Task

Analyze the provided advertising data from "/content/Advertising.csv" to predict future sales, identify key factors influencing sales, and provide actionable insights for marketing strategies.

Load the data

Subtask:

Load the advertising data from "/content/Advertising.csv" into a pandas DataFrame.

Reasoning: Import pandas and load the CSV file into a DataFrame, then display the head to verify the data loaded correctly.

```
import pandas as pd

df = pd.read_csv('/content/Advertising.csv')
display(df.head())
```

{"summary":"{\n \"name\": \"display(df\",\n \"rows\": 5,\n \"fields\": [\n   {\n

Explore and preprocess data

Subtask:

Examine the data to understand its structure, check for missing values, and prepare it for modeling by handling missing values and converting categorical features into numerical representations if needed.

Reasoning: Display the data types, check for missing values, and remove the 'Unnamed: 0' column to prepare the data for modeling.

```
display(df.info())
display(df.isnull().sum())
df = df.drop('Unnamed: 0', axis=1)
display(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  200 non-null    int64
 1   TV          200 non-null    float64
 2   Radio       200 non-null    float64
 3   Newspaper   200 non-null    float64
 4   Sales       200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

```
None
```

```
Unnamed: 0    0
TV            0
Radio         0
```

```
Newspaper     0
Sales         0
dtype: int64
```

{"summary":"{\n \"name\": \"display(df\",\n \"rows\": 5,\n \"fields\": [\n   {\n

Feature selection

Subtask:

Identify the most relevant features for predicting sales. This may involve analyzing correlations or using feature importance techniques.

Reasoning: Calculate and display the correlation matrix to identify the relationship between features and sales.

```
correlation_matrix = df.corr()
display(correlation_matrix)
```

{"summary":"{\n \"name\": \"correlation_matrix\",\n \"rows\": 4,\n \"fields\": [\

Reasoning: Analyze the correlation coefficients to understand the relationship between advertising features and sales.

```
print("Correlation between TV and Sales:", correlation_matrix.loc['TV', 'Sales'])
print("Correlation between Radio and Sales:", correlation_matrix.loc['Radio', 'Sales
print("Correlation between Newspaper and Sales:", correlation_matrix.loc['Newspaper'
```

```
Correlation between TV and Sales: 0.7822244248616067
Correlation between Radio and Sales: 0.5762225745710555
Correlation between Newspaper and Sales: 0.22829902637616545
```

Split data

Subtask:

Divide the dataset into training and testing sets to evaluate the model's performance.

Reasoning: Import the necessary function and split the data into training and testing sets.

```
from sklearn.model_selection import train_test_split

X = df[['TV', 'Radio']]
y = df['Sales']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat

print("Shape of X_train:", X_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (160, 2)
Shape of X_test: (40, 2)
Shape of y_train: (160,)
Shape of y_test: (40,)
```

Model selection and training

Subtask:

Choose appropriate regression models (e.g., Linear Regression, Decision Tree Regression, etc.) and train them on the training data.

Reasoning: Import the LinearRegression model, instantiate it, and fit it to the training data.

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

Model evaluation

Subtask:

Evaluate the trained models using appropriate metrics (e.g., Mean Absolute Error, Mean Squared Error, R-squared) to assess their performance on the testing data.

Reasoning: Import the necessary evaluation metrics, make predictions using the trained model, calculate the evaluation metrics, and print the results.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("R-squared (R2) Score:", r2)
```

```
Mean Absolute Error (MAE): 1.4443023138227578
Mean Squared Error (MSE): 3.137948009068354
R-squared (R2) Score: 0.9005833101920356
```

Predict sales

Subtask:

Use the best-performing model to predict sales based on new advertising data.

Reasoning: Define new advertising data and use the trained model to predict sales.

```
new_advertising_data = pd.DataFrame({
    'TV': [250.0, 100.0, 50.0],
    'Radio': [20.0, 30.0, 10.0]
})

predicted_sales = model.predict(new_advertising_data)
```

```
print("Predicted sales for new advertising data:")
print(predicted_sales)
```

```
Predicted sales for new advertising data:
[18.02367603 13.22110338  7.17134299]
```

Analyze advertising impact

Subtask:

Analyze the model's coefficients or feature importance to understand how changes in advertising spend impact sales outcomes.

Reasoning: Access and display the model's coefficients and intercept, associating coefficients with feature names, to understand their impact on sales.

```
import pandas as pd

coefficients = model.coef_
intercept = model.intercept_

feature_names = X_train.columns
coefficients_series = pd.Series(coefficients, index=feature_names)

print("Model Coefficients:")
display(coefficients_series)
print("\nModel Intercept:")
print(intercept)

print("\nInterpretation:")
print("The coefficients represent the change in sales for a one-unit increase in the
print("For example, a one-unit increase in TV advertising spend is associated with a
print("The intercept represents the baseline sales when both TV and Radio advertisin
```

```
Model Coefficients:

TV       0.044728
Radio    0.190667
dtype: float64

Model Intercept:
3.028255250783305

Interpretation:
The coefficients represent the change in sales for a one-unit increase in the corres
For example, a one-unit increase in TV advertising spend is associated with a change
The intercept represents the baseline sales when both TV and Radio advertising spend
```

Deliver actionable insights

Subtask:

Summarize the findings and provide actionable insights for business marketing strategies based on the analysis.

Reasoning: Summarize the findings and provide actionable insights based on the correlation analysis and linear regression model coefficients.

```
print("Summary of Findings:")
print("Correlation Analysis:")
print(f"- TV advertising has the strongest positive correlation with sales ({correla
print(f"- Radio advertising has a moderate positive correlation with sales ({correla
print(f"- Newspaper advertising has a weak positive correlation with sales ({correla
print("\nLinear Regression Model:")
print(f"- The model explains approximately {r2:.2f}% of the variance in sales.")
print(f"- For every unit increase in TV advertising spend, sales are estimated to in
print(f"- For every unit increase in Radio advertising spend, sales are estimated to
print(f"- The baseline sales when both TV and Radio spend are zero is estimated to b

print("\nActionable Insights for Marketing Strategies:")
print("1. Prioritize TV Advertising: Given its strong correlation and positive coeff
print("2. Leverage Radio Advertising: Radio also shows a positive impact on sales, a
print("3. Re-evaluate Newspaper Advertising: The analysis indicates that Newspaper a
print("4. Data-Driven Budget Allocation: Continuously monitor the performance of TV

print("\nLimitations and Future Steps:")
print("1. Exclusion of Newspaper: The current model does not include Newspaper adver
print("2. Linear Relationship Assumption: The linear regression model assumes a line
print("3. Interaction Effects: The model assumes that the impact of TV and Radio adv
print("4. External Factors: The model only considers advertising spend. Other extern
print("5. Causal Inference: While the model shows correlation and association, it do
```

```
Summary of Findings:
Correlation Analysis:
- TV advertising has the strongest positive correlation with sales (0.78).
- Radio advertising has a moderate positive correlation with sales (0.58).
- Newspaper advertising has a weak positive correlation with sales (0.23).

Linear Regression Model:
- The model explains approximately 0.90% of the variance in sales.
- For every unit increase in TV advertising spend, sales are estimated to increase b
- For every unit increase in Radio advertising spend, sales are estimated to increas
- The baseline sales when both TV and Radio spend are zero is estimated to be 3.028

Actionable Insights for Marketing Strategies:
1. Prioritize TV Advertising: Given its strong correlation and positive coefficient,
2. Leverage Radio Advertising: Radio also shows a positive impact on sales, and its
3. Re-evaluate Newspaper Advertising: The analysis indicates that Newspaper advertis
4. Data-Driven Budget Allocation: Continuously monitor the performance of TV and Rad

Limitations and Future Steps:
1. Exclusion of Newspaper: The current model does not include Newspaper advertising.
2. Linear Relationship Assumption: The linear regression model assumes a linear rela
3. Interaction Effects: The model assumes that the impact of TV and Radio advertisin
4. External Factors: The model only considers advertising spend. Other external fact
5. Causal Inference: While the model shows correlation and association, it doesn't d
```

Summary:

Data Analysis Key Findings

- TV advertising has the strongest positive correlation with sales (approximately 0.78).
- Radio advertising has a moderate positive correlation with sales (approximately 0.58).
- Newspaper advertising has a weak positive correlation with sales (approximately 0.23).
- A Linear Regression model using TV and Radio advertising explained approximately 90% of the variance in sales (R-squared = 0.90).
- For every unit increase in TV advertising spend, sales are estimated to increase by approximately 0.045 units, holding Radio spend constant.
- For every unit increase in Radio advertising spend, sales are estimated to increase by approximately 0.191 units, holding TV spend constant.
- The estimated baseline sales when both TV and Radio spend are zero is approximately 3.028 units.

Insights or Next Steps

- Prioritize investment in TV and Radio advertising due to their significant positive impact on sales, and consider re-evaluating or reducing spend on Newspaper advertising based on its weak correlation.
- Explore non-linear models and the potential interaction effects between TV and Radio advertising to gain a more nuanced understanding of their combined impact on sales, and consider incorporating external factors for a more robust predictive model.

```
!pip install -q gradio
```

```
import gradio as gr
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load the data
df = pd.read_csv('/content/Advertising.csv')

# Preprocess data
df = df.drop('Unnamed: 0', axis=1)

# Feature Selection
X = df[['TV', 'Radio']]
y = df['Sales']

# Split Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_st

# Model Selection and Training
model = LinearRegression()
model.fit(X_train, y_train)

# Model Evaluation (optional for the Gradio interface)
y_pred = model.predict(X_test)
```

```python
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

def predict_sales(tv_spend, radio_spend):
    """
    Predicts sales based on TV and Radio advertising spend.
    """
    new_data = pd.DataFrame({'TV': [tv_spend], 'Radio': [radio_spend]})
    predicted_sales = model.predict(new_data)
    return predicted_sales[0]

# Create Gradio interface
interface = gr.Interface(
    fn=predict_sales,
    inputs=[
        gr.Number(label="TV Advertising Spend"),
        gr.Number(label="Radio Advertising Spend")
    ],
    outputs=gr.Number(label="Predicted Sales"),
    title="Sales Prediction based on Advertising Spend",
    description="Enter the amount spent on TV and Radio advertising to predict sal
)

# Launch the interface
interface.launch(share=True, debug=True)
```

```
Colab notebook detected. This cell will run indefinitely so that you can see error
* Running on public URL: https://9dc8d6a70a02d24f54.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, ru
<IPython.core.display.HTML object>
```