Predicting one's political view using General Social Study data

Hanah Chang

Introduction

The goal of this project is to find a model that has the highest proportion of correct predictions in deciding individuals who have a moderate political views.

This project will primarily use a variable poliviews. It represents how individuals place themselves on the seven-point scale on which the political views that people might hold are arranged from extremely liberal (1) to extremely conservative (7). Out of the seven scales, respondents who consider themselves moderate (4), is subset to make a dichotomous binary variable and used for the outcome. Now that all responses fall into one of two categories (moderate and not moderate), classification techniques are mostly used in this project.

For predictors, I decide to use four continues variables (educ: years of education, prestg80: occupational prestiage, realinc: income, age) and six categorical variables (sex: male true, race: nonwhites true, marital: non-married true, childs:no children true, wrkgovt: working at government true, reg16: lived in a rural at the age of 16 true). I chose these predictors on the base of numerous literature that explain strong evidence of relations between views and social/demographic factors (Description of the outcome and predictors is included in the Appendix, at the end of the research).

The dataset used in this study is 2006 General Social Survey (GSS). GSS is a nationwide survey that collects demographic, behavioral and attitudinal information of residents of United States.

```
vars <- c("polviews","educ", "prestg80", "realinc", "age", "sex", "race",  "marital", "childs", "wrkgov

d <- gssdata[,vars]
d <- na.omit(gssdata[,vars])
```

```
# outcome (polview) recode , subset 1, 2 making binaray outcomes)

library(plyr)
d$polviews <- with(d, polviews == 4)
d$polviews <- as.numeric(d$polviews)

table(d$polviews)
```

```
##
##    0    1
## 2192 1364
```

```
d$sex <-  d$sex== 1 #male
d$sex <- as.factor(d$sex)

d$race <-  d$race == 3 # nonwhite
d$race <- as.factor(d$race)
```

```
d$marital <- with(d, marital == 2 | marital == 3| marital == 4| marital == 5)
#notmarried
d$marital <- as.factor(d$marital)

d$childs <- d$childs == 0 #nochild
d$childs <- as.factor(d$childs)

d$wrkgovt <- d$wrkgovt == 1 # working at government
d$wrkgovt <- as.factor(d$wrkgovt)

d$reg16 <- with(d, reg16 == 1 | reg16 == 2) #lived at rural
d$reg16  <- as.factor(d$reg16 )
```

```
set.seed(123)

Train <- sample(1:nrow(d), size = 1778, replace = FALSE)
training <- d[Train, ]
testing <- d[-Train, ]
```

Initial model: Logit, Stepwise Selection

(Logit)

I estimated a logit model in the training data and calculate the proportion of correct predictions in the testing data. The result shows that the function makes 60.46% correct predictions out of total 1,778 cases.

```
logit <- glm(polviews ~ educ + prestg80+ I(log(realinc))+ age + sex + race + marital + childs + wrkgovt
summary(logit)
```

```
##
## Call:
## glm(formula = polviews ~ educ + prestg80 + I(log(realinc)) +
##     age + sex + race + marital + childs + wrkgovt + reg16, family = binomial,
##     data = training)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.577  -1.006  -0.859   1.307   2.005
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.140290   0.601990  -0.233   0.8157
## educ            -0.085570   0.020210  -4.234  2.3e-05 ***
## prestg80        -0.006853   0.004264  -1.607   0.1080
## I(log(realinc))  0.122200   0.059451   2.055   0.0398 *
## age             -0.002429   0.003278  -0.741   0.4588
## sexTRUE         -0.090880   0.100787  -0.902   0.3672
## raceTRUE         0.097060   0.160898   0.603   0.5464
## maritalTRUE      0.098703   0.111336   0.887   0.3753
## childsTRUE      -0.170523   0.124528  -1.369   0.1709
## wrkgovtTRUE      0.157692   0.129369   1.219   0.2229
## reg16TRUE        0.152140   0.131742   1.155   0.2482
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2377.7  on 1777  degrees of freedom
## Residual deviance: 2336.0  on 1767  degrees of freedom
## AIC: 2358
## 
## Number of Fisher Scoring iterations: 4
```

```
p_logit <- predict(logit, newdata = testing, type = "response")
table(testing$polviews, as.integer(p_logit > 0.5))
```

```
## 
##        0    1
##   0 1044   63
##   1  640   31
```

```
mean(testing$polviews == ((p_logit) > 0.5))
```

```
## [1] 0.6046119
```

(add a step function to logit regression)

Forward selection starts with a model that just has an intercept and sequentially adds predictors. Backward selection starts with the full model and sequentially subtracts predictors. The Step() function I used in here uses a combination of both directions and ranks the models by comparing the Aikaike Information Criterion (AIC).

The result shows that the step function dropped 6 predictors (age, race, marital, childs, wrkgov, reg16). When used to predict outcomes in the testing data, this reduced model makes fewer errors. And now I have 1,081 correct cases which account for 60.57% of total cases. Compare to logit model, the proportion of correct predictions in the testing data slightly increased by 0.11%P.

```
logit2 <- step(logit, trace = FALSE)
names(coef(logit))
```

```
##  [1] "(Intercept)"      "educ"             "prestg80"         "I(log(realinc))"
##  [5] "age"              "sexTRUE"          "raceTRUE"         "maritalTRUE"
##  [9] "childsTRUE"       "wrkgovtTRUE"      "reg16TRUE"
```

```
names(coef(logit2))
```

```
## [1] "(Intercept)"     "educ"            "prestg80"        "I(log(realinc))"
```

```
p_logit2 <- predict(logit2, newdata = testing,  type="response")
table(testing$polviews, as.integer(p_logit2 > 0.5))
```

```
##
##          0    1
##    0 1052   55
##    1  646   25
```

```
mean(testing$polviews == (p_logit2 > 0.5))
```

```
## [1] 0.6057368
```

Expended model 1: Linear Discriminant Analysis, Lasso and Generalized Additive Models

The parameter estimates for the logistic regression model are unstable when the classes are well-separated. On the other hand, a linear discriminant analysis (LDA) does not suffer from this problem. Also, If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model. This is why I used the LDA as a way to improve my initial logit model.

(Linear Discriminant Analysis / Quadratic Discriminant Analysis)

I expected that the LDA will yield the smallest possible total number of misclassified observations, irrespective of which class the errors come from. The result shows 1,077 correct predictions (60.57%) in testing data, and this is no better than the logit + step function.

I also used quadratic discriminant analysis (QDA) approach. While LDA assumes that the observations within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a covariance matrix that is common to all K classes, QDA estimates a separate covariance matrix for each class, for a total of Kp(p+1)/2 parameters.

The QDA result shows 1,100 correct predictions (61.87%). According to the textbook, LDA is popular when we have more than two response classes. In addition, Garrett Grolemund / says QDA is recommended if the training set is very large so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable. Since my data is not applied to these cases, the results from LDA and QDA are not much satisfactory.

```
library(MASS)
LDA <- lda(logit2$formula, data = training)
QDA <- qda(logit2$formula, data = training)

p_LDA <- predict(LDA, newdata = testing)$class
table(testing$polviews, p_LDA)
```

```
##     p_LDA
##          0    1
##    0 1052   55
##    1  646   25
```

```
p_QDA <- predict(QDA, newdata = testing)$class
table(testing$polviews, p_QDA)
```

```
##     p_QDA
##          0    1
##    0 1095   12
##    1  666    5
```

(Lasso)

The maximum number of correct predictions is 1,107 (62.26%). In this case, lasso penalization is slightly better than QDA (correct predictions of 1,100).

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-2
```

```r
Lasso <- glmnet(x = model.matrix(logit2)[, -1], y = training$polviews,
                family = "binomial")


X <- model.matrix(logit2, data = testing)[, -1]
correct <- colSums(testing$polviews == (plogis(predict(Lasso, newx = X)) > 0.5))

# Note that lasso yields a sequence of models, so we have to find the best one:
max(correct)
```

```
## [1] 1107
```

(Generalized Additive Models)

I used a gam() function to fit a Generalized Additive Model (GAM) where *moderate political view* is the outcome using the predictors from the best model found via step() for the training data.

The GAM model is: Moderate political views = B0 + f1(education) + f2(occupational prestige) + f3(income) + e

After running the gam function, the plots tell us how each spline function changes as the corresponding predictor changes. The curvature of each spline function is estimated in order to fit the (training) data. It looks like all of three continuous variables (education, occupational prestige, and Income) exhibit a non-linear relationship with the outcome (moderate political views), conditional on the other predictors. The last plot for the dichotomous factor variable (sex = male true) shows that the outcome (moderate political views) for women (sex = false) is higher than for male (sex = true); holding other variables fixed.

```r
stopifnot(require(gam))
```

```
## Loading required package: gam
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.16.1
```

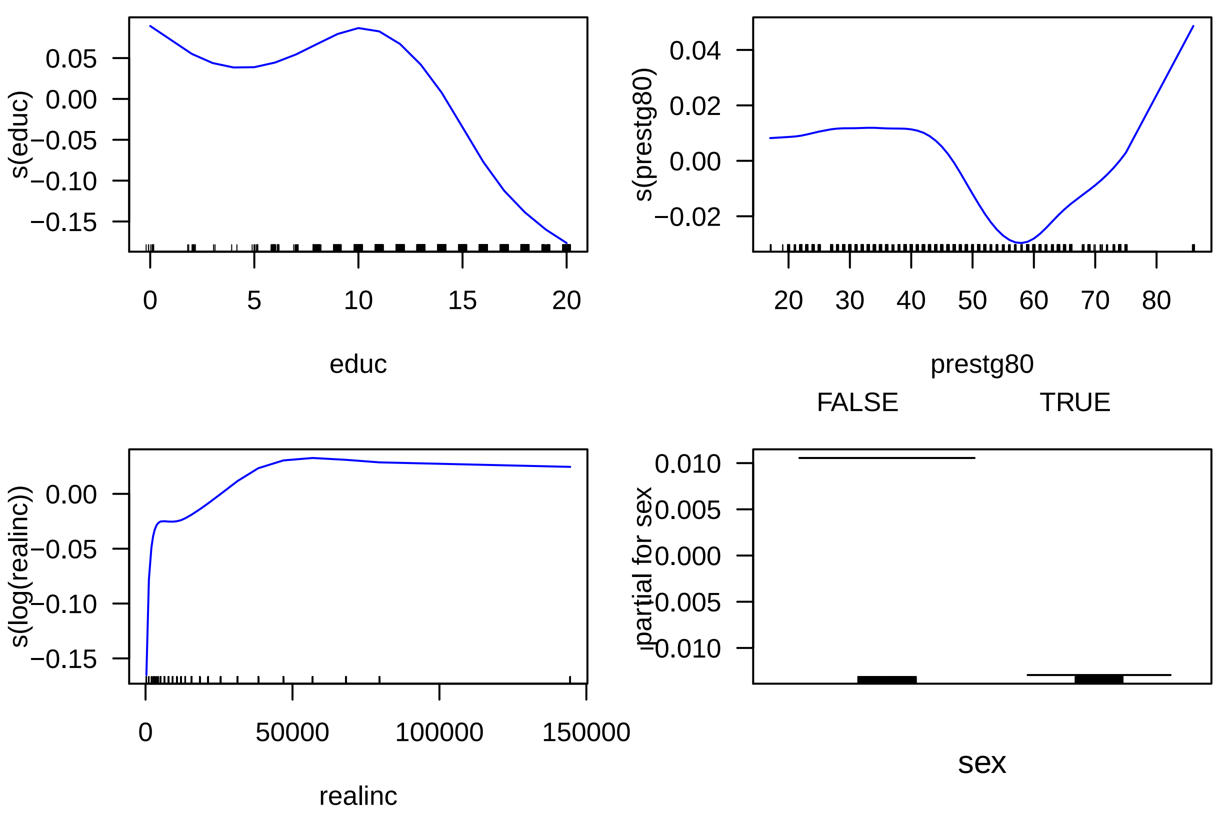
```r
gam_train <- gam(polviews ~ s(educ) + s(prestg80) + s(log(realinc)) + sex, data = training)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
## ignored
```

```
# make Plots

par(mfrow=c(2,2), las = 1, mar = c(5,4,1,1) + .1)

plot(gam_train,col="blue")
```



Since our predictors are non-linear with the outcome, we can take advantage of GAM, because GAM allow us to fit a non-linear fj to each Xj and the non-linear fits can potentially make more accurate predictions for the response Y . The average SSR is smaller in the training data (0.2304) than in the testing data (0.2311), which is to be expected given that the parameters underlying the predictions are estimated from the training data in order to minimize the SSR.

```
# Predicting the gam model in the training data
Yhat_gam00 <- predict(gam_train)
mean( (training$polviews - Yhat_gam00) ^ 2 )
```

```
## [1] 0.2303641
```

```
# Predicting the gam model in the testing data
Yhat_gam <- predict(gam_train, newdata = testing)
mean( (testing$polviews - Yhat_gam) ^ 2 )
```

```
## [1] 0.2310977
```

I also compared mean squared error for logit + step model with a generalized additive model. The MSE for logit+step was 0.9923 Compared to the logit model, the GAM provides a better fit in terms of least squares.

```
MSE_logit2 <- mean( (testing$polviews -
                    predict(logit2, newdata = testing)) ^ 2)
MSE_logit2
```

```
## [1] 0.9922816
```

Expended model 2: Tree-based methods

I chose to use tree-based models(plain tree,bagging, random forest) for classification to investigate they fit better than Logit/LDA/QDA/GAM models.

(plain tree + prune.tree)

The plot shows a classification tree for predicting the moderate political views, based on the number of years of education. As we can see, only one variable (education) is used in tree construction, and the number of terminal nodes is 2.

The plain tree approach is virtually guaranteed to overfit in the training data and predict poorly in the testing data. To compensate, I utilized a tree pruning after the initial algorithm has terminated to choose the best parts of the original tree. However, in this case, the _tree pruning did not result in minimizing mean squared errors.

The left-hand branch corresponds to years of education <14.5, and the right-hand branch corresponds to years of education >=14.5. However, regardless of the value of education, a response value of 0(not moderate political view) is predicted because it leads to increased node purity. this model shows accuracy of 62.26 % but failed to predict true negatives.

```
training$polviews <- as.factor(training$polviews)
testing$polviews <- as.factor(testing$polviews)
```

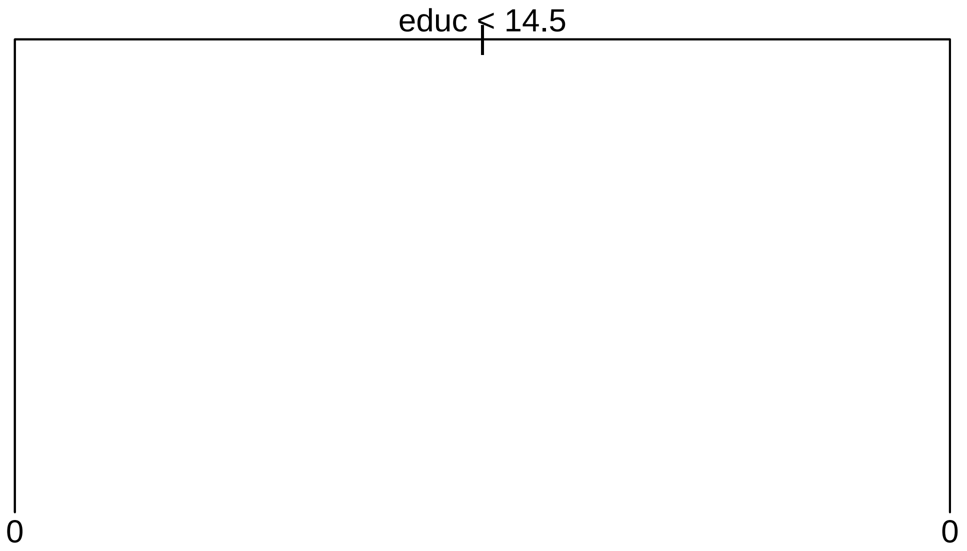
```
stopifnot(require(tree))
```

```
## Loading required package: tree
```

```
tree_model <- tree(polviews ~ ., data = training)
summary(tree_model)
```

```
##
## Classification tree:
## tree(formula = polviews ~ ., data = training)
## Variables actually used in tree construction:
## [1] "educ"
## Number of terminal nodes:  2
## Residual mean deviance:  1.315 = 2335 / 1776
## Misclassification error rate: 0.3898 = 693 / 1778
```

```
plot(tree_model)
text(tree_model, pretty = 0)
```



```
new_tree <- cv.tree(tree_model, FUN = prune.misclass)
new_tree$dev
```

```
## [1] 693 693
```

```
best_tree <- prune.tree(tree_model, best = 2)
summary(best_tree)
```

```
##
## Classification tree:
## tree(formula = polviews ~ ., data = training)
## Variables actually used in tree construction:
## [1] "educ"
## Number of terminal nodes:  2
## Residual mean deviance:  1.315 = 2335 / 1776
## Misclassification error rate: 0.3898 = 693 / 1778
```

```
tree_hat <- predict(best_tree, newdata = testing, type = "class")
table(testing$polviews, tree_hat)
```

```
##    tree_hat
##        0    1
##   0 1107    0
##   1  671    0
```

```r
mean(testing$polviews == tree_hat)
```

```
## [1] 0.6226097
```

(bagging)

As shown in the plot, we can immediately see that *education* is the most important factor in terms of the mean decrease in accuracy, and *age/occupational prestige/income/education* are the most important factor in terms of gini index.(The mean decrease in the Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest)

```r
stopifnot(require(randomForest))
```

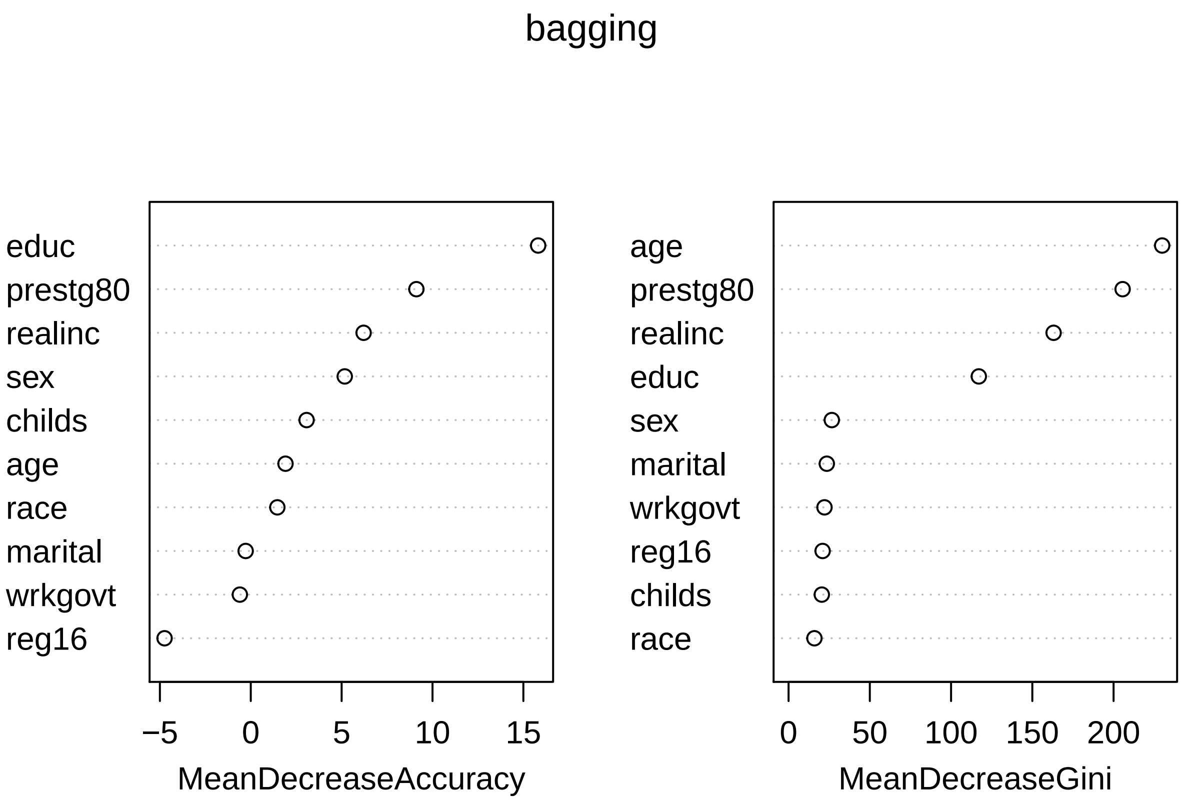
```
## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.
```

```r
bagging <- randomForest(polviews ~ ., data = training, mtry = 10, importance = TRUE)
```

```r
varImpPlot(bagging)
```



Bagging shows 57.59% of correct predictions in the testing data.

```
bagging_hat <- predict(bagging, newdata = testing, type="response")
table(testing$polviews, bagging_hat)
```

```
##    bagging_hat
##       0   1
##   0 808 299
##   1 455 216
```

```
mean(testing$polviews == bagging_hat)
```

```
## [1] 0.575928
```

(random forest)

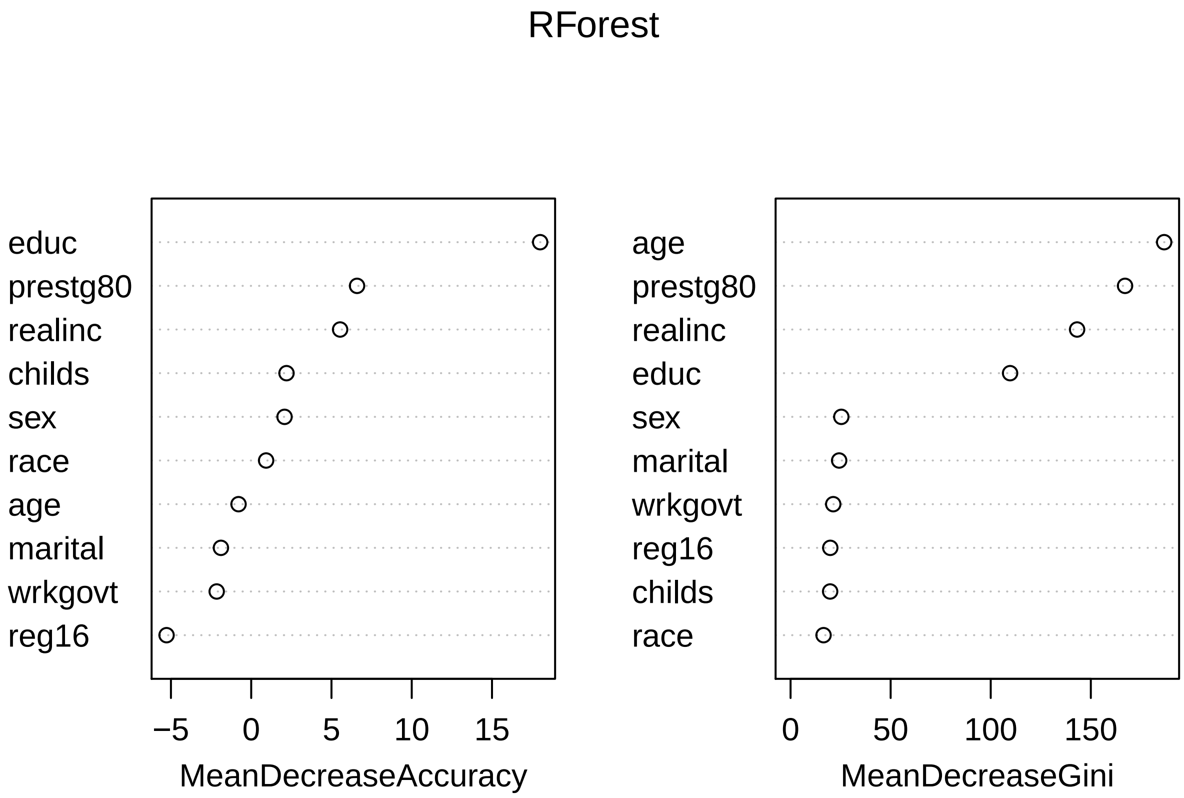
The varImpPlot()shows result similar to bagging. Noticeably, importance of *childs/sex/occupational prestige* is degreased when using random forest.

The proportion of correct predictions for random forest slightly better than bagging, the proportion is now 59.00%

```
stopifnot(require(randomForest))

RForest <- randomForest(polviews ~ ., data = training, importance = TRUE)

varImpPlot(RForest)
```



```r
RForest_hat <- predict(RForest, newdata = testing, type="response")
table(testing$polviews, RForest_hat)
```

```
##    RForest_hat
##       0   1
##   0 886 221
##   1 508 163
```

```r
mean(testing$polviews == RForest_hat)
```

```
## [1] 0.5899888
```

Conclusion

Throughout this project, multiple supervised learning techniques (logit + step, LDA, QDA, lasso, plain tree, random forest, bagging) are utilized to find a best predictive solution.

The predictions of each classification model are as follows. **qda 61.87% > lda 60.57% > logit 60.46% > randomforest 59.00% >bagging 57.59%**

(I did not include tree 62.26% correct prediction, because these approaches compensate all true negatives for true positives, resulting 0 case of true negatives. This is not what I want.)

The complex models (i.e. random forests, bagging) not always provide a significant improvement over the simpler methods (i.e. logit), thus, logit is selected for deployment.

(Appendix: Description of the outcome and predictors)

```r
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:randomForest':
##
##     outlier
```

```r
describe(d)
```

```
##          vars    n     mean       sd  median  trimmed      mad    min      max
## polviews    1 3556     0.38     0.49     0.0     0.35     0.00   0.00      1.0
## educ        2 3556    13.54     3.11    13.0    13.63     1.48   0.00     20.0
## prestg80    3 3556    44.41    14.08    44.0    43.81    14.83  17.00     86.0
## realinc     4 3556 34715.27 32822.09 25582.5 28677.44 20017.88 284.25 144502.7
## age         5 3556    46.78    16.47    45.0    45.85    17.79  18.00     89.0
## sex*        6 3556     1.46     0.50     1.0     1.45     0.00   1.00      2.0
## race*       7 3556     1.12     0.32     1.0     1.02     0.00   1.00      2.0
## marital*    8 3556     1.51     0.50     2.0     1.51     0.00   1.00      2.0
## childs*     9 3556     1.27     0.45     1.0     1.22     0.00   1.00      2.0
## wrkgovt*   10 3556     1.19     0.39     1.0     1.11     0.00   1.00      2.0
## reg16*     11 3556     1.17     0.37     1.0     1.09     0.00   1.00      2.0
##              range  skew kurtosis     se
## polviews      1.0  0.48    -1.77   0.01
## educ         20.0 -0.65     2.03   0.05
## prestg80     69.0  0.39    -0.49   0.24
## realinc  144218.5  2.06     4.34 550.41
## age          71.0  0.43    -0.51   0.28
## sex*          1.0  0.15    -1.98   0.01
## race*         1.0  2.41     3.80   0.01
## marital*      1.0 -0.04    -2.00   0.01
## childs*       1.0  1.01    -0.98   0.01
## wrkgovt*      1.0  1.57     0.46   0.01
## reg16*        1.0  1.77     1.15   0.01
```