In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
Gr=pd.read_excel("google_review_ratings.csv.xlsx")
```

In [3]:
```
Gr
```

Out[3]:

	User	Category 1	Category 2	Category 3	Category 4	Category 5	Category 6	Category 7	Category 8	Category 9	...	Category 16	Category 17	Category 18	Cat
0	User 1	0.00	0.00	3.63	3.65	5.00	2.92	5.00	2.35	2.33	...	0.59	0.50	0.00	
1	User 2	0.00	0.00	3.63	3.65	5.00	2.92	5.00	2.64	2.33	...	0.59	0.50	0.00	
2	User 3	0.00	0.00	3.63	3.63	5.00	2.92	5.00	2.64	2.33	...	0.59	0.50	0.00	
3	User 4	0.00	0.50	3.63	3.63	5.00	2.92	5.00	2.35	2.33	...	0.59	0.50	0.00	
4	User 5	0.00	0.00	3.63	3.63	5.00	2.92	5.00	2.64	2.33	...	0.59	0.50	0.00	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5451	User 5452	0.91	5.00	4.00	2.79	2.77	2.57	2.43	1.09	1.77	...	0.66	0.65	0.66	

In [4]:
```
Gr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5456 entries, 0 to 5455
Data columns (total 26 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   User         5456 non-null   object 
 1   Category 1   5456 non-null   float64
 2   Category 2   5456 non-null   float64
 3   Category 3   5456 non-null   float64
 4   Category 4   5456 non-null   float64
 5   Category 5   5456 non-null   float64
 6   Category 6   5456 non-null   float64
 7   Category 7   5456 non-null   float64
 8   Category 8   5456 non-null   float64
 9   Category 9   5456 non-null   float64
 10  Category 10  5456 non-null   float64
 11  Category 11  5456 non-null   object 
 12  Category 12  5455 non-null   float64
 13  Category 13  5456 non-null   float64
 14  Category 14  5456 non-null   float64
 15  Category 15  5456 non-null   float64
 16  Category 16  5456 non-null   float64
 17  Category 17  5456 non-null   float64
 18  Category 18  5456 non-null   float64
 19  Category 19  5456 non-null   float64
 20  Category 20  5456 non-null   float64
 21  Category 21  5456 non-null   float64
 22  Category 22  5456 non-null   float64
 23  Category 23  5456 non-null   float64
 24  Category 24  5455 non-null   float64
 25  Unnamed: 25  2 non-null      float64
dtypes: float64(24), object(2)
memory usage: 1.1+ MB
```

In [5]:
```
bb=Gr.select_dtypes(include=["object"])
```

In [6]:
```
bb
```

Out[6]:

	User	Category 11
0	User 1	1.7
1	User 2	1.7
2	User 3	1.7
3	User 4	1.73
4	User 5	1.7
...	...	...
5451	User 5452	1.02
5452	User 5453	1.01
5453	User 5454	0.99
5454	User 5455	0.97
5455	User 5456	0.95

5456 rows × 2 columns

In [7]:
```
Gr.drop(["User","Unnamed: 25"],axis=1,inplace=True)
```

In [8]:
```
Gr.replace({"2\t2.":0.92},inplace=True)
```

In [9]:
```
Gr.iloc[2712:,10:]
```

Out[9]:

	Category 11	Category 12	Category 13	Category 14	Category 15	Category 16	Category 17	Category 18	Category 19	Category 20	Category 21	Category 22	Category 23	Categ
2712	0.92	NaN	0.84	5.00	5.00	1.08	1.10	1.04	5.00	4.43	5.00	5.00	5.00	2
2713	0.82	0.83	5.00	5.00	1.07	1.09	0.99	5.00	5.00	3.74	2.62	5.00	2.58	2
2714	0.81	0.82	5.00	5.00	1.06	1.08	0.98	5.00	5.00	5.00	5.00	2.61	2.59	2
2715	0.81	0.83	5.00	5.00	1.06	1.07	0.98	1.03	4.44	5.00	5.00	5.00	2.60	2
2716	0.82	0.82	0.87	5.00	1.06	1.07	1.95	1.03	5.00	3.78	3.74	5.00	2.59	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5451	1.02	1.06	1.29	1.29	5.00	0.66	0.65	0.66	0.69	5.00	1.05	5.00	5.00	1
5452	1.01	1.04	1.27	1.27	0.89	0.65	0.64	0.65	1.59	1.62	1.06	5.00	5.00	1
5453	0.99	1.00	1.08	1.25	0.87	0.65	0.63	0.64	0.74	5.00	1.07	5.00	5.00	1
5454	0.97	0.98	1.06	1.23	5.00	0.64	0.63	0.64	0.75	5.00	1.08	5.00	5.00	1
5455	0.95	0.96	1.01	1.21	0.85	0.64	0.62	0.63	0.78	5.00	1.08	5.00	5.00	1

2744 rows × 14 columns

In [10]:
```
Gr["Category 11"]=pd.to_numeric(Gr["Category 11"],downcast="float")
```

In [11]:
```
Gr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5456 entries, 0 to 5455
Data columns (total 24 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Category 1   5456 non-null   float64
 1   Category 2   5456 non-null   float64
 2   Category 3   5456 non-null   float64
 3   Category 4   5456 non-null   float64
 4   Category 5   5456 non-null   float64
 5   Category 6   5456 non-null   float64
 6   Category 7   5456 non-null   float64
 7   Category 8   5456 non-null   float64
 8   Category 9   5456 non-null   float64
 9   Category 10  5456 non-null   float64
 10  Category 11  5456 non-null   float32
 11  Category 12  5455 non-null   float64
 12  Category 13  5456 non-null   float64
 13  Category 14  5456 non-null   float64
 14  Category 15  5456 non-null   float64
 15  Category 16  5456 non-null   float64
 16  Category 17  5456 non-null   float64
 17  Category 18  5456 non-null   float64
 18  Category 19  5456 non-null   float64
 19  Category 20  5456 non-null   float64
 20  Category 21  5456 non-null   float64
 21  Category 22  5456 non-null   float64
 22  Category 23  5456 non-null   float64
 23  Category 24  5455 non-null   float64
dtypes: float32(1), float64(23)
memory usage: 1001.8 KB
```

In [12]:
```
Gr.describe()
```

Out[12]:

	Category 1	Category 2	Category 3	Category 4	Category 5	Category 6	Category 7	Category 8	Category 9	Category 10	...	Cate
count	5456.000000	5456.000000	5456.000000	5456.000000	5456.000000	5456.00000	5456.000000	5456.000000	5456.000000	5456.000000	...	5456
mean	1.455720	2.319707	2.489331	2.796886	2.958941	2.89349	3.351395	2.540795	3.126019	2.832729	...	2
std	0.827604	1.421438	1.247815	1.309159	1.339056	1.28240	1.413492	1.111391	1.356802	1.307665	...	1
min	0.000000	0.000000	0.000000	0.830000	1.120000	1.11000	1.120000	0.860000	0.840000	0.810000	...	0
25%	0.920000	1.360000	1.540000	1.730000	1.770000	1.79000	1.930000	1.620000	1.800000	1.640000	...	0
50%	1.340000	1.905000	2.060000	2.460000	2.670000	2.68000	3.230000	2.170000	2.800000	2.680000	...	1
75%	1.810000	2.682500	2.740000	4.092500	4.312500	3.84000	5.000000	3.190000	5.000000	3.530000	...	4
max	5.000000	5.000000	5.000000	5.000000	5.000000	5.00000	5.000000	5.000000	5.000000	5.000000	...	5

8 rows × 24 columns

In [13]:
```
Gr_corr=Gr.corr()
Gr_corr
```

Out[13]:

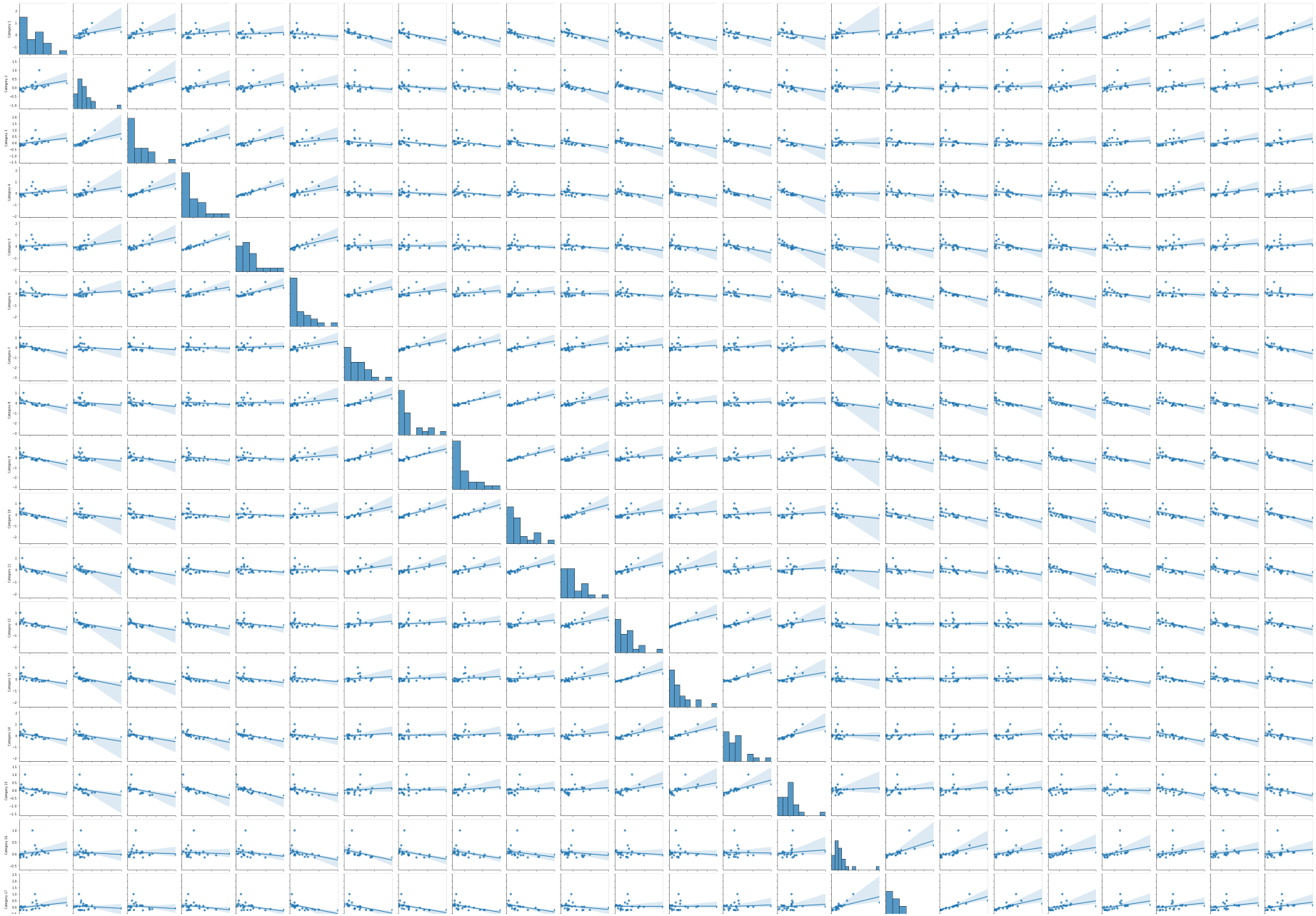
	Category 1	Category 2	Category 3	Category 4	Category 5	Category 6	Category 7	Category 8	Category 9	Category 10	...	Category 15	Category 16	Cate
Category 1	1.000000	0.248991	0.148805	0.070692	0.035451	-0.093143	-0.264632	-0.181267	-0.290214	-0.274683	...	-0.134914	0.067285	0.13
Category 2	0.248991	1.000000	0.325011	0.167338	0.153520	0.053916	-0.050158	-0.004500	-0.050550	-0.092526	...	-0.066670	-0.033119	-0.07
Category 3	0.148805	0.325011	1.000000	0.396786	0.329925	0.161242	-0.072713	-0.186308	-0.219842	-0.179090	...	-0.134203	-0.022249	-0.08
Category 4	0.070692	0.167338	0.396786	1.000000	0.626868	0.315354	-0.067914	-0.128064	-0.169442	-0.115515	...	-0.272130	0.018274	-0.13
Category 5	0.035451	0.153520	0.329925	0.626868	1.000000	0.489937	0.077807	-0.002595	-0.170430	-0.100191	...	-0.323902	-0.056347	-0.18
Category 6	-0.093143	0.053916	0.161242	0.315354	0.489937	1.000000	0.382774	0.200528	0.114719	-0.019007	...	-0.187512	-0.149295	-0.22
Category 7	-0.264632	-0.050158	-0.072713	-0.067914	0.077807	0.382774	1.000000	0.406959	0.432054	0.256563	...	0.092527	-0.142861	-0.20
Category 8	-0.181267	-0.004500	-0.186308	-0.128064	-0.002595	0.200528	0.406959	1.000000	0.536870	0.551409	...	-0.064692	-0.124417	-0.20
Category 9	-0.290214	-0.050550	-0.219842	-0.169442	-0.170430	0.114719	0.432054	0.536870	1.000000	0.564086	...	0.126720	-0.119844	-0.22
Category 10	-0.274683	-0.092526	-0.179090	-0.115515	-0.100191	-0.019007	0.256563	0.551409	0.564086	1.000000	...	0.038084	-0.031927	-0.21
Category 11	-0.214774	-0.221433	-0.157344	-0.116865	-0.123990	-0.147989	0.098364	0.292642	0.262236	0.469737	...	-0.027211	-0.001310	-0.04
Category 12	-0.261190	-0.157070	-0.235708	-0.166140	-0.104966	-0.163945	0.030878	0.003250	-0.012590	0.128420	...	0.150238	-0.047281	0.02
Category 13	-0.178666	-0.213747	-0.182322	-0.145093	-0.090369	-0.136938	0.025643	-0.010413	0.019325	0.066222	...	0.197768	-0.048572	0.02
Category 14	-0.236966	-0.126713	-0.162605	-0.308164	-0.280767	-0.153427	0.089101	-0.021850	0.032260	-0.001803	...	0.371021	-0.006312	0.07

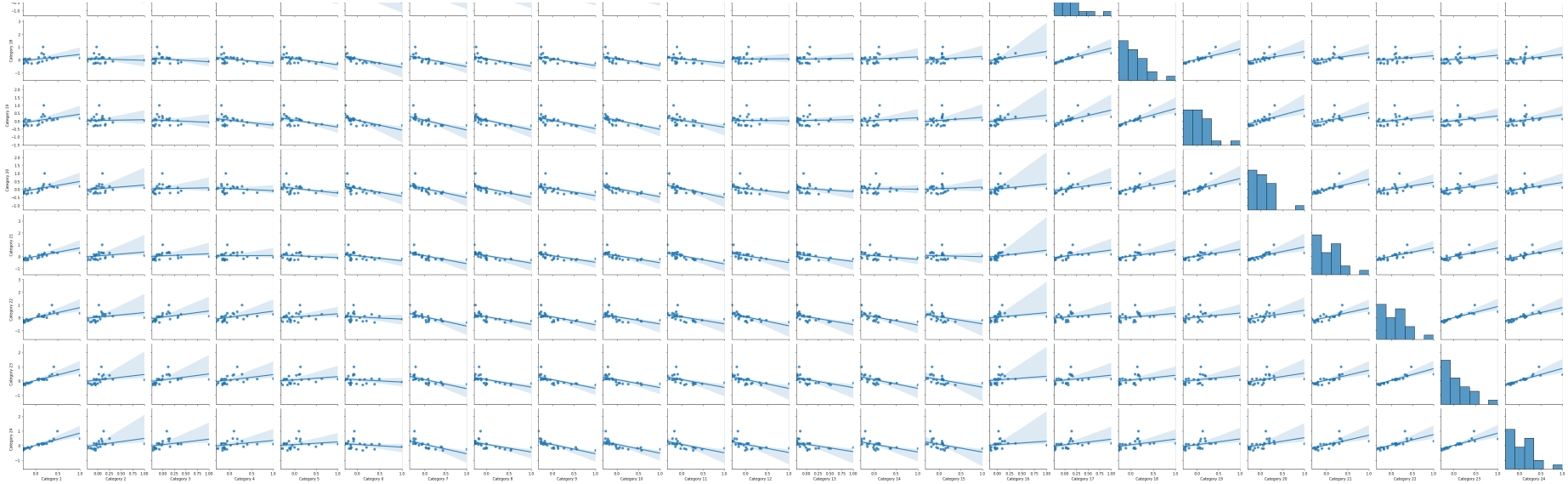
	Category 1	Category 2	Category 3	Category 4	Category 5	Category 6	Category 7	Category 8	Category 9	Category 10	...	Category 15	Category 16	Cate
Category 15	-0.134914	-0.066670	-0.134203	-0.272130	-0.323902	-0.187512	0.092527	-0.064692	0.126720	0.038084	...	1.000000	0.099339	0.06
Category 16	0.067285	-0.033119	-0.022249	0.018274	-0.056347	-0.149295	-0.142861	-0.124417	-0.119844	-0.031927	...	0.099339	1.000000	0.37
Category 17	0.130099	-0.077146	-0.084643	-0.132468	-0.184345	-0.229604	-0.207275	-0.202950	-0.229585	-0.212346	...	0.062586	0.371238	1.00
Category 18	0.160360	-0.025446	-0.119810	-0.183849	-0.243357	-0.271354	-0.226475	-0.236106	-0.266700	-0.273837	...	0.078407	0.217081	0.51
Category 19	0.180271	0.032529	-0.075402	-0.192665	-0.256987	-0.266608	-0.272979	-0.281018	-0.271881	-0.324115	...	0.065196	0.019839	0.27
Category 20	0.199397	0.096974	-0.009202	-0.092453	-0.191997	-0.227362	-0.232408	-0.254393	-0.164033	-0.247502	...	0.071242	0.062146	0.08
Category 21	0.309238	0.088990	0.001105	-0.056226	-0.133433	-0.197454	-0.257728	-0.274733	-0.185958	-0.230389	...	0.059684	0.155171	0.16
Category 22	0.364375	0.018366	0.134571	0.276727	0.124431	-0.090481	-0.360287	-0.255520	-0.268053	-0.183660	...	-0.179217	0.099088	0.10
Category 23	0.407033	0.077616	0.117458	0.173550	0.128055	-0.081109	-0.223080	-0.167927	-0.268314	-0.213030	...	-0.163722	0.050828	0.13
Category 24	0.489260	0.125172	0.082669	0.090533	0.098853	-0.066423	-0.247569	-0.137292	-0.326574	-0.263505	...	-0.210699	0.027896	0.14

24 rows × 24 columns

In [14]:
```
sns.pairplot(Gr_corr,kind="reg")
```

Out[14]: <seaborn.axisgrid.PairGrid at 0x24746567ee0>





In [15]:
```
Gr[:]=Gr[:].fillna(0)
```

In [16]:
```
from sklearn.preprocessing import normalize
data_normalize=normalize(Gr)
```

In [17]:
```
Gr.iloc[2712:,10:].head()
```

Out[17]:

	Category 11	Category 12	Category 13	Category 14	Category 15	Category 16	Category 17	Category 18	Category 19	Category 20	Category 21	Category 22	Category 23	Categ
2712	0.92	0.00	0.84	5.0	5.00	1.08	1.10	1.04	5.00	4.43	5.00	5.00	5.00	2
2713	0.82	0.83	5.00	5.0	1.07	1.09	0.99	5.00	5.00	3.74	2.62	5.00	2.58	2
2714	0.81	0.82	5.00	5.0	1.06	1.08	0.98	5.00	5.00	5.00	5.00	2.61	2.59	2
2715	0.81	0.83	5.00	5.0	1.06	1.07	0.98	1.03	4.44	5.00	5.00	5.00	2.60	2
2716	0.82	0.82	0.87	5.0	1.06	1.07	1.95	1.03	5.00	3.78	3.74	5.00	2.59	2

In [18]:
```
data_normalize
```

Out[18]:
```
array([[0.        , 0.        , 0.33357969, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.33148681, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.3317615 , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.0749856 , 0.39885955, 0.3214808 , ..., 0.39885955, 0.39885955,
        0.08854682],
       [0.07238949, 0.30860782, 0.30860782, ..., 0.38099731, 0.38099731,
        0.0853434 ],
       [0.07508624, 0.32168526, 0.39519074, ..., 0.39519074, 0.39519074,
        0.09247463]])
```

In [19]:
```
data_normalized_one=pd.DataFrame(data_normalize)
```

In [20]:
```
data_normalized_one
```

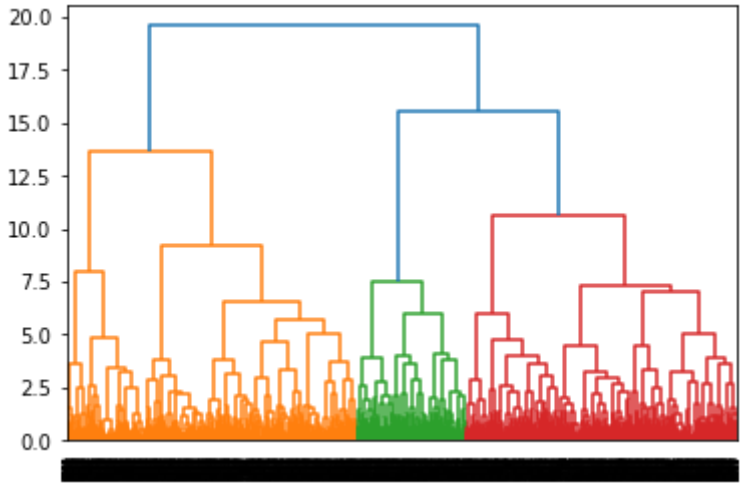
Out[20]:

	0	1	2	3	4	5	6	7	8	9	...	14	15	16	
0	0.000000	0.000000	0.333580	0.335418	0.459476	0.268334	0.459476	0.215954	0.214116	0.242603	...	0.159898	0.054218	0.045948	0.0
1	0.000000	0.000000	0.331487	0.333313	0.456593	0.266651	0.456593	0.241081	0.212773	0.241995	...	0.158895	0.053878	0.045659	0.0
2	0.000000	0.000000	0.331762	0.331762	0.456972	0.266872	0.456972	0.241281	0.212949	0.241281	...	0.159026	0.053923	0.045697	0.0
3	0.000000	0.045907	0.333288	0.333288	0.459075	0.268100	0.459075	0.215765	0.213929	0.242391	...	0.159758	0.054171	0.045907	0.0
4	0.000000	0.000000	0.331762	0.331762	0.456972	0.266872	0.456972	0.241281	0.212949	0.241281	...	0.159026	0.053923	0.045697	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5451	0.066725	0.366622	0.293298	0.204575	0.203109	0.188444	0.178178	0.079924	0.129784	0.076257	...	0.366622	0.048394	0.047661	0.0
5452	0.079340	0.426561	0.342955	0.238021	0.237168	0.219253	0.151003	0.091284	0.150150	0.087019	...	0.075928	0.055453	0.054600	0.0
5453	0.074986	0.398860	0.321481	0.223361	0.221766	0.205014	0.139601	0.083761	0.139601	0.079772	...	0.069402	0.051852	0.050256	0.0
5454	0.072389	0.308608	0.308608	0.214120	0.212596	0.185927	0.134111	0.078485	0.132587	0.074675	...	0.380997	0.048768	0.048006	0.0
5455	0.075086	0.321685	0.395191	0.222888	0.221307	0.203128	0.191272	0.080619	0.137526	0.075877	...	0.067182	0.050584	0.049004	0.0

5456 rows × 24 columns

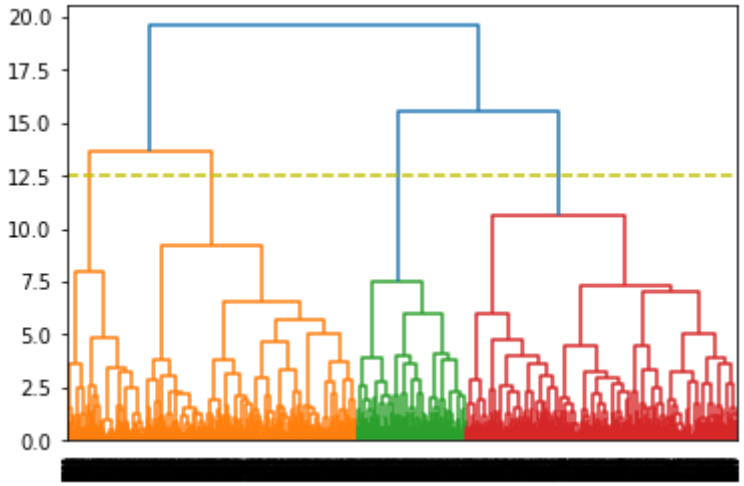
In [21]:
```
import scipy.cluster.hierarchy as HCluster
```

In [22]:
```
dendrogram=HCluster.dendrogram(HCluster.linkage(data_normalized_one,method="ward"))
```



In [23]:
```
dendrogram=HCluster.dendrogram(HCluster.linkage(data_normalized_one,method="ward"))
plt.axhline(y=12.5,color="y",linestyle="--")
```

Out[23]:
```
<matplotlib.lines.Line2D at 0x2476254fc40>
```



In [25]:
```
from sklearn.cluster import AgglomerativeClustering as ag
```

In [37]:
```
cluster=ag(n_clusters=4)
y_cluster=cluster.fit_predict(data_normalized_one)
```

In [39]:
```
Gr["y_cluster"]=y_cluster
```

In [40]:
```
Gr
```

Out[40]:

Category 7	Category 8	Category 9	Category 10	...	Category 16	Category 17	Category 18	Category 19	Category 20	Category 21	Category 22	Category 23	Category 24	y_cluster
5.00	2.35	2.33	2.64	...	0.59	0.50	0.00	0.50	0.00	0.00	0.0	0.0	0.00	0
5.00	2.64	2.33	2.65	...	0.59	0.50	0.00	0.50	0.00	0.00	0.0	0.0	0.00	0
5.00	2.64	2.33	2.64	...	0.59	0.50	0.00	0.50	0.00	0.00	0.0	0.0	0.00	0
5.00	2.35	2.33	2.64	...	0.59	0.50	0.00	0.50	0.00	0.00	0.0	0.0	0.00	0
5.00	2.64	2.33	2.64	...	0.59	0.50	0.00	0.50	0.00	0.00	0.0	0.0	0.00	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2.43	1.09	1.77	1.04	...	0.66	0.65	0.66	0.69	5.00	1.05	5.0	5.0	1.56	2
1.77	1.07	1.76	1.02	...	0.65	0.64	0.65	1.59	1.62	1.06	5.0	5.0	1.09	0
1.75	1.05	1.75	1.00	...	0.65	0.63	0.64	0.74	5.00	1.07	5.0	5.0	1.11	2
1.76	1.03	1.74	0.98	...	0.64	0.63	0.64	0.75	5.00	1.08	5.0	5.0	1.12	2
2.42	1.02	1.74	0.96	...	0.64	0.62	0.63	0.78	5.00	1.08	5.0	5.0	1.17	2

so i choosed 4 as no.of.clusters and predicted

In [ ]:
```

```