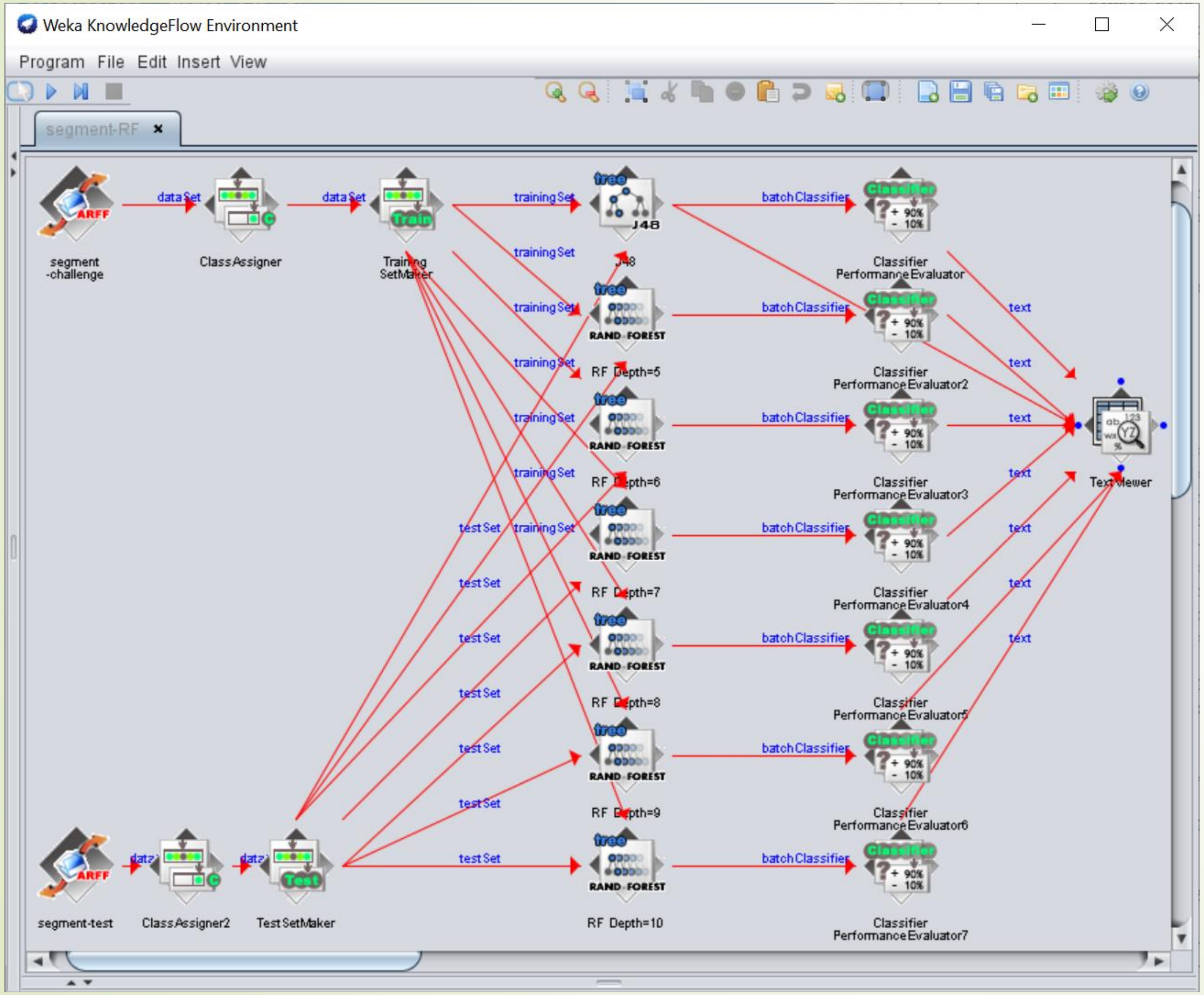
香港城市大學
City University of Hong Kong

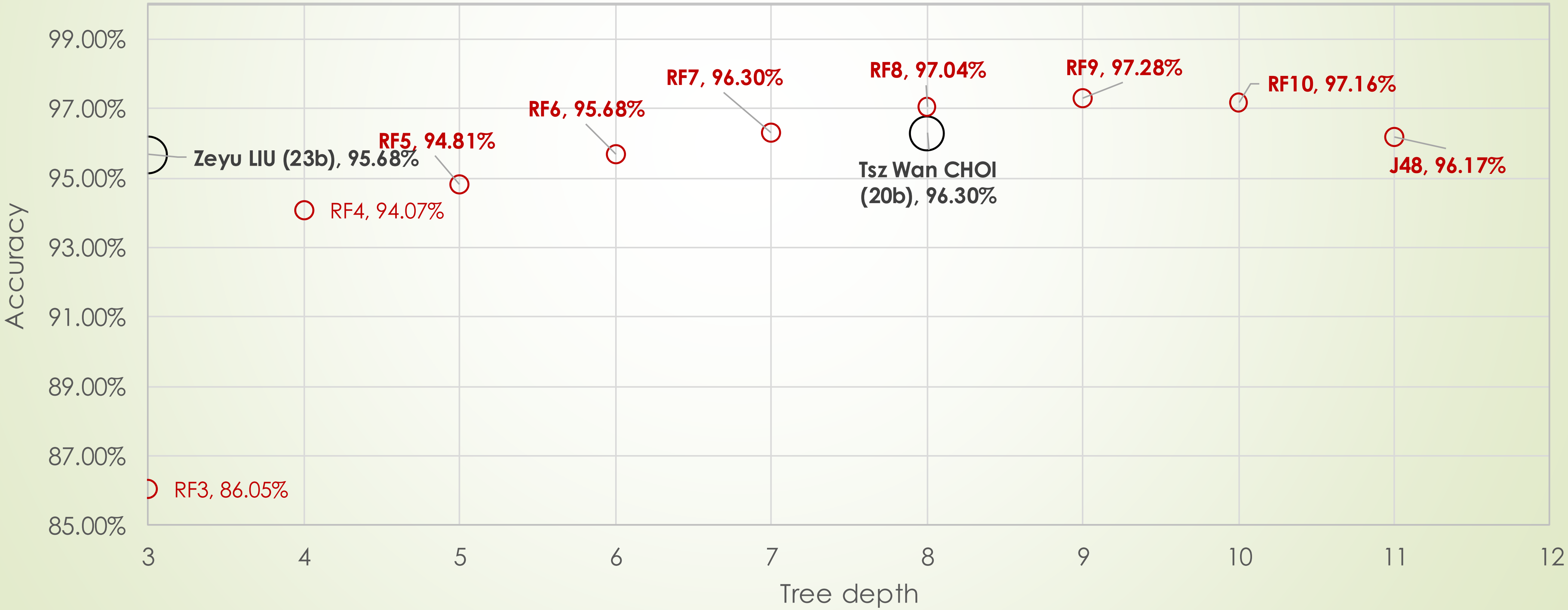
Classification: Ensemble Methods

CS5483 Data Warehousing and Data Mining

Man vs Machine Rematch



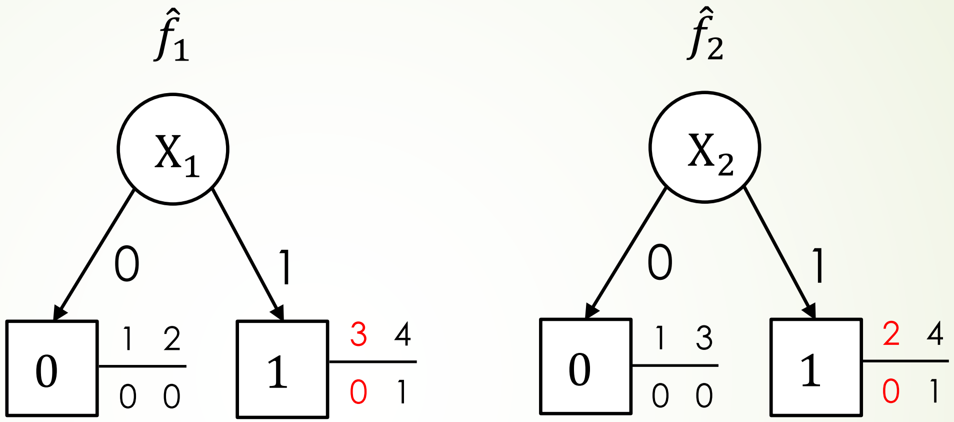
Segment Challenge Results



Two heads are better than one

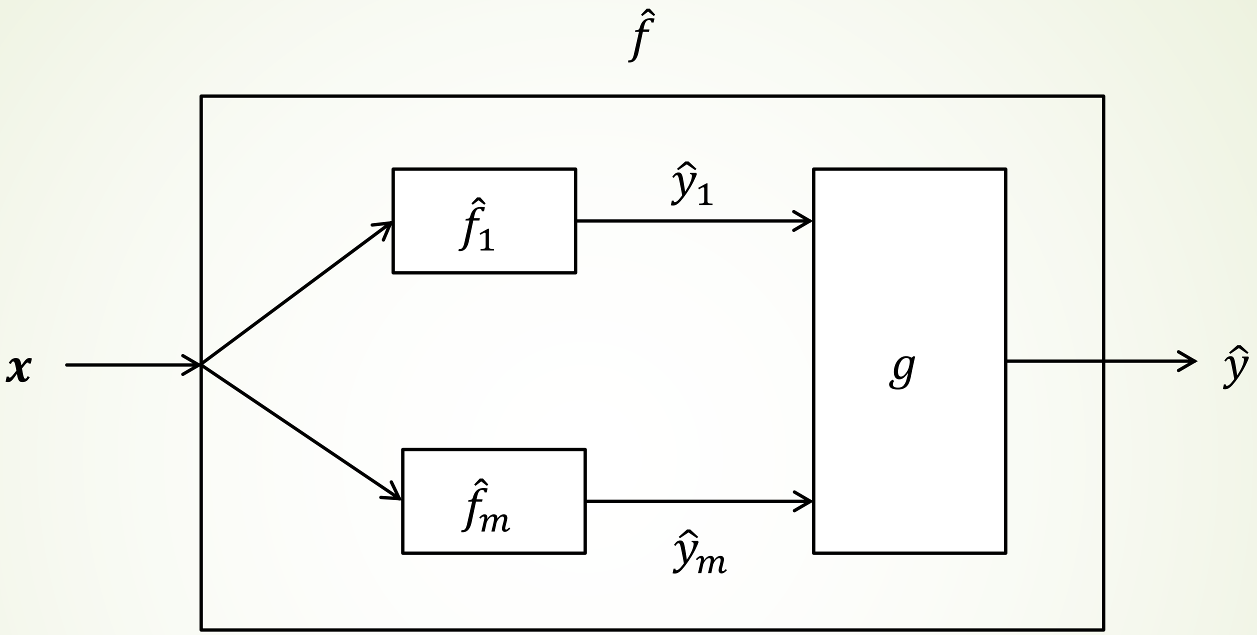
- Bing/Baidu/Google translation.
- The story in Chinese and its translation to English.
- Can we combine two poor classifiers into a good classifier?
- What is the benefit of doing so?

	X_1	X_2	Y
1.	0	0	0
2.	0	1	0
3.	1	0	0
4.	1	1	1



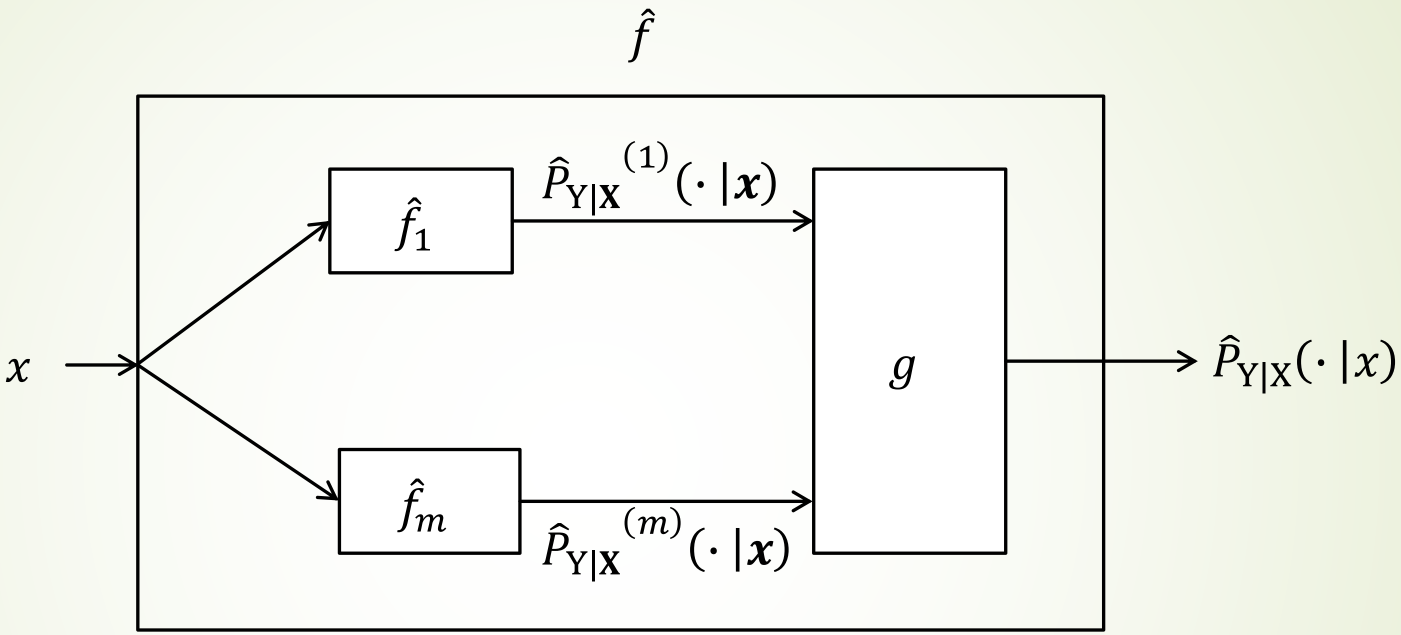
- Accuracies of $\hat{f}_1$ and $\hat{f}_2$ are both ______%. Are they good?
- Can we combine them into a better classifier $\hat{f}(x) := g\left(\hat{f}_1(x), \hat{f}_2(x)\right)$?
- ______$\{\hat{f}_1(x), \hat{f}_2(x)\}$ achieves an accuracy of _________________%.
- How does it work in general?

Architecture



1. **Base classifiers** $\hat{f}_j$'s are simple but possibly have weak preliminary predictions $\hat{y}_j$'s.
2. **Combined classifier** $\hat{f}$ uses the **combination rule** g to merge $\hat{y}_j$'s into a good final prediction $\hat{y}$.

Architecture for probabilistic classifiers



1. **Base classifiers** $\hat{f}_j$'s are simple but possibly have weak probability estimates $\hat{P}_{Y|\mathbf{X}}{}^{(j)}(\cdot\,|x)$.
2. **Combined classifier** $\hat{f}$ uses the **combination rule** g to merge $\hat{P}_{Y|\mathbf{X}}{}^{(j)}(\cdot\,|\boldsymbol{x})$'s into a good final prediction $\hat{P}_{Y|X}(\cdot\,|x)$.

How to get good performance?

- Reduce **risk** by avoiding *underfitting* and *overfitting*.
- For many loss functions L (0-1 loss, sum of squared error, …):

$$\overbrace{E[L(\mathrm{Y}, f_{\mathrm{W}}(\mathrm{X}))]}^{\text{Risk}} \leq \overbrace{E\left[L\left(\mathrm{Y}, \bar{f}(\mathrm{X})\right)\right]}^{\text{Bias}} + \overbrace{E\left[L\left(\bar{f}(\mathrm{X}), f_{\mathrm{W}}(\mathrm{X})\right)\right]}^{\text{Variance}}$$

  where
  - $\bar{f} := x \longmapsto E[f_{\mathrm{W}}(x)]$ is the **expected predictor;** (W is a random variable. Why?)
  - **Variance** is the dependence of $f_{\mathrm{W}}(X)$ on the data aka overfitting/underfitting; and
  - **Bias** is the deviation of $\bar{f}(\mathrm{X})$ from Y aka overfitting/underfitting.
- See Bias-variance trade-off.

Bias and variance for probabilistic classifiers

- For probabilistic classifiers,

$$\overbrace{E\left[L\left(P_{\mathrm{Y|X}}(\cdot\,|\mathrm{X}), P_{\hat{\mathrm{Y}}|\mathrm{X,W}}(\cdot\,|\mathrm{X,W})\right)\right]}^{\text{Risk}} \leq \overbrace{E\left[L\left(P_{\mathrm{Y|X}}(\cdot\,|\mathrm{X}), P_{\hat{\mathrm{Y}}|\mathrm{X}}(\cdot\,|\mathrm{X})\right)\right]}^{\text{Bias}} + \overbrace{I\left(\hat{\mathrm{Y}}; \mathrm{W}|\mathrm{X}\right)}^{\text{Variance}}$$

where

- $f_w(x) := P_{\hat{\mathrm{Y}}|\mathrm{X,W}}(\cdot\,|x, w)$ implies $\bar{f}(x) = E\left[P_{\hat{\mathrm{Y}}|\mathrm{X,W}}(\cdot\,|x, \mathrm{W})\right] = P_{\hat{\mathrm{Y}}|\mathrm{X}}(\cdot\,|x)$, called m____________;
- $P_{\mathrm{Y|X}}(\cdot\,|\mathrm{X})$ instead of Y is used as the ground truth;
- information (or Kullback-Leibler) divergence is used as the loss function

$$L(Q, P) := D_{\mathrm{KL}}(P\|Q) := \int_y (dP) \log \frac{dP}{dQ}; \quad \text{and}$$

- variance becomes the mutual information

$$\mathrm{E}\left[D_{\mathrm{KL}}\left(P_{\hat{\mathrm{Y}}|\mathrm{X,W}}(\cdot\,|\mathrm{X,W}) \| P_{\hat{\mathrm{Y}}|\mathrm{X}}(\cdot\,|\mathrm{X})\right)\right] = I\left(\hat{\mathrm{Y}}; \mathrm{W}|\mathrm{X}\right) \qquad \because I(\mathrm{X}; \mathrm{W}) = 0.$$

How to reduce variance and bias?

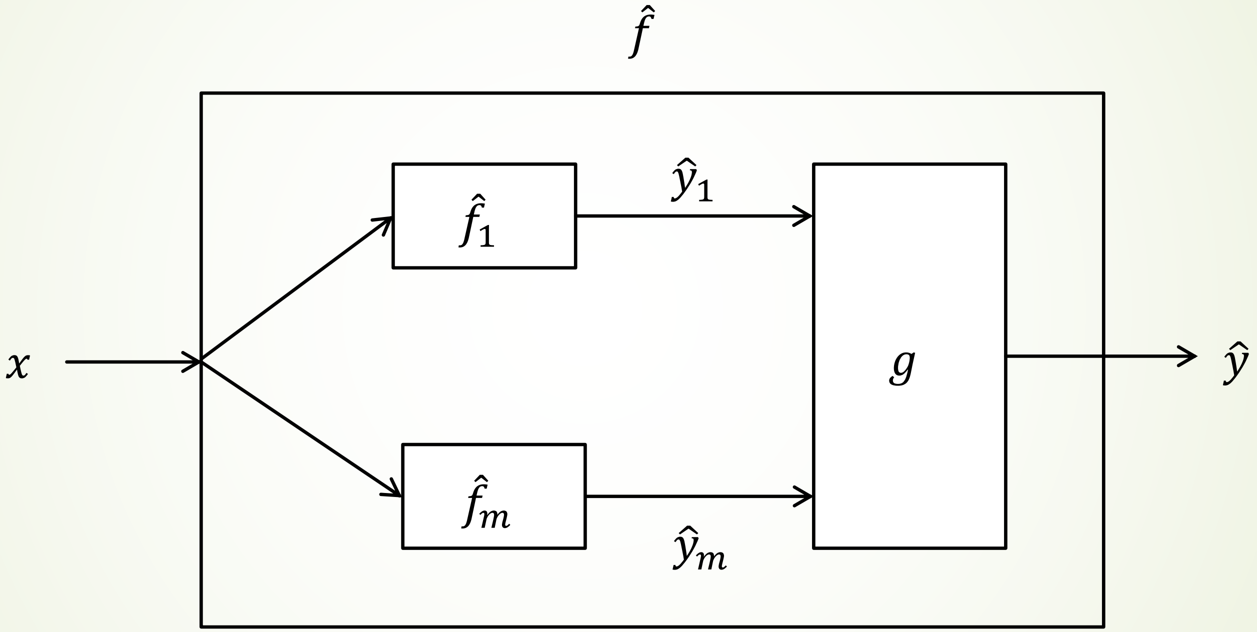
- Base classifiers should be **d________**, i.e., capture **as many different pieces of relevant information** as possible to reduce ______.
- The combination rule should reduce _________ by **smoothing out the noise** while **aggregating relevant information** into the final decision.

Bagging (**B**ootstrap **Agg**regation)
Base classifiers

$$\begin{tikzcd}
 & D_1 \arrow[r] & \hat{f}_1 \\
D \arrow[ur] \arrow[dr] & & \\
 & D_m \arrow[r] & \hat{f}_m
\end{tikzcd}$$

- Construct m bootstrap samples.
- Construct a base classifier for each bootstrap sample.

Bagging (**B**ootstrap **Agg**regation)
Majority voting



$$\hat{f}(x) := \arg\max_{\hat{y}} \overbrace{\sum_j \mathbb{1}\left(\hat{f}_j(x) = \hat{y}\right)}^{|\{j \mid \hat{f}_j(x) = \hat{y}\}| =}$$

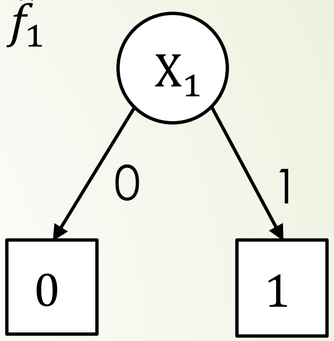
Example

	X_1	X_2	Y
1.	0	0	0
2.	0	1	0
3.	1	0	0
4.	1	1	1

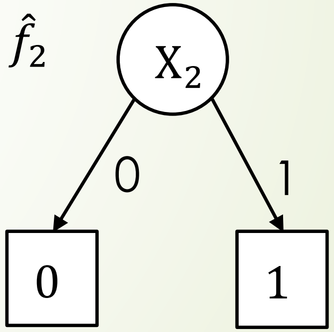
	X_1	X_2	Y
1.	0	0	0
2.	0	1	0
2.	0	1	0
4.	1	1	1

	X_1	X_2	Y
1.	0	0	0
3.	1	0	0
3.	1	0	0
4.	1	1	1

$\hat{f}_1$



$\hat{f}_2$



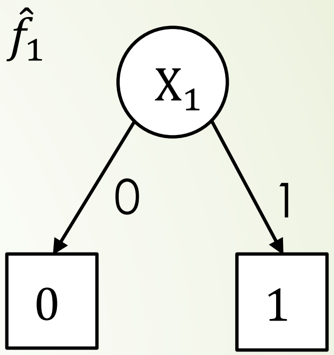
	X_1	X_2	Y	$\hat{f}_1$	$\hat{f}_2$	$\hat{f}$
1.	0	0	0	0	0	0
2.	0	1	0	0	1	?
3.	1	0	0	1	0	?
4.	1	1	1	1	1	1

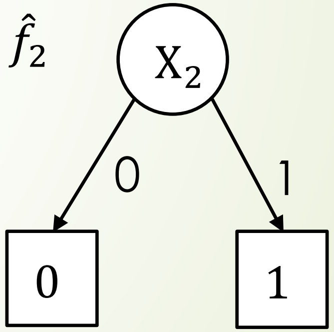
0

1

with equal probability

Accuracy = ____________%



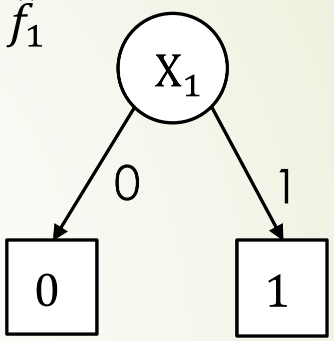


Is it always good to follow the majority?

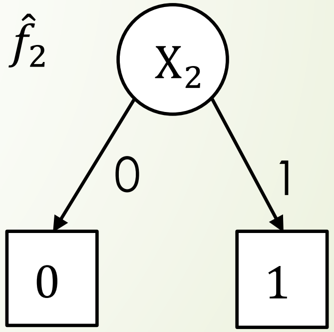
	X_1	X_2	Y	$\hat{f}_1$	$\hat{f}_2$	$\hat{f}$
1.	0	0	0	0	0	0
2.	0	1	0	0	1	?
3.	1	0	0	1	0	?
4.	1	1	1	1	1	1

0

$\hat{f}_1$



$\hat{f}_2$



Accuracy = ____________%

- It is beneficial to return 0 more often because __________________________.
- How to do this in general?

Sum rule and threshold moving

- $\hat{f}(x) = 1$ iff

$$\frac{1}{2}\left[\hat{f}_1(\boldsymbol{x}) + \hat{f}_2(\boldsymbol{x})\right] > \underline{\hspace{6em}}$$

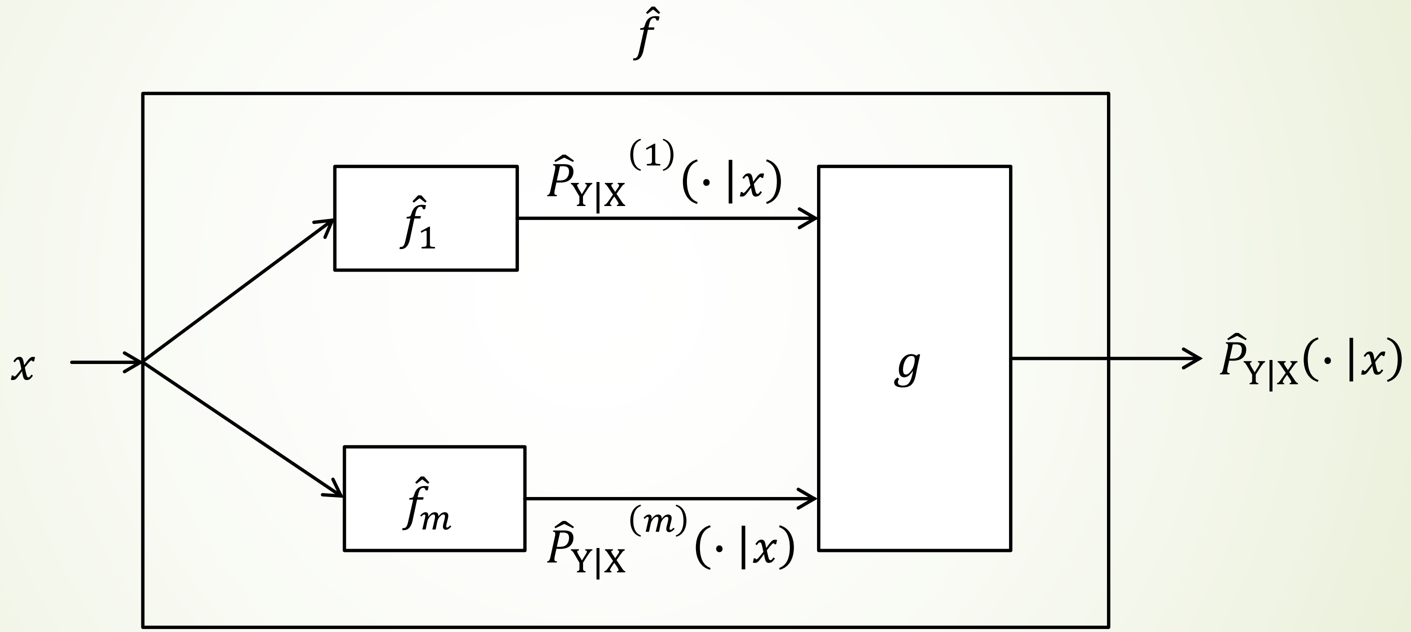
- **Binary classification:** Choose $\hat{f}(x) = 1$ iff

$$\frac{1}{m}\sum_t \hat{f}_t(\boldsymbol{x}) > \gamma$$

  for some chosen threshold γ.
- What about multi-class classification?

Bagging (**B**ootstrap **Ag**gregation)

Average of probabilities



$$\hat{f}(\boldsymbol{x}) := \frac{1}{m}\sum_t \hat{f}_t(\boldsymbol{x})$$

Other techniques to diversify base classifiers

- **Random forest**: Bagging with modified decision tree induction
  - **Forest-RI**: For each split, consider **random i**_________________ **s**________________ where only F randomly chosen features are considered.
  - **Forest-RC**: For each split, consider F **random l**____________________ **c**___________________ of L randomly chosen features.
- **Voting** (weka.classifier.meta.vote) and **Stacking** (weka.classifier.meta.stacking):
  - Use different classification algorithms.
- **Adaptive boosting (Adaboost)**:
  - Each base classifier tries to ____________________________ made by previous base classifiers.

Other techniques to combine decisions

- **Random forest**
  - Majority voting
  - Average of probabilities
- **Voting**
  - Majority voting or median
  - Average/product/minimum/maximum probabilities
- **Stacking**
  - Use a meta classifier.
- **Adaptive boosting (Adaboost)** - 2003 Gödel Prize winner
  - Weighted majority voting

What is Adaboost?

- An ensemble method that **learns from mistakes**:
  - Combined classifier:
    - Majority voting but with more weight on more accurate base classifier.

$$\hat{f}(x) \coloneqq \arg\max_{\hat{y}} \sum_t w_t \cdot \mathbb{1}\left(\hat{f}_t(x) = \hat{y}\right)$$

where $w_t \coloneqq \frac{1}{2} \ln \frac{1-\text{error}(\hat{f}_t)}{\text{error}(\hat{f}_t)}$ is the amount of say of $\hat{f}_t$ and $\text{error}\left(\hat{f}_t\right)$ is the error rate w.r.t. D_t. (See the precise formula below.)

  - Base classifiers:
    - Train $\hat{f}_t$ sequentially in t on D_t obtained by
    - Bagging $(x_i, y_i) \in D$ with

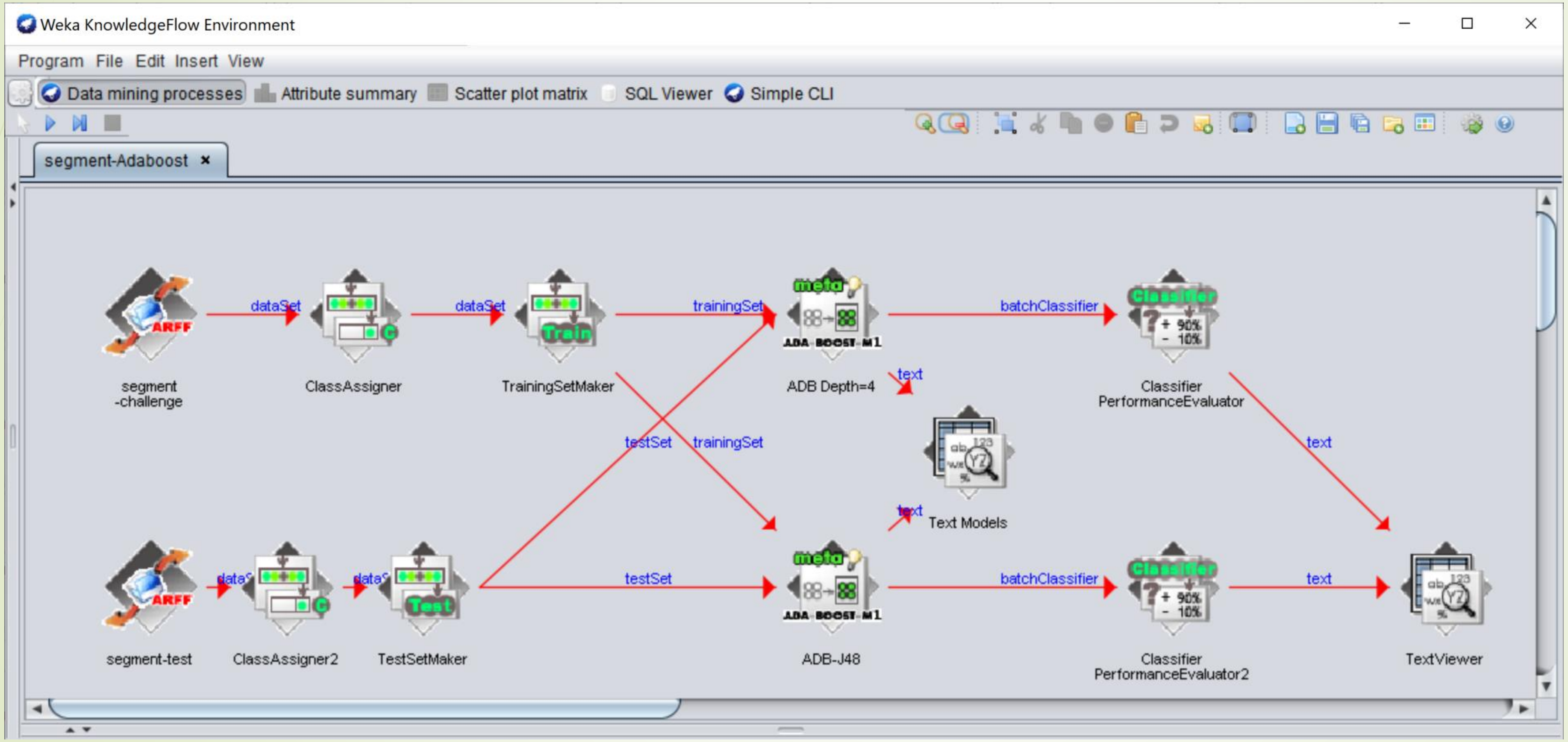
$$p_i^{(t)} \coloneqq \frac{p_i^{(t-1)}}{Z_t} \times \begin{cases} e^{w_{t-1}}, & \hat{f}_{t-1}(x_i) \neq y_i \text{ (incorrectly classified example)} \\ e^{-w_{t-1}}, & \text{otherwise (correctly classified example).} \end{cases}$$

starting with $p_i^{(1)} \coloneqq \frac{1}{|D|}$ and with $Z_t > 0$ chosen so that $\sum_i p_i^{(t)} = 1$.

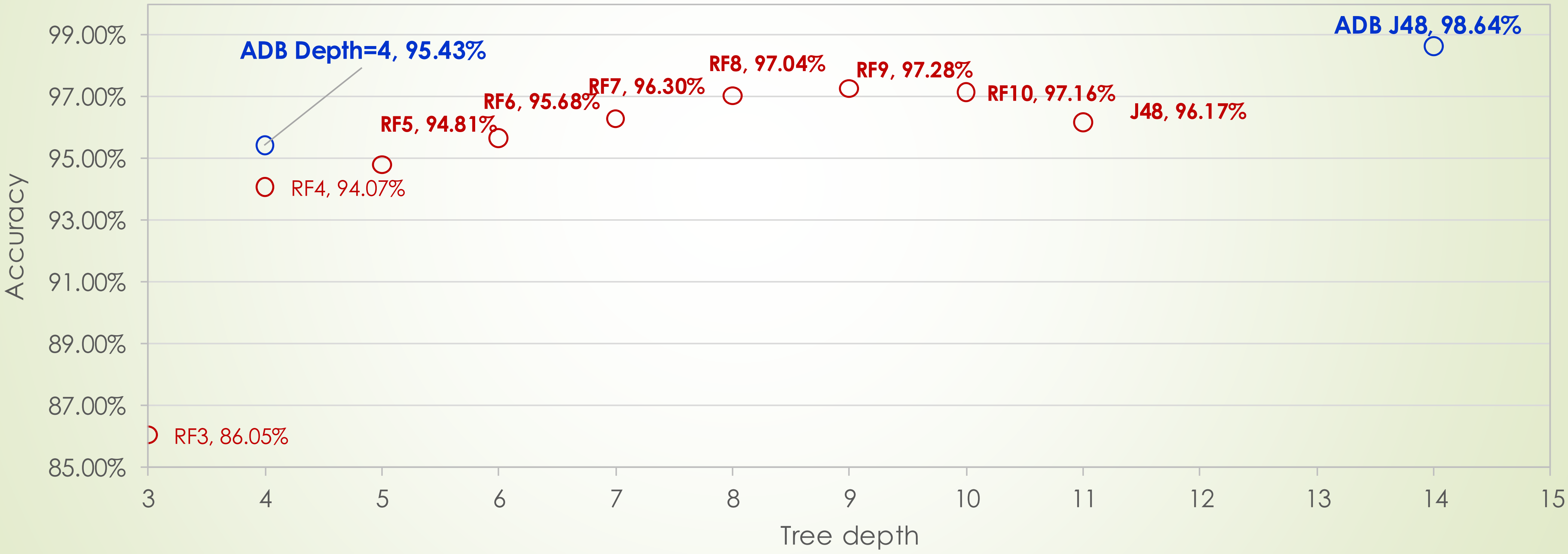
    - Compute the error rate

$$\text{error}\left(\hat{f}_t\right) \coloneqq \sum_i p_i^{(t)} \cdot \mathbb{1}\left(\hat{f}_t(x_i) \neq y_i\right)$$

Machine vs Machine



Machine vs Machine



References

- 8.6 Techniques to improve classification accuracy
- [Witten11] Chapter 8
- *Optional:*

  - Breiman, L. (1996). "Bagging predictors." *Machine learning*, *24*(2), 123-140.
  - Breiman, L. (2001). "Random forests." *Machine learning*, *45*(1), 5-32.
  - Freund Y, Schapire R, Abe N. "A short introduction to boosting." Journal-Japanese Society For Artificial Intelligence. 1999 Sep 1;14(771-780):1612.
  - Zhu, H. Zou, S. Rosset, T. Hastie, "Multi-class AdaBoost", 2009.