HW 6 REPORT

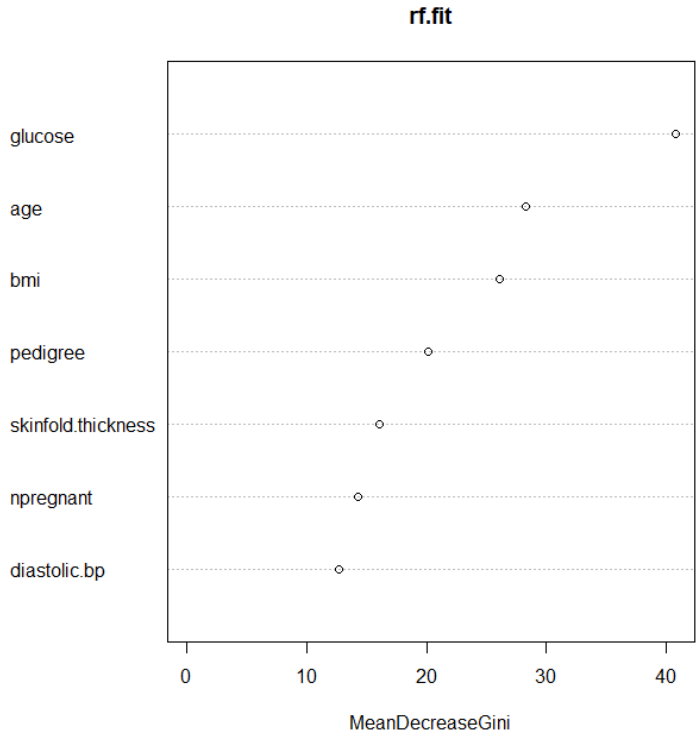
Rohith Reddy Kolla | rkolla

1. The data set chosen for this part is the pima dataset (since it is being allowed). The pima set has 9 columns and since the 8th and 9th column both represent the class, the 9th column has been removed.

```
> names(pima)
[1] "npregnant"          "glucose"            "diastolic.bp"
[4] "skinfold.thickness" "bmi"                "pedigree"
[7] "age"                "classdigit"         "class"
```

a) A training set is created from the 2/3rds of the data and the remaining is the test set. Firstly, a random forest model is fit on the training data with the number of trees = 5000. The variable importance plot from that fit is shown below.

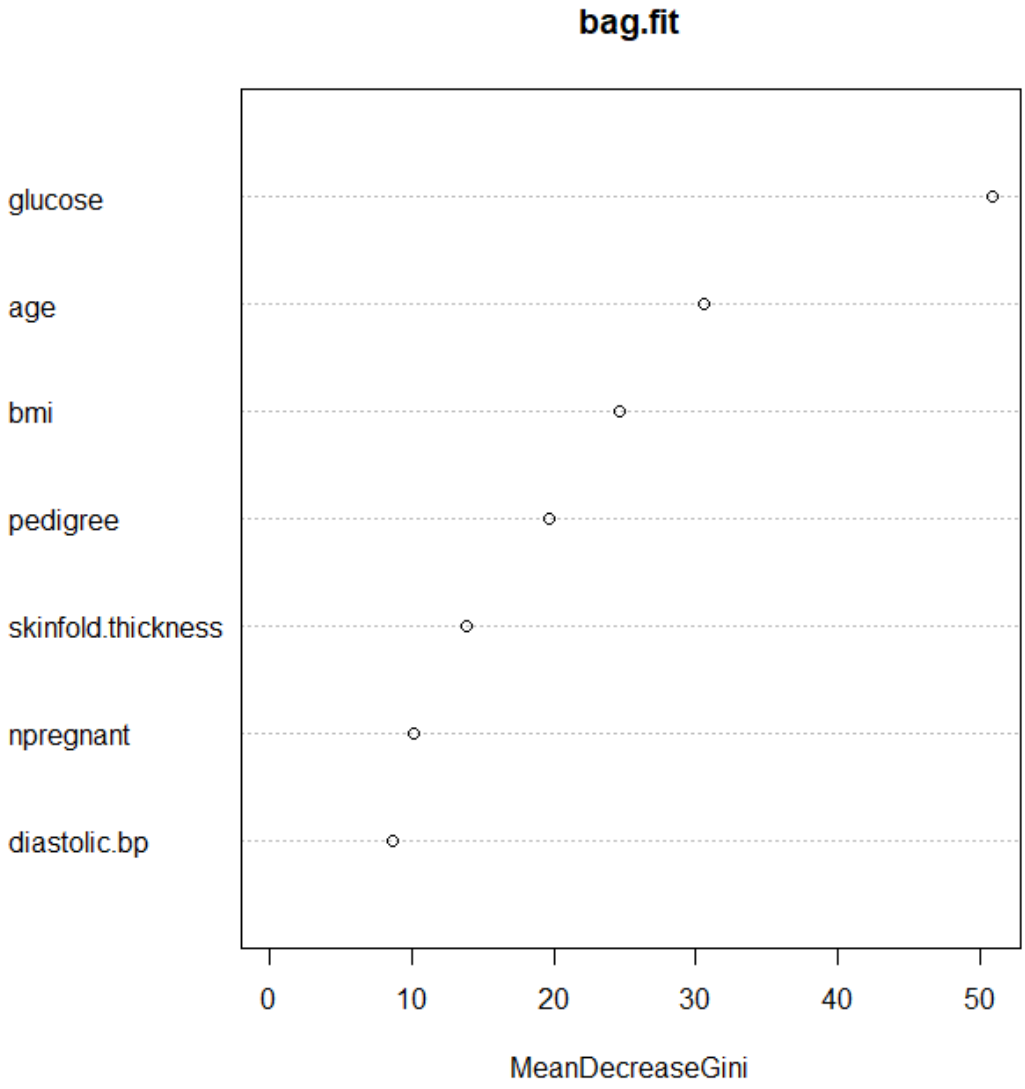
	MeanDecreaseGini
npregnant	14.05347
glucose	42.07753
diastolic.bp	12.23379
skinfold.thickness	16.07095
bmi	25.25146
pedigree	20.58659
age	27.91512



The model is able to predict the test data with a misclassification error rate of 0.276.

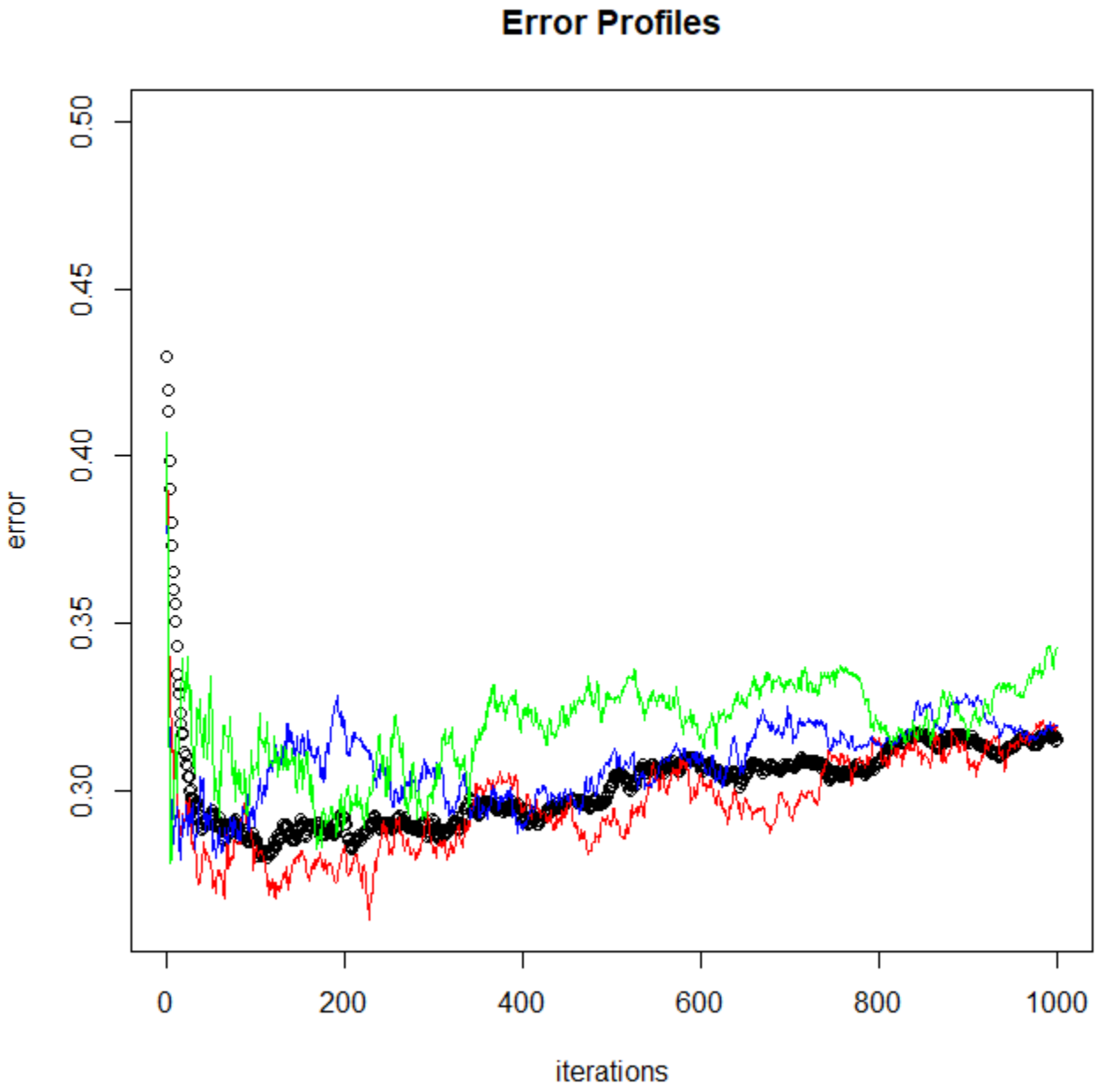
Next a bagged model is fit on the train data with the number of trees = 5000 and the variable importance plot from that fit is shown below. It is essentially similar to the variable importance from the previous fit but notably, the glucose variable is given more importance.

	MeanDecreaseGini
npregnant	10.119345
glucose	50.897504
diastolic.bp	8.607507
skinfold.thickness	13.821261
bmi	24.703368
pedigree	19.640338
age	30.564300



This model is able to predict the test data with a misclassification error rate of 0.265 which is slightly better than the previous model.

Next, 4 different vales of shrinkage (.1 - black, .3 - red, .5 - blue, .7 - green) are considered to fit a boosted model on the training data with 1000 as the max iteration and the resulting error is shown in the graph below.



The best misclassification error rate for this model comes out to be 0.261 which is slightly better than the previous two models.

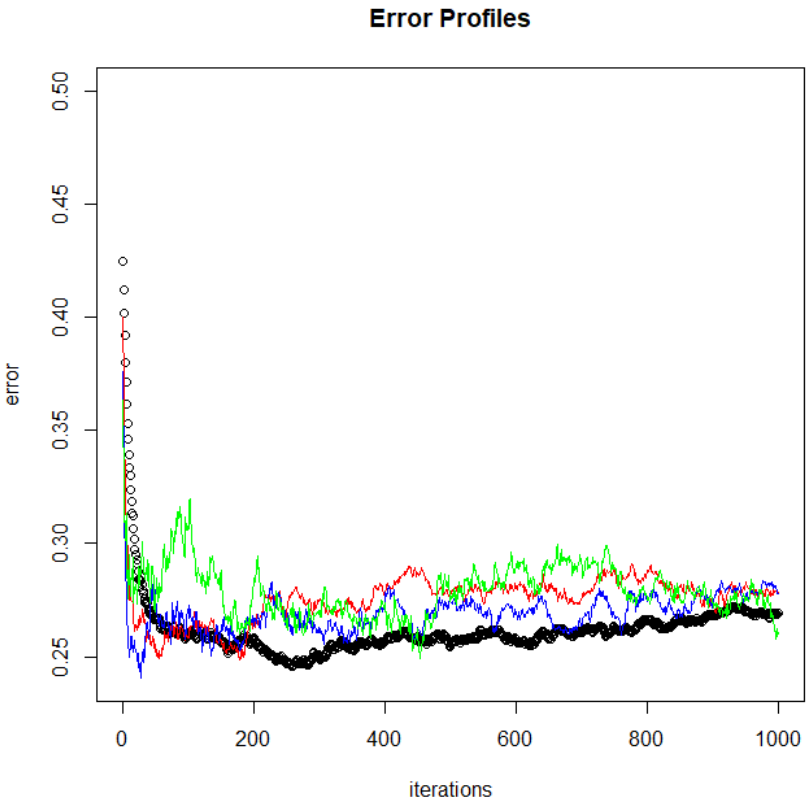
b) In the case of the pima data. The results from the above methods are comparable to the results from logistic regression. The resulting summary from fitting a logistic model to the train data using the glm function is shown below.

```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -10.482140   1.270881  -8.248  < 2e-16 ***
npregnant          0.131672   0.053389   2.466 0.013652 *
glucose            0.033402   0.005214   6.406 1.49e-10 ***
diastolic.bp      -0.007113   0.012264  -0.580 0.561915
skinfold.thickness 0.005175   0.018557   0.279 0.780343
bmi                0.106423   0.029490   3.609 0.000308 ***
pedigree           1.246300   0.442771   2.815 0.004881 **
age                0.038082   0.017234   2.210 0.027126 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The misclassification error rate based on the predictions on the same training set as the methods before comes out to be 0.265 which is better than the random forests method but almost the same as the bagging and boosting method.

c) The pima data considered is used to learn prediction based on diagnostic measurements whether a patient has diabetes. With only 7 diagnostic measurements, the data is relatively simplistic and basic machine learning algorithms are able to do a good enough job at prediction and committee machines do not provide a significant improvement. However, since committee machines are based on a combination of experts, over-fitting is less likely to occur. Furthermore averaging performances generally provides better results and reliability.

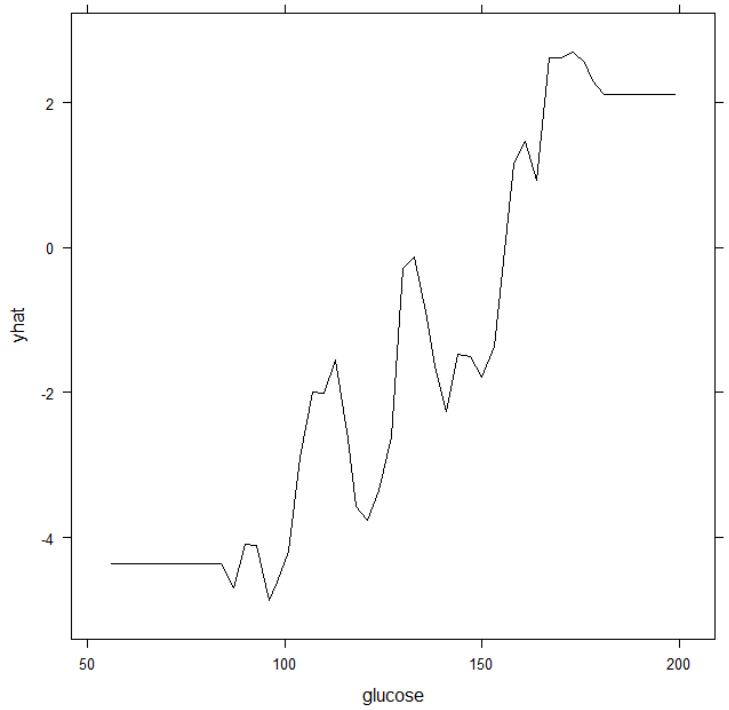
2. The pima data is considered again but a different random seed is used to split the 2/3rd of the data into training and rest into test. Firstly, a boosted model is fit on the training data with a max iteration of 1000 considering the same 4 shrinkage parameters as used before (.1 - black, .3 - red, .5 - blue, .7 - green). The minimum test misclassification error rate comes out to be 0.24.



From the above results, it seems that shrinkage = 0.1 has lesser oscillation and generally more consistent. Therefore, considering that shrinkage to estimate the variable importance provides the results shown below. According to this model, glucose turns out to be the most important variable followed by pedigree, bmi and age.

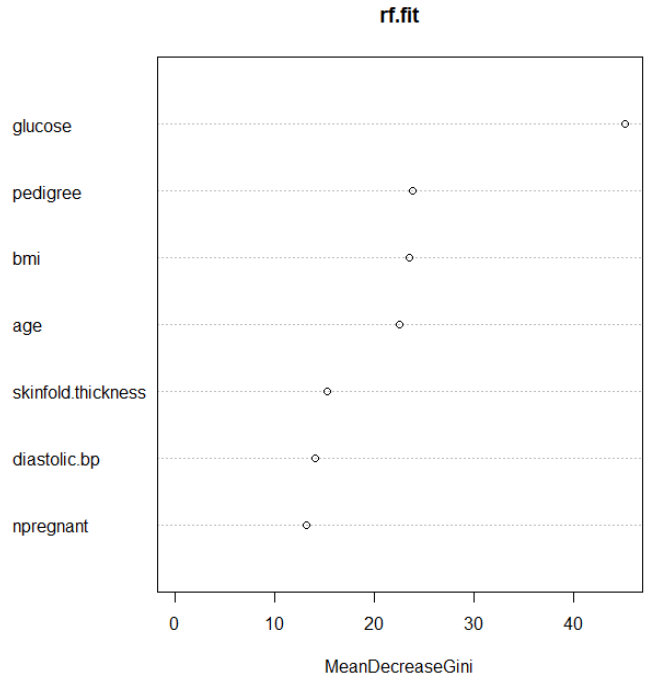
	var	rel.inf
glucose	glucose	25.344989
pedigree	pedigree	19.612101
bmi	bmi	16.220577
age	age	13.280088
skinfold.thickness	skinfold.thickness	10.069195
diastolic.bp	diastolic.bp	7.854389
npregnant	npregnant	7.618661

The partial dependency plot considering the glucose variable for the boosted model is shown below.

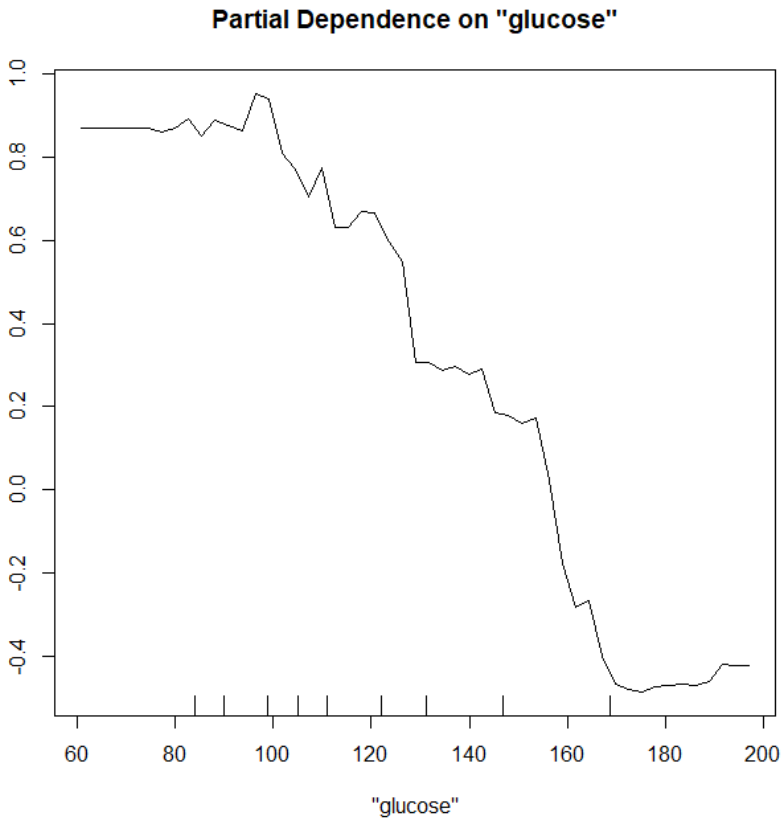


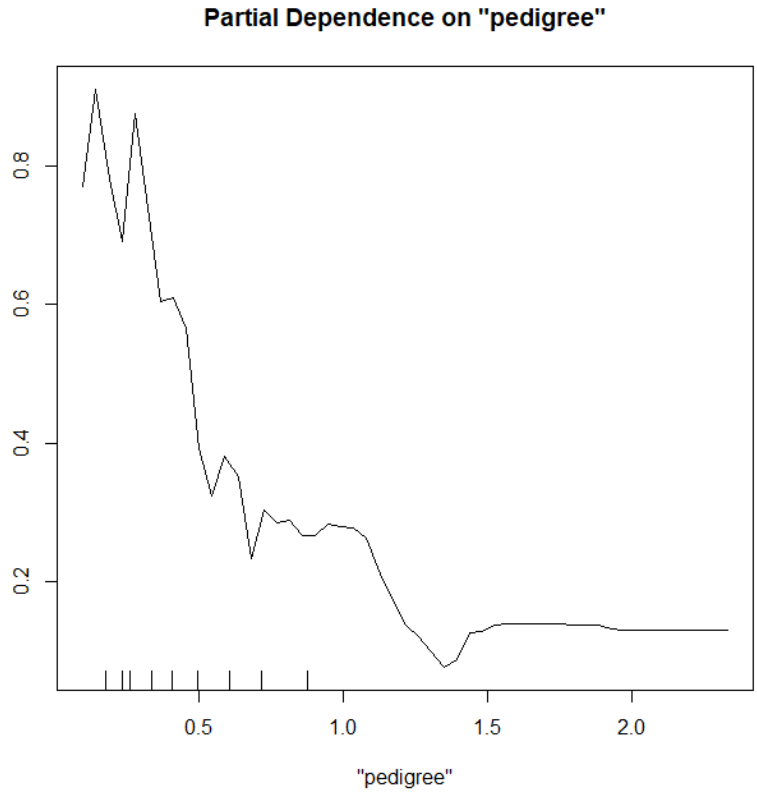
Next, a random forest model is fit on the training data with the number of trees = 10000. The test misclassification error rate comes out to be 0.225 which is slightly better than the above method. The variable importance plot from that fit is shown below. Similar to the boosted model the most important variables turn out to be glucose followed by pedigree, bmi and age.

	MeanDecreaseGini
npregnant	13.14858
glucose	45.10472
diastolic.bp	14.02038
skinfold.thickness	15.22314
bmi	23.46053
pedigree	23.83827
age	22.52033

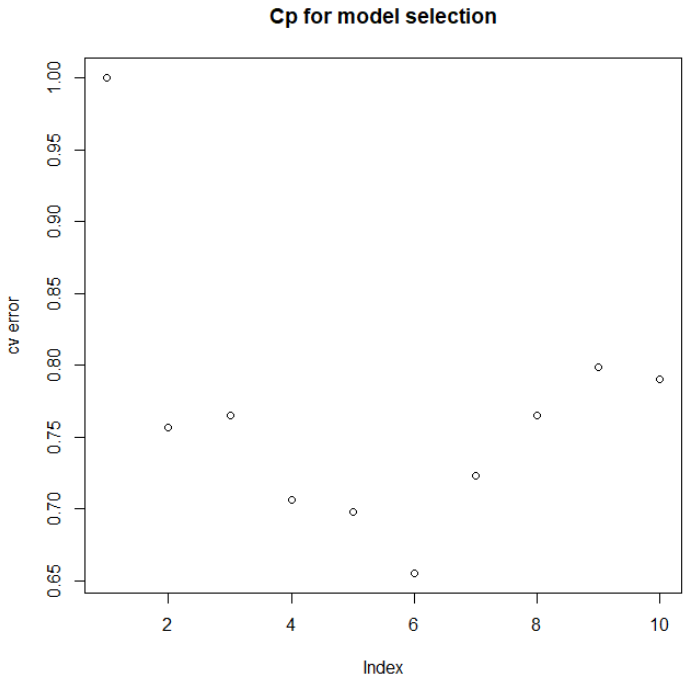


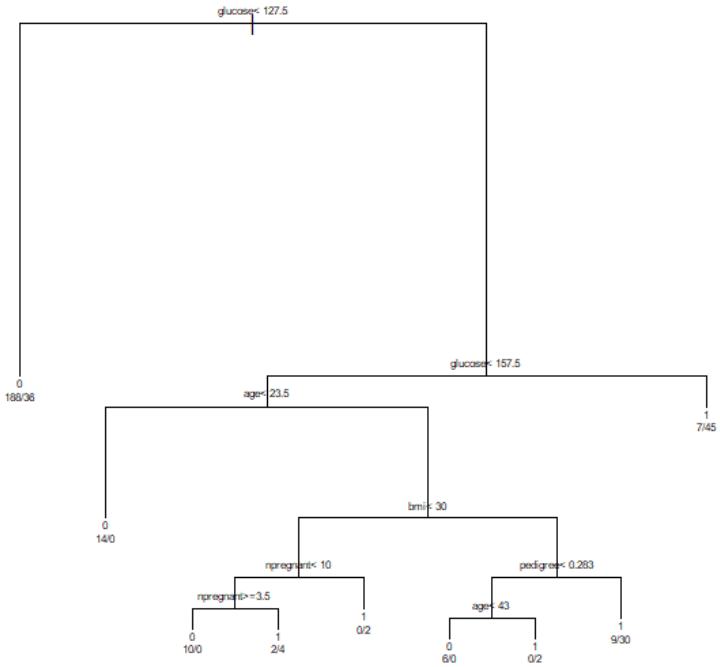
The partial dependence plots for the two most important variables in the random forest method is shown below.





Lastly a single tree (CART model) is fit on the training data with minsplit = 5, cp = 0 and pruned. The plots below indicate the cv error based on cp along and the stricture of the pruned tree. The misclassification error rate is 0.242 which makes it the worst performing of the three methods.





1 represents the diabetic class and 0 represents the normal class

The variable importance of the pruned tree is shown below along with the partial dependency plots for the model's two most important variables - glucose and age.

```
Variable importance
           glucose                age                bmi
                49                 17                 10
          npregnant         diastolic.bp           pedigree
                 8                  7                  6
skinfold.thickness
                 4
```