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()
```

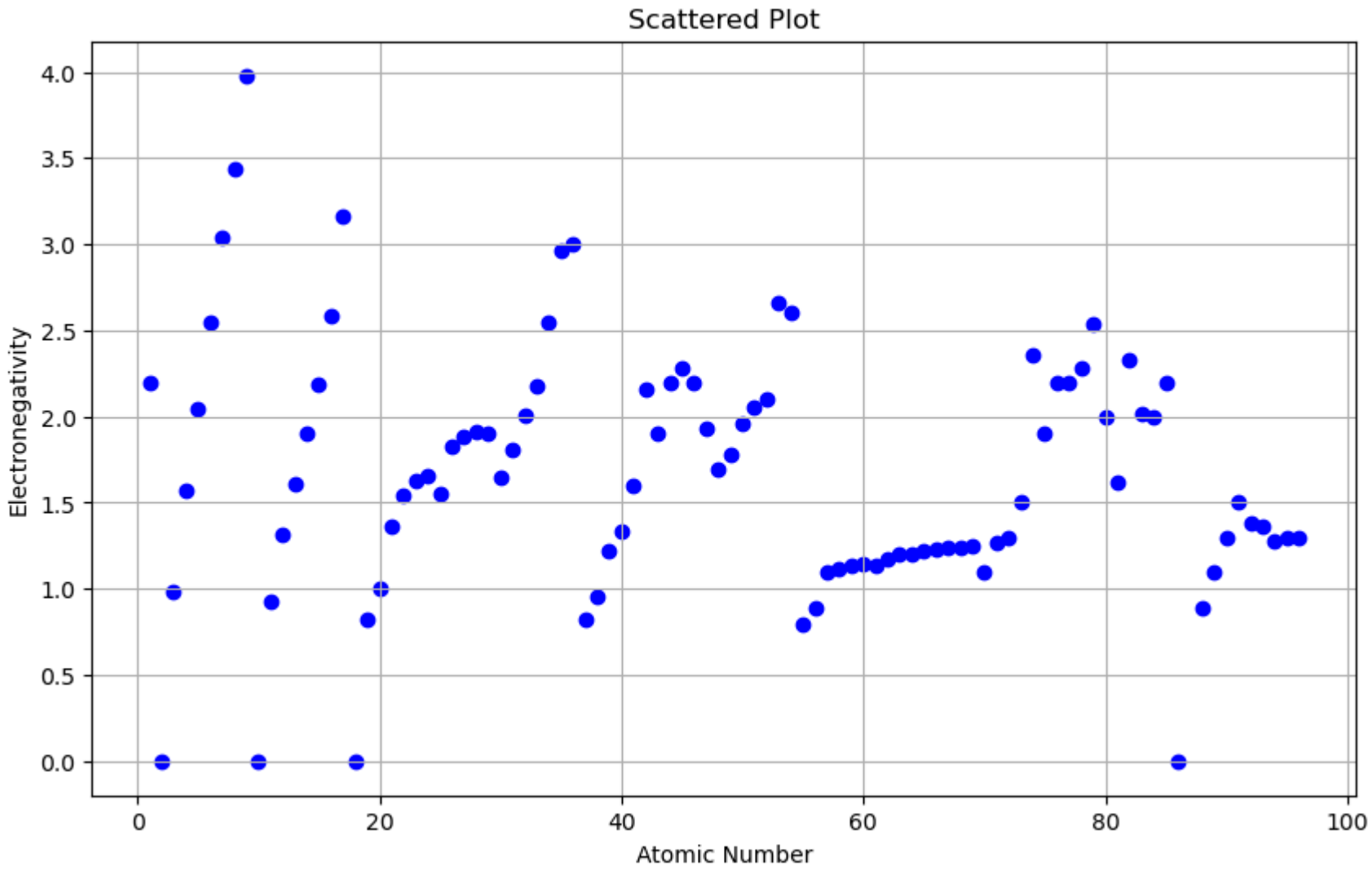
Out[26]:

	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.

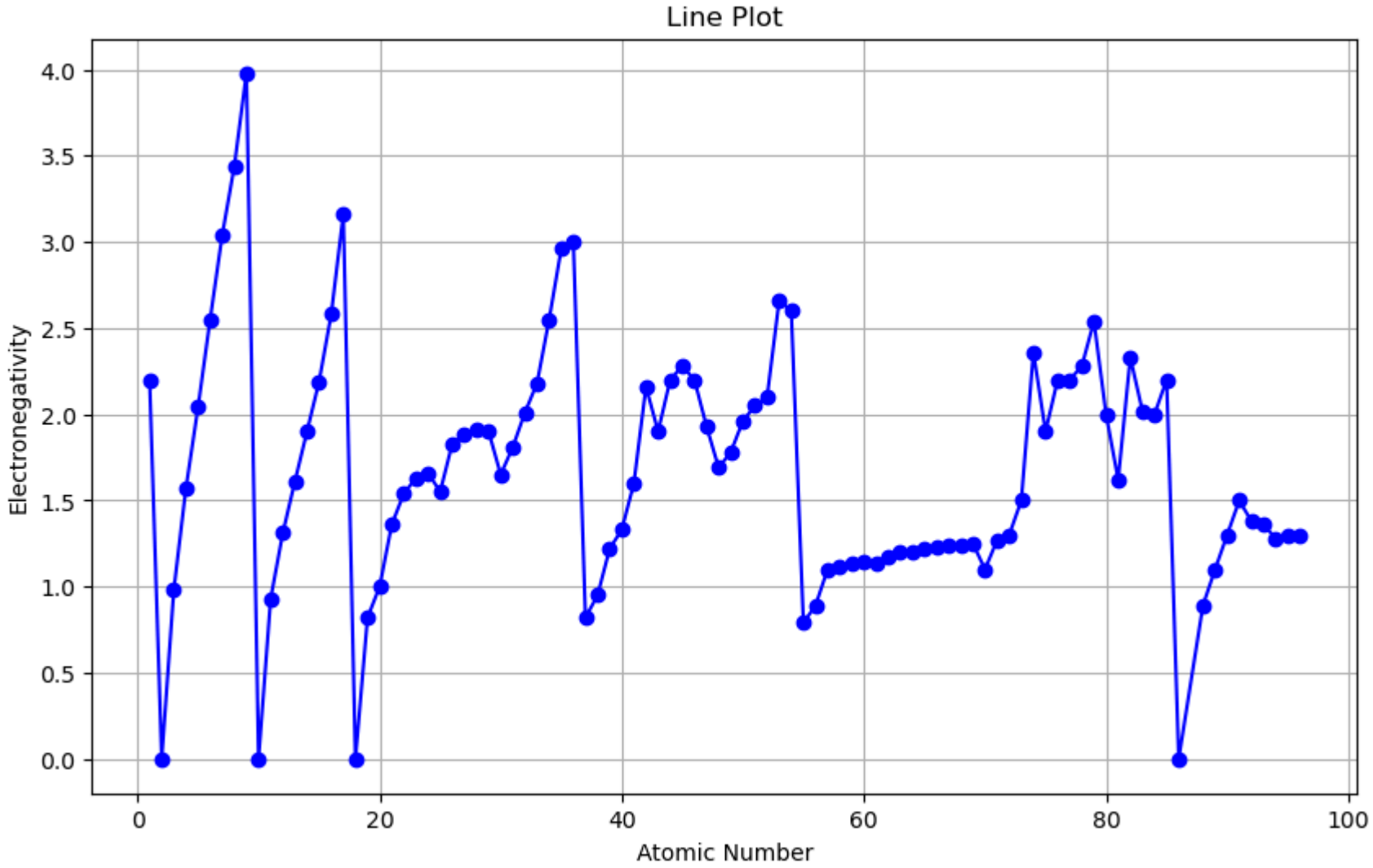
In [27]:
```
plt.figure(figsize=(10, 6))
plt.scatter(data['Atomic_Number'], data['Electronegativity'], color='blue')
plt.title('Scattered Plot')
plt.xlabel('Atomic Number')
```

```
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()
```

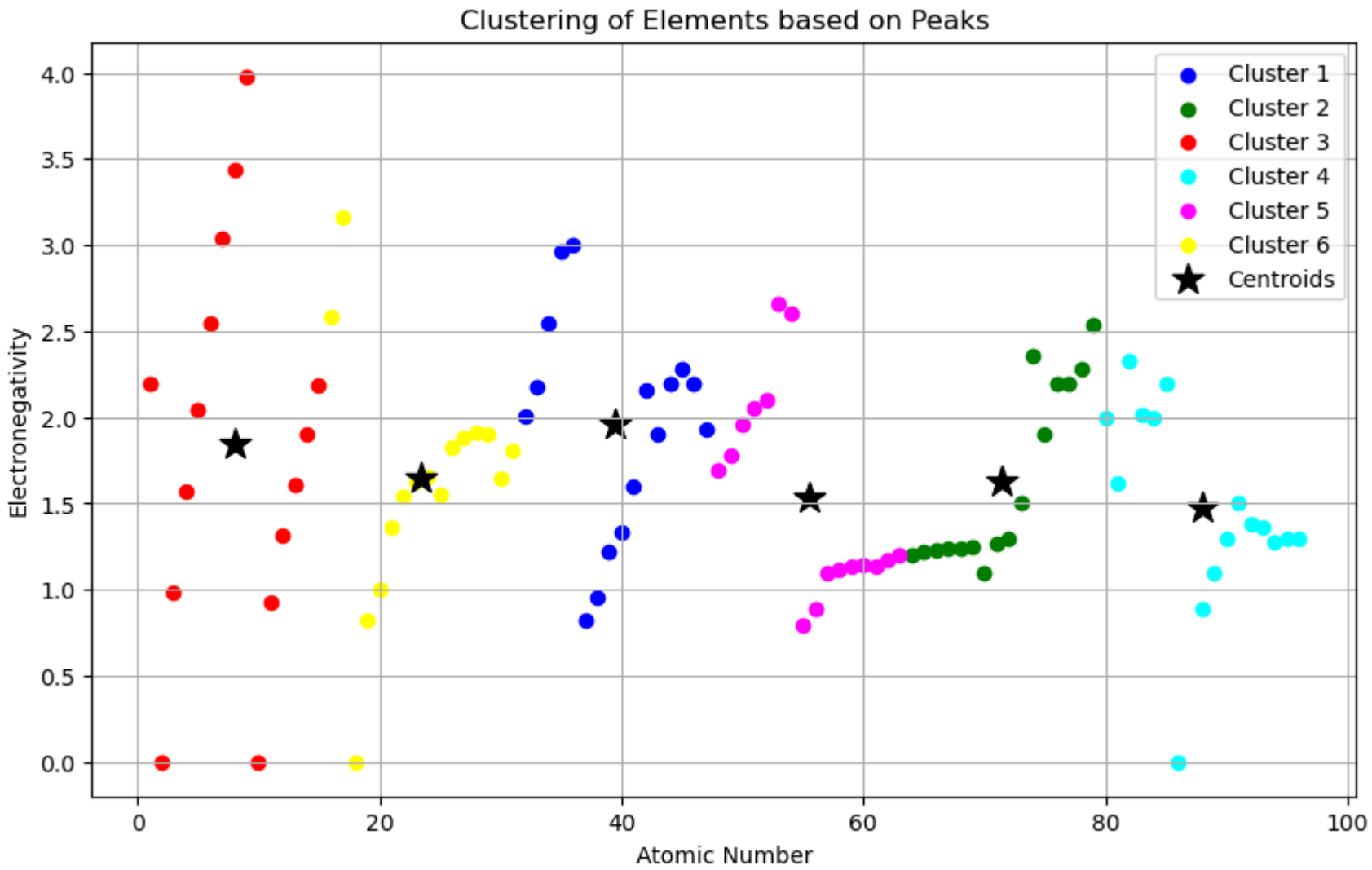


Number of Peaks: 6

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()
```



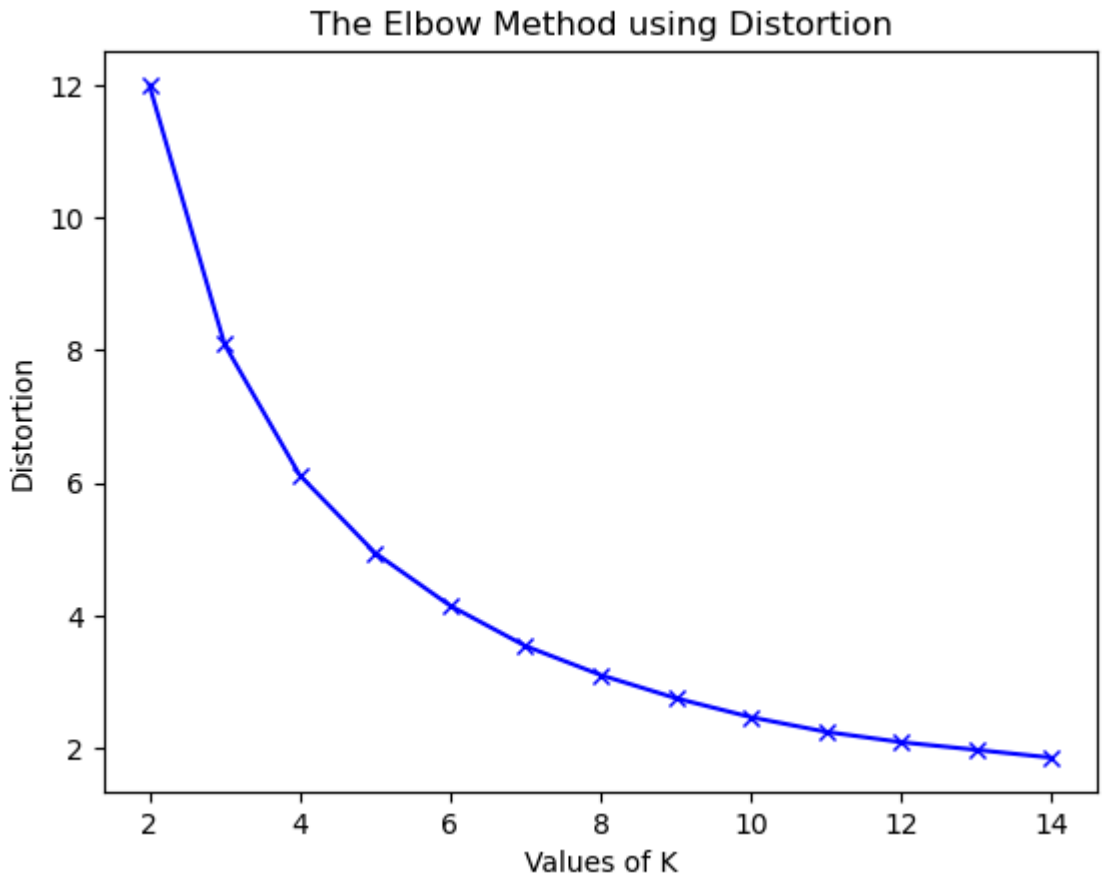
In [30]:
```
data.to_csv('data.csv', index=False)
```

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])
```

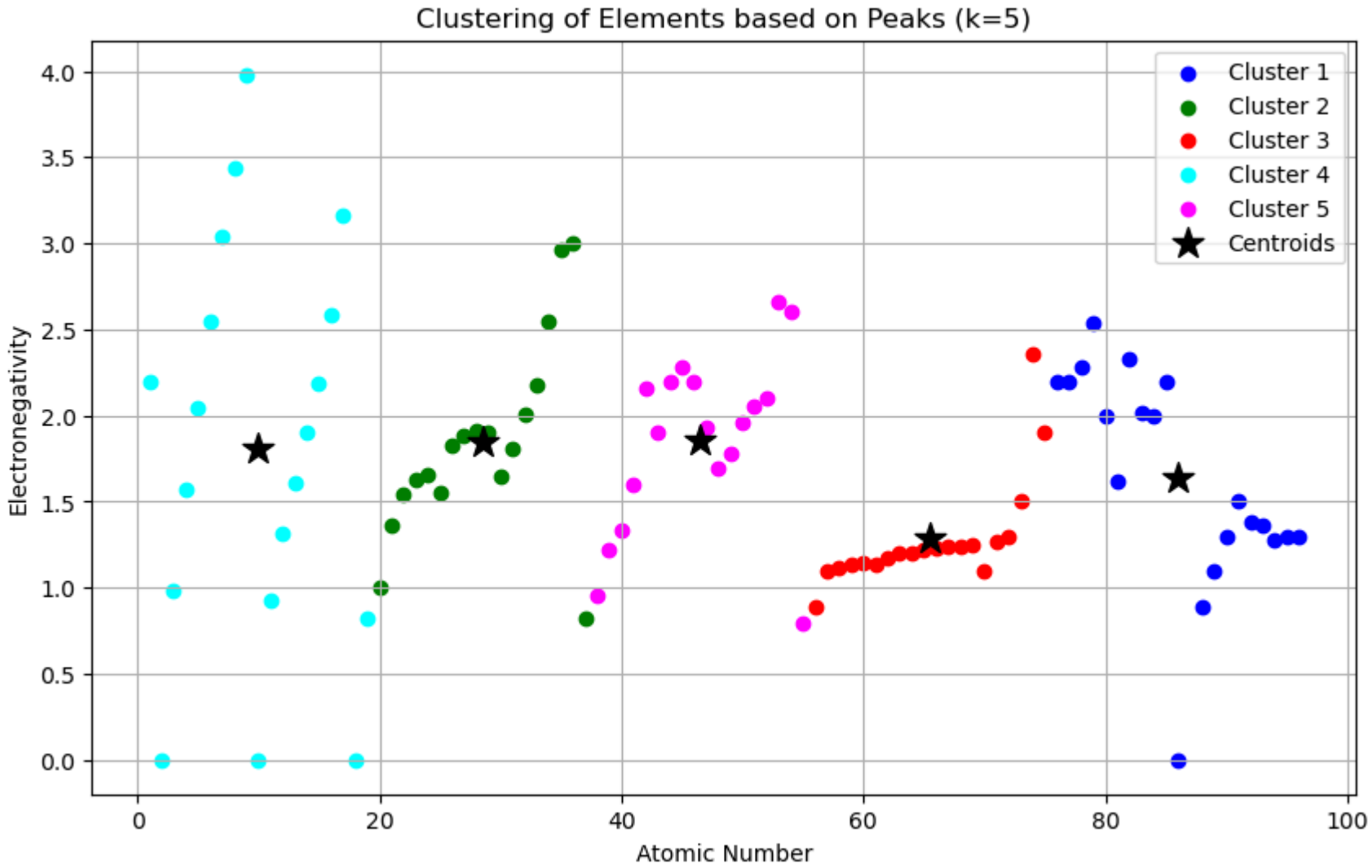
In [32]:
```
plt.plot(K, distortions, 'bx-')
plt.xlabel('Values of K')
plt.ylabel('Distortion')
plt.title('The Elbow Method using Distortion')
plt.show()
```



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

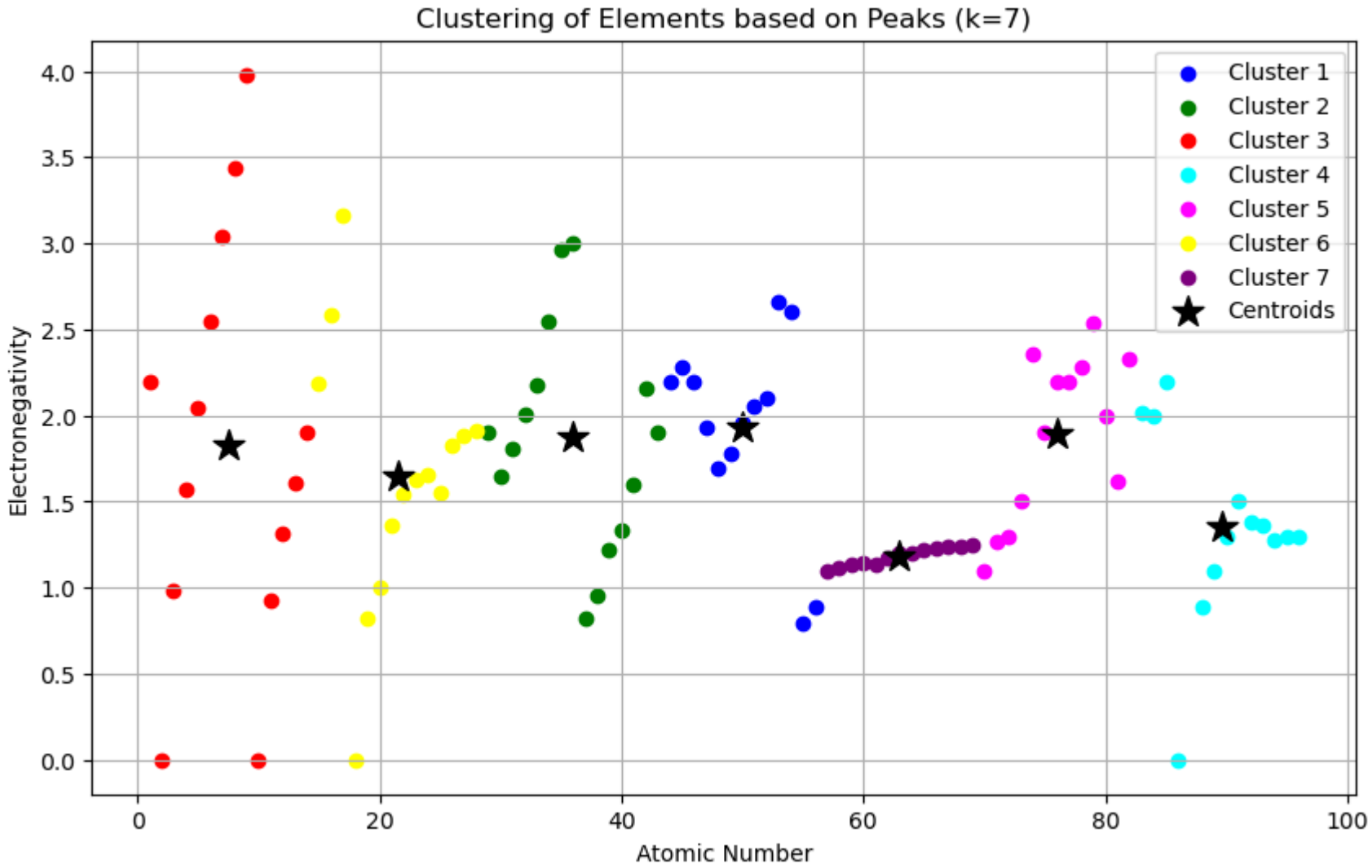
Let's plot for k=5 and k=7

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 [ ]: