**SALES PREDICTION DATASET**

* **Sharon Blessy J V**

1. **What is the average amount spent on TV advertising in the dataset?**

**Ans.**

***Code Snippet:***

**tv\_amt\_avg = sd['TV'].mean()**

**print("Average amount spent on TV advertising: Rs.", tv\_amt\_avg)**

***Output:***

****

1. **What is the correlation between radio advertising expenditure and product sales?**

**Ans.**

***Code Snippet:***

**corr\_radio = sd['Radio'].corr(sd['Sales'])**

**print("Correlation between radio advertising expenditure and product sales: ", corr\_radio)**

***Output:***

This indicates a weak positive relationship(correlation).

1. **Which advertising medium has the highest impact on sales based on the dataset?**

**Ans.**

***Code Snippet:***

**corr\_tv = sd['TV'].corr(sd['Sales'])**

**corr\_radio = sd['Radio'].corr(sd['Sales'])**

**corr\_newspaper = sd['Newspaper'].corr(sd['Sales'])**

**if corr\_tv > corr\_radio and corr\_tv > corr\_newspaper:**

**print("TV advertising has the highest impact on sales.")**

**elif corr\_radio > corr\_tv and corr\_radio > corr\_newspaper:**

**print("Radio advertising has the highest impact on sales.")**

**else:**

**print("Newspaper advertising has the highest impact on sales.")**

***Output:***

1. **Plot a linear regression line that includes all variables (TV, Radio, Newspaper) to predict Sales, and visualize the model's predictions against the actual sales values.**

**Ans.**

***Code Snippet:***

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.linear\_model import LinearRegression**

**from sklearn.metrics import mean\_squared\_error, r2\_score**

**from sklearn.impute import SimpleImputer**

**X = sd[['TV', 'Radio', 'Newspaper']]**

**y = sd['Sales']**

**imputer = SimpleImputer(strategy='mean')**

**X = imputer.fit\_transform(X)**

**# Split the data**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# Initialize and train the model**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**y\_pred = model.predict(X\_test)**

**# Plotting actual vs predicted sales**

**plt.figure(figsize=(12, 6))**

**# Plot predicted vs actual sales**

**plt.subplot(1, 2, 1)**

**plt.scatter(y\_test, y\_pred, color='blue', alpha=0.5)**

**plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], color='red', linestyle='--')**

**plt.xlabel('Actual Sales')**

**plt.ylabel('Predicted Sales')**

**plt.title('Actual vs Predicted Sales')**

**# Plot residuals**

**plt.subplot(1, 2, 2)**

**residuals = y\_test - y\_pred**

**sns.histplot(residuals, kde=True, color='blue')**

**plt.xlabel('Residuals')**

**plt.title('Distribution of Residuals')**

**plt.tight\_layout()**

**plt.show()**

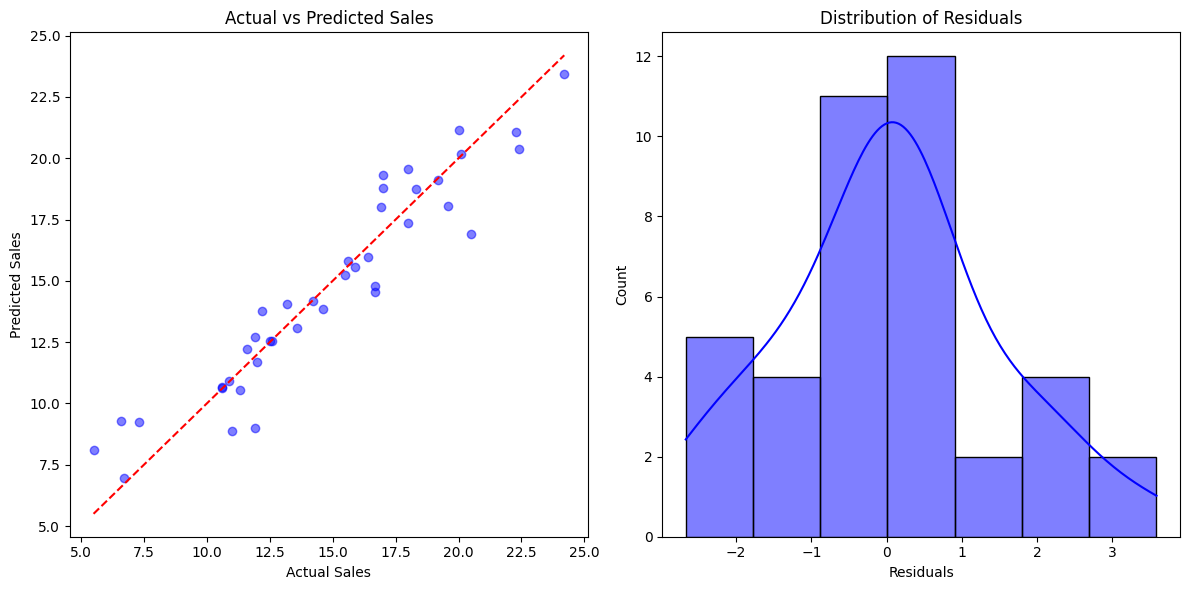
**# Print evaluation metrics**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**r2 = r2\_score(y\_test, y\_pred)**

**print(f"Mean Squared Error: {mse}")**

**print(f"R^2 Score: {r2}")**

***Output:***

****

1. **How would sales be predicted for a new set of advertising expenditures: $200 on TV, $40 on Radio, and $50 on Newspaper?**

**Ans.**

***Code Snippet:***

**X = sd[['TV', 'Radio', 'Newspaper']]**

**y = sd['Sales']**

**imputer = SimpleImputer(strategy='mean')**

**X = imputer.fit\_transform(X)**

**# Splitting data into training and testing sets**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# Fit the linear regression model**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**# TV: $200, Radio: $40, Newspaper: $50**

**new\_ad\_expenditures = [[200, 40, 50]]**

**# prediction for new values**

**predicted\_sales = model.predict(new\_ad\_expenditures)**

**print(f"Predicted Sales for $200 on TV, $40 on Radio, and $50 on Newspaper: {predicted\_sales[0]:.2f}")**

***Output:***

1. **How does the performance of the linear regression model change when the dataset is normalized?**

**Ans.**

***Code Snippet:***

**X = sd[['TV', 'Radio', 'Newspaper']]**

**y = sd['Sales']**

**# Split the data into training and testing sets (80% train, 20% test)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**imputer = SimpleImputer(strategy='mean')**

**X\_train = imputer.fit\_transform(X\_train)**

**X\_test = imputer.transform(X\_test)**

**# Train the model on the unnormalized data**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**y\_pred = model.predict(X\_test)**

**mse\_unnormalized = mean\_squared\_error(y\_test, y\_pred)**

**r2\_unnormalized = r2\_score(y\_test, y\_pred)**

**print(f"Unnormalized Data - MSE: {mse\_unnormalized:.2f}, R²: {r2\_unnormalized:.2f}")**

**# Normalize the feature data using StandardScaler**

**from sklearn.preprocessing import StandardScaler #Import the StandardScaler module**

**scaler = StandardScaler()**

**X\_train\_scaled = scaler.fit\_transform(X\_train)**

**X\_test\_scaled = scaler.transform(X\_test)**

**# Train the model on the normalized data**

**model\_normalized = LinearRegression()**

**model\_normalized.fit(X\_train\_scaled, y\_train)**

**y\_pred\_scaled = model\_normalized.predict(X\_test\_scaled)**

**# Performance on the normalized data**

**mse\_normalized = mean\_squared\_error(y\_test, y\_pred\_scaled)**

**r2\_normalized = r2\_score(y\_test, y\_pred\_scaled)**

**print(f"Normalized Data - MSE: {mse\_normalized:.2f}, R²: {r2\_normalized:.2f}")**

***Output:***

****

1. **What is the impact on the sales prediction when only radio and newspaper advertising expenditures are used as predictors?**

**Ans.**

***Output:***

**X = sd[['TV', 'Radio', 'Newspaper']]**

**y = sd['Sales']**

**X\_train\_all, X\_test\_all, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**imputer = SimpleImputer(strategy='mean')**

**X\_train\_all = imputer.fit\_transform(X\_train\_all)**

**X\_test\_all = imputer.transform(X\_test\_all)**

**model\_all = LinearRegression()**

**model\_all.fit(X\_train\_all, y\_train)**

**y\_pred\_all = model\_all.predict(X\_test\_all)**

**mse\_all = mean\_squared\_error(y\_test, y\_pred\_all)**

**r2\_all = r2\_score(y\_test, y\_pred\_all)**

**print(f"Model with TV, Radio, and Newspaper - MSE: {mse\_all:.2f}, R²: {r2\_all:.2f}")**

**X\_radio\_newspaper = sd[['Radio', 'Newspaper']]**

**X\_train\_rn, X\_test\_rn, y\_train\_rn, y\_test\_rn = train\_test\_split(X\_radio\_newspaper, y, test\_size=0.2, random\_state=42)**

**# Apply the imputer to X\_train\_rn and X\_test\_rn**

**X\_train\_rn = imputer.fit\_transform(X\_train\_rn)**

**X\_test\_rn = imputer.transform(X\_test\_rn)**

**model\_rn = LinearRegression()**

**model\_rn.fit(X\_train\_rn, y\_train\_rn)**

**y\_pred\_rn = model\_rn.predict(X\_test\_rn)**

**mse\_rn = mean\_squared\_error(y\_test\_rn, y\_pred\_rn)**

**r2\_rn = r2\_score(y\_test\_rn, y\_pred\_rn)**

**print(f"Model with Radio and Newspaper - MSE: {mse\_rn:.2f}, R²: {r2\_rn:.2f}")**

***Output:***