campusx-official / **100-days-of-machine-learning** Public

Code Issues 3 Pull requests Actions Projects Security Insights

main

100-days-of-machine-learning / day25-normalization / **day25.ipynb**

campusx-official Add files via upload **History**

1 contributor

```
649 lines (649 sloc) | 247 KB
```

In [29]:

```
import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
import seaborn as sns
```

In [32]:

```
df = pd.read_csv('wine_data.csv',header=None,usecols=[0,1,2])
df.columns=['Class label', 'Alcohol', 'Malic acid']
```

In [33]:

```
df
```

Out[33]:

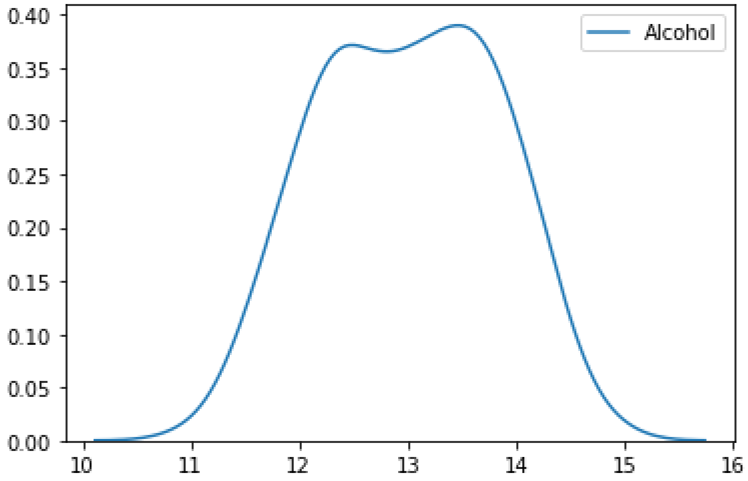
	Class label	Alcohol	Malic acid
0	1	14.23	1.71
1	1	13.20	1.78
2	1	13.16	2.36
3	1	14.37	1.95
4	1	13.24	2.59
...	...	...	...
173	3	13.71	5.65
174	3	13.40	3.91
175	3	13.27	4.28
176	3	13.17	2.59
177	3	14.13	4.10

178 rows × 3 columns

In [34]:

```
sns.kdeplot(df['Alcohol'])
```

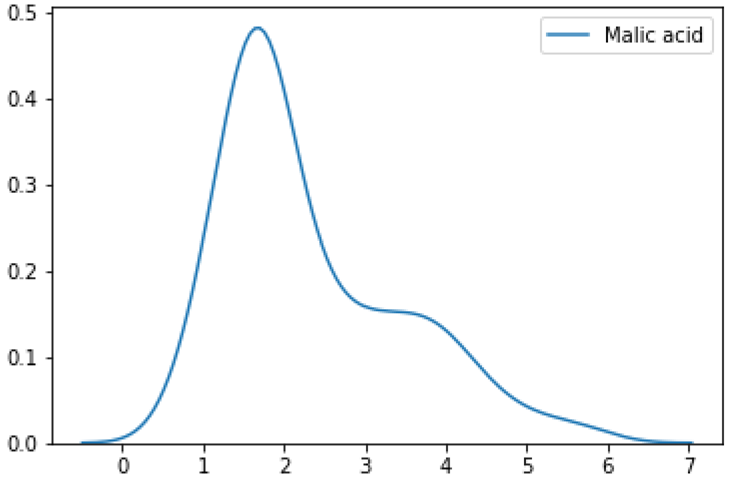
Out[34]:



In [36]:

```
sns.kdeplot(df['Malic acid'])
```

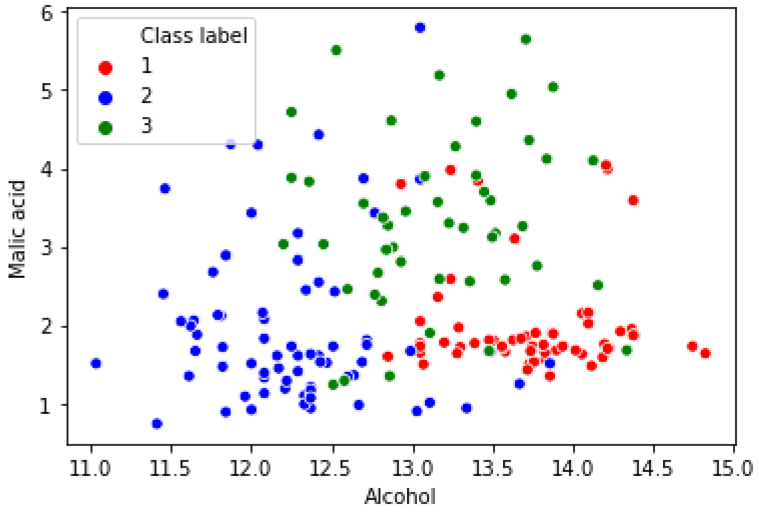
Out[36]:



In [43]:

```
color_dict={1:'red',3:'green',2:'blue'}
sns.scatterplot(df['Alcohol'],df['Malic acid'],hue=df['Class label'],palette=co
```

Out[43]:



In [44]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.drop('Class label', axis
                                                    df['Class label'],
                                                    test_size=0.3,
                                                    random_state=0)


X_train.shape, X_test.shape
```

Out[44]:

```
((124, 2), (54, 2))
```

In [45]:

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
```

```
# fit the scaler to the train set, it will learn the parameters
scaler.fit(X_train)

# transform train and test sets
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [46]:

```
X_train_scaled = pd.DataFrame(X_train_scaled, columns=X_train.columns)
X_test_scaled = pd.DataFrame(X_test_scaled, columns=X_test.columns)
```

In [47]:

```
np.round(X_train.describe(), 1)
```

Out[47]:

	Alcohol	Malic acid
count	124.0	124.0
mean	13.0	2.4
std	0.8	1.1
min	11.0	0.9
25%	12.4	1.6
50%	13.0	1.9
75%	13.6	3.2
max	14.8	5.6

In [48]:

```
np.round(X_train_scaled.describe(), 1)
```

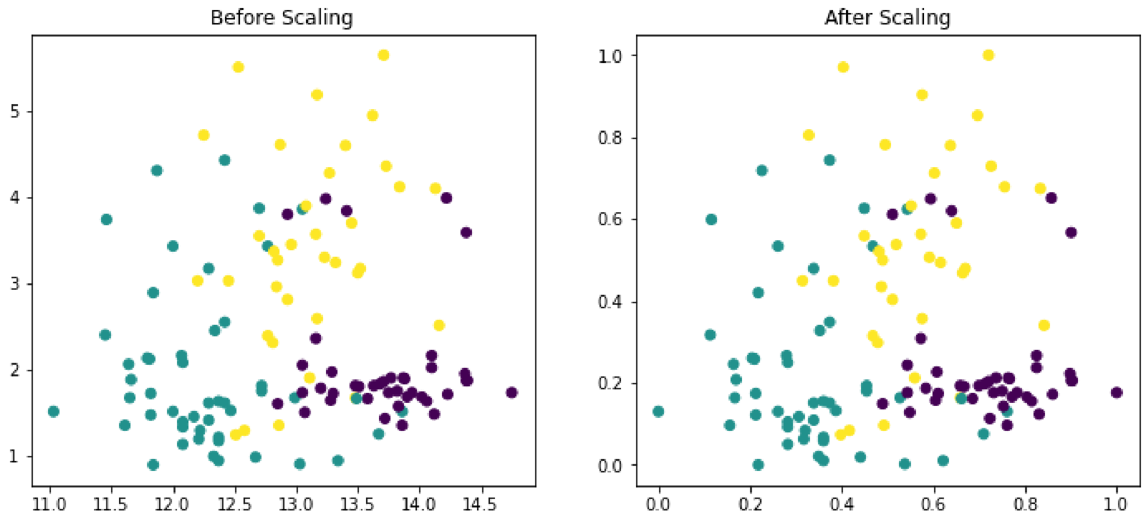
Out[48]:

	Alcohol	Malic acid
count	124.0	124.0
mean	0.5	0.3
std	0.2	0.2
min	0.0	0.0
25%	0.4	0.2
50%	0.5	0.2
75%	0.7	0.5
max	1.0	1.0

In [52]:

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

ax1.scatter(X_train['Alcohol'], X_train['Malic acid'],c=y_train)
ax1.set_title("Before Scaling")
ax2.scatter(X_train_scaled['Alcohol'], X_train_scaled['Malic acid'],c=y_train)
ax2.set_title("After Scaling")
plt.show()
```

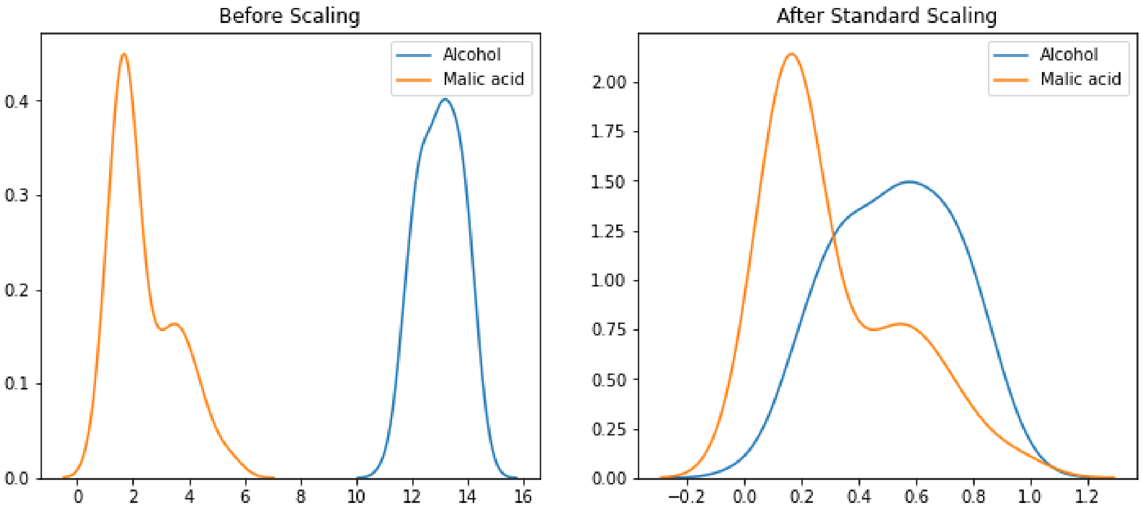


In [53]:

```python
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

# before scaling
ax1.set_title('Before Scaling')
sns.kdeplot(X_train['Alcohol'], ax=ax1)
sns.kdeplot(X_train['Malic acid'], ax=ax1)

# after scaling
ax2.set_title('After Standard Scaling')
sns.kdeplot(X_train_scaled['Alcohol'], ax=ax2)
sns.kdeplot(X_train_scaled['Malic acid'], ax=ax2)
plt.show()
```

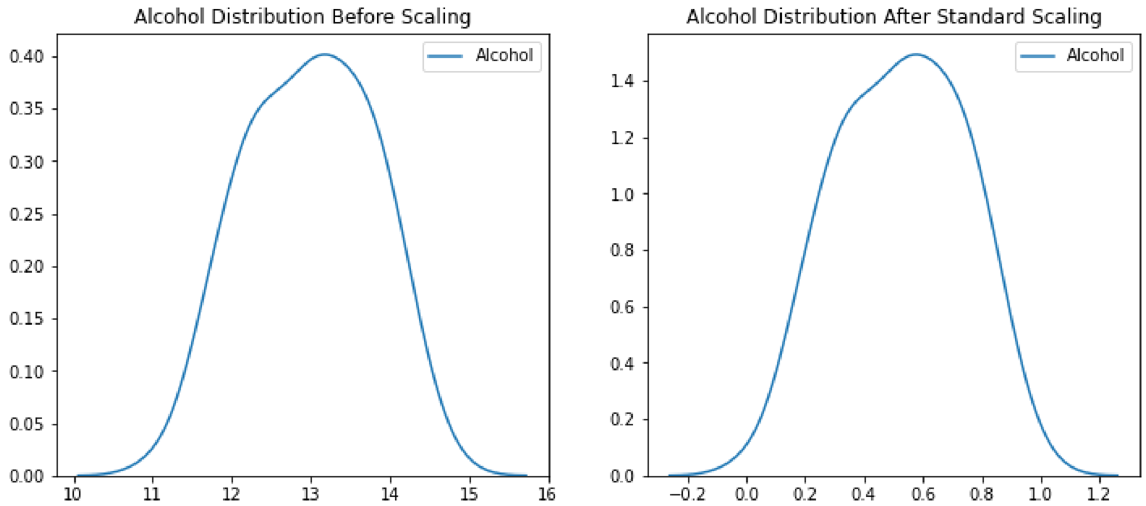


In [54]:

```python
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

# before scaling
ax1.set_title('Alcohol Distribution Before Scaling')
sns.kdeplot(X_train['Alcohol'], ax=ax1)

# after scaling
ax2.set_title('Alcohol Distribution After Standard Scaling')
sns.kdeplot(X_train_scaled['Alcohol'], ax=ax2)
plt.show()
```



Alcohol Distribution Before Scaling

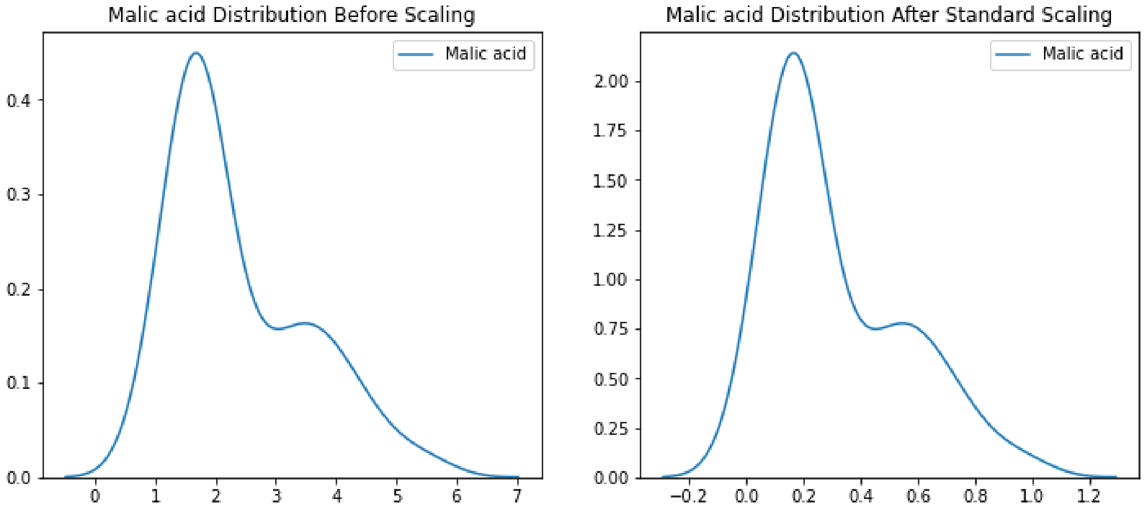
Alcohol Distribution After Standard Scaling

In [55]:

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 5))

# before scaling
ax1.set_title('Malic acid Distribution Before Scaling')
sns.kdeplot(X_train['Malic acid'], ax=ax1)

# after scaling
ax2.set_title('Malic acid Distribution After Standard Scaling')
sns.kdeplot(X_train_scaled['Malic acid'], ax=ax2)
plt.show()
```



Malic acid Distribution Before Scaling

Malic acid Distribution After Standard Scaling

In [ ]: