

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Insert Data set

In [2]:

```
glass = pd.read_csv("F:\dataset\glass.csv")
glass.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 214 entries, 0 to 213
```

```
Data columns (total 10 columns):
```

```
#   Column  Non-Null Count  Dtype
---  -
0    RI      214 non-null    float64
1    Na      214 non-null    float64
2    Mg      214 non-null    float64
3    Al      214 non-null    float64
4    Si      214 non-null    float64
5    K       214 non-null    float64
6    Ca      214 non-null    float64
7    Ba      214 non-null    float64
8    Fe      214 non-null    float64
9    Type    214 non-null    int64
```

```
dtypes: float64(9), int64(1)
```

```
memory usage: 16.8 KB
```

In [3]:

```
glass.head()
```

Out[3]:

| | RI | Na | Mg | Al | Si | K | Ca | Ba | Fe | Type |
|---|---------|-------|------|------|-------|------|------|-----|-----|------|
| 0 | 1.52101 | 13.64 | 4.49 | 1.10 | 71.78 | 0.06 | 8.75 | 0.0 | 0.0 | 1 |
| 1 | 1.51761 | 13.89 | 3.60 | 1.36 | 72.73 | 0.48 | 7.83 | 0.0 | 0.0 | 1 |
| 2 | 1.51618 | 13.53 | 3.55 | 1.54 | 72.99 | 0.39 | 7.78 | 0.0 | 0.0 | 1 |
| 3 | 1.51766 | 13.21 | 3.69 | 1.29 | 72.61 | 0.57 | 8.22 | 0.0 | 0.0 | 1 |
| 4 | 1.51742 | 13.27 | 3.62 | 1.24 | 73.08 | 0.55 | 8.07 | 0.0 | 0.0 | 1 |

In [4]:

```
# To check missing values
```

In [5]:

```
glass.isnull()
```

Out[5]:

| | RI | Na | Mg | Al | Si | K | Ca | Ba | Fe | Type |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | False | False | False | False | False | False | False | False | False | False |
| 1 | False | False | False | False | False | False | False | False | False | False |
| 2 | False | False | False | False | False | False | False | False | False | False |
| 3 | False | False | False | False | False | False | False | False | False | False |
| 4 | False | False | False | False | False | False | False | False | False | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 209 | False | False | False | False | False | False | False | False | False | False |
| 210 | False | False | False | False | False | False | False | False | False | False |
| 211 | False | False | False | False | False | False | False | False | False | False |
| 212 | False | False | False | False | False | False | False | False | False | False |
| 213 | False | False | False | False | False | False | False | False | False | False |

214 rows × 10 columns

In [6]:

```
glass.corr()
```

Out[6]:

| | RI | Na | Mg | Al | Si | K | Ca | Ba |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| RI | 1.000000 | -0.191885 | -0.122274 | -0.407326 | -0.542052 | -0.289833 | 0.810403 | -0.000386 |
| Na | -0.191885 | 1.000000 | -0.273732 | 0.156794 | -0.069809 | -0.266087 | -0.275442 | 0.326603 |
| Mg | -0.122274 | -0.273732 | 1.000000 | -0.481799 | -0.165927 | 0.005396 | -0.443750 | -0.492262 |
| Al | -0.407326 | 0.156794 | -0.481799 | 1.000000 | -0.005524 | 0.325958 | -0.259592 | 0.479404 |
| Si | -0.542052 | -0.069809 | -0.165927 | -0.005524 | 1.000000 | -0.193331 | -0.208732 | -0.102151 |
| K | -0.289833 | -0.266087 | 0.005396 | 0.325958 | -0.193331 | 1.000000 | -0.317836 | -0.042618 |
| Ca | 0.810403 | -0.275442 | -0.443750 | -0.259592 | -0.208732 | -0.317836 | 1.000000 | -0.112841 |
| Ba | -0.000386 | 0.326603 | -0.492262 | 0.479404 | -0.102151 | -0.042618 | -0.112841 | 1.000000 |
| Fe | 0.143010 | -0.241346 | 0.083060 | -0.074402 | -0.094201 | -0.007719 | 0.124968 | -0.058692 |
| Type | -0.164237 | 0.502898 | -0.744993 | 0.598829 | 0.151565 | -0.010054 | 0.000952 | 0.575161 |

In [7]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(glass.drop('Type',axis=1),
                                                    glass['Type'], test_size=0.10,
                                                    random_state=5)
```

In [8]:

```
print(X_train,y_train)
```

| | RI | Na | Mg | Al | Si | K | Ca | Ba | Fe |
|-----|---------|-------|------|------|-------|------|------|------|------|
| 93 | 1.51590 | 13.24 | 3.34 | 1.47 | 73.10 | 0.39 | 8.22 | 0.00 | 0.00 |
| 119 | 1.51652 | 13.56 | 3.57 | 1.47 | 72.45 | 0.64 | 7.96 | 0.00 | 0.00 |
| 46 | 1.51869 | 13.19 | 3.37 | 1.18 | 72.72 | 0.57 | 8.83 | 0.00 | 0.16 |
| 145 | 1.51839 | 12.85 | 3.67 | 1.24 | 72.57 | 0.62 | 8.68 | 0.00 | 0.35 |
| 177 | 1.51937 | 13.79 | 2.41 | 1.19 | 72.76 | 0.00 | 9.77 | 0.00 | 0.00 |
| .. | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 144 | 1.51660 | 12.99 | 3.18 | 1.23 | 72.97 | 0.58 | 8.81 | 0.00 | 0.24 |
| 118 | 1.51673 | 13.30 | 3.64 | 1.53 | 72.53 | 0.65 | 8.03 | 0.00 | 0.29 |
| 189 | 1.52365 | 15.79 | 1.83 | 1.31 | 70.43 | 0.31 | 8.61 | 1.68 | 0.00 |
| 206 | 1.51645 | 14.94 | 0.00 | 1.87 | 73.11 | 0.00 | 8.67 | 1.38 | 0.00 |
| 99 | 1.51811 | 12.96 | 2.96 | 1.43 | 72.92 | 0.60 | 8.79 | 0.14 | 0.00 |

```
[192 rows x 9 columns] 93      2
```

```
119      2
```

```
46       1
```

```
145      2
```

```
177      6
```

```
..
```

```
144      2
```

```
118      2
```

```
189      7
```

```
206      7
```

```
99       2
```

```
Name: Type, Length: 192, dtype: int64
```

In [9]:

```
print(X_test,y_test)
```

| | RI | Na | Mg | Al | Si | K | Ca | Ba | Fe | | |
|-----|---------|-------|------|------|-------|------|------|------|------|----|---|
| 21 | 1.51966 | 14.77 | 3.75 | 0.29 | 72.02 | 0.03 | 9.00 | 0.00 | 0.00 | | |
| 54 | 1.51778 | 13.21 | 2.81 | 1.29 | 72.98 | 0.51 | 9.02 | 0.00 | 0.09 | | |
| 84 | 1.51409 | 14.25 | 3.09 | 2.08 | 72.28 | 1.10 | 7.08 | 0.00 | 0.00 | | |
| 102 | 1.51820 | 12.62 | 2.76 | 0.83 | 73.81 | 0.35 | 9.42 | 0.00 | 0.20 | | |
| 26 | 1.51793 | 13.21 | 3.48 | 1.41 | 72.64 | 0.59 | 8.43 | 0.00 | 0.00 | | |
| 202 | 1.51514 | 14.85 | 0.00 | 2.42 | 73.72 | 0.00 | 8.39 | 0.56 | 0.00 | | |
| 208 | 1.51640 | 14.37 | 0.00 | 2.74 | 72.85 | 0.00 | 9.45 | 0.54 | 0.00 | | |
| 28 | 1.51768 | 12.56 | 3.52 | 1.43 | 73.15 | 0.57 | 8.54 | 0.00 | 0.00 | | |
| 6 | 1.51743 | 13.30 | 3.60 | 1.14 | 73.09 | 0.58 | 8.17 | 0.00 | 0.00 | | |
| 161 | 1.51934 | 13.64 | 3.54 | 0.75 | 72.65 | 0.16 | 8.89 | 0.15 | 0.24 | | |
| 188 | 1.52247 | 14.86 | 2.20 | 2.06 | 70.26 | 0.76 | 9.76 | 0.00 | 0.00 | | |
| 25 | 1.51764 | 12.98 | 3.54 | 1.21 | 73.00 | 0.65 | 8.53 | 0.00 | 0.00 | | |
| 74 | 1.51596 | 13.02 | 3.56 | 1.54 | 73.11 | 0.72 | 7.90 | 0.00 | 0.00 | | |
| 142 | 1.51662 | 12.85 | 3.51 | 1.44 | 73.01 | 0.68 | 8.23 | 0.06 | 0.25 | | |
| 42 | 1.51779 | 13.21 | 3.39 | 1.33 | 72.76 | 0.59 | 8.59 | 0.00 | 0.00 | | |
| 37 | 1.51797 | 12.74 | 3.48 | 1.35 | 72.96 | 0.64 | 8.68 | 0.00 | 0.00 | | |
| 122 | 1.51687 | 13.23 | 3.54 | 1.48 | 72.84 | 0.56 | 8.10 | 0.00 | 0.00 | | |
| 198 | 1.51531 | 14.38 | 0.00 | 2.66 | 73.10 | 0.04 | 9.08 | 0.64 | 0.00 | | |
| 117 | 1.51708 | 13.72 | 3.68 | 1.81 | 72.06 | 0.64 | 7.88 | 0.00 | 0.00 | | |
| 127 | 1.52081 | 13.78 | 2.28 | 1.43 | 71.99 | 0.49 | 9.85 | 0.00 | 0.17 | | |
| 53 | 1.51837 | 13.14 | 2.84 | 1.28 | 72.85 | 0.55 | 9.07 | 0.00 | 0.00 | | |
| 101 | 1.51730 | 12.35 | 2.72 | 1.63 | 72.87 | 0.70 | 9.23 | 0.00 | 0.00 | 21 | 1 |
| 54 | 1 | | | | | | | | | | |
| 84 | 2 | | | | | | | | | | |
| 102 | 2 | | | | | | | | | | |
| 26 | 1 | | | | | | | | | | |
| 202 | 7 | | | | | | | | | | |
| 208 | 7 | | | | | | | | | | |
| 28 | 1 | | | | | | | | | | |
| 6 | 1 | | | | | | | | | | |
| 161 | 3 | | | | | | | | | | |
| 188 | 7 | | | | | | | | | | |
| 25 | 1 | | | | | | | | | | |
| 74 | 2 | | | | | | | | | | |
| 142 | 2 | | | | | | | | | | |
| 42 | 1 | | | | | | | | | | |
| 37 | 1 | | | | | | | | | | |
| 122 | 2 | | | | | | | | | | |
| 198 | 7 | | | | | | | | | | |
| 117 | 2 | | | | | | | | | | |
| 127 | 2 | | | | | | | | | | |
| 53 | 1 | | | | | | | | | | |
| 101 | 2 | | | | | | | | | | |

Name: Type, dtype: int64

In [10]:

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(random_state=40, multi_class='multinomial', solver='newton-cg')
model1=clf.fit(X_train,y_train)
```

In [11]:

```
pred1 = model1.predict(X_test)
```

In [12]:

```
print(pred1)
```

```
[1 2 2 1 2 7 7 1 1 1 2 1 2 2 2 1 2 7 2 2 2 2]
```

In [13]:

```
print(y_test, pred1)
```

```
21      1
54      1
84      2
102     2
26      1
202     7
208     7
28      1
6       1
161     3
188     7
25      1
74      2
142     2
42      1
37      1
122     2
198     7
117     2
127     2
53      1
101     2
```

```
Name: Type, dtype: int64 [1 2 2 1 2 7 7 1 1 1 2 1 2 2 2 1 2 7 2 2 2 2]
```

In [14]:

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, model1.predict(X_test)))
```

```
[[5 4 0 0]
 [1 7 0 0]
 [1 0 0 0]
 [0 1 0 3]]
```

In [15]:

```
print(classification_report(y_test,model1.predict(X_test)))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1 | 0.71 | 0.56 | 0.63 | 9 |
| 2 | 0.58 | 0.88 | 0.70 | 8 |
| 3 | 0.00 | 0.00 | 0.00 | 1 |
| 7 | 1.00 | 0.75 | 0.86 | 4 |
| accuracy | | | 0.68 | 22 |
| macro avg | 0.57 | 0.55 | 0.55 | 22 |
| weighted avg | 0.69 | 0.68 | 0.67 | 22 |

C:\Users\Pratima Dhar\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1272: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

In [16]:

```
import scipy as scp
import sklearn

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn.metrics import confusion_matrix
import statsmodels.api as sm
```

In [17]:

```
logit_model=sm.MNLogit(y_train,sm.add_constant(X_train))  
logit_model  
result=logit_model.fit(method='bfgs')  
stats1=result.summary()  
  
print(stats1)
```

```
Warning: Maximum number of iterations has been exceeded.  
Current function value: 0.795573  
Iterations: 35  
Function evaluations: 43  
Gradient evaluations: 43
```

```
C:\Users\Pratima Dhar\anaconda3\lib\site-packages\statsmodels\base\model.py:568: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals  
"Check mle_retvals", ConvergenceWarning)
```


MNLogit Regression Results

```

=====
====
Dep. Variable:                Type    No. Observations:
192
Model:                        MNLogit   Df Residuals:
142
Method:                       MLE      Df Model:
45
Date:                         Mon, 06 Jul 2020   Pseudo R-squ.:           0.
4801
Time:                         08:52:27   Log-Likelihood:          -15
2.75
converged:                    False    LL-Null:                 -29
3.80
Covariance Type:              nonrobust   LLR p-value:             4.463
e-36
=====

```

```

====
Type=2      coef      std err          z      P>|z|      [0.025      0.
975]
-----

```

```

----
const      0.0959      343.858      0.000      1.000      -673.853      67
4.045
RI          0.1577      208.896      0.001      0.999      -409.271      40
9.586
Na          1.0673        1.863        0.573      0.567        -2.584
4.718
Mg         -1.8112        2.093       -0.865      0.387        -5.914
2.291
Al          2.7635        2.229        1.240      0.215        -1.605
7.132
Si         -0.1817        1.836       -0.099      0.921        -3.780
3.417
K           2.0302        2.640        0.769      0.442        -3.143
7.204
Ca          0.0430        2.024        0.021      0.983        -3.923
4.009
Ba          1.5124        3.673        0.412      0.681        -5.687
8.712
Fe          1.3450        2.122        0.634      0.526        -2.814
5.504
-----

```

```

----
Type=3      coef      std err          z      P>|z|      [0.025      0.
975]
-----

```

```

----
const      0.0189      506.769      3.72e-05      1.000      -993.231      99
3.269
RI         -0.0024      281.533     -8.41e-06      1.000      -551.796      55
1.792
Na          1.1085        2.910        0.381      0.703        -4.596
6.813
Mg         -0.2621        3.182       -0.082      0.934        -6.499
5.975
Al          0.9439        3.473        0.272      0.786        -5.862
7.750
Si         -0.2216        2.841       -0.078      0.938        -5.789
5.346
-----

```

| | | | | | | |
|----------------|---------|----------|-----------|-------|-----------|-----|
| K 7.602 | 0.0577 | 3.849 | 0.015 | 0.988 | -7.486 | |
| Ca 5.887 | -0.0253 | 3.017 | -0.008 | 0.993 | -5.938 | |
| Ba 0.575 | -0.7066 | 5.756 | -0.123 | 0.902 | -11.989 | 1 |
| Fe 6.049 | -0.4216 | 3.301 | -0.128 | 0.898 | -6.892 | |
| ----- | | | | | | |
| ---- | | | | | | |
| Type=5 | coef | std err | z | P> z | [0.025 | 0. |
| 975] | ----- | | | | | |
| ---- | | | | | | |
| const 5.627 | 0.0483 | 1135.520 | 4.26e-05 | 1.000 | -2225.531 | 222 |
| RI 7.974 | 0.0614 | 600.987 | 0.000 | 1.000 | -1177.852 | 117 |
| Na 1.889 | -0.0642 | 6.099 | -0.011 | 0.992 | -12.018 | 1 |
| Mg 8.101 | -4.0148 | 6.182 | -0.649 | 0.516 | -16.131 | |
| Al 8.561 | 6.1606 | 6.327 | 0.974 | 0.330 | -6.239 | 1 |
| Si 2.019 | 0.0983 | 6.082 | 0.016 | 0.987 | -11.822 | 1 |
| K 6.773 | 2.8785 | 7.089 | 0.406 | 0.685 | -11.016 | 1 |
| Ca 1.236 | -0.8087 | 6.145 | -0.132 | 0.895 | -12.853 | 1 |
| Ba 3.714 | 0.2895 | 6.850 | 0.042 | 0.966 | -13.135 | 1 |
| Fe 0.915 | -0.2670 | 5.705 | -0.047 | 0.963 | -11.449 | 1 |
| ----- | | | | | | |
| ---- | | | | | | |
| Type=6 | coef | std err | z | P> z | [0.025 | 0. |
| 975] | ----- | | | | | |
| ---- | | | | | | |
| const 3.843 | -0.0893 | 1553.055 | -5.75e-05 | 1.000 | -3044.021 | 304 |
| RI 0.553 | -0.1534 | 898.336 | -0.000 | 1.000 | -1760.860 | 176 |
| Na 0.128 | 3.0810 | 8.697 | 0.354 | 0.723 | -13.966 | 2 |
| Mg 2.303 | -4.2881 | 8.465 | -0.507 | 0.612 | -20.879 | 1 |
| Al 7.726 | 0.9634 | 8.553 | 0.113 | 0.910 | -15.799 | 1 |
| Si 6.274 | -0.1560 | 8.383 | -0.019 | 0.985 | -16.586 | 1 |
| K 7.928 | -3.0299 | 10.693 | -0.283 | 0.777 | -23.988 | 1 |
| Ca 5.136 | -2.1687 | 8.829 | -0.246 | 0.806 | -19.474 | 1 |
| Ba 4.898 | -2.4770 | 8.865 | -0.279 | 0.780 | -19.852 | 1 |
| Fe 1.577 | -0.5883 | 11.309 | -0.052 | 0.959 | -22.754 | 2 |
| ----- | | | | | | |

| ----- Type=7 975] | coef | std err | z | P> z | [0.025 | 0. |
|-------------------------|---------|----------|-----------|-------|-----------|-----|
| ----- | | | | | | |
| ----- | | | | | | |
| const 3.056 | -0.0474 | 1527.122 | -3.1e-05 | 1.000 | -2993.151 | 299 |
| RI 6.667 | -0.0640 | 865.695 | -7.39e-05 | 1.000 | -1696.795 | 169 |
| Na 0.267 | 0.6996 | 9.983 | 0.070 | 0.944 | -18.868 | 2 |
| Mg 3.817 | -6.1551 | 10.190 | -0.604 | 0.546 | -26.128 | 1 |
| Al 0.539 | 0.8317 | 10.055 | 0.083 | 0.934 | -18.876 | 2 |
| Si 9.202 | 0.4596 | 9.562 | 0.048 | 0.962 | -18.282 | 1 |
| K 3.273 | 0.9101 | 11.410 | 0.080 | 0.936 | -21.453 | 2 |
| Ca 7.257 | -3.3714 | 10.525 | -0.320 | 0.749 | -24.000 | 1 |
| Ba 2.897 | 3.1390 | 10.081 | 0.311 | 0.756 | -16.619 | 2 |
| Fe 5.318 | -0.3038 | 13.072 | -0.023 | 0.981 | -25.925 | 2 |
| ===== | | | | | | |
| ===== | | | | | | |
| ◀ | | | | | | ▶ |

In [18]:

```

glass1 = pd.read_csv("F:\dataset\glass.csv")
glass1.info()
col=['Na', 'Mg', 'Al', 'Si', 'Fe', 'Ba', 'Ca', 'K']
X=glass1[col]
X

```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 214 entries, 0 to 213
```

```
Data columns (total 10 columns):
```

```

#   Column  Non-Null Count  Dtype
---  -----  -
0   RI      214 non-null    float64
1   Na      214 non-null    float64
2   Mg      214 non-null    float64
3   Al      214 non-null    float64
4   Si      214 non-null    float64
5   K       214 non-null    float64
6   Ca      214 non-null    float64
7   Ba      214 non-null    float64
8   Fe      214 non-null    float64
9   Type    214 non-null    int64

```

```
dtypes: float64(9), int64(1)
```

```
memory usage: 16.8 KB
```

Out[18]:

| | Na | Mg | Al | Si | Fe | Ba | Ca | K |
|-----|-------|------|------|-------|-----|------|------|------|
| 0 | 13.64 | 4.49 | 1.10 | 71.78 | 0.0 | 0.00 | 8.75 | 0.06 |
| 1 | 13.89 | 3.60 | 1.36 | 72.73 | 0.0 | 0.00 | 7.83 | 0.48 |
| 2 | 13.53 | 3.55 | 1.54 | 72.99 | 0.0 | 0.00 | 7.78 | 0.39 |
| 3 | 13.21 | 3.69 | 1.29 | 72.61 | 0.0 | 0.00 | 8.22 | 0.57 |
| 4 | 13.27 | 3.62 | 1.24 | 73.08 | 0.0 | 0.00 | 8.07 | 0.55 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 209 | 14.14 | 0.00 | 2.88 | 72.61 | 0.0 | 1.06 | 9.18 | 0.08 |
| 210 | 14.92 | 0.00 | 1.99 | 73.06 | 0.0 | 1.59 | 8.40 | 0.00 |
| 211 | 14.36 | 0.00 | 2.02 | 73.42 | 0.0 | 1.64 | 8.44 | 0.00 |
| 212 | 14.38 | 0.00 | 1.94 | 73.61 | 0.0 | 1.57 | 8.48 | 0.00 |
| 213 | 14.23 | 0.00 | 2.08 | 73.36 | 0.0 | 1.67 | 8.62 | 0.00 |

214 rows × 8 columns

In [19]:

```

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,glass1['Type'], test_size=0.20,
                                                    random_state=5)

```

In [20]:

```
print(X_train,y_train)
```

| | Na | Mg | Al | Si | Fe | Ba | Ca | K |
|-----|-------|------|------|-------|------|------|------|------|
| 39 | 14.21 | 3.82 | 0.47 | 71.77 | 0.00 | 0.00 | 9.57 | 0.11 |
| 56 | 12.99 | 3.47 | 1.12 | 72.98 | 0.31 | 0.00 | 8.35 | 0.62 |
| 40 | 12.79 | 3.50 | 1.12 | 73.03 | 0.00 | 0.00 | 8.77 | 0.64 |
| 51 | 13.20 | 3.33 | 1.28 | 72.36 | 0.11 | 0.00 | 9.14 | 0.60 |
| 160 | 13.33 | 3.34 | 1.54 | 72.14 | 0.00 | 0.00 | 8.99 | 0.56 |
| .. | ... | ... | ... | ... | ... | ... | ... | ... |
| 144 | 12.99 | 3.18 | 1.23 | 72.97 | 0.24 | 0.00 | 8.81 | 0.58 |
| 118 | 13.30 | 3.64 | 1.53 | 72.53 | 0.29 | 0.00 | 8.03 | 0.65 |
| 189 | 15.79 | 1.83 | 1.31 | 70.43 | 0.00 | 1.68 | 8.61 | 0.31 |
| 206 | 14.94 | 0.00 | 1.87 | 73.11 | 0.00 | 1.38 | 8.67 | 0.00 |
| 99 | 12.96 | 2.96 | 1.43 | 72.92 | 0.00 | 0.14 | 8.79 | 0.60 |

```
[171 rows x 8 columns] 39      1
```

```
56      1
```

```
40      1
```

```
51      1
```

```
160     3
```

```
..
```

```
144     2
```

```
118     2
```

```
189     7
```

```
206     7
```

```
99      2
```

```
Name: Type, Length: 171, dtype: int64
```

In [21]:

```
print(X_test,y_test)
```

| | Na | Mg | Al | Si | Fe | Ba | Ca | K | | |
|-----|-------|------|------|-------|------|------|-------|------|----|---|
| 21 | 14.77 | 3.75 | 0.29 | 72.02 | 0.00 | 0.00 | 9.00 | 0.03 | | |
| 54 | 13.21 | 2.81 | 1.29 | 72.98 | 0.09 | 0.00 | 9.02 | 0.51 | | |
| 84 | 14.25 | 3.09 | 2.08 | 72.28 | 0.00 | 0.00 | 7.08 | 1.10 | | |
| 102 | 12.62 | 2.76 | 0.83 | 73.81 | 0.20 | 0.00 | 9.42 | 0.35 | | |
| 26 | 13.21 | 3.48 | 1.41 | 72.64 | 0.00 | 0.00 | 8.43 | 0.59 | | |
| 202 | 14.85 | 0.00 | 2.42 | 73.72 | 0.00 | 0.56 | 8.39 | 0.00 | | |
| 208 | 14.37 | 0.00 | 2.74 | 72.85 | 0.00 | 0.54 | 9.45 | 0.00 | | |
| 28 | 12.56 | 3.52 | 1.43 | 73.15 | 0.00 | 0.00 | 8.54 | 0.57 | | |
| 6 | 13.30 | 3.60 | 1.14 | 73.09 | 0.00 | 0.00 | 8.17 | 0.58 | | |
| 161 | 13.64 | 3.54 | 0.75 | 72.65 | 0.24 | 0.15 | 8.89 | 0.16 | | |
| 188 | 14.86 | 2.20 | 2.06 | 70.26 | 0.00 | 0.00 | 9.76 | 0.76 | | |
| 25 | 12.98 | 3.54 | 1.21 | 73.00 | 0.00 | 0.00 | 8.53 | 0.65 | | |
| 74 | 13.02 | 3.56 | 1.54 | 73.11 | 0.00 | 0.00 | 7.90 | 0.72 | | |
| 142 | 12.85 | 3.51 | 1.44 | 73.01 | 0.25 | 0.06 | 8.23 | 0.68 | | |
| 42 | 13.21 | 3.39 | 1.33 | 72.76 | 0.00 | 0.00 | 8.59 | 0.59 | | |
| 37 | 12.74 | 3.48 | 1.35 | 72.96 | 0.00 | 0.00 | 8.68 | 0.64 | | |
| 122 | 13.23 | 3.54 | 1.48 | 72.84 | 0.00 | 0.00 | 8.10 | 0.56 | | |
| 198 | 14.38 | 0.00 | 2.66 | 73.10 | 0.00 | 0.64 | 9.08 | 0.04 | | |
| 117 | 13.72 | 3.68 | 1.81 | 72.06 | 0.00 | 0.00 | 7.88 | 0.64 | | |
| 127 | 13.78 | 2.28 | 1.43 | 71.99 | 0.17 | 0.00 | 9.85 | 0.49 | | |
| 53 | 13.14 | 2.84 | 1.28 | 72.85 | 0.00 | 0.00 | 9.07 | 0.55 | | |
| 101 | 12.35 | 2.72 | 1.63 | 72.87 | 0.00 | 0.00 | 9.23 | 0.70 | | |
| 93 | 13.24 | 3.34 | 1.47 | 73.10 | 0.00 | 0.00 | 8.22 | 0.39 | | |
| 119 | 13.56 | 3.57 | 1.47 | 72.45 | 0.00 | 0.00 | 7.96 | 0.64 | | |
| 46 | 13.19 | 3.37 | 1.18 | 72.72 | 0.16 | 0.00 | 8.83 | 0.57 | | |
| 145 | 12.85 | 3.67 | 1.24 | 72.57 | 0.35 | 0.00 | 8.68 | 0.62 | | |
| 177 | 13.79 | 2.41 | 1.19 | 72.76 | 0.00 | 0.00 | 9.77 | 0.00 | | |
| 212 | 14.38 | 0.00 | 1.94 | 73.61 | 0.00 | 1.57 | 8.48 | 0.00 | | |
| 87 | 13.40 | 3.49 | 1.52 | 72.65 | 0.10 | 0.00 | 8.08 | 0.67 | | |
| 3 | 13.21 | 3.69 | 1.29 | 72.61 | 0.00 | 0.00 | 8.22 | 0.57 | | |
| 172 | 13.00 | 0.00 | 3.02 | 70.70 | 0.00 | 0.00 | 6.93 | 6.21 | | |
| 48 | 13.21 | 3.77 | 0.79 | 71.99 | 0.00 | 0.00 | 10.02 | 0.13 | | |
| 55 | 12.45 | 2.71 | 1.29 | 73.70 | 0.24 | 0.00 | 9.06 | 0.56 | | |
| 193 | 14.75 | 0.00 | 2.00 | 73.02 | 0.08 | 1.59 | 8.53 | 0.00 | | |
| 34 | 12.69 | 3.54 | 1.34 | 72.95 | 0.00 | 0.00 | 8.75 | 0.57 | | |
| 184 | 17.38 | 0.00 | 0.34 | 75.41 | 0.00 | 0.00 | 6.65 | 0.00 | | |
| 185 | 13.69 | 3.20 | 1.81 | 72.81 | 0.00 | 1.19 | 5.43 | 1.76 | | |
| 130 | 13.75 | 1.01 | 1.36 | 72.19 | 0.00 | 0.00 | 11.14 | 0.33 | | |
| 154 | 12.86 | 3.58 | 1.31 | 72.61 | 0.00 | 0.00 | 8.79 | 0.61 | | |
| 116 | 13.24 | 3.90 | 1.41 | 72.33 | 0.10 | 0.00 | 8.31 | 0.55 | | |
| 61 | 13.81 | 3.58 | 1.32 | 71.72 | 0.00 | 0.69 | 8.67 | 0.12 | | |
| 17 | 14.36 | 3.85 | 0.89 | 71.36 | 0.00 | 0.00 | 9.15 | 0.15 | | |
| 134 | 13.33 | 3.85 | 1.25 | 72.78 | 0.00 | 0.00 | 8.12 | 0.52 | 21 | 1 |
| 54 | 1 | | | | | | | | | |
| 84 | 2 | | | | | | | | | |
| 102 | 2 | | | | | | | | | |
| 26 | 1 | | | | | | | | | |
| 202 | 7 | | | | | | | | | |
| 208 | 7 | | | | | | | | | |
| 28 | 1 | | | | | | | | | |
| 6 | 1 | | | | | | | | | |
| 161 | 3 | | | | | | | | | |
| 188 | 7 | | | | | | | | | |
| 25 | 1 | | | | | | | | | |
| 74 | 2 | | | | | | | | | |
| 142 | 2 | | | | | | | | | |
| 42 | 1 | | | | | | | | | |
| 37 | 1 | | | | | | | | | |
| 122 | 2 | | | | | | | | | |
| 198 | 7 | | | | | | | | | |

```

117    2
127    2
53     1
101    2
93     2
119    2
46     1
145    2
177    6
212    7
87     2
3      1
172    5
48     1
55     1
193    7
34     1
184    6
185    7
130    2
154    3
116    2
61     1
17     1
134    2

```

Name: Type, dtype: int64

In [22]:

```

from sklearn.linear_model import LogisticRegression
clf1 = LogisticRegression(random_state=40, multi_class='multinomial', solver='newton-cg')
model2=clf1.fit(X_train,y_train)

```

In [23]:

```
pred2 = model2.predict(X_test)
```

In [24]:

```
print(pred2)
```

```

[1 2 2 1 2 7 7 1 1 1 2 1 2 2 2 1 2 7 2 2 2 2 2 2 1 2 2 7 2 1 5 1 2 7 1 7 7
 2 2 2 2 1 1]

```

In [25]:

```

from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,model2.predict(X_test)))

```

```

[[10  6  0  0  0  0]
 [ 2 13  0  0  0  0]
 [ 1  1  0  0  0  0]
 [ 0  0  0  1  0  0]
 [ 0  1  0  0  0  1]
 [ 0  1  0  0  0  6]]

```


In [26]:

```
print(classification_report(y_test,model2.predict(X_test)))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1 | 0.77 | 0.62 | 0.69 | 16 |
| 2 | 0.59 | 0.87 | 0.70 | 15 |
| 3 | 0.00 | 0.00 | 0.00 | 2 |
| 5 | 1.00 | 1.00 | 1.00 | 1 |
| 6 | 0.00 | 0.00 | 0.00 | 2 |
| 7 | 0.86 | 0.86 | 0.86 | 7 |
| accuracy | | | 0.70 | 43 |
| macro avg | 0.54 | 0.56 | 0.54 | 43 |
| weighted avg | 0.66 | 0.70 | 0.66 | 43 |

C:\Users\Pratima Dhar\anaconda3\lib\site-packages\sklearn\metrics_classification.py:1272: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```