FINAL PROJECT REPORT– HANH NGUYEN

BIOS 648 HIGH DIMENSION DATA ANALYSIS

1. Background

In this project, regression problem will be solved by using appropriate statistical models on HIV data. In this project, the outcomes, which are the log viral load and CD4 count, would be separately predicted based on the baseline data by the end of year 1 and by the end of year 2. In this project, Random forest and support vector machine (SVM), mixed regression partition tree with SVM and mixed Random forest-SVM will be implemented to predict the outcomes of interest. The prediction power of these methods will be compared based on their mean square error and R^2 in the training period and testing period.

2. Description of Data

The predictors used for prediction includes: log of viral load at time 0 (lrna0), CD4 count at time 0 (cd40), the drug used for the patient (called "arm" – which is a factor of 6 levels presenting 6 therapeutic drug combination), and the data for mutation status which is presented under a binary type of data (1 = mutated and 0 = wild-type) at the first 240 codons of the reverse transcriptase (RT) region and the first 99 codons of the protease (PR) region.

Each dependent variable is predicted based on 300 variables including lrna0, cd40, arm and the other 297 binary variables describing mutation status.

2.1 Distribution of patients under different schemes of drugs (arm)

The number of patients under each "arm" are quite uniform among 6 groups

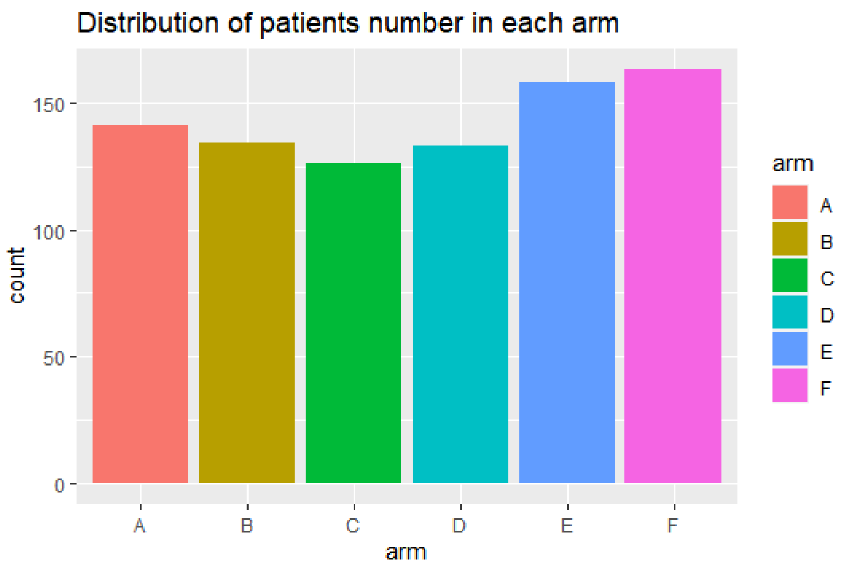


Figure 1 Arm distribution

2.2 Exploring log viral load and CD4 count distributions

The distributions of lrna1, lrna2, cd40, cd41 and cd42 are skewed to the right, especially the first two features (lrna1 and lrna2).

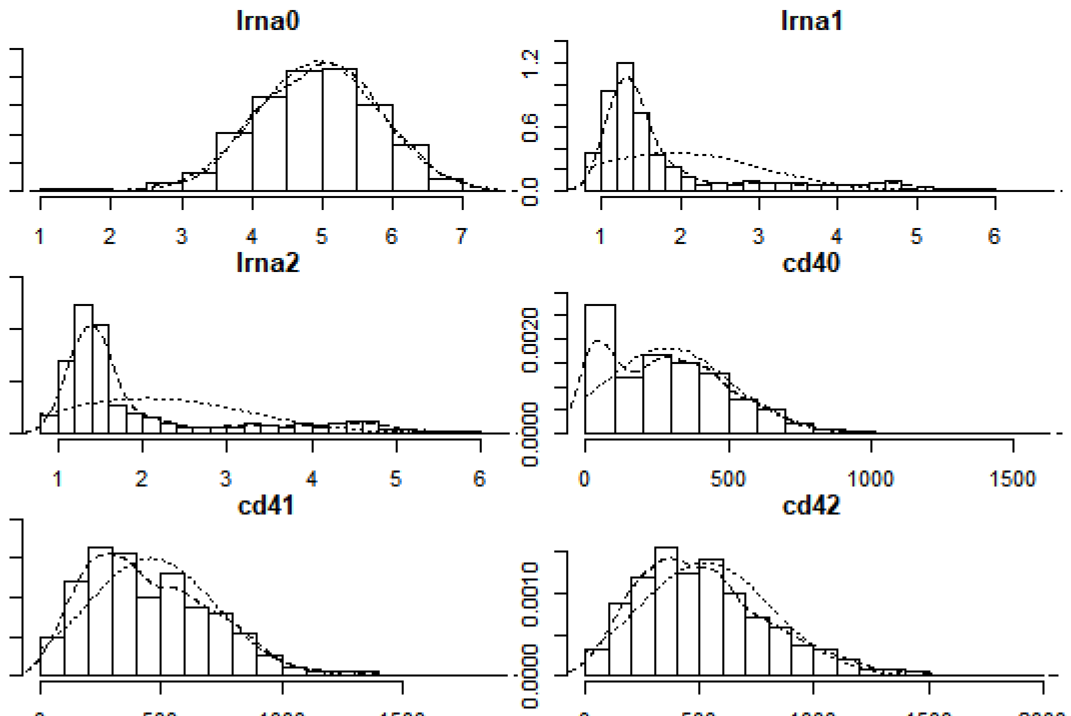


Figure 2 Distributions of lrna and cd4 at time 0, year 1 and year 2

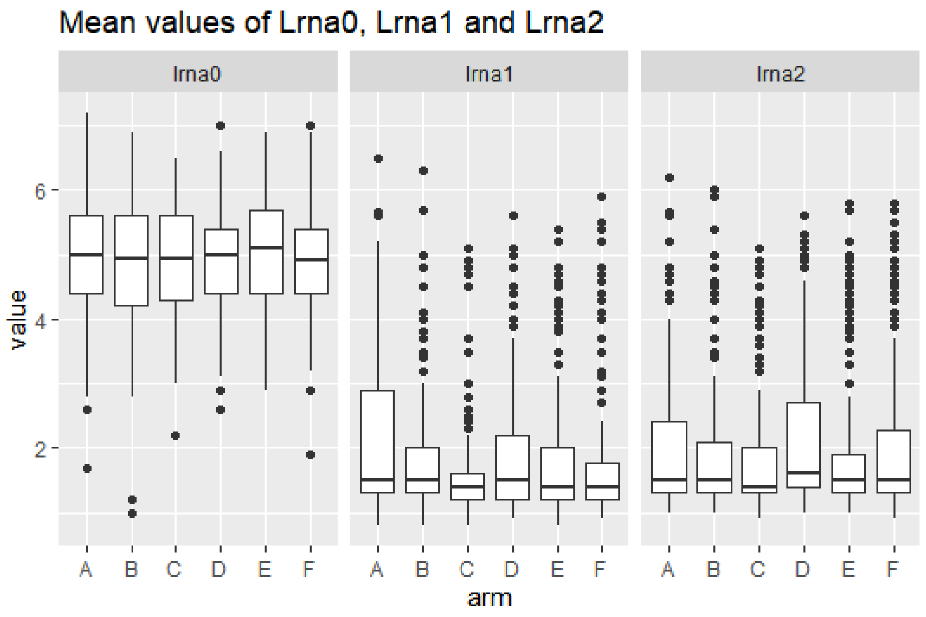


Figure 3 Mean values of Lrna0, Lrna1 and Lrna2

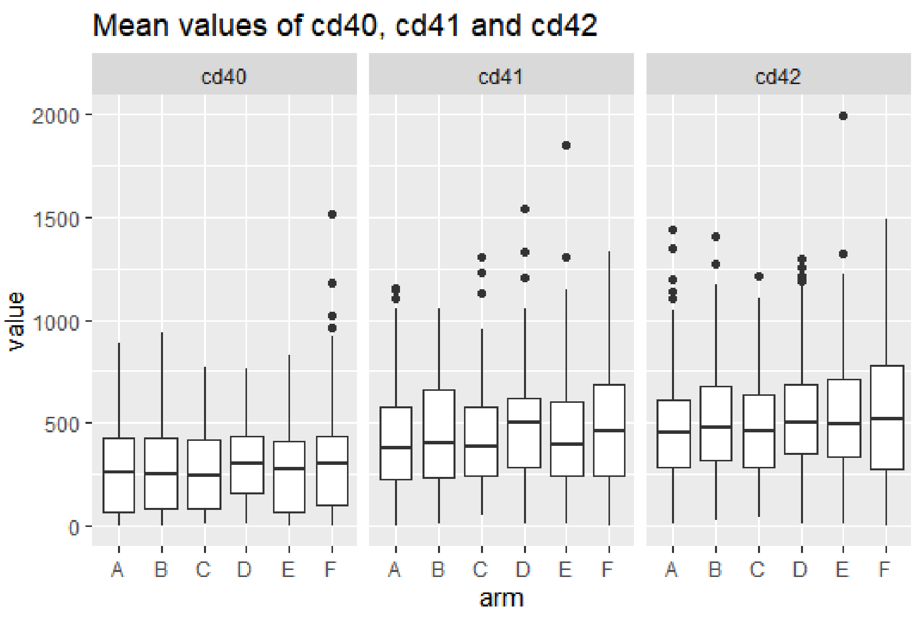


Figure 4 Mean values of cd40, cd41 and cd42

The mean values of lrna0, lrna1, lrna2 (log viral load at time 0, year 1, year 2) and cd40, cd41, cd42 (CD4 count at time 0, year 1 and year 2) are also quite similar among 6 arms

2.3 Missing values distribution

There are 47 missing values found in the total data (including dat384, rtmut0 and prmut0). There are 11 predictors containing missing values (or NA values) out of 300 variables, making up 47 missing values (0.01% of the total data). The distribution of the missing values shows no obvious systematic pattern, so it is unlikely to cause problem with deleting those missing value.

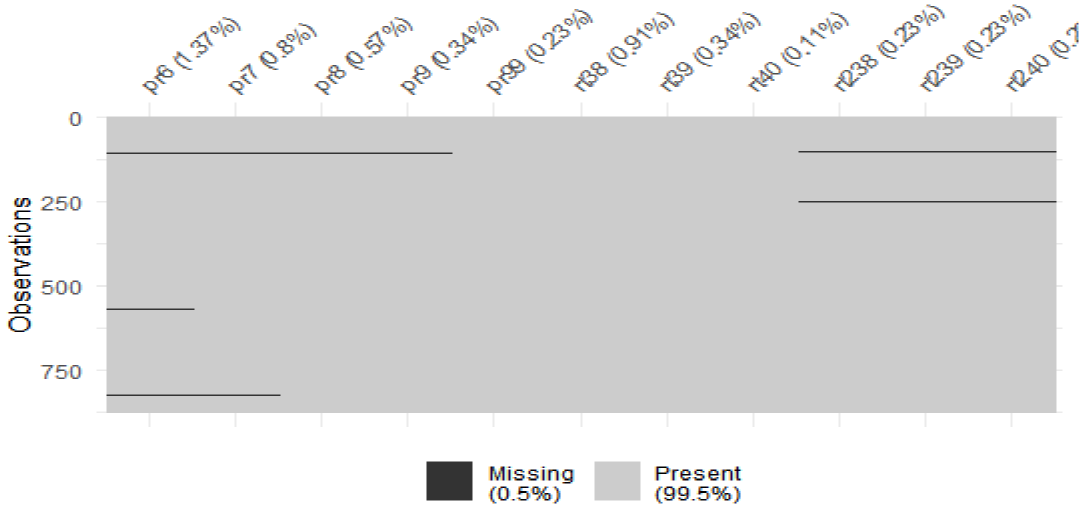


Figure 5 Missing value distribution

2.4 Zero-variance variables

There are 71 mutation positions of both the reverse transcriptase and the protease found to be constant in all patients (71 predictors have zero variance). These variables, along with patient identity (patid) are considered as no-informative variables, and they will be deleted out from the predictor matrix before fitting a model.

2.5 Correlation matrix

The matrix below shows the pairwise correlation between variables where absolute value of correlation is larger than 0.6.

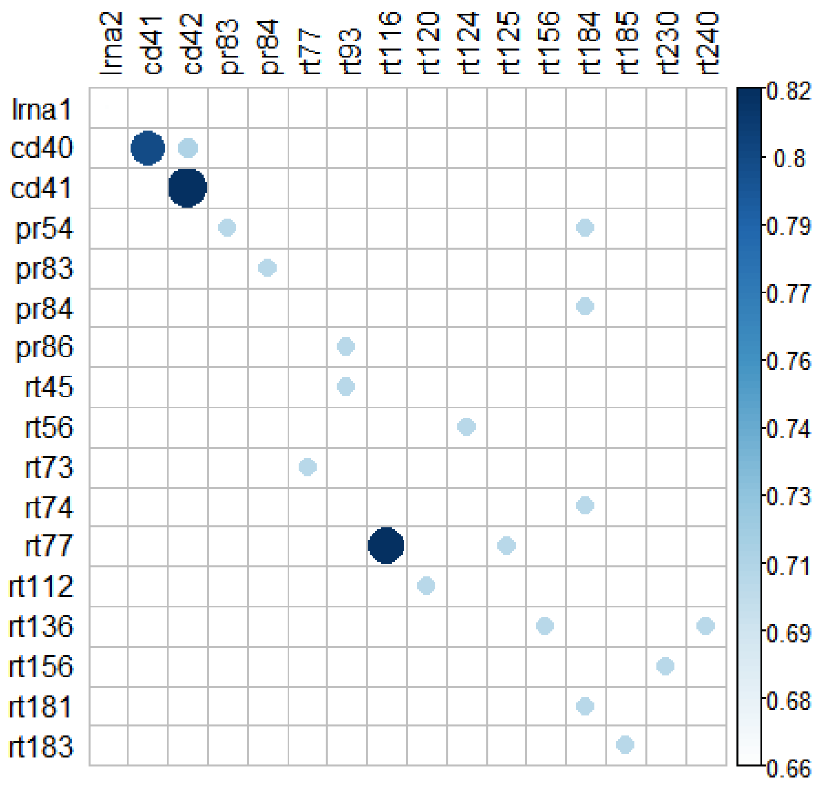


Figure 6 Pairwise correlation matrix (> 0.6 in absolute value)

3. Statistical Methods

There are three models to be implemented: random forest (RF), support vector machine (SVM). Regression partition tree associated with SVM (Rpart-SVM) and RF-SVM. For a fair comparison, all algorithms would be trained on the same train set and tested on the same test set.

Then the results will show the comparison of mean squared error, R squared of each method with or without using regression tree. All the algorithms uses the same data pre-processing method.

3.1 Pre-processing data

3.1.1 Introduction

The variables start with “pr” and “rt” show the mutation status (binary: 1 = mutated and 0 = wild type) in the corresponding codons of the reverse transcriptase (RT) region and the codons of protease (PR) region. Assembling large sets of codons represented in this way will result in sparse data, where majority of the set is zero values. This consumes an excessive amount of computer memory which inhibits the size of data sets that can be used when constructing predictive models. Particularly, when the algorithm randomly choose features for splitting the node, zero features may be selected , which may be one of the reasons for high error rate. For random forest, the number of features that are used to build individual classifier should be less so that uncorrelated trees are build.

For support vector machine, roughly speaking, the SVM maximizes the margin and thus relies on the concept of distance between different points. So, min-max or other scaling is highly recommended at the preprocessing step.

Table 1 Methods of preprocessing data in order

Methods	1. Omit NA	2. Normalizing lrna0-1-2	3. ZV and NZV	4. Center and Scale
RF	X	X*	X	X
SVM	X	X*	X	X
Rpart-SVM	X	X*	X	X
RF-SVM	X	X*	X	X

(*) results with and without normalizing lrna0-lrna1-lrna2

The detailed descriptions of the preprocessing methods are shown in order as the following.

3.1.2 Omitting NA values

As mentioned in the missing value distribution, all observation containing NA values will be omitted from the data. This method is applied for all algorithms (RF and SVM)

3.1.3 Normalizing skewed variables

As lrna1, and lrna2 have very skewed distributions to the right. Typically, random forest/ regression trees are robust to outliers and skewed distributions in explanatory variables. However, in dependent variables, it is worth checking out how normalizing these variables would affect the prediction error. Furthermore,

The method used to normalize lrna0, lrna1 and lrna2 is Yeo-Johnson. This transformation requires the data to be greater than zero, so it was done before center and scale variables.

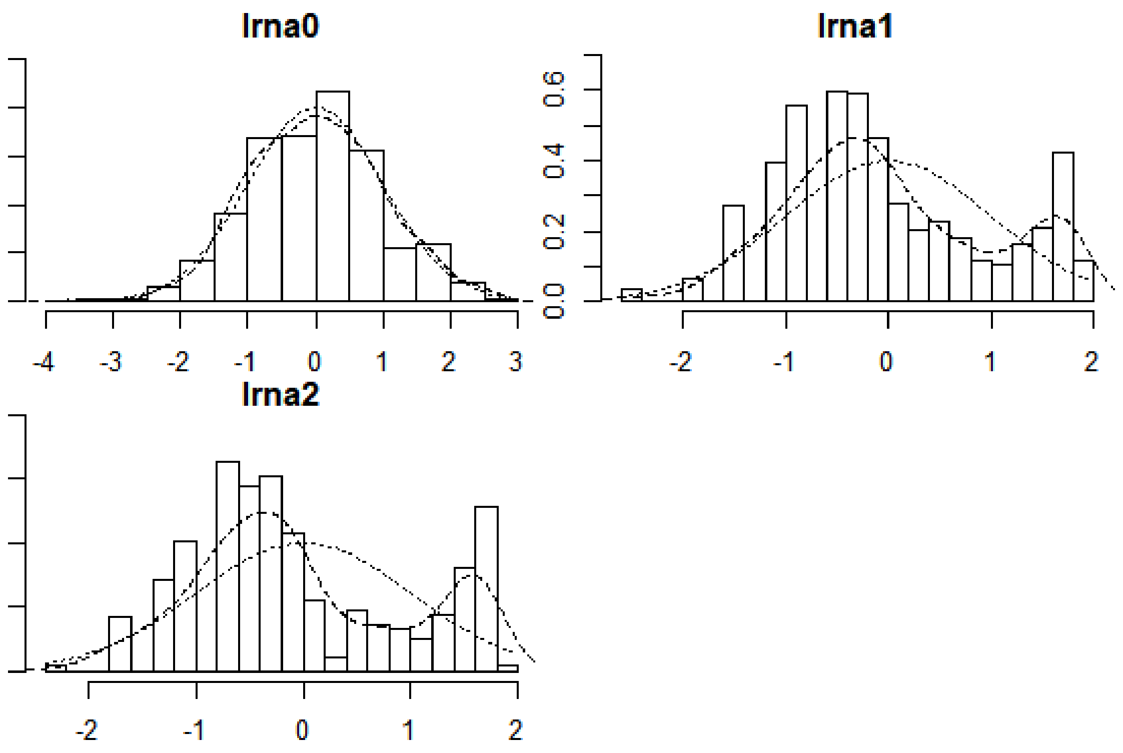


Figure 7 After transforming using Yeo-Johnson method

3.1.4 Removing zero variance (ZV) and near zero variance (NZV) variables

Using "caret" package in R, we can remove variables that contain only one unique value. Those are zero-variance variables.

We can also remove near zero variables, which "have both of the following characteristics: they have very few unique values relative to the number of samples and the ratio of the frequency of the most common value to the frequency of the second most common value is large.", as defined in R document of caret package.

Setting parameters:

- The frequency cutoff is 95/5 (default) which is defined as the ratio of the frequency of the most common values to the second most common value
- The cutoff for the percentage of distinct values out of the number of total samples is 10

This method is applied for all algorithms (RF and SVM).

3.1.5 Center and scale variables

Centering is done by subtracting the column means of x from their corresponding columns.

The value of scale determines how column scaling is performed (after centering). If scale is a numeric-alike vector with length equal to the number of columns of x, then each column of x is divided by the corresponding value from scale. If scale is TRUE then scaling is done by dividing the (centered) columns of x by their standard deviations if center is TRUE, and the root mean square otherwise. If scale is FALSE, no scaling is done.

3.1.6 Correlated variables

After all previous steps of pre-processing data, the correlation matrix shows that no pair of mutation variables is correlated.

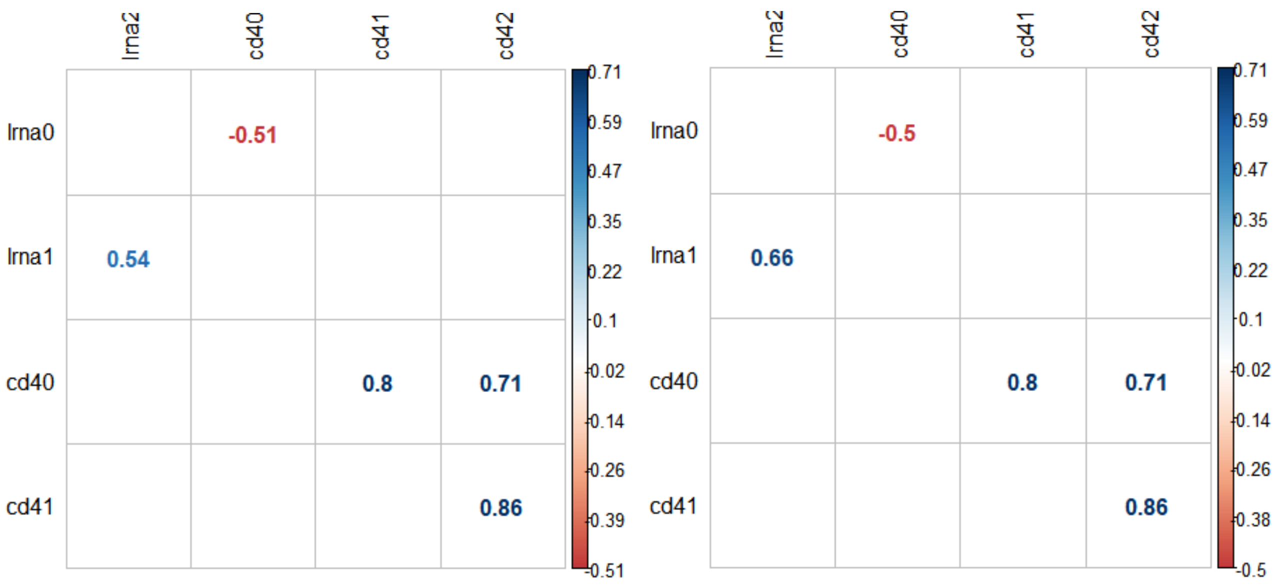


Figure 8 (Left) Without normalizing lrna and (Right) With normalizing lrna

3.2 Splitting train set and test set

Train set contain 75% observation of the whole data. The dividing criterion for splitting is to stratify "arm" in both train set and test set to ensure the distribution of 6 levels of arm in both train and test set similar to that in the data.

3.3 Fitting models

3.3.1 Random forest

Package used in R: ranger

Implementing grid search to find the sets of parameters with the best performances:

- Mtry = (1, 3, 5, 7), node size = (2, 4, 6, 8, 10), number of trees = (300,400,500,600,700,800)
- Number of cross-valiadtion folds: 5

Table 2 Chosen tuning parameters after grid search

Dependent variable	mtry	Node size	numtree
Scenario 1: with normalizing lrna			
Lrna1	1	8	400
Lrna2	1	6	700
Cd41	7	2	300
Cd42	7	8	400
Scenario 2: without normalizing lrna			

Lrna1	1	10	500
Lrna2	1	2	800
Cd41	7	2	300
Cd42	7	8	400

3.3.2 Support vector machine

Implementing grid search to find the sets of parameters with the best performances:

- Gamma = (2,1,0.1); Degree = (2,3,4,5,6); Cost = (0.5,0.1,0.01,0.001,0.0001)
- Number of cross-validation folds: 5

Table 3 Chosen tuning parameters after grid search

Dependent variable	Gamma	Degree	Cost
Scenario 1: with normalizing lrna			
Lrna1	0.1	5	0.01
Lrna2	0.1	6	0.01
Cd41	0.1	3	0.1
Cd42	1	3	0.0001
Scenario 2: without normalizing lrna			
Lrna1	0.1	5	0.01
Lrna2	0.1	5	0.01
Cd41	0.1	3	0.1
Cd42	1	3	0.0001

3.3.3 Rpart-SVM

The Rpart-SVM method handles the regression job with two main steps. In the first step, regression partition tree would be implemented using the whole train set, then pruned back the tree to avoid overfitting. In the second step, each data set (cluster) at each terminal node (leaf) will be used to fit polynomial SVM algorithm. New individual newX will be predicted by Rpart-SVM by pushing newX down the tree to a terminal node i, then use the SVM model for that node i to predict the dependent value. The following diagram depicts the Rpart-SVM work-flow.

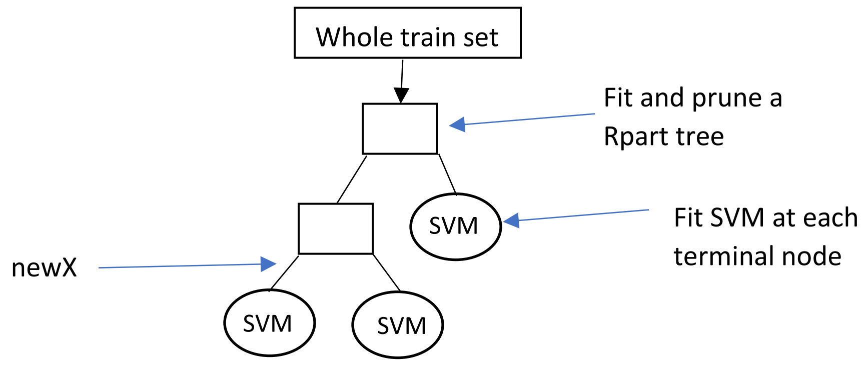


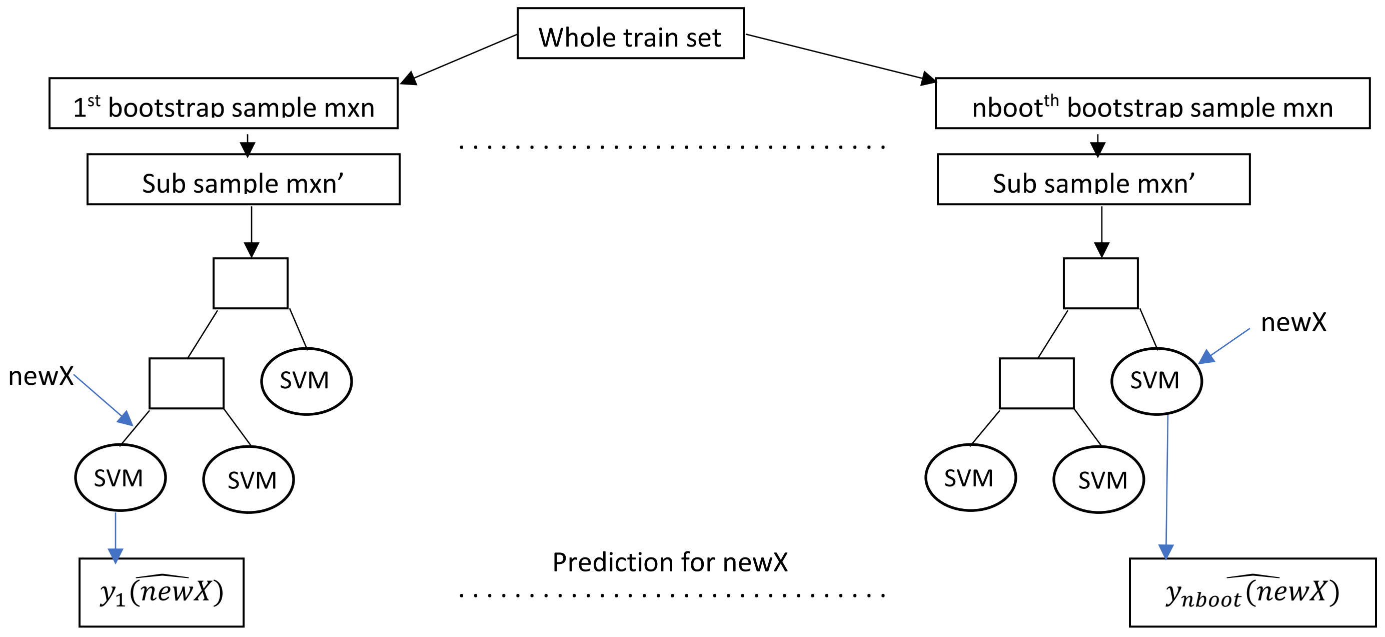
Figure 9 Diagram showing algorithm Rpart-SVM

Parameters:

- Parameters for rpart: minsplit=200, xval=10, cp=0.00001
- Parameters for SVM grid search:

3.3.4 RF-SVM

The method RF-SVM handles the regression tasks in a quite similar manner to the Rpart-SVM, except that it uses bootstrapped samples for training. First, nboot bootstrapped samples of m x n size will be generated from the m x n train set. Following that, from the ith bootstrap sample, a new sample of m x n' is created by randomly choosing a n' subset of variable numbers. In the next step, the regression tree is grown by using the newly created sample of m x n', followed by fitting SVM at each terminal node of the tree. Mean square error (MSE) of each tree is the mean of sum of squares of difference between dependent variables of the sample m x n' and the predicted values from all leaves of the tree. The overall train mean square error is computed by taking the mean of nboot mean square error of those trees. For prediction task, all individuals of the test set will be pushed down ith trees to obtain an ith vector of predicted values, then taking the mean of all nboot predicted vectors to obtain the overall test predicted vector. The diagram below show the working pipeline for RF-SVM method.



$$Result\ y(\widehat{newX}) = \frac{1}{nboot}(y_1(\widehat{newX}) + \ldots + y_{nboot}(\widehat{newX}))$$

Figure 10 DIagram of RF-SVM workflow

Parameters:

- Number of bootstrap samples: nboot = 100
- Number of variables in subset samples n' = subn (=35 for lrna1, =30 for lrna2, =50 for cd41, =50 for cd42)
- Parameters for rpart: minsplit=200, xval=10, cp=0.00001
- Parameters for SVM grid search

4. Results

4.1 Prediction of lrna1

For a fair comparison, all algorithms in each scenario are implemented on the same train set and test set.

Table 4 Results of machine learning prediction for lrna1

Assessment elements	Random forest	SVM	Rpart-SVM	RF-SVM
Scenario 1: With normalizing lrna				
Train error	0.8166068	0.9363951	0.2941733	1.786485
95% CI of train error	0.6865857 0.9466278	0.8374859 1.0353043	0.2305353 0.3578114	Not computed
Test error	1.0434586	1.1086564	1.0441924	**1.0310425**
95% CI of test error	0.8727424 1.2141747	0.9099066 1.3074061	0.8659616 1.2224231	0.8607719 1.2013130
R squared	-0.003752057	-0.0664689	-0.004457948	0.008191576
Correlation between $\hat{y}$ and y	0.1056126	0.1230489	0.1761532	0.1922918
Scenario 2: Without normalizing lrna				
Train error	0.8013674	1.0455621	0.039827635	1.560092
95% CI of train error	0.5734200 1.0293148	0.8441812 1.2469429	0.009798019 0.069857250	Not computed
Test error	1.168299	1.4009150	1.1930941	1.1730571
95% CI of test error	0.832139 1.504459	0.9678177 1.8340122	0.8614556 1.5247326	0.8154654 1.5306488
R squared	-0.01025565	-0.2114042	-0.03169664	-0.01437015
Correlation between $\hat{y}$ and y	0.0467153	-0.001506511	0.104464	0.1629164

In scenario of normalizing lrna, RF-SVM shows the best prediction result with the smallest MSE, followed by RF. Without normalizing lrna, all MSE of four algorithms increases by more than 10%.

However, no algorithm generates predicted values which have linear relationship to actual test values. As can be seen in the plot below. As we expect to see the points are located along the solid line, we have very scattered points instead.

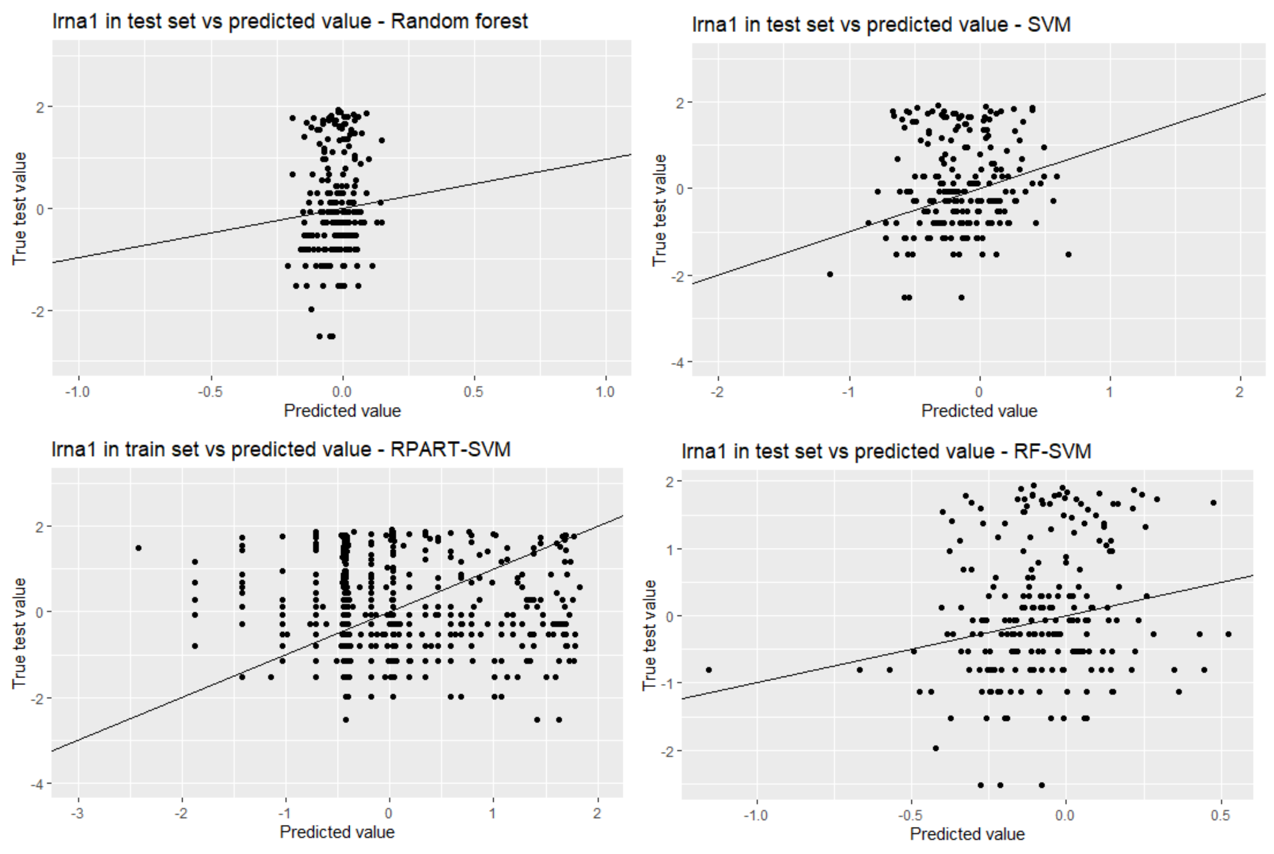


Figure 11 (With normalizing lrna) Scatter plot of actual test value vs. predicted values of lrna1. The solid line has zero intercept and slope = 1 showing relationship between actual and predicted values.

4.2 Prediction of lrna2

Table 5 Results of machine learning prediction for lrna2

Assessment elements	Random forest	SVM	Rpart-SVM	RF-SVM
Scenario 1: With normalizing lrna				
Train error	0.8073620	0.9502335	0.07347604	1.342097
95% CI of train error	0.6885111 0.9262128	0.8512472 1.0492198	0.04902391 0.09792818	Not computed
Test error	1.0685338	1.232343	1.1060359	1.0810834
95% CI of test error	0.9067559 1.2303117	1.034767 1.429920	0.9348668 1.2772050	0.9130087 1.2491582
R squared	-0.02581952	-0.1830808	-0.06182245	-0.03786748
Correlation between $\hat{y}$ and y	0.06512775	0.04175346	0.0413584	0.08957727
Scenario 2: Without normalizing lrna				
Train error	0.7847865	1.1013743	0.11073697	1.490711

95% CI of train error	0.5848335 0.9847394	0.9095464 1.2932023	0.05444087 0.16703307	Not computed
Test error	**1.0600016**	1.322575	1.1423722	1.0896342
95% CI of test error	0.8152417 1.3047615	0.982927 1.662223	0.8566347 1.4281097	0.8321704 1.3470979
R squared	-0.01291523	-0.2638249	-0.0916268	-0.04123147
Correlation between $\hat{y}$ and y	0.06663725	0.07220619	-0.01614301	0.04689836

With normalizing lrna2 scenario, RF is the best algorithm, followed by RF-SVM. Since RF-SVM consumes a long processing time, it gives rise to difficulty in tuning parameters, especially n' (the number of variables for subset samples from bootstrap samples). In future work, we can fine-tune this parameter to obtain better result. RF remains to be the best efficient algorithm in regression task of lrna2 when normalizing it. It works even slightly better when normalizing lrna1.

However, similar to lrna1, no algorithm generates predicted values which have linear relationship to actual test values. Like lrna1, we obtain very scattered points.

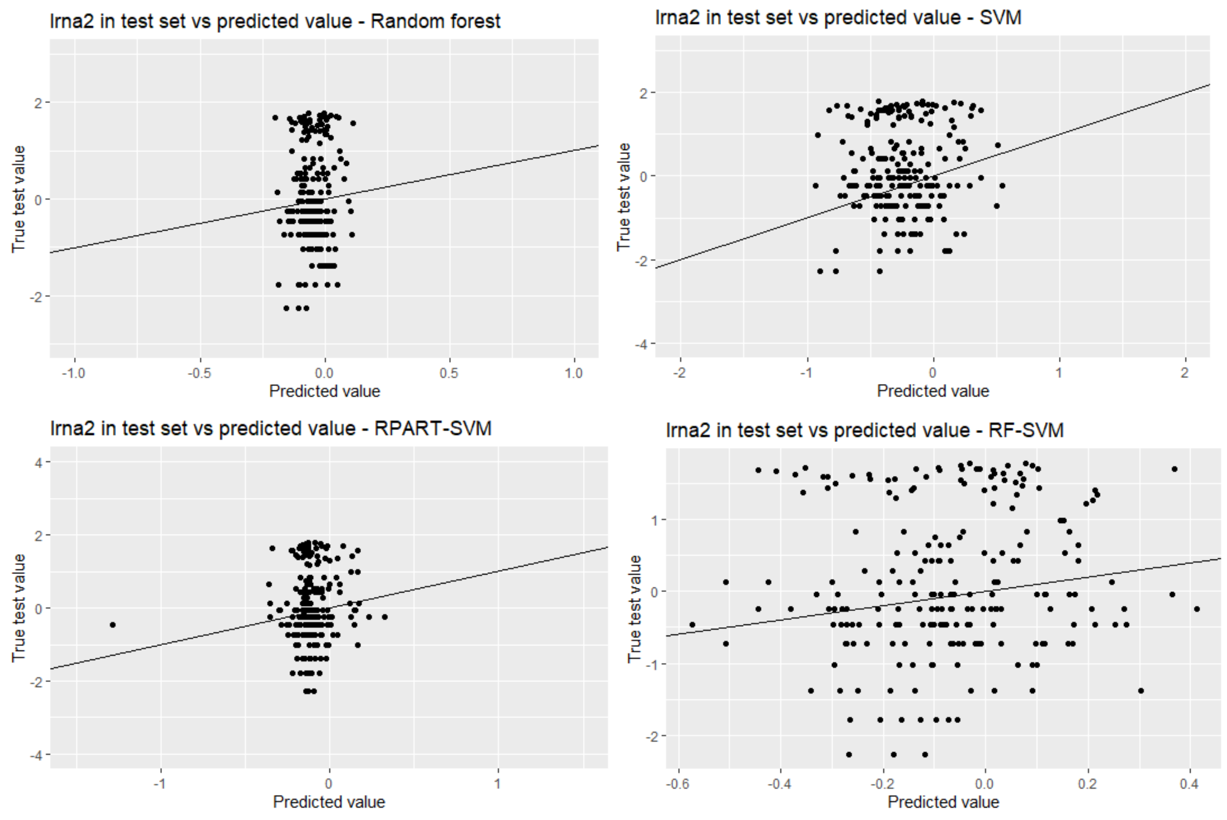


Figure 12 (With normalizing lrna) Scatter plot of actual test value vs. predicted values of lrna2. The solid line has zero intercept and slope = 1 showing relationship between actual and predicted values.

4.3 Prediction of cd41

Table 6 Results of machine learning prediction for cd41

Assessment elements	Random forest	SVM	Rpart-SVM	RF-SVM
Scenario 1: With normalizing lrna				
Train error	0.07723582	0.3303051	0.10785463	1.925316
95% CI of train error	0.05454130 0.09993034	0.2734036 0.3872065	0.07401139 0.14169788	Not computed
Test error	0.4622826	0.3941364	0.3907150	**0.3578774**
95% CI of test error	0.3335215 0.5910437	0.2643469 0.5239259	0.2813045 0.5001255	0.2478698 0.4678850
R squared	0.5191256	0.5900125	0.5935715	0.6277297
Correlation between $\hat{y}$ and y	0.7452225	0.77355	0.774193	0.7935173
Scenario 2: Without normalizing lrna				
Train error	0.07728518	0.3305998	0.10796034	1.925321
95% CI of train error	0.05459292 0.09997744	0.2737272 0.3874725	0.07407921 0.14184148	Not computed
Test error	0.4620930	0.3929662	0.3910254	0.3578934
95% CI of test error	0.3332793 0.5909068	0.2645473 0.5213850	0.2817039 0.5003468	0.2478960 0.4678907
R squared	0.5193228	0.5912298	0.5932486	0.6277131
Correlation between $\hat{y}$ and y	0.7454715	0.7740793	0.7740633	0.793538

All SVM-related algorithms outperform the regression task as compared with RF. The best one is RF-SVM. In future work, we can fine-tune this parameter to obtain better result.

Unlike lrna1, in scenario 2 that not normalizing lrna, the prediction performance of RF and SVM seems not significantly affected. RF and SVM even work slightly better without normalizing, though the improvement is very subtle. In both scenarios, RF-SVM appears to be the best algorithm, and even slightly better with lrna transformation.

The regression prediction for cd41 is significant efficient than predicting lrna1 and lrna2. As can be seen in the plot, the points are located along the solid line showing predicted and actual values

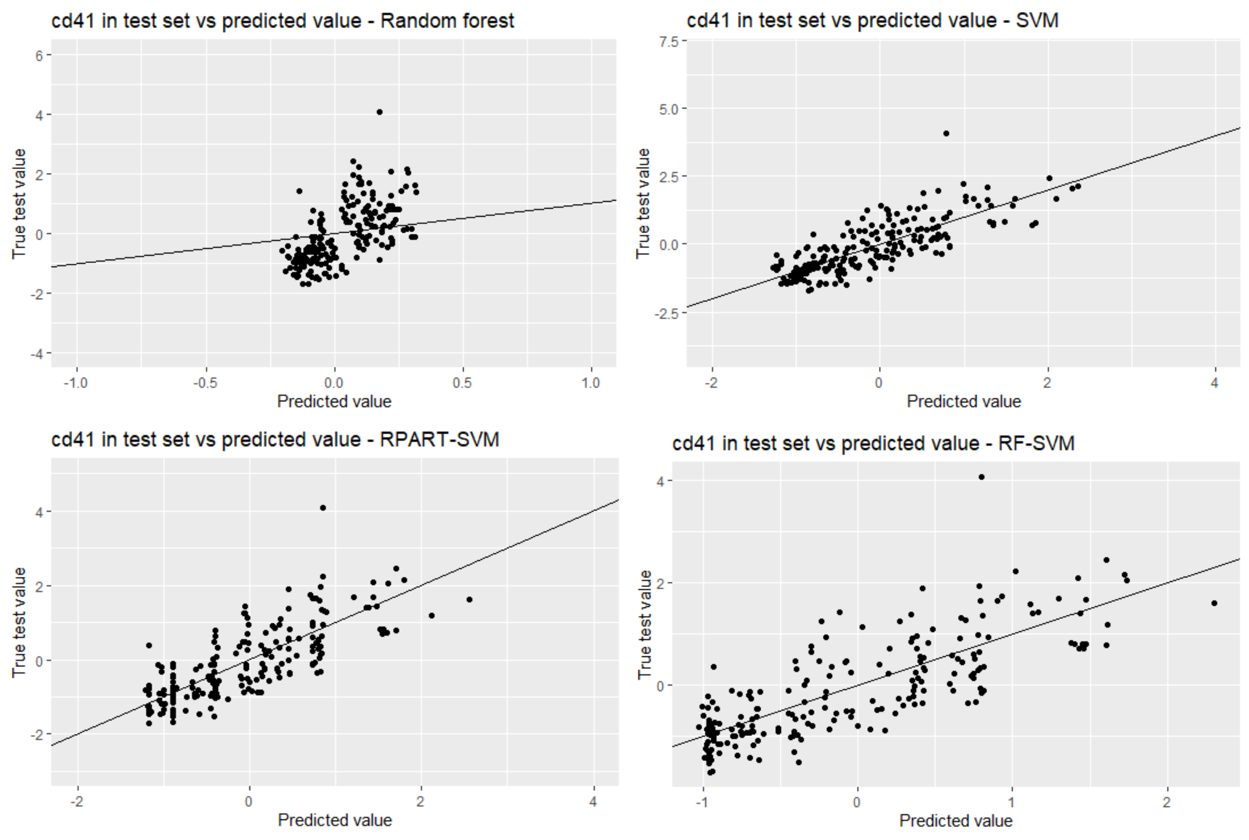


Figure 13 (With normalizing lrna) Scatter plot of actual test value vs. predicted values of cd41. The solid line has zero intercept and slope = 1 showing relationship between actual and predicted values.

4.4 Prediction of cd42

Table 7 Results of machine learning prediction for cd42

Assessment elements	Random forest	SVM	Rpart-SVM	RF-SVM
Scenario 1: With normalizing lrna				
Train error	0.1984074	0.9635539	0.1888904	1.926338
95% CI of train error	0.1402103 0.2566045	0.8308614 1.0962464	0.1534690 0.2243118	Not computed
Test error	0.5456170	0.8608483	0.5142835	**0.4735745**
95% CI of test error	0.4076962 0.6835379	0.6649768 1.0567199	0.3736248 0.6549423	0.3483682 0.5987809
R squared	0.4110808	0.07083165	0.444901	0.4888409
Correlation between $\hat{y}$ and y	0.6598071	0.5882513	0.677784	0.7015884
Scenario 2: Without normalizing lrna				
Train error	0.1983728	0.9637777	0.1890803	1.926921
95% CI of train error	0.1401784 0.2565672	0.8310954 1.0964600	0.1536559 0.2245047	Not computed
Test error	0.5454629	0.8609834	0.5143287	0.4740636

95% CI of test error	0.4075271 0.6833988	0.6652583 1.0567084	0.3736884 0.6549690	0.3490716 0.5990556
R squared	0.4112471	0.07068585	0.4448523	0.7013223
Correlation between $\hat{y}$ and y	0.6598839	0.5892669	0.6780574	0.488313

RF-SVM outperforms other three algorithms and with normalizing lrna (skewed data).

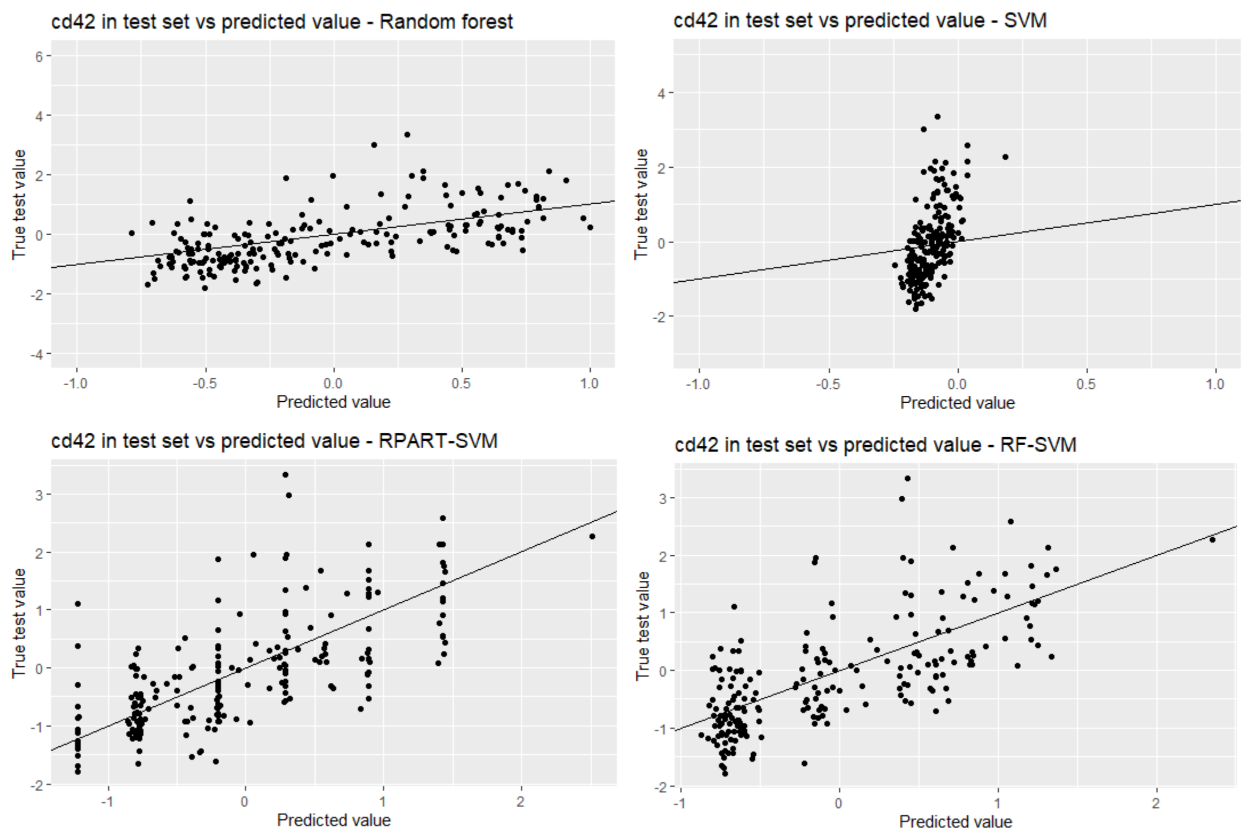


Figure 14 (With normalizing lrna) Scatter plot of actual test value vs. predicted values of cd42. The solid line has zero intercept and slope = 1 showing relationship between actual and predicted values.

5. Conclusion

For lnra1 prediction, it is better to normalizing this dependent variable before implementing regression task and the best algorithm is RF-SVM. Unlike lrna1, it is not necessary to normalizing lrna2 in fitting RF algorithm. In this case, RF is the best algorithm.

Except predicting cd41, RF outperform SVM in predicting cd42, lrna1 and lrna2. RF-SVM association makes a significant improvement in performance of SVM algorithm and outperforms other three algorithms in most cases. The only disadvantage of this algorithm is its time-consuming property.