Task: classify origin of wine based on physio-chemical analysis data.

You are provided data that are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Details can be **found here** (http://archive.ics.uci.edu/ml/datasets/Wine).

In [1]:
```
import pandas as pd
import seaborn as sns
import statsmodels.formula.api as sm

%matplotlib inline
import matplotlib.pyplot as plt
import csv
import pandas
import sklearn
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer, TfidfTran
sformer
from sklearn.svm import SVC, LinearSVC
from sklearn.metrics import classification_report, f1_score, accuracy_
score, confusion_matrix
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold, cross_val_score,
train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import learning_curve
```

Read in the data

Data set

In [2]:
```
df = pd.read_csv('wine.data.csv', header=None)
df.columns = ['Class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline']
df.head(10)
```

Out[2]:

	Class	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proa
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	
5	1	14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	
6	1	14.39	1.87	2.45	14.6	96	2.50	2.52	0.30	
7	1	14.06	2.15	2.61	17.6	121	2.60	2.51	0.31	
8	1	14.83	1.64	2.17	14.0	97	2.80	2.98	0.29	
9	1	13.86	1.35	2.27	16.0	98	2.98	3.15	0.22	

Describe the basic statistics of the features

In [3]:
```
print("The data size is\n")
df.shape
```

```
The data size is


(178, 14)
```

In [4]:
```
df['Class'].value_counts()
```

```
2    71
1    59
3    48
Name: Class, dtype: int64
```

In [5]:
```
a = df['Alcohol'].describe()
b = df['Malic acid'].describe()
c = df['Ash'].describe()
d = df['Alcalinity of ash'].describe()
e = df['Magnesium'].describe()
f = df['Total phenols'].describe()
g = df['Flavanoids'].describe()
h = df['Nonflavanoid phenols'].describe()
i = df['Proanthocyanins'].describe()
j = df['Color intensity'].describe()
k = df['Hue'].describe()
l = df['OD280/OD315 of diluted wines'].describe()
m = df['Proline'].describe()
dfs = pd.concat([a, b, c, d, e, f, g, h, i, j, k, l, m], axis = 1)
print("The basic statistics of the features are\n")
dfs
```

```
The basic statistics of the features are
```

Out[5]:

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	N
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	

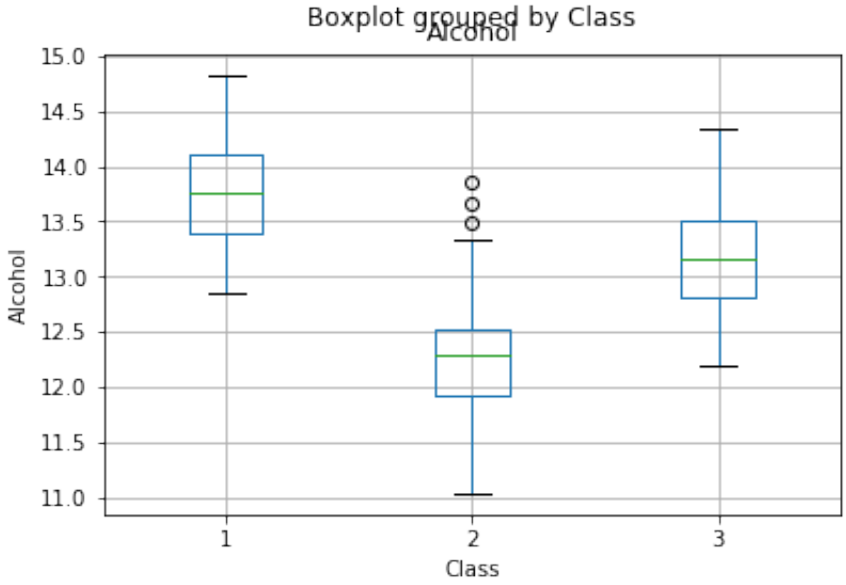
Make boxplots by output labels/classes - do any features classify the wine based on these figures?

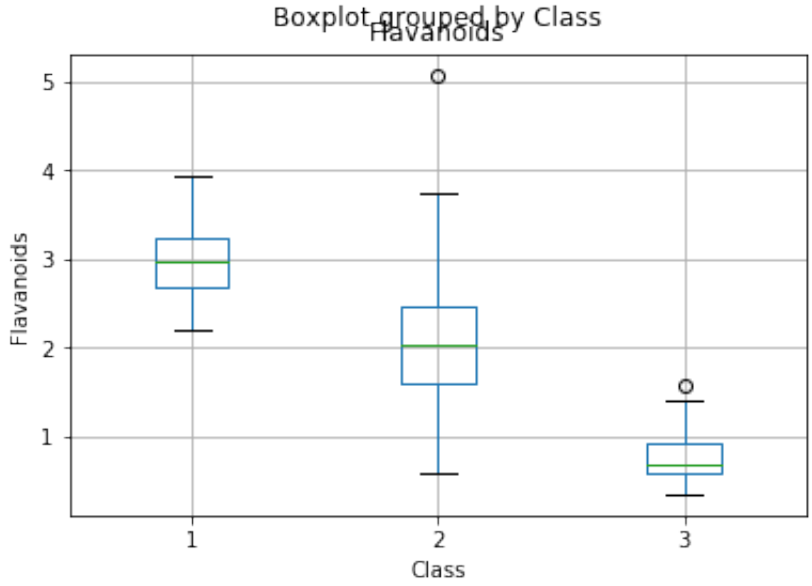
If so (and hint, they do!), make a scatter plot showing the correlation of two features showing the correlation of two features and class separation by these features

In [6]:
```
df.boxplot('Alcohol', by = 'Class')
plt.ylabel("Alcohol")

df.boxplot('Flavanoids', by = 'Class')
plt.ylabel("Flavanoids")
print("Yes, the boxplots of alcohol and flavanoids features showing th
is two features classify wine.")
```

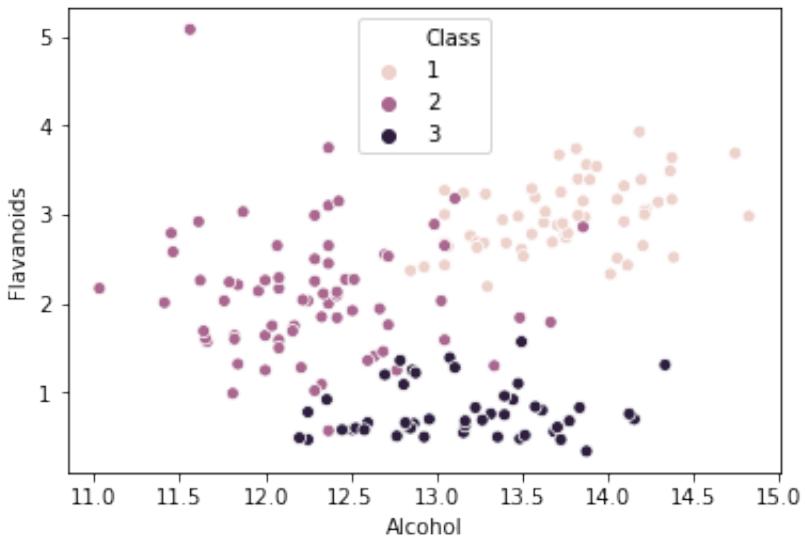
```
Yes, the boxplots of alcohol and flavanoids features showing this tw
o features classify wine.
```





In [7]:
```
sns.scatterplot(x="Alcohol", y="Flavanoids", hue = "Class", legend = "
full", data = df)
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1a21abcb70>



Naive Bayes Classification

Use Naive Bayes Classification (https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html) to create a model to classify wine base on attributes. Justify how good the model is for the wine classification. Note that some of the metrics we've used in class are only for *binary* classifications, so may not be applicable here.

In [8]:
```
# Define x and y
x = df.drop('Class', 1)
y = df['Class']
print (x.shape)
print (y.shape)
```

```
(178, 13)
(178,)
```

In [9]:
```
# Split into training and testing dataset
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.5, random_state=1)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(89, 13)
(89,)
(89, 13)
(89,)
```

In [10]:
```
# Import and instantiate the model
from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()
clf.fit(x_train, y_train)
```

Out[10]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [11]:
```
y_pred_class = clf.predict(x_test)
print(y_pred_class.shape)
```

```
(89,)
```

In [12]:
```
from sklearn import metrics
metrics.accuracy_score(y_test, y_pred_class)
```

Out[12]:
```
0.9775280898876404
```

In [13]:
```
# Print the confusion matrix
cm = metrics.confusion_matrix(y_test, y_pred_class)
cmdf = pd.DataFrame(cm,index=['1', '2', '3'], columns=['1','2','3'])
print("The confusion matrix for three wine classes looks like followin
g...\n")
cmdf
```

```
The confusion matrix for three wine classes looks like following...
```

Out[13]:

	1	2	3
1	33	0	0
2	1	32	1
3	0	0	22

After create and test the model, the accuracy of the model is about 0.977 which is really high accuracy. So the model is working effectively for wine classification. One of reason that we are getting high accuracy is probably because the data size is relatively small. The confusion matrix above shows the three wine types and the predicted values. Original code and data are posted on github. (https://github.com/joyleeisu/ABE516X-Naive-Bayes.git (https://github.com/joyleeisu/ABE516X-Naive-Bayes.git))