Statistical Learning Lab

Assignment - 4

Cross-validation and Bootstrapping

NAME: SUNNY KUMAR , ROLL NO: 22IM10040

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

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

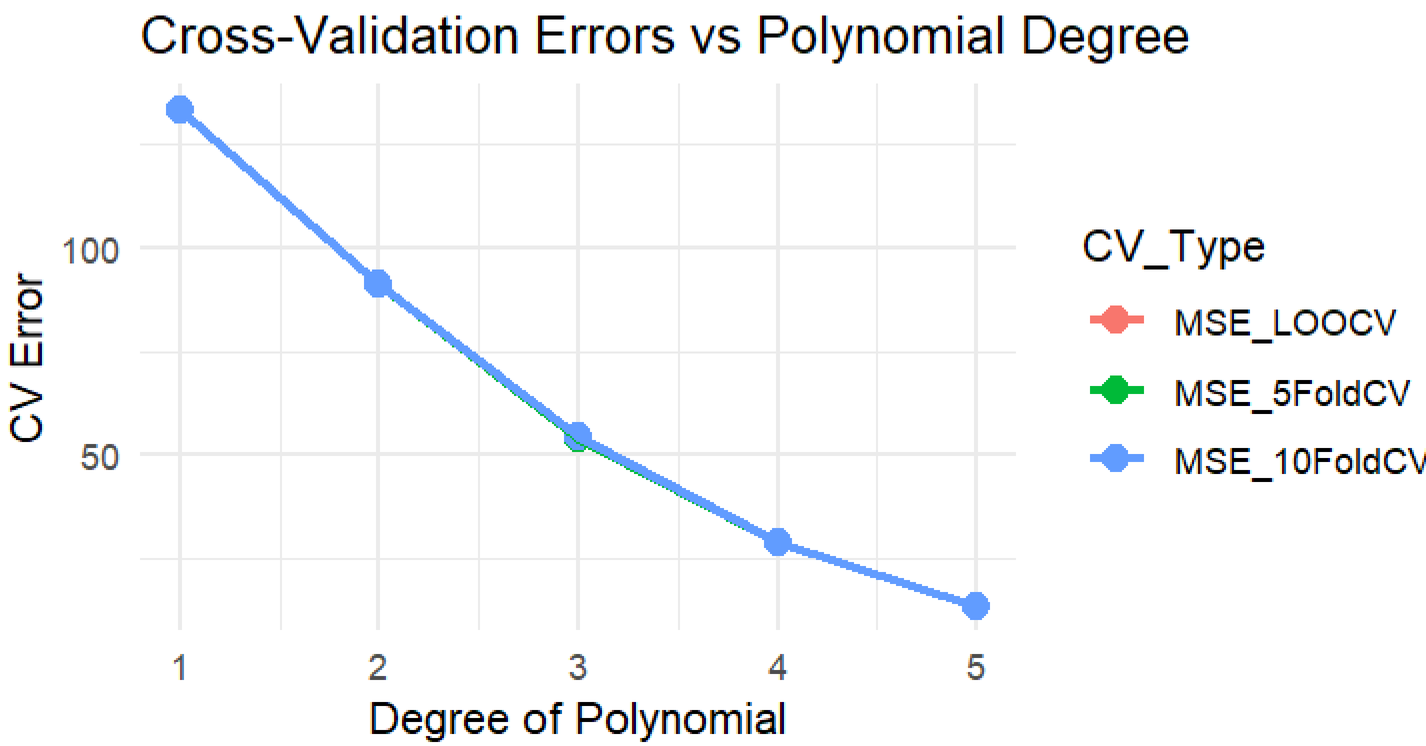
```
> df<- manufacturing
> head(df)
  Temperature...C. Pressure..kPa. Temperature.x.Pressure Material.Fusion.Metric Material.Transformation.Metric Quality.Rating
1         209.7627       8.050855               1688.769               44522.22                        9229576       99.99997
2         243.0379      15.812068               3842.931               63020.76                       14355367       99.98570
3         220.5527       7.843130               1729.823               49125.95                       10728389       99.99976
4         208.9766      23.786089               4970.737               57128.88                        9125702       99.99997
5         184.7310      15.797812               2918.345               38068.20                        6303792      100.00000
6         229.1788       8.498306               1947.632               53136.69                       12037072       99.99879
```

2. **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.**

```
> cv.error = rep(0,5)
> for(i in 1:5){
+   glm.fit = glm(Quality.Rating ~ poly(Temperature...C.,i),data = df)
+   cv.error[i] = cv.glm(df, glm.fit)$delta[1]
+ }
> cv.error
[1] 133.07880  91.19322  54.23607  28.78949  13.52725
> cv.error.5 = rep(0,5)
> for(i in 1:5){
+   glm.fit = glm(Quality.Rating ~ poly(Temperature...C.,i),data = df)
+   cv.error.5[i] = cv.glm(df, glm.fit, K =5)$delta[1]
+ }
> cv.error.5
[1] 133.35171  91.31538  54.12140  28.96107  13.54865
> cv.error.10 = rep(0,5)
> for(i in 1:5){
+   glm.fit = glm(Quality.Rating ~ poly(Temperature...C.,i),data = df)
+   cv.error.10[i] = cv.glm(df, glm.fit, K =10)$delta[1]
+ }
> cv.error.10
[1] 133.03142  91.18129  54.28092  28.91394  13.46476
> my_table <- data.frame(
+   MSE_LOOCV = cv.error,
+   MSE_5FoldCV = cv.error.5,
+   MSE_10FoldCV = cv.error.10
+ )
> print(my_table)
  MSE_LOOCV MSE_5FoldCV MSE_10FoldCV
1 133.07880   133.35171    133.03142
2  91.19322    91.31538     91.18129
3  54.23607    54.12140     54.28092
4  28.78949    28.96107     28.91394
5  13.52725    13.54865     13.46476
```

We can clearly see from here that for polynomial degree 1 and 2 , 10- fold Cross Validation gave the least error in these three models and for 3rd degree polynomial , 5-fold CV gives the least error and for 4degree polynomial , LOOCV gives the least error and for 5 degree polynomial , 10-fold Cross Validation gives the least error . Graphs are shown below.

From



From the graphs , we can clearly say that , Polynomial of degree 5 is favourable because it is giving the least error .

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.)

Code:

```
models <- list(
  "Temp" = "Quality.Rating ~ Temperature...C.",
  "Temp-Press" = "Quality.Rating ~ Temperature...C. + Pressure..kPa.",
  "Temp-MatFus" = "Quality.Rating ~ Temperature...C. + Material.Fusion.Metric",
  "Temp-MatFus-MatTrans" = "Quality.Rating ~ Temperature...C. + Material.Fusion.Metric + Material.Transformation.Metric",
  "Temp-Press-MatFus-MatTrans" = "Quality.Rating ~ Temperature...C. + Pressure..kPa. + Material.Fusion.Metric + Material.Transformation.Metric"
)

# Initialize error storage
cv_errors <- data.frame(Model = character(), CV_Error = numeric())

# Perform 5-Fold Cross-Validation
for (name in names(models)) {
  formula <- as.formula(models[[name]])  # Convert to formula
  glm.fit <- glm(formula, data = df)  # Fit model
  cv_result <- cv.glm(df, glm.fit, K = 5)  # Cross-validation
  cv_errors <- rbind(cv_errors, data.frame(Model = name, CV_Error = cv_result$delta[1]))
}

# Print results
print(cv_errors)
```

Output :

```
> print(cv_errors)
                          Model   CV_Error
1                          Temp 133.29321
2                    Temp-Press 133.00601
3                  Temp-MatFus 119.85305
4          Temp-MatFus-MatTrans  84.54390
5 Temp-Press-MatFus-MatTrans  83.76069
```

From the above graph we can conclude that the combination of Temperature...C. + Pressure..kPa. + Material.Fusion.Metric + Material.Transformation.Metric" gives the least error .

4. **Generate 50 random numbers from Normal Distribution $N(\mu = 50, \sigma^2 = 2)$. Now create 100 bootstrap samples with 20 datapoints each, with replacement. Estimate the mean and variance of the population from the bootstrap samples.**

```
> set.seed(3)
> population_data <- rnorm(50, mean = 50, sd = sqrt(2))
> # Bootstrap: 100 samples of size 20
> boot_means <- boot_vars <- numeric(100)
> for (i in 1:100) {
+   samp <- sample(data, 20, replace = TRUE)
+   boot_means[i] <- mean(samp)
+   boot_vars[i] <- var(samp)
+ }
There were 50 or more warnings (use warnings() to see the first 50)
> cat("Estimated Mean:", mean(boot_means), "\n")
Estimated Mean: NA
> cat("Estimated Variance:", mean(boot_vars), "\n")
Estimated Variance: 1.212761e+13
```