**EE219 Project 1**

**Regression Analysis**

**Winter 2017**

**Team members:**

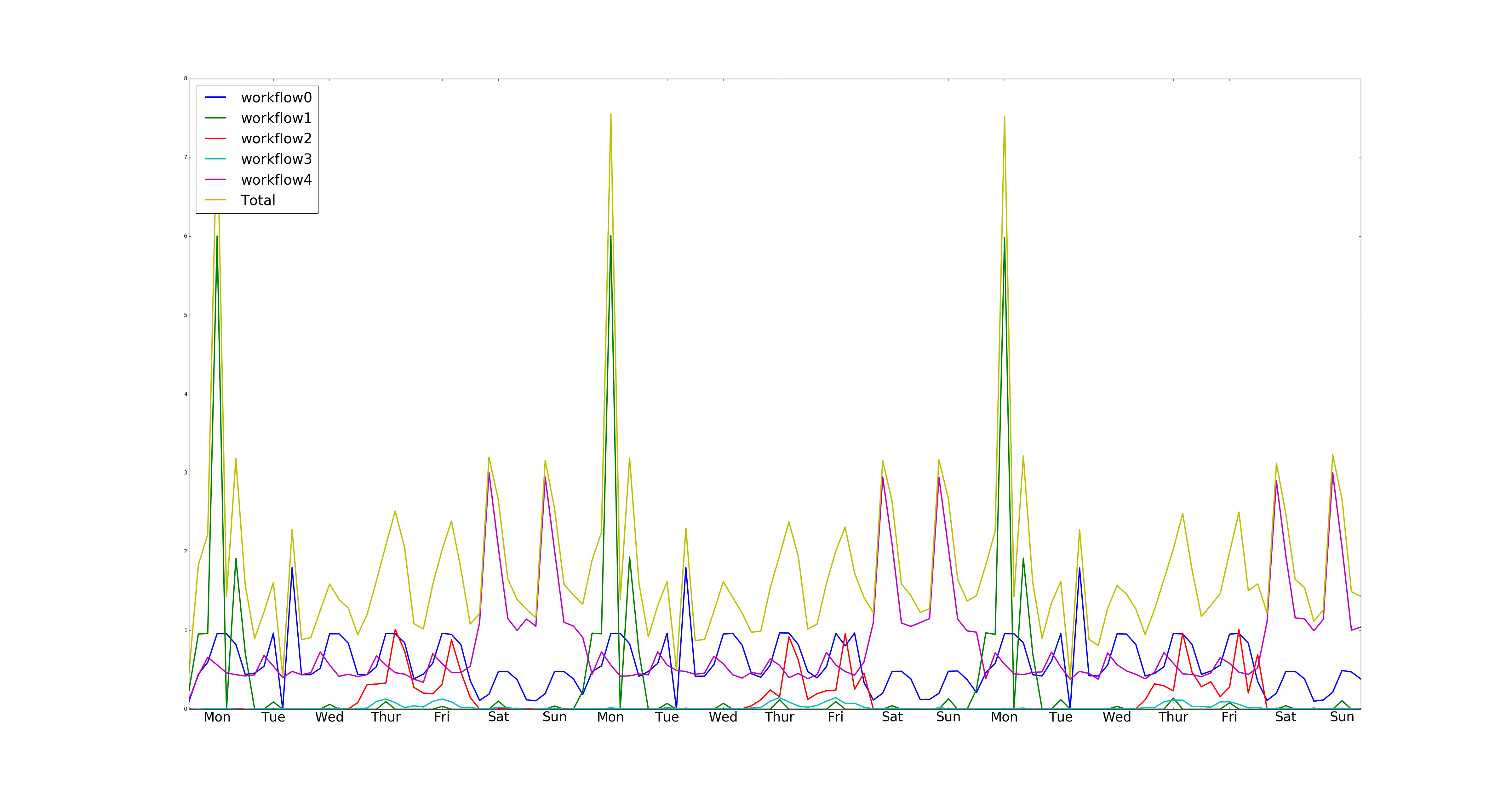
**Mojtaba Sahraee 204517437**

**Kia Karbasi 804617014**

**Heng Zhao 304436191**

1. **Plot of the size of the backup vs other features**

Size of Backup vs. Day for each workflow and all the workflows:



We plotted the size of backup for 3 weeks and obviously there is periodicity in the plot in which the period is a week. And it can be seen that the backup size has been the highest in Mondays and we also have relatively large backup size on Saturday and Sunday as well.

1. **Modeling the Network Backup Dataset**
2. **Linear Regression**

The linear regression minimizes

where is the output vector and is the matrix of feature vectors with each row being the feature vector of one of the input points.

Using the raw numbers in the file as the features we get the following coefficients for each of the features:

Week: 0.000 Day of the week: 0.001 Backup start time: 0.001 Work flow: 0.003 File number: -0.000 Backup Time: 0.071

Therefore, almost the only feature that has been used in the linear regression is the backup time.

In this regression the mean squared error on the training data is 0.006332 and the RMSE for 10 fold cross validation is .006334.

But since the features like file number, day of the week, … are categorical variables

If we use this minimization, the entries of vector of coefficients, , would become very large, indicating that the features in are not independent.

The main parameters of the model as can be seen are the T parameter and the P parameter. The t-value measures the size of the difference relative to the variation in your sample data. Put another way, T is simply the calculated difference represented in units of standard error. The greater the magnitude of T (it can be either positive or negative), the greater the evidence against the null hypothesis that there is no significant difference. The closer T is to 0, the more likely there isn't a significant difference. On the other hand P-value is a metric of rejecting the null hypothesis and concludes that there is a statically significant difference. So the parameters with high T value and low P values are significant in the model. For this model as we saw, the days of the week are mostly significant variables as can also be seen in the dataset plot (because clearly there is an obvious data size difference between Monday and Wednesday). But the week number is not significant and its T value is not very large and p value not very small. And that is also reasonable because the we have periodicity in the data with period of a week so week number cannot distinguish the results.

Here is the result for RMSE:

**RMSE= 0. 006334**

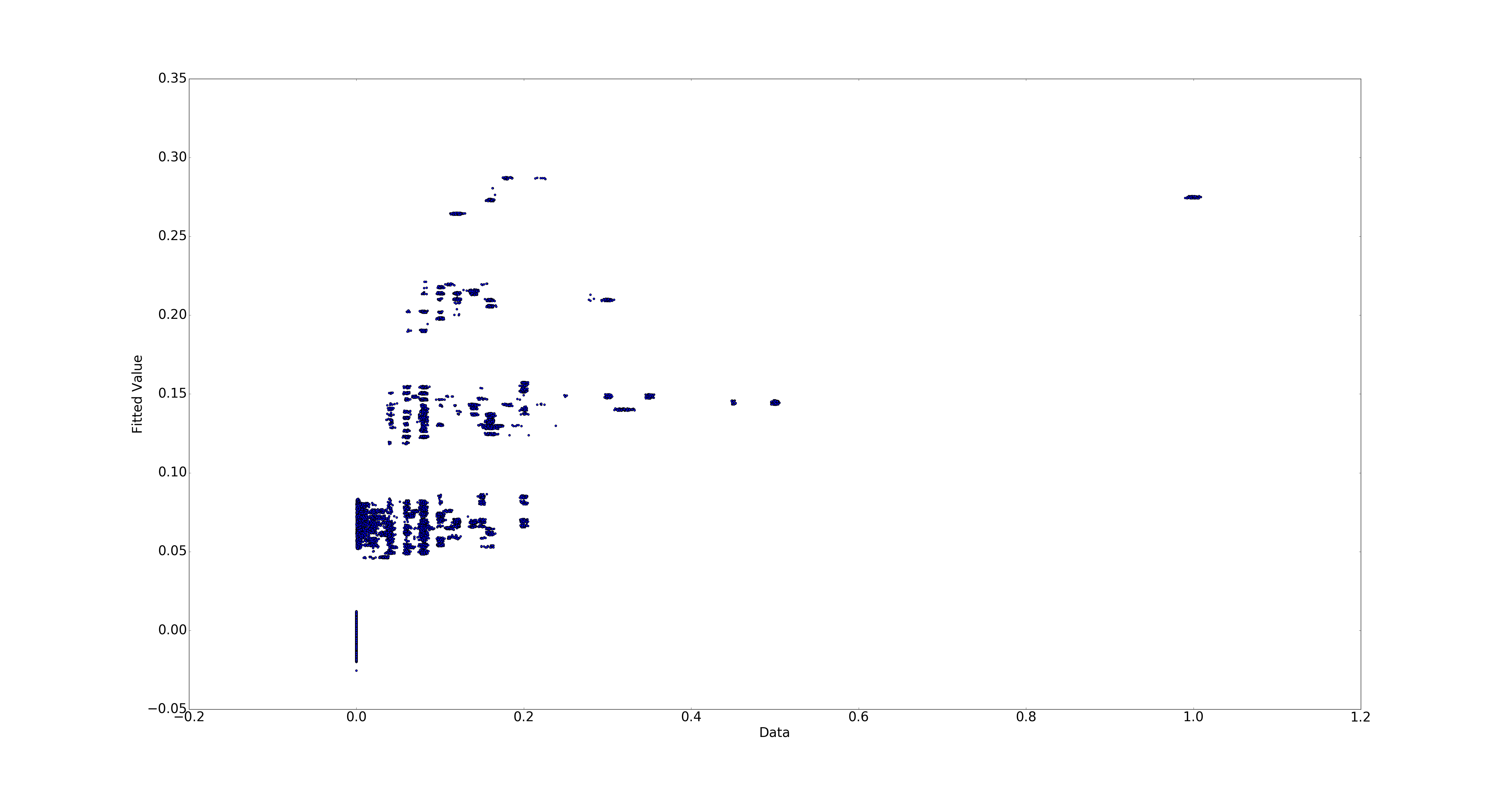
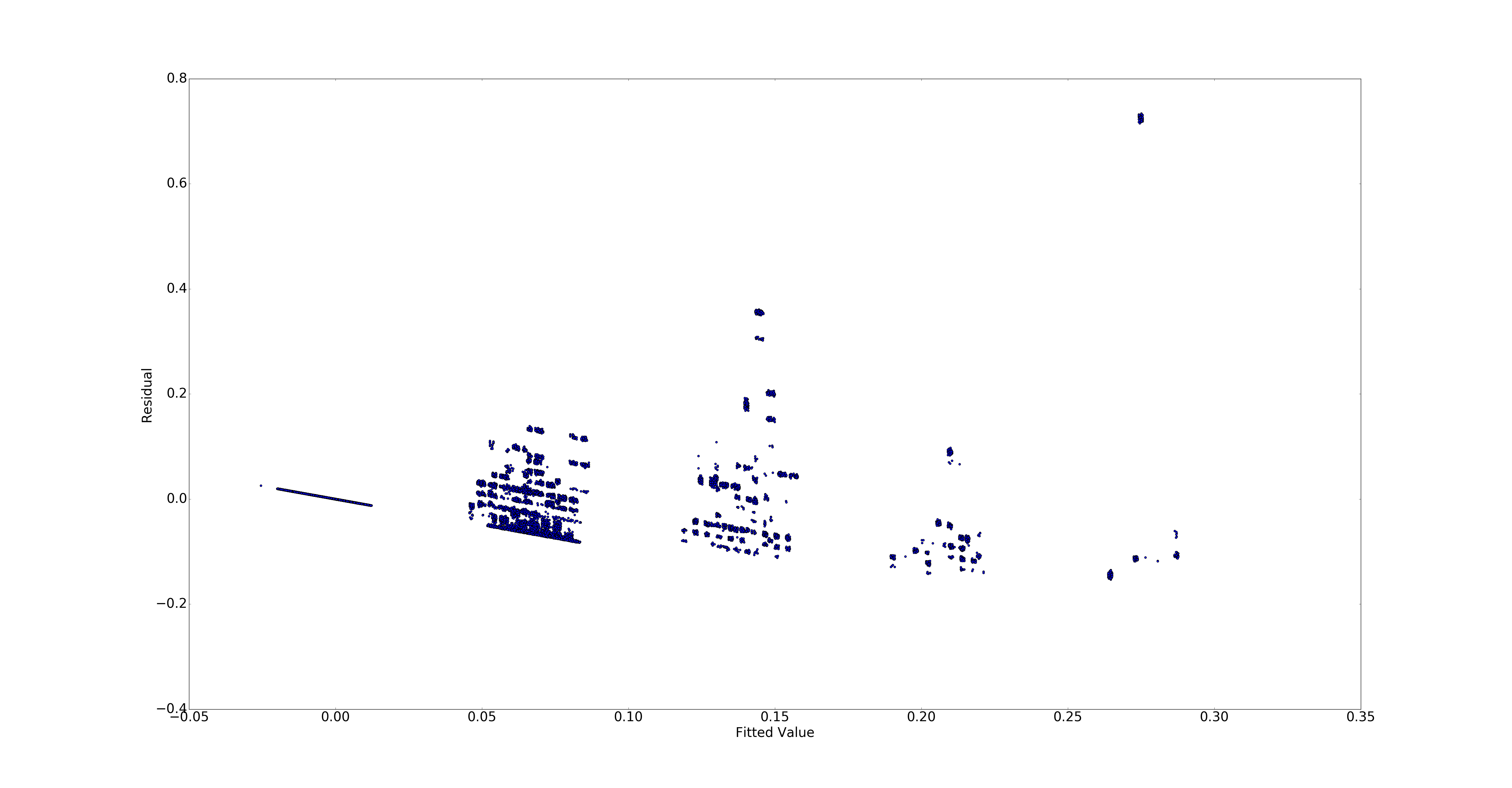
****

Figure 1- Fitted value vs Actual for Linear regression

As can be seen the results are fitted on a linear regression line and the next plot shows the residual which gives us better idea how far the predictions are from the actual values:

Figure 2-Residual vs Fitted value for Linear regression

The residual for most of the data points are close to zero which shows that the linear model efficiently predicts the “size of the backup”. Although 7% of RMSE shows that still linear regression might not be the best fit for this dataset.

**b) Random Forest:**

In this part Random Forest algorithm is been used instead of linear regression which might be good for not a linear dataset. The default has been set in the problem to be:

Tree number= 20 , Tree depth= 4

Now we plot the residual and fitted value.

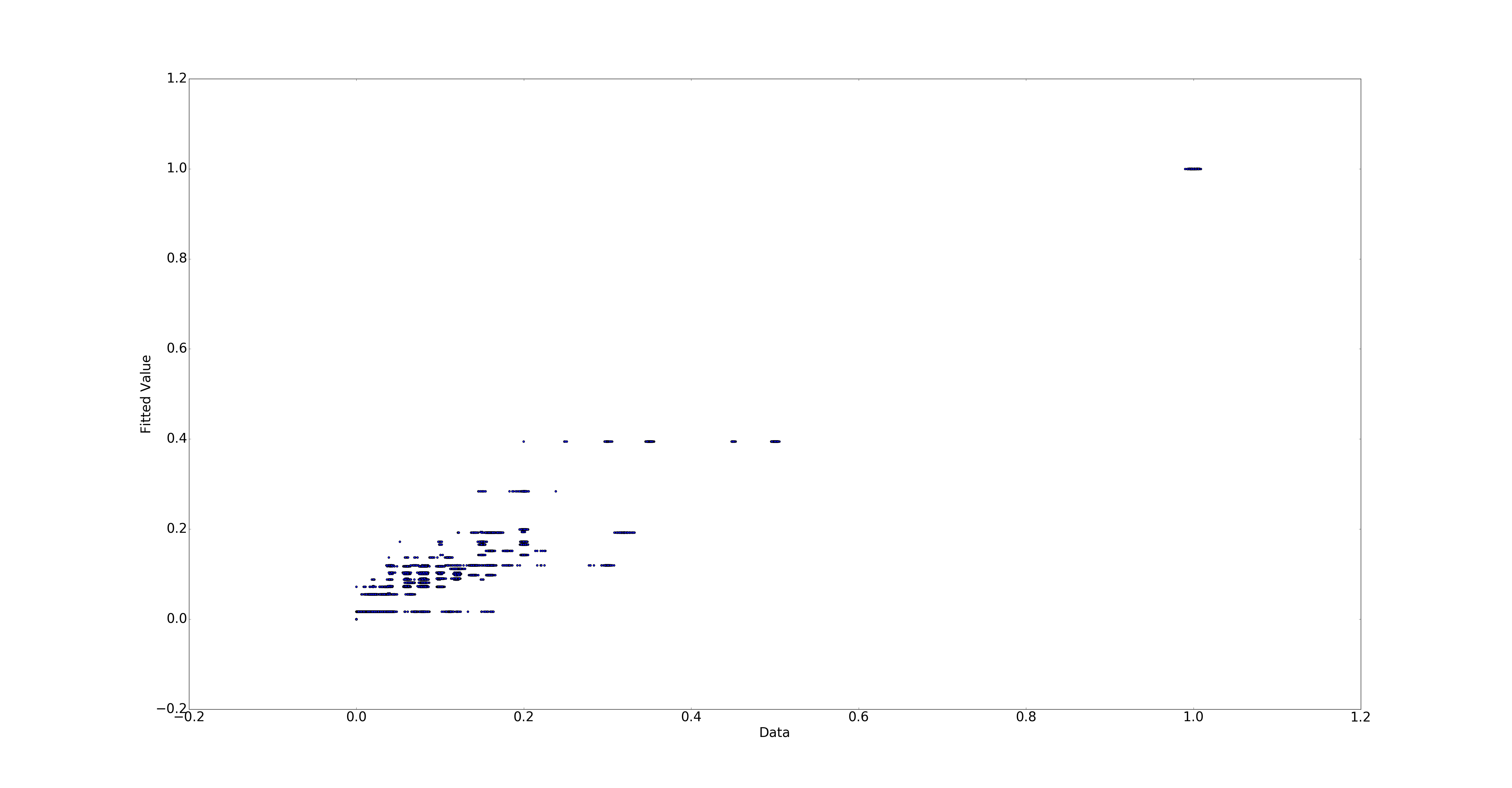


Figure 3-Fitted value vs Actual for Random Forest ntree=20 depth=4

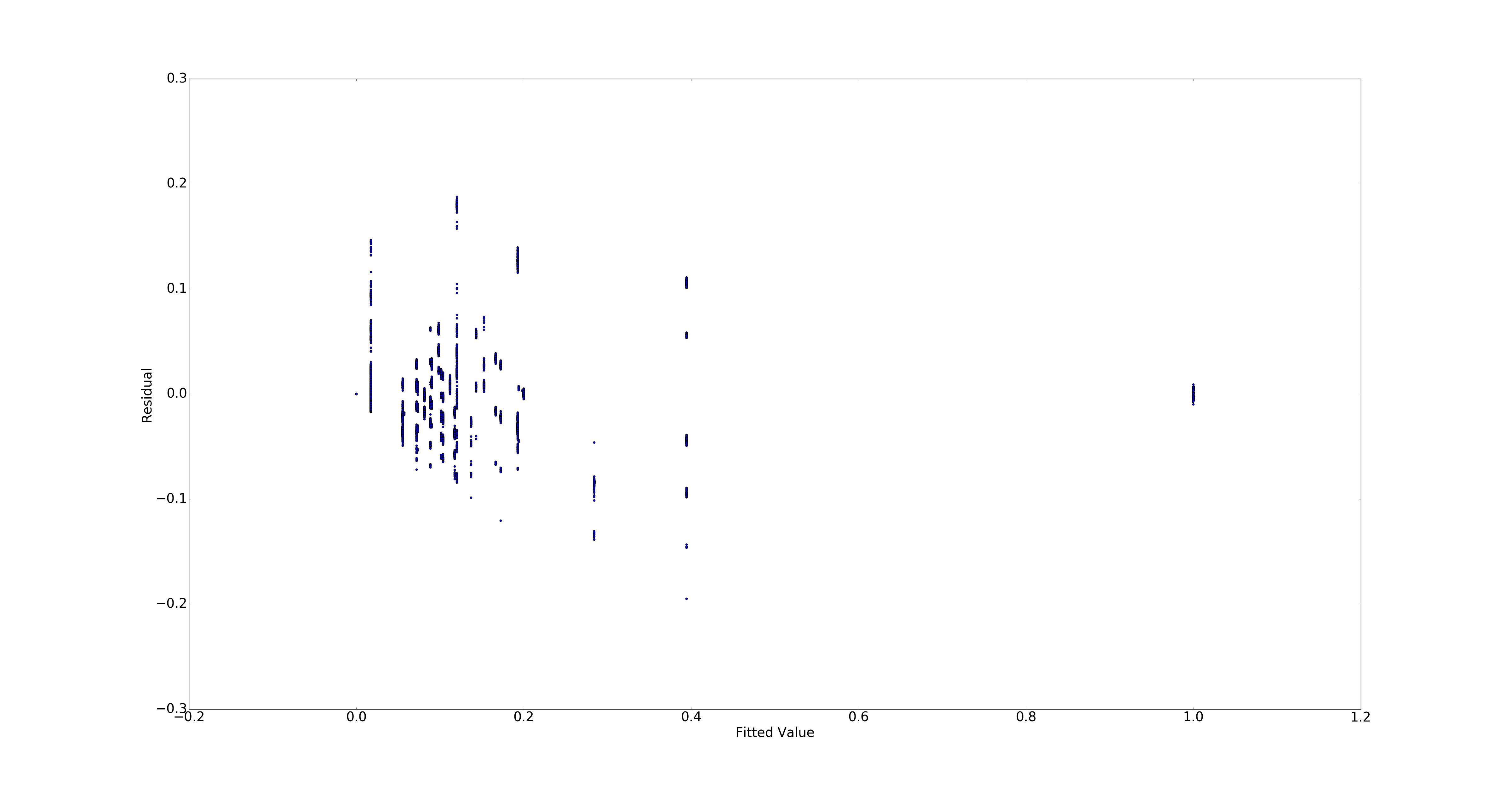


Figure 4- Residual vs Fitted value for Random Forest ntree=20 depth=4

The RMSE found to be: RMSE = 0.01327423

Training MSE =0.00087269073363

Test MSE mean (RMSE)= 0.000881887492204

Test\_MSE std = 6.18609778079e-05

Now tuning the parameters hopefully are giving us better results:

**Tree number= 60, Tree depth= 4 RMSE= 0.** **000893360499162**

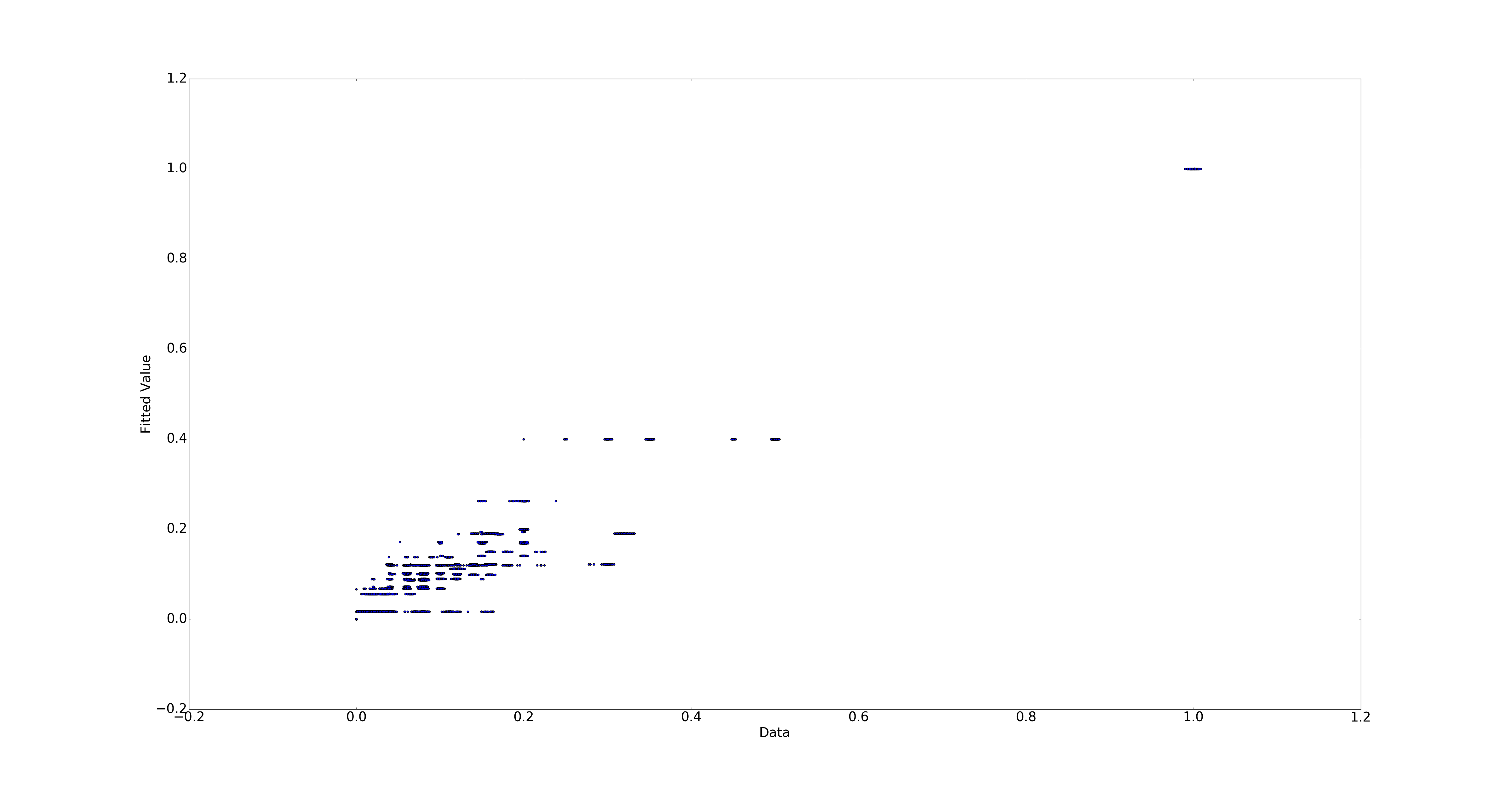


Figure 5-Fitted value vs Actual for Random Forest ntree=60 depth=4

**Tree number= 40 , Tree depth= 10 RMSE= 8.7380954522e-05**

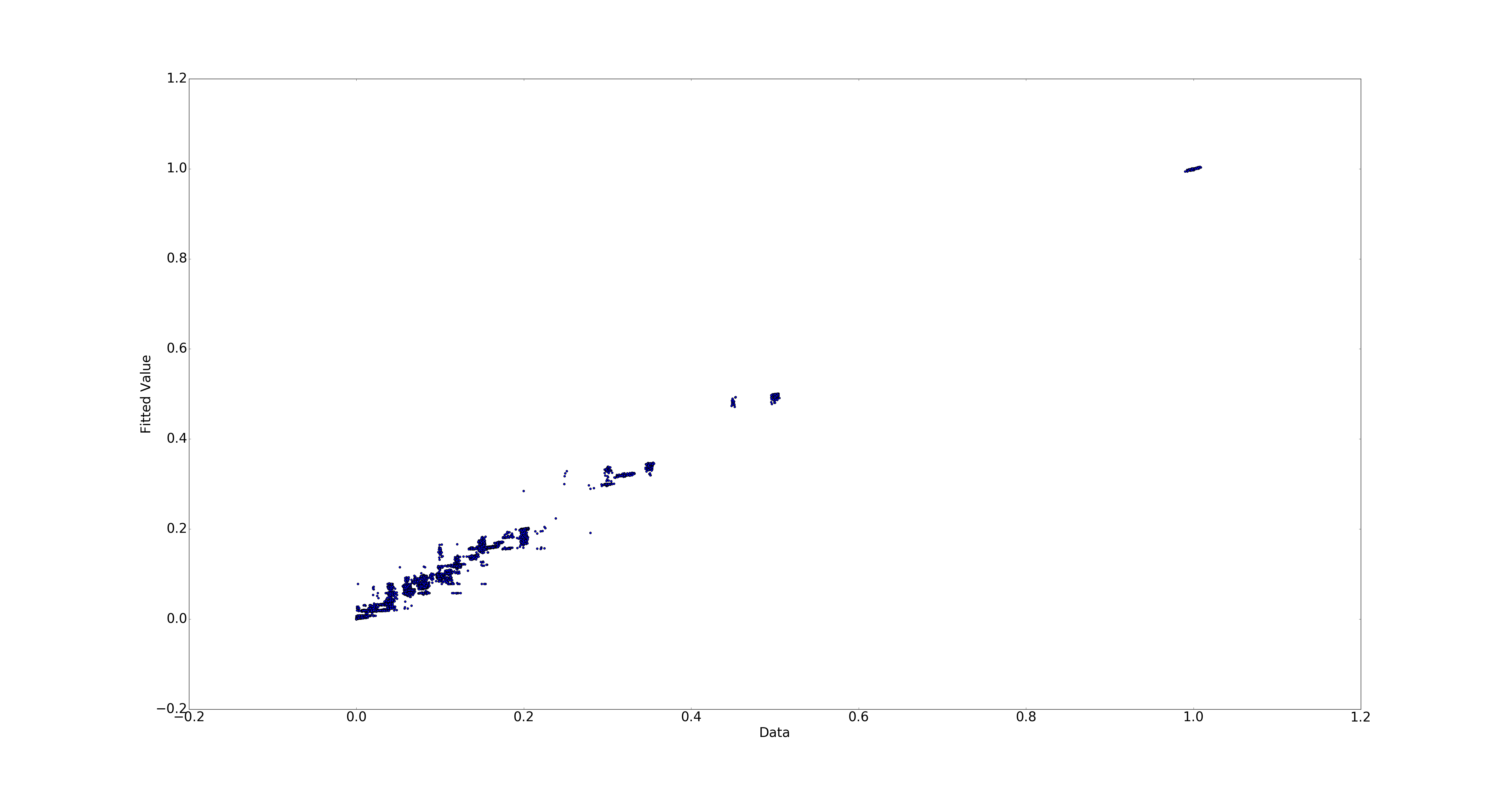


Figure 5-Fitted value vs Actual for Random Forest ntree=40 depth=10

**Tree number= 40 , Tree depth= 12 RMSE= 9.2266126707e-05**

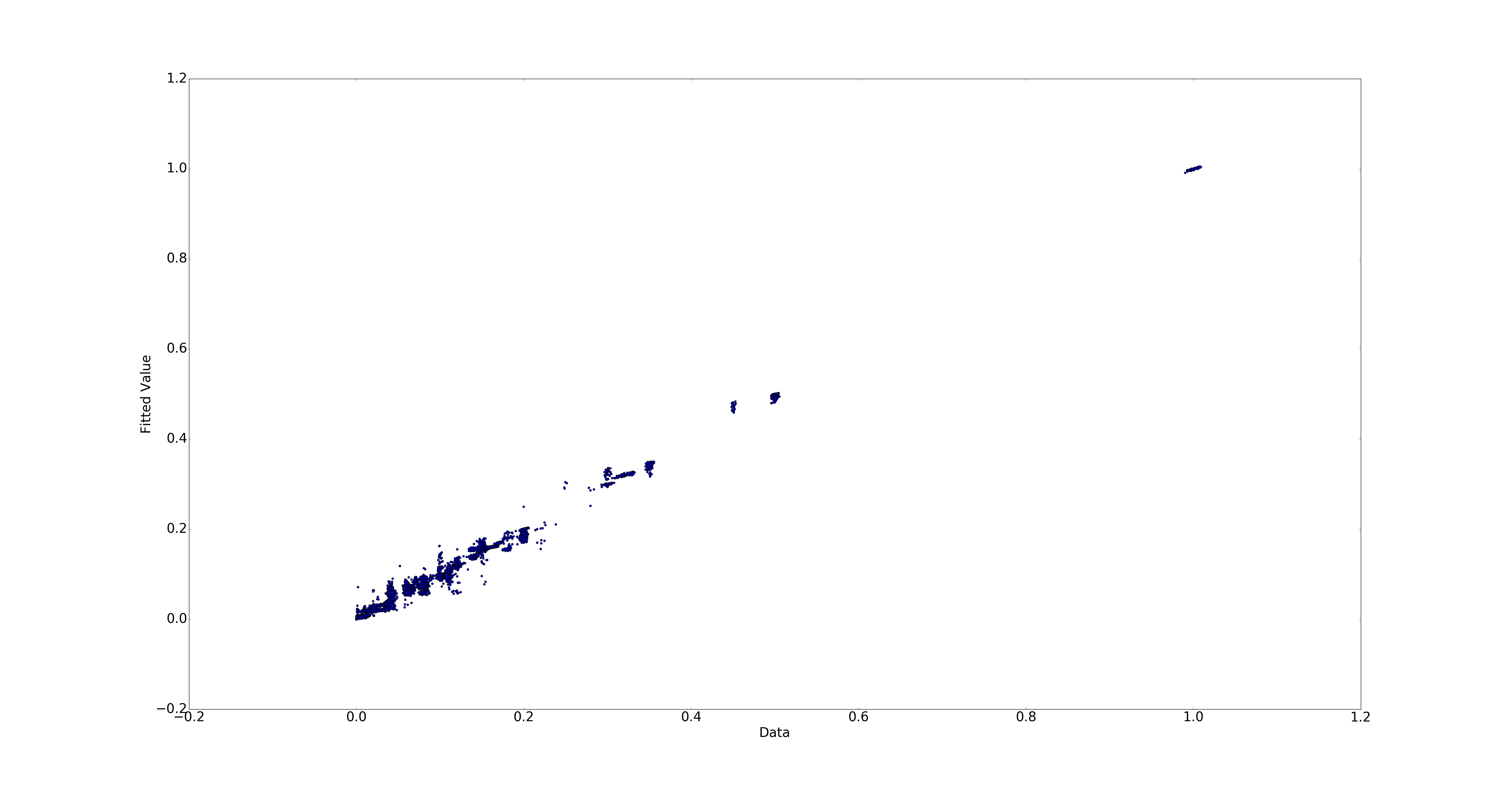


Figure 6-Fitted value vs Actual for Random Forest ntree=40 depth=12

**Tree number= 100 , Tree depth= 10 RMSE= 8.68736594831e-05**

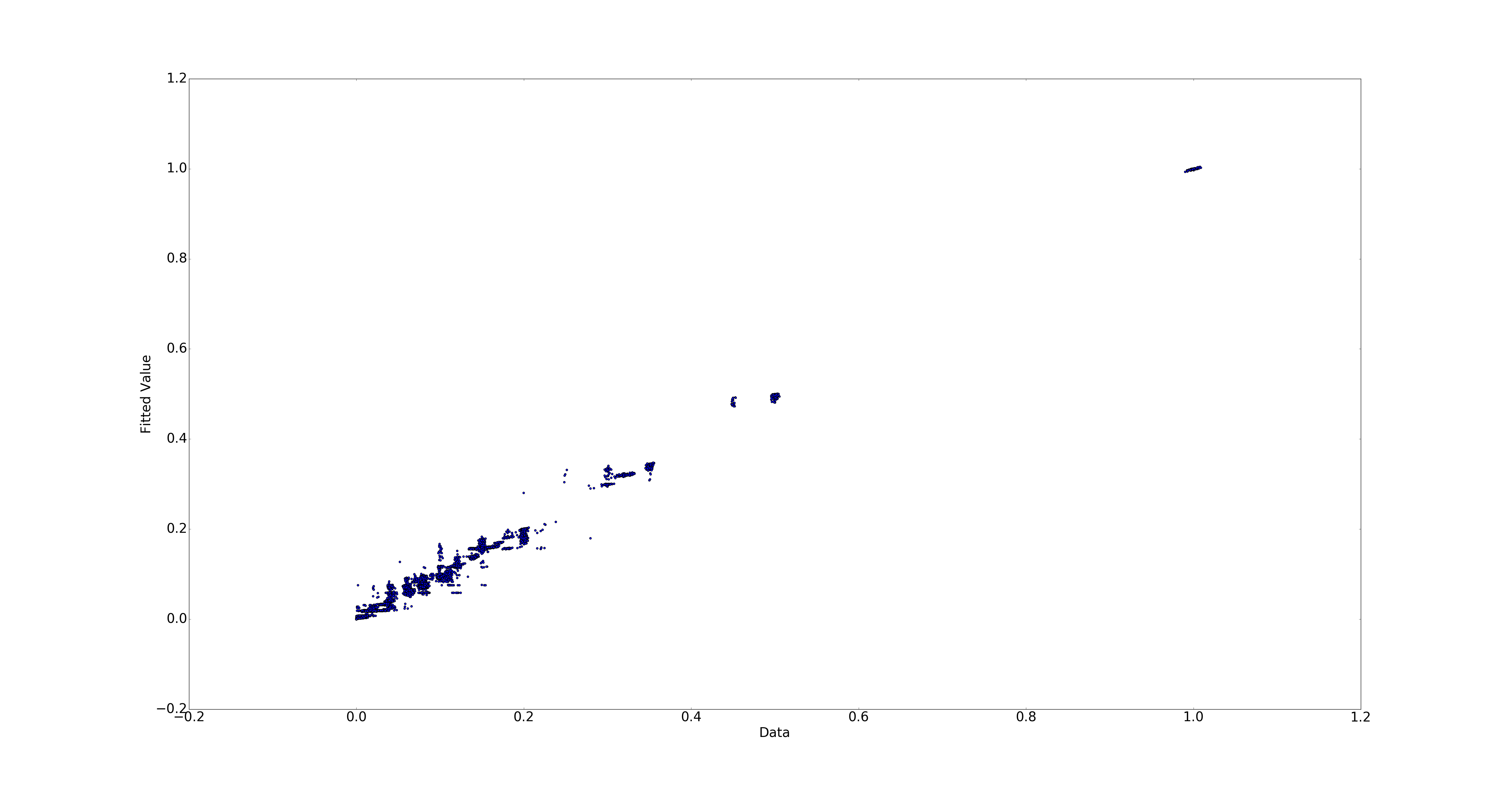


Figure 7-Fitted value vs Actual for Random Forest ntree=100 depth=10

So increasing number of trees and more depth will help us get better result and less RMSE. This is expected because more trees help diminishing returns and reduces the variance. Also deeper trees reduces the bias. However, too much depth will result in overfitting because doing so in reality increases the number of parameters while the number of data points stays constant. Moreover, the computation becomes expensive as we increase these parameters. Using a utilitarian approach, we can assume that the following result is practically optimum:

Tree number= 40 , Tree depth= 10 RMSE= **8.7380954522e-05**

**c) Neural Network:**

For neural networks, the major parameter is the number of hidden layers. Also, for weight optimization, number of iterations are important parameters.

To fix the test result, we set the randon\_state to 1 for all test results. We discover several algorithms and their performance with different number of hidden layers. We test the RMSE with different hidden layer sizes, and we plotted the following:

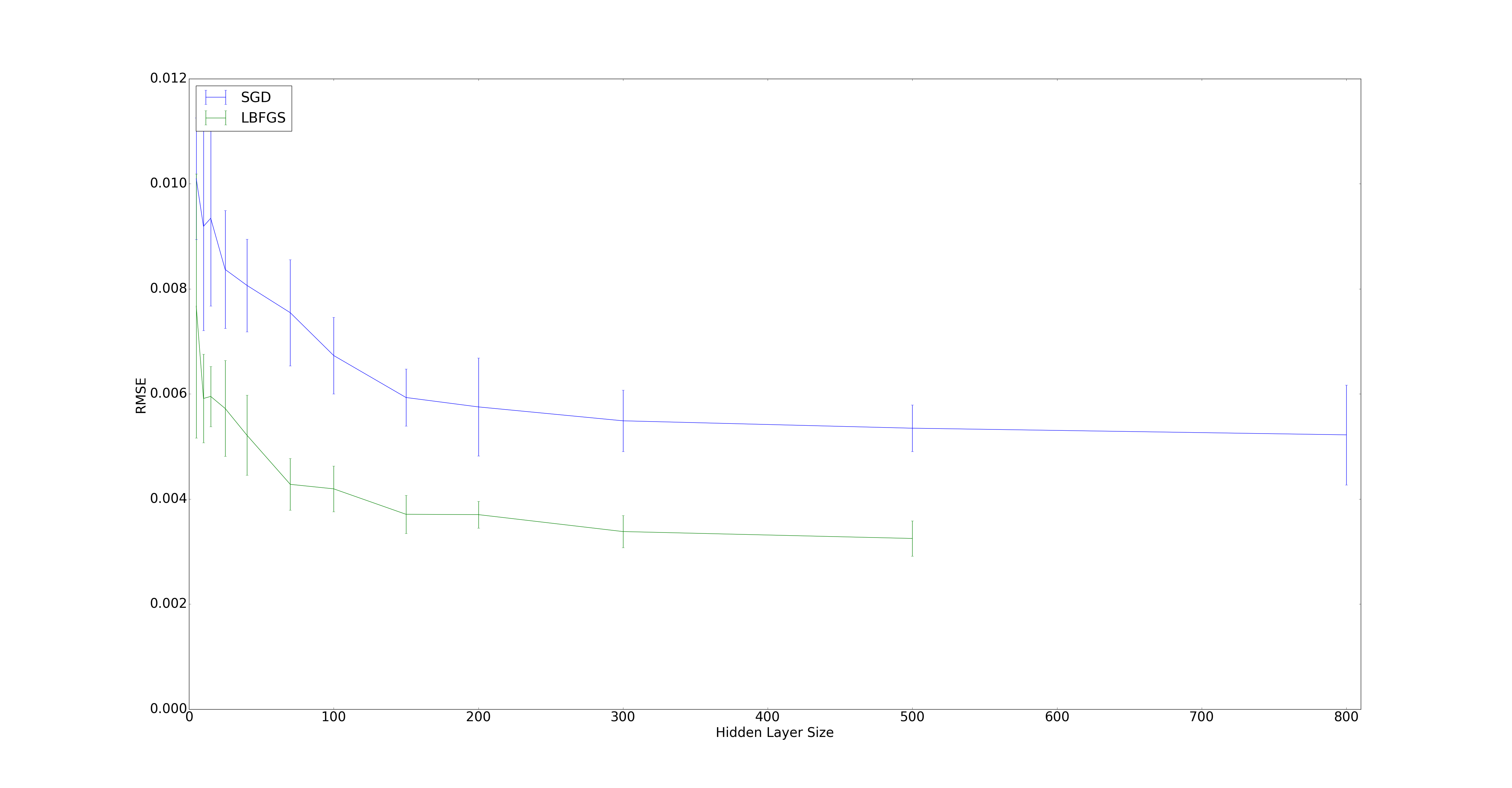


Figure 8- RMSE of SGD and L-BFGS vs Hidden Layer Size

We realized that for size of the hidden greater than 100, the RMSE flattens. Also, the L-BFGS is much more computationally expensive since it took much longer to plot the L-BFGS portion of the above graph than the SGD. However, the L-BFGS provides about %30 less RMSE. For that reason, the L-BFGS is recommended for smaller data sets while the SGD works better and quicker for larger data sets.



First, we calculated the **RMSE** for the different workflows using linear regression. The following graph was plotted:

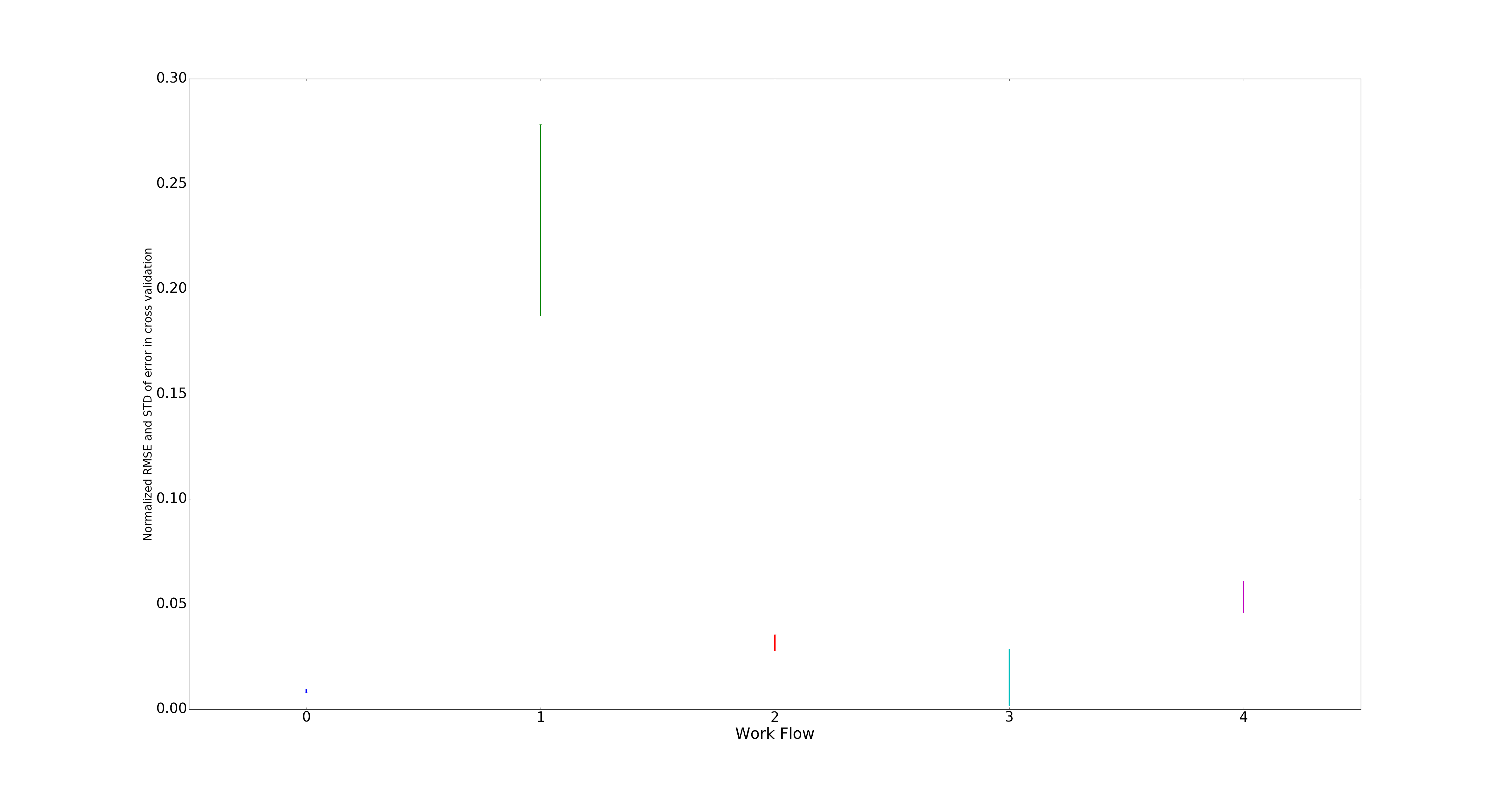
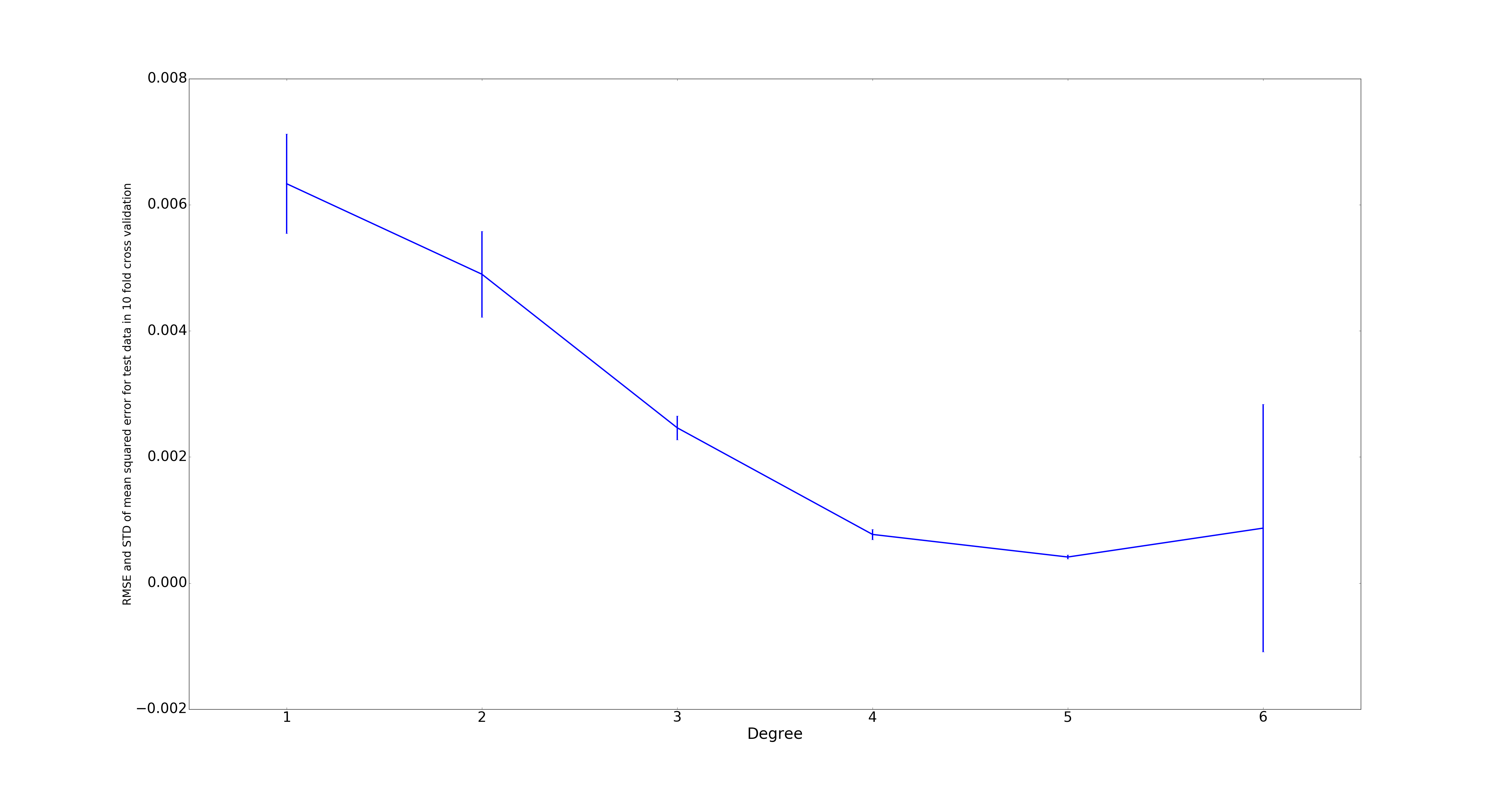


Figure 9- RMSE vs Workflow

Each **RMSE** has been normalized. The length of the bars indicate the standard deviation of the **RMSE** for each workflow. We can see that predicting workflow\_0 has the smallest “overall” error and thus best prediction since the error is almost always small and mostly the same. We can also see that for workflow\_3, the error is also almost always small while it could deviate to zero or 0.05. We can also see that our prediction of file sizes for workflow\_1 would be the worst. Next we try to use polynomial fits for our prediction model. We calculated the RMSE using a 10-fold cross validation polynomial fits with different degrees. The result is plotted below:

 Figure 10- RMSE vs Polynomial Degree

The length of each bar represents the standard deviation of **RMSE** (there are 10 values for 10 different training sets). We can see that initially, increasing the degree of the polynomial would decrease the **RMSE** and therefore better prediction. While increasing the degree of the polynomial results in computational cost, it doesn’t always result in better prediction. We can see that fitting to a 6th degree polynomial worsens our prediction as **RMSE** grows, making a 5th degree polynomial fit as the optimum. The reason is because of overfitting as we saw that during our 10-fold cross validation, the training error was almost zero while the testing error had become significant.

1. **Boston data set**

Now we do the same thing we did in part 2 here:

Here are the results

Model : least square error – linear regression

RMSE= 4.861832

Model : Random Forest ntree=20

RMSE= 3.411698

Model : Neural Network

RMSE= 23.41194

Feature Ranking:

CRIM RMSE=8.603689

ZN: RMSE=8.615033

INDUS RMSE= 8.080178

CHAS: RMSE= 9.098518

NOX: RMSE= 8.36158

RM: RMSE= 6.65742

AGE: RMSE=8.534641

DIS: RMSE=8.951091

RAD: RMSE=8.563641

TAX: RMSE= 8.160803

PTRATIO RMSE= 7.966722

B: RMSE= 8.704385

LSTAT RMSE= 6.228161

Now we model the dataset in polynomial only using the RM and LSTAT which are the most informative features. Here you can see how RMSE changes with the degree of the polynomial:

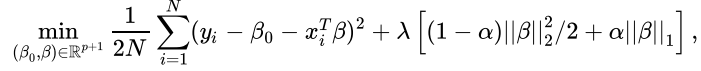
|  |  |
| --- | --- |
| degree | RMSE |
| 2 | 4.696783 |
| 3 | 4.999404 |
| 4 | 4.913379 |
| 5 | 8.356929 |
| 6 | 4.979818 |

Again as can be seen the RMSE saturates for about 4.7 as the degree of poly goes up and this means that for this dataset lower degrees of polynomial fits better. And after 3 and 4 degrees we have overfitting occur.

1. **Regularization of the Parameters:**

In this part, we use glmnet package which is designed in Stanford university. Glmnet is a package that fits a generalized linear model via penalized maximum likelihood. The regularization path is computed for the lasso or elasticnet penalty at a grid of values for the regularization parameter lambda.

Gaussian is the default family option in the function glmnet . Suppose we have observations and the xi ∈ Rp , responses yi  ∈ R, i = 1, … , N. The objective function for the Gaussian family is



where l> 0 is a complexity parameter and 0 < α< 1 is a compromise between ridge (α=0 ) and lasso (α=1 ).

The regularization parameter lis a control on your fitting parameters. As the magnitudes of the fitting parameters increase, there will be an increasing penalty on the cost function.

In other words, l is a parameter to control overfitting and underfitting. As illustration, if you chose l=0 you have over fitting, because the function solve the least mean square error and fit a model either linear or polynomial to reach least RMSE. Although, if you chose l too large, underfitting occurs, and the curve does not follow the direction of the points as well is it did before. Therefore, in order to have a good fitting, the l should be chosen not too large or not too small.

1. lamda=0.001 RMSE= 4.873139

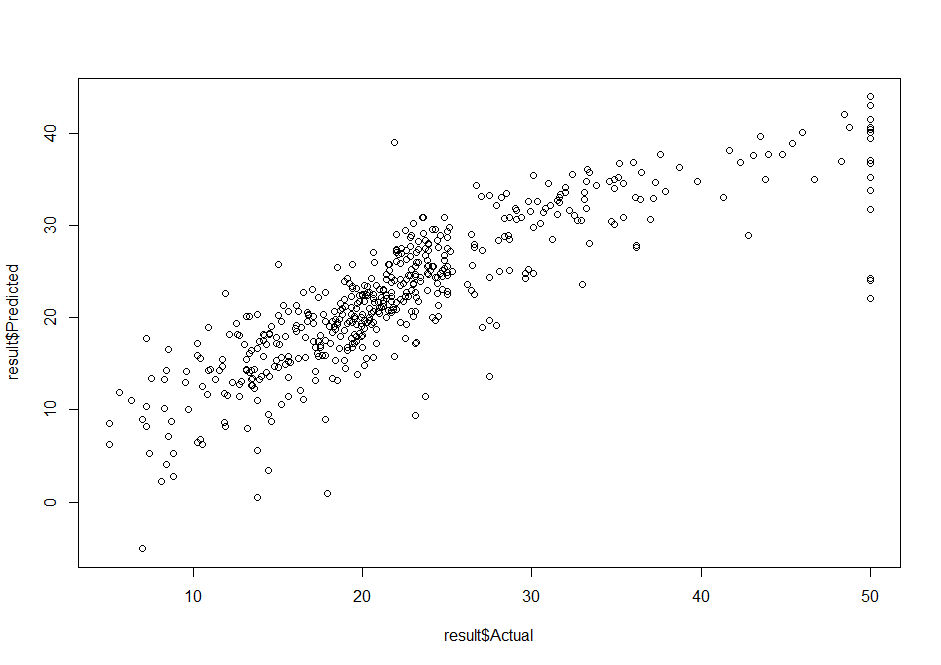


Figure 7-predicted vs actual - lamda=0.001

lamda=0.01 RMSE=4.890635

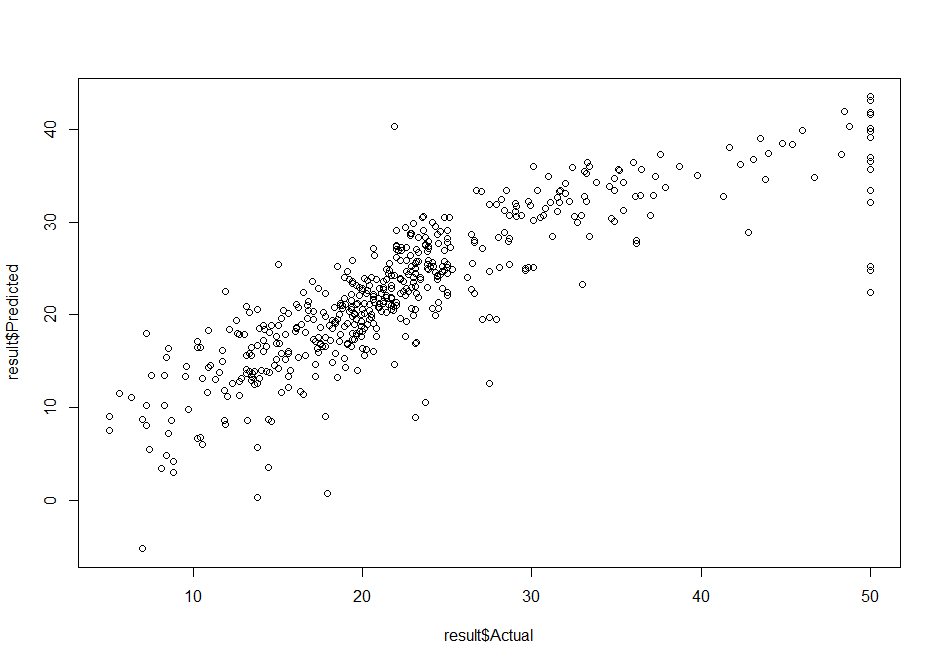


Figure 8-predicted vs actual - lamda=0.01

lamda=0.1 RMSE= 4.83065

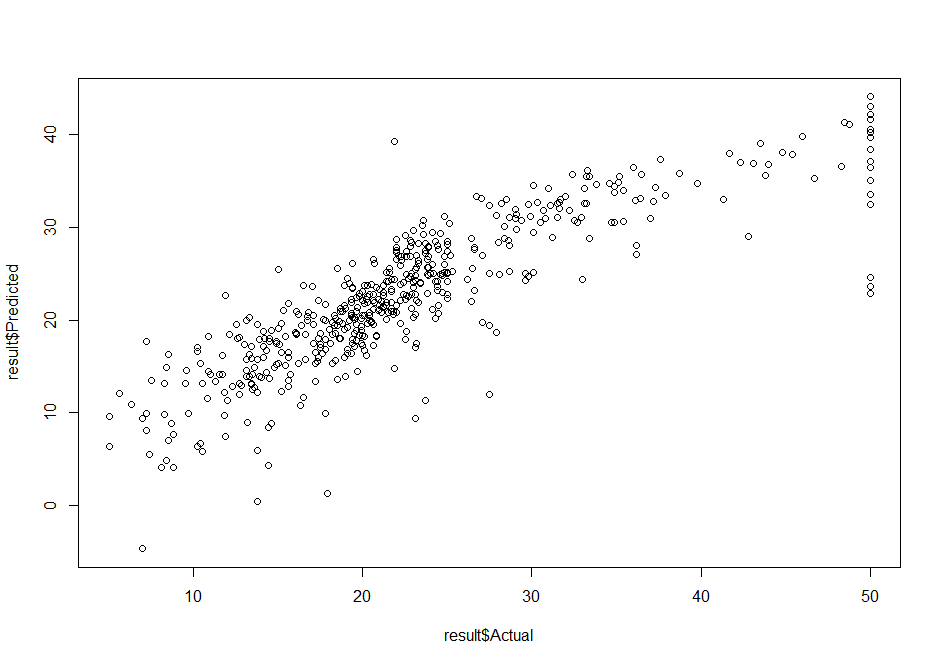


Figure 9-predicted vs actual - lamda=0.1

b)

|  |  |  |  |
| --- | --- | --- | --- |
|  | l=0.1 | l=0.01 | l=0.01 |
| RMSE | 4.928247 | 4.866313 | 4.888762 |