In [ ]:

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

In [82]:

```
dataset = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.data")


dataset.columns=["No.","RI","Na","Mg","Al","Si","K","Ca","Ba","Fe","Type"]
```

In [83]:

```
dataset=dataset.drop("No.",axis=1)
```

In [84]:

```
dataset.head()
```

Out[84]:

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
0	1.51761	13.89	3.60	1.36	72.73	0.48	7.83	0.0	0.00	1
1	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0.0	0.00	1
2	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0.0	0.00	1
3	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0.0	0.00	1
4	1.51596	12.79	3.61	1.62	72.97	0.64	8.07	0.0	0.26	1

In [85]:

```
print(dataset.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 213 entries, 0 to 212
Data columns (total 10 columns):
RI      213 non-null float64
Na      213 non-null float64
Mg      213 non-null float64
Al      213 non-null float64
Si      213 non-null float64
K       213 non-null float64
Ca      213 non-null float64
Ba      213 non-null float64
Fe      213 non-null float64
Type    213 non-null int64
dtypes: float64(9), int64(1)
memory usage: 16.7 KB
None
```

In [86]:

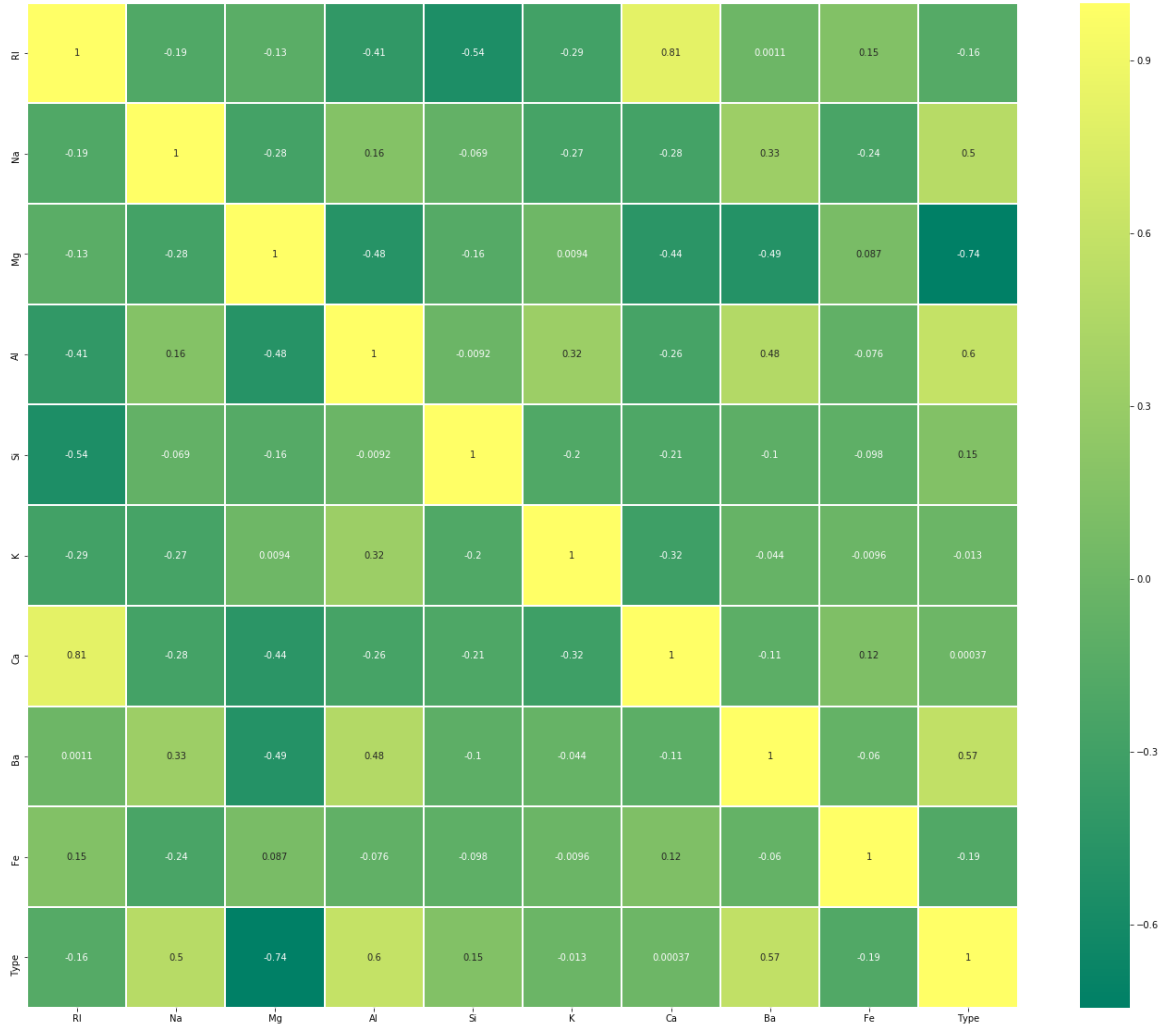
```
dataset.describe()
```

Out[86]:

	RI	Na	Mg	Al	Si	K	Ca	
count	213.000000	213.000000	213.000000	213.000000	213.000000	213.000000	213.000000	2
mean	1.518353	13.406761	2.676056	1.446526	72.655023	0.499108	8.957934	
std	0.003039	0.818371	1.440453	0.499882	0.774052	0.653035	1.426435	
min	1.511150	10.730000	0.000000	0.290000	69.810000	0.000000	5.430000	
25%	1.516520	12.900000	2.090000	1.190000	72.280000	0.130000	8.240000	
50%	1.517680	13.300000	3.480000	1.360000	72.790000	0.560000	8.600000	
75%	1.519150	13.830000	3.600000	1.630000	73.090000	0.610000	9.180000	
max	1.533930	17.380000	3.980000	3.500000	75.410000	6.210000	16.190000	

In [87]:

```
plt.figure(figsize=(24,20))
sns.heatmap(dataset.corr(),annot=True,linecolor="white",linewidths=(1,1),cmap="summer")
plt.show()
```



In [183]:

```
X = dataset.iloc[:, [1,2,3,4,7]].values
y = dataset.iloc[:, 9].values
```

In [184]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

In [185]:

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 2, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
```

Out[185]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkows
ki',
           metric_params=None, n_jobs=None, n_neighbors=2, p=2,
           weights='uniform')
```

In [186]:

```
y_pred = classifier.predict(X_test)
```

In [187]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[187]:

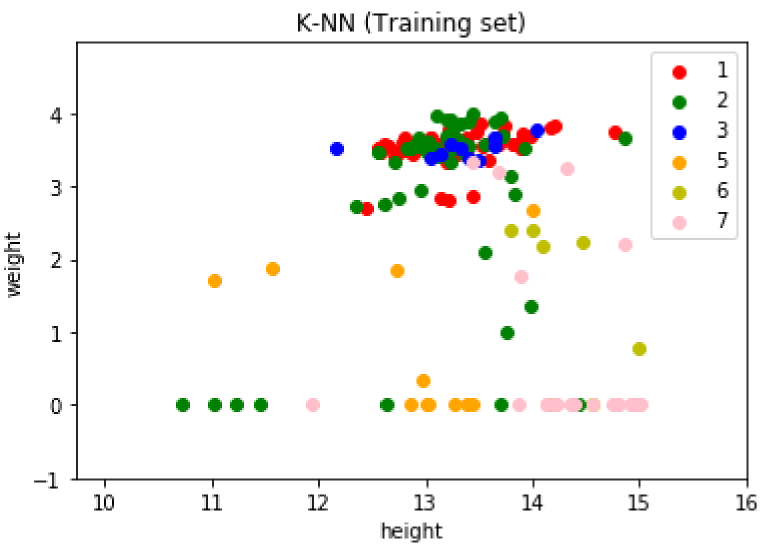
```
array([[12,  2,  0,  0,  0,  0],
       [ 9, 13,  0,  1,  0,  0],
       [ 4,  2,  0,  0,  0,  0],
       [ 0,  1,  0,  1,  0,  0],
       [ 0,  0,  0,  0,  2,  0],
       [ 0,  0,  0,  0,  0,  7]])
```

In [188]:

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0]
.max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1]
.max() + 1, step = 0.01))

plt.xlim(X1.min(), X1.max())#Setting limits turns autoscaling off for the x-axi
s.
plt.ylim(X2.min(), X2.max())
#enumerate allows us to loop over something and have an automatic counter.
#np.unique returns the unique values from the input array
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green','b','orange','y','pink'))(i),
label = j)
plt.title('K-NN (Training set)')
plt.xlabel('height')
plt.ylabel('weight')
plt.legend()
plt.show()
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which
should be avoided as value-mapping will have precedence in case its
length matches with 'x' & 'y'.  Please use a 2-D array with a single
row if you really want to specify the same RGB or RGBA value for all
points.
'c' argument looks like a single numeric RGB or RGBA sequence, which
should be avoided as value-mapping will have precedence in case its
length matches with 'x' & 'y'.  Please use a 2-D array with a single
row if you really want to specify the same RGB or RGBA value for all
points.
'c' argument looks like a single numeric RGB or RGBA sequence, which
should be avoided as value-mapping will have precedence in case its
length matches with 'x' & 'y'.  Please use a 2-D array with a single
row if you really want to specify the same RGB or RGBA value for all
points.
'c' argument looks like a single numeric RGB or RGBA sequence, which
should be avoided as value-mapping will have precedence in case its
length matches with 'x' & 'y'.  Please use a 2-D array with a single
row if you really want to specify the same RGB or RGBA value for all
points.
'c' argument looks like a single numeric RGB or RGBA sequence, which
should be avoided as value-mapping will have precedence in case its
length matches with 'x' & 'y'.  Please use a 2-D array with a single
row if you really want to specify the same RGB or RGBA value for all
points.
'c' argument looks like a single numeric RGB or RGBA sequence, which
should be avoided as value-mapping will have precedence in case its
length matches with 'x' & 'y'.  Please use a 2-D array with a single
row if you really want to specify the same RGB or RGBA value for all
points.
```



In [189]:

```
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
scores=[]
for k in range(1,32):
    knn=KNeighborsClassifier(k)
    score_val=cross_val_score(knn,X_train,y_train,scoring='accuracy',cv=10)
    score_mean=score_val.mean()
    scores.append(score_mean)

# get index of maxium score along axis, default axis=0 for 1 dimensional array
best_k=np.argmax(scores)+1
print(best_k)
# generate KNN model
knn=KNeighborsClassifier(best_k)
# fit with train data set
knn.fit(X_train,y_train)
# get Modes presicion rate using test set
print("prediction precision rate:",knn.score(X_test,y_test))
```

```
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
```

```
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
```

```
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
```

```
2
prediction precision rate: 0.6481481481481481
```

```
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
/Users/nandanadileep/anaconda3/lib/python3.7/site-packages/sklearn/m
odel_selection/_split.py:652: Warning: The least populated class in
y has only 7 members, which is too few. The minimum number of member
s in any class cannot be less than n_splits=10.
  % (min_groups, self.n_splits)), Warning)
```

In [190]:

```
#to increase the accuracy rate, we over sample the skewed data


df3=dataset[dataset['Type']==3]
df3=pd.concat([df3]*4)
df5=dataset[dataset['Type']==5]
df5=pd.concat([df5]*5)
df6=dataset[dataset['Type']==6]
df6=pd.concat([df6]*7)
df7=dataset[dataset['Type']==7]
df7=pd.concat([df7]*2)
df1=dataset[dataset['Type']==1]
df2=dataset[dataset['Type']==2]
df_balanced=pd.concat([df1,df2,df3,df5,df6,df7])
df_balanced.shape

dataset.head()
type=df_balanced['Type'].groupby(df_balanced['Type']).count()
type
df6
```

Out[190]:

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
175	1.51905	14.00	2.39	1.56	72.37	0.0	9.57	0.0	0.0	6
176	1.51937	13.79	2.41	1.19	72.76	0.0	9.77	0.0	0.0	6
177	1.51829	14.46	2.24	1.62	72.38	0.0	9.26	0.0	0.0	6
178	1.51852	14.09	2.19	1.66	72.67	0.0	9.32	0.0	0.0	6
179	1.51299	14.40	1.74	1.54	74.55	0.0	7.59	0.0	0.0	6
180	1.51888	14.99	0.78	1.74	72.50	0.0	9.95	0.0	0.0	6
181	1.51916	14.15	0.00	2.09	72.74	0.0	10.88	0.0	0.0	6
182	1.51969	14.56	0.00	0.56	73.48	0.0	11.22	0.0	0.0	6
183	1.51115	17.38	0.00	0.34	75.41	0.0	6.65	0.0	0.0	6
175	1.51905	14.00	2.39	1.56	72.37	0.0	9.57	0.0	0.0	6
176	1.51937	13.79	2.41	1.19	72.76	0.0	9.77	0.0	0.0	6
177	1.51829	14.46	2.24	1.62	72.38	0.0	9.26	0.0	0.0	6
178	1.51852	14.09	2.19	1.66	72.67	0.0	9.32	0.0	0.0	6
179	1.51299	14.40	1.74	1.54	74.55	0.0	7.59	0.0	0.0	6
180	1.51888	14.99	0.78	1.74	72.50	0.0	9.95	0.0	0.0	6
181	1.51916	14.15	0.00	2.09	72.74	0.0	10.88	0.0	0.0	6
182	1.51969	14.56	0.00	0.56	73.48	0.0	11.22	0.0	0.0	6
183	1.51115	17.38	0.00	0.34	75.41	0.0	6.65	0.0	0.0	6
175	1.51905	14.00	2.39	1.56	72.37	0.0	9.57	0.0	0.0	6
176	1.51937	13.79	2.41	1.19	72.76	0.0	9.77	0.0	0.0	6
177	1.51829	14.46	2.24	1.62	72.38	0.0	9.26	0.0	0.0	6
178	1.51852	14.09	2.19	1.66	72.67	0.0	9.32	0.0	0.0	6
179	1.51299	14.40	1.74	1.54	74.55	0.0	7.59	0.0	0.0	6
180	1.51888	14.99	0.78	1.74	72.50	0.0	9.95	0.0	0.0	6
181	1.51916	14.15	0.00	2.09	72.74	0.0	10.88	0.0	0.0	6
182	1.51969	14.56	0.00	0.56	73.48	0.0	11.22	0.0	0.0	6
183	1.51115	17.38	0.00	0.34	75.41	0.0	6.65	0.0	0.0	6
175	1.51905	14.00	2.39	1.56	72.37	0.0	9.57	0.0	0.0	6
176	1.51937	13.79	2.41	1.19	72.76	0.0	9.77	0.0	0.0	6
177	1.51829	14.46	2.24	1.62	72.38	0.0	9.26	0.0	0.0	6
...	...	...	...	...	...	...	...	...	...	...
181	1.51916	14.15	0.00	2.09	72.74	0.0	10.88	0.0	0.0	6
182	1.51969	14.56	0.00	0.56	73.48	0.0	11.22	0.0	0.0	6
183	1.51115	17.38	0.00	0.34	75.41	0.0	6.65	0.0	0.0	6
175	1.51905	14.00	2.39	1.56	72.37	0.0	9.57	0.0	0.0	6
176	1.51937	13.79	2.41	1.19	72.76	0.0	9.77	0.0	0.0	6

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
177	1.51829	14.46	2.24	1.62	72.38	0.0	9.26	0.0	0.0	6
178	1.51852	14.09	2.19	1.66	72.67	0.0	9.32	0.0	0.0	6
179	1.51299	14.40	1.74	1.54	74.55	0.0	7.59	0.0	0.0	6
180	1.51888	14.99	0.78	1.74	72.50	0.0	9.95	0.0	0.0	6
181	1.51916	14.15	0.00	2.09	72.74	0.0	10.88	0.0	0.0	6
182	1.51969	14.56	0.00	0.56	73.48	0.0	11.22	0.0	0.0	6
183	1.51115	17.38	0.00	0.34	75.41	0.0	6.65	0.0	0.0	6
175	1.51905	14.00	2.39	1.56	72.37	0.0	9.57	0.0	0.0	6
176	1.51937	13.79	2.41	1.19	72.76	0.0	9.77	0.0	0.0	6
177	1.51829	14.46	2.24	1.62	72.38	0.0	9.26	0.0	0.0	6
178	1.51852	14.09	2.19	1.66	72.67	0.0	9.32	0.0	0.0	6
179	1.51299	14.40	1.74	1.54	74.55	0.0	7.59	0.0	0.0	6
180	1.51888	14.99	0.78	1.74	72.50	0.0	9.95	0.0	0.0	6
181	1.51916	14.15	0.00	2.09	72.74	0.0	10.88	0.0	0.0	6
182	1.51969	14.56	0.00	0.56	73.48	0.0	11.22	0.0	0.0	6
183	1.51115	17.38	0.00	0.34	75.41	0.0	6.65	0.0	0.0	6
175	1.51905	14.00	2.39	1.56	72.37	0.0	9.57	0.0	0.0	6
176	1.51937	13.79	2.41	1.19	72.76	0.0	9.77	0.0	0.0	6
177	1.51829	14.46	2.24	1.62	72.38	0.0	9.26	0.0	0.0	6
178	1.51852	14.09	2.19	1.66	72.67	0.0	9.32	0.0	0.0	6
179	1.51299	14.40	1.74	1.54	74.55	0.0	7.59	0.0	0.0	6
180	1.51888	14.99	0.78	1.74	72.50	0.0	9.95	0.0	0.0	6
181	1.51916	14.15	0.00	2.09	72.74	0.0	10.88	0.0	0.0	6
182	1.51969	14.56	0.00	0.56	73.48	0.0	11.22	0.0	0.0	6
183	1.51115	17.38	0.00	0.34	75.41	0.0	6.65	0.0	0.0	6

63 rows × 10 columns

In [191]:

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier

# df.columns is column labels property
features=df_balanced.columns[:-1].tolist()
X=df_balanced[features].values
y=df_balanced['Type']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=1)
score=[]
for i in range(32):
    knn=KNeighborsClassifier(k)
    score_val=cross_val_score(knn,X_train,y_train,scoring='accuracy',cv=10)
    score_mean=score_val.mean()
    scores.append(score_mean)
best_K=np.argmax(scores)+1
print('best K is:',best_K)
knn=KNeighborsClassifier(best_K)
knn.fit(X_train,y_train)
print("prediction precision rate:",knn.score(X_test,y_test))
```

```
best K is: 2
prediction precision rate: 0.8375
```

In [ ]:

In [ ]: