**STAT 4360 (Introduction to Statistical Learning, Spring 2023)**

**Mini Project 5  
Name: Ann Biju**

1a.) Standardizing the variables before performing analysis would be a good idea since PCA is sensitive to the scaling of variables. If the variables are measured in different units or have different scales, a certain set of variables may dominate the final analysis. Standardization of variables ensures equal variable contribution.

1b.)

A white background with black numbers

Description automatically generated

A graph of a scree plot

Description automatically generated

As per the Scree Plot above, I would recommend 3 PCs, since this is where the “elbow” appears.

1c.

A screenshot of a computer

Description automatically generated

*Correlation table between standardized quantitative variables and the 1st two PCs*

A graph with black dots and red lines

Description automatically generated  
  
*Biplot between PC1 and PC2 shows us that Division has the least weight out of all other predictors, as its vector has the least length. Additionally there is a rather tight cluster in the top left of the biplot, showing a more significant contribution to each of the 2 PCs.*

2a.



2b.

A number of numbers and lines

Description automatically generated with medium confidence

M = 16 because the CV Error rate is smallest here.



2c.

A screenshot of a computer

Description automatically generated

M = 12 because the CV Error rate is smallest here.



2d.



2e. I would recommend the PLS model with M chosen optimally via LOOCV, since it has the smallest Test MSE rate of 0.08257%. A smaller test error rate points to more accuracy.

3a.



CWalks has the largest absolute coefficient (1.466e-03) and hence the most impact on the model. Therefore, it was deemed most important.

3b.

A close-up of a white background

Description automatically generated



3c.

A screenshot of a computer program

Description automatically generated

****

The GAM model and linear model both agree that the most important predictor of Salary would be CWalks.

**Python Code (or R Code)**

setwd("C:/ann/fall 2023/stat 4360/project 5")

**# Question 1b**

# Install and load required libraries

install.packages("ISLR")

library(ISLR)

#Hitters dataset

data("Hitters")

# Remove the response variable (Salary) and select only the predictor variables

hitters\_data <- Hitters[, -which(names(Hitters) == "Salary")]

# Standardize the variables

#convert League, Division, and NewLeague to numeric

hitters\_data$League <- as.numeric(as.factor(hitters\_data$League))

hitters\_data$Division <- as.numeric(as.factor(hitters\_data$Division))

hitters\_data$NewLeague <- as.numeric(as.factor(hitters\_data$NewLeague))

scaled\_hitters <- scale(hitters\_data)

# Perform PCA

pca\_result <- prcomp(scaled\_hitters, scale. = TRUE)

# Summary of PCA

summary(pca\_result)

# Scree plot - to determine where the elbow falls

plot(pca\_result, type = "l", main = "Scree Plot")

# how many variables to retain, with at least 70% of variance

cumulative\_variance <- cumsum(pca\_result$sdev^2) / sum(pca\_result$sdev^2)

num\_components <- which(cumulative\_variance >= 0.70)[1]

num\_components # 3 components

**#Question 1c**

# Extract the scores for the first 2 PCs

pc\_scores <- as.data.frame(pca\_result$x[, 1:2])

pc\_scores

# Extract the loadings for the first first 2 PCs

pc\_loadings <- as.data.frame(pca\_result$rotation[, 1:2])

pc\_loadings

# Correlations of standardized quantitative variables with the first two PCs

cor\_with\_pcs <- cor(scaled\_hitters, pc\_scores)

# Create a table

cor\_table <- as.table(cor\_with\_pcs)

cor\_table

# install.packages("remotes")

# remotes::install\_github("vqv/ggbiplot")

# ggbiplot(fit, labels =  rownames(hitters\_data))

require(ggbiplot)

ggbiplot(pca\_result)

**# Question 2a**

# Install and load required libraries

library(ISLR)

#Hitters dataset

data("Hitters")

Hitters <- na.omit(Hitters)

model <- lm(log(Hitters$Salary) ~ ., data = Hitters)

model.summary()

#no need to split data into test and training sets, since "all data is taken as training data"

predictions <- predict(model, newdata = Hitters)

# Compute the test MSE

test\_mse\_linear <- mean((log(Hitters$Salary) - predictions)^2)

# Print the results

cat("Test MSE (Linear model):", test\_mse\_linear, "\n")

**# Question 2b**

library(pls)

# Separate predictors and response

X <- model.matrix(Salary ~ ., data = Hitters)[, -1]

y <- log(Hitters$Salary)

# Perform Principal Component Regression with LOOCV

pcr\_model <- pcr(y ~ X, scale = TRUE, validation = "LOO")

summary(pcr\_model)

optimal\_components <- 16 #this is when the CV error is the smallest (0.6406)

pcr\_pred <- predict(pcr\_model, ncomp=16)

#test mse

test\_mse\_pcr <- mean((pcr\_pred-predictions)^2)

# Print the results

cat("Test MSE (PCR):", test\_mse\_pcr, "\n")

**# Question 2c**

pls\_model = plsr(y ~ X, data = Hitters, scale = TRUE, validation = "LOO")

summary(pls\_model)

#M=12

pls\_pred <- predict(pls\_model, ncomp=12)

#test mse

test\_mse\_pls <- mean((pls\_pred-predictions)^2)

# Print the results

cat("Test MSE (PLS):", test\_mse\_pls, "\n")

**# Question 2d**

library(glmnet)

# ridge regression model with LOOCV

ridge\_model <- cv.glmnet(x = X, y = y, alpha = 0)  # alpha = 0 for ridge regression

# Optimal lambda (penalty parameter) from LOOCV

optimal\_lambda <- ridge\_model$lambda.min

# Fit Ridge Regression model with the optimal penalty parameter

final\_ridge\_model <- glmnet(x = X, y = y, alpha = 0, lambda = optimal\_lambda)

# Make predictions on the test data

X\_test <- model.matrix(Salary ~ ., data = Hitters)[, -1]

y\_test <- log(Hitters$Salary)

predictions\_ridge <- predict(final\_ridge\_model, s = optimal\_lambda, newx = X\_test)

# Compute the test MSE

test\_mse\_ridge <- mean((y\_test - predictions\_ridge)^2)

cat("Test MSE (Ridge Regression):", test\_mse\_ridge, "\n")

**# Question 3a**

# Separate predictors and response

X <- model.matrix(Salary ~ ., data = Hitters)[, -1]

y <- log(Hitters$Salary)

# Fit a linear model

model <- lm(y ~ X)

model

# Extract coefficients and their names

coefficients <- coef(model)

predictor\_names <- names(coefficients)

# Identify the most important predictor (variable with the largest absolute coefficient)

most\_important\_predictor <- predictor\_names[which.max(abs(coefficients[-1]))]

# Print the results

cat("Most important predictor:", most\_important\_predictor, "\n")

**# Question 3b**

# Load necessary packages

install.packages("splines")

library(splines)

# Fit natural cubic spline regression model with LOOCV

spline\_model <- lm(y ~ ns(Hitters$CWalks, df = 7), data = Hitters)

spline\_model

# Make predictions on the test data

predictions\_spline <- predict(spline\_model, newdata = Hitters)

# Compute the test MSE

test\_mse\_spline <- mean((y - predictions\_spline)^2)

cat("Estimated test MSE (spline):", test\_mse\_spline, "\n")

**# Question 3c**

install.packages("gam")

library(gam)

# Fit a Generalized Additive Model (GAM)

gam\_model <- gam(y ~ s(AtBat, 3) + s(Hits,4) + s(HmRun, 5)

                 + s(Runs, 5) + s(RBI, 5) + s(Walks,4)

                 + s(Years, 4) + s(CAtBat, 4) + s(CHits, 4)

                 + s(CHmRun,4) + s(CRuns, 5) + s(CRBI, 5)

                 + s(CWalks, 3) + League + Division + s(PutOuts, 5)

                 + s(Assists, 5) + s(Errors, 5) + s(Salary, 5)

                 + NewLeague, data = Hitters)

# Summarize the model results

summary(gam\_model)

# find the most important predictor

name <- names(coef(gam\_model))

most\_important\_gam\_predictor <- name[which.max(abs(coefficients[-1]))]

most\_important\_gam\_predictor