Name: Saloni Vishwakarma

Batch-Roll number: C1-13

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # for data visualization
import seaborn as sns # for statistical data visualization
%matplotlib inline
```

```
df = pd.read_csv('Live.csv')
```

```
df.drop(['Column1', 'Column2', 'Column3', 'Column4'], axis=1, inplace=True)
```

```
df.drop(['status_id', 'status_published'], axis=1, inplace=True)
```

```
X = df
```

```
y = df['status_type']
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
X['status_type'] = le.fit_transform(X['status_type'])
```

```
y = le.transform(y)
```

```
cols = X.columns
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
ms = MinMaxScaler()
```

```
X = ms.fit_transform(X)
```

```
X = pd.DataFrame(X, columns=[cols])
```

```
X.head()
```

	status_type	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys
0	1.000000	0.112314	0.024393	0.076519	0.091720	0.140030	0.010791	0.006369	0.019608	0.0
1	0.333333	0.031847	0.000000	0.000000	0.031847	0.000000	0.000000	0.000000	0.000000	0.0
2	1.000000	0.048195	0.011243	0.016647	0.043312	0.031963	0.003597	0.006369	0.000000	0.0
3	0.333333	0.023567	0.000000	0.000000	0.023567	0.000000	0.000000	0.000000	0.000000	0.0
4	0.333333	0.045223	0.000000	0.000000	0.043312	0.013699	0.000000	0.000000	0.000000	0.0

```
pip install sklearn.cluster
```

```
ERROR: Could not find a version that satisfies the requirement sklearn.cluster (from versions: none)
ERROR: No matching distribution found for sklearn.cluster
```

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7050 entries, 0 to 7049
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   (status_type,)    7050 non-null   float64
 1   (num_reactions,)  7050 non-null   float64
 2   (num_comments,)   7050 non-null   float64
 3   (num_shares,)     7050 non-null   float64
 4   (num_likes,)      7050 non-null   float64
 5   (num_loves,)      7050 non-null   float64
 6   (num_wows,)       7050 non-null   float64
```

```
 7   (num_hahas,)      7050 non-null   float64
 8   (num_sads,)       7050 non-null   float64
 9   (num_angrys,)     7050 non-null   float64
dtypes: float64(10)
memory usage: 550.9 KB
```

```
X.shape
```

```
(7050, 10)
```

```
X.describe()
```

	status_type	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angr
count	7050.000000	7050.000000	7050.000000	7050.000000	7050.000000	7050.000000	7050.000000	7050.000000	7050.000000	7050.000
mean	0.568322	0.048857	0.010689	0.011689	0.045657	0.019374	0.004638	0.004436	0.004778	0.003
std	0.314133	0.098222	0.042384	0.038435	0.095429	0.060842	0.031366	0.025205	0.031317	0.023
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	0.333333	0.003609	0.000000	0.000000	0.003609	0.000000	0.000000	0.000000	0.000000	0.000
50%	0.333333	0.012633	0.000191	0.000000	0.012314	0.000000	0.000000	0.000000	0.000000	0.000
75%	1.000000	0.046497	0.001096	0.001168	0.039225	0.004566	0.000000	0.000000	0.000000	0.000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters=2, random_state=0)
```

```
kmeans.fit(X)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
```

```
KMeans(n_clusters=2, random_state=0)
```

```
kmeans.cluster_centers_
```

```
array([[3.28506857e-01, 3.90710874e-02, 7.54854864e-04, 7.53667113e-04,
        3.85438884e-02, 2.17448568e-03, 2.43721364e-03, 1.20039760e-03,
        2.75348016e-03, 1.45313276e-03],
       [9.54921576e-01, 6.46330441e-02, 2.67028654e-02, 2.93171709e-02,
        5.71231462e-02, 4.71007076e-02, 8.18581889e-03, 9.65207685e-03,
        8.04219428e-03, 7.19501847e-03]])
```

```
kmeans.inertia_
```

```
237.75726404419646
```

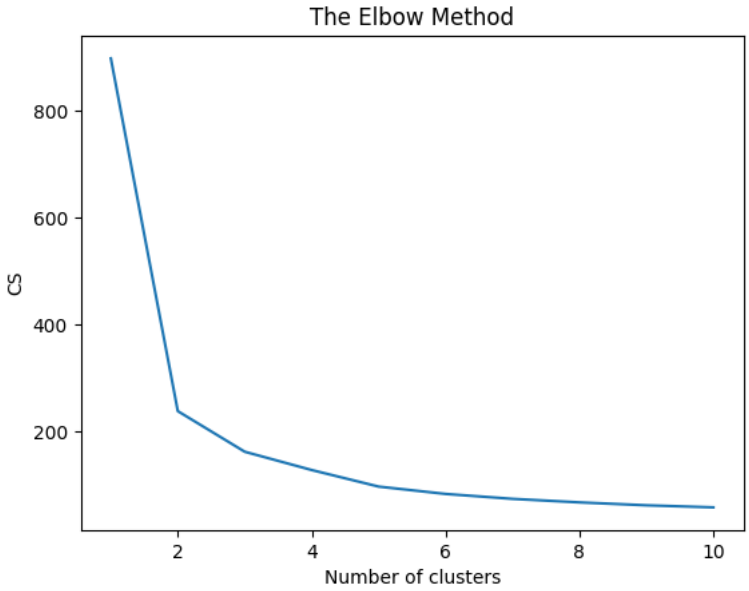
```
labels = kmeans.labels_

correct_labels = sum(y == labels)
print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))
```

```
Result: 63 out of 7050 samples were correctly labeled.
```

```
print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))
```

```
Accuracy score: 0.01
```

```
from sklearn.cluster import KMeans
cs = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    cs.append(kmeans.inertia_)
plt.plot(range(1, 11), cs)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('CS')
plt.show()
```




```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=4,random_state=0)

kmeans.fit(X)

labels = kmeans.labels_

# check how many of the samples were correctly labeled

correct_labels = sum(y == labels)

print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))

print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
Result: 4340 out of 7050 samples were correctly labeled.
Accuracy score: 0.62
```

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3,random_state=0)

kmeans.fit(X)

labels = kmeans.labels_

# check how many of the samples were correctly labeled

correct_labels = sum(y == labels)

print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))

print('Accuracy score: {0:0.2f}'. format(correct_labels/float(y.size)))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
Result: 138 out of 7050 samples were correctly labeled.
Accuracy score: 0.02
```