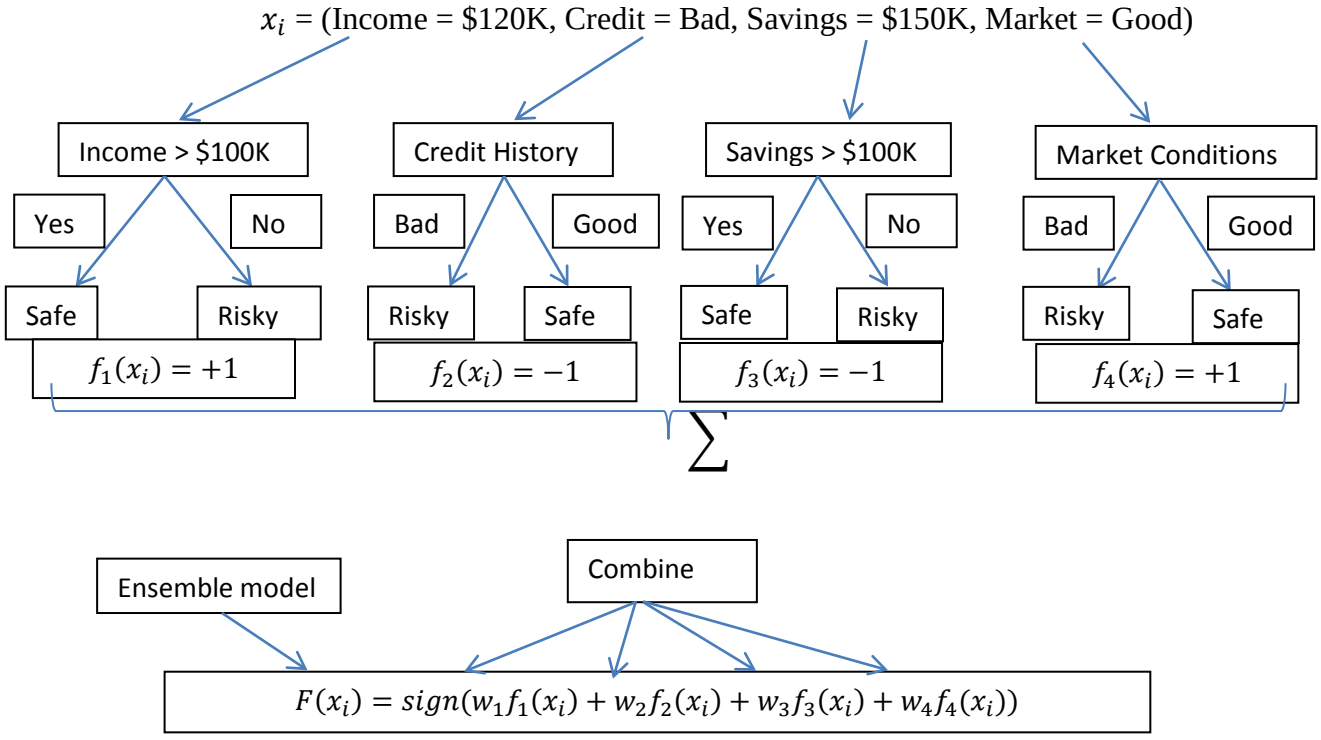
- The way we can think intuitively about boosting is that we can start to fork out week classifiers.
- The question is, how do we go from a weak classifier to something that has lower error? One approach is to add more features. Another is boosting.
- At the core of boosting is the idea of an ensemble classifier.

- Ensemble methods: Each classifier "votes" on prediction.



- Ensemble classifier in general

- Goal:
Predict output y (+1 or -1) from input x
- Learn ensemble model:
Classifiers: $f_1(x), f_2(x), \ldots, f_T(x)$
Coefficients: $\widehat{w}_1, \widehat{w}_2, \ldots, \widehat{w}_T$
- Prediction:
$\hat{y} = sign(\sum_{t=1}^{T} \widehat{w}_t f_t(x))$.

- Boosting: Learn where f(x) makes mistakes and focus next classifier on places where f(x) does less well.

- Learning on weighted data:

--- More weight on “hard” or more important points.
-- Weighted dataset:
- Each x_i, y_i weighted by α_i
More important point = higher weight α_i.
-- Learning:
- Data point j counts as α_i data points.
E.g., $\alpha_i = 2 \rightarrow$ count point twice

- Learning from weighted data in general

--- Usually, learning from weighted data
-- Data point I counts as α_i data points.
--- E.g., gradient ascent for logistic regression:

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta \sum_{i=1}^{N} \alpha_i h_j(x_i)(I[y_i = +1] - P(y = +1|x_i, w^{(t)}))$$

- AdaBoost: learning ensemble

--- Start same weight for all points: $\alpha_i = 1/N$
--- For t = 1, …, T:
-- Learn $f_t(x)$ with data weights α_i
-- Compute coefficient $\widehat{w}_t$
-- Recompute weights α_i.
--- Final model predicts by: $\hat{y} = sign(\sum_{t=1}^{T} \widehat{w_t} f_t(x))$.

- AdaBoost: Computing coefficient $\widehat{w}_t$ of classifier $f_t(x)$

--- $f_t(x)$ is good $\rightarrow$ f_t has low training error.
--- Measuring error in weighted data?
Just weighted # of misclassified points.

- Weighted classification error

--- Total weight of mistakes = $\sum_{i=1}^{N} \alpha_i \mathrm{I}(\hat{y}_i \neq y_i)$
--- Total weight of all points = $\sum_{i=1}^{N} \alpha_i$
--- Weighted error measures fraction of weight of mistakes:

$$weighted\_error = \frac{Total\ weight\ of\ mistakes}{Total\ weight\ of\ all\ data\ points}$$

- AdaBoost: Formula for computing coefficient $\widehat{w}_t$ of classifier $f_t(x)$

$$\widehat{w}_t = \frac{1}{2}\ln\left(\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}\right)$$

- AdaBoost: Updating weights α_i based on where classifier $f_t(x)$ makes mistakes.

- AdaBoost: Formula for updating weights α_i

$$\alpha_i \leftarrow \begin{cases} \alpha_i e^{-\widehat{w}_t}, & if\ f_t(x_i) = y_i \\ \alpha_i e^{\widehat{w}_t}, & if\ f_t(x_i) \neq y_i \end{cases}$$

- AdaBoost: Normalizing weights α_i

Normalize weights to add up to 1 after every iteration.

$$\alpha_i \leftarrow \frac{\alpha_i}{\sum_{i=1}^{N} \alpha_i}$$