1.) Given that the dataset for both the Logistic Regression and CART model will be the same, i.e., the Heart dataset. The Y (dependent variable) for both the analyses is "AHD" from the original dataset. In my research, I have renamed it to "Diagnosis_Heart_Disease." Furthermore, I changed it to a binary (0-1) factor variable, "Heart_Disease_Binary," for the remainder of my analysis.

The X (independent variables) will be all the remaining fields/variables comprising the overall dataset structure.

Here's a detailed description of all the columns ->

Age: Age of subject

Sex: Gender of subject:
0 = female 1 = male

Chest-pain type: Type of chest-pain experienced by the individual:
1 = typical angina
2 = atypical angina
3 = non-angina pain
4 = asymptomatic angina

Resting Blood Pressure: Resting blood pressure in mm Hg

Serum Cholesterol: Serum cholesterol in mg/dl

Fasting Blood Sugar: Fasting blood sugar level relative to 120 mg/dl: 0 = fasting blood sugar <= 120 mg/dl
1 = fasting blood sugar > 120 mg/dl

Resting ECG: Resting electrocardiographic results
0 = normal
1 = ST-T wave abnormality
2 = left ventricle hyperthrophy

Max Heart Rate Achieved: Max heart rate of the subject.

Exercise Induced Angina:
0 = no 1 = yes

ST Depression Induced by Exercise Relative to Rest: ST Depression of the subject.

Peak Exercise ST Segment:

1 = Up-sloping
2 = Flat
3 = Down-sloping

Number of Major Vessels (0-3) Visible on Fluoroscopy: Number of visible vessels under fluoroscopy

Thal: Form of thalassemia: 3
3 = normal
6 = fixed defect
7 = reversible defect

Diagnosis of Heart Disease: Indicates whether the subject is suffering from heart disease or not:
0 = absence
1, 2, 3, 4 = heart disease present

Let's conduct a few data preparation and preliminary EDL steps, which will serve as a basis for both the models ->

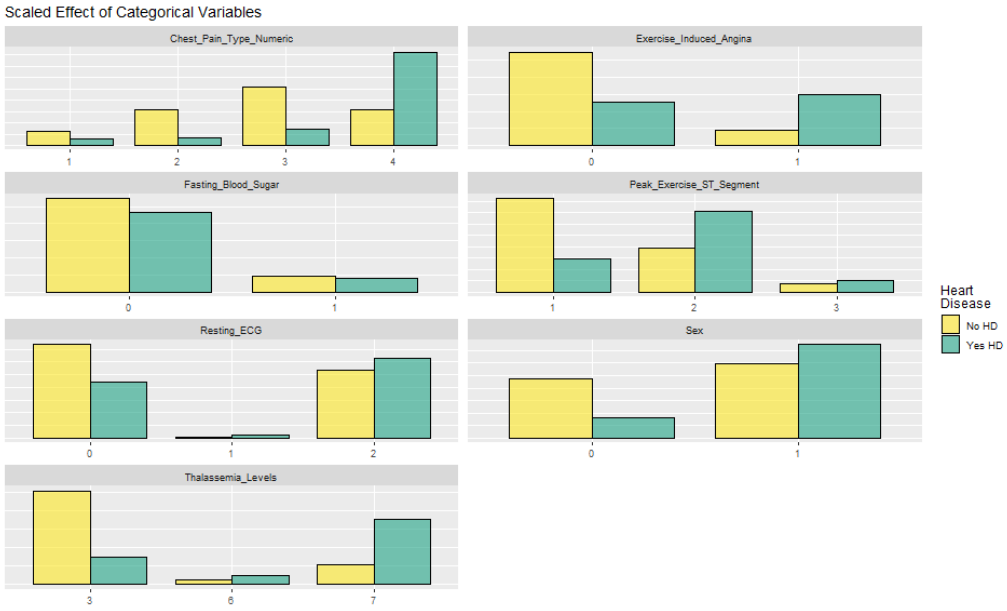
```
names(heart) # See the current fields and field names

heart_lr <- heart[,-1] # Remove the first column as it serves no purpose

names <- c("Age", # Create a vector containing the desired column names
        "Sex",
        "Chest_Pain_Type",
        "Resting_Blood_Pressure",
        "Serum_Cholesterol",
        "Fasting_Blood_Sugar",
        "Resting_ECG",
        "Max_Heart_Rate_Achieved",
        "Exercise_Induced_Angina",
        "ST_Depression_Exercise",
        "Peak_Exercise_ST_Segment",
        "Num_Major_Vessels_Flouro",
        "Thalassemia",
        "Diagnosis_Heart_Disease")

colnames(heart_lr) <- names # Change the existing column names

names(heart_lr) # Check to ensure that the changes have taken place
```

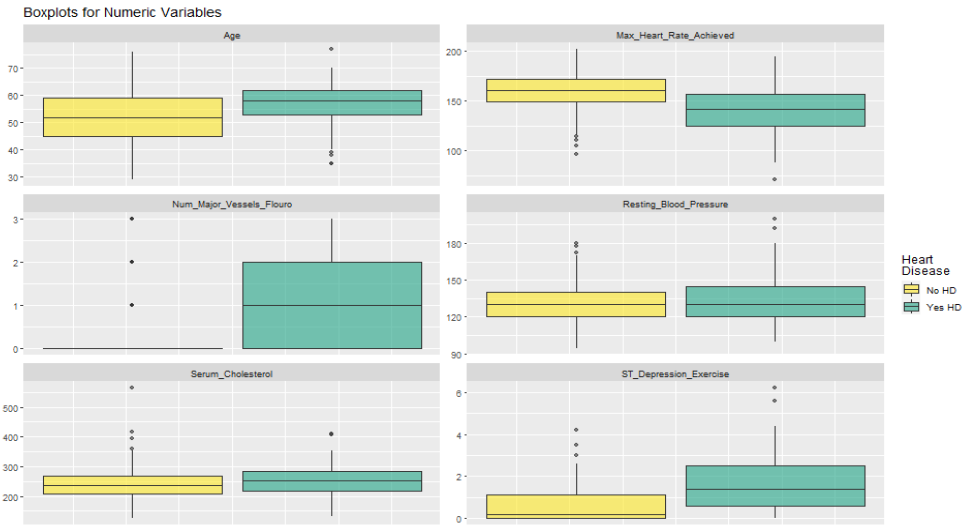
```
head(heart_lr) # Check the first six records of the dataset

sum(is.na(heart_lr)) # Check the total number of NA's present

str(heart_lr) # Check the metadata of the dataset

attach(heart_lr) # Optional step -> doing this makes accessing the columns easier

############ Convert a few required categorical variables from integers to factors
##############
heart_lr$Sex <- factor(heart_lr$Sex)
heart_lr$Fasting_Blood_Sugar <- factor(heart_lr$Fasting_Blood_Sugar)
heart_lr$Resting_ECG <- factor(heart_lr$Resting_ECG)
heart_lr$Exercise_Induced_Angina <- factor(heart_lr$Exercise_Induced_Angina)
heart_lr$Peak_Exercise_ST_Segment <- factor(heart_lr$Peak_Exercise_ST_Segment)

glimpse(heart_lr) # a dplyr equivalent of str()

summary(heart_lr) # Get a basic data profile

# table(Num_Major_Vessels_Flouro)

# dplyr provides a lot of commands for data wrangling. One of the most powerful of them
# being mutate() which can be used to create new fields or edit the existing fields similar
# to the CASE statement on SQL.

heart_lr <- heart_lr %>% mutate(Chest_Pain_Type_Numeric = case_when(Chest_Pain_Type ==
"typical" ~ 1,
Chest_Pain_Type == "nontypical" ~ 2,
Chest_Pain_Type == "nonanginal" ~ 3,
Chest_Pain_Type == "asymptomatic" ~ 4
))
heart_lr$Chest_Pain_Type_Numeric <- factor(heart_lr$Chest_Pain_Type_Numeric)

heart_lr <- heart_lr %>% mutate(Resting_ECG_Results = case_when(Resting_ECG == 0 ~
"Normal",
Resting_ECG == 1 ~ "ST-T Wave Abnormality",
Resting_ECG == 2 ~ "Left Ventricle Hyperthropy"
))

heart_lr <- heart_lr %>% mutate(Heart_Disease_Binary = case_when(Diagnosis_Heart_Disease
== "No" ~ 0,
Diagnosis_Heart_Disease == "Yes" ~ 1
))
```

```
heart_lr$Heart_Disease_Binary <- factor(heart_lr$Heart_Disease_Binary)

heart_lr <- heart_lr %>% mutate(Thalassemia_Levels = case_when(Thalassemia == "normal" ~ 3,
Thalassemia == "fixed" ~ 6,
Thalassemia == "reversable" ~ 7
))
heart_lr$Thalassemia_Levels <- factor(heart_lr$Thalassemia_Levels)

glimpse(heart_lr)

which(is.na(heart_lr),arr.ind = TRUE) # Figure out which rows have blanks or NA's present in
them

heart_clean <- heart_lr %>% drop_na() %>% filter(Thalassemia != "?") # dplyr provides the
piping
# capability that can be used to chain multiple commands together

# View(heart_clean)
glimpse(heart_clean)
# heart_clean <- heart_clean[,-c(3,13,14,16)]
heart_clean <- heart_clean[,-c(3,13,16)] # Subset out a few columns which we don't need
glimpse(heart_clean)


heart_tbl <- heart_clean %>% select(Sex,Fasting_Blood_Sugar,Resting_ECG,
Exercise_Induced_Angina,Peak_Exercise_ST_Segment,
Thalassemia_Levels,Heart_Disease_Binary,Chest_Pain_Type_Numeric) %>%
gather(key = "key", value = "value", -Heart_Disease_Binary)

heart_tbl %>% ggplot(aes(value)) +
geom_bar(aes(x       = value,
fill     = Heart_Disease_Binary),
alpha    = .6,
position = "dodge",
color    = "black",
width    = .8
) +
labs(x = "",  # Create basic barplots to understand the prevalence and causes of heart attacks
y = "",
title = "Scaled Effect of Categorical Variables") +
theme(
axis.text.y  = element_blank(),
axis.ticks.y = element_blank()) +
facet_wrap(~ key, scales = "free", nrow = 4) +
```

```
scale_fill_manual(
values = c("#fde725ff", "#20a486ff"),
name   = "Heart\nDisease",
labels = c("No HD", "Yes HD"))

heart_tbl_cont <- heart_clean %>% select(Age, Resting_Blood_Pressure,
Serum_Cholesterol,
Max_Heart_Rate_Achieved,
ST_Depression_Exercise,
Num_Major_Vessels_Flouro,
Heart_Disease_Binary) %>%
gather(key = "key", value = "value", -Heart_Disease_Binary)

heart_tbl_cont %>% ggplot(aes(y = value)) +
geom_boxplot(aes(fill = Heart_Disease_Binary),
alpha  = .6,
fatten = .7) +
labs(x = "",
y = "",
title = "Boxplots for Numeric Variables") + # Create boxplots to see the outliers in the
continuous fields
scale_fill_manual(
values = c("#fde725ff", "#20a486ff"),
name   = "Heart\nDisease",
labels = c("No HD", "Yes HD")) +
theme(
axis.text.x  = element_blank(),
axis.ticks.x = element_blank()) +
facet_wrap(~ key,
scales = "free",
ncol   = 2)

heart_tbl_corr <- heart_clean %>% ggcorr(high = "#20a486fe",
low = "#fde725ee",
label = TRUE,)
heart_tbl_corr <- heart_clean %>% ggcorr(high = "#20a486fe",
low = "#fde725ee",
label = TRUE,
hjust = 0.80,
size = 3,
label_size = 3,
nbreaks = 5) +
labs(title = "Correlation Matrix",
subtitle = "Pearson Method for pair-wise correlations")
```

heart_tbl_corr # The above lines plot the one -to-one correlation between the fields





Basis the above plots, a few conclusions can be drawn ->

a.) Presence of exercise-induced angina seems to be moderately associated with increased incidence of heart attacks.
b.) Males tend to have a higher incidence of heart attacks.
c.) Higher Thalassemia score may indicate a higher chance of heart attack.
d.) Higher cholesterol levels correspond to higher levels of heart attacks.

It should be noted that the above observations do not correspond to causality.

2.) **Logistic Regression** -> The original document undergoes a 75/25 train/test split for the logistic regression model. Concurrently, a few factor character variables have either been re-coded or dropped.

The regression has been performed once, each using the training and full datasets, respectively. The outputs of the same can be found below:

```
logit_train <- glm(Heart_Disease_Binary ~ .,data = heart_train, family = "binomial")
summary(logit_train)
```

```
logit_full <- glm(Heart_Disease_Binary ~ .,data = heart_clean, family = "binomial")
summary(logit_full)
```

```
logLik(logit_train)
```

logLik(logit_full)

with(logit_full, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))

with(logit_train, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))

```
> logit_train <- glm(Heart_Disease_Binary ~ .,data = heart_train, family = "binomial")
> summary(logit_train)

Call:
glm(formula = Heart_Disease_Binary ~ ., family = "binomial",
    data = heart_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8294  -0.4071  -0.1050   0.4068   2.9127

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -9.466247   3.845686  -2.462  0.01383 *
Age                       0.005733   0.030072   0.191  0.84880
Sex1                      1.909087   0.683659   2.792  0.00523 **
Resting_Blood_Pressure    0.022173   0.013201   1.680  0.09304 .
Serum_Cholesterol         0.006138   0.005789   1.060  0.28894
Fasting_Blood_Sugar1     -0.987916   0.692281  -1.427  0.15357
Resting_ECG1              1.259679   2.543250   0.495  0.62039
Resting_ECG2              0.712501   0.459165   1.552  0.12073
Max_Heart_Rate_Achieved  -0.004759   0.014372  -0.331  0.74055
Exercise_Induced_Angina1  1.281022   0.528573   2.424  0.01537 *
ST_Depression_Exercise    0.142495   0.256309   0.556  0.57825
Peak_Exercise_ST_Segment2 1.709255   0.587061   2.912  0.00360 **
Peak_Exercise_ST_Segment3 0.788383   1.012399   0.779  0.43614
Num_Major_Vessels_Flouro  1.376739   0.344565   3.996 6.45e-05 ***
Chest_Pain_Type_Numeric2  0.858626   0.927278   0.926  0.35446
Chest_Pain_Type_Numeric3 -0.102645   0.790239  -0.130  0.89665
Chest_Pain_Type_Numeric4  1.317692   0.794099   1.659  0.09704 .
Thalassemia_Levels6       0.102800   0.901384   0.114  0.90920
Thalassemia_Levels7       1.587204   0.524552   3.026  0.00248 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 306.60  on 221  degrees of freedom
Residual deviance: 135.36  on 203  degrees of freedom
AIC: 173.36

Number of Fisher Scoring iterations: 6
```

```
> logit_full <- glm(Heart_Disease_Binary ~ .,data = heart_clean, family = "binomial")
> summary(logit_full)

Call:
glm(formula = Heart_Disease_Binary ~ ., family = "binomial",
    data = heart_clean)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7614  -0.5211  -0.1450   0.3181   2.7881

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -6.029468   2.891768  -2.085  0.03707 *
Age                      -0.013763   0.024745  -0.556  0.57808
Sex1                      1.546014   0.529995   2.917  0.00353 **
Resting_Blood_Pressure    0.024364   0.011269   2.162  0.03062 *
Serum_Cholesterol         0.004448   0.003993   1.114  0.26526
Fasting_Blood_Sugar1     -0.596246   0.607848  -0.981  0.32664
Resting_ECG1              0.810202   2.435102   0.333  0.73935
Resting_ECG2              0.473895   0.383518   1.236  0.21659
Max_Heart_Rate_Achieved  -0.017723   0.011109  -1.595  0.11065
Exercise_Induced_Angina1  0.709456   0.440018   1.612  0.10689
ST_Depression_Exercise    0.357875   0.230070   1.556  0.11983
Peak_Exercise_ST_Segment2 1.155286   0.473794   2.438  0.01475 *
Peak_Exercise_ST_Segment3 0.525147   0.919661   0.571  0.56798
Num_Major_Vessels_Flouro  1.311510   0.279276   4.696 2.65e-06 ***
Chest_Pain_Type_Numeric2  1.239566   0.770874   1.608  0.10784
Chest_Pain_Type_Numeric3  0.245959   0.663312   0.371  0.71078
Chest_Pain_Type_Numeric4  2.086480   0.666547   3.130  0.00175 **
Thalassemia_Levels6      -0.010974   0.790210  -0.014  0.98892
Thalassemia_Levels7       1.392715   0.425194   3.275  0.00105 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 409.95  on 296  degrees of freedom
Residual deviance: 191.64  on 278  degrees of freedom
AIC: 229.64

Number of Fisher Scoring iterations: 6
```

The dataset “heart_clean” is the entire dataset post the initial data cleaning phase. The log-likelihood and the p-values of both the models are given below:

```
> logLik(logit_train)
'log Lik.' -67.6824 (df=19)
>
> logLik(logit_full)
'log Lik.' -95.82101 (df=19)
>
> with(logit_full, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 2.12447e-36
>
> with(logit_train, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 5.165902e-27
```

The regression was re-run to basis the significant variables observed to yield a more parsimonious model as shown below:

logit_train_parsimonious <- glm(Heart_Disease_Binary ~ Resting_Blood_Pressure +
Exercise_Induced_Angina +
Thalassemia_Levels +

Num_Major_Vessels_Flouro,data = heart_train, family = "binomial")

summary(logit_train_parsimonious)

logLik(logit_train_parsimonious)

logit_full_parsimonious <- glm(Heart_Disease_Binary ~ Sex + Resting_Blood_Pressure + Chest_Pain_Type_Numeric +
Thalassemia_Levels +
Num_Major_Vessels_Flouro,data = heart_clean, family = "binomial")

summary(logit_full_parsimonious)

logLik(logit_full)

with(logit_full_parsimonious, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))

with(logit_train_parsimonious, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))

```
> logit_train_parsimonious <- glm(Heart_Disease_Binary ~ Resting_Blood_Pressure +
+ Exercise_Induced_Angina +
+ Thalassemia_Levels +
+ Num_Major_Vessels_Flouro,data = heart_train, family = "binomial")
>
> summary(logit_train_parsimonious)

Call:
glm(formula = Heart_Disease_Binary ~ Resting_Blood_Pressure +
    Exercise_Induced_Angina + Thalassemia_Levels + Num_Major_Vessels_Flouro,
    family = "binomial", data = heart_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7224  -0.4252  -0.3276   0.5396   2.4402

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)            -4.3829     1.3874  -3.159  0.00158 **
Resting_Blood_Pressure  0.0135     0.0100   1.350  0.17713
Exercise_Induced_Angina1  1.9877     0.4234   4.695 2.67e-06 ***
Thalassemia_Levels6     1.3903     0.7232   1.922  0.05455 .
Thalassemia_Levels7     2.3580     0.4171   5.654 1.57e-08 ***
Num_Major_Vessels_Flouro  1.3170     0.2566   5.133 2.86e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 306.60  on 221  degrees of freedom
Residual deviance: 170.61  on 216  degrees of freedom
AIC: 182.61

Number of Fisher Scoring iterations: 5
```

```
> logit_full_parsimonious <- glm(Heart_Disease_Binary ~ Sex + Resting_Blood_Pressure +
+ Chest_Pain_Type_Numeric +
+ Thalassemia_Levels +
+ Num_Major_Vessels_Flouro,data = heart_clean, family = "binomial")
>
> summary(logit_full_parsimonious)

Call:
glm(formula = Heart_Disease_Binary ~ Sex + Resting_Blood_Pressure +
    Chest_Pain_Type_Numeric + Thalassemia_Levels + Num_Major_Vessels_Flouro,
    family = "binomial", data = heart_clean)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0147  -0.5173  -0.2223   0.5233   2.4635

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            -6.052196   1.575921  -3.840 0.000123 ***
Sex1                    0.951544   0.415518   2.290 0.022020 *
Resting_Blood_Pressure  0.020678   0.009541   2.167 0.030216 *
Chest_Pain_Type_Numeric2  0.174435   0.678391   0.257 0.797078
Chest_Pain_Type_Numeric3 -0.117730   0.612701  -0.192 0.847625
Chest_Pain_Type_Numeric4  2.040611   0.578981   3.524 0.000424 ***
Thalassemia_Levels6     0.892642   0.687293   1.299 0.194019
Thalassemia_Levels7     1.835924   0.376814   4.872 1.10e-06 ***
Num_Major_Vessels_Flouro  1.174845   0.219314   5.357 8.47e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 409.95  on 296  degrees of freedom
Residual deviance: 228.27  on 288  degrees of freedom
AIC: 246.27

Number of Fisher Scoring iterations: 5
```

Similarly, the log-likelihood and the p-values of the above two models are also shown below:

```
> logLik(logit_train_parsimonious)
'log Lik.' -85.30377 (df=6)
```

```
> logLik(logit_full)
'log Lik.' -95.82101 (df=19)
>
> with(logit_full_parsimonious, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 4.582788e-35
>
> with(logit_train_parsimonious, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 1.269312e-27
```

We’ll also do regression by setting the dependent variable against no independent variable, i.e., regressing it against the intercept. The results of the above are given below:

```
> logit_none <- glm(Heart_Disease_Binary ~ 1, data = heart_clean, family = "binomial")
> summary(logit_none)

Call:
glm(formula = Heart_Disease_Binary ~ 1, family = "binomial",
    data = heart_clean)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.112  -1.112  -1.112   1.244   1.244

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1552     0.1164  -1.333    0.182

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 409.95  on 296  degrees of freedom
Residual deviance: 409.95  on 296  degrees of freedom
AIC: 411.95

Number of Fisher Scoring iterations: 3

> logLik(logit_none)
'log Lik.' -204.9732 (df=1)
> with(logit_none, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 1
```

Model Interpretations: The benefit of doing the regression each on the training and the entire dataset alongside their parsimonious models respectively allows us to see which independent variables are significant in predicting the chance of inducing a heart attack. Let us go through both the parsimonious models and interpret what they mean ->

a.) Train -> Mathematically,

Heart_Disease_Binary = -4.829 + 0.0135 * (Resting_Blood_Pressure) + 1.9877 * (Exercise Induced Angina – Lv1) + 1.3903 * (Thalassemia – Level 6) + 2.358 * (Thalassemia – Level 7) + 1.317 * (Num_Major_Vessels_Fluoro)

The above equation yields the following interpretations ->

i.) For a unit increase in Resting BP, the odds of having a heart attack increase by a factor of 0.0135.

ii.) Having an Exercise-Induced Angina – Lv1 is associated with a 1.9877 odds increase for a heart attack.

iii.) Having a Fixed Defect Thalassemia, i.e., Thalassemia – Lv6, is associated with a 1.3903 odds increase for a heart attack.

iv.) Having a Reversible Defect Thalassemia, i.e., Thalassemia – Lv7, is associated with a 2.358 odds increase for a heart attack.

v.) For a unit increase in the Number of Major vessels visible under Fluoroscopy, the odds of having a heart attack increase by a factor of 1.317.

b.) Full data -> Mathematically,

Heart_Disease_Binary = -6.052 + 0.951 * (Sex-Male) + 0.02 * (Resting_Blood_Pressure) + 0.174 * (Chest_Pain_Type_Numeric – Level 2) + (-0.11) * (Chest_Pain_Type_Numeric – Level 3) + 2.04 * (Chest_Pain_Type_Numeric – Level 4) + 0.892 * (Thalassemia – Level 6) + 1.835 * (Thalassemia – Level 7) + 1.174 * (Num_Major_Vessels_Fluoro)

The above equation yields the following interpretations ->

i.) For males, the odds of having a heart attack increase by a factor of 0.951.

ii.) For a unit increase in Resting BP, the odds of incurring a heart attack increase by a factor of 0.02

iii.) Having an Atypical Angina, i.e., Chest_Pain_Type_Numeric – Level 2, is associated with a 0.174 odds increase of a heart attack.

iv.) Having Non-Angina pain, i.e., Chest_Pain_Type_Numeric – Level 3 is associated with a 0.11 odds decrease of a heart attack.

v.) Having an Asymptomatic Angina, i.e., Chest_Pain_Type_Numeric – Level 4 is associated with a 2.04 odds increase of a heart attack.

vi.) Having a Fixed Defect Thalassemia, i.e., Thalassemia – Lv6, is associated with a 0.892 odds increase for a heart attack.

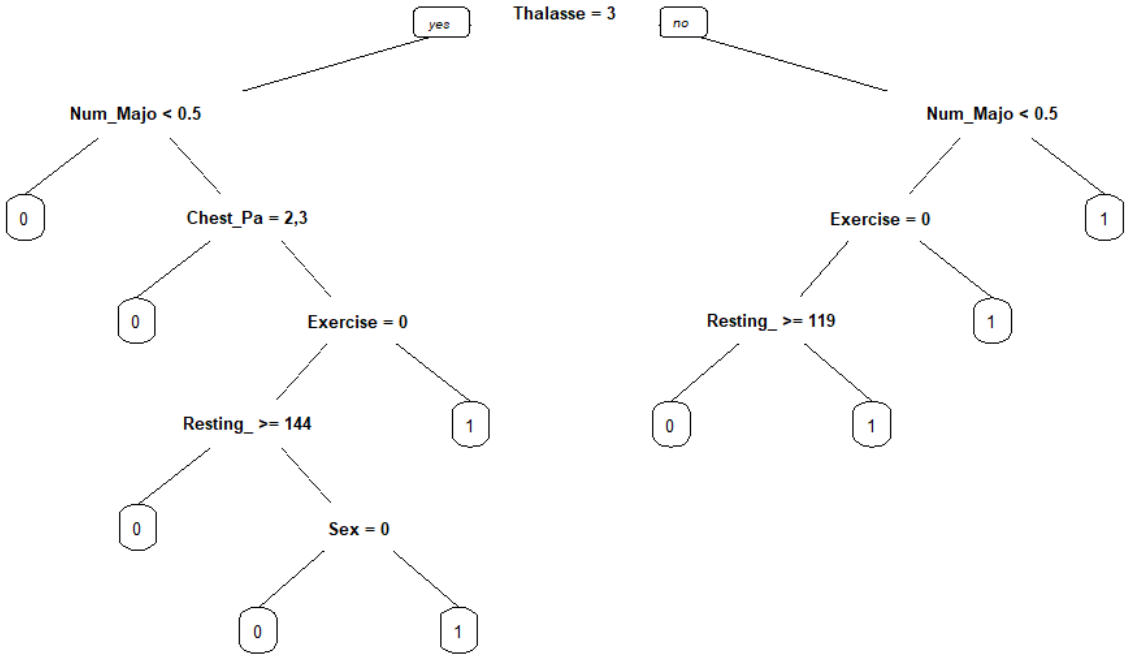
vii.) Having a Reversible Defect Thalassemia, i.e., Thalassemia – Lv7, is associated with a 1.835 odds increase for a heart attack.

viii.) For a unit increase in the Number of Major vessels visible under Fluoroscopy, the heart attack odds increase by a factor of 1.174.

Although the train and the entire dataset models have overlapping variables present, we'll consider the full dataset parsimonious model as the model of choice owing to its lower p-value, log-likelihood, and confidence interval. Even though the full dataset model has a higher AIC value, thus diminishing its out-of-sample predictability, it is still the right due to a lower record to fields ratio.

CART Model -> The basic data preparations steps will be the same as that executed in Logistic Regression. Here, we will be using the training dataset to carry out our analysis.

Let us have a look below at the initial tree which we obtain.



The above tree has been obtained using a 10-fold Cross-validation indicated by the "xval" argument in the rpart. Control command.

Let us now look at the Complexity Parameter and identify the depth by which we can prune to obtain the optimal tree. The C_p values can be found below ->

```
> class_param <- heart_cart$cptable
> class_param
          CP nsplit rel error    xerror       xstd
1 0.52427184      0 1.0000000 1.0000000 0.07214037
2 0.05339806      1 0.4757282 0.4757282 0.05999394
3 0.03883495      3 0.3689320 0.4757282 0.05999394
4 0.01941748      5 0.2912621 0.4271845 0.05766637
5 0.01294498      6 0.2718447 0.4757282 0.05999394
6 0.01000000      9 0.2330097 0.4757282 0.05999394
```

From the above table, we see that the 5th Complexity parameter with nine splits and at a depth of six results in the minimum relative error, and hence we will use this to prune the tree.

We will run a for-loop that takes the index equal to our depth and calculates the training, testing, and cross-validation error,

train_error <- double(6)

test_error <- double(6)

cv_error <- double(6)

```
for(i in 1:nrow(class_param)){
alpha <- class_param[i, 'CP']
train <- table(heart_train$Heart_Disease_Binary, predict(prune(heart_cart, cp = alpha),
heart_train, type = "class"))
train_error[i] <- 1 - sum(diag(train))/sum(train)
cv_error[i] <- class_param[i, "xerror"] * class_param[i, 'rel error']
test <- table(heart_test$Heart_Disease_Binary, predict(prune(heart_cart, cp = alpha),
heart_test, type = "class"))
test_error[i] <- 1 - sum(diag(test))/sum(test)
}
```

```
train_error
```

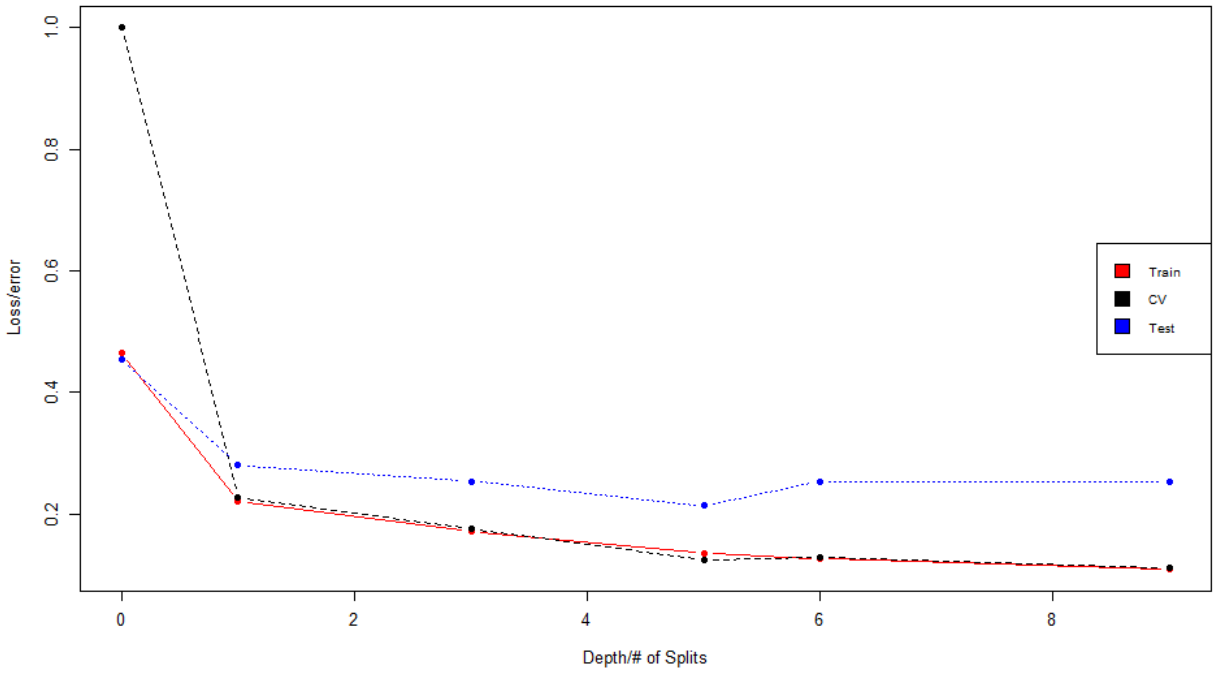
```
test_error
```

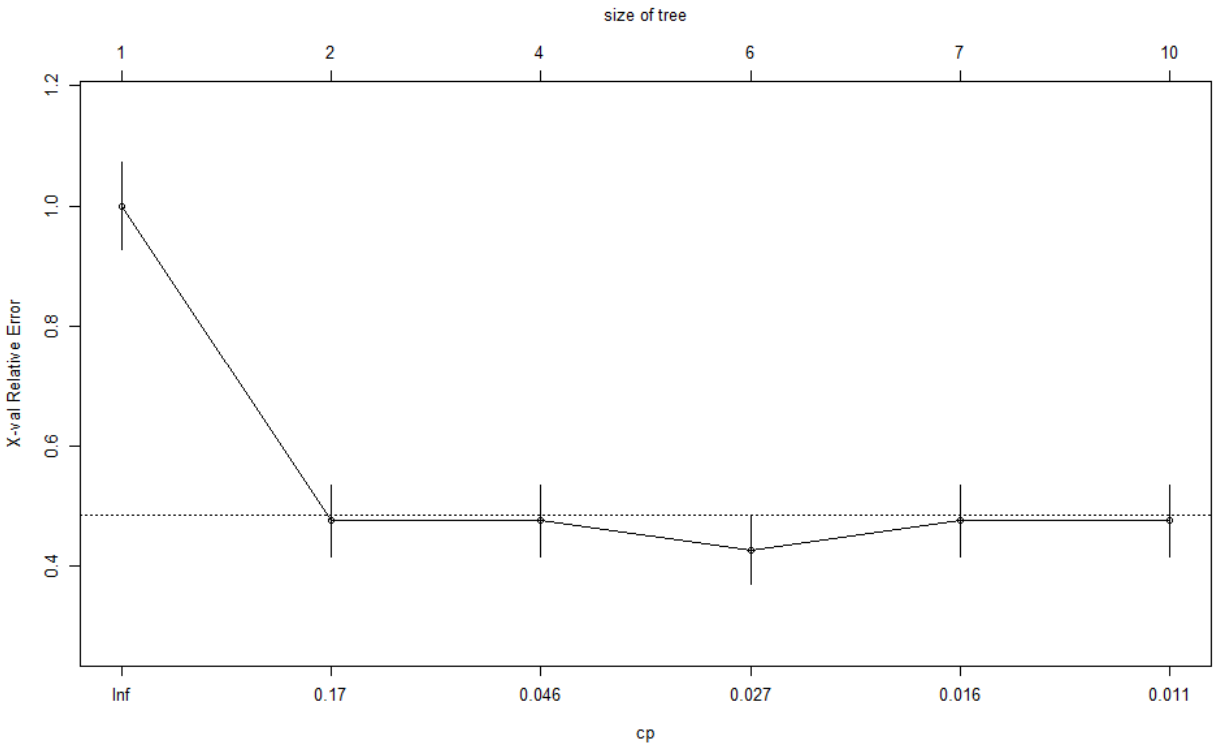
```
cv_error
```

```
> train_error <- double(6)
> test_error <- double(6)
> cv_error <- double(6)
>
> for(i in 1:nrow(class_param)){
+ alpha <- class_param[i, 'CP']
+ train <- table(heart_train$Heart_Disease_Binary, predict(prune(heart_cart, cp = alpha),
+ heart_train, type = "class"))
+ train_error[i] <- 1 - sum(diag(train))/sum(train)
+ cv_error[i] <- class_param[i, "xerror"] * class_param[i, 'rel error']
+ test <- table(heart_test$Heart_Disease_Binary, predict(prune(heart_cart, cp = alpha),
+ heart_test, type = "class"))
+ test_error[i] <- 1 - sum(diag(test))/sum(test)
+ }
>
> train_error
[1] 0.4639640 0.2207207 0.1711712 0.1351351 0.1261261 0.1081081
>
> test_error
[1] 0.4533333 0.2800000 0.2533333 0.2133333 0.2533333 0.2533333
>
> cv_error
[1] 1.0000000 0.2263173 0.1755114 0.1244227 0.1293242 0.1108493
```

Let us now plot the above-derived errors.

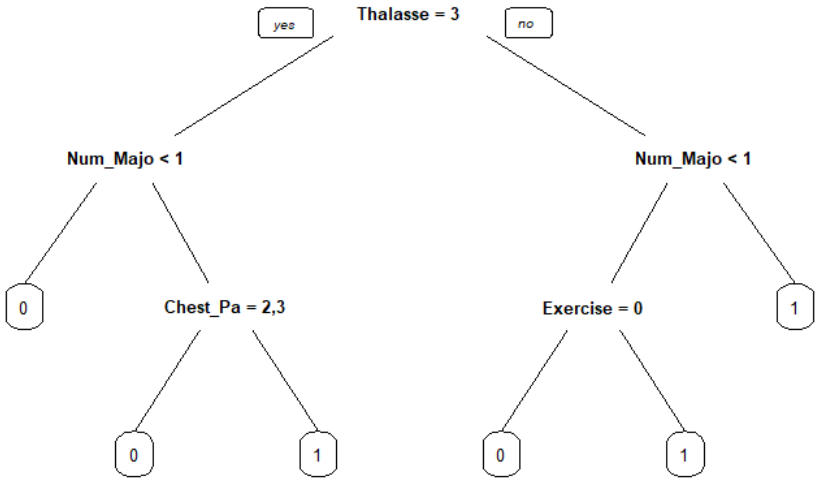
```
matplot(class_param[,'nsplit'], cbind(train_error, cv_error, test_error),
        pch=19, col=c("red", "black", "blue"), type="b",
        ylab="Loss/error", xlab="Depth/# of Splits")

legend("right", c('Train', 'CV', 'Test') ,col=seq_len(3),cex=0.8,fill=c("red", "black", "blue"))

plotcp(heart_cart)
```





So, from the above two plots, we can see that basis the Complexity Parameters obtained, the number of splits should be > 2 as anything less will not be decipherable. However, anything excessive will render the pruning ineffective. Hence, we will choose nsplit = 4 to prune the tree.

```
heart_prune <- prune(heart_cart,cp = class_param[4, 'CP'])

prp(heart_prune)
```



Let us now examine the confusion matrix and the accuracy of the pruned and unpruned trees.

```
heart_cfm_cart <- table(heart_test$Heart_Disease_Binary,predict(heart_prune,
type = "class", newdata = heart_test))
heart_cfm_cart

cart_accuracy <- sum(diag(heart_cfm_cart))/sum(heart_cfm_cart)
cart_accuracy

cart_train_accuracy <- 1-train_error[6]
cart_train_accuracy

cart_test_accuracy <- 1-test_error[6]
cart_test_accuracy
```

cart_cv_accuracy <- 1-cv_error[6]

cart_cv_accuracy

prune_train_acc <- 1-train_error[4]

prune_train_acc

prune_test_acc <- 1-test_error[4]

prune_test_acc

prune_cv_acc <- 1-cv_error[4]

prune_cv_acc

```
> heart_cfm_cart <- table(heart_test$Heart_Disease_Binary,predict(heart_prune,
+ type = "class", newdata = heart_test))
> heart_cfm_cart

    0  1
  0 35  6
  1 10 24
>
> cart_accuracy <- sum(diag(heart_cfm_cart))/sum(heart_cfm_cart)
> cart_accuracy
[1] 0.7866667
>
> cart_train_accuracy <- 1-train_error[6]
> cart_train_accuracy
[1] 0.8918919
>
> cart_test_accuracy <- 1-test_error[6]
> cart_test_accuracy
[1] 0.7466667
>
> cart_cv_accuracy <- 1-cv_error[6]
> cart_cv_accuracy
[1] 0.8891507
>
> prune_train_acc <- 1-train_error[4]
> prune_train_acc
[1] 0.8648649
>
> prune_test_acc <- 1-test_error[4]
> prune_test_acc
[1] 0.7866667
>
> prune_cv_acc <- 1-cv_error[4]
> prune_cv_acc
[1] 0.8755773
```

3.) **Summary Table: Logistic Regression** ->

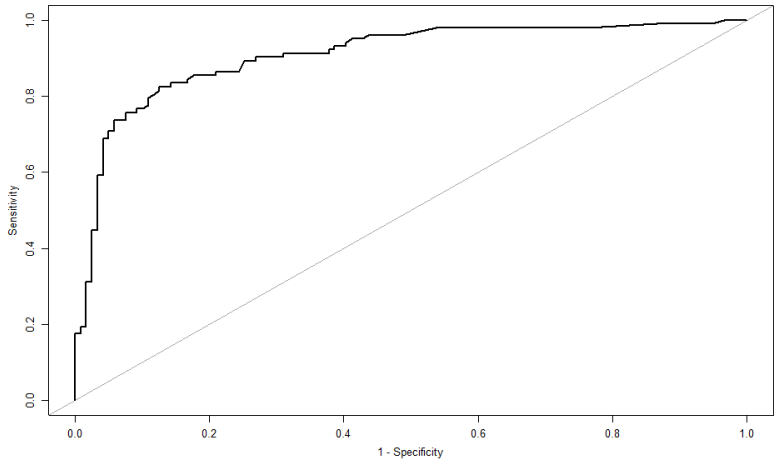
Metric	Testing	Training	Full Dataset
Accuracy	0.773	0.846	0.841
Sensitivity	0.780	0.865	0.868
Specificity	0.764	0.825	0.810
PPV	0.8	0.851	0.842
NPV	0.742	0.841	0.840
AUC	0.843	0.907	0.908

Summary Table: CART Model ->

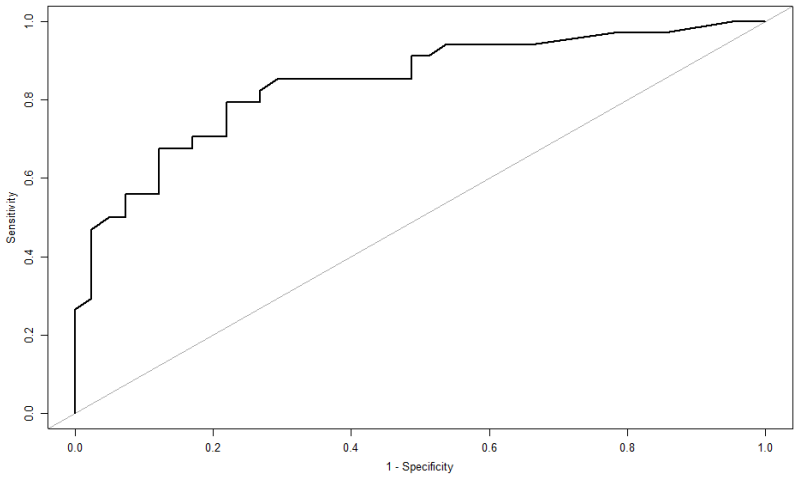
		Unpruned Tree	Pruned Tree
Tree Size		9	4
Accuracy	Train	0.891	0.864
	Test	0.746	0.786
	All (Cross-validated)	0.889	0.875

4.) **ROC Curves** ->

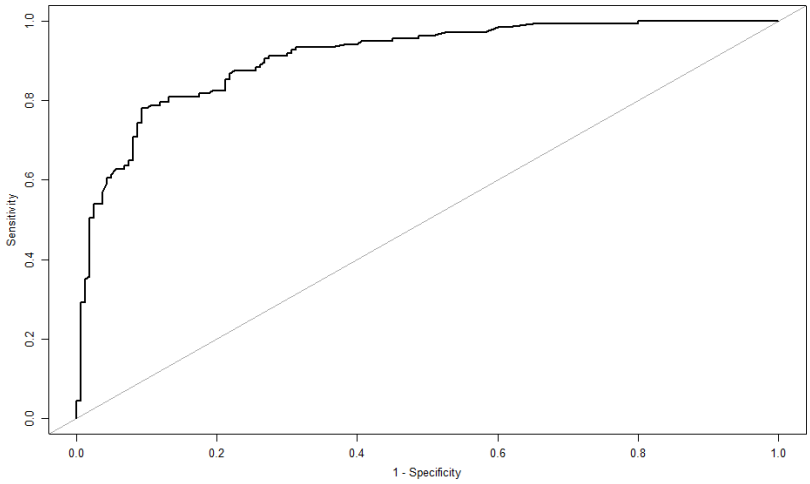
a.) **Train** ->



b.) **Test** ->

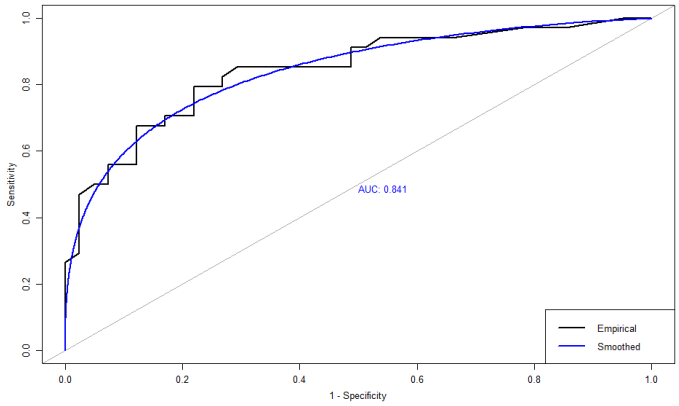

Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0

c.) **Full** ->

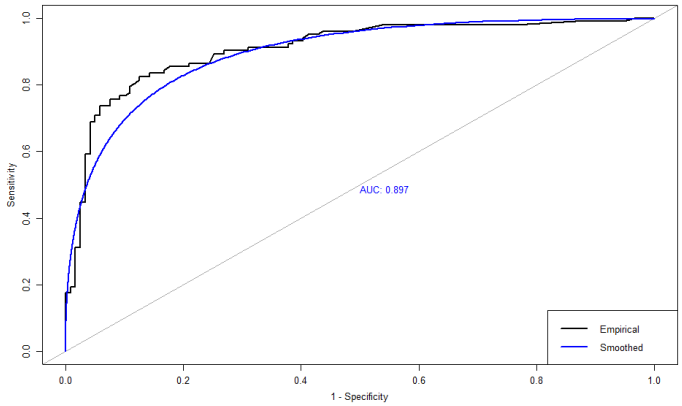

Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0

AUC Curves ->

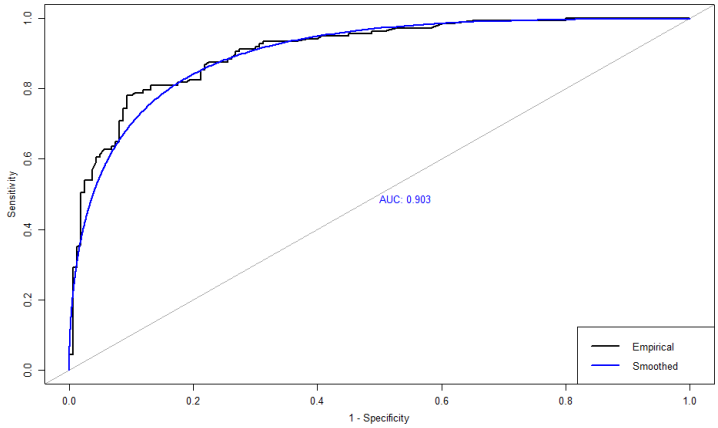
a.) **Test** ->



b.) **Train** ->

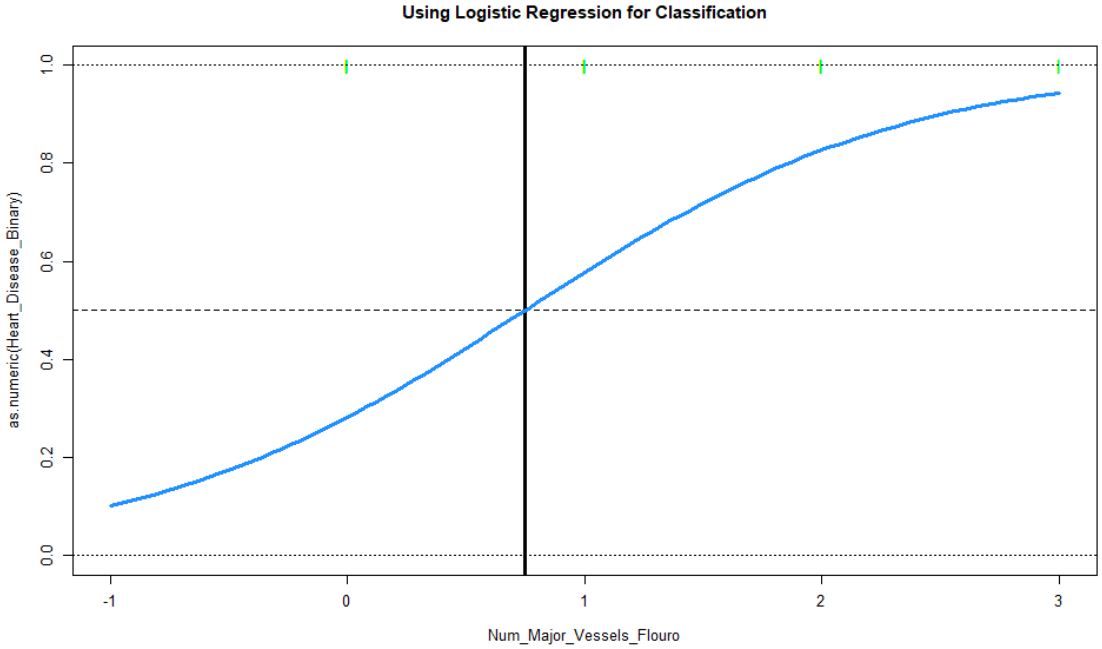


c.) **Full** ->



5.) **S-Curve** -> The S-Curve can be generated using the following code:

```
heart_scurve <- glm(Heart_Disease_Binary ~ Num_Major_Vessels_Flouro, data = heart_clean,
family = "binomial")
plot(as.numeric(Heart_Disease_Binary) ~ Num_Major_Vessels_Flouro, data = heart_clean,
    col = "green", pch = "|", xlim = c(-1,3), ylim = c(0,1),
    main = "Using Logistic Regression for Classification") +
abline(h = 0, lty = 3) +
abline(h = 1, lty = 3) +
abline(h = 0.5, lty = 2) +
curve(predict(heart_scurve, data.frame(Num_Major_Vessels_Flouro = x), type = "response"),
      add = TRUE, lwd = 3, col = "dodgerblue") +
abline(v = -coef(heart_scurve)[1] / coef(heart_scurve)[2], lwd = 3)
```



6.) **References** ->

a.) R-Bloggers -> Primarily used for the EDA.
b.) UCLA
c.) Cross-Validated -> 1, 2, 3, 4
d.) CRAN