ECE219 Project 4

Regression Analysis

Zhilai Shen, Yufei Hu, Zheang Huai, and Tianyi Liu
105023454, 404944367, 505222324, 705035425

March 7, 2019

Contents

1 Network Backup Dataset

1.1 Load the dataset

(a) The plot is as in figure 1:

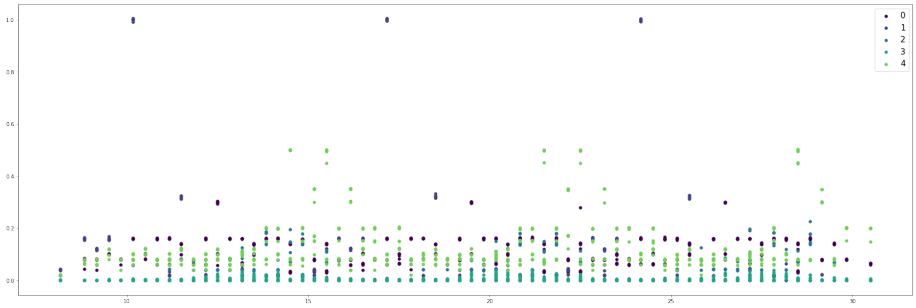

Figure 1: 20 days time vs 20 days backup size

(b)The plot is as in figure 2:

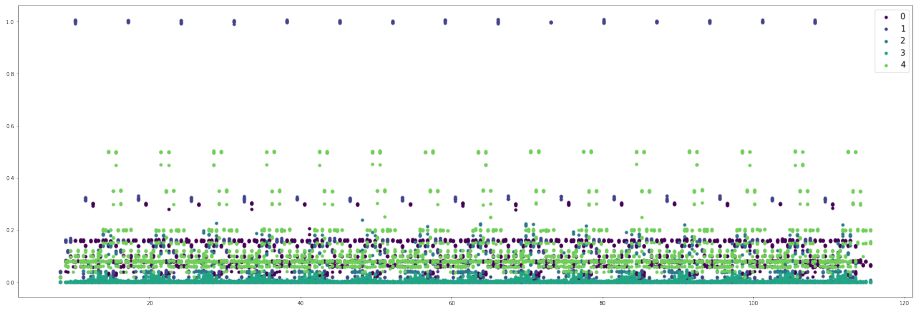

Figure 2: 105 days time vs 105 days backup size

(c)As shown in the above graphs, there are in fact repeating patterns. For a constant period of time, work-flow1 will upload a file with size around 1GB. Work-flow3 barely made any backup. Work-flow 0 is constantly used to backup some small files with size around 0.2GB.

1.2 Predict the backup size of a file given the other attributes

1.2.1 linear regression model

Fit the linear regression model.
All the data are converted into one-dimension. The RMSE for training across ten sets is 0.10358719585332854, and RMSE for testing across ten sets is 0.10358269769007267.

The fitted value against true value is in figure 3, and the residuals vs fitted values is in figure 4.

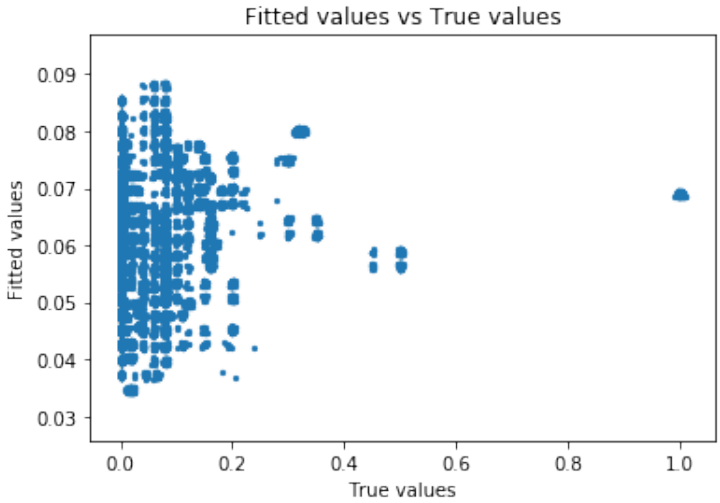

Figure 3: Fitted Value vs True Value

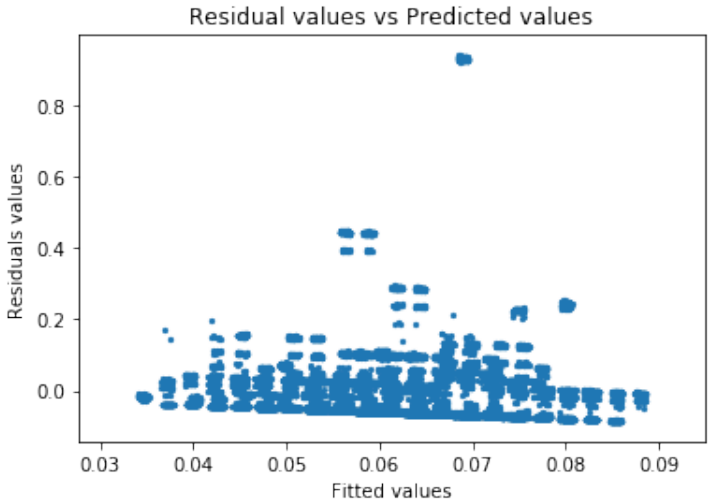

Figure 4: Residual vs Fitted Value

1.2.2 random forest regression model

Fit the random forest tree model.

(i)For the initial model, the RMSE for training is 0.06041914907193139, and the average RMSE for testing is 0.06077076528311328. The average out of bag error is 0.3411795413509088.

(ii)After sweeping the maximum number of trees from 1 to 200, and maximum number of features from 1 to 5 and applying the cross-validation using 10 fold, the plot for out of bag error vs number of trees is in figure 5, and the plot for average test-RMSE vs number of trees is in figure 6.

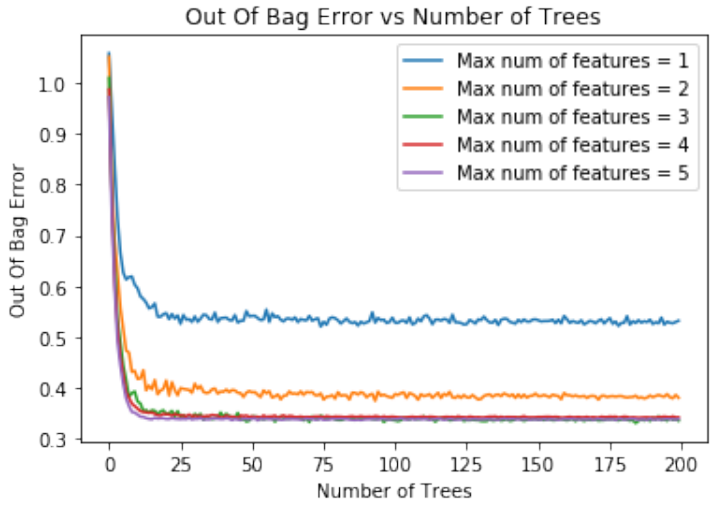

Figure 5: Out of bag error vs number of trees

(iii)For further test, the number of maximum depth is chosen. The depth of each tree is swept from 1 to 30 to find a good performance. The plot for out of bag error vs number of trees is in figure 7, and the plot for average test-RMSE vs number of trees is in figure 8. As seen from plots in part (ii) and part (iii), the best hyperparameter is to use maximum number of features = 3, maximum number of trees = 25, and maximum depth = 7.

(iv)The feature importance for the best model in part (iii) is in table 1.
The average RMSE for training is 0.021061323772690425, and the average RMSE for testing is 0.021657162672114462. The fitted value against true value is in figure 9, and the residuals vs fitted values is in figure 10.

(v)The decision tree is in figure 11. The tree node is work-flow-id, which is

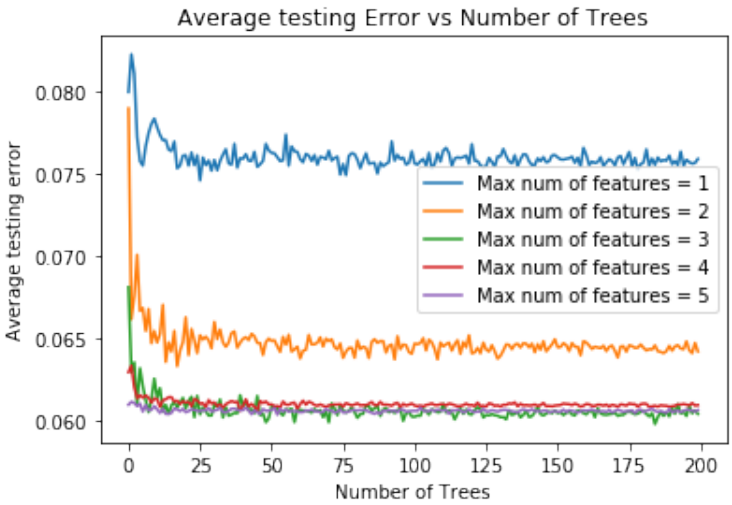

Figure 6: Average test-RMSE vs number of trees

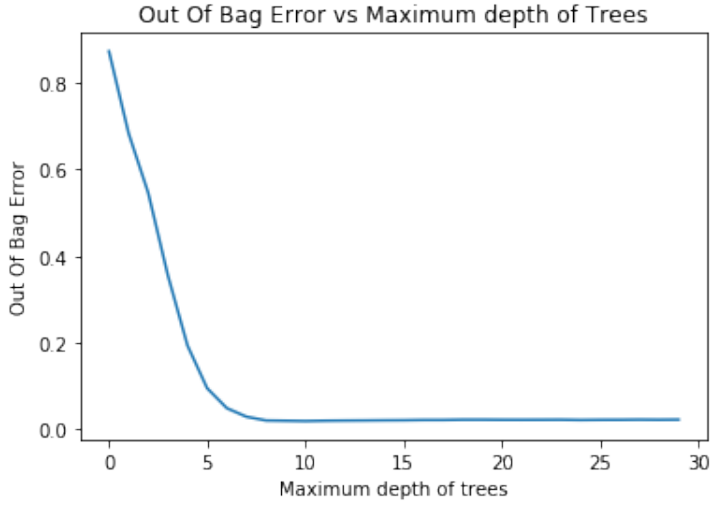

Figure 7: Out of bag error vs number of trees

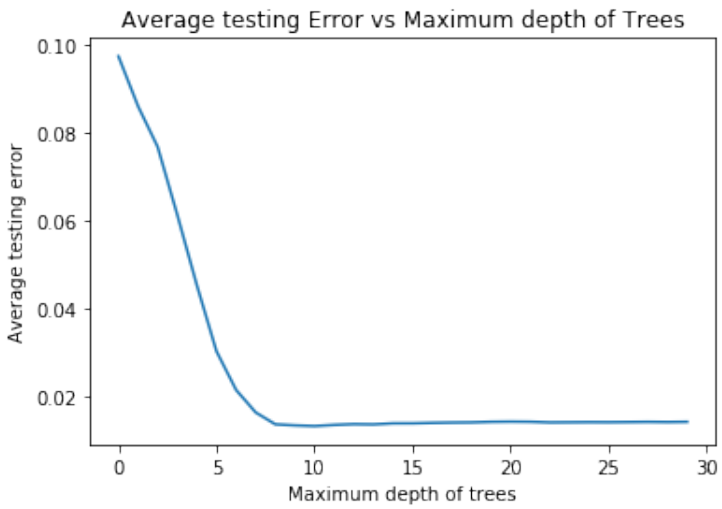

Figure 8: Average test-RMSE vs number of trees

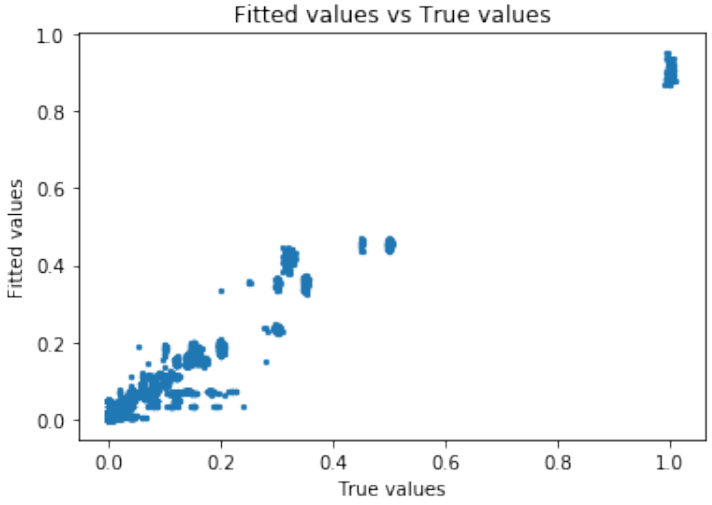

Figure 9: Fitted Value vs True Value

Table 1: Feature Importance

Feature Name	Importance
Day of the Week	0.28760693
Hour of the Day	0.3216406
Work-flow-ID	0.20244174
File Name	0.18558651
Week Number	0.00272422

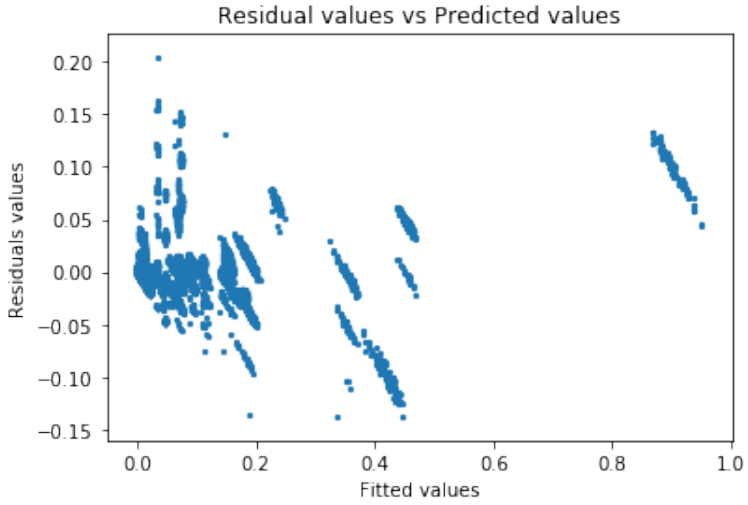

Figure 10: Residual vs Fitted Value

not the most important feature but a relatively important feature according to the feature importance in part (iv).

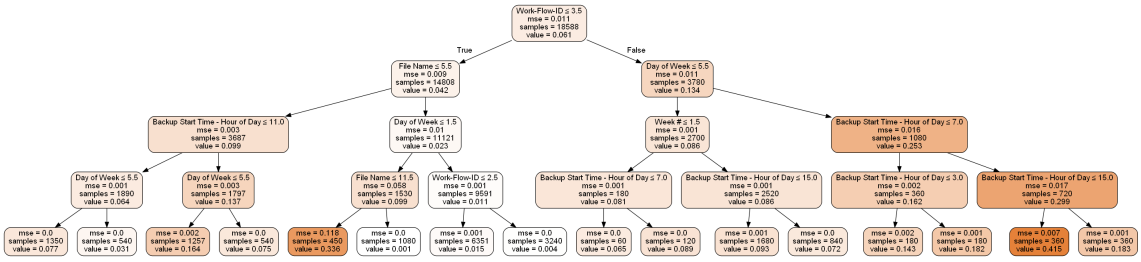

Figure 11: Decision Tree

1.2.3 neural network regression model

Now we used a neural network regression model with one hidden layer and all features one-hot encoded. Again we used k-folder cross-validation to find the optimal combination of the number of hiddens units and activation functions. We tried $2, 5, 10, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600$ with relu, logistic, tanh. The plot of the test-RMSE against the number of hidden units for different activation functions is shown in Fig 12.

The best model chosen from the cross-validation is a neural network model with 550 hidden units and relu activation and the train and test RMSE are 0.04054727126423967 and 0.03918910976276833. The plot of fitted values against true values and residuals versus fitted values are shown in Fig 13 and Fig 14.

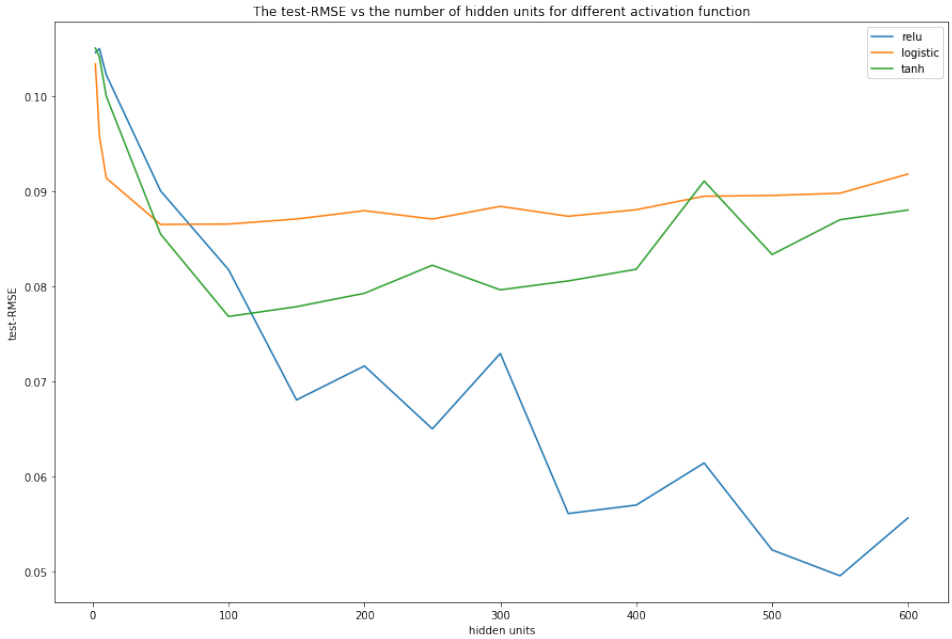

Figure 12: the test-RMSE against the number of hidden units for different activation functions

1.2.4 separate workflows

(i) This time we predicted the Backup size for each of the workflows separately. Using linear regression model, the scatter plot of those 5 workflows are shown in Fig 15, 16, 17, 18, 19, 20, 21, 22, 23, and 24. The train and test error of each separate workflow can be seen in Table 2. We could see that the performance of the linear prediction model has been dramatically improved. This is very intuitive since each workflow may has each different inside parameters

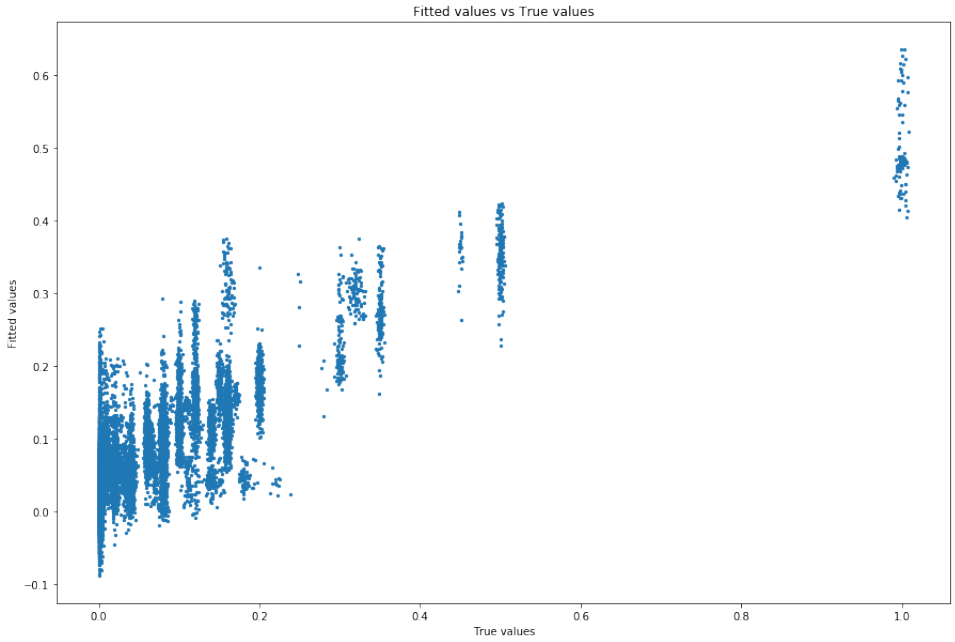

Figure 13: fitted values against true values

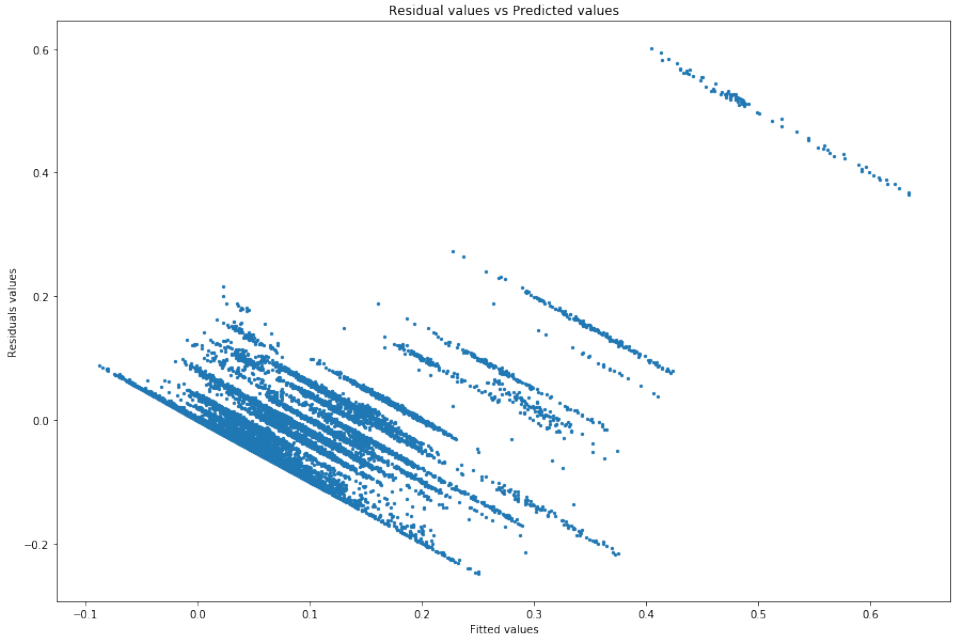

Figure 14: residuals versus fitted values

or models. By separating those difference the linear model would fit the data more accurately.

Table 2: Train and test RMSE

Workflow ID	train RMSE	test RMSE
0	0.035836058473410995	0.035864233365527486
1	0.14873667005167984	0.14718022940875605
2	0.042912741990063744	0.042871138232155256
3	0.007244122686732686	0.007236474722628693
4	0.08591840773195633	0.08587749300354083

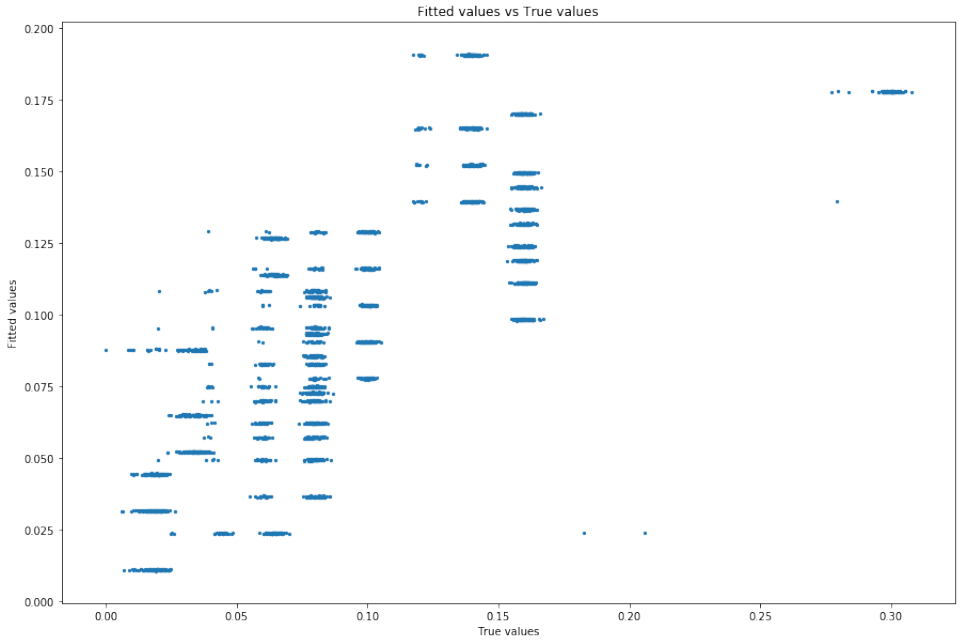

Figure 15: fitted values against true values for workflow 0

(ii) We tried different degree of the polynomial of the model. The plot of the train and test RMSE against the degree is shown in Fig 25, 26, 27, 28, and 29 separately. The train and test error and best degree that we find of each separate workflow can be seen in Table 3. It's obvious that as we increase the degree the training error can always decrease, which seems like a good thing but what comes aside is the overfitting problem. The plot proves it that the test error becomes increase if the degree is way too high.

The new scatter plot with best parameter for those 5 workflows are shown in Fig 30, 31, 32, 33, 34, 35, 36, 37, 38, and 39.

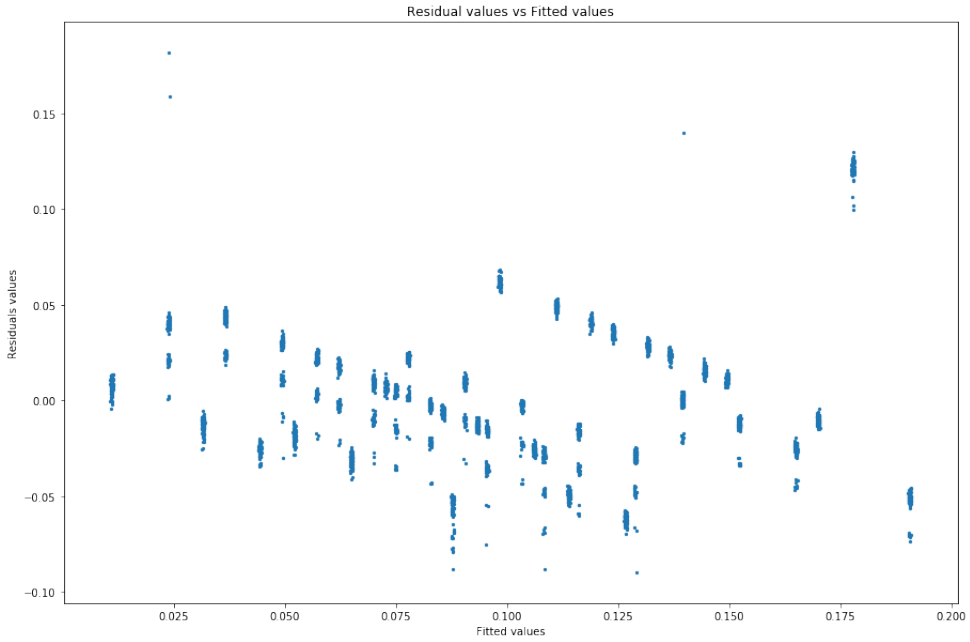

Figure 16: residuals versus fitted values for workflow 0

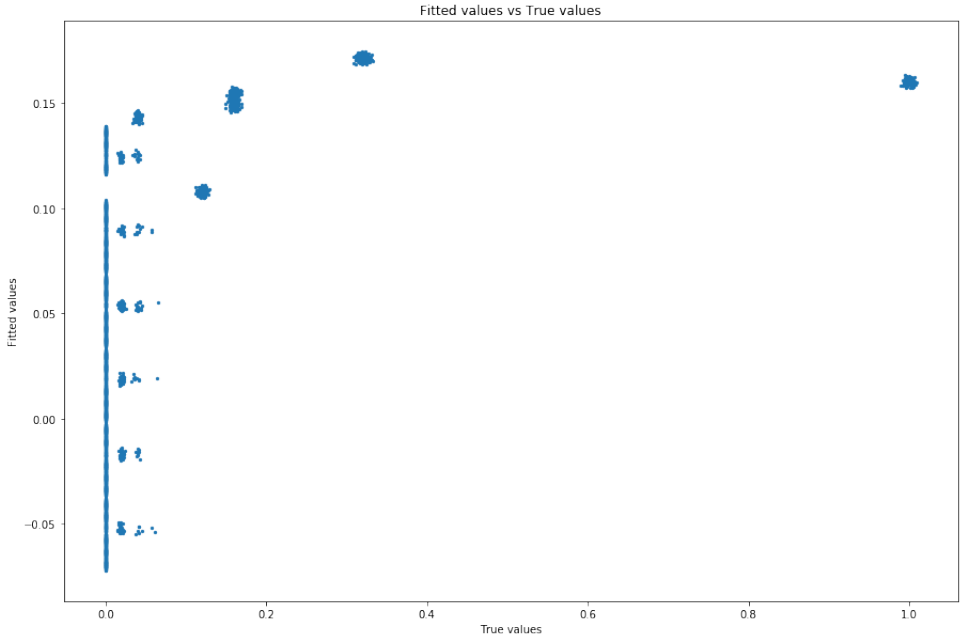

Figure 17: fitted values against true values for workflow 1

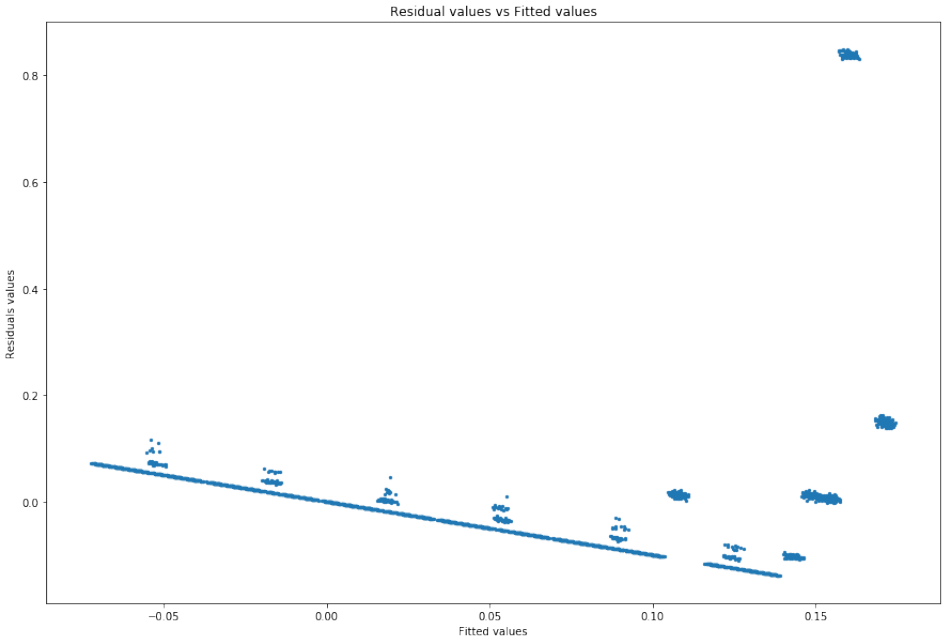

Figure 18: residuals versus fitted values for workflow 1

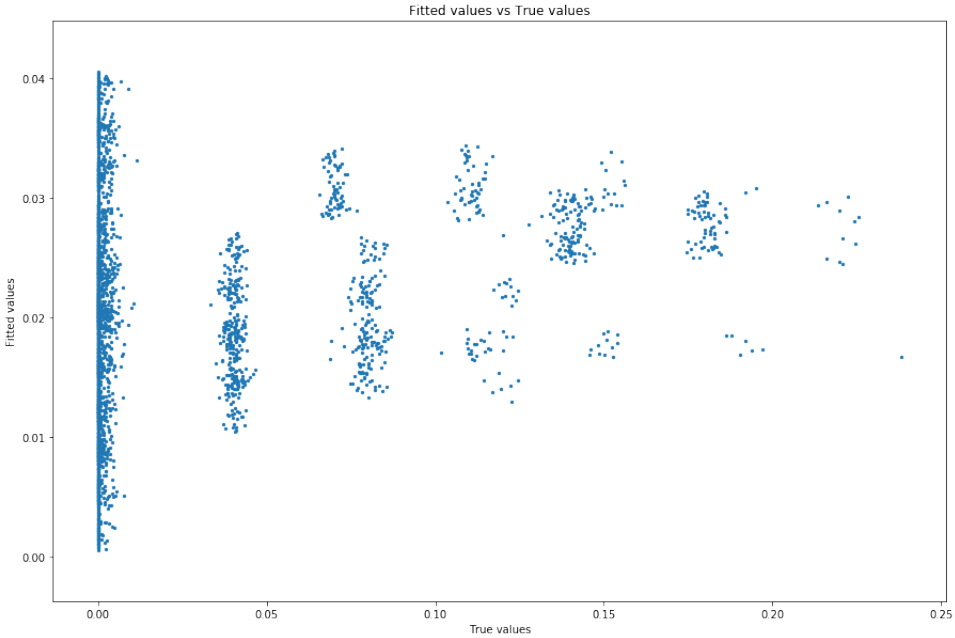

Figure 19: fitted values against true values for workflow 2

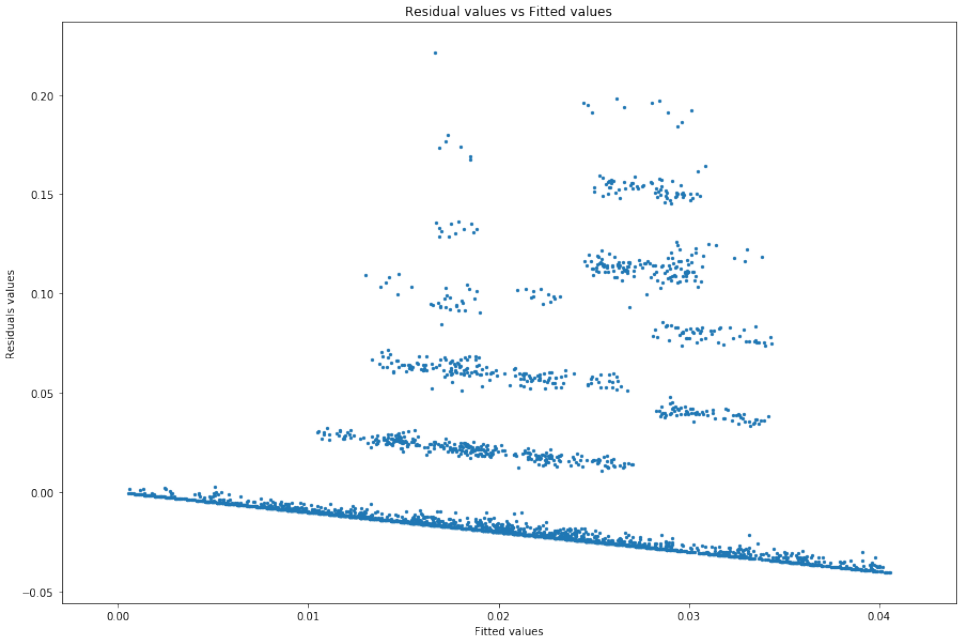

Figure 20: residuals versus fitted values for workflow 2

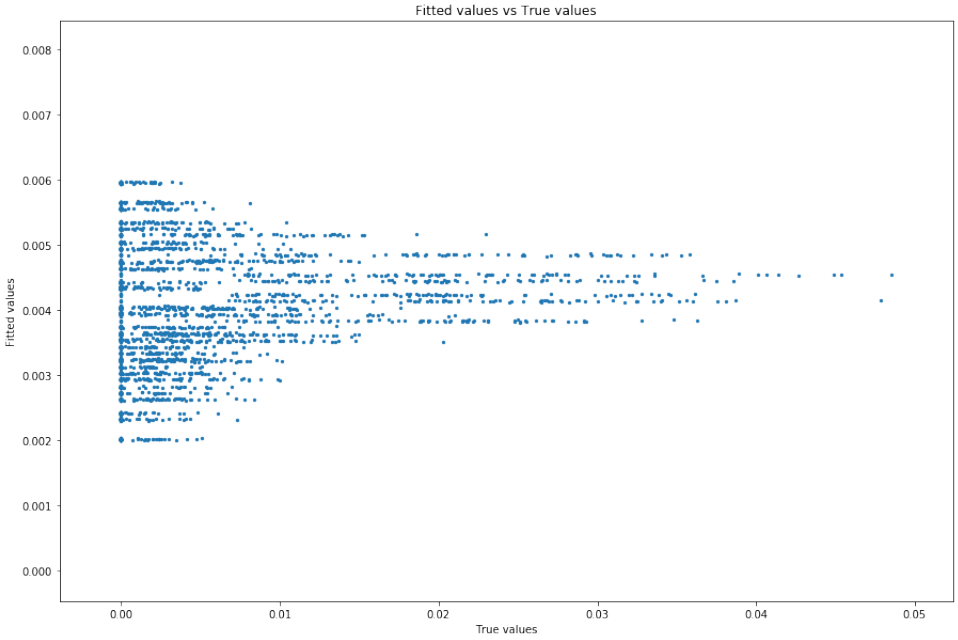

Figure 21: fitted values against true values for workflow 3

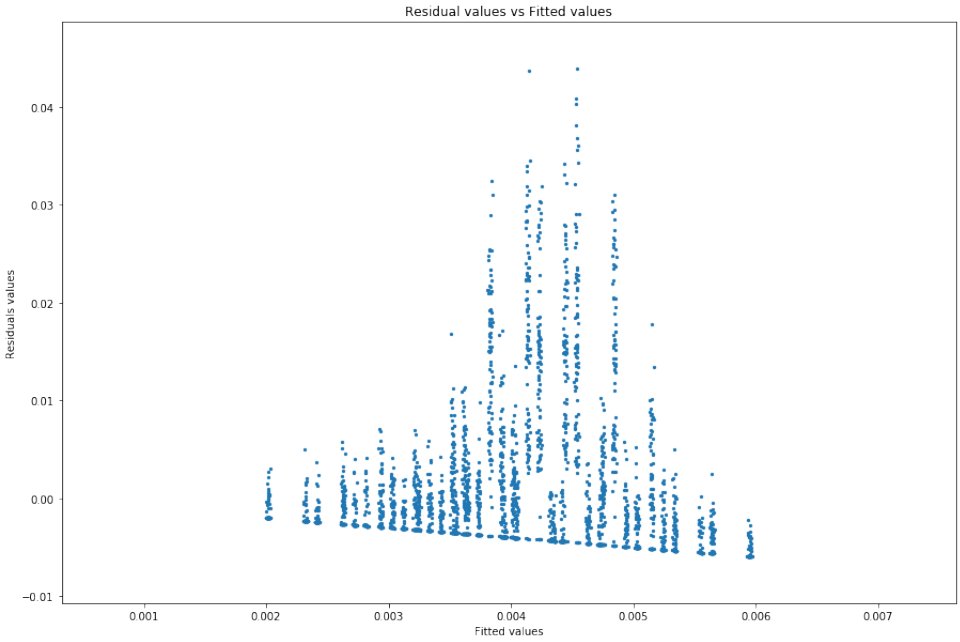

Figure 22: residuals versus fitted values for workflow 3

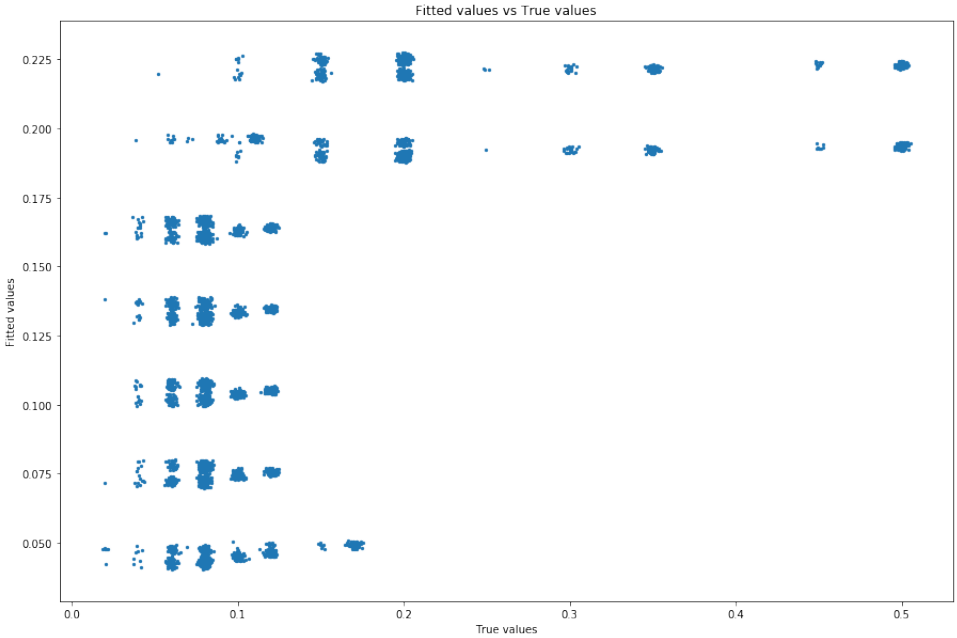

Figure 23: fitted values against true values for workflow 4

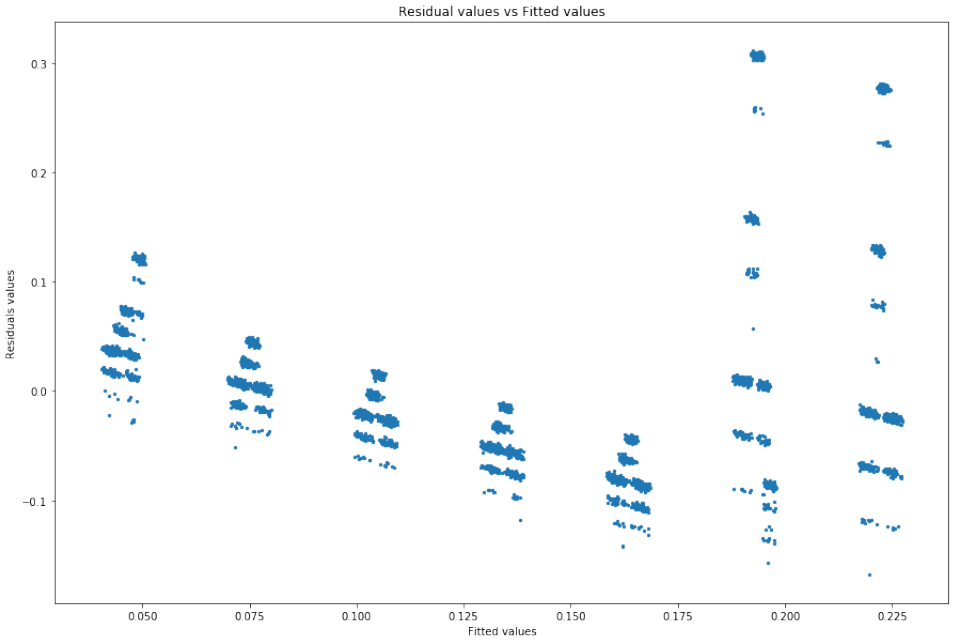

Figure 24: residuals versus fitted values for workflow 4

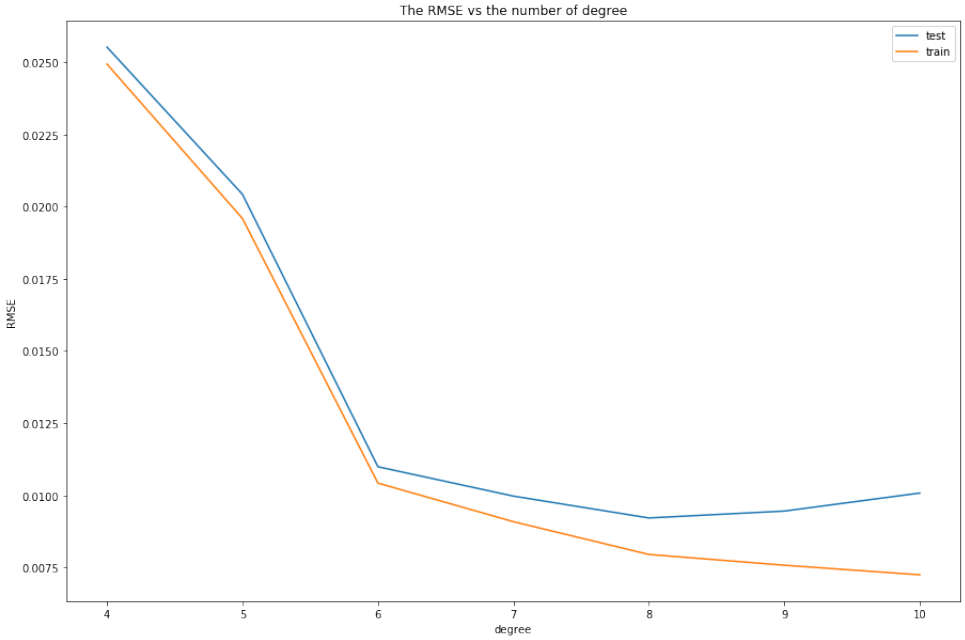

Figure 25: the RMSE against the degree for workflow 0

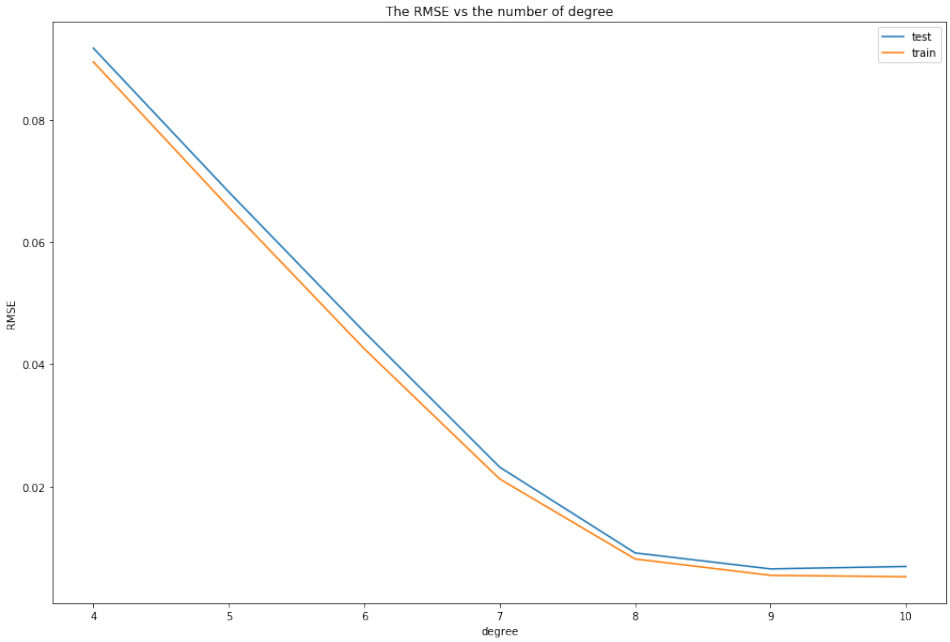

Figure 26: the RMSE against the degree for workflow 1

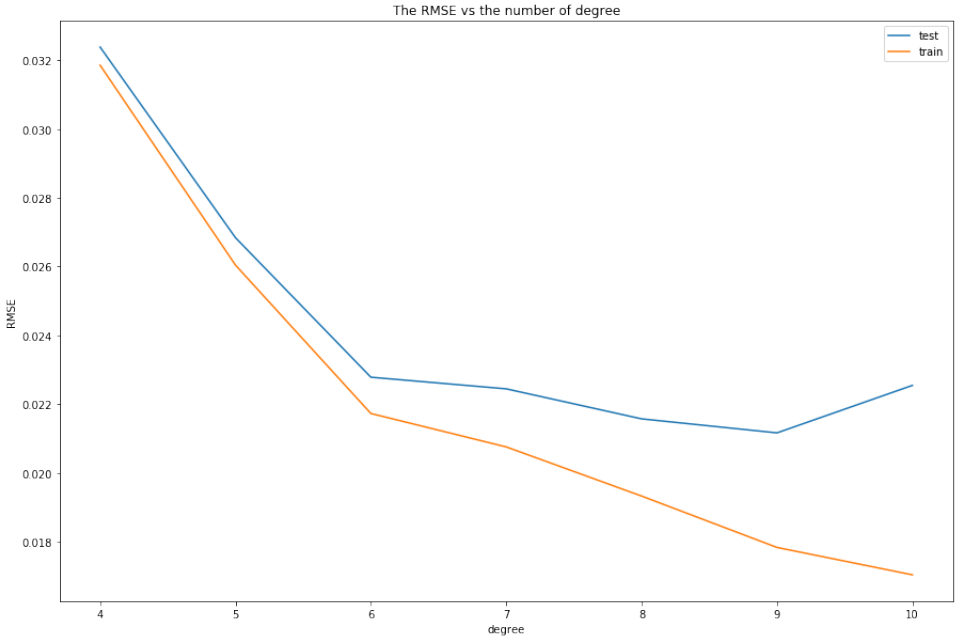

Figure 27: the RMSE against the degree for workflow 2

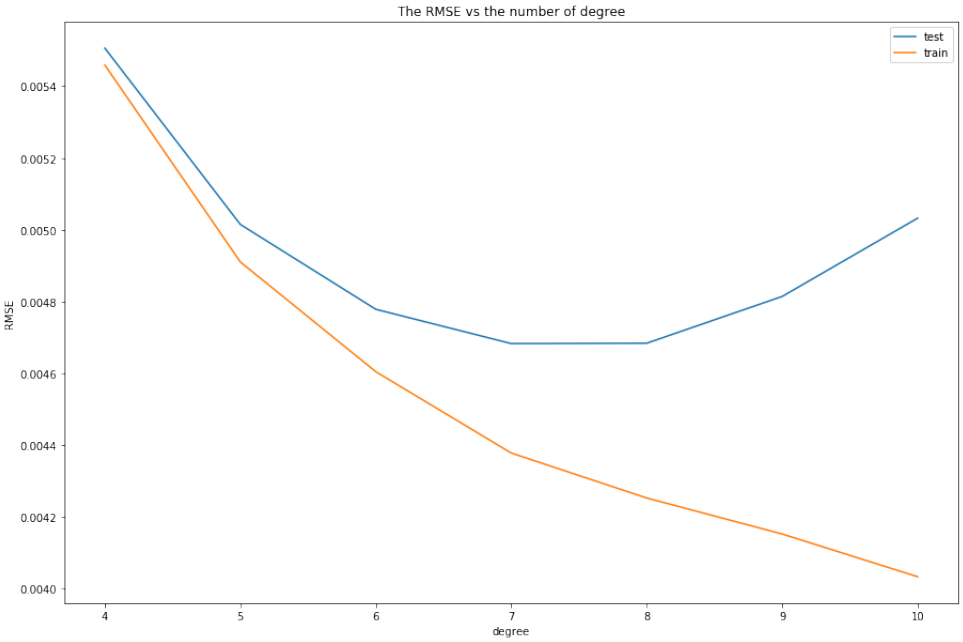

Figure 28: the RMSE against the degree for workflow 3

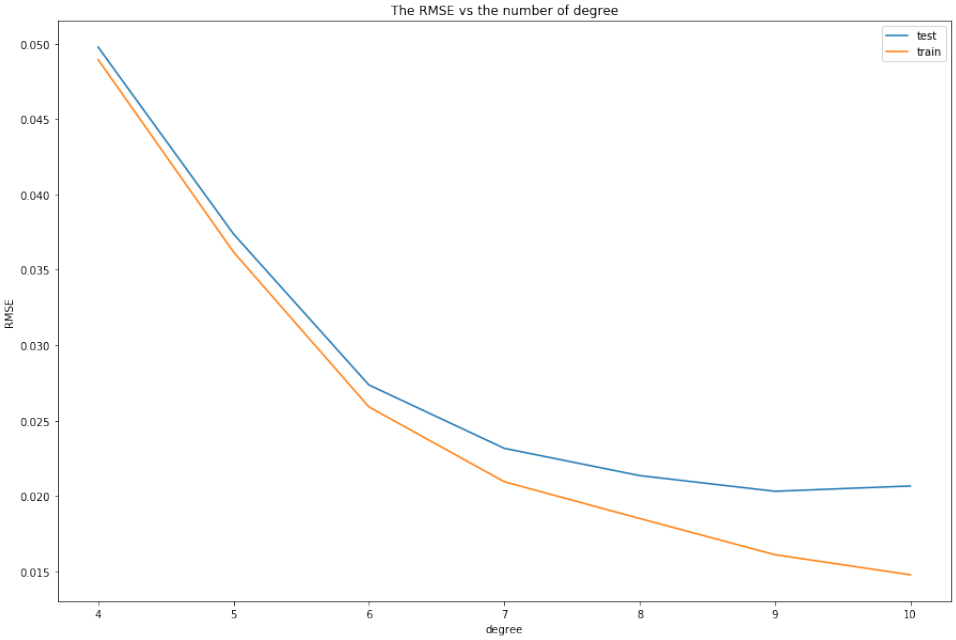

Figure 29: the RMSE against the degree for workflow 4

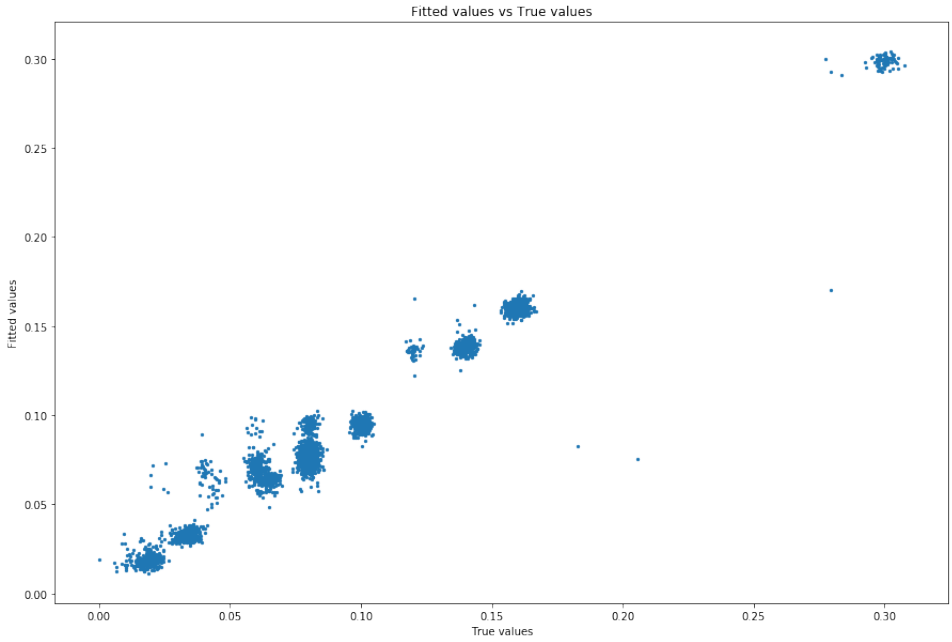

Figure 30: fitted values against true values for workflow 0

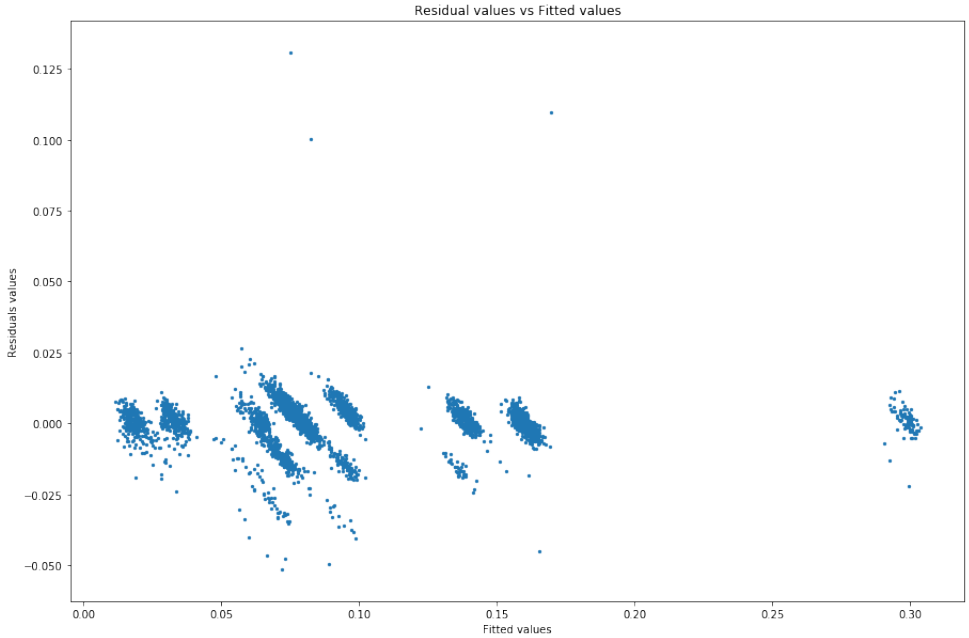

Figure 31: residuals versus fitted values for workflow 0

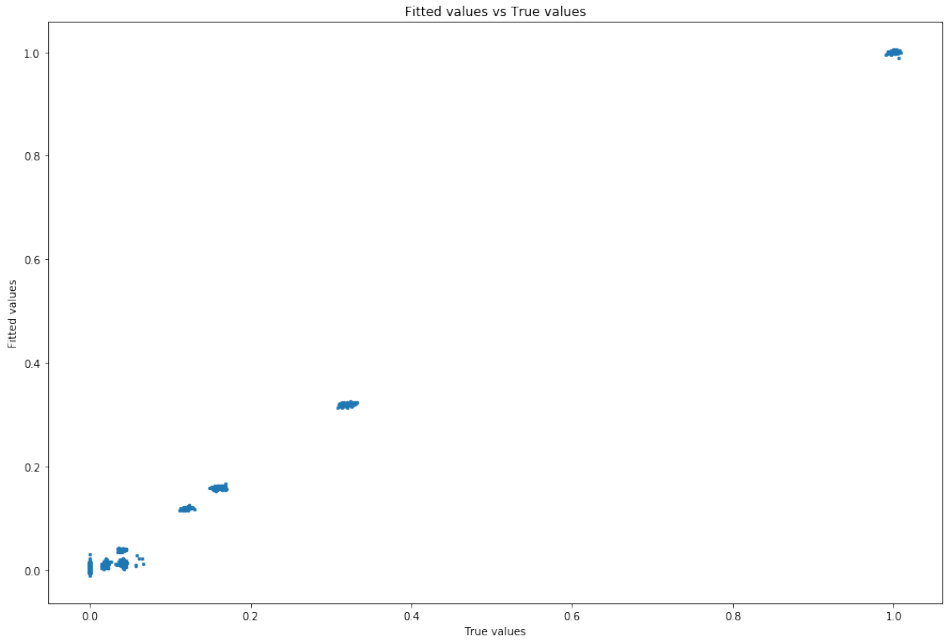

Figure 32: fitted values against true values for workflow 1

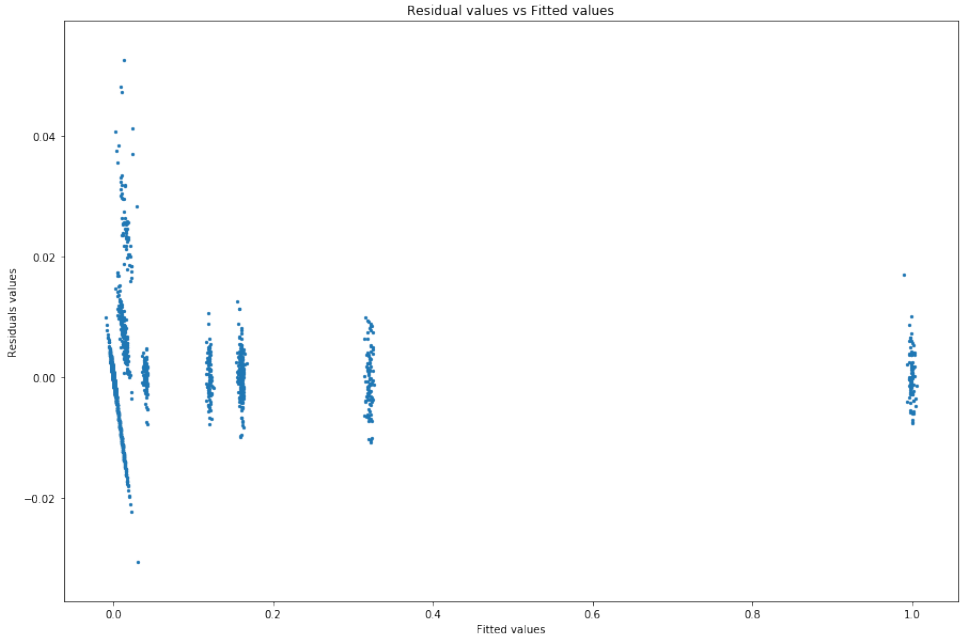

Figure 33: residuals versus fitted values for workflow 1

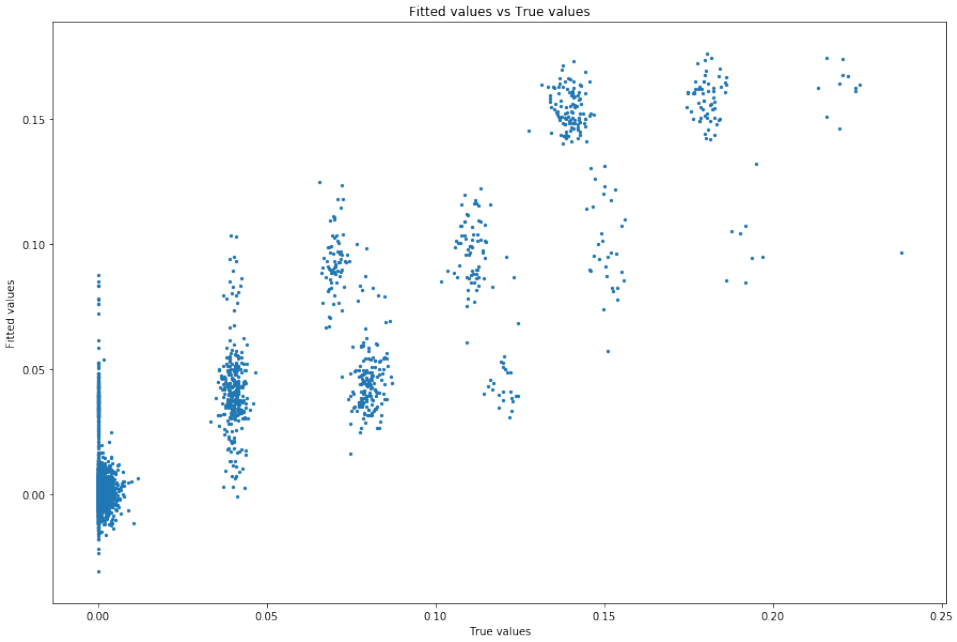

Figure 34: fitted values against true values for workflow 2

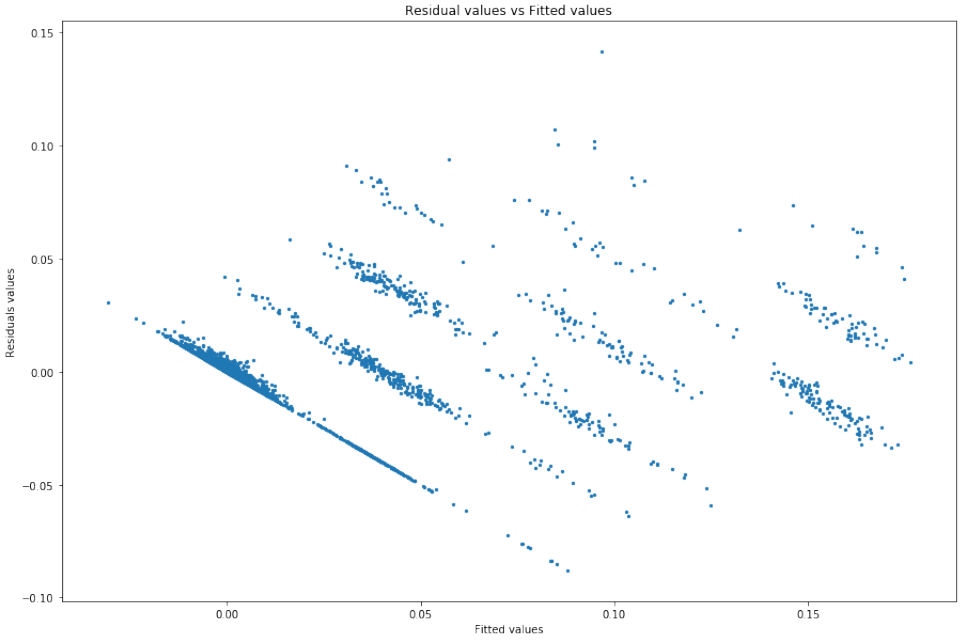

Figure 35: residuals versus fitted values for workflow 2

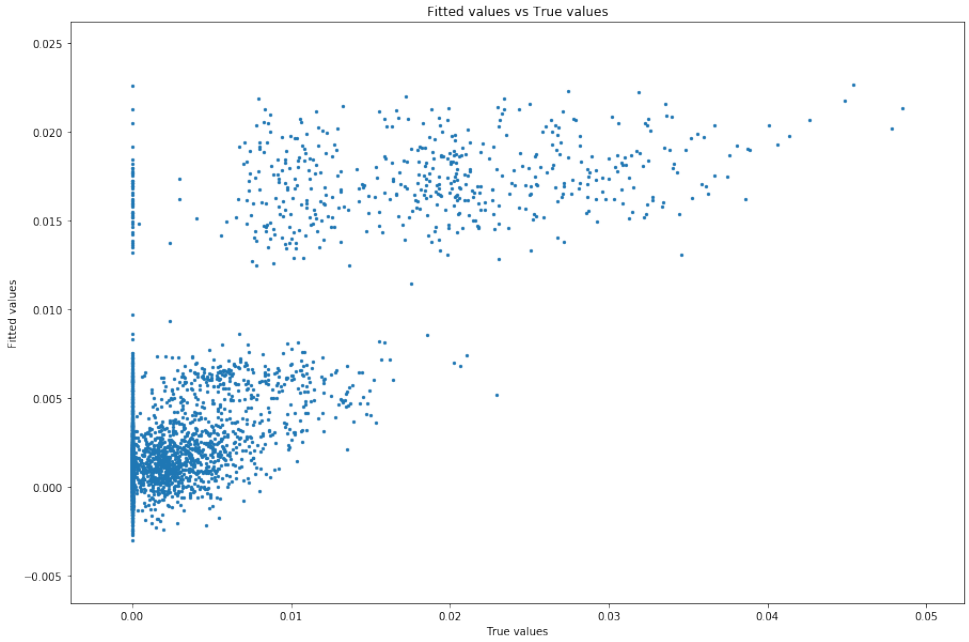

Figure 36: fitted values against true values for workflow 3

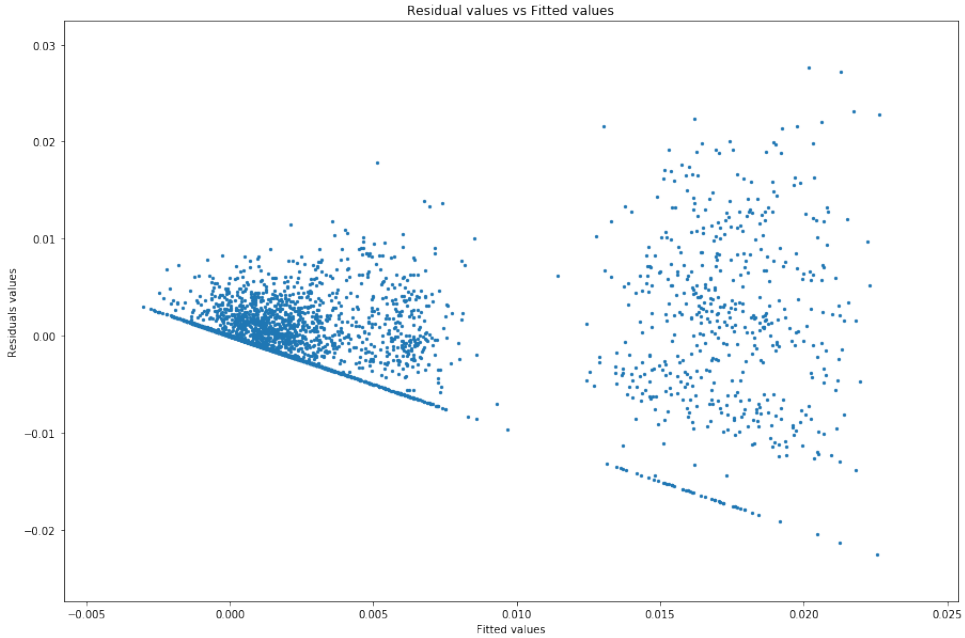

Figure 37: residuals versus fitted values for workflow 3

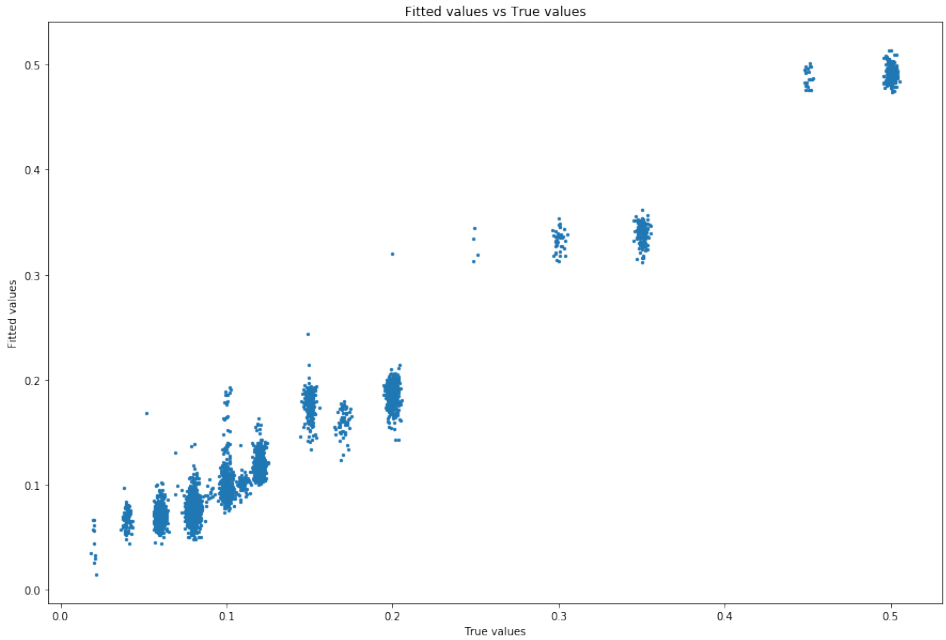

Figure 38: fitted values against true values for workflow 4

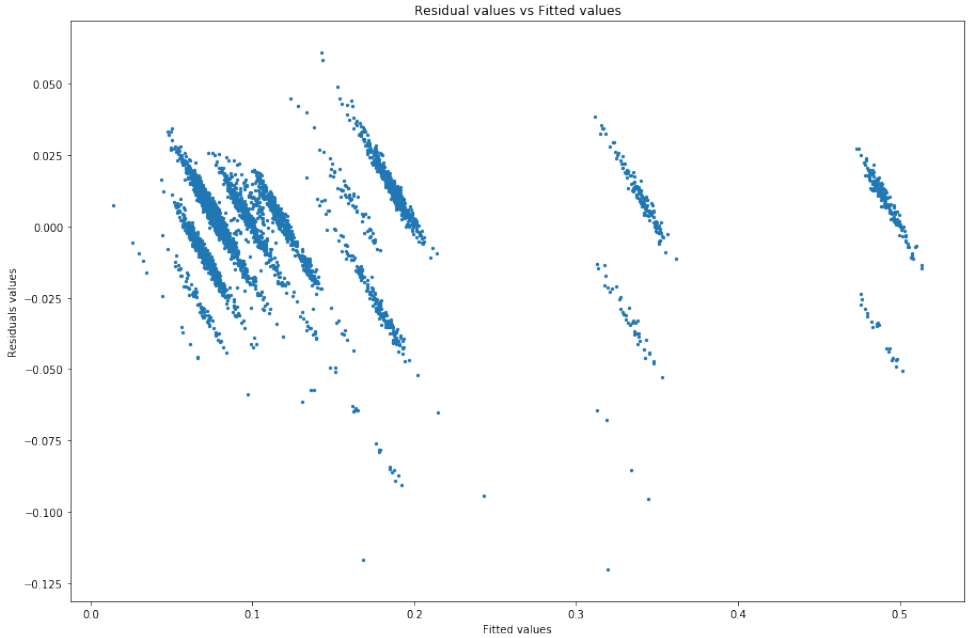

Figure 39: residuals versus fitted values for workflow 4

Table 3: Train and test RMSE and best degree

Workflow ID	degree	train RMSE	test RMSE
0	8	0.009215949356264234	0.007952311037775975
1	9	0.006489646031637972	0.005441175333664292
2	9	0.021160106836710026	0.017828380830571006
3	7	0.004682304240652268	0.004377076086892894
4	9	0.020306227755373964	0.016098651643866833

1.2.5 k-nearest neighbor regression

Using k-nearest neighbor regression, the plot of the test-RMSE against the number of neighbor is shown in Fig 40. The best parameter is 4 and the train and test RMSE of that model are 0.03509321892803684 and 0.027896132153362275. The plot of fitted values against true values and residuals versus fitted values are shown in Fig 41 and Fig 42.

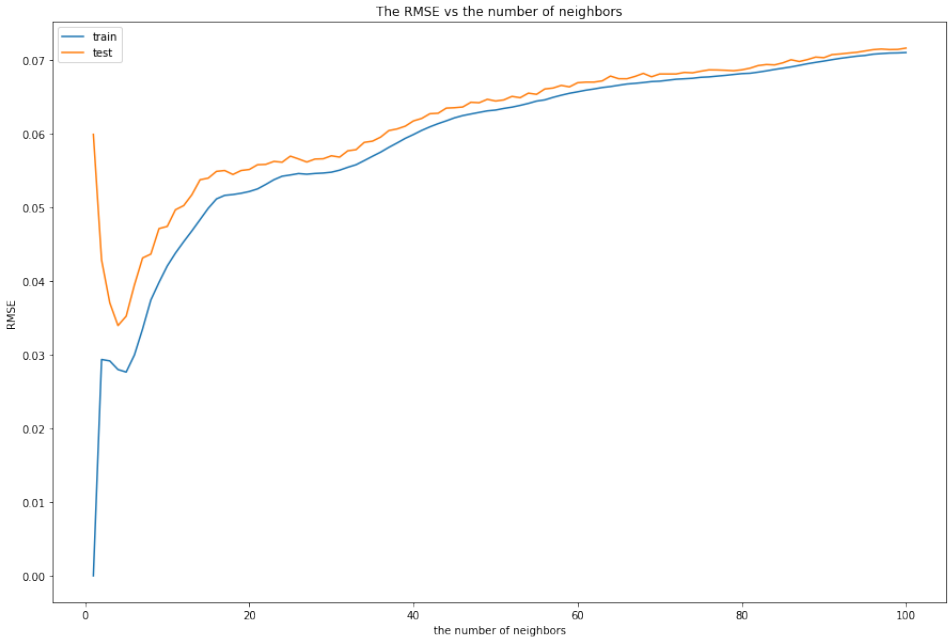

Figure 40: the test-RMSE against the number of neighbor

1.3 Compare these regression models

The performance of all these regression models we discussed before can be found in Table 4. We could conclude from the observation that the random forest, neural network and knn all give us good enough performance under

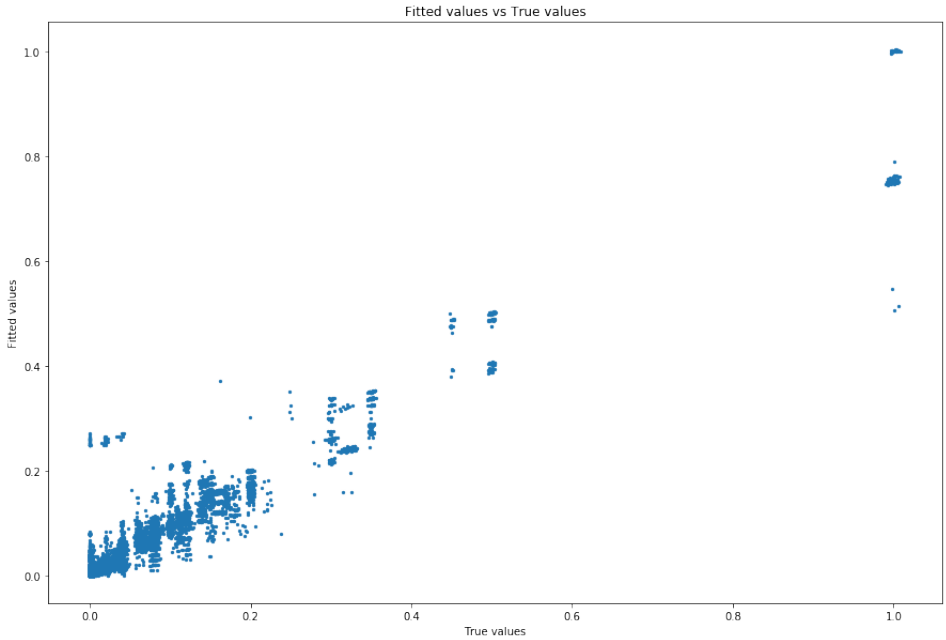

Figure 41: fitted values against true values

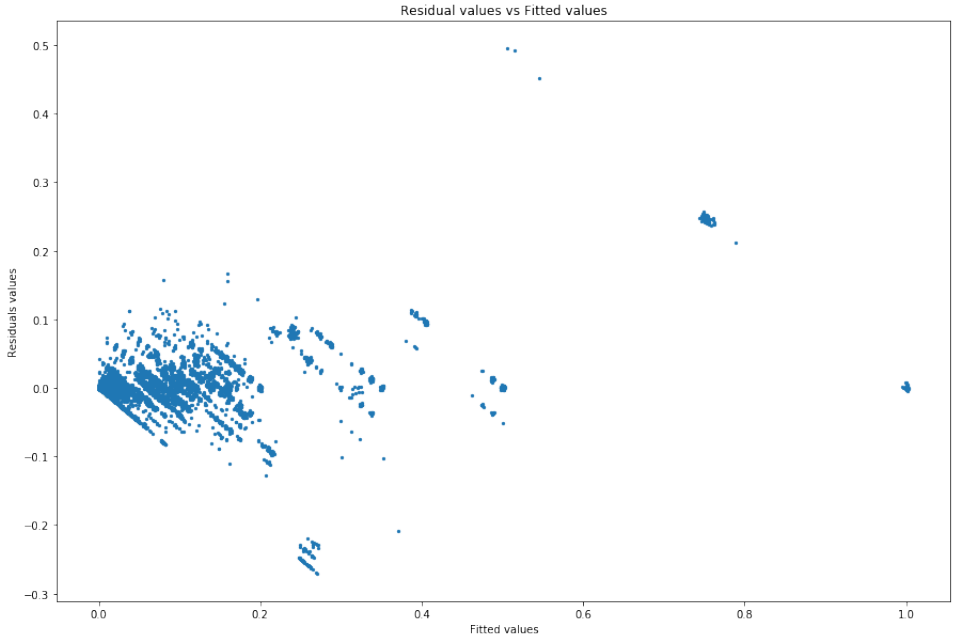

Figure 42: residuals versus fitted values

the measurement of test RMSE. Even though the original linear model doesn't work well, we still can extremely improve it by separating each workflow and increasing the degrees of the model.

Table 4: Best train and test RMSE among all regression models

model	train RMSE	test RMSE
linear	0.10358719585332854	0.10358269769007267
linear workflow 0	0.035836058473410995	0.035864233365527486
linear workflow 1	0.14873667005167984	0.14718022940875605
linear workflow 2	0.042912741990063744	0.042871138232155256
linear workflow 3	0.007244122686732686	0.007236474722628693
linear workflow 4	0.08591840773195633	0.08587749300354083
high degree linear workflow 0	0.009215949356264234	0.007952311037775975
high degree linear workflow 1	0.006489646031637972	0.005441175333664292
high degree linear workflow 2	0.021160106836710026	0.017828380830571006
high degree linear workflow 3	0.004682304240652268	0.004377076086892894
high degree linear workflow 4	0.020306227755373964	0.016098651643866833
random forest	0.021061323772690425	0.021657162672114462
neural network	0.04054727126423967	0.03918910976276833
k-nearest neighbour	0.03509321892803684	0.027896132153362275.

2 Boston Housing Dataset

2.1 Load the dataset

2.2 Fit a linear regression model

The F-values and p-values are:

F = [89.27867007 75.08340878 153.982021 15.95925779 112.35134671 470.92252602 83.29994976 33.50345866 85.77751739 141.48567511 175.33780624 62.90074607 602.30110356]

p = [1.29165892e-19 6.19893794e-17 4.93018661e-31 7.43915002e-05 7.87623345e-24 3.45360629e-74 1.70543500e-18 1.25304781e-08 5.83365934e-19 6.37376693e-29 1.50648864e-34 1.41751808e-14 4.90947265e-88]

The variable that has a low p-value is likely to be significant. Conversely, a larger p-value suggests the variable is likely to be insignificant. So the order of significance of variables is: $LSTAT > RM > PTRATIO > INDUS > TAX > NOX > CRIM > RAD > AGE > ZN > B > DIS > CHAS$

Averaged Root Mean Squared Errors of test and training are: Average test RMSE: 5.191327655559577 ; Average training RMSE: 4.605185401794117

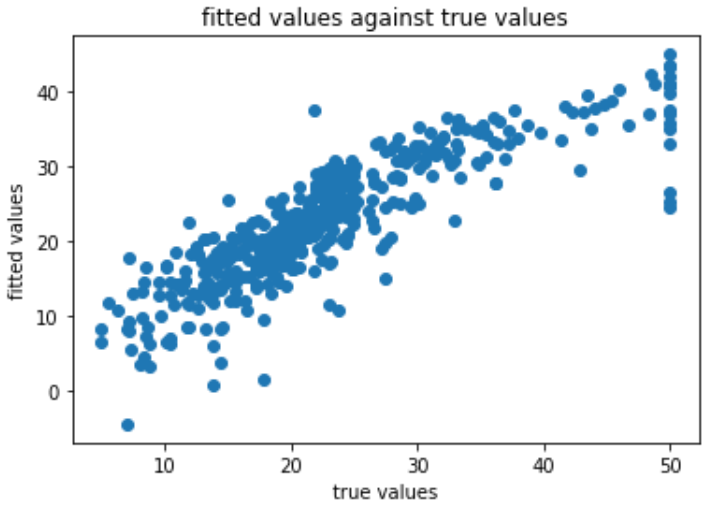

Figure 43: fitted values against true values

Scatter plots of fitted values against true values and residuals versus fitted values using the whole dataset can be seen in figure 43 and figure 44.

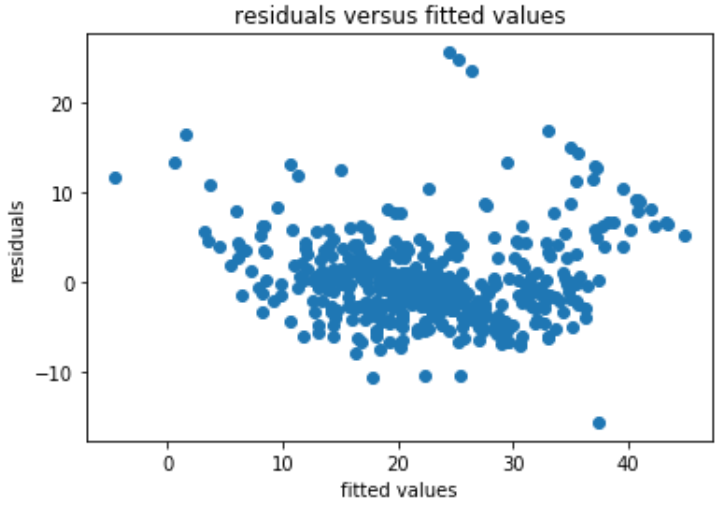

Figure 44: residuals versus fitted values

2.3 Control overfitting via regularization of the parameters

After trying different values, best choices of α, $\lambda 1$, $\lambda 2$ and corresponding RMSEs obtained via 10-fold cross validation and estimated coefficients are:

Regularization method: Ridge
Best parameter Test RMSE: 5.037718653318939
Best parameter Train RMSE: 4.842570572044513
Best α: 100
Estimated coefficients: [-9.87348682e-02 5.75000733e-02 -5.24367753e-02 6.30453235e-01 -2.50839386e-01 2.20491982e+00 -1.13944138e-04 -1.23509929e+00 3.09145266e-01 -1.55229229e-02 -8.01378743e-01 9.38365517e-03 -6.94791951e-01]

Regularization method: Lasso
Best parameter Test RMSE: 5.1083599482410325
Best parameter Train RMSE: 4.729257067596344
Best α: 0.1
Estimated coefficients: [-0.09499876 0.05404697 -0.03796886 1.1136662 -0. 3.55262497 -0.01277121 -1.2804953 0.2682344 -0.01408458 -0.7423286 0.01017644 -0.60427997]

Regularization method: Elastic Net
Best parameter Test RMSE: 5.022347107449401
Best parameter Train RMSE: 4.761934847801649
Best α: 0.1
Best l1Ratio: 0

Best $\lambda 1$: 0.0
Best $\lambda 2$: 0.05
Estimated coefficients: [-0.0989382 0.05676159 -0.04914643 1.05795062 -0.55761236 2.85560729 -0.00776635 -1.30952237 0.29329637 -0.01473338 -0.7874883 0.0097517 -0.64968596]

Estimated coefficients of unregularized best model are: [-1.06498784e-01 5.14808179e-02 2.90518248e-02 2.69067244e+00 -1.80855060e+01 3.66226749e+00 -1.00886887e-03 -1.57259804e+00 3.09021741e-01 -1.19639182e-02 -9.72294752e-01 9.43731639e-03 -5.54185329e-01]

Among the three kinds regularization, the one that has smallest test RMSE is Elastic Net Regularizer with parameter $\lambda 1$=0.0, $\lambda 2$=0.05. And the best RMSE is 5.02235. (Annotation: Here I justify why best α for Ridge is 100 whereas best $\lambda 2$ for Elastic Net is 0.05. I found the test RMSE varies little when α changes from 0.1 to 100 for Ridge. The test RMSEs for α=0.1,1,10,100 are 5.170394749496701, 5.096697120354928 5, 5.054582577537348, 5.037718653318939,respectively.)

On the whole, the values of the estimated coefficients for these regularized good models are smaller than those of unregularized best model.(Most coefficients of the former are smaller than the latter, although some are not.) The reason is that we add the norm of coefficients to the cost function.

3 Car Insurance Dataset

3.1 Feature Preprocessing

(a) Feature Encoding

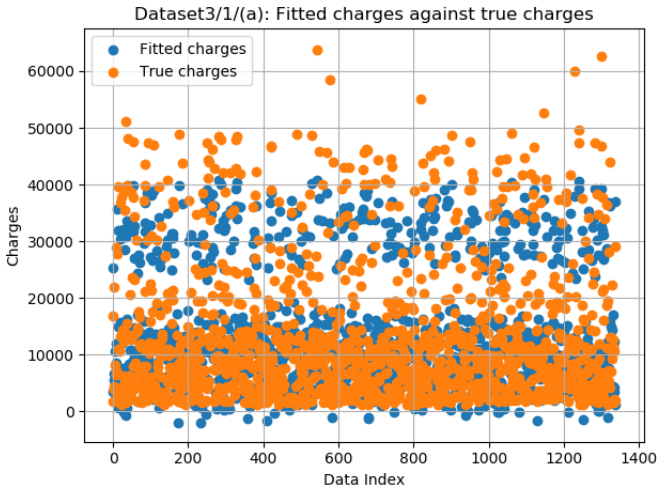

Figure 45: Fitted charges against true charges without standardizing categorical features

Average training RMSE: 6045.093.
Average testing RMSE: 6082.919.
The fitted against true charges plot and residual plot suggest that the linear regression does not fit very well.

(b) Standardization

Average training RMSE: 6045.093.
Average testing RMSE: 6082.919.
After standardization, the testing and training RMSE are still the same as before standardization. After we plot the fitted against true value and residual, the plots look the same as before standardization as well. With the standardization for linear regression, we get the exact same model, and thus same accuracy. This is why we get same testing RMSE, training RMSE, and same graphs.

(c) Divide Numerical Features

Average training RMSE: 6198.081.
Average testing RMSE: 6240.027.
After dividing feature 1 into 3 levels, the training and testing RMSE got

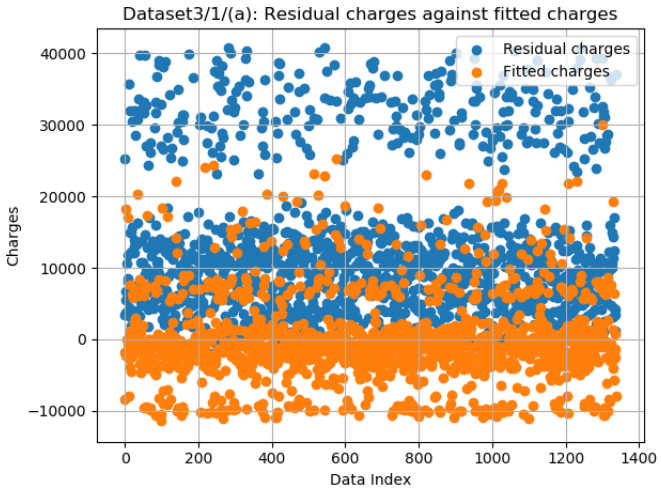

Figure 46: Residual charges against fitted charges without standardizing categorical features

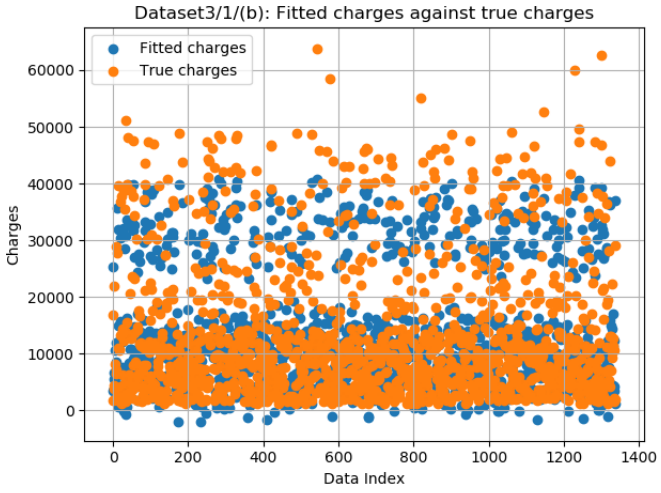

Figure 47: Fitted charges against true charges with standardizing categorical features

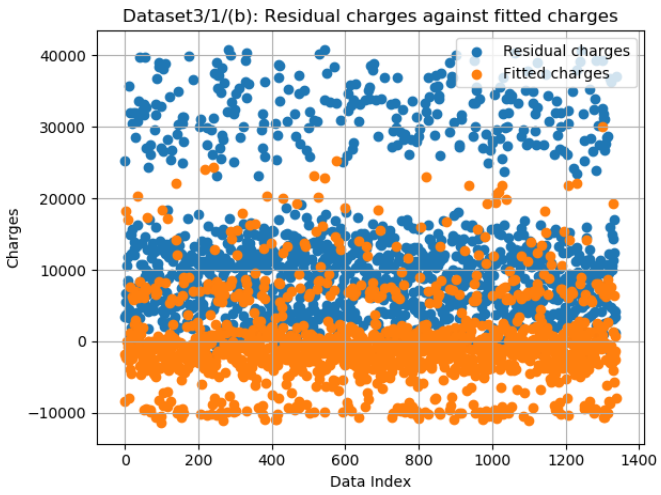

Figure 48: Residual charges against fitted charges with standardizing categorical features

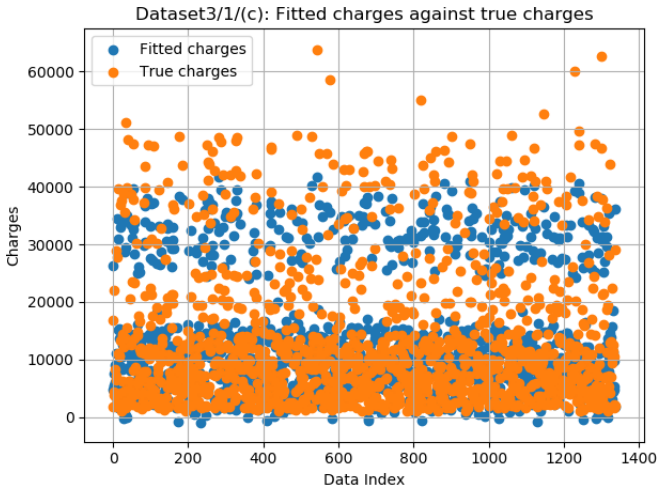

Figure 49: Fitted charges against true charges with dividing feature 1

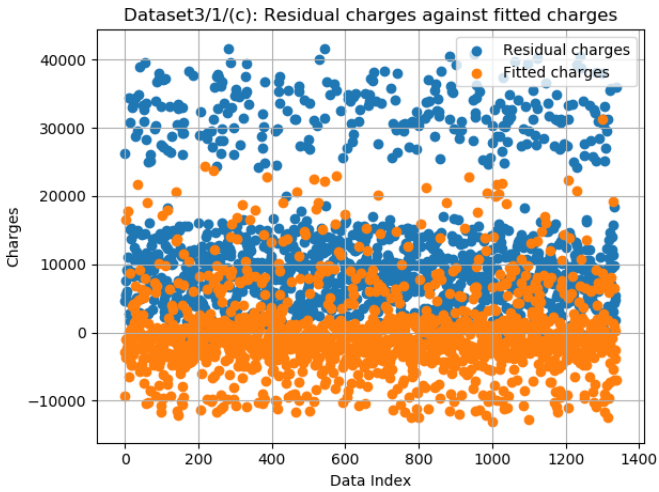

Figure 50: Residual charges against fitted charges with dividing feature 1

worse. The reason behind is probably due to a loss of information brought by such discretization.

3.2 Correlation Exploration

(a)

We use f regression and mutual information regression to select the two most important variables respectively. And we get:

F: 131.174, 54.709, 6.206, 4.400, 2177.615, 3.379

Mutual information: 1.500, 0.075, 0.161, 0.177, 0.369, 0.076

They all suggest that the first and fifth feafures are the most important ones selected.

(b)

The figure can be found at 51. It seems that the feature 2 whose feature 5 value is 0 has a seemingly linear relation with charges.

(c)

The figure can be found at 52. It seems that the feature 1 also has a seemingly linear relation with charges.

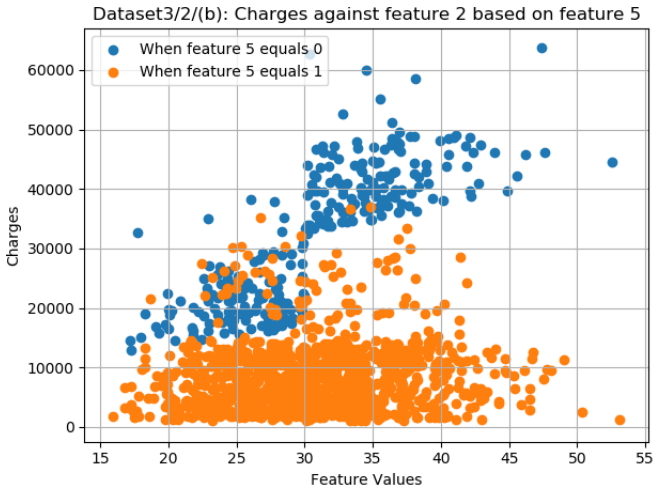

Figure 51: Charges against features 2 and color points based on feature 5

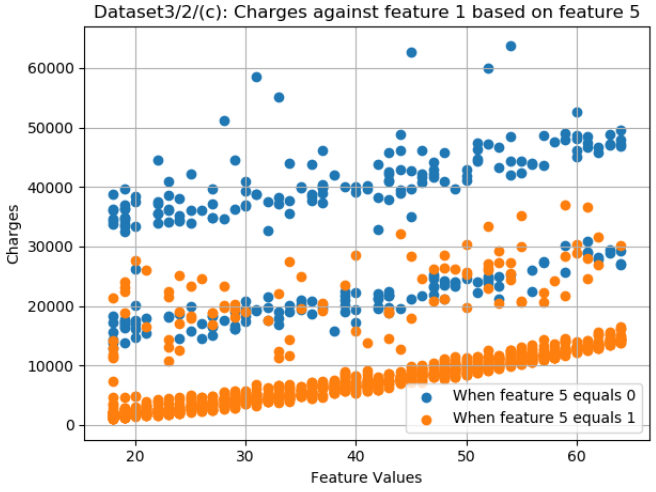

Figure 52: Charges against features 1 and color points based on feature 5

3.3 Modify the Target Variable

(a)

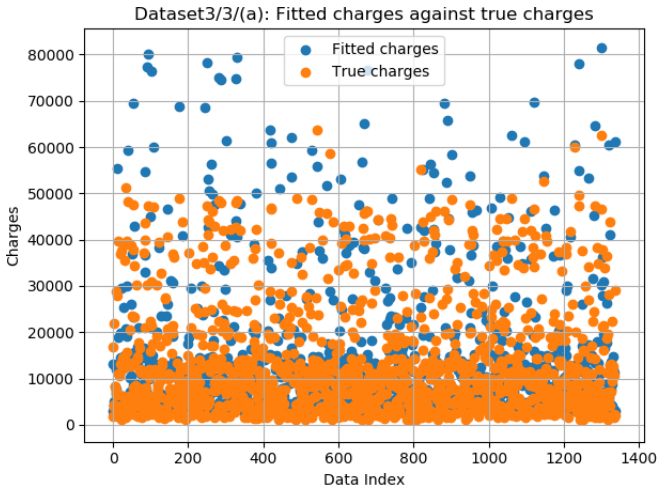

Figure 53: Fitted log of charges against true log of charges

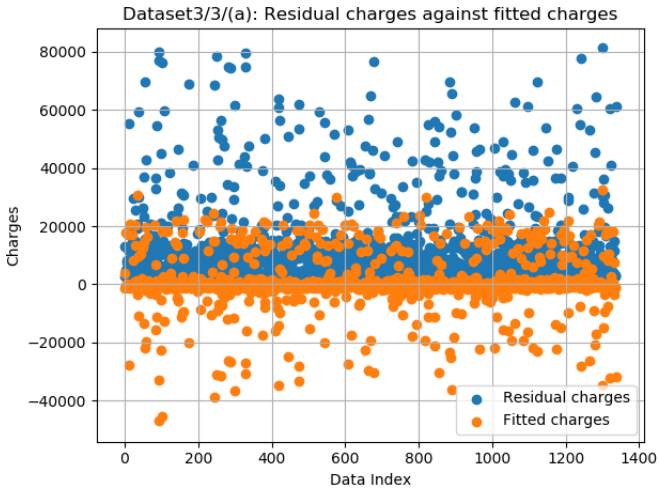

Figure 54: Residual log of charges against fitted log of charges

Average training RMSE: 8359.526.
Average testing RMSE: 8437.228.

The preprocessing we picked is standardizing all these numerical features and keeping the one-hot-encoded features. It seems from the results above that this technique failed to bring any improvement to the regression performance.

(b)

We use f regression and mutual information regression to select the two most important variables respectively. And we get:

F: 515.977, 23.936, 35.705, 0.042, 1062.124, 2.923

Mutual information: 1.501, 0.067, 0.161, 0.176, 0.369, 0.078

They all once again suggest that the first and fifth feafures are the most important ones selected.

3.4 Bonus Questions

(a)

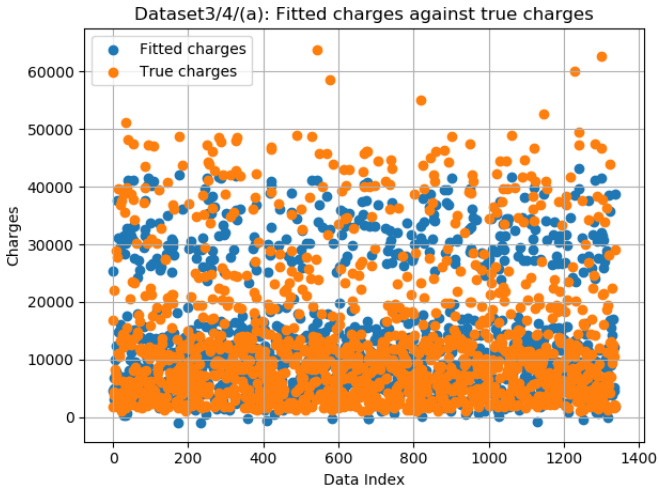

Figure 55: Fitted charges against true charges using polynomial features

Average training RMSE: 5999.819.

Average testing RMSE: 6052.478.

After trying polynomial features (applied on feature 1 and feature 5 with a factor of 2), the regression performance is successfully improved. These two features are selected based on the f regression and mutual information analysis conducted above. The possible reason for this improvement is probably due to the fact that the weights of the two most important features are increased, thus bringing more effective information to the regression model.

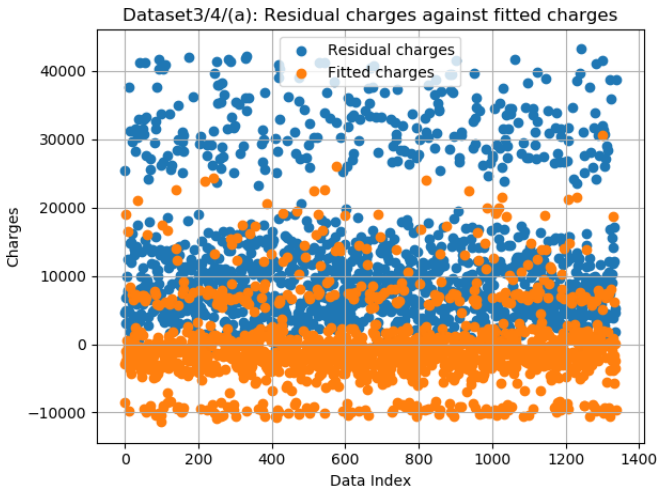

Figure 56: Residual charges against fitted charges using polynomial features

(b) Random Forest

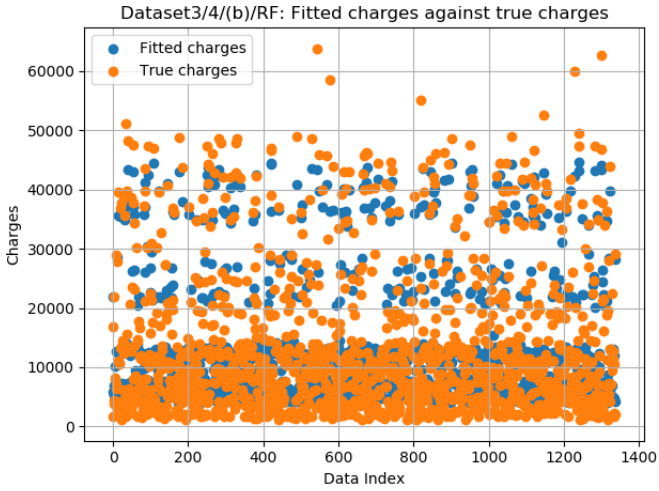

Figure 57: Fitted charges against true charges using random forest

Average training RMSE: 4483.339.
Average testing RMSE: 4681.565.

A random forest model is trained using the same feature preprocessing techniques applied in Question 1/(b). The optimal hyperparameters set for the

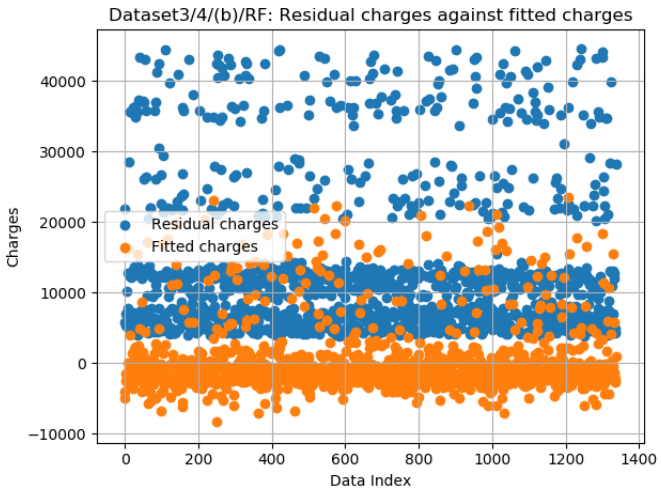

Figure 58: Residual charges against fitted charges using random forest

random forest model is fully explored using grid search. The number of estimators is scanned from 1 to 200. The number of max features is scanned from 1 to 5. Finally, the optimal number of estimators is 51 and the optimal number of max features is 5. The result is much better than that of a linear regression model. This improvement is brought by the higher model capacity of random forest.

(b) Neural Network

Average training RMSE: 4940.481.
Average testing RMSE: 5099.837.

A neural network model is trained using the same feature preprocessing techniques applied in Question 3/(a). The optimal hyperparameters set for the neural network model is also fully explored using grid search. The number of hidden layers is scanned from 5 to 250. The activation function is either ReLU or tanh. Finally, the optimal number of hidden layers is 250 and the optimal activation function is ReLU. The result is slight worse than that of a random forest model.

(b) K Nearest Neighbors

Average training RMSE: 4331.033.
Average testing RMSE: 5840.346.

A KNN model is trained using the same feature preprocessing techniques applied in Question 1/(b). The optimal hyperparameters set for the KNN model is also fully explored using grid search. The number of nearest neighbors is scanned from 1 to 50. Finally, the optimal number of nearest neighbors is 50.

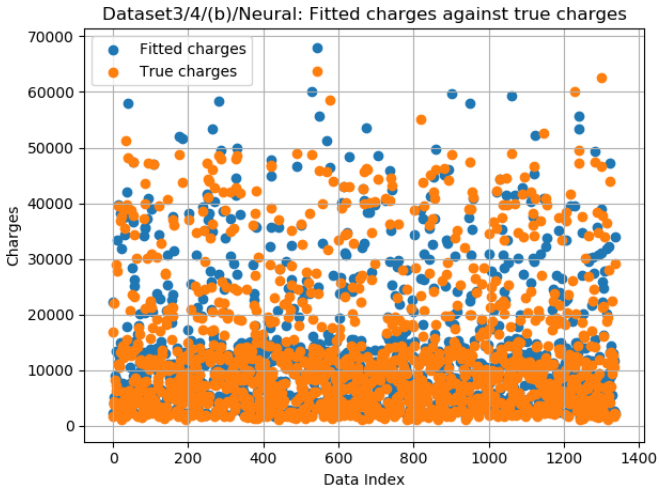

Figure 59: Fitted charges against true charges using neural network

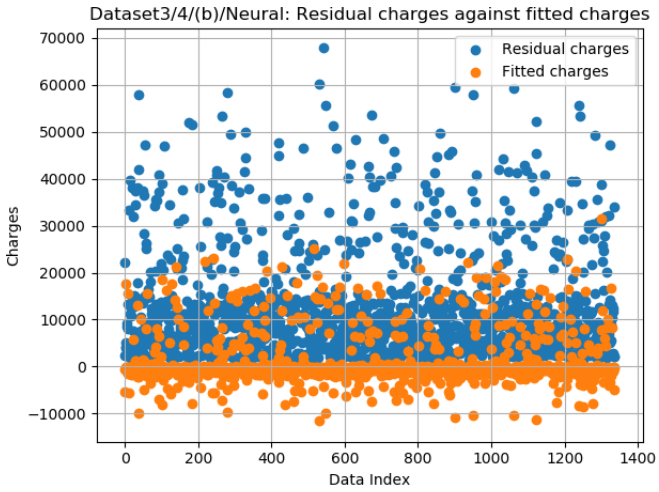

Figure 60: Residual charges against fitted charges using neural network

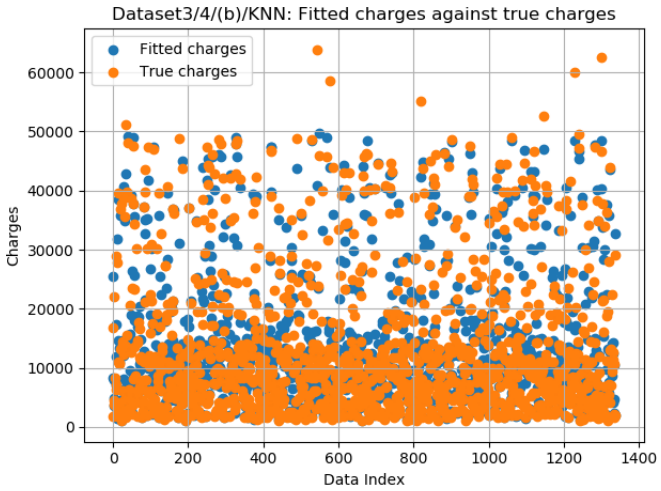

Figure 61: Fitted charges against true charges using KNN

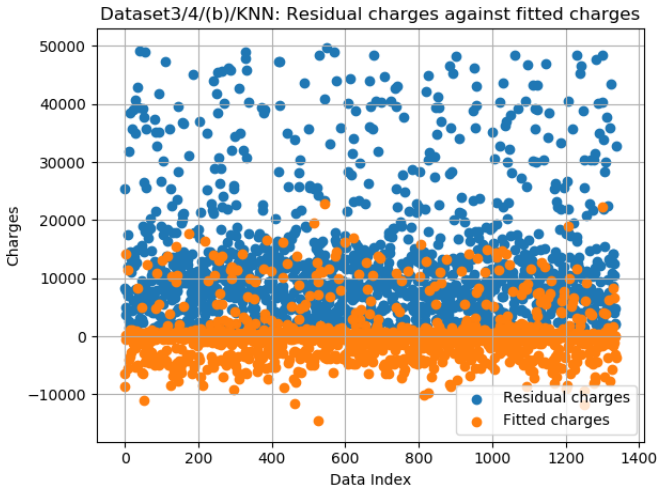

Figure 62: Residual charges against fitted charges using KNN

4 Conclusion

Linear Regression model and the random forest model are suitable for the dataset which performs periodically according to features.

For dataset such as car insurance dataset where many features are categorical, it seems that Random Forest is the best model to fit those data, followed by Neural Network, K Nearest Neighbor, and Linear Regression. The best preprocessing technique to apply to those categorical features is one-hot encoding and that to those numerial features is standardization.