

Question 10.1

Using the same crime data set `uscrime.txt` as in Questions 8.2 and 9.1, find the best model you can using

- (a) a regression tree model, and
- (b) a random forest model.

In R, you can use the `tree` package or the `rpart` package, and the `randomForest` package. For each model, describe one or two qualitative takeaways you get from analyzing the results (i.e., don't just stop when you have a good model, but interpret it too).

Summary :

1. Build Simple tree with 7 nodes
 - a. $R^2 = 0.7244962$
 - b. $\text{Adj } R^2 = 0.6750468$
2. Build tree with Sample Data 1:1000
 - a. $R^2 = 0.3629629$
 - b. $\text{Adj } R^2 = 0.3340067$

Observation 1:

insample adjusted R^2 has dropped significantly from .67 to .334. However we are expecting cross validated R^2 to be less than the full model.

3. Build Random Forest Model

Number of Trees built from 500

RMSE is 278.7323 0.5181691 213.7942.

Number of Trees built from 400

RMSE is 278.8589, rsquared is 0.5187796, and MAE is 214.4396

Number of Trees built from 300

RMSE is 278.0788 , rSquared is 0.5212120 and MAE is 212.4929

Observation 2 :

small improvement in cross validated performance by limiting the number of trees grown to 400
And no improvement with trees built from 300 so use the previous model with 400 trees constructed.

RSCRIPT :

```
# ----- Data manipulation -----
```

```
#First, Read in the data
```

```
#
```

```
data_crime = read.table("uscrime.txt",sep = "",header=TRUE)
```

```
#
```

```
# optional check to make sure the data is read correctly
```

```
#
```

```
head(data_crime)
```

```
# It appears the data has a varying scale from hundreth and thousanth decimal place
```

```
# Load in library tree, perform tree regression
```

```
#and use cross validation to find the optimal size of tree
```

```
#
```

```
library(tree)
```

```
crimetree<-tree(Crime~.,data=data_crime)
```

```
# examine the tree we built first
```

```
#
```

```
summary(crimetree)
```

```
# there are actually 7 terminal nodes with Residual mean deviance of 47390
```

```
# Check if the splits make sense
```

```
plot(crimetree)
```

```
text(crimetree ,pretty =0)
```

```
crimetree
```

```
# From the result, Po1 is the most important variable and therefore used as the first split
```

```
#
```

```
# calculate R^2 and adjusted R^2 for the tree model
```

```
#
```

```
SST<-sum((data_crime$Crime-mean(data_crime$Crime))^2)
```

```
SSE<-sum((data_crime$Crime-predict(crimetree,newdata=data_crime))^2)
```

```
SST
```

```
SSE
```

```
r12<-1-SSE/SST
```

```
r12
```

```
#adjusted R^2 Value
```

```
n=nrow(data_crime)
```

```
k=7#number of terminal nodes
```

```
adjR12<-1-(1-r12)*(n-1)/(n-k-1)
```

```
adjR12
```

```

#
# build a better tree with less sum of squared residuals
#
sample(1:1000,1)
set.seed(555)
crvs<-cv.tree(crimetree, FUN = prune.tree, K=5)
crvs
plot(crvs$size ,crvs$dev ,type="b")

#
# 5 fold cross validation indicates that optimal size is 2 since dev of 6456140 is the lowest error.
# apply this limit to tree.control option if tree is bigger than 2 terminal nodes to prevent overfit.
# Try rebuilding the tree with mincut =23 which will give us 2 terminal nodes.
#

crimetree2<-tree(Crime~.,data=data_crime, control = tree.control(nobs = 47, mincut = 23))
crimetree2
plot(crimetree2)
text(crimetree2 ,pretty =0)

#
#we see that this give the treee with 2 terminal nodes or leaves.
#

summary(crimetree2)

#in-sample residual mean deviance is now 97410 which is twice as high as
#it was with all 7 leaves which makes sense

```

```
#since we now have a much simpler model that is not able to explain the training data  
#as well so residual deviance is higher.
```

```
#Calculate R^2 for the pruned model
```

```
#
```

```
SSE2<-sum((data_crime$Crime-predict(crimetree2,newdata=data_crime))^2)
```

```
SSE2
```

```
r22<-1-SSE2/SST
```

```
r22
```

```
n=nrow(data_crime)
```

```
k=2#number of terminal nodes
```

```
adjR22<-1-(1-r22)*(n-1)/(n-k-1)
```

```
adjR22
```

```
#
```

```
#insample adjusted R^2 has dropped significantly frkm .67 to .334.
```

```
#However we are expecting cross validated R^2 to be less than the full model.
```

```
#use the randomForest package to build a randomForest model
```

```
#using first the recommended number of variables to split on
```

```
# Try mtry =4 - sqrt( number of predictors)
```

```
#
```

```
library(randomForest)
```

```
rmf<-randomForest(Crime~., data=data_crime, mtry=4)
```

```
rmf
```

```
#
```

```
#we see here that squared is 43.13% which is higher than
```

```
#our selected tree model r squared of 0.3629626.
```

```
#Find the importance of the predictors using the Importance function and VarImpPlot
```

```
#
```

```
importance(rmf)
```

```
varImpPlot(rmf)
```

```
#
```

```
#Increase in node purity for Random Forest regression is just the increase
```

```
#in RSS averaged over all trees that occurs from splitting over that variable.
```

```
#Again we see that Po1 is the most important variable.
```

```
#Use cross validation to see if we can improve the model with different mtry levels
```

```
library(caret)
```

```
set.seed(555)
```

```
crvrm<-train(Crime~., data=data_crime,
```

```
            trControl=trainControl(method = 'cv', number = 5, savePredictions = TRUE),
```

```
            tuneGrid=expand.grid(mtry=1:15))
```

```
crvrm
```

```
SSErm<-sum((crvrm$pred[,2]-crvrm$pred[,1])^2)
```

```
SSErm
```

```
SSTrm<-sum((crvrm$pred[,2]-mean(crvrm$pred[,2]))^2)
```

```
SSTrm
```

```
r32<-1-(SSErm/SSTrm)
```

```
r32
```

```
#
```

```
#The mtry level that corresponds to the highest cross validated R^2 is 4 just as we hypothesized.
```

```
#It appears that the random forest model has outperformed the single decision tree pruned tree model.
```

```
#Cross validated r squared is 0.4435946.
```

#this result is not suprising because we are creating many tree to explain the data.

#try lowering the number of trees built from 500 to 400 and compare cross validated performance perhaps we can prevent overfit

#

set.seed(555)

crvrm2<-train(Crime~., data=data_crime,

trControl=trainControl(method = 'cv', number = 5, savePredictions = TRUE),

tuneGrid=expand.grid(mtry=1:15), ntree=400)

crvrm2

SSErm<-sum((crvrm2\$pred[,2]-crvrm2\$pred[,1])^2)

SSErm

SSTrm<-sum((crvrm2\$pred[,2]-mean(crvrm2\$pred[,2]))^2)

SSTrm

r42<-1-(SSErm/SSTrm)

r42

#

#small improvement in cross validated performance by limiting the number of trees grown to 400

#since RMSE is 278.8589, rsquared is 0.5187796, and MAE is 214.4396 and was 278.7323 0.5181691 213.7942.

#

#try lowering the number of trees built from 400 to 300

#and compare cross validated performance perhaps we can still prevent overfit

#

set.seed(555)

crvrm3<-train(Crime~., data=data_crime,

trControl=trainControl(method = 'cv', number = 5, savePredictions = TRUE),

tuneGrid=expand.grid(mtry=1:15), ntree=300)

```

crvm3
SSErm<-sum((crvm3$pred[,2]-crvm3$pred[,1])^2)
SSErm
SSTrm<-sum((crvm3$pred[,2]-mean(crvrm3$pred[,2]))^2)
SSTrm
r52<-1-(SSErm/SSTrm)
r52
#no improvement so use the previous model crvm2 instead with 400 trees constructed.

```

RSCRIPT with Results

```

> # ----- Data manipulation -----
>
> #First, Read in the data
> #
> data_crime = read.table("uscrime.txt",sep = "",header=TRUE)

```



```

>
> #
> # optional check to make sure the data is read correctly
> #
>
> head(data_crime)
  M So  Ed Po1 Po2  LF  M.F Pop  NW  U1 U2 Wealth Ineq  Prob  Time Crime
1 15.1 1  9.1  5.8  5.6 0.510 95.0 33 30.1 0.108 4.1  3940 26.1 0.084602 26.2011  791
2 14.3 0 11.3 10.3  9.5 0.583 101.2 13 10.2 0.096 3.6  5570 19.4 0.029599 25.2999 1635
3 14.2 1  8.9  4.5  4.4 0.533  96.9 18 21.9 0.094 3.3  3180 25.0 0.083401 24.3006  578
4 13.6 0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9  6730 16.7 0.015801 29.9012 1969
5 14.1 0 12.1 10.9 10.1 0.591  98.5 18  3.0 0.091 2.0  5780 17.4 0.041399 21.2998 1234
6 12.1 0 11.0 11.8 11.5 0.547  96.4 25  4.4 0.084 2.9  6890 12.6 0.034201 20.9995  682
>
> # It appears the data has a varying scale from hundreth and thousanth decimal place
>
>
> # Load in library tree, perform tree regression
> #and use cross validation to find the optimal size of tree
> #
> library(tree)
> crimetree<-tree(Crime~.,data=data_crime)
>
> # examine the tree we built first
>
> #
> summary(crimetree)

```

Regression tree:

```
tree(formula = Crime ~ ., data = data_crime)
```

Variables actually used in tree construction:

```
[1] "Po1" "Pop" "LF" "NW"
```

Number of terminal nodes: 7

Residual mean deviance: 47390 = 1896000 / 40

Distribution of residuals:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-573.900	-98.300	-1.545	0.000	110.600	490.100

```
>
```

```
> # there are actually 7 terminal nodes with Residual mean deviance of 47390
```

```
>
```

```
> # Check if the splits make sense
```

```
> plot(crimetree)
```

```
> text(crimetree ,pretty =0)
```

```
> crimetree
```

node), split, n, deviance, yval

* denotes terminal node

1) root 47 6881000 905.1

2) Po1 < 7.65 23 779200 669.6

4) Pop < 22.5 12 243800 550.5

8) LF < 0.5675 7 48520 466.9 *

9) LF > 0.5675 5 77760 667.6 *

5) Pop > 22.5 11 179500 799.5 *

3) Po1 > 7.65 24 3604000 1131.0

6) NW < 7.65 10 557600 886.9

12) Pop < 21.5 5 146400 1049.0 *

13) Pop > 21.5 5 147800 724.6 *

7) NW > 7.65 14 2027000 1305.0

14) Po1 < 9.65 6 170800 1041.0 *

15) Po1 > 9.65 8 1125000 1503.0 *

>

> # From the result, Po1 is the most important variable and therefore used as the first split

>

> #

> # calculate R^2 and adjusted R^2 for the tree model

> #

> SST<-sum((data_crime\$Crime-mean(data_crime\$Crime))^2)

> SSE<-sum((data_crime\$Crime-predict(crimetree,newdata=data_crime))^2)

> SST

[1] 6880928

> SSE

[1] 1895722

> r12<-1-SSE/SST

> r12

[1] 0.7244962

>

> #adjusted R^2 Value

>

> n=nrow(data_crime)

> k=7#number of terminal nodes

> adjR12<-1-(1-r12)*(n-1)/(n-k-1)

> adjR12

[1] 0.6750468

>

>

> #

> # build a better tree with less sum of squared residuals

```

> #
> sample(1:1000,1)
[1] 940
> set.seed(555)
> crvs<-cv.tree(crimetree, FUN = prune.tree, K=5)
> crvs
$size
[1] 7 6 5 4 3 2 1

$dev
[1] 6985079 6920884 6754652 6790966 6469270 6456140 7308121

$k
[1] -Inf 117534.9 263412.9 355961.8 731412.1 1019362.7 2497521.7

$method
[1] "deviance"

attr("class")
[1] "prune" "tree.sequence"
> plot(crvs$size ,crvs$dev ,type="b")
>
> #
> # 5 fold cross validation indicates that optimal size is 2 since dev of 6456140 is the lowest error.
> # apply this limit to tree.control option if tree is bigger than 2 terminal nodes to prevent overfit.
> # Try rebuilding the tree with mincut =23 which will give us 2 terminal nodes.
> #
>
> crimetree2<-tree(Crime~.,data=data_crime, control = tree.control(nobs = 47, mincut = 23))

```

```
> crimetree2
```

```
node), split, n, deviance, yval
```

```
* denotes terminal node
```

```
1) root 47 6881000 905.1
```

```
2) Po1 < 7.65 23 779200 669.6 *
```

```
3) Po1 > 7.65 24 3604000 1131.0 *
```

```
> plot(crimetree2)
```

```
> text(crimetree2 ,pretty =0)
```

```
>
```

```
>
```

```
> #
```

```
> #we see that this give the treee with 2 terminal nodes or leaves.
```

```
> #
```

```
>
```

```
> summary(crimetree2)
```

Regression tree:

```
tree(formula = Crime ~ ., data = data_crime, control = tree.control(nobs = 47,  
  mincut = 23))
```

Variables actually used in tree construction:

```
[1] "Po1"
```

Number of terminal nodes: 2

Residual mean deviance: 97410 = 4383000 / 45

Distribution of residuals:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-622.800	-193.200	-5.609	0.000	147.300	862.200

```
>
```

```
> #in-sample residual mean deviance is now 97410 which is twice as high as
```

```

> #it was with all 7 leaves which makes sense

> #since we now have a much simpler model that is not able to explain the training data

> #as well so residual deviance is higher.

>

> #Calculate R^2 for the pruned model

> #

> SSE2<-sum((data_crime$Crime-predict(crimetree2,newdata=data_crime))^2)

> SSE2

[1] 4383406

> r22<-1-SSE2/SST

> r22

[1] 0.3629629

> n=nrow(data_crime)

> k=2#number of terminal nodes

> adjR22<-1-(1-r22)*(n-1)/(n-k-1)

> adjR22

[1] 0.3340067

> #

> #insample adjusted R^2 has dropped significantly frkm .67 to .334.

> #However we are expecting cross validated R^2 to be less than the full model.

> #use the randomForest package to build a randomForest model

> #using first the recommended number of variables to split on

> # Try mtry =4 - sqrt( number of predictors)

>

> #

> library(randomForest)

> rmf<-randomForest(Crime~., data=data_crime, mtry=4)

> rmf

```

Call:

```
randomForest(formula = Crime ~ ., data = data_crime, mtry = 4)
```

Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 4

Mean of squared residuals: 83266.41

% Var explained: 43.13

```
> #
```

```
> #we see here that squared is 43.13% which is higher than
```

```
> #our selected tree model r squared of 0.3629626.
```

```
> #Find the importance of the predictors using the Importance function and VarImpPlot
```

```
> #
```

```
>
```

```
> importance(rmf)
```

Error in UseMethod("importance") :

no applicable method for 'importance' applied to an object of class "c('randomForest.formula', 'randomForest')"

```
> #First, Read in the data
```

```
> #
```

```
> data_crime = read.table("uscrime.txt", sep = "", header=TRUE)
```

```
>
```

```
> #
```

```
> # optional check to make sure the data is read correctly
```

```
> #
```

```
>
```

```
> head(data_crime)
```

	M So	Ed	Po1	Po2	LF	M.F Pop	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime		
1	15.1	1	9.1	5.8	5.6	0.510	95.0	33	30.1	0.108	4.1	3940	26.1	0.084602	26.2011	791

```

2 14.3 0 11.3 10.3 9.5 0.583 101.2 13 10.2 0.096 3.6 5570 19.4 0.029599 25.2999 1635
3 14.2 1 8.9 4.5 4.4 0.533 96.9 18 21.9 0.094 3.3 3180 25.0 0.083401 24.3006 578
4 13.6 0 12.1 14.9 14.1 0.577 99.4 157 8.0 0.102 3.9 6730 16.7 0.015801 29.9012 1969
5 14.1 0 12.1 10.9 10.1 0.591 98.5 18 3.0 0.091 2.0 5780 17.4 0.041399 21.2998 1234
6 12.1 0 11.0 11.8 11.5 0.547 96.4 25 4.4 0.084 2.9 6890 12.6 0.034201 20.9995 682
>

```

```

> # It appears the data has a varying scale from hundreth and thousanth decimal place
>
>

```

```

> # Load in library tree, perform tree regression
> #and use cross validation to find the optimal size of tree
> #
> library(tree)
> crimetree<-tree(Crime~.,data=data_crime)
>
> # examine the tree we built first
>
> #
> summary(crimetree)

```

Regression tree:

```
tree(formula = Crime ~ ., data = data_crime)
```

Variables actually used in tree construction:

```
[1] "Po1" "Pop" "LF" "NW"
```

Number of terminal nodes: 7

Residual mean deviance: 47390 = 1896000 / 40

Distribution of residuals:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-573.900	-98.300	-1.545	0.000	110.600	490.100


```

>
> # there are actually 7 terminal nodes with Residual mean deviance of 47390
>
> # Check if the splits make sense
> plot(crimetree)
> text(crimetree ,pretty =0)
> crimetree
node), split, n, deviance, yval
    * denotes terminal node

1) root 47 6881000 905.1
  2) Po1 < 7.65 23 779200 669.6
    4) Pop < 22.5 12 243800 550.5
      8) LF < 0.5675 7 48520 466.9 *
      9) LF > 0.5675 5 77760 667.6 *
    5) Pop > 22.5 11 179500 799.5 *
  3) Po1 > 7.65 24 3604000 1131.0
    6) NW < 7.65 10 557600 886.9
      12) Pop < 21.5 5 146400 1049.0 *
      13) Pop > 21.5 5 147800 724.6 *
    7) NW > 7.65 14 2027000 1305.0
      14) Po1 < 9.65 6 170800 1041.0 *
      15) Po1 > 9.65 8 1125000 1503.0 *
>
> # From the result, Po1 is the most important variable and therefore used as the first split
>
> #
> # calculate R^2 and adjusted R^2 for the tree model
> #

```

```

> SST<-sum((data_crime$Crime-mean(data_crime$Crime))^2)
> SSE<-sum((data_crime$Crime-predict(crimetree,newdata=data_crime))^2)
> SST
[1] 6880928
> SSE
[1] 1895722
> r12<-1-SSE/SST
> r12
[1] 0.7244962
>
> #adjusted R^2 Value
>
> n=nrow(data_crime)
> k=7#number of terminal nodes
> adjR12<-1-(1-r12)*(n-1)/(n-k-1)
> adjR12
[1] 0.6750468
>
>
> #
> # build a better tree with less sum of squared residuals
> #
> sample(1:1000,1)
[1] 328
> set.seed(555)
> crvs<-cv.tree(crimetree, FUN = prune.tree, K=5)
> crvs
$size
[1] 7 6 5 4 3 2 1

```

```
$dev
```

```
[1] 6985079 6920884 6754652 6790966 6469270 6456140 7308121
```

```
$k
```

```
[1] -Inf 117534.9 263412.9 355961.8 731412.1 1019362.7 2497521.7
```

```
$method
```

```
[1] "deviance"
```

```
attr("class")
```

```
[1] "prune" "tree.sequence"
```

```
> plot(crvs$size ,crvs$dev ,type="b")
```

```
>
```

```
> #
```

```
> # 5 fold cross validation indicates that optimal size is 2 since dev of 6456140 is the lowest error.
```

```
> # apply this limit to tree.control option if tree is bigger than 2 terminal nodes to prevent overfit.
```

```
> # Try rebuilding the tree with mincut =23 which will give us 2 terminal nodes.
```

```
> #
```

```
>
```

```
> crimetree2<-tree(Crime~.,data=data_crime, control = tree.control(nobs = 47, mincut = 23))
```

```
> crimetree2
```

```
node), split, n, deviance, yval
```

```
  * denotes terminal node
```

```
1) root 47 6881000 905.1
```

```
 2) Po1 < 7.65 23 779200 669.6 *
```

```
 3) Po1 > 7.65 24 3604000 1131.0 *
```

```
> plot(crimetree2)
```

```
> text(crimetree2 ,pretty =0)
```

```
>
```

```
>
```

```
> #
```

```
> #we see that this give the treee with 2 terminal nodes or leaves.
```

```
> #
```

```
>
```

```
> summary(crimetree2)
```

Regression tree:

```
tree(formula = Crime ~ ., data = data_crime, control = tree.control(nobs = 47,  
  mincut = 23))
```

Variables actually used in tree construction:

```
[1] "Po1"
```

Number of terminal nodes: 2

Residual mean deviance: 97410 = 4383000 / 45

Distribution of residuals:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-622.800	-193.200	-5.609	0.000	147.300	862.200

```
>
```

```
> #in-sample residual mean deviance is now 97410 which is twice as high as
```

```
> #it was with all 7 leaves which makes sense
```

```
> #since we now have a much simpler model that is not able to explain the training data
```

```
> #as well so residual deviance is higher.
```

```
>
```

```
> #Calculate R^2 for the pruned model
```

```
> #
```

```
> SSE2<-sum((data_crime$Crime-predict(crimetree2,newdata=data_crime))^2)
```

```
> SSE2
```

```

[1] 4383406
> r22<-1-SSE2/SST
> r22
[1] 0.3629629
> n=nrow(data_crime)
> k=2#number of terminal nodes
> adjR22<-1-(1-r22)*(n-1)/(n-k-1)
> adjR22
[1] 0.3340067
> #
> #insample adjusted R^2 has dropped significantly frkm .67 to .334.
> #However we are expecting cross validated R^2 to be less than the full model.
> #use the randomForest package to build a randomForest model
> #using first the recommended number of variables to split on
> # Try mtry =4 - sqrt( number of predictors)
>
> #
> library(randomForest)
> rmf<-randomForest(Crime~., data=data_crime, mtry=4)
> rmf

```

Call:

```
randomForest(formula = Crime ~ ., data = data_crime, mtry = 4)
```

Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 4

Mean of squared residuals: 83266.41

% Var explained: 43.13

```

> #
> #we see here that squared is 43.13% which is higher than
> #our selected tree model r squared of 0.3629626.
> #Find the importance of the predictors using the Importance function and VarImpPlot
> #
>
> #importance(rmf)
> varImpPlot(rmf)
> #
>
> #Increase in node purity for Random Forest regression is just the increase
> #in RSS averaged over all trees that occurs from splitting over that variable.
> #Again we see that Po1 is the most important variable.
>
> #Use cross validation to see if we can improve the model with different mtry levels
>
> library(caret)
> set.seed(555)
> crvrm<-train(Crime~., data=data_crime,
+   trControl=trainControl(method = 'cv', number = 5, savePredictions = TRUE),
+   tuneGrid=expand.grid(mtry=1:15))
> crvrm
Random Forest

47 samples
15 predictors

No pre-processing
Resampling: Cross-Validated (5 fold)

```

Summary of sample sizes: 36, 38, 38, 38, 38

Resampling results across tuning parameters:

	mtry	RMSE	Rsquared	MAE
1	282.2148	0.5109789	232.2553	
2	270.4879	0.5399055	222.0057	
3	267.9380	0.5543635	214.3942	
4	268.5870	0.5537980	212.0183	
5	270.1196	0.5464426	213.2321	
6	274.4155	0.5359920	213.7011	
7	274.1235	0.5295017	211.8640	
8	272.5812	0.5394876	212.9723	
9	277.0303	0.5270595	214.4619	
10	279.2353	0.5187866	214.7263	
11	275.0933	0.5389526	212.4839	
12	276.8314	0.5332234	212.7497	
13	279.1010	0.5258992	213.4738	
14	275.2154	0.5376047	210.8170	
15	278.7323	0.5181691	213.7942	

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was mtry = 3.

>

```
> SSErm<-sum((crvrm$pred[,2]-crvrm$pred[,1])^2)
```

```
> SSErm
```

```
[1] 65521786
```

```
> SSTrm<-sum((crvrm$pred[,2]-mean(crvrm$pred[,2]))^2)
```

```
> SSTrm
```

```
[1] 103213915
```

```

> r32<-1-(SSErm/SSTrm)

> r32

[1] 0.3651846

>

> #

> #The mtry level that corresponds to the highest cross validated R^2 is 4 just as we hypothesized.

> #It appears that the random forest model has outperformed the single decision tree pruned tree
model.

> #Cross validated r squared is 0.4435946.

> #this result is not suprising because we are creating many tree to explain the data.

>

> #try lowering the number of trees built from 500 to 400 and compare cross validated performance
perhaps we can prevent overfit

> #

> set.seed(555)

> crvrm2<-train(Crime~., data=data_crime,
+      trControl=trainControl(method = 'cv', number = 5, savePredictions = TRUE),
+      tuneGrid=expand.grid(mtry=1:15), ntree=400)

> crvrm2

Random Forest

47 samples

15 predictors

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 36, 38, 38, 38, 38

Resampling results across tuning parameters:

```


	mtry	RMSE	Rsquared	MAE
1	281.3281	0.5107555	231.0260	
2	271.2270	0.5397441	222.5078	
3	267.8948	0.5577242	213.9204	
4	269.2855	0.5510344	212.6778	
5	270.2207	0.5441216	213.7219	
6	273.4922	0.5414202	213.5780	
7	276.6178	0.5247428	213.2216	
8	273.3596	0.5407124	213.7767	
9	277.1620	0.5291552	215.0050	
10	278.3759	0.5275974	214.2065	
11	275.1091	0.5436061	211.9783	
12	279.2716	0.5226313	213.9776	
13	278.5464	0.5326273	213.3096	
14	273.9128	0.5434945	210.1580	
15	278.8589	0.5187796	214.4396	

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was mtry = 3.

```
> SSErm<-sum((crvrm2$pred[,2]-crvrm2$pred[,1])^2)
> SSErm
[1] 65591805
> SSTrm<-sum((crvrm2$pred[,2]-mean(crvrm2$pred[,2]))^2)
> SSTrm
[1] 103213915
> r42<-1-(SSErm/SSTrm)
> r42
[1] 0.3645062
> #
```

```
> #small improvement in cross validated performance by limiting the number of trees grown to 400
> #since RMSE is 278.8589, rsquared is 0.5187796, and MAE is 214.4396 and was 278.7323 0.5181691
213.7942.
```

```
>
```

```
> #
```

```
> #try lowering the number of trees built from 400 to 300
```

```
> #and compare cross validated performance perhaps we can still prevent overfit
```

```
> #
```

```
> set.seed(555)
```

```
> crvm3<-train(Crime~., data=data_crime,
```

```
+      trControl=trainControl(method = 'cv', number = 5, savePredictions = TRUE),
```

```
+      tuneGrid=expand.grid(mtry=1:15), ntree=300)
```

```
> crvm3
```

Random Forest

47 samples

15 predictors

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 36, 38, 38, 38, 38

Resampling results across tuning parameters:

mtry	RMSE	Rsquared	MAE
1	281.1145	0.5120530	230.1269
2	269.3940	0.5389937	220.3498
3	267.6460	0.5604548	213.8100
4	268.5961	0.5506388	212.5859
5	270.3770	0.5426735	212.7279

6	273.8725	0.5389338	213.5581
7	274.4479	0.5294671	211.6421
8	273.9322	0.5412890	215.0722
9	275.6087	0.5401420	213.6776
10	278.5269	0.5232155	213.8719
11	274.7225	0.5513087	211.6254
12	283.1814	0.5060676	217.5156
13	278.1487	0.5325734	213.6487
14	273.2436	0.5471932	209.0589
15	278.0788	0.5212120	212.4929

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was mtry = 3.

```
> SSErm<-sum((cvrm3$pred[,2]-cvrm3$pred[,1])^2)
```

```
> SSErm
```

```
[1] 65635355
```

```
> SSTrm<-sum((cvrm3$pred[,2]-mean(cvrm3$pred[,2]))^2)
```

```
> SSTrm
```

```
[1] 103213915
```

```
> r52<-1-(SSErm/SSTrm)
```

```
> r52
```

```
[1] 0.3640842
```

```
> #no improvement so use the previous model cvrm2 instead with 400 trees constructed.
```