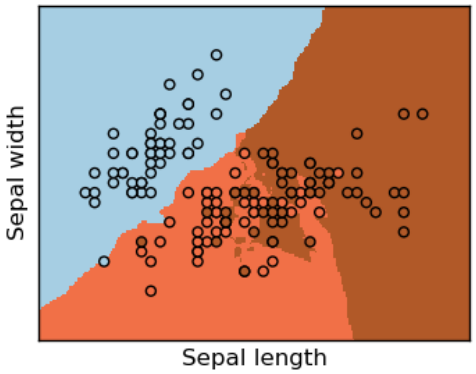
Training set and testing set

- Machine learning is about learning some properties of a data set and applying them to new data.
- This is why a common practice in machine learning to evaluate an algorithm is to split the data at hand in two sets ***training data*** and ***testing data***.
- We learn data properties and fit the model with the training data,
- On the testing set, we test the mode that we have fitted.

When experimenting with a learning algorithm, it is important not to test the prediction of an estimator on the data used to fit the estimator, as this would not be evaluating the performance of the estimator on new data.

KNN (k nearest neighbors) classification example:



Firstly, we will split the iris data in training and testing data sets. We will use a random permutation, to split the data randomly

```
>>> np.random.seed(0)
>>> indices = np.random.permutation(len(iris_X))
>>> iris_X_train = iris_X[indices[:-10]]
>>> iris_y_train = iris_y[indices[:-10]]
>>> iris_X_test  = iris_X[indices[-10:]]
>>> iris_y_test  = iris_y[indices[-10:]]
```

Now we will create and fit a nearest-neighbor classifier

```
>>> from scikits.learn.neighbors import NeighborsClassifier
>>> knn = NeighborsClassifier()
>>> knn.fit(iris_X_train, iris_y_train)

NeighborsClassifier(n_neighbors=5,
leaf_size=20, algorithm='auto')
```

Now lets use our model on the test data

```
>>> knn.predict(iris_X_test)

array([1, 2, 1, 0, 0, 0, 2, 1, 2, 0])

>>> iris_y_test

array([1, 1, 1, 0, 0, 0, 2, 1, 2, 0])
```