

```
In [1]: # Author : Amir Shokri
# github link : https://github.com/amirshnll/Wine
# dataset link : http://archive.ics.uci.edu/ml/datasets/Wine
# email : amirsh.nll@gmail.com
```

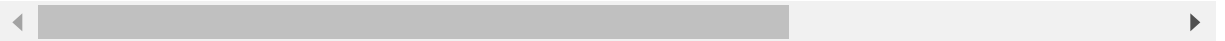
```
In [27]: import matplotlib.pyplot as plt
import pandas as pd
```

```
In [21]: col_names = ['class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Ma
gnesium', 'Total phenols', 'Flavanoids', ' Nonflavanoid phenols', 'Proanthocyan
ins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline']
wine =pd.read_csv("wine.csv",header=None, names=col_names)
```

```
In [22]: wine.head()
```

Out[22]:

	class	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proan
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	



```
In [28]: inputs =wine.drop('class',axis='columns')
target = wine['class']
```

In [24]: inputs

Out[24]:

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthoc
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	
...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	

178 rows × 13 columns



In [29]: `from sklearn.model_selection import train_test_split`

```
X_train, X_test, y_train, y_test = train_test_split(inputs, target, test_size=0.3, random_state=109)
```

In [30]: `from sklearn.naive_bayes import GaussianNB`

```
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
```

In [31]: `from sklearn import metrics`

```
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.9074074074074074

In [ ]: