KJ 6.3

Group 3

2024-07-03

# 6.3 Chemical Manufacturing

**A chemical manufacturing process for a pharmaceutical product was discussed in Sect.1.4. In this problem, the objective is to understand the relationship between biological measurements of the raw materials (predictors),measurements of the manufacturing process (predictors), and the response of product yield. Biological predictors cannot be changed but can be used to assess the quality of the raw material before processing. On the other hand, manufacturing process predictors can be changed in the manufacturing pro- cess. Improving product yield by 1% will boost revenue by approximately one hundred thousand dollars per batch**

## 1.

**Start R and use these commands to load this data. The matrix processPredictors contains the 57 predictors (12 describing the input biological material and 45 describing the process predictors) for the 176 manufacturing runs. yield contains the percent yield for each run.**

data(ChemicalManufacturingProcess)  
dim(ChemicalManufacturingProcess)

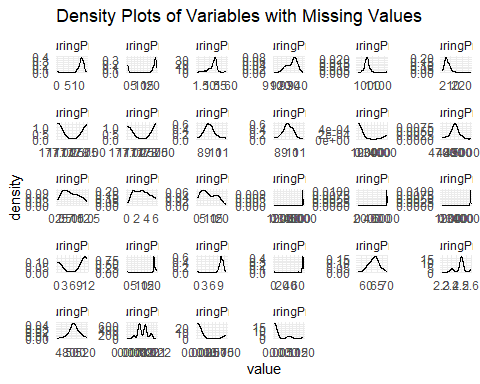
## [1] 176 58

## 2.

**A small percentage of cells in the predictor set contain missing values. Use an imputation function to fill in these missing values (e.g., see Sect. 3.8).**

In order to determine a valid imputation method, it is important to visualize the data. Different distributions require different methods of imputation

# Check the distributions of the columns with missing values  
cols\_with\_missing\_values <- colnames(ChemicalManufacturingProcess)[apply(ChemicalManufacturingProcess, 2, function(x) any(is.na(x)))]  
  
# Reshape the data to long format  
long\_data <- ChemicalManufacturingProcess %>%  
 select(all\_of(cols\_with\_missing\_values)) %>%  
 pivot\_longer(cols = everything(), names\_to = "variable", values\_to = "value")  
  
# Create density plots of those columns  
ggplot(long\_data, aes(x = value)) +  
 geom\_density() +  
 facet\_wrap(~ variable, scales = "free") +  
 labs(title = "Density Plots of Variables with Missing Values") +  
 theme\_minimal()



Given that each variable has a unique distribution that may respond differently to various imputation methods, I’ve decided to employ the k-nearest neighbors imputation method. This method is quite robust and can accommodate a wide range of distributions. If a specific column had a more substantial percentage of missing values, I would have chosen an imputation method better suited to that column’s distribution. However, as the missing values are minimal, I will proceed with a general and robust method.

imputed\_data <- preProcess(ChemicalManufacturingProcess, method = 'knnImpute')  
complete\_data <- predict(imputed\_data, ChemicalManufacturingProcess)  
# Check if there are any missing values  
any(is.na(complete\_data))

## [1] FALSE

There is no longer any miissing data in the dataset.

## 3.

**Split the data into a training and a test set, pre-process the data, and tune a model of your choice from this chapter. What is the optimal value of the performance metric?**

# Split the data into training and test sets  
set.seed(1125)  
splitIndex <- createDataPartition(complete\_data$Yield, p = 0.70, list = F)  
train\_data <- complete\_data[splitIndex, ]  
test\_data <- complete\_data[-splitIndex, ]

set.seed(1125)  
# Define training control with preprocessing and cross-validation  
train\_control <- trainControl(method = "cv", number = 10, preProcOptions = c("center", "scale", "nzv"))  
  
# Train the PLS model with preprocessing and cross-validation  
pls\_model <- train(Yield ~ .,   
 data = train\_data,   
 method = "pls",   
 metric = "RMSE",  
 trControl = train\_control)  
  
# Print the model results  
print(pls\_model)

## Partial Least Squares   
##   
## 124 samples  
## 57 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 112, 111, 111, 112, 112, 111, ...   
## Resampling results across tuning parameters:  
##   
## ncomp RMSE Rsquared MAE   
## 1 0.7577436 0.4572835 0.6160768  
## 2 0.6660160 0.5736380 0.5304066  
## 3 0.6539096 0.5906426 0.5256635  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was ncomp = 3.

The optimal value of the performance metric, RMSE, is *0.654*.

## 4.

**Predict the response for the test set. What is the value of the performance metric and how does this compare with the resampled performance metric on the training set?**

set.seed(1125)  
# Predict on the test set  
test\_predictions <- predict(pls\_model, newdata = test\_data)  
  
# Calculate RMSE on the test set  
test\_rmse <- sqrt(mean((test\_data$Yield - test\_predictions)^2))  
  
# Compare with resampled performance metric on training set  
resampled\_rmse <- min(pls\_model$results$RMSE)  
  
# Print the results  
cat("Performance Metric (RMSE) on Test Set:", test\_rmse, "\n")

## Performance Metric (RMSE) on Test Set: 0.6244794

cat("Resampled Performance Metric (RMSE) on Training Set:", resampled\_rmse, "\n")

## Resampled Performance Metric (RMSE) on Training Set: 0.6539096

The performance metric (RMSE) on the test set is *0.624* and the resampled performance metric (RMSE) on the training set is *0.654*. Overall, similar enough to suggest the model is generalizing well and stable.

## 5.

**Which predictors are most important in the model you have trained? Do either the biological or process predictors dominate the list?**

importance <- varImp(pls\_model)

##   
## Attaching package: 'pls'

## The following object is masked from 'package:caret':  
##   
## R2

## The following object is masked from 'package:stats':  
##   
## loadings

importance

## pls variable importance  
##   
## only 20 most important variables shown (out of 57)  
##   
## Overall  
## ManufacturingProcess09 100.00  
## ManufacturingProcess32 98.83  
## ManufacturingProcess13 90.75  
## ManufacturingProcess36 87.11  
## ManufacturingProcess17 86.74  
## BiologicalMaterial03 65.17  
## BiologicalMaterial02 65.13  
## ManufacturingProcess33 62.63  
## ManufacturingProcess11 62.43  
## BiologicalMaterial06 62.12  
## ManufacturingProcess12 61.60  
## BiologicalMaterial08 60.23  
## ManufacturingProcess06 59.43  
## BiologicalMaterial12 57.88  
## BiologicalMaterial11 53.09  
## BiologicalMaterial01 51.70  
## ManufacturingProcess28 49.25  
## BiologicalMaterial04 49.20  
## ManufacturingProcess02 45.13  
## ManufacturingProcess04 42.03

# Assess dominance of biological vs. process predictors  
biological\_predictors <- subset(importance$importance, grepl("Biological", rownames(importance$importance)))  
process\_predictors <- subset(importance$importance, grepl("Process", rownames(importance$importance)))  
  
# Print dominance assessment  
cat("Biological Predictors Dominance:", sum(biological\_predictors), "\n")

## Biological Predictors Dominance: 572.0013

cat("Process Predictors Dominance:", sum(process\_predictors), "\n")

## Process Predictors Dominance: 1364.141

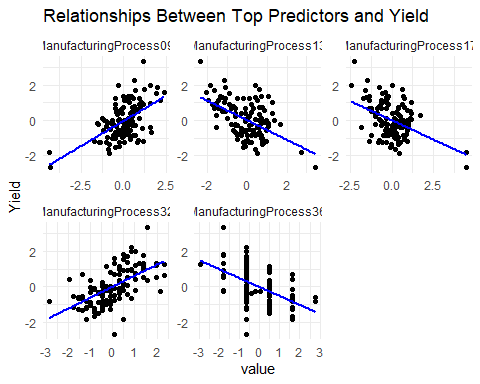
Viewing the top 20 predictors shows 12 process predictors and 8 biological predictors. Moreover, the first 5 predictors are all process predictors. This suggests that the process predictors dominate the list of important predictors in the model. Calculating the dominance of the predictors confirms this, with the process predictors having a dominance score of 1364 compared to the biological predictors’ dominance score of 572.

## 6.

**Explore the relationships between each of the top predictors and the response. How could this information be helpful in improving yield in future runs of the manufacturing process?**

# Get the top 5 predictors  
top\_5\_predictors <- rownames(importance$importance)[order(importance$importance$Overall, decreasing = TRUE)[1:5]]  
  
long\_data <- train\_data %>%  
 select(all\_of(top\_5\_predictors), Yield) %>%  
 pivot\_longer(cols = -Yield, names\_to = "variable", values\_to = "value")  
  
# Create scatter plots of the top predictors  
ggplot(long\_data, aes(x = value, y = Yield)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE, color = "blue") +  
 facet\_wrap(~ variable, scales = "free") +  
 labs(title = "Relationships Between Top Predictors and Yield") +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



# correlation matrix  
correlation\_matrix <- cor(train\_data[, c(top\_5\_predictors, "Yield")])  
cor\_yield <- correlation\_matrix["Yield", -length(correlation\_matrix)]  
cor\_yield

## ManufacturingProcess09 ManufacturingProcess32 ManufacturingProcess13   
## 0.5529263 0.5944541 -0.5186381   
## ManufacturingProcess36 ManufacturingProcess17 Yield   
## -0.5112481 -0.4200562 1.0000000

Since I do not know anything about the data or the processes, I am not looking at correlation between the predictors. I am only focusing on the relationship between the predictors and the response variable. Both the plots and the correltion matrix clearly reveal some of the top predictors hve a strong positive relationship with the response variable, while others have a negative relationship. Without knowing anything about the data, the obvious recommendation would be to increase the predictors with a positive relationship and decrease the predictors with a negative relationship. However, this is a very simplistic view and may not be the best course of action.