KT_11-15_Vishal Arora

Vishal Arora

11/13/2020

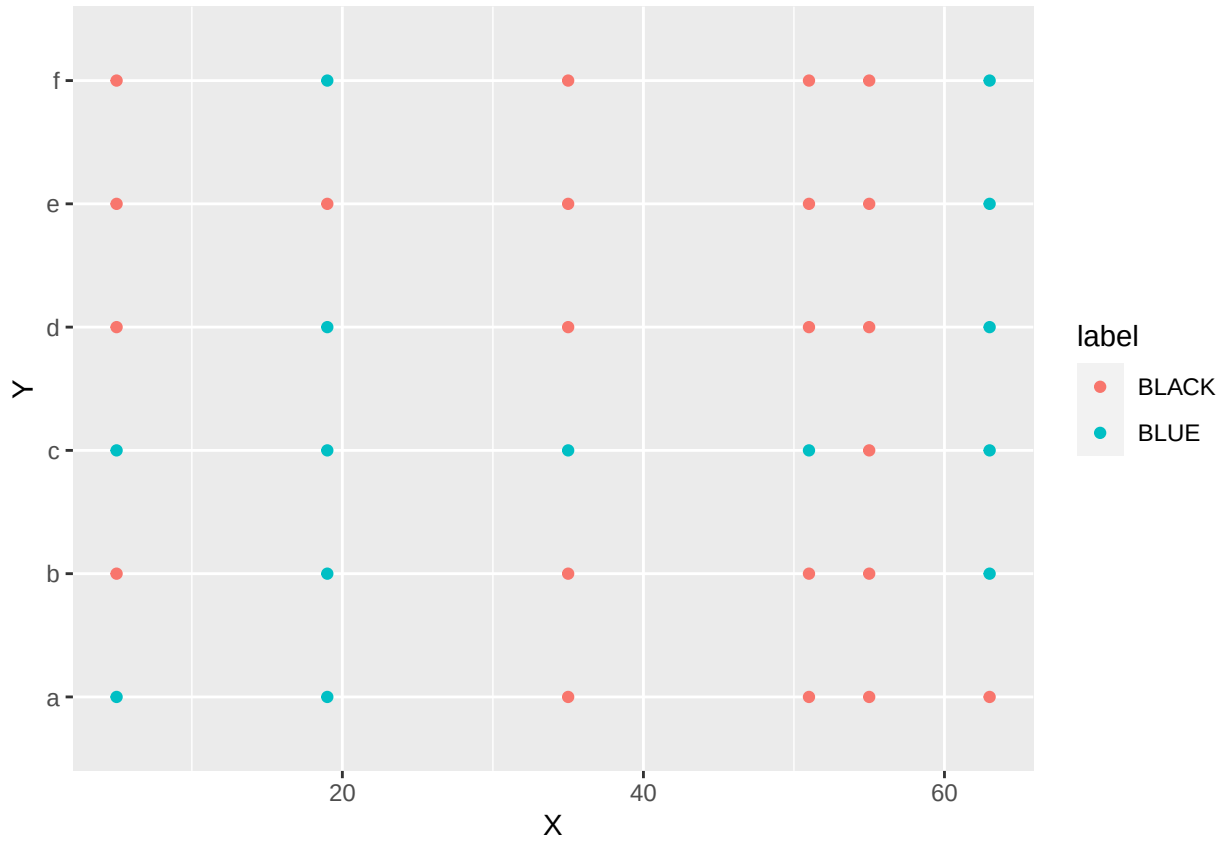
```
df <- read.csv("data_hw1_622.csv", header = TRUE, sep =",")
str(df)
```

```
## '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.

Conversion & summarizing 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)
```

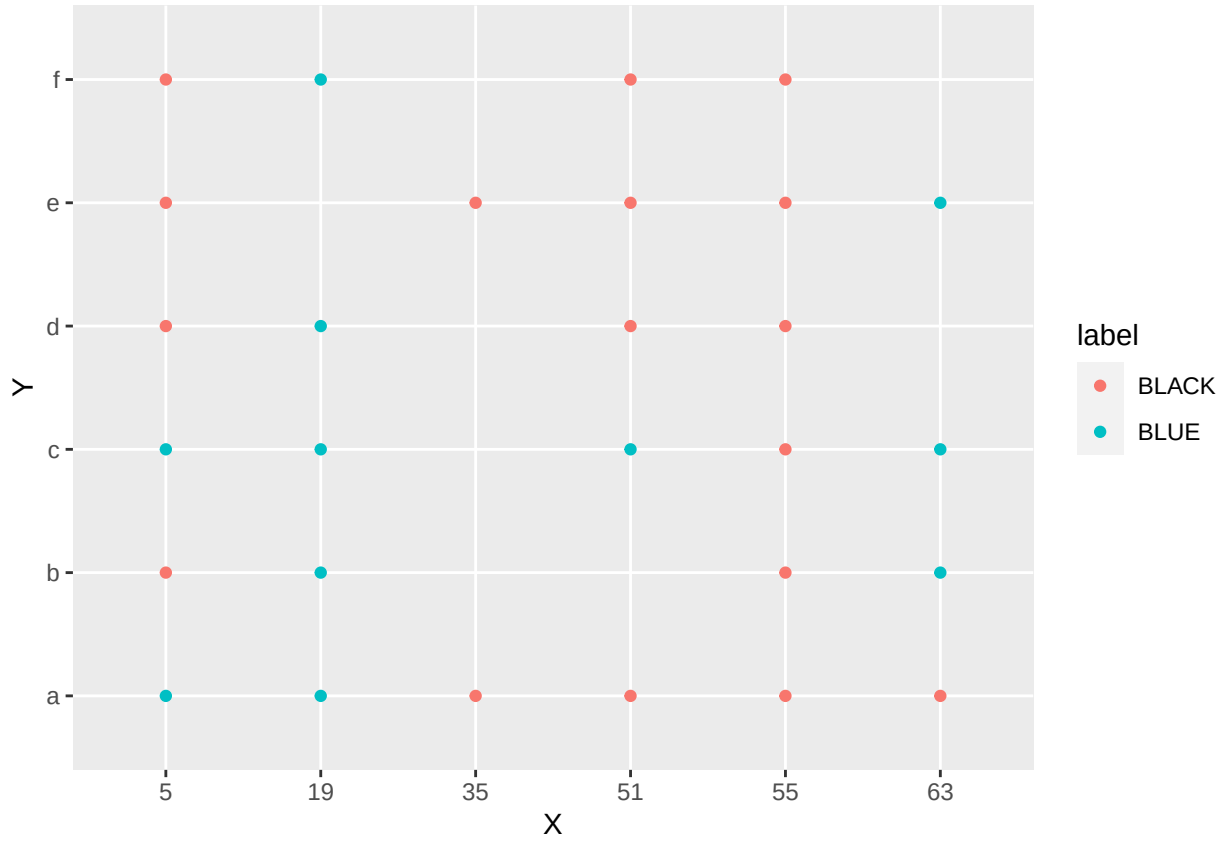
```
##   X      Y       label
##  5 :6   a:6   BLACK:22
##  19:6   b:6   BLUE :14
##  35:6   c:6
##  51:6   d:6
##  55:6   e:6
##  63:6   f:6
```

```
str(df)
```

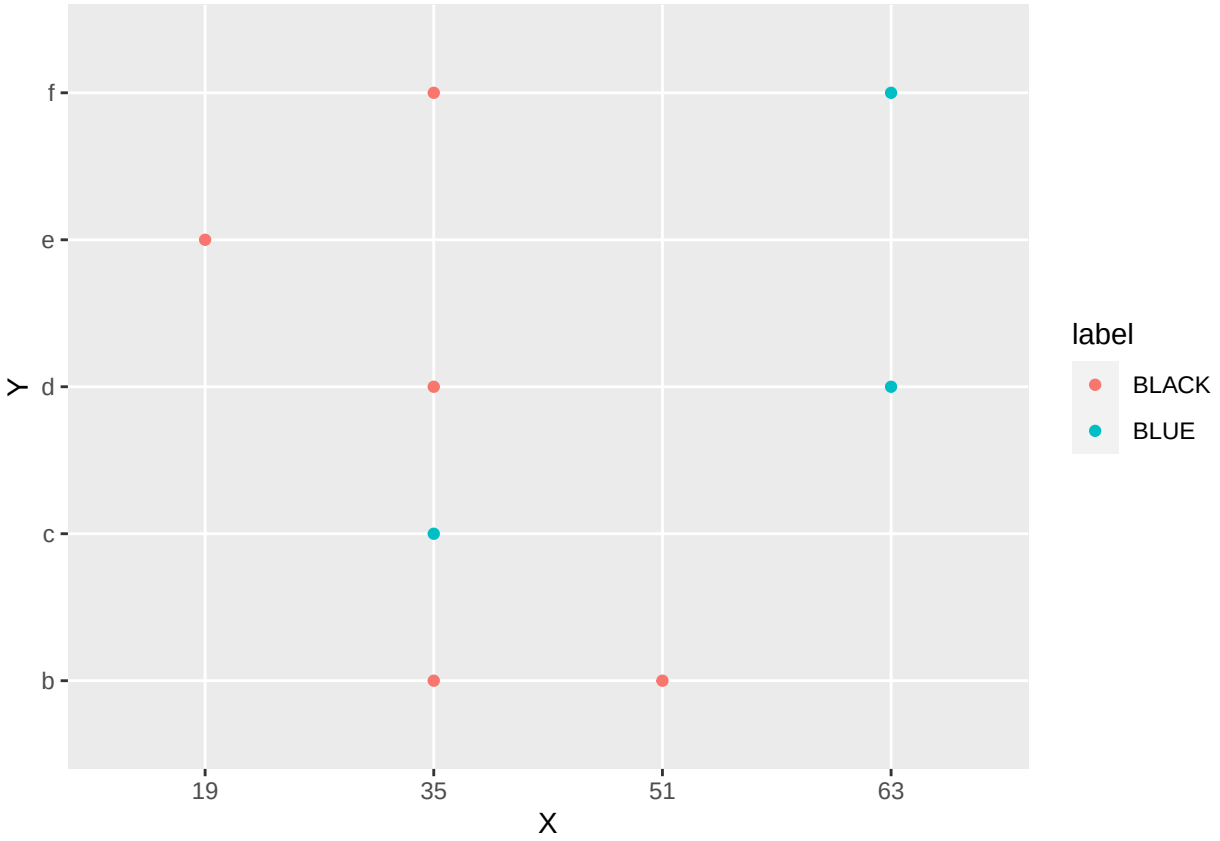
```
## 'data.frame':    36 obs. of  3 variables:
##  $ X    : Factor w/ 6 levels "5","19","35",..: 1 1 1 1 1 1 2 2 2 2 ...
##  $ Y    : Factor w/ 6 levels "a","b","c","d",..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ label: Factor w/ 2 levels "BLACK","BLUE": 2 1 2 1 1 1 2 2 2 2 ...
```

Splitting Data & Models

```
#setting seed number
set.seed(53)



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

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




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



Bagging

```
# 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")
```

LOOCV

GLM

```
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
```

```
}

# Average accuracy of the model

mean(acc)
```

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

NB

```
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 for Bagging and LOOCV(GLM) has increased but still in the overall run the KNN=3 from original model has the most accuracy . The LOOVC (NB) has shown a degrading performance in respect to original NB model.Bagging is a method to reduce overfitting. You train many models on resampled data and then take their average to get an averaged model. One disadvantage of bagging is that it introduces a loss of interpretability of a model. The resultant model can experience lots of bias when the proper procedure is ignored.