In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
```

In [3]:

```
glass=pd.read_csv('glass.csv')
```

In [4]:

```
glass
```

Out[4]:

	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	0.00	0.0	1
1	1.51761	13.89	3.60	1.36	72.73	0.48	7.83	0.00	0.0	1
2	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0.00	0.0	1
3	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0.00	0.0	1
4	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0.00	0.0	1
...	...	...	...	...	...	...	...	...	...	...
209	1.51623	14.14	0.00	2.88	72.61	0.08	9.18	1.06	0.0	7
210	1.51685	14.92	0.00	1.99	73.06	0.00	8.40	1.59	0.0	7
211	1.52065	14.36	0.00	2.02	73.42	0.00	8.44	1.64	0.0	7
212	1.51651	14.38	0.00	1.94	73.61	0.00	8.48	1.57	0.0	7
213	1.51711	14.23	0.00	2.08	73.36	0.00	8.62	1.67	0.0	7

214 rows × 10 columns

In [5]:

```
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 [6]:

```
glass.duplicated()
```

Out[6]:

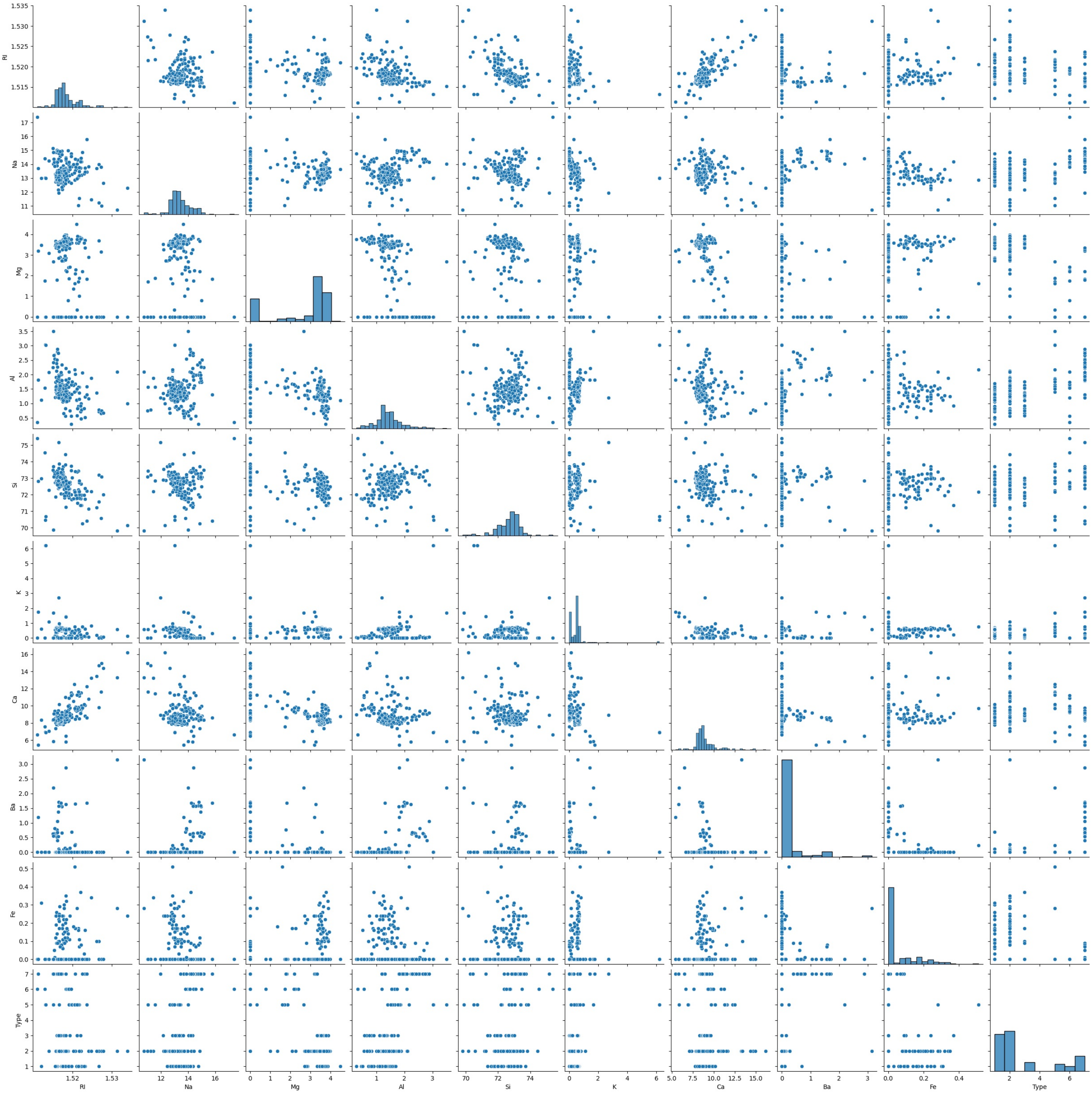
```
0      False
1      False
2      False
3      False
4      False
       ...  
209    False
210    False
211    False
212    False
213    False
Length: 214, dtype: bool
```

In [7]:

```
sns.pairplot(glass)
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x18eea4f15b0>
```



In [8]:

```
sns.heatmap(glass.isnull(),cmap='bone')
```

Out[8]:

```
<AxesSubplot:>
```

In [9]:

```
array = glass.values
X = array[:, 0:9]
Y = array[:, 9]
```

In [10]:

```
num_folds = 70
kfold = KFold(n_splits=70)
```

In [15]:

```
model = KNeighborsClassifier(n_neighbors=50)
results = cross_val_score(model, X, Y, cv=kfold)
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

In [16]:

```
print(results.mean())
```

```
0.5678571428571428
```

In [17]:

```
from sklearn.model_selection import GridSearchCV
import pandas as pd
import numpy
```

In [18]:

```
n_neighbors = numpy.array(range(1,40))
param_grid = dict(n_neighbors=n_neighbors)
```

In [19]:

```
model = KNeighborsClassifier()
grid = GridSearchCV(estimator=model, param_grid=param_grid)
grid.fit(X, Y)
```

Out[19]:

```
GridSearchCV(estimator=KNeighborsClassifier(),
             param_grid={'n_neighbors': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39])})
```

In [20]:

```
print(grid.best_score_)
print(grid.best_params_)
```

```
0.640531561461794
{'n_neighbors': 1}
```

In [21]:

```
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
```

In [22]:

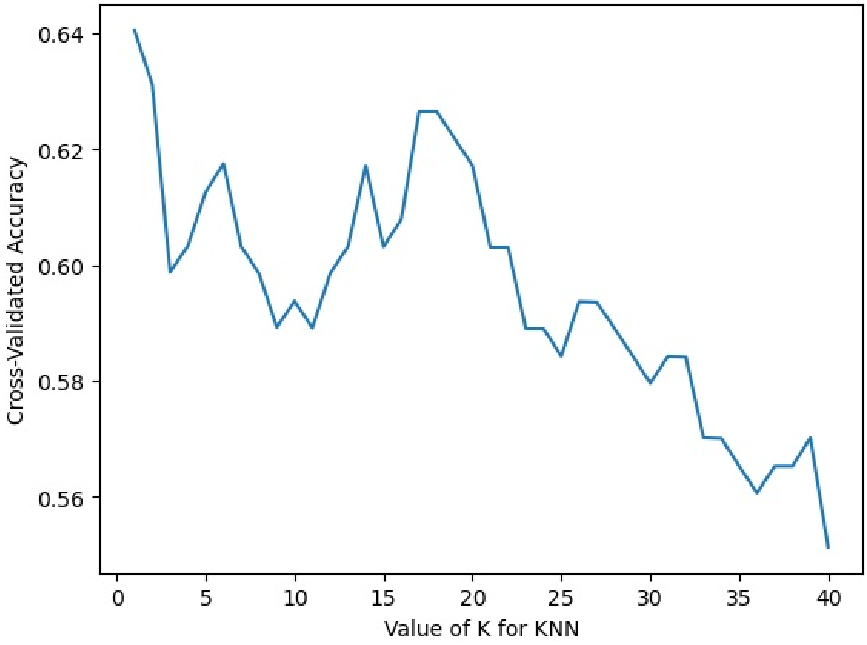
```
k_range = range(1, 41)
k_scores = []
```

In [23]:

```
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X, Y, cv=5)
    k_scores.append(scores.mean())
```

In [24]:

```
plt.plot(k_range, k_scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Cross-Validated Accuracy')
plt.show()
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js