1 Splitting data into training and test sets

To test the accuracy of a model we will test the model on data that it has not seen before. We will divide available data into two sets: a training set that the model will learn from, and a test set which will be used to test the accuracy of the model on new data. A convenient way to split the data is to use scikit-learn's train_test_split method. This randomly divides the data between training and test sets. We may specify what proportion to keep for the test set (0.2 - 0.3 is common)

```
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split

# Load the iris data

iris=datasets.load_iris()

# Extra out the feature data (data), and the classification (target)

X=iris.data
y=iris.target

X_train,X_test,y_train,y_test=train_test_split(
        X,y,test_size=0.3,random_state=0)

# Random_state is integer seed.
# If this is omitted than a different seed will be used each time
```

Let's look at the size of the data sets:

```
print ('Shape of X:', X.shape)
print ('Shape of y:', y.shape)
print ('Shape of X_train:', X_train.shape)
print ('Shape of y_train:', y_train.shape)
print ('Shape of X_test:', X_test.shape)
print ('Shape of y_test:', y_test.shape)

OUT:

Shape of X: (150, 4)
Shape of y: (150,)
Shape of X_train: (105, 4)
Shape of y_train: (105,)
Shape of X_test: (45, 4)
Shape of y_test: (45,)
```

The data has been split randomly, 70% into the training set and 30% into the test set.