Module 3: GLM and Logistic Regression

Mohit Ravindra Kamble

College of Professional Studies – Northeastern University

ALY6015: Intermediate Analytics

Prof. Roy Wada

January 30, 2024

01. Overview:

In this analysis, I explored a dataset on college attributes using descriptive statistics and visualizations. After splitting the data into training and test sets, I employed the glm() function to fit a logistic regression model to predict whether a college is private or public based on acceptance and enrollment. The confusion matrix for the training set showed the model's performance, highlighting the balance between false positives and false negatives. Evaluating accuracy, precision, recall, and specificity metrics provided a comprehensive understanding of the model's strengths and weaknesses. Applying the model to the test set produced another confusion matrix, demonstrating its generalization ability. The ROC curve visually depicted the trade-off between sensitivity and specificity, with the Area Under the Curve (AUC) quantifying overall performance. This analysis helps in assessing the model's predictive power and understanding potential misclassifications, crucial for decision-making in diverse applications.

02. Analysis:

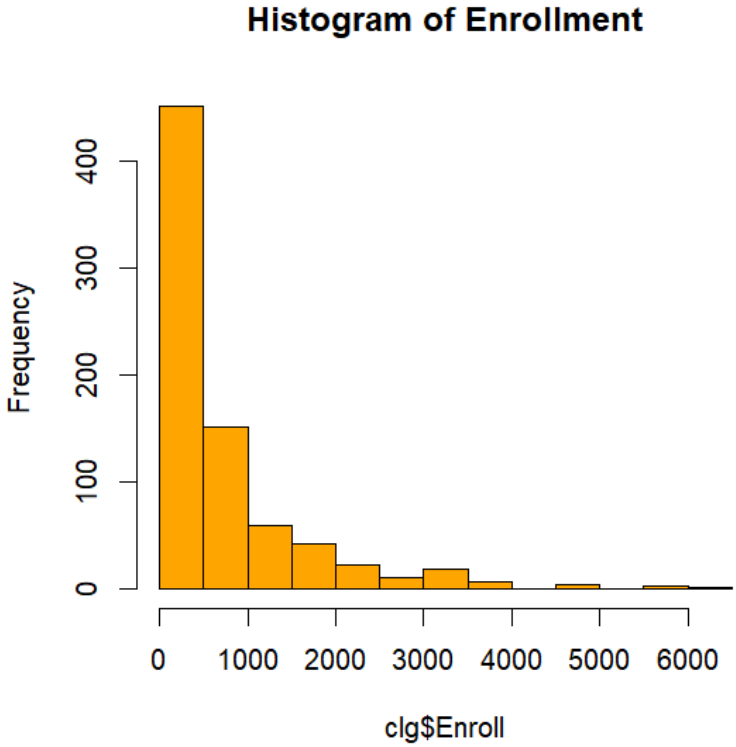
2.1) Import the dataset and perform Exploratory Data Analysis by using descriptive statistics and plots to describe the dataset.

➔ I began by installing and loading the 'ISLR' package to access the 'College' dataset. After attaching the dataset, I examined its contents using 'View(College)' and extracted variable names with 'names(College).' To better manipulate the data, I created a data frame called **'clg**.' Utilizing the 'summary()' function on 'clg' provided a quick snapshot of its statistical summary. This initial examination serves as a foundation for more in-depth exploratory data analysis in subsequent steps. Used str() to know the internal structure.

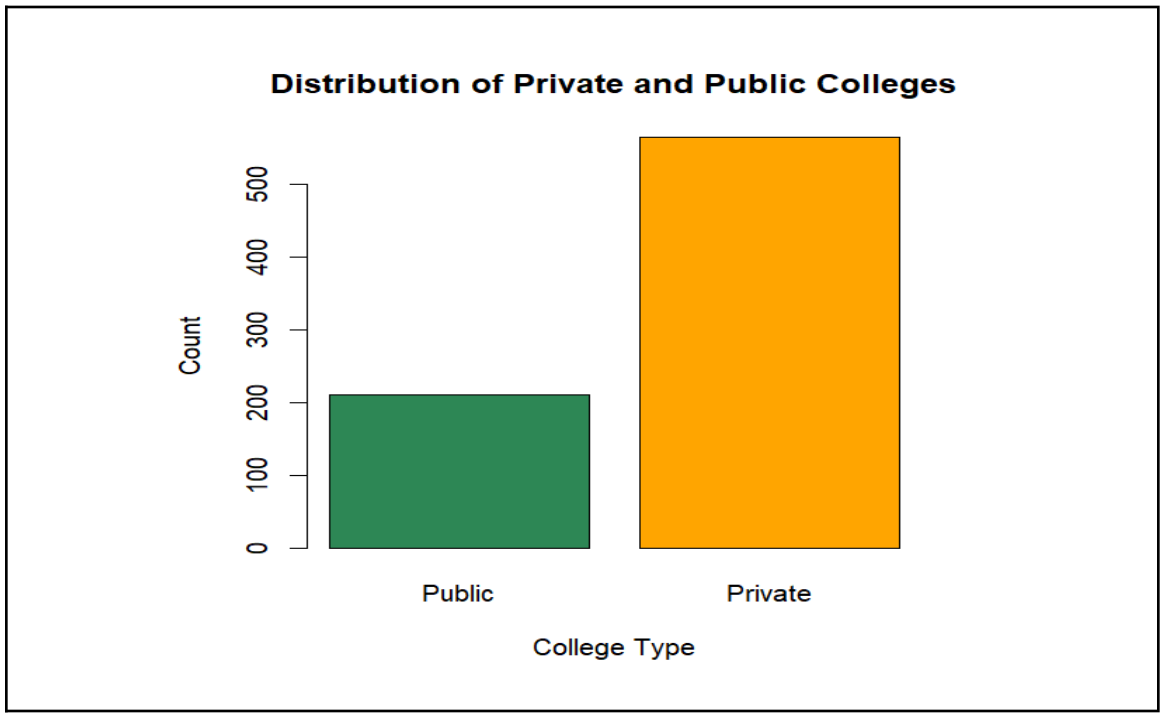
```
> str(clg)
'data.frame':      777 obs. of  18 variables:
 $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ Apps       : num  1660 2186 1428 417 193 ...
 $ Accept     : num  1232 1924 1097 349 146 ...
 $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
 $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
 $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
 $ F.Undergrad: num  2885 2683 1036 510 249 ...
 $ P.Undergrad: num  537 1227 99 63 869 ...
 $ Outstate   : num  7440 12280 11250 12960 7560 ...
 $ Room.Board : num  3300 6450 3750 5450 4120 ...
 $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
 $ Personal   : num  2200 1500 1165 875 1500 ...
 $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
 $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
 $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
 $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
```

```
$ Expend     : num  7041 10527 8735 19016 10922 ...
$ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

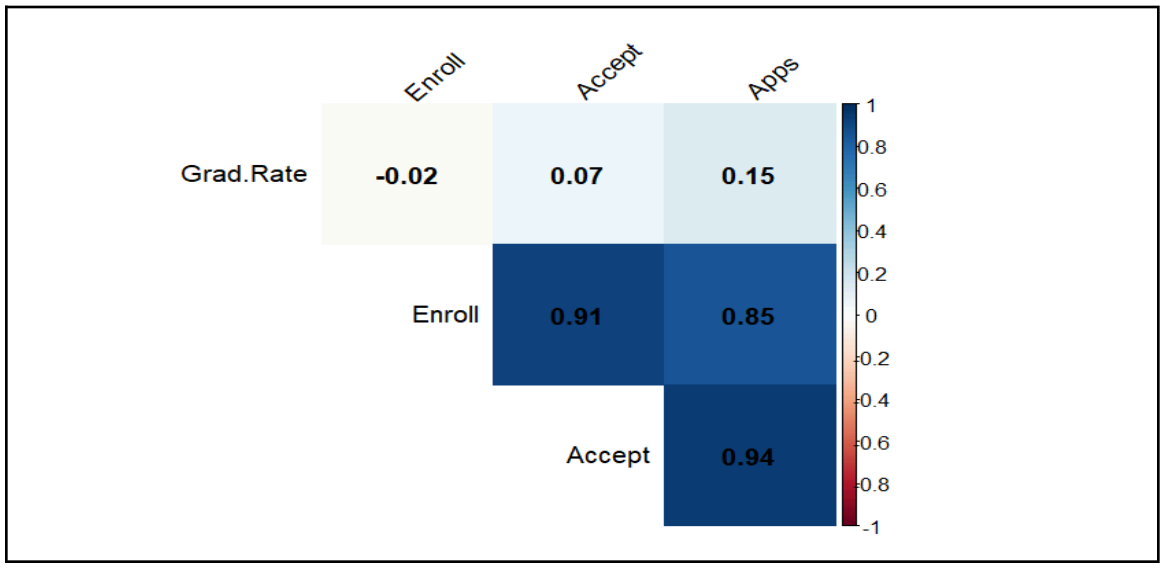
➔ created a histogram to visualize the distribution of college enrollments in the dataset. *The histogram of enrollment is right skewed.*



➔ I created a barplot to visualize the distribution of private and public colleges. Each bar represents the count of colleges falling into the respective categories, with 'seagreen' indicating public and 'orange' indicating private colleges. *There are more private colleges compared to public colleges.*



➔ I computed and visualized a correlation matrix for key variables in the 'College' dataset—'Enroll,' 'Accept,' 'Grad.Rate,' and 'Apps.' The color-coded matrix provides a quick overview of how these variables relate to each other. Darker colors indicate stronger correlations, with positive correlations represented in one color gradient and negative correlations in another.



★ **Insights:**

- From the correlation plot we can see the acceptance has strong positive relations with applications and enrollments.
- Graduation rate has a negative relation with enrollment.

2.2) Split the data into a train and test set – refer to the Feature_Selection_R.pdf document for information on how to split a dataset.

➔ I utilized the 'caTools' package to split the 'College' dataset into training and test sets. The code utilizes the 'caTools' package to split the 'College' dataset into training and test sets based on the 'Private' variable. A random seed ensures reproducibility, and a 70-30 split ratio is applied using 'sample.split()'. The resulting *split vector is used to subset the data into 'train_data' and 'test_data'.* The 'View()' function allows inspection of the contents and structures of the resulting datasets, facilitating subsequent model training and evaluation.

```
> head(train_data)
                            Private Apps Accept Enroll Top10perc Top25perc
F.Undergrad P.Undergrad Outstate Room.Board
Abilene Christian University     Yes 1660   1232    721        23        52
2885         537     7440       3300
Adrian College                   Yes 1428   1097    336        22        50
1036          99    11250       3750
Albertson College                Yes  587    479    158        38        62
678           41    13500       3335
Albertus Magnus College          Yes  353    340    103        17        45
416          230    13290       5720
Albright College                 Yes 1038    839    227        30        63
973          306    15595       4400
Alderson-Broaddus College        Yes  582    498    172        21        44
799           78    10468       3380

> head(test_data)
                              Private Apps Accept Enroll Top10perc Top25perc
F.Undergrad P.Undergrad Outstate Room.Board
Adelphi University                 Yes 2186   1924    512        16        29
2683        1227    12280       6450
Agnes Scott College                Yes  417    349    137        60        89
510           63    12960       5450
Alaska Pacific University          Yes  193    146     55        16        44
249          869     7560       4120
Albion College                     Yes 1899   1720    489        37        68
1594          32    13868       4826
Alfred University                  Yes 1732   1425    472        37        75
1830         110    16548       5406
American International College     Yes 1420   1093    220         9        22
1018         287     8700       4780
```

2.3) Use the glm() function in the 'stats' package to fit a logistic regression model to the training set using at least two predictors.

➔ I installed and loaded the 'caret' package for machine learning tasks. I specified the *response variable as "Private"* and selected *"Accept" and "Enroll" as predictors.* The logistic regression model was built using 'glm()', with the family parameter set to 'binomial' and a logit link function. The 'summary()' function was then applied to display essential information about the model's coefficients and significance levels. This step is crucial for understanding the predictive relationships between the chosen predictors and the outcome variable.

```
> summary(mod1)

Call:
glm(formula = Private ~ Accept + Enroll, family = binomial(link =
"logit"),
    data = train_data)

Coefficients:
             Estimate Std. Error z value             Pr(>|z|)
(Intercept)  2.7713910  0.2095844  13.223 < 0.0000000000000002 ***
Accept       0.0005489  0.0001694   3.241              0.00119 **
Enroll      -0.0036342  0.0005148  -7.059     0.00000000000167 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 636.80  on 543  degrees of freedom
Residual deviance: 417.41  on 541  degrees of freedom
AIC: 423.41

Number of Fisher Scoring iterations: 6
```

★ **Insights:**

- The logistic regression model indicates that 'Accept' and 'Enroll' are statistically significant predictors of whether a college is private.
- The positive coefficient for 'Accept' suggests that an increase in acceptance rate is associated with higher odds of being a private college.
- Conversely, the negative coefficient for 'Enroll' implies that higher enrollment is associated with lower odds of being private.
- The model's deviance values and AIC indicate a good fit, suggesting its effectiveness in predicting college type based on the given predictors.

2.4) Create a confusion matrix and report the results of your model for the train set. Interpret and discuss the confusion matrix. Which misclassifications are more damaging for the analysis, False Positives or False Negatives?

➔ I applied the logistic regression model to predict whether colleges in the training set are private. Utilizing the 'predict()' function, I obtained probabilities that were then converted into binary predictions using a threshold of 0.5. Subsequently, a confusion matrix ('cm_train') was generated to compare these predictions with the actual 'Private' values in the training set. The matrix offers insights into the model's performance, distinguishing between true positives, true negatives, false positives, and false negatives. This evaluation step is crucial for understanding the accuracy and effectiveness of the logistic regression model on the training data.

```
> cm_train
      Predicted
Actual   0   1
   No   79  69
   Yes  16 380
```

★ **Insights:**

- The confusion matrix for the logistic regression model on the training set is as follows:
- True Positives (Yes): 380
- True Negatives (No): 79
- False Positives: 69
- False Negatives: 16
- This matrix provides a breakdown of the model's predictions, showing how many instances were correctly or incorrectly classified as private (Yes) or not private (No). The model appears to perform well in correctly identifying private colleges (high True Positives) but has some misclassifications, as indicated by the False Positives and False Negatives.

2.5) Report and interpret metrics for Accuracy, Precision, Recall, and Specificity.

➔ I calculated key metrics, such as accuracy, sensitivity, specificity, precision, recall, and the F1 score, to evaluate the logistic regression model's performance on the training set. The metrics provide insights into how well the model correctly classifies private and non-private colleges, addressing both true positives and true negatives. The results were rounded to two decimal places for clarity, ensuring a concise presentation of the model's performance metrics. A table named *'metrics_table_train' was created to organize and summarize these metrics*, facilitating easy interpretation and comparison. This comprehensive evaluation assists in gauging the model's overall effectiveness and identifying potential areas for improvement.

```
> metrics_table_train
       Metric Value
1    Accuracy  0.84
2 Sensitivity  0.96
3 Specificity  0.53
4   Precision  0.85
5      Recall  0.96
6    F1 Score  0.90
```

★ **Insights:**

- The logistic regression model on the training set demonstrates overall good performance with an **accuracy of 84%.**
- While sensitivity (true positive rate) is high at 96%, specificity (true negative rate) is relatively lower at 53%, indicating a potential challenge in correctly identifying non-private colleges.

2.6) Create a confusion matrix and report the results of your model for the test set.

➔ I extended the logistic regression model to predict college types on the test set, converting probabilities to binary predictions and creating a confusion matrix ('cm_test'). The computed metrics for the test set include accuracy (rounded to 2 decimal places), sensitivity, specificity, precision, recall, and the F1 score. The resulting *'metrics_table_test' provides a clear overview of the model's performance on data*, aiding in understanding its generalization capabilities. Accuracy on the test set is a key indicator of the model's overall correctness, while sensitivity and specificity offer insights into its ability to correctly classify private and non-private colleges. This evaluation step is crucial for assessing the logistic regression model's practical utility and reliability when applied to new, previously unseen data.

```
> cm_test
      Predicted
Actual   0   1
   No   35  29
   Yes   8 161


> metrics_table_test
       Metric Value
1    Accuracy  0.84
2 Sensitivity  0.95
3 Specificity  0.55
4   Precision  0.85
5      Recall  0.95
6    F1 Score  0.90
```

★ **Insights:**

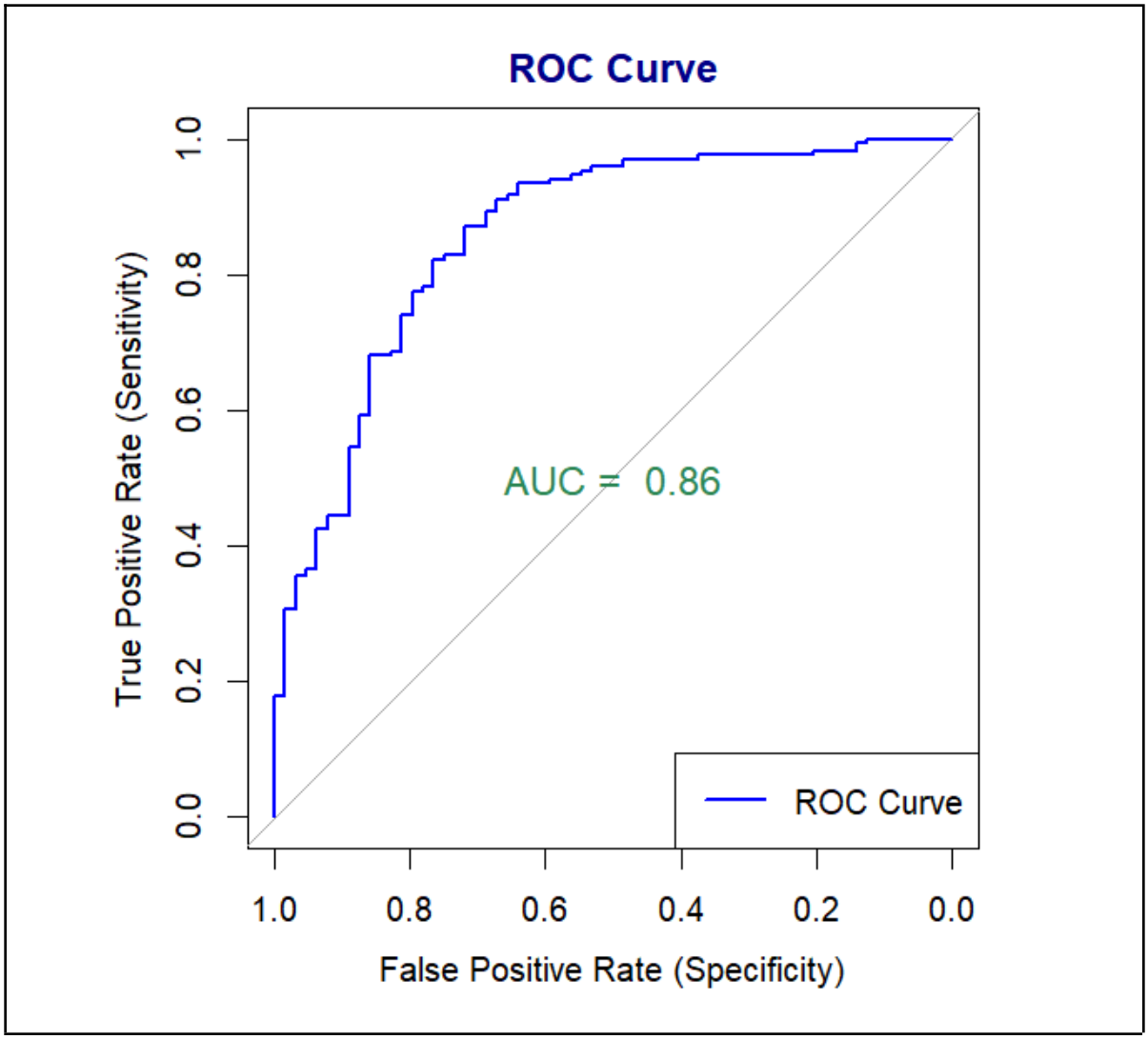
- The confusion matrix for the test set indicates 35 true negatives, 29 false positives, 8 false negatives, and 161 true positives.
- The logistic regression model exhibits an **accuracy of 84%** on the test set, with high sensitivity (95%), but relatively lower specificity (55%).
- Precision, measuring the proportion of correctly predicted positives among all predicted positives, is at 85%, reflecting the model's ability to minimize false positives.
- The F1 score, a balance between precision and recall, stands at 90%, suggesting an overall effective performance of the logistic regression model on the test data.

2.7) Plot and interpret the ROC curve.

❖ **ROC and AUC:**

The Receiver Operating Characteristic (ROC) curve is a graphical representation that illustrates the performance of a binary classification model across various decision thresholds. It plots the true positive rate (sensitivity) against the false positive rate (1-specificity), providing a visual trade-off analysis. A steeper ROC curve indicates better discrimination ability, while the Area Under the Curve (AUC) quantifies the model's overall performance. A higher AUC, ranging from 0 to 1, signifies superior predictive power, making the ROC curve and AUC valuable tools for assessing and comparing classification models.

➔ I utilized the 'pROC' package to construct and visualize the ROC curve for the logistic regression model on the test set. The 'roc()' function compared true 'Private' values to predicted probabilities, generating a curve that depicts the model's discrimination performance. The resulting ROC curve is plotted with customized aesthetics, facilitating a clear understanding of the trade-off between sensitivity and specificity. *Calculating the AUC here itself*, displayed on the plot, provides a quantitative summary of the model's overall discriminative ability. This visual representation enhances the interpretation of the logistic regression model's performance on the test data, particularly in terms of its ability to balance true positive and true negative rates.



★ **Insights:**

- The logistic regression model exhibits a high **AUC of 0.86**, indicating strong discriminative performance in distinguishing positive and negative cases.
- The ROC curve's steepness suggests a clear separation between true positive and false positive rates, emphasizing the model's effectiveness.
- The ROC curve starts with an ideal high true positive rate and a low false positive rate, showcasing the model's accuracy in classifying positive cases.
- The curve levels off at a true positive rate of approximately 1, indicating optimal performance in correctly identifying all positive cases.

03. Conclusion:

In conclusion, this analysis delved into a dataset of college attributes, employing descriptive statistics, visualizations, and logistic regression modeling. After splitting the data into training and test sets, I built a logistic regression model with 'Accept' and 'Enroll' as predictors, showcasing their significance. The evaluation of the training set revealed a balance between true positives and true negatives, with an 84% accuracy. False positives and false negatives, while present, did not dominate. Applying the model to the test set maintained a similar level of accuracy, with a high sensitivity of 95%. The ROC curve emphasized the model's strong discriminative performance, supported by an AUC of 0.86. This comprehensive analysis helps in understanding the model's predictive power and areas for improvement, which are crucial for informed decision-making.

04. Citations:

➢ Splitting the data into a train and test set using split vector: source.

➢ ROC and AUC: source.

05. Appendix:

```
cat("\014") # clears console
rm(list = ls()) # clears global environment
try(dev.off(dev.list()["RStudioGD"]), silent = TRUE) # clears plots
try(p_unload(p_loaded(), character.only = TRUE), silent = TRUE) # clears packages
options(scipen = 100) # disables scientific notation for entire R session
#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Week 3<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
install.packages("ISLR")
library('ISLR')
# College Dataset
attach(College)
View(College)
names(College)
clg <- data.frame(College)
# Q1)
summary(clg)
str(clg)
# Histogram of Enrollment
hist(clg$Enroll, main = "Histogram of Enrollment", col = "orange", border = "black")
# Barplot of the distribution of Private and Public Colleges
barplot(table(clg$Private), col = c("seagreen", "orange"),
      main = "Distribution of Private and Public Colleges",
      xlab = "College Type", ylab = "Count", names.arg = c("Public", "Private"))
# Correlation matrix
cor_matrix <- cor(clg[, c("Enroll", "Accept", "Grad.Rate", "Apps")])
install.packages("corrplot")
library(corrplot)
corrplot(cor_matrix, method = "color", addCoef.col = "black", tl.col = "black",
```

```
    diag = FALSE, type = "upper", order = "hclust", tl.srt = 45)
# Q2)
install.packages("caTools")
library(caTools)
set.seed(123)
split_vec <- sample.split(clg$Private, SplitRatio = 0.7)
# Training and Test sets based on the split vector
train_data <- subset(clg, split_vec == TRUE)
test_data <- subset(clg, split_vec == FALSE)
View(train_data)
View(test_data)
head(train_data)
head(test_data)
# Q3)
install.packages("caret")
library(caret)
response_variable <- "Private"
predictors <- c("Accept", "Enroll")
# logistic regression model
mod1 <- glm(Private ~ Accept + Enroll, data = train_data, family = binomial(link = 'logit'))
# Display the summary of the logistic regression model
summary(mod1)
# Q4)
# Predictions on the training set
train_predictions <- predict(mod1, newdata = train_data, type = "response")
# Convert probabilities to binary predictions (0 or 1)
train_predictions_binary <- ifelse(train_predictions > 0.5, 1, 0)
# Confusion matrix
```

```
cm_train <- table(Actual = train_data$Private, Predicted = train_predictions_binary)

# Q5)

# Accuracy, Precision, Recall, and Specificity metrics for Train Data

accuracy_train <- round(sum(diag(cm_train)) / sum(cm_train),2)

sensitivity_train <- round(cm_train[2, 2] / sum(cm_train[2, ]),2)

specificity_train <- round(cm_train[1, 1] / sum(cm_train[1, ]),2)

precision_train <- round(cm_train[2, 2] / sum(cm_train[, 2]),2)

recall_train <- round(sensitivity_train,2)

f1_score_train <- round(2 * (precision_train * recall_train) / (precision_train + recall_train),2)

# Creating a table to store the metrics for training set

metrics_table_train <- data.frame(

  Metric = c("Accuracy", "Sensitivity", "Specificity", "Precision", "Recall", "F1 Score"),

      Value = c(accuracy_train, sensitivity_train, specificity_train, precision_train, recall_train, f1_score_train))

# Q6)

# Predictions on the test set

test_predictions <- predict(mod1, newdata = test_data, type = "response")

# Convert probabilities to binary predictions (0 or 1)

test_predictions_binary <- ifelse(test_predictions > 0.5, 1, 0)

# Create confusion matrix for the test set

cm_test <- table(Actual = test_data$Private, Predicted = test_predictions_binary)

# Accuracy, Precision, Recall, and Specificity metrics for Test Data

accuracy_test <- round(sum(diag(cm_test)) / sum(cm_test),2)

sensitivity_test <- round(cm_test[2, 2] / sum(cm_test[2, ]),2)

specificity_test <- round(cm_test[1, 1] / sum(cm_test[1, ]),2)

precision_test <- round(cm_test[2, 2] / sum(cm_test[, 2]),2)

recall_test <- round(sensitivity_test,2)

f1_score_test <- round(2 * (precision_test * recall_test) / (precision_test + recall_test),2)

# Creating a table to store the metrics for test set
```

```
metrics_table_test <- data.frame(
  Metric = c("Accuracy", "Sensitivity", "Specificity", "Precision", "Recall", "F1 Score"),
  Value = c(accuracy_test, sensitivity_test, specificity_test, precision_test, recall_test, f1_score_test))
```

 # Q7)

```
install.packages("pROC")
library(pROC)
roc_curve <- roc(test_data$Private, test_predictions)
plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2, cex.main = 1.2, col.main = "darkblue", lty = 1,
     xlab = "False Positive Rate (Specificity)", ylab = "True Positive Rate (Sensitivity)")
text(0.5, 0.5, paste("AUC = ", round(auc(roc_curve), 2)), col = "seagreen", cex = 1.2)
legend("bottomright", legend = c("ROC Curve"), col = "blue", lty = 1, lwd = 2)
```

 # Q8)

```
area_uc <- round(auc(roc_curve),2)
```

 # Area Under Curve is mentioned in ROC curve plot