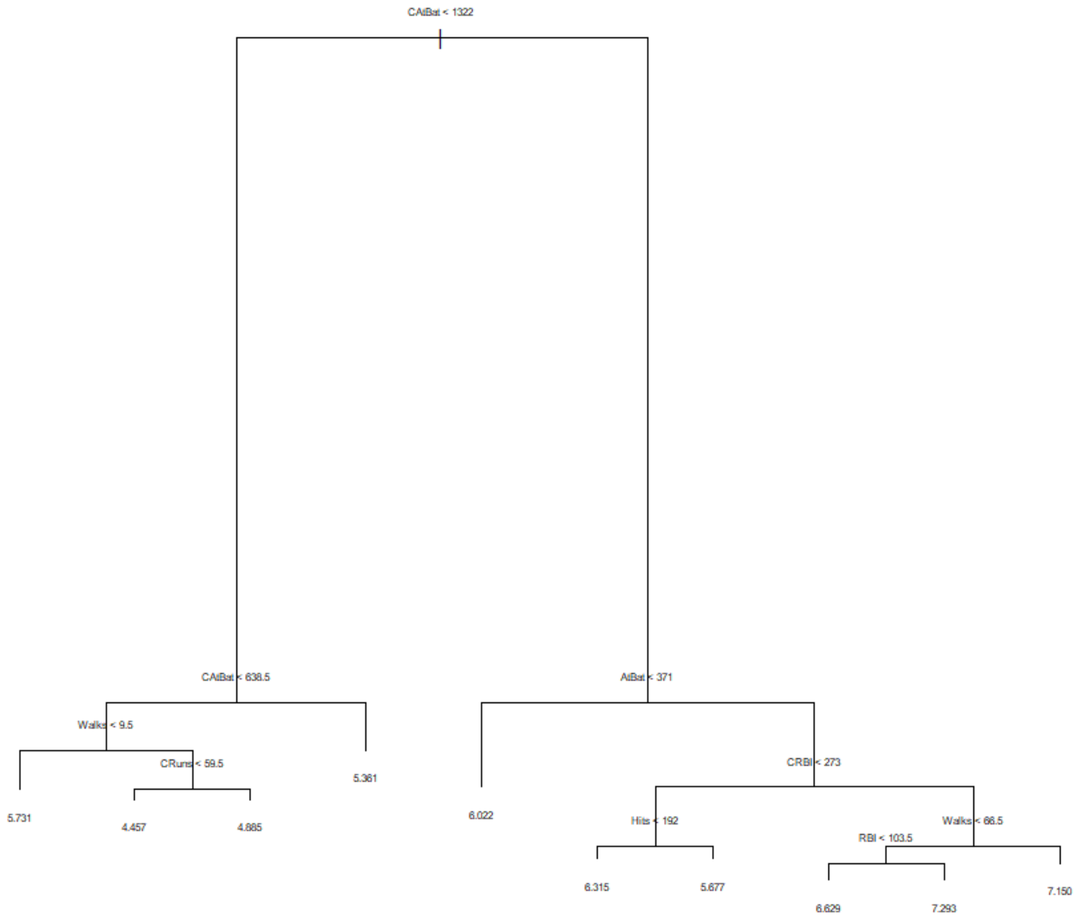
1.

I used Hitters dataset from ISLR. There are 20 variables, and 263 baseball players. Before splitting the dataset, I omitted the NAs. The data is spitted into 80:20. First thing I did is making a decision tree. The minimum tree is 24, and I pruned the decision tree. The pruned tree is the figure below:



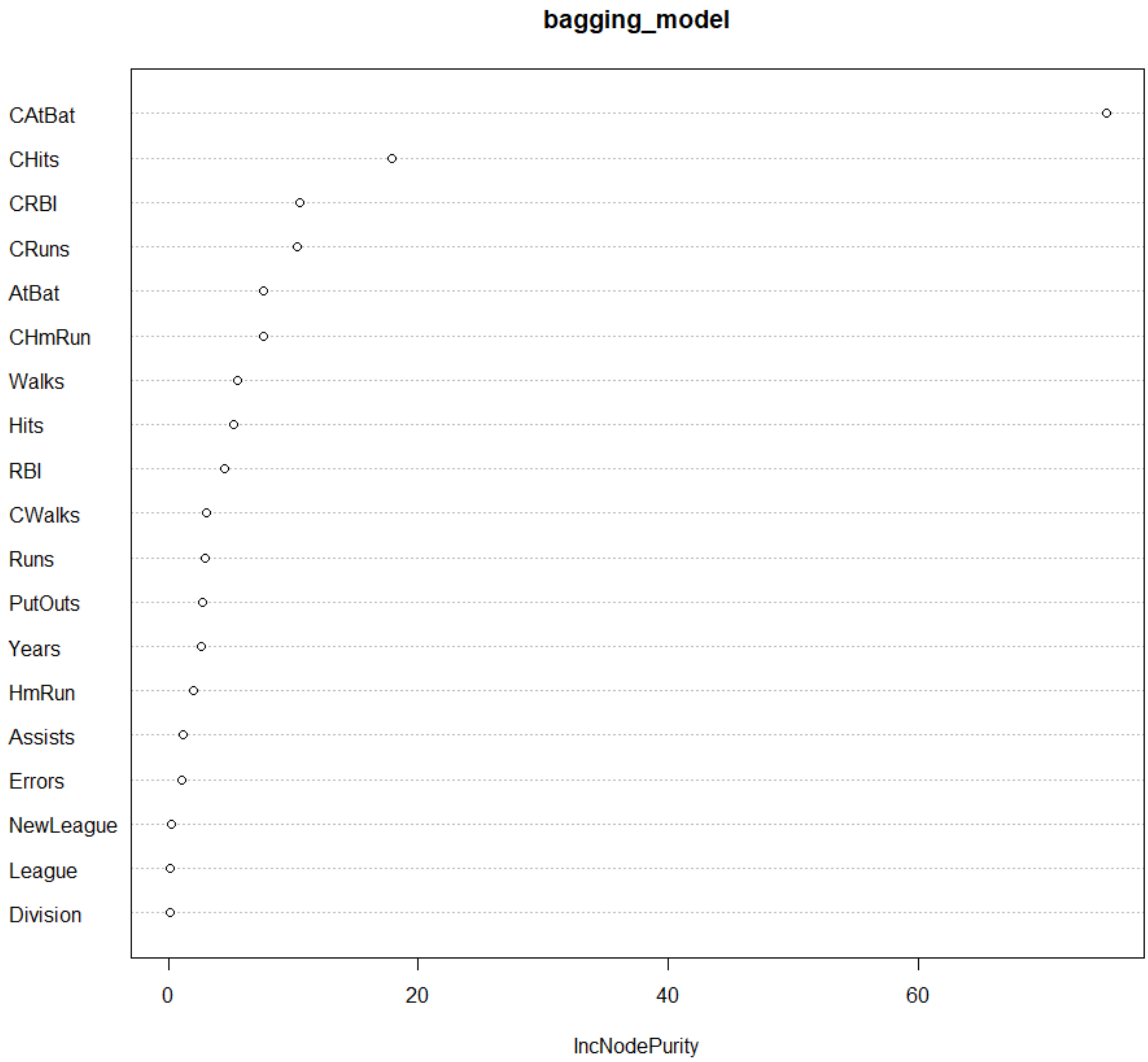

```
Regression tree:
tree(formula = Salary ~ ., data = train)
Variables actually used in tree construction:
[1] "CAtBat"  "CHits"   "Hits"    "CWalks"  "Walks"   "CHmRun"  "Assists"
Number of terminal nodes:  11
Residual mean deviance:  0.1881 = 37.44 / 199
Distribution of residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-1.534000 -0.191300  0.005533  0.000000  0.238400  1.902000
```

I compared the pruned mean squared error and unpruned mean squared error.

Pruned mean squared error: 0.1542393, Unpruned mean squared error: 0.1709158.

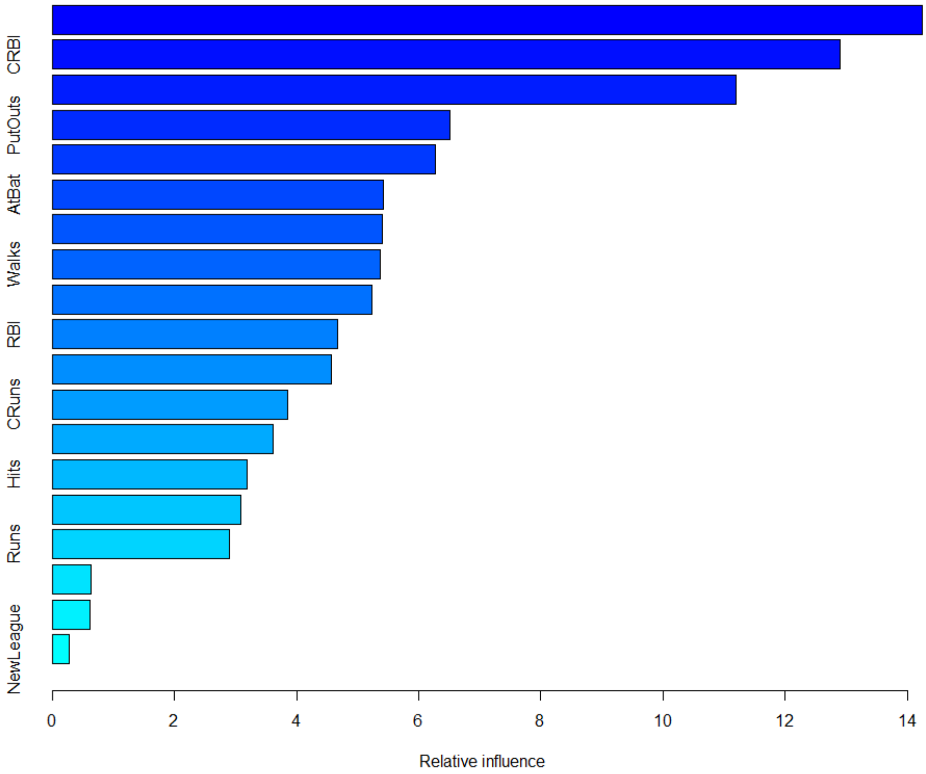
As you can see, the error rate is lowered after being pruned.

#Bagging



The figure above is variable importance plot of bagging. As you can see, The highest importance is CatBat, Chits, CRBI, ... , league and division. Bagging mean squared error is 0.2094662.

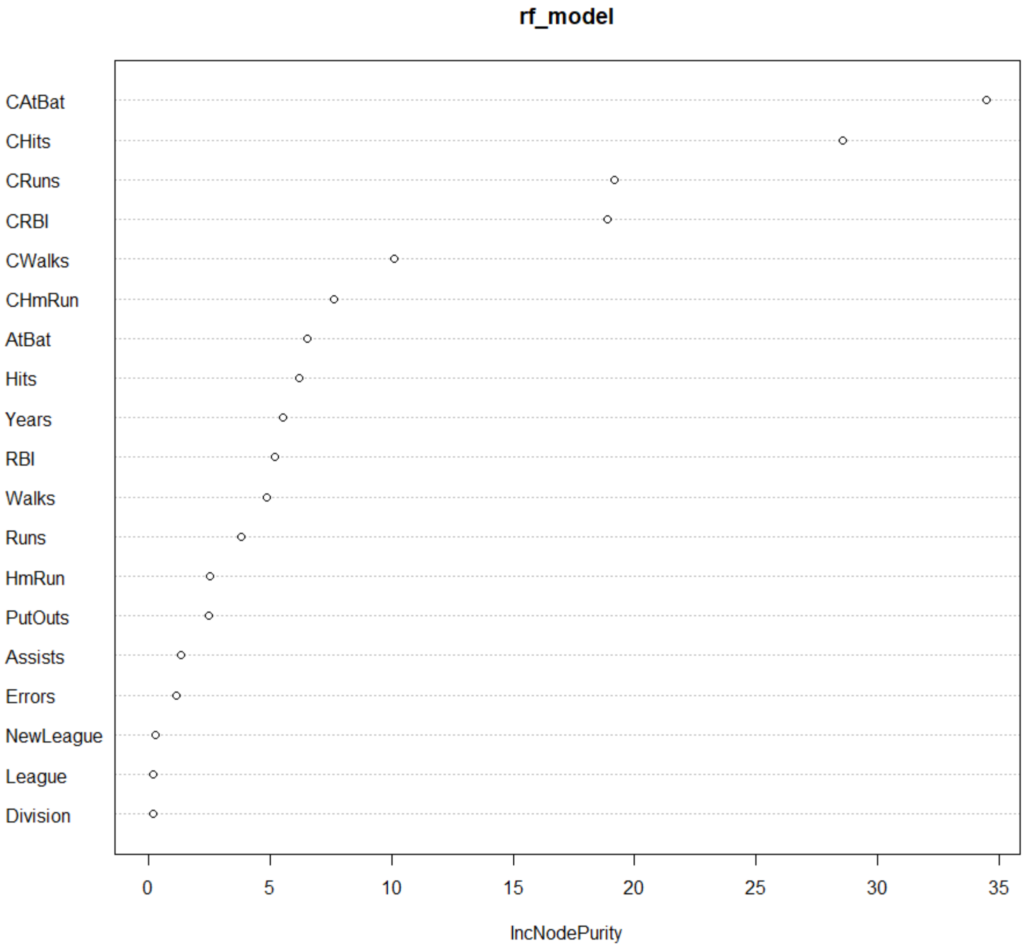
#Boosting




```
> summary(boost_model)
                var     rel.inf
CAtBat       CAtBat 26.3370252
CHits         CHits 10.2727188
Walks         Walks  7.8802829
PutOuts     PutOuts  7.6460360
CRBI           CRBI  5.3230236
CHmRun       CHmRun  5.0675629
RBI             RBI  4.1389311
HmRun         HmRun  3.9430323
AtBat         AtBat  3.9323211
Assists     Assists  3.8191267
Runs           Runs  3.6886404
Errors       Errors  3.6879829
CWalks       CWalks  3.2608391
CRuns         CRuns  3.2512290
Hits           Hits  3.1013060
Years         Years  2.6666242
League       League  0.8786551
Division   Division  0.5850133
NewLeague NewLeague  0.5196494
```

The figure above is variable importance plot of boosting. As you can see, The highest importance is CatBat, Chits, walks, … , league, division and newleague. Boosting mean squared error is 0.2206112.

#Random Forest




```
Call:
 randomForest(formula = Salary ~ ., data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 0.1879442
                    % Var explained: 75.82
```

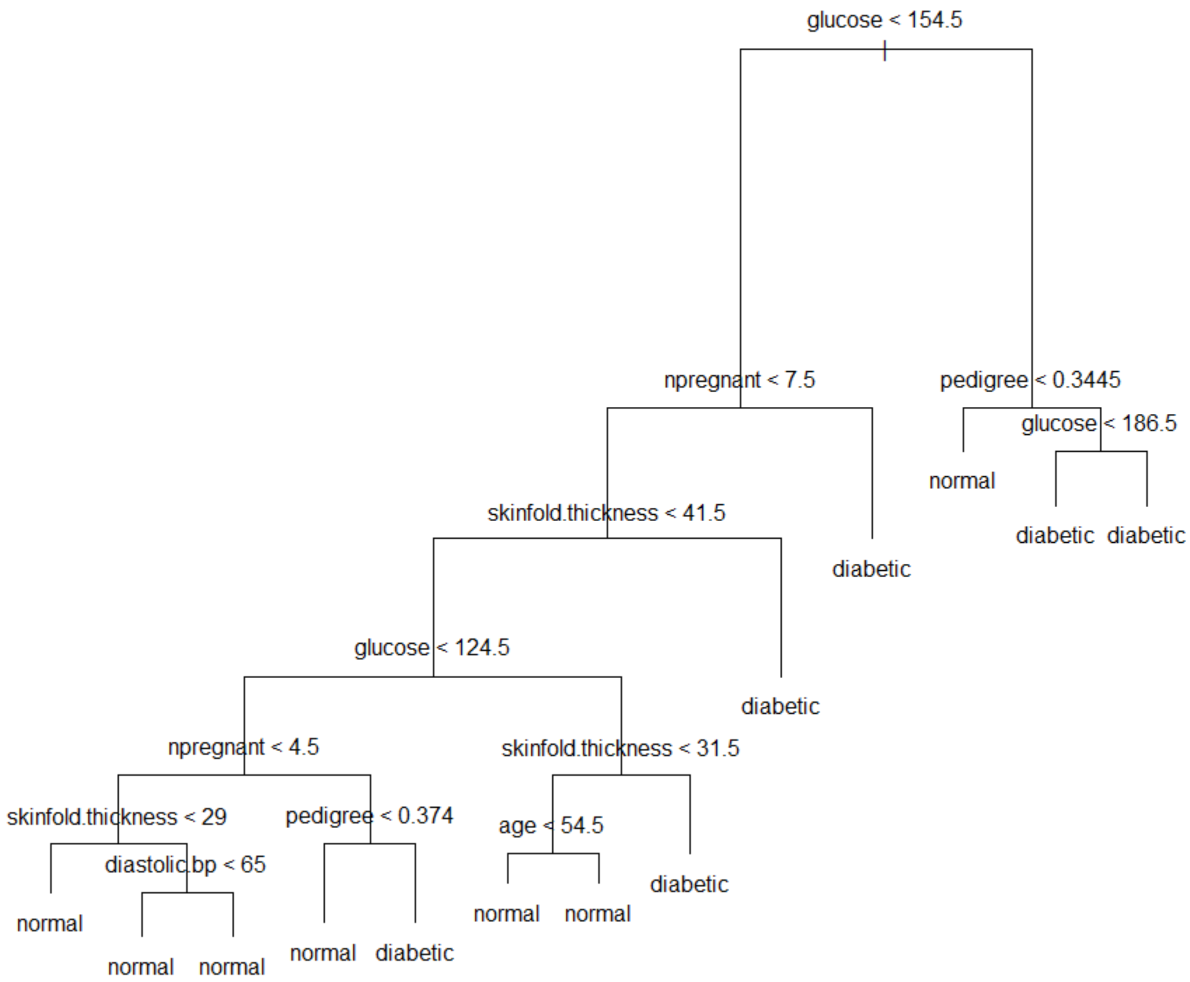
The figure above is variable importance plot of random forest. As you can see, the highest importance is CatBat, Chits, CRuns, …, newleague, league, and division. Random Forest mean squared error is 0.1996369.

Lastly, I did linear regression, and the mean squared error of linear regression is 0.3567222.

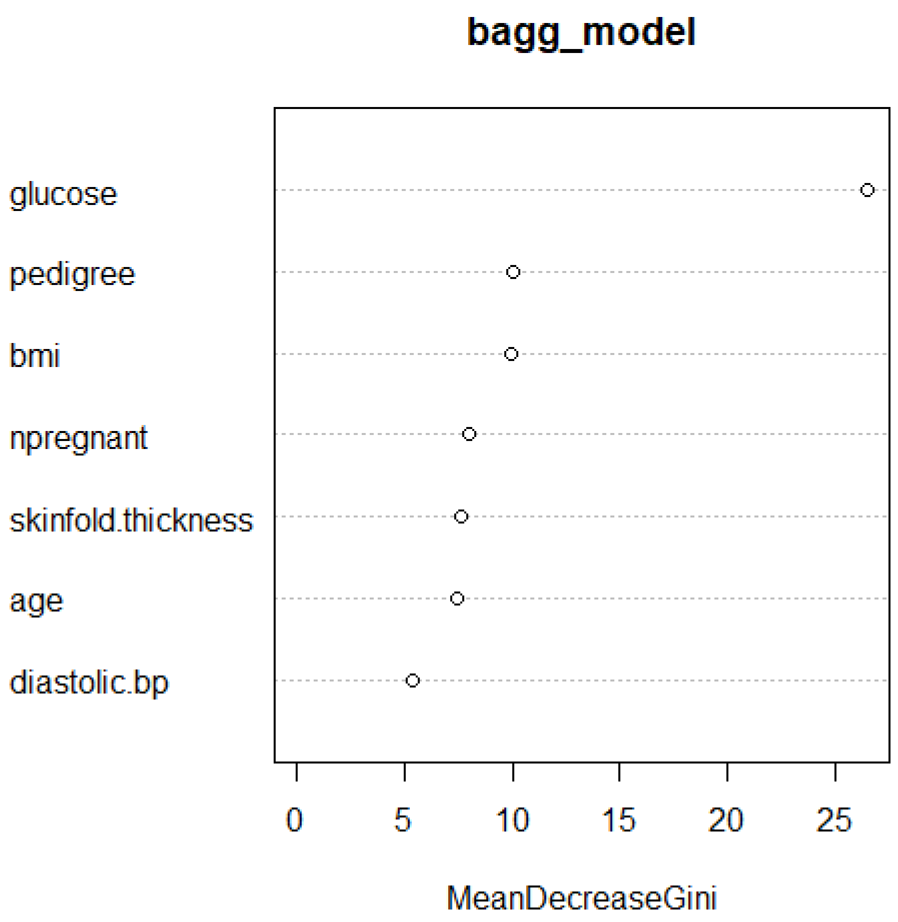
As a result, the highest error rate is linear regression, and the lowest error rate is pruned tree, which is 0.1542393. Linear regression had higher error rate compared to other methods, which means in this exercise, ensemble methods were more accurate then un-ensemble.

Advantage of committee machines is that methods like tree model is more visible and simpler than other methods. Moreover, it can combine multiple data easily by using ensemble, in order to get the optimal result. However, disadvantage is, since it is collaborating thousands of data, I also felt that the computer is lagging while doing the computing. Also, for decision tree, it overfits the data. Therefore, pruning is used to prevent this.

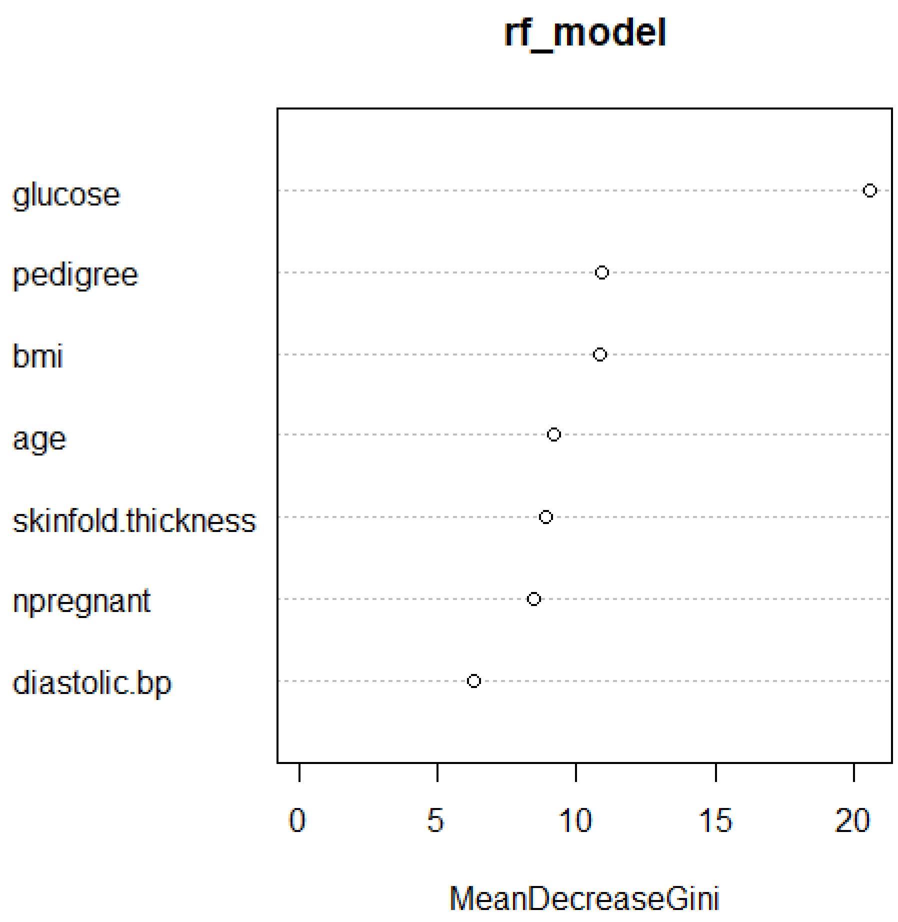
2.



This is the tree model for pima data, setting best = 13. It starts with glucose, then npregnant, pedigree, skinfold.thickness.



Variable Importance for bagging model is glucose, pedigree, bmi, ..., age and diastolic.bp.

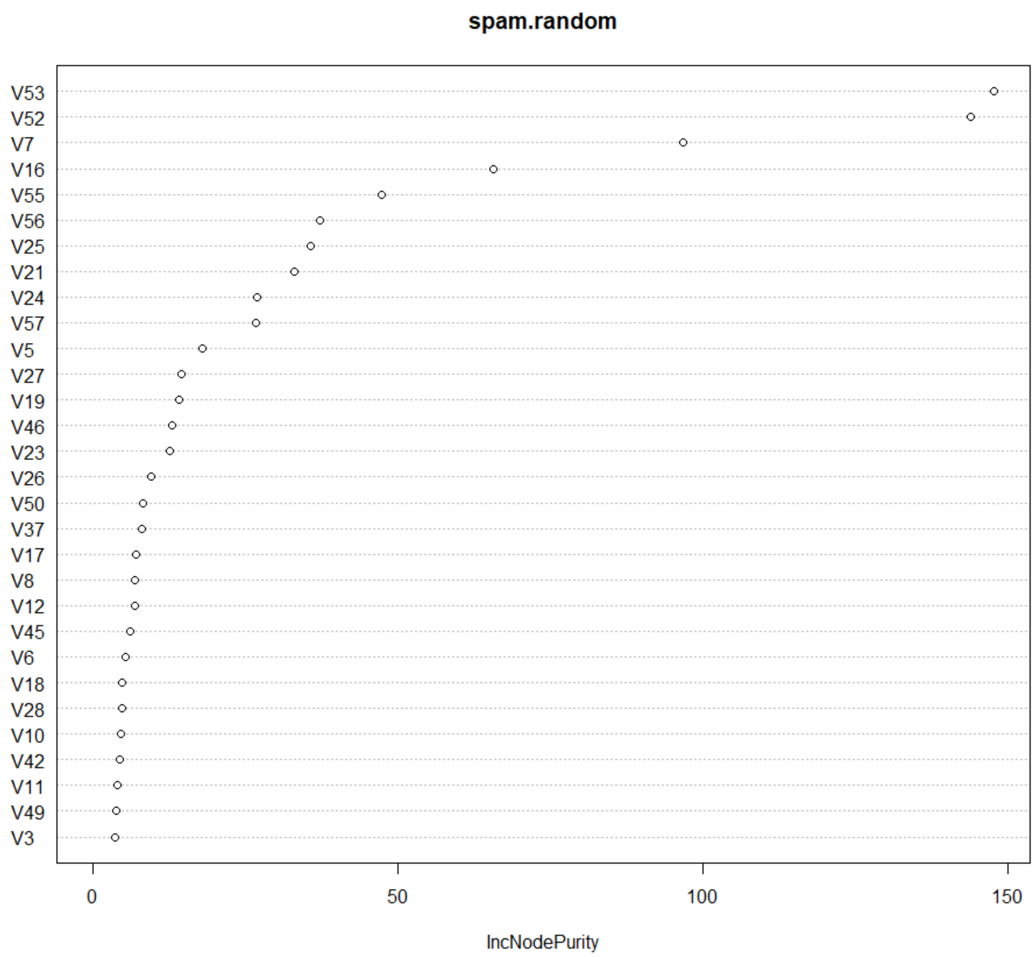


Variable Importance for random forest model is glucose, pedigree, bmi, ..., npregnant and diastolic.bp.

3.

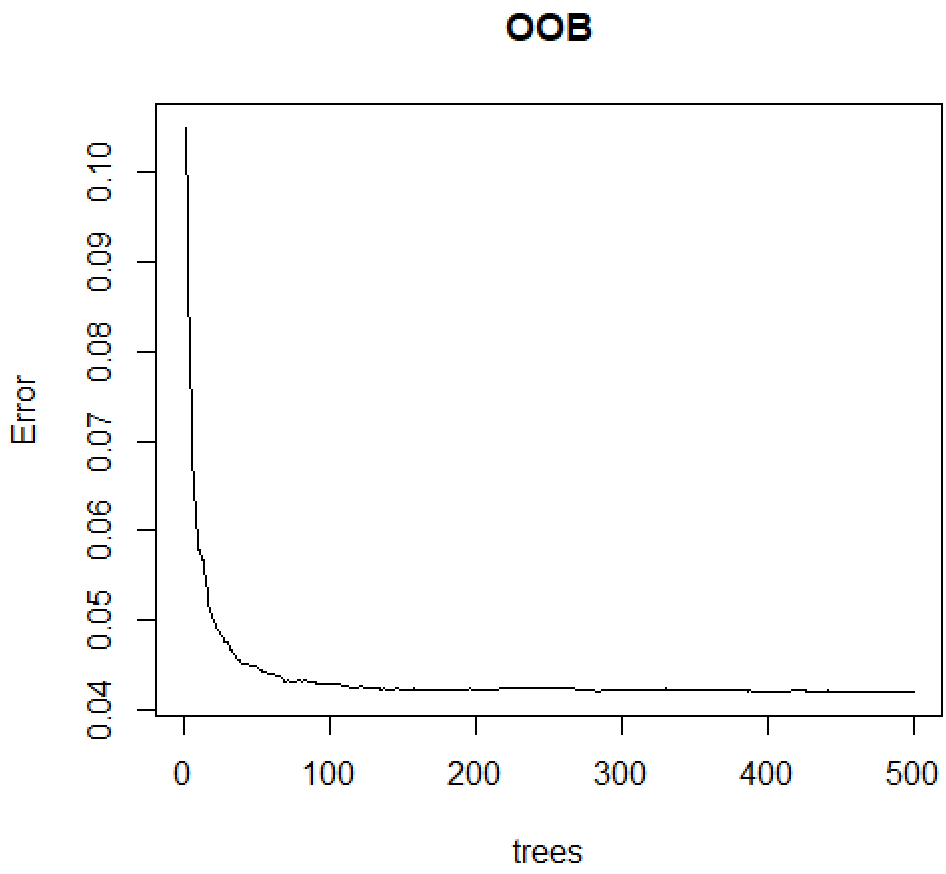
```
Call:
 randomForest(formula = V58 ~ ., data = train, n.tree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 19

          Mean of squared residuals: 0.04156554
                    % Var explained: 82.59
```



The figure above is the random forest. The mean squared error is 0.04318207.

	Out-of-bag	
Tree	MSE	%Var(y)
50	0.04311	18.06
100	0.04161	17.43
150	0.04084	17.11
200	0.04068	17.04
250	0.04075	17.07
300	0.04049	16.96
350	0.04032	16.89
400	0.0402	16.84
450	0.04018	16.83
500	0.04019	16.84



The figure above is the OOB. The data is sliced by 50 and as you can see, the MSE decreases dramatically between 0 ~ 100. After 150 trees, it becomes stable.