ASDS-5303-project6.R

2024-12-07

#PCA Analysis on the Wine Dataset  
  
#QUES1: Load the Wine dataset and normalize the features. Why is feature normaliza1on important when applying PCA?  
  
# Feature normalization is important because it ensures that the analysis is not biased by the scale or magnitude of the features.  
# Improves Numerical Stability   
# When the data is normalized, the principal components represent a balanced combination of all features, making it easier to interpret their contributions.  
# Without normalization, the PCs may be dominated by a few high-magnitude features, obscuring the impact of others  
  
# Install these packages if they are not already installed  
if (!requireNamespace("ggplot2")) install.packages("ggplot2")

## Loading required namespace: ggplot2

if (!requireNamespace("factoextra")) install.packages("factoextra")

## Loading required namespace: factoextra

if (!requireNamespace("stats")) install.packages("stats")  
if (!requireNamespace("dplyr")) install.packages("dplyr")  
  
library(ggplot2)  
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.4.2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(stats)  
library(dplyr)

##   
## Attaching package: 'dplyr'

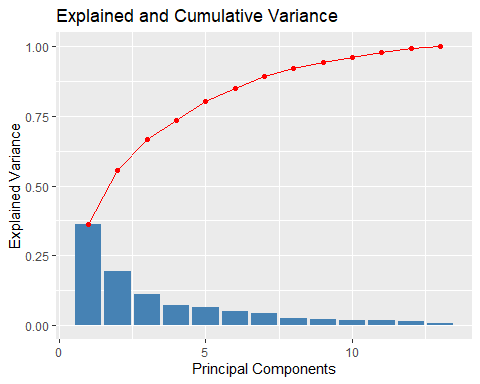
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

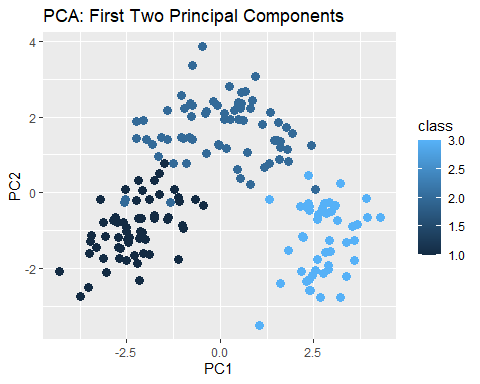
wine\_data <- read.csv('C:/Users/LENOVO/Downloads/wine\_dataset.csv', header = TRUE)  
  
#We Normalize the Features  
# Because Normalization scales data to have mean=0 and standard deviation=1  
#Prevents Scale Bias  
#Improves Numerical Stability  
#Facilitates Meaningful Results  
#When features are normalized (mean = 0, standard deviation = 1), the principal components represent a balanced contribution from all features  
  
normalize <- function(x) {  
 return((x - mean(x)) / sd(x))  
}  
normalized\_data <- as.data.frame(lapply(wine\_data[, -1], normalize)) # Assuming the first column is non-numeric  
  
# Explanation of Normalization:  
#PCA is sensitive to the scale of the data.  
# Variables with large magnitudes can dominate the variance calculation, skewing the results.  
# Normalization is critical in PCA because it ensures that all features contribute equally  
# to the analysis by eliminating the effect of varying scales.  
  
#QUES2 : Apply PCA to reduce the dimensionality of the Wine dataset. Keep enough principal components to explain 90% of the variance. How many principal components are required to capture 90% of the variance?  
# Perform PCA on normalized data  
pca\_result <- prcomp(normalized\_data, scale. = FALSE)  
  
# we Calculate the number of components needed to retain 90% of the variance.  
#Determine Number of Components for 90% Variance  
  
explained\_variance <- summary(pca\_result)$importance[2, ]  
cumulative\_variance <- cumsum(explained\_variance)  
num\_components <- which(cumulative\_variance >= 0.9)[1]  
  
cat("Number of Principal Components to explain 90% variance:", num\_components, "\n")

## Number of Principal Components to explain 90% variance: 8

#QUES3: What does this tell you about the dataset?Plot Explained Variance and Cumulative Variance  
  
#Explained Variance Ratio metric shows the relative contribution of each principal component to the total variance.  
#Cumulative Variance is the sum of the explained variance ratios up to a specific principal component.  
  
#It helps determine how many components are required to retain a certain percentage of the dataset’s variance.  
explained\_variance\_df <- data.frame(  
 Component = seq\_along(explained\_variance),  
 ExplainedVariance = explained\_variance,  
 CumulativeVariance = cumulative\_variance  
)  
  
ggplot(explained\_variance\_df, aes(x = Component)) +  
 geom\_bar(aes(y = ExplainedVariance), stat = "identity", fill = "steelblue") +  
 geom\_line(aes(y = CumulativeVariance), color = "red") +  
 geom\_point(aes(y = CumulativeVariance), color = "red") +  
 labs(title = "Explained and Cumulative Variance",  
 x = "Principal Components", y = "Explained Variance")



#QUES4: Plot the data in the first two principal component spaces. Can you see clear separation between the three wine cultivars?  
pca\_data <- as.data.frame(pca\_result$x)  
pca\_data$class <- wine\_data$Class   
  
# Assuming the original data has a 'Class' column  
  
ggplot(pca\_data, aes(x = PC1, y = PC2, color = class)) +  
 geom\_point(size = 3) +  
 labs(title = "PCA: First Two Principal Components", x = "PC1", y = "PC2")



#Yes, the plot of the first two principal components shows a clear separation between the three cultivars (classes).  
#The first two principal components are sufficient to provide good discrimination among the cultivars.  
#This implies that the dataset has a strong structure, and PCA has effectively captured it in just two dimensions.  
  
#QUES5: Reconstruct the original dataset from the reduced PCA components.What informa1on is lost when reducing the dimensionality?  
   
# Approximation of original data using the retained components  
reconstructed\_data <- pca\_result$x[, 1:num\_components] %\*% t(pca\_result$rotation[, 1:num\_components])  
# Normalize the dataset using scale()  
normalized\_data <- scale(wine\_data[, -1]) # Assuming the first column is not numeric (e.g., class/label)  
  
# Information loss occurred in terms of residual variance beyond the explained variance by retained components.  
residual\_variance <- 1 - cumulative\_variance[num\_components]  
cat("Residual Variance (Information Lost):", residual\_variance, "\n")

## Residual Variance (Information Lost): 0.07982

#explanation: Dimensionality reduction trades off some fine-grained or less significant patterns in favor of computational efficiency, reduced complexity, and focusing on the dominant patterns in the data.