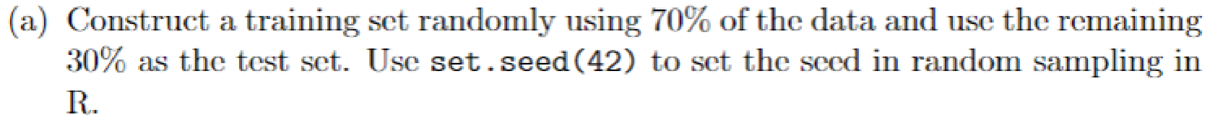
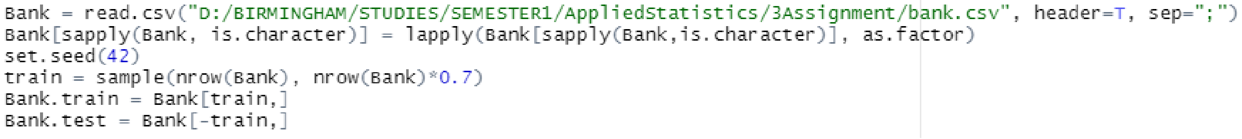
Applied Statistics

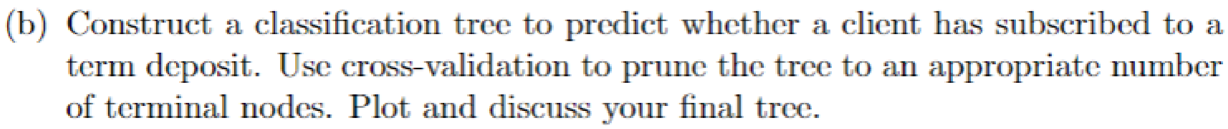
Assignment 3



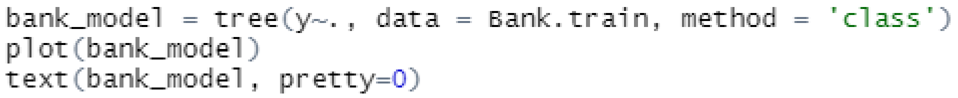
(a) Construct a training set randomly using 70% of the data and use the remaining 30% as the test set. Use `set.seed(42)` to set the seed in random sampling in R.



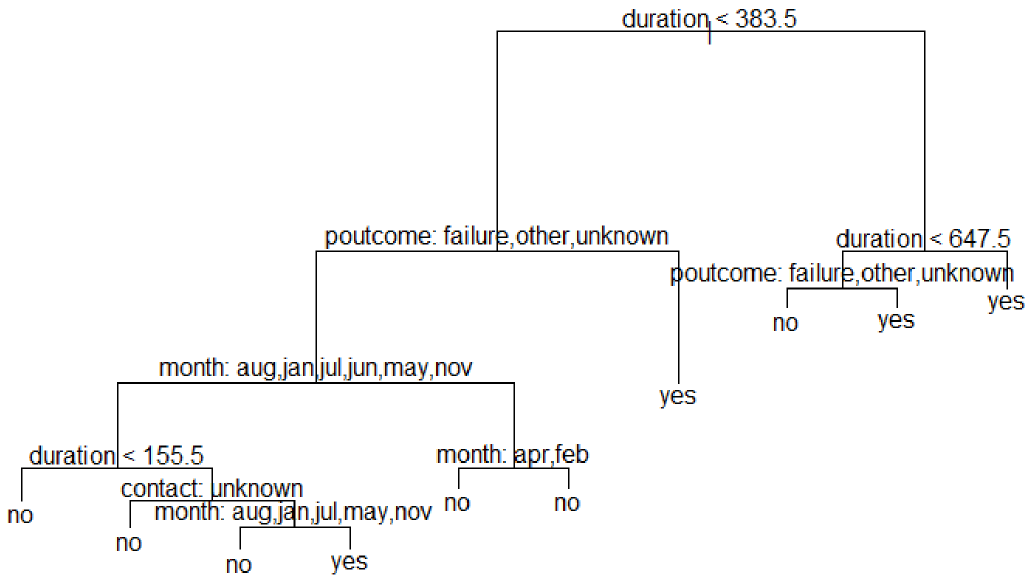
```
Bank = read.csv("D:/BIRMINGHAM/STUDIES/SEMESTER1/AppliedStatistics/3Assignment/bank.csv", header=T, sep=";")
Bank[sapply(Bank, is.character)] = lapply(Bank[sapply(Bank,is.character)], as.factor)
set.seed(42)
train = sample(nrow(Bank), nrow(Bank)*0.7)
Bank.train = Bank[train,]
Bank.test = Bank[-train,]
```



(b) Construct a classification tree to predict whether a client has subscribed to a term deposit. Use cross-validation to prune the tree to an appropriate number of terminal nodes. Plot and discuss your final tree.



```
bank_model = tree(y~., data = Bank.train, method = 'class')
plot(bank_model)
text(bank_model, pretty=0)
```



duration < 383.5
poutcome: failure,other,unknown
duration < 647.5
poutcome: failure,other,unknown
no
yes
yes
month: aug,jan,jul,jun,may,nov
yes
duration < 155.5
month: apr,feb
contact: unknown
month: aug,jan,jul,may,nov
no
no
no
no
no
no
yes

Figure 1. Initial classification tree model

```
bank_model_cv = cv.tree(bank_model, FUN = prune.misclass)
plot(bank_model_cv$size,bank_model_cv$dev,type="b") #choose 5
```

We are using cross-validation to find an appropriate number of terminal nodes.

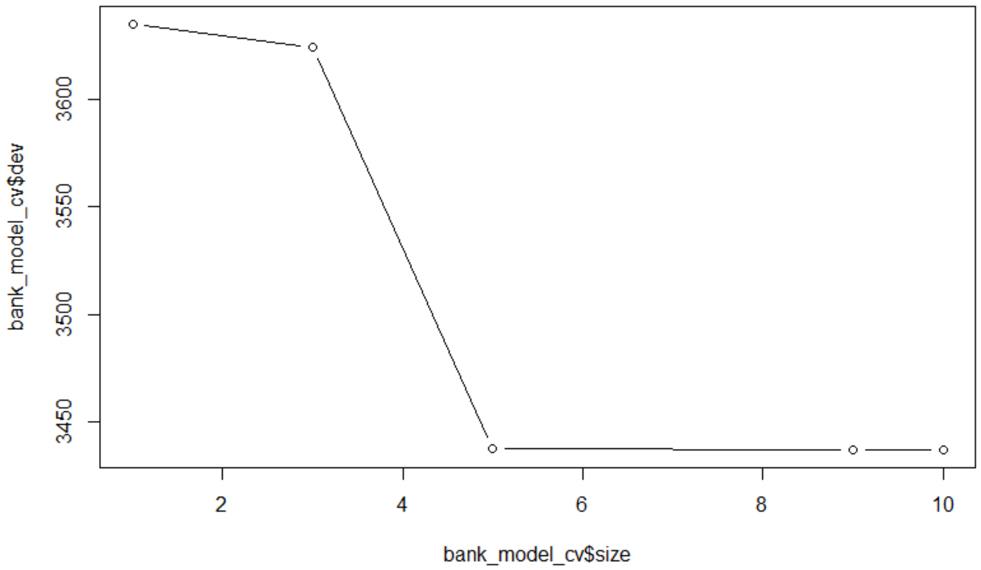
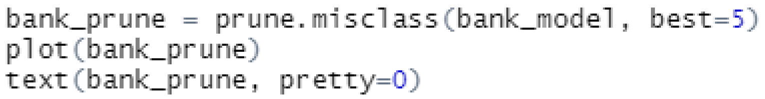


Figure 2. Size of the tree of model deviation

Choose 5 terminal nodes as the most appropriate. After 5, the deviation of the model does not change significantly.



```
bank_prune = prune.misclass(bank_model, best=5)
plot(bank_prune)
text(bank_prune, pretty=0)
```

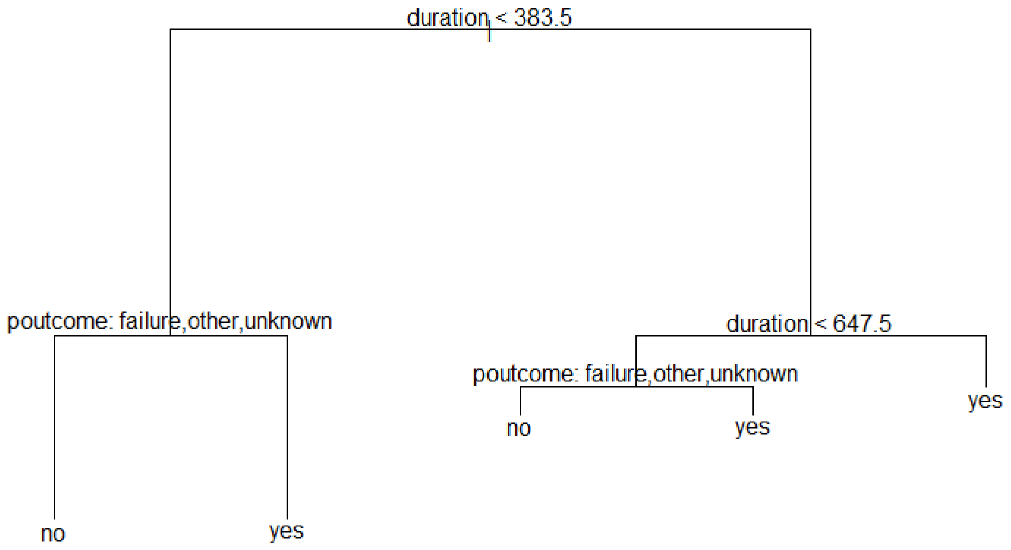
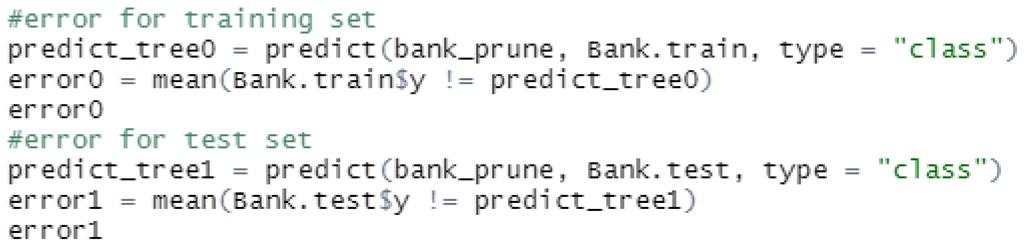

duration < 383.5
poutcome: failure,other,unknown
no
yes
duration < 647.5
poutcome: failure,other,unknown
no
yes
yes

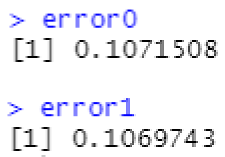
Figure 3. Final tree

The final tree consists of two variables: duration and poutcome. According to the tree, we can classify the clients relying on the two factors.

(c) Compute the training and test misclassification errors.



```
#error for training set
predict_tree0 = predict(bank_prune, Bank.train, type = "class")
error0 = mean(Bank.train$y != predict_tree0)
error0
#error for test set
predict_tree1 = predict(bank_prune, Bank.test, type = "class")
error1 = mean(Bank.test$y != predict_tree1)
error1
```

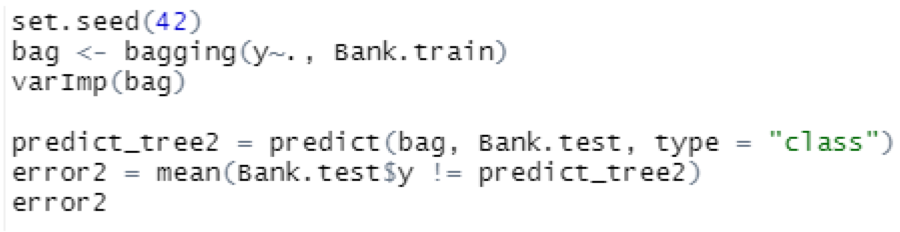


```
> error0
[1] 0.1071508

> error1
[1] 0.1069743
```

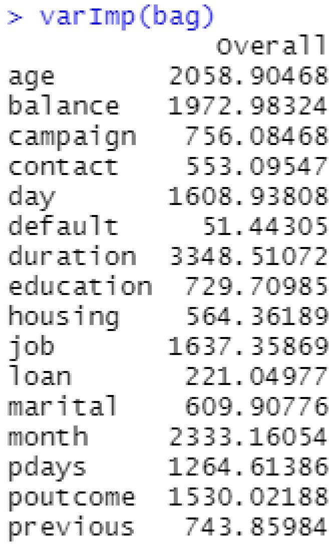
We got quite similar errors for training and test sets.

(d) Use the bagging approach in order to analyze this data. What test error do you obtain? Which variables are most important?

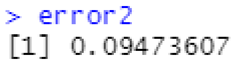


```
set.seed(42)
bag <- bagging(y~., Bank.train)
varImp(bag)

predict_tree2 = predict(bag, Bank.test, type = "class")
error2 = mean(Bank.test$y != predict_tree2)
error2
```



```
> varImp(bag)
             Overall
age       2058.90468
balance   1972.98324
campaign   756.08468
contact    553.09547
day       1608.93808
default     51.44305
duration  3348.51072
education  729.70985
housing    564.36189
job       1637.35869
loan       221.04977
marital    609.90776
month     2333.16054
pdays     1264.61386
poutcome  1530.02188
previous   743.85984
```



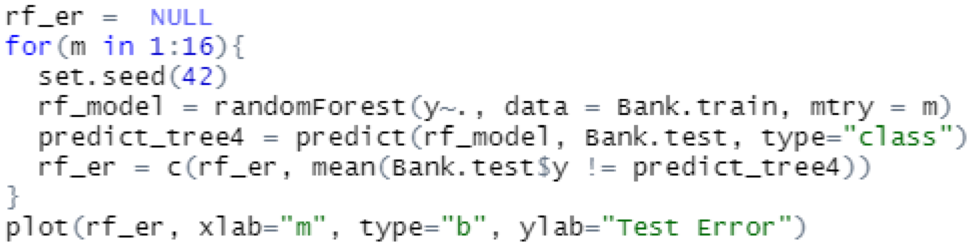
```
> error2
[1] 0.09473607
```

Bagging gives us less error than the classification tree model.

The most significant variable is duration. The variable poutcome is not as important in this method as in the classification tree.

(e) Use random forests to analyze this data. What test error rate do you obtain? Which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

Find the most appropriate number of variables considered at each slip (m).



```
rf_er =  NULL
for(m in 1:16){
  set.seed(42)
  rf_model = randomForest(y~., data = Bank.train, mtry = m)
  predict_tree4 = predict(rf_model, Bank.test, type="class")
  rf_er = c(rf_er, mean(Bank.test$y != predict_tree4))
}
plot(rf_er, xlab="m", type="b", ylab="Test Error")
```

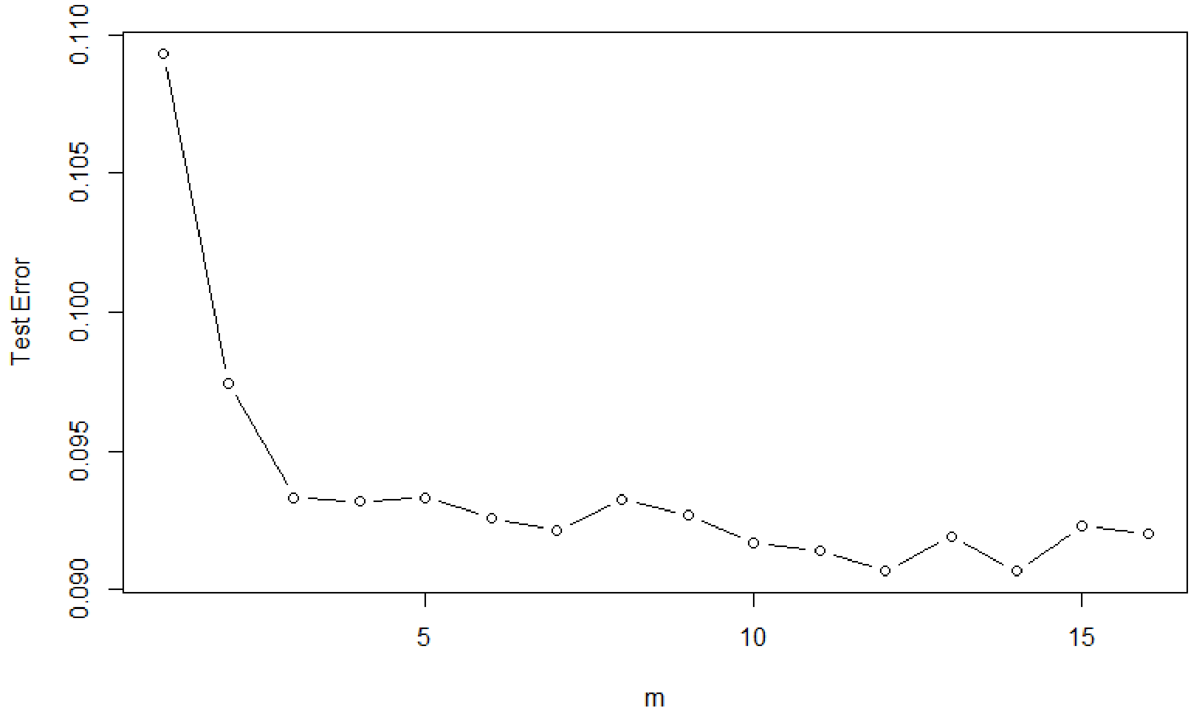
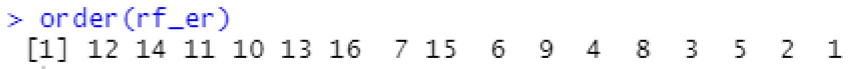


Figure 4. The effect of m on the error rate obtained

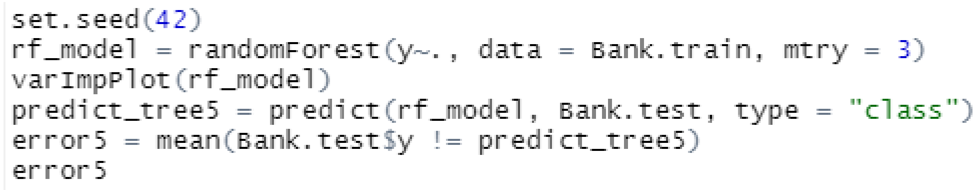
Figure 5 shows that when m increases, the test error decreases. But after m = 3, the decrease is not significant.



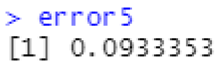
```
> order(rf_er)
 [1] 12 14 11 10 13 16  7 15  6  9  4  8  3  5  2  1
```

The minimum error is when m = 12, and the maximum error is when m = 1.

Take m = 3 and fit the model.



```
set.seed(42)
rf_model = randomForest(y~., data = Bank.train, mtry = 3)
varImpPlot(rf_model)
predict_tree5 = predict(rf_model, Bank.test, type = "class")
error5 = mean(Bank.test$y != predict_tree5)
error5
```



```
> error5
[1] 0.0933353
```

The error is less than the bagging approach error.

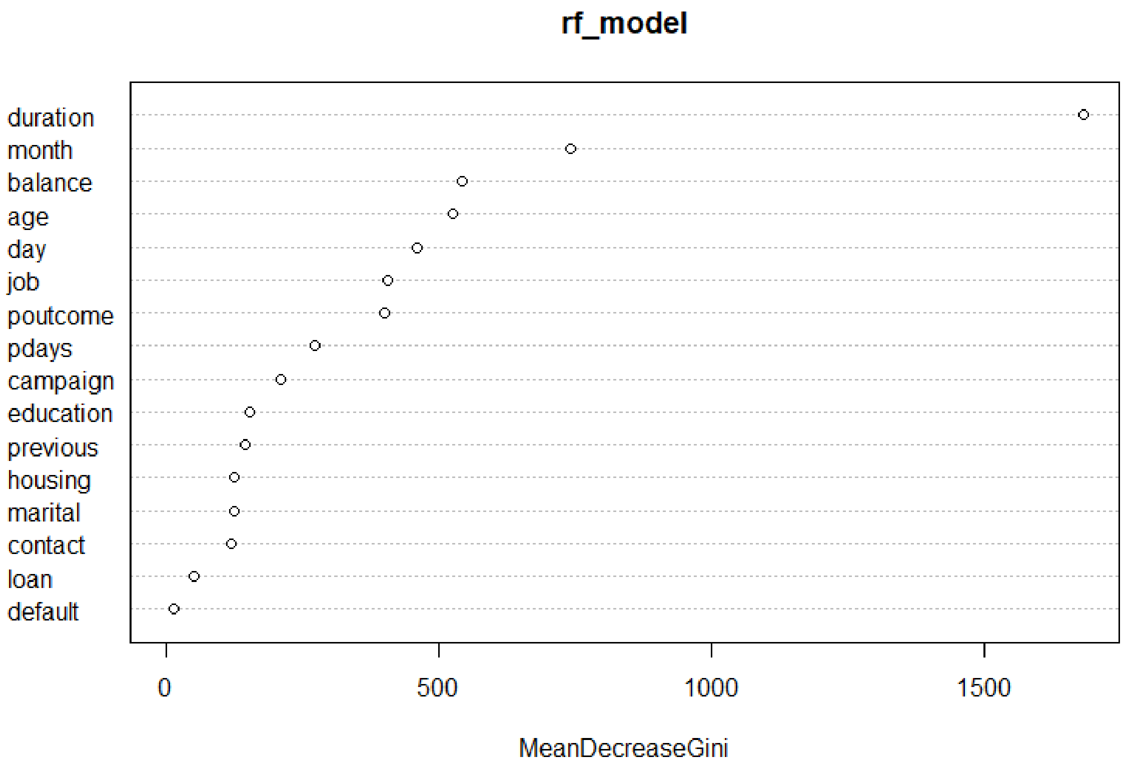


Figure 5. Significance of variables

The most significant variable is still duration.

(f) Compare the test error rates from all the classifiers above.

According to the test error rates, the random forest method performs better than the bagging method, and bagging performs better than the pruned classification tree. That means the random forest performs better than others with an error of 0.0933353. And the classification tree method is the least accurate, with an error of 0.1069743.