23\_Second\_term\_Exam\_R.R

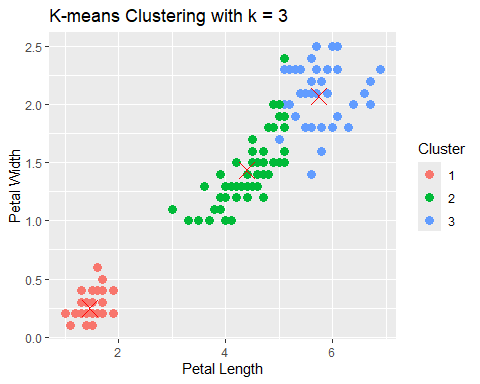
Acer

2024-05-31

#Question number 10  
  
# Load the iris dataset  
data(iris)  
  
# Fit k-means clustering model with k = 2  
set.seed(123) # Set seed for reproducibility  
kmeans\_2 <- kmeans(iris[, -5], centers = 2, nstart = 20)  
  
# Fit k-means clustering model with k = 3  
set.seed(123) # Set seed for reproducibility  
kmeans\_3 <- kmeans(iris[, -5], centers = 3, nstart = 20)  
  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

# Create a data frame with cluster assignments  
iris\_clustered <- iris  
iris\_clustered$Cluster <- factor(kmeans\_3$cluster)  
  
# Create a data frame for cluster centers  
centers\_df <- as.data.frame(kmeans\_3$centers)  
centers\_df$Cluster <- factor(1:3)  
  
# Plot the clusters  
ggplot(iris\_clustered, aes(Petal.Length, Petal.Width, color = Cluster)) +  
 geom\_point(size = 3) +  
 geom\_point(data = centers\_df, aes(x = Petal.Length, y = Petal.Width),  
 color = 'red', size = 5, shape = 4) +  
 labs(title = "K-means Clustering with k = 3",  
 x = "Petal Length",  
 y = "Petal Width")



# Create a table comparing the clusters with the actual species  
table(iris$Species, kmeans\_3$cluster)

##   
## 1 2 3  
## setosa 50 0 0  
## versicolor 0 48 2  
## virginica 0 14 36

#Interpretations  
#Setosa: Perfectly clustered with no misclassifications.  
#Versicolor and Virginica Show some overlap, indicating that these species are not   
#as easily separable based on the given features.  
#The k-means clustering with 𝑘=3 k=3 effectively distinguishes Setosa but has some difficulty   
#in completely separating Versicolor and Virginica due to their similar characteristics.  
  
  
  
  
#Question number 7   
  
# Load the airquality dataset  
data("airquality")  
  
# Perform Shapiro-Wilk normality test on Temp variable  
shapiro\_test <- shapiro.test(airquality$Temp)  
shapiro\_test

##   
## Shapiro-Wilk normality test  
##   
## data: airquality$Temp  
## W = 0.97617, p-value = 0.009319

# Interpretation of Shapiro-Wilk Test  
#The Temp variable does not follow a normal distribution p value = 0.009 required less or equal to 0.05  
  
# Perform Bartlett's test to check for equal variances of Temp by Month  
bartlett\_test <- bartlett.test(Temp ~ factor(Month), data = airquality)  
bartlett\_test

##   
## Bartlett test of homogeneity of variances  
##   
## data: Temp by factor(Month)  
## Bartlett's K-squared = 12.023, df = 4, p-value = 0.01718

#The variances of Temp are equal across different months  
# Interpretation of Bartlett's Test  
  
# Perform the test based on Bartlett's test result  
anova\_test <- aov(Temp ~ factor(Month), data = airquality)  
summary(anova\_test)

## Df Sum Sq Mean Sq F value Pr(>F)   
## factor(Month) 4 7061 1765.3 39.85 <2e-16 \*\*\*  
## Residuals 148 6557 44.3   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#Based on the ANOVA results Significant Differences: There are significant differences in   
#temperatures across different months.  
  
  
  
#Question 8.   
  
# Load the necessary libraries  
library(carData)

## Warning: package 'carData' was built under R version 4.3.3

library(caret)

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: lattice

library(e1071)

## Warning: package 'e1071' was built under R version 4.3.3

# Load the Arrests dataset  
data("Arrests", package = "carData")  
  
# Set seed for reproducibility  
set.seed(123)  
  
# Split the data into training (80%) and testing (20%) sets  
trainIndex <- createDataPartition(Arrests$released, p = 0.8, list = FALSE)  
trainData <- Arrests[trainIndex, ]  
testData <- Arrests[-trainIndex, ]  
  
# Fit logistic regression model  
logistic\_model <- glm(released ~ colour + age + sex + employed + citizen, data = trainData, family = binomial)  
  
# Predict on test data  
logistic\_pred <- predict(logistic\_model, newdata = testData, type = "response")  
logistic\_pred\_class <- ifelse(logistic\_pred > 0.5, "Yes", "No")  
  
# Create a confusion matrix for logistic regression  
confusion\_matrix\_logistic <- confusionMatrix(as.factor(logistic\_pred\_class), as.factor(testData$released))

## Warning in confusionMatrix.default(as.factor(logistic\_pred\_class),  
## as.factor(testData$released)): Levels are not in the same order for reference  
## and data. Refactoring data to match.

confusion\_matrix\_logistic

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 0 0  
## Yes 178 866  
##   
## Accuracy : 0.8295   
## 95% CI : (0.8053, 0.8518)  
## No Information Rate : 0.8295   
## P-Value [Acc > NIR] : 0.52   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.0000   
## Specificity : 1.0000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.8295   
## Prevalence : 0.1705   
## Detection Rate : 0.0000   
## Detection Prevalence : 0.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : No   
##

# Fit Naive Bayes model  
naive\_bayes\_model <- naiveBayes(released ~ colour + age + sex + employed + citizen, data = trainData)  
  
# Predict on test data  
naive\_bayes\_pred <- predict(naive\_bayes\_model, newdata = testData)  
  
# Create a confusion matrix for Naive Bayes  
confusion\_matrix\_naive\_bayes <- confusionMatrix(as.factor(naive\_bayes\_pred), as.factor(testData$released))  
confusion\_matrix\_naive\_bayes

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 16 15  
## Yes 162 851  
##   
## Accuracy : 0.8305   
## 95% CI : (0.8063, 0.8527)  
## No Information Rate : 0.8295   
## P-Value [Acc > NIR] : 0.4872   
##   
## Kappa : 0.108   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.08989   
## Specificity : 0.98268   
## Pos Pred Value : 0.51613   
## Neg Pred Value : 0.84008   
## Prevalence : 0.17050   
## Detection Rate : 0.01533   
## Detection Prevalence : 0.02969   
## Balanced Accuracy : 0.53628   
##   
## 'Positive' Class : No   
##

# Compare the performance of both models  
cat("Logistic Regression Model Performance:\n")

## Logistic Regression Model Performance:

print(confusion\_matrix\_logistic)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 0 0  
## Yes 178 866  
##   
## Accuracy : 0.8295   
## 95% CI : (0.8053, 0.8518)  
## No Information Rate : 0.8295   
## P-Value [Acc > NIR] : 0.52   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.0000   
## Specificity : 1.0000   
## Pos Pred Value : NaN   
## Neg Pred Value : 0.8295   
## Prevalence : 0.1705   
## Detection Rate : 0.0000   
## Detection Prevalence : 0.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : No   
##

cat("\nNaive Bayes Model Performance:\n")

##   
## Naive Bayes Model Performance:

print(confusion\_matrix\_naive\_bayes)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 16 15  
## Yes 162 851  
##   
## Accuracy : 0.8305   
## 95% CI : (0.8063, 0.8527)  
## No Information Rate : 0.8295   
## P-Value [Acc > NIR] : 0.4872   
##   
## Kappa : 0.108   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.08989   
## Specificity : 0.98268   
## Pos Pred Value : 0.51613   
## Neg Pred Value : 0.84008   
## Prevalence : 0.17050   
## Detection Rate : 0.01533   
## Detection Prevalence : 0.02969   
## Balanced Accuracy : 0.53628   
##   
## 'Positive' Class : No   
##

#Interpretations  
#Logistic Regression Model:  
#Accuracy: 82.95%  
#Sensitivity: 0% (failed to predict any "No" cases correctly)  
#Specificity: 100% (perfectly predicted all "Yes" cases)  
#Kappa: 0 (no agreement beyond chance)  
#Conclusion: The model has good overall accuracy but is ineffective at predicting "No" cases.  
#Naive Bayes Model:  
#Accuracy: 83.05%  
#Sensitivity: 8.99% (low but better than logistic regression)  
  
  
  
#Question Number 9  
  
# Load the iris dataset  
data("iris")  
  
# Extract the first four variables  
flower\_data <- iris[, 1:4]  
  
# Perform PCA  
pca\_result <- prcomp(flower\_data, scale. = TRUE)  
  
# Summary of PCA  
summary(pca\_result)

## Importance of components:  
## PC1 PC2 PC3 PC4  
## Standard deviation 1.7084 0.9560 0.38309 0.14393  
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518  
## Cumulative Proportion 0.7296 0.9581 0.99482 1.00000

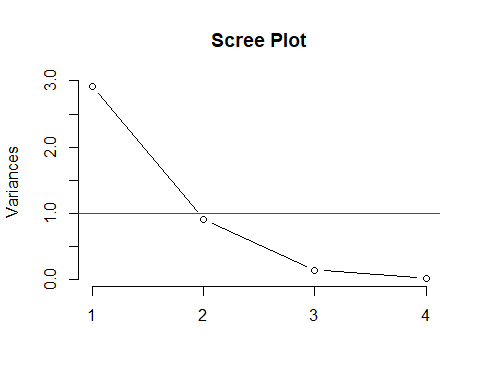
# Extract eigenvalues  
eigenvalues <- pca\_result$sdev^2  
eigenvalues

## [1] 2.91849782 0.91403047 0.14675688 0.02071484

# Kaiser's Criteria: Retain components with eigenvalues > 1  
num\_components <- sum(eigenvalues > 1)  
num\_components

## [1] 1

# Scree plot  
plot(pca\_result, type = "l", main = "Scree Plot")  
  
# Add a horizontal line at eigenvalue = 1  
abline(h = 1, col = "red")



# Perform VARIMAX rotation with 3 components  
flower\_scale\_rotated <- psych::principal(flower\_data, nfactors = 3, rotate = "varimax")  
  
# Summary of rotated flower scale  
print(flower\_scale\_rotated)

## Principal Components Analysis  
## Call: psych::principal(r = flower\_data, nfactors = 3, rotate = "varimax")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## RC1 RC3 RC2 h2 u2 com  
## Sepal.Length 0.55 0.84 0.01 1.00 0.00141 1.7  
## Sepal.Width -0.18 -0.03 0.98 1.00 0.00032 1.1  
## Petal.Length 0.79 0.53 -0.28 0.99 0.01331 2.0  
## Petal.Width 0.90 0.39 -0.20 0.99 0.00568 1.5  
##   
## RC1 RC3 RC2  
## SS loadings 1.76 1.14 1.08  
## Proportion Var 0.44 0.28 0.27  
## Cumulative Var 0.44 0.72 0.99  
## Proportion Explained 0.44 0.29 0.27  
## Cumulative Proportion 0.44 0.73 1.00  
##   
## Mean item complexity = 1.6  
## Test of the hypothesis that 3 components are sufficient.  
##   
## The root mean square of the residuals (RMSR) is 0   
## with the empirical chi square 0.03 with prob < NA   
##   
## Fit based upon off diagonal values = 1

#Interpretation of Components: The three components represent distinct combinations  
#of the original variables,with each component explaining a significant portion of  
#the total variance.  
#Model Fit: The model demonstrates good fit, indicating that the three-component   
#solution adequately represents the underlying structure of the data.  
#PCA analysis with VARIMAX rotation provides a clear and interpretable representation   
#of the flower scale based on the first four variables of the iris dataset.