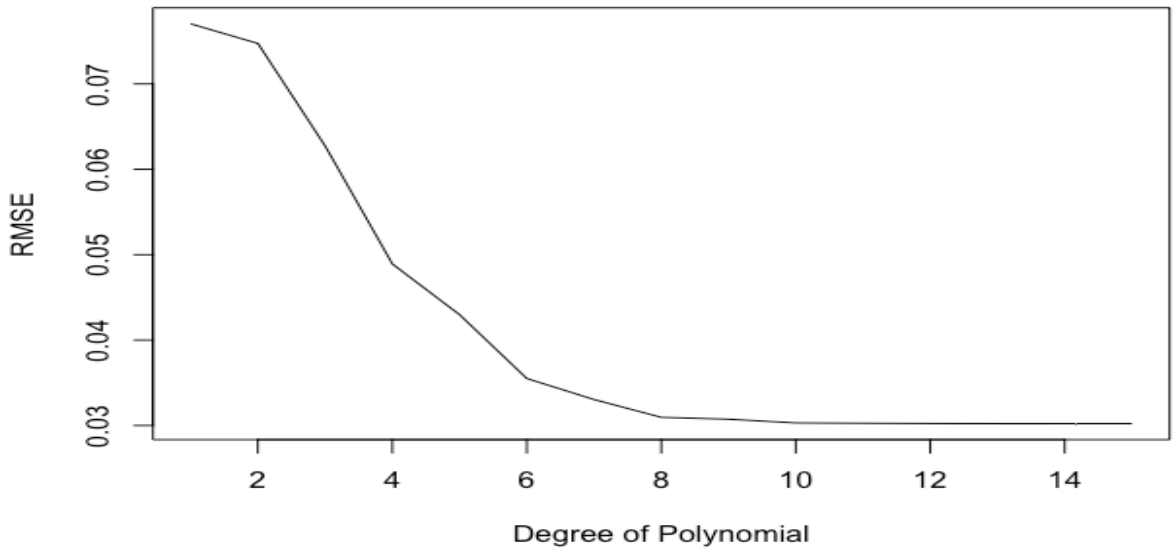


For comparison purpose, we performed Random Forest modeling by excluding the Workflow ID to maintain consistency with our linear model. With such a random forest we obtain an RMSE of **0.04180376493.** Piecewise Linear performs better than even Random Forest in this case.

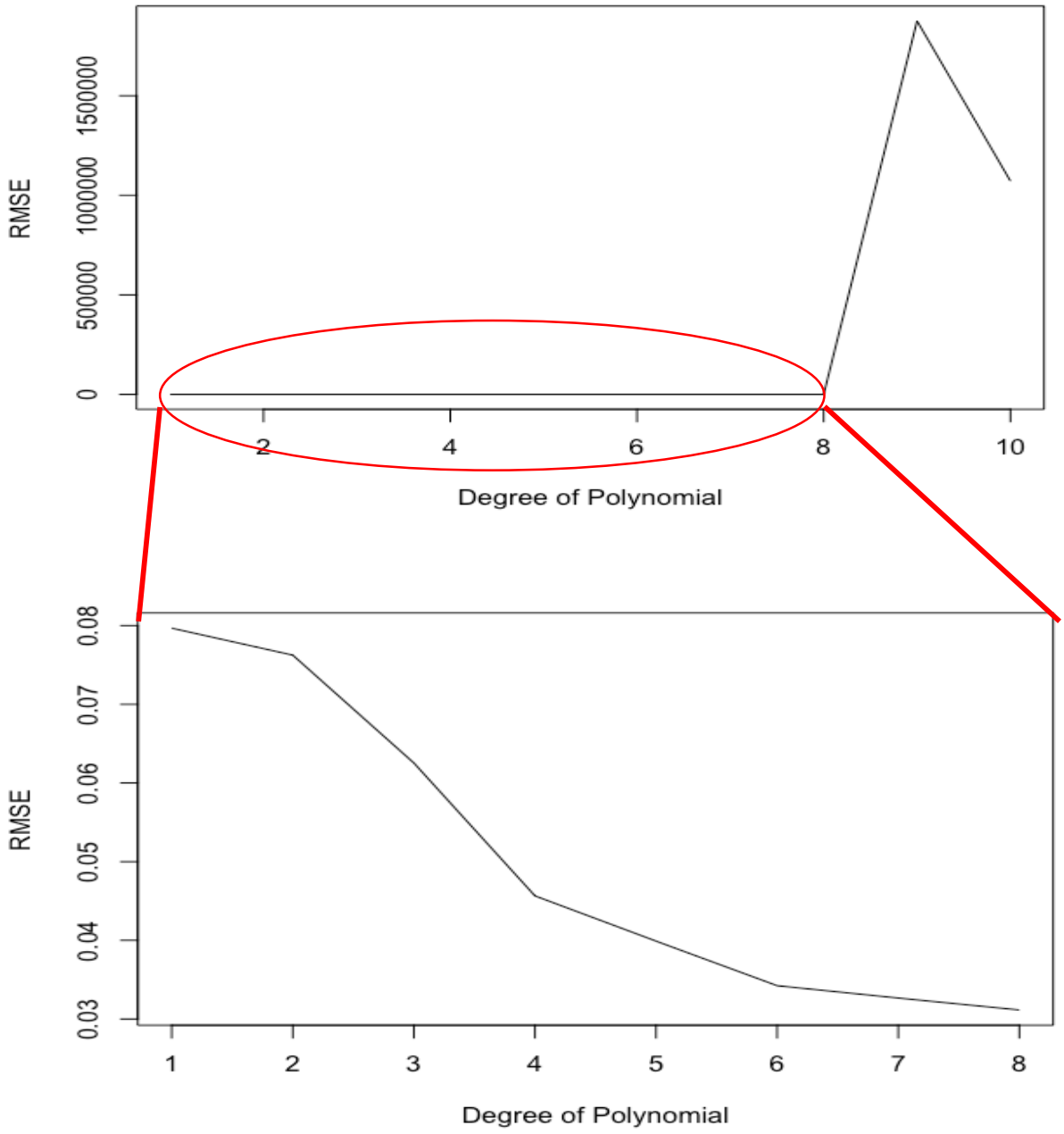
Polynomial Regression

We use the function 'polym' to formulate polynomial formula for the target variable. The function generates all possible combinations of input features in the given degree. We then model a regression model using this polynomial formula.

We first the training set as the first 90% observations from the given data and set test data as the remaining 10% observations. We then vary the degree and plot the RMSE values obtained-



We then performed 10-fold cross-validation for polynomial regression and obtained the following results-



We obtained minimum **RMSE of 0.03114577 for degree 8.**

PART 2

Dataset used : Boston Housing Dataset.
Description of dataset :
This dataset concerns housing values in the suburbs of the greater Boston area and is taken from the StatLib library which is maintained at Carnegie Mellon University. There are around 500 data points with the following features

- CRIM: per capita crime rate by town
- ZN: proportion of residential land zoned for lots over 25,000 sq. ft.
- INDUS: proportion of non-retail business acres per town
- CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX: nitric oxides concentration (parts per 10 million)
- RM: average number of rooms per dwelling
- AGE: proportion of owner-occupied units built prior to 1940
- DIS: weighted distances to five Boston employment centers
- RAD: index of accessibility to radial highways
- TAX: full-value property-tax rate per $10,000
- PTRATIO: pupil-teacher ratio by town
- B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT: % lower status of the population
- MEDV: Median value of owner-occupied homes in $1000's

Problem 4

Linear Regression

We first choose the best parameters that can be used to fit a linear model. We first train a model using all parameters and obtain the following results –

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.634941   5.102043   7.180 2.59e-12 ***
CRIM         -0.107417   0.032847  -3.270 0.001150 **
ZN            0.046121   0.013721   3.361 0.000836 ***
INDUS         0.014269   0.061653   0.231 0.817071
CHAS          2.671108   0.861115   3.102 0.002033 **
NOX         -17.633641   3.818719  -4.618 4.96e-06 ***
RM            3.794307   0.417835   9.081  < 2e-16 ***
AGE           0.001076   0.013205   0.081 0.935079
DIS          -1.479179   0.199347  -7.420 5.19e-13 ***
RAD           0.301534   0.066398   4.541 7.04e-06 ***
TAX          -0.012053   0.003765  -3.202 0.001454 **
PTRATIO      -0.958874   0.130831  -7.329 9.60e-13 ***
B             0.009305   0.002684   3.467 0.000573 ***
LSTAT        -0.527600   0.050732 -10.400  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The number of stars beside each parameter indicates the importance of that parameter in predicting based on its p-value, as discussed in part 2. We choose the parameters with 3 stars besides them, namely ZN, NOX, RM, DIS, RAD, PTRATIO, B and LSTAT. Using these we train the model again and get the following results –

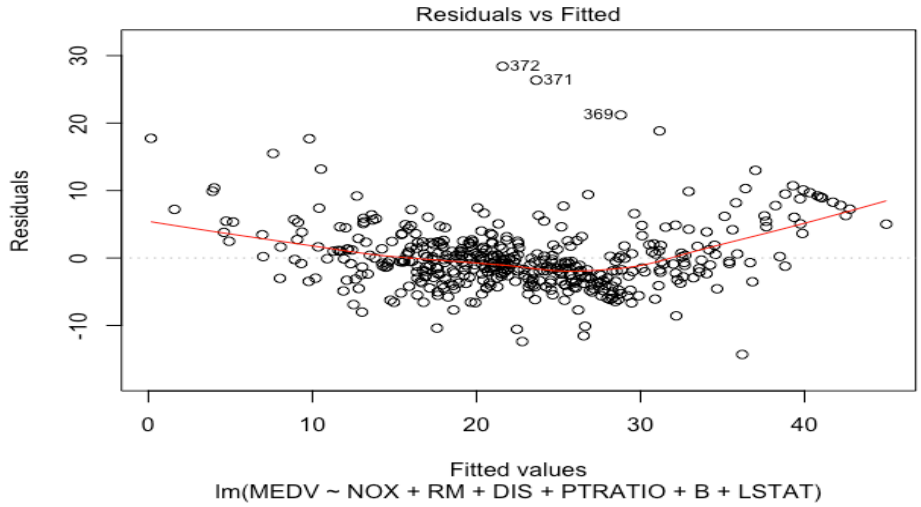
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  34.052804   5.195601   6.554 1.41e-10 ***
ZN            0.031680   0.013621   2.326   0.0204 *
NOX         -18.460559   3.520901  -5.243 2.34e-07 ***
RM            4.054542   0.415388   9.761  < 2e-16 ***
DIS          -1.392167   0.189169  -7.359 7.72e-13 ***
RAD           0.074176   0.037800   1.962   0.0503 .
PTRATIO      -1.052303   0.131062  -8.029 7.18e-15 ***
B             0.011276   0.002731   4.129 4.28e-05 ***
LSTAT        -0.563739   0.048227 -11.689  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.884 on 496 degrees of freedom
Multiple R-squared:  0.723,     Adjusted R-squared:  0.7185
F-statistic: 161.8 on 8 and 496 DF,  p-value: < 2.2e-16
```

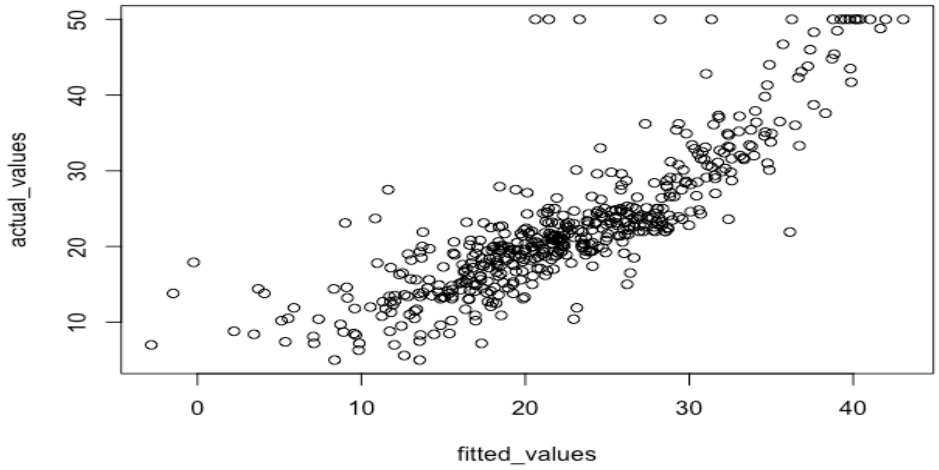
We fine tune the model more and choose only those parameters with 3 stars beside them namely NOX, RM, DIS, PTRATIO, B, and LSTAT. Using these we finally obtain our linear model –

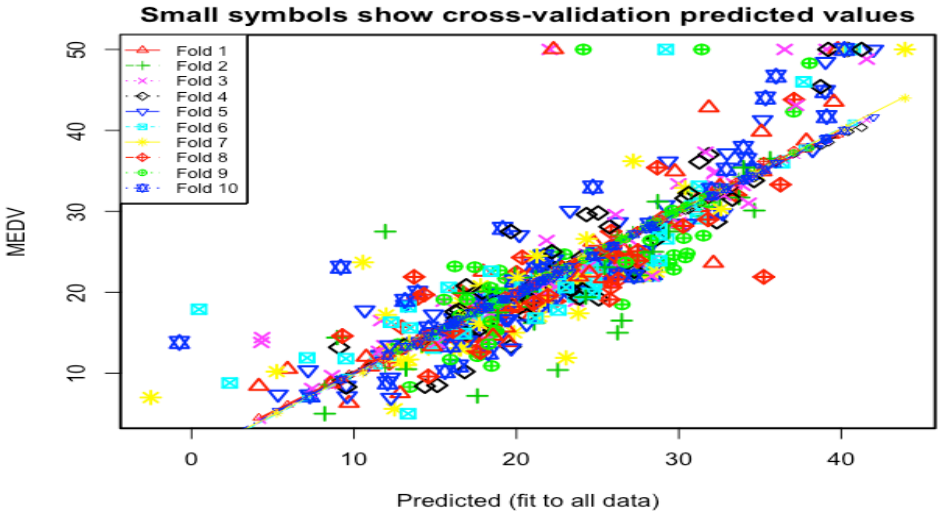
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  30.822005   4.957079   6.218 1.07e-09 ***
NOX         -15.782835   3.274798  -4.819 1.91e-06 ***
RM            4.338385   0.410351  10.572  < 2e-16 ***
DIS          -1.162611   0.166410  -6.986 9.08e-12 ***
PTRATIO      -1.021336   0.112613  -9.069  < 2e-16 ***
B             0.009586   0.002674   3.585  0.00037 ***
LSTAT        -0.548661   0.048395 -11.337  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.929 on 498 degrees of freedom
Multiple R-squared:  0.7167,    Adjusted R-squared:  0.7133
F-statistic:   210 on 6 and 498 DF,  p-value: < 2.2e-16
```

We observe that all parameters are important and thus stop fine tuning at this point. The RMSE obtained was **4.895144**.

We then perform 10-fold cross validation using the final parameters obtained an RMSE of **5.022797.** We plot residual vs fitted values , actual vs fitted values for each fold -

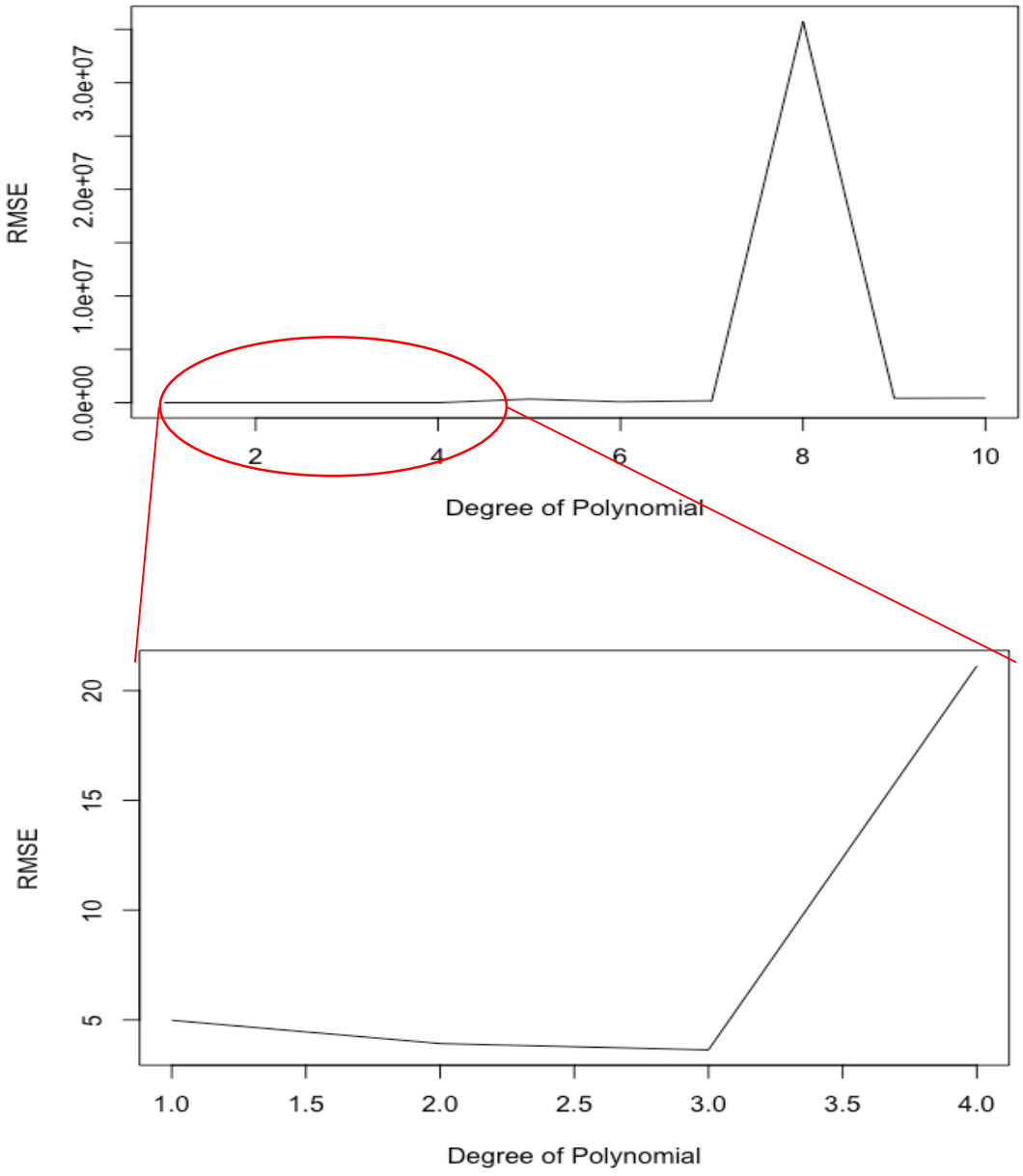

Residuals vs Fitted
Residuals
Fitted values
lm(MEDV ~ NOX + RM + DIS + PTRATIO + B + LSTAT)
372
371
369


actual_values
fitted_values


Small symbols show cross-validation predicted values
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Fold 7
Fold 8
Fold 9
Fold 10
MEDV
Predicted (fit to all data)

Polynomial Regression

We then performed Polynomial Regression using the same parameters as obtained previously. We perform 10-fold cross validation to find the best degree. **The best degree obtained was 3 and the corresponding RMSE was 3.63.**



Problem 5 : Regularization

To perform Lasso and Ridge Regression, we use the R function 'glmnet'. 'glmnet' solves the following problem :

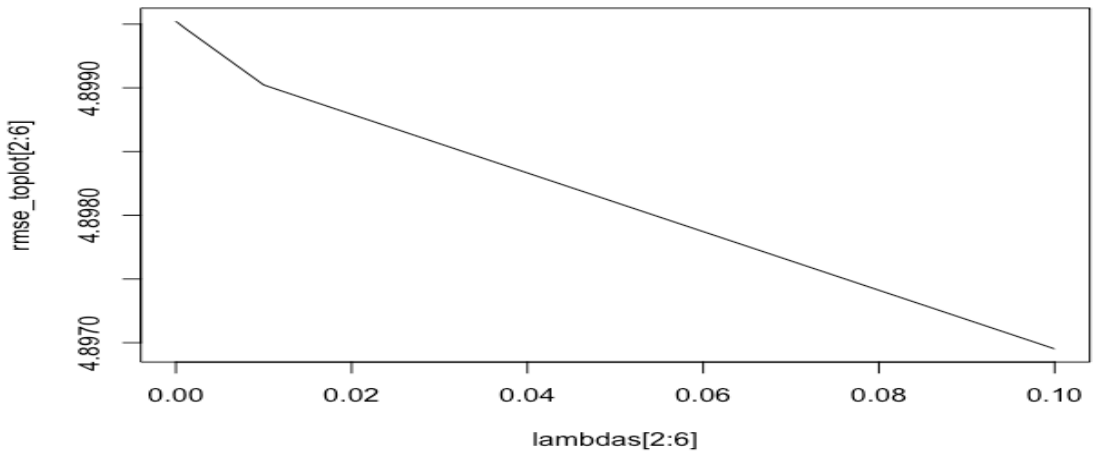
$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda \left[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1 \right],$$

over a grid of values of λ covering the entire range. Here l(y, η) is the negative log-likelihood contribution for observation i; e.g. for the Gaussian case it is 1 2 (y − η) 2 . The elastic-net penalty is controlled by α, and bridges the gap between lasso (α = 1, the default) and ridge (α = 0). The tuning parameter λ controls the overall strength of the penalty.

Thus for Ridge regression , we set alpha = 0 and vary the values of lambda in the vector (1.0,0.1,0.01,0.001,0.0001,0.00001). We perform 10-fold cross-validation and get the following results-

Lambda	RMSE
1.0	4.921144
0.1	**4.861304**
0.01	4.862714
0.001	4.863097
0.0001	4.863142
0.00001	4.8631462

We observe that there is not much difference in the RMSE values. Thus lambda does not have much effect. Minimum was obtained at **lamda = 0.1, RMSE = 4.681.**



For Lasso Regression, we set alpha = 1 and vary lamda as above. We use 10-fold cross-validation and obtain the following results –

Lambda	RMSE
1.0	5.427831
0.1	4.905366
0.01	**4.861268**
0.001	4.862909
0.0001	4.863154
0.00001	4.863144

As was the case with Ridge Regression, in Lasso Regression also we do not observe much variation in RMSE with change in lamda. The minimum **RMSE of 4.861268 was obtained at lambda = 0.01**