Log[a]b = (log[10]b/log[10]a)

Filter - pre classifiear.
-information gain
"how much information about the Classification the feature provides"
-variance of individual features
""
-correlation among features

Wrapper -

Filter Methods
Pros: Fast
Cons: Chosen filter might not be relevant for a specific kind of classifier.
Doesn't take into account interactions among features
Often hard to know how many features to select.

Wrapper Methods
Pros: Features are evaluated in context of classification
Wrapper method selects number of features to use
Cons: Slow

Embedded methods
-Result is that most of the weights go to zero, leaving a small subset of the weights.

information gain. How it's used for feature selection.

Adaboost algorithm

- Given $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ where $\mathbf{x} \in \boldsymbol{X}, y_i \in \{+1, -1\}$
- Initialize $\mathbf{w}_1(i) = 1/N$. (Uniform distribution over data)
- For $t = 1, \ldots, K$:
  - Select new training set S_t from S with replacement, according to $\mathbf{w}_t$
  - Train L on S_t to obtain hypothesis h_t
  - Compute the training error ε_t of h_t on $\boldsymbol{S}$:

$$\varepsilon_t = \sum_{j=1}^{N} \mathbf{w}_t(j)\ \delta(y_j \neq h_t(\mathbf{x}_j)),\ \text{where}$$

$$\delta(y_j \neq h_t(\mathbf{x}_j)) = \begin{cases} 1 \text{ if } y_j \neq h_t(\mathbf{x}_j) \\ 0 \text{ otherwise} \end{cases}$$

  - Compute coefficient
  - Compute new weights on data:

For $i = 1$ to N

$$\mathbf{w}_{t+1}(i) = \frac{\mathbf{w}_t(i)\ \exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t}$$

where Z_t is a normalization factor chosen so that $\mathbf{w}_{t+1}$ will be a probability distribution:

$$Z_t = \sum_{i=1}^{N} \mathbf{w}_t(i)\ \exp(-\alpha_t y_i h_t(\mathbf{x}_i))$$

- At the end of K iterations of this algorithm, we have

$h_1, h_2, \ldots, h_K$

We also have

$\alpha_1, \alpha_2, \ldots, \alpha_K$, where

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

- Ensemble classifier:

$$H(\mathbf{x}) = \operatorname{sgn}\sum_{t=1}^{K}\alpha_t h_t(\mathbf{x})$$

- Note that hypotheses with higher accuracy on their training sets are weighted more strongly.

Then define:

$$Entropy(S_f^{high}) = -(p_+^{high}\log_2 p_+^{high} + p_-^{high}\log_2 p_-^{high})$$

$$Entropy(S_f^{low}) = -(p_+^{low}\log_2 p_+^{low} + p_-^{low}\log_2 p_-^{low})$$

$$Entropy(S_f) = \frac{|S_f^{high}|}{|S|}Entropy(S_f^{high}) + \frac{|S_f^{low}|}{|S|}Entropy(S_f^{low})$$

$$InformationGain(S_f) = Entropy(S) - Entropy(S_f)$$