0.0.1 2.2.4. Classification

For classification, as in the labeling iris task, linear regression is not the right approach, as it will give too much weight to data far from the decision frontier. A linear approach is to fit a sigmoid function, or logistic function:

$$y = \text{sigmoid}(X\beta - \text{offset}) + \epsilon = \frac{1}{1 + \exp(-X\beta + \text{offset})} + \epsilon$$

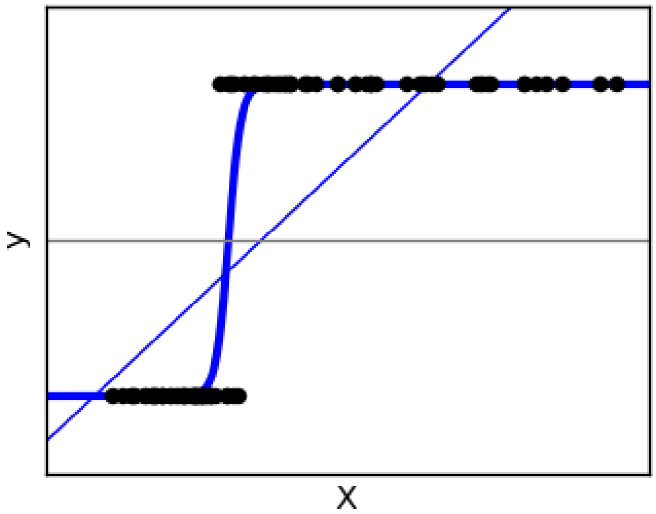
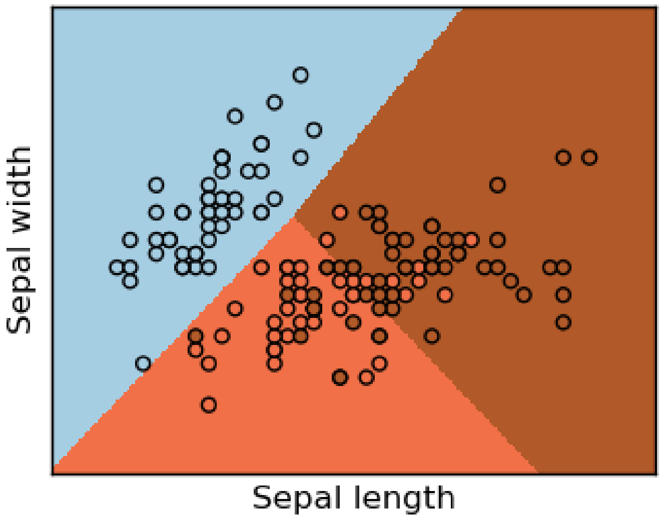


Figure 1

```
>>> logistic = linear_model.LogisticRegression(C=1e5)
>>> logistic.fit(iris_X_train, iris_y_train)
LogisticRegression(C=100000.0, intercept_scaling=1, dual=False,
          fit_intercept=True, penalty='l2', tol=0.0001)
```



0.1 Multiclass classification

If you have several classes to predict, an option often used is to fit one-versus-all classifiers, and use a voting heuristic for the final decision.

0.2 Shrinkage and sparsity with logistic regression

The C parameter controls the amount of regularization in the LogisticRegression object, the bigger C, the less regularization. penalty="l2" gives shrinkage (i.e. non-sparse coefficients), while penalty="l1" gives sparsity.

Excercise

Try classifying the digits dataset with nearest neihbors and a linear model. Leave out the last 10% and test prediction performance on these observations.