```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
from sklearn.datasets import load_wine
winedataset = load_wine()
```

```
winedataset.target_names
```

```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

```
winedataset.feature_names
```

```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

```
winedataset.data
```

```
array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
        1.065e+03],
       [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
        1.050e+03],
       [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
        1.185e+03],
       ...,
       [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
        8.350e+02],
       [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
        8.400e+02],
       [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
        5.600e+02]])
```

```
winedataset.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])
```

```
dir(winedataset)
```

```
['DESCR', 'data', 'feature_names', 'frame', 'target', 'target_names']
```

```
df = pd.DataFrame(winedataset.data,columns=winedataset.feature_names)
df['target'] = winedataset.target
```

```
df
```

...nity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	col...
15.6	127.0	2.80	3.06	0.28	2.29	
11.2	100.0	2.65	2.76	0.26	1.28	
18.6	101.0	2.80	3.24	0.30	2.81	
16.8	113.0	3.85	3.49	0.24	2.18	

```
df[df['target']==0].head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonfl...
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	

```
df[df['target']==1].head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonf...
59	12.37	0.94	1.36	10.6	88.0	1.98	0.57	
60	12.33	1.10	2.28	16.0	101.0	2.05	1.09	
61	12.64	1.36	2.02	16.8	100.0	2.02	1.41	
62	13.67	1.25	1.92	18.0	94.0	2.10	1.79	
63	12.37	1.13	2.16	19.0	87.0	3.50	3.10	

```
df[df['target']==2].head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	non...
130	12.86	1.35	2.32	18.0	122.0	1.51	1.25	
131	12.88	2.99	2.40	20.0	104.0	1.30	1.22	
132	12.81	2.31	2.40	24.0	98.0	1.15	1.09	
133	12.70	3.55	2.36	21.5	106.0	1.70	1.20	
134	12.51	1.24	2.25	17.5	85.0	2.00	0.58	

```
x = df.drop(['target'], axis='columns')
y = df.target
```

```
x
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	non-
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	

```
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
173    2
174    2
175    2
176    2
177    2
Name: target, Length: 178, dtype: int64
```

176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	

train test split

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.3)
```

```
from sklearn.naive_bayes import GaussianNB, MultinomialNB
model1 = GaussianNB()
model2 = MultinomialNB()
model1.fit(xtrain,ytrain)
model2.fit(xtrain,ytrain)
```

```
MultinomialNB()
```

```
#GaussianNB
#Accuracy
model1.score(xtest,ytest)
```

```
0.9629629629629629
```

```
#MultinomialNB
#Accuracy
model2.score(xtest,ytest)
```

```
0.8148148148148148
```

```
#cross validation
from sklearn.model_selection import cross_val_score
s1 = cross_val_score(GaussianNB(),xtrain,ytrain)
```

```
s1
```

```
array([1.        , 1.        , 1.        , 0.88      , 0.95833333])
```

```
#Average
np.average(s1)
```

```
0.9676666666666666
```

```
s2 = cross_val_score(MultinomialNB(),xtrain,ytrain)
s2
```

```
array([0.8 , 0.96, 0.92, 0.88, 0.75])
```

```
#Average
np.average(s2)
```

```
0.8620000000000001
```

```
#GaussianNB is the best working
```

0s completed at 11:01 AM