lab 2 of machile learning

Submitted by

Sarvesh kumar sharma

Section- A

Roll number- 50

university roll number- 181500625

importing Essential libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Reading CSV file using read_csv function

In [2]:

```
df=pd.read_csv("headbrain.csv")
df.head()
```

Out[2]:

	Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590

splitting data for prediction

x- input data

y- output data

In [3]:

```
x=df["Head Size(cm^3)"].values
y=df["Brain Weight(grams)"].values
print(x)
print(y)
```

```
[4512 3738 4261 3777 4177 3585 3785 3559 3613 3982 3443 3993 3640 4208
 3832 3876 3497 3466 3095 4424 3878 4046 3804 3710 4747 4423 4036 4022
 3454 4175 3787 3796 4103 4161 4158 3814 3527 3748 3334 3492 3962 3505
 4315 3804 3863 4034 4308 3165 3641 3644 3891 3793 4270 4063 4012 3458
 3890 4166 3935 3669 3866 3393 4442 4253 3727 3329 3415 3372 4430 4381
 4008 3858 4121 4057 3824 3394 3558 3362 3930 3835 3830 3856 3249 3577
```

```
 4006 3858 4121 4057 3824 3594 3558 3562 3950 3855 3850 3856 3249 3577
 3933 3850 3309 3406 3506 3907 4160 3318 3662 3899 3700 3779 3473 3490
 3654 3478 3495 3834 3876 3661 3618 3648 4032 3399 3916 4430 3695 3524
 3571 3594 3383 3499 3589 3900 4114 3937 3399 4200 4488 3614 4051 3782
 3391 3124 4053 3582 3666 3532 4046 3667 2857 3436 3791 3302 3104 3171
 3572 3530 3175 3438 3903 3899 3401 3267 3451 3090 3413 3323 3680 3439
 3853 3156 3279 3707 4006 3269 3071 3779 3548 3292 3497 3082 3248 3358
 3803 3566 3145 3503 3571 3724 3615 3203 3609 3561 3979 3533 3689 3158
 4005 3181 3479 3642 3632 3069 3394 3703 3165 3354 3000 3687 3556 2773
 3058 3344 3493 3297 3360 3228 3277 3851 3067 3692 3402 3995 3318 2720
 2937 3580 2939 2989 3586 3156 3246 3170 3268 3389 3381 2864 3740 3479
 3647 3716 3284 4204 3735 3218 3685 3704 3214 3394 3233 3352 3391]
[1530 1297 1335 1282 1590 1300 1400 1255 1355 1375 1340 1380 1355 1522
 1208 1405 1358 1292 1340 1400 1357 1287 1275 1270 1635 1505 1490 1485
 1310 1420 1318 1432 1364 1405 1432 1207 1375 1350 1236 1250 1350 1320
 1525 1570 1340 1422 1506 1215 1311 1300 1224 1350 1335 1390 1400 1225
 1310 1560 1330 1222 1415 1175 1330 1485 1470 1135 1310 1154 1510 1415
 1468 1390 1380 1432 1240 1195 1225 1188 1252 1315 1245 1430 1279 1245
 1309 1412 1120 1220 1280 1440 1370 1192 1230 1346 1290 1165 1240 1132
 1242 1270 1218 1430 1588 1320 1290 1260 1425 1226 1360 1620 1310 1250
 1295 1290 1290 1275 1250 1270 1362 1300 1173 1256 1440 1180 1306 1350
 1125 1165 1312 1300 1270 1335 1450 1310 1027 1235 1260 1165 1080 1127
 1270 1252 1200 1290 1334 1380 1140 1243 1340 1168 1322 1249 1321 1192
 1373 1170 1265 1235 1302 1241 1078 1520 1460 1075 1280 1180 1250 1190
 1374 1306 1202 1240 1316 1280 1350 1180 1210 1127 1324 1210 1290 1100
 1280 1175 1160 1205 1163 1022 1243 1350 1237 1204 1090 1355 1250 1076
 1120 1220 1240 1220 1095 1235 1105 1405 1150 1305 1220 1296 1175  955
 1070 1320 1060 1130 1250 1225 1180 1178 1142 1130 1185 1012 1280 1103
 1408 1300 1246 1380 1350 1060 1350 1220 1110 1215 1104 1170 1120]
```

taking mean of input data

In [4]:

```
xmean=np.mean(x)
xmean
```

Out[4]:

```
3633.9915611814345
```

taking mean of output data

In [5]:

```
ymean=np.mean(y)
ymean
```

Out[5]:

```
1282.873417721519
```

Size of input data

In [6]:

```
n=len(x)
n
```

Out[6]:

```
237
```

derieving b0 and b1 Cofficients

In [7]:

```
num=0
```

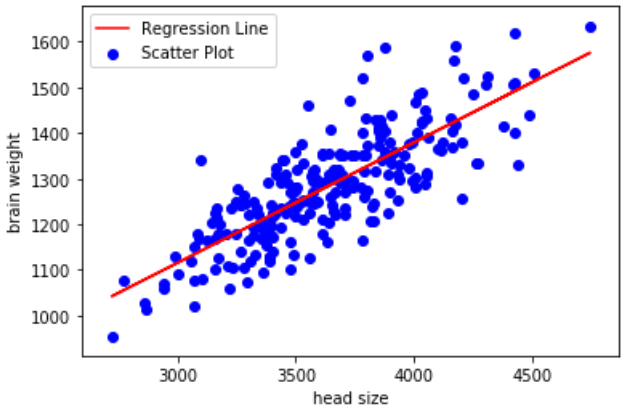
```
den=0
for i in range(n):
    num+=((x[i]-xmean)*(y[i]-ymean))
    den+=(x[i]-xmean)**2
b1=num/den
b0=ymean-(b1*xmean)
print(b1)
print(b0)
```

```
0.26342933948939945
325.57342104944223
```

Visualizing Realationship

In [8]:

```
xmax=np.max(x)+100
xmin=np.min(x)+100
X=np.linspace(xmin,xmax,1000)
yhat=b0+b1*x
plt.plot(x,yhat,color="red",label="Regression Line")
plt.scatter(x,y,color="blue",label="Scatter Plot")
plt.xlabel("head size")
plt.ylabel("brain weight")
plt.legend()
plt.show()
```



calculating Root mean Square Error value

In [9]:

```
    from sklearn.metrics import mean_squared_error
    from math import sqrt
    rmse = sqrt(mean_squared_error(y, yhat))
    print(rmse)
```

```
72.1206213783709
```

Function of predicting value of brain weight of different value of head size

In [10]:

```
def model_predction(x):
    b1=0.26342933948939945
    b0=325.57342104944223
    y_pred=b0+b1*x
    return y_pred
```

predicting value of brain weight of different value of head size

In [11]:

```
x=np.random.randint(100,10000,size=10)
for i in x:
    print("for head size=",i)
    print("The brain weight is",model_predction(i))
```

```
for head size= 5035
The brain weight is 1651.9401453785686
for head size= 6164
The brain weight is 1949.3518696621004
for head size= 216
The brain weight is 382.4741583791525
for head size= 3867
The brain weight is 1344.25467685495
for head size= 1651
The brain weight is 760.4952605464407
for head size= 6668
The brain weight is 2082.120256764758
for head size= 1415
The brain weight is 698.3259364269425
for head size= 4530
The brain weight is 1518.9083289364216
for head size= 1102
The brain weight is 615.8725531667604
for head size= 7212
The brain weight is 2225.425817446991
```

thank you