Statistical Learning Assignment

Bella Shao

December 17, 2021

Abstract

In this report, we apply three different methods, generalized additive model, ensemble trees and support vector machines, to predict depression severity *dep_sev_fu* and compare their prediction errors for our specific data set.

1 Data exploration

At the first look of the data set, we have 20 potential predictors. The first step is to identify which predictors are highly related to the response variable *dep_sev_fu*. By plotting the response variable to all the 20 candidate predictors respectively, we can have a basic and direct impression of how each potential predictors are related to the response variable. The plot 1 shows that some predictors such as *disType* and *IDS* clearly have relationships with response variable, other predictors might also contribute to response variable.

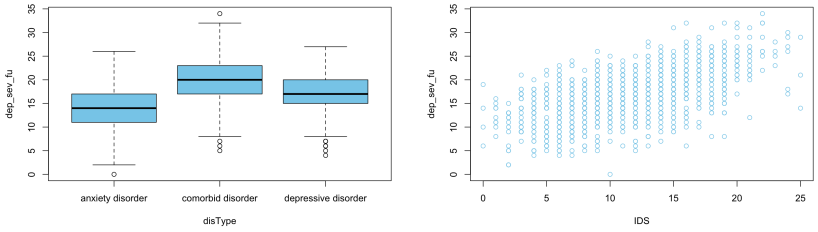

Figure 1: Distributions of response variable to predictors

Furthermore, as to explore which candidate predictors are related to response variable, we also run subset selection regression, the resulted Mallow's Cp and BIC plots are shown in figure 2, where approximately 8 to 11 predictors might be suitable for our model. To this end, in the following sections, we will figure out which predictors are the most important to the response variable *dep_sev_fu*.

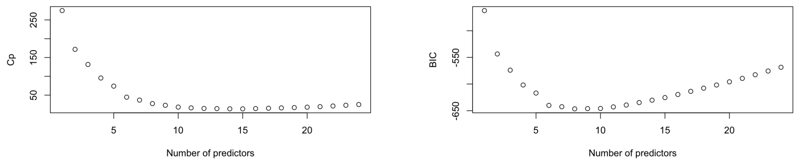

Figure 2: Mallow's Cp and BIC

2 Comparisons with GAM, Ensemble trees and SVM

In order to explore the relationships between *dep_sev_fu* and candidate predictors and which predictors are more important to response variable, we will use three methods. The first method is Generalized Additive model with smoothing splines for related predictors. The reasons are as follows:

- We have multiple predictors in the data
- Some predictors might have nonlinear relationship with response variable
- GAM is additive model, and the data set has numeric and factor predictors
- Smoothing spline is a good method to penalize over-fitting of data points

Even though GAM is not a suitable method to perform predictor selection, we can still inspect the significance for each candidate predictor and only keep those statistical significant ones.

The second method we are going to experiment with is Ensemble trees, which includes random forest and boosting method. We choose this method is because:

- Since we have many candidate predictors, ensemble method is good for feature selection
- The variance can decrease substantially in the prediction without increasing much bias and the bias can remain relatively stable because of Out-Of-Bag property
- Avoid overfitting
- Boosting ensemble can sequentially add predictors in the previous residual errors and arrive at a correct predictor and the error can be reduced significantly

Besides the advantages of random forest and boosting, we will compare the two methods and to check which method is more suitable for out data predictions.

At the final experiment, we will implement support vector machines method to predict *dep_sev_fu* scores. The benefits of implementing support vector machines method are as follows:

- We don't need to consider specifically the distribution of the data
- Since there are quite a few predictors in the data set, SVM can handle high dimensions well with kernel functions, which can map n-dimensional data to different dimensions by using dot products
- Support vector machines only rely on relevant support vectors, not all the data points

3 Method evaluations

In this section, we will implement each method mentioned in the previous question and evaluate which method can predict the best results. We split the data into training and test data sets with 1000 samples and 152 samples respectively.

3.1 Generalized additive model(GAM)

Firstly, in order to assess the significance of each candidate predictors to the response variable, we fit GAM model with all predictors and full data set. Here comes to a problem with fitting the model. Since the candidate predictor "aedu" (Years of education completed) does not have enough data points to fit the model, we have to expand the basis complexity of "aedu" to $k = 5$, which means adding more basis functions to this predictor. k equals to 5 is randomly chosen, which is already big enough to stabilize the model fitting. In addition, in this implementation, we set the method to restricted maximum likelihood estimation. With this REML estimation method, the linear effects are treated as unpenalized fixed effects and the random effects are for non-linear effects, which penalizes towards to zero. Generalized cross validation is another method can be used for fitting the model. We will evaluate which fitting method performs better in our case.

According to the summary results of GAM model with full data, it is clear that almost half of the candidate predictors are not even statistically significant. This situation is inline with what we anticipated in the first step. In this case, we keep all the predictors with minimum significance level smaller than 0.05. These predictors are "disType", "Sexe", "bGAD", "bPanic", "sample", "PsychTreat", "aedu", "IDS", "AO".

Next, we refit the GAM model with remaining selected predictors. At this time, we use training data set. The results reveal that the majority of the predictors are statistically significant at 5% significance level, where the most significant predictors are *disType* and *IDS* with the smallest p value, followed by *PsychTreatTRUE* and *AO*. The rest of the predictors are less statistically significant with higher p values, but they are still below 5%. Additionally, when inspecting the smooth terms, the effective degrees of freedom for *aedu* is 1.015, which is approximately linear. We can also observe this linear relationship from plot 3. The first subplot of *aedu* is relative linear to the response variable. And the other subplots display non linearity of the two predictors. Hence we refit this model without smoothing *aedu*. The results reveal that there is not much changes to the previous fitting and the relevant predictors are still statistically significant. Furthermore, we observe that the two categories of *disType*, which are "comorbid disorder" and "depressive disorder" are positively correlated to response variable, whereas the TRUE category of *PsychTreat* is negatively related to *dep_sev_fu*. For smoothing terms, both of *IDS* and *AO* positively contribute to response variable *dep_sev_fu*. Moreover, the coefficients of statistically significant predictors remain the same.

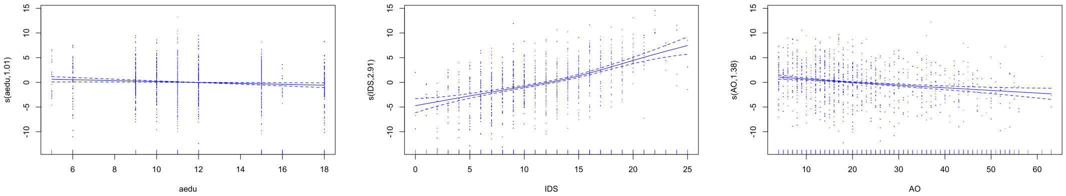

Figure 3: GAM model inspection

Subsequently we fit the GAM model again with method of GCV, which is the default setting of GAM. At most of cases we use REML method for the reason that it is more stable at predictions. Nevertheless, we can still perform 10-fold cross validation on the training data to evaluate which fitting method yields lower prediction errors. As to the predicted mean squared errors, REML method yields slightly smaller MSE than GCV method, which are 14.73 for REML and 14.77 for GCV respectively. Thus we use GAM model with REML method to predict *dep_sev_fu* using test data set. This yields an MSE of 19.26.

3.2 Random forest and Boosting

In this section, we implement random forest regression from r package *randomForest*. Since we are experimenting with random forest regression, the setting for "mtry" parameter should be the number of predictors divided by 3. Keeping other function parameters as default, the resultant graph 4 reveals that after generating around 100 trees, this model starts to converge with a mean squared error of 15.61 for training data. And the test MSE by plugging in test data in predict function is 20.11. The predicted test error is not ideal. Hence, we will subsequently implement boosting ensemble method and then evaluate whether the results can improve.

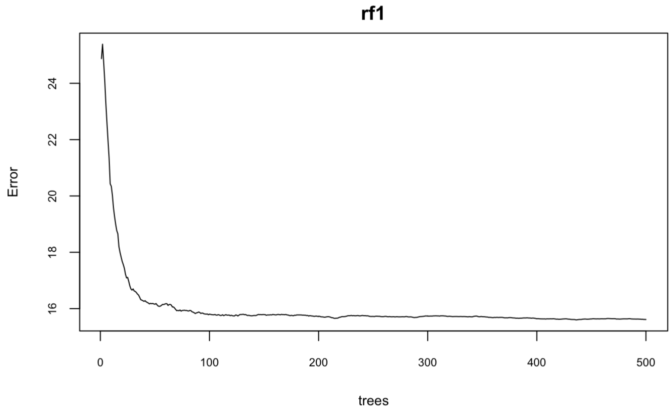

Figure 4: randomForest model convergence

Nevertheless, we can still get an idea of which predictors are strongly contributed to the response variable. Regarding to the importance plots(Figure 5) of this random forest model, we can clearly observe that *IDS* contributes the most to the increase in MSE, which is around 55.91%, and *disType* comes to the second with 31.52%. But at the third, *bTypeDep* takes the position with 17.34%, while it is not even statistically significant in the previous GAM spline regression. *PsychTreat* comes to the fourth important position with 10.63% contribution to the increase in MSE. In the IncNodePurity plot, order of the predictors is slightly different, and *IDS* and *disType* are still the most and second most influenced predictors. In addition, from the partial dependence plots 6, we can observe that *IDS* is positive related to response variable, while *AO* is in the opposite direction.

For boosting ensemble method, the r package and function of boosting ensemble algorithm is *gbm*. At the first attempt, we set the number of trees equal to 1000, tree depth to 4 and shrinkage to 0.01. The resulted plot 7 indicate that squared error loss start to increase from approximately 320 iterations and OOB change in squared error loss start to become negative instead of positive. The predicted test error is about 19.13, which is lower than previous random forest method. As to improving the performance of this boosting ensemble predictions, we create a parameter tuning grid which consists of n.trees equals to 100, 500 and 1000, interaction.depth equals to 3,4 and 5, and shrinkage equals to 0.1, 0.01 and 0.001. Using *train()* function from *caret* r package, the tuning results (Figure 8) show that the best-performing parameters are n.trees = 500, interaction.depth = 3 and shrinkage = 0.01. By plugging in these parameters, predicted test error is 19.60, which is even worse than

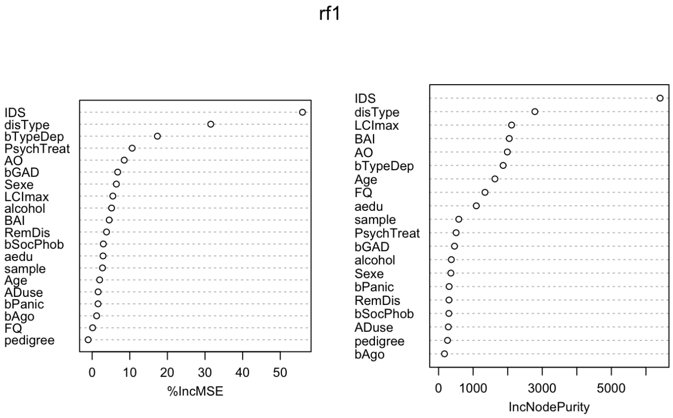

Figure 5: Importance of randomForest predictors

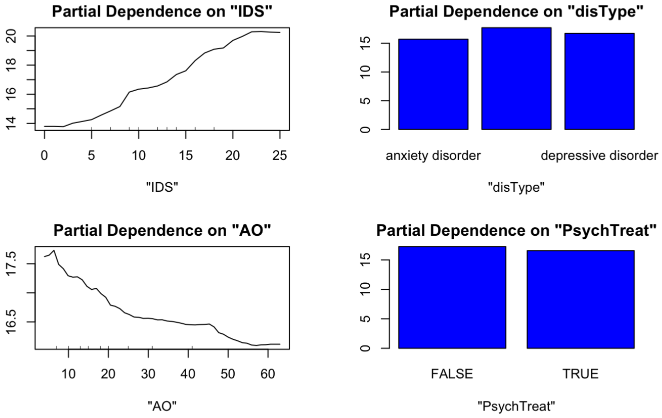

Figure 6: Dependence plots of important predictors

our first attempt. Thus we keep the first attempt boosting ensemble model as our final model.

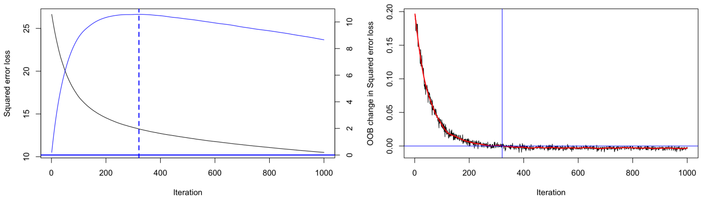

Figure 7: Boosting ensemble tree model convergence

Subsequently, we inspect the importance of predictors. Based on the graph below, predictor *IDS* has the highest influence on *dep_sev_fu* with magnitude of 38.96, and with higher IDS, *dep_sev_fu* increases. *disType* follows as the second highest influence predictor on response variable with magnitude of 14.64. This is a categorical predictor, and "comorbid disorder" category has the most impact on response. Here follows the third influenced predictor, which is *AO* with magnitude of 7.94 and it has a negative relationship with response *dep_sev_fu*.

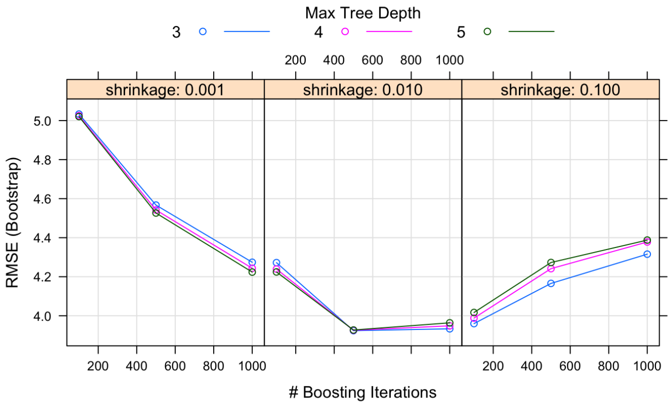

Figure 8: Boosting ensemble tree tuning parameters

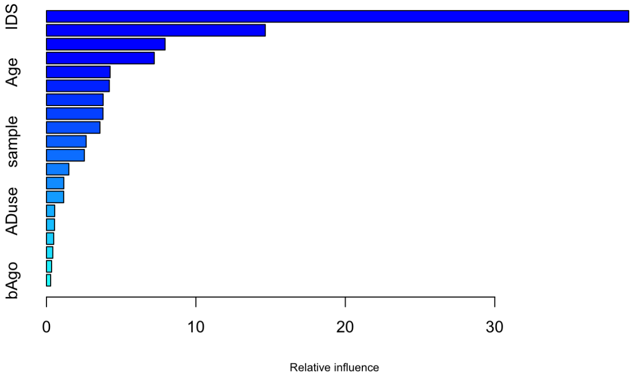

Figure 9: Relative influence

3.3 Support vector machine

Support vector machines can also be used for regression analysis. In this part, we are experimenting with SVM method to predict *dep_sev_fu*. The main principle of support vector machines regression is the same as SVM for classifications, which is maximizing margin and the margin only depends on some specific vectors. Cost constraint as well determines the level of violating margins. For this method, we experiment with three different kernel functions. They are linear, polynomial and radial kernels. When implementing each kernel, we will tune parameters respect to the specific kernel function. We use r package *e1071* to run SVM algorithm.

3.3.1 Linear kernel

For linear kernel, we only need to tune the cost parameters. We define a cost vector that consists of 0.001, 0.01, 0.1, 1, 5, 10, 100. We employ *tune()* function in r to help us do the task. The best parameters for cost is 1. By fitting SVM with the best parameter, our predicted test error based on linear kernel is 20.41.

The drawbacks of SVM is that it cannot distinguish important predictors, instead it only relies on supporting data points. Even though the unsupported data points are unaffected to the margin in a classification setting, in our case, those unrelated data can still have impacts on the test errors in this regression setting. To this end, we employ the results from GAM regression, where we refit support vector machines algorithm with only statistically significant predictors. The predicted test error improves to 19.81. Thus, for the rest of SVM experiments, we only use these statistically significant predictors.

3.3.2 Polynomial kernel

For polynomial kernel function, we define a grid vector of polynomial orders, which are 2, 3, 4 and 5. We supply polynomial orders grid and costs grid together into the tuning function. The resulted best parameters are cost = 1 and order = 2 and the predicted test error is 22.08.

3.3.3 Radial kernel

The procedure is the same as before for radial kernel function, but the difference is that we use predefined gamma grid and cost grid together when tuning the parameters. Gamma parameter is the smoothness of the decision hyper plane, in our case, it's the wiggliness of the regression line. The gamma grid is a vector consists of 0.1, 0.3, 0.5, 0.8 and 1.The resulted best parameters are $cost = 1$ and $gamma = 0.1$. Moreover, the predicted test mean squared error based on best performance model is 19.75.

Overall, radial kernel SVM generates the lowest mean squared error in the test data set. Thus our final SVM predictor is with radial kernel.

4 Model selections

In summary, from Table 1, we can see that boosting ensemble method yields the lowest predicted test MSE of 19.13. The advantage of tree based ensemble methods is that it can automatically select most significant predictors when running the algorithms. Moreover, the most influenced predictors are almost the same estimated by all three methods, but with slightly variations in orders. Overall, the most impacted predictors are *IDS* with positive effect, *disTpye* with positive effect, *AO* with negative effect as well as *PsychTreat* also with negative effect.

	GAM(REML)	Ensemble(boosting)	SVM(radial)
Train MSE	14.74	10.51	12.63
Test MSE	19.26	19.13	19.75

Table 1: Train and test MSE for three models

By comparing the pairwise differences of predicted values estimated by either two of the methods, we can generate confidence intervals of each pair difference by using *t.test()* function in r. In table 2, we display 95% confidence intervals for each pairwise predictions differences. All three pairwise prediction differences are not statistically significant. Predicted values from boosting ensemble method and SVM method have the widest intervals, which means they produce the most different response values, whereas GAM(REML) method and boosting ensemble method produce the most similar values.

	Lower bound	Upper bound	p-value
GAM(REML)-Ensemble	-0.0519	0.2675	0.1842
GAM(REML)-SVM(radial)	-0.0362	0.2842	0.1282
Ensemble-SVM(radial)	-0.1568	0.1892	0.8533

Table 2: Confidence intervals for pairwise differences

5 Model prediction

Finally, given the information data of David, we can predict his depression symptom severity and decide whether sending him to intensive treatment program. As applying boosting ensemble model, the predicted depression symptom severity for David is 19.42, which is bigger than 17. Thus the psychologist has to refer David to intensive treatment. We can also confirm this prediction by looking important predictors. For instance, David has *disType* of "depressive disorder", which has significantly positive effects on his depression severity, and his *IDS* of 13 (maximum is 25) also has strong positive effects. On the other hand, *AO* has significant negative influence, David's "AO" is 9, which impacts strongly to his depression severity. True *PsychTreat* is also negatively related to *dep_sev_fu*, and David's has no *PsychTreat*, which means it can increase his depression severity.