In [101…

```
import os
import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.metrics import classification_report, confusion_matrix
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PowerTransformer
from sklearn.cluster import KMeans
fixed_random_state = random.seed(42)
import umap
from sklearn.datasets import load_digits
%matplotlib inline
```

In [100…

```
pip install umap-learn
```

```
Collecting umap-learn
  Downloading umap-learn-0.5.3.tar.gz (88 kB)
Requirement already satisfied: numpy>=1.17 in c:\users\tural\anaconda3\lib\site-packa
ges (from umap-learn) (1.21.5)
Requirement already satisfied: scikit-learn>=0.22 in c:\users\tural\anaconda3\lib\sit
e-packages (from umap-learn) (1.0.2)
Requirement already satisfied: scipy>=1.0 in c:\users\tural\anaconda3\lib\site-packag
es (from umap-learn) (1.7.3)
Requirement already satisfied: numba>=0.49 in c:\users\tural\anaconda3\lib\site-packa
ges (from umap-learn) (0.55.1)
Collecting pynndescent>=0.5
  Downloading pynndescent-0.5.7.tar.gz (1.1 MB)
Requirement already satisfied: tqdm in c:\users\tural\anaconda3\lib\site-packages (fr
om umap-learn) (4.64.0)
Requirement already satisfied: setuptools in c:\users\tural\anaconda3\lib\site-packag
es (from numba>=0.49->umap-learn) (61.2.0)
Requirement already satisfied: llvmlite<0.39,>=0.38.0rc1 in c:\users\tural\anaconda3
\lib\site-packages (from numba>=0.49->umap-learn) (0.38.0)
Requirement already satisfied: joblib>=0.11 in c:\users\tural\anaconda3\lib\site-pack
ages (from pynndescent>=0.5->umap-learn) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\tural\anaconda3\lib\s
ite-packages (from scikit-learn>=0.22->umap-learn) (2.2.0)
Requirement already satisfied: colorama in c:\users\tural\anaconda3\lib\site-packages
(from tqdm->umap-learn) (0.4.4)
Building wheels for collected packages: umap-learn, pynndescent
  Building wheel for umap-learn (setup.py): started
  Building wheel for umap-learn (setup.py): finished with status 'done'
  Created wheel for umap-learn: filename=umap_learn-0.5.3-py3-none-any.whl size=82829
sha256=f1cb7042a75a252027dc41f08db5b896e400732ec06c7e9d78f04765ba50c869
  Stored in directory: c:\users\tural\appdata\local\pip\cache\wheels\f4\3e\1c\596d0a4
63d17475af648688443fa4846fef624d1390339e7e9
  Building wheel for pynndescent (setup.py): started
  Building wheel for pynndescent (setup.py): finished with status 'done'
  Created wheel for pynndescent: filename=pynndescent-0.5.7-py3-none-any.whl size=542
86 sha256=ecc517ff2d0531e781b63bc7e02c9b2ca9265e3e4dd085f783c27382848531dd
  Stored in directory: c:\users\tural\appdata\local\pip\cache\wheels\5b\f5\6e\aac11d6
9fe2115d9ac871d6c148b361f0d3f8a35ed7354fa03
Successfully built umap-learn pynndescent
Installing collected packages: pynndescent, umap-learn
Successfully installed pynndescent-0.5.7 umap-learn-0.5.3
Note: you may need to restart the kernel to use updated packages.
```

In [3]:

```python
os.getcwd()
```

Out[3]:

```
'C:\\Users\\tural'
```

In [37]:

```python
os.chdir('C:\\Users\\tural\\OneDrive\\Desktop\\Study Materials\\Datasets')
```

In [64]:

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

In [66]:

```python
df.head()
```

Out[66]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phe
0	14.23	1.71	2.43	15.6	127	2.80	3.06	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	

In [68]:

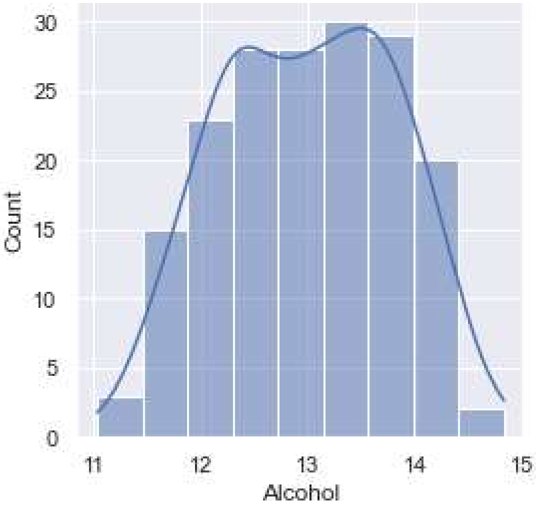
```
df.isna().sum()
```

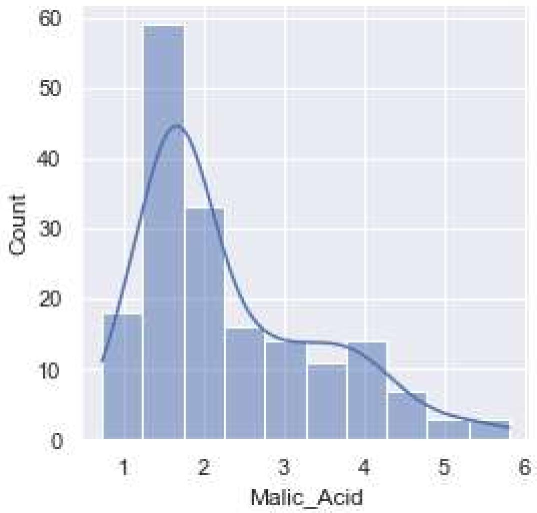
Out[68]:

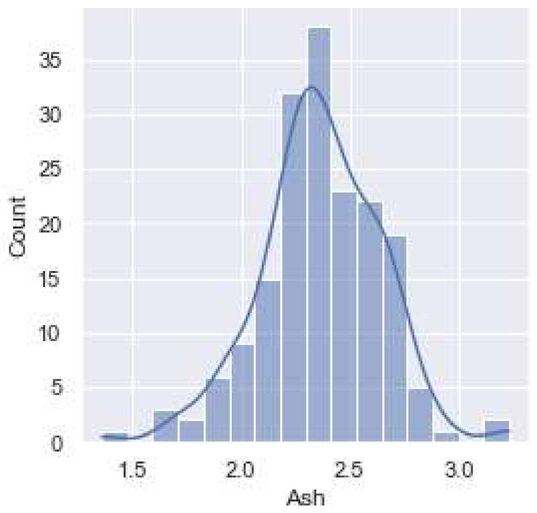
```
Alcohol                 0
Malic_Acid              0
Ash                     0
Ash_Alcanity            0
Magnesium               0
Total_Phenols           0
Flavanoids              0
Nonflavanoid_Phenols    0
Proanthocyanins         0
Color_Intensity         0
Hue                     0
OD280     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

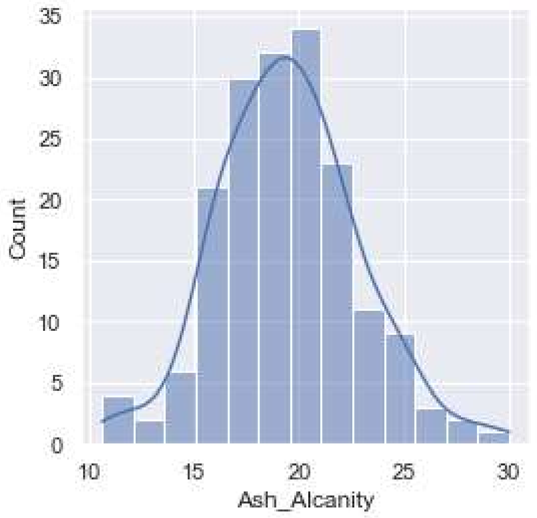
In [78]:

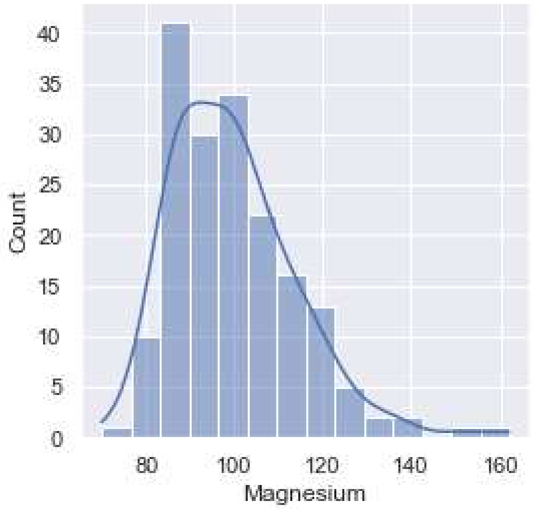
```
for col in df.columns:
    fig, axs = plt.subplots(figsize = (4,4))
    sns.histplot(data=df, x=col, kde=True, ax=axs)
    plt.show()
```

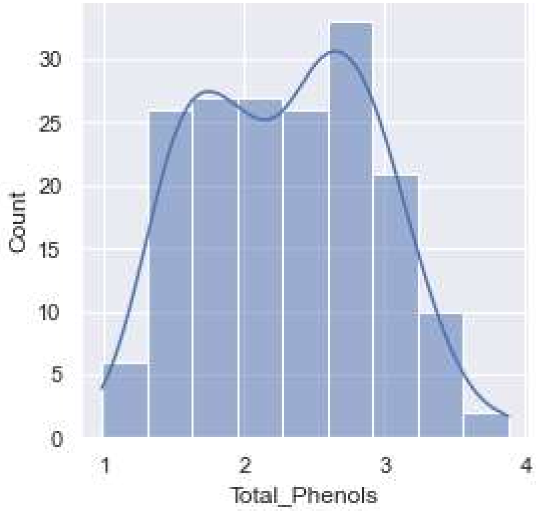


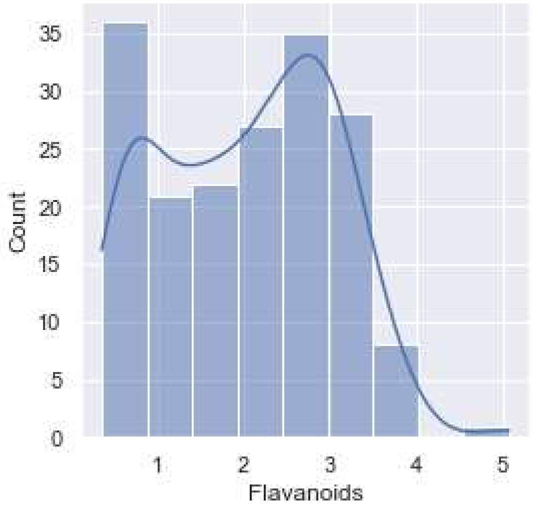

Count
0
10
20
30
40
50
60
1
2
3
4
5
6
Malic_Acid

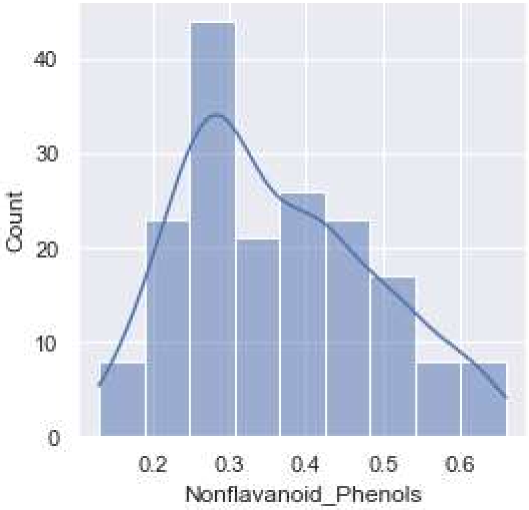

Count
0
5
10
15
20
25
30
35
1.5
2.0
2.5
3.0
Ash

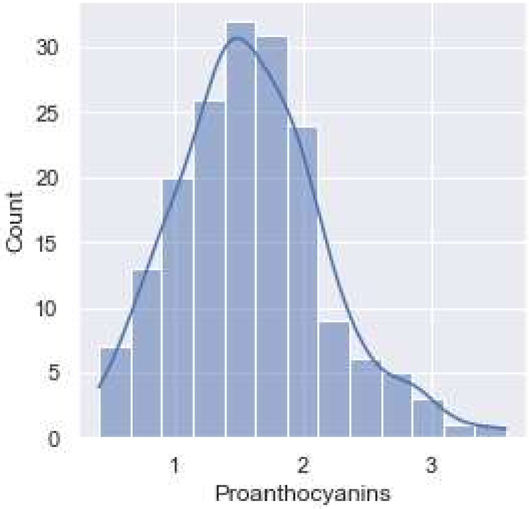

Count
0
5
10
15
20
25
30
35
10
15
20
25
30
Ash_Alcanity

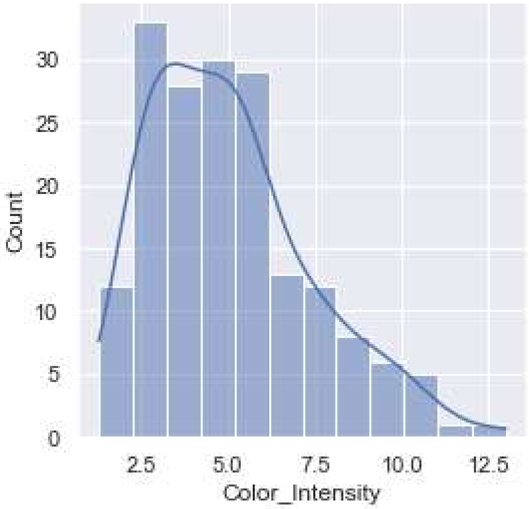

Count
0
5
10
15
20
25
30
35
40
80
100
120
140
160
Magnesium

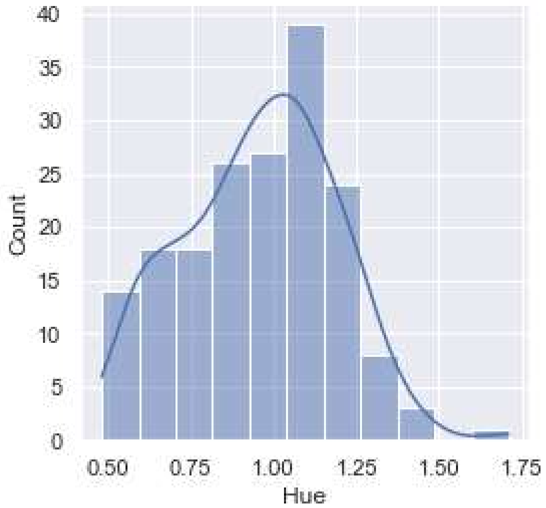

Count
0
5
10
15
20
25
30
1
2
3
4
Total_Phenols

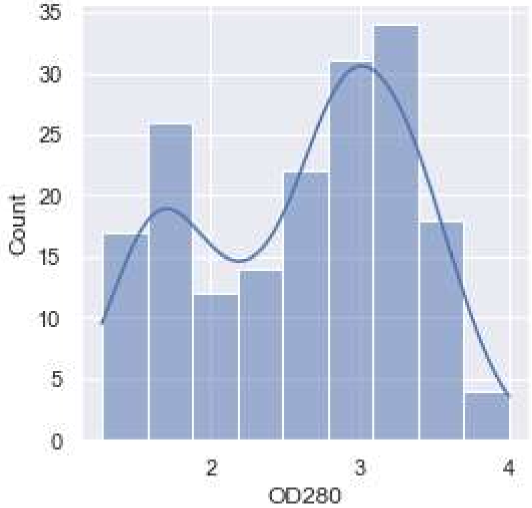

Count
0
5
10
15
20
25
30
35
1
2
3
4
5
Flavanoids

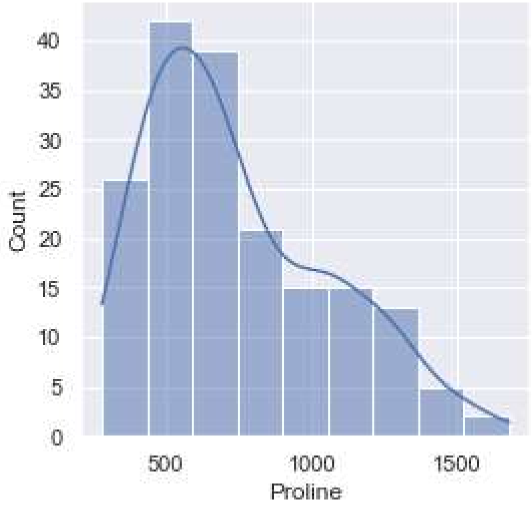

Count
0
10
20
30
40
0.2
0.3
0.4
0.5
0.6
Nonflavanoid_Phenols


Count
0
5
10
15
20
25
30
1
2
3
Proanthocyanins


Count
0
5
10
15
20
25
30
2.5
5.0
7.5
10.0
12.5
Color_Intensity







In [80]:

```python
X = df.copy()
X = StandardScaler().fit_transform(df)
X = PowerTransformer(standardize = False).fit_transform(X)
```

In [81]:

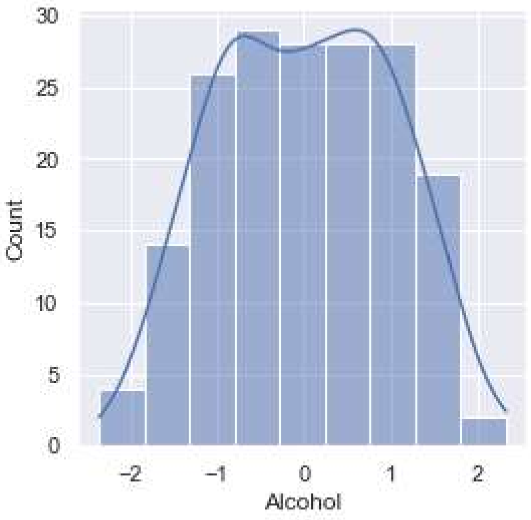
```python
X = pd.DataFrame(X, columns = df.columns)
X.head()
```

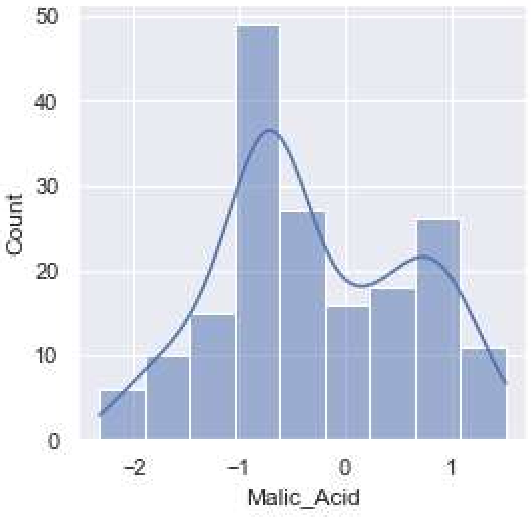
Out[81]:

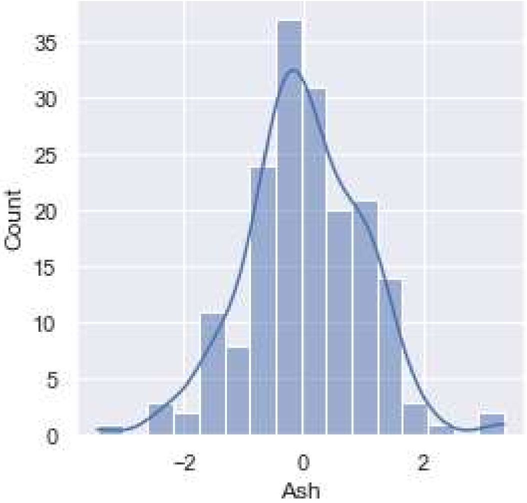
	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoi
0	1.552972	-0.702436	0.233769	-1.224710	1.378174	0.796708	1.043661	
1	0.247472	-0.610198	-0.809540	-2.696922	0.018056	0.562211	0.738364	
2	0.197645	0.021031	1.141607	-0.272253	0.086325	0.796708	1.227314	
3	1.732995	-0.400457	0.495037	-0.837478	0.767296	2.405004	1.482916	
4	0.297382	0.207225	1.919073	0.442690	1.001563	0.796708	0.667285	

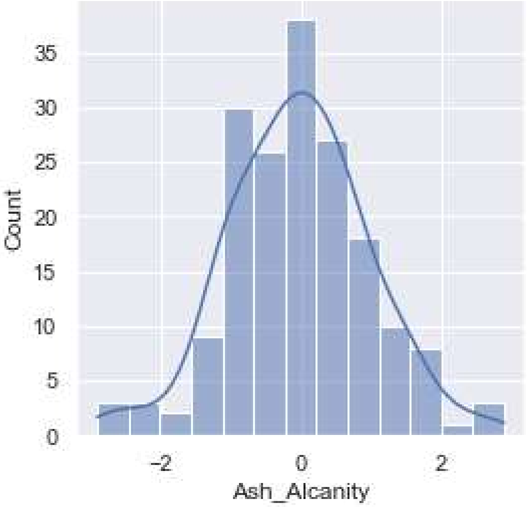
In [82]:

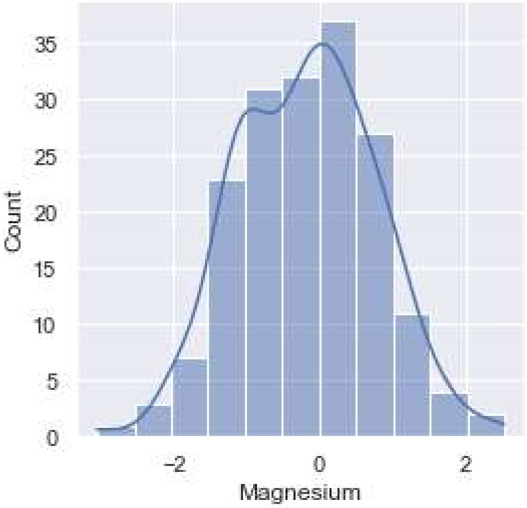
```python
for col in X.columns:
    fig, axs = plt.subplots(figsize = (4,4))
    sns.histplot(data=X, x=col, kde=True, ax=axs)
    plt.show()
```

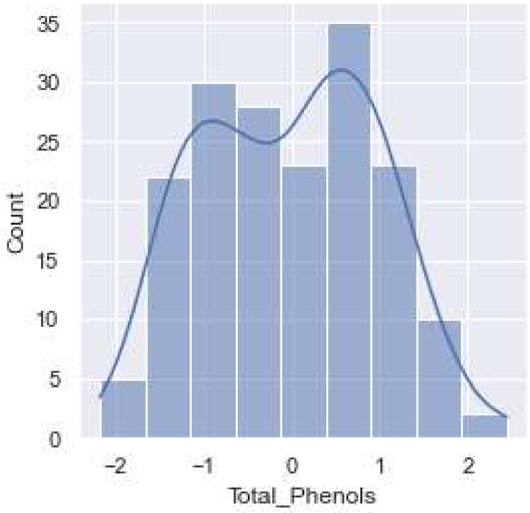


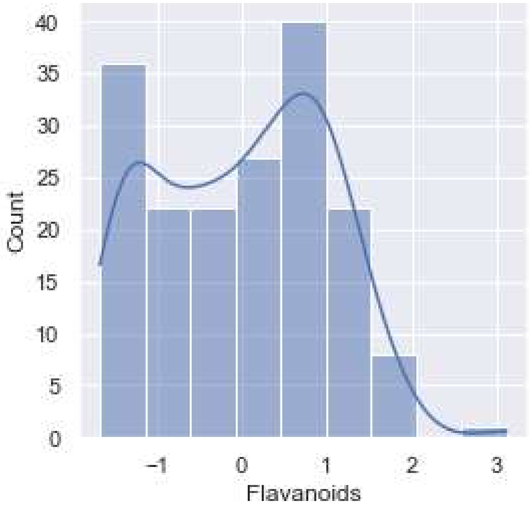


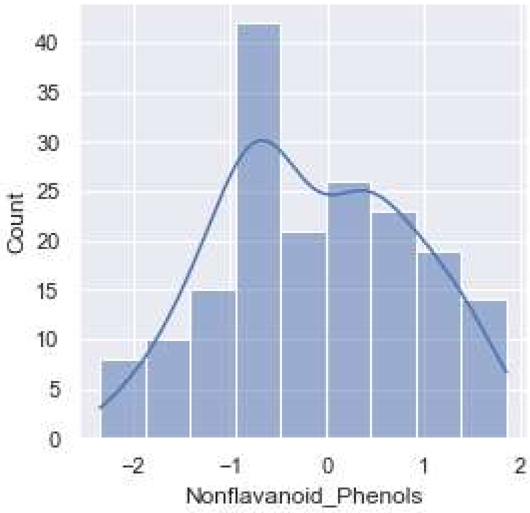

Count
0
5
10
15
20
25
30
35
−2
0
2
Ash

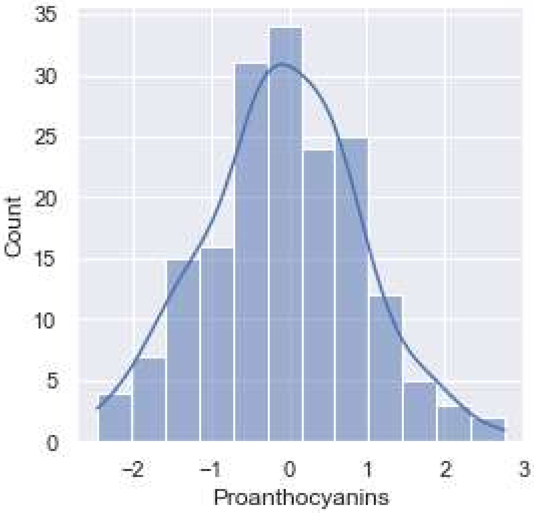

Count
0
5
10
15
20
25
30
35
−2
0
2
Ash_Alcanity

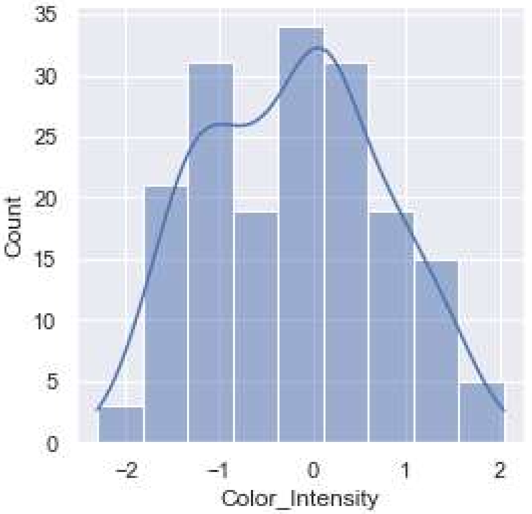

Count
0
5
10
15
20
25
30
35
−2
0
2
Magnesium

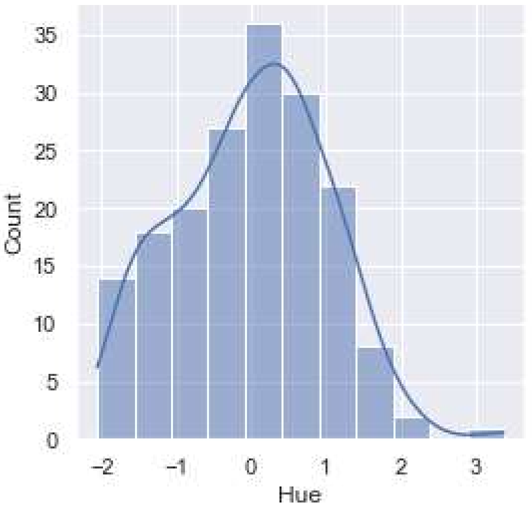

Count
0
5
10
15
20
25
30
35
−2
−1
0
1
2
Total_Phenols

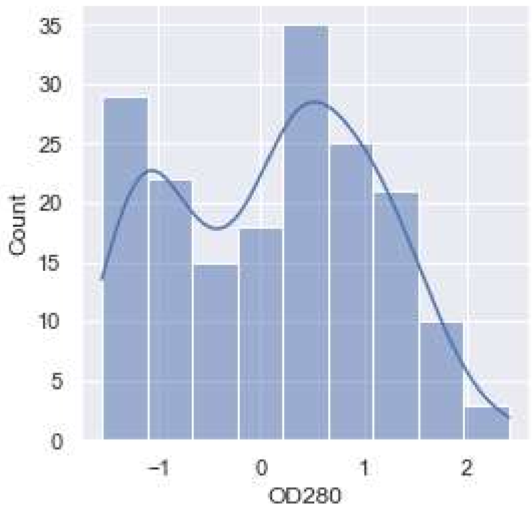

Count
0
5
10
15
20
25
30
35
40
−1
0
1
2
3
Flavanoids

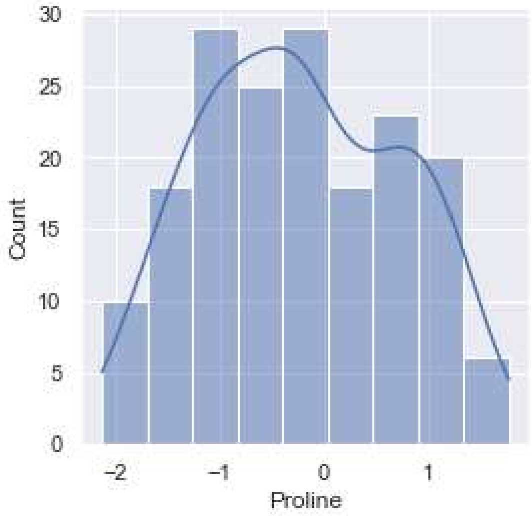

Count
0
5
10
15
20
25
30
35
40
−2
−1
0
1
2
Nonflavanoid_Phenols


35
30
25
20
15
10
5
0
Count
−2
−1
0
1
2
3
Proanthocyanins


35
30
25
20
15
10
5
0
Count
−2
−1
0
1
2
Color_Intensity


35
30
25
20
15
10
5
0
Count
−2
−1
0
1
2
3
Hue





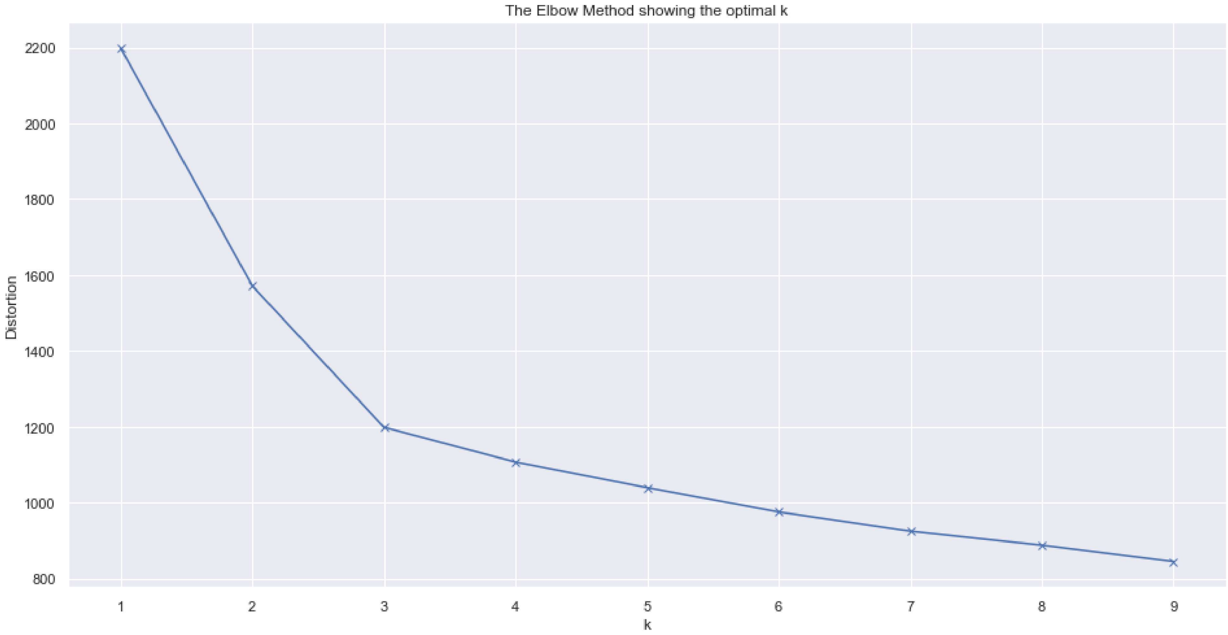
In [90]:

```
distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(X)
    distortions.append(kmeanModel.inertia_)
```

```
C:\Users\tural\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: UserWarni
ng: KMeans is known to have a memory leak on Windows with MKL, when there are less ch
unks than available threads. You can avoid it by setting the environment variable OMP
_NUM_THREADS=1.
  warnings.warn(
```

In [91]:

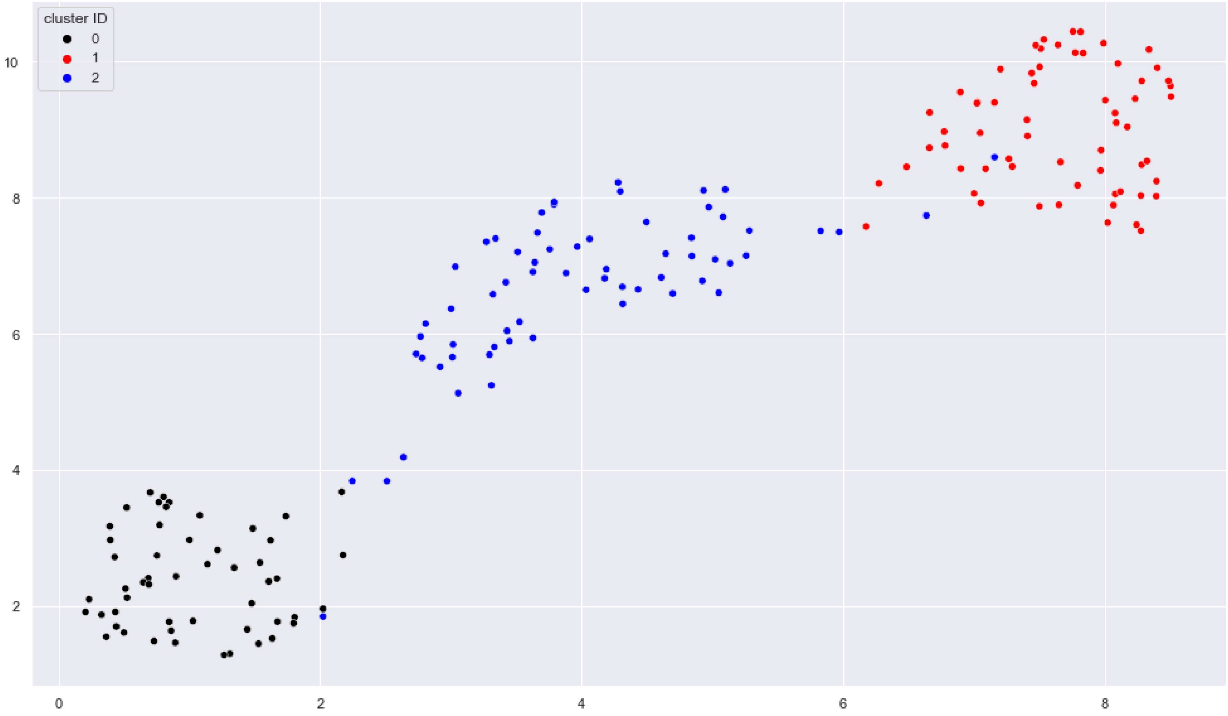
```
plt.figure(figsize=(16,8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



In [92]:
```
labels = KMeans(n_clusters=3, random_state=fixed_random_state).fit_predict(X)
```

In [104…
```
reducer = umap.UMAP(n_components=2, n_neighbors=15, n_jobs=-1, random_state=fixed_ranc
embedding = reducer.fit_transform(X)
```

In [109…
```
plt.figure(figsize = (17,10))
sns.scatterplot(x=embedding[:, 0], y=embedding[:, 1], hue=labels, palette=['black', 'r
plt.legend(title="cluster ID")
plt.show()
```


cluster ID
● 0
● 1
● 2