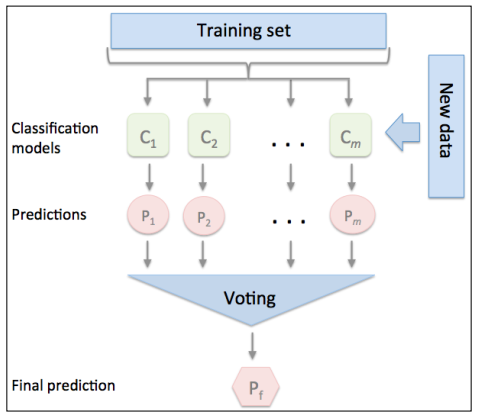
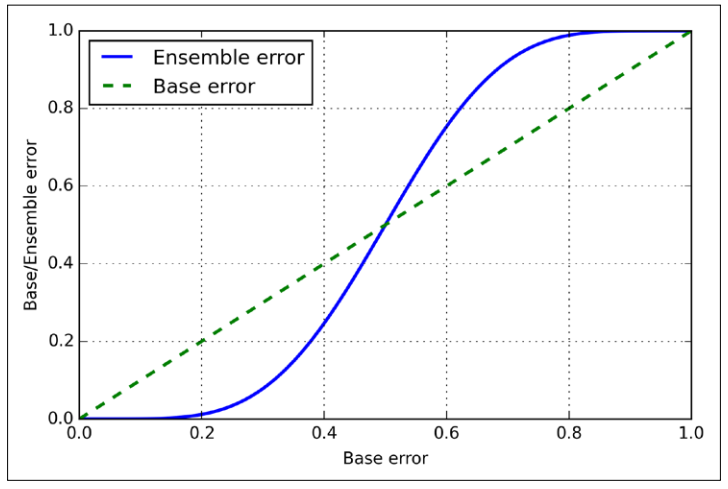
**Majority Voting**

Majority voting simply means that we select the class label that has been predicted by the majority of classifiers, that is, received more than 50 percent of the votes.

Using the training set, we start by training m different classifiers (C1,…,Cm). Depending on the technique, the ensemble can be built from different classification algorithms, for examples, decision trees, support vector machines, logistic regression classifiers, and so on. Alternatively, we can also use the same base classification algorithm fitting different subsets of the training set. One prominent example of this approach would be the random forest algorithm, which combines different decision tree classifiers. The following diagram illustrates the concept of a general ensemble approach using majority voting.



The Error probability of an ensemble is always better than the error of an individual base classifier as long as the base classifiers perform better than random guessing . Note that the y-axis depicts the base error (dotted line) as well as the ensemble error (continuous line):

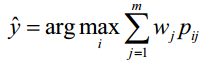


**Simple majority vote classifier:**

The algorithm that we are going to implement will allow us to combine different classification algorithms associated with individual weights for confidence. Our goal is to build a stronger meta-classifier that balances out the individual classifiers weaknesses on a particular dataset. In more precise mathematical terms, we can write the weighted majority vote as follows:



Using the predicted class probabilities instead of the class labels for majority voting can be useful if the classifiers in our ensemble are well calibrated. The modified version of the majority vote for predicting class labels from probabilities can be written as follows:



Here, pij is the predicted probability of the jth classifier for class label i.

Code:github.

**Evaluating and tuning the ensemble classifier**

Let's now tune the inverse regularization parameter C of the logistic regression classifer and the decision tree depth via a grid search for demonstration purposes.

Reference:

1. Python machine learning