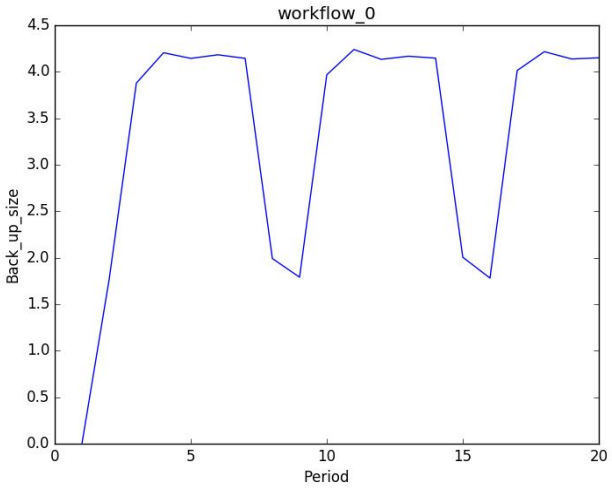
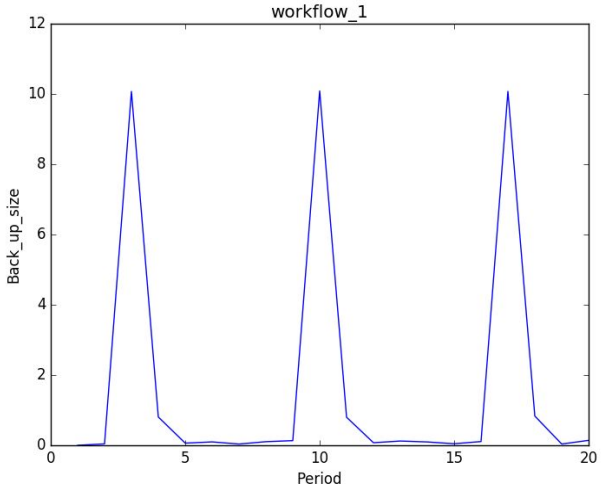
EE219 Project 1
Regression Analysis
Winter 2017

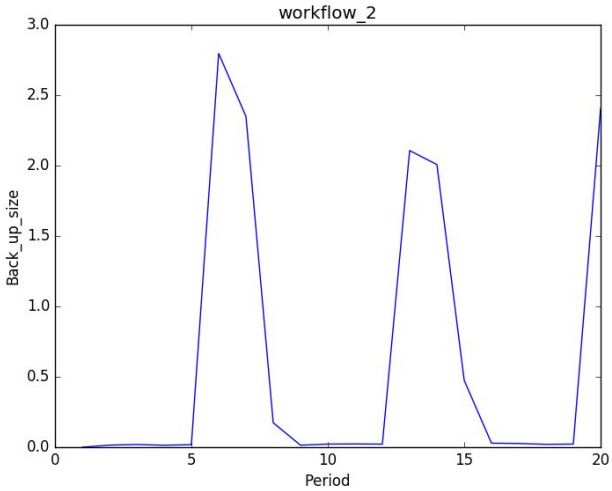
Boyang Cai 304330123
Manni Chen 304145309
Zhuoqi Li 004855607

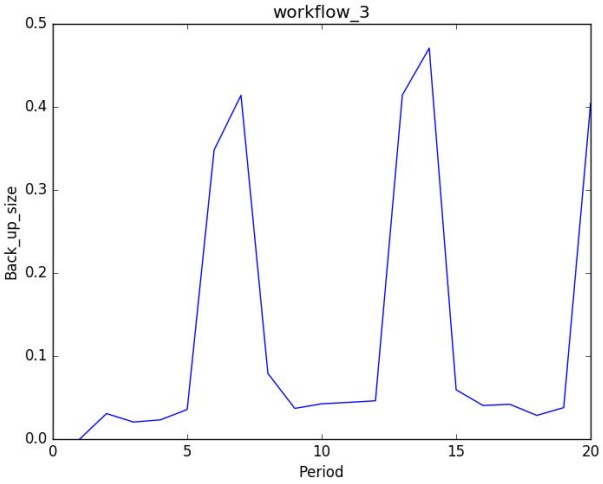
Network Backup Dataset

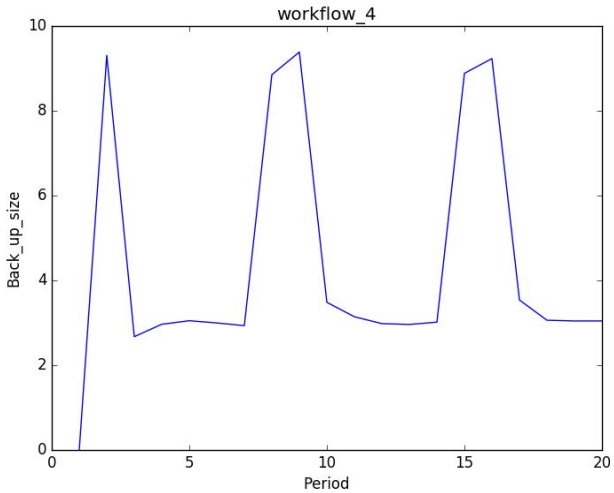
1)











By plotting the backup size for each workflow via time period in the first 20 days. We have the following observations:

Workflow 0: the copy size maintains in a high value for weekdays and drops to a lower value on weekends. Therefore, files in workflow 0 are mainly used during office day.

Workflow 1: the copy size has a peak value every week. For other days in this week, the copy size drops to 0. This indicates that files in workflow 1 are only used in a certain day of the week.

Workflow 2: the plot for workflow 2 is similar to the plot for workflow 1. However, for each week, the copy size maintains a high value for 2 days(Thursday and Friday) in workflow 2. For other days, the copy size drops to 0. So files in workflow 2 are only used in the last two days of the weekday.

Workflow 3: the plot for workflow 3 is similar to the plot for workflow 2. It also has peak values for Thursday and Friday. However, for other days, the copy size maintains in a lower value but not 0. In conclusion, files in workflow 3 are mainly used in two days for a week. For other days, they are slightly used.

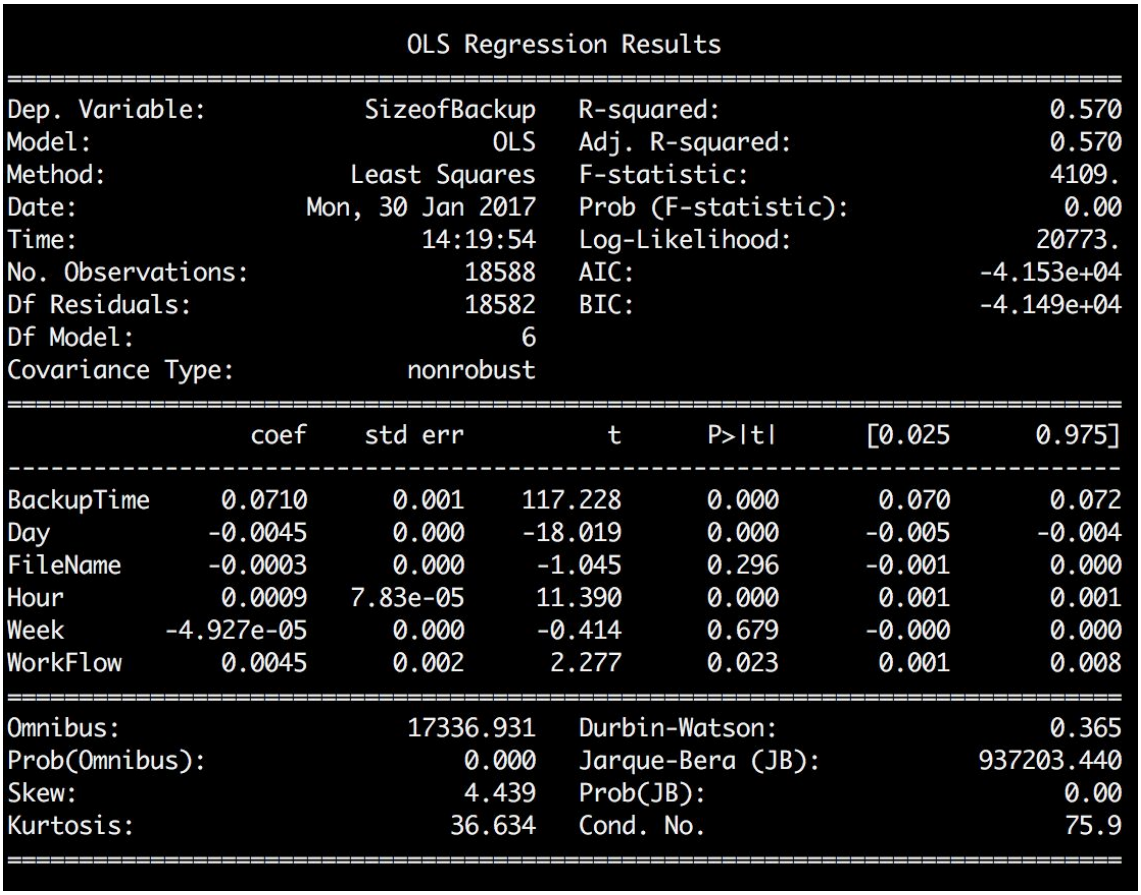
Workflow 4: the plot has peak values on the first two weekdays. For other days, it has lower values than it for these two days but higher than rest value for other plots. Files for workflow 4 are used everyday and they are more used in the first two weekdays.

Question 2

a) Linear regression model

In this section, we run the linear regression model on the dataset to see whether a linear regression model could give a good prediction on the backup size of the data based on other features in the data set. Also, we would also like to know which features have more weights in determining our targets.

We performed 10-fold cross validation linear regression on the dataset. To achieve this, we take advantage of the *Scikit_sklearn* library and *OLS library for pandas* to do the analysis. The following diagram displays the linear regression summary:



OLS Regression Results

Dep. Variable:	SizeofBackup	R-squared:	0.570
Model:	OLS	Adj. R-squared:	0.570
Method:	Least Squares	F-statistic:	4109.
Date:	Mon, 30 Jan 2017	Prob (F-statistic):	0.00
Time:	14:19:54	Log-Likelihood:	20773.
No. Observations:	18588	AIC:	-4.153e+04
Df Residuals:	18582	BIC:	-4.149e+04
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
BackupTime	0.0710	0.001	117.228	0.000	0.070	0.072
Day	-0.0045	0.000	-18.019	0.000	-0.005	-0.004
FileName	-0.0003	0.000	-1.045	0.296	-0.001	0.000
Hour	0.0009	7.83e-05	11.390	0.000	0.001	0.001
Week	-4.927e-05	0.000	-0.414	0.679	-0.000	0.000
WorkFlow	0.0045	0.002	2.277	0.023	0.001	0.008

Omnibus:	17336.931	Durbin-Watson:	0.365
Prob(Omnibus):	0.000	Jarque-Bera (JB):	937203.440
Skew:	4.439	Prob(JB):	0.00
Kurtosis:	36.634	Cond. No.	75.9

From the screenshot above we can observe that *Backup Time, Day of week and Hour of day* has significant influence on the prediction of *Backup Size* as their P values is **0.0**. The *Workflow number* also has some significance over the prediction as its p value is only **0.023**, which is smaller than the common alpha value 0.05. As for the *Filename and Week*, it does not really impact on the prediction due to their large P value.

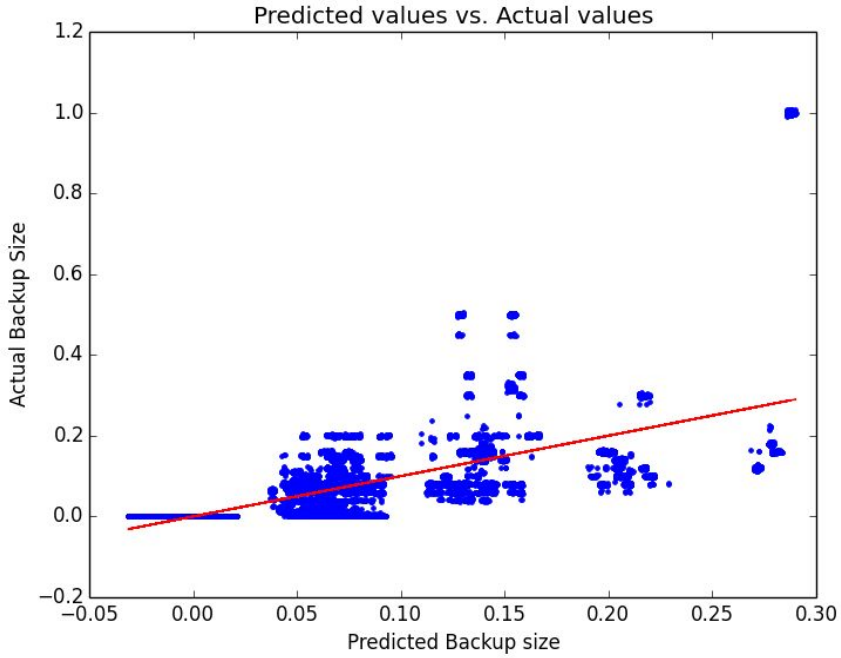
Note that the coefficients generated with OLS library of pandas actually differs slightly with those obtained with sklearn libraries. This maybe due to the different algorithm adopted by two libraries.

The R-squared value is only **0.57,** indicating that linear regression may not be the best fit.

The 10-fold cross validation RMSE value obtained for this regression is **0.079.**

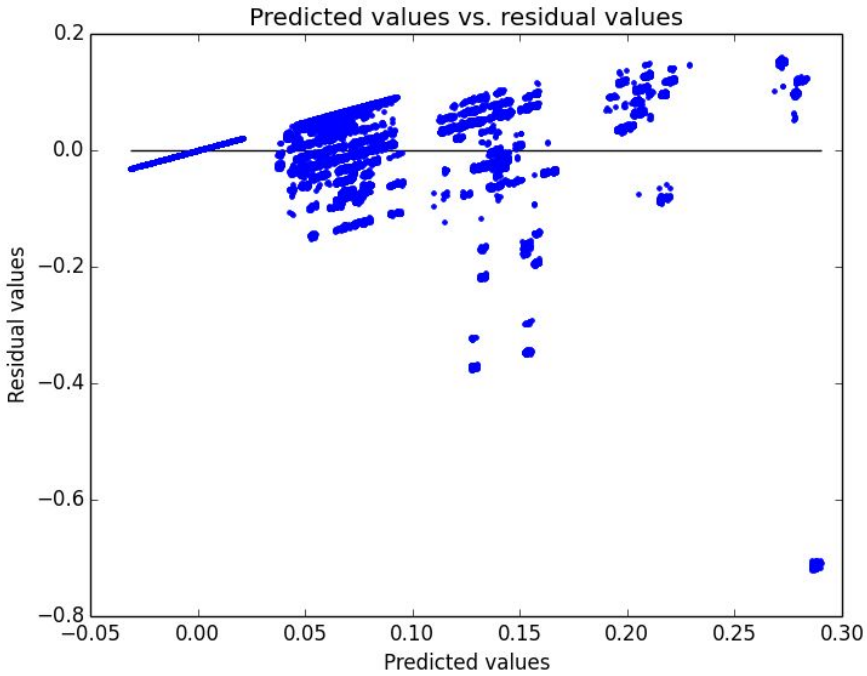
Predicted vs. Actual Values

To better visualize how well our model fits the data and whether linear regression is a good model in our case, we plotted the following Predicted vs. Actual Values diagram:



The red line in the diagram represent the result of our linear regression model. As revealed from the diagram, although many data points does follow the pattern of our model, especially for those with smaller backup size, there still exist some outliers which deviates from the linear model completely. This indicates that linear regression may not be the best model for our dataset.

Predicted vs. residual values

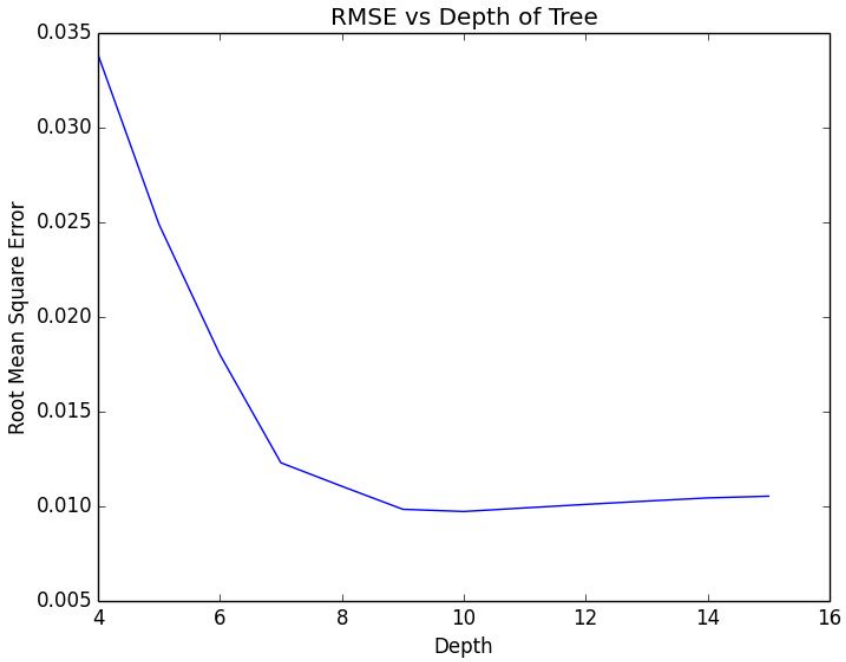


A better way to visualize how data points deviate from our model is to use the “Predicted vs. residual values” plot shown above. For a good prediction model, most data points should gather around the y = 0 line in the diagram above. Clearly, this is not our case. As we can see, there are still many data points deviate from y = 0 line and this trend gets more obvious when the predicted values get larger. Also, there seems to be recognizable upward pattern in the diagram.

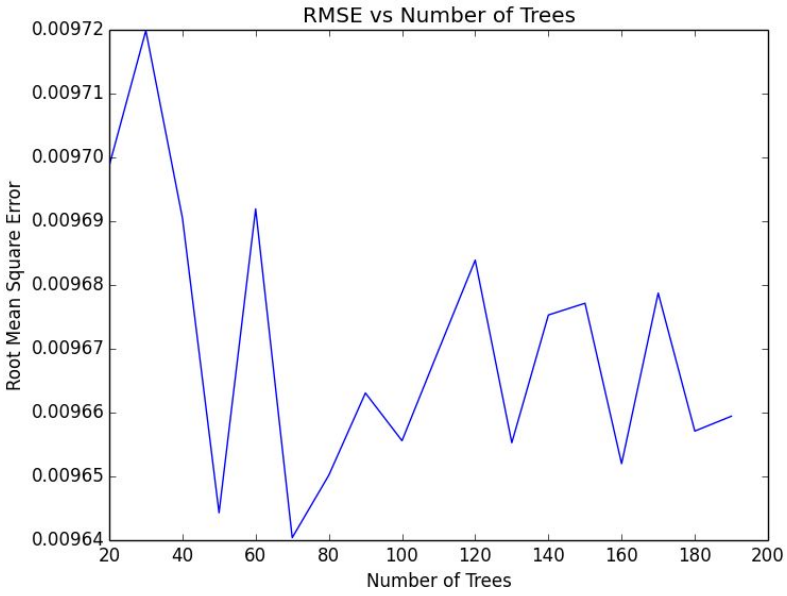
Based on the analysis to the two diagram above, we can tell that linear regression is not a good model to fit over the network backup size.

b) Random Forest Regression model

After we get the data using linear regression, we then change our algorithm to Random Forest Regression and use this regression to estimate our data and compare to the linear regression model.

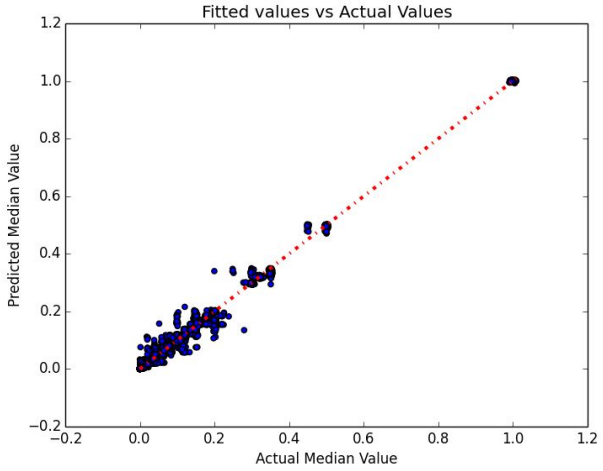


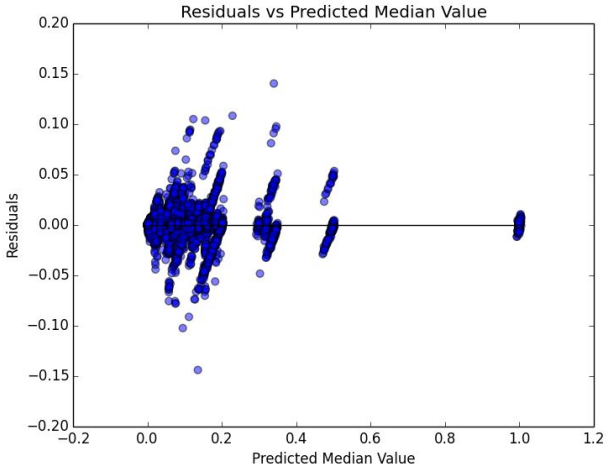
In Random Forest Regression, there are two main parameter: number of the tree and depth of the tree. First, we use 20 as our number of trees and simulate alone the depth of the tree. As we can see from the figure, the minimum RMSE occurs at the Depth 10. Then, we use the optimized Depth to generate our RMSE vs tree number graph to see the optimum RMSE for number of trees.



In this graph we discovered that the graph is not as smooth as the RMSE vs Depth. However, we also notices that the increasing number of trees can reduce the RMSR slightly, to a minimum of **0.00964**, which is small enough for a estimator. In this case, We see that the number of Trees here is 70.

For further verification, we also plot the predicted value vs actual value and residual vs predicted value for comparison with linear regression.





As we can see compared to the linear model, the Random Forest Regression has more accuracy than the linear regression since every point pair is really close to the predicted line and there are less outliers than a linear model. As we see the RMSE, the linear regression got a RMSE of **0.079** where random forest got a RMSE of **0.00964**, So we can see that the Random Forest Regression has a significant improvement on RMSE. By comparing the RMSE vs Depth and the RMSE vs Number of trees. We can also see that the number of depth will influence RMSE more than the number of trees.

Result:
Optimized Max Depth: 10
Optimized of Maximum Tree: 70
Root Mean Squared Error: 0.00964039654747

c) Neural Network Regression

We then use Neural Network Regression to estimate our data. In this case, we use MLP Regression from sklearn to achieve our goals.

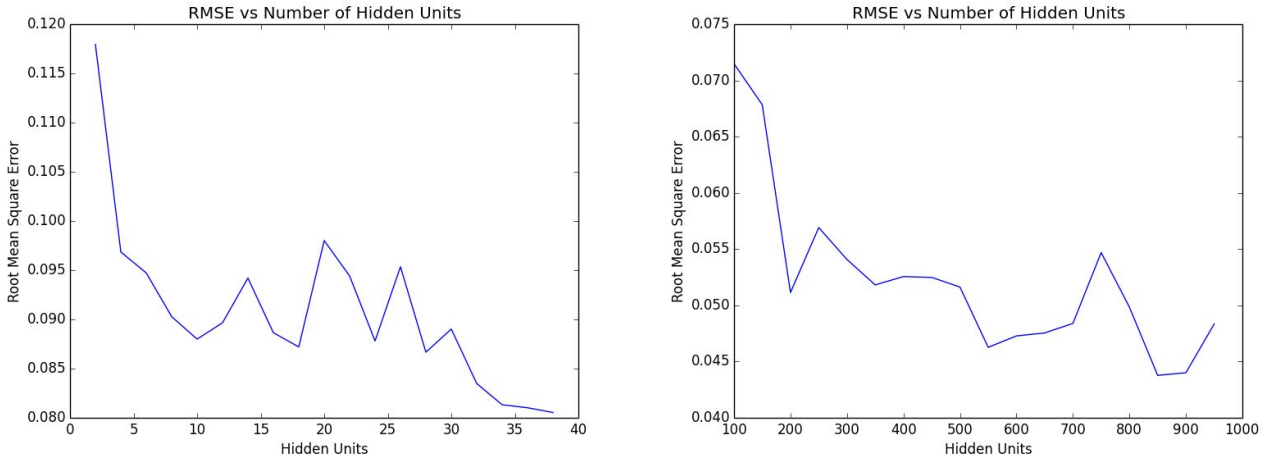
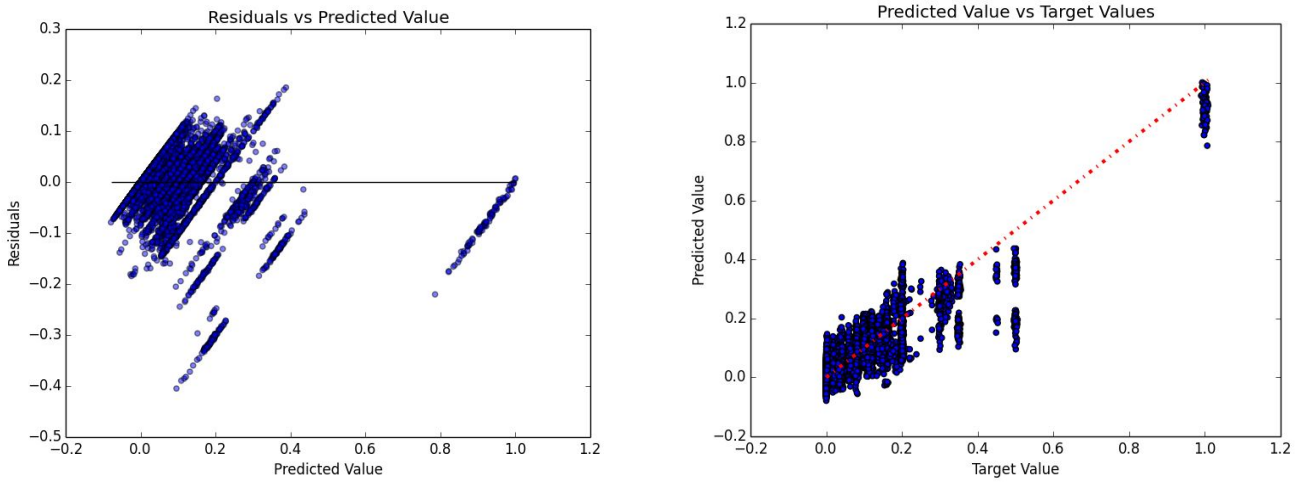


figure 2c-1 & 2c-2

In Neural Network Regression, we know that RMSE is determined by the number of hidden units and number of the hidden layers. However, we can not control the number of hidden layers too much because it can not exceed the number of the features. **In the graph, we see that overall the RMSE is decreasing as Hidden Units increase.** We then sketch the Predicted vs Actual and the Residual plot to see how the data shows



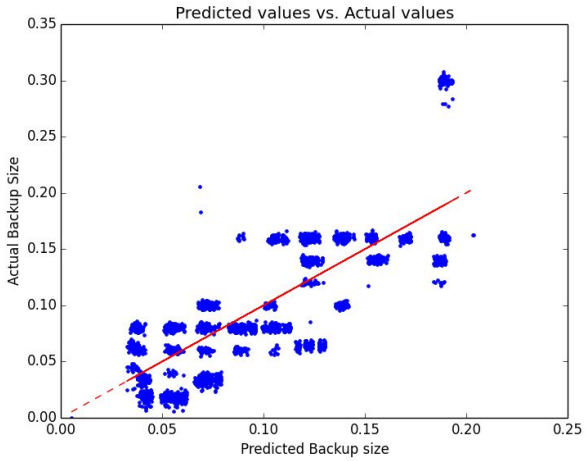
We found that the RMSE is less than the linear regression, however the minimum RMSE we got equals to 0.0438, which is greater than the **0.00964** we got from the Random Forest Regression. Due to the computer

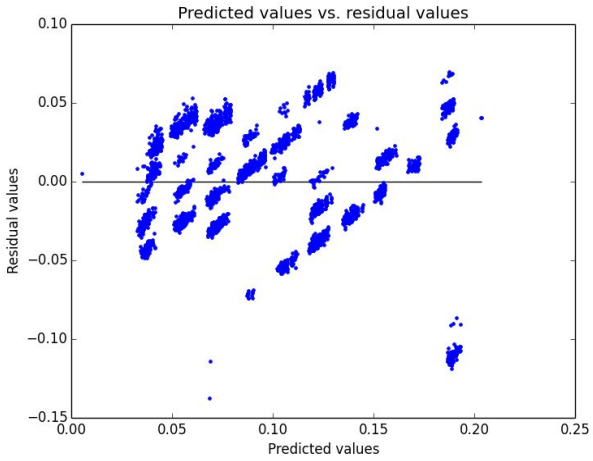
limit we are unable to verify higher number of unknown units. We think that the Random Forest Regression is the best match for this case and the Neural Network is not suitable for this case.

Figure 2c-1 result:
Optimized Hidden Unit: 38
Root Mean Squared Error: 0.0805394925887
Figure 2c-2 result:
Optimized Hidden Unit: 850
Root Mean Squared Error: 0.043774437539

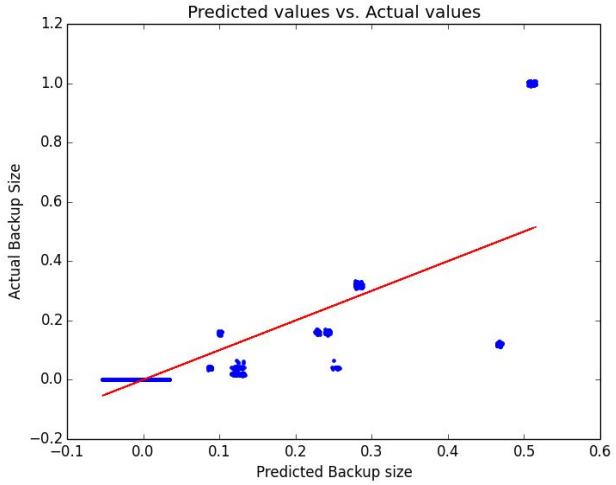
Question 3:
Linear Regression on WorkFlows

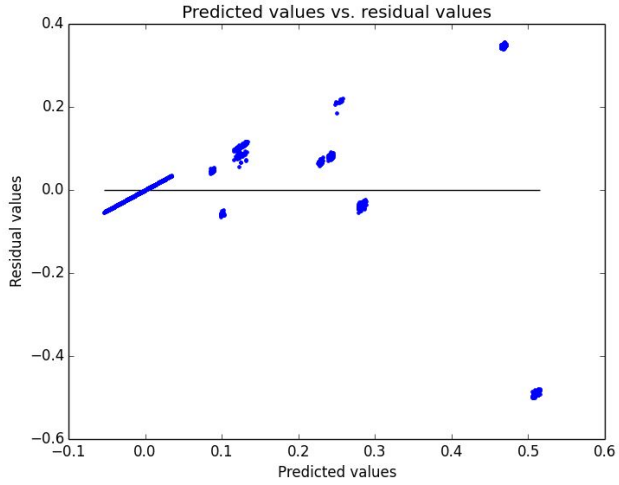
The following diagrams reveal the piece-wise linear regression on 5 workflows. Note that since diagram axes are not normalized, some workflow may have exaggerated diagrams but very small RMSE.



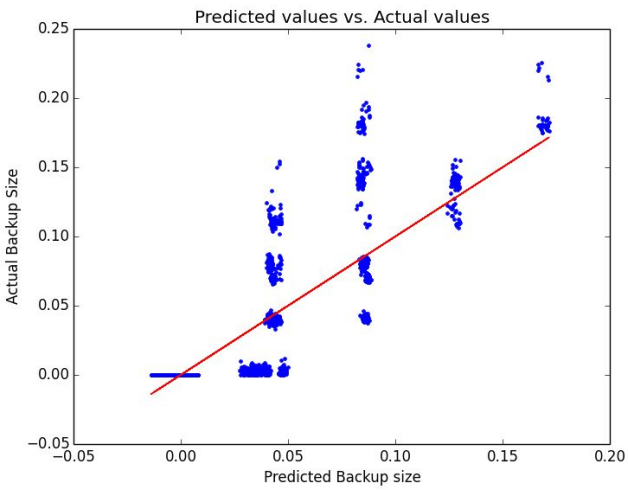


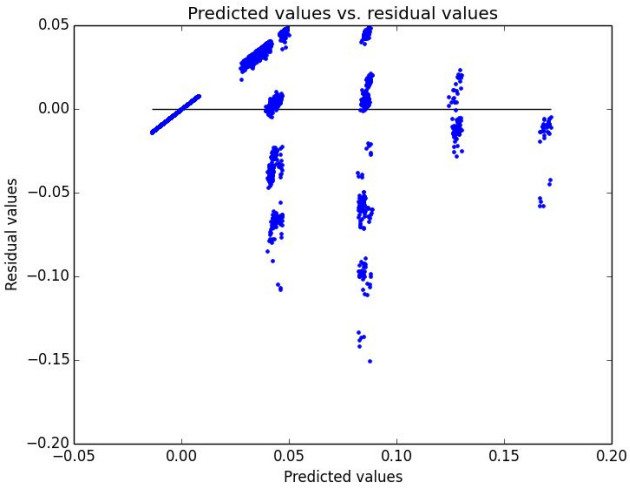
WorkF_#	Coef 1	Coef 2	Coef 3	Coef 4	Coef 5	Coef 6	RMSE
0	-3.297e-05	-1.002e-03	4.187e-03	0	-3.834e-05	3.326e-02	0.03539



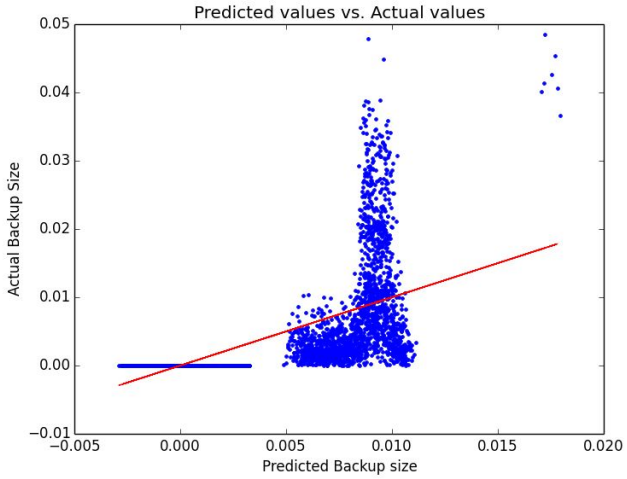


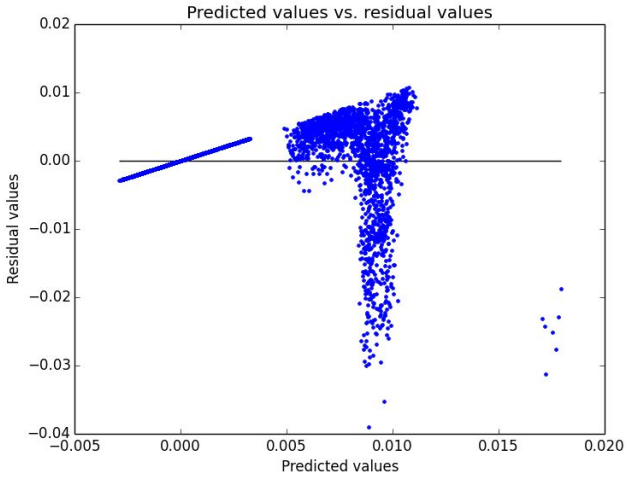
WorkF_#	Coef 1	Coef 2	Coef 3	Coef 4	Coef 5	Coef 6	RMSE
1	-2.802e-05	-1.92e-03	3.436e-03	0	5.56e-04	1.304e-01	0.1037



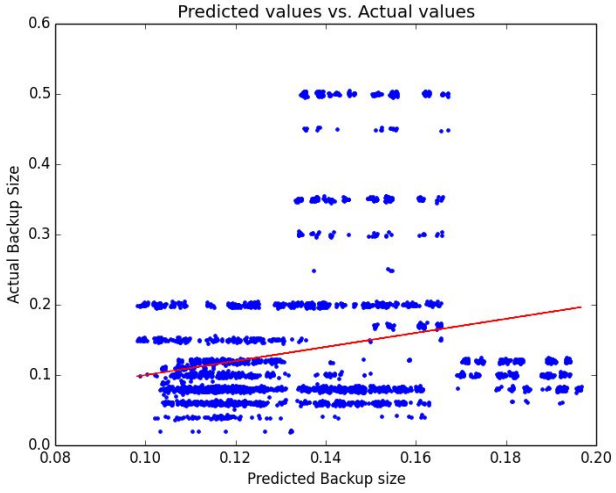


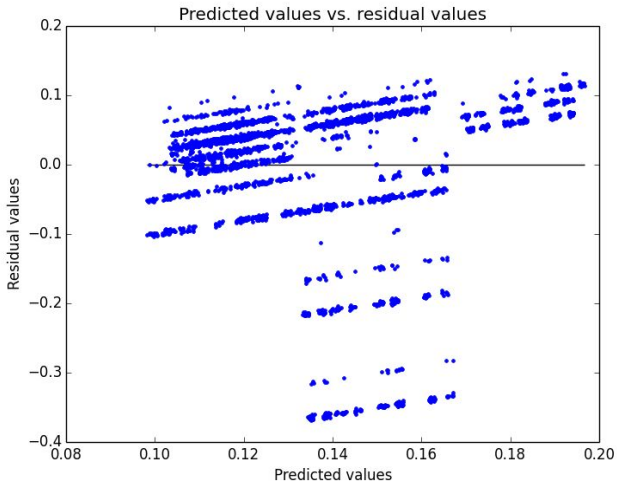
WorkF_#	Coef 1	Coef 2	Coef 3	Coef 4	Coef 5	Coef 6	RMSE
2	1.108e-04	2.69e-03	1.729e-04	0	2.257e-04	4.164e-02	0.025



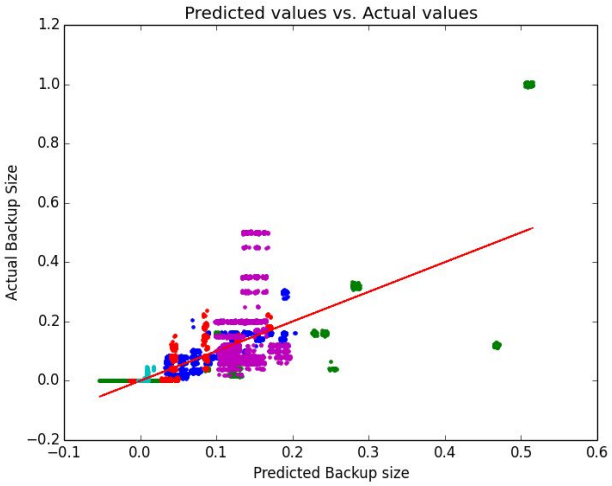


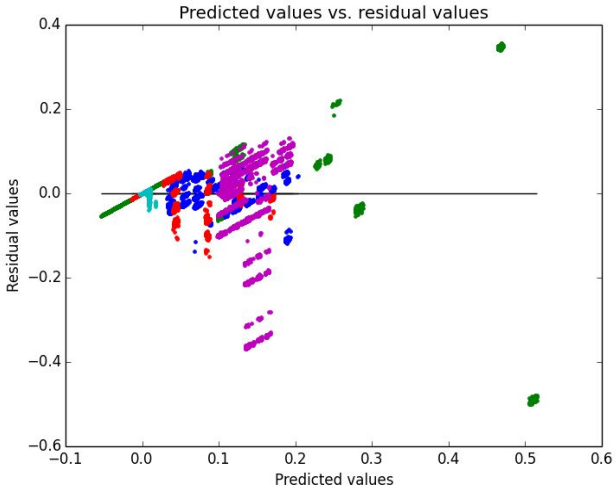
WorkF_#	Coef 1	Coef 2	Coef 3	Coef 4	Coef 5	Coef 6	RMSE
3	1.456e-05	7.353e-04	5.56e-05	0	-3.288e-05	7.875e-03	0.00576





WorkF_#	Coef 1	Coef 2	Coef 3	Coef 4	Coef 5	Coef 6	RMSE
4	-7.767e-05	-2.957e-03	-2.428e-04	0	-1.3498e-05	3.381e-02	0.102





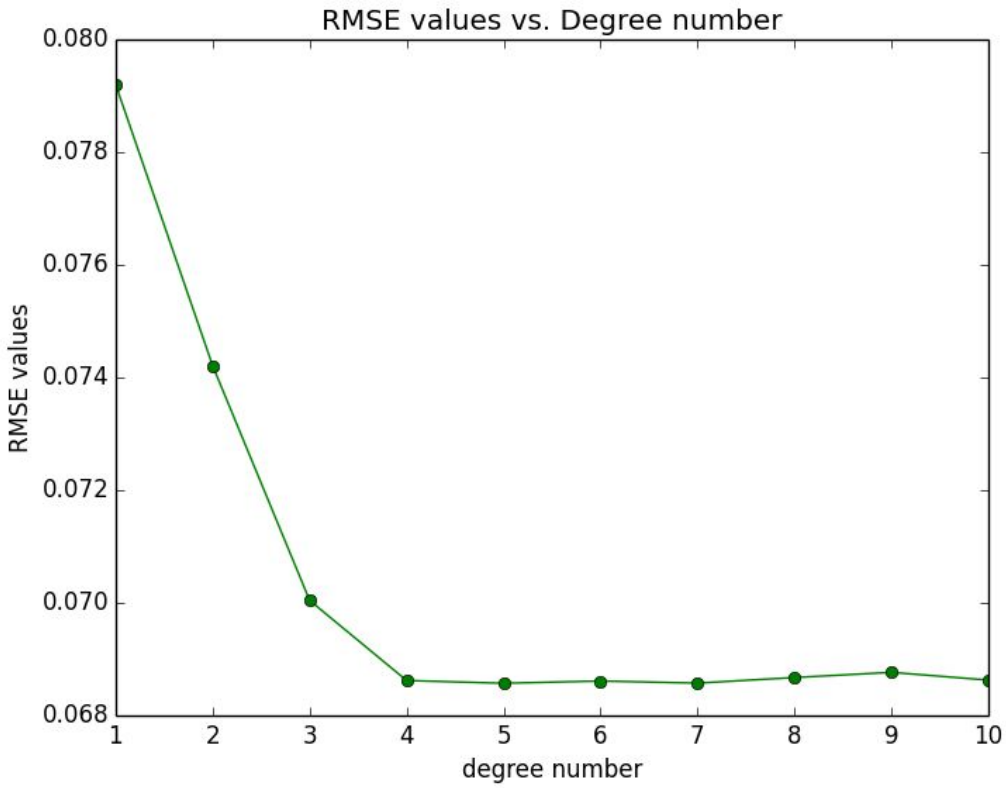
The two diagrams above show the piece-wise linear regression fitting over the network backup size data sets. Clearly the piece-wise regression fits the dataset much better than single linear regression as more data points now cluster around the predication line and y = 0 line in the residual graph. However, some outliers still exist and cannot be predicted by the piece-wise linear model.

Another way we can tell the piece-wise solution is better is that the average RMSE over 5 workflow is **0.054**, which is smaller than **0.079** obtained with single linear regression model.

Polynomial Regression

From above analysis we have learned that linear regression is not enough to give an accurate prediction, in order to obtain a better prediction on outliers, a more complex regression model, the polynomial regression method will be used.

However, since we have no idea which degree of the polynomial will yield the best result, we performed 10-fold cross validation from degree 1 to 10. By looking at the evolution of RMSE values with growing degrees, we could find a threshold degree value that may give us the best fitting result.



Degree #	1	2	3	4	**5**	6	7	8	9	10
RMSE	0.0792	0.0742	0.0699	0.068604	**0.0686**	0.06869	0.06864	0.068763	0.068761	0.068764

According to the table above, we can see that the RMSE value stops to decrease and starts to rise with when degree is larger than 5. Therefore, we determined the degree threshold for our polynomial regression to be **5**.

Benefits of cross validation

Since test data set is not used in model estimation, thus the MSE calculated on this data will give a more predictive accuracy. This because multi-fold calculation yields more conservative result. Therefore, this method provides an unbiased biased measure of MSE on new observations.

Therefore, cross validation helps fit the data better because it separates test and train datasets and use that test data set to mitigate the bias in the fitting model. This prevents from overfitting and help reduce RMSEs.

Question 4: Boston Housing Dataset

We perform similar linear regression as we did in problem 2 in problem 4 and the following diagram shows the regression summary:

OLS Regression Results

Dep. Variable:	MEDV	R-squared:	0.959
Model:	OLS	Adj. R-squared:	0.958
Method:	Least Squares	F-statistic:	891.3
Date:	Mon, 30 Jan 2017	Prob (F-statistic):	0.00
Time:	21:29:17	Log-Likelihood:	-1523.8
No. Observations:	506	AIC:	3074.
Df Residuals:	493	BIC:	3128.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
CRIM	-0.0929	0.034	-2.699	0.007	-0.161	-0.025
ZN	0.0487	0.014	3.382	0.001	0.020	0.077
INDUS	-0.0041	0.064	-0.063	0.950	-0.131	0.123
CHAS	2.8540	0.904	3.157	0.002	1.078	4.630
NOX	-2.8684	3.359	-0.854	0.394	-9.468	3.731
RM	5.9281	0.309	19.178	0.000	5.321	6.535
AGS	-0.0073	0.014	-0.526	0.599	-0.034	0.020
DIS	-0.9685	0.196	-4.951	0.000	-1.353	-0.584
RAD	0.1712	0.067	2.564	0.011	0.040	0.302
TAX	-0.0094	0.004	-2.395	0.017	-0.017	-0.002
PTRATIO	-0.3922	0.110	-3.570	0.000	-0.608	-0.176
B	0.0149	0.003	5.528	0.000	0.010	0.020
LSTAT	-0.4163	0.051	-8.197	0.000	-0.516	-0.317

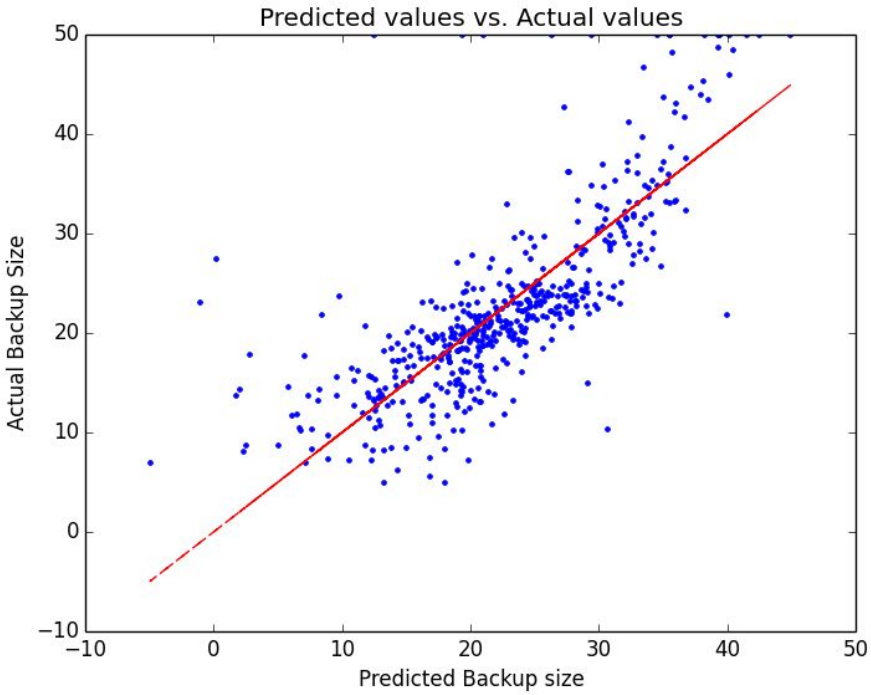
Omnibus:	204.082	Durbin-Watson:	0.999
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1374.225
Skew:	1.609	Prob(JB):	3.90e-299
Kurtosis:	10.404	Cond. No.	8.50e+03

Similarly, from the p values, we can tell ***CRIM, ZN, CHAS, RM, DIS, PTRATIO, B and LSTA*** has siginificant impact on MEDV while ***INDUS, NOX, AGS*** seems to not really impact our prediction.

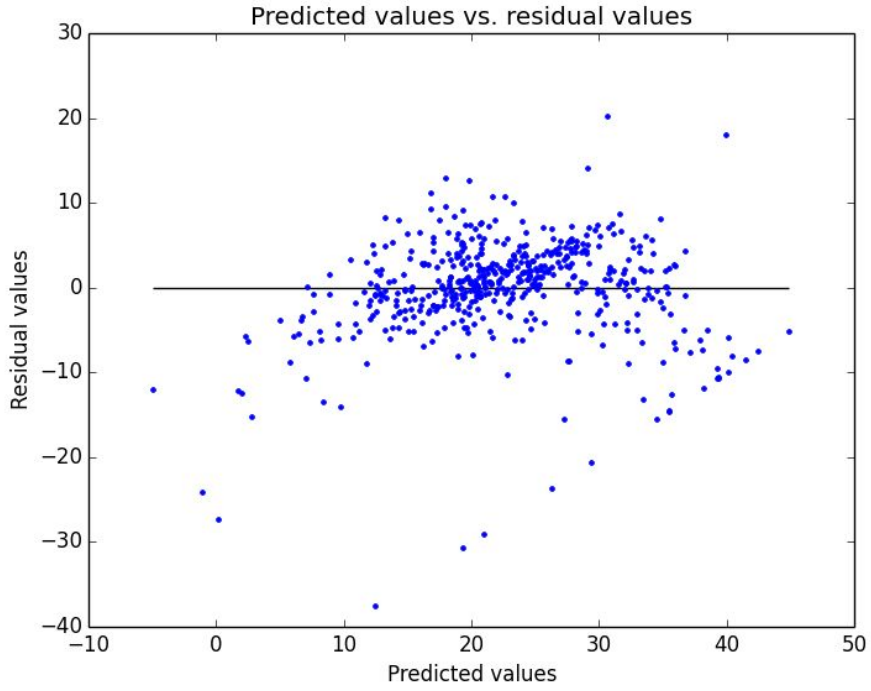
Also, the R squared value of 0.959 shows that the variance in MEDV is dependent by 95% on the remaining features, which means the linear regression model may be a good fit.

Also, the RMSE of the linear model is **5.877.**

Just as we did back in problem 2a, to evaluate whether linear model could be a good fit, we draw the “Predicted vs. Actual value” and “Predicted vs. residual value” diagrams:

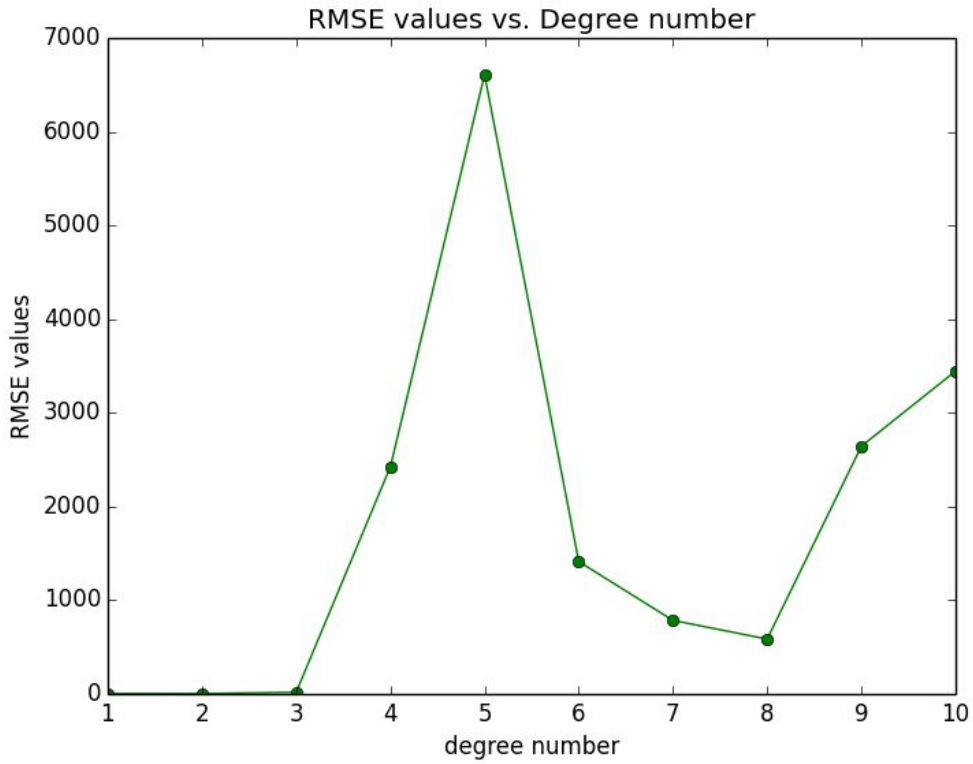


Compared with what we had in problem 2, this diagram reveals that linear regression does a much better job in fitting over the dataset as most data points do cluster around the prediction line.



The residual diagram further demonstrates above claim. However, as we can see, there still exist many outliers that do not get predicted well at all.

To get a better fitting diagram, we also performed the polynomial regression with degree 1 to 10:

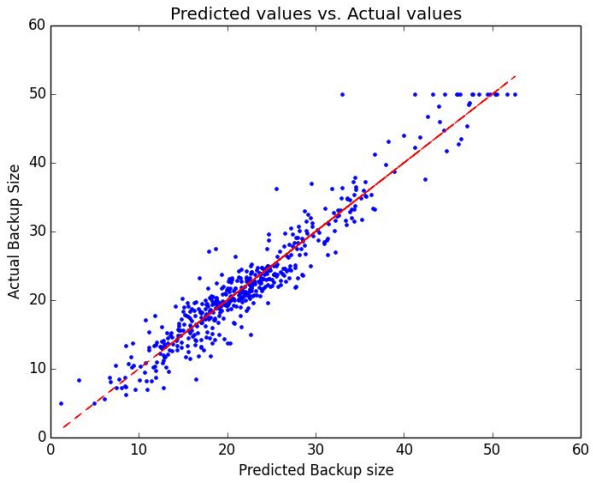


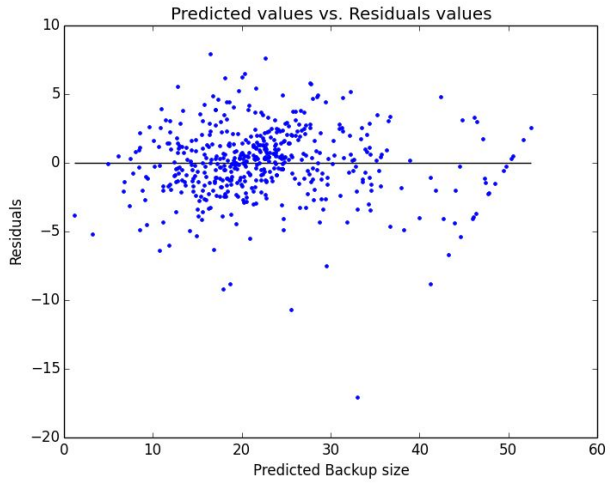
Degree #	1	**2**	3	4	5	6	7	8	9	10
RMSE	4.8673	**3.595**	18.408	2424.584	6607.58	1412.67	785.72	585.91	2643.7	3445.99

The diagram above shows the evolution of 10-fold cross validation RMSE. As we can see, the RMSE value gets very large when degree is larger than 3, and the min RMSE appears at degree equal to **2**. Therefore, the threshold degree for the boston housing polynomial regression is **2.**

One interesting observation we found while doing this problem is that the built-in function "*cross_val_score(lr, X, Y, cv=10, scoring='neg_mean_squared_error')* fails to yield correct 10-fold cross validation results in polynomial regression cases. In order to get the correct values, we changed 'cv = 10' To 'cv = KFold(n_splits = 10, shuffle = True)'.

We also plotted the following two diagrams at degree 2 to compare with the linear model:





According to the two graphs above, we can see that the second order polynomial regression is indeed better than the linear regression model.

Question 5: Ridge & Lasso Rregression

a)

When alpha in range[1, 0.1, 0.01, 0.001], take the ridge regression using 10-fold cross validation.
The best alpha for ridge regression is **1**.
The best RMSE value under this alpha is **4.6952**

b)

When alpha in range[1, 0.1, 0.01, 0.001], take the lasso regression using 10-fold cross validation.
The best alpha for ridge regression is **0.01**.
The best RMSE value under this alpha is **4.8659**

Coefficients comparision

Un-regularized Coeffs	**Ridge Regression Coeffs**	**Lasso Regression Coeffs**
-0.0929	-0.1046	-0.0364
0.0487	0.0474	0.0132
-0.041	**-0.0089**	**0**
2.854	2.5524	2.3535
-2.8684	-10.777	-8.5552
5.9281	3.854	4.2359
-0.073	**-0.0054**	**0**

-0.9685	-1.3727	-0.7434
0.1712	0.2901	**0**
-0.0094	-0.0129	**0**
-.3922	-0.8761	-0.8185
0.0149	**0.0097**	**0.0072**
-0.4163	-.5333	-0.5207

According to the table, there are 4 coefficients in lasso regression equals to 0. This indicates that lasso regression regression reduces the features from 13 to 9, the left 4 features are not very important. In ridge regression, there are also 3 coefficients which absolute values are less than 0.1. These coefficients can also be considered very small and corresponding features are less important. However, for un-regularized coefficients, all values are relatively great comparing with coefficients in lasso and ridge regression. Therefore, we can conclude ridge and lasso regression constrain the coefficient vector to lie in a less complex manifold.