```
import pandas as pd
import seaborn as sns


dt = pd.read_csv('/content/DATA (1).csv')


dt
```

	STUDENT ID	1	2	3	4	5	6	7	8	9	...	23	24	25	26	27	28	29	30	COURSE ID	GRADE
0	STUDENT1	2	2	3	3	1	2	2	1	1	...	1	1	3	2	1	2	1	1	1	1
1	STUDENT2	2	2	3	3	1	2	2	1	1	...	1	1	3	2	3	2	2	3	1	1
2	STUDENT3	2	2	2	3	2	2	2	2	4	...	1	1	2	2	1	1	2	2	1	1
3	STUDENT4	1	1	1	3	1	2	1	2	1	...	1	2	3	2	2	1	3	2	1	1
4	STUDENT5	2	2	1	3	2	2	1	3	1	...	2	1	2	2	2	1	2	2	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
140	STUDENT141	2	1	2	3	1	1	2	1	1	...	1	1	2	1	2	1	3	3	9	5
141	STUDENT142	1	1	2	4	2	2	2	1	4	...	1	1	3	2	2	1	5	3	9	5
142	STUDENT143	1	1	1	4	2	2	2	1	1	...	1	1	3	3	2	1	4	3	9	1
143	STUDENT144	2	1	2	4	1	1	1	5	2	...	2	1	2	1	2	1	5	3	9	4
144	STUDENT145	1	1	1	5	2	2	2	3	1	...	2	1	3	2	3	1	5	4	9	3

145 rows × 33 columns

```
dt.describe()
```

	1	2	3	4	5	6	7	8	9	
count	145.000000	145.000000	145.000000	145.000000	145.000000	145.000000	145.000000	145.000000	145.000000	145
mean	1.620690	1.600000	1.944828	3.572414	1.662069	1.600000	1.579310	1.627586	1.620690	1
std	0.613154	0.491596	0.537216	0.805750	0.474644	0.491596	0.495381	1.020245	1.061112	0
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
25%	1.000000	1.000000	2.000000	3.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
50%	2.000000	2.000000	2.000000	3.000000	2.000000	2.000000	2.000000	1.000000	1.000000	2
75%	2.000000	2.000000	2.000000	4.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2
max	3.000000	2.000000	3.000000	5.000000	2.000000	2.000000	2.000000	5.000000	4.000000	4

8 rows × 32 columns

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145 entries, 0 to 144
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   STUDENT ID  145 non-null    object
 1   1           145 non-null    int64
 2   2           145 non-null    int64
 3   3           145 non-null    int64
 4   4           145 non-null    int64
 5   5           145 non-null    int64
 6   6           145 non-null    int64
 7   7           145 non-null    int64
 8   8           145 non-null    int64
 9   9           145 non-null    int64
 10  10          145 non-null    int64
 11  11          145 non-null    int64
```

```
 12  12          145 non-null    int64
 13  13          145 non-null    int64
 14  14          145 non-null    int64
 15  15          145 non-null    int64
 16  16          145 non-null    int64
 17  17          145 non-null    int64
 18  18          145 non-null    int64
 19  19          145 non-null    int64
 20  20          145 non-null    int64
 21  21          145 non-null    int64
 22  22          145 non-null    int64
 23  23          145 non-null    int64
 24  24          145 non-null    int64
 25  25          145 non-null    int64
 26  26          145 non-null    int64
 27  27          145 non-null    int64
 28  28          145 non-null    int64
 29  29          145 non-null    int64
 30  30          145 non-null    int64
 31  COURSE ID   145 non-null    int64
 32  GRADE       145 non-null    int64
dtypes: int64(32), object(1)
memory usage: 37.5+ KB
```

```
from sklearn.model_selection import train_test_split
```

```
x=dt.drop(['GRADE','STUDENT ID'],axis=1)
y=dt['GRADE']
```

x

	1	2	3	4	5	6	7	8	9	10	...	22	23	24	25	26	27	28	29	30	COURSE ID
0	2	2	3	3	1	2	2	1	1	1	...	1	1	1	3	2	1	2	1	1	1
1	2	2	3	3	1	2	2	1	1	1	...	1	1	1	3	2	3	2	2	3	1
2	2	2	2	3	2	2	2	2	4	2	...	1	1	1	2	2	1	1	2	2	1
3	1	1	1	3	1	2	1	2	1	2	...	1	1	2	3	2	2	1	3	2	1
4	2	2	1	3	2	2	1	3	1	4	...	1	2	1	2	2	2	1	2	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
140	2	1	2	3	1	1	2	1	1	2	...	1	1	1	2	1	2	1	3	3	9
141	1	1	2	4	2	2	2	1	4	2	...	2	1	1	3	2	2	1	5	3	9
142	1	1	1	4	2	2	2	1	1	1	...	1	1	1	3	3	2	1	4	3	9
143	2	1	2	4	1	1	1	5	2	3	...	1	2	1	2	1	2	1	5	3	9
144	1	1	1	5	2	2	2	3	1	1	...	1	2	1	3	2	3	1	5	4	9

145 rows × 31 columns

y

	GRADE
0	1
1	1
2	1
3	1
4	1
...	...
140	5
141	5
142	1
143	4
144	3

145 rows × 1 columns

dtype: int64

```
dt = dt.drop(['STUDENT ID'],axis=1)
```

```
dt.corr()
```

	1	2	3	4	5	6	7	8	9	10	...
1	1.000000	0.138233	0.294426	-0.316506	-0.181019	0.138233	0.042572	0.127853	0.001472	-0.199257	...
2	0.138233	1.000000	0.126218	-0.206876	0.190476	0.224138	-0.125471	-0.022154	-0.013313	-0.154957	...
3	0.294426	0.126218	1.000000	-0.086965	-0.155331	0.073627	0.016557	-0.126440	0.023944	-0.018990	...
4	-0.316506	-0.206876	-0.086965	1.000000	0.055351	-0.154281	0.120346	-0.119024	0.028288	0.036543	...
5	-0.181019	0.190476	-0.155331	0.055351	1.000000	0.220238	0.100010	-0.118284	0.102223	-0.152641	...
6	0.138233	0.224138	0.073627	-0.154281	0.220238	1.000000	-0.011406	0.060922	0.026626	-0.227030	...
7	0.042572	-0.125471	0.016557	0.120346	0.100010	-0.011406	1.000000	-0.078556	0.037811	-0.168201	...
8	0.127853	-0.022154	-0.126440	-0.119024	-0.118284	0.060922	-0.078556	1.000000	-0.015926	0.030177	...
9	0.001472	-0.013313	0.023944	0.028288	0.102223	0.026626	0.037811	-0.015926	1.000000	0.327283	...
10	-0.199257	-0.154957	-0.018990	0.036543	-0.152641	-0.227030	-0.168201	0.030177	0.327283	1.000000	...
11	-0.068972	0.062370	-0.166336	-0.109006	0.153780	-0.122430	-0.043001	0.123932	0.168831	0.159529	...
12	-0.070108	0.120638	-0.134322	-0.222776	0.205138	-0.113252	0.045241	0.031212	0.090659	0.206439	...
13	0.069748	0.060216	0.051827	-0.037831	-0.134003	0.143273	0.033182	0.007866	-0.253099	-0.276875	...
14	0.172668	-0.028781	0.009990	-0.005450	-0.165492	-0.057563	-0.128034	0.101380	-0.033565	-0.059119	...
15	-0.200812	-0.056143	-0.050269	0.023919	-0.044113	-0.161412	-0.211091	0.028451	0.274118	0.384892	...
16	-0.141571	-0.033997	-0.131680	-0.012785	0.005919	0.019123	-0.176901	-0.109689	-0.116259	0.016447	...
17	-0.172833	-0.067750	-0.174719	0.003758	-0.114824	-0.129342	0.048897	-0.097935	0.007134	-0.021241	...
18	0.059712	0.045206	-0.010144	0.008877	-0.096332	-0.281283	0.090581	-0.072357	-0.046942	-0.081128	...
19	0.078849	-0.005233	0.050529	0.061543	-0.171384	-0.214557	-0.004119	0.009390	-0.124146	-0.155241	...
20	-0.226882	-0.363971	-0.009101	0.214838	-0.018640	-0.020602	0.103634	-0.106800	0.059791	0.179520	...
21	0.007306	-0.144138	0.124317	0.041443	-0.120973	-0.024023	0.038636	0.059872	0.070998	0.001039	...
22	0.060027	-0.065795	-0.032180	-0.141190	0.028198	0.098693	0.154225	0.111506	-0.056759	-0.135860	...
23	-0.044462	-0.032164	-0.027258	0.027357	-0.153273	-0.124062	-0.191197	0.213079	-0.015048	0.132236	...
24	0.030595	-0.013833	-0.053034	0.005529	-0.246770	-0.117580	-0.031005	0.048955	-0.094471	0.009870	...
25	-0.181121	0.065013	0.053969	0.164447	-0.033757	-0.035007	0.005818	-0.235900	0.022769	0.066609	...
26	0.235576	-0.184238	0.123405	0.018147	-0.158217	0.004187	-0.054734	-0.050648	-0.019064	0.146396	...
27	-0.100812	0.065449	-0.082461	0.147922	0.078973	-0.028050	-0.000480	-0.031225	-0.134057	-0.024462	...
28	0.228884	0.223095	0.055107	-0.095411	0.027762	0.135949	-0.082664	-0.037184	-0.158432	-0.136949	...
29	0.172597	0.241033	0.019802	0.236460	0.034666	0.002171	-0.058479	-0.100950	-0.036078	-0.082775	...
30	0.121441	0.292842	0.053497	0.243517	0.007706	-0.092476	-0.012658	-0.058644	0.005909	-0.007332	...
COURSE ID	-0.304994	-0.430704	-0.118761	0.317563	-0.114794	-0.326711	0.068769	-0.123024	0.114839	0.271997	...
GRADE	-0.095251	0.335533	0.104821	0.023963	0.167445	-0.062993	-0.051778	-0.166352	-0.156289	0.023683	...

32 rows × 32 columns

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25)
```

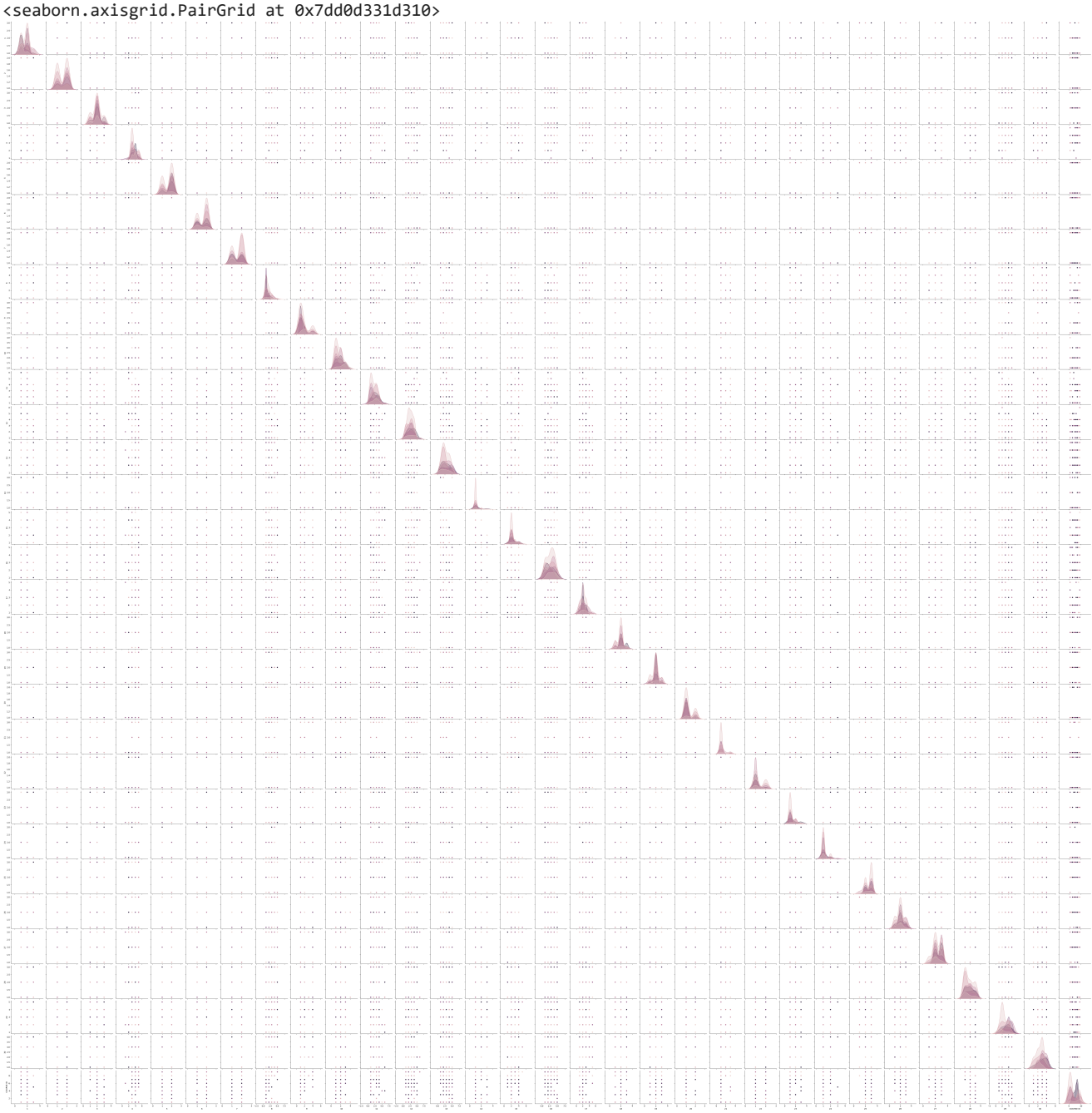
```
xtest
```

	1	2	3	4	5	6	7	8	9	10	...	22	23	24	25	26	27	28	29	30	COURSE ID
17	2	2	2	3	2	2	2	1	1	1	...	2	1	1	2	2	2	2	2	2	1
5	2	2	2	3	2	2	2	2	1	1	...	1	1	1	1	2	1	2	4	4	1
33	2	1	2	3	1	2	1	1	1	1	...	1	1	1	1	3	2	2	2	3	1
77	1	2	1	2	2	2	1	2	2	2	...	2	3	1	2	2	2	2	1	1	4
68	2	1	2	4	1	2	2	1	1	1	...	2	1	1	2	2	3	2	4	3	3
54	2	2	2	3	2	2	2	3	4	2	...	1	1	2	3	1	3	1	5	3	1
23	3	2	2	2	1	1	2	5	1	1	...	2	1	1	3	2	3	3	3	3	1
135	2	1	2	3	1	1	2	1	4	2	...	1	1	1	3	3	2	1	3	2	9
25	2	2	2	3	2	2	1	1	1	2	...	1	1	1	2	1	3	2	1	2	1
108	2	1	1	5	2	1	2	2	2	1	...	2	1	1	2	3	1	2	3	3	7
76	2	2	1	2	2	1	2	2	1	1	...	1	3	2	2	1	2	3	2	2	4
69	2	1	2	4	2	2	1	1	1	1	...	1	2	1	3	2	3	1	3	2	3
87	2	2	2	3	2	1	2	2	1	1	...	1	2	1	2	2	2	2	1	1	6
74	1	2	2	4	2	2	2	1	1	1	...	1	2	1	3	2	3	2	3	3	3
18	1	1	2	4	2	2	2	3	1	1	...	2	1	1	3	1	3	3	3	3	1
127	1	1	2	4	2	2	2	1	4	3	...	1	1	1	3	2	2	1	2	2	9
131	1	1	1	5	2	1	2	1	2	2	...	1	1	2	3	2	3	1	5	3	9
132	1	1	1	5	2	1	2	1	1	2	...	1	2	1	3	2	3	1	5	4	9
11	1	1	1	4	1	1	2	4	2	3	...	1	3	2	3	1	3	3	4	3	1
48	1	2	2	3	2	1	1	1	1	2	...	1	1	1	3	2	3	2	2	2	1
130	1	1	2	3	1	1	2	1	1	1	...	1	1	2	3	1	3	2	2	3	9
133	1	1	2	5	2	2	1	1	1	1	...	1	1	1	3	3	3	2	5	3	9
140	2	1	2	3	1	1	2	1	1	2	...	1	1	1	2	1	2	1	3	3	9
35	1	2	1	4	2	2	2	3	1	1	...	2	1	1	2	1	3	1	1	1	1
22	2	2	2	3	1	2	1	1	1	1	...	1	1	2	3	1	2	3	3	3	1
94	2	2	2	3	2	2	1	1	1	1	...	1	1	1	3	2	2	3	5	4	6
7	1	1	2	3	1	1	1	2	2	3	...	1	3	1	3	2	2	1	1	1	1
96	1	2	2	4	2	1	1	1	2	3	...	1	1	1	3	2	3	2	2	3	7
8	2	1	3	3	2	1	1	1	1	3	...	1	1	1	3	2	2	2	4	3	1
27	1	2	1	3	1	2	2	1	1	1	...	2	1	1	3	1	2	1	2	1	1
89	2	2	2	3	2	2	2	1	1	1	...	2	1	1	3	1	3	3	3	2	6
10	1	1	1	3	2	2	2	3	2	3	...	2	1	1	2	2	2	2	1	1	1
21	1	2	2	5	2	2	1	1	4	2	...	1	1	1	3	1	3	3	3	3	1
40	1	2	1	3	2	2	2	2	1	1	...	1	1	1	3	1	2	1	2	3	1
95	1	2	3	5	1	1	1	1	2	3	...	1	1	1	3	2	2	1	4	3	7
43	1	2	2	3	2	2	1	1	1	1	...	2	1	1	3	1	3	3	3	2	1
104	1	1	2	3	2	2	2	1	2	2	...	1	1	1	2	3	2	2	1	1	7

37 rows × 31 columns

```
sns.pairplot(dt,hue='GRADE')
```

```
<seaborn.axisgrid.PairGrid at 0x7dd0d331d310>
```



```
from sklearn.linear_model import LogisticRegression

lr=LogisticRegression()
```

Start coding or generate with AI.

Start coding or generate with AI.

+ Code + Text

Start coding or generate with AI.

Start coding or generate with AI.