EE219 Project 4 – Report
Regression Analysis

Fangyao Liu 204945018
Xuan Hu 505031796
Yanzhe Xu 404946757
Zhechen Xu 805030074

Content

Introduction:

In this project, we use the method of regression to analyze Network Backup Data. Regression analysis is a statistical procedure for estimating the relationship between a target variable and a set of potentially relevant variables. We need use several regression models to accomplish the goal, such as Linear Regression, Polynomial Regression and Neural Network Regression models. Through using cross validation and regularization, we can also further improve the prediction result from the datasets

The Network backup Dataset is comprised of simulated traffic data on a backup system over a network, which contains the files residing in a destination machine and copies their changes in four hour cycles. In this project, we need to predict the backup size of the traffic depending on the file name and time of backup and we used Linear Regression, Random Forest, Neural Network and Polynomial Regression models. The features contained in data set are as follows:

• **Week index**

• **Day of the week** at which the file back up has started

• **Backup start time**: Hour of the day

• **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

Part a:

i) After we convert the features into scalars, the fitted value-actual value can be shown in figure 1 and the residuals-fitted value can be shown in figure 2.

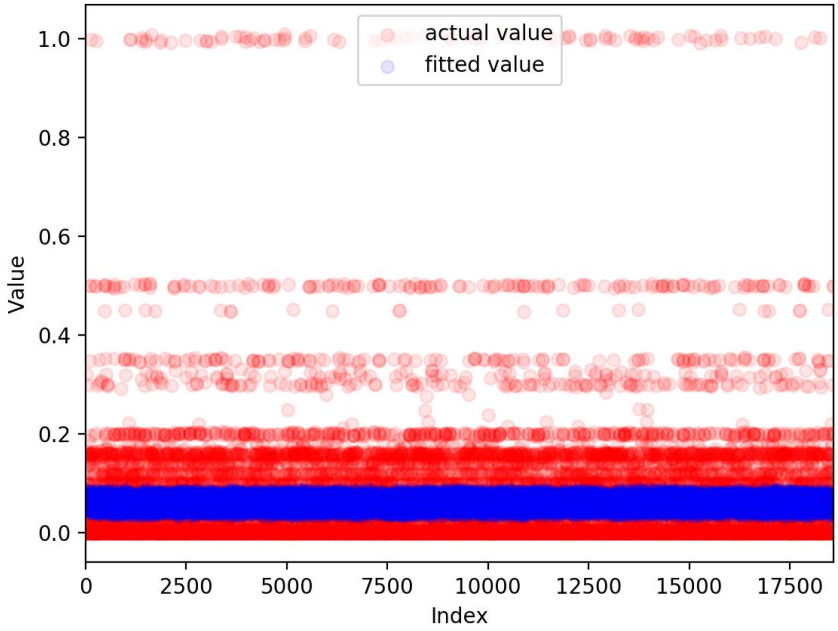


Figure 1 Fitted value - Actual value with linear model

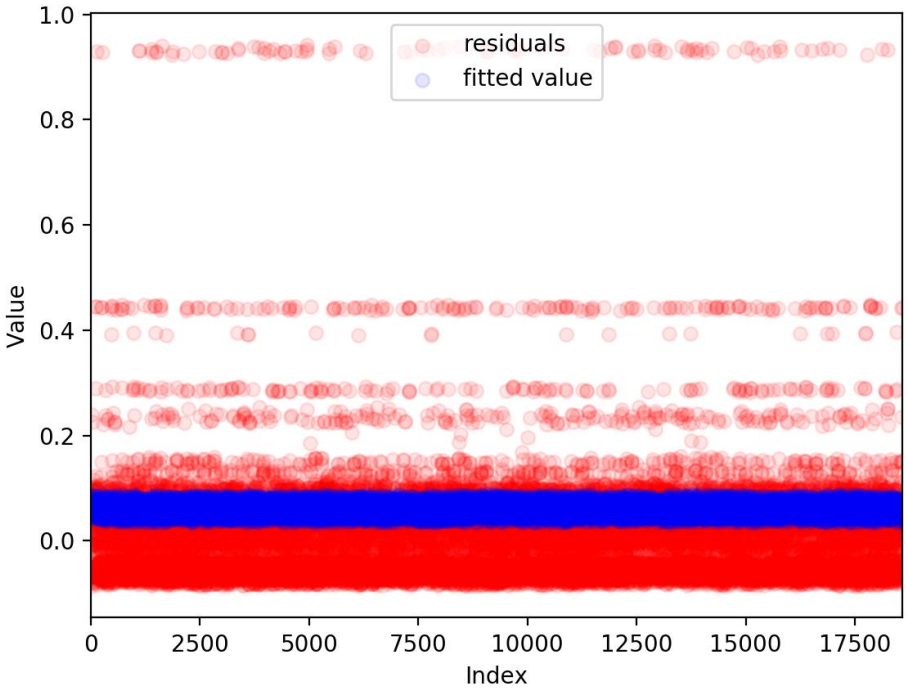


Figure 2 Residual value - Fitted value with linear model

From these two graphs, it is easy to tell that the result is not good. If we predict the back size precisely, the fitted value should be very close to actual values so that the line should lie on y1=y2 which means these two kinds of points should be covered by each other. However, according to the first graph, the line looks like they are not same at all, not even close. This means that no matter how actual value changes, the predicted value remain the same. Thus, it is pretty bad performance. The residual value is the difference between actual value and fitted value. Therefore, the second graph looks similar to the first one. In addition, the average test RMSE is 0.1036.

ii) Considered the fact that there are 5 features that we used to predict back-up size. It is possible that these 5 features can have different contribution to prediction since they have different variance and mean. To eliminate this factor, one way is to standardize those features so that they will contribute to prediction equally. The result of standardize features can be shown in figure 3.

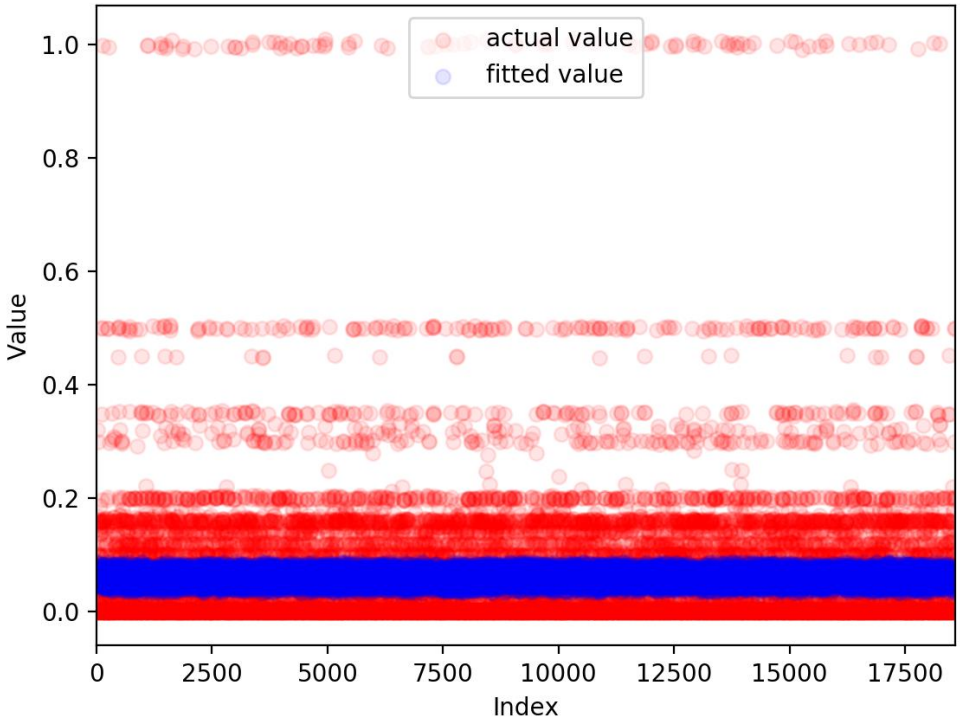


Figure 3 Fitted value - Actual value with linear model(Standardize features)

From graph above, it is obvious that the difference between this and figure 1 is too small to notice. And the fact is that the test RMSE is almost the same as the result of directly using scalar data. The performance of the model doesn't get improved at all. Maybe standardization of features is not the main factor that influences the result.

iii) As mentioned in question ii, the influence of each feature might be different on the result. So, it might be useful to find the most important features in the prediction. It will help us understand what will contribute to the result most and what we should pay less attention to. By using f_regression and mutual information regression measure, we can get two array which show the score of each feature. In f_regression, the result is [0.008, 38.816, 150.741, 26.139, 25.320]. From the result, the most important 3 features are backup start time, day of week and workflow id. In mutual information regression measure, the result is [0.0025, 0.2251, 0.2340, 0.7139, 0.4927]. This result shows that the most important 3 features are workflow id, filename and backup start time. Although these 2 measures give different result, it is obvious that both of them show that the first feature-week is much smaller than other features. Besides, the second feature and fifth feature have same magnitude and their difference is just usual compared to the difference with first feature. In conclusion, workflow id and backup start time are the most 2 important features and week is least important feature while the other two feature share similar importance to prediction.
Then we use [day of week, backup start time, workflow id] to predict the backup size, the result is shown in figure 4.

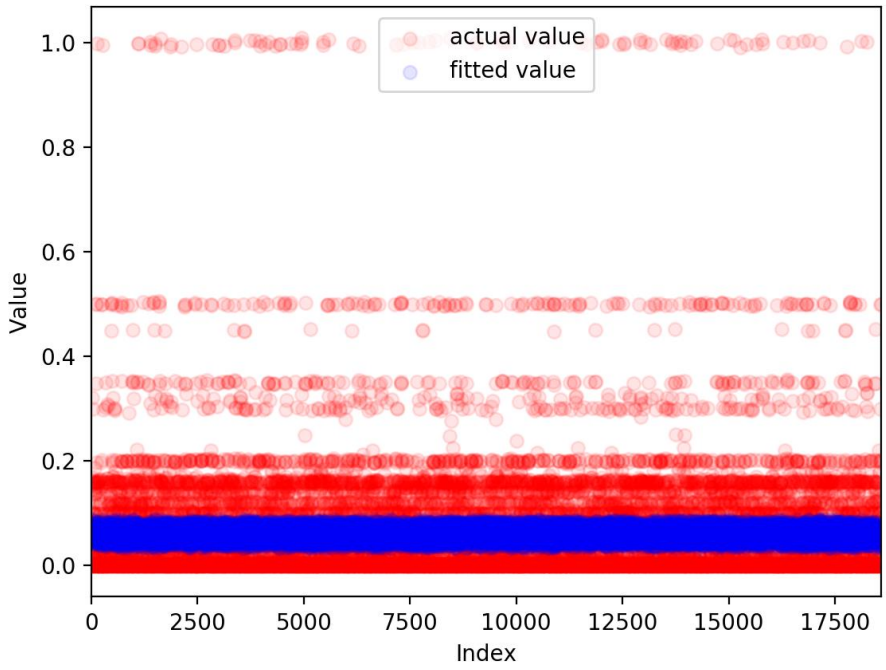


Figure 4 Fitted value - Actual value with selected features

The average RMSE is a little smaller than before, but the difference is too small to ignore. From the graph, the performance of selected features doesn't get significant improvement and the line still looks like y=c. In other words, the most important features can improve the performance a little, but it can't change the fact that this linear model has a bad performance on predicting this kind of data.

iv) In the first 2 questions, we use scalar to describe the features. This definition of feature may not be exactly right. Thus, we introduce another method of encoding-one-hot-encoding. By using the combinations of different method of encoding 5 features, there are 32 different kinds of combinations. We use 1 to indicate scalar coding and 0 to indicate one-hot-encoding. And we use 1st bit to indicate week, 2nd bit to indicate day of week, 3rd bit to indicate backup start time, 4th bit to indicate workflow id, 5th bit to indicate filename. For example, 00001 means that we only use scalar encoding on week feature, others are encoded by one-hot-encoding, after applying linear model to each of 32 combinations, the average test RMSE and train RMSE of 32

combinations can be shown in figure 5.

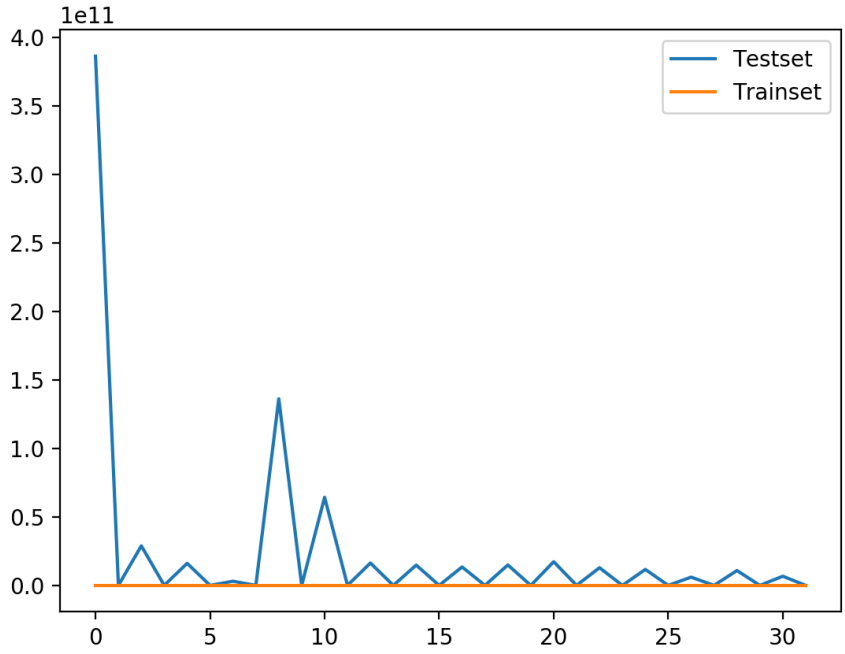


Figure 5 RMSE of trainset and testset of 32 combinations

From the graph above, it is interesting that odd number of combination get normal test RMSE which is less than 1 and close to 0. Nevertheless, all of even number of combination has extremely large test RMSE(>>1). The difference between even and odd number is whether we use one-hot-encoding on week feature or not. That means if we use one-hot-encoding on week, the RMSE will become incredible large. But when we set the shuffle parameter to be true and then the result shows in figure 6.

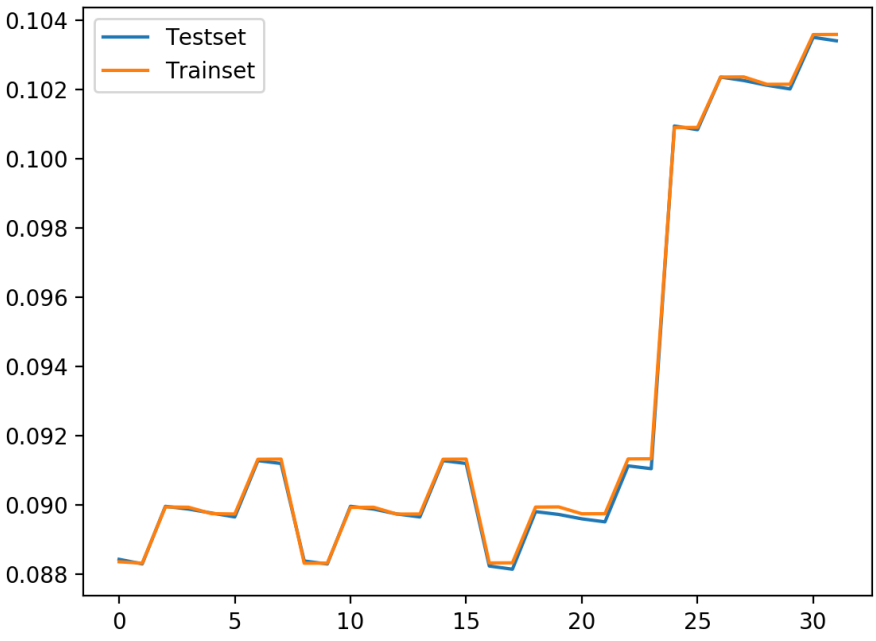


Figure 6 RMSE of trainset and testset of 32 combinations(shuffle=True)

From Figure 6, all of the RMSE of 32 combinations are reasonable now. The reason of this phenomenon might be that the change of choice of K-Fold. If we use shuffle=False which means that the testset and trainset are successive. The difference between i and i+1 point is very small while the backup size may be much larger compared to the input difference. Then the predicted value will be much larger due to this extremely large rate of output to input. However, if we select data point randomly, the similarity of these data points will be much smaller and then the predicted value will be normal. And the two successive points always share same week, then we can't use one-hot-encoding on week feature because one-hot-encoding will enlarge the change of input if shuffle=True.

The least RMSE appears at combination number 17 which means we use one-hot-encoding on 2-4 features. This is exactly the same 3 features as what we conclude from f_regression. By encoding these 3 most important features, the main characteristic will be highlighted so that the best result will be achieved. That is why the least RMSE appears at combination number 17.

v) In question iv, we found that there might be some significant increase in some specific combinations, to avoid this happen or decrease the influence of the one-hot-encoding on week, there are 3 linear models we can use to improve the performance- Ridge, Lasso, ElasticNet. By applying these 3 methods to each of 32 combinations while trying different parameter, we can find the best combination and most efficient parameter which fits one particular model.

From the result, the best combination of Ridge model will still be number 17, and the alpha is 0.9. The average test RMSE is 0.088. When it comes to Lasso model, the best combination will be number 20 with alpha=0.01 and the average test RMSE is 0.099. In ElasticNet model, we fixed the l1_ratio to 0.5 and then sweep alpha from 0.01 to 1. Then the best combination is number 20 with alpha=0.01 as well and the average test RMSE is 0.094. It is interesting that in Lasso and ElasticNet model, the best combination is number 20 which is even number. That means these two model eliminate the influence of one-hot-encoding on week feature. ElasticNet and Lasso model share one similar property that they all includes $|| \, ||_1$ into their model while Ridge doesn't. This might be the explanation to the sudden increase of RMSE when we apply one-hot-encoding on week feature. This dataset doesn't fit for linear model since the data might not be linearly distributed. By taking norm 1 into prediction, the result of prediction might be more persuasive and general.

Part b

i) Using random forest classifier with default parameters to predict the backup size.

Test RMSE is 0.060132685841569347.

Train RMSE is 0.060470422449299843.

Out Of Bag error is 0.33645064586028328.

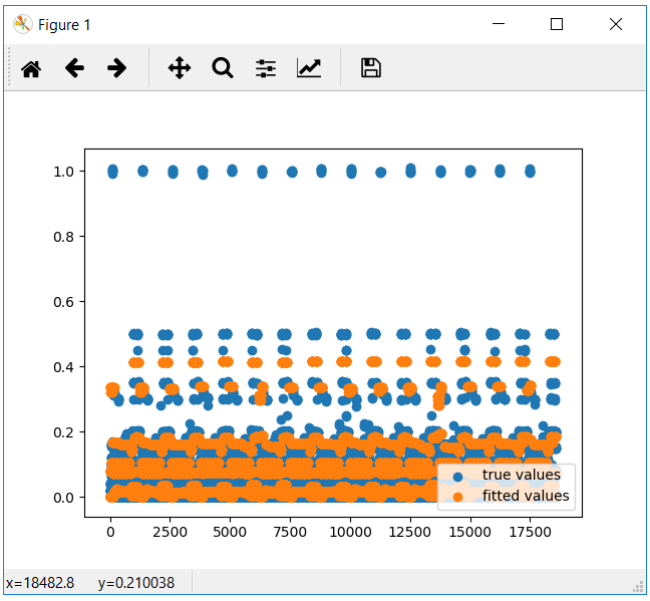


Figure 7 true value – fitted value plot

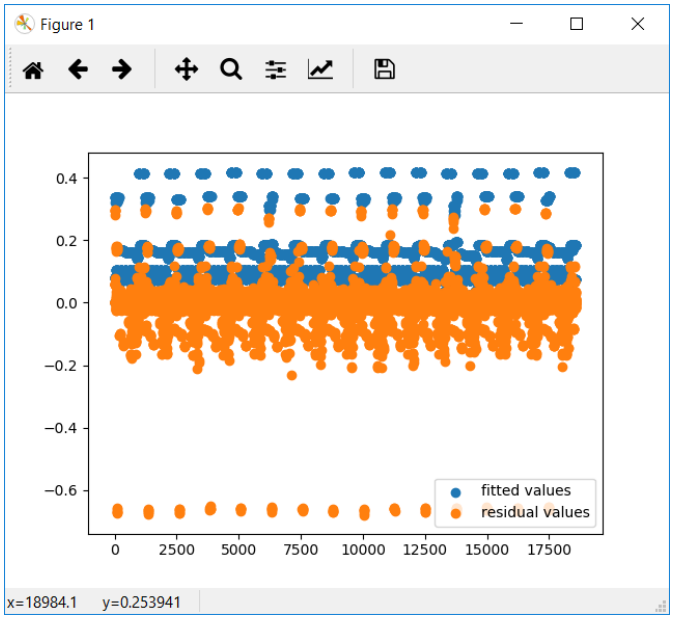


Figure 8 residual value – fitted value plot

The predict plot implies that using default parameters, random forest didn't do a very good job.

ii) In this question, we are asked to sweep through parameters to find optimal number of trees and maximum features. We can see from the plots that when we select maximum features above 4, number of trees above 25, the error keeps at a low and stable state. These plots make sense because with more trees and more features, each decision tree has more information to split data points and forest has a larger number of trees to help classify. In order to avoid unnecessary computation, the best parameter for random tree are selected as: maximum features = 4, number of trees = 25

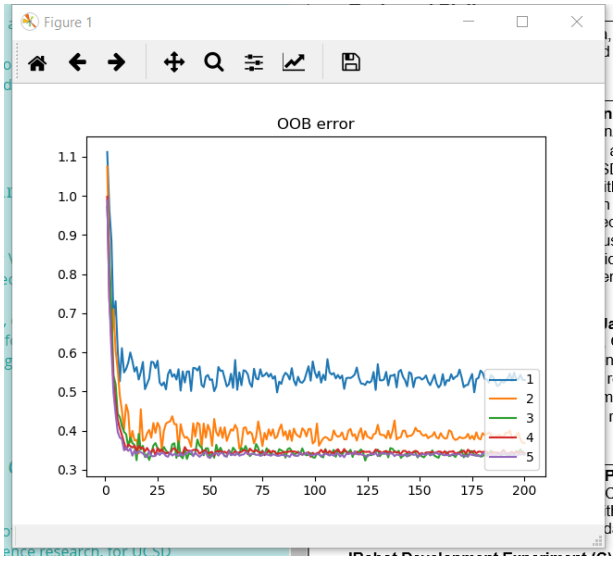


Figure 9 number of trees – OOB error plot

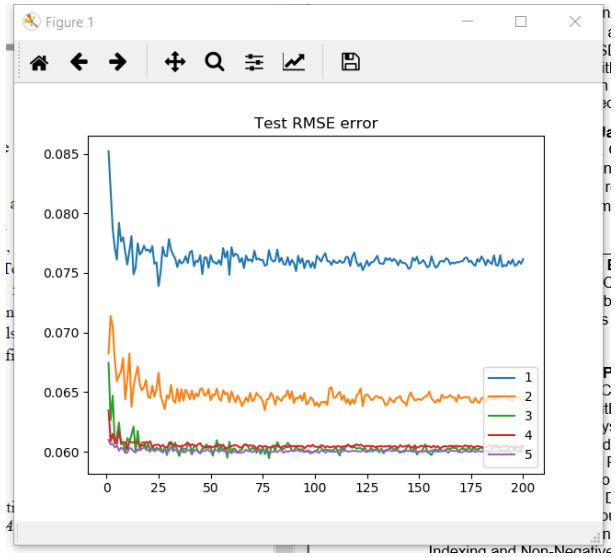


Figure 10 number of trees – Test RMSE plot

iii) Choose maximum depth as the parameter that we are going to test. According to the result above, set number of trees to be 25, number of features to be 4. Sweep maximum depth from 1 to 20. It can be observed that when maximum depth is above 9, the error stays at a low and stable state. Thus, optimum maximum depth is selected as 19.

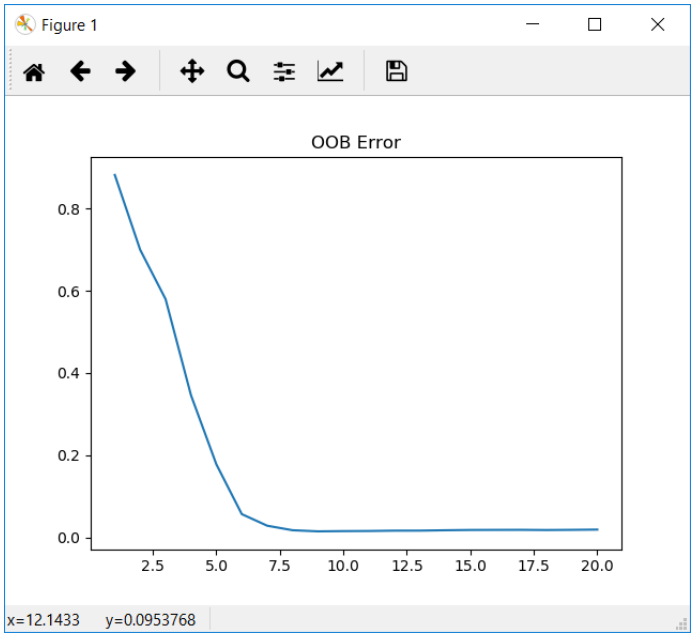


Figure 11 number of trees – OOB error plot

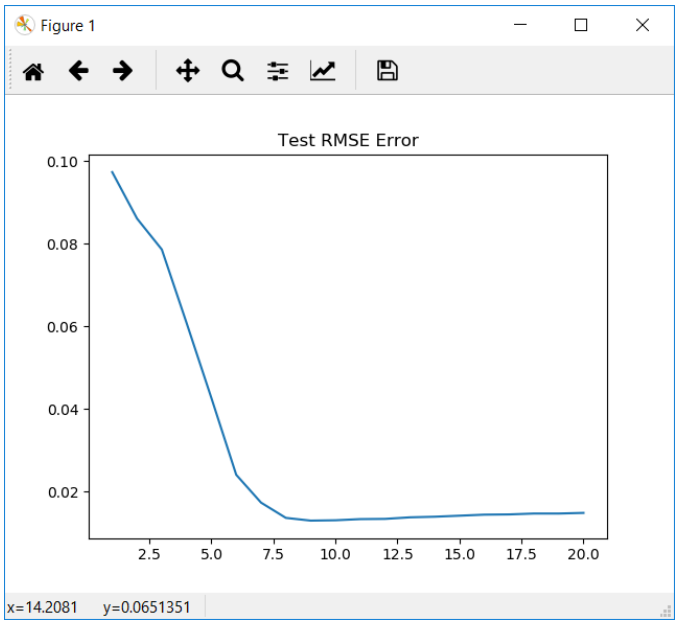


Figure 12 number of trees – Test RMSE plot

iv) The best Random forest parameter would be number of trees to be 25, number of features to be 4, maximum depth as 10. It can be observed from the plot that almost all the predicted values match ground truth values. Residual values are also low (around 0)

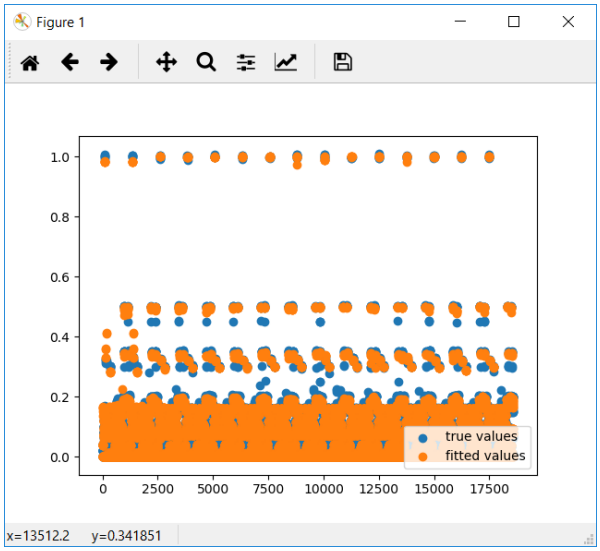


Figure 13 true value – fitted value plot

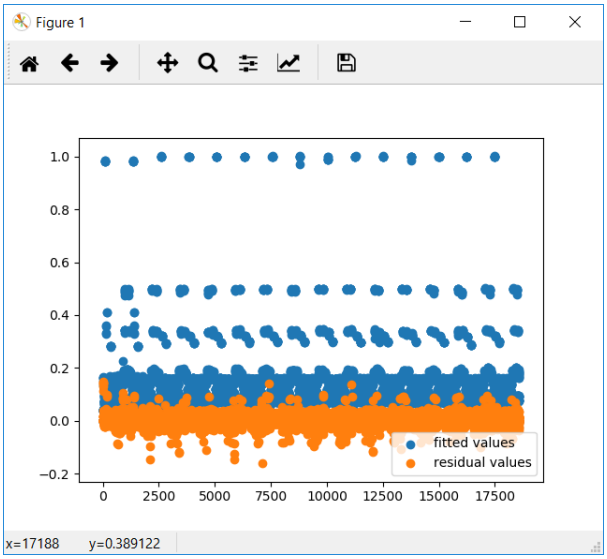


Figure 14 residual value – fitted value plot

$$feature\ names = [\text{“}week\text{”}, \text{“}day\_of\_week\text{”}, \text{“}backup\_start\_time\text{”}, \text{“}workflow\_id\text{”}, \text{“}filename\text{”}]$$

$$feature\ importance = [0.0027, 0.242, 0.383, 0.158, 0.213]$$

$$importance\ ranking: backup\_start\_time > day\_of\_week > filename > workflow\_id > week$$

v) In order to visualize the tree, we choose max depth as 4 to make the tree smaller.

$$feature\ names = [\text{“}week\text{”}, \text{“}day\_of\_week\text{”}, \text{“}backup\_start\_time\text{”}, \text{“}workflow\_id\text{”}, \text{“}filename\text{”}]$$

$$feature\ importance = [1.746e-04, 3.258e-01, 1.239e-01, 2.169e-01, 3.330e-01]$$

$$importance\ ranking: filename > day\_of\_week > workflow\_id > backup\_start\_time > week$$

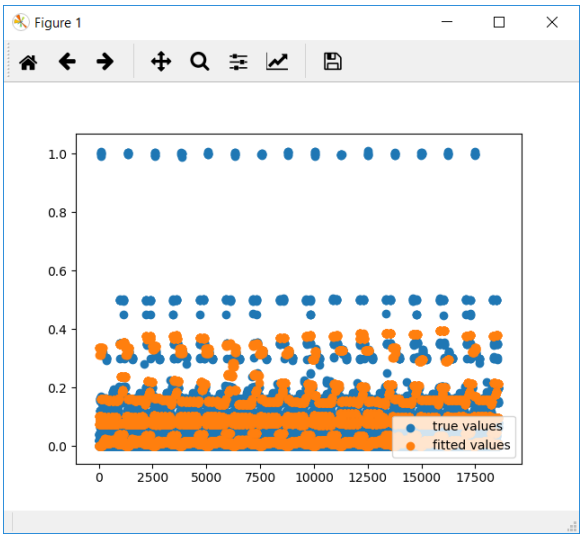


Figure 15 true value – fitted value plot

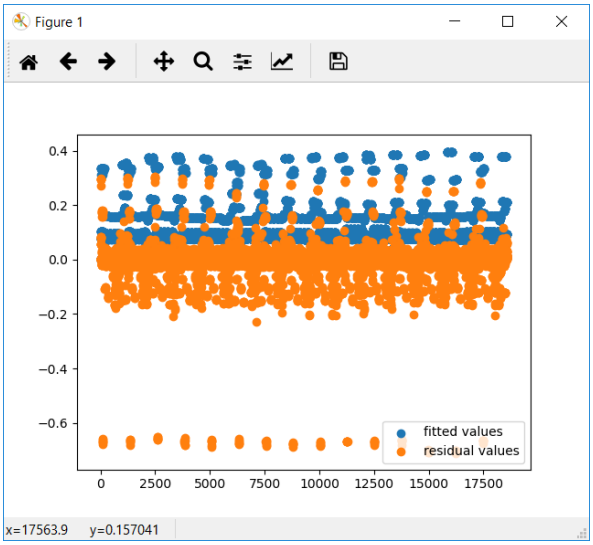
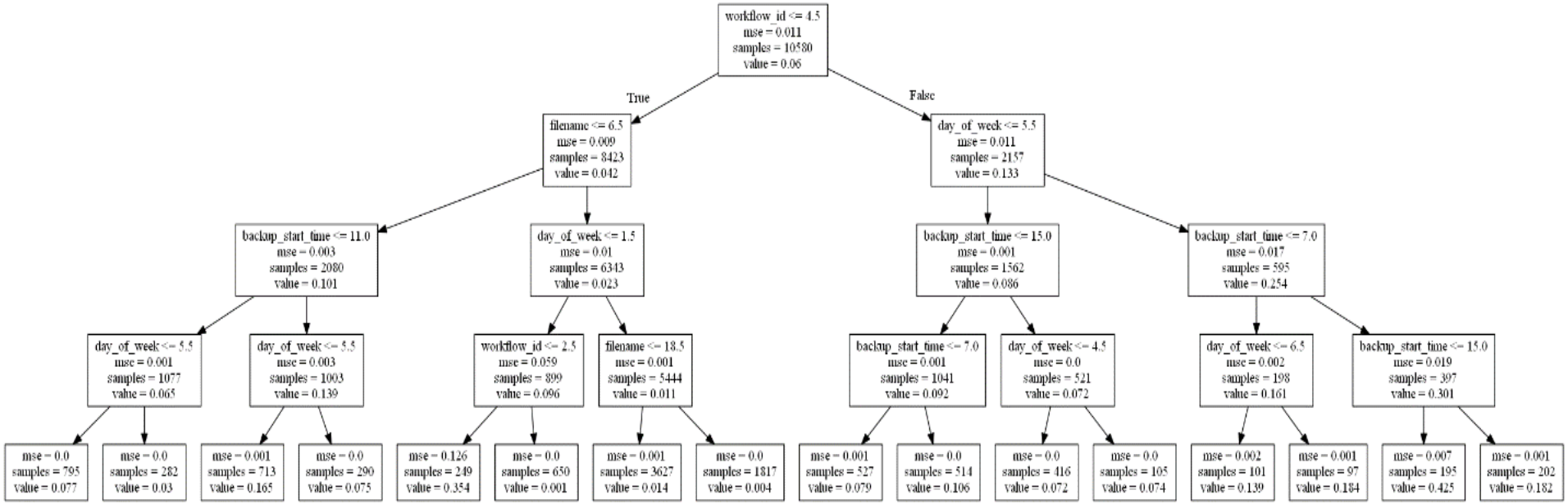
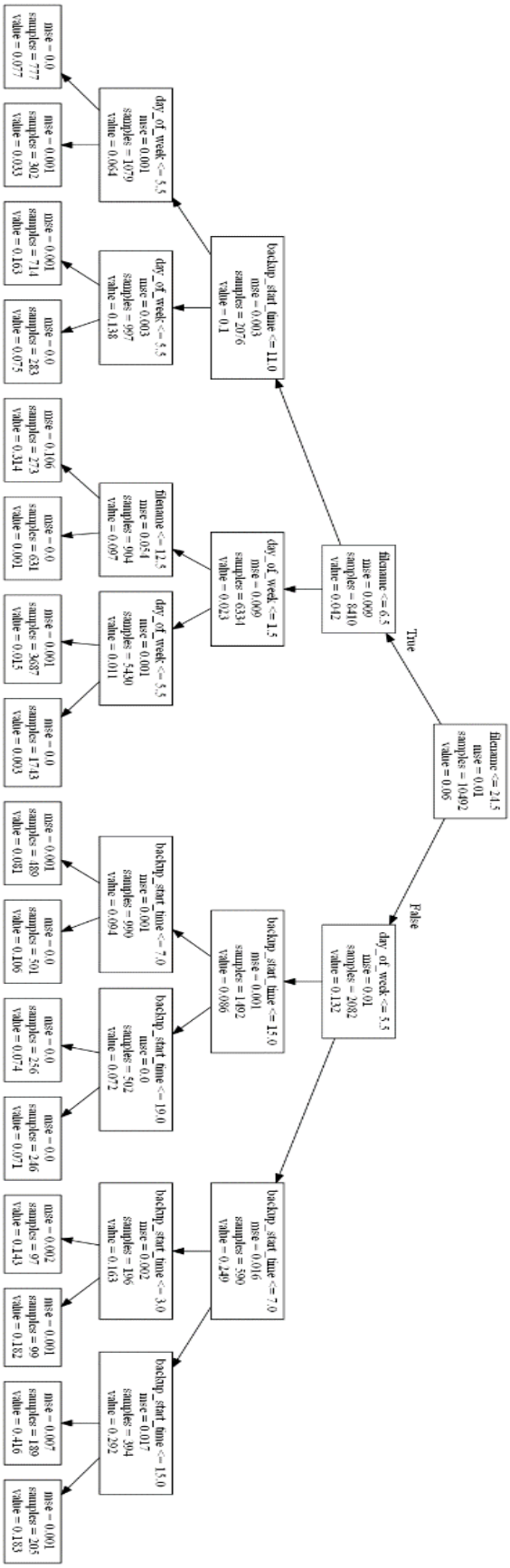


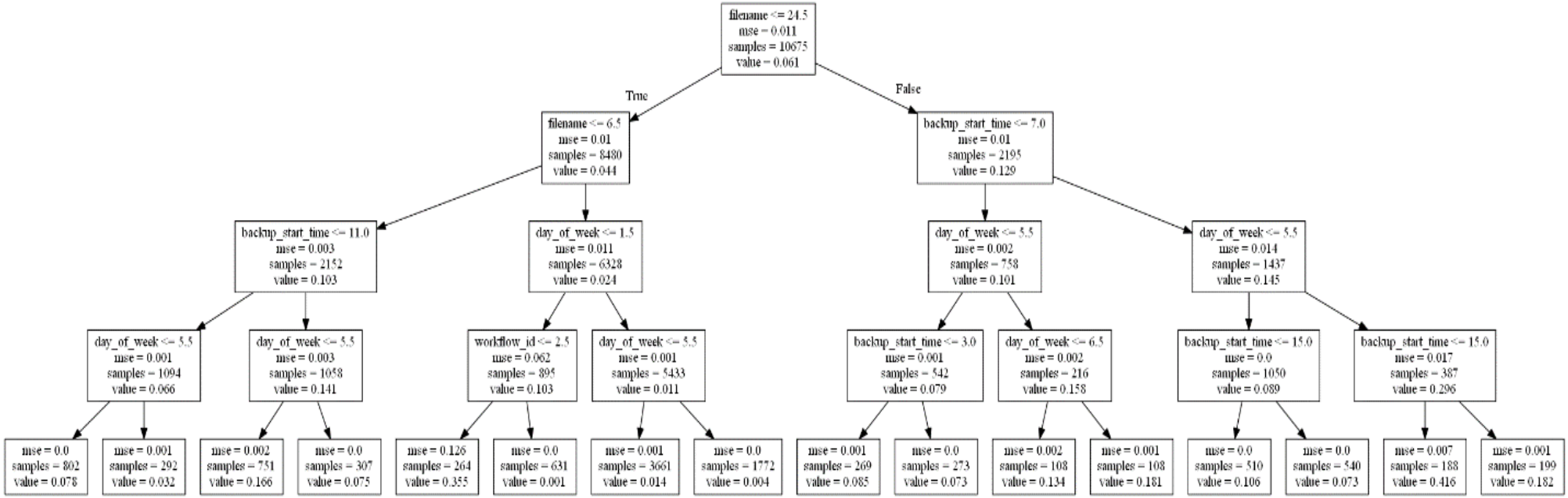
Figure 16 residual value – fitted value plot


workflow_id <= 4.5
mse = 0.011
samples = 10580
value = 0.06
True
False
filename <= 6.5
mse = 0.009
samples = 8423
value = 0.042
day_of_week <= 5.5
mse = 0.011
samples = 2157
value = 0.133
backup_start_time <= 11.0
mse = 0.003
samples = 2080
value = 0.101
day_of_week <= 1.5
mse = 0.01
samples = 6343
value = 0.023
backup_start_time <= 15.0
mse = 0.001
samples = 1562
value = 0.086
backup_start_time <= 7.0
mse = 0.017
samples = 595
value = 0.254
day_of_week <= 5.5
mse = 0.001
samples = 1077
value = 0.065
day_of_week <= 5.5
mse = 0.003
samples = 1003
value = 0.139
workflow_id <= 2.5
mse = 0.059
samples = 899
value = 0.096
filename <= 18.5
mse = 0.001
samples = 5444
value = 0.011
backup_start_time <= 7.0
mse = 0.001
samples = 1041
value = 0.092
day_of_week <= 4.5
mse = 0.0
samples = 521
value = 0.072
day_of_week <= 6.5
mse = 0.002
samples = 198
value = 0.161
backup_start_time <= 15.0
mse = 0.019
samples = 397
value = 0.301
mse = 0.0
samples = 795
value = 0.077
mse = 0.0
samples = 282
value = 0.03
mse = 0.001
samples = 713
value = 0.165
mse = 0.0
samples = 290
value = 0.075
mse = 0.126
samples = 249
value = 0.354
mse = 0.0
samples = 650
value = 0.001
mse = 0.001
samples = 3627
value = 0.014
mse = 0.0
samples = 1817
value = 0.004
mse = 0.001
samples = 527
value = 0.079
mse = 0.0
samples = 514
value = 0.106
mse = 0.0
samples = 416
value = 0.072
mse = 0.0
samples = 105
value = 0.074
mse = 0.002
samples = 101
value = 0.139
mse = 0.001
samples = 97
value = 0.184
mse = 0.007
samples = 195
value = 0.425
mse = 0.001
samples = 202
value = 0.182

Choose the first tree in the forest. Root node is work-flow id. It is not the most important feature in the feature importance. But the first three node corresponds to the most three important features. The explanation of this is that there are 20 trees in the forest, it is likely that most of other trees' first nodes are the most important features, since random forest are predicted by majority.

Try to visualize some other trees.




filename <= 24.5
mse = 0.011
samples = 10675
value = 0.061
True
False
filename <= 6.5
mse = 0.01
samples = 8480
value = 0.044
backup_start_time <= 7.0
mse = 0.01
samples = 2195
value = 0.129
backup_start_time <= 11.0
mse = 0.003
samples = 2152
value = 0.103
day_of_week <= 1.5
mse = 0.011
samples = 6328
value = 0.024
day_of_week <= 5.5
mse = 0.002
samples = 758
value = 0.101
day_of_week <= 5.5
mse = 0.014
samples = 1437
value = 0.145
day_of_week <= 5.5
mse = 0.001
samples = 1094
value = 0.066
day_of_week <= 5.5
mse = 0.003
samples = 1058
value = 0.141
workflow_id <= 2.5
mse = 0.062
samples = 895
value = 0.103
day_of_week <= 5.5
mse = 0.001
samples = 5433
value = 0.011
backup_start_time <= 3.0
mse = 0.001
samples = 542
value = 0.079
day_of_week <= 6.5
mse = 0.002
samples = 216
value = 0.158
backup_start_time <= 15.0
mse = 0.0
samples = 1050
value = 0.089
backup_start_time <= 15.0
mse = 0.017
samples = 387
value = 0.296
mse = 0.0
samples = 802
value = 0.078
mse = 0.001
samples = 292
value = 0.032
mse = 0.002
samples = 751
value = 0.166
mse = 0.0
samples = 307
value = 0.075
mse = 0.126
samples = 264
value = 0.355
mse = 0.0
samples = 631
value = 0.001
mse = 0.001
samples = 3661
value = 0.014
mse = 0.0
samples = 1772
value = 0.004
mse = 0.001
samples = 269
value = 0.085
mse = 0.0
samples = 273
value = 0.073
mse = 0.002
samples = 108
value = 0.134
mse = 0.001
samples = 108
value = 0.181
mse = 0.0
samples = 510
value = 0.106
mse = 0.0
samples = 540
value = 0.073
mse = 0.007
samples = 188
value = 0.416
mse = 0.001
samples = 199
value = 0.182

It is found out that some trees have the most important features as their first node. But the trees that don't still outnumber (2:1). I believe the reason is that when max_depth = 4, our classifier is not doing a good prediction work here, shown in the "fitted value vs true values" plot. Therefore, most of the trees in forest don't take the most important features as the first node. If we can select the optimum max_depth, most of the trees will take the most important features as the first node.

Part c

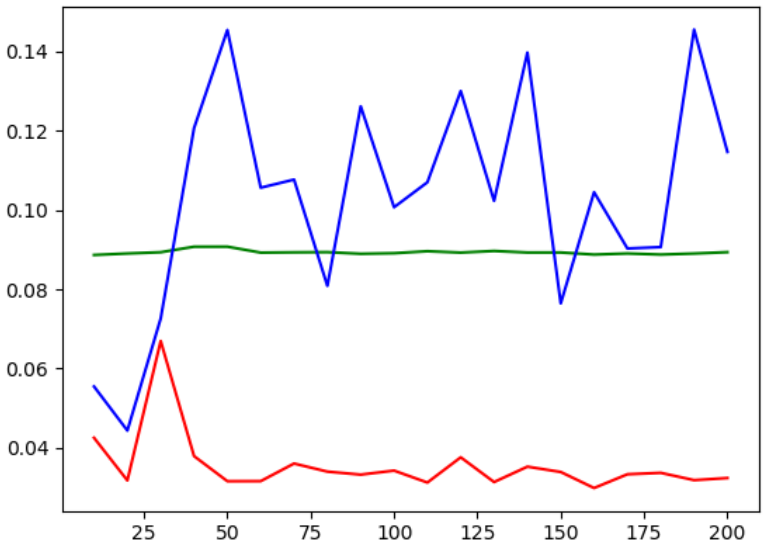


Figure 17 number of hidden units – RMSE plot

In figure 17, blue line is tanh; green line is logistic; red line is relu.

Best combination is activity function = relu and hidden units = 20. When hidden units equals to 20, activity function equals to relu, we can get best combination with RMSE less than 0.01.

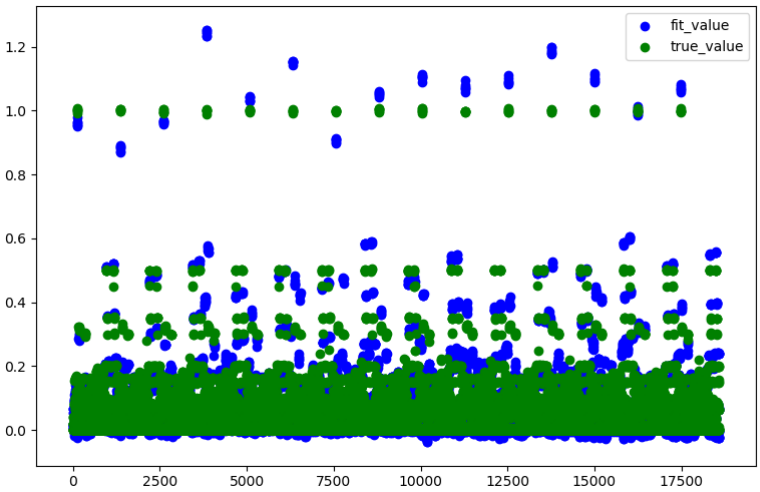


Figure 18 fitted value vs. true value plot

The picture shows we get a good performance of Neural Network with best combination. Fit values and true values distribute along line y=x.

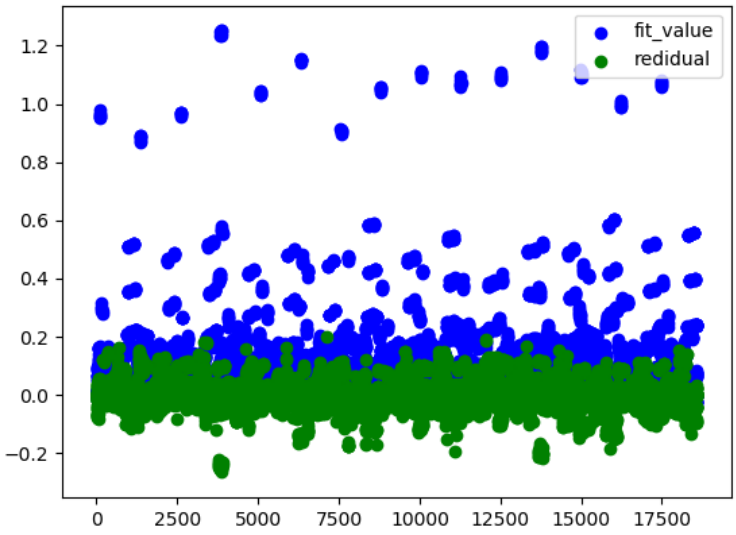


Figure 19 residual value vs. fitted value plot

Part d

The RMSE of ordinary linear regression is 0.1036. And the RMSE of each workflow are 0.035, 0.147,0.04,0.007,0.08. Expect workflow1, all other's rese is lower than 0.1036. In general, we could say the result improved a lot by predicting backup size of each workflow.

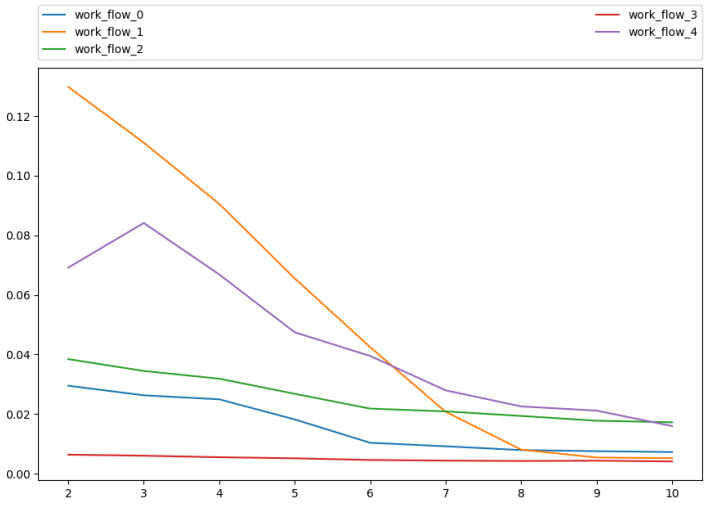


Figure 20 Train RMSE for polynomial (a)

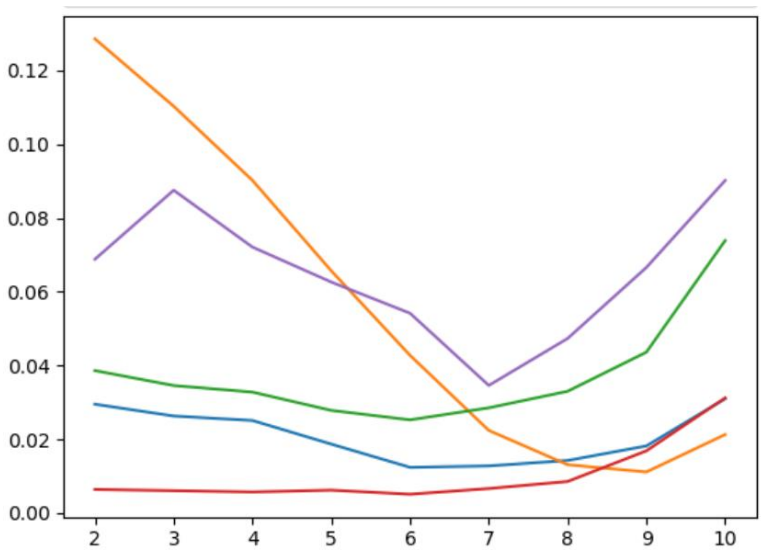


Figure 21 Train RMSE for polynomial (b)

Except workflow1, threshold of degree of all other four workflow is 7. The generalization error of model gets worse when degree beyond this number. As for workflow1, threshold is 7.

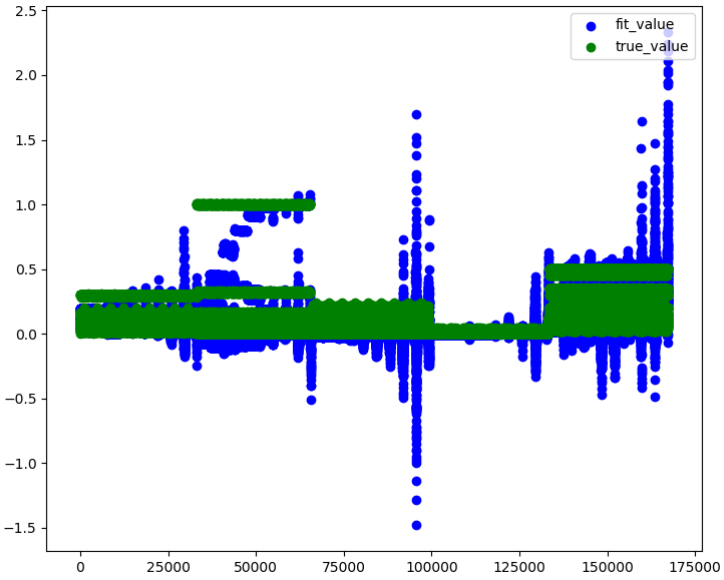


Figure 22 true value – fitted value plot

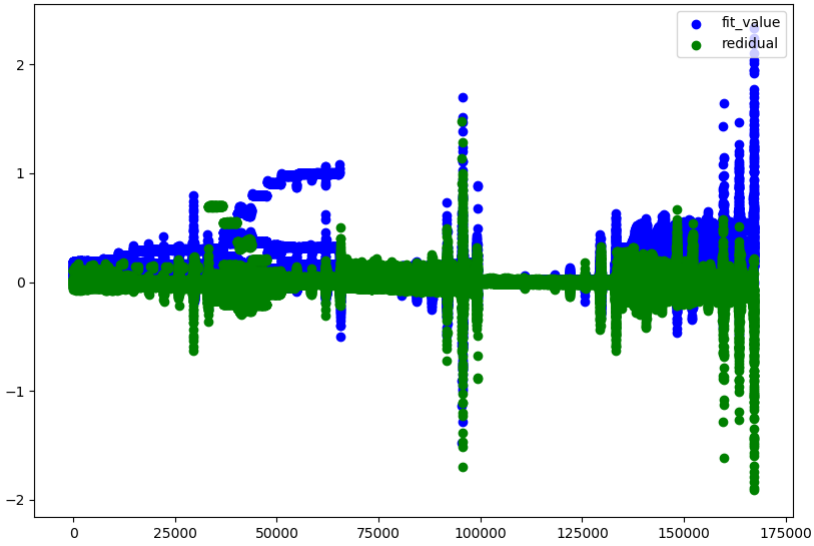


Figure 23 residual value – fitted value plot

Cross validation can avoid overfitting of the model. In cross validation, when we get a low error of train dataset, but a high error of test dataset, it means the model has a high complexity and we need to rebuild the classifier. When this event happened, we need to split dataset again and fit new model to get better prediction.

Part e

In part e, we are using KNN to predict labels. First we select n_neighbors as the one to sweep. We sweep n_neighbors from 1~50 and plot train error and test error in the plot. The plot makes sense. When n_neighbors is small, the classifier overfit and we have a low training error and a high testing error. When n_neighbors is high, the classifier underfit, so both of the training and testing error are high.

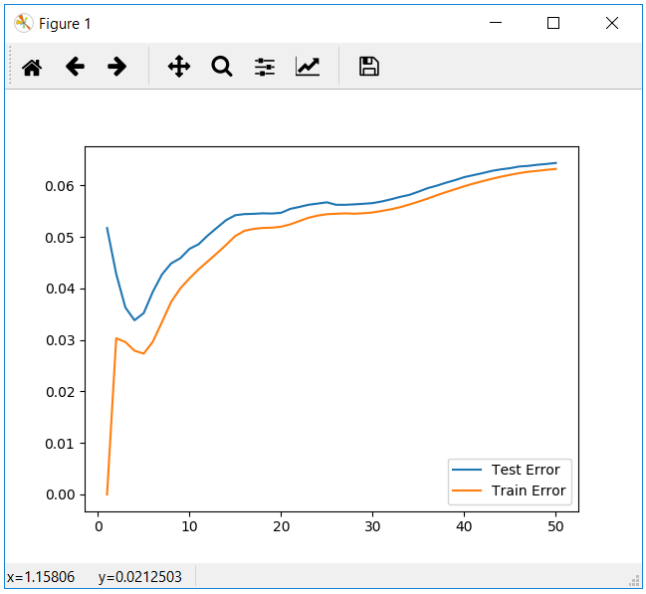


Figure 24 k-NN plot (n_neighbors parameter changing from 1~50)

The optimum value of n_neighbors is 4. Error plots are shown as below:

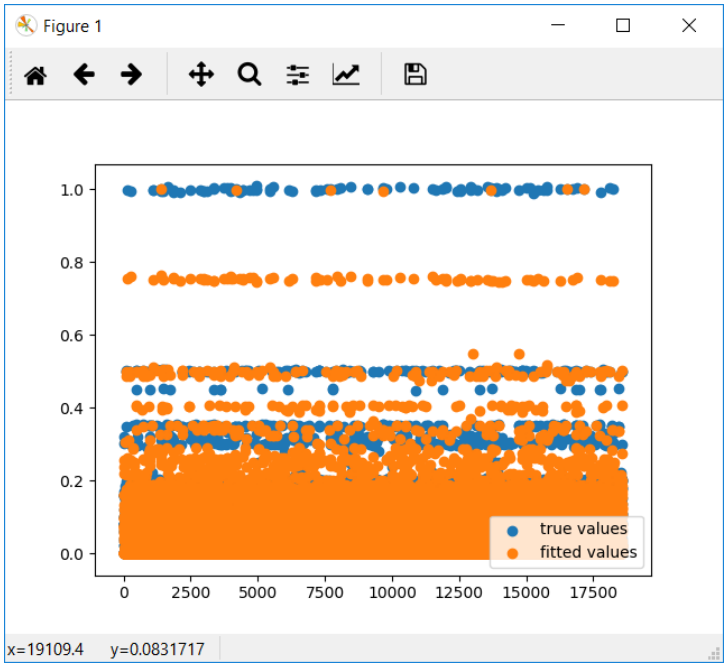


Figure 25 true value - fitted value plot

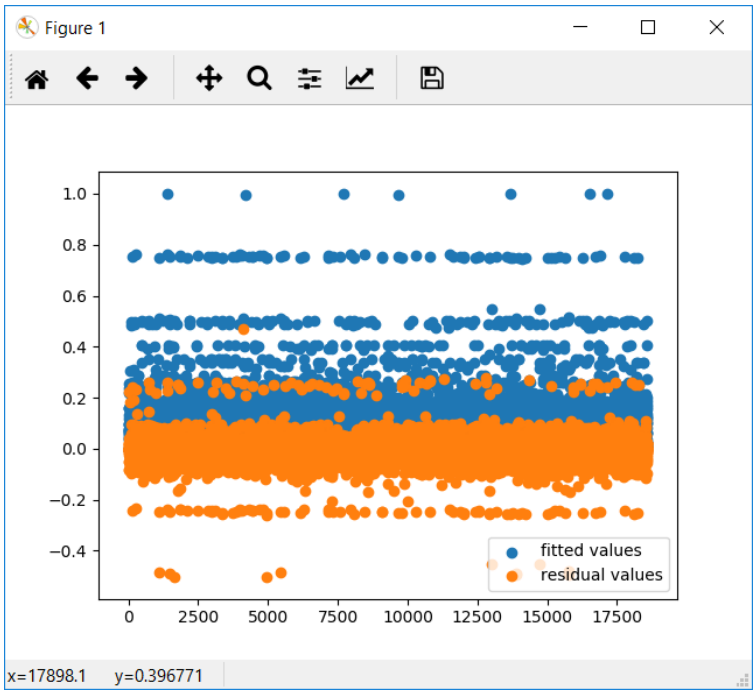


Figure 26 fitted value - residual value

Regression models comparison:

a) Linear Regression isn't a good model for this problem. We separately plot fitted value vs actual value and residual value versus fitted value to see whether we can get a good result from linear regression. For residual value, it equals to actual value minus fitted value. Obviously, in first picture, the best result we can get is that points distributed along straight line y=x, which means the value we predict is exactly the true value. In another plot, the best result we can get is that points distribute along y=0 which implies we obtain small error in this model.

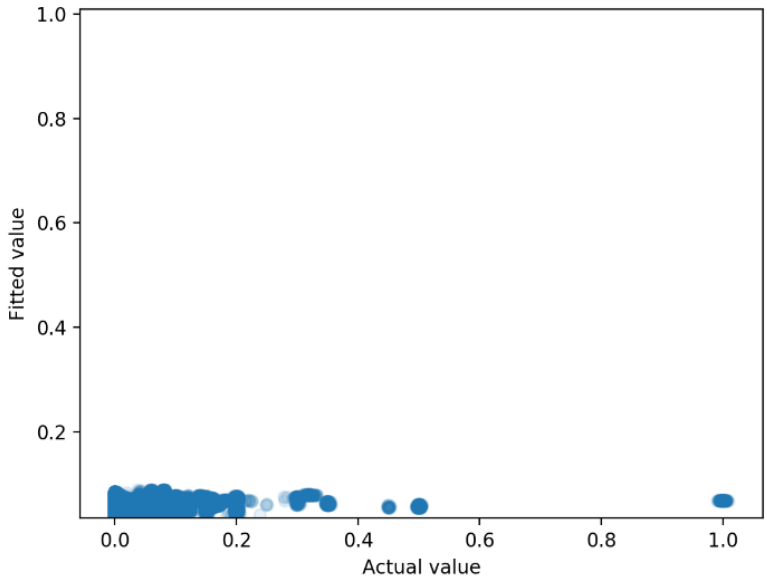


Figure 27 actual value – fitted value plot

The plot above shows that linear regression make little difference to predict backup size in this situation. There isn't a point even higher that y=0.1. Every point with a value bigger than 0.1, we can't fit them properly. Also the residual we calculate is really high, compared with the best result (along y=0), we get an opposite result with points distribute along x=0.

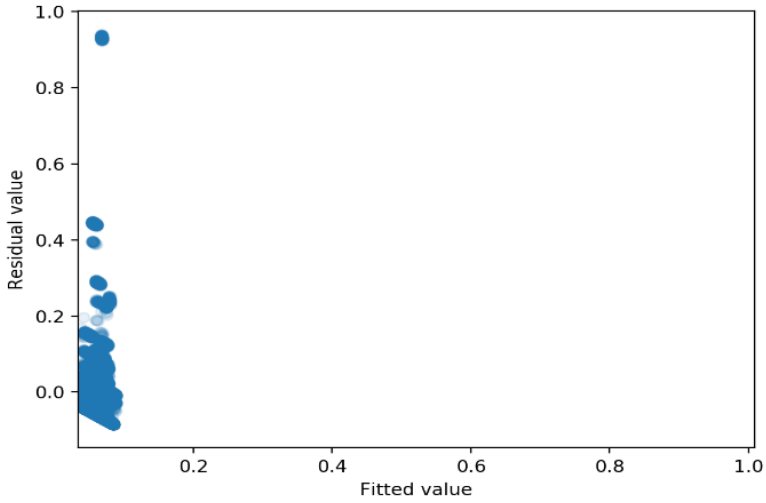


Figure 28 fitted value –resifual value plot

b) Neural Network, Random Forest and KNN regression solves this problem well. We did same operation to the result of Neural Network as the plot above. The picture below shows we get a good prediction result by Neural Network. The points distribute with a tendency along y=x, in other word, we predict a good value. As for residual, the range of error is about -0.2 to 0.2, in a reasonable scope.

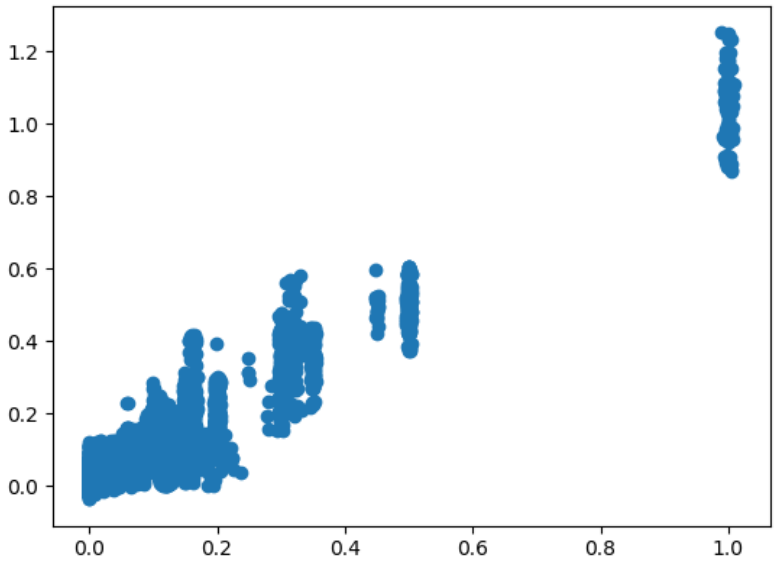


Figure 29 fitted value - true value plot

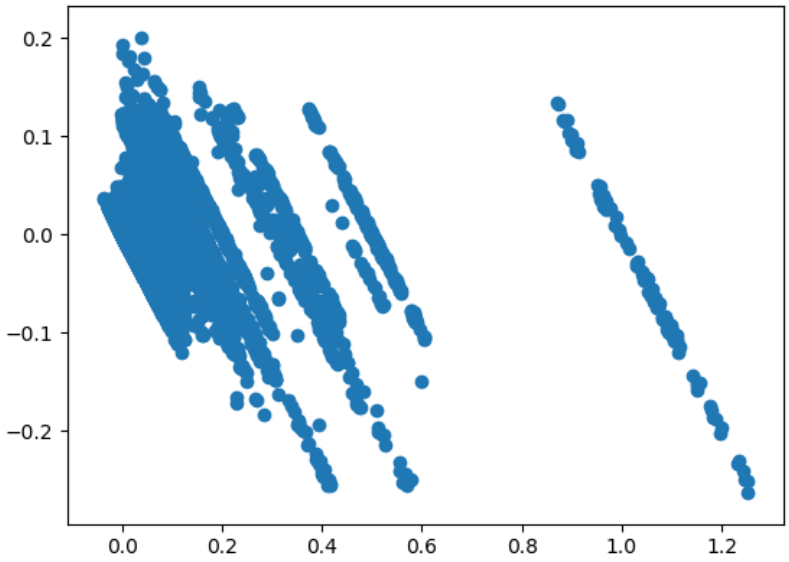


Figure 30 residual value – fitted value plot

c) For linear regression combined with polynomial function, it did a better but limited work compared with single linear regression.