**Program:**

import warnings

warnings.simplefilter("ignore")

import numpy as np

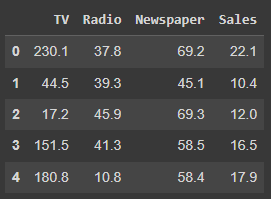
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

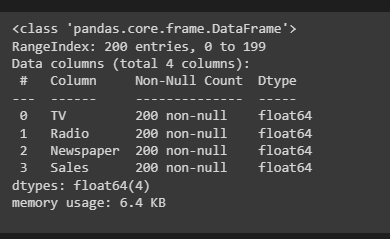
advertising =pd.DataFrame(pd.read\_csv("/content/advertising.csv"))

advertising.head()

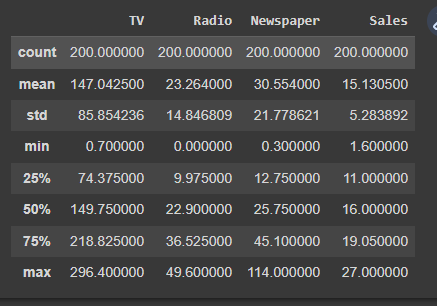


advertising.shape

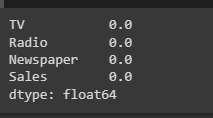
advertising.info()



advertising.describe()



advertising.isnull().sum()\*100/advertising.shape[0]



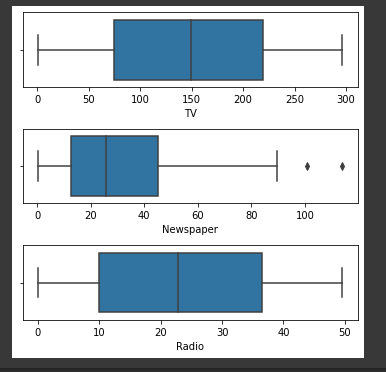
fig,axs=plt.subplots(3,figsize=(5,5))

plt1=sns.boxplot(advertising['TV'],ax=axs[0])

plt2=sns.boxplot(advertising['Newspaper'],ax=axs[1])

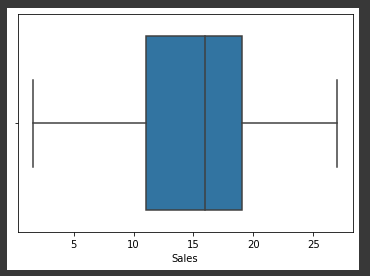
plt3=sns.boxplot(advertising['Radio'],ax=axs[2])

plt.tight\_layout()



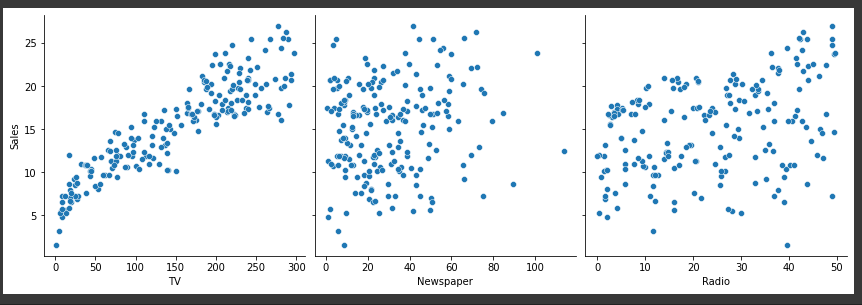
sns.boxplot(advertising['Sales'])

plt.show()



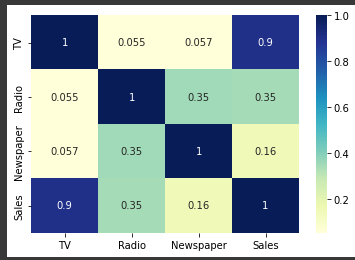
sns.pairplot(advertising,x\_vars=['TV','Newspaper','Radio'],y\_vars='Sales',height=4,aspect=1,kind='scatter')

plt.show()



sns.heatmap(advertising.corr(),cmap="YlGnBu",annot=True)

plt.show()



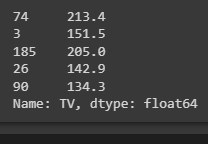
X=advertising['TV']

y=advertising['Sales']

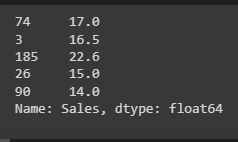
from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,train\_size=0.7,test\_size=0.3,random\_state=100)

X\_train.head()



y\_train.head()

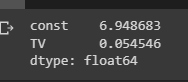


import statsmodels.api as sm

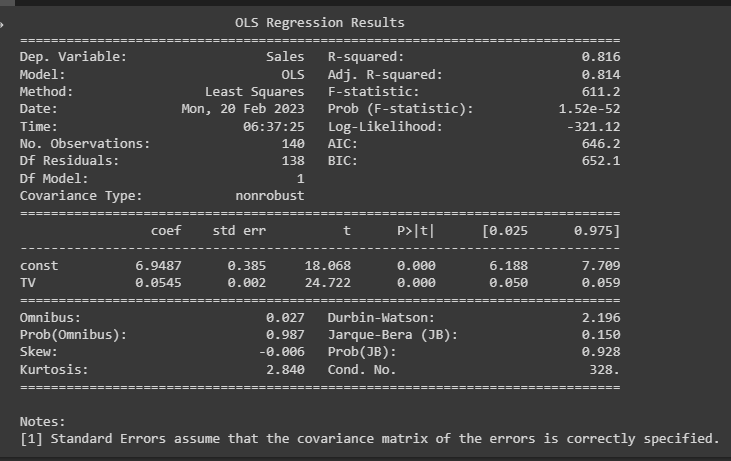
X\_train\_sm=sm.add\_constant(X\_train)

lr=sm.OLS(y\_train,X\_train\_sm).fit()

lr.params



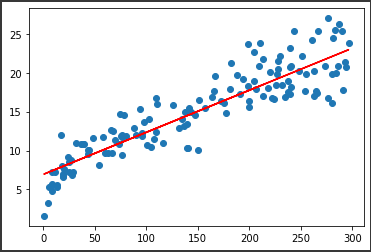
print(lr.summary())



plt.scatter(X\_train,y\_train)

plt.plot(X\_train,6.948+0.054\*X\_train,'r')

plt.show()



y\_train\_pred=lr.predict(X\_train\_sm)

res=(y\_train-y\_train\_pred)

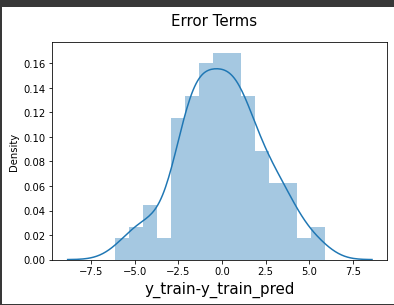
fig=plt.figure()

sns.distplot(res,bins=15)

fig.suptitle('Error Terms',fontsize=15)

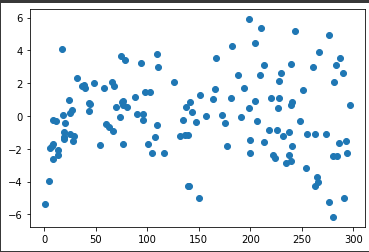
plt.xlabel('y\_train-y\_train\_pred',fontsize=15)

plt.show()



plt.scatter(X\_train,res)

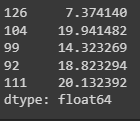
plt.show()



X\_test\_sm=sm.add\_constant(X\_test)

y\_pred=lr.predict(X\_test\_sm)

y\_pred.head()



from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import r2\_score

np.sqrt(mean\_squared\_error(y\_test,y\_pred))

2.019296008966232

r\_squared=r2\_score(y\_test,y\_pred)

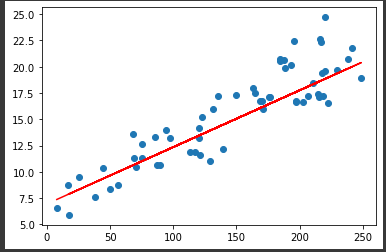
r\_squared

0.792103160124566

plt.scatter(X\_test,y\_test)

plt.plot(X\_test,6.948+0.054\*X\_test,'r')

plt.show()



Program:

import csv

num\_attributes = 6

a = []

print("\n The Given Training Data Set \n")

with open('/content/ENJOYSPORT.csv', 'r') as csvfile:

reader = csv.reader(csvfile)

for row in reader:

a.append (row)

print(row)

print("\n The initial value of hypothesis: ")

hypothesis = ['0'] \* num\_attributes

print(hypothesis)

for j in range(0,num\_attributes):

hypothesis[j] = a[0][j];

print("\n Find S: Finding a Maximally Specific Hypothesis\n")

for i in range(0,len(a)):

if a[i][num\_attributes]=='yes':

for j in range(0,num\_attributes):

if a[i][j]!=hypothesis[j]:

hypothesis[j]='?'

else :

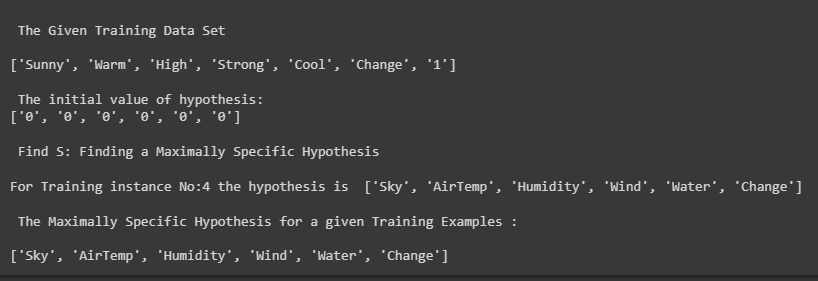
hypothesis[j]= a[i][j]

print("For Training instance No:{0} the hypothesis is ".format(i),hypothesis)

print("\n The Maximally Specific Hypothesis for a given Training Examples :\n")

print(hypothesis)

output:



Program:

import numpy as np

import pandas as pd

data = pd.DataFrame(data=pd.read\_csv('/content/ENJOYSPORT.csv'))

concepts = np.array(data.iloc[:,0:-1])

print(concepts)

target = np.array(data.iloc[:,-1])

print(target)

def learn(concepts, target):

specific\_h = concepts[0].copy()

print("initialization of specific\_h and general\_h")

print(specific\_h)

general\_h = [["?" for i in range(len(specific\_h))] for i in range(len(specific\_h))]

print(general\_h)

for i, h in enumerate(concepts):

if target[i] == "yes":

for x in range(len(specific\_h)):

if h[x]!= specific\_h[x]:

specific\_h[x] ='?'

general\_h[x][x] ='?'

print(specific\_h)

print(specific\_h)

if target[i] == "no":

for x in range(len(specific\_h)):

if h[x]!= specific\_h[x]:

general\_h[x][x] = specific\_h[x]

else:

general\_h[x][x] = '?'

print(" steps of Candidate Elimination Algorithm",i+1)

print(specific\_h)

print(general\_h)

indices = [i for i, val in enumerate(general\_h) if val == ['?', '?', '?', '?', '?', '?']]

for i in indices:

general\_h.remove(['?', '?', '?', '?', '?', '?'])

return specific\_h, general\_h

s\_final, g\_final = learn(concepts, target)

print("Final Specific\_h:", s\_final, sep="\n")

print("Final General\_h:", g\_final, sep="\n")

output:

