Assignment 3 ME 793

Roll No.: 19D100011

In [25]:
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore", category=UserWarning)
warnings.filterwarnings("ignore", category=FutureWarning)
```

In [26]:
```
# Loading the data
data = pd.read_csv("material_data.csv")
data.head()
```

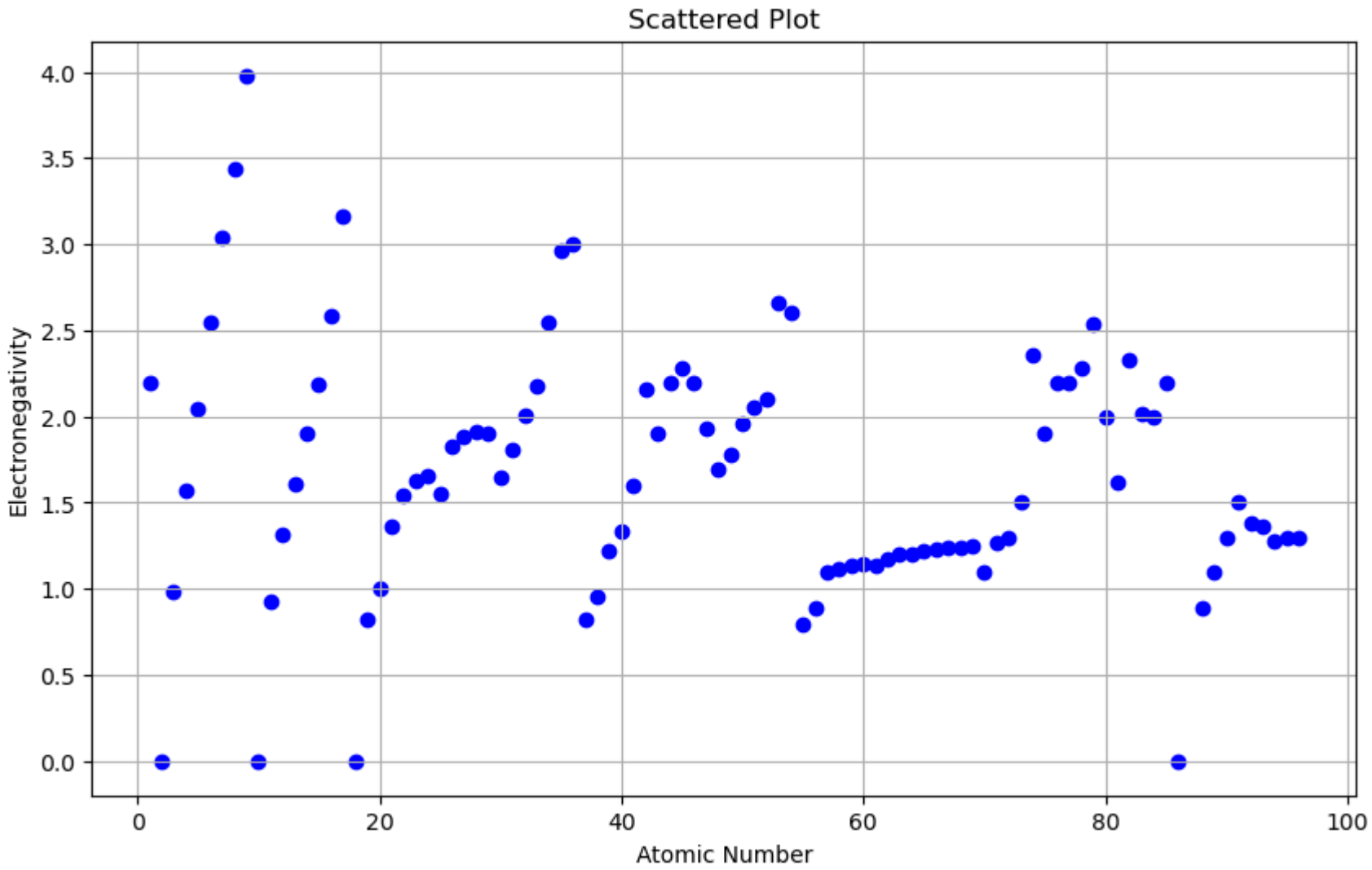
Out[26]:

	Elements	Atomic_Number	Electronegativity	Atomic_Radius	Thermal_Conductivity	Density	Crystal_System
0	H	1	2.20	0.25	0.1805	0.09	HEX
1	He	2	0.00	1.20	0.1513	0.18	HCP
2	Li	3	0.98	1.45	85.0000	530.00	BCC
3	Be	4	1.57	1.05	190.0000	1850.00	HCP
4	B	5	2.04	0.85	27.0000	2340.00	RHO

Plotting electronegativity vs. atomic number.

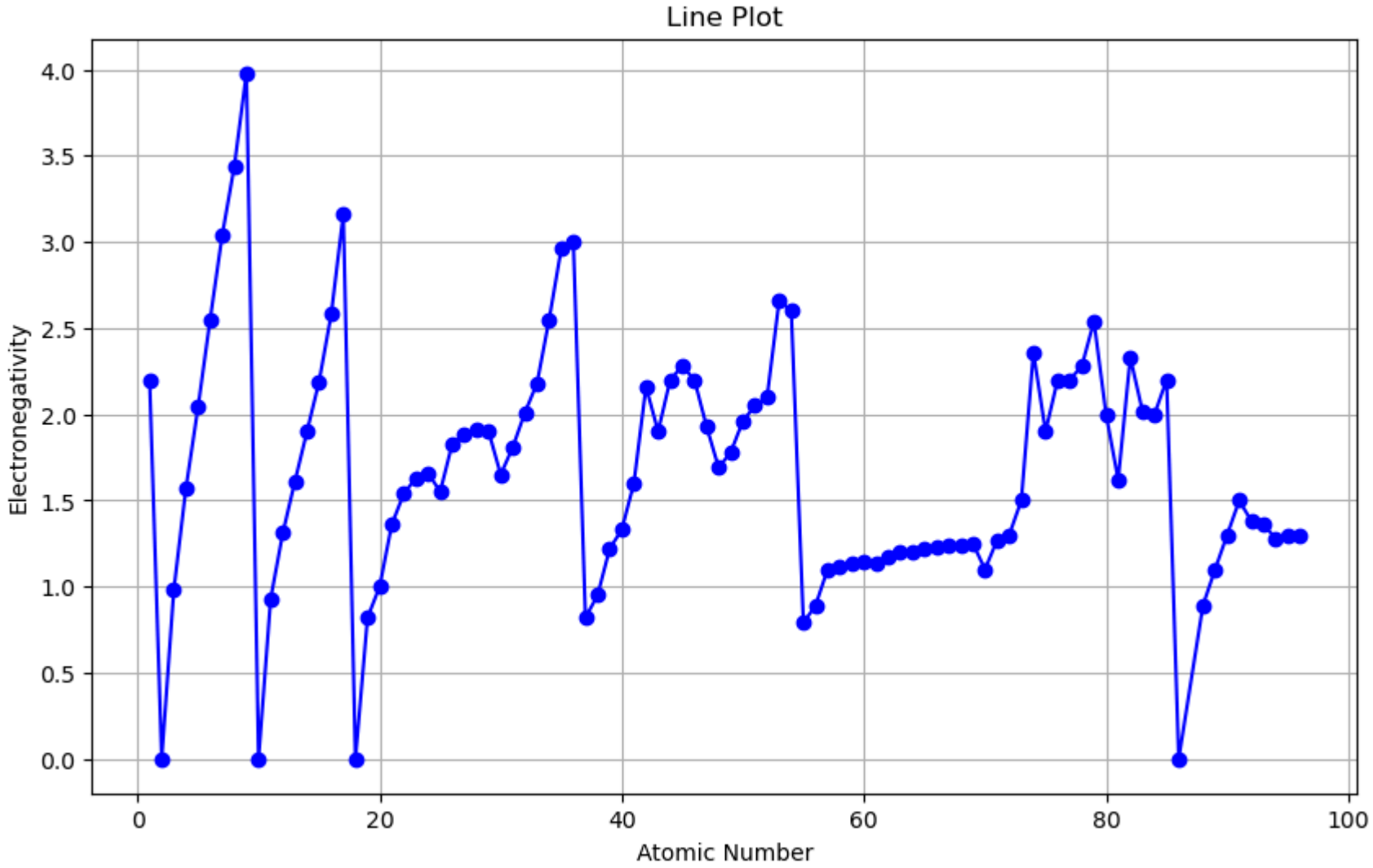
In [27]:
```
plt.figure(figsize=(10, 6))
plt.scatter(data['Atomic_Number'], data['Electronegativity'], color='blue')
plt.title('Scattered Plot')
plt.xlabel('Atomic Number')
```

```
plt.ylabel('Electronegativity')
plt.grid(True)
plt.show()
```



In [28]:
```
plt.figure(figsize=(10, 6))
plt.plot(data['Atomic_Number'], data['Electronegativity'], marker='o', color='blue')
plt.title('Line Plot')
plt.xlabel('Atomic Number')
plt.ylabel('Electronegativity')
```

```
plt.grid(True)
plt.show()
```

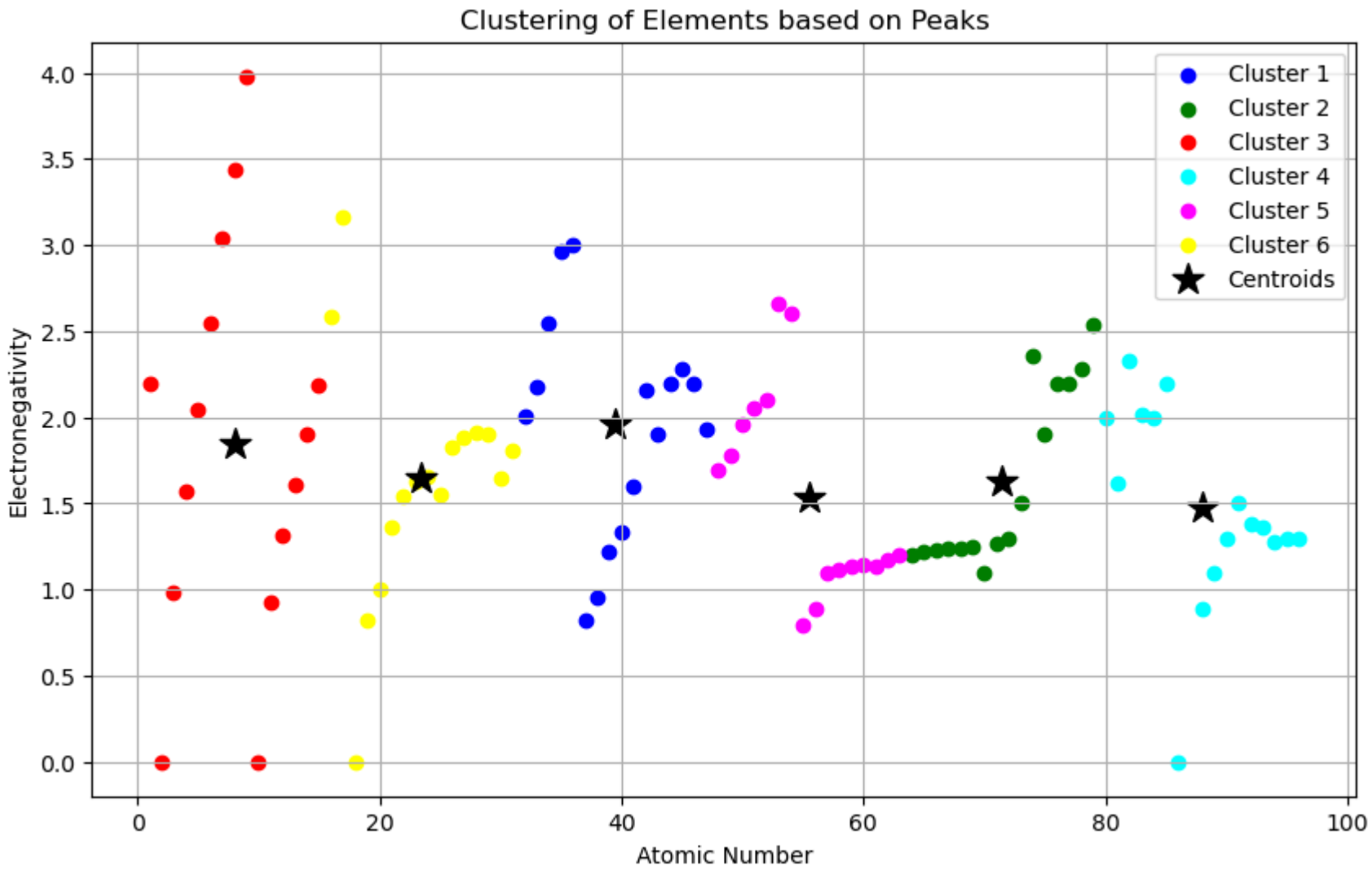


Number of Peaks: 6

Based on Electronegativity vs. Atomic Number graph there are 6 peaks

Performing K Means clustering

In [29]:
```
# Select relevant features for clustering
features = data[['Electronegativity', 'Atomic_Number']]
k = 6

# Color map
colors = ['blue', 'green', 'red', 'cyan', 'magenta', 'yellow', 'purple']

# Perform KMeans clustering
kmeans = KMeans(n_clusters=k, random_state=42)
data['Cluster'] = kmeans.fit_predict(features) + 1

# Plot clusters
plt.figure(figsize=(10, 6))

# Plot each cluster
for cluster_num, color in zip(range(1, k+1), colors):
    cluster_data = data[data['Cluster'] == cluster_num]
    plt.scatter(cluster_data['Atomic_Number'], cluster_data['Electronegativity'], color=color, label=f'Cluster {cluster_num}')

# Plot centroids
centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 1], centroids[:, 0], marker='*', s=200, c='black', label='Centroids')

# Labels and legend
plt.xlabel('Atomic Number')
plt.ylabel('Electronegativity')
plt.title('Clustering of Elements based on Peaks')
plt.legend()
plt.grid(True)
plt.show()
```



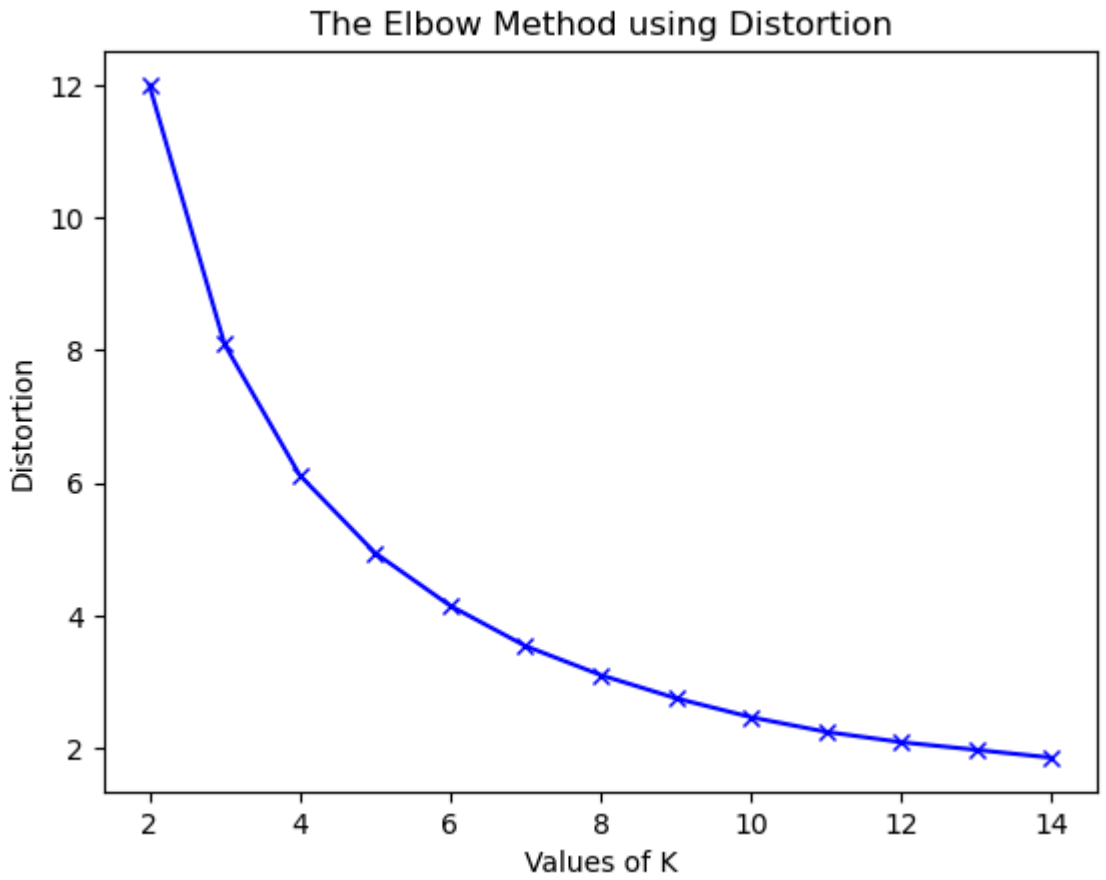
In [30]:
```
data.to_csv('data.csv', index=False)
```

Elbow Method

Finding best k using the Elbow method using Distortion

In [31]:
```
distortions = []
K = range(2, 15)
for k in K:
    k_means = KMeans(n_clusters=k, random_state=42)
    k_means.fit(features)
    distortions.append(sum(np.min(cdist(features, k_means.cluster_centers_,'euclidean'), axis=1)) / features.shape[0])
```

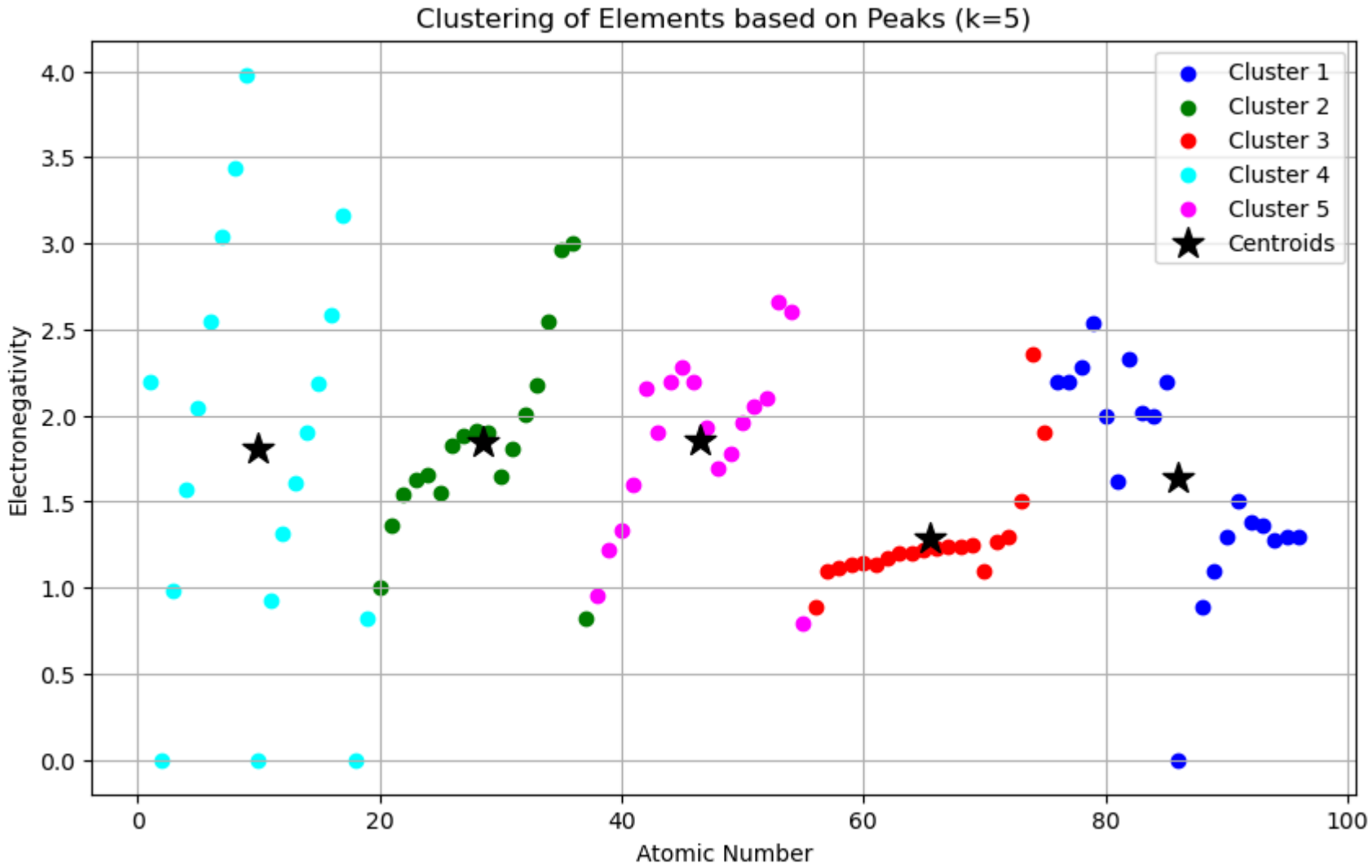
In [32]:
```
plt.plot(K, distortions, 'bx-')
plt.xlabel('Values of K')
plt.ylabel('Distortion')
plt.title('The Elbow Method using Distortion')
plt.show()
```



This plot doesn't have pointy elbow but we can see k=5 and k=6 are relatively appropriate values

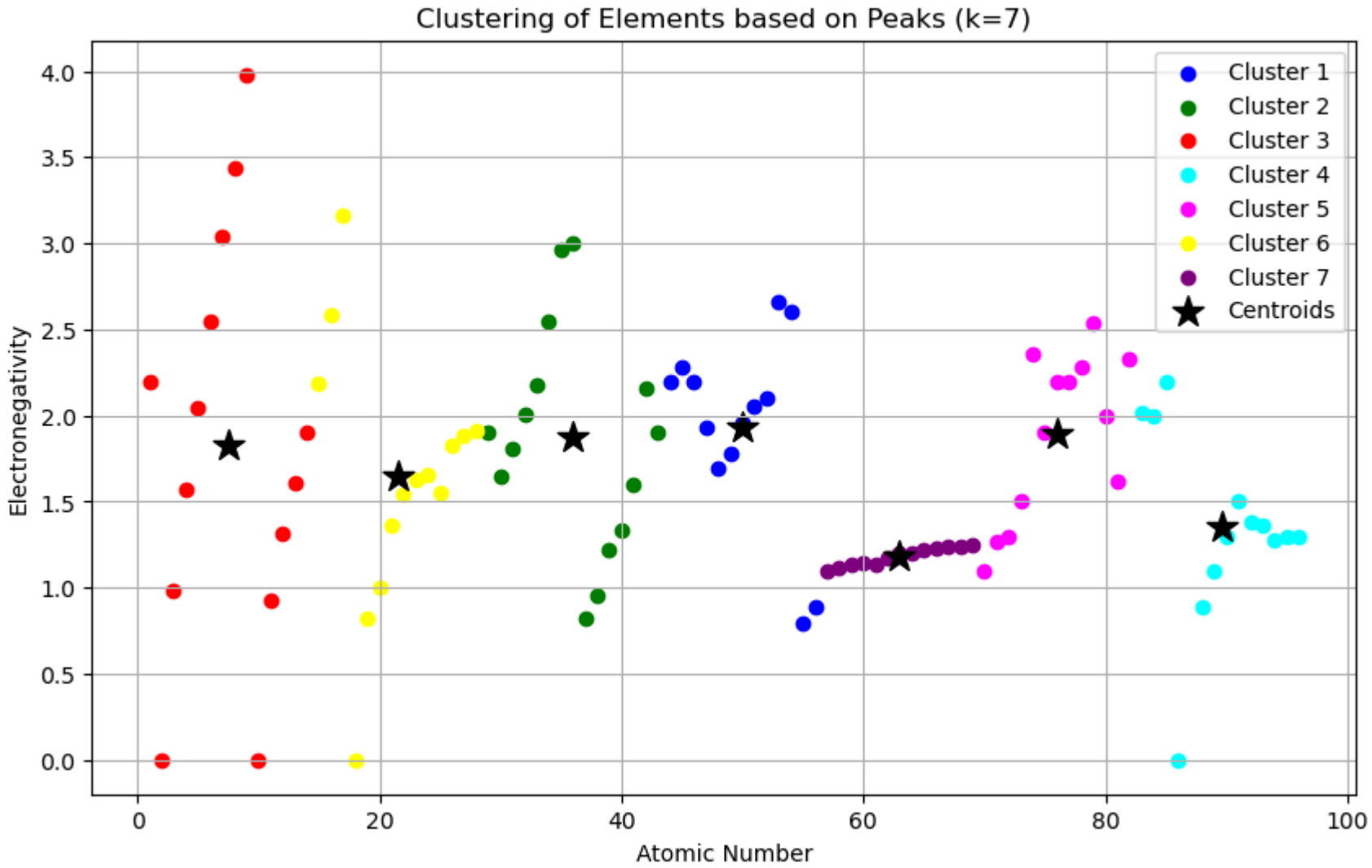
Let's plot for k=5 and k=7

In [33]:
```
# Perform KMeans clustering
k = 5
kmeans = KMeans(n_clusters=k, random_state=42)
data['Cluster'] = kmeans.fit_predict(features) + 1

# Plot clusters
plt.figure(figsize=(10, 6))

# Plot each cluster
for cluster_num, color in zip(range(1, k+1), colors):
    cluster_data = data[data['Cluster'] == cluster_num]
    plt.scatter(cluster_data['Atomic_Number'], cluster_data['Electronegativity'], color=color, label=f'Cluster {cluster_num}')

# Plot centroids
centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 1], centroids[:, 0], marker='*', s=200, c='black', label='Centroids')

# Labels and legend
plt.xlabel('Atomic Number')
plt.ylabel('Electronegativity')
plt.title('Clustering of Elements based on Peaks (k=5)')
plt.legend()
plt.grid(True)
plt.show()
```



In [34]:
```
# Perform KMeans clustering
k = 7
kmeans = KMeans(n_clusters=k, random_state=42)
data['Cluster'] = kmeans.fit_predict(features) + 1

# Plot clusters
plt.figure(figsize=(10, 6))

# Plot each cluster
```

```
for cluster_num, color in zip(range(1, k+1), colors):
    cluster_data = data[data['Cluster'] == cluster_num]
    plt.scatter(cluster_data['Atomic_Number'], cluster_data['Electronegativity'], color=color, label=f'Cluster {cluster_num}')

# Plot centroids
centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 1], centroids[:, 0], marker='*', s=200, c='black', label='Centroids')

# Labels and Legend
plt.xlabel('Atomic Number')
plt.ylabel('Electronegativity')
plt.title('Clustering of Elements based on Peaks (k=7)')
plt.legend()
plt.grid(True)
plt.show()
```


Clustering of Elements based on Peaks (k=7)
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Centroids
Electronegativity
Atomic Number
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
20
40
60
80
100

In [ ]:

Elements	Atomic_Number	Electronegativity	Atomic_Radius	Thermal_Conductivity	Density	Cluster
H	1	2.2	0.25	0.1805	0.09	3
He	2	0	1.2	0.1513	0.18	3
Li	3	0.98	1.45	85	530	3
Be	4	1.57	1.05	190	1850	3
B	5	2.04	0.85	27	2340	3
C	6	2.55	0.7	140	2260	3
N	7	3.04	0.65	0.02583	1.25	3
O	8	3.44	0.6	0.02658	1.43	3
F	9	3.98	0.5	0.0277	1.7	3
Ne	10	0	1.6	0.0491	0.9	3
Na	11	0.93	1.8	140	970	3
Mg	12	1.31	1.5	160	1740	3
Al	13	1.61	1.25	235	2700	3
Si	14	1.9	1.1	150	2330	3
P	15	2.19	1	0.236	1820	3
S	16	2.58	1	0.205	2070	6
Cl	17	3.16	1	0.0089	3.21	6
Ar	18	0	0.71	0.01772	1.78	6
K	19	0.82	2.2	100	860	6
Ca	20	1	1.8	200	1550	6
Sc	21	1.36	1.6	16	2990	6
Ti	22	1.54	1.4	22	4540	6
V	23	1.63	1.35	31	6110	6
Cr	24	1.66	1.4	94	7190	6
Mn	25	1.55	1.4	7.8	7430	6
Fe	26	1.83	1.4	80	7870	6
Co	27	1.88	1.35	100	8900	6
Ni	28	1.91	1.35	91	8900	6
Cu	29	1.9	1.35	400	8960	6
Zn	30	1.65	1.35	120	7130	6
Ga	31	1.81	1.3	29	5910	6
Ge	32	2.01	1.25	60	5320	1
As	33	2.18	1.15	50	5720	1
Se	34	2.55	1.15	0.52	4790	1
Br	35	2.96	1.15	0.12	3120	1
Kr	36	3	0.88	0.00943	3.75	1
Rb	37	0.82	2.35	58	1630	1
Sr	38	0.95	2	35	2540	1
Y	39	1.22	1.8	17	4470	1
Zr	40	1.33	1.55	23	6510	1
Nb	41	1.6	1.45	54	8570	1
Mo	42	2.16	1.45	139	10220	1
Tc	43	1.9	1.35	51	11500	1

Ru	44	2.2	1.3	120	12370	1
Rh	45	2.28	1.35	150	12410	1
Pd	46	2.2	1.4	72	12020	1
Ag	47	1.93	1.6	430	10500	1
Cd	48	1.69	1.55	97	8650	5
In	49	1.78	1.55	82	7310	5
Sn	50	1.96	1.45	67	7310	5
Sb	51	2.05	1.45	24	6680	5
Te	52	2.1	1.4	3	6240	5
I	53	2.66	1.4	0.449	4930	5
Xe	54	2.6	1.08	0.00565	5.9	5
Cs	55	0.79	2.6	36	1870	5
Ba	56	0.89	2.15	18	3590	5
La	57	1.1	1.95	13	6150	5
Ce	58	1.12	1.85	11	6770	5
Pr	59	1.13	1.85	13	6770	5
Nd	60	1.14	1.85	17	7010	5
Pm	61	1.13	1.85	15	7300	5
Sm	62	1.17	1.85	13	7520	5
Eu	63	1.2	1.85	14	5240	5
Gd	64	1.2	1.8	11	7900	2
Tb	65	1.22	1.75	11	8230	2
Dy	66	1.23	1.75	11	8550	2
Ho	67	1.24	1.75	16	8800	2
Er	68	1.24	1.75	15	9070	2
Tm	69	1.25	1.75	17	9320	2
Yb	70	1.1	1.75	39	6900	2
Lu	71	1.27	1.75	16	9840	2
Hf	72	1.3	1.55	23	13310	2
Ta	73	1.5	1.45	57	16650	2
W	74	2.36	1.35	170	19350	2
Re	75	1.9	1.35	48	21040	2
Os	76	2.2	1.3	88	22600	2
Ir	77	2.2	1.35	150	22400	2
Pt	78	2.28	1.35	72	21450	2
Au	79	2.54	1.35	320	19320	2
Hg	80	2	1.5	8.3	13550	4
Tl	81	1.62	1.9	46	11850	4
Pb	82	2.33	1.8	35	11350	4
Bi	83	2.02	1.6	8	9750	4
Po	84	2	1.9	20	9300	4
At	85	2.2	1.27	2	6350	4
Rn	86	0	1.2	0.00361	9.73	4
Ra	88	0.89	2.15	19	5500	4

Ac	89	1.1	1.95	12	10070	4
Th	90	1.3	1.8	54	11720	4
Pa	91	1.5	1.8	47	15400	4
U	92	1.38	1.75	27	18950	4
Np	93	1.36	1.75	6	20200	4
Pu	94	1.28	1.75	6	19840	4
Am	95	1.3	1.75	10	13670	4
Cm	96	1.3	1.76	8.8	13500	4

Crystal_System
HEX
HCP
BCC
HCP
RHO
HEX
HEX
SC
SC
FCC
BCC
HCP
FCC
DC
ORTH
ORTH
ORTH
FCC
BCC
FCC
HCP
HCP
BCC
BCC
BCC
BCC
HCP
FCC
FCC
HCP
ORTH
DC
RHO
HEX
ORTH
FCC
BCC
FCC
HCP
HCP
BCC
BCC
HCP

HCP
FCC
FCC
FCC
HCP
TETR
TETR
RHO
HEX
ORTH
FCC
BCC
BCC
HCP
HCP
HCP
HCP
HCP
HCP
RHO
BCC
HCP
HCP
HCP
HCP
HCP
HCP
FCC
HCP
HCP
BCC
BCC
HCP
HCP
FCC
FCC
FCC
RHO
HCP
FCC
RHO
SC
FCC
FCC
BCC

FCC
FCC
TETR
ORTH
ORTH
MON
HCP
HCP