In [455...
```
import pandas as pd
import numpy as np
```

In [456...
```
df = pd.read_csv("C:\\users\\solun\\Advertising.csv")
```

In [457...
```
df.head()
```

Out[457...

	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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

In [458...
```
## check for missing values

#df.isnull().sum()
```

In [459...
```
## check for data types

#df.dtypes
```

In [460...
```
## check for the distribution of varaibles

#import seaborn as sns
```

In [461...
```
#sns.distplot(df["TV"])
```

In [462...
```
#df.describe()
```

In [463...
```
#sns.distplot(df["newspaper"])
```

In [464...
```
### log transformation for the skewed variable

#df["newspaper"] = np.log1p(df["newspaper"])
```

In [465...
```
#sns.distplot(df["newspaper"])
```

In [466...
```
#df.describe()
```

In [467...
```
df_num = df.drop("sales",axis=1)
```

In [468...
```
## scaling tchnique for numerical variables

from sklearn.preprocessing import MinMaxScaler
```

In [469...
```
# apply scaling on numerical independent variables

mn = MinMaxScaler()
df_sc = mn.fit_transform(df_num)
```

In [470...
```
## convert array t0 dataframe

df_sc_df = pd.DataFrame(df_sc, columns=df_num.columns, index=df.index)
```

In [471...
```
df_sc_df.head()
```

Out[471...

	TV	radio	newspaper
0	0.775786	0.762097	0.605981
1	0.148123	0.792339	0.394019
2	0.055800	0.925403	0.606860
3	0.509976	0.832661	0.511873
4	0.609063	0.217742	0.510994

In [472...
```
x = df_sc_df
y = df["sales"]
```

In [473...
```
## train test split

from sklearn.model_selection import train_test_split
```

In [474...
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state= 88)
```

In [475...
```
print("Shape of Training set input:",X_train.shape)
print("Shape of Testing set input:",X_test.shape)
```

```
Shape of Training set input: (160, 3)
Shape of Testing set input: (40, 3)
```

In [476...
```
print("Total VALUES in Training set output:",y_train.size)
print("Total VALUES in Testing set output:",y_test.size)
```

```
Total VALUES in Training set output: 160
Total VALUES in Testing set output: 40
```

In [477...
```
from sklearn.linear_model import LinearRegression
```

In [478...
```
lr = LinearRegression()
lr.fit(X_train, y_train)
```

Out[478...
```
LinearRegression()
```

In [479...
```
## predict

y_pred = lr.predict(X_test)
```

In [480...
```
import numpy as np
ytest=np.array(y_test)
```

In [486...
```
df_actual=pd.DataFrame(ytest,columns=['Actual_Output'])
df_predicted=pd.DataFrame(y_pred,columns=['Predicted_Output'])
df_output=pd.concat([df_actual,df_predicted],axis=1)
df_output.head()
```

Out[486...

	Actual_Output	Predicted_Output
0	13.4	15.088370
1	9.3	12.492225
2	7.3	10.623041
3	9.7	8.893642
4	21.5	20.522669

In [ ]:
```

```

In [482...
```
from sklearn.metrics import r2_score, mean_squared_error
```

In [452...
```
## check for overfitting

print("Training Accuracy:",r2_score(y_train, lr.predict(X_train)))
```

```
Training Accuracy: 0.9095202869396248
```

In [453...
```
## score on test data

print("Testing Accuracy:",r2_score(y_test, pred))
```

```
Testing Accuracy: -1.0660520781400495
```

In [454...
```
#mean_squared_error(y_test, pred)
```

In [270...
```
#y =  m1x1+ m2x2+m3x3...+c
```

In [354...
```
lr.coef_
```

Out[354...
```
array([13.18725332,  9.76171993, -0.61325372])
```

In [397...
```
X_test.head(1)
```

Out[397...

	TV	radio	newspaper
199	0.78255	0.173387	0.448351

In [403...
```
#pred_sale=m1*TV+m2*Radio+m3*Newspaper
predy0=13.18725332*0.782550+9.76171993*0.173387+-0.61325372*0.448351
print("Estimated output:",predy0)
print("Actual output:",ytest[0])
y_test.head(1)
```

```
Estimated output: 11.737287500455319
Actual output: 13.4
199    13.4
Name: sales, dtype: float64
```

In [302...
```
## model is overfitting

## newspaper is not important
```

In [301...
```
x = df_sc_df.drop("newspaper",axis=1)
y = df["sales"]
```

In [304...
```
X_train1, X_test1, y_train1, y_test1 = train_test_split(x, y, test_size=0.2, random_state= 89)
```

In [305...
```
lr.fit(X_train1, y_train1)
```

Out[305...
```
LinearRegression()
```

In [306...
```
ypred1 = lr.predict(X_test1)
```

In [307...
```
import numpy as np
ytest1=np.array(y_test1)
```

In [358...
```
df_actual=pd.DataFrame(ytest1,columns=['Actual_Output'])
df_predicted=pd.DataFrame(ypred1,columns=['Predicted_Output'])
df_output=pd.concat([df_actual,df_predicted],axis=1)
df_output.head(15)
```

Out[358...

	Actual_Output	Predicted_Output
0	7.3	4.191805
1	12.8	12.540793
2	8.8	6.255200
3	19.0	18.485802
4	12.0	15.317401
5	16.9	16.358726
6	9.3	7.596242
7	8.4	7.184837
8	20.7	20.326066
9	9.7	7.872899
10	15.5	14.394697
11	11.4	9.937876
12	23.8	22.121433
13	12.6	12.158402
14	5.3	8.790742

In [309...
```
r2_score(y_test1, ypred1)
```

Out[309...
```
0.8804767444671727
```

In [310...
```
r2_score(y_train1, lr.predict(X_train1))
```

Out[310...
```
0.8996373226035996
```

In [311...
```
lr.coef_
```

Out[311...
```
array([13.53799565,  9.88698984])
```