Assignment 5

Ben Abraham Biju

2025-03-04

1. Load the Dataset and Display the First Few Rows

```
library(readxl)

cellphone <- read_excel("C:/Users/benab/OneDrive - iitkgp.ac.in/Desktop/Sem 6/SL Lab/Lab 6/Phone-price/Cellphone.xlsx")

head(cellphone)
```

```
## # A tibble: 6 × 14
##   Product_id Price  Sale weight resoloution   ppi `cpu core` `cpu freq`
##        <dbl> <dbl> <dbl>  <dbl>       <dbl> <dbl>      <dbl>      <dbl>
## 1        203  2357    10   135          5.2   424          8       1.35
## 2        880  1749    10   125          4     233          2       1.3
## 3         40  1916    10   110          4.7   312          4       1.2
## 4         99  1315    11   118.         4     233          2       1.3
## 5        880  1749    11   125          4     233          2       1.3
## 6        947  2137    12   150          5.5   401          4       2.3
## # ℹ 6 more variables: `internal mem` <dbl>, ram <dbl>, RearCam <dbl>,
## #   Front_Cam <dbl>, battery <dbl>, thickness <dbl>
```

2. Preliminary Analysis and Exploratory Visualizations

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

Remove the product id from the dataset

```
# Exclude the Product_id variable if it exists:
df <- if ("Product_id" %in% colnames(cellphone)) {
  cellphone[, !(names(cellphone) %in% "Product_id")]
} else {
  cellphone
}
head(df)
```

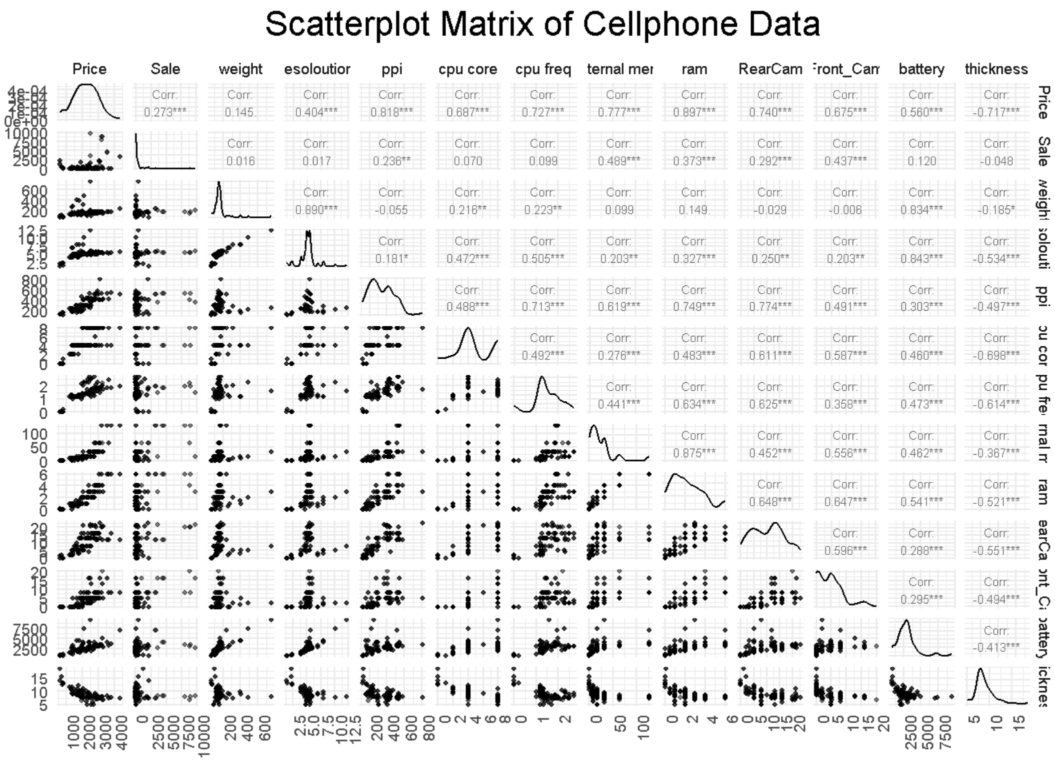
```
## # A tibble: 6 × 13
##   Price  Sale weight resoloution   ppi `cpu core` `cpu freq` `internal mem`
##   <dbl> <dbl>  <dbl>       <dbl> <dbl>      <dbl>      <dbl>          <dbl>
## 1  2357    10   135          5.2   424          8       1.35             16
## 2  1749    10   125          4     233          2       1.3               4
## 3  1916    10   110          4.7   312          4       1.2               8
## 4  1315    11   118.         4     233          2       1.3               4
## 5  1749    11   125          4     233          2       1.3               4
## 6  2137    12   150          5.5   401          4       2.3              16
## # ℹ 5 more variables: ram <dbl>, RearCam <dbl>, Front_Cam <dbl>, battery <dbl>,
## #   thickness <dbl>
```

Creating a pairwise scatter plot to see the interrelations among variables

```
options(repr.plot.width = 40, repr.plot.height = 40)

# Create a scatter plot matrix
ggpairs_plot <- ggpairs(df,
                        title = "Scatterplot Matrix of Cellphone Data",
                        progress = TRUE,
                        upper = list(continuous = wrap("cor", size = 2)),
                        lower = list(continuous = wrap("points", alpha = 0.5, size = 0.7)),
                        diag = list(continuous = wrap("densityDiag", alpha = 0.6))) +
  theme_minimal(base_size = 9) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        plot.title = element_text(hjust = 0.5, size = 16))

ggpairs_plot
```



3. Best Subset Selection

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.4.3
```

```
# Prepare the dataset by removing Product_id (if present):
data_bs <- if ("Product_id" %in% names(cellphone)) {
  cellphone[, !(names(cellphone) %in% "Product_id")]
} else {
  cellphone
}
# Perform best subset selection for predicting Price:
best_fit <- regsubsets(Price ~ ., data = data_bs, nvmax = ncol(data_bs) - 1)
best_summary <- summary(best_fit)

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

selected_vars <- summary(best_fit)$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")
```

```
## Best model (by Adjusted R^2) has 11 predictors
```

```
cat("Best model predictors (by Adjusted R^2):", paste(selected_vars, collapse=", "), "\n")
```

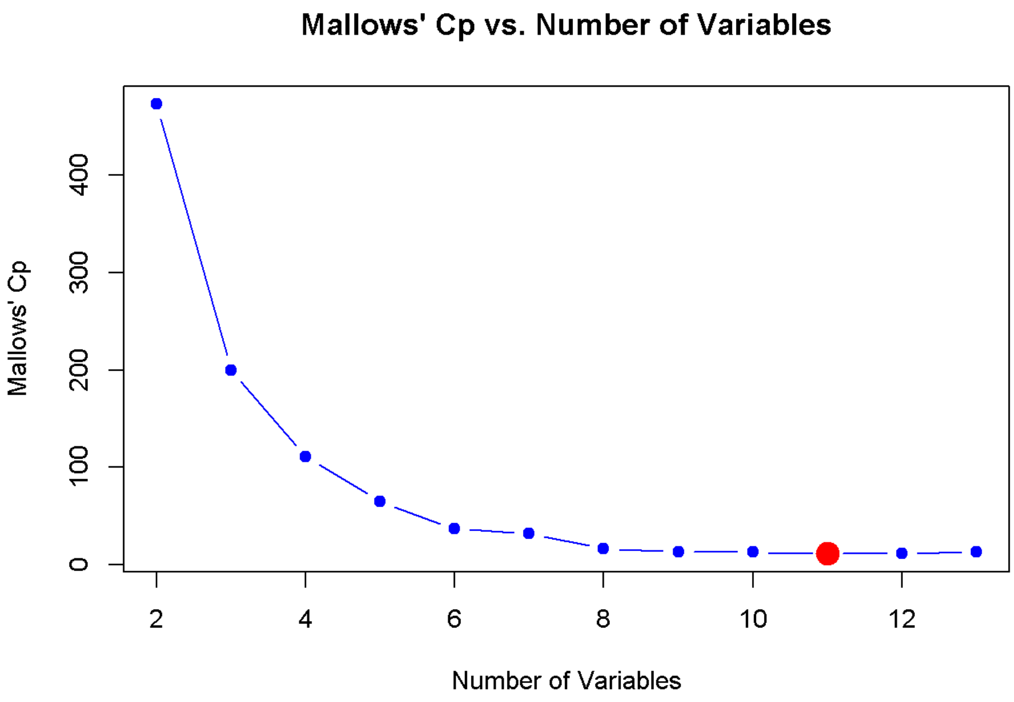
```
## Best model predictors (by Adjusted R^2): (Intercept), Sale, resoloution, ppi, `cpu core`, `cpu freq`, `internal mem`,
ram, RearCam, Front_Cam, battery, thickness
```

4. Create a plot with Cp on y-axis and number of variables on the x-axis. Determine the lowest Cp and report how many variables are included in the lowest Cp model

```
cp_values <- best_summary$cp
num_variables <- apply(best_summary$which, 1, sum) # Count number of selected variables for each model

# Find the model with the lowest Cp
best_cp_index <- which.min(cp_values)
best_num_variables <- num_variables[best_cp_index]

# Plot Cp vs. number of variables
plot(num_variables, cp_values, type = "b", pch = 19, col = "blue",
     xlab = "Number of Variables", ylab = "Mallows' Cp",
     main = "Mallows' Cp vs. Number of Variables")
points(best_num_variables, cp_values[best_cp_index], col = "red", pch = 19, cex = 2) # Highlight best model
```

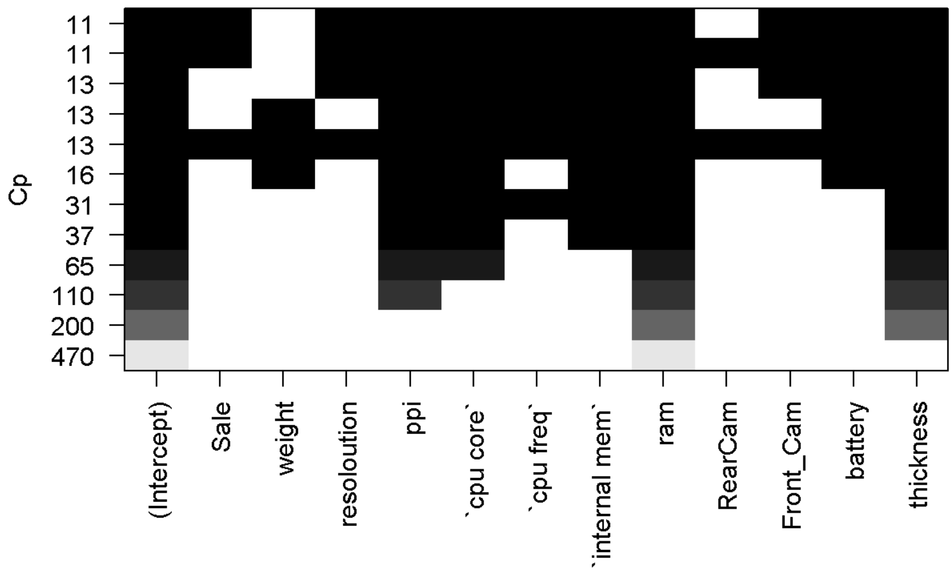



```
# Print results
cat("Lowest Cp Model includes", best_num_variables, "variables with Cp =", cp_values[best_cp_index], "\n")
```

```
## Lowest Cp Model includes 11 variables with Cp = 11.10326
```

5. Plot the best subset selection output and explain the plot.

```
par(mfrow = c(1, 1))  # Reset plotting area
plot(best_fit, scale = "Cp")  # Mallows' Cp as selection criteria
```



- This plot visualizes the model selection process using Mallows' Cp criterion, showing how different predictor variables contribute to models of varying complexity. The y-axis represents the Cp values, with lower values indicating better models that balance simplicity and accuracy. Each row corresponds to a specific model, and the black tiles indicate which predictors are included in that model, while white tiles indicate exclusion.
- The x-axis lists the predictor variables, such as **Sale, weight, ppi, ram,** etc. As we move down the plot (toward higher Cp values), more predictors are included, representing increasingly complex models.
- Key variables like **ram** and **ppi** appear in most models with low Cp values, suggesting they are significant predictors, while others like **Front_Cam** and **thickness** are included only in more complex models with higher Cp values, indicating they have less explanatory power. This plot helps identify the optimal subset of predictors for a regression model by balancing predictive performance and simplicity.

6. Use principal component regression on the same dataset with 5 components and 7 components. How much variability is explained by these two models?

```
library(pls)
```

```
## Warning: package 'pls' was built under R version 4.4.3
```

```
##
## Attaching package: 'pls'
```

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

```
set.seed(123)
data_pcr <- if ("Product_id" %in% names(cellphone)) {
  cellphone[, !(names(cellphone) %in% "Product_id")]
} else {
  cellphone
}
# Fit the PCR model with cross-validation
pcr_fit <- pcr(Price ~ ., data = data_pcr, scale = TRUE, validation = "CV")
summary(pcr_fit)
```

```
## Data:    X dimension: 161 12
##  Y dimension: 161 1
## Fit method: svdpc
## Number of components considered: 12
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           770.6    292.2    242.5    240.9    237.4    187.9    189.0
## adjCV        770.6    290.8    241.7    240.2    236.5    186.3    188.2
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
## CV       190.9    187.9    189.4     187.7     185.1     186.0
## adjCV    189.7    186.8    188.3     185.6     183.9     184.7
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X        48.24    67.71    78.61    85.55    90.02    93.40    95.59    97.37
## Price    87.40    90.91    91.15    91.92    94.50    94.52    95.01    95.13
##        9 comps  10 comps  11 comps  12 comps
## X        98.57     99.23     99.82    100.00
## Price    95.13     95.39     95.39     95.41
```

PCR reduces the data dimensionality. Here, the cumulative explained variance by the first 5 and 7 components is computed.

- The first 5 principal components together explain 90.02% of the total variability in the dataset, meaning they capture most of the important information in the original data.
- Adding two more components (for a total of 7) increases the explained variance to 95.59%, showing diminishing returns as more components are included.

```
# Extract the percentage variance explained by each principal component:
explained_var <- explvar(pcr_fit)
cum_explained_var <- cumsum(explained_var)
# Variability explained by the first 5 and first 7 components:
variance_5 <- cum_explained_var[5]
variance_7 <- cum_explained_var[7]
variance_5  # Variability explained by 5 components
```

```
##   Comp 5
## 90.01716
```

```
variance_7  # Variability explained by 7 components
```

```
##   Comp 7
## 95.58727
```

7. Perform Lasso on the model and explain the results.

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.4.3
```

```
## Loading required package: Matrix
```

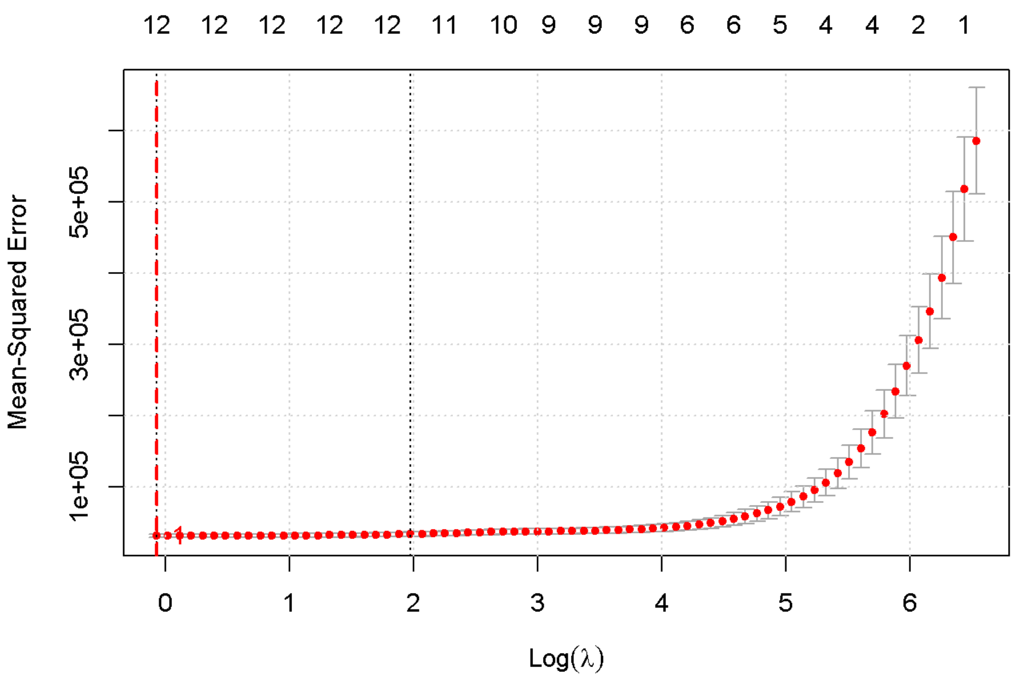
```
## Loaded glmnet 4.1-8
```

```
if ("Product_id" %in% names(cellphone)) {
  data_lasso <- cellphone[, !(names(cellphone) %in% "Product_id")]
} else {
  data_lasso <- cellphone
}

x <- model.matrix(Price ~ ., data = data_lasso)[, -1]
y <- data_lasso$Price

set.seed(123)
cv_lasso <- cv.glmnet(x, y, alpha = 1)
best_lambda <- cv_lasso$lambda.min

plot(cv_lasso, col.main = "blue", cex.main = 1)
grid()
abline(v = log(best_lambda), col = "red", lty = 2, lwd = 2)  # Marking best λ
text(log(best_lambda), min(cv_lasso$cvm), pos = 4, col = "red")
```




```
lasso_fit <- glmnet(x, y, alpha = 1, lambda = best_lambda)
lasso_coefs <- coef(lasso_fit)

# Print Selected Coefficients
print(lasso_coefs)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)    1705.74933498
## Sale             -0.02135994
## weight           -0.45523499
## resoloution     -66.71562740
## ppi               1.02291850
## `cpu core`       54.11494752
## `cpu freq`      125.48189555
## `internal mem`    6.19528538
## ram              95.91893217
## RearCam           4.68732450
## Front_Cam         8.58641496
## battery           0.12081951
## thickness       -71.71894132
```

- This plot shows the relationship between the **regularization parameter** (λ) and **mean squared error** (MSE).

- As λ **increases**, more coefficients **shrink to zero**, reducing the number of selected variables (displayed at the top).
- The **optimal λ**, marked by **vertical dotted lines**, balances model complexity and prediction error.
- This results in a **subset of important predictors**, reducing overfitting while maintaining predictive accuracy