```
from google.colab import files
upload=files.upload()
```

```
Choose Files glass.csv
• glass.csv(text/csv) - 10053 bytes, last modified: 2/28/2023 - 100% done
Saving glass.csv to glass.csv
```

```
import pandas as pd
df=pd.read_csv("glass.csv")
df.head()
```

	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

```
df["Type"]
```

```
0      1
1      1
2      1
3      1
4      1
      ..
209    7
210    7
211    7
212    7
213    7
Name: Type, Length: 214, dtype: int64
```

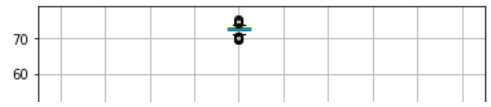
```
df.isnull().sum()
x=df.iloc[:,0:9]
y=df["Type"]
```

```
df.corr()
```

	RI	Na	Mg	Al	Si	K	Ca	
RI	1.000000	-0.191885	-0.122274	-0.407326	-0.542052	-0.289833	0.810403	-0.000
Na	-0.191885	1.000000	-0.273732	0.156794	-0.069809	-0.266087	-0.275442	0.326
Mg	-0.122274	-0.273732	1.000000	-0.481799	-0.165927	0.005396	-0.443750	-0.492
Al	-0.407326	0.156794	-0.481799	1.000000	-0.005524	0.325958	-0.259592	0.479
Si	-0.542052	-0.069809	-0.165927	-0.005524	1.000000	-0.193331	-0.208732	-0.102
K	-0.289833	-0.266087	0.005396	0.325958	-0.193331	1.000000	-0.317836	-0.042
Ca	0.810403	-0.275442	-0.443750	-0.259592	-0.208732	-0.317836	1.000000	-0.112
Ba	-0.000386	0.326603	-0.492262	0.479404	-0.102151	-0.042618	-0.112841	1.000
Fe	0.143010	-0.241346	0.083060	-0.074402	-0.094201	-0.007719	0.124968	-0.058
Type	-0.164237	0.502898	-0.744993	0.598829	0.151565	-0.010054	0.000952	0.575

```
df.boxplot()
```

```
<AxesSubplot:>
```




```
from sklearn.preprocessing import StandardScaler
SS=StandardScaler()
X_SS=SS.fit_transform(x)
df1=pd.DataFrame(X_SS)
df1
```

	0	1	2	3	4	5	6	7	8
0	0.872868	0.284953	1.254639	-0.692442	-1.127082	-0.671705	-0.145766	-0.352877	-0.586451
1	-0.249333	0.591817	0.636168	-0.170460	0.102319	-0.026213	-0.793734	-0.352877	-0.586451
2	-0.721318	0.149933	0.601422	0.190912	0.438787	-0.164533	-0.828949	-0.352877	-0.586451
3	-0.232831	-0.242853	0.698710	-0.310994	-0.052974	0.112107	-0.519052	-0.352877	-0.586451
4	-0.312045	-0.169205	0.650066	-0.411375	0.555256	0.081369	-0.624699	-0.352877	-0.586451
...	...	...	...	...	...	...	...	...	...
209	-0.704815	0.898681	-1.865511	2.881125	-0.052974	-0.640968	0.157088	1.783978	-0.586451
210	-0.500178	1.856097	-1.865511	1.094342	0.529374	-0.763919	-0.392276	2.852405	-0.586451
211	0.754046	1.168721	-1.865511	1.154570	0.995252	-0.763919	-0.364103	2.953200	-0.586451
212	-0.612399	1.193270	-1.865511	0.993960	1.241133	-0.763919	-0.335931	2.812087	-0.586451
213	-0.414363	1.009152	-1.865511	1.275028	0.917606	-0.763919	-0.237327	3.013677	-0.586451

214 rows × 9 columns

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X_SS,y, test_size=0.3)


from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=4,p=2)
knn.fit(X_train,Y_train)

y_pred_train = knn.predict(X_train)
y_pred_test = knn.predict(X_test)

from sklearn.metrics import accuracy_score
Training_accuracy = accuracy_score(Y_train,y_pred_train)
Test_accuracy = accuracy_score(Y_test,y_pred_test)

print("Training accuracy",Training_accuracy.round(4))
print("Test accuracy",Test_accuracy.round(4))
```

```
Training accuracy 0.7651
Test accuracy 0.6615
```

✓ 0s completed at 11:02 AM