ML models analysis on Heart Disease Dataset

Krishna Sai Pendem

```
setwd("/Users/krishna/Desktop/SEM 1/STAT/Final Project/Datasets/")
heart_df <- read.csv("heart.csv")
head(heart_df)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  52   1  0      125  212   0       1     168     0     1.0     2  2    3
## 2  53   1  0      140  203   1       0     155     1     3.1     0  0    3
## 3  70   1  0      145  174   0       1     125     1     2.6     0  0    3
## 4  61   1  0      148  203   0       1     161     0     0.0     2  1    3
## 5  62   0  0      138  294   1       1     106     0     1.9     1  3    2
## 6  58   0  0      100  248   0       0     122     0     1.0     1  0    2
##   target
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
```

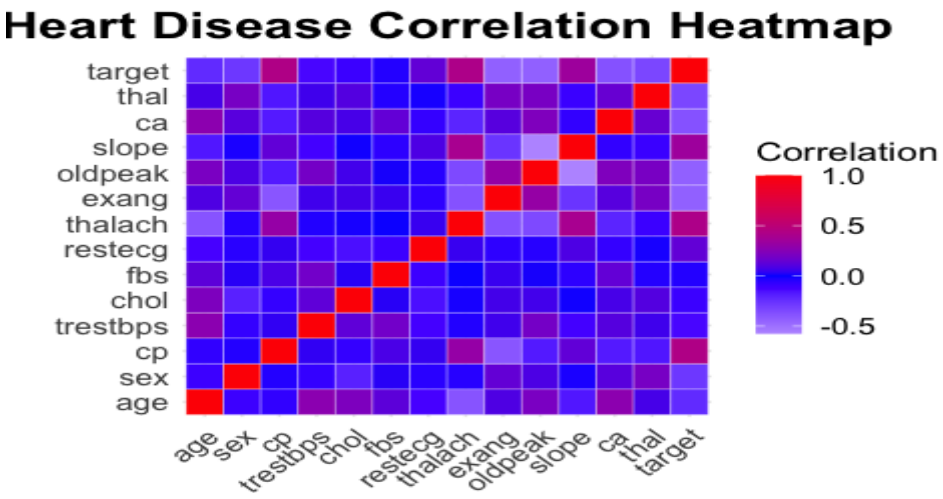
```
summary(heart_df)
```

```
##       age             sex               cp            trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.0000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Median :1.0000   Median :130.0
##  Mean   :54.43   Mean   :0.6956   Mean   :0.9424   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.0000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.0000   Max.   :200.0
##       chol           fbs             restecg          thalach
##  Min.   :126   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:132.0
##  Median :240   Median :0.0000   Median :1.0000   Median :152.0
##  Mean   :246   Mean   :0.1493   Mean   :0.5298   Mean   :149.1
##  3rd Qu.:275   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##      exang           oldpeak          slope             ca
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.800   Median :1.000   Median :0.0000
##  Mean   :0.3366   Mean   :1.072   Mean   :1.385   Mean   :0.7541
##  3rd Qu.:1.0000   3rd Qu.:1.800   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.200   Max.   :2.000   Max.   :4.0000
##       thal           target
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:2.000   1st Qu.:0.0000
```

```
##  Median :2.000   Median :1.0000
##  Mean   :2.324   Mean   :0.5132
##  3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000

dim(heart_df)

## [1] 1025   14

cat("There are ", nrow(heart_df), "number of rows \n")

## There are  1025 number of rows

cat("There are ", ncol(heart_df), "number of columns")

## There are  14 number of columns

heart_df_corr <- cor(heart_df)
heart_df_corr

##                  age         sex          cp    trestbps        chol
## age       1.00000000 -0.10324030 -0.07196627  0.27112141  0.21982253
## sex      -0.10324030  1.00000000 -0.04111909 -0.07897377 -0.19825787
## cp       -0.07196627 -0.04111909  1.00000000  0.03817742 -0.08164102
## trestbps  0.27112141 -0.07897377  0.03817742  1.00000000  0.12797743
## chol      0.21982253 -0.19825787 -0.08164102  0.12797743  1.00000000
## fbs       0.12124348  0.02720046  0.07929359  0.18176662  0.02691716
## restecg  -0.13269617 -0.05511721  0.04358061 -0.12379409 -0.14741024
## thalach  -0.39022708 -0.04936524  0.30683928 -0.03926407 -0.02177209
## exang     0.08816338  0.13915681 -0.40151271  0.06119697  0.06738223
## oldpeak   0.20813668  0.08468656 -0.17473348  0.18743411  0.06488031
## slope    -0.16910511 -0.02666629  0.13163278 -0.12044531 -0.01424787
## ca        0.27155053  0.11172891 -0.17620647  0.10455372  0.07425934
## thal      0.07229745  0.19842425 -0.16334148  0.05927618  0.10024418
## target   -0.22932355 -0.27950076  0.43485425 -0.13877173 -0.09996559
##                  fbs     restecg      thalach       exang     oldpeak
## age       0.121243479 -0.13269617 -0.390227075  0.08816338  0.20813668
## sex       0.027200461 -0.05511721 -0.049365243  0.13915681  0.08468656
## cp        0.079293586  0.04358061  0.306839282 -0.40151271 -0.17473348
## trestbps  0.181766624 -0.12379409 -0.039264069  0.06119697  0.18743411
## chol      0.026917164 -0.14741024 -0.021772091  0.06738223  0.06488031
## fbs       1.000000000 -0.10405124 -0.008865857  0.04926057  0.01085948
## restecg  -0.104051244  1.00000000  0.048410637 -0.06560553 -0.05011425
## thalach  -0.008865857  0.04841064  1.000000000 -0.38028087 -0.34979616
## exang     0.049260570 -0.06560553 -0.380280872  1.00000000  0.31084376
## oldpeak   0.010859481 -0.05011425 -0.349796163  0.31084376  1.00000000
## slope    -0.061902374  0.08608609  0.395307843 -0.26733547 -0.57518854
## ca        0.137156259 -0.07807235 -0.207888416  0.10784854  0.22181603
## thal     -0.042177320 -0.02050406 -0.098068165  0.19720104  0.20267203
## target   -0.041163547  0.13446821  0.422895496 -0.43802855 -0.43844127
##                slope          ca        thal      target
## age      -0.16910511  0.27155053  0.07229745 -0.22932355
```

```
## sex      -0.02666629  0.11172891  0.19842425 -0.27950076
## cp        0.13163278 -0.17620647 -0.16334148  0.43485425
## trestbps -0.12044531  0.10455372  0.05927618 -0.13877173
## chol     -0.01424787  0.07425934  0.10024418 -0.09996559
## fbs      -0.06190237  0.13715626 -0.04217732 -0.04116355
## restecg   0.08608609 -0.07807235 -0.02050406  0.13446821
## thalach   0.39530784 -0.20788842 -0.09806817  0.42289550
## exang    -0.26733547  0.10784854  0.19720104 -0.43802855
## oldpeak  -0.57518854  0.22181603  0.20267203 -0.43844127
## slope     1.00000000 -0.07344041 -0.09409006  0.34551175
## ca       -0.07344041  1.00000000  0.14901387 -0.38208529
## thal     -0.09409006  0.14901387  1.00000000 -0.33783815
## target    0.34551175 -0.38208529 -0.33783815  1.00000000
```

```
# Load libraries
library(ggplot2)
library(reshape2)

# Melt correlation matrix into long format
heart_corr_melted <- melt(heart_df_corr)

# Create heatmap
ggplot(heart_corr_melted, aes(x = Var2, y = Var1, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "#f7fbff", mid = "blue", high = "red", midpoint
= 0,
                       name = "Correlation", na.value = "white", guide = "col
orbar") +
  labs(title = "Heart Disease Correlation Heatmap", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 12),
        axis.text.y = element_text(size = 12),
        plot.title = element_text(size = 20, face = "bold", hjust = 0.5),
        legend.title = element_text(size = 14),
        legend.text = element_text(size = 12))
```




```
heart_df$target <- as.factor(heart_df$target)
is.factor(heart_df$target)

## [1] TRUE
```

1) RESEARCH QUESTION Which classification method (logistic regression, decision trees, KNN, random forest) resulted in the highest accuracy in predicting the presence of heart disease in individuals?

Using Train and Test method

```
#make this example reproducible
set.seed(234)

#use 80% of dataset as training set and 20% as test set
sample <- sample(c(TRUE, FALSE), nrow(heart_df), replace=TRUE, prob=c(0.7,0.3
))
train  <- heart_df[sample, ]
test   <- heart_df[!sample, ]

print("70% of Train Data")

## [1] "70% of Train Data"

dim(train)

## [1] 756  14

print("30% of Test Data")

## [1] "30% of Test Data"

dim(test)

## [1] 269  14
```

LOGISTIC REGRESSION

```
# Logistic Regression Model with Train & Test Method
library(caret)

## Loading required package: lattice

# Fit the model on the training set
logistic_classification <- glm(target ~ ., data = train, family = 'binomial')
summary(logistic_classification)

##
## Call:
## glm(formula = target ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.67858  -0.35724   0.08508   0.57332   2.52677
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.402697   1.699992   2.002 0.045328 *
## age         -0.001805   0.015380  -0.117 0.906565
## sex         -1.858825   0.314642  -5.908 3.47e-09 ***
## cp           1.028463   0.128123   8.027 9.98e-16 ***
## trestbps    -0.012178   0.006780  -1.796 0.072483 .
## chol        -0.004752   0.002520  -1.886 0.059331 .
## fbs         -0.472835   0.350944  -1.347 0.177877
## restecg      0.628854   0.228371   2.754 0.005893 **
## thalach      0.017494   0.006585   2.657 0.007891 **
## exang       -1.052008   0.273164  -3.851 0.000118 ***
## oldpeak     -0.709656   0.140722  -5.043 4.58e-07 ***
## slope        0.507604   0.237316   2.139 0.032441 *
## ca          -0.864774   0.125031  -6.916 4.63e-12 ***
## thal        -0.894935   0.182721  -4.898 9.69e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1047.78  on 755  degrees of freedom
## Residual deviance:  500.31  on 742  degrees of freedom
## AIC: 528.31
##
## Number of Fisher Scoring iterations: 6

# Make predictions on the test set
predictions_logistic_classification <- predict(logistic_classification, newda
ta = test, type = "response")

# Convert probabilities to class predictions
```

```
predictionClass_logistic_classification <- ifelse(predictions_logistic_classi
fication > 0.5, 1, 0)

# Create the confusion matrix
conf_matrix_logistic_classification <- confusionMatrix(factor(predictionClass
_logistic_classification), factor(test$target))
conf_matrix_logistic_classification
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  91  10
##          1  37 131
##
##                Accuracy : 0.8253
##                  95% CI : (0.7745, 0.8687)
##     No Information Rate : 0.5242
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6463
##
##  Mcnemar's Test P-Value : 0.0001491
##
##             Sensitivity : 0.7109
##             Specificity : 0.9291
##          Pos Pred Value : 0.9010
##          Neg Pred Value : 0.7798
##              Prevalence : 0.4758
##          Detection Rate : 0.3383
##    Detection Prevalence : 0.3755
##       Balanced Accuracy : 0.8200
##
##        'Positive' Class : 0
##
```

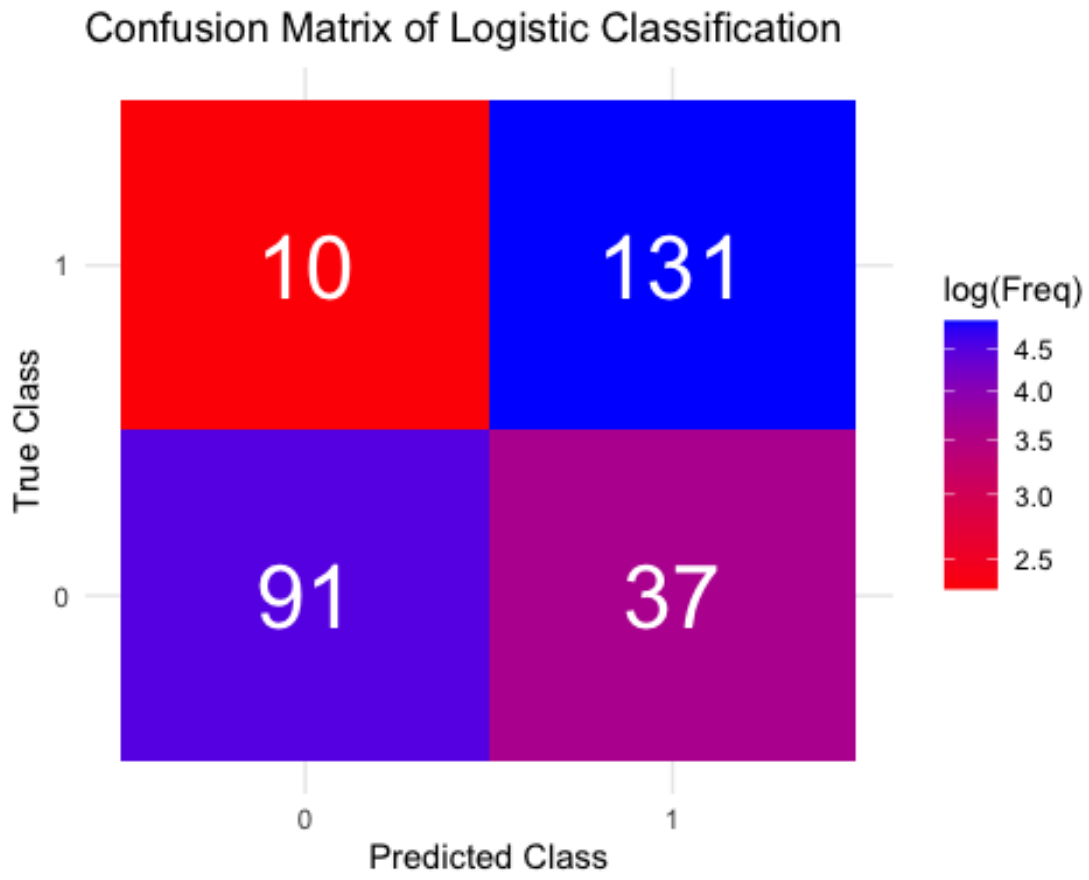
```
# Extract the accuracy metric
accuracy_logistic_classification <- conf_matrix_logistic_classification$overa
ll['Accuracy']
accuracy_logistic_classification
```

```
##  Accuracy
## 0.8252788
```

CONFUSION MATRIX OF LOGISTIC CLASSIFICATION

```
# Assuming conf_matrix is a table object created by confusionMatrix()
conf_matrix_logistic_classification <- as.data.frame(conf_matrix_logistic_cla
ssification$table)

# Plot confusion matrix as a heatmap
```

```
library(ggplot2)
ggplot(conf_matrix_logistic_classification, aes(Prediction, Reference, fill =
log(Freq))) +
  geom_tile() +
  scale_fill_gradient(low = "red", high = "blue", na.value = "grey50", trans
= "log") +
  geom_text(aes(label = Freq), size = 10, color = "white") +
  xlab("Predicted Class") +
  ylab("True Class") +
  ggtitle("Confusion Matrix of Logistic Classification") +
  theme_minimal()
```



K-NEAREST NEIGHBOURS CLASSIFICATION

```
# Scale the predictors
preProcValues_knn <- preProcess(train[, -14], method = c("center", "scale"))
train[, -14] <- predict(preProcValues_knn, train[, -14])
test[, -14] <- predict(preProcValues_knn, test[, -14])
```

```r
# Train the KNN model
k <- 5
knnmodel <- train(target ~ ., data = train, method = "knn", trControl = train
Control(method = "cv", number = 5), preProcess = c("center", "scale"), tuneLe
ngth = 10, metric = "Accuracy", tuneGrid = expand.grid(k = k))

# Print the accuracy and confusion matrix on the testing set
predictions_knn <- predict(knnmodel, newdata = test)
confusion_matrix_knn <- confusionMatrix(predictions_knn, test$target)
print(paste0("Accuracy: ", confusion_matrix_knn$overall["Accuracy"]))
```

```
## [1] "Accuracy: 0.83271375464684"
```

```r
print(confusion_matrix_knn$table)
```

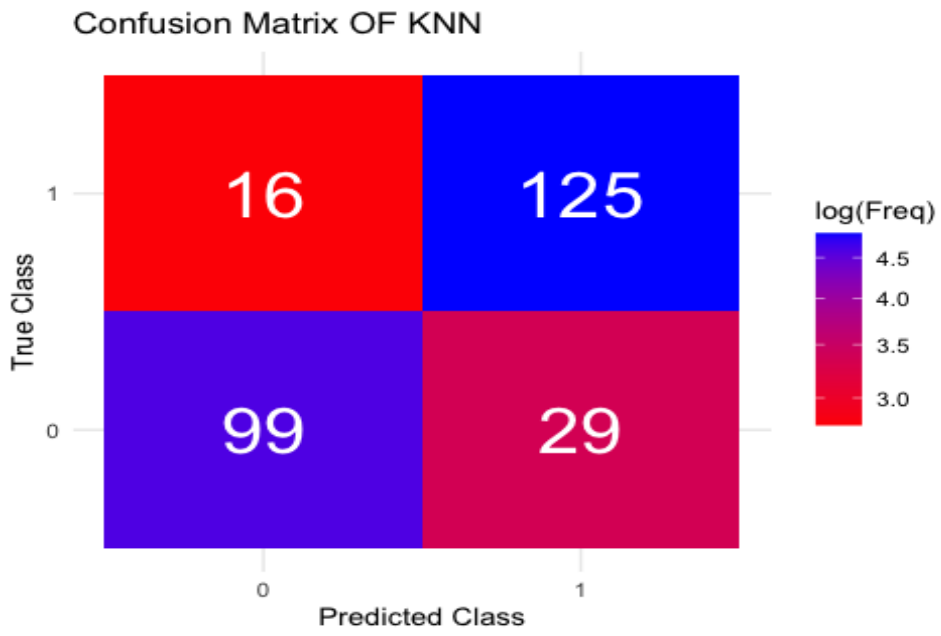
```
##            Reference
## Prediction   0   1
##          0  99  16
##          1  29 125
```

```r
accuracy_knn_classification <- confusion_matrix_knn$overall['Accuracy']
accuracy_knn_classification
```

```
##  Accuracy
## 0.8327138
```

CONFUSION MATRIX OF K-NEAREST NEIGHBOURS CLASSIFICATION

```r
# Assuming conf_matrix is a table object created by confusionMatrix()
conf_matrix_knn <- as.data.frame(confusion_matrix_knn$table)

# Plot confusion matrix as a heatmap
library(ggplot2)
ggplot(conf_matrix_knn, aes(Prediction, Reference, fill = log(Freq))) +
  geom_tile() +
  scale_fill_gradient(low = "red", high = "blue", na.value = "grey50", trans
= "log") +
  geom_text(aes(label = Freq), size = 10, color = "white") +
  xlab("Predicted Class") +
  ylab("True Class") +
  ggtitle("Confusion Matrix OF KNN") +
  theme_minimal()
```



Random Forest Classification

```
#Random forest Classification
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

library(caret)


# Train the random forest model
rf_model <- randomForest(target ~ ., data=train, ntree=500, mtry=3)

# Make predictions on the test data
rf_predictions <- predict(rf_model, test)

# Evaluate the model
rf_cm <- table(rf_predictions, test$target)
rf_acc <- sum(diag(rf_cm)) / sum(rf_cm)
```

```r
# Print the results
print(paste("Random Forest Accuracy:", rf_acc))
```

```
## [1] "Random Forest Accuracy: 0.988847583643123"
```

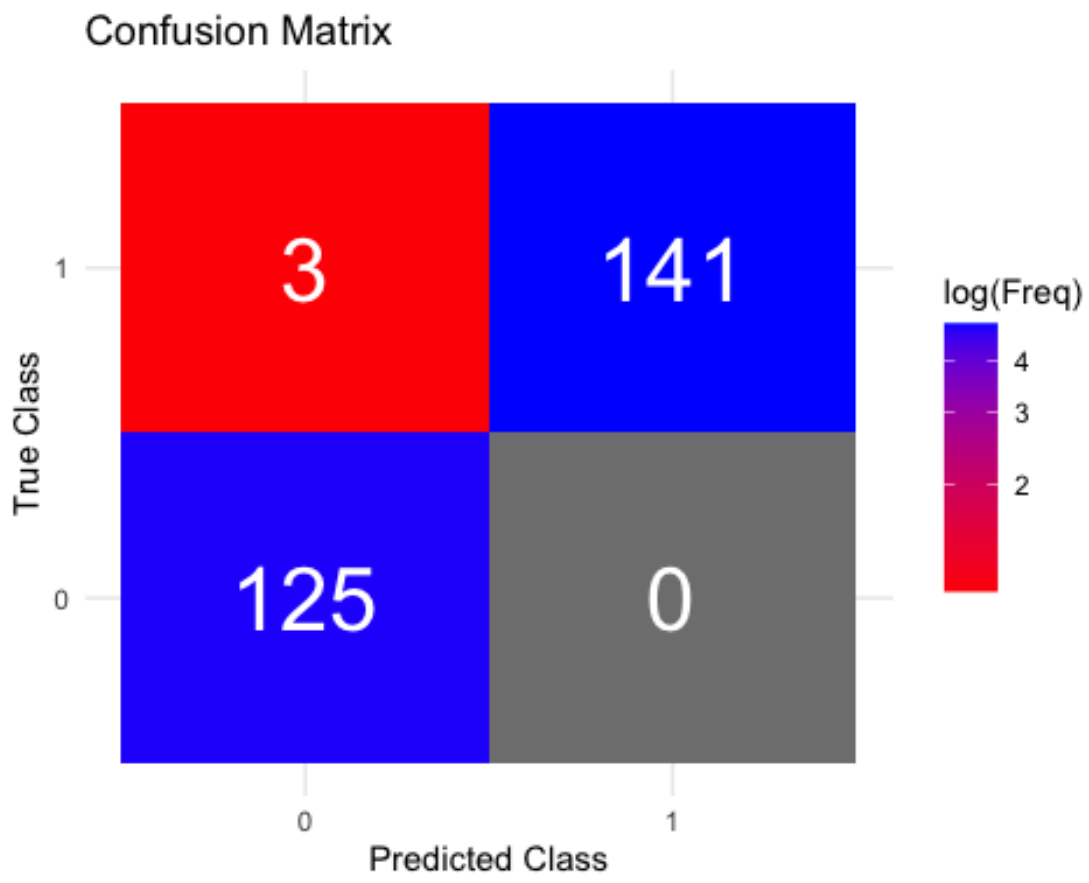
```r
rf_cm
```

```
##
## rf_predictions   0   1
##              0 125   0
##              1   3 141
```

CONFUSION MATRIX OF RANDOM FOREST CLASSIFICATION

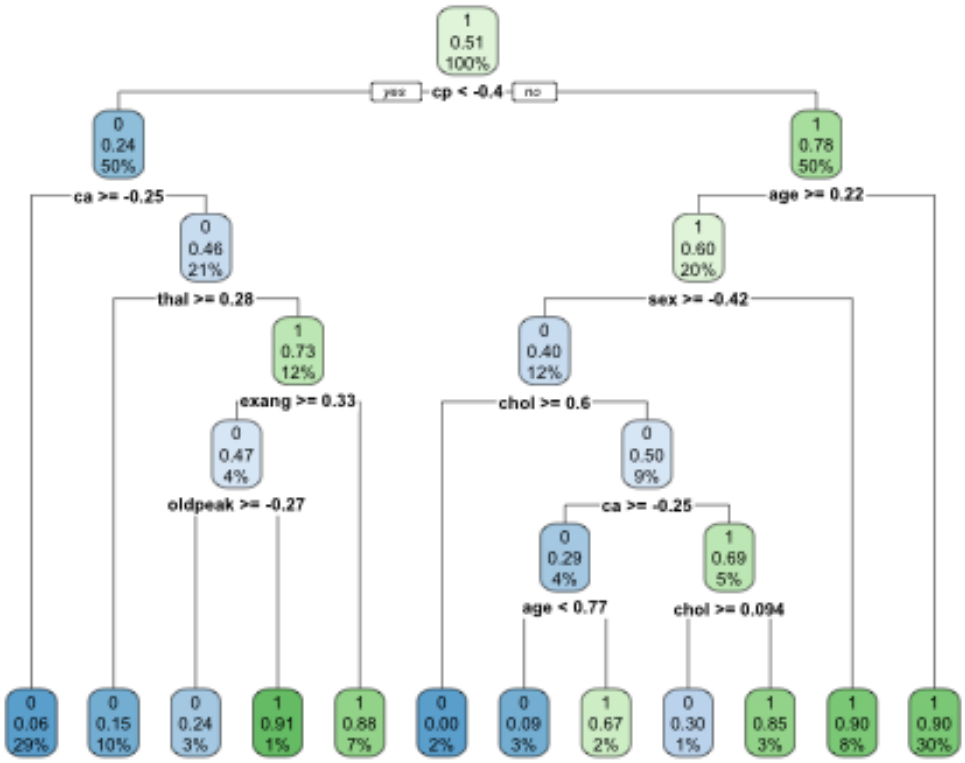
```r
# Assuming conf_matrix is a table object created by confusionMatrix()
conf_matrix_RFC <- as.data.frame(rf_cm)

# Rename the column names in conf_matrix_RFC to match the expected names
names(conf_matrix_RFC) <- c("Reference", "Prediction", "Freq")

# Plot confusion matrix as a heatmap
library(ggplot2)
ggplot(conf_matrix_RFC, aes(Prediction, Reference, fill = log(Freq))) +
  geom_tile() +
  scale_fill_gradient(low = "red", high = "blue", na.value = "grey50", trans
= "log") +
  geom_text(aes(label = Freq), size = 10, color = "white") +
  xlab("Predicted Class") +
  ylab("True Class") +
  ggtitle("Confusion Matrix") +
  theme_minimal()
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning in self$trans$transform(x): NaNs produced
```



Decision Tree Classification

```
library(rpart)
# fit Decision Tree model
DecisiontreeModel <- rpart(target ~., data=train, method="class")

library(rpart.plot)
# plot Decision Tree
rpart.plot(DecisiontreeModel)
```




```
# make predictions on testing set
predictions_Decision_tree <- predict(DecisiontreeModel, test, type="class")

# evaluate model performance
conf_matix_Decision_tree <- confusionMatrix(predictions_Decision_tree, test$t
arget)
conf_matix_Decision_tree
```

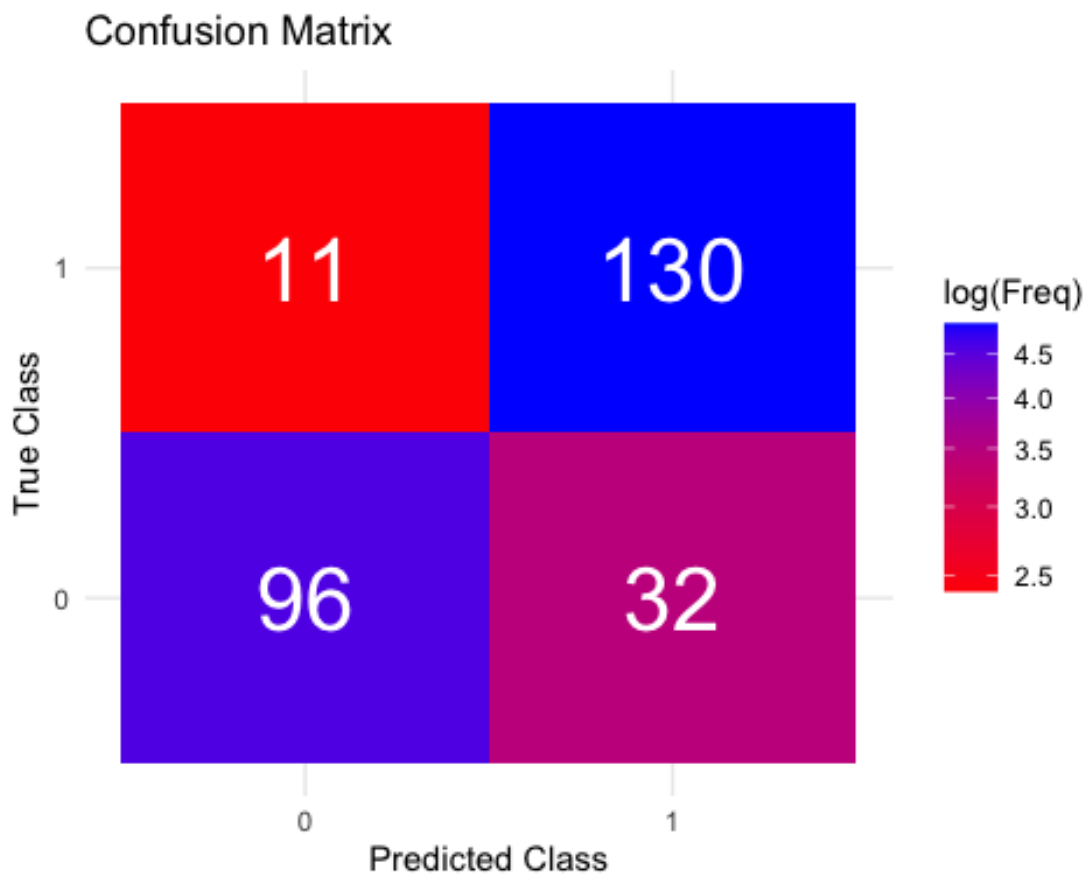
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  96  11
##          1  32 130
##
##                Accuracy : 0.8401
##                  95% CI : (0.7908, 0.8818)
##     No Information Rate : 0.5242
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6771
##
##  Mcnemar's Test P-Value : 0.002289
```

```
##
##             Sensitivity : 0.7500
##             Specificity : 0.9220
##          Pos Pred Value : 0.8972
##          Neg Pred Value : 0.8025
##              Prevalence : 0.4758
##          Detection Rate : 0.3569
##    Detection Prevalence : 0.3978
##       Balanced Accuracy : 0.8360
##
##        'Positive' Class : 0
##
```

```
# Extract the accuracy metric
accuracy_decision_tree <- conf_matix_Decision_tree$overall['Accuracy']
accuracy_decision_tree
```

```
##  Accuracy
## 0.8401487
```

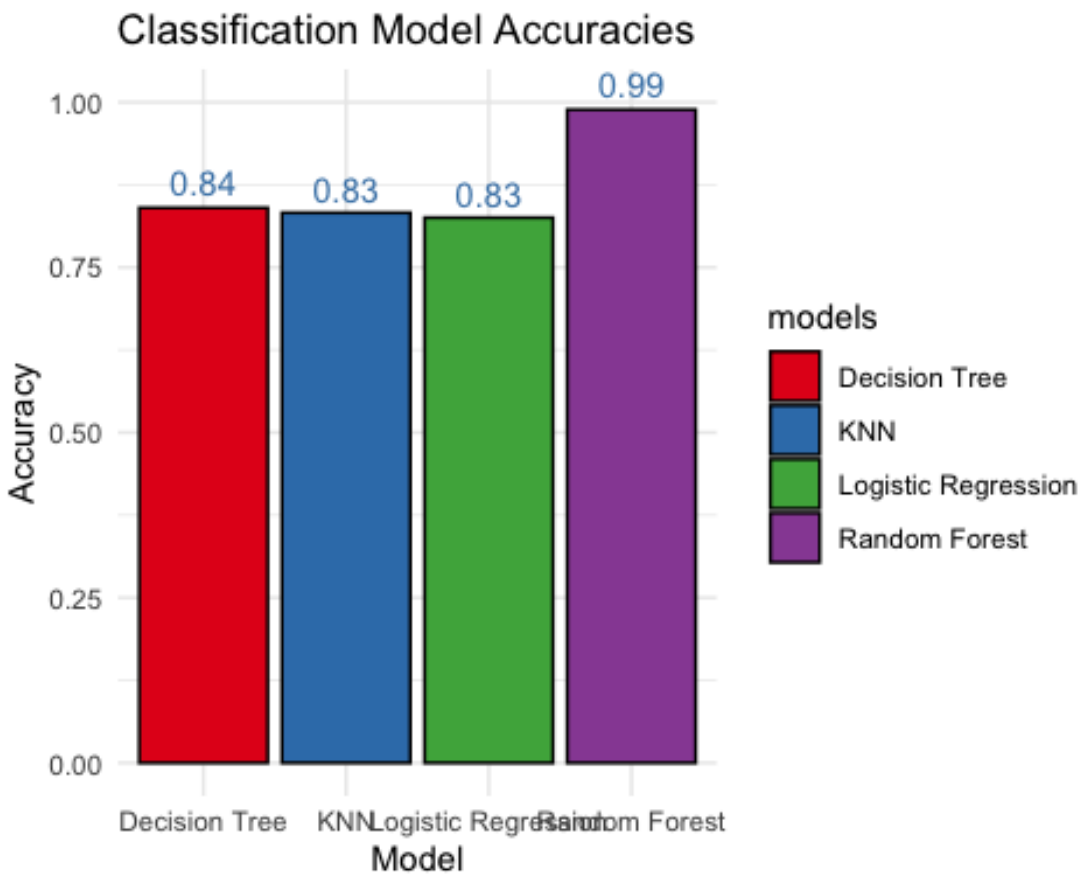
```
# Assuming conf_matrix is a table object created by confusionMatrix()
conf_matrix_Decision_tree <- as.data.frame(conf_matix_Decision_tree$table)

# Plot confusion matrix as a heatmap
library(ggplot2)
ggplot(conf_matrix_Decision_tree, aes(Prediction, Reference, fill = log(Freq)
)) +
  geom_tile() +
  scale_fill_gradient(low = "red", high = "blue", na.value = "grey50", trans
= "log") +
  geom_text(aes(label = Freq), size = 10, color = "white") +
  xlab("Predicted Class") +
  ylab("True Class") +
  ggtitle("Confusion Matrix") +
  theme_minimal()
```



Printing all the classification Accuracies

```
cat("Accuracy of Logistic Classification:", accuracy_logistic_classification,
"\n")

## Accuracy of Logistic Classification: 0.8252788

cat("Accuracy of KNN Classification:", accuracy_knn_classification, "\n")

## Accuracy of KNN Classification: 0.8327138

cat("Accuaracy of Random Forest Classification:", rf_acc, "\n")

## Accuaracy of Random Forest Classification: 0.9888476

cat("Accuracy of Decision Tree Classification:", accuracy_decision_tree, "\n"
)

## Accuracy of Decision Tree Classification: 0.8401487

library(ggplot2)
library(RColorBrewer)
```

```r
# create a data frame with the model names and accuracies
models <- c("Logistic Regression", "KNN", "Random Forest", "Decision Tree")
accuracies <- c(accuracy_logistic_classification, accuracy_knn_classification
, rf_acc, accuracy_decision_tree)
Classifications_accuracies_df <- data.frame(models, accuracies)

# specify the color palette
colors <- brewer.pal(length(models), "Set1")

# create the bar plot with the specified colors
ggplot(Classifications_accuracies_df, aes(x = models, y = accuracies, fill =
models)) +
  geom_bar(stat = "identity", color = "black") +
  scale_fill_manual(values = colors) +
  ylim(0,1) +
  ggtitle("Classification Model Accuracies") +
  xlab("Model") +
  ylab("Accuracy") +
  theme_minimal() + geom_text(aes(label = round(accuracies, 2)), vjust = -0.5
, color = "steelblue")
```



2) RESEARCH QUESTION Do patients with heart disease have significantly higher mean blood pressure compared to patients without heart disease?

To compare the mean blood pressure between patients with and without heart disease in R, you can use the t-test.

```
# Load the required package for reading the dataset
library(readr)

# Separate the blood pressure values for patients with and without heart dise
ase
bp_with_heart_disease <- heart_df$trestbps[heart_df$target == 1]

bp_without_heart_disease <- heart_df$trestbps[heart_df$target == 0]

# Perform the t-test
result <- t.test(bp_with_heart_disease, bp_without_heart_disease)

# Print the t-test results
cat("t-Statistic:", result$statistic, "\n")

## t-Statistic: -4.465215

cat("p-value:", result$p.value, "\n")

## p-value: 8.922492e-06
```

t-Statistic: -4.465215

The t-statistic represents the calculated value of the t-test. In this case, the t-statistic is -4.465215. The sign indicates the direction of the difference between the means (whether one group has a higher mean than the other), and the magnitude represents the size of the difference.

p-value: 8.922492e-06

The p-value is a measure of the evidence against the null hypothesis. In this case, the p-value is 8.922492e-06, which is very small. The “e-06” represents scientific notation, and it means that the p-value is extremely close to zero (e.g., 0.000008922492). The small p-value suggests strong evidence to reject the null hypothesis, which typically states that there is no difference between the means of the two groups. In this case, it indicates that there is a significant difference in mean blood pressure between patients with and without heart disease. Since the p-value is less than the commonly used significance level of 0.05 (or 0.01), it indicates that the observed difference in mean blood pressure is unlikely to have occurred by chance alone, assuming the null hypothesis is true. In summary, the t-test output suggests that there is a statistically significant difference in mean blood pressure between patients with and without heart disease, with the t-statistic indicating the direction and magnitude of the difference.

Effect size calculation: Although the t-test determines whether there is a significant difference, it does not provide information about the magnitude of the difference. You can calculate an effect size measure, such as Cohen's d or Hedges' g, to quantify the practical significance of the observed effect.

```
# Load the required package for effect size calculation
library(effsize)

# Calculate Cohen's d
cohen_d <- cohen.d(bp_with_heart_disease, bp_without_heart_disease)

# Print the effect size measures
cohen_d
```

```
##
## Cohen's d
##
## d estimate: -0.2800787 (small)
## 95 percent confidence interval:
##      lower      upper
## -0.4033036 -0.1568539
```

The output suggests the following information regarding Cohen's d:

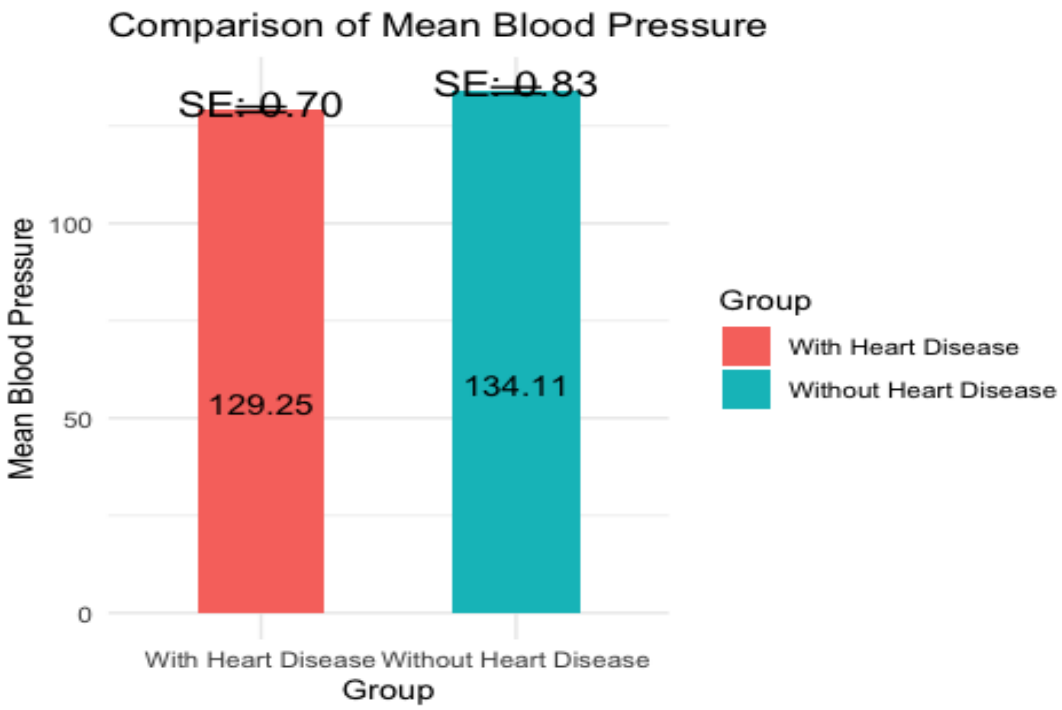
Cohen's d estimate: -0.2800787 The calculated value of Cohen's d is -0.2800787. Cohen's d represents the standardized difference between the means of the two groups (patients with heart disease and those without heart disease). In this case, the negative sign indicates that patients without heart disease have, on average, slightly higher blood pressure than those with heart disease. 95 percent confidence interval: The confidence interval provides a range of plausible values for the true effect size in the population. In this case, the 95 percent confidence interval for Cohen's d is (-0.4033036, -0.1568539). This means that we can be 95 percent confident that the true value of Cohen's d falls within this interval. Interpretation: The magnitude of Cohen's d is typically interpreted as follows: A small effect size: Cohen's d around 0.2. A medium effect size: Cohen's d around 0.5. A large effect size: Cohen's d of 0.8 or higher. In this case, with Cohen's d estimated as -0.2800787, it falls within the small effect size range. This suggests that the difference in mean blood pressure between patients with and without heart disease is relatively small, but still statistically significant.

```
# Load the required packages
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

# Calculate mean and standard error for each group
mean_with_heart_disease <- mean(bp_with_heart_disease)
mean_without_heart_disease <- mean(bp_without_heart_disease)
se_with_heart_disease <- sd(bp_with_heart_disease) / sqrt(length(bp_with_hear
t_disease))
se_without_heart_disease <- sd(bp_without_heart_disease) / sqrt(length(bp_wit
hout_heart_disease))

mean_with_heart_disease

## [1] 129.2452

mean_without_heart_disease

## [1] 134.1062

se_with_heart_disease

## [1] 0.7025248

se_without_heart_disease

## [1] 0.8316089

# Create a data frame for plotting
df_group_mean_se <- data.frame(
  Group = c("With Heart Disease", "Without Heart Disease"),
  Mean = c(mean_with_heart_disease, mean_without_heart_disease),
  SE = c(se_with_heart_disease, se_without_heart_disease)
)

# Plot the bar plot with error bars
ggplot(df_group_mean_se, aes(x = Group, y = Mean, fill = Group)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.5) +
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE), width = 0.2) +
  labs(x = "Group", y = "Mean Blood Pressure") +
  ggtitle("Comparison of Mean Blood Pressure") +
  theme_minimal() +
  geom_text(aes(label = sprintf("%.2f", Mean)), vjust = 15.0, color = "black"
, size = 4, position = position_dodge(width = 0.5)) +
  geom_text(aes(y = Mean + SE + 1, label = sprintf("SE: %.2f", SE)), color =
"black", size = 5, position = position_dodge(width = 0.5))
```




```
cat("mean with heart disease:",mean_with_heart_disease,"\n")

## mean with heart disease: 129.2452

cat("mean without heart disease:", mean_without_heart_disease,"\n")

## mean without heart disease: 134.1062

cat("se with heart disease:", se_with_heart_disease,"\n")

## se with heart disease: 0.7025248

cat("se without heart disease:", se_without_heart_disease,"\n")

## se without heart disease: 0.8316089
```

This visualization directly compares the mean blood pressure for each group using bars. The inclusion of error bars provides a visual representation of the uncertainty or variability around the mean estimates. If the error bars do not overlap or show minimal overlap, it visually indicates a significant difference in mean blood pressure between the two groups. This clear visual distinction allows viewers to immediately understand the significant difference. By using a bar plot with error bars, you can effectively communicate the key finding that patients with heart disease have a significantly higher mean blood pressure compared to patients without heart disease. Make sure to include appropriate labels, legends, and statistical annotations (such as asterisks denoting the significance level or p-value) to enhance the clarity and interpretability of the visualization.