---
title: "Subset selection regularization dimensionality reduction"
author: "Semanti Ghosh"
date: "2025-02-27"
output: pdf_document

###Loading the data set

The directory is changed to the directory containing the data sets
The data is loaded and the first six lines are printed

```{r}
setwd("C:\\Study\\Semester_6\\Statistical_Learning_Lab\\assignment_5")
getwd()
data <- read.csv("Cellphone.csv")
head(data)
```

###Preliminary Analysis

We have to determine how different variables affect the target variable ("Price").
For that, first we have analysed through scatter plots

```{r}
library(ggplot2)

predictors <- setdiff(names(data), "Price")

for (var in predictors)
{
  p <- ggplot(data, aes(x = !!sym(var), y = Price)) +  # Use `!!sym(var)` inside aes()
    geom_point(color = "blue", alpha = 0.6) +
    labs(title = paste("Price vs", var), x = var, y = "Price") +
    theme_minimal()
  print(p)  # Print each plot
}
```

Then, boxplots
```{r}
for (var in predictors) {
  p <- ggplot(data, aes(x = "", y = !!sym(var))) +  # Use !!sym(var) for tidy evaluation
    geom_boxplot(fill = "red") +
    labs(title = paste("Box Plot of", var), x = "", y = var) +
    theme_minimal()
  print(p)
}
```

And although this has not been specifically mentioned in the question, for the sake of my own understanding, I will compute and print the correlation matrix and also the correlation heatmap
```{r}
install.packages("ggcorrplot")
library(ggcorrplot)
cor_matrix <- cor(data, use = "complete.obs")
print(cor_matrix)
ggcorrplot(cor_matrix, method = "square", lab = TRUE)
```


###Best Subset Selection

First installing the leaps package
```{r}
install.packages("leaps")
```

And then, the actual process of the best subset selection (brute force, trying every subset possible).Product ID has been dropped. To tackle overfitting, we have used \(R^2_{adj} \) and BIC as goodness of fit metrics. Based on those values, we will determine what the best fit was.Initially, the best fit model taking a certain number of predictors (1 to 12) is printed. After that, the best model is selected using highest \(R^2_{adj} \) statistic (that was the statistic decided on). The results have been printed as below.
```{r}
library(leaps)
data_subset <- data[, !(names(data) %in% c("Product_id"))]
best_subset <- regsubsets(Price ~ ., data = data_subset, nvmax = ncol(data_subset) - 1)
best_subset_summary <- summary(best_subset)
print(best_subset_summary)

adj_r2 <- summary(best_subset)$adjr2
best_model_adj_r2 <- which.max(adj_r2)
bic_values <- summary(best_subset)$bic
best_model_bic <- which.min(bic_values)

selected_vars <- summary(best_subset)$which[best_model_adj_r2, ]
selected_vars <- names(selected_vars[selected_vars == TRUE])
cat("Best model (by Adjusted R^2) has", best_model_adj_r2, "predictors\n")
cat("Best model predictors (by Adjusted R^2):", paste(selected_vars, collapse=", "), "\n")
```

###Creating a \(C_p \) plot

The \(C_p \) values have been extracted and then a graph has been plotted, keeping the number of variables or parameters on the X-axis and the \(C_p \) value on the Y-axis. The number of parameters for which the Cp value is be minimum has been highlighted in green.

```{r}
cp_values <- summary(best_subset)$cp
cp_df <- data.frame(Num_Variables = 1:length(cp_values),  Cp = cp_values)

best_cp_model <- which.min(cp_values)

ggplot(cp_df, aes(x = Num_Variables, y = Cp)) +
  geom_point(size = 3, color = "blue") +
  geom_line(color = "red") +
  annotate("point", x = best_cp_model, y = min(cp_values), color = "green", size = 4) +
  labs(title = "Mallows' Cp vs. Number of Variables",
       x = "Number of Variables",
       y = "Mallows' Cp") +
  theme_minimal()
```
And it has been observed that the Cp value is minimum when the number of parameters is 10.

###Plotting the best subset selection

For each value of n from 1 to 12 (X-axis), we are plotting the corresponding value of a statistic of the best model having that many parameter. The parameters considered in the plots are \(R^2_{adj} \), BIC, Mallow's \(C_p \) and RSS
```{r}
best_subset_summary <- summary(best_subset)
num_vars <- 1:length(best_subset_summary$cp)

cp_values <- best_subset_summary$cp
bic_values <- best_subset_summary$bic
```

```r
adj_r2_values <- best_subset_summary$adjr2
rss_values <- best_subset_summary$rss

# Adjusted R² Plot
ggplot(data.frame(num_vars, adj_r2_values), aes(x = num_vars, y = adj_r2_values)) +
  geom_point(color = "blue", size = 3) +
  geom_line(color = "red") +
  labs(title = "Adjusted R² vs. Number of Variables", x = "Number of Variables", y =
"Adjusted R²") +
  theme_minimal()

# Cp Plot
ggplot(data.frame(num_vars, cp_values), aes(x = num_vars, y = cp_values)) +
  geom_point(color = "blue", size = 3) +
  geom_line(color = "red") +
  labs(title = "Mallows' Cp vs. Number of Variables", x = "Number of Variables", y = "Cp")
+
  theme_minimal()

# BIC Plot
ggplot(data.frame(num_vars, bic_values), aes(x = num_vars, y = bic_values)) +
  geom_point(color = "blue", size = 3) +
  geom_line(color = "red") +
  labs(title = "BIC vs. Number of Variables", x = "Number of Variables", y = "BIC") +
  theme_minimal()

# RSS Plot
ggplot(data.frame(num_vars, rss_values), aes(x = num_vars, y = rss_values)) +
  geom_point(color = "blue", size = 3) +
  geom_line(color = "red") +
  labs(title = "RSS vs. Number of Variables", x = "Number of Variables", y = "RSS") +
  theme_minimal()

```
With the increase in variables, R^2_{adj} increases and then stabilises while the other three statistics decrease and eventtually stabilise. This happens because the model fits the training set better when there are greater number of parameters.

Principal Component Analysis

First, the pls library has got to be installed.
```{r}
install.packages("pls")
```

Fitting a PCR model(with cross-validation), and then printing it
```{r}
library(pls)
pcr_model <- pcr(Price ~ ., data = data, scale = TRUE, validation = "CV")
summary(pcr_model)
```

Now that the model has been fit, we have got to extract the variance explained for 5 and 7 components. After that, we have to convert the variance explained to percentage and then print the percentage explained in both cases. It has been printed below.
```{r}
expl_var <- pcr_model$Xvar / sum(pcr_model$Xvar)
cum_var <- cumsum(expl_var)
print(cum_var)
var_5 <- cum_var[5] * 100
var_7 <- cum_var[7] * 100
cat("Variance explained by 5 components:", var_5, "%\n")
```

```
cat("Variance explained by 7 components:", var_7, "%\n")
```
The percentage of variance explained by 5 components = 86.85746%
The percentage of variance explained by 7 components = 93.97701%

###Lasso Regression

Installing the glmnet model (because I dont think I have it)
```{r}
install.packages("glmnet")
```

Now carrying out the actual lasso regression
```{r}
library(glmnet)
X <- as.matrix(data[, !names(data) %in% c("Price", "Product_ID")])
Y <- data$Price
lasso_model <- glmnet(X, Y, alpha = 1)
set.seed(123)
cv_lasso <- cv.glmnet(X, Y, alpha = 1)
best_lambda <- cv_lasso$lambda.min
cat("Best Lambda:", best_lambda, "\n")
lasso_best <- glmnet(X, Y, alpha = 1, lambda = best_lambda)
lasso_coefs <- coef(lasso_best)
print(lasso_coefs)
```
So, on performing lasso regression, it is observed that the coefficients of the columns Product ID and sales come very close to zero. This is expected because the Product ID should not affect anything (it should have been dropped, but since it was not specifically mentioned in the question, I have not dropped it). Also, the weight of Sale column comes pretty close to zero. This happens due to lasso regression. Lasso regression is a much faster alternative to best subset selection (that uses brute force).