Data 622 Test1

Samriti Malhotra

11/11/2020

Read Data

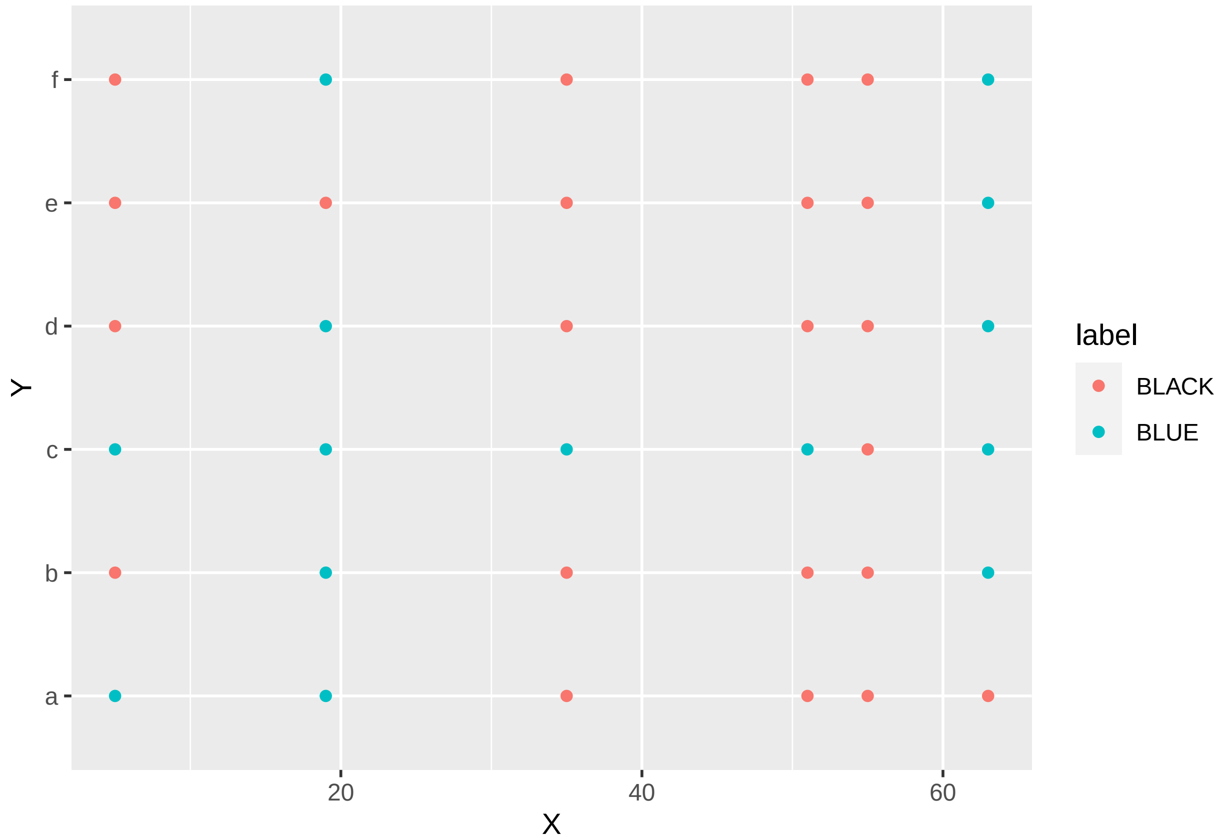
```
df <- read.csv("data_hw1_622.csv", header = TRUE, sep =",")
str(df)
```

```
## 'data.frame':    36 obs. of  3 variables:
##  $ X    : int  5 5 5 5 5 5 19 19 19 19 ...
##  $ Y    : chr  "a" "b" "c" "d" ...
##  $ label: chr  "BLUE" "BLACK" "BLUE" "BLACK" ...
```

```
# printing the top 5 rows of the data frame.
kable(head(df)) %>%
  kable_styling(bootstrap_options = c("striped","hover","condensed","responsive"),full_width   = F,posi
  row_spec(0, background ="gray")
```

X	Y	label
5	a	BLUE
5	b	BLACK
5	c	BLUE
5	d	BLACK
5	e	BLACK
5	f	BLACK

```
ggplot(df,aes(y=Y,x=X,color=label)) + geom_point()
```



The data has 36 rows and 3 columns , out of which columns 'Y' and 'label' are Character and column 'X' is int, but all the columns are categorical in nature and hence can be converted to factors to be consistent.

Prepare data

```
df$X = as.factor(df$X)
df$Y = as.factor(df$Y)
df$label = as.factor(df$label)
#df[sapply(df, is.character)] <- lapply(df[sapply(df, is.character)], as.factor)

# summary statistics of the columns
summary(df)
```

```
##    X       Y        label
##  5 :6   a:6   BLACK:22
##  19:6   b:6   BLUE :14
##  35:6   c:6
##  51:6   d:6
##  55:6   e:6
##  63:6   f:6
```

```
str(df)
```

```
## 'data.frame':    36 obs. of  3 variables:
##  $ X    : Factor w/ 6 levels "5","19","35",..: 1 1 1 1 1 1 2 2 2 2 ...
##  $ Y    : Factor w/ 6 levels "a","b","c","d",..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ label: Factor w/ 2 levels "BLACK","BLUE": 2 1 2 1 1 1 2 2 2 2 ...
```

Split Dataset

```
set.seed(53)

trainidx<-sample(1:nrow(df) , size=round(0.77*nrow(df)),replace=F)
train_set <- df[trainidx,]
test_set <- df[-trainidx,]
```

(A) Run Bagging (ipred package)

– sample with replacement

– estimate metrics for a model

– repeat as many times as specied and report the average

```
# Bagging
trainBgModel <- bagging(label ~ ., data=train_set, nbagg = 100, coob = TRUE)
trainBgModel
```

```
##
## Bagging classification trees with 100 bootstrap replications
##
## Call: bagging.data.frame(formula = label ~ ., data = train_set, nbagg = 100,
##     coob = TRUE)
##
## Out-of-bag estimate of misclassification error:  0.2143
```

```
confMat_train <- table(predict(trainBgModel), train_set$label)
confMat_train
```

```
##
##         BLACK BLUE
##   BLACK    15    4
##   BLUE      2    7
```

```
testbag = predict(trainBgModel, newdata=test_set)
confusionMat_bg <- table(testbag, test_set$label)
confusionMat_bg
```

```
##
## testbag BLACK BLUE
##   BLACK     4    0
##   BLUE      1    3
```

```
# Calculating  the ACC,TPR,FPR,TNR & FNR from confusion matrix
acc_bag <- sum(diag(confusionMat_bg)) / sum(confusionMat_bg)
tpr_bag <- confusionMat_bg[1,1]/sum(confusionMat_bg[1,1], confusionMat_bg[2,1])
fpr_bag <- confusionMat_bg[1,2]/sum(confusionMat_bg[1,2], confusionMat_bg[2,2])
tnr_bag <- confusionMat_bg[2,2]/sum(confusionMat_bg[2,2], confusionMat_bg[1,2])
fnr_bag <- confusionMat_bg[2,1]/sum(confusionMat_bg[2,1], confusionMat_bg[1,1])
auc_bag <- auc(roc(testbag, ifelse(test_set$label == 'BLUE', 1, 0)))
```

```
## Setting levels: control = BLACK, case = BLUE
```

```
## Setting direction: controls < cases
```

```
Bgrow <- c("Bagging ",round(auc_bag,2), round(acc_bag,2),round(tpr_bag,2),round(fpr_bag,2), round(tnr_ba
```

```
Bgrow
```

```
## [1] "Bagging " "0.88"     "0.88"     "0.8"      "0"        "1"        "0.2"
```

```
resMatrix <- data.frame(matrix(ncol = 6, nrow = 0))
resMatrix <- rbind(resMatrix,Bgrow)
colnames(resMatrix) <- c("ALGO", "AUC","ACC", "TPR", "FPR", "TNR ", "FNR")
```

(B) Run LOOCV (jacknife) for the same dataset

— iterate over all points

```
 -- keep one observation as test

-- train using the rest of the observations

-- determine test metrics

-- aggregate the test metrics
```

end of loop

find the average of the test metric(s)

Compare (A), (B) above with the results you obtained in HW-1 and write 3 sentences explaining the observed difference.

```
data <- df
acc <- NULL
for(i in 1:nrow(data))
{
    # Train-test splitting
    # 35 samples -> fitting
    # 1 sample -> testing
    train <- data[-i,]
    test <- data[i,]


    # Fitting
    model <- glm(label~.,family=binomial,data=train)
    pred_glm <- predict(model,test,type='response')

   # If prob > 0.5 then 1, else 0
    results <- ifelse(pred_glm > 0.5,"BLUE","BLACK")

    # Actual answers
    answers <- test$label

    # Calculate accuracy
    misClasificError <- mean(answers != results)

    # Collecting results
    acc[i] <- 1-misClasificError


}
```

```r
# Average accuracy of the model

mean(acc)
```

```
## [1] 0.7777778
```

Naive Bayes

```r
data <- df
acc <- NULL
for(i in 1:nrow(data))
{
    # Train-test splitting
    # 35 samples -> fitting
    # 1 sample -> testing
    train <- data[-i,]
    test <- data[i,]


    # Fitting
    model <- naiveBayes(label~.,data=train)
    pred_nb <- predict(model,test,type='raw')

   # If prob > 0.5 then 1, else 0
    results <- ifelse(pred_nb > 0.5,"BLUE","BLACK")

    # Actual answers
    answers <- test$label

    # Calculate accuracy
    misClasificError <- mean(answers != results)

    # Collecting results
    acc[i] <- 1-misClasificError
}

mean(acc)
```

```
## [1] 0.5138889
```

Conclusion:

The accuracy of Bagging method is (.88) and LOOCV produced accuracy of (.51) for NB and (.77) for GLB. Both models performed differently and score better. Bagging is a method to reduce over fitting. It trains many models on resampled data and then take their average to get an averaged model.