```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

```
d=pd.read_csv('/content/university admission.csv')
```

```
d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 15,\n  \"fields\": [\n
{\n      \"column\": \"GRE Score\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 301,\n
\"max\": 337,\n        \"num_unique_values\": 15,\n
\"samples\": [\n          323,\n          327,\n          337\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"TOEFL Score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5,\n        \"min\": 100,\n        \"max\": 118,\n
\"num_unique_values\": 14,\n        \"samples\": [\n          108,\n
111,\n          118\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"University Rating\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1,\n        \"min\": 1,\n
\"max\": 5,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          3,\n          1,\n          2\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"SOP\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.7761320457119086,\n
\"min\": 2.0,\n        \"max\": 4.5,\n        \"num_unique_values\":
6,\n        \"samples\": [\n          4.5,\n          4.0,\n
2.5\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"LOR\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.9411239481143203,\n        \"min\": 1.5,\n        \"max\":
4.5,\n        \"num_unique_values\": 7,\n        \"samples\": [\n
4.5,\n          3.5,\n          1.5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"CGPA\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.5541376827364941,\n
\"min\": 7.8,\n        \"max\": 9.65,\n        \"num_unique_values\":
12,\n        \"samples\": [\n          9.0,\n          8.4,\n
9.65\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Research\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Admission Chance\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"d"}
```

```
d.shape
```

```
(15, 8)
```

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   GRE Score          15 non-null     int64  
 1   TOEFL Score        15 non-null     int64  
 2   University Rating  15 non-null     int64  
 3   SOP                15 non-null     float64
 4   LOR                15 non-null     float64
 5   CGPA               15 non-null     float64
 6   Research           15 non-null     int64  
 7   Admission Chance   15 non-null     int64  
dtypes: float64(3), int64(5)
memory usage: 1.1 KB
```

```
d.isnull().sum()
```

```
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Admission Chance     0
dtype: int64
```

```
x=d.iloc[:,[3,3]].values
```

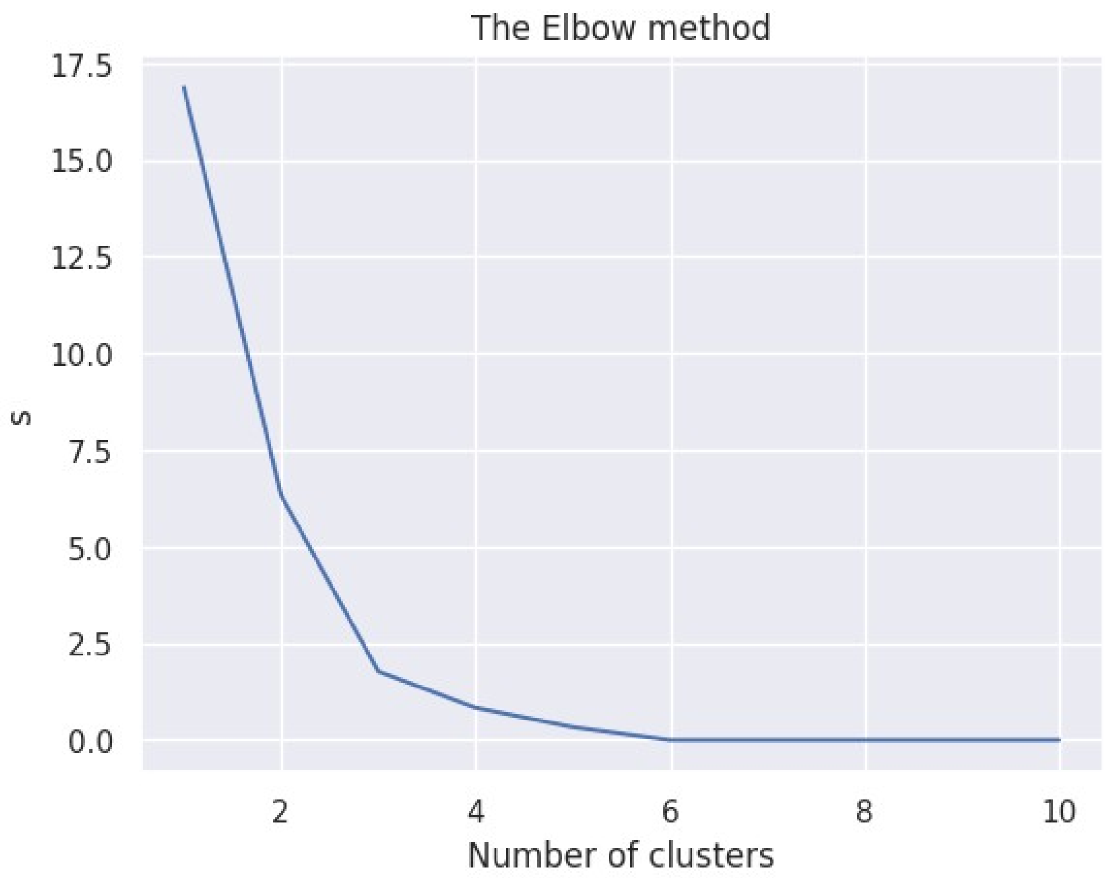
```
print(x)
```

```
[[4.5 4.5]
 [4.  4. ]
 [3.  3. ]
 [3.5 3.5]
 [2.  2. ]
 [4.5 4.5]
 [3.  3. ]
 [3.  3. ]
 [2.  2. ]
 [3.5 3.5]
 [3.5 3.5]
 [4.  4. ]
 [3.  3. ]
 [3.  3. ]
 [2.5 2.5]]
```

```python
s=[]
for i in range(1,11):
  kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)
  kmeans.fit(x)
  s.append(kmeans.inertia_)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/base.py:1389:
ConvergenceWarning: Number of distinct clusters (6) found smaller than
n_clusters (7). Possibly due to duplicate points in X.
  return fit_method(estimator, *args, **kwargs)
/usr/local/lib/python3.11/dist-packages/sklearn/base.py:1389:
ConvergenceWarning: Number of distinct clusters (6) found smaller than
n_clusters (8). Possibly due to duplicate points in X.
  return fit_method(estimator, *args, **kwargs)
/usr/local/lib/python3.11/dist-packages/sklearn/base.py:1389:
ConvergenceWarning: Number of distinct clusters (6) found smaller than
n_clusters (9). Possibly due to duplicate points in X.
  return fit_method(estimator, *args, **kwargs)
/usr/local/lib/python3.11/dist-packages/sklearn/base.py:1389:
ConvergenceWarning: Number of distinct clusters (6) found smaller than
n_clusters (10). Possibly due to duplicate points in X.
  return fit_method(estimator, *args, **kwargs)
```

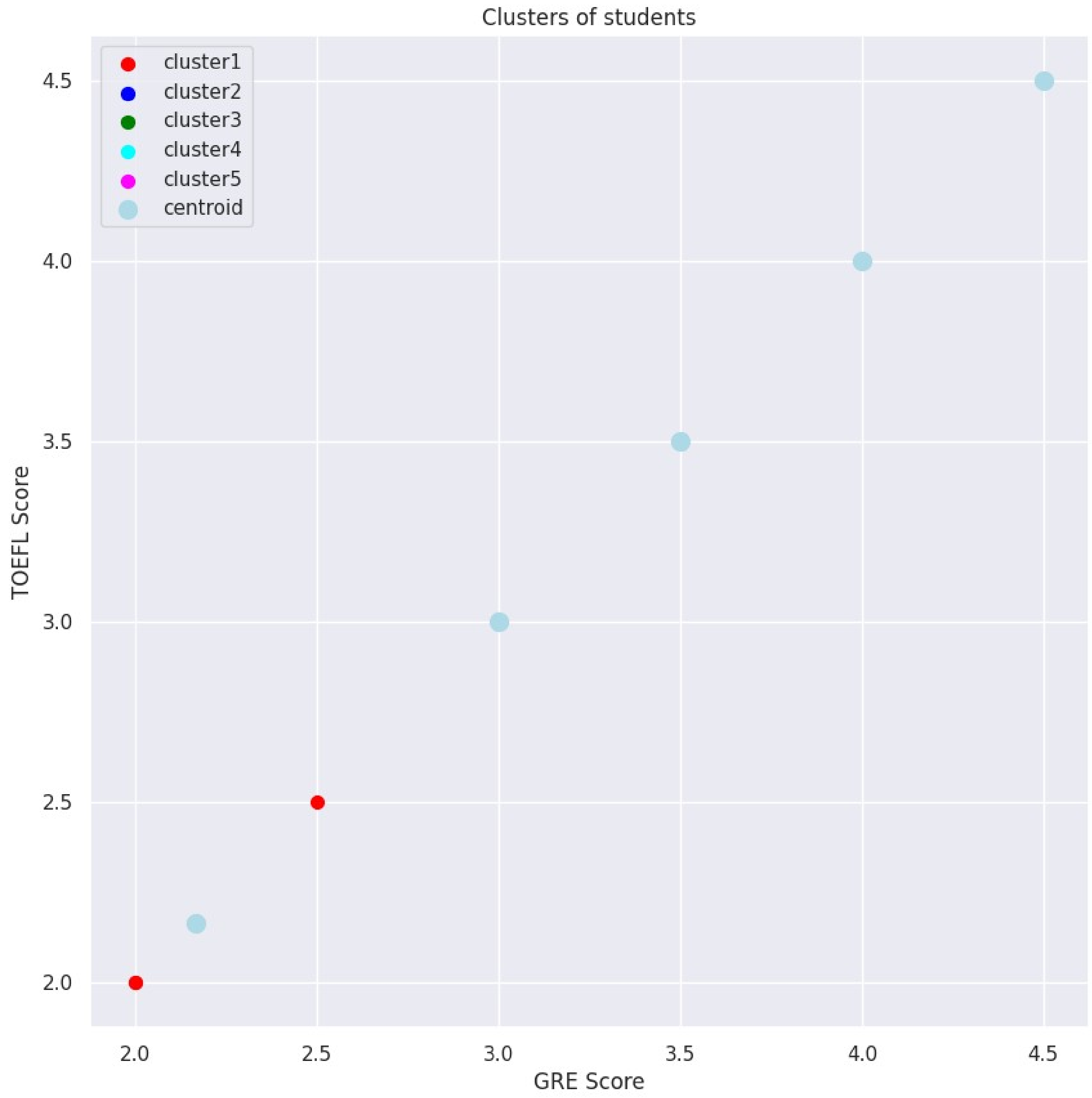
```python
sns.set()
plt.plot(range(1,11),s)
plt.title('The Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('s')
plt.show()
```



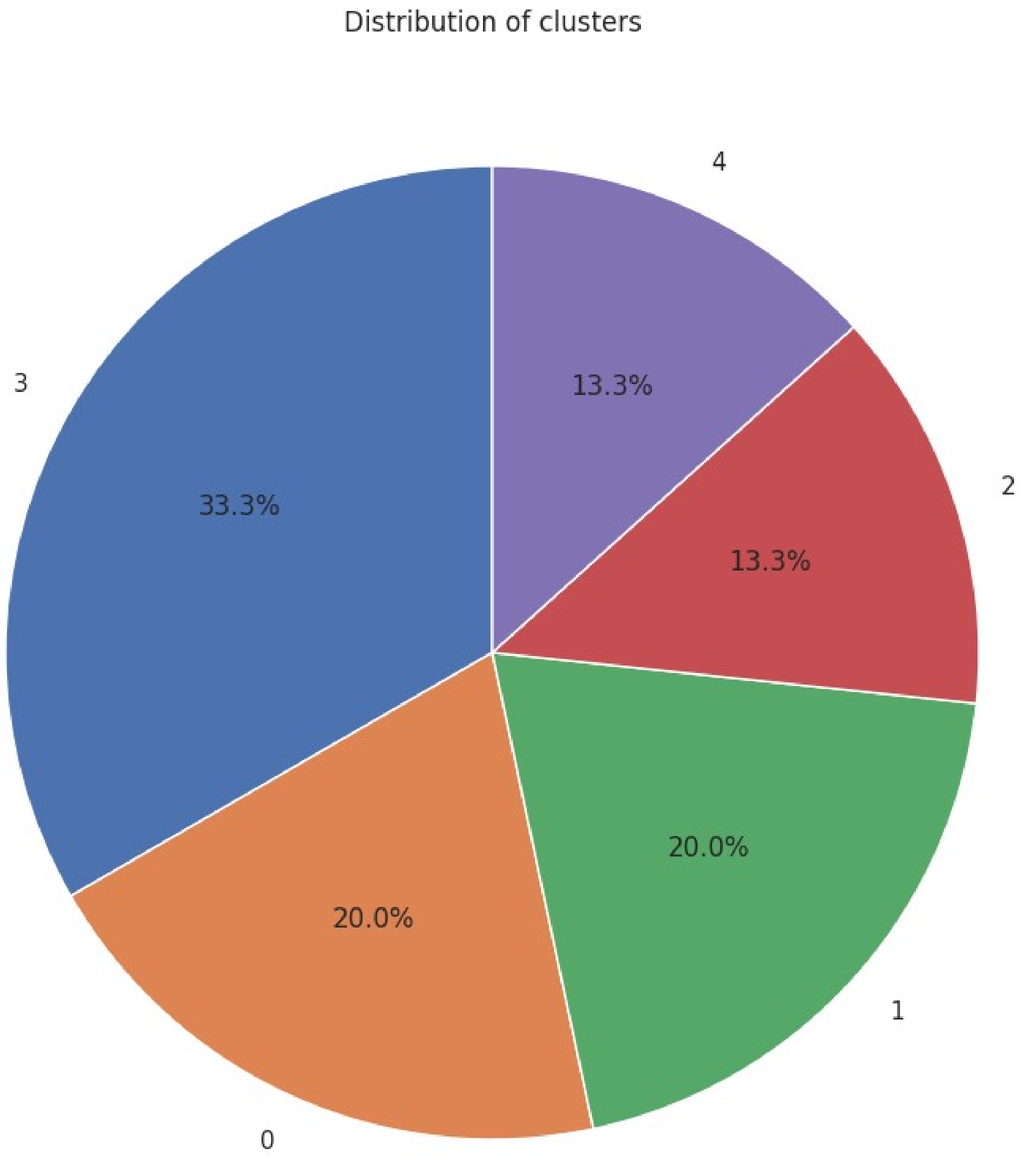
```
kmeans=KMeans(n_clusters=5,init='k-means++',random_state=0)
y=kmeans.fit_predict(x)
print(y)
```

```
[2 4 3 1 0 2 3 3 0 1 1 4 3 3 0]
```

```
plt.figure(figsize=(10,10))
plt.scatter(x[y==0,0],x[y==0,1],s=50,c='red',label='cluster1')
plt.scatter(x[y==1,0],x[y==1,1],s=50,c='blue',label='cluster2')
plt.scatter(x[y==2,0],x[y==2,1],s=50,c='green',label='cluster3')
plt.scatter(x[y==3,0],x[y==3,1],s=50,c='cyan',label='cluster4')
plt.scatter(x[y==4,0],x[y==4,1],s=50,c='magenta',label='cluster5')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],
s=100,c='lightblue',label='centroid')
plt.title('Clusters of students')
plt.xlabel('GRE Score')
plt.ylabel('TOEFL Score')
plt.legend()
plt.show()
```



```
c=pd.Series(y).value_counts()
plt.figure(figsize=(10,10))
plt.pie(c,labels=c.index,autopct='%1.1f%%',startangle=90)
plt.title('Distribution of clusters')
plt.show()
```


Distribution of clusters
4
13.3%
3
33.3%
2
13.3%
20.0%
20.0%
1
0