SALES PREDICTION USING PYTHON

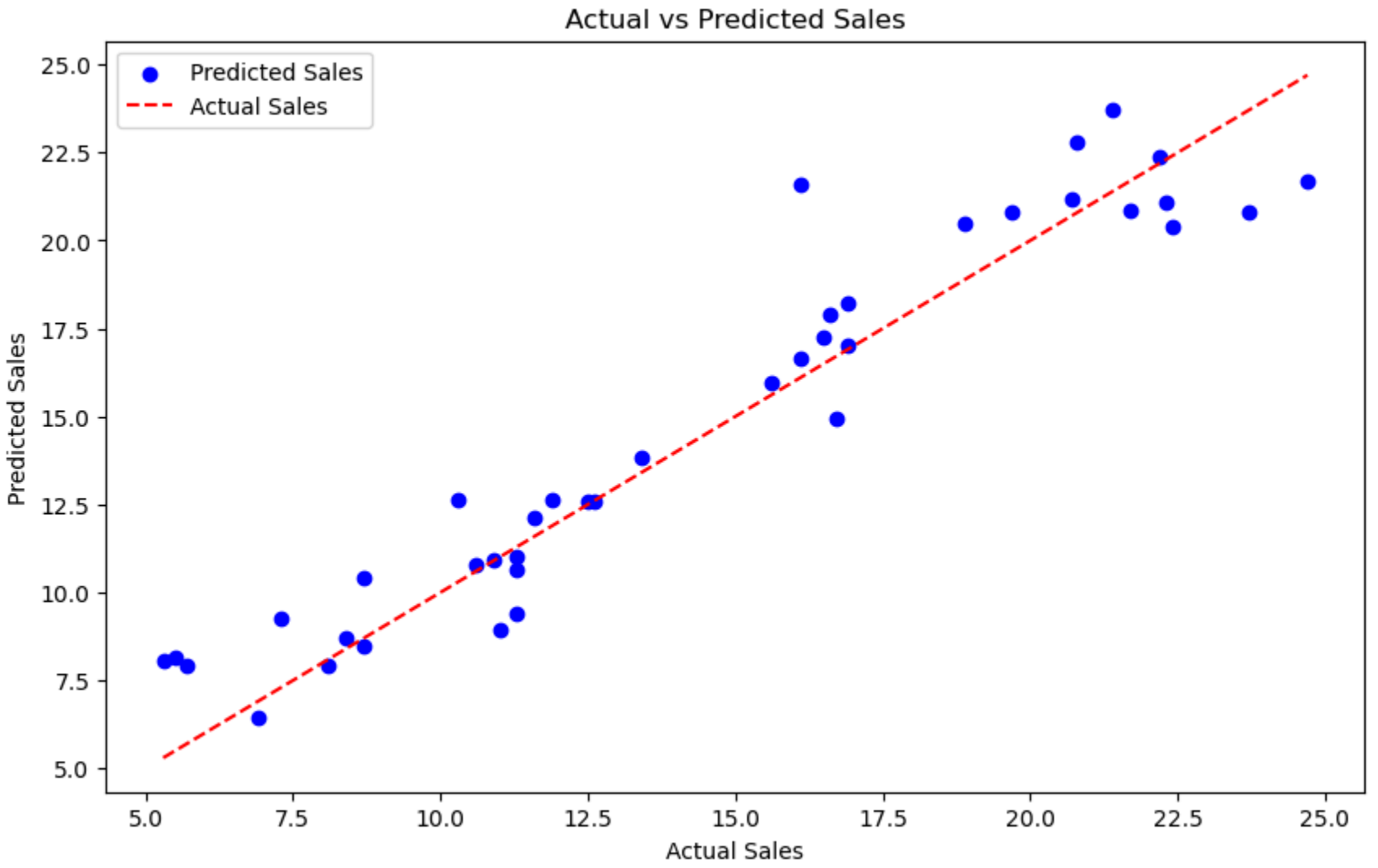
Sales prediction involves forecasting the amount of a product that customers will purchase, taking into account various factors such as advertising expenditure, target audience segmentation, and advertising platform selection. In businesses that offer products or services, the role of a Data Scientist is crucial for predicting future sales. They utilize machine learning techniques in Python to analyze and interpret data, allowing them to make informed decisions regarding advertising costs. By leveraging these predictions, businesses can optimize their advertising strategies and maximize sales potential. Let'

s embark on

the journey of sales prediction using machine learning in Python.

In [7]:
```python
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

try:
    # Load the dataset (replace 'advertising[1].csv' with your dataset file path)
    data = pd.read_csv('advertising[1].csv')

    # Print existing column names in the dataset
    print("Existing columns in the dataset:")
    print(data.columns.tolist())

    # Check if required columns exist in the dataset
    required_columns = ['TV', 'Radio', 'Newspaper', 'Sales']
    for col in required_columns:
        if col not in data.columns:
            raise ValueError(f"Missing column: {col}")

    # Data preprocessing
    # Selecting relevant features and target variable
    X = data[['TV', 'Radio', 'Newspaper']]
    y = data['Sales']

    # Splitting the dataset 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)

    # Initialize Linear Regression model
    model = LinearRegression()

    # Training the model
    model.fit(X_train, y_train)

    # Making predictions
    y_pred = model.predict(X_test)

    # Evaluate the model
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    # Printing the evaluation metrics
    print(f"Mean Absolute Error: {mae}")
    print(f"Mean Squared Error: {mse}")
    print(f"R-squared Score: {r2}")

    # Visualization of actual vs predicted sales
    plt.figure(figsize=(10, 6))
    plt.scatter(y_test, y_pred, color='blue', label='Predicted Sales')
    plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--', label='Actual Sales')
    plt.title('Actual vs Predicted Sales')
    plt.xlabel('Actual Sales')
    plt.ylabel('Predicted Sales')
    plt.legend()
    plt.show()

except FileNotFoundError:
    print("Error: File not found. Please check the file path.")
except ValueError as ve:
    print(f"Error: {ve}")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

```
Existing columns in the dataset:
['TV', 'Radio', 'Newspaper', 'Sales']
Mean Absolute Error: 1.274826210954934
Mean Squared Error: 2.907756910271091
R-squared Score: 0.9059011844150826
```



DOCMENATION:

This Python script predicts sales using a Linear Regression model. It utilizes pandas for data handling, matplotlib for plotting, and scikit-learn for machine learning.

Sections Breakdown Imports:

Essential libraries for data handling, plotting, and ML are imported. Data Loading:

Reads the advertising data from a CSV file. Data Prep:

Chooses features (TV, Radio, Newspaper) and target (Sales). Splits data into training and test sets. Modeling:

Sets up and trains the Linear Regression model. Predictions:

Predicts sales using the trained model on test data. Evaluation:

Computes MAE, MSE, and R-squared to assess model performance. Visualization:

Plots actual vs. predicted sales. Error Handling Handles file issues, missing columns, and unexpected errors. Usage Guide Update the file path to your dataset. Run the script to load data, train the model, and generate predictions. Dependencies pandas matplotlib scikit-learn

In [ ]: