```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
\textbf{from sklearn.decomposition import} \ \texttt{PCA}
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.model selection import train test split
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.model selection import cross val score
import warnings
warnings.filterwarnings('ignore')
In [2]:
sonar=pd.read csv('sonar.csv')
In [3]:
sonar
Out[3]:
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	 V52	V53	V54	V55	V56	V57
0	0.0200	0.0371	0.0428	0.0207	0.0954	0.0986	0.1539	0.1601	0.3109	0.2111	 0.0027	0.0065	0.0159	0.0072	0.0167	0.0180
1	0.0453	0.0523	0.0843	0.0689	0.1183	0.2583	0.2156	0.3481	0.3337	0.2872	 0.0084	0.0089	0.0048	0.0094	0.0191	0.0140
2	0.0262	0.0582	0.1099	0.1083	0.0974	0.2280	0.2431	0.3771	0.5598	0.6194	 0.0232	0.0166	0.0095	0.0180	0.0244	0.0316
3	0.0100	0.0171	0.0623	0.0205	0.0205	0.0368	0.1098	0.1276	0.0598	0.1264	 0.0121	0.0036	0.0150	0.0085	0.0073	0.0050
4	0.0762	0.0666	0.0481	0.0394	0.0590	0.0649	0.1209	0.2467	0.3564	0.4459	 0.0031	0.0054	0.0105	0.0110	0.0015	0.0072
203	0.0187	0.0346	0.0168	0.0177	0.0393	0.1630	0.2028	0.1694	0.2328	0.2684	 0.0116	0.0098	0.0199	0.0033	0.0101	0.0065
204	0.0323	0.0101	0.0298	0.0564	0.0760	0.0958	0.0990	0.1018	0.1030	0.2154	 0.0061	0.0093	0.0135	0.0063	0.0063	0.0034
205	0.0522	0.0437	0.0180	0.0292	0.0351	0.1171	0.1257	0.1178	0.1258	0.2529	 0.0160	0.0029	0.0051	0.0062	0.0089	0.0140
206	0.0303	0.0353	0.0490	0.0608	0.0167	0.1354	0.1465	0.1123	0.1945	0.2354	 0.0086	0.0046	0.0126	0.0036	0.0035	0.0034
207	0.0260	0.0363	0.0136	0.0272	0.0214	0.0338	0.0655	0.1400	0.1843	0.2354	 0.0146	0.0129	0.0047	0.0039	0.0061	0.0040

208 rows × 61 columns

In [4]:

sonar.describe()

Out[4]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	
count	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	20
mean	0.029164	0.038437	0.043832	0.053892	0.075202	0.104570	0.121747	0.134799	0.178003	0.208259	
std	0.022991	0.032960	0.038428	0.046528	0.055552	0.059105	0.061788	0.085152	0.118387	0.134416	
min	0.001500	0.000600	0.001500	0.005800	0.006700	0.010200	0.003300	0.005500	0.007500	0.011300	
25%	0.013350	0.016450	0.018950	0.024375	0.038050	0.067025	0.080900	0.080425	0.097025	0.111275	
50%	0.022800	0.030800	0.034300	0.044050	0.062500	0.092150	0.106950	0.112100	0.152250	0.182400	
75%	0.035550	0.047950	0.057950	0.064500	0.100275	0.134125	0.154000	0.169600	0.233425	0.268700	
	0.407400	0 000000	0.000000	0.400400	0.404000	0.00000	0.070000	0.450000	0.00000	0.740000	

```
U.3U59UU
V3
                                        U.4264UU
V4
                                                                                  U.459UUU
V8
                                                                                            ∪.ხ828∪∪
V9
                                                                                                       U.71U6UU ...
V10 ...
         บ.า37 าบบ
V1
                                                             บ.3ช2300
V6
  max
                   0.233900
V2
                                                   0.401000
V5
                                                                       0.372900
V7
8 rows × 61 columns
4
                                                                                                                  •
In [5]:
sonar.columns
Out[5]:
'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'V29', 'V30', 'V31',
        'V32', 'V33', 'V34', 'V35', 'V36', 'V37', 'V38', 'V39', 'V40', 'V41', 'V42', 'V43', 'V44', 'V45', 'V46', 'V47', 'V48', 'V49', 'V50', 'V51', 'V52', 'V53', 'V54', 'V55', 'V56', 'V57', 'V58', 'V59', 'V60', 'Class'],
       dtype='object')
In [6]:
sonar.skew()
Out[6]:
          2.131088
V1
          2.155644
V3
          2.652518
V4
          3.401697
V5
          2.018141
         1.653090
V58
          2.098330
V59
          1.737506
V60
          2.775754
Class
          0.135903
Length: 61, dtype: float64
In [7]:
for col in sonar.columns:
     if sonar.skew().loc[col]>0.55:
         sonar[col]=np.log1p(sonar[col])
In [8]:
sonar.skew()
Out[8]:
V1
          2.036001
V2
          1.969917
V3
          2.344713
V4
          2.818320
V5
          1.698684
            . . .
V57
          1.629182
V58
          2.058207
          1.713349
V59
V60
          2.711412
Class
         0.135903
Length: 61, dtype: float64
In [9]:
sonar.Class.unique()
Out[9]:
array([1, 0], dtype=int64)
```

```
In [10]:
sns.heatmap(sonar.isnull())
Out[10]:
<matplotlib.axes. subplots.AxesSubplot at 0x1c0511ae520>
                                                  0.100
 0
10
20
30
40
50
70
80
90
110
120
130
140
150
160
170
180
190
                                                 -0.075
                                                  0.050
                                                 - 0.025
                                                 0.000
                                                 -0.025
                                                  -0.050
                                                  -0.075
                                                  -0.100
    In [11]:
from scipy.stats import zscore
z_score=abs(zscore(sonar))
print(sonar.shape)
sonar df=sonar.loc[(z score<3).all(axis=1)]</pre>
print(sonar df.shape)
(208, 61)
(173, 61)
In [12]:
sonar df
Out[12]:
           V1
                    V2
                             V3
                                      ۷4
                                               V5
                                                        V6
                                                                  V7
                                                                           V8
                                                                                    V9
                                                                                            V10 ...
                                                                                                         V52
                                                                                                                  V53
   0 0.019803 0.036428 0.041909
                                 0.020489 \quad 0.091120 \quad 0.094037 \quad 0.143148 \quad 0.148506 \quad 0.270714
                                                                                       0.191529 ... 0.002696
                                                                                                                      0.015
   1 0.044304 0.050978 0.080935 0.066630 0.111810 0.229762 0.195238 0.298696 0.287957 0.252469 ... 0.008365 0.008861
                                                                                                                      0.004
   3 0.009950 0.016955 0.060436 0.020293 0.020293 0.036139 0.104180 0.120091 0.058080 0.119027 ... 0.012027
                                                                                                                      0.014
   4 0.073436 0.064476 0.046979 0.038644 0.057325 0.062881 0.114132 0.220500 0.304834 0.368732 ... 0.003095 0.005385
                                                                                                                      0.010
     0.028199 0.044304 0.027323
                                 0.017250 0.037681 0.094401 0.113418 0.168307 0.191033 0.265360 ...
                                                                                                    0.004490
                                                                                                             0.001399
                                                                                                                      0.003
 203 0.018527 0.034015 0.016660 0.017545 0.038547 0.151003 0.184652 0.156491 0.209288 0.237756 ... 0.011533 0.009752 0.019
 204 0.031789 0.010049 0.029365 0.054867 0.073250 0.091485 0.094401 0.096945 0.098034 0.195073 ... 0.006081 0.009257 0.013
 205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 ... 0.015873 0.002896 0.005
 206 0.029850 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177728 0.211395 ... 0.008563 0.004589
                                                                                                                      0.012
 207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 ... 0.014494 0.012818 0.004
173 rows × 61 columns
4
                                                                                                                         Þ
In [13]:
sonar df=pd.DataFrame(data=sonar_df)
sonar df
Out[13]:
```

**V/1** 

V2

V5

٧a

V10

V52

V53

1 0.044304 0.050978 0.080935 0.066360 0.111810 0.229762 0.196238 0.2298696 0.287957 0.252469 0.008365 0.008861 0.00 3 0.009950 0.016955 0.060436 0.020293 0.020293 0.036139 0.104180 0.120091 0.058000 0.119027 0.012027 0.003594 0.04 4 0.073430 0.064476 0.046979 0.038644 0.057325 0.026281 0.1141312 0.220500 0.304634 0.368732 0.030306 0.003385 0.05 5 0.02039 0.04304 0.027323 0.017250 0.037681 0.094401 0.113418 0.16807 0.19030 0.265360 0.004460 0.01139 0.00 203 0.016527 0.034015 0.016660 0.017545 0.038547 0.151003 0.184652 0.156491 0.209288 0.237756 0.011503 0.009752 0.10 204 0.031789 0.04040 0.029365 0.054867 0.073250 0.091485 0.046401 0.068645 0.080034 0.195073 0.006081 0.009257 0.10 205 0.056083 0.042772 0.017340 0.029282 0.034498 0.110736 0.116360 0.116374 0.106430 0.177728 0.211395 0.058631 0.008567 0.026686 0.028856 0.038657 0.015508 0.036691 0.047837 0.059023 0.016562 0.105808 0.106540 0.015404 0.131028 0.16912 0.18195 0.014494 0.1016410 0.0000000000000000000000000000000000		V 1	¥4 V2	¥3 V2	V-+	¥3 VE	¥0	¥ /	¥0 \/0	v 3	¥ 10	 ¥ J Z	¥ JJ	
3 0.00960 0.016955 0.060436 0.020239 0.020293 0.020293 0.036139 0.101800 0.100901 0.050800 0.119027 0.012027 0.003584 0.14 4 0.073436 0.004476 0.046979 0.033644 0.057325 0.026881 0.114132 0.220500 0.304834 0.368732 0.000305 0.050586 0.10 5 0.028199 0.044304 0.27373 0.017550 0.037861 0.034401 0.113418 0.120000 0.304834 0.368532 0.000449 0.00199 0.02038 0.016827 0.03584 0.034834 0.150739 0.010049 0.023855 0.054867 0.073260 0.074865 0.084401 0.08945 0.088034 0.150737 0.00681 0.009257 0.00280 0.05685 0.054867 0.073260 0.07486 0.110736 0.111405 0.111405 0.111405 0.111405 0.111405 0.111405 0.111405 0.111405 0.00687 0.00861 0.009257 0.00860 0.028850 0.034681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048881 0.034681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.048681 0.038681 0.038681 0.028850 0.0368681 0.038681 0.028850 0.0368681 0.038681 0.028850 0.0368681 0.028850 0.0368681 0.038681 0.028850 0.0368681 0.028850 0.0368681 0.028850 0.0368681 0.028850 0.036861 0.028850 0.036861 0.028850 0.036861 0.028850 0.036861 0.028850 0.036861 0.028850 0.036861 0.028850 0.036861 0.028850 0.036861 0.028850 0.028850 0.036861 0.028850 0.028850 0.036861 0.028850 0.02	0	0.019803	0.036428	0.041909	0.020489	0.091120	0.094037	0.143148	0.148506	0.270714	0.191529	 0.002696	0.006479	0.015
4 0.073436 0.064476 0.046979 0.038644 0.057325 0.062881 0.14132 0.220500 0.304854 0.308473	1	0.044304	0.050978	0.080935	0.066630	0.111810	0.229762	0.195238	0.298696	0.287957	0.252469	 0.008365	0.008861	0.004
5 0/22199 0.044304 0.027323 0.017250 0.037881 0.094401 0.113418 0.168307 0.191033 0.268380 0.004400 0.001399 0.002308 0.016407 0.029388 0.016407 0.036445 0.058457 0.034015 0.016660 0.017545 0.038547 0.151003 0.184652 0.156491 0.20288 0.237758 0.015133 0.009752 0.012240 0.031759 0.04680 0.029385 0.042872 0.034898 0.110730 0.118405 0.111342 0.118494 0.225461 0.015873 0.002886 0.02808 0.038687 0.033687 0.047897 0.028682 0.028688 0.035857 0.013508 0.026837 0.02174 0.033241 0.063444 0.131028 0.169152 0.211395 0.006861 0.002557 0.02868 0.038687 0.035857 0.021548 0.02174 0.033241 0.063444 0.131028 0.169152 0.211395 0.006861 0.002557 0.02868 0.038687 0.035867 0.021548 0.02174 0.033241 0.03444 0.031028 0.169152 0.211395 0.014494 0.046878 0.028181 0.046878 0.036844 0.056978 0.046878 0.036844 0.056978 0.036884 0.056978 0.036884 0.056978 0.036884 0.056978 0.036884 0.056978 0.036884 0.056978 0.036884 0.046878 0.046	3	0.009950	0.016955	0.060436	0.020293	0.020293	0.036139	0.104180	0.120091	0.058080	0.119027	 0.012027	0.003594	0.014
202 0.016827 0.034015 0.016860 0.017545 0.038647 0.151003 0.184862 0.156491 0.209288 0.237756 0.011533 0.009752 0.010204 0.023762 0.0104940 0.054872 0.034498 0.114945 0.019401 0.096945 0.096946 0.198944 0.225491 0.016873 0.002696 0.02660 0.02660 0.02660 0.02660 0.02660 0.02660 0.03660	4	0.073436	0.064476	0.046979	0.038644	0.057325	0.062881	0.114132	0.220500	0.304834	0.368732	 0.003095	0.005385	0.010
203 0.018527 0.034015 0.016660 0.017545 0.038547 0.151003 0.184652 0.156491 0.209288 0.237756 0.011533 0.009752 0.102404 0.031789 0.010049 0.029365 0.054867 0.073250 0.001445 0.008401 0.008045 0.098034 0.195073 0.006081 0.000257 0.01806 0.038680 0.038691 0.047837 0.059033 0.016562 0.128095 0.1180405 0.111302 0.1181404 0.224601 0.0118373 0.008563 0.004589 0.018070 0.025680 0.038667 0.033508 0.028837 0.021174 0.033241 0.033241 0.033444 0.131028 0.109152 0.211395 0.014404 0.122818 0.0084737 0.025680 0.038657 0.013508 0.028837 0.021174 0.033241 0.033241 0.033444 0.131028 0.109152 0.211395 0.014404 0.122818 0.0084737 0.02568 0.008663 0.036657 0.013508 0.026837 0.021174 0.033241 0.033241 0.033444 0.131028 0.109152 0.211395 0.014404 0.122818 0.008473 0.03464 0.03	5	0.028199	0.044304	0.027323	0.017250	0.037681	0.094401	0.113418	0.168307	0.191033	0.265360	 0.004490	0.001399	0.003
204 0.031789 0.010049 0.029385 0.054887 0.073250 0.091485 0.094401 0.098943 0.098034 0.195073 0.000001 0.009257 0.01 205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.119405 0.111362 0.118494 0.225461 0.015873 0.002896 0.00 206 0.029860 0.039867 0.013508 0.026337 0.021174 0.033241 0.06344 0.131028 0.169152 0.211395 0.014494 0.012818 0.00 207 0.025686 0.039657 0.013508 0.026337 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.014494 0.012818 0.00 208 0.0737 rows x 61 columns  Put												 		
205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 0.015873 0.002896 0.0266 0.028650 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177778 0.211395 0.014494 0.02683 0.04278 0.02896 0.035667 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.014494 0.012818 0.00737 rows × 61 columns  1 [14]:  1 [15]:  1 [15]:  1 [15]:  1 [15]:  1 [15]:  1 [15]:  2 [16]:  2 [17]:	203	0.018527	0.034015	0.016660	0.017545	0.038547	0.151003	0.184652	0.156491	0.209288	0.237756	 0.011533	0.009752	0.019
205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 0.015873 0.002896 0.0266 0.028650 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177778 0.211395 0.014494 0.02683 0.04278 0.02896 0.035667 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.014494 0.012818 0.00737 rows × 61 columns  1 [14]:  1 [15]:  1 [15]:  1 [15]:  1 [15]:  1 [15]:  1 [15]:  2 [16]:  2 [17]:	204	0.031789	0.010049	0.029365	0.054867	0.073250	0.091485	0.094401	0.096945	0.098034	0.195073	 0.006081	0.009257	0.013
2006 0.029850 0.034691 0.047837 0.059023 0.018592 0.18592 0.186915 0.186714 0.108430 0.177728 0.211395 0.008563 0.004589 0.012818 0.0073 rows × 61 columns														
207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.014494 0.012818 0.00737 rows × 61 columns														
73 rows × 61 columns    141														
n [14]:	201	0.023000	0.000001	0.010000	0.020037	0.021174	0.000241	0.005444	0.131020	0.103132	0.211333	 0.014454	0.012010	0.004
n [14]: ==sonar_df.iloc[:,0:-1]  n [15]:  v1 v2 v3 v4 v5 v6 v7 v8 v9 v1 0 v51 v52  0 0.019803  0.036428  0.041909  0.020489  0.091120  0.094037  0.143148  0.148506  0.270714  0.191529  0.022835  0.002696  0.00  1 0.044304  0.050978  0.080935  0.066630  0.111810  0.229762  0.195238  0.29869  0.287957  0.252469  0.012423  0.003665  0.00  3 0.009950  0.016955  0.060436  0.020293  0.020293  0.036139  0.104180  0.120991  0.058080  0.119027  0.023814  0.012027  0.00  4 0.073436  0.064476  0.046979  0.038644  0.057325  0.062881  0.114132  0.220500  0.304834  0.368732  0.015480  0.003055  0.00  5 0.028199  0.044304  0.027323  0.017250  0.037681  0.04401  0.114132  0.220500  0.304834  0.368732  0.015480  0.003065  0.00  203  0.018527  0.034015  0.016660  0.017545  0.038647  0.151003  0.184652  0.156491  0.20288  0.237756  0.02097  0.011533  0.00  204  0.031789  0.010049  0.029365  0.054867  0.073250  0.091485  0.094401  0.096945  0.098034  0.195073  0.005087  0.005081  0.00  205  0.050883  0.042772  0.017840  0.026782  0.034498  0.110736  0.118405  0.118405  0.118494  0.225461  0.015381  0.016573  0.00  206  0.029650  0.034691  0.047837  0.059023  0.016562  0.126985  0.136714  0.106430  0.177728  0.211395  0.017938  0.014949  0.01  73 rows × 60 columns  1	173 r	rows × 61	columns											
151   151	4													·······································
15   15   15   15   15   15   15   15	In [	14]:												
15   15   15   15   15   15   15   15	x=sc	nar df.	iloc[:,(	):-1]										
V1		_												
V1	T [	161.												
V1	ın [	15]:												
V1   V2   V3   V3   V4   V5   V6   V7   V8   V9   V10   W   V5   V5   V5   V5   V5   V5   V5	X													
V1   V2   V3   V3   V4   V5   V6   V7   V8   V9   V10   W   V5   V5   V5   V5   V5   V5   V5	O11+ [	151.												
0 0.019803 0.036428 0.041909 0.020489 0.091120 0.094037 0.143148 0.148506 0.270714 0.191529 0.022935 0.002696 0.00 1 0.044304 0.050978 0.080935 0.066630 0.111810 0.229762 0.195238 0.298696 0.287957 0.252469 0.012423 0.008365 0.00 3 0.009950 0.016955 0.060436 0.020293 0.020293 0.036139 0.104180 0.120091 0.058080 0.119027 0.023814 0.012027 0.00 4 0.073436 0.064476 0.046979 0.038644 0.057325 0.062881 0.114132 0.222500 0.304834 0.368732 0.015480 0.003095 0.00 5 0.028199 0.044304 0.027323 0.017250 0.037681 0.094401 0.113418 0.168307 0.191033 0.265360 0.010346 0.004490 0.00	Juc	13]:												
1 0.044304 0.050978 0.080935 0.066630 0.111810 0.229762 0.195238 0.298696 0.287957 0.252469 0.012423 0.008365 0.00 3 0.009950 0.016955 0.060436 0.020293 0.020293 0.036139 0.104180 0.120091 0.058080 0.119027 0.023814 0.012027 0.00 4 0.073436 0.064476 0.046979 0.038644 0.057325 0.062881 0.114132 0.220500 0.304834 0.368732 0.015480 0.003095 0.00 5 0.028199 0.044304 0.027323 0.017250 0.037681 0.094401 0.113418 0.168307 0.191033 0.265360 0.010346 0.004490 0.00		V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	 V51	V52	
3 0.009950 0.016955 0.060436 0.020293 0.020293 0.036139 0.104180 0.120091 0.058080 0.119027 0.023814 0.012027 0.00 4 0.073436 0.064476 0.046979 0.038644 0.057325 0.062881 0.114132 0.220500 0.304834 0.368732 0.015480 0.003095 0.00 5 0.028199 0.044304 0.027323 0.017250 0.037681 0.094401 0.113418 0.168307 0.191033 0.265360 0.010346 0.004490 0.00	0	0.019803	0.036428	0.041909	0.020489	0.091120	0.094037	0.143148	0.148506	0.270714	0.191529	 0.022935	0.002696	0.006
4 0.073436 0.064476 0.046979 0.038644 0.057325 0.062881 0.114132 0.220500 0.304834 0.368732 0.015480 0.003095 0.00 5 0.028199 0.044304 0.027323 0.017250 0.037681 0.094401 0.113418 0.168307 0.191033 0.265360 0.010346 0.004490 0.00	1	0.044304	0.050978	0.080935	0.066630	0.111810	0.229762	0.195238	0.298696	0.287957	0.252469	 0.012423	0.008365	0.008
5 0.028199 0.044304 0.027323 0.017250 0.037681 0.094401 0.113418 0.168307 0.191033 0.265360 0.010346 0.004490 0.00	3	0.009950	0.016955	0.060436	0.020293	0.020293	0.036139	0.104180	0.120091	0.058080	0.119027	 0.023814	0.012027	0.003
203 0.018527 0.034015 0.016660 0.017545 0.038547 0.151003 0.184652 0.156491 0.209288 0.237756 0.020097 0.011533 0.00 204 0.031789 0.010049 0.029365 0.054867 0.073250 0.091485 0.094401 0.096945 0.098034 0.195073 0.005087 0.006081 0.00 205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 0.015381 0.015873 0.00 206 0.029850 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177728 0.211395 0.004191 0.008563 0.00 207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 73 rows × 60 columns	4	0.073436	0.064476	0.046979	0.038644	0.057325	0.062881	0.114132	0.220500	0.304834	0.368732	 0.015480	0.003095	0.005
203 0.018527 0.034015 0.016660 0.017545 0.038547 0.151003 0.184652 0.156491 0.209288 0.237756 0.020097 0.011533 0.00 204 0.031789 0.010049 0.029365 0.054867 0.073250 0.091485 0.094401 0.096945 0.098034 0.195073 0.005087 0.006081 0.00 205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 0.015381 0.015873 0.00 206 0.029850 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177728 0.211395 0.004191 0.008563 0.00 207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 73 rows × 60 columns	5	0.028199	0.044304	0.027323	0.017250	0.037681	0.094401	0.113418	0.168307	0.191033	0.265360	 0.010346	0.004490	0.001
203 0.018527 0.034015 0.016660 0.017545 0.038547 0.151003 0.184652 0.156491 0.209288 0.237756 0.020097 0.011533 0.00 204 0.031789 0.010049 0.029365 0.054867 0.073250 0.091485 0.094401 0.096945 0.098034 0.195073 0.005087 0.006081 0.00 205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 0.015381 0.015873 0.00 206 0.029850 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177728 0.211395 0.004191 0.008563 0.00 207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 208 0.029850 0.034691 0.047837 0.056837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 200 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01	_													
204 0.031789 0.010049 0.029365 0.054867 0.073250 0.091485 0.094401 0.096945 0.098034 0.195073 0.005087 0.006081 0.00 205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 0.015381 0.015873 0.00 206 0.029850 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177728 0.211395 0.004191 0.008563 0.00 207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 208 0.029850 0.034691 0.047837 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.0017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.0017938 0.014494 0.01 209 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.0017938 0.014494 0.01	203	0.018527	0.034015	0.016660					0 156491	n 209288	0.237756	 0.020097	0.011533	0 009
205 0.050883 0.042772 0.017840 0.028782 0.034498 0.110736 0.118405 0.111362 0.118494 0.225461 0.015381 0.015873 0.00 206 0.029850 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177728 0.211395 0.004191 0.008563 0.00 207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 73 rows × 60 columns														
206 0.029850 0.034691 0.047837 0.059023 0.016562 0.126985 0.136714 0.106430 0.177728 0.211395 0.004191 0.008563 0.00 207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01 73 rows × 60 columns														
207 0.025668 0.035657 0.013508 0.026837 0.021174 0.033241 0.063444 0.131028 0.169152 0.211395 0.017938 0.014494 0.01. 73 rows × 60 columns  n [16]: shape  ut[16]:  173, 60)														
73 rows × 60 columns  n [16]:shape  ut[16]: 173, 60)  n [17]:														
n [16]: :.shape  ut[16]: 173, 60)  n [17]:	207	0.025000	0.035657	0.013506	0.020037	0.021174	0.033241	0.063444	0.131026	0.169152	0.211395	 0.017936	0.014494	0.012
n [16]:shape ut[16]: 173, 60) n [17]:	173 r	rows × 60	columns											
n [17]:	4													Þ
n [17]:	In [	16]:												
ut[16]: 173, 60) n [17]:	x sh	lane												
173, 60) n [17]:	A . OI.	арс												
n [17]:	Out[	16]:												
n [17]:	(173	60)												
	,	/												
ca=PCA(n_components=12)	In [	17]:												
	oca=	PCA(n c	omponent	cs=12)										

x=pca.fit\_transform(x)

In [18]:

In [19]:

```
x.shape
Out[19]:
(173, 12)
In [20]:
y=sonar df.iloc[:,-1]
In [21]:
Out[21]:
1
       1
       1
203
     0
204
       0
205
206
       0
207
       0
Name: Class, Length: 173, dtype: int64
In [22]:
y.shape
Out[22]:
(173,)
In [23]:
sonar.corr()
Out[23]:
                    V2
                             V3
                                      V4
                                              V5
                                                       V6
                                                                V7
                                                                        V8
                                                                                 V9
                                                                                         V10 ...
                                                                                                     V52
                                                                                                              V53
   V1 1.000000 0.735222 0.570685 0.485743 0.336469 0.244380 0.260978 0.351766 0.348769 0.312130 ... 0.353387 0.312563 0.3
   V2 0.735222 1.000000 0.773040 0.590833 0.404392 0.334846 0.278614 0.337122 0.318649 0.269347 ... 0.434345 0.348536 0.3
   V3 0.570685 0.773040 1.000000 0.765006 0.524464 0.351185 0.192678 0.246778 0.263476 0.226283 ... 0.396473 0.335055 0.3
   V4 0.485743 0.590833 0.765006 1.000000 0.708444 0.362016 0.252227 0.260257 0.252504 0.242135 ... 0.379065 0.370981 0.3
   V5 0.336469 0.404392 0.524464 0.708444 1.000000 0.605497 0.340052 0.212995 0.188492 0.194106 ... 0.267553 0.315872 0.2
  V57 0.316519 0.286026 0.388956 0.351808 0.216137 0.164213 0.183444 0.259359 0.179931 0.128099 ... 0.190659 0.306836 0.3
  V58 0.371517 0.358558 0.342265 0.357230 0.236755 0.208964 0.241244 0.283423 0.223121 0.203778 ... 0.309571 0.369978 0.4
  V59 0.358920 0.354793 0.431001 0.430703 0.288460 0.224321 0.179633 0.186581 0.082903 0.052438 ... 0.297710 0.346919 0.4
```

V60 0.348455 0.358427 0.375295 0.400941 0.247061 0.183374 0.220841 0.142316 0.082875 0.093219 ... 0.196169 0.282185 0.2

Class 0.272710 0.232735 0.194547 0.257889 0.227755 0.141348 0.120424 0.196120 0.337322 0.354554 ... 0.289856 0.141687 0.1

61 rows × 61 columns

```
In [24]:
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.22,random_state=45)
```

#### In [25]:

```
lr=LogisticRegression()
```

#### In [26]:

```
lr.fit(x_train,y_train)
lr.score(x_train,y_train)
pred=lr.predict(x_test)
print(accuracy_score(y_test,pred))
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

# 0.8205128205128205

[[17 5] [ 2 15]]

2 15]]	precision	recall	f1-score	support
0	0.89	0.77	0.83	22
1	0.75	0.88	0.81	17
accuracy			0.82	39
macro avg	0.82	0.83	0.82	39
ghted avg	0.83	0.82	0.82	39

#### In [27]:

wei

```
svc=SVC(kernel='rbf')
svc.fit(x_train,y_train)
svc.score(x_train,y_train)
predsvc=svc.predict(x_test)
print(accuracy_score(y_test,predsvc))
print(confusion_matrix(y_test,predsvc))
print(classification_report(y_test,predsvc))
```

### 0.8461538461538461

[[16 6] [ 0 17]]

[ 0 1/]]		precision	recall	f1-score	support
	0	1.00	0.73	0.84	22
	1	0.74	1.00	0.85	17
accur	racy			0.85	39
macro	avg	0.87	0.86	0.85	39
weighted	avg	0.89	0.85	0.85	39

## In [28]:

```
dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)
dtc.score(x_train,y_train)
preddtc=dtc.predict(x_test)
print(accuracy_score(y_test,preddtc))
print(confusion_matrix(y_test,preddtc))
print(classification_report(y_test,preddtc))
```

# 0.7948717948717948

[[18 4] [ 4 13]]

```
precision recall f1-score support
```

0 0.82 0.82 0.82

22

```
1 0.76 0.76
                                  0.76
                                              17
                                    0.79
                                                39
   accuracy
                0.79
                         0.79
                                    0.79
                                                39
  macro avg
                0.79
                          0.79
                                    0.79
                                               39
weighted avg
In [29]:
gnb=GaussianNB()
gnb.fit(x_train,y_train)
gnb.score(x_train,y_train)
predgnb=gnb.predict(x_test)
print(accuracy_score(y_test,predgnb))
print(confusion_matrix(y_test,predgnb))
print(classification_report(y_test,predgnb))
0.8717948717948718
[[19 3]
[ 2 15]]
            precision recall f1-score support
                                    0.88
          0
                 0.90
                       0.86
                                                22
                         0.88
                                    0.86
                                               17
                 0.83
                                    0.87
                                               39
   accuracy
                 0.87
                           0.87
                                    0.87
                                               39
  macro avg
weighted avg
                 0.87
                           0.87
                                    0.87
                                               39
In [30]:
import joblib
joblib.dump(gnb,'sonar.pkl')
Out[30]:
['sonar.pkl']
In [ ]:
```