Major project 2

Name-Anusha Patnaik
College – GIET University

Branch-CSE(AIML)
Year: 2021-2025

```
#dataset = https://www.kaggle.com/code/prashant111/k-means-clustering-with-python?scriptVersionId=48823469&cellId=18

import pandas as pd
df = pd.read_csv('/content/Live.csv')
df

df.shape#7050 rows and 16 cols

df.info()

df.isnull().sum() #checking for missing values

df.describe()

# view the labels in the variable

df['status_id'].unique()

# view how many different types of variables are there

len(df['status_id'].unique())

# view the labels in the variable

df['status_published'].unique()

# view how many different types of variables are there

len(df['status_published'].unique())

# view the labels in the variable

df['status_type'].unique()

# view how many different types of variables are there

len(df['status_type'].unique())

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

df.info()

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

df.info()

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)

X.info()

#feature scaling
cols = X.columns

from sklearn.preprocessing import MinMaxScaler

ms = MinMaxScaler()

X = ms.fit_transform(X)

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

X.head()

#ELBOW Method

from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
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=2,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)))
```

```
#with 4 clusters

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

kmeans.fit(X)

# check how many of the samples were correctly labeled
labels = kmeans.labels_

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)))

#with 3 clusters

#K-Means model with 3 clusters
kmeans = KMeans(n_clusters=3, random_state=0)

kmeans.fit(X)

# check how many of the samples were correctly labeled
labels = kmeans.labels_

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)))
```

OUTPUTS:

	status_id	status_type	status_published	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_
0	246675545449582_1649696485147474	video	4/22/2018 6:00	529	512	262	432	92	3	1	
1	246675545449582_1649426988507757	photo	4/21/2018 22:45	150	0	0	150	0	0	0	
2	246675545449582_1648730588577397	video	4/21/2018 6:17	227	236	57	204	21	1	1	
3	246675545449582_1648576705259452	photo	4/21/2018 2:29	111	0	0	111	0	0	0	
4	246675545449582_1645700502213739	photo	4/18/2018 3:22	213	0	0	204	9	0	0	
...	...	...	...	...	...	...	...	...	...	...	
7045	1050855161656896_1061863470556065	photo	9/24/2016 2:58	89	0	0	89	0	0	0	
7046	1050855161656896_1061334757275603	photo	9/23/2016 11:19	16	0	0	14	1	0	1	
7047	1050855161656896_1060126464063099	photo	9/21/2016 23:03	2	0	0	1	1	0	0	
7048	1050855161656896_1058663487542730	photo	9/20/2016 0:43	351	12	22	349	2	0	0	
7049	1050855161656896_1050858841656528	photo	9/10/2016 10:30	17	0	0	17	0	0	0	

7050 rows × 16 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7050 entries, 0 to 7049
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   status_id         7050 non-null   object
 1   status_type       7050 non-null   object
 2   status_published  7050 non-null   object
 3   num_reactions     7050 non-null   int64
 4   num_comments      7050 non-null   int64
 5   num_shares        7050 non-null   int64
 6   num_likes         7050 non-null   int64
 7   num_loves         7050 non-null   int64
 8   num_wows          7050 non-null   int64
 9   num_hahas         7050 non-null   int64
 10  num_sads          7050 non-null   int64
 11  num_angrys        7050 non-null   int64
 12  Column1           0 non-null      float64
 13  Column2           0 non-null      float64
 14  Column3           0 non-null      float64
 15  Column4           0 non-null      float64
dtypes: float64(4), int64(9), object(3)
memory usage: 881.4+ KB
```

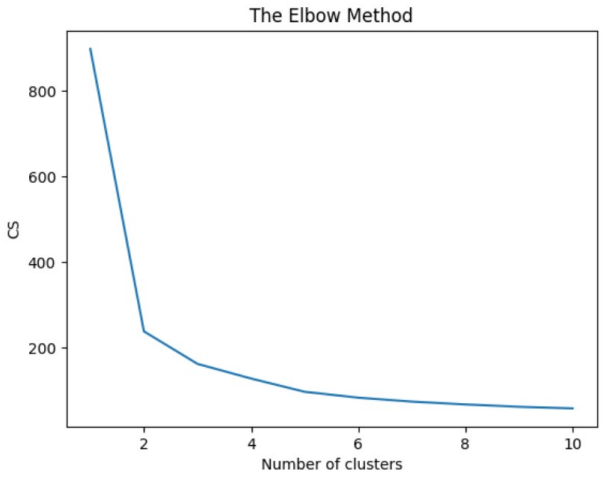
```
[34] df.info()
```

```
<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   object
 1   num_reactions   7050 non-null   int64
 2   num_comments    7050 non-null   int64
 3   num_shares      7050 non-null   int64
 4   num_likes       7050 non-null   int64
 5   num_loves       7050 non-null   int64
 6   num_wows        7050 non-null   int64
 7   num_hahas       7050 non-null   int64
 8   num_sads        7050 non-null   int64
 9   num_angrys      7050 non-null   int64
dtypes: int64(9), object(1)
memory usage: 550.9+ KB
```

```
[35] X = df

     y = df['status_type']
```

```
[4]
```



```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set t
  warnings.warn(
Result: 4340 out of 7050 samples were correctly labeled.
Accuracy score: 0.62
```

```
#K-Means model with 3 clusters
kmeans = KMeans(n_clusters=3, random_state=0)

kmeans.fit(X)

# check how many of the samples were correctly labeled
labels = kmeans.labels_

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 from 10 to 'auto' in 1.4. Set t
  warnings.warn(
Result: 138 out of 7050 samples were correctly labeled.
Accuracy score: 0.02
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set t
  warnings.warn(
Result: 63 out of 7050 samples were correctly labeled.
Accuracy score: 0.01
```