Assignment – 2

Question-1:

Load the hurricane dataset.

Code:

```
# Loading the dataset
file_name <- loadWorkbook("E:/Downloads/hurricanesNew.xlsx")
data <- read.xlsx(file_name, sheet = "hurricanes")
str(data)
```

Output:

```
> data
   RowNames Number     Name Year Type FirstLat FirstLon MaxLat MaxLon LastLat LastLon MaxInt
1         1    430 NOTNAMED 1944    1     30.2    -76.1   32.1  -74.8    35.1   -69.2     80
2         2    432 NOTNAMED 1944    0     25.6    -74.9   31.0  -78.1    32.6   -78.2     80
3         3    433 NOTNAMED 1944    0     14.2    -65.2   16.6  -72.2    20.6   -88.5    105
4         4    436 NOTNAMED 1944    0     20.8    -58.0   26.3  -72.3    42.1   -71.5    120
5         5    437 NOTNAMED 1944    0     20.0    -84.2   20.6  -84.9    19.1   -93.9     70
6         6    438 NOTNAMED 1944    1     29.2    -55.8   38.0  -53.2    50.0   -46.5     85
7         7    440 NOTNAMED 1944    0     16.1    -80.8   21.9  -82.9    28.4   -82.1    105
8         8    441 NOTNAMED 1945    1     27.6    -85.6   27.6  -85.6    31.7   -79.1    100
9         9    445 NOTNAMED 1945    0     21.6    -95.2   28.6  -96.1    29.5   -96.0    120
10       10    449 NOTNAMED 1945    0     19.0    -56.6   24.9  -79.6    28.9   -81.8    120
11       11    450 NOTNAMED 1945    0     16.2    -82.6   16.5  -85.6    16.4   -88.3     85
12       12    451 NOTNAMED 1945    0     19.6    -80.2   21.6  -79.3    29.9   -68.0     85
13       13    453 NOTNAMED 1946    1     36.5    -72.3   36.7  -70.8    39.0   -63.0     70
14       14    455 NOTNAMED 1946    0     26.4    -77.9   28.4  -75.0    40.7   -66.0     85
15       15    456 NOTNAMED 1946    0     19.6    -85.6   25.4  -83.2    27.0   -82.8    115
16       16    459 NOTNAMED 1947    0     21.0    -92.5   22.0  -96.4    22.0   -97.2     95
17       17    460 NOTNAMED 1947    0     26.5    -90.6   26.9  -91.2    29.2   -94.8     70
18       18    461 NOTNAMED 1947    0     14.1    -24.0   26.5  -75.4    30.4   -91.0    140
19       19    465 NOTNAMED 1947    0     24.1    -82.3   25.8  -80.6    31.8   -82.3     75
20       20    466 NOTNAMED 1947    0     19.7    -66.6   31.4  -66.9    37.5   -59.0    105
21       21    469 NOTNAMED 1948    0     20.9    -61.0   27.6  -70.4    37.0   -68.7    105
22       22    471 NOTNAMED 1948    0     25.8    -92.6   26.6  -91.9    28.8   -90.5     70
23       23    472 NOTNAMED 1948    0     14.3    -23.0   28.7  -64.6    46.9   -48.8    115
24       24    473 NOTNAMED 1948    0     18.5    -80.8   24.3  -81.7    37.1   -66.9    105
25       25    474 NOTNAMED 1948    0     19.4    -85.1   23.3  -82.5    32.2   -51.3    115
26       26    475 NOTNAMED 1948    1     25.9    -68.8   26.3  -70.8    30.1   -74.4     70
27       27    476 NOTNAMED 1949    0     22.3    -64.7   30.9  -76.2    44.2   -49.3     95
28       28    477 NOTNAMED 1949    0     23.4    -73.0   26.1  -79.0    28.3   -82.2    130
29       29    479 NOTNAMED 1949    0     20.9    -66.6   31.7  -63.5    45.5   -55.1    110
30       30    483 NOTNAMED 1949    0     23.0    -94.9   21.9  -95.9    20.3   -95.8     85
31       31    484 NOTNAMED 1949    0     16.4    -65.3   16.9  -66.6    18.2   -69.9     70
32       32    485 NOTNAMED 1949    0     22.0    -94.3   29.1  -95.4    29.1   -95.4    115
33       33    486 NOTNAMED 1949    0     24.2    -71.9   32.4  -68.3    35.7   -65.5     90
34       34    489     ABLE 1950    0     21.0    -62.5   26.1  -73.8    41.8   -67.0    120
35       35    490    BAKER 1950    0     16.5    -57.4   16.7  -60.0    30.8   -87.8    105
36       36    491  CHARLIE 1950    0     22.0    -52.8   29.2  -58.0    38.4   -58.1    100
37       37    492      DOG 1950    0     15.7    -56.5   26.7  -68.4    40.5   -68.8    160
38       38    493     EASY 1950    0     21.0    -82.8   27.4  -83.2    28.2   -82.2    110
39       39    494      FOX 1950    0     18.9    -50.2   24.6  -59.4    41.9   -42.8    120
```

Interpretation:

First of all, the openxlsx library is installed and then it is called, which is then used to create a dataset named file_name, that loads the hurricane dataset to it. Then we extract the sheet containing “hurricanes” and analyse it through a data frame called data.

Question 2:

Preprocess the dataset to convert it into a binary classification problem by considering only tropical hurricanes and non-tropical hurricanes. Thus type 1 and type 3 will be represented as non-tropical hurricanes.

Code:

```r
# Map the target variable to binary format
# Making tropical region or type = 0 as 1 and else as 0
data$binary_target <- ifelse(data$Type == 0, 0, 1)
```

Output:

```
> data
   RowNames Number     Name Year Type FirstLat FirstLon MaxLat MaxLon LastLat LastLon MaxInt binary_target
1         1    430 NOTNAMED 1944    1     30.2    -76.1   32.1  -74.8    35.1   -69.2     80             1
2         2    432 NOTNAMED 1944    0     25.6    -74.9   31.0  -78.1    32.6   -78.2     80             0
3         3    433 NOTNAMED 1944    0     14.2    -65.2   16.6  -72.2    20.6   -88.5    105             0
4         4    436 NOTNAMED 1944    0     20.8    -58.0   26.3  -72.3    42.1   -71.5    120             0
5         5    437 NOTNAMED 1944    0     20.0    -84.2   20.6  -84.9    19.1   -93.9     70             0
6         6    438 NOTNAMED 1944    1     29.2    -55.8   38.0  -53.2    50.0   -46.5     85             1
7         7    440 NOTNAMED 1944    0     16.1    -80.8   21.9  -82.9    28.4   -82.1    105             0
8         8    441 NOTNAMED 1945    1     27.6    -85.6   27.6  -85.6    31.7   -79.1    100             1
9         9    445 NOTNAMED 1945    0     21.6    -95.2   28.6  -96.1    29.5   -96.0    120             0
10       10    449 NOTNAMED 1945    0     19.0    -56.6   24.9  -79.6    28.9   -81.8    120             0
11       11    450 NOTNAMED 1945    0     16.2    -82.6   16.5  -85.6    16.4   -88.3     85             0
12       12    451 NOTNAMED 1945    0     19.6    -80.2   21.6  -79.3    29.9   -68.0     85             0
13       13    453 NOTNAMED 1946    1     36.5    -72.3   36.7  -70.8    39.0   -63.0     70             1
14       14    455 NOTNAMED 1946    0     26.4    -77.9   28.4  -75.0    40.7   -66.0     85             0
15       15    456 NOTNAMED 1946    0     19.6    -85.6   25.4  -83.2    27.0   -82.8    115             0
16       16    459 NOTNAMED 1947    0     21.0    -92.5   22.0  -96.4    22.0   -97.2     95             0
17       17    460 NOTNAMED 1947    0     26.5    -90.6   26.9  -91.2    29.2   -94.8     70             0
18       18    461 NOTNAMED 1947    0     14.1    -24.0   26.5  -75.4    30.4   -91.0    140             0
19       19    465 NOTNAMED 1947    0     24.1    -82.3   25.8  -80.6    31.8   -82.3     75             0
20       20    466 NOTNAMED 1947    0     19.7    -66.6   31.4  -66.9    37.5   -59.0    105             0
21       21    469 NOTNAMED 1948    0     20.9    -61.0   27.6  -70.4    37.0   -68.7    105             0
22       22    471 NOTNAMED 1948    0     25.8    -92.6   26.6  -91.9    28.8   -90.5     70             0
23       23    472 NOTNAMED 1948    0     14.3    -23.0   28.7  -64.6    46.9   -48.8    115             0
24       24    473 NOTNAMED 1948    0     18.5    -80.8   24.3  -81.7    37.1   -66.9    105             0
25       25    474 NOTNAMED 1948    0     19.4    -85.1   23.3  -82.5    32.2   -51.3    115             0
26       26    475 NOTNAMED 1948    1     25.9    -68.8   26.3  -70.8    30.1   -74.4     70             1
27       27    476 NOTNAMED 1949    0     22.3    -64.7   30.9  -76.2    44.2   -49.3     95             0
28       28    477 NOTNAMED 1949    0     23.4    -73.0   26.1  -79.0    28.3   -82.2    130             0
29       29    479 NOTNAMED 1949    0     20.9    -66.6   31.7  -63.5    45.5   -55.1    110             0
30       30    483 NOTNAMED 1949    0     23.0    -94.9   21.9  -95.9    20.3   -95.8     85             0
31       31    484 NOTNAMED 1949    0     16.4    -65.3   16.9  -66.6    18.2   -69.9     70             0
32       32    485 NOTNAMED 1949    0     22.0    -94.3   29.1  -95.4    29.1   -95.4    115             0
33       33    486 NOTNAMED 1949    0     24.2    -71.9   32.4  -68.3    35.7   -65.5     90             0
34       34    489     ABLE 1950    0     21.0    -62.5   26.1  -73.8    41.8   -67.0    120             0
35       35    490    BAKER 1950    0     16.5    -57.4   16.7  -60.0    30.8   -87.8    105             0
36       36    491  CHARLIE 1950    0     22.0    -52.8   29.2  -58.0    38.4   -58.1    100             0
37       37    492      DOG 1950    0     15.7    -56.5   26.7  -68.4    40.5   -68.8    160             0
38       38    493     EASY 1950    0     21.0    -82.8   27.4  -83.2    28.2   -82.2    110             0
39       39    494      FOX 1950    0     18.9    -50.2   24.6  -59.4    41.9   -42.8    120             0
```

Interpretation:

A vector is created by the name binary_target and then the column ‘Type’ of the dataset in data is updated with the values 0 and 1. When the value is Tropical or 0 in ‘Type’, assign it 0, else assign it 1.

Question 3:

Split the dataset into training and testing sets.

Code:

```r
# Train-Test Split
# Typically, the data is split into a majority (80%) for training and a minority (20%) for testing.
set.seed(123) # for reproducibility
train_index <- createDataPartition(data$binary_target, p = 0.8, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]
```

Output:

```
> train_data
  RowNames Number     Name Year Type FirstLat FirstLon MaxLat MaxLon LastLat LastLon MaxInt binary_target
1        1    430 NOTNAMED 1944    1     30.2    -76.1   32.1  -74.8    35.1   -69.2     80             1
2        2    432 NOTNAMED 1944    0     25.6    -74.9   31.0  -78.1    32.6   -78.2     80             0
4        4    436 NOTNAMED 1944    0     20.8    -58.0   26.3  -72.3    42.1   -71.5    120             0
5        5    437 NOTNAMED 1944    0     20.0    -84.2   20.6  -84.9    19.1   -93.9     70             0
6        6    438 NOTNAMED 1944    1     29.2    -55.8   38.0  -53.2    50.0   -46.5     85             1
7        7    440 NOTNAMED 1944    0     16.1    -80.8   21.9  -82.9    28.4   -82.1    105             0
8        8    441 NOTNAMED 1945    1     27.6    -85.6   27.6  -85.6    31.7   -79.1    100             1
9        9    445 NOTNAMED 1945    0     21.6    -95.2   28.6  -96.1    29.5   -96.0    120             0
10      10    449 NOTNAMED 1945    0     19.0    -56.6   24.9  -79.6    28.9   -81.8    120             0
11      11    450 NOTNAMED 1945    0     16.2    -82.6   16.5  -85.6    16.4   -88.3     85             0
13      13    453 NOTNAMED 1946    1     36.5    -72.3   36.7  -70.8    39.0   -63.0     70             1
14      14    455 NOTNAMED 1946    0     26.4    -77.9   28.4  -75.0    40.7   -66.0     85             0
16      16    459 NOTNAMED 1947    0     21.0    -92.5   22.0  -96.4    22.0   -97.2     95             0
17      17    460 NOTNAMED 1947    0     26.5    -90.6   26.9  -91.2    29.2   -94.8     70             0
18      18    461 NOTNAMED 1947    0     14.1    -24.0   26.5  -75.4    30.4   -91.0    140             0
20      20    466 NOTNAMED 1947    0     19.7    -66.6   31.4  -66.9    37.5   -59.0    105             0
21      21    469 NOTNAMED 1948    0     20.9    -61.0   27.6  -70.4    37.0   -68.7    105             0
22      22    471 NOTNAMED 1948    0     25.8    -92.6   26.6  -91.9    28.8   -90.5     70             0
23      23    472 NOTNAMED 1948    0     14.3    -23.0   28.7  -64.6    46.9   -48.8    115             0
24      24    473 NOTNAMED 1948    0     18.5    -80.8   24.3  -81.7    37.1   -66.9    105             0
25      25    474 NOTNAMED 1948    0     19.4    -85.1   23.3  -82.5    32.2   -51.3    115             0
26      26    475 NOTNAMED 1948    1     25.9    -68.8   26.3  -70.8    30.1   -74.4     70             1
27      27    476 NOTNAMED 1949    0     22.3    -64.7   30.9  -76.2    44.2   -49.3     95             0
29      29    479 NOTNAMED 1949    0     20.9    -66.6   31.7  -63.5    45.5   -55.1    110             0
30      30    483 NOTNAMED 1949    0     23.0    -94.9   21.9  -95.9    20.3   -95.8     85             0
31      31    484 NOTNAMED 1949    0     16.4    -65.3   16.9  -66.6    18.2   -69.9     70             0
32      32    485 NOTNAMED 1949    0     22.0    -94.3   29.1  -95.4    29.1   -95.4    115             0
33      33    486 NOTNAMED 1949    0     24.2    -71.9   32.4  -68.3    35.7   -65.5     90             0
34      34    489     ABLE 1950    0     21.0    -62.5   26.1  -73.8    41.8   -67.0    120             0
35      35    490    BAKER 1950    0     16.5    -57.4   16.7  -60.0    30.8   -87.8    105             0
36      36    491  CHARLIE 1950    0     22.0    -52.8   29.2  -58.0    38.4   -58.1    100             0
37      37    492      DOG 1950    0     15.7    -56.5   26.7  -68.4    40.5   -68.8    160             0
38      38    493     EASY 1950    0     21.0    -82.8   27.4  -83.2    28.2   -82.2    110             0
39      39    494      FOX 1950    0     18.9    -50.2   24.6  -59.4    41.9   -42.8    120             0
40      40    495   GEORGE 1950    1     30.3    -63.8   33.0  -68.0    44.6   -56.7     95             1
41      41    497     ITEM 1950    1     21.0    -93.2   19.9  -95.3    18.8   -95.9     95             1
```

Interpretation:

2 new datasets are created by the name test_data and train_data by splitting the new dataset, where train_data contains 80% of the data and the rest 20% is stored by test_data.

Question 4 & Question 5:

Implement logistic regression classifiers using appropriate R packages. Also, train the classifier using a single predictor

Code:

```
# Fit logistic regression model using a single predictor
# Let's say the predictor be FirstLat
# Train logistic regression model
logistic_model <- glm(binary_target~FirstLat, data = data, family = "binomial")

# Summarize the model
summary(logistic_model)

# Making the predictions on training data
train_predictions <- predict(logistic_model, type = "response")

# Convert predicted probabilities to class labels (Tropical or Non-Tropical)
train_predicted_classes <- ifelse(train_predictions > 0.5 , 1, 0)

# Compute accuracy on the training data
train_accuracy <- mean(train_predicted_classes == data$binary_target)
cat("We get Training Accuracy as:", train_accuracy, "\n")
```

Output:

```
> # Summarize the model
> summary(logistic_model)

Call:
glm(formula = binary_target ~ FirstLat, family = "binomial",
    data = data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.08263    0.96148  -9.446   <2e-16 ***
FirstLat     0.37283    0.03947   9.447   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 463.11  on 336  degrees of freedom
Residual deviance: 232.03  on 335  degrees of freedom
AIC: 236.03

Number of Fisher Scoring iterations: 6

>
> # Making the predictions on training data
> train_predictions <- predict(logistic_model, type = "response")
>
> # Convert predicted probabilities to class labels (Tropical or Non-Tropical)
> train_predicted_classes <- ifelse(train_predictions > 0.5 , 1, 0)
>
> # Compute accuracy on the training data
> train_accuracy <- mean(train_predicted_classes == data$binary_target)
> cat("We get Training Accuracy as:", train_accuracy, "\n")
We get Training Accuracy as: 0.851632
```

Interpretation:

Logistic regression is implemented using glm package. Also, accuracy is found on the testing data which is quite huge (suggesting our approach is correct).

Question 6:

Evaluate the performance of the classifier for five different thresholds (0.5, 0.6, 0.7, 0.8, 0.9) using the following metrics:

_ Residual Difference

_ Confusion matrix

_ Accuracy

_ Precision

_ Recall

_ F-measure

_ ROC Curve.

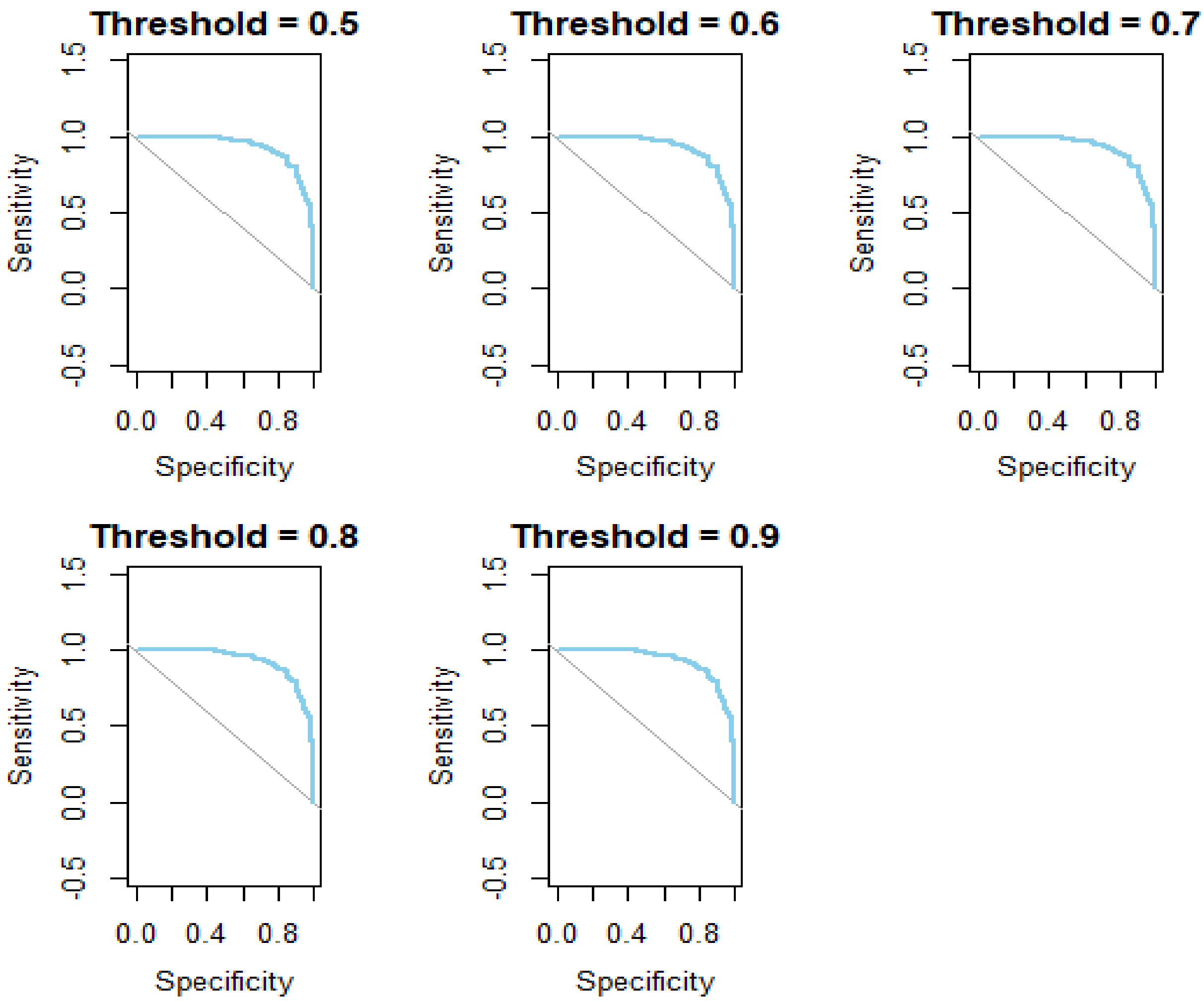
Code:

```r
# Define threshold values
thresholds <- c(0.5, 0.6, 0.7, 0.8, 0.9)

# Required lists that will store the results
residual_differences <- numeric(length(thresholds))
confusion_matrices <- vector("list", length(thresholds))
accuracies <- numeric(length(thresholds))
precisions <- numeric(length(thresholds))
recalls <- numeric(length(thresholds))
f_measures <- numeric(length(thresholds))

# Evaluate the classifier for each threshold
for (i in seq_along(thresholds)) {
  # Adjust threshold for class prediction
  threshold <- thresholds[i]
  train_predicted_classes <- ifelse(train_predictions > threshold, 0, 1)

  # Calculate residual difference
  residual_differences[i] <- mean(train_predicted_classes != data$binary_target)

  # Compute confusion matrix
  confusion_matrices[[i]] <- table(Actual = data$binary_target, Predicted = train_predicted_classes)

  # Compute accuracy
  accuracies[i] <- mean(train_predicted_classes == data$binary_target)

  confusion_matrix <- table(Actual = data$binary_target, Predicted = train_predicted_classes)
  print(confusion_matrix)

  # Compute precision and recall
  if (0 %in% rownames(confusion_matrix) && 0 %in% colnames(confusion_matrix)) {
    TP <- confusion_matrix[1, 1]  # True Positives
    FP <- confusion_matrix[2, 1]  # False Positives
    FN <- confusion_matrix[1, 2]  # False Negatives
    if ((TP + FN) > 0) {
      recalls[i] <- TP / (TP + FN)
    } else {
      recalls[i] <- 0  # Set recall to zero if there are no true positives or false negatives
    }
```

```r
    if ((TP + FN) > 0) {
      recalls[i] <- TP / (TP + FN)
    } else {
      recalls[i] <- 0  # Set recall to zero if there are no true positives or false negatives
    }

    if ((TP + FP) > 0) {
      precisions[i] <- TP / (TP + FP)
    } else {
      precisions[i] <- 0  # Set precision to zero if there are no true positives
    }
  }
  else {
    precisions[i] <- NA  # Set precision to NA if class '0' is not present in confusion matrix
    recalls[i] <- NA  # Set recall to NA if class '0' is not present in confusion matrix
  }

  # Compute F-measure
  if (precisions[i] + recalls[i] > 0) {
    f_measures[i] <- 2 * precisions[i] * recalls[i] / (precisions[i] + recalls[i])
  } else {
    f_measures[i] <- 0  # Set F-measure to zero if both precision and recall are zero
  }
}
```

Output:

```
At threshold: 0.5

Residual Difference:  0.851632
Confusion Matrix:
 24 124 163 26
Accuracy: 0.148368
Precision: 0.1621622
Recall: 0.1283422
F-measure: 0.1432836
-----************---------

At threshold: 0.6

Residual Difference:  0.8456973
Confusion Matrix:
 18 116 169 34
Accuracy: 0.1543027
Precision: 0.1343284
Recall: 0.09625668
F-measure: 0.1121495
-----************---------

At threshold: 0.7

Residual Difference:  0.8308605
Confusion Matrix:
 11 104 176 46
Accuracy: 0.1691395
Precision: 0.09565217
Recall: 0.05882353
F-measure: 0.07284768
-----************---------
At threshold: 0.8

Residual Difference:  0.7982196
Confusion Matrix:
 5 87 182 63
Accuracy: 0.2017804
Precision: 0.05434783
Recall: 0.02673797
F-measure: 0.03584229
-----************---------

At threshold: 0.9

Residual Difference:  0.735905
Confusion Matrix:
 2 63 185 87
Accuracy: 0.264095
Precision: 0.03076923
Recall: 0.01069519
F-measure: 0.01587302
-----************---------
```



Interpretation:

Change in the threshold results in difference in performance matrix. Precision decreases with the increasing value of threshold and recall increases with the increasing value of threshold. Thus, F1 score also decrease with increase in threshold. Accuracy also follows the same suit. This is also reflected in the ROC curve as well.

Question 7:

Carry out the same experiment with two predictor variables- FirstLat and FirstLon. Compare the result with the previous model.

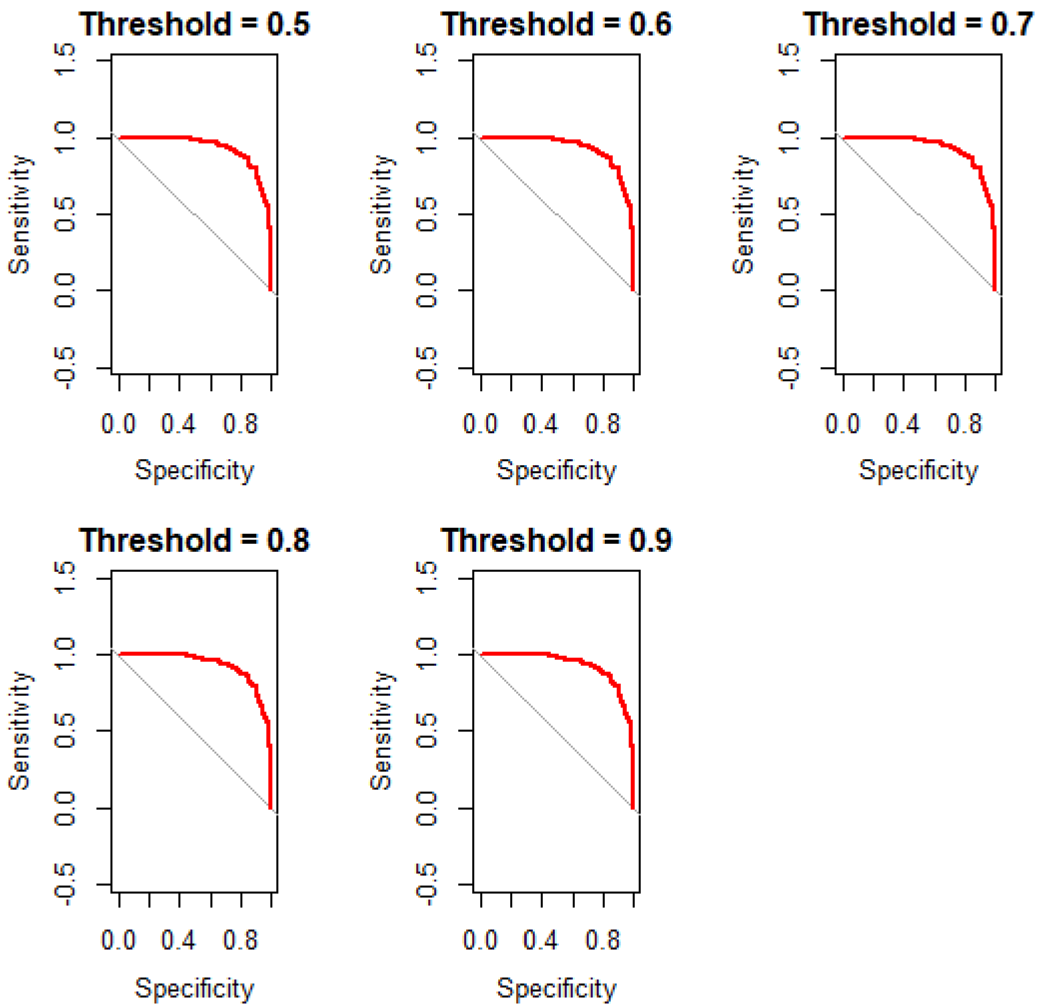
Code:

Remains the same as previous model but only a minor change shown in below image.

```
model_<-glm(binary_target ~ FirstLat+FirstLon, data = data, family = binomial)
summary(model)

# Making the predictions on training data
train_predictions <- predict(model_, type = "response")
```

Output:

```
At threshold: 0.5

Residual Difference:  0.8545994
Confusion Matrix:
 24 125 163 25
Accuracy: 0.1454006
Precision: 0.1610738
Recall: 0.1283422
F-measure: 0.1428571
-----************---------

At threshold: 0.6

Residual Difference:  0.8486647
Confusion Matrix:
 18 117 169 33
Accuracy: 0.1513353
Precision: 0.1333333
Recall: 0.09625668
F-measure: 0.1118012
-----************---------

At threshold: 0.7

Residual Difference:  0.8278932
Confusion Matrix:
 12 104 175 46
Accuracy: 0.1721068
Precision: 0.1034483
Recall: 0.06417112
F-measure: 0.07920792
-----************---------

At threshold: 0.8

Residual Difference:  0.7982196
Confusion Matrix:
 6 88 181 62
Accuracy: 0.2017804
Precision: 0.06382979
Recall: 0.03208556
F-measure: 0.04270463
-----************---------

At threshold: 0.9

Residual Difference:  0.7329377
Confusion Matrix:
 1 61 186 89
Accuracy: 0.2670623
Precision: 0.01612903
Recall: 0.005347594
F-measure: 0.008032129
-----************---------
```



Interpretation:

The increase in the number of predictors gives us a better model, which is indicated in the ROC Curve. This is due to feature engineering, where we must identify the features, which control the behaviour of the model carefully, to get a better model.