In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing, svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
#Step-2: Reading the Dataset
df=pd.read_csv(r"C:\Users\sudheer\Downloads\Advertising.csv")
df
```

Out[2]:

	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]:
```
df=df[['TV','Radio']]
df.columns=['tv','radio']
```

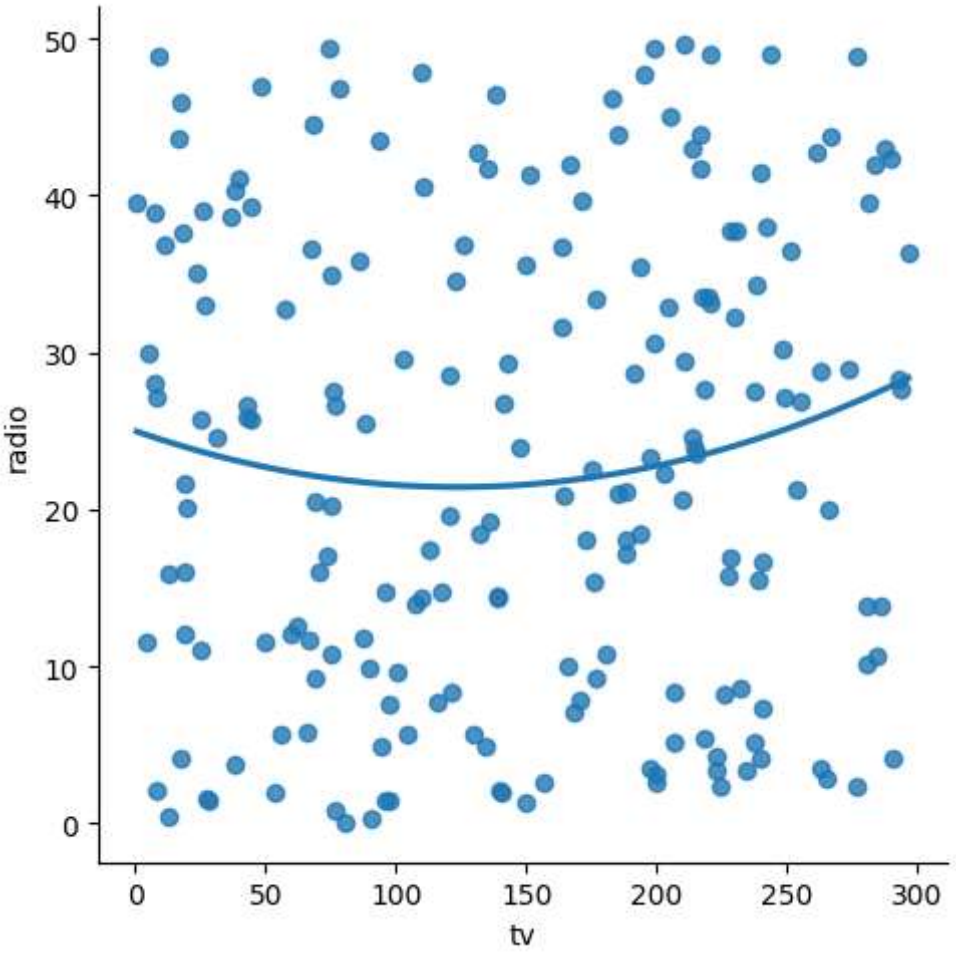
In [6]:
```
df.head(10)
```

Out[6]:

	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
5	8.7	48.9
6	57.5	32.8
7	120.2	19.6
8	8.6	2.1
9	199.8	2.6

In [8]:
```
#Step-3: Exploring the Data Scatter - plotting the data scatter
sns.lmplot(x="tv",y="radio", data = df, order = 2, ci = None)
df.describe()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   tv      200 non-null    float64
 1   radio   200 non-null    float64
dtypes: float64(2)
memory usage: 3.2 KB
```



In [9]:
```
#Step-4: Data cleaning - Eliminating NaN OR missing input numbers
df.fillna(method ='ffill', inplace = True)
```

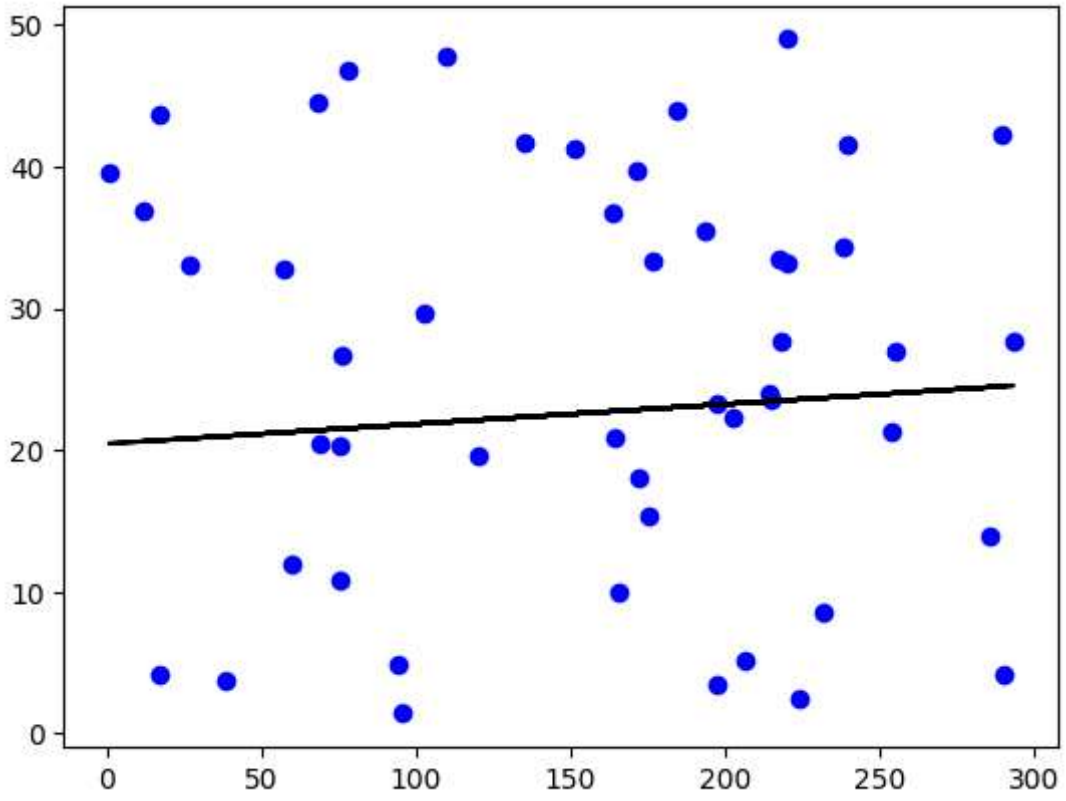
```
C:\Users\sudheer\AppData\Local\Temp\ipykernel_14712\3221840372.py:2: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df.fillna(method ='ffill', inplace = True)
```

In [10]:
```
# Step-5: Training Our Model
X = np.array(df['tv']).reshape(-1, 1)
y = np.array(df['radio']).reshape(-1, 1)
#Seperating the data into independent and dependent variables and convert
#Now each dataset contains only one column
```

In [11]:
```
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size = 0.25)
# Splitting the data into training data and test data
regr = LinearRegression()
regr.fit(X_train, y_train)
print(regr.score(X_test, y_test))
```

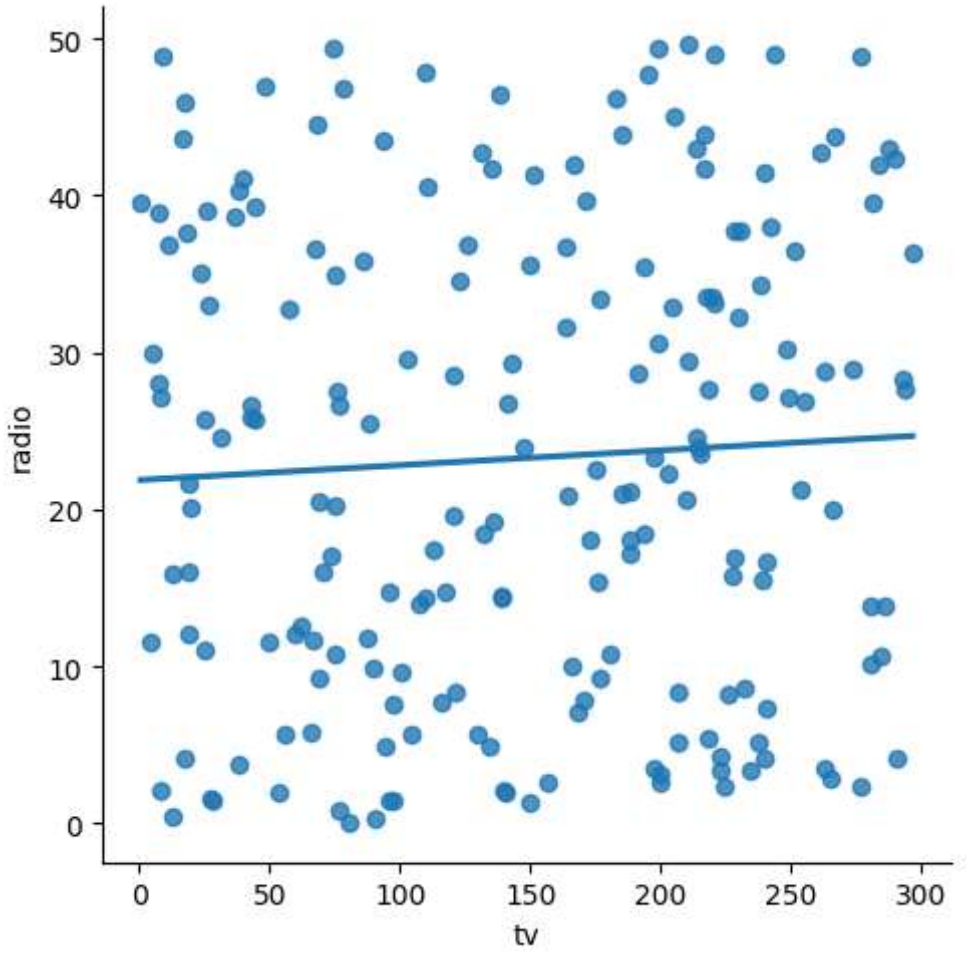
```
-0.06239630222121151
```

In [12]:
```
#step-6: Exploring Our Results
y_pred = regr.predict(X_test)
plt.scatter(X_test, y_test, color = 'b')
plt.plot(X_test, y_pred,color='k')
plt.show()
# Data scatter of predicted values
```



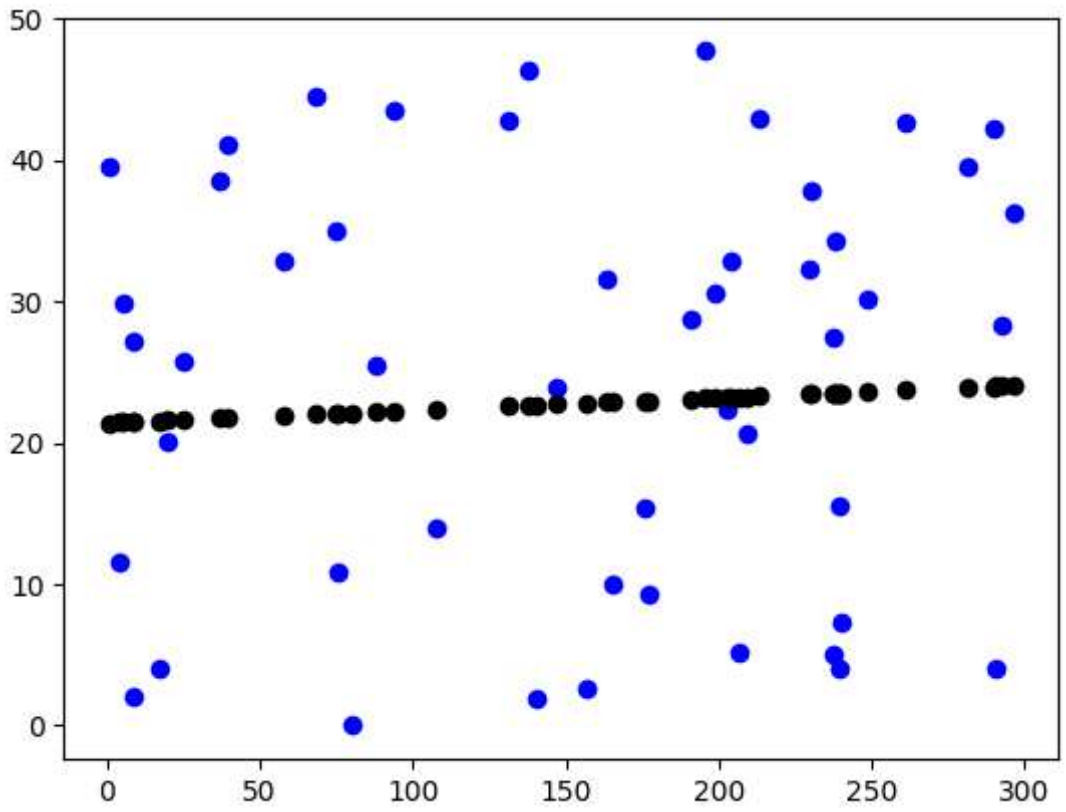
In [13]:
```
# Step-7: Working with a smaller Dataset
df500 = df[:][:500]
# Selecting the 1st 500 rows of the data
sns.lmplot(x = "tv", y ="radio", data = df500, order = 1, ci = None)
```

Out[13]: <seaborn.axisgrid.FacetGrid at 0x2e023b883a0>



In [14]:
```
df500.fillna(method = 'ffill', inplace = True)
X = np.array(df500['tv']).reshape(-1, 1)
y = np.array(df500['radio']).reshape(-1, 1)
df500.dropna(inplace = True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
regr = LinearRegression()
regr.fit(X_train, y_train)
print("Regression:",regr.score(X_test, y_test))
y_pred = regr.predict(X_test)
plt.scatter(X_test, y_test, color = 'b')
plt.scatter(X_test, y_pred, color = 'k')
plt.show()
```

```
Regression: -0.019615203158702244
```



In [16]:
```
#Step-8: Evaluation of model

from sklearn.linear_model import LinearRegression

from sklearn.metrics import r2_score

#Train the model

model = LinearRegression()

model.fit(X_train, y_train)

#Evaluating the model on the test set

y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)

print("R2 score:",r2)
```

```
R2 score: -0.019615203158702244
```

In [ ]: