Large Scale Data Mining
Homework 1

Anurag Pande - 604749647
Brendon Faleiro - 704759004
Sachin Krishna Bhat - 304759727

1 Introduction

This project analyzes two data sets - **Network Backup Data** and **Boston Housing Data**. We use the method of regression while anaysing the datasets. A method for fitting a curve (not necessarily a straight line) through a set of points using some goodness-of-fit criterion[1]. During this process we have tudied the use of several regression tools, including the Linear Regression, Random Forest Regression, Polynomial Regression and Neural Network Regression models. We have also studied the concepts of cross validation and regularization to improve the prediction of dependent variables in the datasets.

The **Network-Backup Dataset** is comprised of simulated traffic data on a backup system in a network. It contains information about both the size of the data and the time taken for moving the data. In this project, we have tried to predict the **backup size** of the traffic depending on the file name, day/time of backup. This prediction was done with the use of Linear, Random Forest, Neural Network and Polynomial Regression models.

The **Boston Housing Dataset** contains housing values of the suburbs. In this project, we have tried to estimate the **value of owner-occupied homes**. We have used Linear and Polynomial Regressions to create a predictive model.

2 Network Backup

The system monitors the files residing in a destination machine and copies their changes in four hours cycles. At the end of each backup process, the size of the data moved to the destination as well as the duration it took are logged, to be used for developing prediction models. The Network-Backup Dataset has information of files maintained in destination machine and it monitors and copies their changes in four hours cycle. The features captured in data set are as follows:

1. **Week index**
2. **Day of the week:** at which the file is backed up starts
3. **Backup start time - Hour of the day:** the exact time that the backup process is completed
4. **Workflow ID**
5. **File name**
6. **Backup size:** the size of the file that is backed up in that cycle in GB
7. **Backup time:** the duration of the backup procedure in hour

2.1 Question 1: Relationships in the Dataset

We try to develop prediction models for predicting the size of the data being backed up as well as the time a backup process may take. To get an idea on the type of relationships in the dataset, for each workflow, we plot the actual copy sizes of all the files on a time period of 20 days.
On analysis of each of the plots, we see the following trends in the workflows.

- **Workflow 0:** There is a clear drop in the copy sizes towards the weekends. The data copy sizes vary between 0.3 and 0.7 GBs. This would indicate that Workflow_0 (Files 0 to 5) typically are used to log content during the weekdays.

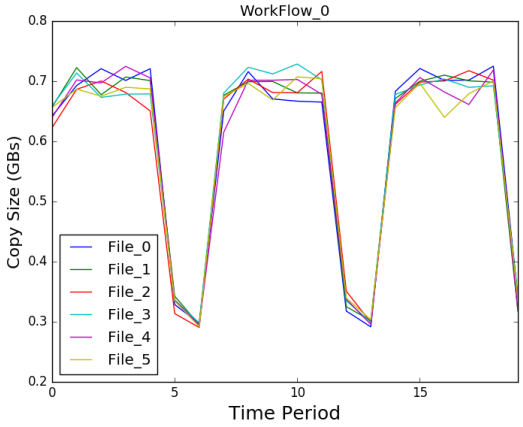

Figure 1: Copy Size vs Time Period for Workflow 0

- **Workflow 1:** The copy sizes in Workflow_1 peaks towards the start of every week (Mondays). Thereafter the copy sizes falls to 0GB for the rest of the week. This could mean that the files included in Workflow_1 (Files 6 to 11) are typically only used on Mondays, thus needing a backup and then cleared for the rest of the week.

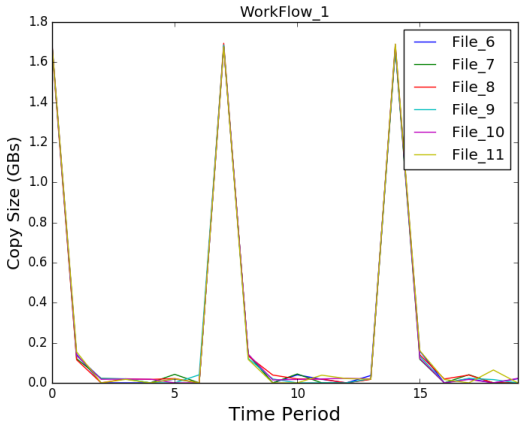

Figure 2: Copy Size vs Time Period for Workflow 1

- **Workflow 2:** Workflow_2 (Files 12 to 17) shows a rise in copy sizes between Wednesdays and Saturdays with the copy sizes peaking on Thursdays. The rest of the days the copy sizes are almost 0GB.

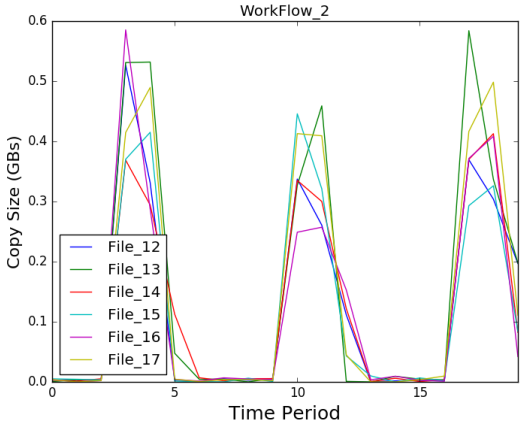

Figure 3: Copy Size vs Time Period for Workflow 2

- **Workflow 3:** Workflow_3 (Files 18 to 23) shows a trend similar to Workflow_2 with copy sizes peaking between Wednesday and Saturdays at around 0.07 GB. However, unlike Workflow_2, the copy sizes during the rest of the week is uneven and not 0GB.

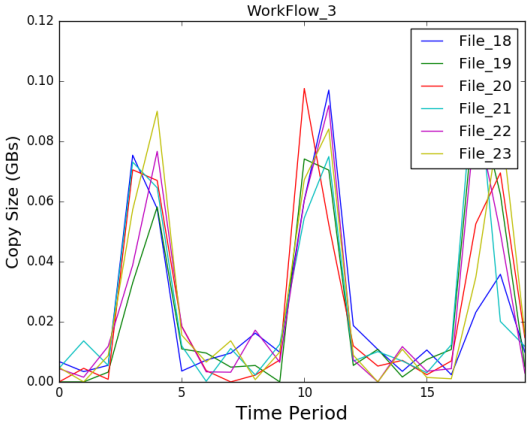

Figure 4: Copy Size vs Time Period for Workflow 3

- **Workflow 4:** The data backup trends in Workflow_4 are almost the inverse of Workflow_1. It would seem like the files in Workflow_4 (Files 24 to 29) are worked on during the weekends and only contain a small amount of data during the rest of the week. The copy sizes vary between 0.5 and 1.5GB.

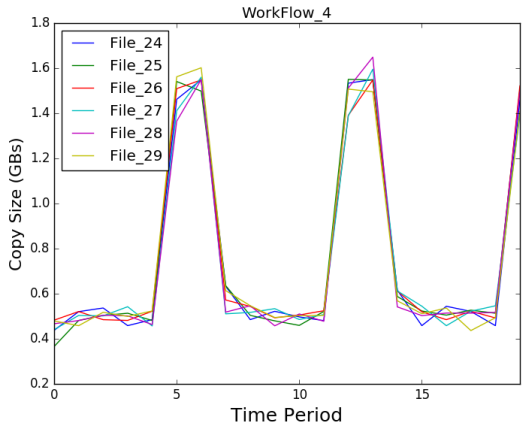

Figure 5: Copy Size vs Time Period for Workflow 4

Thus we have seen, almost all the dataset workflows vary largely based on the day of the week.

2.2 Question 2: Copy Size Prediction

In this section we attempt to predict the copy size of a file, given other attributes. To achieve this, three different regression techniques were used:

- Linear Regression
- Random Forest Regression
- Complex Regression (Polynomial Function)

2.2.1 Linear Regression

In order to predict the copy size, a Linear Regression model was built with the copy size as the target variable and the other attributes were used as features. The ordinary least squares function was used to calculate the penalty on the regression. This model was tested using 10 folds cross validation.The model was created using the *Linear_Models from Scikit-Learn Library* and the *OLS library for Pandas*.

As can be seen from the Pandas OLS summary presented in Figure 6, the **RMSE** value obtained after 10 fold Cross-Validation is **0.07956**. On the basis of the output we can conclude that the p values for all the variables is 0.00. Thus showing a strong dependency between the copy size and the variables.

OLS Regression Results

Dep. Variable:	backUpSize	R-squared:	0.563
Model:	OLS	Adj. R-squared:	0.563
Method:	Least Squares	F-statistic:	3985.
Date:	Mon, 30 Jan 2017	Prob (F-statistic):	0.00
Time:	12:53:07	Log-Likelihood:	20612.
No. Observations:	18588	AIC:	-4.121e+04
Df Residuals:	18582	BIC:	-4.116e+04
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
0	-0.0084	0.002	-4.942	0.000	-0.012	-0.005
1	-0.0017	0.002	-1.069	0.285	-0.005	0.001
2	0.0118	0.002	7.414	0.000	0.009	0.015
3	0.0434	0.008	5.441	0.000	0.028	0.059
4	-0.0472	0.009	-5.177	0.000	-0.065	-0.029
5	0.2758	0.002	115.216	0.000	0.271	0.281

Omnibus:	17809.065	Durbin-Watson:	0.364
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1019551.524
Skew:	4.621	Prob(JB):	0.00
Kurtosis:	38.085	Cond. No.	24.9

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

RMSE Values of Estimator : 0.079560357063

Figure 6: Results from Linear Regression on the Network Backup Dataset

Predicted vs Actual Values

The graph in Figure 7 shows the mapping of predicted values against the actual values of the Copy Size of the network backup data set when the Linear regression model is applied. In most cases, the predicted values have an extremely small deviation from the actual values.

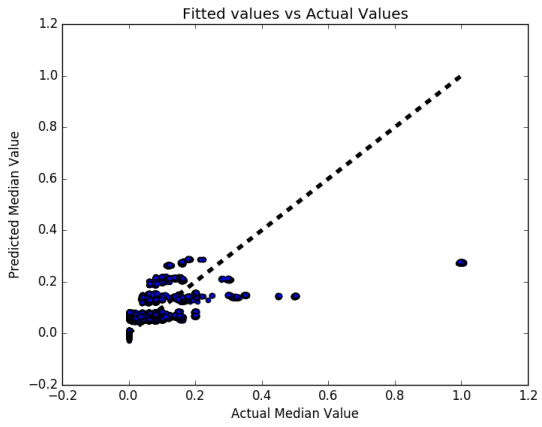

Figure 7: Linear Regression - Predicted Values vs Actual Values

Residuals vs Predicted Values

The **Residual Value** is the difference between the actual values and the predicted values. In Figure 8, we see the mapping between the predicted values and residual values of the copy sizes of the network backup data when the Linear Regression model is used. Since most

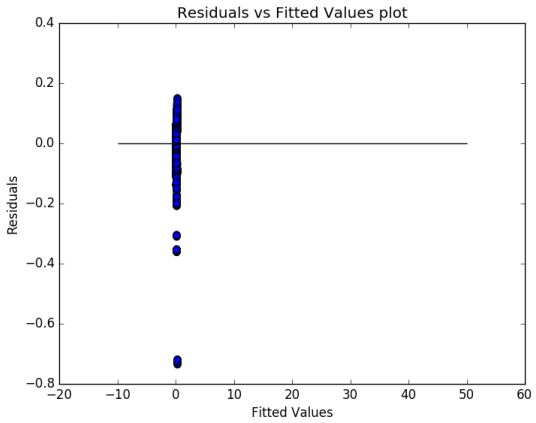

Figure 8: Linear Regression - Residual Values vs Predicted Values

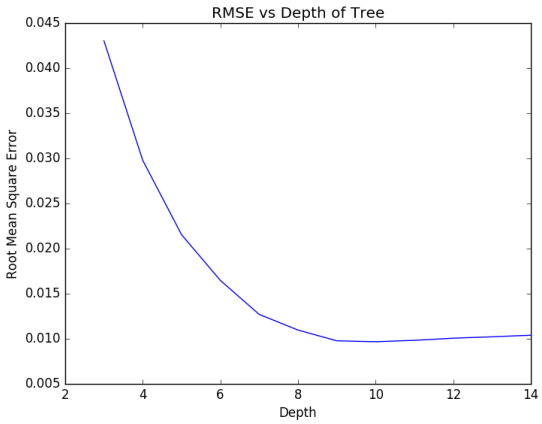

Figure 9: Random Forest Regression - RMSE vs Maximum Depth

of the residuals are concentrated close to the zero mark, and thereby indicate a good fit. While the model works well with majority of the data points, we still see a considerable error in mapping certain outliers.

2.2.2 Random Forest Regression

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks. Random Forests operate by constructing multiple decision trees at training time[2]. These trees then output the mode of the classes (classification) or mean prediction (regression) of the individual trees. Individual decision trees are known to overfit for the given data and thus, Random Forests help in avoiding such overfitting.

In our analysis, we used the *scikit-learn library*. The model was tuned by varying the number of trees in the model and the maximum depth of each tree. Initially, the tree depth was decided by creating models by varying the depths of trees between 4 and 15, and setting the number of trees to 20. As can be seen in Figure 9, minimum RMS error was found at a depth of 10.

Once the best depth was fixed at 10, we then tuned the model to find the optimum number of trees. Models were created for varying numbers of trees between 20 and 220. As shown in Figure 10, the RMSE is minimized when 40 trees are used. When we take **best depth of 10** and **number of trees as 40**, a 10-fold Cross-Validation on the model gives us an **RMSE of 0.009597**.

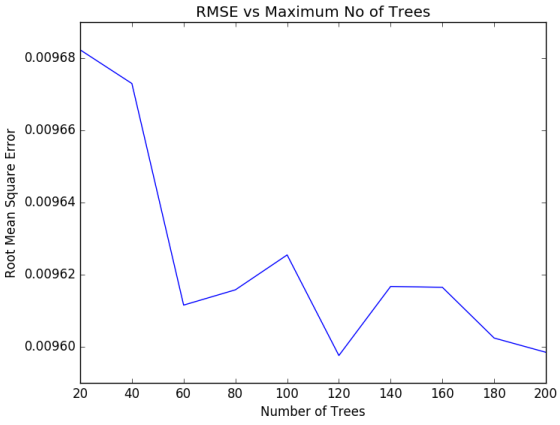

Figure 10: Random Forest Regression - RMSE vs Number of Trees

Predicted vs Actual Values

Figure 11 shows the mapping of the Predicted values against the Actual data values. The values in the figure lie closer to the line generated by the model.

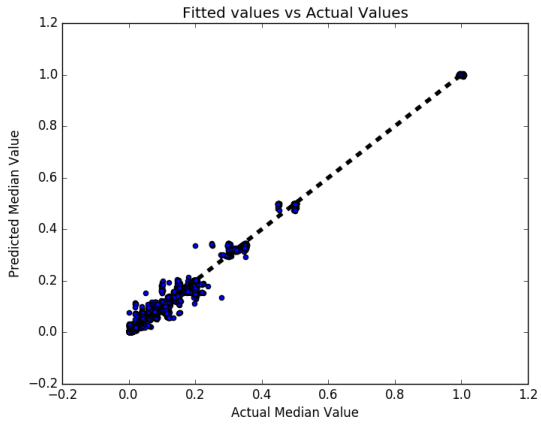

Figure 11: Random Forest Regression - Predicted Values vs Actual Values

Residuals vs Actual Values Figure 12 shows a mapping of the Residuals from the predictions against the Predicted Values. Since all the values are mapped close to the 0 line, we know that this model is good fit.

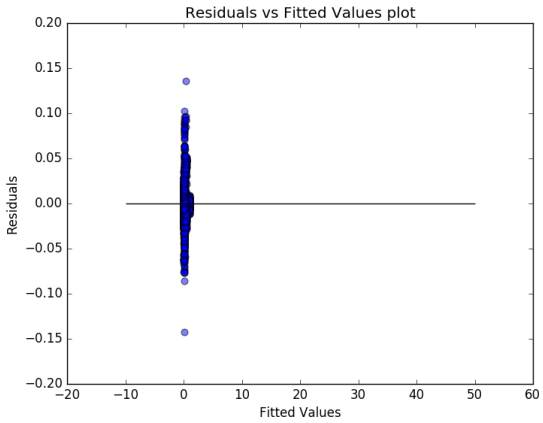

Figure 12: Random Forest Regression - Residual Values vs Predicted Values

Relationship between Linear and Random Forest Regression models

The RMSE value obtained from the Linear Regression model was **0.07956**. The Random Forest Regressor on the other hand has an RMSE of **0.009578**. The Random Forest regressor thus gives us a better prediction model that the Linear Regressor. However, unlike the Linear Regressor, the Random Forest has uncertainties in the RMSE values due the fact that the start node is selected at random.

As seen in Figures 7 and 11, the outliers are removed by the Random Forest Regressor. Also the points now lie closer to the regressor line. As a result the RMSE is much smaller with the Random Forest Regressor than the Linear Regressor.

2.2.3 Neural Network Regression

Neural network regressors are used to map a continuous input to a continuous output. We built a NeuralNetwork Regressor using the *PyBrain Library*. The model was tuned by varying parameters such as the number of epochs run by the model and the number of nodes in the hidden layer of the network.

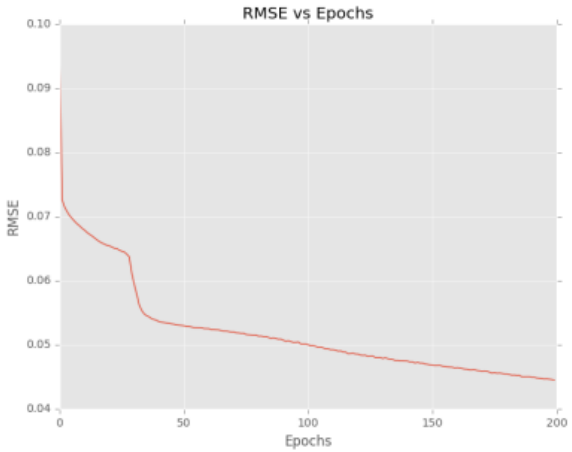

Figure 13: RMSE vs Epochs

The neural network is modelled to get best parameters by varying the number of epoch for 100 hidden nodes. We identified that the model worked best with 100 epochs and achieved an RMSE of **0.0483153**.

Major Parameters and their Effect on the RMSE

The main parameters we used in tuning our model were:

- Number of Epochs
  The ideal number of epochs needed to tune a model is dependent on the dataset being used. In order to achieve an optimum number of epochs, a threshold value is set on the error. Once the error value falls below this threshold, we consider the model to be sufficiently tuned.

- Number of Nodes in Hidden layer
  Like the number of epochs needed, the number of hidden nodes needed for an optimum performance is highly dependent on the dataset being used. In order to achieve optimum performance, models were created by varying the number of hidden nodes. The RMSE was found to be minimum for the given dataset when 100 hidden nodes were used.

2.3 Question 3:

In this section we try to predict the backup size for each workflow independently. The model was built using the Linear regression technique.

2.3.1 Linear Regression on WorkFlow 0:

```
Estimated intercept coefficient:  0.0439790236565
   Features  EstimatedCoefficients
0         0              -0.000971
1         1              -0.059171
2         2               0.088472
3         3               0.000645
4         4               0.105937

RMSE Values of Estimator : 0.0294817316878
```

Figure 14: Linear Regression Estimated Coefficients for Workflow 0

The RMSE values for all the workflows together as seen in section 1 is **0.07956**. However, when the RMSE was calculated for Workflow-0 alone, the value is minimized to **0.02948**. The Figures 15 and 16 shown below also prove that there is a mich smaller deviation between the predicted and actual values and the residual is kept below 0.1. Thus the individual fit is much better than the overall fit.

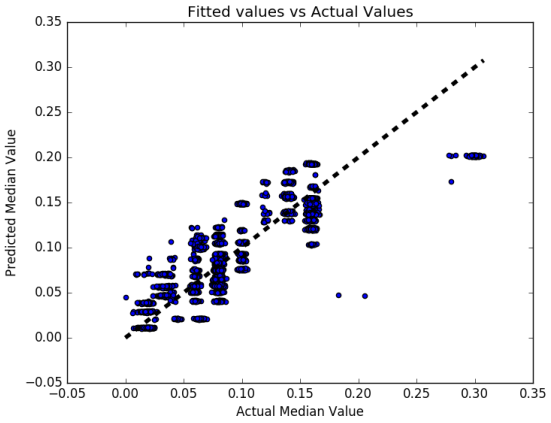

Figure 15: Linear Regression - Predicted Values vs Actual Values

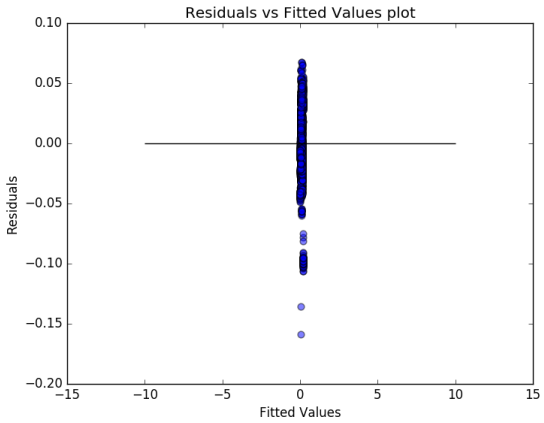

Figure 16: Linear Regression - Residuals vs Predicted Values

2.3.2 Linear Regression on WorkFlow_1:

```
Estimated intercept coefficient:  -0.0397178986831
   Features  EstimatedCoefficients
0         0                0.003924
1         1               -0.011384
2         2                0.068631
3         3                0.000845
4         4                0.495040

RMSE Values of Estimator : 0.103741097579
```

Figure 17: Linear Regression Estimated Coefficients for Workflow 1

For Workflow-1, the RMSE value is **0.10374**. This value is higher than the overall RMSE. The residual values are higher too.

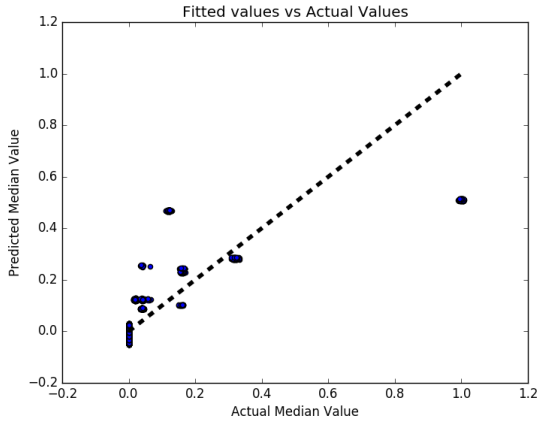

Figure 18: Linear Regression - Predicted Values vs Actual Values

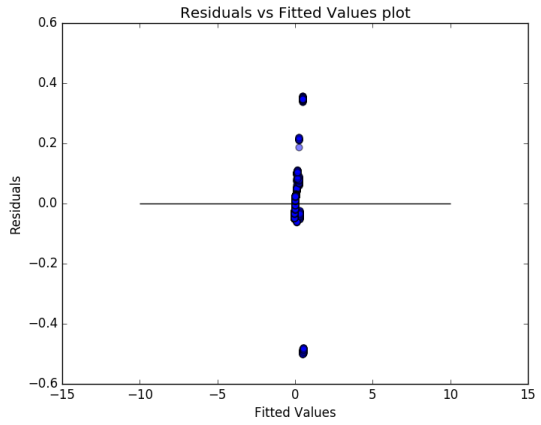

Figure 19: Linear Regression - Residuals vs Predicted Values

2.3.3 Linear Regression on WorkFlow_2:

```
Estimated intercept coefficient:  -0.00834410516762
   Features  EstimatedCoefficients
0         0                0.002232
1         1                0.004597
2         2                0.002039
3         3                0.000532
4         4                0.173552

RMSE Values of Estimator : 0.0255676397262
```

Figure 20: Linear Regression Estimated Coefficients for Workflow 2

For Workflow-2, the RMSE value is **0.0255**. This value is lesser than the overall RMSE. Figure 22 shows how the residual values lie between the range **0.05** and **-0.15**. The mapping of predicted and actual values is also scattered.

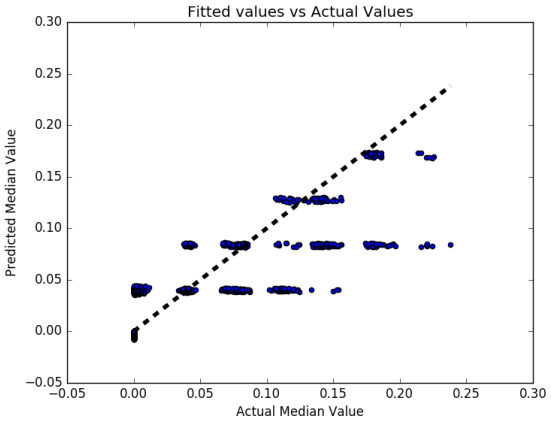

Figure 21: Linear Regression - Predicted Values vs Actual Values

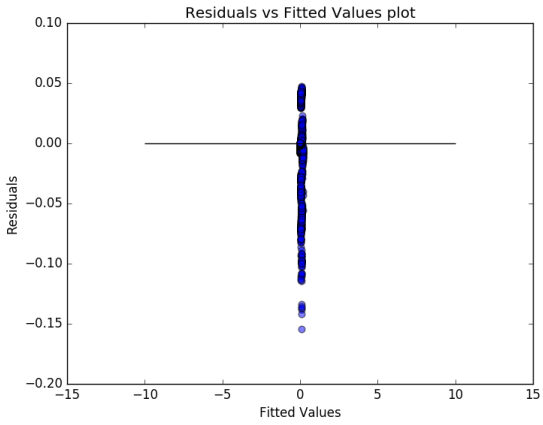

Figure 22: Linear Regression - Residuals vs Predicted Values

2.3.4 Linear Regression on WorkFlow_3:

The RMSE in this case is as low as **0.05917**. This is mainly due to the small copy sizes in this workflow. Figure 24 shows us how there is a constant prediction around 0.01.

```
Estimated intercept coefficient:  -0.00141100496131
   Features  EstimatedCoefficients
0         0                0.000498
1         1                0.001488
2         2                0.000955
3         3               -0.000207
4         4                0.016815

RMSE Values of Estimator : 0.00591750708412
```

Figure 23: Linear Regression Estimated Coefficients for Workflow 3

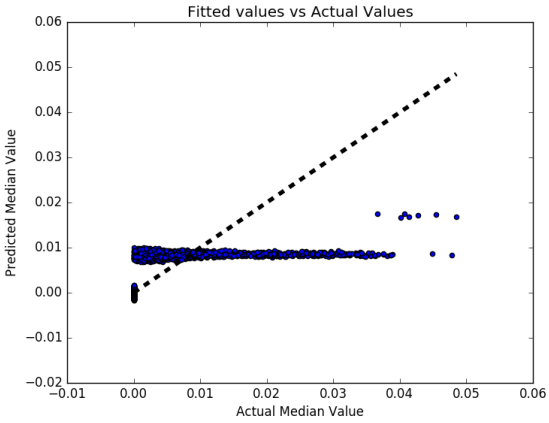

Figure 24: Linear Regression - Predicted Values vs Actual Values

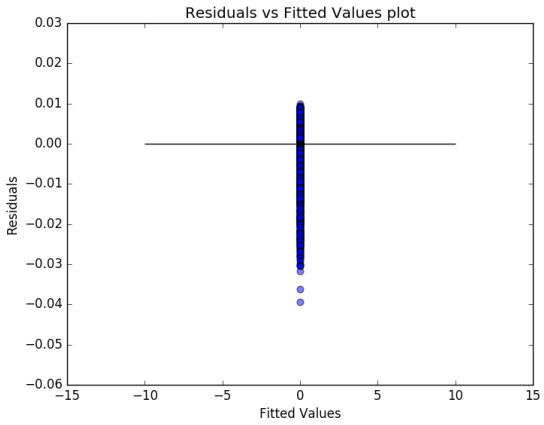

Figure 25: Linear Regression - Residuals vs Predicted Values

2.3.5 Linear Regression on WorkFlow_4:

```
Estimated intercept coefficient:  0.0336441737663
   Features  EstimatedCoefficients
0         0                0.001222
1         1                0.172215
2         2               -0.003978
3         3                0.001471
4         4                0.053850

RMSE Values of Estimator : 0.0842290809082
```

Figure 26: Linear Regression Estimated Coefficients for Workflow 4

The RMSE value for workflow 4 is **0.08422**. This is higher than the overal RMSE. Also, the residual values vary between **0.16** and **-1.3**. This is quite a large range of values.

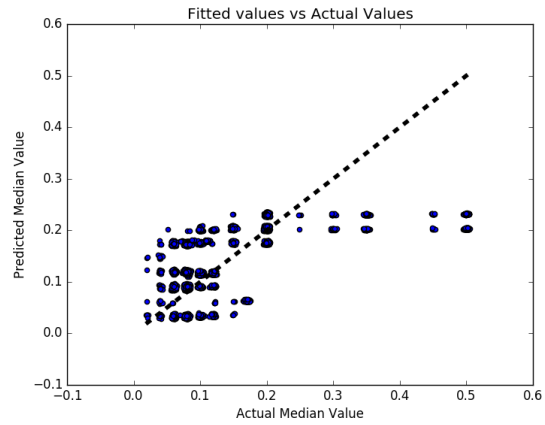

Figure 27: Linear Regression - Predicted Values vs Actual Values

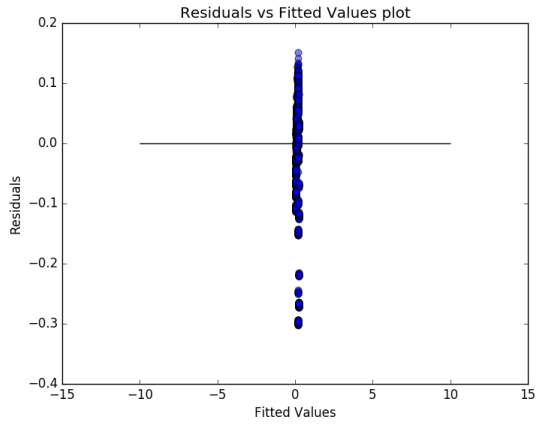

Figure 28: Linear Regression - Residuals vs Predicted Values

Thus the RMSE for individual workflows was better than the overall value whenever the fluctuation in copy sizes was comparatively smaller (in cases 0, 2, 3), while it was worse when there was a large swing in the copy sizes (cases 1 and 4).

2.3.6 Polynomial Regression

This section uses the Polynomial Regression function to improve the fit of the variables and improve the prediction of the copy size. The model was tested by fitting the polynomial functions with **degrees between 1 and 15**. By plotting the **RMSE vs polynomial degree** we see that the polynomial with degree 5 and above has the minimum RMSE value. Using this degree and splitting the entire dataset as 90 percent training and 10 percent testing, we get an RMSE value of 0.16.

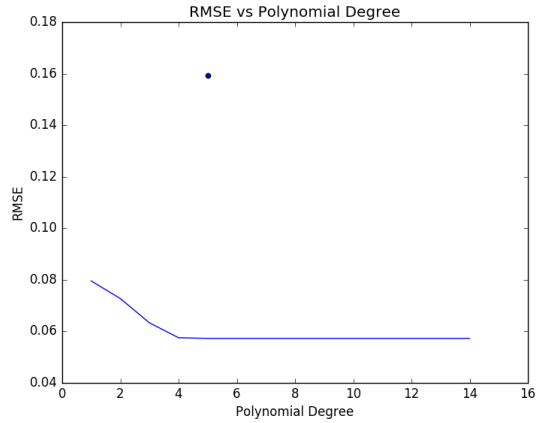

Figure 29: RMSE vs Polynomial Degree

Cross-Validation and Complexity

- Cross Validation is used to measure the performance of the predictive model. Since we are trying to achieve the best fit for model, we need a method to measure the performance of the model. Setting high degrees of freedom for a polynomial regression function often results in overfitting the data. In such cases, Cross-Validation helps in achieving the best fit on the data without overfitting the model.
- Too small a training dataset will not be able to give the right performance and too large a training dataset results in overfitting. It is necessary to maintain the right balance in the training and testing dataset. The Cross-Validation technique achieves a balance in the training and testin gof models as the testing data is got from the training data itself.
- We can use cross validation on a number of different training models to choose the best model.

3 Boston Housing

The Boston Housing Dataset has information about the housing values in the suburbs of the greater Boston area. The features captured are as follows:

1. **CRIM:** per capita crime rate by town
2. **ZN:** proportion of residential land zoned for lots over 25,000 sq. ft.
3. **INDUS:** proportion of non-retail business acres per town
4. **CHAS:** Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. **NOX:** nitric oxides concentration (parts per 10 million)
6. **RM:** average number of rooms per dwelling
7. **AGE:** proportion of owner-occupied units built prior to 1940
8. **DIS:** weighted distances to five Boston employment centers
9. **RAD:** index of accessibility to radial highways
10. **TAX:** full-value property-tax rate per $10,000
11. **PTRATIO:** pupil-teacher ratio by town
12. **B:**
$$1000(B_k - 0.63)^2$$
where Bk is the proportion of blacks by town
13. **LSTAT:** % lower status of the population
14. **MEDV:** Median value of owner-occupied homes in $1000s

3.1 Predicting MEDV using other attributes

In this section, we attempt to predict the value of MEDV based on the other attributes. The ordinary least square function is used as the penalty function. Two models were used to achieve this:

- Linear Regression
- Polynomial Regression

The method of 10-folds Cross-Validation was used to measure the performance of the model.

3.1.1 Linear Regression

The results obtained from running Linear Regression on the Housing dataset were as follows:

```
Executing...
                            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:                        19:27:19   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|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
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
AGE           -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
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.5e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

RMSE Values of Estimator : 5.89111669758
```

Figure 30: Visualization of the Linear Regression results

We see that the fitting of MEDV is dependent largely (95%) on the values of the remaining variables as described by the R-squares value. Also, except INDUS, NOX and AGE, all the other values are significant due to their low p values. The RMSE value of the estimator is **5.8911**.

Predicted vs Actual Values

In Figure 31, we see that the predicted values are close to the regressed diagonal line. Thus, the predicted and actual values are quite similar. However, the model underpredicts for values close to 50.

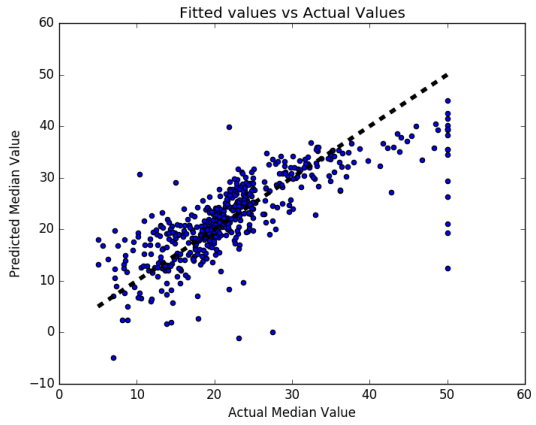

Figure 31: Linear Regression - Predicted vs Actual Values

Residuals vs Fitted Values

From the plot of the residuals shown below we can see that data is moore widespread and there is a wide spread in the residual values. There are also a large number of outliers due to the high absolute values of the residuals for some data points.

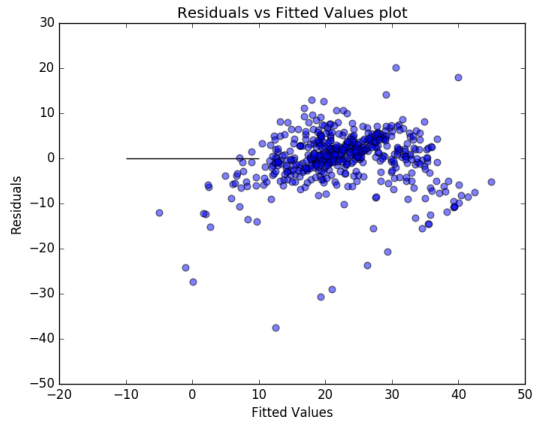

Figure 32: Linear Regression - Residuals vs Fitted Values of MEDV

3.1.2 Polynomial Regression

Here we see that the best fit is observed with a polynomial degree is from 1 to 3. The RMSE obtained for this model was **1.58317**.

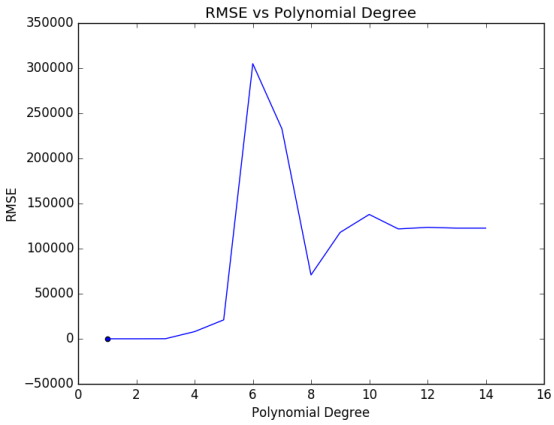

Figure 33: Linear Regression - Residuals vs Fitted Values of MEDV

3.2 Regularization of the Parameters

3.2.1 Ridge Regression

Best Alpha value for Ridge Regression : 1
Best RMSE for corresponding Alpha = 4.69515199361

3.2.2 Lasso Regression

Best Alpha value for Lasso Regularization : 0.01
Best RMSE for corresponding Alpha = 4.86585388387

References

[1] Defining Regression. `http://mathworld.wolfram.com/Regression.html`.

[2] Ho, Tin Kam (1995). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 1416 August 1995. pp. 278282. `http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf`.