

## DATA 621 Homework 2

### Classification Metrics

Trishita Nath

#### Description

In this homework assignment, you will work through various classification metrics. You will be asked to create functions in R to carry out the various calculations. You will also investigate some functions in packages that will let you obtain the equivalent results. Finally, you will create graphical output that also can be used to evaluate the output of classification models, such as binary logistic regression.

#### Question1: The Data set

	pregnant	glucose	diastolic	skinfold	insulin	bmi	pedigree	age	class	scored.class	scored.probability
176	5	144	82	26	285	32.0	0.452	58	1	1	0.6764516
177	5	123	74	40	77	34.1	0.269	28	0	0	0.3114196
178	4	146	78	0	0	38.5	0.520	67	1	1	0.7072096
179	8	188	78	0	0	47.9	0.137	43	1	1	0.8882766
180	9	120	72	22	56	20.8	0.733	48	0	0	0.4224679
181	0	102	86	17	105	29.3	0.695	27	0	0	0.1199810

From the provided dataset, it appears that the dependent variable – class, was regressed against several variables. The scored.class variable is the predicted variable and the scored.probability class gives the probability that the scored.class belongs to a class of 1.

#### Question 2: Raw confusion matrix

```
key_columns_df <- dplyr::select(data, scored.class, class)
table(key_columns_df)
```

```
##           class
## scored.class  0  1
##           0 119 30
##           1   5 27
```

The rows represent the predicted, the columns represent the actual classes.

#### Question 3: Accuracy of predictions

```

accuracy_fn <- function(df){
  TP <- sum(df$class == 1 & df$scored.class == 1)
  TN <- sum(df$class == 0 & df$scored.class == 0)
  accuracy <- (TP + TN)/nrow(df)
  return(accuracy)
}

# Run the function on the dataset
accuracy_fn(data)

```

```
## [1] 0.8066298
```

#### Question 4: Classification error rate of the predictions

```

cf_err_rate_fn <- function(df){
  FP <- sum(df$class == 0 & df$scored.class == 1)
  FN <- sum(df$class == 1 & df$scored.class == 0)
  cf_err_rate <- (FP + FN)/nrow(df)
  return(cf_err_rate)
}

# run the function on the dataset
cf_err_rate_fn(data)

```

```
## [1] 0.1933702
```

The sum of Accuracy and Error rates:

```
print(paste0("The sum of Accuracy and Error Rates: ", (accuracy_fn(data) + cf_err_rate_fn(data))))
```

```
## [1] "The sum of Accuracy and Error Rates: 1"
```

#### Question 5: Precision of the predictions

```

precision_fn <- function(df){
  TP <- sum(df$class == 1 & df$scored.class == 1)
  FP <- sum(df$class == 0 & df$scored.class == 1)
  precision <- (TP)/(TP+FP)
  return(precision)
}

#Run it on the dataset
precision_fn(data)

```

```
## [1] 0.84375
```

#### Question 6: Sensitivity of the predictions

```
sensitivity_fn <- function(df){
  TP <- sum(df$class == 1 & df$scored.class == 1)
  FN <- sum(df$class == 1 & df$scored.class == 0)
  sensitivity <- (TP)/(TP+FN)
  return(sensitivity)
}
```

```
#Running it on the data
sensitivity_fn(data)
```

```
## [1] 0.4736842
```

### Question 7: Specificity of the predictions

```
specificity_fn <- function(df){
  TN <- sum(df$class == 0 & df$scored.class == 0)
  FP <- sum(df$class == 0 & df$scored.class == 1)
  specificity <- (TN)/(TN+FP)
  return(specificity)
}
```

```
#Run it on the data
specificity_fn(data)
```

```
## [1] 0.9596774
```

### Question 8: F1 score of the predictions

```
f1_score_fn <- function(df){
  f1_score <- (2*precision_fn(df)*sensitivity_fn(df))/(precision_fn(df)+sensitivity_fn(df))
  return(f1_score)
}
```

```
#Run it on the data
f1_score_fn(data)
```

```
## [1] 0.6067416
```

### Question 9: Bounds of F1 score

Precision values between from 0 to 1:  $0 \leq p \leq 1$

Sensitivity values also range between 0 to 1:  $0 \leq s \leq 1$

Using the relation: If  $0 < a < 1$  and  $0 < b < 1$  then  $a \cdot b < a$ , we have:  $p \cdot s \leq s$  and  $p \cdot s \leq p$

This implies that:  $0 \leq p \cdot s \leq p \leq 1$  and  $0 \leq p \cdot s \leq s \leq 1$

The numerator in the equation ranges from 0 to 1

The denominator ranges from 0 to 2 Hence the quotient will range from 0 to 1.

### Question 10: ROC Curve

```

roc_curve_fn <- function(df){
  # sequence of thresholds ranging from 0 to 1 at 0.01 intervals.
  seq_int <- seq(0,1,by=0.01)
  TPR_vector <- c()
  FPR_vector <- c()

  for (i in 1:length(seq_int)){
    scored_class <- ifelse(df$scored.probability >= seq_int[i], 1, 0)
    rev_df <- data.frame(scored.class = scored_class, class = df$class)
    df_table <- with(rev_df, table(scored.class, class))
    TPR <- (df_table[4])/(df_table[4] + df_table[3])
    FPR <- (df_table[2])/(df_table[2] + df_table[1])
    TPR_vector[i] <- TPR
    FPR_vector[i] <- FPR
  }

  plot_df <- data.frame(TRUE_POSITIVE = TPR_vector, FALSE_POSITIVE = FPR_vector)
  ROC_plot <- ggplot(plot_df, aes(x=FALSE_POSITIVE, y=TRUE_POSITIVE)) + geom_point() + geom_line(col="blue") + geom_abline(intercept = 0, slope = 1) + labs(title="ROC Curve for the Dataset", x = "False Positive Rate (1 - Specificity)", y = "True Positive Rate (Sensitivity)")

  # Remove the NA values to calculate area under the curve
  auc_df <- plot_df[complete.cases(plot_df),]

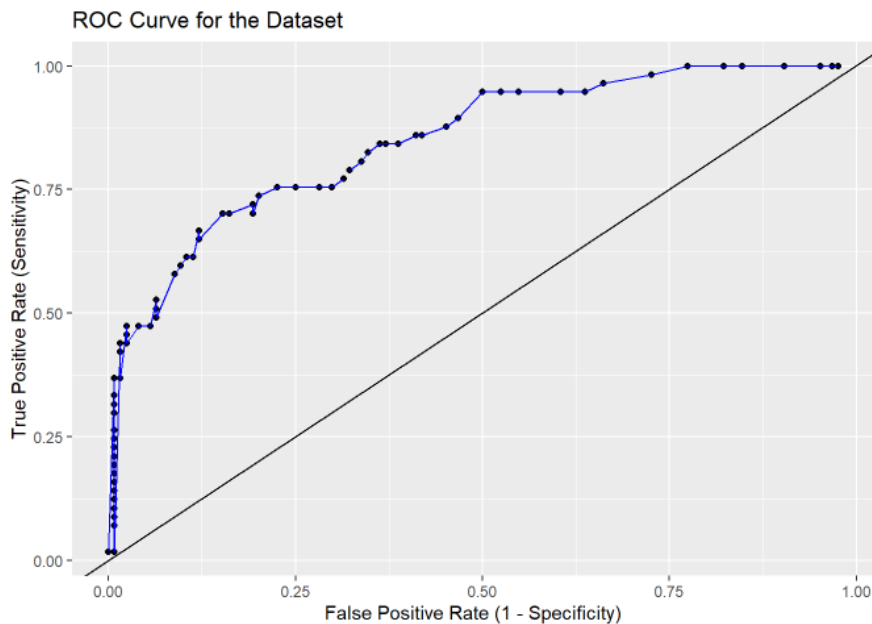
  # Calculation AUC (Area under the curve)
  x <- abs(diff(auc_df$FALSE_POSITIVE))
  y <- auc_df$TRUE_POSITIVE

  area_under_curve <- sum(x*y)

  return(list(ROC_plot, area_under_curve))
}

ROC_list <- roc_curve_fn(data)

```



```
print(paste0("Area Under the Curve: ", area_under_curve))
```

```
## [1] "Area Under the Curve: 0.829937747594793"
```

Question 11: All classification metrics

```

functions_df <- c(accuracy_fn(data),
                 cf_err_rate_fn(data),
                 precision_fn(data),
                 sensitivity_fn(data),
                 specificity_fn(data),
                 f1_score_fn(data))
names(functions_df) <- c("Accuracy", "Classification Error", "Precision",
                        "Sensitivity", "Specificity", "F1 Score")
functions_df<-as.data.frame(functions_df)
names(functions_df)[1]<-'Scores'
kbl(functions_df)%>%
  kable_classic("hover", full_width = F, html_font = "Helvetica")

```

	Scores
Accuracy	0.8066298
Classification Error	0.1933702
Precision	0.8437500
Sensitivity	0.4736842
Specificity	0.9596774
F1 Score	0.6067416

## Question 12: Caret package

```

scored_df <- data %>%
  select(scored.class, class) %>%
  mutate(scored.class = as.factor(scored.class),
         class = as.factor(class))

c_matrix <- confusionMatrix(scored_df$scored.class, scored_df$class, positive = "1")

Caret_Package <- c(c_matrix$overall["Accuracy"], c_matrix$byClass["Sensitivity"], c_matrix$byClass["Specificity"]
)
Written_Functions <- c(accuracy_fn(data), sensitivity_fn(data), specificity_fn(data))
d <- cbind(Caret_Package, Written_Functions)
kbl(d)%>%
  kable_classic("hover", full_width = F, html_font = "Garamond")

```

	Caret_Package	Written_Functions
Accuracy	0.8066298	0.8066298
Sensitivity	0.4736842	0.4736842
Specificity	0.9596774	0.9596774

The results from the caret package and the functions Confusion matrix, sensitivity, and specificity are the same.

## Question 13: pROC package

```

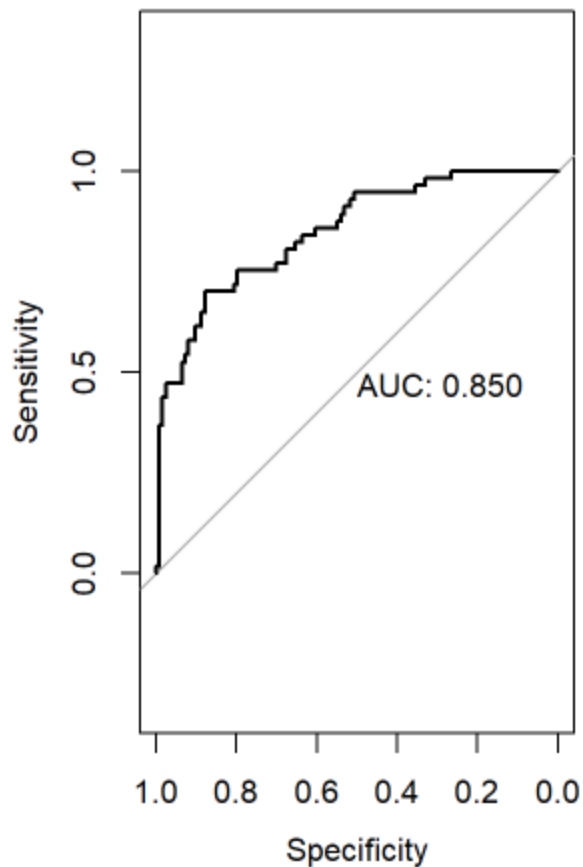
#The results are exactly the same
par(mfrow = c(1, 2))
plot(roc(data$class, data$scored.probability), print.auc = TRUE, main="ROC Curve from pROC Package")

```

```
## Setting levels: control = 0, case = 1
```

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

## ROC Curve from pROC Package



This plot looks very similar to our manual plot, the only difference being the area under the curve.

## References

- <https://www.kaggle.com/kumargh/pimaindiansdiabetescsv>
- [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix)
- <https://rdrr.io/cran/caret/man/sensitivity.html>
- <https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>

## Code

GitHub: [https://github.com/nathtrish334/Data-621/blob/main/TrishitaNath\\_Data\\_621\\_Hw\\_02.pdf](https://github.com/nathtrish334/Data-621/blob/main/TrishitaNath_Data_621_Hw_02.pdf)

RPubs: <https://rpubs.com/trishitanath/819863>