Math 133 - Group Work 2

Pranav Jayakumar

January 24, 2025

Abstract

In this assignment, we compare the different predictors of sales (TV, radio, newspaper).

1 Data Analysis

1.1 Fitting Linear Models

We will first create three linear models for the three predictors of sales. We will use a 80-20 training-testing split. We initialize a seed of 123 to maintain reproducibility.

```
fit_linear_model <- function(y, x, raw_data) {

  # train test split
  n <- nrow(raw_data)
  trainIndex <- sample(n, round(0.8 * n, 0))
  train <- raw_data[trainIndex, ]
  test <- raw_data[-trainIndex, ]

  # construct formula
  formula <- as.formula(paste(y, "~", x))

  # fit model
  model <- lm(formula, data = train)

  # predict on testing data
  y_test <- test[[y]]
  y_hat <- predict(model, newdata = test)

  # analyze accuracy
  SSE <- sum((y_test - y_hat)^2)
  MSE <- SSE / nrow(test)
  RMSE <- sqrt(MSE)
  SST <- sum((y_test - mean(y_test))^2)
  R2 <- 1 - SSE / SST

  return(list(SSE = SSE, MSE = MSE, RMSE = RMSE, SST = SST, R2 = R2))
}
```

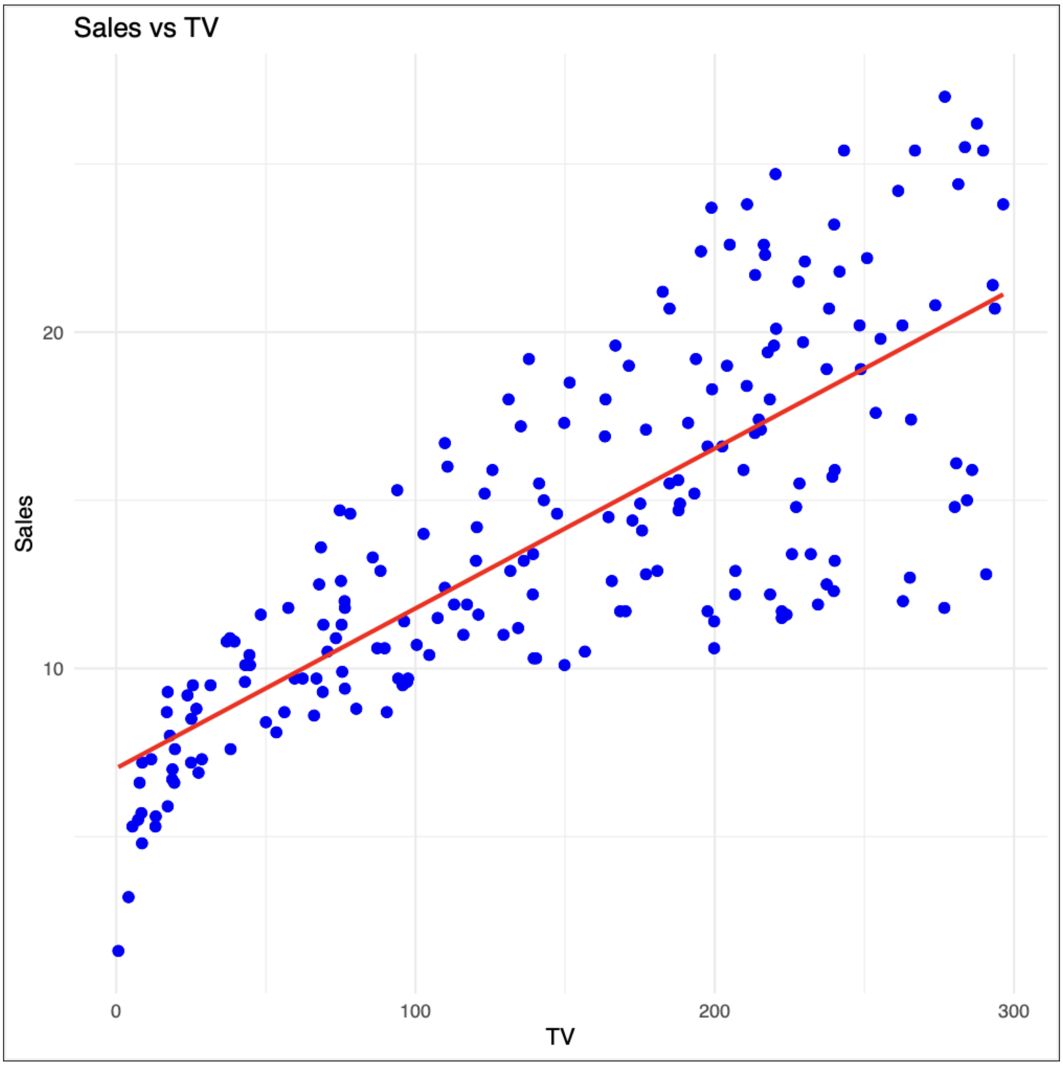
We observe that for the linear model `sales~TV`, $R^2 = 0.6053$. For the linear model `sales~radio`, we observe that $R^2 = 0.2692$. For the linear model `sales~newspaper`, we observe that $R^2 = -0.0693$.

1.2 Interpretation of Results

Based on the results from the R^2 tests, we determine `TV` to be the best predictor of `sales`.

1.3 Visualization

Below is a scatterplot denoting sales vs x where x is the `TV` predictor.



2 Complete R Code

```r
#!/usr/bin/env Rscript
library(ggplot2)

set.seed(123)

fit_linear_model <- function(y, x, raw_data) {

  # train test split
  n <- nrow(raw_data)
  trainIndex <- sample(n, round(0.8 * n, 0))
  train <- raw_data[trainIndex, ]
  test <- raw_data[-trainIndex, ]

  # construct formula
  formula <- as.formula(paste(y, "~", x))

  # fit model
  model <- lm(formula, data = train)

  # predict on testing data
  y_test <- test[[y]]
  y_hat <- predict(model, newdata = test)

  # analyze accuracy
  SSE <- sum((y_test - y_hat)^2)
  MSE <- SSE / nrow(test)
  RMSE <- sqrt(MSE)
  SST <- sum((y_test - mean(y_test))^2)
  R2 <- 1 - SSE / SST

  return(list(SSE = SSE, MSE = MSE, RMSE = RMSE, SST = SST, R2 = R2))
}

main <- function() {
    # Load data
  advertising <- read.csv("../../data/Advertising.csv")

  # Define predictors
  predictors <- c("TV", "radio", "newspaper")
  results <- list()

  # Iterate over predictors
  for (predictor in predictors) {
    if (!predictor %in% colnames(advertising)) {
      cat("\nWarning:␣Predictor", predictor, "not␣found␣in␣dataset.␣Skipping...\n")
      next
    }

    cat("\nLinear␣model␣for␣sales␣~", predictor, "\n")
    result <- fit_linear_model("sales", predictor, advertising)

    # Store results for later comparison
    results[[predictor]] <- result

    # Format and print results
    formatted_results <- lapply(result[1:5], function(x) format(round(x, 4), nsmall = 4))
    print(formatted_results)
  }

  # Determine the best predictor (highest R^2)
  best_predictor <- names(results)[which.max(sapply(results, function(r) r$R2))]
  cat("\nBest␣predictor␣based␣on␣R^2:", best_predictor, "\n")

  # Create scatterplot for the best predictor
  ggplot(advertising, aes_string(x = best_predictor, y = "sales")) +
    geom_point(color = "blue", size = 2) +
    geom_smooth(method = "lm", color = "red", se = FALSE) +
    ggtitle(paste("Sales␣vs", best_predictor)) +
    xlab(best_predictor) +
    ylab("Sales") +
    theme_minimal()
}

# Run the script if executed directly
if (interactive() || identical(Sys.getenv("R_SCRIPT"), "")) {
  main()
```