Task 4

sales prediction using python

In [3]:
```
import pandas as pd
import numpy as anp
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn import metrics
```

In [4]:
```
df = pd.read_csv("Advertising.csv")
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')
```

In [6]:
```
df.head()
```

Out[6]:

	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

In [7]:
```
df.info()
```

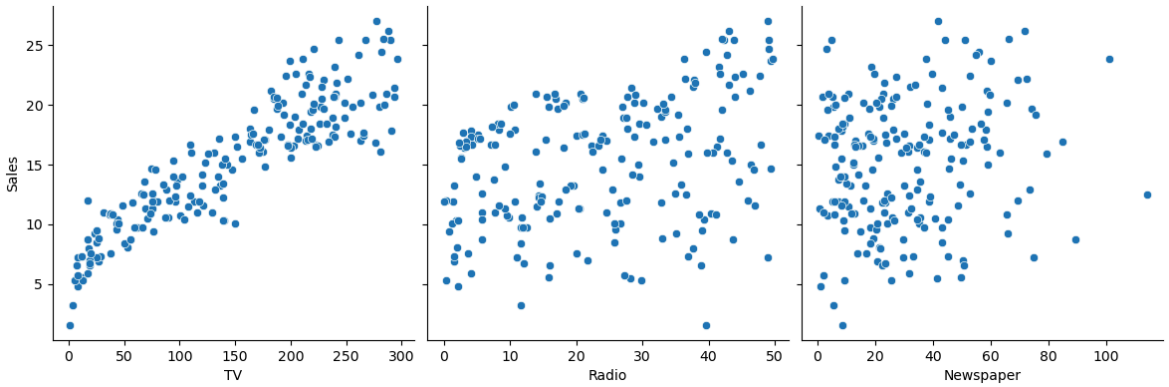
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64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

In [8]:
```
df.describe()
```

Out[8]:

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

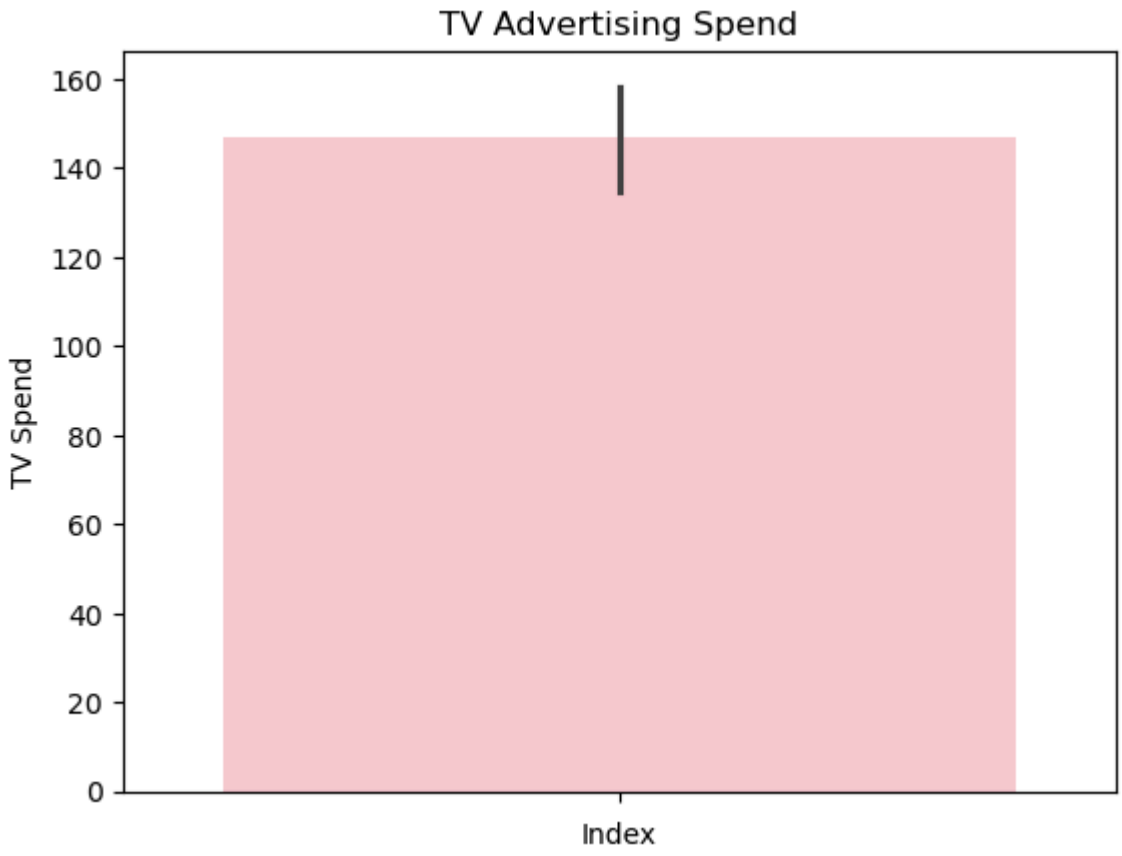
In [9]:
```
sns.pairplot(df, x_vars=['TV', 'Radio', 'Newspaper'], y_vars='Sales', height=4,
plt.show()
```



In [10]:
```
sns.heatmap(df.corr(), annot=True,cmap='plasma')
plt.title("Feature Correlation")
plt.show()
```

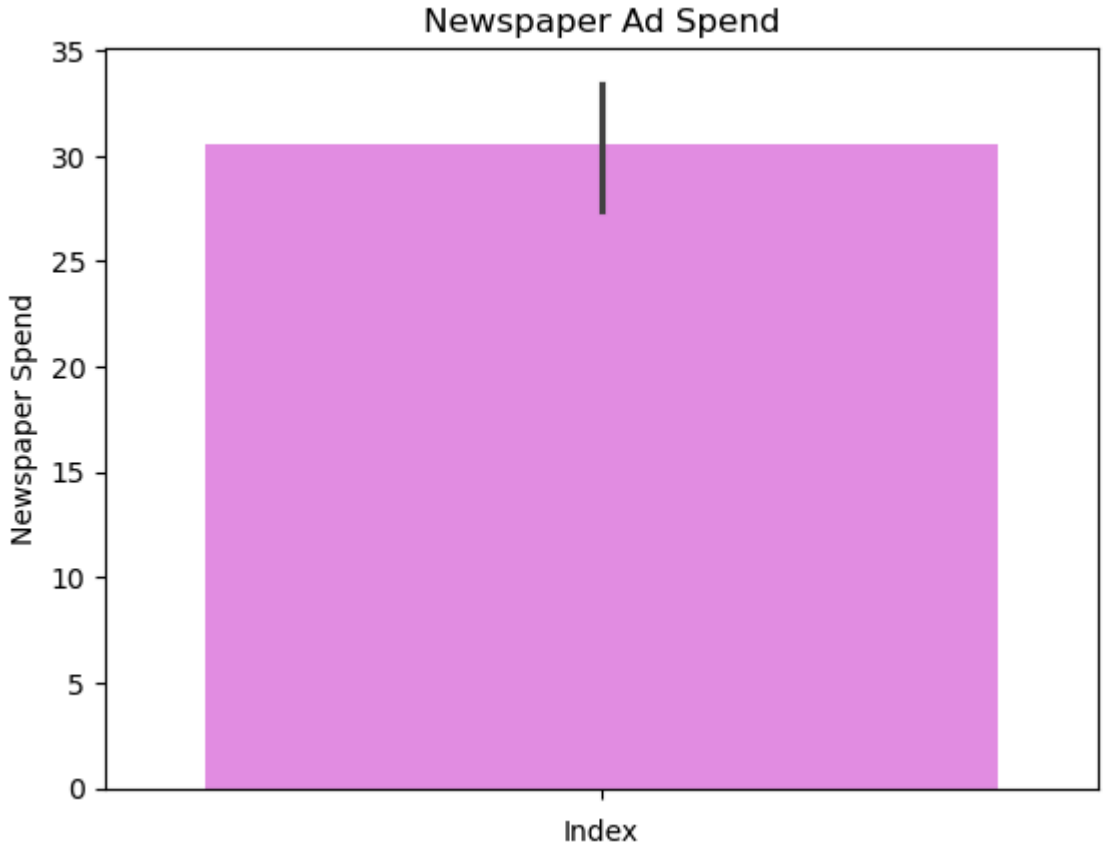
In [14]:
```
sns.barplot(y=df['TV'], color='pink').set(title='TV Advertising Spend', xlabel='
```

Out[14]:
```
[Text(0.5, 1.0, 'TV Advertising Spend'),
 Text(0.5, 0, 'Index'),
 Text(0, 0.5, 'TV Spend')]
```



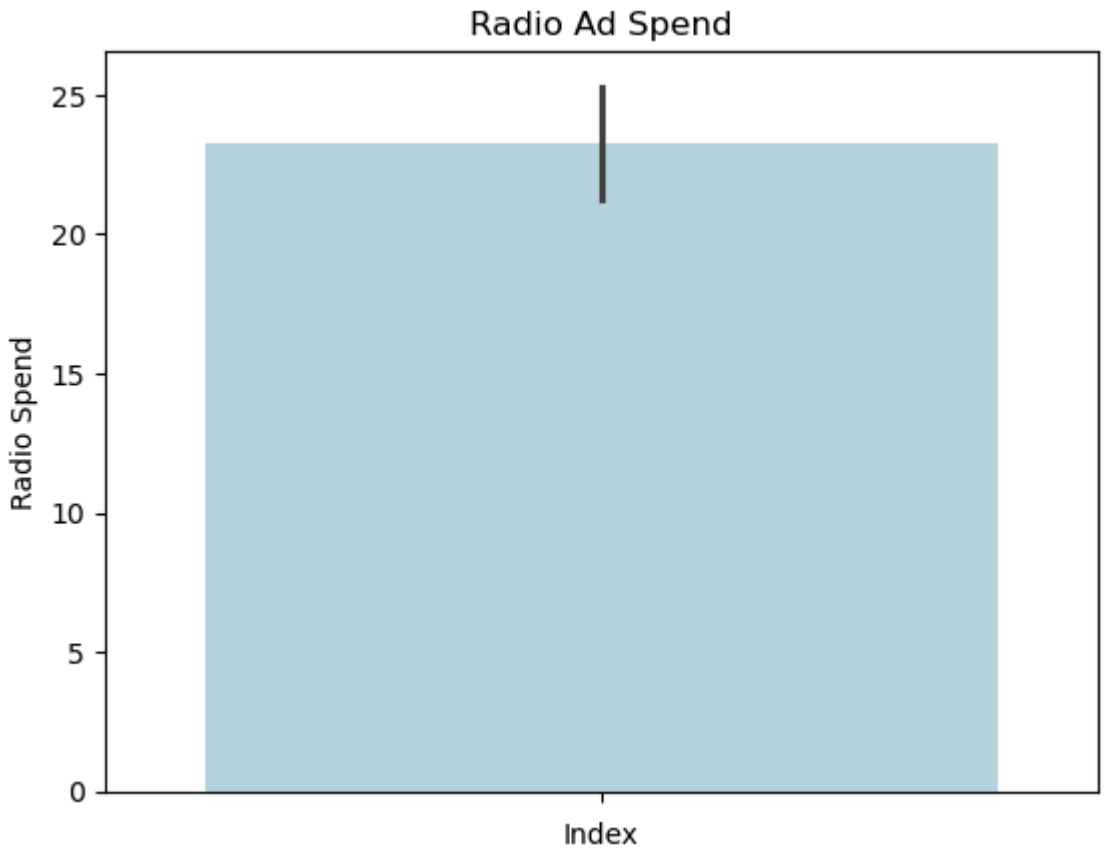
In [15]:
```
sns.barplot(y=df['Newspaper'], color='violet').set(title='Newspaper Ad Spend', x
```

Out[15]:
```
[Text(0.5, 1.0, 'Newspaper Ad Spend'),
 Text(0.5, 0, 'Index'),
 Text(0, 0.5, 'Newspaper Spend')]
```



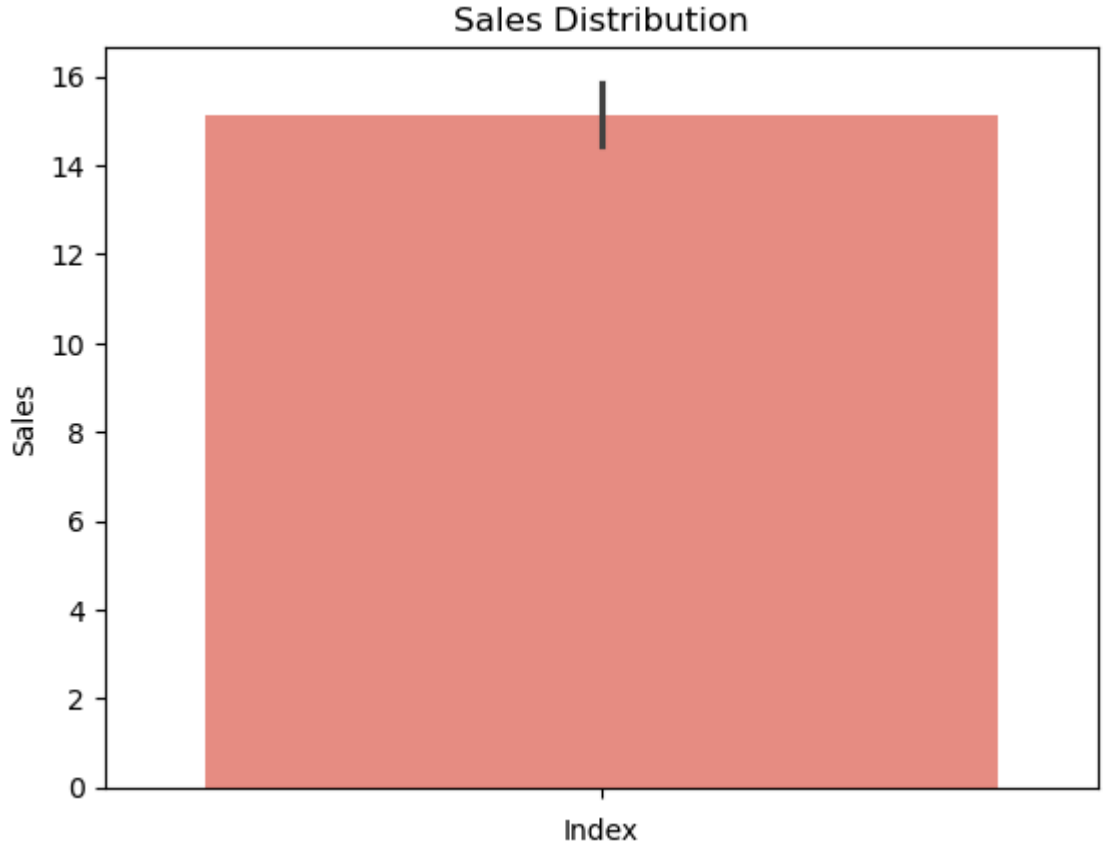
In [16]:
```
sns.barplot(y=df['Radio'], color='lightblue').set(title='Radio Ad Spend', xlabel
```

Out[16]:
```
[Text(0.5, 1.0, 'Radio Ad Spend'),
 Text(0.5, 0, 'Index'),
 Text(0, 0.5, 'Radio Spend')]
```



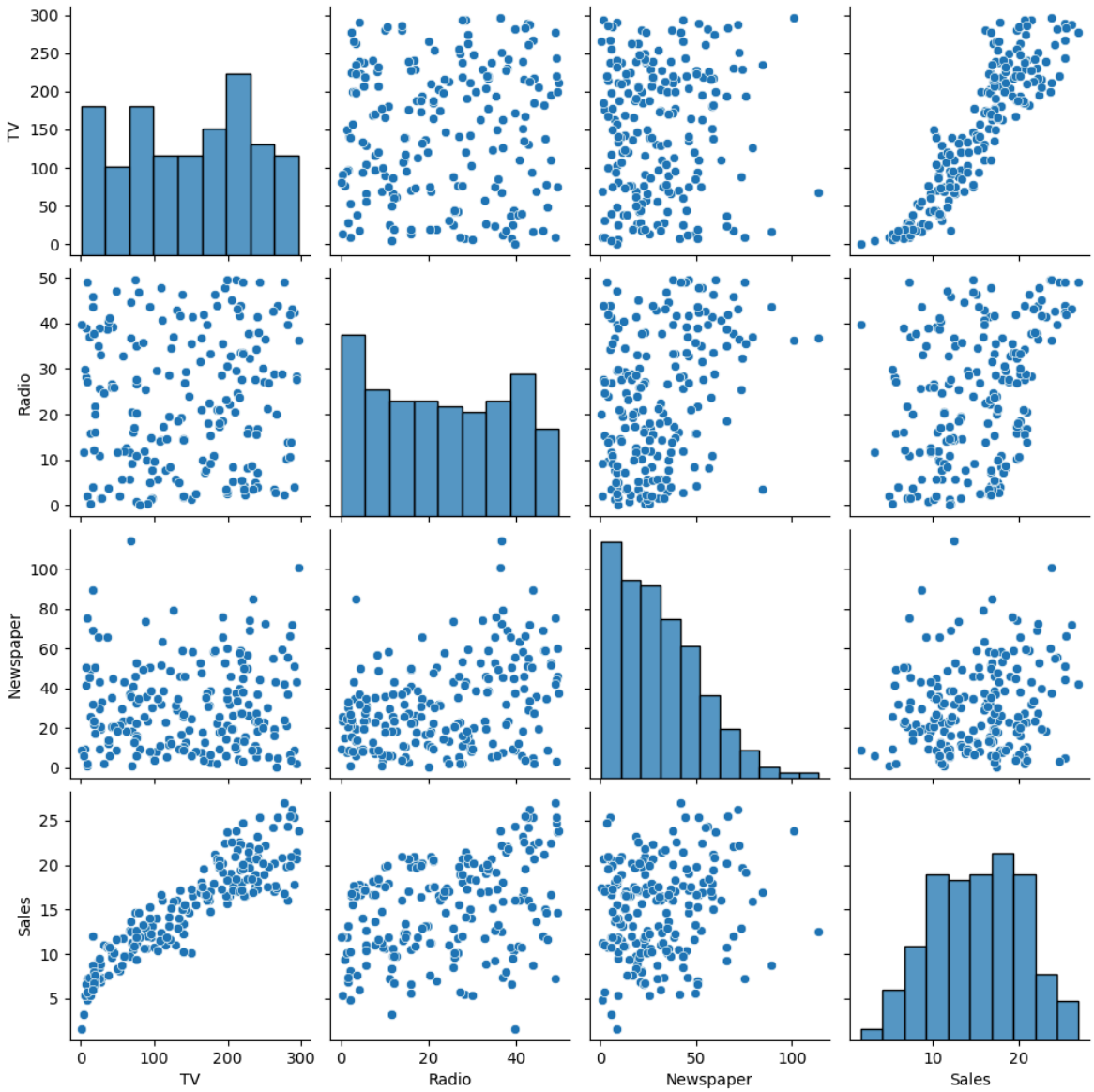
In [17]:
```
sns.barplot(y=df['Sales'], color='salmon').set(title='Sales Distribution', xlabe
```

Out[17]:
```
[Text(0.5, 1.0, 'Sales Distribution'),
 Text(0.5, 0, 'Index'),
 Text(0, 0.5, 'Sales')]
```



In [61]:
```
sns.pairplot(df)
```

Out[61]: <seaborn.axisgrid.PairGrid at 0x189260db820>



In [66]:
```
print(df.isnull().any())
```

```
TV           False
Radio        False
Newspaper    False
Sales        False
dtype: bool
```

In [67]:
```
x = df.iloc[:, 0:3]     # Independent variables
y = df['Sales']         # Dependent variable
```

In [69]:
```
x.head()
```

Out[69]:

	TV	Radio	Newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4

In [70]:
```
y.head()
```

Out[70]:
```
0    22.1
1    10.4
2    12.0
3    16.5
4    17.9
Name: Sales, dtype: float64
```

In [72]:
```
x.shape
```

Out[72]:
```
(200, 3)
```

In [73]:
```
y.shape
```

Out[73]:
```
(200,)
```

Since all the columns in the dataset are already numerical, there is no need for encoding or any additional transformation of categorical variables.

In [74]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_
```

In [75]:
```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(160, 3) (40, 3) (160,) (40,)
```

Model Building

In [76]:
```
lr = LinearRegression()
lr.fit(x_train, y_train)
```

Out[76]:
```
LinearRegression
LinearRegression()
```

In [79]:
```
lr.coef_
```

Out[79]:
```
array([0.05450927, 0.10094536, 0.00433665])
```

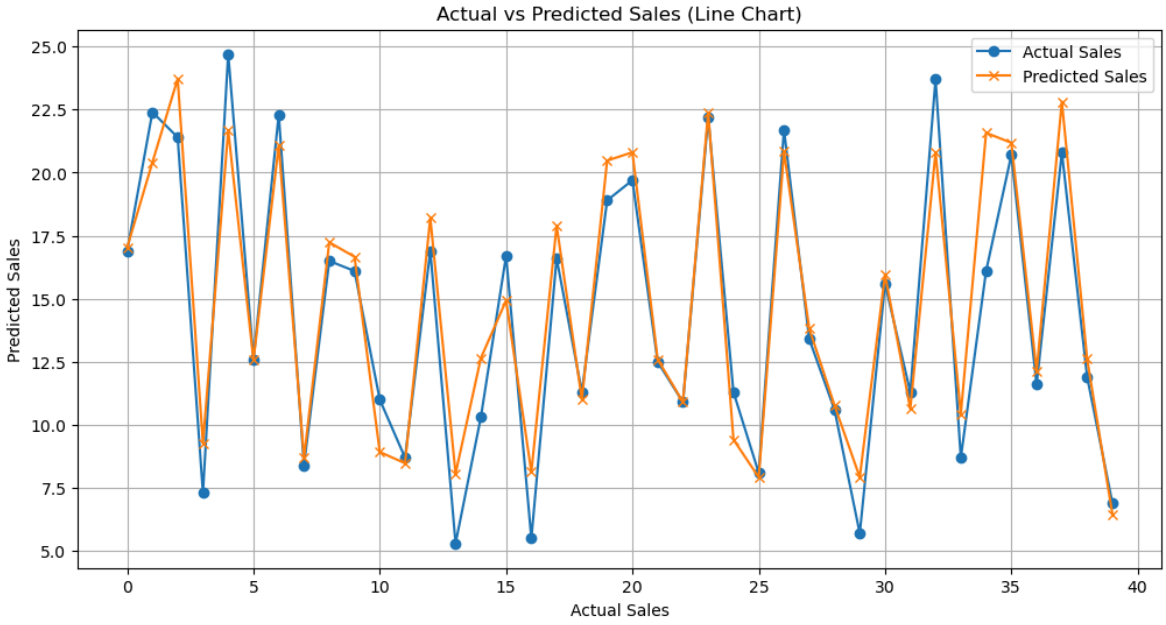
In [80]:
```
lr.intercept_
```

Out[80]:
```
np.float64(4.714126402214134)
```

In [85]:
```
y_pred=lr.predict(x_test)
```

In [99]:
```
plt.figure(figsize=(12,6))
plt.plot(y_test.values, label='Actual Sales', marker='o')
plt.plot(y_pred, label='Predicted Sales', marker='x')

plt.xlabel('Actual Sales')
plt.ylabel('Predicted Sales')
plt.title('Actual vs Predicted Sales (Line Chart)')
plt.legend()
plt.grid(True)
plt.show()
```



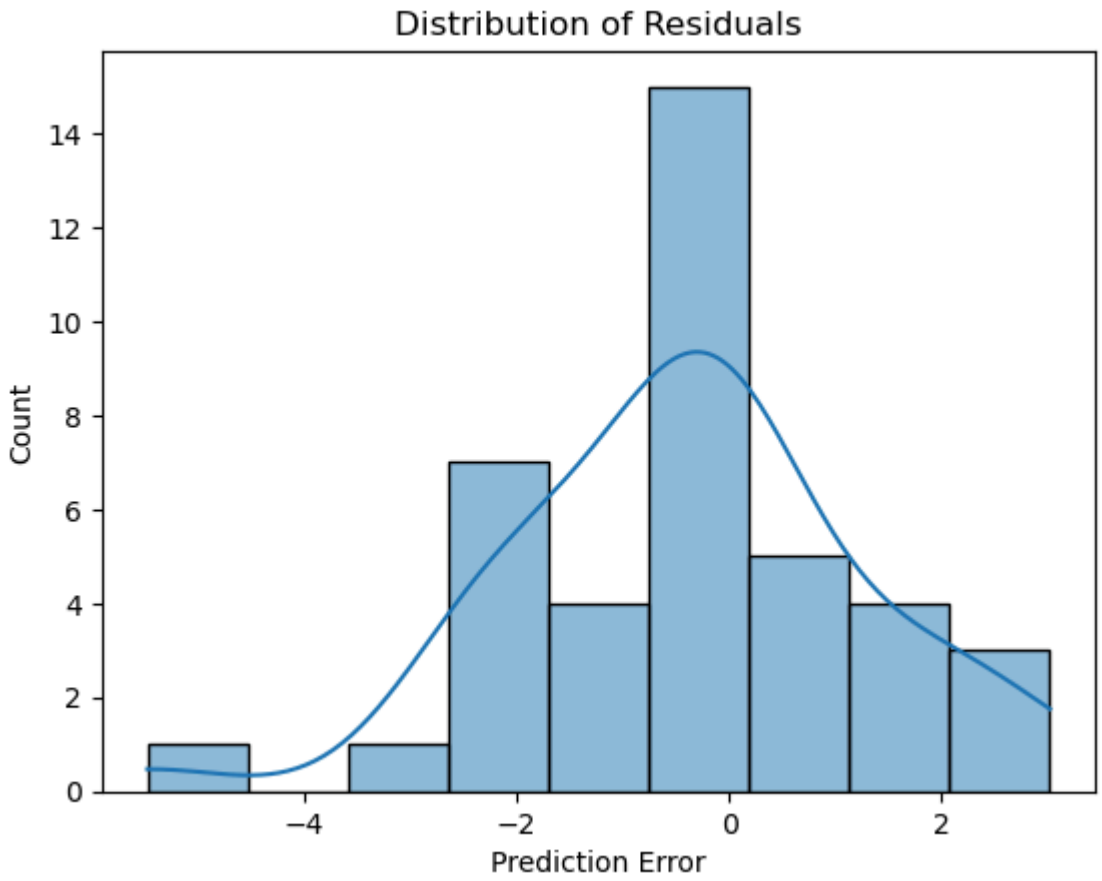
In [87]:
```
r2_score(y_test, y_pred)
```

Out[87]:
```
0.9059011844150826
```

In [88]:
```
metrics.mean_squared_error(y_test, y_pred)
```

Out[88]:
```
2.9077569102710923
```

In [100…
```
residuals = y_test - y_pred
sns.histplot(residuals, kde=True)
plt.title("Distribution of Residuals")
plt.xlabel("Prediction Error")
plt.show()
```



Conclusion

The linear regression model shows strong predictive performance with an R^2 score of approximately 0.91, indicating that over 90% of the variance insales can be explained by advertising spends across TV, Radio, and Newspaper. The low Mean Squared Error confirms the model's accuracy, making itsuitable for predicting future sales trends.

In [ ]: