EE219 Project 4 - Report

Regression Analysis

Winter 2018

Introduction

In this project, our goal is to explore various regression models on the given dataset along with basic techniques like cross-validation and regularization to handle overfitting.

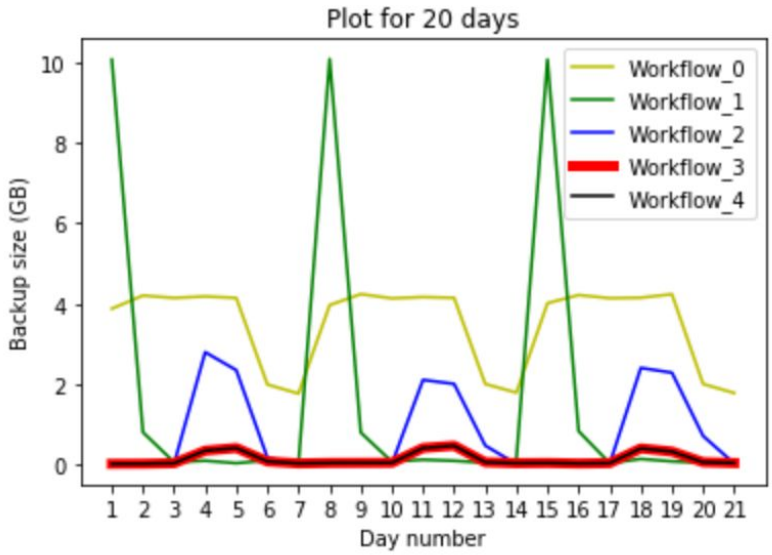
Dataset

We use a Network backup dataset, which comprises of simulated traffic data on a backup system over a network. The dataset has around 18000 points with the following fields: Week number, day of the week, backup start time, workflow ID, file name, backup size and backup time. We have to predict the backup size of the file using all the attributes except backup time, as the candidate features.

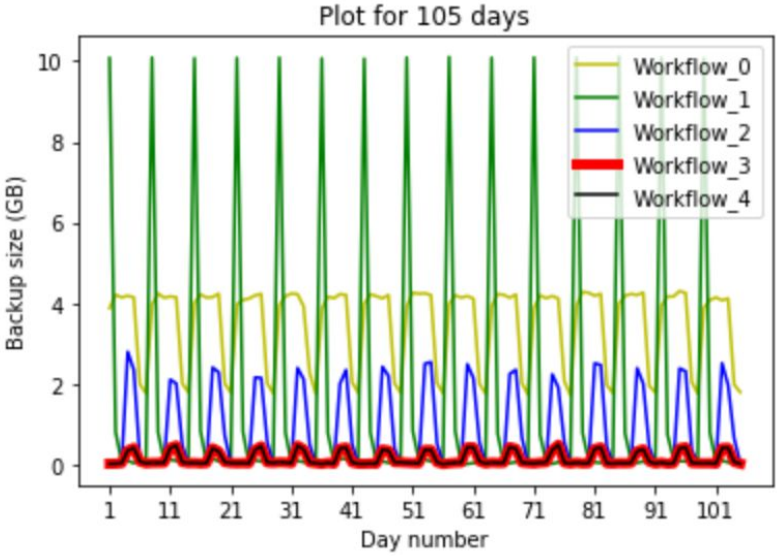
Q1: Plot backup sizes for all workflows

In this task, we plot backup sizes for all the workflows over a period of 20 days to understand the relationship between variables in the dataset. We first preprocess the data by changing the categorical variables to numeric values using scalar encoding and plot the relationship between Backup size and the day number as shown in the figure below:

(a) For a 20-day period



(b) For a 105-day period



From the above figures, we observe repeating patterns for different workflows. Each workflow follows a pattern which is repeated over a period of 7 days or a week. For instance:

- For workflow-0, the backup sizes from Monday to Friday are higher compared to the backup sizes on Saturday and Sunday.
- For workflow-1, there is a sudden decrease in the backup size on day 2 i.e. Tuesday and becomes zero Wednesday onwards.

- For workflow-2, backup is only done on day 4 and day 5 i.e. Thursday and Friday. No backup is done on rest of the days.
- Backup size pattern for workflow-3 is also similar to that of workflow-2. However, the size of the data that is being backed up in workflow-3 is very less compared to the data backed up in workflow-2.
- Also, it can be observed that workflow-4 follows the exact same pattern as workflow-3

Q2(a): Linear Regression Model

(i) Scalar Encoding

Each categorical feature was converted into numerical values using scalar encoding and were then used to fit a basic linear regression model.

The RMSE values reported vere as follows:

Test RMSE = 0.10193944624209859

Train RMSE = 0.10183435819796752

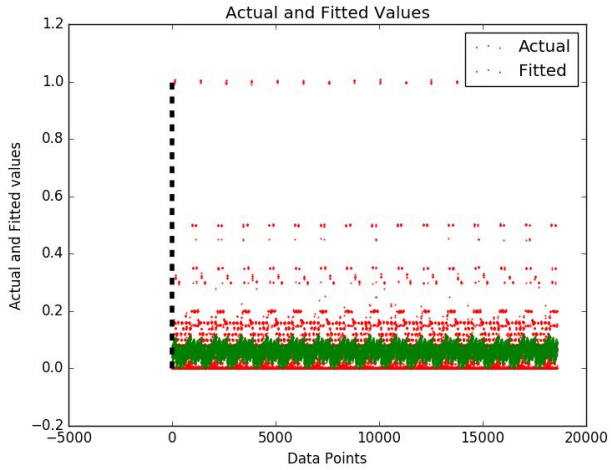


Figure: Actual and Fitted Values

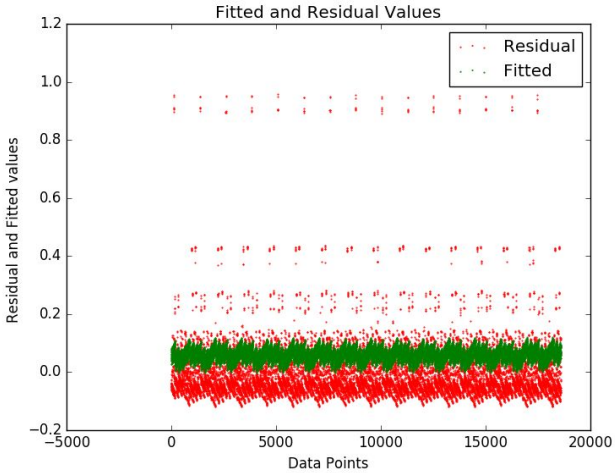


Figure: Residual and Fitted Values

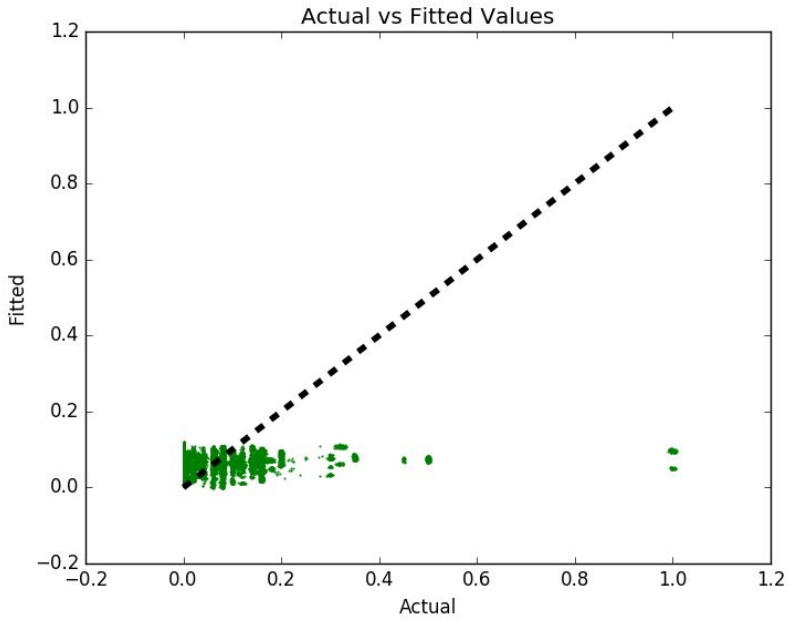


Figure: Actual vs Fitted values

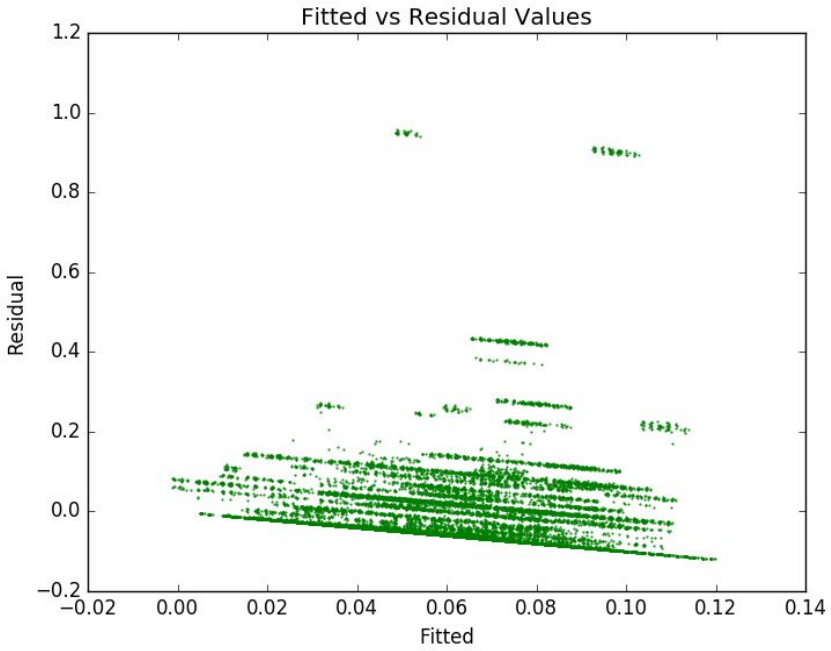


Figure: Residual vs fitted values

(ii) Data-preprocessing

Standardization (Centering and Scaling) was done on all the features.

The RMSE values reported after standardizations were as follows:
Test RMSE = 0.10193944624209859
Train RMSE = 0.10183435819796753

Standardization does not bring about any improvements in accuracy.

The basic linear regression model used above is **invariant to standardization**. The parameters learnt after standardization gives exactly the same model as the one before standardization and this results in the exact same test RMSE values.

In general, standardization which involves **linear transformations**(scaling and centering) does **not bring about any changes to linear models**. This may not be true for non-linear transformations (polynomial regression, ridge and lasso regression etc.)

(iii) Feature selection

Normalised F-regression values for the features:
Week index: 1.83383477e-05
Day of the week: 4.78772999e-01
Backup Start time - Hour of the day : 3.27138366e-01
Workflow ID : 5.67261995e-02
File Name : 1.00000000e+00

Using f_regression the 3 most important features were found to be: **File Name, Backup Start time - Hour of the day and Day of the week**

After selecting the best 3 features, the RMSE values reported were as follows:
Test RMSE : 0.101893426765
Train RMSE: 0.101871974794
The test RMSE was found to be nearly the same (improved by a very small amount 0.046%).

Normalised Mutual Information values for the features:
Week index: 0. 00000007

Day of the week: 0.43676439
Backup Start time - Hour of the day : 0.59709141
Workflow ID : 0.68792305
File Name : 1.

Using mutual information the 3 most important features were found to be: **File name, Workflow ID and Backup Start time-Hour of the day**

After selecting the best 3 features, the RMSE values reported were as follows:
Test RMSE : 0.102547286771
Train RMSE: 0.102465412867

In both the feature selection methods the Week Index was found to be an irrelevant feature.

(iv) Feature encoding

Feature Encoding (S=scalar, V=vector)	Test RMSE	Train RMSE
S S S S S	0.10193944624209859	0.10183435819796752
S S S S V	0.090968244789234071	0.090793594435595593
S S S V S	0.090965435745335804	0.090796238285714856
S S S V V	0.090967777891921967	0.090793626996754584
S S V S S	0.1007047686560026	0.10058577720887095
S S V S V	0.089573220589374003	0.089385277370006191
S S V V S	0.089574171641399442	0.089386331588134707
S S V V V	0.089584844887701701	0.089385689070760124
S V S S S	0.1009757241245799	0.10088992814449357
S V S S V	0.089910883168603975	0.089758534916399665
S V S V S	0.089908188253164198	0.089758188931314264
S V S V V	0.089920089156094091	0.089758870079687775

S V V S S	0.099738534894066852	0.099637685671444709
S V V S V	0.088507099324723421	0.088338505839689607
S V V V S	0.088507360144404648	0.088339889994547352
S V V V V	**0.088506373687921721**	**0.08834013560010659**
V S S S S	8065370829	0.10182832364470487
V S S S V	17461609275.458393	0.090787811865523385
V S S V S	15615450317.490131	0.090789528399592756
V S S V V	26896548047.500687	0.090787165405974535
V S V S S	7001681485.6499958	0.10057945858835408
V S V S V	36800500816.852264	0.08937824464040843
V S V V S	22583523876.873058	0.089379496546441534
V S V V V	109184256519.27597	0.089392972368840057
V V S S S	13148003254.87126	0.10088738931541147
V V S S V	17207308281.642895	0.089753728608563621
V V S V S	19447376948.255455	0.089755368318915452
V V S V V	52596226886.327904	0.089754490107035734
V V V S S	23400956820.067589	0.099634661654162071
V V V S V	245424233914.4487	0.088342175774585169
V V V V S	18502358572.67556	0.088337748757025519
V V V V V	555674402231.9502	0.088368971131411705

The best accuracy was obtained when the first feature (Week Index) was scalar encoded and the rest vector (one hot) encoded.

In general, the models from 1-15, which had different combinations of the features : File name, Workflow Index, Backup Start time and Day of the week one hot encoded gave low RMSE values. The test RMSE values shot up when we one hot encoded the week index feature.

One hot encoding improves model accuracy when there is no ordinal relationship of a categorical variable. In such a case scalar encoding allows the model to assume a natural ordering which results in poor performance.

The categorical features in the data set do not have an ordinal relationship i.e the magnitude of the categorical variable has no significance on the backup time and is just used as a label for the different values . Hence one hot encoding improves the accuracy of the model.

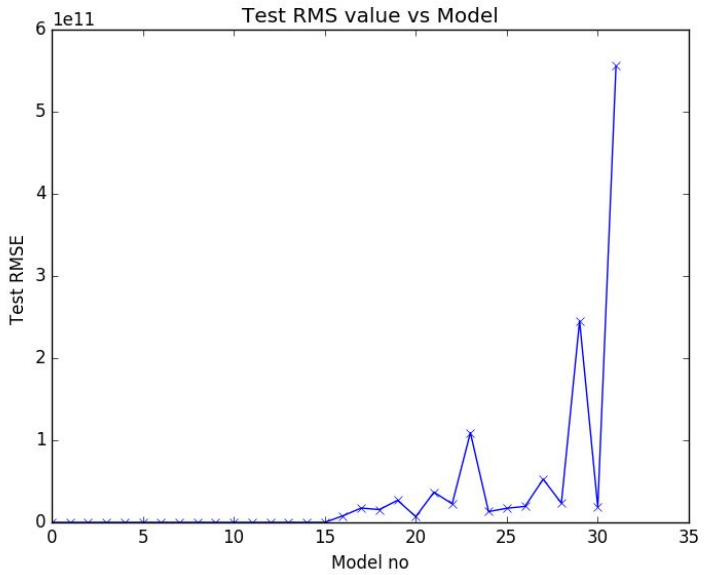


Figure: Test RMSE vs model

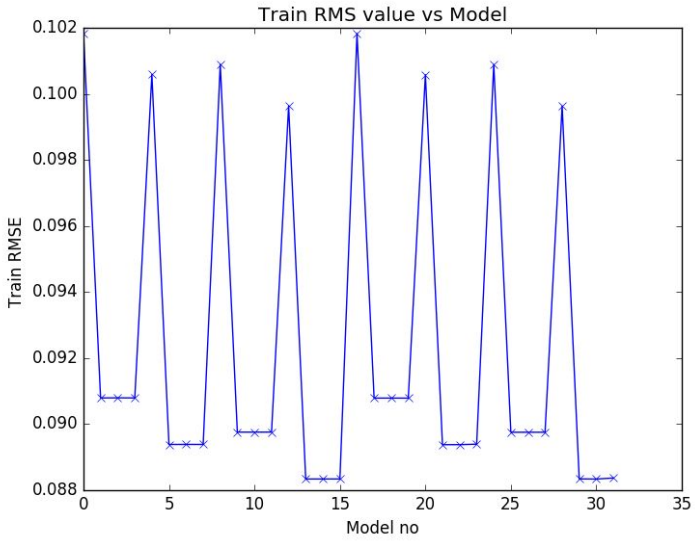


Figure: Train RMSE vs model

v. Controlling ill-conditioning and over-fitting:

Ridge Regularization:

We observe extremely high values of Test RMSE for certain combinations of encodings especially when an irrelevant feature like Week Index is one hot encoded. The train RMSE however remains consistent across all encodings. This is because of overfitting the model.
This can be fixed by using regularization. The model with the Week ID encoded as a scalar and the remaining features encoded as vector was found to give low test RMSE values for any value of alpha. Hence this model was chosen to determine the optimum alpha value for ridge regularization.

The Test RMSE for the chosen model was found to be nearly the same for alpha >= 24

The average Test RMSE across all the 32 models for alpha = 24 was found to be 0.09256014015238212

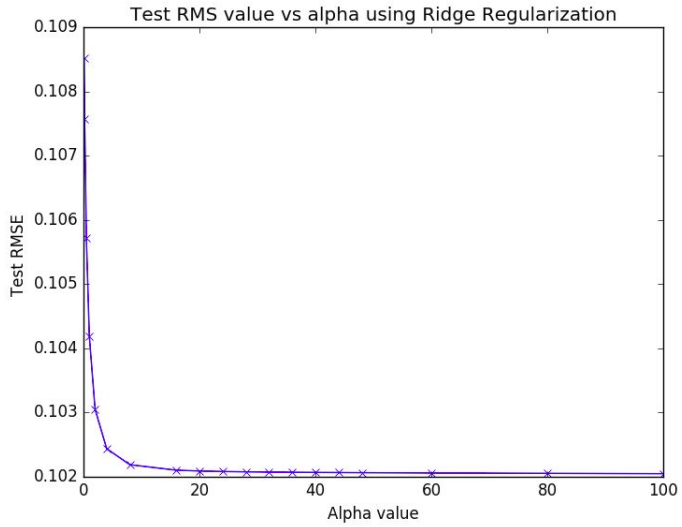


Figure: Plot of Test RMSE for vs alpha for the chosen model

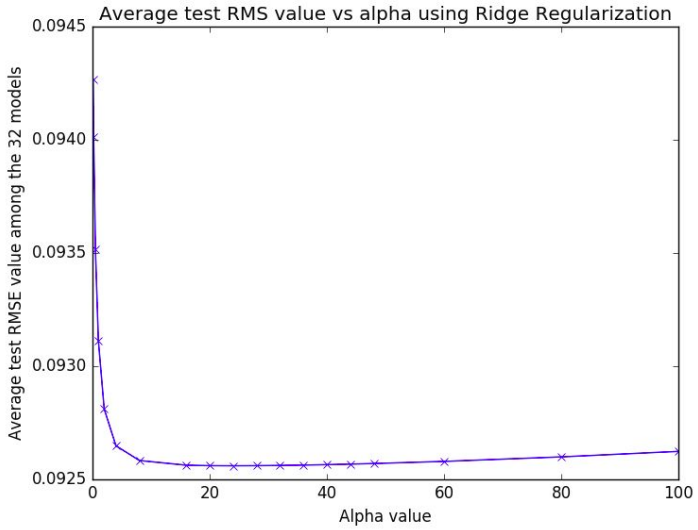


Figure: Plot of average Test RMSE for the 32 models vs alpha

Lasso Regularization:

The Test RMSE for the chosen model was found to be nearly the same for alpha >= 0.001

The average Test RMSE across all the 32 models for alpha = 0.001 was found to be the minimum.

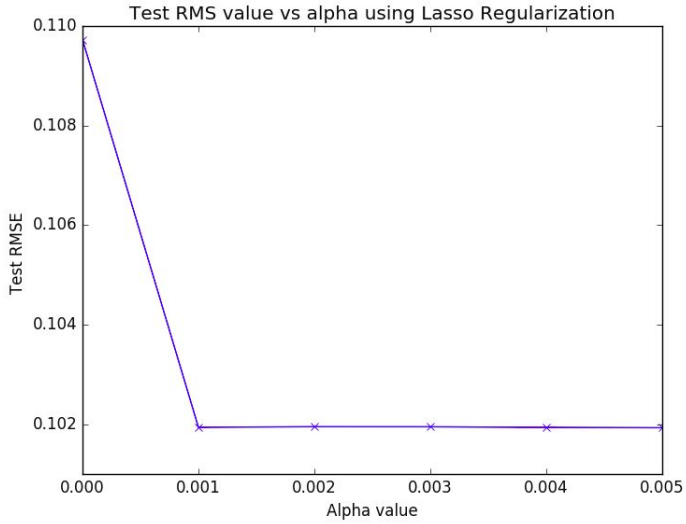


Figure: Plot of Test RMSE for vs alpha for the chosen model

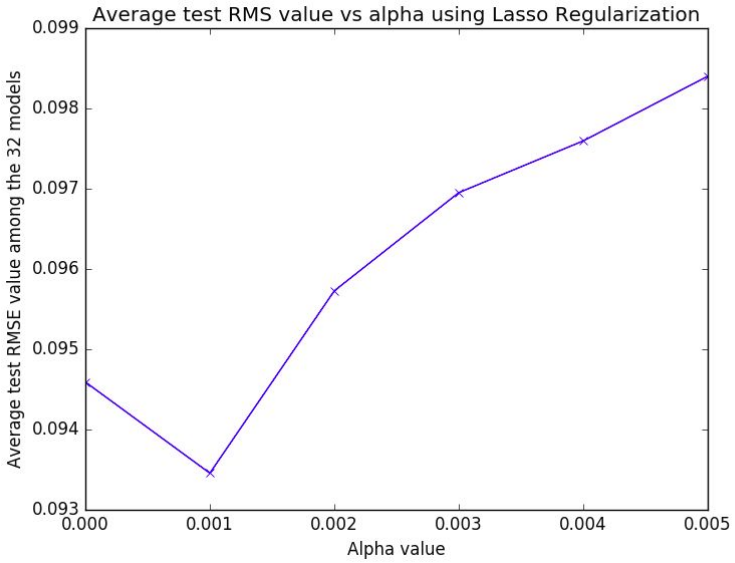
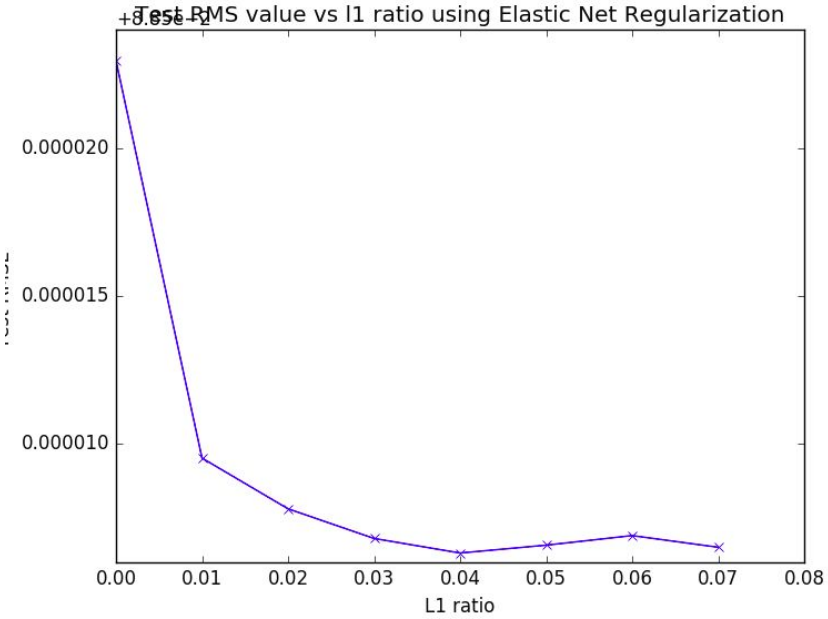


Figure: Plot of average Test RMSE for the 32 models vs alpha

The optimum value for alpha for lasso regression was found to be **0.001**

Elastic Net Regularization



For alpha=0.001 and L1 ratio=0.04 ,the Test RMSE was minimum for the chosen value and was found to be equal to **0.0863380774977**

The **best mode**l was found to be the one with the following settings:

1. Scalar Encoding for the first feature and vector encoding for the remaining 4 features
2. Elastic Net Regularization with alpha=0.001 and L1 ratio=0.04

The test RMSE for the best model with regularization is **0.0863380774977**

The estimated coefficients were compared before and after regularization:

Coefficients of the best model without regularization:

[-1.05677056e+09 -1.05677056e+09 -1.05677056e+09 -1.05677056e+09
-1.05677056e+09 -1.05677056e+09 -1.05677056e+09 -5.10305327e+09
-5.10305327e+09 -5.10305327e+09 -5.10305327e+09 -5.10305327e+09
-5.10305327e+09 1.01834650e+10 5.39658123e+09 -8.77655282e+08
1.82220973e+09 -2.19268368e+09 -1.05719955e+10 -1.05719955e+10
-5.78511172e+09 -5.78511172e+09 4.89124787e+08 4.89124787e+08
4.89124787e+08 4.89124787e+08 4.89124787e+08 4.89124787e+08
-2.21074022e+09 -2.21074022e+09 -1.05719955e+10 -2.21074022e+09

-2.21074022e+09 -2.21074022e+09 -2.21074022e+09 1.80415319e+09
1.80415319e+09 1.80415319e+09 1.80415319e+09 1.80415319e+09
1.80415319e+09 -1.05719955e+10 -1.05719955e+10 -1.05719955e+10
-5.78511172e+09 -5.78511172e+09 -5.78511172e+09 -5.78511172e+09
1.14440918e-05]

The test RMSE for the best model without regularization is **0.088506373687921721**
The estimated coefficients are:

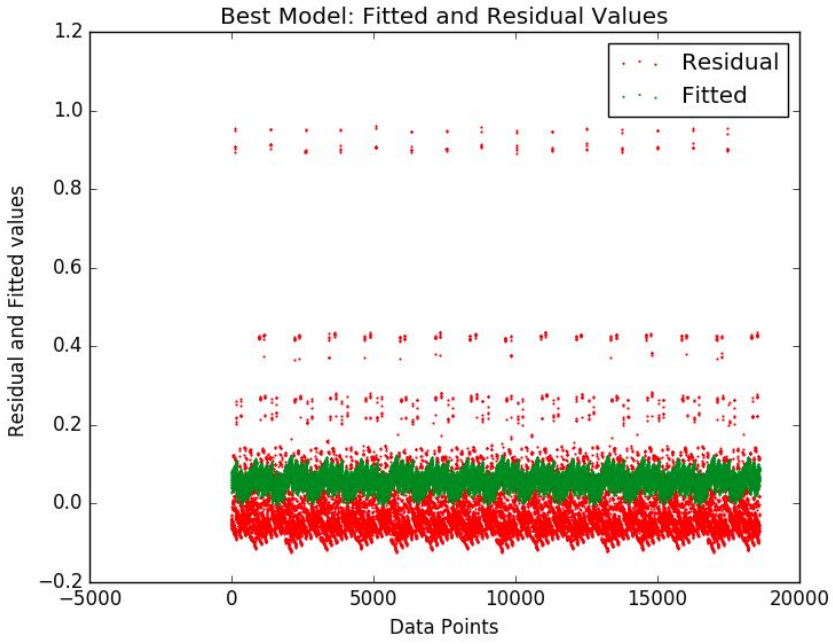
Coefficients of the best model with **ridge regularization**:
[6.91525761e-02 1.14170617e-01 7.81361495e-02 7.63556617e-02
6.96107027e-02 6.19948980e-02 5.45918826e-02 -1.16997620e-01
-1.17854090e-01 -8.90010344e-02 -6.33377841e-02 -9.87796364e-02
-9.47849160e-02 3.49228418e-01 7.51978960e-02 5.34549834e-01
-3.76816186e-01 3.06126122e-02 -5.97191138e-01 -5.95942301e-01
-3.75008894e-01 -3.74680980e-01 -8.60964953e-01 -8.60550397e-01
-8.60076124e-01 -8.61144132e-01 -8.58624772e-01 -8.59956507e-01
3.42824819e-02 3.46095667e-02 -5.96271307e-01 3.46684393e-02
3.43088051e-02 3.42601993e-02 3.45293433e-02 -2.43392032e-01
-2.42828627e-01 -2.42639131e-01 -2.43705378e-01 -2.43059269e-01
-2.43168363e-01 -5.96297804e-01 -5.96688346e-01 -5.96441899e-01
-3.74345593e-01 -3.74829267e-01 -3.74620550e-01 -3.74904119e-01
1.12089931e-05]

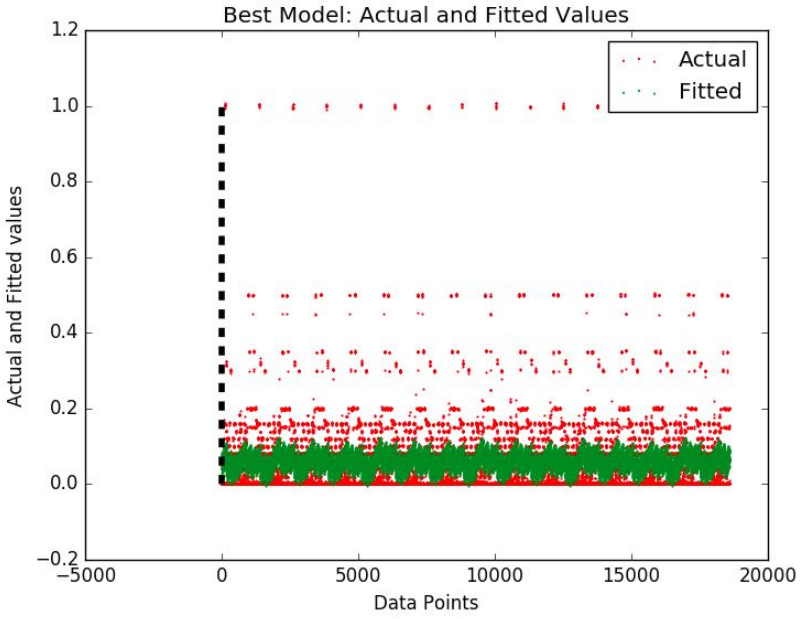
Coefficients of the best model with l**asso regularization**:
[-0. 0.0043795 0. 0. -0. -0. -0.
-0. -0. 0. 0.00475332 -0. 0.
0.02745336 -0. -0.00122776 -0.01821043 0.0624691 0. 0.
-0. 0. -0. -0. -0. -0. -0.
-0. -0. -0. 0. -0. -0. -0.
-0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. -0. 0. -0.
0.]

Coefficients of the best model with **elastic Net regularization**:
[-4.07476692e-04 4.37600970e-02 7.96510354e-03 6.19630585e-03
-0.00000000e+00 -7.48444138e-03 -1.48774398e-02 -1.99000103e-02
-2.07238293e-02 7.46962620e-03 3.29884026e-02 -1.78137106e-03
1.72273021e-03 4.58248317e-02 -5.81021772e-03 -3.18337774e-02

-4.84679681e-02 8.03846427e-02 -0.00000000e+00 0.00000000e+00
-0.00000000e+00 -0.00000000e+00 -0.00000000e+00 -0.00000000e+00
-0.00000000e+00 -8.40378310e-05 4.65828299e-05 -0.00000000e+00
-0.00000000e+00 -0.00000000e+00 0.00000000e+00 -0.00000000e+00
-0.00000000e+00 -0.00000000e+00 -0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 -0.00000000e+00 -0.00000000e+00 -0.00000000e+00
 9.10093408e-06]

We can observe that the learned coefficients are **very large prior to regularization**. The coefficients decrease from ridge, lasso to elastic net regularization with elastic net regularization having the smallest coefficients. Many of the coefficients have a value of zero after lasso/elastic regularization.





Q2(b): Random Forest Regression Model

Random Forest Regression model is a ensemble learning method that fits a number of different classifying decision trees on different subsets of the data and then averages to obtain more accuracy in prediction. This method corrects for overfitting on the training set.

(i) Cross Validation

We perform 10 fold cross validation on the network backup dataset, using the five features except backup time in order to predict the backup size. The model is trained and used to predict on the testing data as well as the training data in order to obtain average Root Mean Square Error values. The left over points while building a tree can be used to test it and the RMSE of the prediction so obtained over all the trees is known as the Out of Bag (OOB) score, which gives the Out of Bag Error as 1-OOB score.

The initial parameters are as follows:

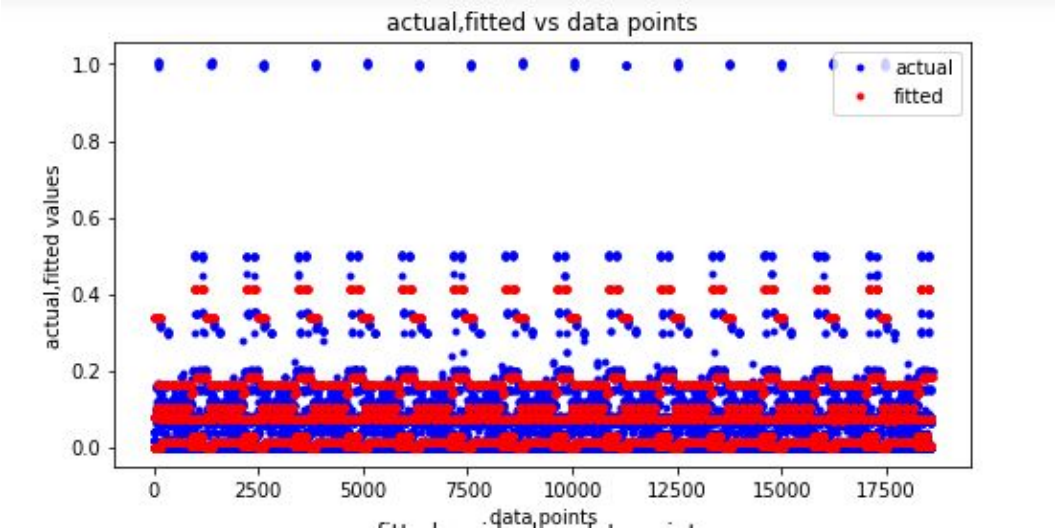
Number of Trees: 20
Depth of each tree: 4
Maximum number of features: 5

The values obtained are:

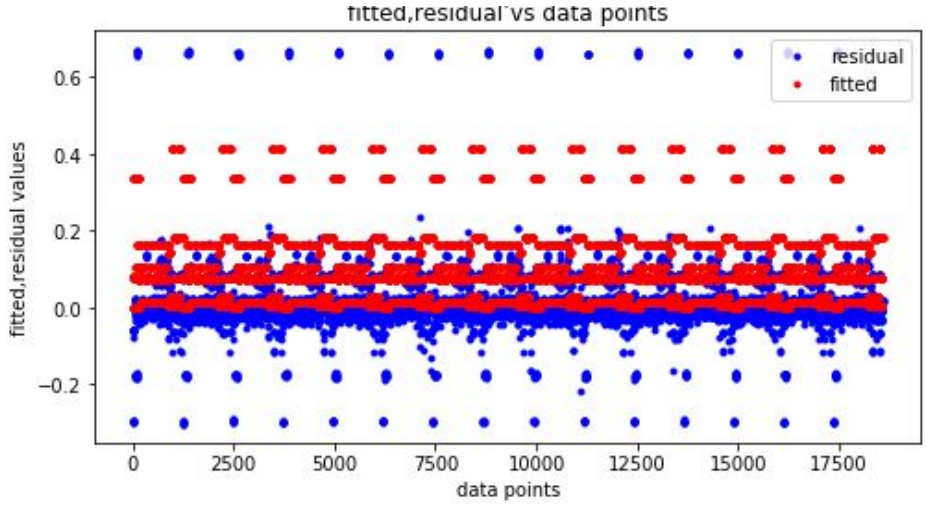
Average Test RMSE	Average Train RMSE	OOB error
0.06056957691429288	0.06039639292754405	0.332637709343

The model is then fit on the whole data set and used to predict backup size. The obtained fitted and actual values are plotted over the set of data points to see how well the model fits the data. The residual values, obtained as (true value-predicted value) are similarly plotted with the actual values over the data points

ACTUAL AND FITTED VALUES VS DATA POINTS



FITTED AND RESIDUAL VALUES VS DATA POINTS



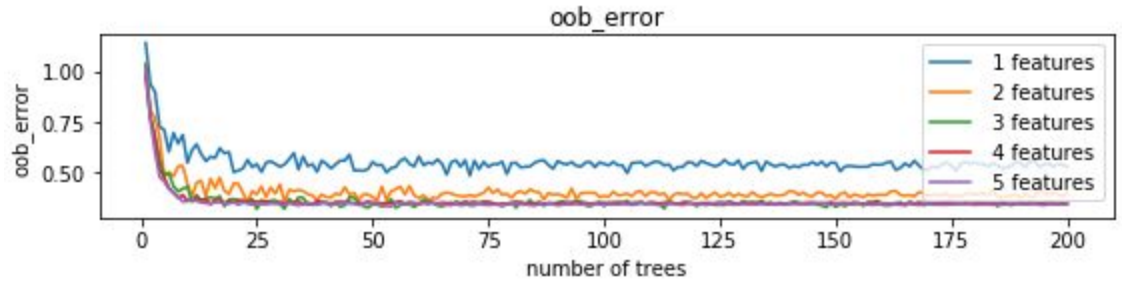
Clearly, there is an overlap of the predicted values over the actual values showing that the model works well but there are still outliers indicating the need for finding the more optimal parameters.

The residual values lie around zero indicating that it's a good model.

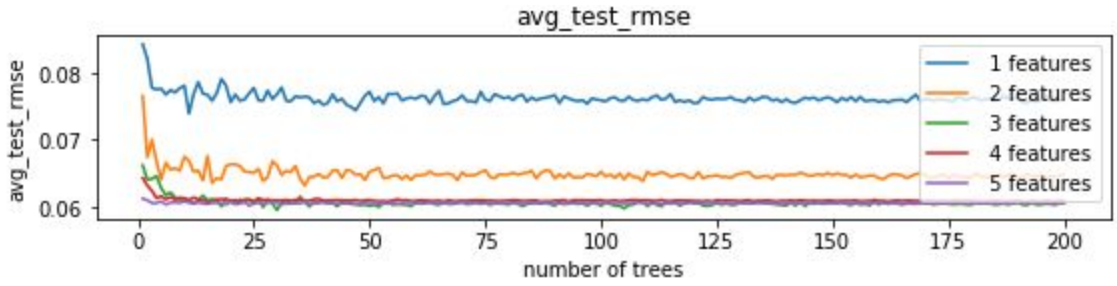
(ii) Maximum Feature Sweep

We try to find the better model by sweeping the number of trees upto 200 for each value of maximum feature from 1 to 5. For each feature, the oob error and the average test rmse error is plotted against the number of trees The model which yields the lowest value of RMSE is selected as the best model and the fitted and residual values are visualized for the same

OUT OF BAG ERROR VS NUMBER OF TREES



AVERAGE TEST RMSE VS NUMBER OF TREES

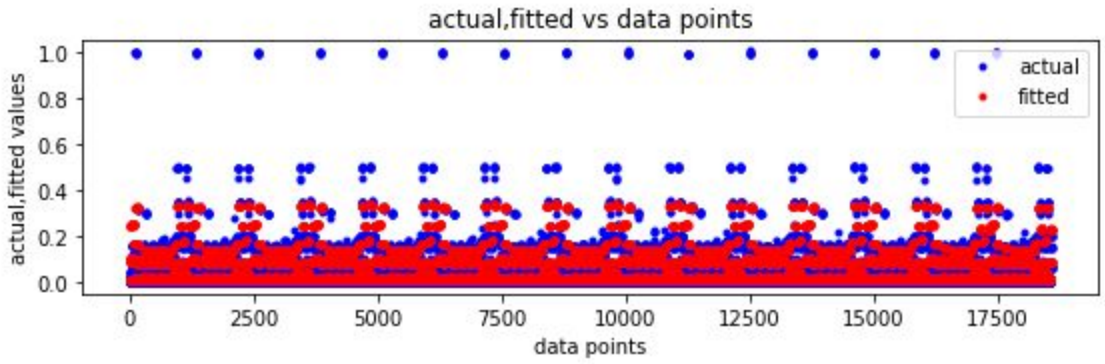


The best model obtained has the following parameters

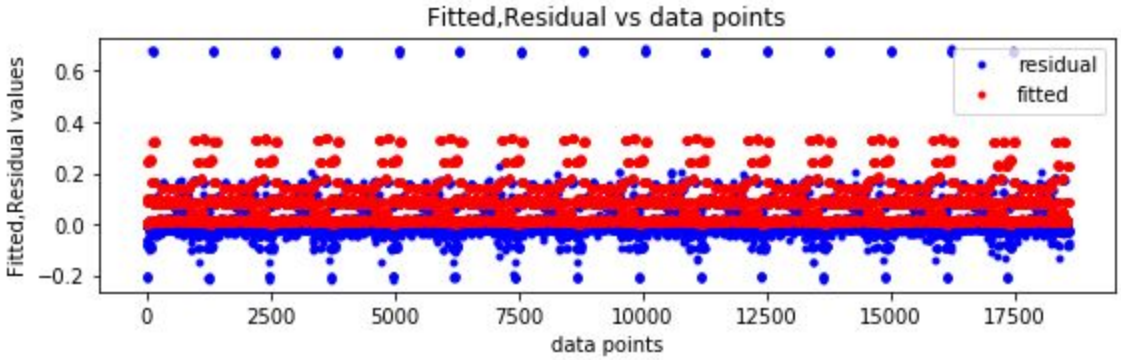
RMSE	**Number of Trees**	**Maximum Feature**
0.06043915396449869	55	3

This matches the intuition from the graphs
For corresponding fitted and residual points are plotted as follows:

ACTUAL AND FITTED VALUES VS DATA POINTS



FITTED AND RESIDUAL VALUES VS DATA POINTS

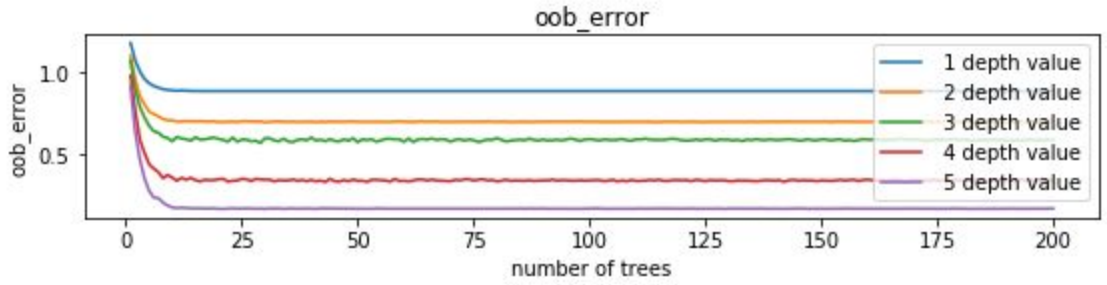


Clearly the model performs better than the initial model, when one parameter is optimized as observed from the large overlap between actual and fitted values. However, not all outliers have been eliminated

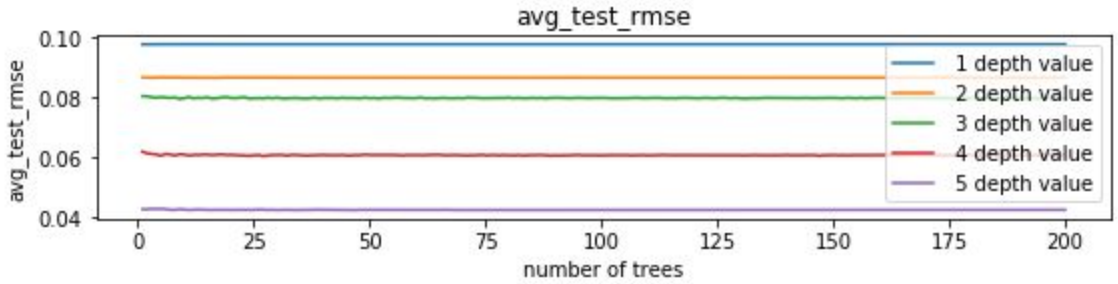
(iii) Depth of Tree Sweep

In order to find the best possible parameter which minimizes the prediction error, we pick another parameter, namely depth of each tree, and vary it over a range from 1 to 5. For each value of depth, the oob error and the average test rmse error are plotted against the number of trees. The model which yields the lowest value of RMSE is selected as the best model and the fitted and residual values are visualized for the same

OUT OF BAG ERROR VS NUMBER OF TREES



AVERAGE TEST RMSE VS NUMBER OF TREES

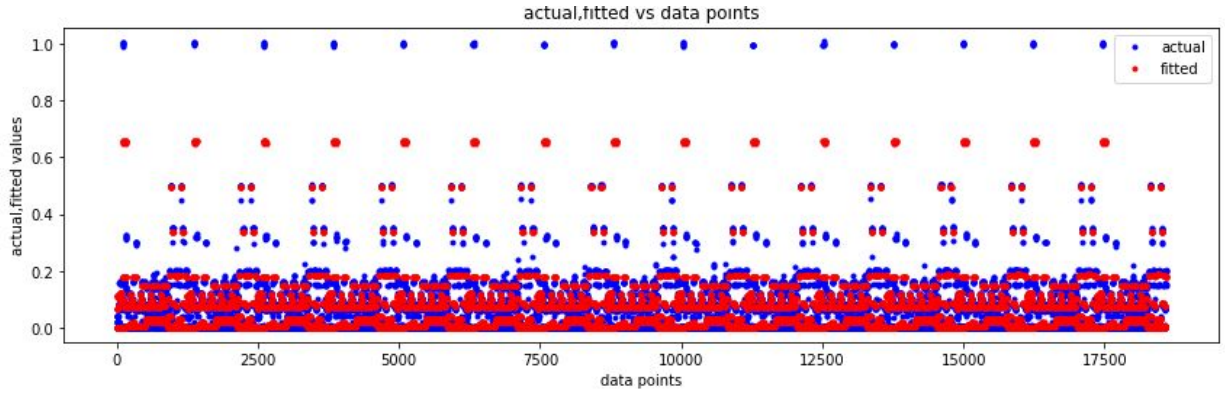


The best model obtained has the following parameters:

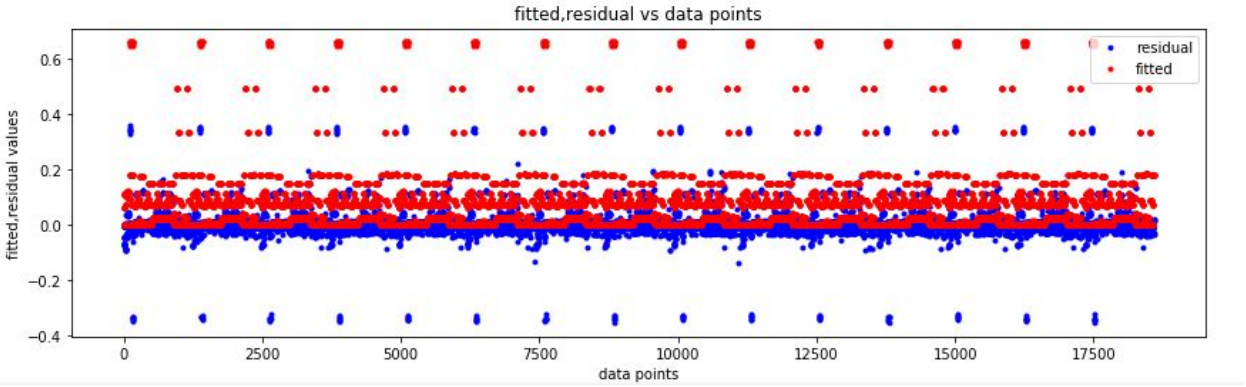
RMSE	Number of Trees	Depth of Tree
0.04224188728779368	74	5

There seems to be no significant decrease in the RMSE value over this range of depths as it is varied over the selected number of trees. Thus the range of search is extended in attempting to find the best model and from a grid search, the value is obtained as 9

ACTUAL AND FITTED VALUES VS DATA POINTS



FITTED AND RESIDUAL VALUES VS DATA POINTS



In keeping with the observation that the model is not varied upto the best possible depth, the overlap between the predicted and actual values does not seem as evident as the previous case for the best maximum feature.

Though the computing time as a very realistic constraint in finding the best value of depth, based on the observed error values, the Depth of the tree is the best parameter used to achieve better performance.

In order to find the best possible model, a exhaustive grid search is performed to find the optimal values for the different number of trees, maximum number of features and depth of trees. Different initial values as before are fixed for two of the three parameters and the other is varied over a selected range to find the best value and each parameter is progressively tuned to find the best model.

The best values thus obtained are:

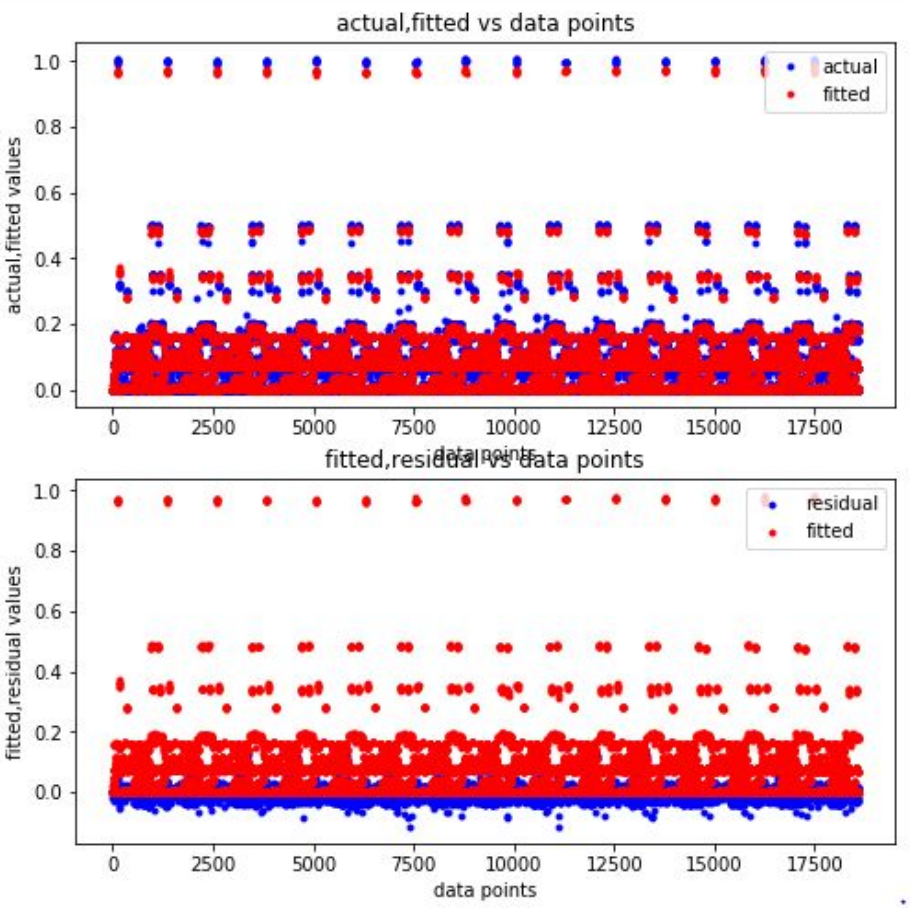
Maximum features	**Number of Trees**	**Depth of Tree**
3	120	9

While determining each of these parameters with the other two fixed, the obtained RMSE values are:

	Maximum features	**Number of Trees**	**Depth of Tree**
RMSE	0.012948024711	0.0603813600023	0.0600512838141

This reaffirms are observations that Depth of the tree is the best parameter to achieve the best performance

The fitted and residual values for this model are as follows:



The residual values are around zero and there is a good overlap between the predicted and actual values. Clearly the random forest model performs well.

(iv) Feature Importances

For this model, the feature importances are obtained as:

Week Index	**Day of the week**	**Backup Start Time: Hour of day**	**Workflow ID**	**File name**
0.00423868	0.27383218	0.35297722	0.15422674	0.21472519

Clearly, the most informative features which are most useful in predicting the size of the backup file are Backup start time, Day of the week and File name . On the other hand, Week index is not very useful for the same task.

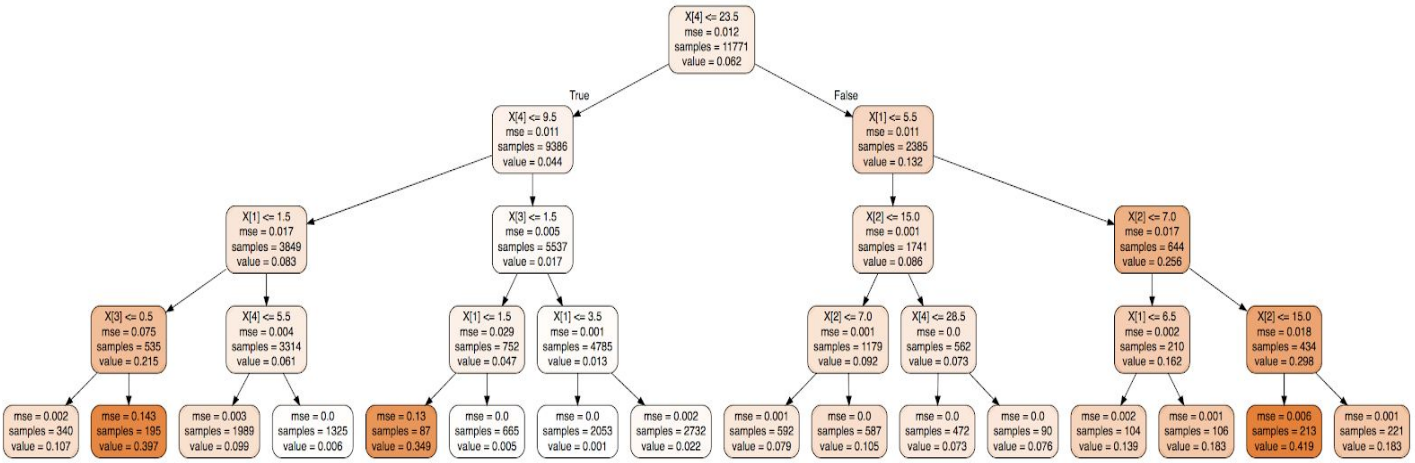
(v) Visualization:

From the above best random forest with maximum depth=4, one tree is picked and visualized using the graphviz method.

The feature importances when the maximum depth is 4 is obtained as follows:

Week Index	**Day of the week**	**Backup Start Time: Hour of day**	**Workflow ID**	**File name**
0.00033444	0.32097783	0.11629388	0.24077253	0.32162132

From the tree, it is observed that the root of the tree corresponds to Backup start time while the observed best feature is Backup start time as well. Thus, when the depth is the same, the most important feature is uniform across the two.

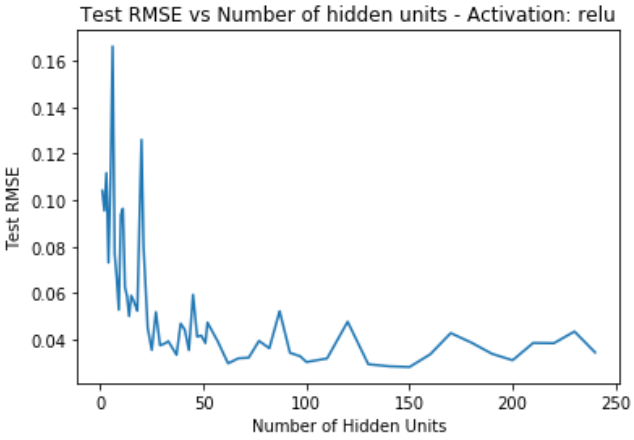


Q2(c): Neural Network Regression Model

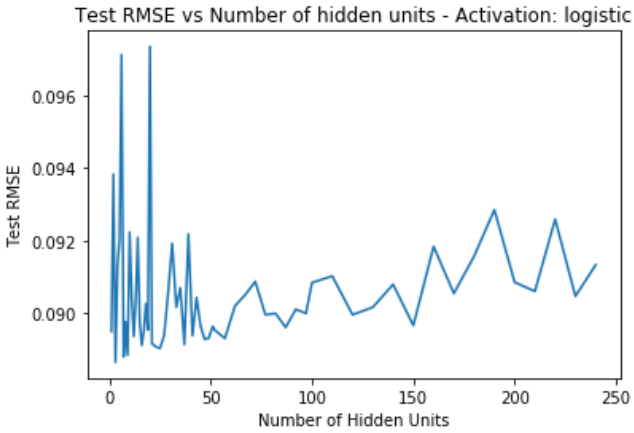
In this task, we used scikit-learn's MLPRegressor to implement a neural network regression model. We one-hot encode all the features and also perform 10-fold cross validation to avoid overfitting. We only have one hidden layer in the neural network and vary the number of hidden units from [1, 250] and activation functions (relu, logistic and tanh). Our observations and plots are as follows:

Activation	Minimum Test RMSE	Hidden units	Minimum Train RMSE	Hidden units
Relu	0.0283282997893980	150	0.01730109545997070	240
Logistic	0.0886389829934241	3	0.08829939410831376	4
Tanh	0.0604909019414150	39	0.05329525258277334	52

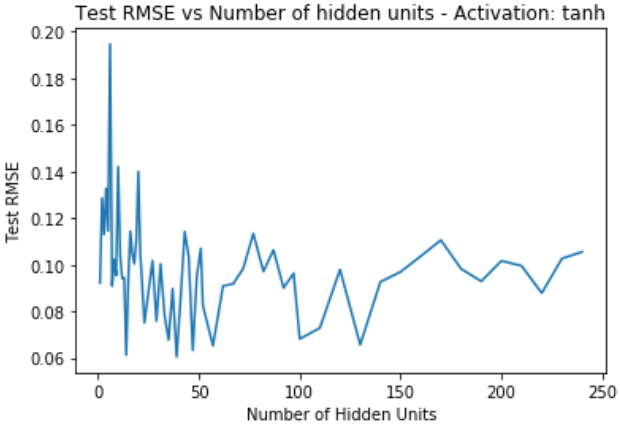
Activation: Relu



Activation: Logistic



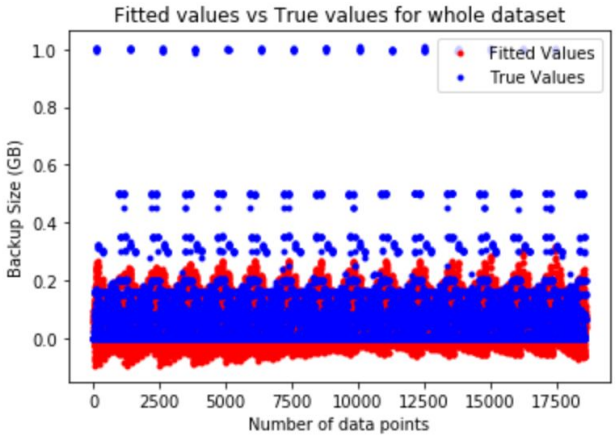
Activation: Tanh

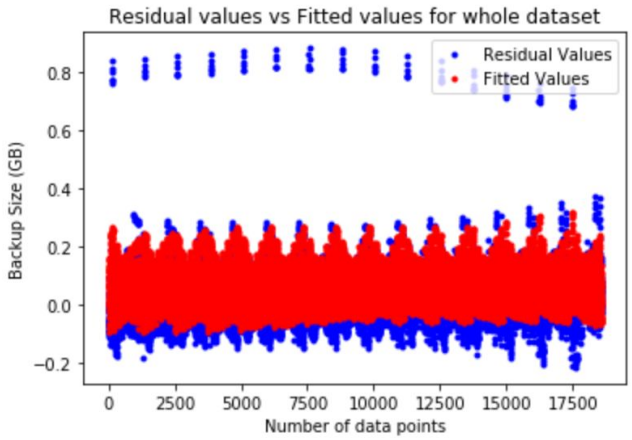


Best model: Hidden units: 231, activation = 'relu'

It can be observed from the above plots that Test RMSE values decrease with increase in the number of hidden units. The lowest RMSE value is observed when we use 'relu' activation function with 231 hidden units. Hence, we conclude that a neural network with 'tanh' activation function and 231 hidden units is the best model.

Also, to visualize how well the best model fits the data, we plot the following plots of fitted and true values vs the data points and residual and true values vs the data points for the whole dataset.





It can be observed that the neural network regression models fits the data better than the previous models.

Q2(d): Backup Size for each workflow

(i) Using Linear Regression Model

In this part, we use the linear regression model with 10-fold cross-validation on different workflow datasets separately. The minimum and best RMSE values observed for each workflow are as shown in the table below:

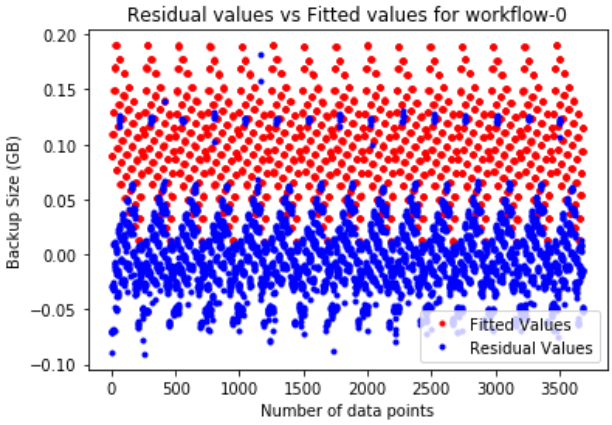
Workflow	Average RMSE	Best RMSE
Workflow-0	0.0358869702489	0.0340834964739
Workflow-1	0.148918602014	0.123710561768
Workflow-2	0.0430669058479	0.0354878192652
Workflow-3	0.0072608942421	0.00567623648231
Workflow-4	0.0859906141157	0.073903648246

For linear regression with 10-fold cross validation on the entire dataset, we observe that the average RMSE value is 0.103675847676 and the best RMSE value is 0.0999471208611.

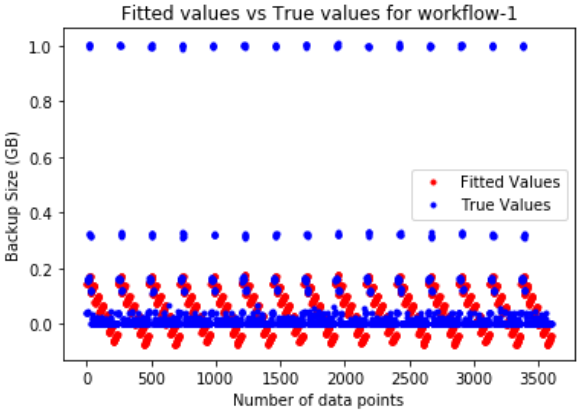
Since each workflow has a different pattern for changing the backup size, separate data for each workflow would perform better than the entire dataset. When compared to the average and the best RMSE values for the entire dataset, it can be seen that, all the workflows, except workflow-1, have lesser average and best RMSE values. So it can be said that the model works better on separate datasets for each workflow as opposed to the entire dataset. This can also be observed from the visualization plots shown below.

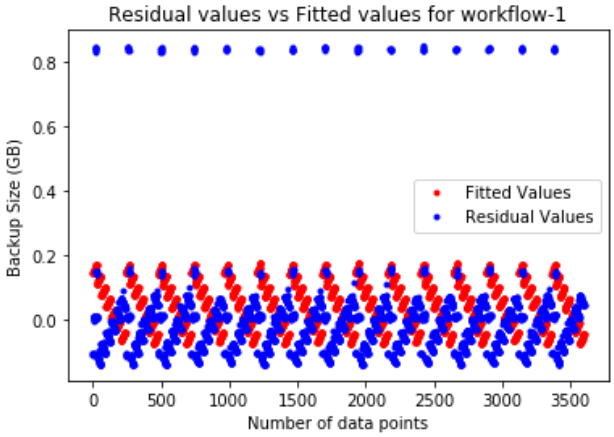
The plots for visualizing fitted and true values vs data points and residual and fitted values vs data points for different workflows are follows:

Workflow-0

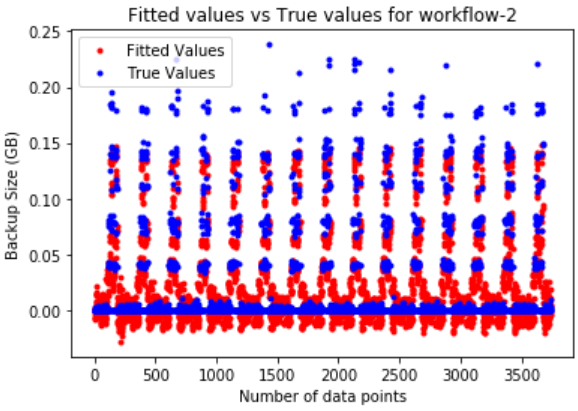


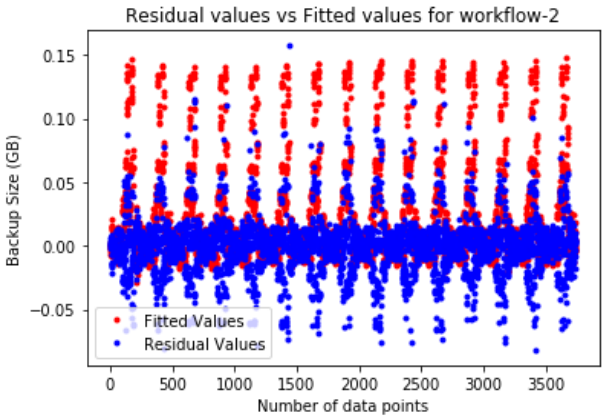
Workflow-1



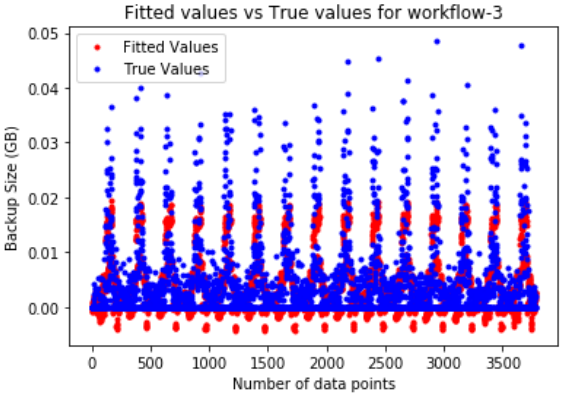


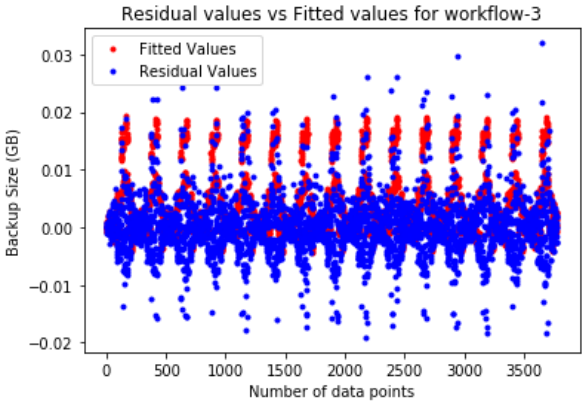
Workflow-2



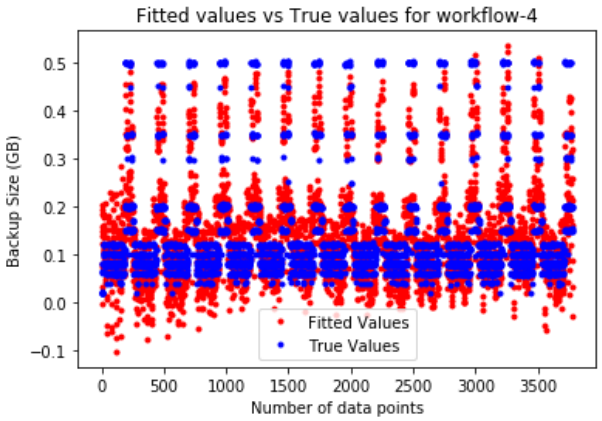


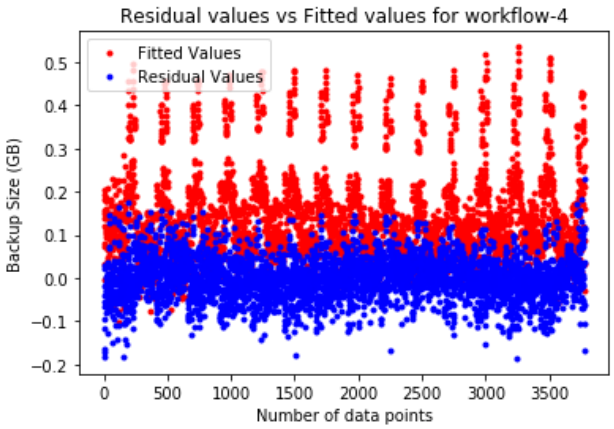
Workflow-3





Workflow-4





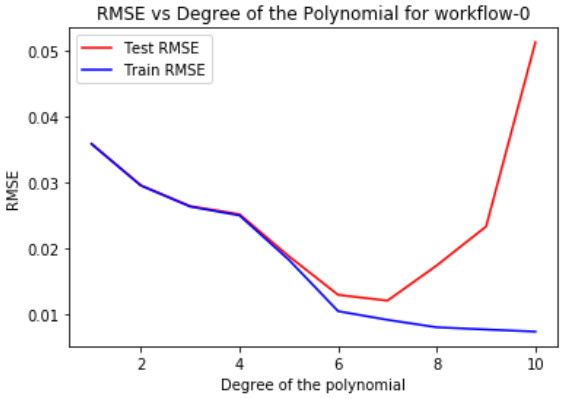
(ii) Polynomial Regression

In this task, we fit a more complex, polynomial regression model to the data and sweep through degree values [1, 10] to find the degree of the polynomial which gives the lowest RMSE values. Again, we use 10 fold cross validation and different data is used for different workflows. The following table shows the minimum RMSE values and the corresponding degrees.

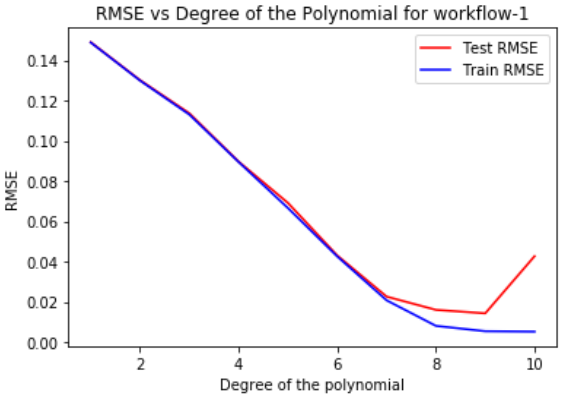
Workflow	Minimum Test RMSE	Degree	Minimum Train RMSE	Degree
Workflow-0	0.0119870872493733	7	0.00726983296872082	10
Workflow-1	0.0143085506470206	9	0.00523593121122613	10
Workflow-2	0.0255677547437117	6	0.01736336072272044	10
Workflow-3	0.0050906503275782	5	0.00425872511046220	8
Workflow-4	0.0487124376252589	5	0.01538087832264816	10

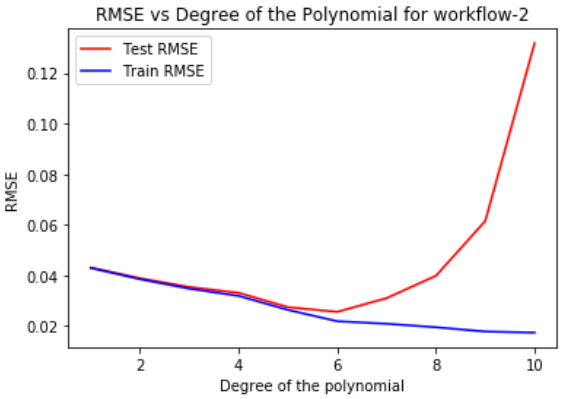
The plots for RMSE vs Degree of the polynomial for each workflow are as shown below:

Workflow-0



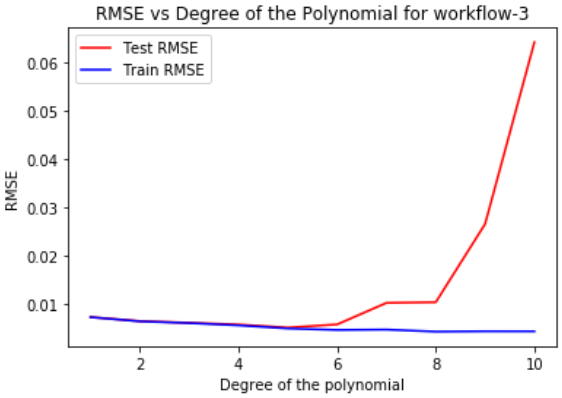
Workflow-1



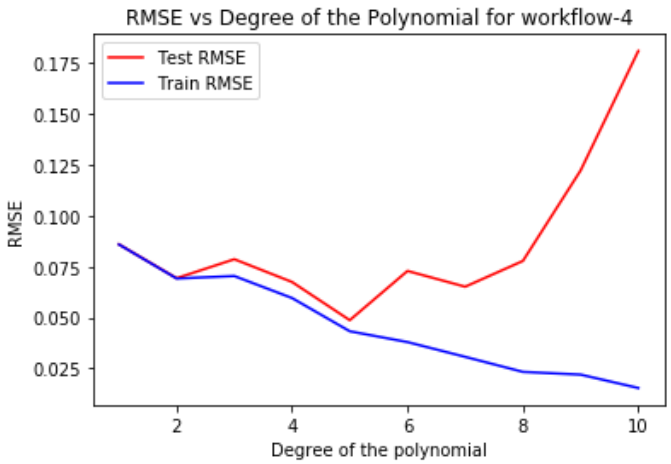


Workflow-2

Workflow-3



Workflow-4

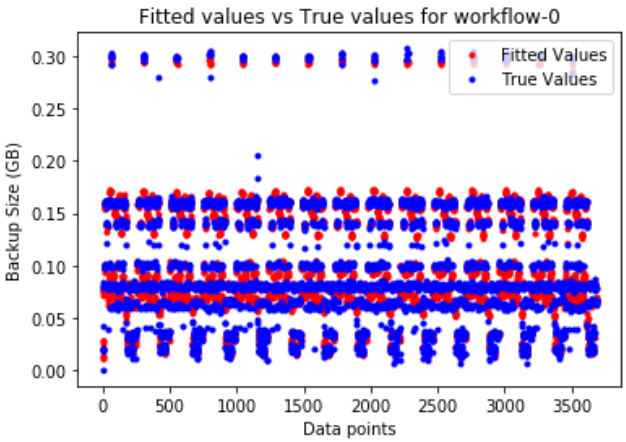


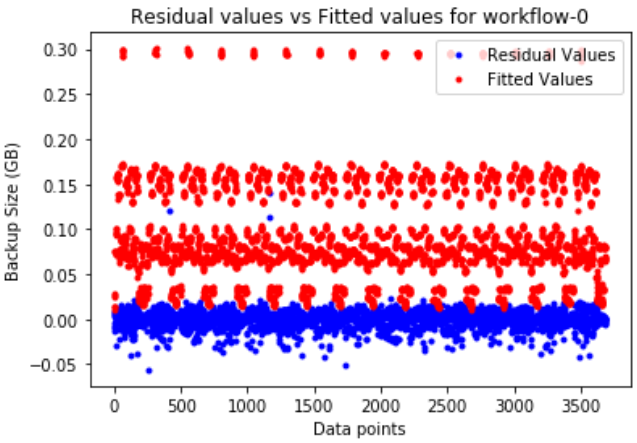
From the above table and plots, it can be observed that degree 7 works best for workflow-0, degree 9 works best for workflow-1, degree 6 for workflow-2 and degree 5 for workflow-3 and workflow-4. These are the threshold degrees for the fitted polynomial model, for each workflow, beyond which the generalization or test error of the model gets worse.

From the above results, it can be seen that different data sets generate different best-fit polynomial model. Cross validation lets the model traverse through the entire dataset while training the model over a particular dataset. A part of the training data itself is used for model estimation. This makes the model less prone to overfitting. Also, we calculate error for multiple folds of data, thus, cross validation provides an unbiased measure of errors. Moreover, cross validation can also be applied on several different methods of model training and the method with the least cross validation error can be chosen. This helps in reducing the model complexity and makes the model more general for prediction.

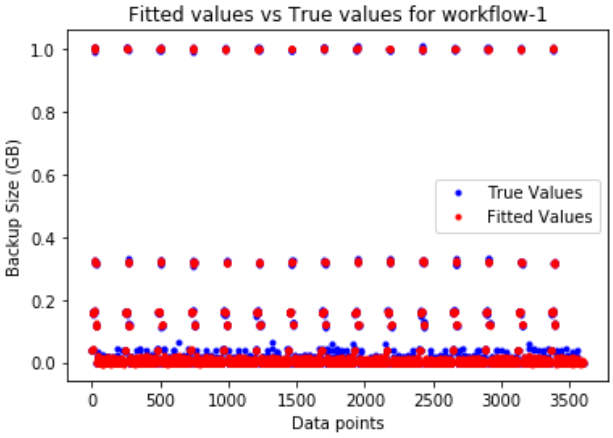
The visualization plots are as follows:

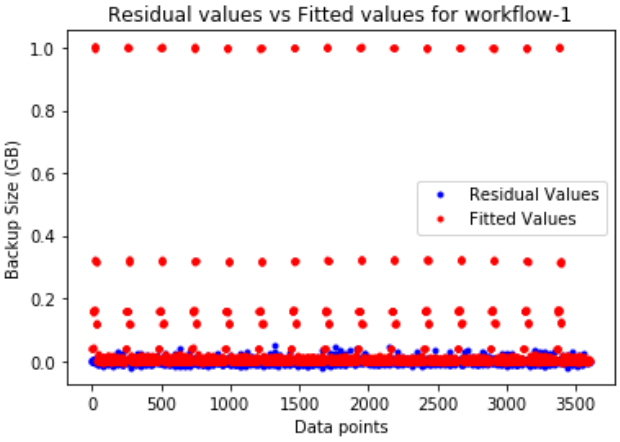
Workflow-0, Degree: 7



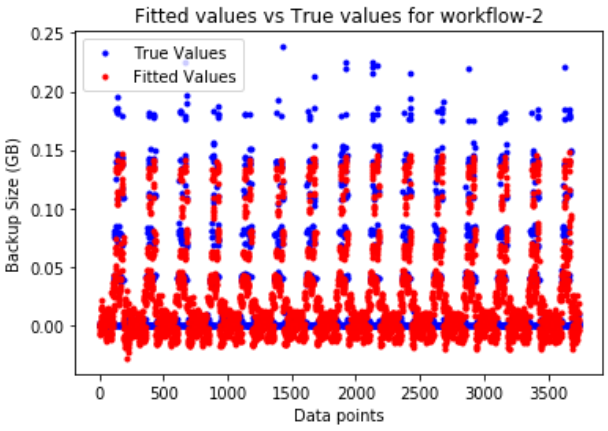


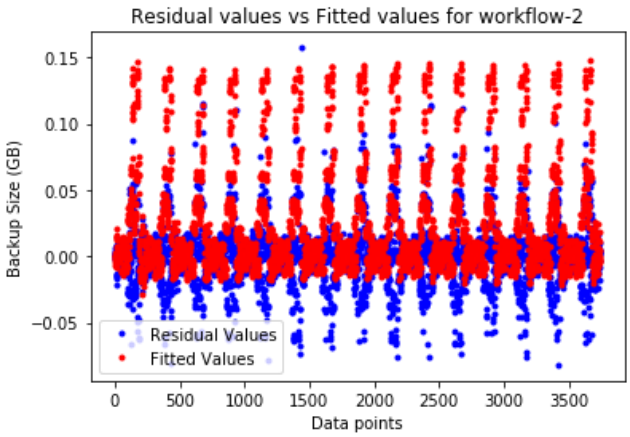
Workflow-1, Degree: 9



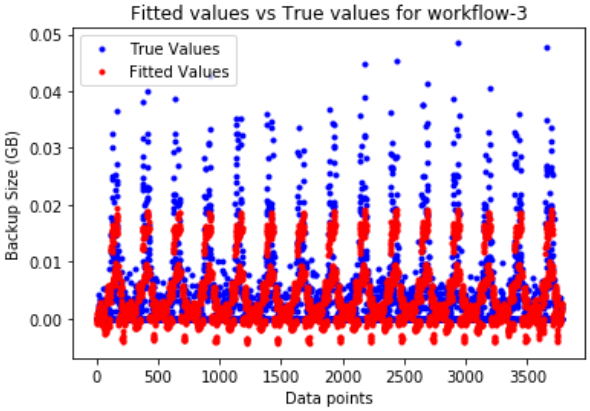


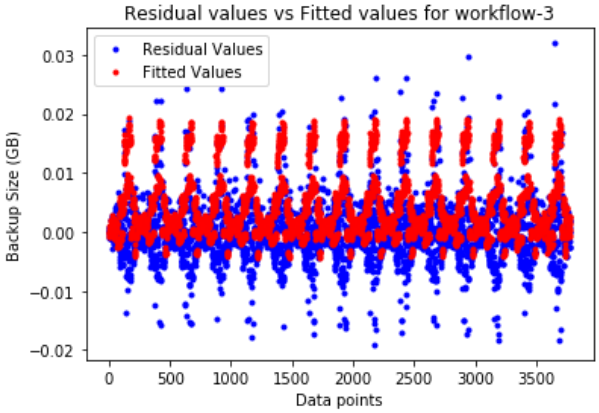
Workflow-2, Degree: 6



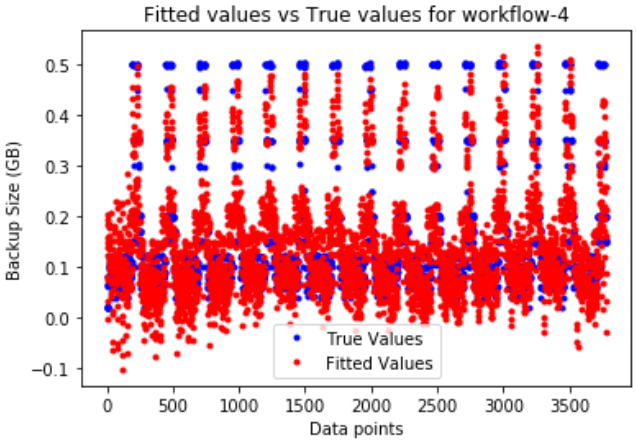


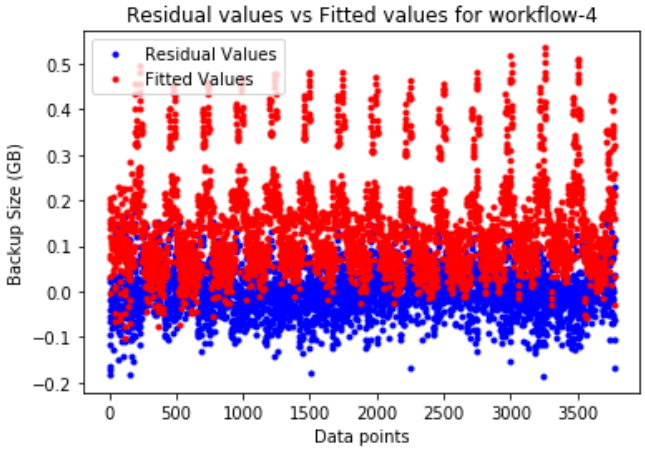
Workflow-3, Degree: 5





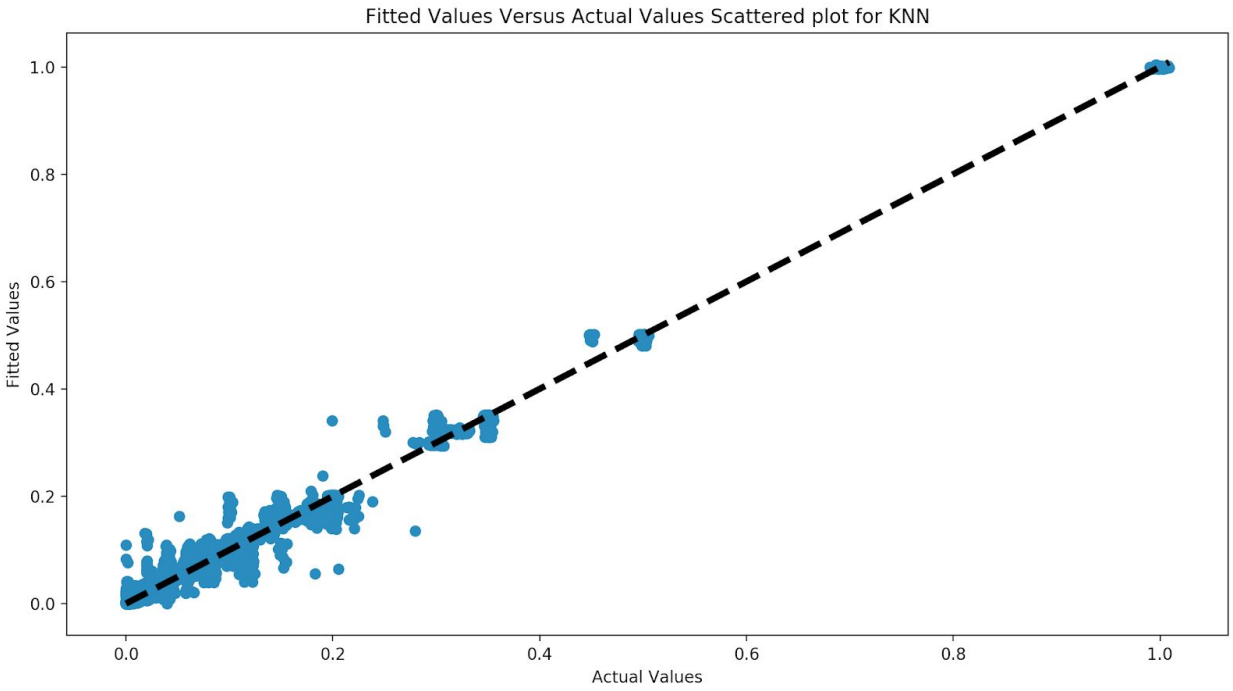
Workflow-4, Degree: 5

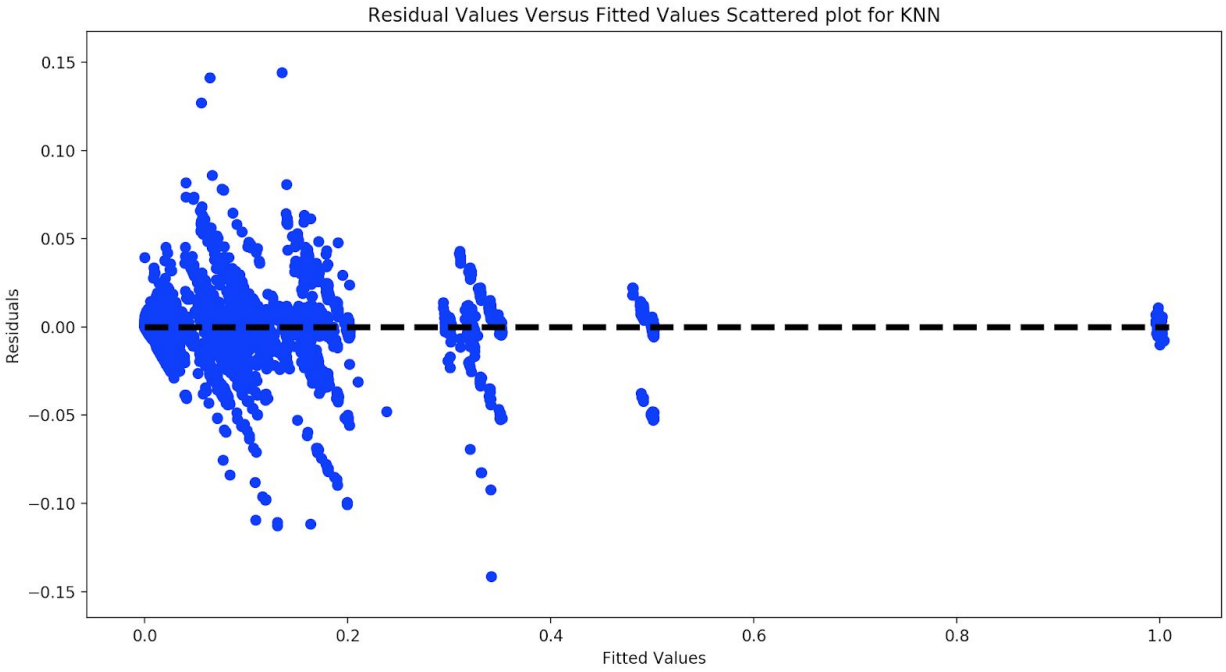


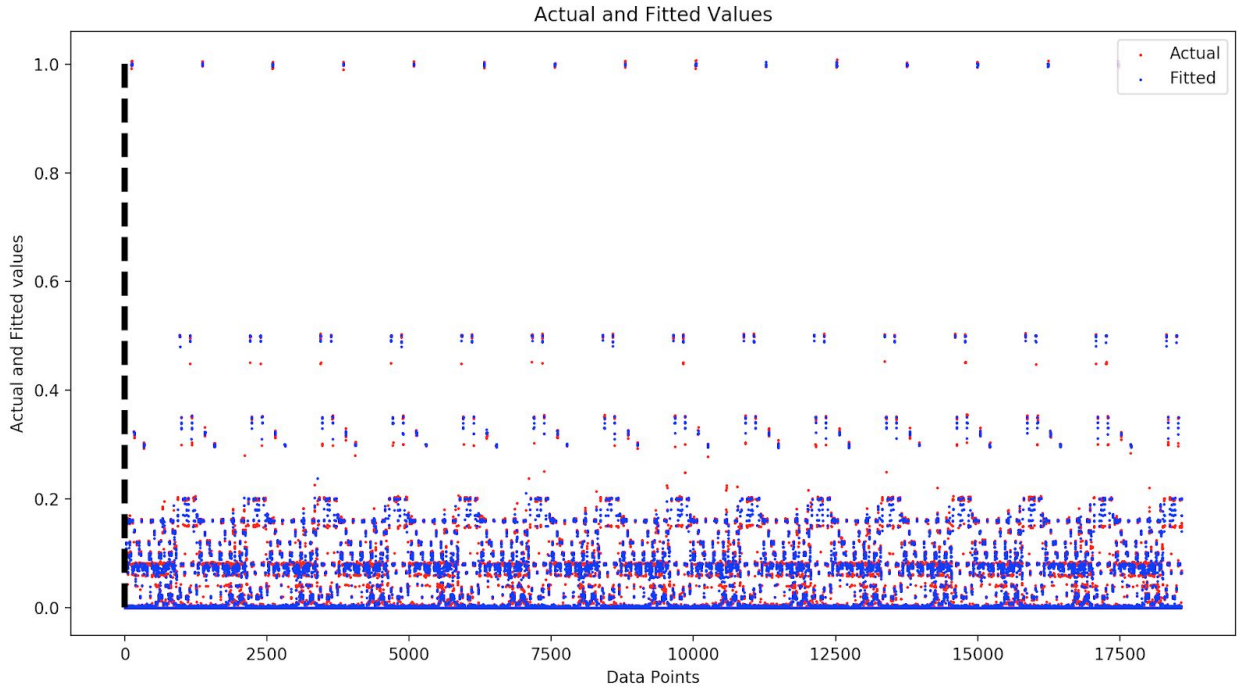


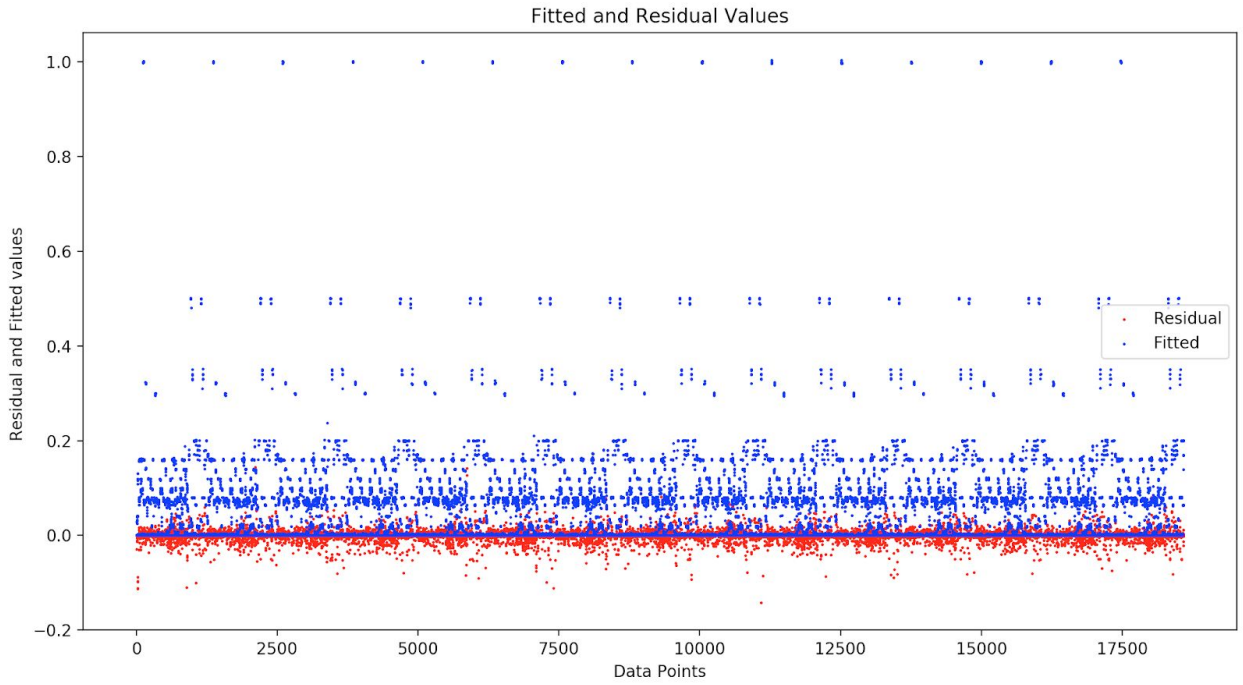
Q2(e): k-Nearest Neighbors Regression Model

Maximum accuracy is received with 5 neighbors and 30 leaves from the data collected above.
All the optimum KNeighborsRegressor parameters are :-
(n_neighbors=5,leaf_size=30,p=1,metric='minkowski',weights='distance')
Accuracy is :- 0.989221088037018
Test RMSE for this model is : 0.017161532652346243
Train RMSE for this model is: 0.010648429314147704
RMSE for this model is : 0.010414950644312217




Residual Values Versus Fitted Values Scattered plot for KNN
Residuals
Fitted Values
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
0.0
0.2
0.4
0.6
0.8
1.0


Actual and Fitted Values
Actual
Fitted
Actual and Fitted values
Data Points
1.0
0.8
0.6
0.4
0.2
0.0
0
2500
5000
7500
10000
12500
15000
17500



Q3: Comparison of all the regression models

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. A linear regression line has an equation of the form $Y = a + bX$, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept. The most common method for fitting a regression line is the method of least-squares. It cannot handle categorical data directly and we need to apply one hot encoding to use categorical data. It is good for handling sparse features.
Test RMSE = 0.10193944624209859

Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. Random decision forests correct for decision trees' habit of overfitting to their training set. The **random forest** model is a type of additive model that makes predictions by combining decisions from a sequence of base models. This broad technique of using multiple models to obtain better predictive performance is called model ensembling. In random forests, all the base models are constructed independently using a different subsample of the data. The random forest model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Tree based models are not designed to work with very sparse features.
Test RMSE = 0.012948024711

A **Neural Network** performing regression will have one output node, and that node will just multiply the sum of the previous layer's activations by 1.The result will be the network's estimate(y), the dependent variable that all your inputs(x) map to. To perform backpropagation and make the network learn, you simply compare y to the ground-truth value of y and adjust the weights and biases of the network until error is minimized. The goal is then to find the weights the provide the best fit to our training data. One way to measure our fit is to calculate the leasts squares error over our dataset. We use different activation functions such as relu, logistic and tan but we get the best results using tan activation function.Using neural networks for regression is overkill.
Test RMSE = 0.0283282997893980

K-Nearest Neighbors (KNN) is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure. KNN is a non-parametric method i.e. it does not assume an explicit form for $f(X)$, providing a more flexible approach. We start by assuming a value for the number of nearest neighbors K and a prediction point xo. Then KNN identifies the training observations No closest to the prediction point xo. KNN estimates f (xo) using the average of all the responses in No. The optimal value for K will depend on the bias-variance tradeoff. KNN works well with a small number of input variables, but struggles when the number of inputs is very large. KNN does not work well for categorical data.
Test RMSE= 0.017161532652346243

Thus it can be seen that the best performance is obtained by using Random Forest regressor.