

## Import Libraries

```
# Import necessary libraries
import pandas as pd
import numpy as np

import plotly.express as px
import plotly.graph_objects as go

#%%matplotlib inline

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

## Data Collection

```
# Import required libraries
import pandas as pd
from google.colab import files

# Upload CSV file from laptop
uploaded = files.upload()

# Uploaded data is in dictionary form; convert keys to a list and get the first filename
file_name = list(uploaded.keys())[0]

# Read CSV file into DataFrame
df = pd.read_csv(file_name)

# OR
#df=pd.read_csv("/content/advertising.csv")

# Print first 5 rows of the dataset
df.head()
```

[Choose Files](#) advertising.csv

**advertising.csv**(text/csv) - 4062 bytes, last modified: 3/20/2025 - 100% done  
Saving advertising.csv to advertising (1).csv

	TV	Radio	Newspaper	Sales	
0	230.1	37.8	69.2	22.1	
1	44.5	39.3	45.1	10.4	
2	17.2	45.9	69.3	12.0	
3	151.5	41.3	58.5	16.5	
4	180.8	10.8	58.4	17.9	

Next steps: [New interactive sheet](#)

```
# Read the CSV file from Colab's content folder into a DataFrame
#df = pd.read_csv("/content/advertising.csv")

# Display the first 5 rows of the DataFrame
#df.head()
```

## Information of Data

```
df.shape
```

```
(200, 4)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    TV          200 non-null    float64
1    Radio        200 non-null    float64
2    Newspaper    200 non-null    float64
3    Sales        200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

## ✓ Separate Input and Output

```
# Drop 'Newspaper' column from the input
X = df.drop(columns=['Sales', 'Newspaper'])
y = df['Sales']
```

X

	TV	Radio
0	230.1	37.8
1	44.5	39.3
2	17.2	45.9
3	151.5	41.3
4	180.8	10.8
...	...	...
195	38.2	3.7
196	94.2	4.9
197	177.0	9.3
198	283.6	42.0
199	232.1	8.6

200 rows × 2 columns

Next steps: [New interactive sheet](#)

y

	Sales
0	22.1
1	10.4
2	12.0
3	16.5
4	17.9
...	...
195	7.6
196	14.0
197	14.8
198	25.5
199	18.4

200 rows × 1 columns

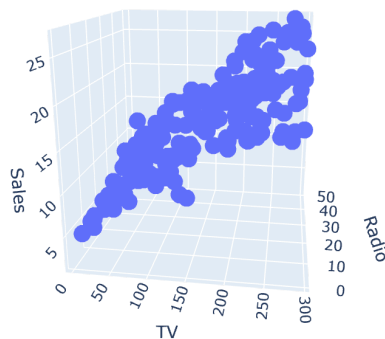
dtype: float64

## ✓ Plot the original 3D scatter plot

```
# Plot the original 3D scatter plot

fig = px.scatter_3d(df, x='TV', y='Radio', z='Sales')

fig.show()
```



### Train Linear Regression model

```
# Step 4: Create and train Linear Regression model
model = LinearRegression()
model.fit(X,y)
```

LinearRegression

LinearRegression()

model.coef\_

array([0.05444896, 0.10717457])

model.intercept\_

np.float64(4.630879464097768)

```
# Step 5: Take new input from user
tv = float(input("Enter TV advertisement budget: "))
radio = float(input("Enter Radio advertisement budget: "))
```

```
# Step 6: Create input DataFrame for prediction
new_data = pd.DataFrame({'TV': [tv], 'Radio': [radio]})
```

```
# Step 7: Predict sales
predicted_sales = model.predict(new_data)
```

```
# Step 8: Display result
print(f"Predicted Sales: {predicted_sales[0]:.2f}")
```

```
Enter TV advertisement budget: 85
Enter Radio advertisement budget: 65
Predicted Sales: 16.23
```

```
# Plot the original 3D scatter plot
fig = px.scatter_3d(df, x='TV', y='Radio', z='Sales')
```

```
# Add new predicted point (RED)
fig.add_scatter3d(x=[tv],y=[radio],z=[predicted_sales[0]],mode='markers',
marker=dict(size=8, color='red'),name='New Prediction')
```

```
fig.show()
```

