EE219 Project 1

Regression Analysis

Winter 2017

Xiongfeng Hu, 304753117

Yanming Zhang, 004761717

Cong Peng, 904760493

PART I. Network backup Dataset

Problem 1

Load the dataset and get an idea on the type of relationships in the dataset by plotting the actual copy sizes of all the files on a time period of 20 days for each workflow.

Solution:

According to the requirement, we first preprocess the data by changing the attributes to numeric datatype for the purposes of further Regression model creation. We replace the attribute 'Day of Week' with 0~6 integers, representing Monday to Sunday, and generate a new attribute 'Time' by combination of 'Week #', 'Day of Week' and 'Backup Start Time – Hour of Day'.

Then we plot the actual copy sizes of all the files on a time period of 20 days for each workflow. We first select the first 20 days as the sample, and then group the data using groupby 'Work-Flow-ID' and groupby 'File Name'. Then we can plot the relationship between 'Size of Backup (GB)' and 'Time'. The figures are as follows.

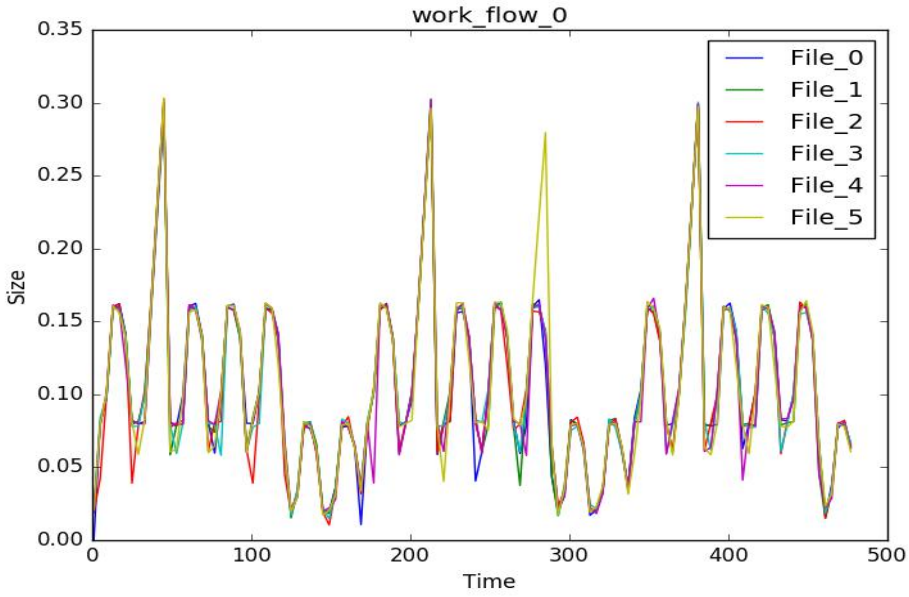


Fig. 1 Size of Backup - Time of work_flow_0

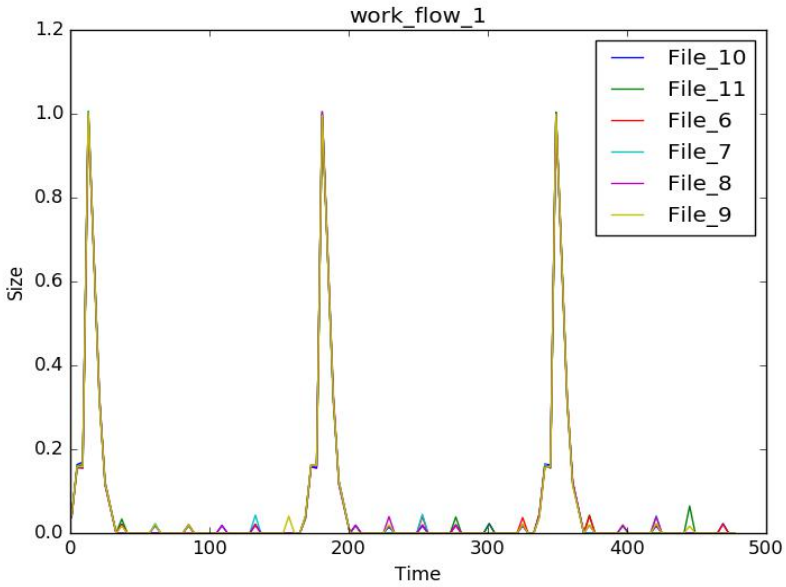


Fig.2 Size of Backup - Time of work_flow_1

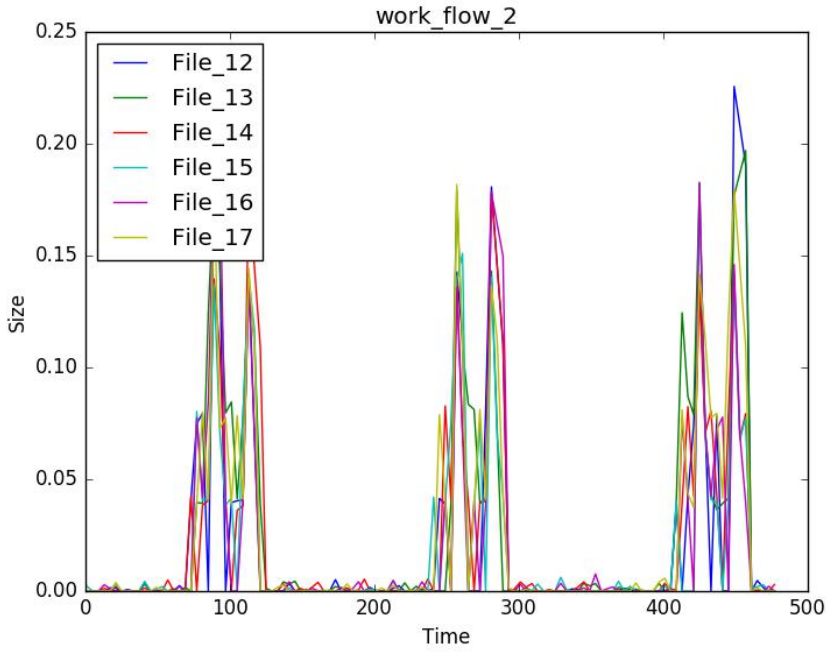


Fig.3 Size of Backup - Time of work_flow_2

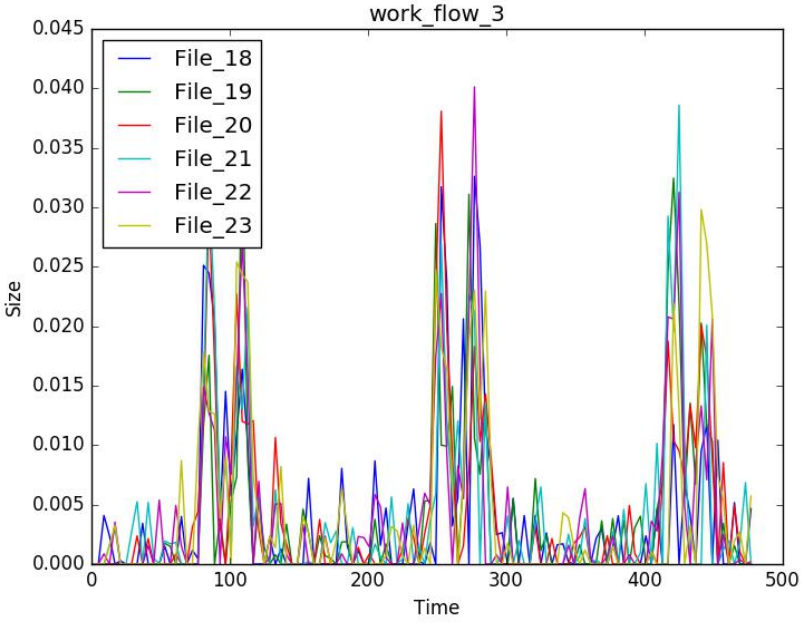


Fig.4 Size of Backup - Time of work_flow_3

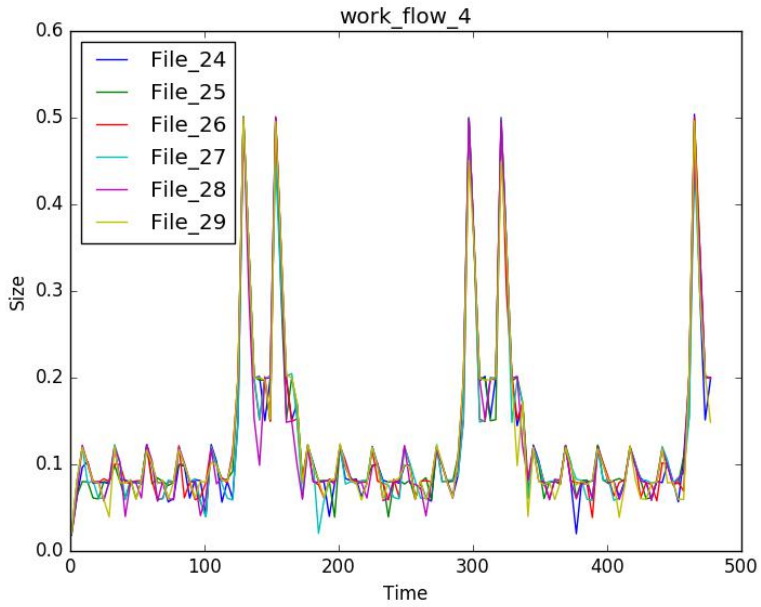


Fig.5 Size of Backup - Time of work_flow_4

From these figures, we can easily recognize patterns of different workflows. Each workflow has different patterns and all of them share the period of 7 days. We can also obtain some detailed features of the relationship between backup sizes of files and time. For example, for work_flow_4 (the figure above), backup sizes from every Monday to Friday are similar and relatively small, and there are obvious increases of backup sizes on every Saturday and Sunday.

Problem 2

Predict the copy size of a file given the other attributes.

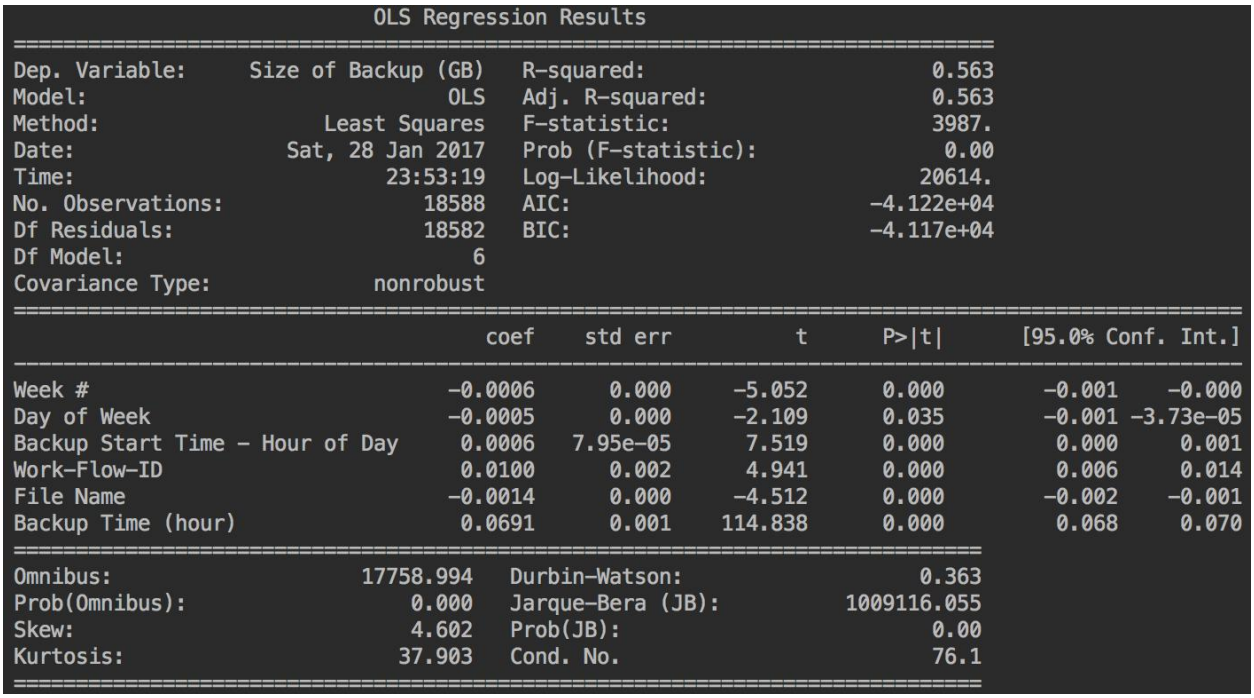
(a) Linear Regression

Fit a linear regression model with backup size as the target variable and the other attributes as the features. Perform a 10-fold cross validation. And analyze the significance of different features and the Root Mean Squared Error (RMSE). Evaluate how well our model fits the data by providing “Fitted values and actual values scattered plot over time”, and “residuals versus fitted values plot”.

Solution:

We first preprocess the data and then use OLS in Pandas statsmodels library and linear_model in Scikit-Learn Libraries to create the model and fit the data.

The Figure 6 shows the summary of the OLS Regression Results of our data. The column of "P>| t |" shows the p-value of different variables. As we can see from the figure, the p-value of 'Day of Week' is 0.035, which is still less thatn the common α level of 0.05. The p-value of other variables are 0.000, showing a strong relationships between them and the Backup Size.



OLS Regression Results

Dep. Variable:	Size of Backup (GB)	R-squared:	0.563
Model:	OLS	Adj. R-squared:	0.563
Method:	Least Squares	F-statistic:	3987.
Date:	Sat, 28 Jan 2017	Prob (F-statistic):	0.00
Time:	23:53:19	Log-Likelihood:	20614.
No. Observations:	18588	AIC:	-4.122e+04
Df Residuals:	18582	BIC:	-4.117e+04
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
Week #	-0.0006	0.000	-5.052	0.000	-0.001	-0.000
Day of Week	-0.0005	0.000	-2.109	0.035	-0.001	-3.73e-05
Backup Start Time - Hour of Day	0.0006	7.95e-05	7.519	0.000	0.000	0.001
Work-Flow-ID	0.0100	0.002	4.941	0.000	0.006	0.014
File Name	-0.0014	0.000	-4.512	0.000	-0.002	-0.001
Backup Time (hour)	0.0691	0.001	114.838	0.000	0.068	0.070

Omnibus:	17758.994	Durbin-Watson:	0.363
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1009116.055
Skew:	4.602	Prob(JB):	0.00
Kurtosis:	37.903	Cond. No.	76.1

Fig.6 OLS Regression Results

We then perform a 10-fold cross validation. The best Root Mean Squared Error (RMSE) that we get is 0.0796044888128 and the coefficients are

[2.97492409e-05, 1.30817809e-03, 9.76184764e-04, 1.66474886e-03, 2.02427098e-04, 7.12596790e-02] with Intercept of -0.0276726701584.

In order to evaluate how well our model fits the data, we plot "Fitted values and actual values scattered plot over time" and further plot "Fitted values vs. actual values" to show a clearer relationship.

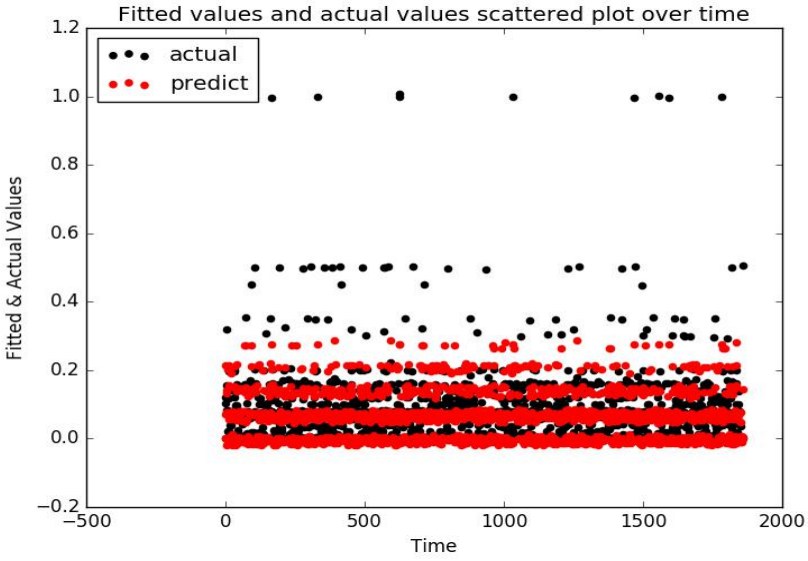
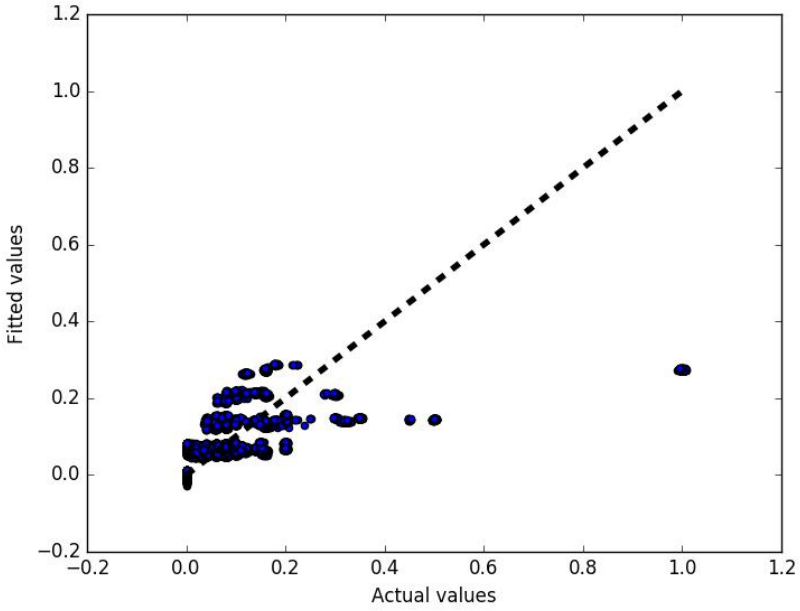


Fig.7 Fitted values and actual values scattered plot over



time

Fig.8 Fitted values vs. actual values

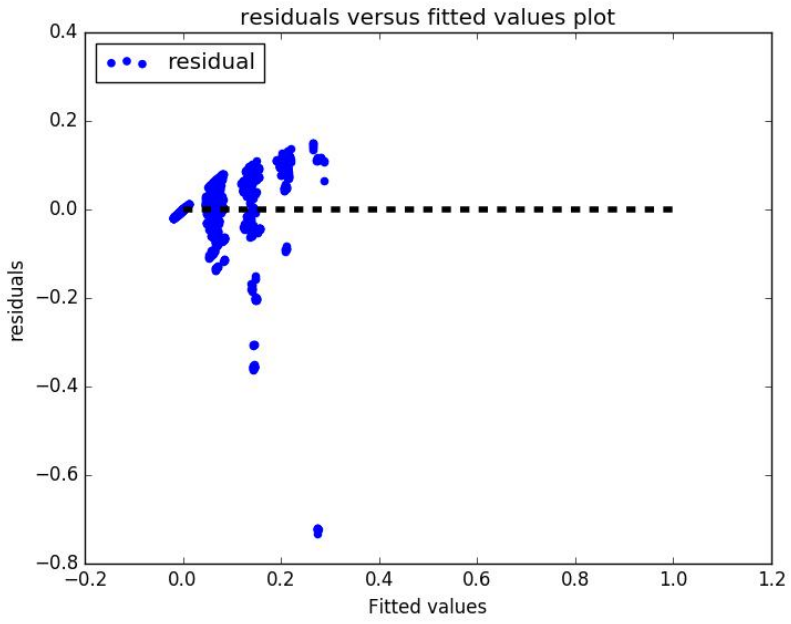


Fig.9 Residuals vs. fitted values

As we can see from these figures, majority of the fitted values have a small error compared to the actual values by our model. But there is still some outliers in this model.

Our conclusion is that the linear regression model does not perform well, the residuals cannot be ignored and the outliers are not handled properly. Actually, the r2 score of this model is 0.4163677626, while best possible score should be 1.

(b) Random Forest Regression

We use GridSearchCV to tune the parameters of the model. We first initialize the parameters with the following values: best_num_of_trees = 20, best_max_depth = 4, best_max_features = 6. Then we fix two of the parameters and tune the other one.

We first tune the parameter of max_depth. The tuning results are as follows.

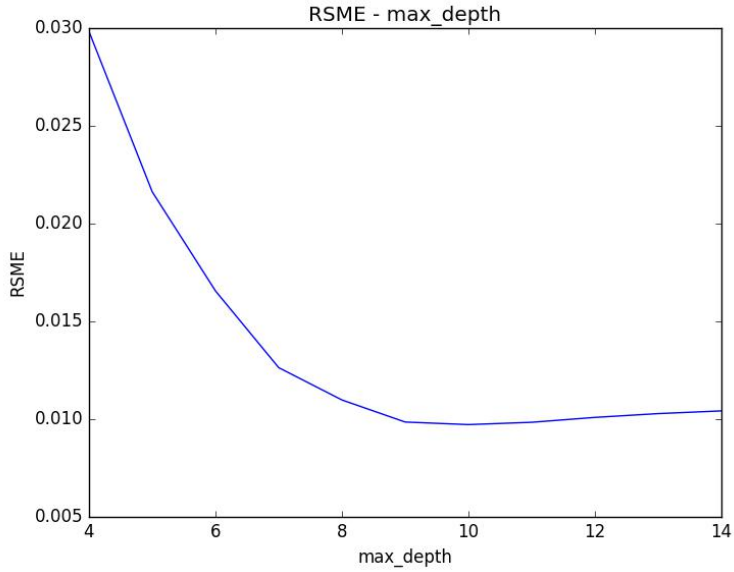


Fig.10 RMSE – max_depth

The tuning range is [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], and the best RMSE that we can get is 0.00964470407733 when max_depth = 10.

Then we set the max_depth as 10 and tune the parameter of number of trees(n_estimators). The tuning results are as follows.

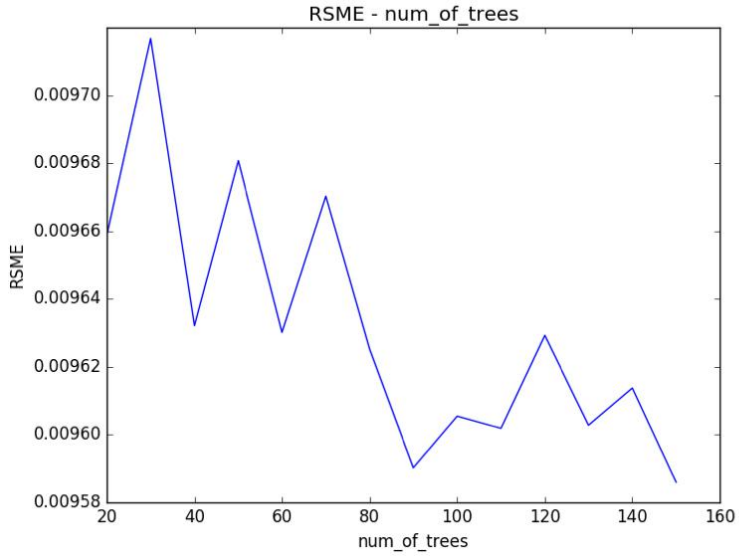


Fig.11 RMSE – num_of_trees

The tuning range is [20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150], and the best RMSE that we can get is 0.00958596247387 when num_of_trees 150. This is reasonable, because the performance will get better when num_of_trees become larger, but the tradeoff is the computing time.

At last, we further try to tuning the parameter of the max_features. The tuning results are as follows.

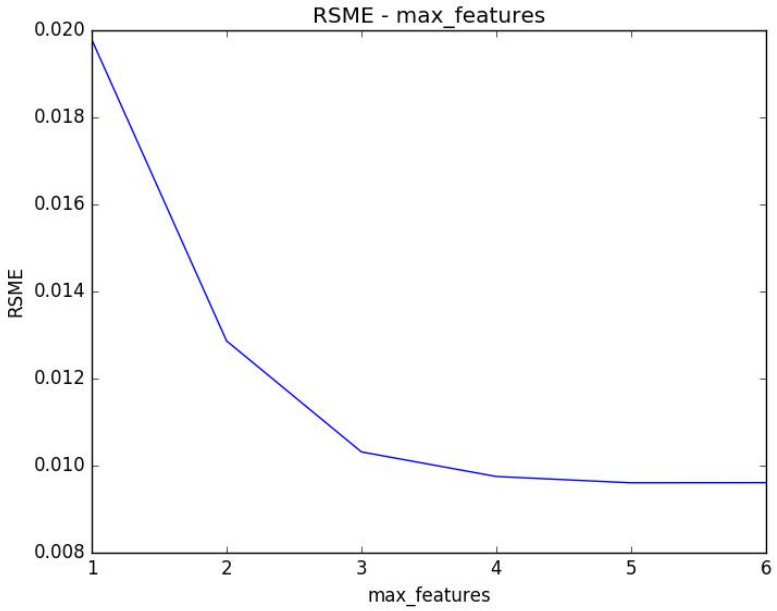


Fig.12 RMSE – max_features

The tuning range is [6, 5, 4, 3, 2, 1], and the best RMSE that we can get is 0.00959746161422 when max_features = 5.

Therefore, we get the best parameters:

best_num_of_trees: 150, best_max_depth: 10, best_max_features 5

And then we use the best parameters to set the model and get the coefficients and scores as follows:

Coefficients: [0.00131858, 0.30700416, 0.08937096, 0.0548849, 0.08131822, 0.46610317]

RMSE: 0.00962793692127

R2 score: 0.991392636154

The RMSE value obtained from Random Forest is 0.00962793692127 and that of Linear Regression is 0.0796044888128. The R2 value obtained from Random Forest = 0.991392636154 and that of Linear Regression is 0.4163677626. Therefore, Random Forest provides a much better model for prediction than Linear Regression.

The coefficients of the features are [0.00131858, 0.30700416, 0.08937096, 0.0548849, 0.08131822, 0.46610317], corresponding to ['Week #', 'Day of Week', 'Backup Start Time - Hour of Day', 'Work-Flow-ID', 'File Name', 'Backup Time (hour)']. Therefore, we can say, from this model, 'Backup Time (hour)', 'Day of Week' play most important roles in the model, 'Backup Start Time - Hour of Day', 'Work-Flow-ID', 'File Name' also have influence on the backup size. However, 'Week #' is trivial for the backup size, which we already know from the 7-day period of the patterns in problem1. This is also in accordance with the best_max_features = 5.

(c) Neural Networks

In this part, a feed forward neural network model is implemented and trained by given data in 10-fold cross validation methodology to make predictions. A few experiments are conducted to study the effect of particular major parameters. The best model is selected from above experiments based on RMSE and R^2 score. And "Fitted values and actual values scattered plot" and "residuals versus fitted values plot" are provided to further evaluate how well the model fits the data.

1) Pre-processing of data

We're faced with some challenges to process our raw dataset. First, some data is in language/word format, not numerical form, which we need to convert into a vector of features. So Day of Week features {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday} are mapped into {0, 1, 2, 3, 4, 5, 6}, Workflow-IDs are mapped into {0, 1, 2, 3, 4, 5} and File Names are mapped into {0, 1, 2,..., 29}. Furthermore, Day of Week and Week # are combined into one label - Day # which is {0, 1, 2,..., 153}. The conversion algorithm is: [Day #] = [Week #] * 7 + [Day of Week].

2) Major parameters and corresponding effects

Our major parameters are the number of data our model process once - batch_size, the number of nodes of each hidden layer - layer_units, and the number of layers. As for other parameters, we

verify that the reasonable range of epoches is 10 -14 in order to avoid overfitting. The activation function type for hidden layer is set to be 'Rectifier' and 'Linear' for output layer. And learning rate is 0.01 which can shorten the runtime as possible without sacrificing the accuracy of prediction.

i) batch_size

Below we use neural network model with one hidden layer of 10 nodes and 4 different batch_sizes {1, 10, 100, 1000} to evaluate the relationship between batch_size and RMSE performance.

a. layer_units = [10] batch_size = 1

RMSE = 0.06676271217146426

r2 = 0.580381858027

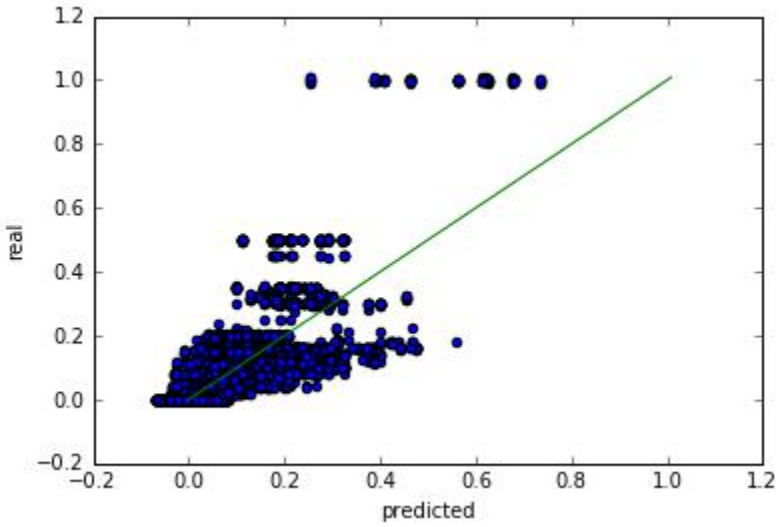


Fig.13 Predicted value vs real values, layer_units = [10] batch_size = 1

b. layer_units = [10] batch_size = 10

RMSE = 0.07579391630937388

r2 = 0.425878628321

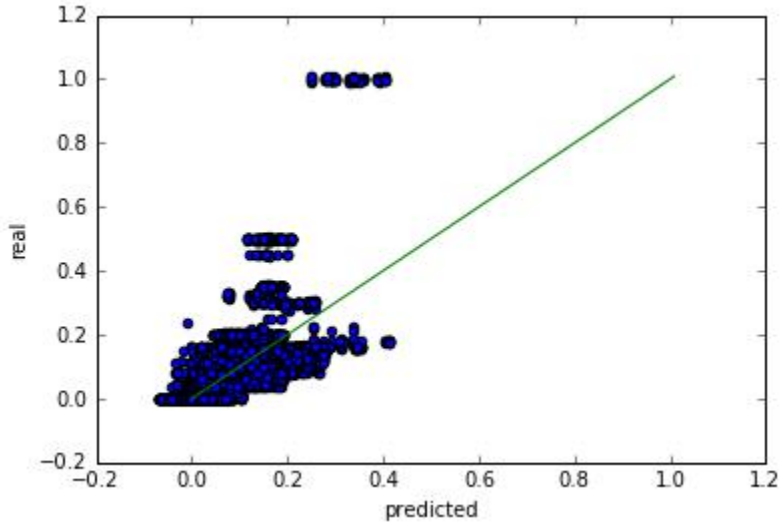


Fig.14 Predicted value vs real values, layer_units = [10] batch_size = 10

c. layer_units = [10] batch_size = 100

RMSE = 0.09283168743902143

r2 = 0.246362075828

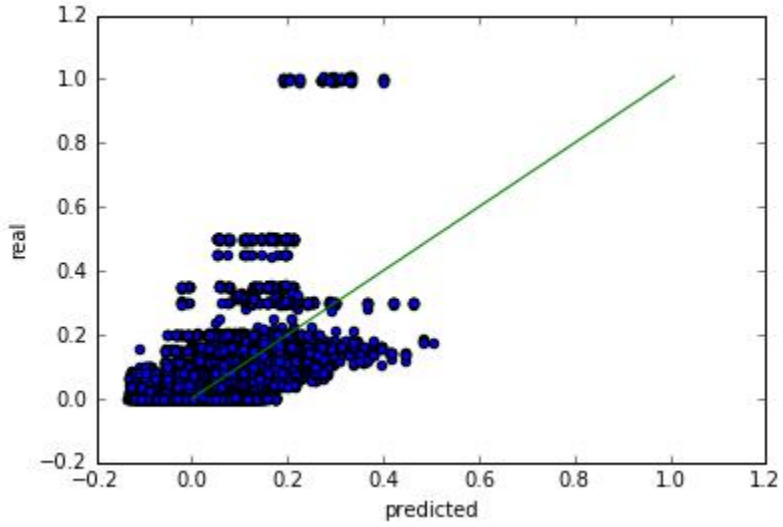


Fig.15 Predicted value vs real values, layer_units = [10] batch_size = 100

d. layer_units = [10] batch_size = 1000

RMSE = 0.19104889442907128

r2 = -1.67009351517

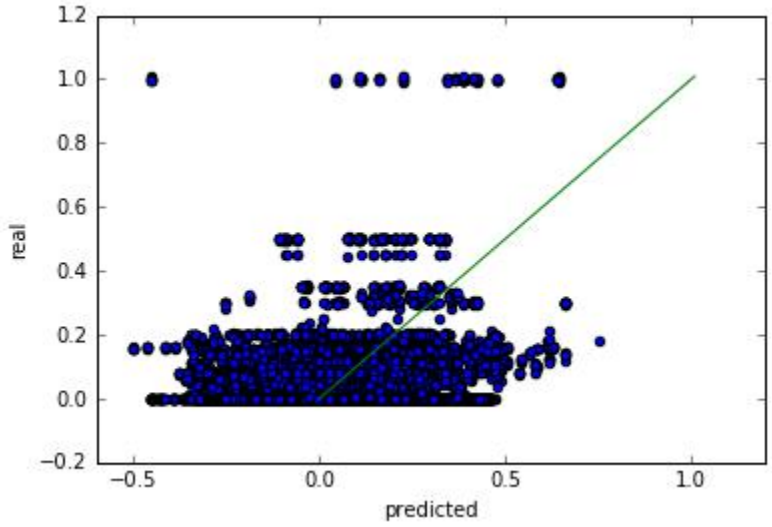


Fig.16 Predicted value vs real values, layer_units = [10] batch_size = 1000

Conclusion:

Above experiments imply that lower batch_size leads to better RMSE performance. When batch_size = 1 we have the best result RMSE = 0.0667627121714642, r2 = 0.580381858027 and when batch_size = 1000 we have the worst result RMSE = 0.19104889442907128, r2 = -1.67009351517. r2 score is coefficient of determination, best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse).

However, even if the batch_size has a negative effect on the predicted results, the relationship is nonlinear.

ii) Layer size

In this part, we have fixed batch_size = 1, and try different layer sizes using one hidden layer model.

a. layer_units = [1] batch_size = 1

RMSE = 0.09232411121562661

r2 = 0.222182585324

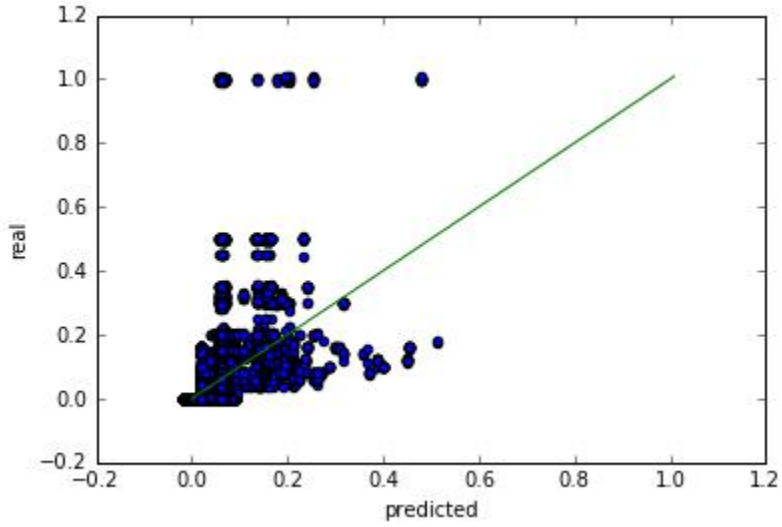


Fig.17 Predicted value vs real values, layer_units = [1] batch_size = 1

b. layer_units = [10] batch_size = 1

RMSE = 0.06676271217146426

r2 = 0.580381858027

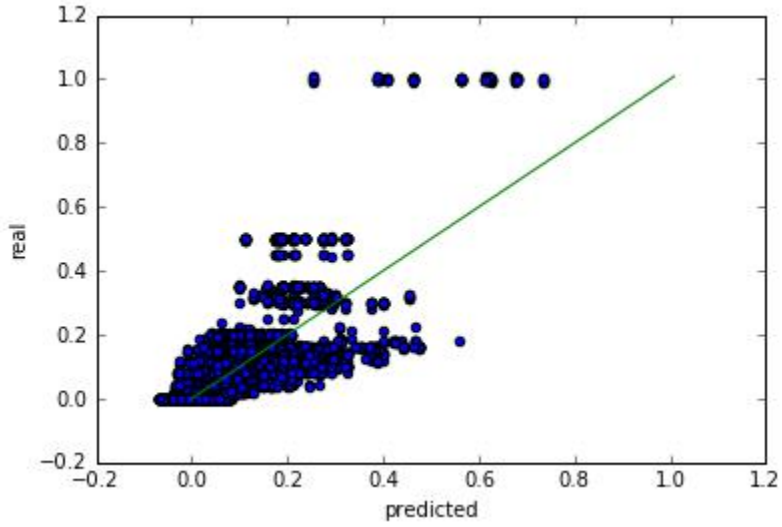


Fig.18 Predicted value vs real values, layer_units = [10] batch_size = 1

c. layer_units = [100] batch_size = 1

RMSE = 0.0444149838058606

r2 = 0.839944355941

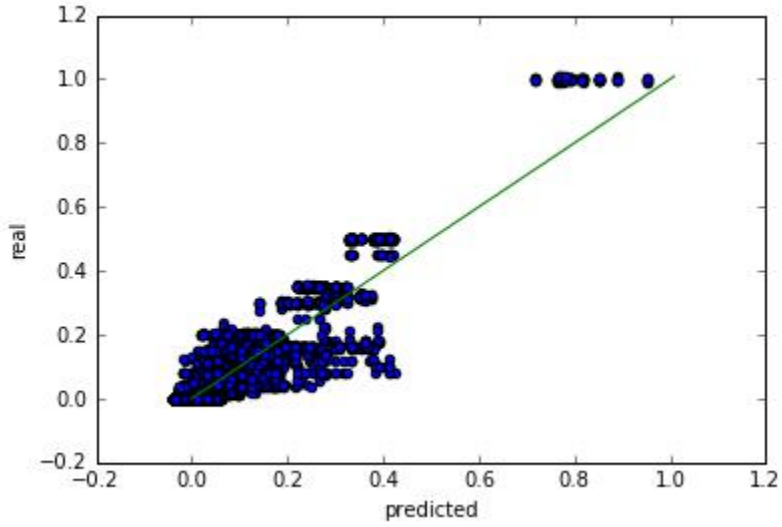


Fig.19 Predicted value vs real values, layer_units = [100] batch_size = 1

d. layer_units = [1000] batch_size = 1

RMSE = 0.04077953303724675

r2 = 0.831556751441

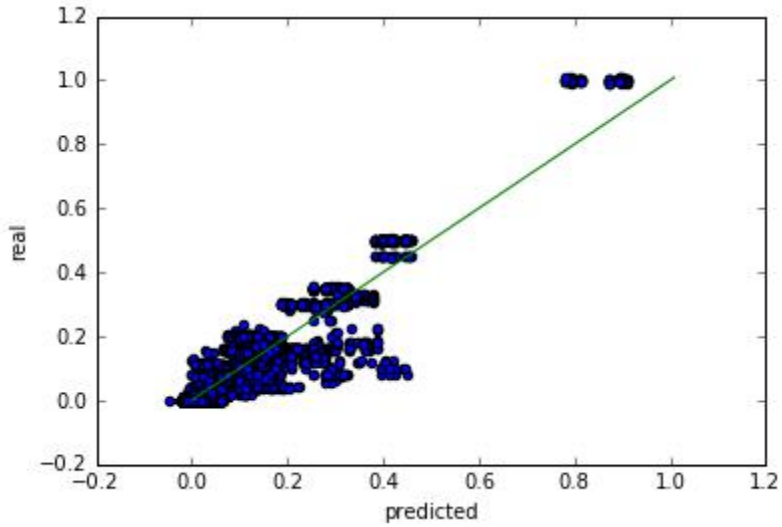


Fig.20 Predicted value vs real values, layer_units = [1000] batch_size = 1

Conclusion:

In this part, the number of layer size is changing from small to large and we have try four different layer sizes {1, 10, 100, 1000} which have 10 times difference between neighbors.

Figures and results show that the layer size has a positive influence on predicted results, which means when layer size comes larger the results comes better. Layer size = 1 corresponds to the worst result in experiments RMSE = 0.09232411121562661, r2 = 0.222182585324 and layer size = 1000 corresponds to the best result RMSE = 0.04077953303724675 ,r2 = 0.831556751441. But the results tell us the relationship between layer size and prediction results is nonlinear and the positive effect of layer size has a diminishing return. The performance between layer size = 100 and layer size = 1000 is very small, therefore in order to make our model as simple as possible we will use layer size = 100 rather than 1000 in further experiments.

iii) Number of hidden layers

In this part, we will test multiple layer neural networks model and study the influence of number layers on prediction accuracy.

a. layer_units = [100] batch_size = 1

RMSE = 0.0444149838058606

r2 = 0.839944355941

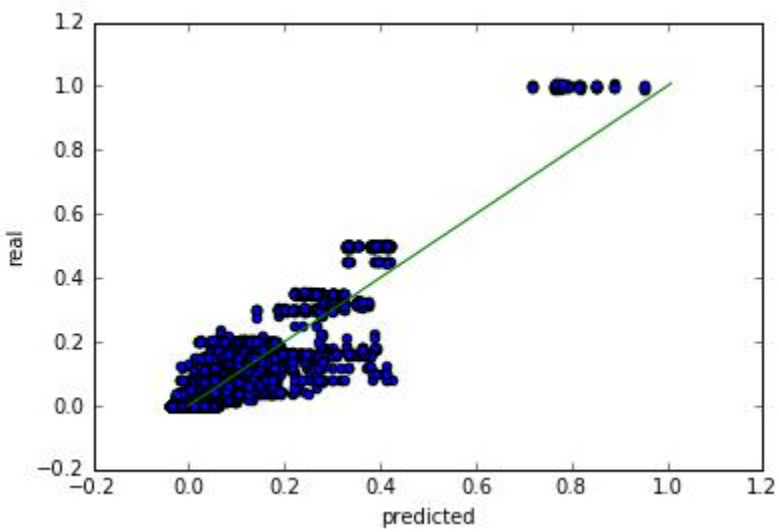


Fig.21 Predicted value vs real values, layer_units = [100], batch_size = 1

b. Two hidden layers:

layer_units = [100, 10] batch_size = 1

RMSE = 0.036545657868091976

r2 = 0.867744471264

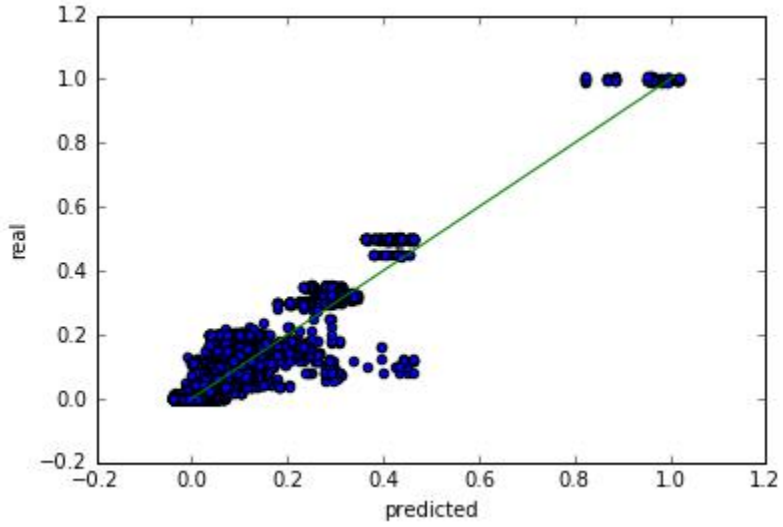


Fig.22 Predicted value vs real values, layer_units = [1000], batch_size = 1

c. Three hidden layers:

layer_units = [100, 100, 10] batch_size = 1

RMSE = 0.035855885201561355

r2 = 0.881567291831

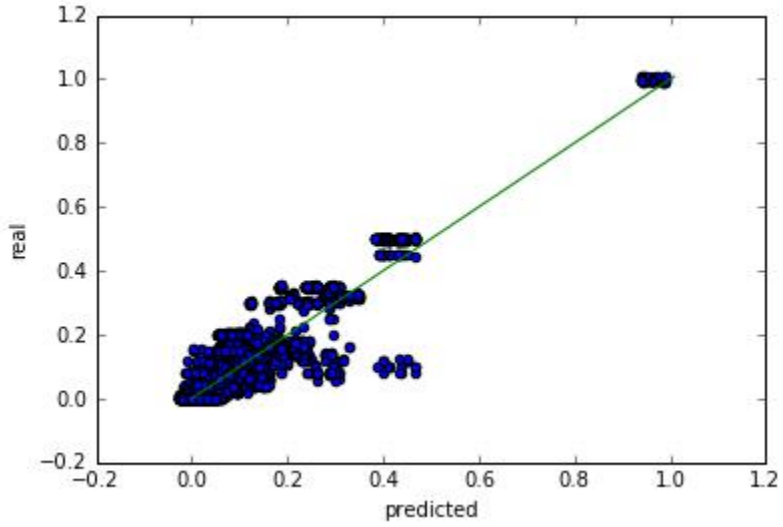


Fig.23 Predicted value vs real values, layer_units = [1000], batch_size = 1

d. Four hidden layers：

layer_units = [100, 100, 10, 10] batch_size = 1

RMSE = 0.03528450095162287

r2 = 0.885311803998

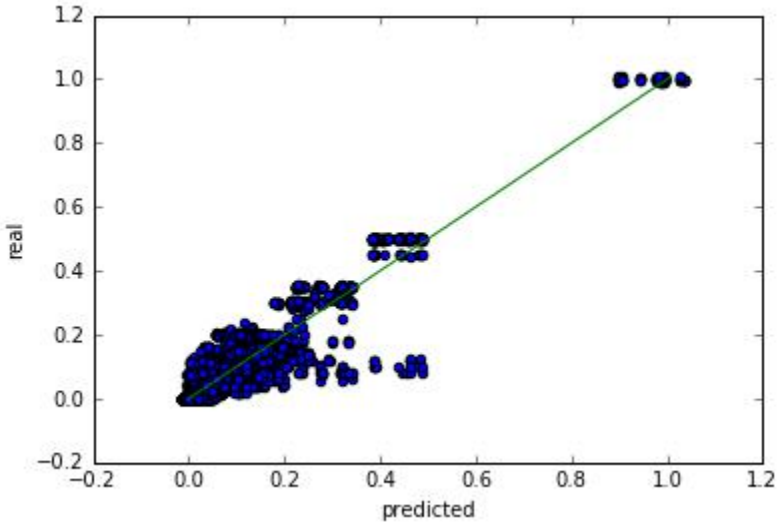


Fig.24 Predicted value vs real values, layer_units = [1000], batch_size = 1

Conclusion:

In this part we try four neural networks models with four different number of hidden layers ranging from 1 to 4. And above test results inform that as we add one layer each time, RMSE decreases from 0.0444149838058606 to 0.03528450095162287 and r2 increases from 0.839944355941 to 0.885311803998. Even though the trend is obvious, the absolute difference among models are not every obvious especially for three-hidden-layer model and four-hidden-layer model. When number of layers increase, the model becomes more complicated and the possibility of overfitting increases. Therefore, three-hidden-layer model is enough for prediction presicion.

3) Best Model and fitting evaluation

Based on above experiments, the best model we choose is three-hidden-layer neural network model with detailed parameters and testing results as below:

layer_units = [100, 100, 10] batch_size = 1

RMSE = 0.035855885201561355

r2 = 0.881567291831

In order to further evaluate how well this model fits our data, we have a fitted values v.s. residuals plot as below:

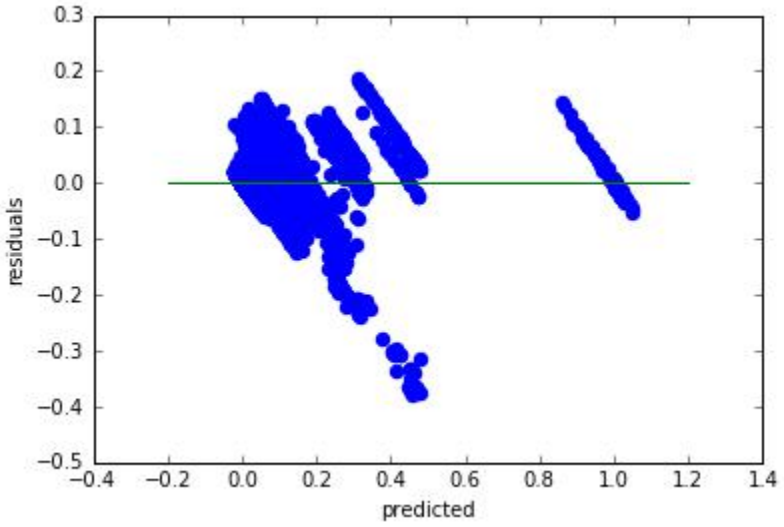


Fig.25 Predicted value vs residuals, layer_units = [100, 100, 10], batch_size = 1

Problem 3

1) Predict the Backup size for each of the workflows separately

In this part, we still use 10-fold cross validation for linear regression model, but perform on each workflow separately.

workflow#	RMSE_CV	r2	number of samples
0	0.03541777128245046	0.594350525679	3687
1	0.10377211175129185	0.602623917055	3600
2	0.025605404535885327	0.659197491653	3741
3	0.005941262537976126	0.339850492673	3780
4	0.10199772898463291	0.0366542711383	3780
Total	0.054546855818447336	0.44653533963965553	

From the table, we improve the average RMSE from 0.0796 to 0.0545.

Conclusion:

From above table, we can conclude that RMSE decrease from from 0.0796044888128 to 0.054546855818447336 compared with all workflows dataset. The separate workflows perform better than all data together which means different workflow has a different pattern of changing on backup size.

2) Polynomial regression for each of the workflows separately

workflow 0

degree	RMSE	RMSE_CV
1	0.035587882981460525	0.03541777128245046
2	0.0334440325988276	0.033674913527946354
3	0.03571365074133022	0.03284106844143193
4	0.0320266482366199	0.034966718205182545
5	0.03167052687351986	0.03794509373349786

==========

workflow 1

degree	RMSE	RMSE_CV
1	0.10629289053192541	0.10377211175129183
2	0.04791694491544227	0.04875372239441639
3	0.009982929666993497	0.014890919446841793
4	0.0029135431270720692	0.0031062531050998453
5	0.0024999080281927696	0.0032902323784129127

==========

workflow 2

degree	RMSE	RMSE_CV
1	0.026608082670442023	0.03541777128245046
2	0.02291378223014487	0.033674913527946354
3	0.022979234232048427	0.03284106844143193
4	0.02184074004679266	0.034966718205182545
5	0.023988640670321613	0.03794509373349786

==========

workflow 3

degree	RMSE	RMSE_CV
1	0.006342713182954391	0.005941262537976126
2	0.005634097369173509	0.005782022880574916
3	0.0051788694577714735	0.0057253923029786935
4	0.005961473977817003	0.006022253422175451

5	0.014551483689120553	0.010402735290184206

==========

workflow 4

degree	RMSE	RMSE_CV
1	0.09097726307299528	0.10199772898463291
2	0.09366650992168386	0.08879751744399497
3	0.08295362165127225	0.08833329587473754
4	0.07506385352575103	0.0744269370318884
5	0.0547633691763881	0.09963992045203147

Conclusion:

The above experiments show that degree 3 is best for workflow 3, degree 4 is best for workflow 4 and degree 5 is best for workflow 1, workflow 2 and workflow 4. Again, this results confirm that different workflow may have different pattern of backup size changing and hence different thresholds for fit polynomials.

Above results also inform that different data set may generate different best-fit polynomial, so for particular data set cross validation let model traverse all data in model training which help reduce model complexity and overfitting possibility and make model more general for prediction.

PART II. Boston Housing Dataset

Problem 4

1) Linear Regression

In linear regression, we model the dataset on all the features and predict the target (MEDV). Also, we perform a 10-fold cross validation, which we split the data randomly into 10 parts and each time take 90% of the data for training and intentionally regard the other 10% to have an unknown response variable for testing. We obtain the following parameters from the model with randomly-chosen dataset.

Best RMSE: 3.926

Best r2_score: 0.834

Optimal coefficient vector: [-0.109, 0.035, 0.009, 2.834, -18.359, 3.870, 0.000, -1.412, 0.298, -1.058, -0.965, 0.001, -0.528]

Moreover, we analyze the significance of different variables with statistics obtained from the model I have already trained, and it returns with **p-value: 0.0099**

Finally, we evaluate how well the model fits the data by providing "Fitted values and actual values scattered plot over time", and "residential vs fitted values plot". For simplicity, we only plot 10% of the entire dataset, which is only the test data after 10-fold split.

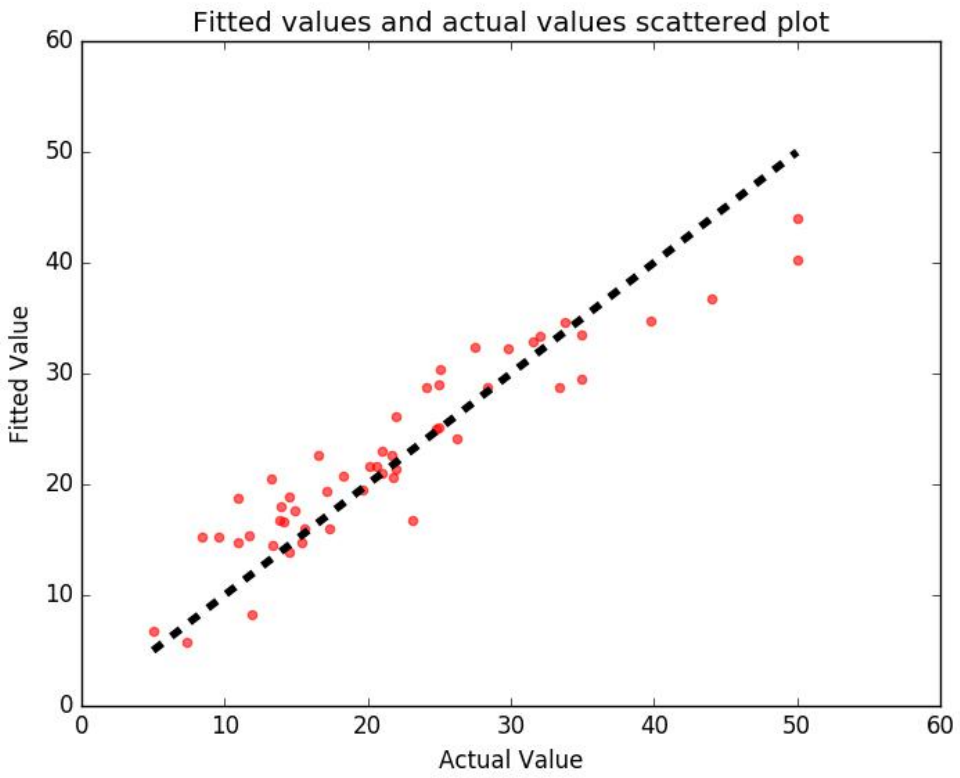


Fig.26 Fitted values and actual values scattered plot

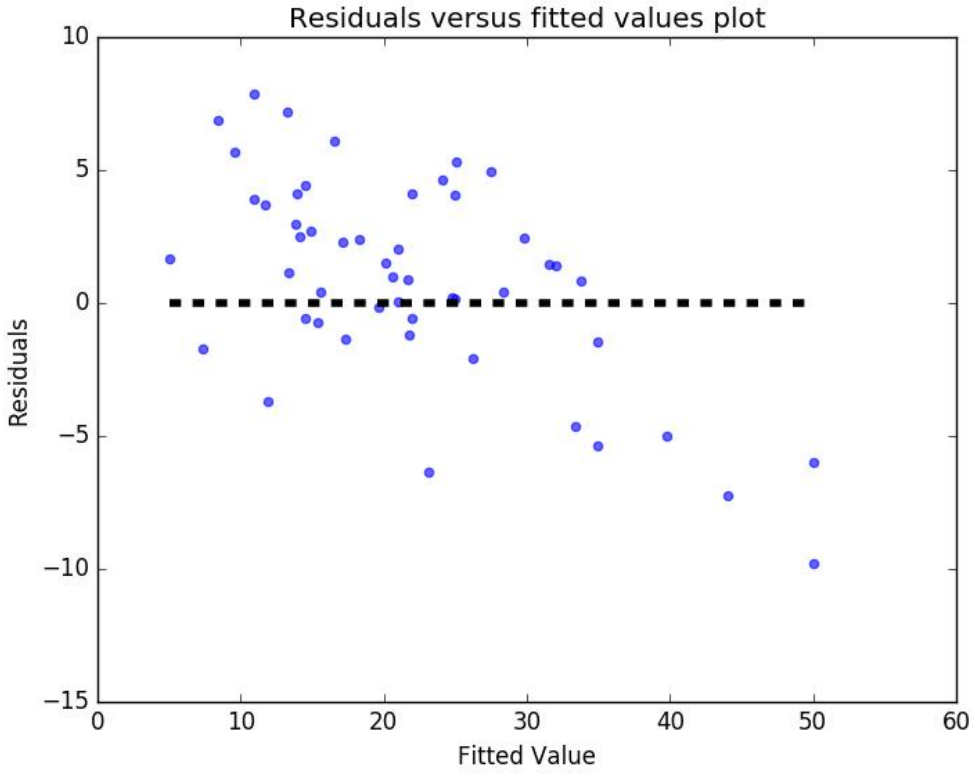


Fig.27 Residuals vs fitted values plot

2) Polynomial Regression

Similarly, we re-model the data into polynomial regression under 10-fold cross validation. We try increasing the degree of the polynomial to improve the fit. We cover the degree from 1 to 9, and plot the RMSE of the trained model against the degree of the polynomial respectively.

The best polynomial regression in our case is with degree 1 to 4. According to the curve below, there is a significant increase in RMSE. Hence, we can safely set the threshold to degree of 4.

Best degree: 1
Best RMSE: 1.446
Best r2_score: 0.855

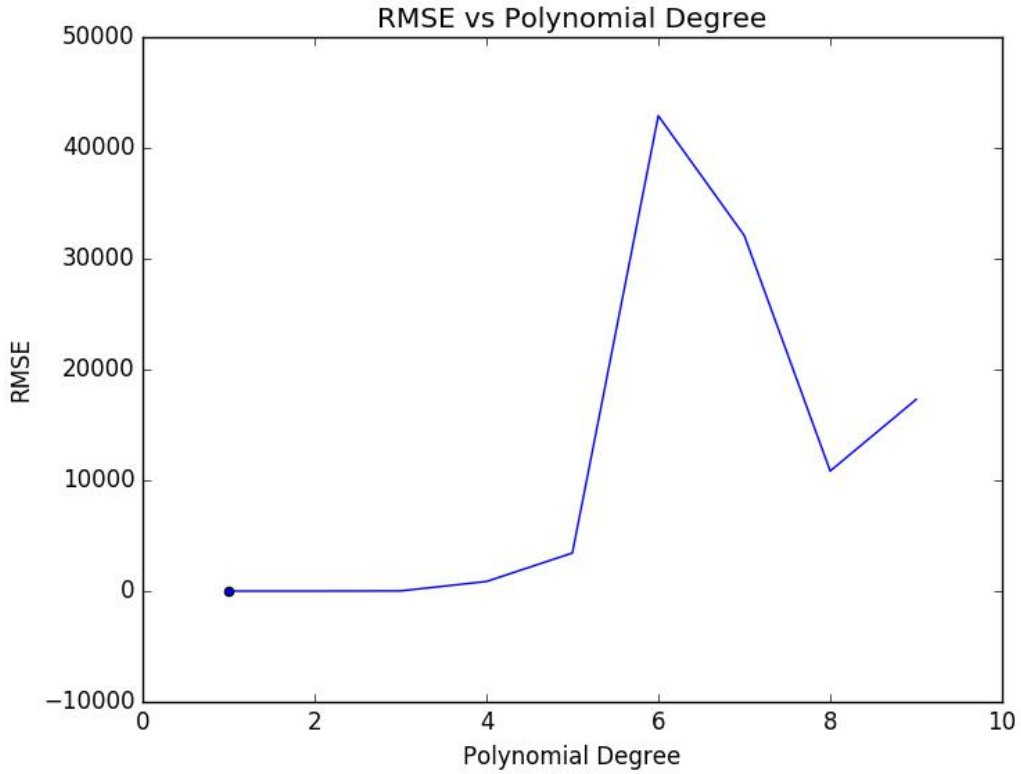


Fig.28 RMSE vs Polynomial Degree

Problem 5

1) Ridge Regression

We tune the complexity parameter α of the ridge regression below in the range {1, 0.1, 0.01, 0.001} and report best RMSE via 10-fold cross validation.

We obtain the following parameters through 10-fold cross validation, and plot the figures as required only in the test case, (10% of the dataset for simplicity).

Optimal α: 1
Best RMSE: 3.442
Best r2_score: 0.835

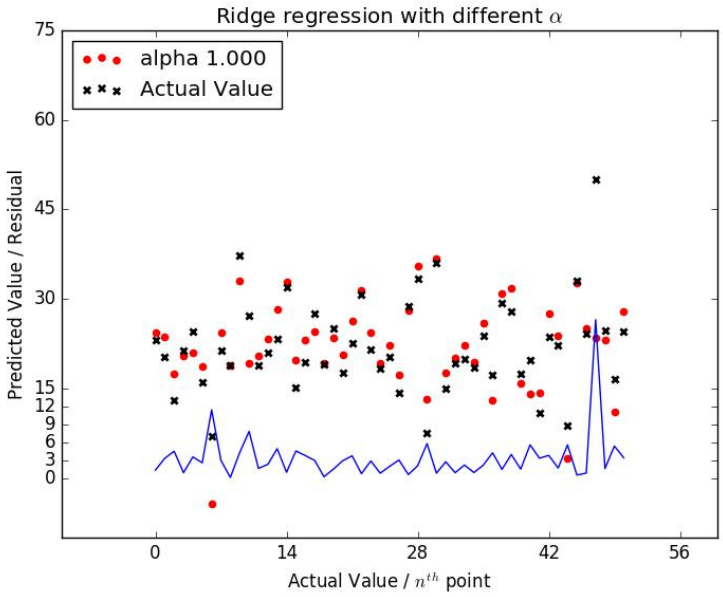


Fig.29 Ridge Regression Model

2) Lasso Regression

In this scenario, we model the data with Lasso regularization, using an appropriate normalization for the range of α.

We obtain the following parameters through 10-fold cross validation, and plot the figures as required only in the test case, (10% of the dataset for simplicity).

Optimal α: 0.001

Best RMSE: 5.079

Best r2_score: 0.787

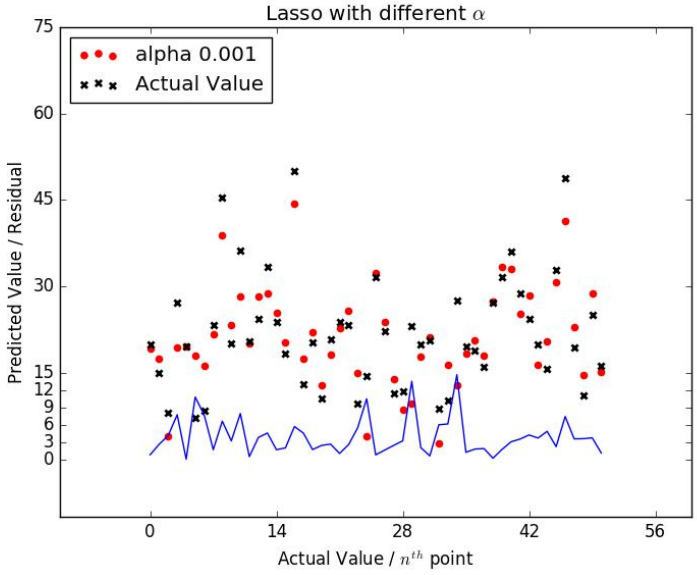


Fig.30 Lasso Regression Model