In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
df=pd.read_csv(r"C:\Users\Mastan Reddy\Downloads\Advertising.csv")
```

In [3]:

```
df.head()
```

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

In [4]:

```
df.tail()
```

Out[4]:

	TV	Radio	Newspaper	Sales
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

In [5]:

```
df.shape
```

Out[5]:

```
(200, 4)
```

In [6]:

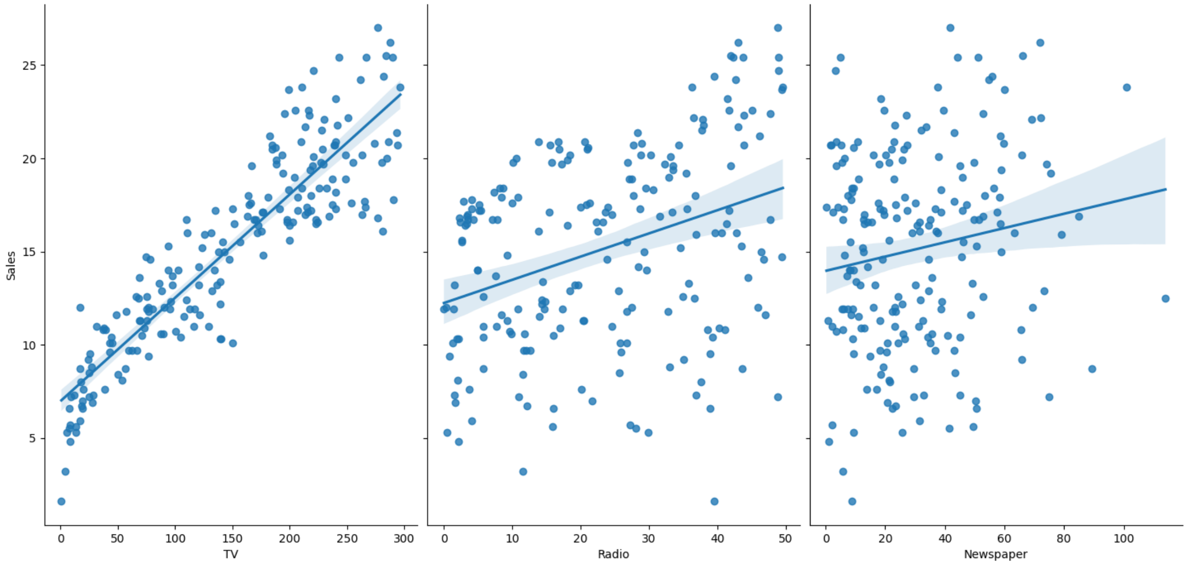
```
df.info()
```

```
<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 [7]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [8]:
```
sns.pairplot(df,x_vars=['TV','Radio','Newspaper'],y_vars='Sales',height=7,aspe
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0xe9f9f4bf48>
```



In [9]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [10]:
```
features=['TV','Radio','Newspaper']
x=df[features]
```

In [11]:
```
x=df[['TV','Radio','Newspaper']]
```

In [12]:
```
x.head()
```

Out[12]:

	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 [13]:
```
print(type(x))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [14]:
```
print(x.shape)
```
```
(200, 3)
```

In [15]:
```
y = df['Sales']
y = df.Sales


y.head()
```
Out[15]:
```
0    22.1
1    10.4
2    12.0
3    16.5
4    17.9
Name: Sales, dtype: float64
```

In [16]:
```
print(type(y))
print(y.shape)
```
```
<class 'pandas.core.series.Series'>
(200,)
```

In [17]:
```
from sklearn.model_selection import  train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=1)
```

In [18]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```
```
(150, 3)
(50, 3)
(150,)
(50,)
```

ELASTICNET REGRESSION

In [19]:
```
from sklearn.linear_model import ElasticNet
regr=ElasticNet()
regr.fit(x,y)
print(regr.coef_)
print(regr.intercept_)
```
```
[0.05440081 0.1046715  0.        ]
4.696191158087224
```

In [21]:
```
y_predict_elastic = regr.predict(x_train)
```

In [22]:

```
mean_squared_error=np.mean((y_predict_elastic-y_train)**2)
print("mean squared error on test set",mean_squared_error)
```

```
mean squared error on test set 2.860921026839881
```

In [ ]: