In [8]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

In [9]:

```
wine = pd.read_csv('wine.data', names = ["Cultivator", "Alchol", "Malic_Acid", "Ash", "Alcalinity_of_Ash", "Magnesium", "Total_phenols", "Falvanoids", "Nonflavanoid_phenols", "Proanthocyanins", "Color_intensity", "Hue", "OD280", "Proline"])
print(wine.shape)
wine.head()
# set seed for consistent random values
np.random.seed(42)
```

```
(178, 14)
```

In [10]:

```
X = wine.drop('Cultivator',axis=1)
y = wine['Cultivator']
```

In [11]:

```
def build_model(X, y, test_size):
    # split dataset into traing and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = test_size)

    # normalize data
    scaler = StandardScaler()
    scaler.fit(X_train)
    X_train = scaler.transform(X_train)
    X_test = scaler.transform(X_test)


    # build model & train
    model = MLPClassifier(hidden_layer_sizes=(13,13,13), max_iter=500)
    model.fit(X_train, y_train)

    # test model
    y_hat = model.predict(X_test)

    confusion_mtx = confusion_matrix(y_test, y_hat)
    classification_rpt = classification_report(y_test, y_hat)

    return model, confusion_mtx, classification_rpt
```

In [16]:

```
test_sizes = [.2, .3, .4]

for test_size in test_sizes:
    print('=============================\nTest size: {}\n============================='.format(test_size))
    model, cm, cr = build_model(X, y, test_size)

    # print('model weights: {}, model intercepts: {}'.format(model.coefs_, model.intercepts_))
    print('Confusion Matrix:')
    print(model.classes_)
    print(cm)
    print('Classification Report:')
    print(cr)
```

```
=============================
Test size: 0.2
=============================
Confusion Matrix:
[1 2 3]
[[10  0  0]
 [ 0 17  1]
 [ 0  1  7]]
Classification Report:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        10
           2       0.94      0.94      0.94        18
           3       0.88      0.88      0.88         8

    accuracy                           0.94        36
   macro avg       0.94      0.94      0.94        36
weighted avg       0.94      0.94      0.94        36

=============================
Test size: 0.3
=============================
Confusion Matrix:
[1 2 3]
[[17  1  0]
 [ 0 22  0]
 [ 0  2 12]]
Classification Report:
              precision    recall  f1-score   support

           1       1.00      0.94      0.97        18
           2       0.88      1.00      0.94        22
           3       1.00      0.86      0.92        14

    accuracy                           0.94        54
   macro avg       0.96      0.93      0.94        54
weighted avg       0.95      0.94      0.94        54

=============================
Test size: 0.4
=============================
Confusion Matrix:
[1 2 3]
[[25  1  0]
 [ 1 23  0]
 [ 0  1 21]]
```

```
Classification Report:
              precision    recall  f1-score   support

           1       0.96      0.96      0.96        26
           2       0.92      0.96      0.94        24
           3       1.00      0.95      0.98        22

    accuracy                           0.96        72
   macro avg       0.96      0.96      0.96        72
weighted avg       0.96      0.96      0.96        72
```

In [ ]:

```

```

In [ ]:

```
The confusion matrix shows the number of entries the model that were classified for each label.  We look at the
diagonal line as the entries correctly labled (TP/TN) in the set, while any other cell shows as a mis-classific
ation (FP/TN).
```