

INTRODUCTION TO MACHINE LEARNING

MINI PROJECT.5

Lecturer : Mr. TOCH Sopheak



Group 05

Start Slide

01

OUR TEAMS(GROUP 5)

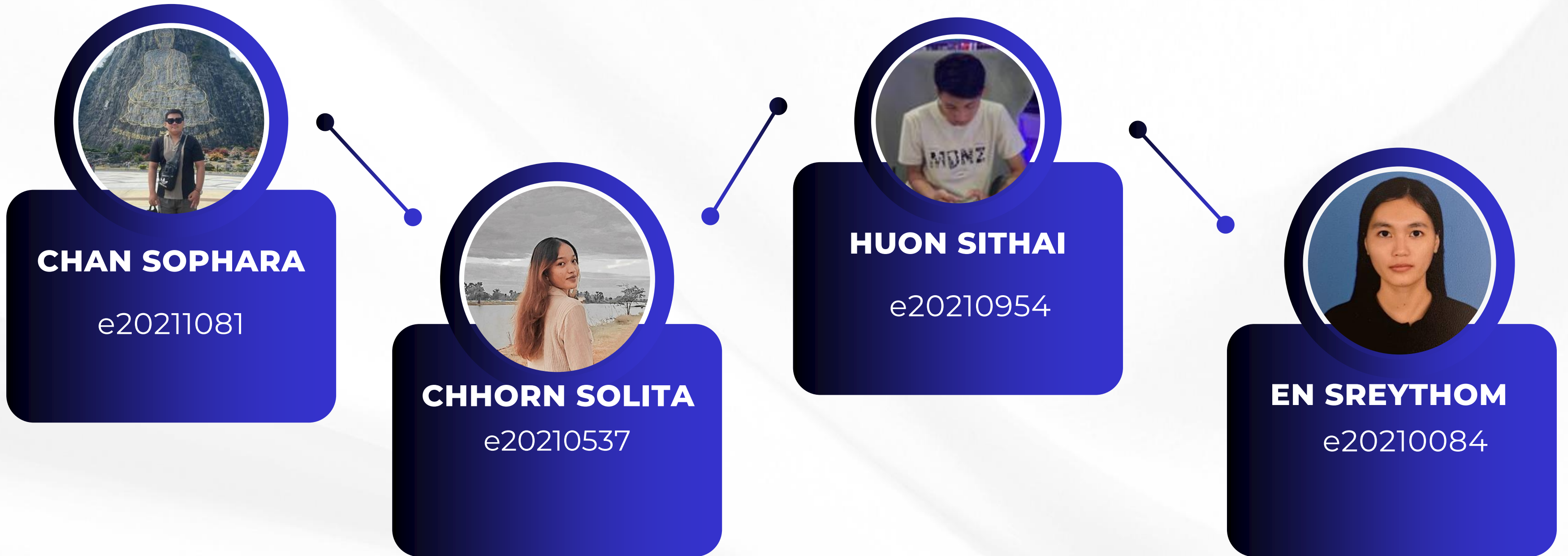


TABLE OF CONTENTS

01

About
projects

02

Datasets

03

Elbow Method &
K-Means algorithm

04

Model
visualisation

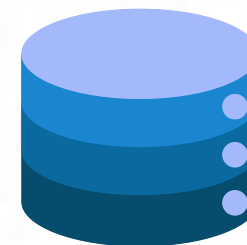
03

ABOUT **OUR PROJECT.5**

For our project today we are working on wine clustering dataset. Wine clustering is a fascinating application of machine learning, used to group different wines based on their characteristics. We apply K-means algorithm and the Elbow method helps classify wines by their properties and visualize the result.



DATASETS



For our dataset we get from kaggle about wholesales dataset.

wine-clustering.csv

05

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proline
2	14.23	1.71	2.43	15.6	127	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1065
3	13.2	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1050
4	13.16	2.36	2.67	18.6	101	2.8	3.24	0.3	2.81	5.68	1.03	3.17	1185
5	14.37	1.95	2.5	16.8	113	3.85	3.49	0.24	2.18	7.8	0.86	3.45	1480
6	13.24	2.59	2.87	21	118	2.8	2.69	0.39	1.82	4.32	1.04	2.93	735
7	14.2	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1450
8	14.39	1.87	2.45	14.6	96	2.5	2.52	0.3	1.98	5.25	1.02	3.58	1290
9	14.06	2.15	2.61	17.6	121	2.6	2.51	0.31	1.25	5.05	1.06	3.58	1295
10	14.83	1.64	2.17	14	97	2.8	2.98	0.29	1.98	5.2	1.08	2.85	1045
11	13.86	1.35	2.27	16	98	2.98	3.15	0.22	1.85	7.22	1.01	3.55	1045
12	14.1	2.16	2.3	18	105	2.95	3.32	0.22	2.38	5.75	1.25	3.17	1510
13	14.12	1.48	2.32	16.8	95	2.2	2.43	0.26	1.57	5	1.17	2.82	1280
14	13.75	1.73	2.41	16	89	2.6	2.76	0.29	1.81	5.6	1.15	2.9	1320
15	14.75	1.73	2.39	11.4	91	3.1	3.69	0.43	2.81	5.4	1.25	2.73	1150
16	14.38	1.87	2.38	12	102	3.3	3.64	0.29	2.96	7.5	1.2	3	1547
17	13.63	1.81	2.7	17.2	112	2.85	2.91	0.3	1.46	7.3	1.28	2.88	1310
18	14.3	1.92	2.72	20	120	2.8	3.14	0.33	1.97	6.2	1.07	2.65	1280
19	13.83	1.57	2.62	20	115	2.95	3.4	0.4	1.72	6.6	1.13	2.57	1130
20	14.19	1.59	2.48	16.5	108	3.3	3.93	0.32	1.86	8.7	1.23	2.82	1680


178 rows x 13 columns

IMPORT LIBRARY

```
# Importing important libraries that help our analysis
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
```

06

```
df = pd.read_csv('wine-clustering.csv')
df.head()
```



	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue	OD280	Proline
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

`df.shape`



`(178, 13)`

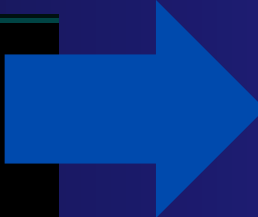
```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 178 entries, 0 to 177  
Data columns (total 13 columns):  
#   Column                                Non-Null Count  Dtype  
---  -  
0   Alcohol                             178 non-null    float64  
1   Malic_Acid                           178 non-null    float64  
2   Ash                                  178 non-null    float64  
3   Ash_Alcanity                         178 non-null    float64  
4   Magnesium                           178 non-null    int64  
5   Total_Phenols                        178 non-null    float64  
6   Flavanoids                           178 non-null    float64  
7   Nonflavanoid_Phenols                 178 non-null    float64  
8   Proanthocyanins                      178 non-null    float64  
9   Color_Intensity                      178 non-null    float64  
10  Hue                                  178 non-null    float64  
11  OD280                               178 non-null    float64  
12  Proline                             178 non-null    int64  
dtypes: float64(11), int64(2)  
memory usage: 18.2 KB
```

`df.info()`



07

df.isnull().sum()



Alcohol	0
Malic_Acid	0
Ash	0
Ash_Alcanity	0
Magnesium	0
Total_Phenols	0
Flavanoids	0
Nonflavanoid_Phenols	0
Proanthocyanins	0
Color_Intensity	0
Hue	0
OD280	0
Proline	0
dtype:	int64

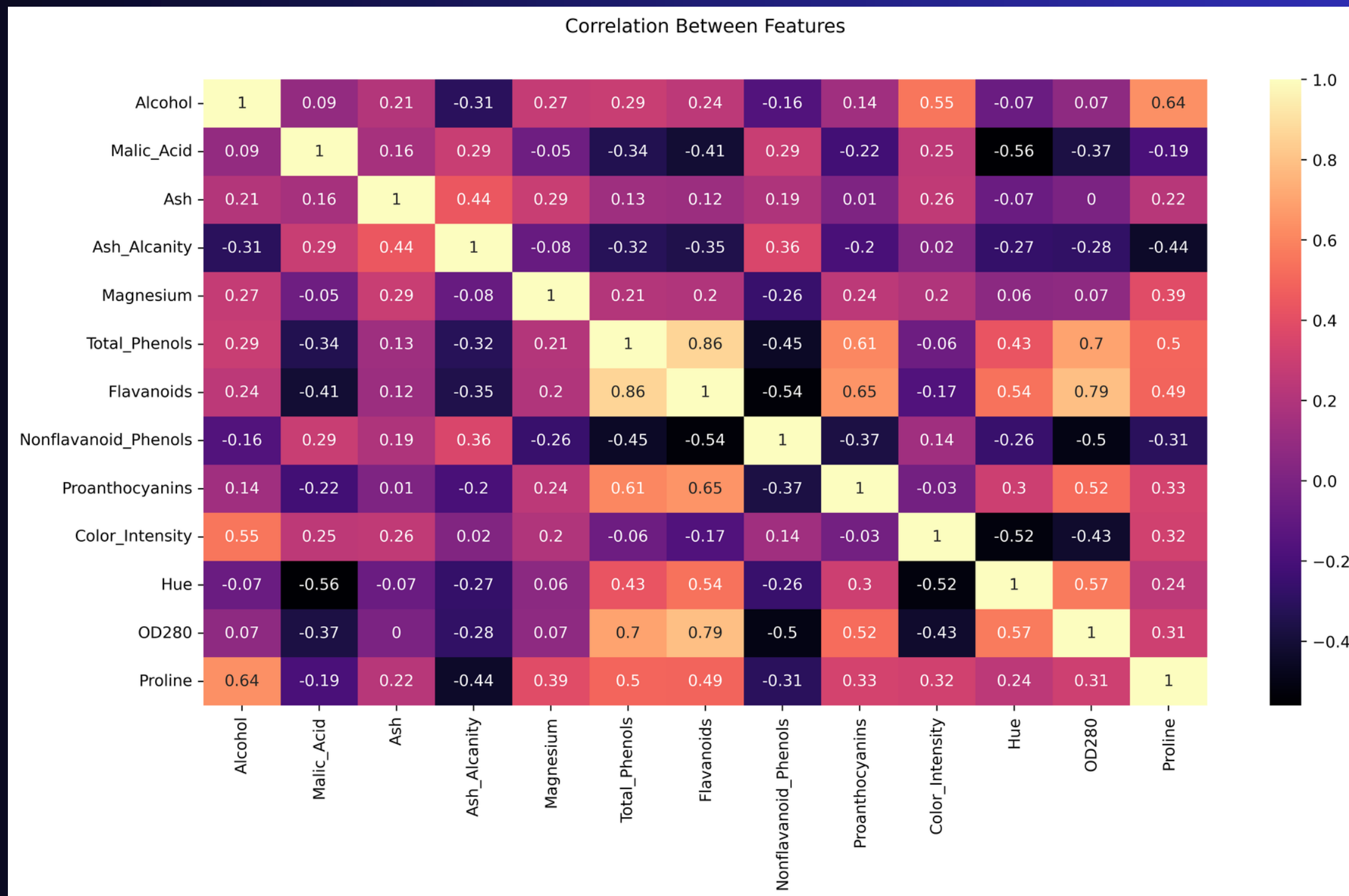
df.describe()

	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	5.058090
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.318286
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.280000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.220000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.690000
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


```
# Correlation between features
plt.figure(figsize=(13,8), dpi=500)
sns.heatmap(round(df.corr(),2), annot=True, cmap='magma')
plt.title("Correlation Between Features", pad=30)
plt.tight_layout()
plt.show()
```



09

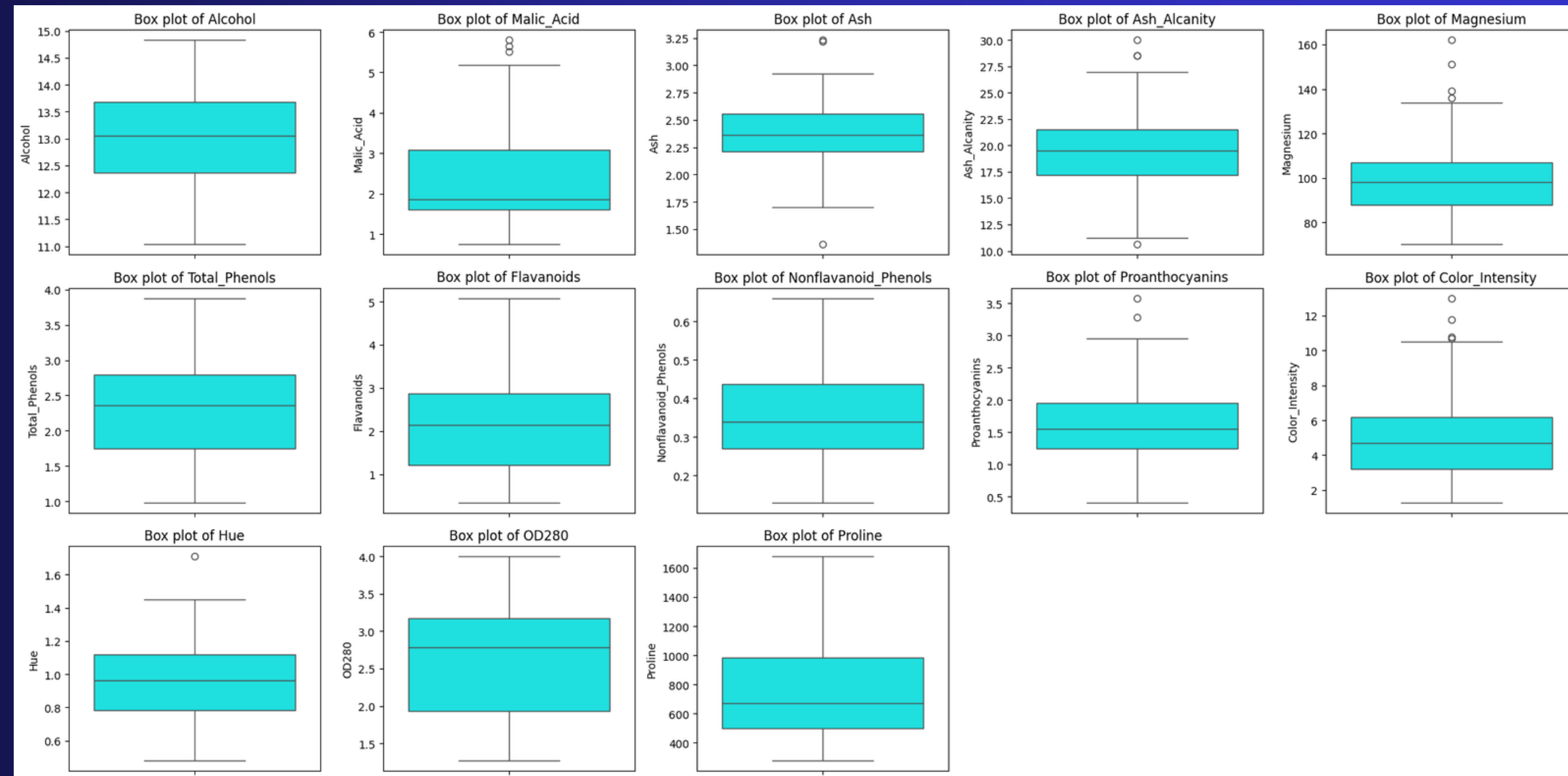


```
# Set up the matplotlib figure
plt.figure(figsize=(20, 10))

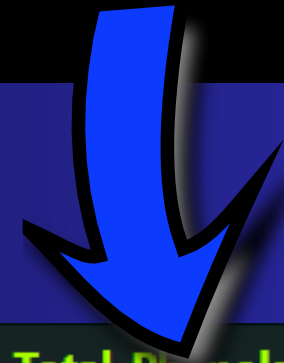
# Plotting boxplots for each numerical column
boxplot_color = 'cyan'
for i, col in enumerate(df.select_dtypes(include=['float64', 'int64']).columns):
    plt.subplot(3, 5, i+1) # Adjust the grid dimensions (3x4) based on your number of columns
    sns.boxplot(y=df[col] , color=boxplot_color)
    plt.title(f'Box plot of {col}')

plt.tight_layout()
plt.show()
```

10



```
# Scaling the dataset
df = pd.DataFrame(StandardScaler().fit_transform(df), columns = df.columns)
df.head()
```



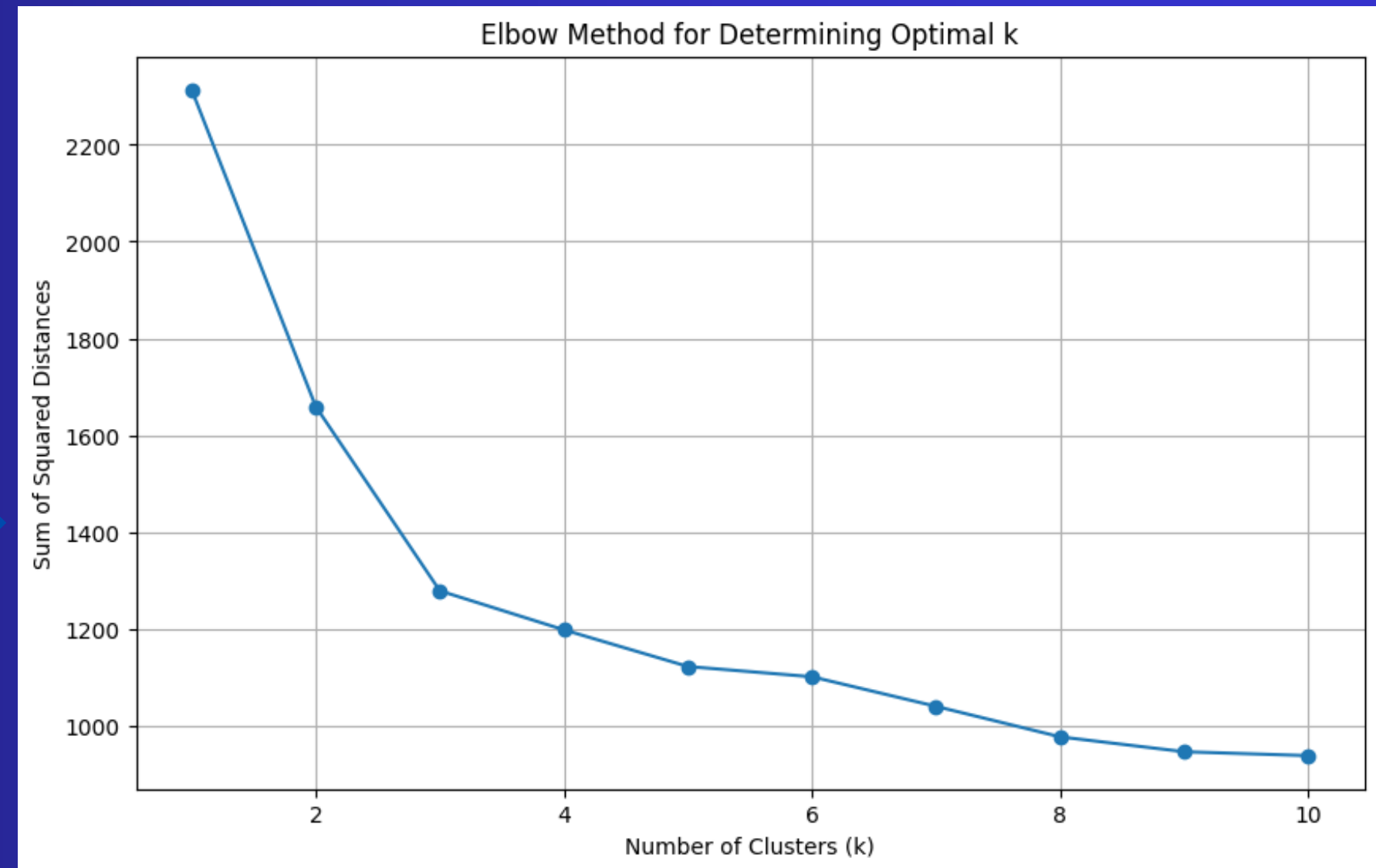
	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity	Hue
0	1.518613	-0.562250	0.232053	-1.169593	1.913905	0.808997	1.034819	-0.659563	1.224884	0.251717	0.362177
1	0.246290	-0.499413	-0.827996	-2.490847	0.018145	0.568648	0.733629	-0.820719	-0.544721	-0.293321	0.406051
2	0.196879	0.021231	1.109334	-0.268738	0.088358	0.808997	1.215533	-0.498407	2.135968	0.269020	0.318304
3	1.691550	-0.346811	0.487926	-0.809251	0.930918	2.491446	1.466525	-0.981875	1.032155	1.186068	-0.427544
4	0.295700	0.227694	1.840403	0.451946	1.281985	0.808997	0.663351	0.226796	0.401404	-0.319276	0.362177

ELBOW METHOD

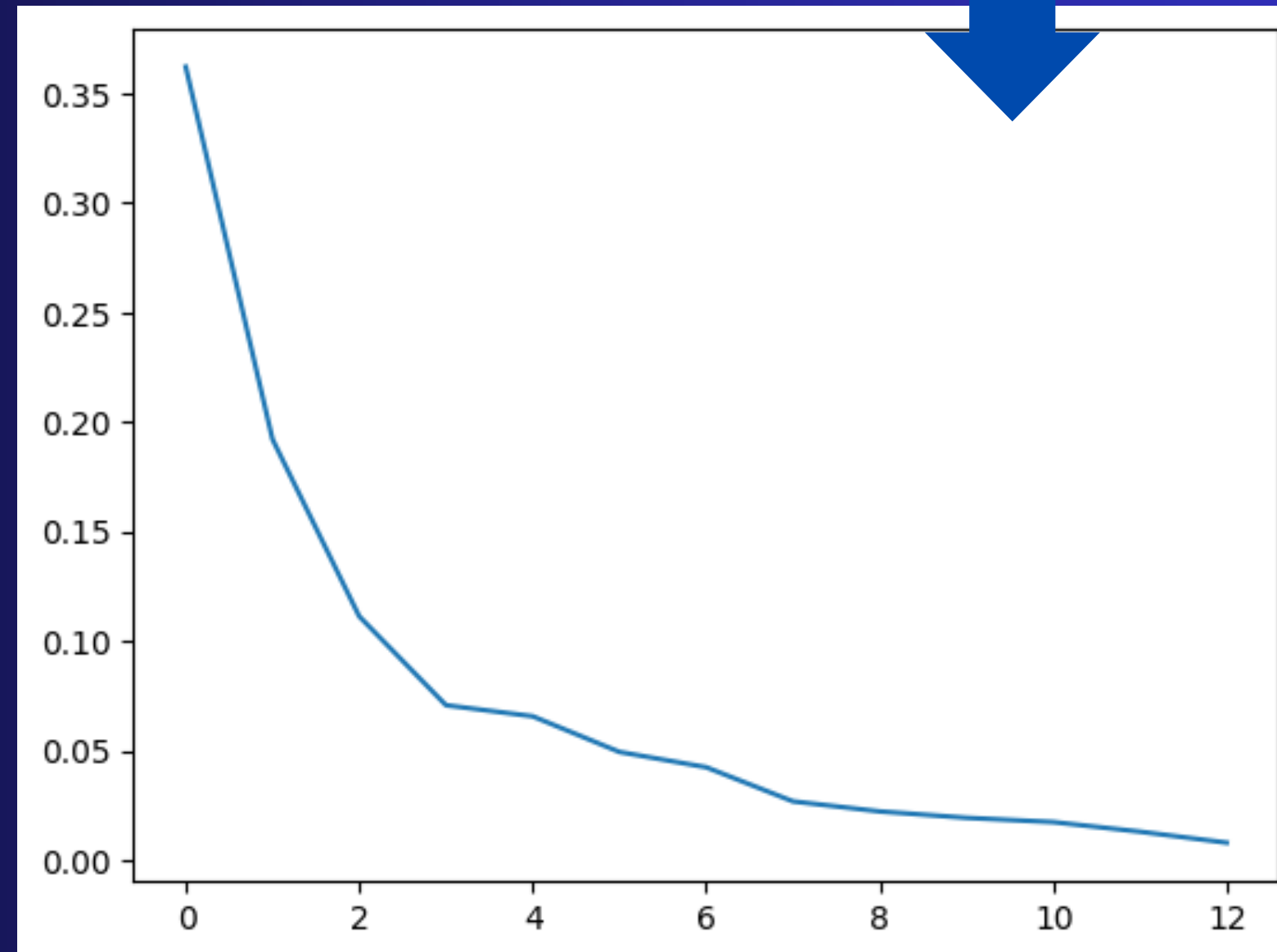
```
# Apply the Elbow Method to find the optimal number of clusters
sse = []
k_range = range(1, 11)

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=0)
    kmeans.fit(df)
    sse.append(kmeans.inertia_)
```

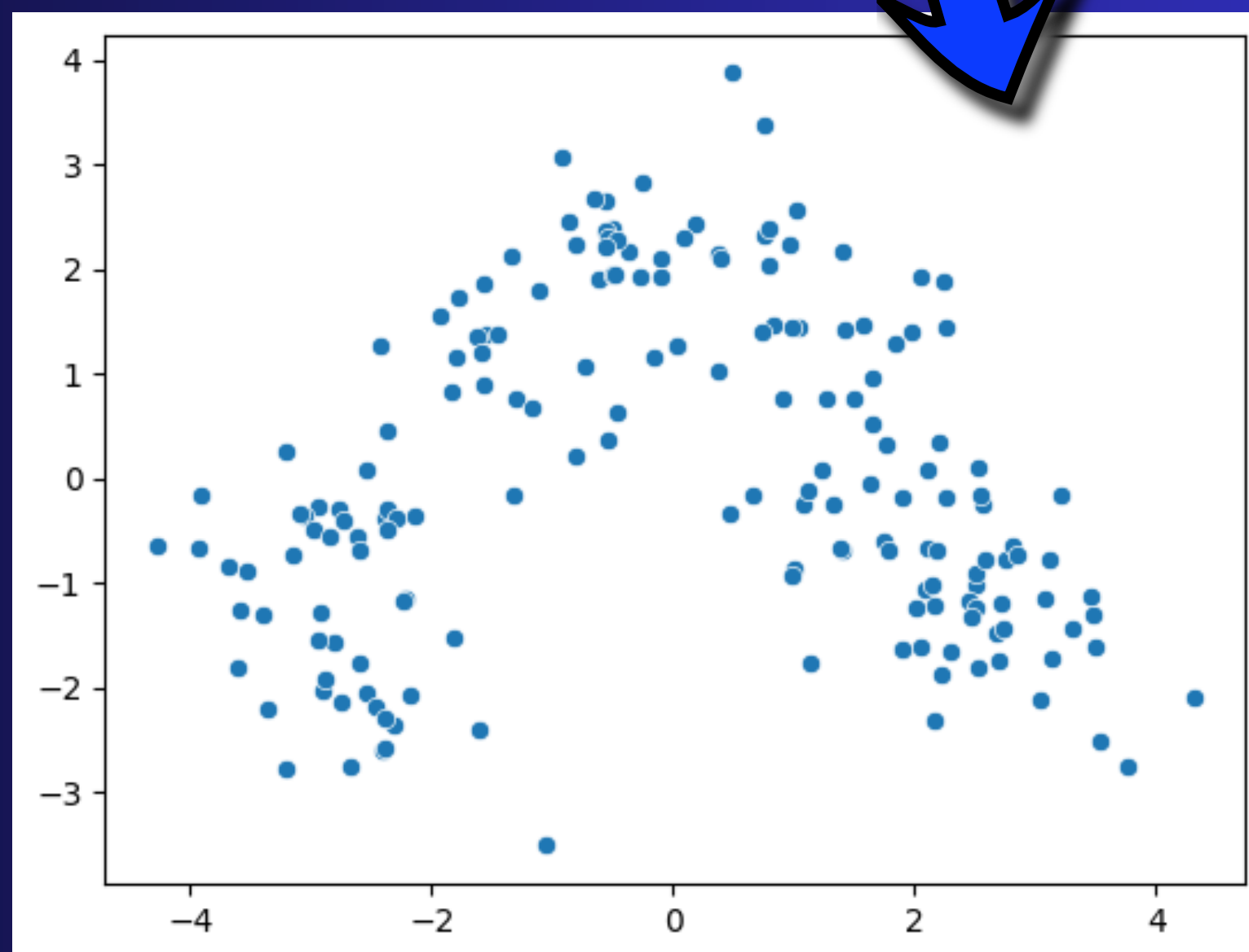
```
# Plot the Elbow Method results
plt.figure(figsize=(10, 6))
plt.plot(k_range, sse, marker='o')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Sum of Squared Distances')
plt.title('Elbow Method for Determining Optimal k')
plt.grid(True)
plt.show()
```



```
# Using PCA to find contributing variables to the variance  
# As we can see, only 2 variables contribute the most, that is "Alcohol" and "Malic_Acid"  
pca = PCA()  
df_tf = pca.fit_transform(df)  
plt.plot(pca.explained_variance_ratio_)
```



```
# Plotting our transformed dataset gives us 3 clusters as seen below
sns.scatterplot(
    x = df_tf[:, 0],
    y = df_tf[:, 1])
```




```
# Converting the data type into Data Frame for further analysis
df_new = pd.DataFrame(df_tf)
df_new.head()
```



	0	1	2	3	4	5	6	7	8	9	10	11	12
0	3.316751	-1.443463	-0.165739	-0.215631	0.693043	-0.223880	0.596427	0.065139	0.641443	1.020956	-0.451563	0.540810	-0.066239
1	2.209465	0.333393	-2.026457	-0.291358	-0.257655	-0.927120	0.053776	1.024416	-0.308847	0.159701	-0.142657	0.388238	0.003637
2	2.516740	-1.031151	0.982819	0.724902	-0.251033	0.549276	0.424205	-0.344216	-1.177834	0.113361	-0.286673	0.000584	0.021717
3	3.757066	-2.756372	-0.176192	0.567983	-0.311842	0.114431	-0.383337	0.643593	0.052544	0.239413	0.759584	-0.242020	-0.369484
4	1.008908	-0.869831	2.026688	-0.409766	0.298458	-0.406520	0.444074	0.416700	0.326819	-0.078366	-0.525945	-0.216664	-0.079364

```
# Getting the first and second columns corresponding to the contributing variables`  
df_new = pd.DataFrame(df_new.iloc[:,0:2])  
df_new.head()
```



	0	1
0	3.316751	-1.443463
1	2.209465	0.333393
2	2.516740	-1.031151
3	3.757066	-2.756372
4	1.008908	-0.869831

```
# Renaming the columns for clarity  
df_new.columns = ["Alcohol", "Malic_Acid"]  
df_new.head()
```

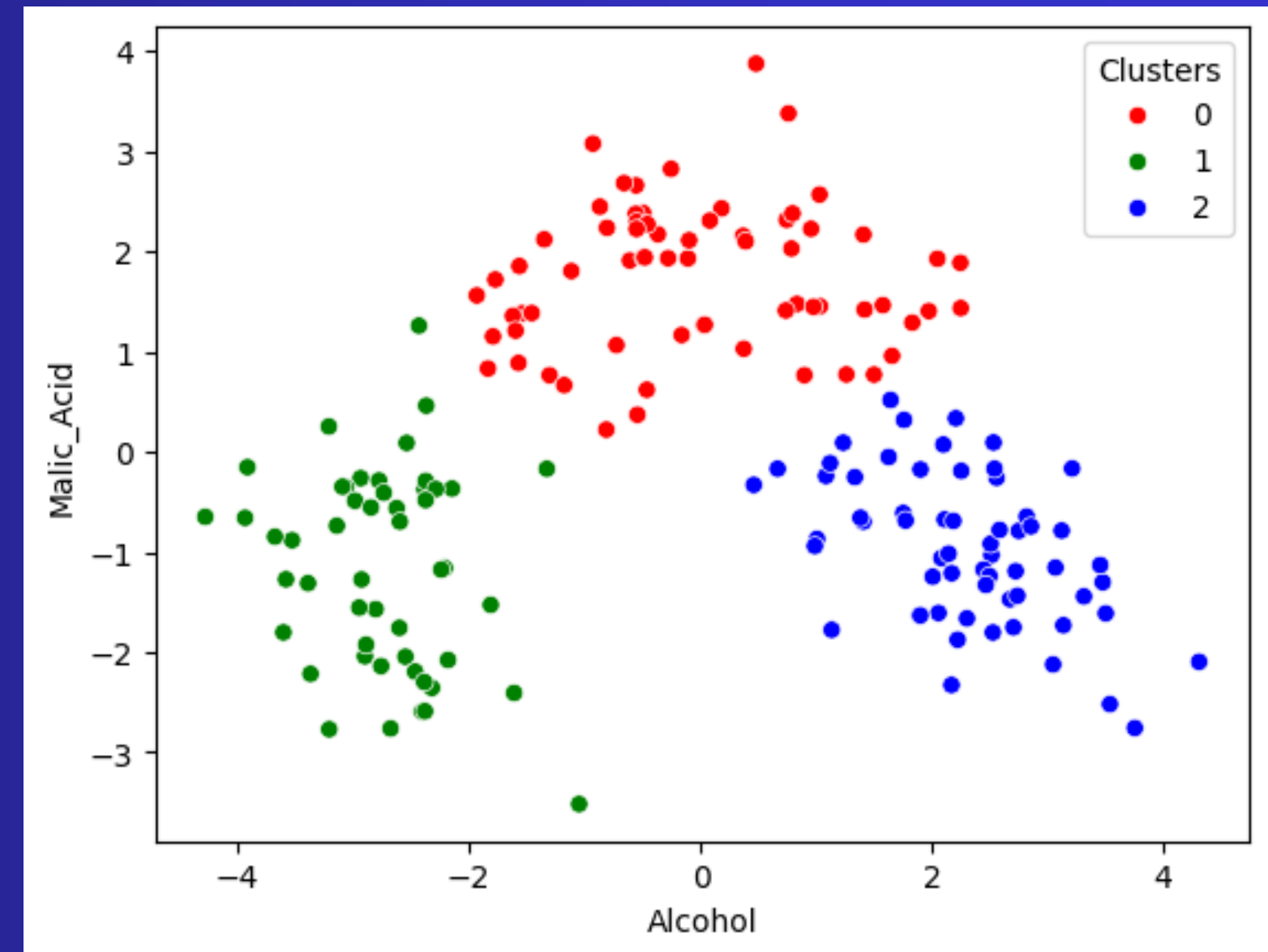


	Alcohol	Malic_Acid
0	3.316751	-1.443463
1	2.209465	0.333393
2	2.516740	-1.031151
3	3.757066	-2.756372
4	1.008908	-0.869831

K-MEAN CLUSTERING

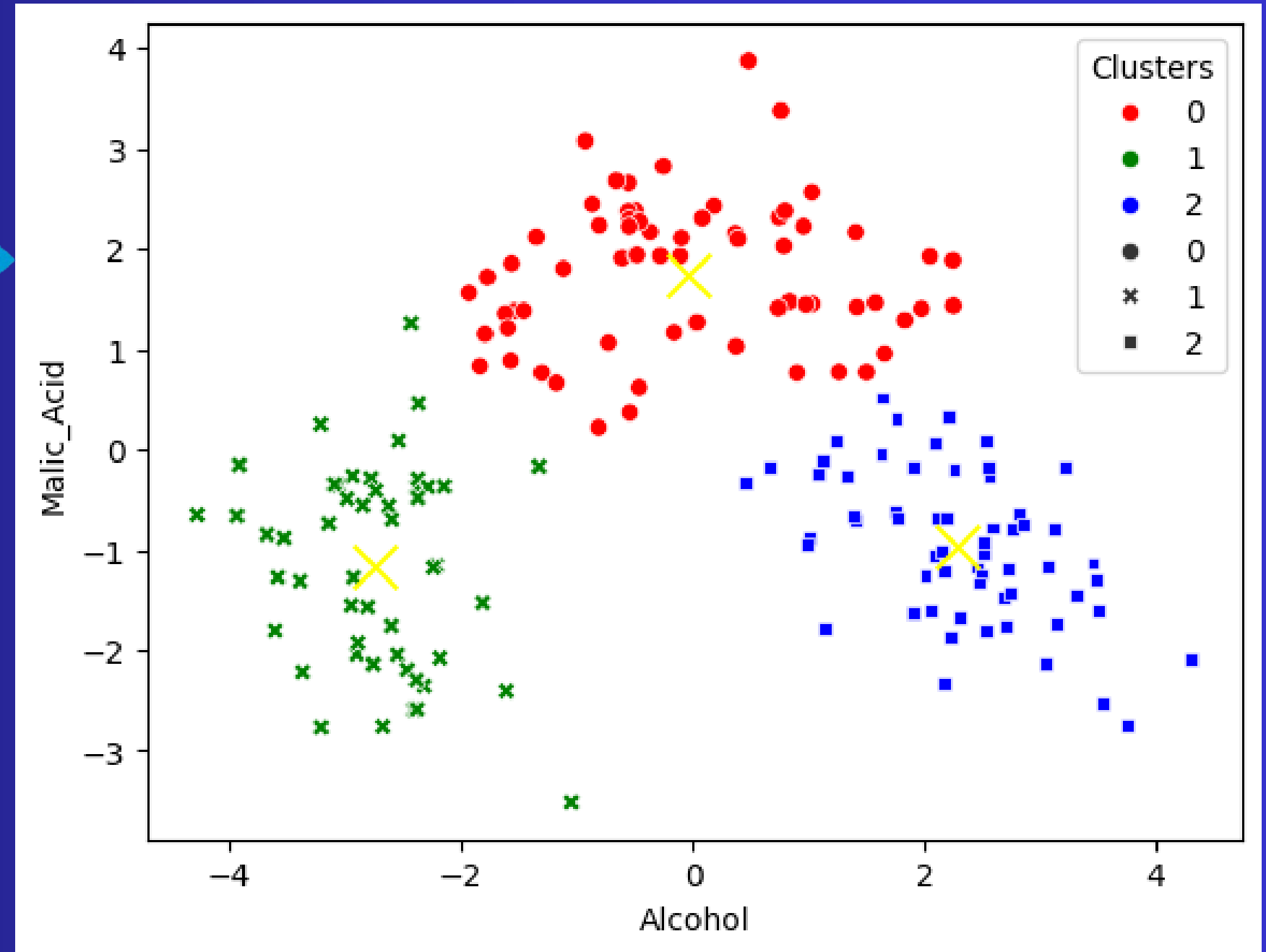
```
# Using KMeans for clustering
km = KMeans(
    n_clusters = 3,
    init = "k-means++",
    n_init = 10)
km.fit(df_new)
df_new["Clusters"] = km.labels_
```

```
# Visualizing the data
sns.scatterplot(
    x = df_new["Alcohol"],
    y = df_new["Malic_Acid"],
    hue = df_new['Clusters'],
    palette = ["red" , "green" , "blue"])
```




```
# There are 3 Clusters seen separated clearly
sns.scatterplot(
    x = df_new["Alcohol"],
    y = df_new["Malic_Acid"],
    hue = df_new["Clusters"],
    style = km.labels_,
    palette = ["red", "green", "blue"])

plt.scatter(
    km.cluster_centers_[0,0],
    km.cluster_centers_[0,1],
    marker = "x",
    s = 200,
    c = "yellow")
```

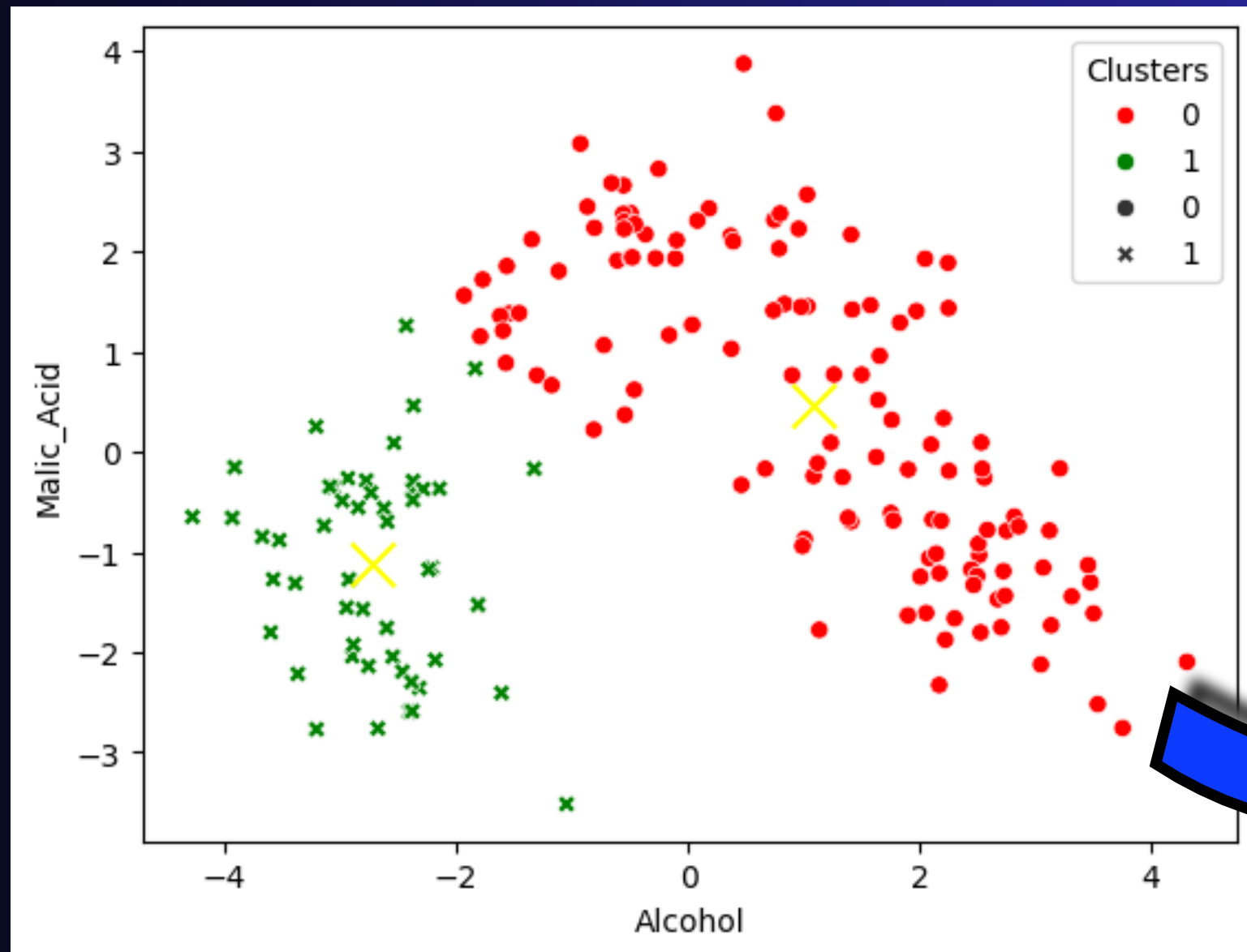


```
columns_to_cluster = ['Alcohol','Malic_Acid', 'Ash', 'Ash_Alcanity', 'Magnesium', 'Total_Phenols', 'Flavanoids', 'Nonflavanoid_Phenols',  
# Extracting the selected columns for clustering  
X = df[columns_to_cluster]  
# Instantiate KMeans  
kmeans = KMeans(n_clusters=3, random_state=42)  
# Fit KMeans and predict cluster labels  
df['Cluster'] = kmeans.fit_predict(X)  
# Print the counts of each cluster  
print(df['Cluster'].value_counts())
```



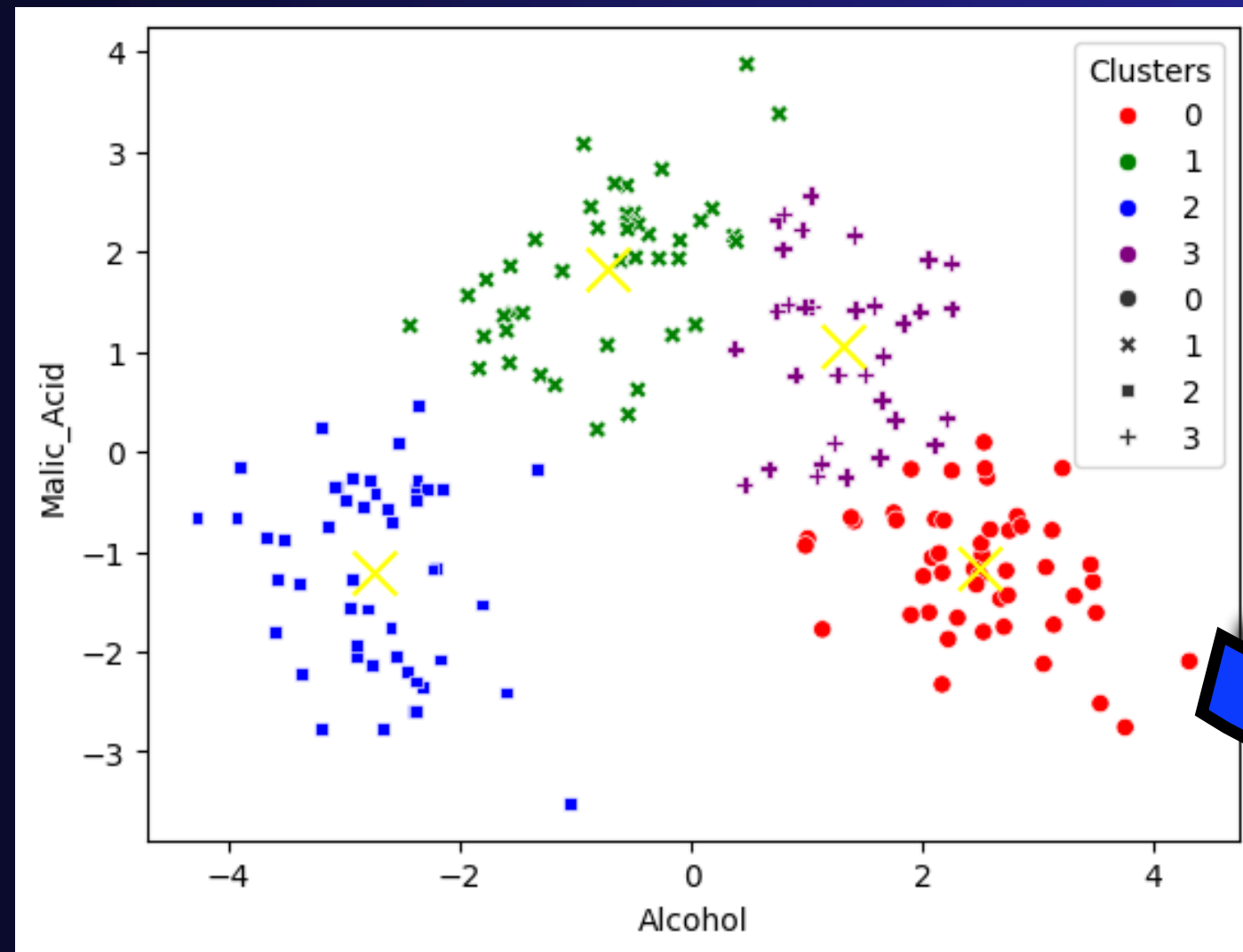
```
Cluster  
0      65  
2      62  
1      51  
Name: count, dtype: int64
```

LET TRY WITH K = 2



```
Cluster
0      107
1       71
Name: count, dtype: int64
```


LET TRY WITH K = 4



```
Cluster
2      58
3      58
1      51
0      11
Name: count, dtype: int64
```

THANK YOU !!!