

SulekhaAloorravi_SL_Lab

Sulekha Aloorravi

Lab Question : For the following 50 observations (predicted probability and class label), what is the best accuracy value and at what cut-off?

Answer:

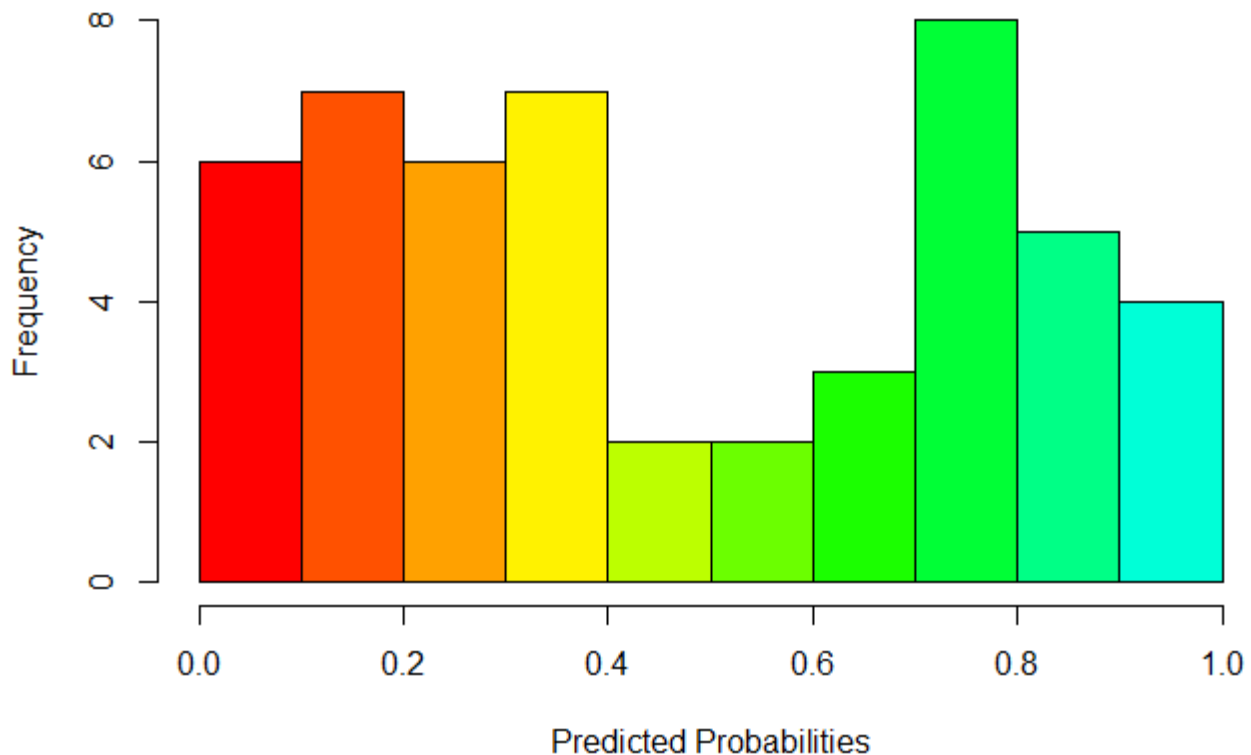
Save predicted probabilities and label data from Question into a csv file.

```
data <- read.csv("E:/Sulekhas BOOKSHELF/GreatLakes/StatisticalLearning/Lab/Lab.csv",header
= TRUE)
head(data)
```

```
## Predicted.... Y
## 1 0.31507530 1
## 2 0.24711658 0
## 3 0.01993854 0
## 4 0.63522368 1
## 5 0.81651226 0
## 6 0.76867154 0
```

```
hist(data$Predicted....,xlab = "Predicted Probabilities", main = "Display of Predicted Pro
babilities", col = rainbow(sample(1:50,50)))
```

Display of Predicted Probabilities



Input data file (data) consists of the following data:

1. predicted probabilities or estimated probabilities
2. labels that are binary values

In this assignment, model has already been applied and predicted probabilities are already calculated. Our aim is to make use of predicted probabilities and labels and calculate best accuracy and its corresponding cutoff.

ROCR is the package used to evaluate the performance of the model which has predicted these probabilities.

“Prediction” and “Performance” functions are part of ROCR package and they are used here.

prediction function is used to transform input data into standardized format.

performance function is used to perform all kinds of predictor evaluations

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##  
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':  
##  
##      lowess
```

```
pred <- prediction(data$Predicted,data$Y)  
pred
```

```

## An object of class "prediction"
## Slot "predictions":
## [[1]]
## [1] 0.315075300 0.247116583 0.019938541 0.635223676 0.816512256
## [6] 0.768671539 0.901123086 0.207731809 0.626827129 0.724754964
## [11] 0.004070023 0.833692794 0.123358244 0.249252743 0.573036542
## [16] 0.190291112 0.356869033 0.811016988 0.306974869 0.041476616
## [21] 0.920796024 0.960045044 0.099158739 0.467322766 0.190094205
## [26] 0.151809214 0.820857232 0.077273261 0.552612980 0.732081003
## [31] 0.758074351 0.390286229 0.724388645 0.020296189 0.942697427
## [36] 0.893232176 0.743319952 0.773019863 0.229053364 0.219980348
## [41] 0.156213842 0.371065335 0.690554685 0.127486623 0.372439317
## [46] 0.183769105 0.472558978 0.724079378 0.384598448 0.226369171
##
##
## Slot "labels":
## [[1]]
## [1] 1 0 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 0 0 1 1 0
## [36] 1 1 0 1 1 0 1 0 1 1 0 0 0 0 0
## Levels: 0 < 1
##
##
## Slot "cutoffs":
## [[1]]
## [1] Inf 0.960045044 0.942697427 0.920796024 0.901123086
## [6] 0.893232176 0.833692794 0.820857232 0.816512256 0.811016988
## [11] 0.773019863 0.768671539 0.758074351 0.743319952 0.732081003
## [16] 0.724754964 0.724388645 0.724079378 0.690554685 0.635223676
## [21] 0.626827129 0.573036542 0.552612980 0.472558978 0.467322766
## [26] 0.390286229 0.384598448 0.372439317 0.371065335 0.356869033
## [31] 0.315075300 0.306974869 0.249252743 0.247116583 0.229053364
## [36] 0.226369171 0.219980348 0.207731809 0.190291112 0.190094205
## [41] 0.183769105 0.156213842 0.151809214 0.127486623 0.123358244
## [46] 0.099158739 0.077273261 0.041476616 0.020296189 0.019938541
## [51] 0.004070023
##
##
## Slot "fp":
## [[1]]
## [1] 0 1 2 3 4 4 4 4 5 5 6 7 8 8 8 9 9 10 11 11 11 12 13
## [24] 14 14 15 16 16 16 16 16 16 16 17 17 18 18 18 18 19 20 21 22 22 23 24
## [47] 24 24 24 25 26
##
##
## Slot "tp":
## [[1]]
## [1] 0 0 0 0 0 1 2 3 3 4 4 4 4 5 6 6 7 7 7 8 9 9 9

```

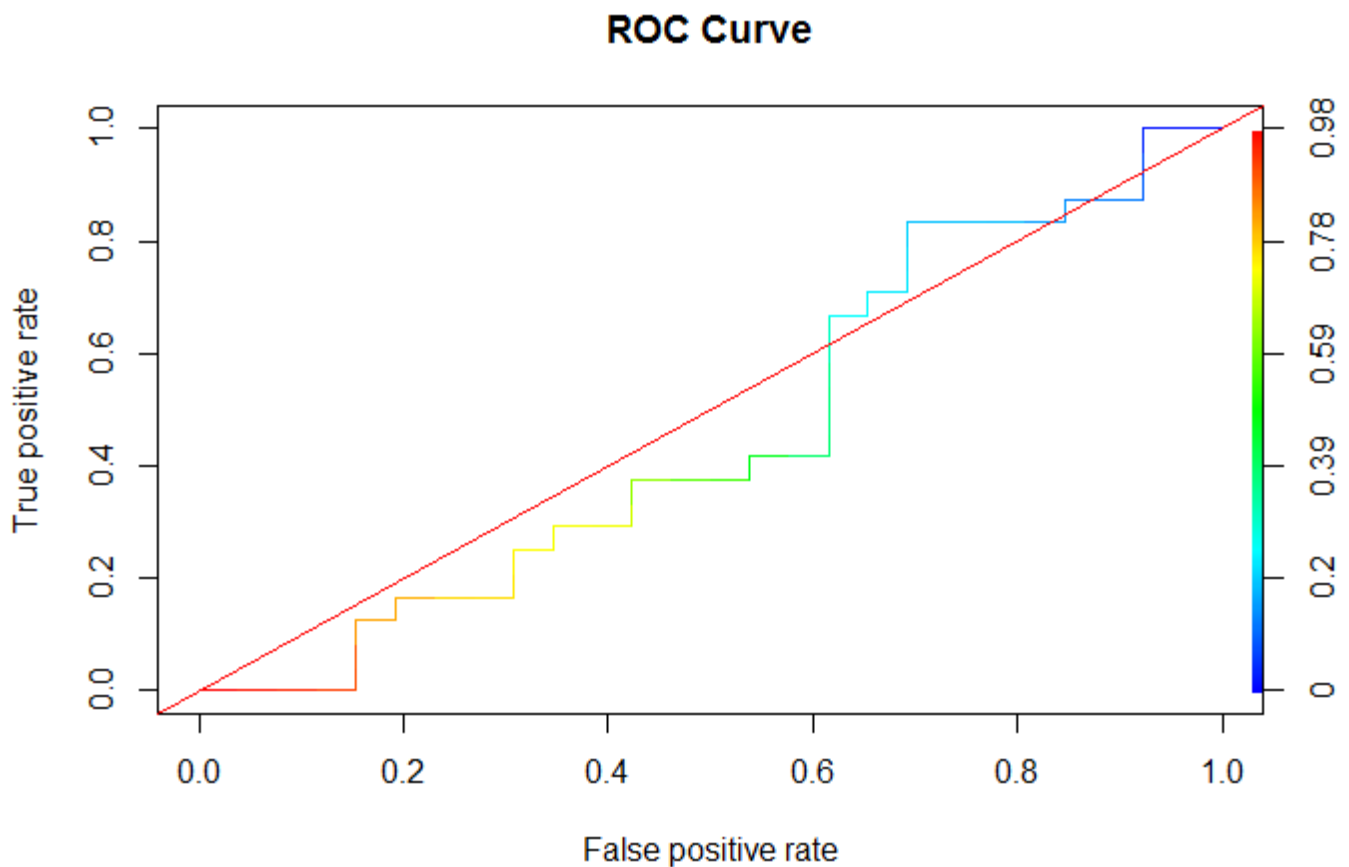
```
## [24] 9 10 10 10 11 12 13 14 15 16 16 17 17 18 19 20 20 20 20 21 21 21
## [47] 22 23 24 24 24
##
##
## Slot "tn":
## [[1]]
## [1] 26 25 24 23 22 22 22 22 21 21 20 19 18 18 18 17 17 16 15 15 15 14 13
## [24] 12 12 11 10 10 10 10 10 10 10 9 9 8 8 8 8 7 6 5 4 4 3 2
## [47] 2 2 2 1 0
##
##
## Slot "fn":
## [[1]]
## [1] 24 24 24 24 24 23 22 21 21 20 20 20 20 19 18 18 17 17 17 16 15 15 15
## [24] 15 14 14 14 13 12 11 10 9 8 8 7 7 6 5 4 4 4 4 4 3 3 3
## [47] 2 1 0 0 0
##
##
## Slot "n.pos":
## [[1]]
## [1] 24
##
##
## Slot "n.neg":
## [[1]]
## [1] 26
##
##
## Slot "n.pos.pred":
## [[1]]
## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
## [24] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
## [47] 46 47 48 49 50
##
##
## Slot "n.neg.pred":
## [[1]]
## [1] 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28
## [24] 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
## [47] 4 3 2 1 0
```

tpr - True Positive Rate or Sensitivity:- True positive rate. $P(\hat{Y} = + | Y = +)$. Estimated as: TP/P.

fpr - False Positive Rate or (1 - Specificity):- False positive rate. $P(\hat{Y} = + | Y = -)$. Estimated as: FP/N.

Draw an ROC curve based on tpr and fpr calculated using **performance** function.

```
perf_tpr_fpr <- performance(pred, "tpr", "fpr")
plot(perf_tpr_fpr,col="blue", colorize=T, main = "ROC Curve")
abline(a=0, b=1,col="red")
```

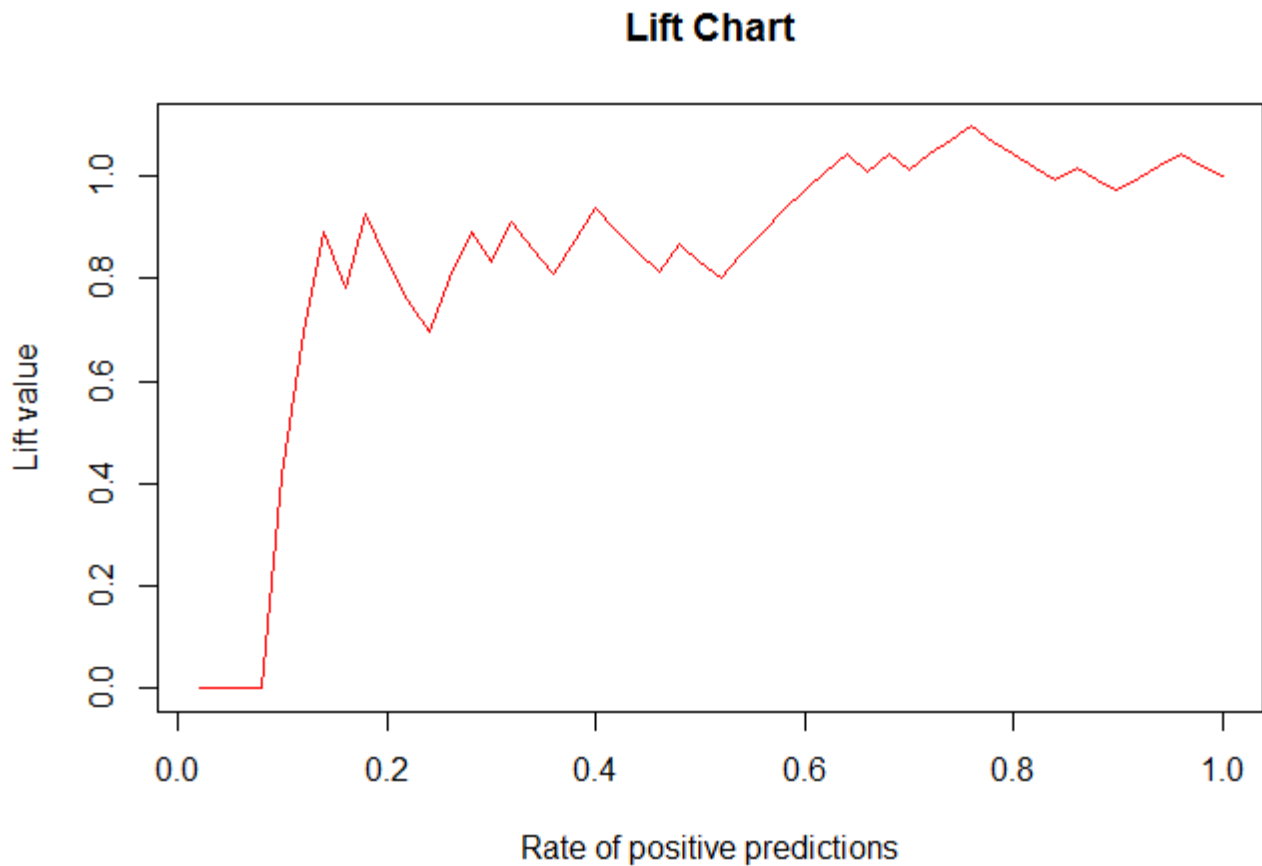


In the above graph, majority of the line is above the curve and this signifies that the model used has a poor performance.

lift - Lift value:- $P(\hat{Y} = + | Y = +) / P(\hat{Y} = +)$.

rpp - Rate of positive predictions:- $P(\hat{Y} = +)$. Estimated as: $(TP+FP)/(TP+FP+TN+FN)$.

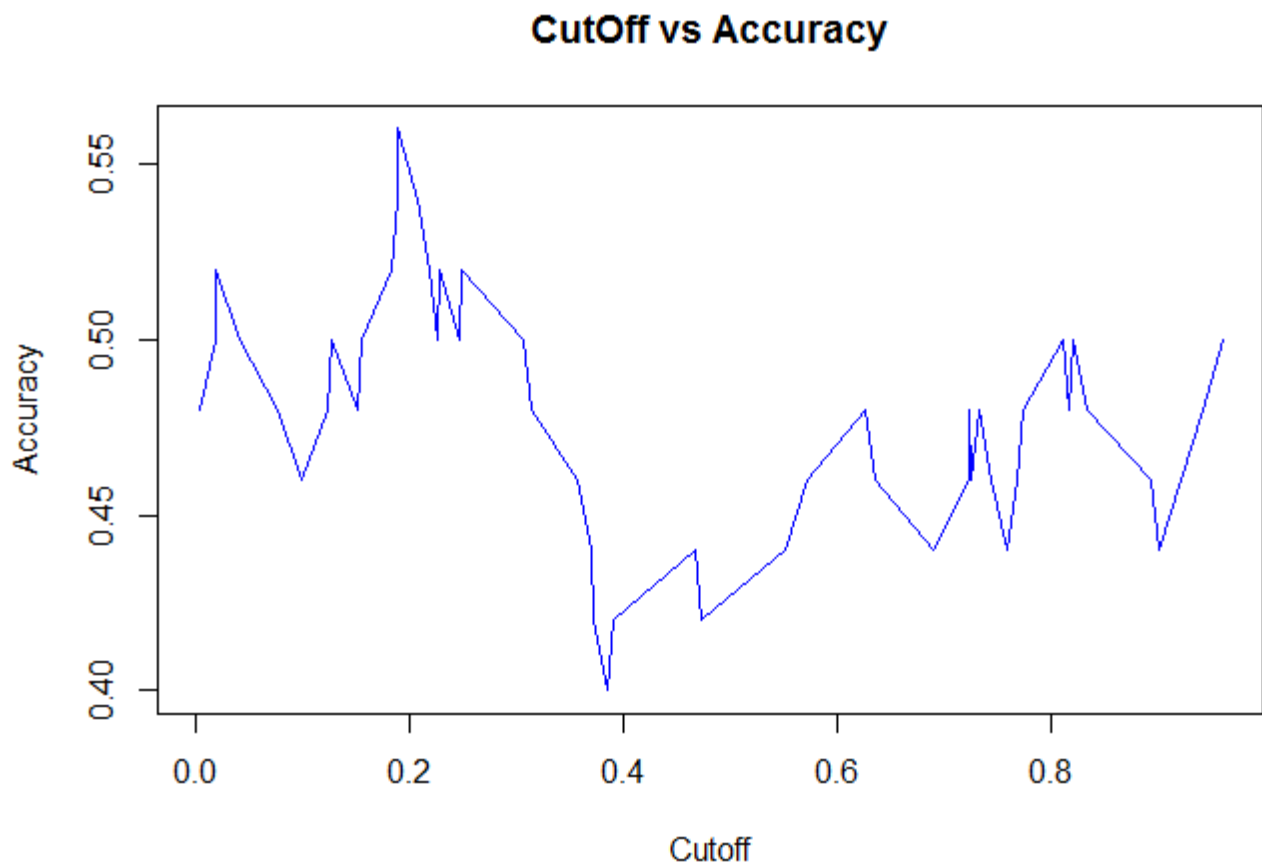
```
perf_lift_rpp <- performance(pred, "lift", "rpp")
plot(perf_lift_rpp, col="red", main = "Lift Chart")
```



acc - Accuracy:- Accuracy. $P(\hat{Y} = Y)$. Estimated as: $(TP+TN)/(P+N)$.

Cutoff of predicted values are calculated and plotted against their corresponding accuracy values.

```
perf_acc <- performance(pred, "acc")  
plot(perf_acc, col = "blue", main="CutOff vs Accuracy")
```



Accuracy and Cutoff values calculated from the above performance function is used to calculate the best accuracy and its corresponding cutoff.

```
i = which.max( slot(perf_acc, "y.values")[[1]] )  
bestacc = slot(perf_acc, "y.values")[[1]][i]  
cutoff = slot(perf_acc, "x.values")[[1]][i]
```

Best Accuracy Value with its corresponding CutOff are as follows:

```
print(c(accuracy= bestacc, cutoff = cutoff))
```

```
## accuracy    cutoff  
## 0.5600000 0.1902911
```