

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import os
6 #from sklearn.cluster import KMeans
7 #from yellowbrick.cluster import KElbowVisualizer
8 #from sklearn import metrics

```

```

1 df = pd.read_csv('/content/Live.csv')
2 df.drop(['status_id', 'status_published', 'Column1', 'Column2', 'Column3', 'Column4'], axis=1, i
3 df.head()

```

	status_type	num_reactions	num_comments	num_shares	num_likes	num_loves	num_w
0	video	529	512	262	432	92	
1	photo	150	0	0	150	0	
2	video	227	236	57	204	21	
3	photo	111	0	0	111	0	
4	photo	213	0	0	204	9	

```

1 from sklearn.preprocessing import LabelEncoder
2 le = LabelEncoder()
3 df['status_type'] = le.fit_transform(df['status_type'])
4 df.head()

```

	status_type	num_reactions	num_comments	num_shares	num_likes	num_loves	num_w
0	3	529	512	262	432	92	
1	1	150	0	0	150	0	
2	3	227	236	57	204	21	
3	1	111	0	0	111	0	
4	1	213	0	0	204	9	

```

1 df.dropna(inplace=True)
2 df.head()

```

```

status_type num reactions num comments num shares num likes num loves num w
1 X = df
2 y = df['status_type']

1 y

0      3
1      1
2      3
3      1
4      1
..
7045    1
7046    1
7047    1
7048    1
7049    1
Name: status_type, Length: 7050, dtype: int64

```

```

1 from sklearn.preprocessing import MinMaxScaler
2 ms = MinMaxScaler()
3 X = ms.fit_transform(X)

```

```

1 from sklearn.cluster import KMeans
2 kmeans = KMeans(n_clusters=2, random_state=0)
3 kmeans.fit(X)

```

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

```

1 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]])

```

```

1 kmeans.inertia_

237.75726404419646

```

```

1 labels = kmeans.labels_
2 # check how many of the samples were correctly labeled
3 correct_labels = sum(y == labels)
4 print("Result: %d out of %d samples were correctly labeled." % (correct_labels, y.size))

```

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

