**Statistical Learning Lab**

Assignment - 4

Linear Regression Assignment

Anurag Singha | 22IM30007 

Show the code snippets and the corresponding output for the following:

Q1)Load the dataset “manufacturing.csv”. Display first few rows of the dataset. Take “Quality Rating” as response variable.

Ans:-

Code:-

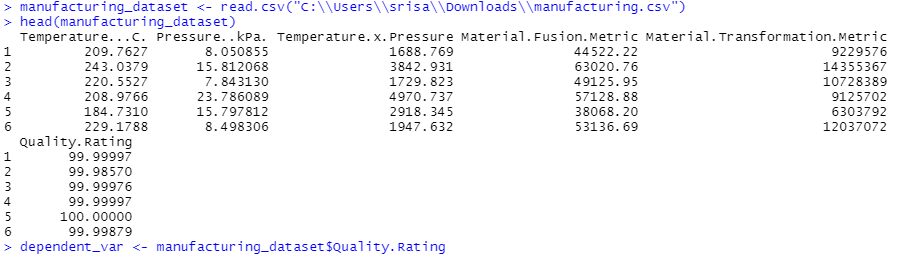
**setwd('C:/Users/USER/OneDrive - iitkgp.ac.in/Desktop/Stat Lab/Validation and Bootstrating')**

**data <- read.csv("manufacturing.csv")**

**head(data)**

**target\_variable <- data$Quality.Rating**

Output:-



Q2)Fit polynomial models between Quality ~ Temp. Vary the degree of polynomial on temperature from 1 to 5 (temp, temp^2, temp^3 etc.). Perform LOOCV, k-fold CV for k=5 and 10 and compare the cross-validation MSE errors for different degrees of polynomials. Create a table showing the CV errors for different degree of polynomials and for different CV techniques. Plot the results. Discuss which degree of polynomial is preferable.

Ans:-

Code:-

**library(boot)**

**colnames(data)**

**poly\_degrees <- 1:5**

**cv\_results <- data.frame(Degree = poly\_degrees, LOOCV\_Error = NA, K5\_Error = NA, K10\_Error = NA)**

**kfold\_cross\_validation <- function(model\_formula, dataset, num\_folds) {**

**total\_rows <- nrow(dataset)**

**split\_size <- floor(total\_rows / num\_folds)**

**mse\_accumulator <- 0**

**for (fold in 1:num\_folds) {**

**test\_idx <- ((fold - 1) \* split\_size + 1):(fold \* split\_size)**

**test\_data <- dataset[test\_idx, ]**

**train\_data <- dataset[-test\_idx, ]**

**fitted\_model <- lm(model\_formula, data = train\_data)**

**predicted\_vals <- predict(fitted\_model, newdata = test\_data)**

**actual\_vals <- test\_data$Quality.Rating**

**mse\_accumulator <- mse\_accumulator + mean((predicted\_vals - actual\_vals)^2)**

**}**

**return(mse\_accumulator / num\_folds)**

**}**

**for (deg in poly\_degrees) {**

**poly\_features <- paste("I(`Temperature...C.`^", 1:deg, ")", collapse = " + ")**

**model\_formula <- as.formula(paste("Quality.Rating ~", poly\_features))**

**print(paste("Training model for Degree", deg, ":", model\_formula))**

**loocv\_eval <- cv.glm(data, glm(model\_formula, data = data), K = nrow(data))**

**print(paste("LOOCV Error for Degree", deg, ":", loocv\_eval$delta[1]))**

**cv\_results$LOOCV\_Error[cv\_results$Degree == deg] <- loocv\_eval$delta[1]**

**k5\_eval <- kfold\_cross\_validation(model\_formula, data, 5)**

**k10\_eval <- kfold\_cross\_validation(model\_formula, data, 10)**

**print(paste("5-Fold CV Error for Degree", deg, ":", k5\_eval))**

**print(paste("10-Fold CV Error for Degree", deg, ":", k10\_eval))**

**cv\_results$K5\_Error[cv\_results$Degree == deg] <- k5\_eval**

**cv\_results$K10\_Error[cv\_results$Degree == deg] <- k10\_eval**

**}**

**print(cv\_results)**

**library(ggplot2)**

**cv\_results\_long <- reshape(cv\_results,**

**varying = c("LOOCV\_Error", "K5\_Error", "K10\_Error"),**

**v.names = "MSE",**

**timevar = "CV\_Method",**

**times = c("LOOCV", "K5CV", "K10CV"),**

**direction = "long")**

**ggplot(cv\_results\_long, aes(x = Degree, y = MSE, color = CV\_Method)) +**

**geom\_line() +**

**geom\_point() +**

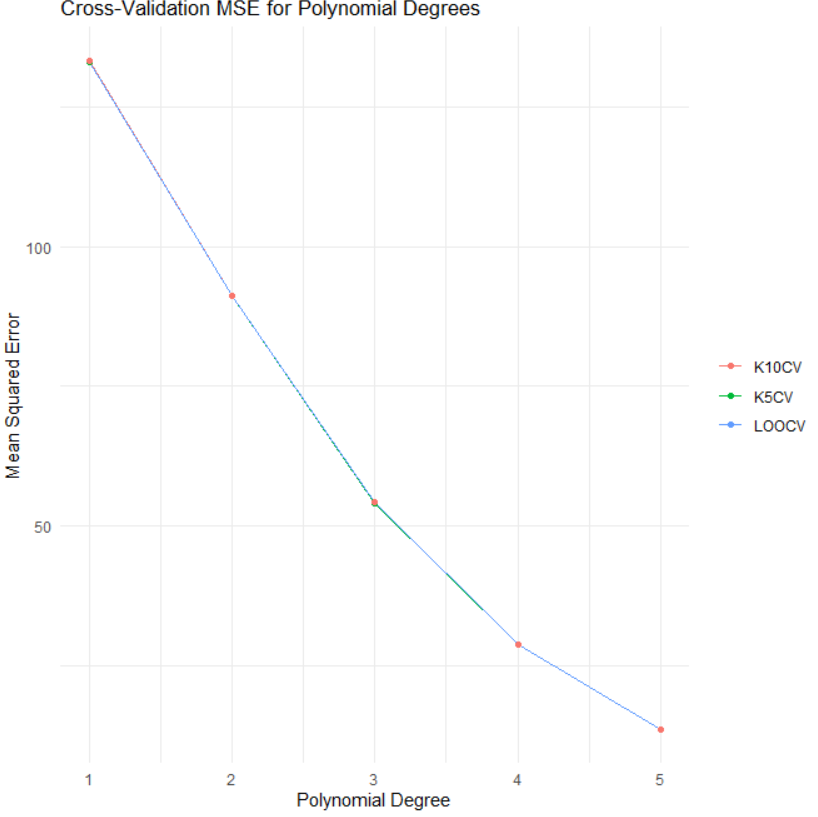
**theme\_minimal() +**

**labs(title = "Cross-Validation MSE for Polynomial Degrees",**

**x = "Polynomial Degree", y = "Mean Squared Error") +**

**theme(legend.title = element\_blank())**

Output:-



**Conclusion:Degrre 5 is Preferable as it has least MSE among them**

3.Perform the analysis in problem no. 2, but this time, fit linear models with different combination of X variables, without interaction. Discuss which model is most preferable based on the cross-validation results. Plot the results and on X-axis labels, provide the X-variable combinations used in the model, e.g. (temp, temp-press, temp-matfus, temp-matfus-mattr etc.)

Ans:-

Code:-

**feature\_combinations <- list(**

**c("Temperature...C."),**

**c("Temperature...C.", "Pressure..kPa."),**

**c("Temperature...C.", "Material.Fusion.Metric"),**

**c("Temperature...C.", "Material.Transformation.Metric"),**

**c("Temperature...C.", "Pressure..kPa.", "Material.Fusion.Metric"),**

**c("Temperature...C.", "Pressure..kPa.", "Material.Transformation.Metric"),**

**c("Temperature...C.", "Material.Fusion.Metric", "Material.Transformation.Metric"),**

**c("Temperature...C.", "Pressure..kPa.", "Material.Fusion.Metric", "Material.Transformation.Metric")**

**)**

**cv\_summary\_results <- data.frame(Combination = character(0), LOOCV\_Error = numeric(0), K5\_Error = numeric(0), K10\_Error = numeric(0))**

**for (features in feature\_combinations) {**

**model\_expression <- paste("Quality.Rating ~", paste(features, collapse = " + "))**

**model\_formula <- as.formula(model\_expression)**

**print(paste("Evaluating model with variables:", model\_expression))**

**loocv\_res <- cv.glm(data, glm(model\_formula, data = data), K = nrow(data))**

**loocv\_mse <- loocv\_res$delta[1]**

**k5\_mse <- kfold\_cross\_validation(model\_formula, data, 5)**

**k10\_mse <- kfold\_cross\_validation(model\_formula, data, 10)**

**cv\_summary\_results <- rbind(cv\_summary\_results, data.frame(Combination = model\_expression, LOOCV\_Error = loocv\_mse, K5\_Error = k5\_mse, K10\_Error = k10\_mse))**

**}**

**print(cv\_summary\_results)**

**library(reshape2)**

**cv\_results\_long <- reshape(cv\_summary\_results,**

**varying = c("LOOCV\_Error", "K5\_Error", "K10\_Error"),**

**v.names = "MSE",**

**timevar = "CV\_Method",**

**times = c("LOOCV", "K5CV", "K10CV"),**

**direction = "long")**

**ggplot(cv\_results\_long, aes(x = Combination, y = MSE, color = CV\_Method, group = CV\_Method)) +**

**geom\_point() +**

**geom\_line() +**

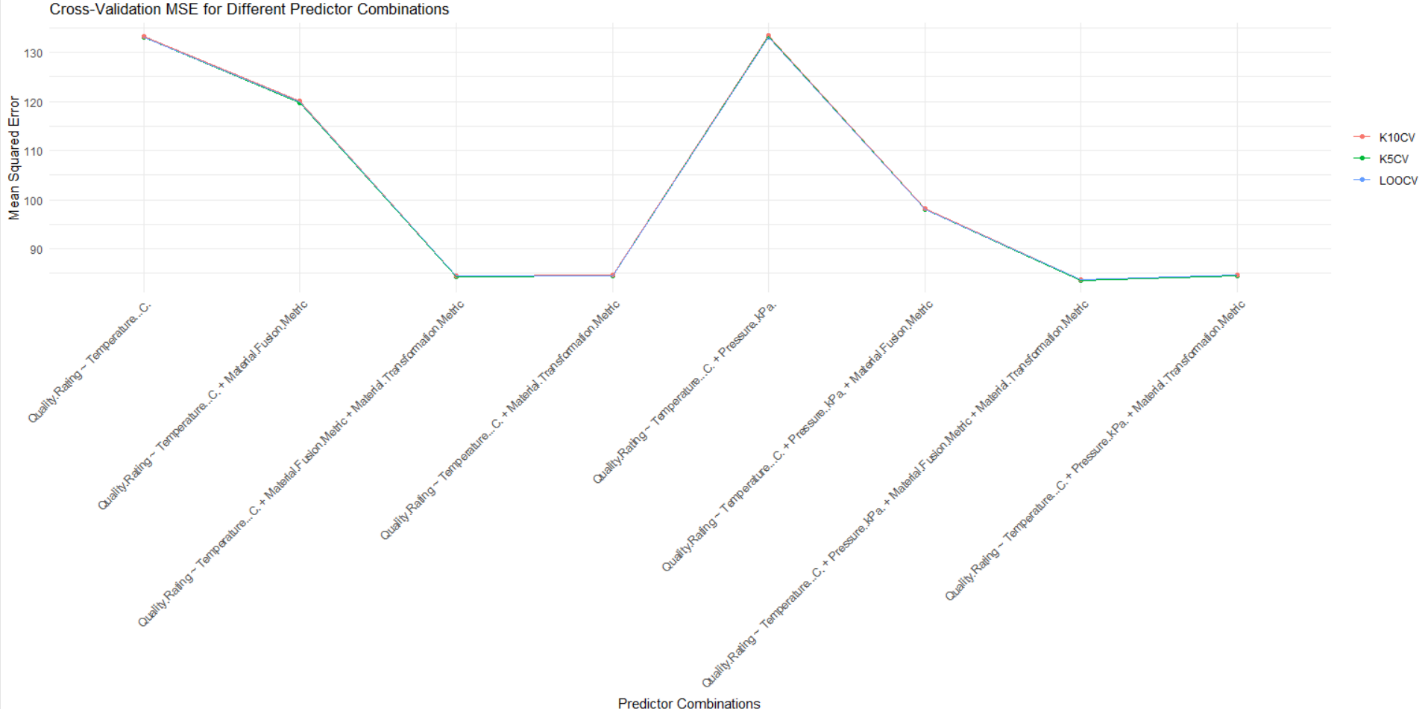
**theme\_minimal() +**

**labs(title = "Cross-Validation MSE for Different Predictor Combinations",**

**x = "Predictor Combinations", y = "Mean Squared Error") +**

**theme(axis.text.x = element\_text(angle = 45, hjust = 1), legend.title = element\_blank())**

Output:-



4)Generate 50 random numbers from Normal Distribution . Now create 100 bootstrap samples with 20 datapoints each, with replacement. Estimate the mean and variance of the population from the bootstrap samples.

Ans:-

Code:-

**set.seed(123)**

**random\_population <- rnorm(50, mean = 50, sd = sqrt(2))**

**bootstrap\_means <- numeric(100)**

**bootstrap\_variances <- numeric(100)**

**for (iteration in 1:100) {**

**sample\_subset <- sample(random\_population, size = 20, replace = TRUE)**

**bootstrap\_means[iteration] <- mean(sample\_subset)**

**bootstrap\_variances[iteration] <- var(sample\_subset)**

**}**

**estimated\_mean <- mean(bootstrap\_means)**

**estimated\_variance <- mean(bootstrap\_variances)**

**estimated\_mean**

**estimated\_variance**

**Output:**

> estimated\_mean

[1]50.05668

> estimated\_variance

[1] 1.733931

**Conclusion:**

Estimated Population Mean from Bootstrap Samples: 50.05668

Estimated Population Variance from Bootstrap Samples: 1.733931