Machine Learning Homework 2

Jin Miao

February 24, 2018

Task 1

1.1 Load Data

Go to the UCI ML Repository (click on the hyperlink) and download the data. There you will find two .zip files, you should use the one called “bank-additional.zip” (ignore the “bank.zip” file which is an older version of the same data). In the .zip file you dowloaded you should find and use the “bank-additional-full” .csv file. Import the bank-additional-full.csv in R Studio. You can do that either using the R Studio dataset GUI or running the command read.table().

```
mlmkt = read.csv(file = "M:/A Master of Science in Marketing Sciences/MS Machine Learning/Homewo
rk2/bank-additional-full.csv",sep=";",header = TRUE)
```

1.2 Remove Irrelevant Variables

Remove the variables duration, date_of_week, month and nr.employed and explain why removing these variables makes sense.

```
drops <- c("month","day_of_week","duration","nr.employed")
mlmkt = mlmkt[ , !(names(mlmkt) %in% drops)]
```

Reasoning:

According to the “bank-additional-names.txt” file, the attribute information of the removed variables is shown as follows:

1. month: last contact month of year (categorical: “jan”, “feb”, “mar”, …, “nov”, “dec”)
2. day_of_week: last contact day of the week (categorical: “mon”,“tue”,“wed”,“thu”,“fri”)
3. duration: last contact duration, in seconds (numeric).
4. nr.employed: number of employees - quarterly indicator (numeric)

“month” and “day_of_week” are irrelevant for the classification goal because these variables cannot provide substantive marketing insights with respects to predicting whether the client will subscribe a term deposit.

As noted by the “bank-additional-names.txt” file, “duration” is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, “duration” should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

I remove “nr.employed” because “emp.var.rate” (employment variation rate) has already been included. These two variables are highly correlated so including both of them results in overfitting.

1.3 Summarize Dataset

```
summary(mlmkt)
```

```
##       age                  job            marital     
##  Min.   :17.00   admin.     :10422   divorced: 4612  
##  1st Qu.:32.00   blue-collar: 9254   married :24928  
##  Median :38.00   technician : 6743   single  :11568  
##  Mean   :40.02   services   : 3969   unknown :   80  
##  3rd Qu.:47.00   management : 2924                   
##  Max.   :98.00   retired    : 1720                   
##                  (Other)    : 6156                   
##                education        default         housing     
##  university.degree  :12168   no     :32588   no     :18622  
##  high.school        : 9515   unknown: 8597   unknown:  990  
##  basic.9y           : 6045   yes    :    3   yes    :21576  
##  professional.course: 5243                                  
##  basic.4y           : 4176                                  
##  basic.6y           : 2292                                  
##  (Other)            : 1749                                  
##       loan            contact         campaign          pdays      
##  no     :33950   cellular :26144   Min.   : 1.000   Min.   :  0.0  
##  unknown:  990   telephone:15044   1st Qu.: 1.000   1st Qu.:999.0  
##  yes    : 6248                     Median : 2.000   Median :999.0  
##                                    Mean   : 2.568   Mean   :962.5  
##                                    3rd Qu.: 3.000   3rd Qu.:999.0  
##                                    Max.   :56.000   Max.   :999.0  
##                                                                    
##     previous            poutcome      emp.var.rate      cons.price.idx 
##  Min.   :0.000   failure    : 4252   Min.   :-3.40000   Min.   :92.20  
##  1st Qu.:0.000   nonexistent:35563   1st Qu.:-1.80000   1st Qu.:93.08  
##  Median :0.000   success    : 1373   Median : 1.10000   Median :93.75  
##  Mean   :0.173                       Mean   : 0.08189   Mean   :93.58  
##  3rd Qu.:0.000                       3rd Qu.: 1.40000   3rd Qu.:93.99  
##  Max.   :7.000                       Max.   : 1.40000   Max.   :94.77  
##                                                                        
##  cons.conf.idx     euribor3m       y        
##  Min.   :-50.8   Min.   :0.634   no :36548  
##  1st Qu.:-42.7   1st Qu.:1.344   yes: 4640  
##  Median :-41.8   Median :4.857              
##  Mean   :-40.5   Mean   :3.621              
##  3rd Qu.:-36.4   3rd Qu.:4.961              
##  Max.   :-26.9   Max.   :5.045              
## 
```

Task 2

2.1 Create Input and Output Variables

You will use the input variables (minus the variables which we deleted) to predict the output variable (whether the client subscribed for a term deposit).

```
x <- model.matrix(y~.,mlmkt)[,-1]
y <- mlmkt$y
```

2.2 Remove Missing Values

```
mlmkt$default[mlmkt$default == "unknown"] = NA
mlmkt$housing[mlmkt$housing == "unknown"] = NA
mlmkt$loan[mlmkt$loan == "unknown"] = NA
mlmkt$job[mlmkt$job == "unknown"] = NA
mlmkt$marital[mlmkt$marital == "unknown"] = NA
mlmkt$loan[mlmkt$loan == "unknown"] = NA
mlmkt$education[mlmkt$education == "unknown"] = NA

mlmkt = na.omit(mlmkt)
```

2.3 Reshape Categorical Variables

```
mlmkt$job = 1 - (mlmkt$job == "unemployed")
mlmkt$marital = mlmkt$marital == "married"
mlmkt$marital = as.numeric(mlmkt$marital)

edu = as.character(mlmkt$education)

edu[edu == "illiterate"] = 0
edu[edu == "basic.4y"] = 1
edu[edu == "basic.6y"] = 2
edu[edu == "basic.9y"] = 3
edu[edu == "high.school"] = 4
edu[edu == "professional.course"] = 5
edu[edu == "university.degree"] = 6
edu = as.numeric(edu)
mlmkt$education = edu
```

Task 3

3.1 Create Training and Test Set

Now split the sample into two equal sub-samples, for training and testing. Use set.seed(1) and the sample() command like in the the R Lab to create a training set and a test set.

```
set.seed(1)
train <- sample( 1: nrow(mlmkt),nrow(mlmkt)/2)
test <- -train
subscription.test = mlmkt[test,]
```

Thus, the input for the Training Set is x[train] and the output is y[train]. The input for the Test Set is x[test] and the output is y[test].

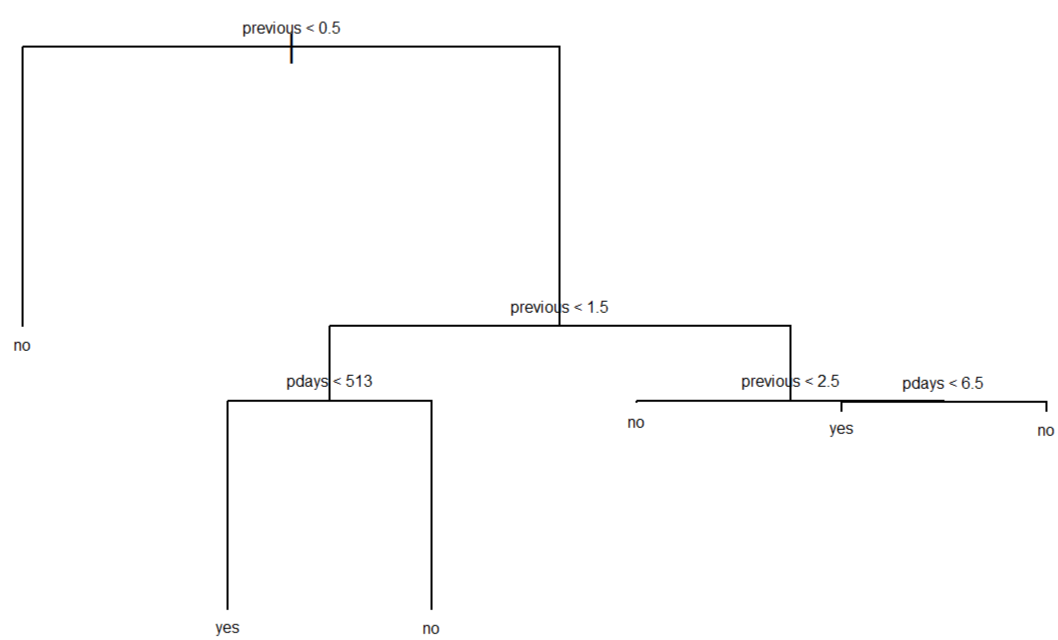
Task 4

4.1 Model Fitting

4.1.1 Gini Tree

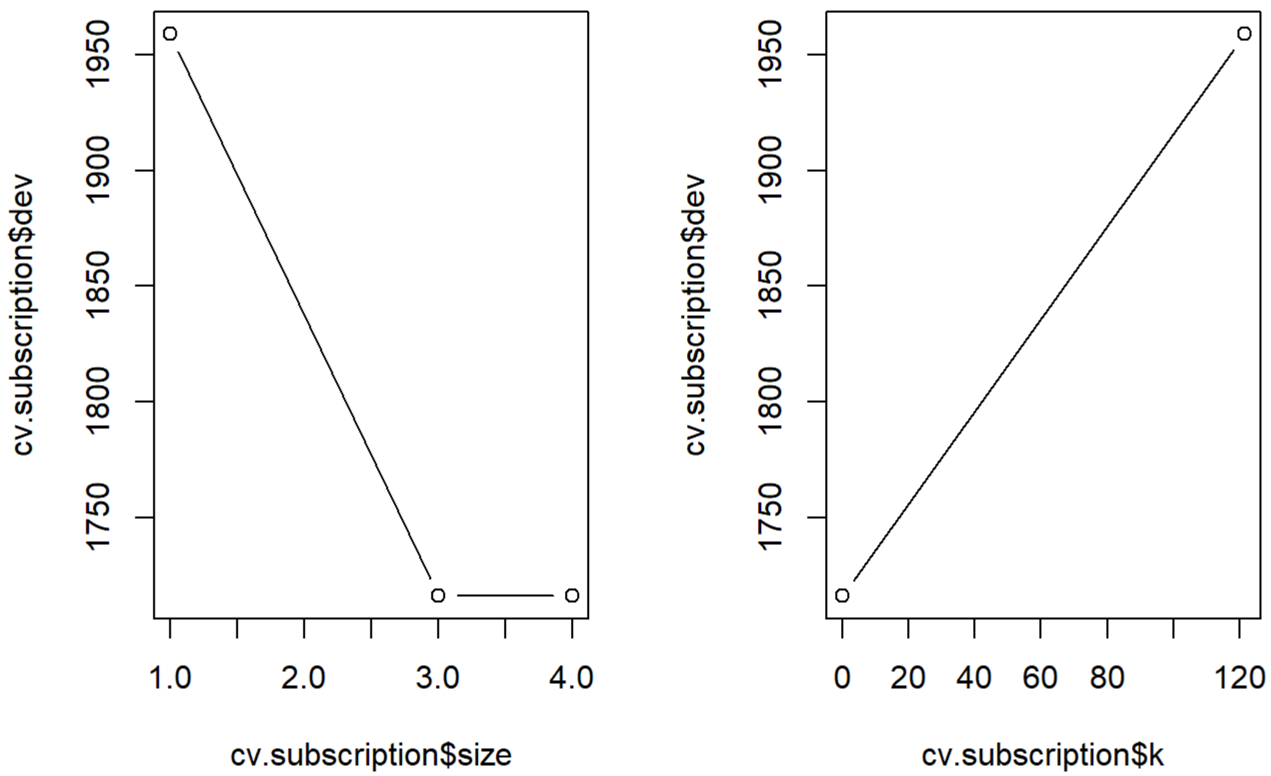
Fit a simple classification tree to your training data to predict the output variable. Try using both “gini” and “deviance” as the splitting criteria; what do you observe? (hint: check out the tree.control() function). Print your trees.

```
library(tree)
library(ISLR)

ginitree.subscription <- tree(y ~. -y  , mlmkt[train,], split = "gini", control = tree.control(n
obs = 15244, mincut = 50))

set.seed(33)
prune.ginitree <- prune.misclass(ginitree.subscription, newdata = mlmkt[train,], best = 6)
plot(prune.ginitree)
text(prune.ginitree, pretty = 0, cex = 0.5)
```

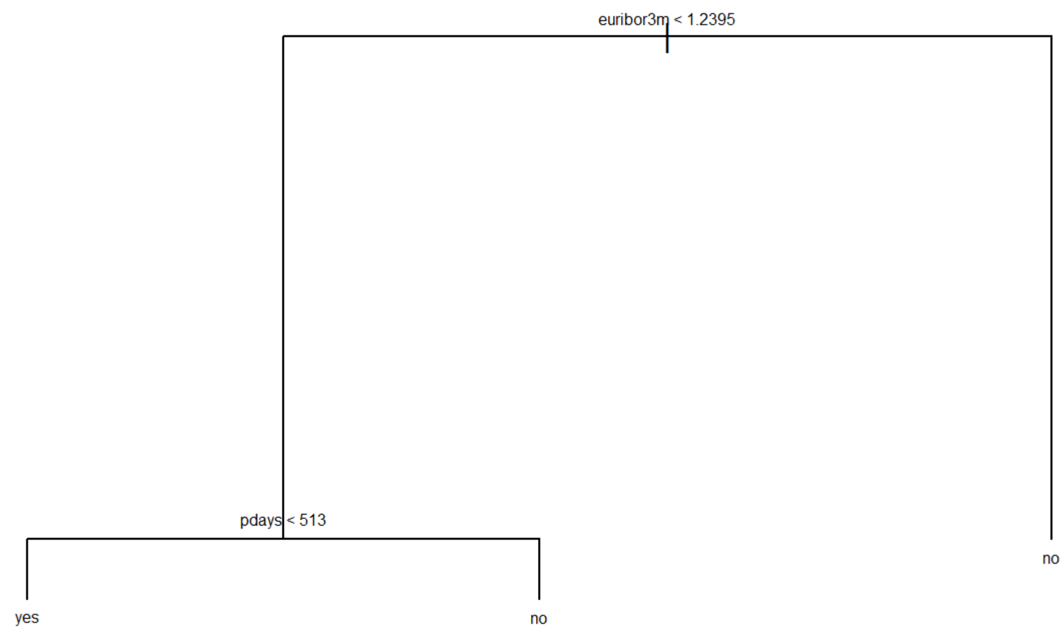


4.1.2 Daviance Tree

```
devtree.subscription <- tree(y ~. -y  , mlmkt , subset = train, split = "deviance")
# plot(devtree.subscription, main = "Tree Based on Deviance Splitting Creterion Before Pruning")
# text(devtree.subscription, pretty = 0, cex = .5)

set.seed(1)
cv.subscription <- cv.tree(devtree.subscription, FUN = prune.misclass)
par(mfrow = c(1, 2))
plot(cv.subscription$size, cv.subscription$dev,type = "b")
plot(cv.subscription$k, cv.subscription$dev,type = "b")
```




```
set.seed(1)
par(mfrow = c(1, 1))
prune.devtree <- prune.misclass(devtree.subscription, best = 3)
plot(prune.devtree, main = "Tree Based on Deviance Splitting Creterion After Pruning")
text(prune.devtree, pretty = 0, cex = 0.5)
```



4.2 Insights and Observations

1. The tree based on Gini splitting creterion predicts that customer will subscribe the term deposit under the following two scenarios:

*i) **pdays** (number of days that passed by after the client was last contacted from a previous campaign) is smaller than 513 AND **previous** (number of contacts performed before this campaign and for this client) is larger than 0.5 and smaller than 1.5;*

*ii) **pdays** (number of days that passed by after the client was last contacted from a previous campaign) is smaller than 6.5 AND **previous** (number of contacts performed before this campaign and for this client) is larger than 2.5.*

2. According to Gini splitting creterion, the most important factor in determining Deposit Subscription is **previous** (number of contacts performed before this campaign and for this client). If **previous** is 0, indicating no previous coorperation, then the deposit will be rejected.

3. The (pruned) tree based on Deviance splitting creterion predicts that customers will subscribe the term deposit only when **pdays** (number of days that passed by after the client was last contacted from a previous campaign) is smaller than 513 and **euribor3m** (euribor 3 month rate) is lower than 1.2395.

4. According to Gini splitting creterion, the most important factor in determining Deposit Subscription is **euribor3m** (euribor 3 month rate). If **euribor3m** is too high (larger than 1.2395), then the deposit will be rejected.

Task 5

Fit a random forest to you training data and print the variable importance graph.

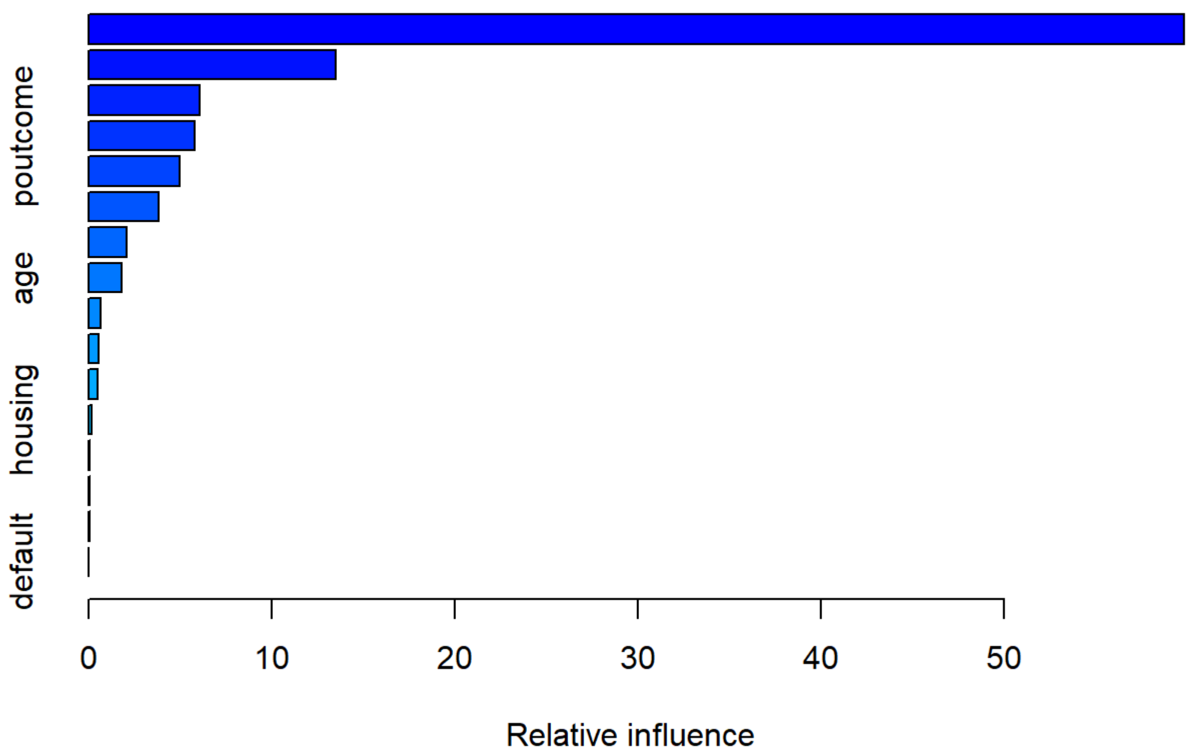
```r
library(randomForest)
library(MASS)
set.seed(33)
rf.subscription <- randomForest(y~.-y, data = mlmkt, subset = train, importance = TRUE)
importance(rf.subscription)
```

```
##                        no         yes MeanDecreaseAccuracy
## age            26.6623419  -2.5175347            24.536240
## job             0.7538281   0.6645684             1.033773
## marital         9.1770230  -7.4469814             4.963145
## education       7.5757572   5.2292777             9.388313
## default         0.0000000   0.0000000             0.000000
## housing         2.5308166  -0.9533033             1.759813
## loan           -0.7054146   3.5600254             1.214068
## contact         9.1870358  24.3520837            11.527637
## campaign       10.1204017   4.1456340            11.054764
## pdays          10.8881228  26.2493176            22.555936
## previous        9.8611161  -3.0716467             8.851333
## poutcome       14.9179409  11.6257663            18.061003
## emp.var.rate   27.8876179   9.0376906            29.550261
## cons.price.idx 26.0169135 -11.3174672            26.424166
## cons.conf.idx  28.9552430  -7.7090324            30.100733
## euribor3m      39.9449783  10.4029943            45.002343
##                MeanDecreaseGini
## age                4.062407e+02
## job                2.215246e+01
## marital            6.443042e+01
## education          1.712638e+02
## default            1.577373e-03
## housing            7.025481e+01
## loan               5.338489e+01
## contact            4.837394e+01
## campaign           1.796175e+02
## pdays              1.680684e+02
## previous           5.972415e+01
## poutcome           1.382651e+02
## emp.var.rate       1.305384e+02
## cons.price.idx     1.286495e+02
## cons.conf.idx      1.496877e+02
## euribor3m          5.830653e+02
```

Task 6

Fit a boosted tree to your training data and print the variable importance graph.

```r
library(gbm)
set.seed(33)
boost.subscription <- gbm(y ~.-y, data = mlmkt[train,], n.trees = 5000, distribution = "gaussia
n", interaction.depth = 4)
summary(boost.subscription)
```




```
##                             var     rel.inf
## euribor3m             euribor3m 59.83643521
## pdays                     pdays 13.51716070
## cons.conf.idx     cons.conf.idx  6.09278429
## poutcome               poutcome  5.79929737
## emp.var.rate       emp.var.rate  4.98248661
## cons.price.idx cons.price.idx  3.80934879
## contact                 contact  2.10112834
## age                         age  1.79383755
## previous               previous  0.64711157
## campaign               campaign  0.53059141
## education             education  0.49646226
## housing                 housing  0.15788452
## marital                 marital  0.08679290
## loan                       loan  0.07509829
## job                         job  0.07358020
## default                 default  0.00000000
```

With a different R package **adabag**, the importance matrix is shown as follows:

```r
library(adabag)
formula <- y ~.-y
cntrl <- rpart.control(maxdepth = 1, minsplit = 0, cp = -1)
mfinal <- 400
data.boosting <- boosting(formula = formula,  data = mlmkt[train,], mfinal = mfinal, coeflearn =
 "Breiman", boos = TRUE, control = cntrl)
data.boosting$importance
```

```
##            age       campaign  cons.conf.idx cons.price.idx        contact
##    0.047752312    0.010983173    1.315100553    0.201078692    0.777221802
##        default      education   emp.var.rate      euribor3m        housing
##    0.000000000    0.019818349    2.991841419   85.720301003    0.003952463
##            job           loan        marital          pdays       poutcome
##    0.000000000    0.000000000    0.000000000    6.823871246    2.088078988
##       previous
##    0.000000000
```

Task 7

7.1 Model Comparison

```r
ginitree.pred <- predict(prune.ginitree,subscription.test,type = "class")
y.test = subscription.test$y
table(Prediction = ginitree.pred,Truth = y.test)
```

```
##           Truth
## Prediction    no   yes
##        no  13180  1561
##        yes   164   339
```

For the (pruned) tree method based on Gini splitting creterion, the predictive accuracy for the Test Set is 0.887.

```r
## Tree Prediction
devtree.pred <- predict(prune.devtree,subscription.test,type = "class")
y.test = subscription.test$y
table(Prediction = devtree.pred, Truth = subscription.test$y)
```

```
##           Truth
## Prediction    no   yes
##        no  13180  1561
##        yes   164   339
```

For the (pruned) tree method based on Deviance splitting creterion, the predictive accuracy for the Test Set is 0.887.

```
yhat.rf = predict(rf.subscription,newdata=mlmkt[-train,],type = "class")
y.test = subscription.test$y
table(Prediction = yhat.rf, Truth = y.test)
```

```
##           Truth
## Prediction    no   yes
##        no  12789  1281
##        yes   555   619
```

For the Random Forest method, the predictive accuracy for the Test Set is 0.88.

```
yhat.boost <- predict(boost.subscription, newdata = mlmkt[-train, ], type = "response", n.trees
= 5000)
y.test = subscription.test$y
predict_class <- yhat.boost > 0.5
#table(Prediction = predict_class, Truth = y.test)
```

The Confusion Matrix for the Boosting Method is shown as follows:

```
data.predboost <- predict.boosting(data.boosting, newdata = mlmkt[-train,])
data.predboost$confusion
```

```
##                Observed Class
## Predicted Class    no   yes
##             no  13164  1551
##             yes   180   349
```

For the Boosting method, the predictive accuracy for the Test Set is 0.886.

7.2 Conclusions

Based on the results above, the highest prediction accuracy comes from the (pruned) tree methods based on Gini/Deviance splitting creterion. The second best prediction accuracy derives from the boosted tree on the test data. Actually, the difference in predictability between simple tree and boosted tree methods is negligible. The random forest method has the lowest prediction accuracy for this study.