In [1]:

```python
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```python
data=pd.read_csv("C:\\Users\\USER\\Documents\\Sales.csv")
```

In [3]:

```python
data
```

Out[3]:

	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
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

In [4]:

```python
data.columns
```

Out[4]:

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

In [5]:

```python
data.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 [6]:  data.isnull().sum()
```

```
Out[6]: TV           0
        Radio        0
        Newspaper    0
        Sales        0
        dtype: int64
```

```
In [7]:  data.shape
```

```
Out[7]: (200, 4)
```

```
In [8]:  data.drop_duplicates()
```

Out[8]:

	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
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

```
In [9]:  data.count()
```

```
Out[9]: TV           200
        Radio        200
        Newspaper    200
        Sales        200
        dtype: int64
```

In [10]:

```
data.describe()
```

Out[10]:

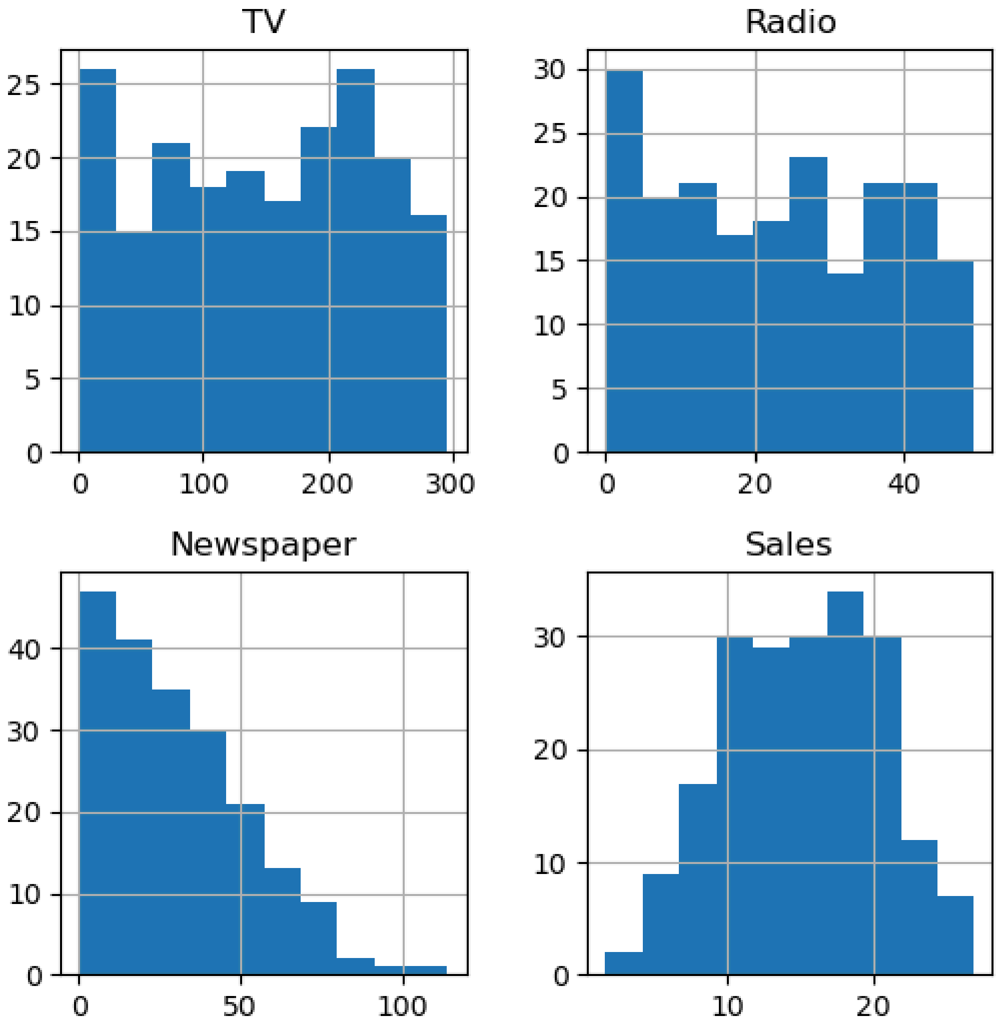
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000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

In [11]:

```
data.hist(figsize=[6,6])
```

Out[11]:

```
array([[<Axes: title={'center': 'TV'}>,
        <Axes: title={'center': 'Radio'}>],
       [<Axes: title={'center': 'Newspaper'}>,
        <Axes: title={'center': 'Sales'}>]], dtype=object)
```

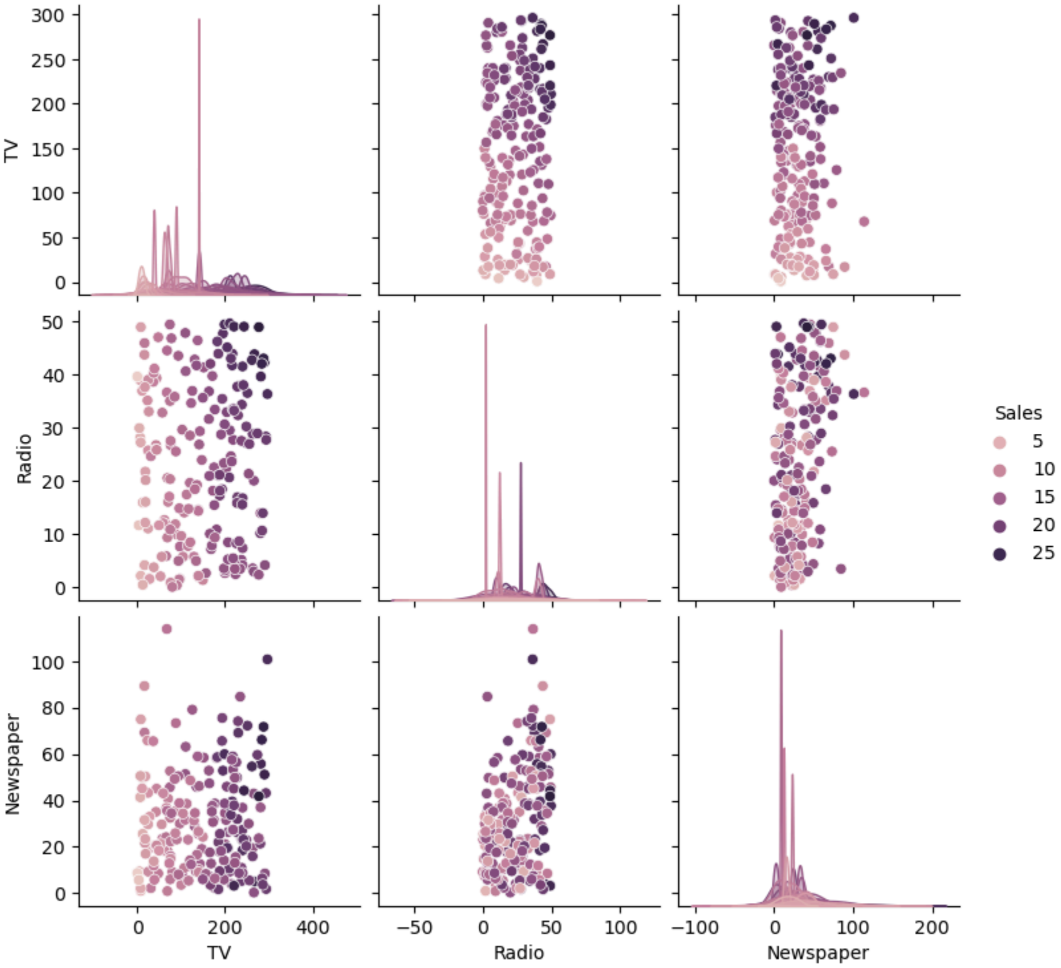


In [12]:

```python
import matplotlib.pyplot as plt
import seaborn as sns
```

In [13]:

```python
sns.pairplot(data, hue="Sales")
plt.show()
```



In [14]:

```
X=data.iloc[:,:3]
X
```

Out[14]:

	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
...	...	...	...
195	38.2	3.7	13.8
196	94.2	4.9	8.1
197	177.0	9.3	6.4
198	283.6	42.0	66.2
199	232.1	8.6	8.7

200 rows × 3 columns

In [15]:

```
y=data.iloc[:,3]
y
```

Out[15]:

```
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
Name: Sales, Length: 200, dtype: float64
```

In [16]:

```
from sklearn.model_selection import train_test_split
```

In [17]:

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

In [18]:

```
from sklearn.linear_model import LinearRegression
```

In [19]:

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

Out[19]:

```
▾ LinearRegression
LinearRegression()
```

In [20]:

```
y_pre=model.predict(X_test)
y_pre
```

Out[20]:

```
array([17.0347724 , 20.40974033, 23.72398873,  9.27278518, 21.6827187
9,
       12.56940161, 21.08119452,  8.69035045, 17.23701254, 16.6665747
5,
        8.92396497,  8.4817344 , 18.2075123 ,  8.06750728, 12.6455097
5,
       14.93162809,  8.12814594, 17.89876565, 11.00880637, 20.4783278
8,
       20.80631846, 12.59883297, 10.9051829 , 22.38854775,  9.4179609
4,
        7.92506736, 20.83908497, 13.81520938, 10.77080925,  7.9268250
9,
       15.95947357, 10.63490851, 20.80292008, 10.43434164, 21.5784752
,
       21.18364487, 12.12821771, 22.80953262, 12.60992766,  6.4644125
2])
```

In [21]:

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

In [22]:

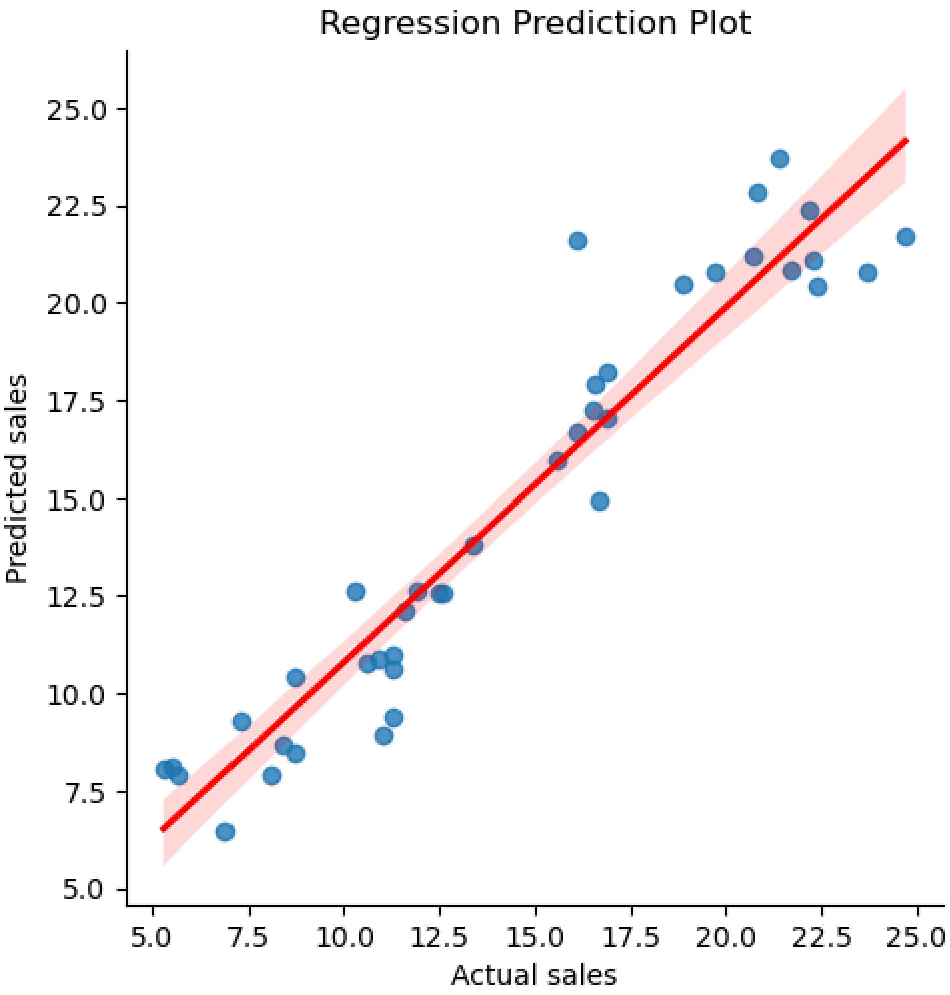
```
prediction=mean_squared_error(y_test,y_pre)
r2=r2_score(y_test,y_pre)
print(f'Mean Squared Error:',prediction)
print(f'R^2 Score:',r2)
```

```
Mean Squared Error: 2.907756910271089
R^2 Score: 0.9059011844150826
```

In [23]:

```
y_test =y_test
y_pred =y_pre
results = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
sns.lmplot(x='Actual', y='Predicted', data=results, line_kws={'colo
plt.xlabel('Actual sales')
plt.ylabel('Predicted sales')
plt.title('Regression Prediction Plot')
print('r2 Squared Error :',r2)
plt.show()
```

```
r2 Squared Error : 0.9059011844150826
```



Conclusion

Mainly we focused on Linear Regression

We took movie rating dataset and performed a linear regression algorithm

Finally, it classified into their ratings of movies accurately.

And I got an accuracy of r2 error 0.90, which shows that the model we built is very accurate.

In [ ]:

```
```