In [35]:
```
import pandas as pd
from sklearn.datasets import load_wine
wine = load_wine()
#imported wine dataset
```

In [36]:
```
wine.feature_names
```

Out[36]:
```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

In [37]:
```
wine.target_names
```

Out[37]:
```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

In [38]:
```
df = pd.DataFrame(wine.data,columns=wine.feature_names)
```

In [39]:
```
df.head()
```

Out[39]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0

In [40]:
```
#craeting target for wine data to divide into 3 categories
df['target'] = wine.target
df.head()
```

Out[40]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0

In [41]:
```
df[df.target==1].head()
```

Out[41]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
59	12.37	0.94	1.36	10.6	88.0	1.98	0.57	0.28	0.42	1.95	1.05	1.82	520.0	1
60	12.33	1.10	2.28	16.0	101.0	2.05	1.09	0.63	0.41	3.27	1.25	1.67	680.0	1
61	12.64	1.36	2.02	16.8	100.0	2.02	1.41	0.53	0.62	5.75	0.98	1.59	450.0	1
62	13.67	1.25	1.92	18.0	94.0	2.10	1.79	0.32	0.73	3.80	1.23	2.46	630.0	1
63	12.37	1.13	2.16	19.0	87.0	3.50	3.10	0.19	1.87	4.45	1.22	2.87	420.0	1

In [42]:
```
df[df.target==2].head()
```

Out[42]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
130	12.86	1.35	2.32	18.0	122.0	1.51	1.25	0.21	0.94	4.10	0.76	1.29	630.0	2
131	12.88	2.99	2.40	20.0	104.0	1.30	1.22	0.24	0.83	5.40	0.74	1.42	530.0	2
132	12.81	2.31	2.40	24.0	98.0	1.15	1.09	0.27	0.83	5.70	0.66	1.36	560.0	2
133	12.70	3.55	2.36	21.5	106.0	1.70	1.20	0.17	0.84	5.00	0.78	1.29	600.0	2
134	12.51	1.24	2.25	17.5	85.0	2.00	0.58	0.60	1.25	5.45	0.75	1.51	650.0	2

In [43]:
```
df['wine_name'] = df.target.apply(lambda x: wine.target_names[x])
df.head()
#checking the dataset after applying categories
```

Out[43]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target	wine_name
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0	class_0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0	class_0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0	class_0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0	class_0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0	class_0

In [44]:
```
df.shape
```

Out[44]:
```
(178, 15)
```

In [45]:
```
df[55:75]
```

Out[45]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target	wine_name
55	13.56	1.73	2.46	20.5	116.0	2.96	2.78	0.20	2.45	6.25	0.980	3.03	1120.0	0	class_0
56	14.22	1.70	2.30	16.3	118.0	3.20	3.00	0.26	2.03	6.38	0.940	3.31	970.0	0	class_0
57	13.29	1.97	2.68	16.8	102.0	3.00	3.23	0.31	1.66	6.00	1.070	2.84	1270.0	0	class_0
58	13.72	1.43	2.50	16.7	108.0	3.40	3.67	0.19	2.04	6.80	0.890	2.87	1285.0	0	class_0
59	12.37	0.94	1.36	10.6	88.0	1.98	0.57	0.28	0.42	1.95	1.050	1.82	520.0	1	class_1
60	12.33	1.10	2.28	16.0	101.0	2.05	1.09	0.63	0.41	3.27	1.250	1.67	680.0	1	class_1
61	12.64	1.36	2.02	16.8	100.0	2.02	1.41	0.53	0.62	5.75	0.980	1.59	450.0	1	class_1
62	13.67	1.25	1.92	18.0	94.0	2.10	1.79	0.32	0.73	3.80	1.230	2.46	630.0	1	class_1
63	12.37	1.13	2.16	19.0	87.0	3.50	3.10	0.19	1.87	4.45	1.220	2.87	420.0	1	class_1
64	12.17	1.45	2.53	19.0	104.0	1.89	1.75	0.45	1.03	2.95	1.450	2.23	355.0	1	class_1
65	12.37	1.21	2.56	18.1	98.0	2.42	2.65	0.37	2.08	4.60	1.190	2.30	678.0	1	class_1
66	13.11	1.01	1.70	15.0	78.0	2.98	3.18	0.26	2.28	5.30	1.120	3.18	502.0	1	class_1
67	12.37	1.17	1.92	19.6	78.0	2.11	2.00	0.27	1.04	4.68	1.120	3.48	510.0	1	class_1
68	13.34	0.94	2.36	17.0	110.0	2.53	1.30	0.55	0.42	3.17	1.020	1.93	750.0	1	class_1
69	12.21	1.19	1.75	16.8	151.0	1.85	1.28	0.14	2.50	2.85	1.280	3.07	718.0	1	class_1
70	12.29	1.61	2.21	20.4	103.0	1.10	1.02	0.37	1.46	3.05	0.906	1.82	870.0	1	class_1
71	13.86	1.51	2.67	25.0	86.0	2.95	2.86	0.21	1.87	3.38	1.360	3.16	410.0	1	class_1
72	13.49	1.66	2.24	24.0	87.0	1.88	1.84	0.27	1.03	3.74	0.980	2.78	472.0	1	class_1
73	12.99	1.67	2.60	30.0	139.0	3.30	2.89	0.21	1.96	3.35	1.310	3.50	985.0	1	class_1
74	11.96	1.09	2.30	21.0	101.0	3.38	2.14	0.13	1.65	3.21	0.990	3.13	886.0	1	class_1

In [46]:
```
from sklearn.model_selection import train_test_split
#train&test&split the dataset
```

In [47]:
```
x = df.drop(['target','wine_name'],axis='columns')
y = df.target
```

In [48]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

In [49]:
```
len(x_train)
```

Out[49]:
```
142
```

In [50]:
```
len(x_test)
```

Out[50]:
```
36
```

In [51]:
```
#applying multinomialNB
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(x_train,y_train)
```

Out[51]:
```
MultinomialNB()
```

In [52]:
```
model.score(x_test,y_test)
```

Out[52]:
```
0.8611111111111112
```

In [53]:
```
#applying GaussianNB
from sklearn.naive_bayes import GaussianNB
Model = GaussianNB()
Model.fit(x_train,y_train)
```

Out[53]:
```
GaussianNB()
```

In [57]:
```
Model.score(x_test,y_test)
```

Out[57]:
```
0.9722222222222222
```

In [61]:
```
#predicting the test data
model.predict(x_test)     #for multinomialNB
```

Out[61]:
```
array([1, 1, 1, 0, 2, 2, 1, 1, 2, 1, 1, 2, 2, 0, 1, 2, 0, 0, 1, 1, 0, 2,
       1, 2, 1, 0, 1, 1, 0, 2, 0, 1, 1, 2, 0, 2])
```

In [62]:
```
Model.predict(x_test)   #for gaussianNB
```

Out[62]:
```
array([1, 1, 1, 0, 2, 1, 2, 1, 1, 1, 1, 2, 2, 0, 1, 2, 0, 1, 1, 1, 0, 2,
       1, 2, 1, 0, 1, 1, 0, 2, 0, 0, 1, 2, 0, 2])
```

In [ ]: