

dataset using: <https://www.kaggle.com/datasets/harrywang/wine-dataset-for-clustering>

Read in the data and preprocessing

```
In [ ]: import pandas as pd
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Read the CSV file
df = pd.read_csv('wine-clustering.csv')
df.fillna(0, inplace=True)
```

EDA

```
In [ ]: df.head()
```

```
Out[ ]:
```

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OL
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	

```
In [ ]: df.describe()
```

Out[]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	5.000000
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.300000
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.200000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.200000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.600000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.200000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.000000

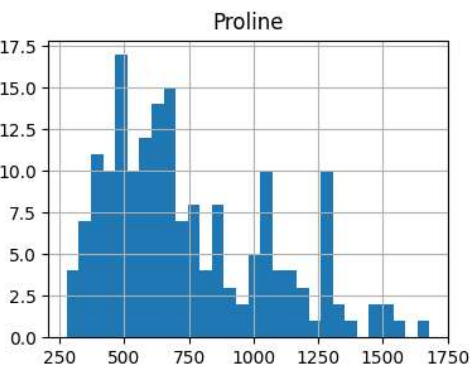
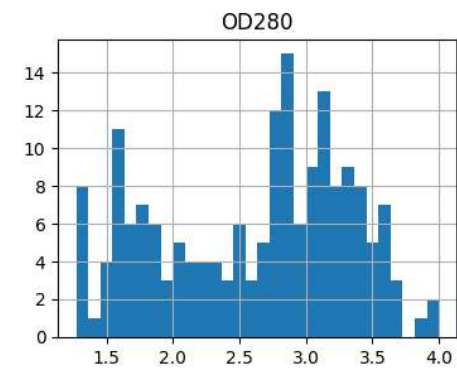
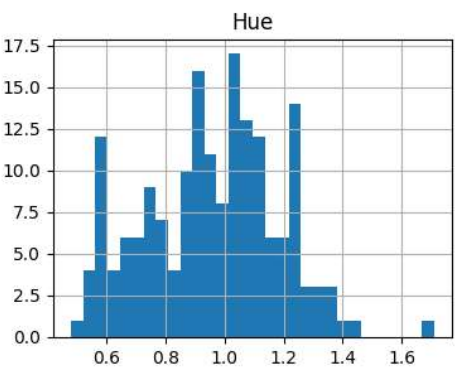
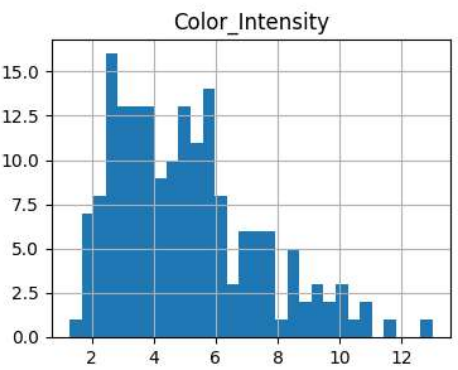
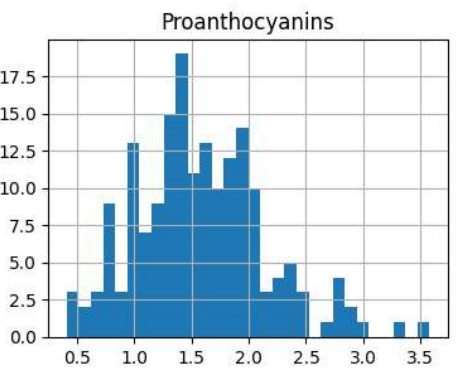
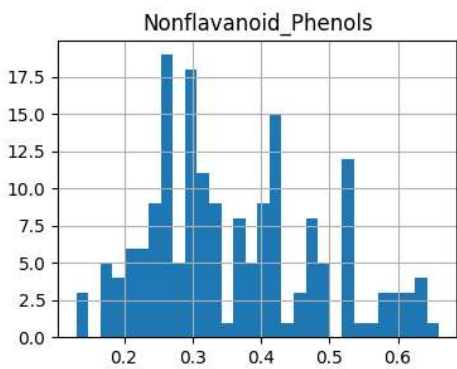
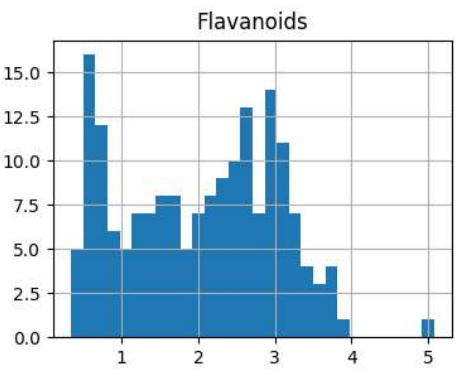
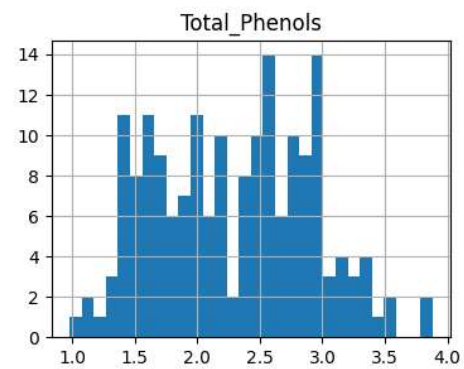
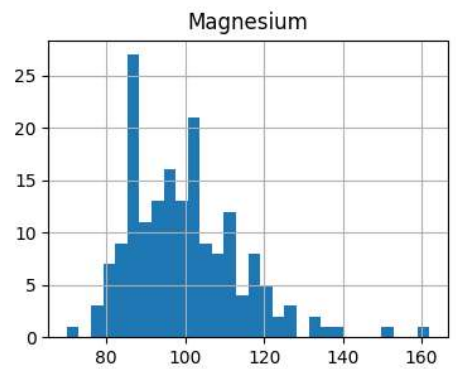
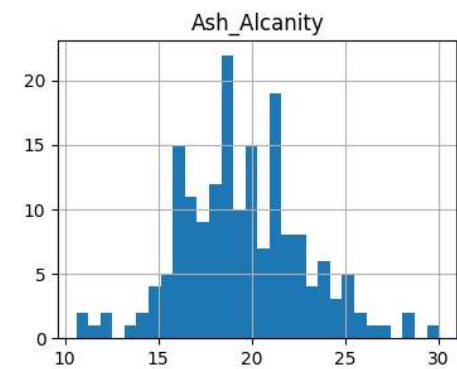
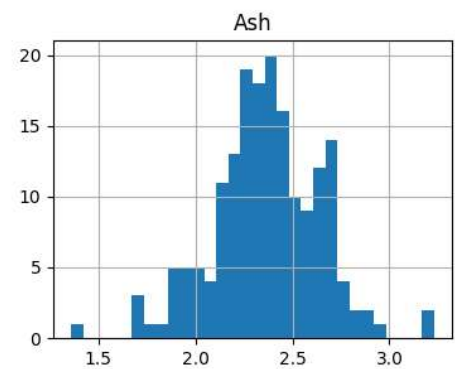
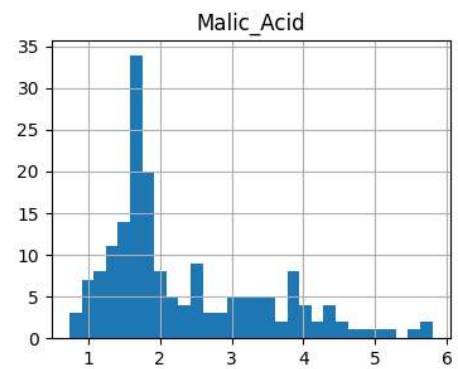
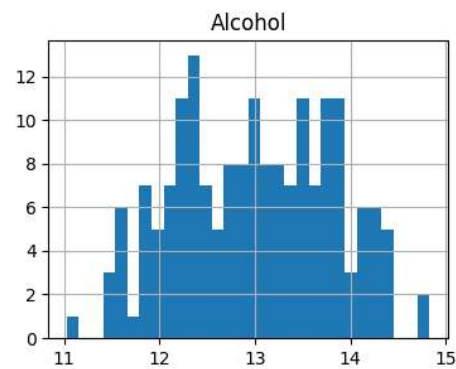
◀

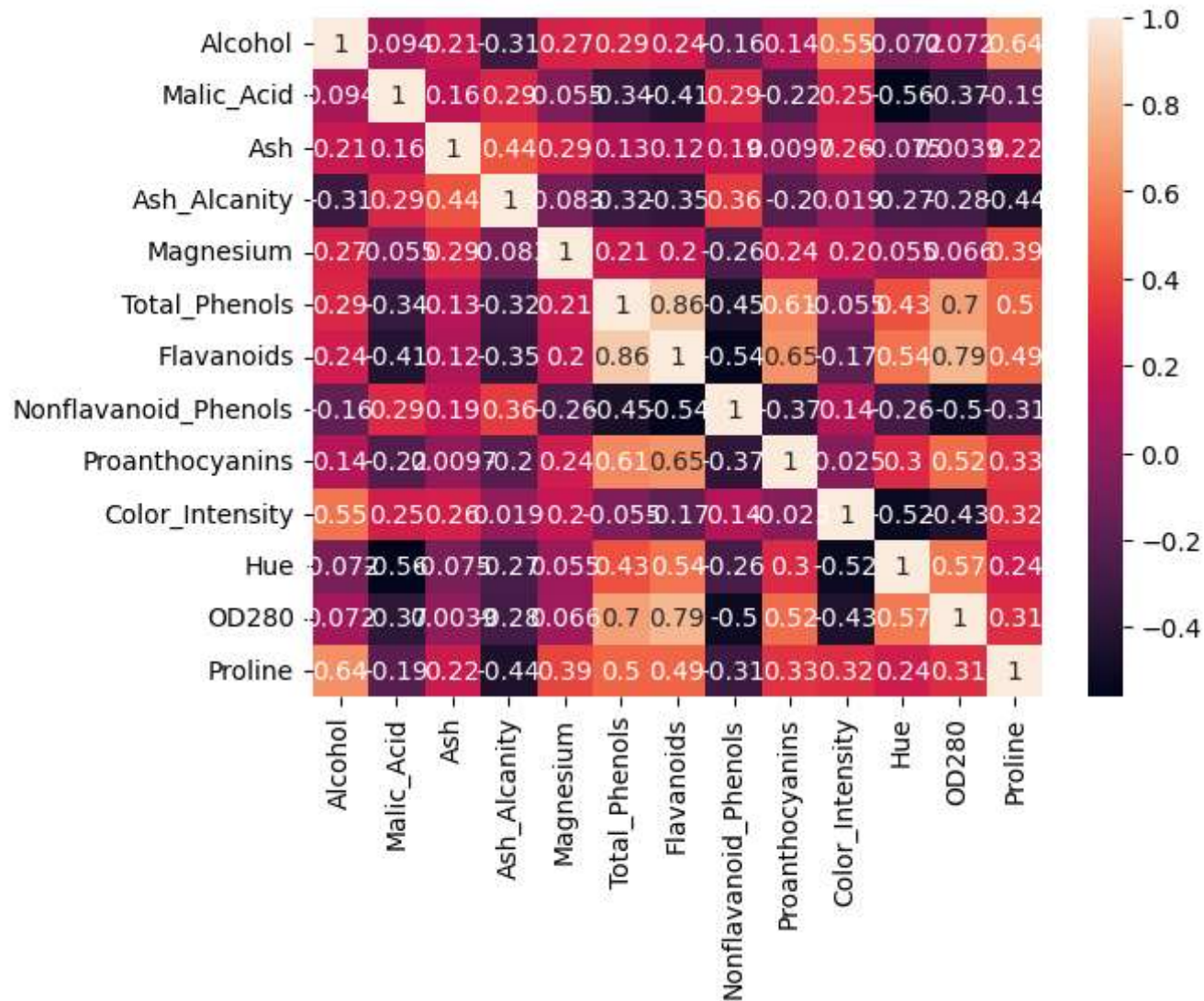
▶

In []:

```
df.hist(bins=30, figsize=(20, 15))
plt.show()

corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True)
plt.show()
```





KMeans clustering

```
In [ ]: kmeans = KMeans(n_clusters=3)
labels = kmeans.fit_predict(df)
df['Cluster'] = labels

print(df.head())
```

c:\Users\Patrick\Documents\GitHub\DataMiningNTNU\env\lib\site-packages\sklearn\cluster_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
super()._check_params_vs_input(X, default_n_init=10)

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	\
0	14.23	1.71	2.43	15.6	127	2.80	
1	13.20	1.78	2.14	11.2	100	2.65	
2	13.16	2.36	2.67	18.6	101	2.80	
3	14.37	1.95	2.50	16.8	113	3.85	
4	13.24	2.59	2.87	21.0	118	2.80	

	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	\
0	3.06	0.28	2.29	5.64	1.04	
1	2.76	0.26	1.28	4.38	1.05	
2	3.24	0.30	2.81	5.68	1.03	
3	3.49	0.24	2.18	7.80	0.86	
4	2.69	0.39	1.82	4.32	1.04	

	OD280	Proline	Cluster
0	3.92	1065	0
1	3.40	1050	0
2	3.17	1185	0
3	3.45	1480	0
4	2.93	735	2

```
In [ ]: kmeans = KMeans(n_clusters=3)
labels = kmeans.fit_predict(df)
df['Cluster'] = labels

print(df.head())
```

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	\
0	14.23	1.71	2.43	15.6	127	2.80	
1	13.20	1.78	2.14	11.2	100	2.65	
2	13.16	2.36	2.67	18.6	101	2.80	
3	14.37	1.95	2.50	16.8	113	3.85	
4	13.24	2.59	2.87	21.0	118	2.80	

	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	\
0	3.06	0.28	2.29	5.64	1.04	
1	2.76	0.26	1.28	4.38	1.05	
2	3.24	0.30	2.81	5.68	1.03	
3	3.49	0.24	2.18	7.80	0.86	
4	2.69	0.39	1.82	4.32	1.04	

	OD280	Proline	Cluster
0	3.92	1065	1
1	3.40	1050	1
2	3.17	1185	1
3	3.45	1480	1
4	2.93	735	2

```
c:\Users\Patrick\Documents\GitHub\DataMiningNTNU\env\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
super()._check_params_vs_input(X, default_n_init=10)
```

There are too many features and hard to visualize, so we apply PCA first.

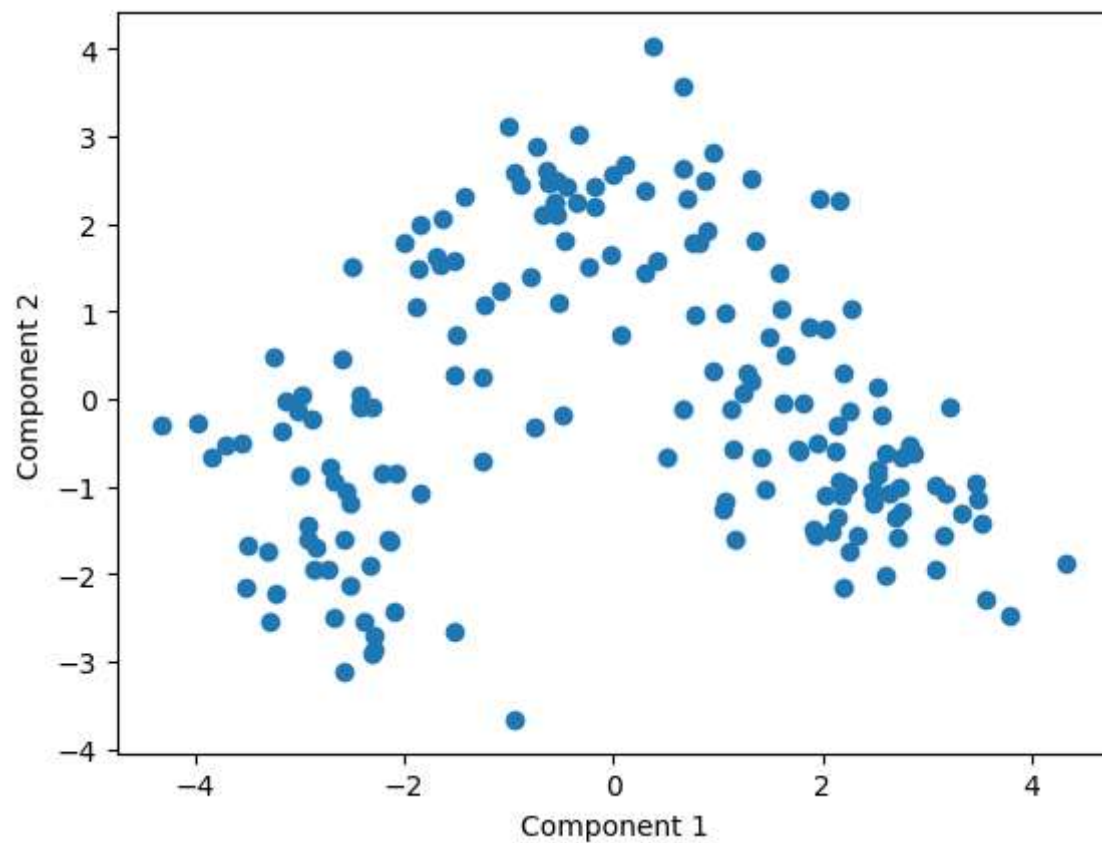
Doing PCA (dimension reduction)

```
In [ ]: # PCA
scaler = StandardScaler()
X = scaler.fit_transform(df)

pca = PCA(n_components=2)

X_pca = pca.fit_transform(X)

# Plot the first two principal components
plt.scatter(X_pca[:, 0], X_pca[:, 1], )
plt.xlabel('Component 1')
plt.ylabel('Component 2')
plt.show()
```

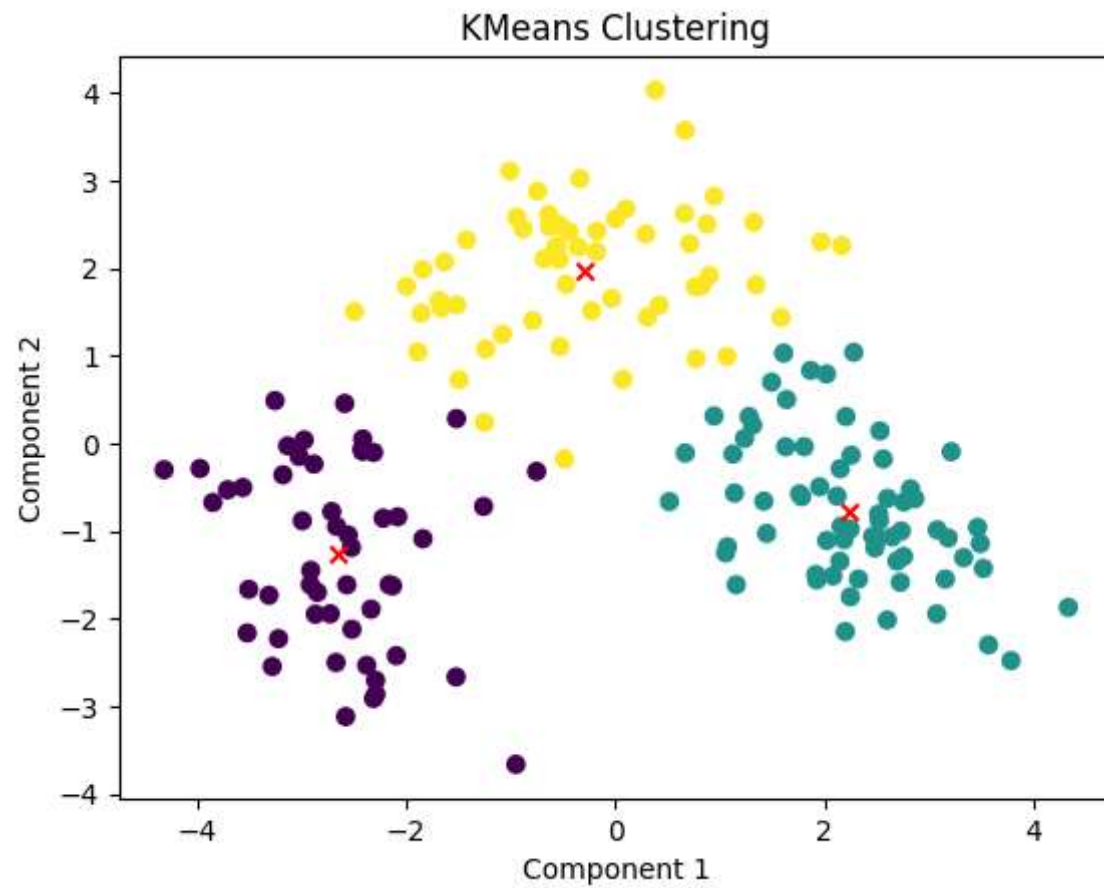



Apply KMeans clustering again

```
In [ ]: # Apply KMeans Clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_pca)

plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans.labels_, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], marker='x', color='red')
plt.xlabel('Component 1')
plt.ylabel('Component 2')
plt.title('KMeans Clustering')
plt.show()
```

c:\Users\Patrick\Documents\GitHub\DataMiningNTNU\env\lib\site-packages\sklearn\cluster_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
super()._check_params_vs_input(X, default_n_init=10)



Use Silhouette Score to see how well the clustering does

```
In [ ]: from sklearn.metrics import silhouette_score, normalized_mutual_info_score

# Silhouette Score
sil_score = silhouette_score(X_pca, kmeans.labels_)
print(f'Silhouette Score: {sil_score}')
```

Silhouette Score: 0.5723812080548977