**ISL ASSIGNMENT**

**Stat Learning: Features and Mode Selection Lab**

**Name: Sowmya Patlolla**

**PART A:**

**Q) Generate the list of selected features generated by the "forward" and "backward" selection methods for a new value of "nvmax" (number of max variables) between 9 and 18. What did you observe as a result of this change?**

**A) SOURCE CODE**

install.packages(“ISLR”)

library (ISLR)

install.packages(“leaps”)

library(leaps)

#For forward stepwise selection

regfit.fwd <- regsubsets(Salary ~ . , data = Hitters, nvmax = 16, method = “forward”)

summary(regfit.fwd)

#For backward selection

regfit.bwd <- regsubsets(Salary ~ ., data = Hitters, nvmax = 16, method = “backward”) summary(regfit.bwd)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The code specifies Salary ~ ., which fits a linear regression model with Salary as the response

variable and all other variables in the Hitters dataset as predictors. The acronym "all other variables" is denoted by the. The code can fit models with up to 16 predictors because the nvmax argument is set to 16, the total number of variables in the dataset.

**Q) Perform the Ridge Regression for the given line again using a new value (replace '50') of Lambda equal to the last 3 (non-zero) digits of your student ID.**

****

**How did this change impact the quality of your shrinkage method? What is the theoretical impact of selecting a larger value for Lambda for the L2 method?**

1. **SOURCE CODE:**

**#Ridge Regrression**

install.packages(“glmnet”)

library(glmnet)

# pre-process the Hitters

Hitters$Salary[is.na(Hitters$Salary)]<- mean(Hitters$Salary,na.rm=TRUE)

Y<-Hitters$Salary

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

Y <- Hitters$Salary

dim(x)

dim(y)

grid <- 10^seq(10, -2, length = 1000)

ridge.mod <- glmnet(x,y, alpha = 0, lambda = grid)

dim(coef(ridge.mod))

#last 3 non zero digits of student id 256

ridge.mod$lambda[256]

predict(ridge.mod, s=50, type = “coefficients”)[1:20, ]

**A computer screen shot of a computer code

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

The provided code uses the R glmnet library to fit a ridge regression model. The algorithm first preprocesses the Hitters dataset by separating the response variable (y) from the predictor variables (x). It does this by substituting the mean salary for any missing values in the Salary variable. The code fits the model with the glmnet function with alpha = 0 and generates a series of lambda values for the ridge regression. The ridge.mod object contains the final coefficient estimates for every value of lambda. The model's prediction accuracy and its capacity to estimate the true underlying coefficients determine how well the shrinkage technique performs. The degree to which the data had missing values and the correlation between Salary and the other predictor variables determine how much of an impact the code modification will have. The degree of shrinkage applied to the coefficient estimations in ridge regression is increased theoretically when a bigger value for lambda is used. The model's ability to minimize overfitting and enhance generalization can be increased by a drop in coefficient magnitude as lambda grows. But when lambda rises, so does the bias in the coefficient estimations, which may cause the genuine coefficients to be underestimated. The trade-off between bias and variance, which can be assessed using methods like cross-validation, must be taken into consideration while selecting a lambda. The trade-off between variance and bias must be considered when selecting a lambda, and this trade-off can be assessed using methods like cross-validation.

**Q) What changes did you observe? Capture a screenshot of your MSEP graph.**

**A) SOURCE CODE**

#PLS

install.packages(“pls”)

library(pls)

Set.seed(256)

Train <- sample(1:nrow(x), nrow(x) / 2)

Pls.fit <- plsr(Salary ~ ., data = Hitters, subset = train, scale =TRUE, validation = “CV”)

summary(pls.fit)

Par(mar=c(1,1,1,1))

Validationplot(pls.fit, val.type = “MSEP”)

**A computer code with black text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a graph

Description automatically generated**

**Part B**

**of your plot and console for wage vs age for Smooth Spline with df 16 (as given in the lab) and two new fits of df 9 and 22. (substitute the values as given in the assignment) What would you conclude from the results generated from the fits given by df 9,16,22? Include a screenshot\* of your results for each case. Be sure to make all elements of your workspace visible in your screenshot.**

1. **Source Code**

library (ISLR2)

attach (Wage)

install.packages(“splines”)

library(splines)

agelims <- range(age)

plot (age, wage, xlim = agelims, cex = .5, col = “darkgrey”)

fit1 <- smooth.spline(age, wage, df = 9)

fit2 <- smooth.spline(age, wage, df = 16)

fit3 <-smooth.spline(age, wage, df = 22)

lines (fit1, col = “red”, lwd = 2)

lines (fit2, col = “blue”, lwd=2)

lines (fit3, col = “green”, lwd = 2)

legend (“topright”, legend =c (“9 DF”, “16 DF”, “22 DF”), col = c(“red”, “blue”, “green”), lty = 1,

lwd = 2, cex = .8)

**A computer screen shot of a white background

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a graph

Description automatically generated**

According to the plots at DF = 9, 16, and 22. After 65, it is observed that the difference between the real line (BLUE) and the df line (RED) grows as the DF increases.