In [1]:
```
# import libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
x=pd.read_csv(r"C:\Users\user\Downloads\17_student_marks - 17_student_marks.cs
```

Out[2]:

	Student_ID	Test_1	Test_2	Test_3	Test_4	Test_5	Test_6	Test_7	Test_8	Test_9	Test_10
0	22000	78	87	91	91	88	98	94	100	100	100
1	22001	79	71	81	72	73	68	59	69	59	60
2	22002	66	65	70	74	78	86	87	96	88	82
3	22003	60	58	54	61	54	57	64	62	72	63
4	22004	99	95	96	93	97	89	92	98	91	98
5	22005	41	36	35	28	35	36	27	26	19	22
6	22006	47	50	47	57	62	64	71	75	85	87
7	22007	84	74	70	68	58	59	56	56	64	70
8	22008	74	64	58	57	53	51	47	45	42	43
9	22009	87	81	73	74	71	63	53	45	39	43
10	22010	40	34	37	33	31	35	39	38	40	48
11	22011	91	84	78	74	76	80	80	73	75	71
12	22012	81	83	93	88	89	90	99	99	95	85
13	22013	52	50	42	38	33	30	28	22	12	20
14	22014	63	67	65	74	80	86	95	96	92	83
15	22015	76	82	88	94	85	76	70	60	50	58
16	22016	83	78	71	71	77	72	66	75	66	61
17	22017	55	45	43	38	43	35	44	37	45	37
18	22018	71	67	76	74	64	61	57	64	61	51
19	22019	62	61	53	49	54	59	68	74	65	55
20	22020	44	38	36	34	26	34	39	44	36	45
21	22021	50	56	53	46	41	38	47	39	44	36
22	22022	57	48	40	45	43	36	26	19	9	12
23	22023	59	56	52	44	50	40	45	46	54	57
24	22024	84	92	89	80	90	80	84	74	68	73
25	22025	74	80	86	87	90	100	95	87	85	79
26	22026	92	84	74	83	93	83	75	82	81	73
27	22027	63	70	74	65	64	55	61	58	48	46
28	22028	78	77	69	76	78	74	67	69	78	68
29	22029	55	58	59	67	71	62	53	61	67	76
30	22030	54	54	48	38	35	45	46	47	41	37
31	22031	84	93	97	89	86	95	100	100	100	99
32	22032	95	100	94	100	98	99	100	90	80	84

	Student_ID	Test_1	Test_2	Test_3	Test_4	Test_5	Test_6	Test_7	Test_8	Test_9	Test_10
33	22033	64	61	63	73	63	68	64	58	50	51
34	22034	76	79	73	77	83	86	95	89	90	95
35	22035	78	71	61	55	54	48	41	32	41	40
36	22036	95	89	91	84	89	94	85	91	100	100
37	22037	99	89	79	87	87	81	82	74	64	54
38	22038	82	83	85	86	89	80	88	95	87	93
39	22039	65	56	64	62	58	51	61	68	70	70
40	22040	100	93	92	86	84	76	82	74	79	72
41	22041	78	72	73	79	81	73	71	77	83	92
42	22042	98	100	100	93	94	92	100	100	98	94
43	22043	58	62	67	77	71	63	64	73	83	76
44	22044	96	92	94	100	99	95	98	92	84	84
45	22045	86	87	85	84	85	91	86	82	85	87
46	22046	48	55	46	40	34	29	37	34	39	41
47	22047	56	52	54	47	40	35	43	44	40	39
48	22048	42	44	46	53	62	59	57	53	43	35
49	22049	64	54	49	59	54	55	57	59	63	73
50	22050	50	44	37	29	37	46	53	57	55	61
51	22051	70	60	70	62	67	67	68	67	72	69
52	22052	63	73	70	63	60	67	61	59	52	58
53	22053	92	100	100	100	100	100	92	87	94	100

In [3]:
```
x=x.head(10)
```

Out[3]:

	Student_ID	Test_1	Test_2	Test_3	Test_4	Test_5	Test_6	Test_7	Test_8	Test_9	Test_10
0	22000	78	87	91	91	88	98	94	100	100	100
1	22001	79	71	81	72	73	68	59	69	59	60
2	22002	66	65	70	74	78	86	87	96	88	82
3	22003	60	58	54	61	54	57	64	62	72	63
4	22004	99	95	96	93	97	89	92	98	91	98
5	22005	41	36	35	28	35	36	27	26	19	22
6	22006	47	50	47	57	62	64	71	75	85	87
7	22007	84	74	70	68	58	59	56	56	64	70
8	22008	74	64	58	57	53	51	47	45	42	43
9	22009	87	81	73	74	71	63	53	45	39	43

In [4]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Student_ID  10 non-null     int64
 1   Test_1      10 non-null     int64
 2   Test_2      10 non-null     int64
 3   Test_3      10 non-null     int64
 4   Test_4      10 non-null     int64
 5   Test_5      10 non-null     int64
 6   Test_6      10 non-null     int64
 7   Test_7      10 non-null     int64
 8   Test_8      10 non-null     int64
 9   Test_9      10 non-null     int64
 10  Test_10     10 non-null     int64
 11  Test_11     10 non-null     int64
 12  Test_12     10 non-null     int64
dtypes: int64(13)
memory usage: 1.1 KB
```

In [5]:

Out[5]:

```
Index(['Student_ID', 'Test_1', 'Test_2', 'Test_3', 'Test_4', 'Test_5',
       'Test_6', 'Test_7', 'Test_8', 'Test_9', 'Test_10', 'Test_11',
       'Test_12'],
      dtype='object')
```

In [6]:

```
d=x[['Student_ID', 'Test_1', 'Test_2', 'Test_3', 'Test_4', 'Test_5']]
```

Out[6]:

	Student_ID	Test_1	Test_2	Test_3	Test_4	Test_5
0	22000	78	87	91	91	88
1	22001	79	71	81	72	73
2	22002	66	65	70	74	78
3	22003	60	58	54	61	54
4	22004	99	95	96	93	97
5	22005	41	36	35	28	35
6	22006	47	50	47	57	62
7	22007	84	74	70	68	58
8	22008	74	64	58	57	53
9	22009	87	81	73	74	71

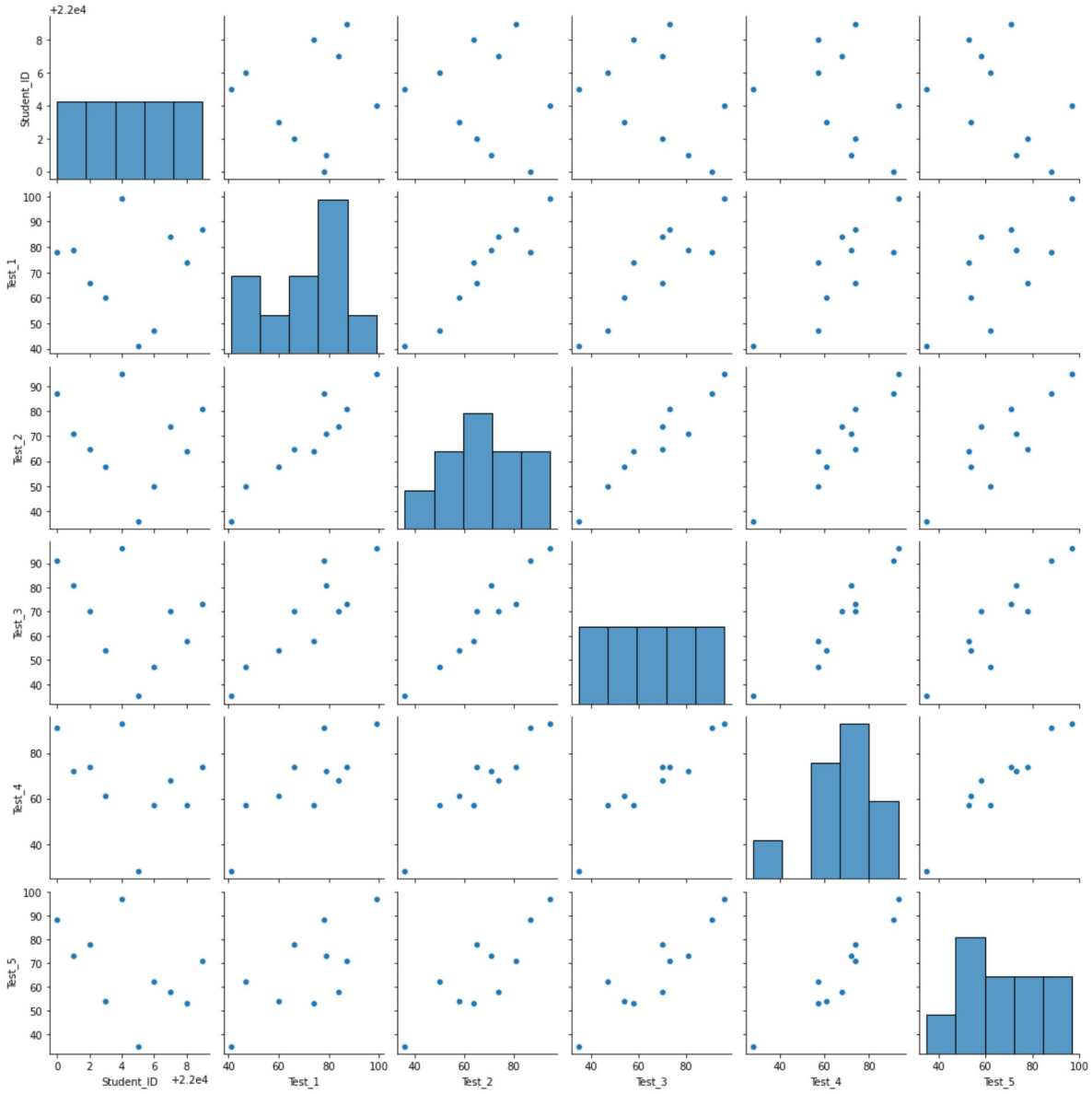
In [7]:

Out[7]:

	Student_ID	Test_1	Test_2	Test_3	Test_4	Test_5	Test_6	Test_7	
count	10.00000	10.000000	10.000000	10.000000	10.00000	10.00000	10.00000	10.000000	
mean	22004.50000	71.500000	68.100000	67.500000	67.50000	66.90000	67.10000	65.000000	
std	3.02765	18.106168	17.565433	19.259918	18.65029	18.28448	18.89415	21.395742	
min	22000.00000	41.000000	36.000000	35.000000	28.00000	35.00000	36.00000	27.000000	
25%	22002.25000	61.500000	59.500000	55.000000	58.00000	55.00000	57.50000	53.750000	
50%	22004.50000	76.000000	68.000000	70.000000	70.00000	66.50000	63.50000	61.500000	
75%	22006.75000	82.750000	79.250000	79.000000	74.00000	76.75000	81.50000	83.000000	
max	22009.00000	99.000000	95.000000	96.000000	93.00000	97.00000	98.00000	94.000000	1

In [8]:

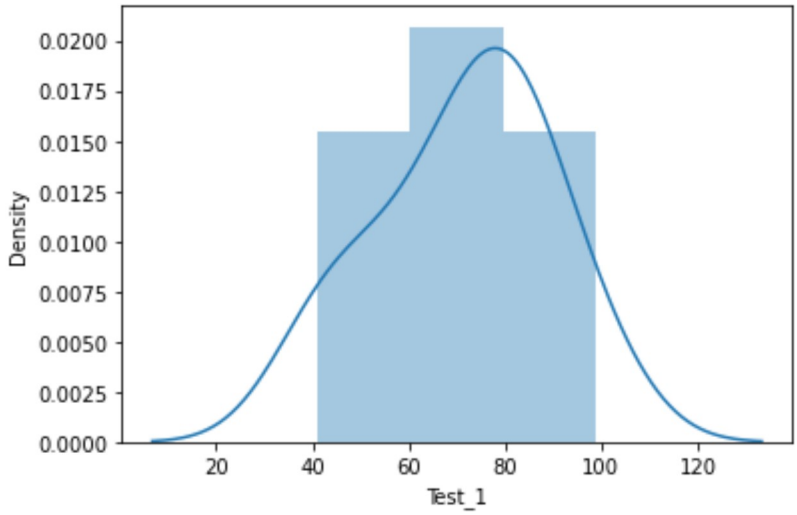
Out[8]: <seaborn.axisgrid.PairGrid at 0x18665cb9160>



In [9]:

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

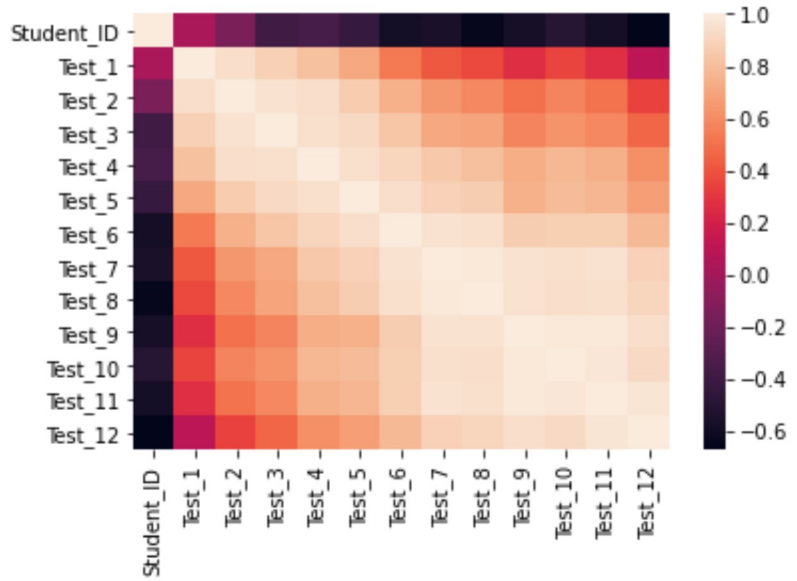
Out[9]: <AxesSubplot:xlabel='Test_1', ylabel='Density'>



In [10]:
```
x1=x[['Student_ID', 'Test_1', 'Test_2', 'Test_3', 'Test_4', 'Test_5',
      'Test_6', 'Test_7', 'Test_8', 'Test_9', 'Test_10', 'Test_11',
```

In [11]:

Out[11]: <AxesSubplot:>



In [12]:
```
x=x1[[  'Student_ID', 'Test_1', 'Test_2', 'Test_3', 'Test_4', 'Test_5',
       'Test_6', 'Test_7', 'Test_8', 'Test_9', 'Test_10', 'Test_11',
       'Test_12']]
```

In [13]:
```
# to split my dataset into traning and test date

from sklearn.model_selection import train_test_split
```

In [14]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Out[14]:
```
LinearRegression()
```

In [15]:

```
-2224.2475269460174
```

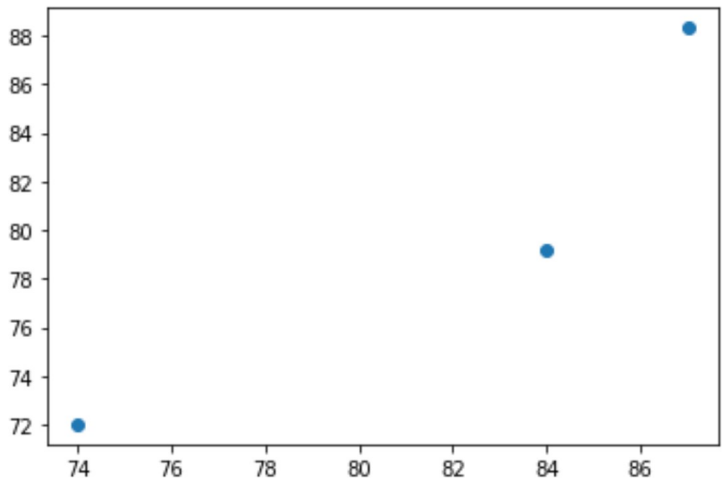
In [16]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[16]:

	Co-efficient
Student_ID	0.101224
Test_1	0.664669
Test_2	0.191991
Test_3	0.052605
Test_4	0.124526
Test_5	0.191652
Test_6	-0.284262
Test_7	0.118802
Test_8	0.066434
Test_9	-0.065366
Test_10	-0.128763
Test_11	0.014357
Test_12	0.019546

In [17]:
```
prediction=lr.predict(x_test)
```

Out[17]: <matplotlib.collections.PathCollection at 0x1866929d220>



In [18]:

Out[18]: 0.6909423794473857

In [19]:

Out[19]: 1.0

In [20]:

In [21]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[21]: 0.672903986179758

In [22]:
```
la=Lasso(alpha=10)
```

Out[22]: Lasso(alpha=10)

In [23]:

Out[23]: 0.9704021397186801

In [ ]: