Naive Bayes Exercise

Use wine dataset from sklearn.datasets to classify wines into 3 categories. Load the dataset and split it into test and train. After taht train the model using Gaussian and Multinomial classifier and find which model performs better. Use the train model to perform some predictions on test data.

In [2]:
```
from sklearn import datasets
wine = datasets.load_wine()
```

In [3]:
```
dir(wine)
```

Out[3]:
```
['DESCR', 'data', 'feature_names', 'frame', 'target', 'target_names']
```

In [4]:
```
wine.data[0:2]
```

Out[4]:
```
array([[1.423e+01, 1.710e+00, 2.430e+00, 1.560e+01, 1.270e+02, 2.800e+00,
        3.060e+00, 2.800e-01, 2.290e+00, 5.640e+00, 1.040e+00, 3.920e+00,
        1.065e+03],
       [1.320e+01, 1.780e+00, 2.140e+00, 1.120e+01, 1.000e+02, 2.650e+00,
        2.760e+00, 2.600e-01, 1.280e+00, 4.380e+00, 1.050e+00, 3.400e+00,
        1.050e+03]])
```

In [5]:
```
wine.feature_names
```

Out[5]:
```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

In [6]:
```
wine.target_names
```

Out[6]:
```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

In [7]:
```
wine.target[0:2]
```

Out[7]:
```
array([0, 0])
```

In [8]:
```
import pandas as pd
df = pd.DataFrame(wine.data,columns=wine.feature_names)
df.head()
```

Out[8]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_
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	

In [10]:
```
df['target'] = wine.target
df[50:60]
```

Out[10]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoic
50	13.05	1.73	2.04	12.4	92.0	2.72	3.27	
51	13.83	1.65	2.60	17.2	94.0	2.45	2.99	
52	13.82	1.75	2.42	14.0	111.0	3.88	3.74	
53	13.77	1.90	2.68	17.1	115.0	3.00	2.79	
54	13.74	1.67	2.25	16.4	118.0	2.60	2.90	
55	13.56	1.73	2.46	20.5	116.0	2.96	2.78	
56	14.22	1.70	2.30	16.3	118.0	3.20	3.00	
57	13.29	1.97	2.68	16.8	102.0	3.00	3.23	
58	13.72	1.43	2.50	16.7	108.0	3.40	3.67	
59	12.37	0.94	1.36	10.6	88.0	1.98	0.57	

In [11]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(wine.data, wine.target, test_
```

In [12]:
```
from sklearn.naive_bayes import GaussianNB, MultinomialNB
model = GaussianNB()
model.fit(X_train,y_train)
```

Out[12]:
```
GaussianNB()
```

In [13]:
```
model.score(X_test,y_test)
```

Out[13]:
```
0.9722222222222222
```

In [14]:
```
mn = MultinomialNB()
mn.fit(X_train,y_train)
mn.score(X_test,y_test)
```

Out[14]:
```
0.8055555555555556
```

GaussianNB has a better performan than MultinomialNB in this exercise.

	GaussianNB	MultinomialNB
Model Score(%)	97	80

Date	Author
2021-10-13	**Ehsan Zia**