# Measuring Error

Adopted from slides by Alexander Ihler

# A Bayes Classifier

- Given training data, compute $p(\, y = c |\, x)$ and choose largest
- What's the (training) error rate of this method?

| Features | # bad | # good |
|---|---|---|
| X=0 | 42 | 15 |
| X=1 | 338 | 287 |
| X=2 | 3 | 5 |

# A Bayes classifier

- Given training data, compute $p(\, y = c |\, x)$ and choose largest
- What's the (training) error rate of this method?

| Features | # bad | # good |
|---|---|---|
| X=0 | 42 | 15 |
| X=1 | 338 | 287 |
| X=2 | 3 | 5 |

**Gets these examples wrong:**

Pr[ error ] = (15 + 287 + 3) / (690)

(empirically on training data: better to use test data)

# Measuring errors

- Confusion matrix
- Can extend to more classes

| | Predict 0 | Predict 1 |
|---|---|---|
| Y=0 | 380 | 3 |
| Y=1 | 302 | 5 |

# Classification Metrics: Precision, Recall, and F1

- In binary classification, we evaluate model performance using several metrics:
- - Precision: How many predicted positives are actually positive.
- - Recall: How many actual positives are correctly predicted.
- - F1-score: The balance between precision and recall.

## Confusion Matrix Components

Below are different components of a confusion matrix for a binary classification task with classes **Positive** and **Negative**.

| | | Predicted | | |
|---|---|---|---|---|
| | | **Positive** | **Negative** | **Total** |
| Actual | **Positive** | True positive (TP) | False negative (FN) (Type 2 error) | # positives |
| | **Negative** | False positive (FP) (Type 1 error) | True negative (TN) | # negatives |
| | **Total** | TP + FP | FN + TN | # examples |

## Confusion Matrix Example

| | | Predicted | | |
|---|---|---|---|---|
| | | **Positive** | **Negative** | **Total** |
| Actual | **Positive** | 80 | 40 | 120 |
| | **Negative** | 20 | 60 | 80 |
| | **Total** | 100 | 100 | 200 |

## Accuracy and Error

$accuracy = \frac{TP+TN}{TP+FP+TN+FN} = \frac{TP+TN}{\#examples}$

$error = \frac{FP+FN}{TP+FP+TN+FN} = \frac{FP+FN}{\#examples}$

**Examples**

$accuracy = \frac{80+60}{200} = \frac{140}{200} = 0.70$

$error = \frac{20+40}{200} = \frac{60}{200} = 0.30$

## Precision

$precision = \frac{TP}{TP+FP}$

**Example**

$precision = \frac{80}{100} = 0.80$

## Recall/TP rate/sensitivity

$recall = \frac{TP}{TP+FN} = \frac{TP}{\#positives}$

**Example**

$recall = \frac{80}{120} = 0.666$

## $F_1$ score

$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$

**Example**

$F_1 = 2 \times \frac{0.8 \times 0.666}{0.8 + 0.666} = 0.727$

## True Negative Rate (specificity)

$tnr = \frac{TN}{\#negatives}$

**Example**

$specificity = \frac{60}{80} = 0.75$

## False Positive Rate

$fpr = \frac{FP}{FP+TN} = \frac{FP}{\#negatives}$

**Example**

$fpr = \frac{20}{80} = 0.25$

## False Negative Rate

$fnr = \frac{FN}{FN+TP} = \frac{FN}{\#positives}$

**Example**

$fnr = \frac{40}{120} = 0.333$

# Example: Confusion Matrix

- Predicted Bad Predicted Good
- Actual Bad 380 3
- Actual Good 302 5

- From this, we derived:
- TP=380, FP=302, FN=3, TN=5.

# Precision

- Definition:
- Precision = TP / (TP + FP)

- Intuition:
- - Among all instances predicted as positive, how many are correct?
- - High precision = few false positives.

- Example:
- From our data: TP=380, FP=302
- Precision = 380 / (380 + 302) ≈ 0.557

# Recall

- Definition:
- Recall = TP / (TP + FN)

- Intuition:
- - Among all actual positives, how many are correctly predicted?
- - High recall = few false negatives.

- Example:
- From our data: TP=380, FN=3
- Recall = 380 / (380 + 3) ≈ 0.992

# F1-score

- Definition:
- F1 = 2 * (Precision * Recall) / (Precision + Recall)

- Why use F1:
- - Balances precision and recall.
- - Useful when dataset is imbalanced.

- Example:
- Precision ≈ 0.557, Recall ≈ 0.992
- F1 ≈ 2 * (0.557 * 0.992) / (0.557 + 0.992) ≈ 0.711

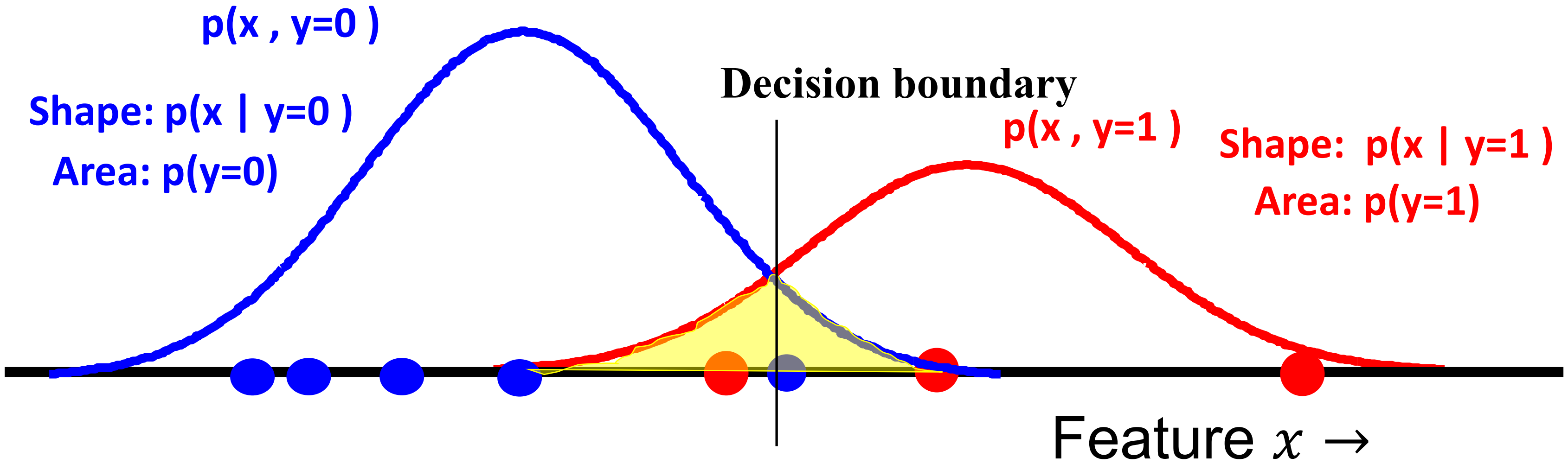
# A Bayes classifier

- Bayes classification decision rule compares probabilities:

$$p(y=0|x) \underset{>}{<} p(y=1|x)$$

$$= \; p(y=0,x) \underset{>}{<} p(y=1,x)$$

- Can visualize this nicely if x is a scalar:

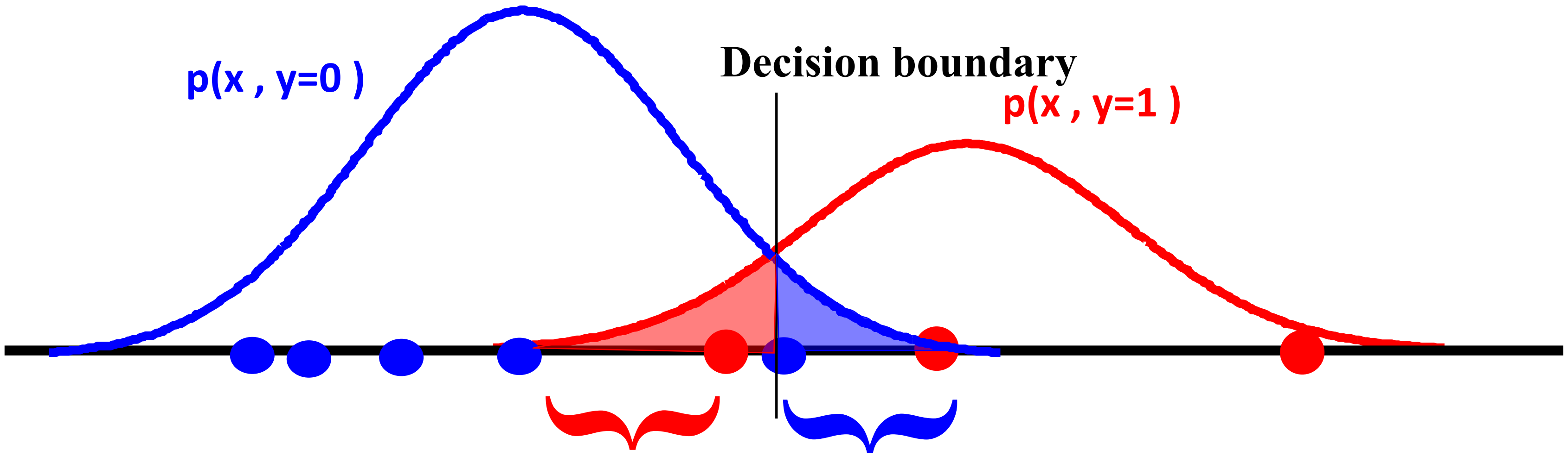


# A Bayes classifier

Add multiplier alpha:

$$\alpha \; p(y=0,x) \;\overset{<}{>}\; p(y=1,x)$$

- Not all errors are created equally…
- Risk associated with each outcome?




Type 2 errors: false negatives

Type 1 errors: false positives

False positive rate: (# y=0, ŷ=1) / (#y=0)

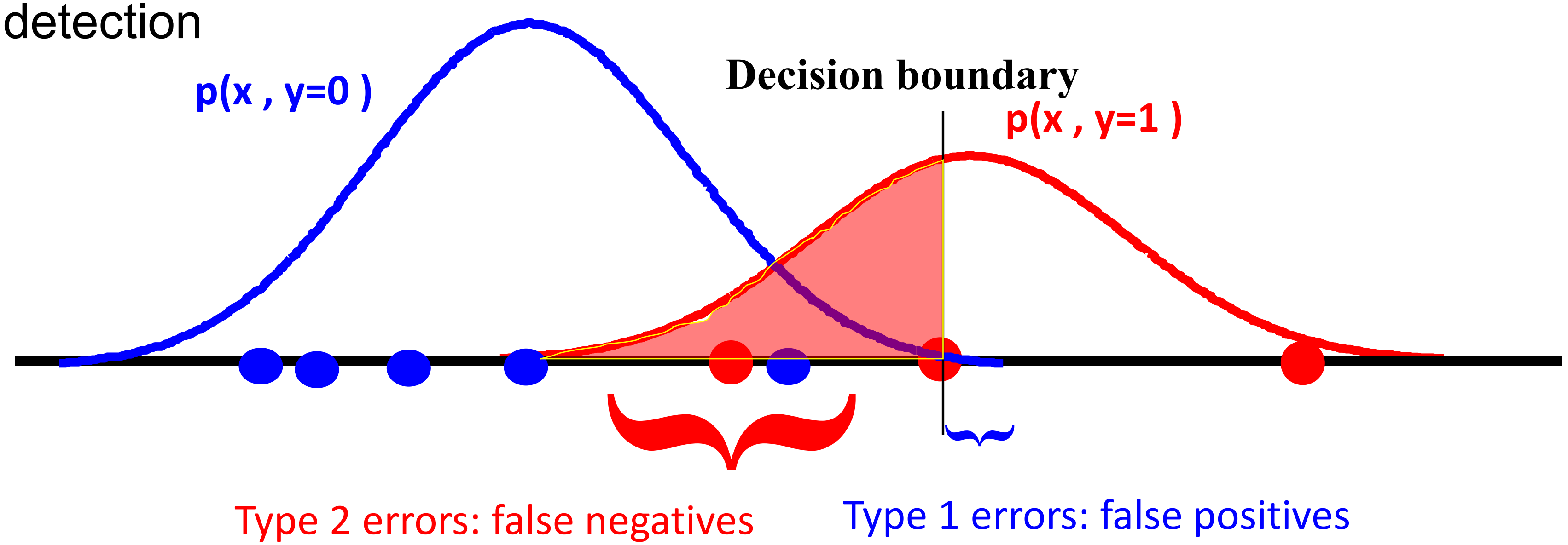
False negative rate: (# y=1, ŷ=0) / (#y=1)

# A Bayes classifier

Add multiplier alpha:

$$\alpha \; p(y=0,x) \;\underset{>}{<}\; p(y=1,x)$$

- Increase alpha: prefer class 0
- Spam detection




False positive rate: (# y=0, ŷ=1) / (#y=0)

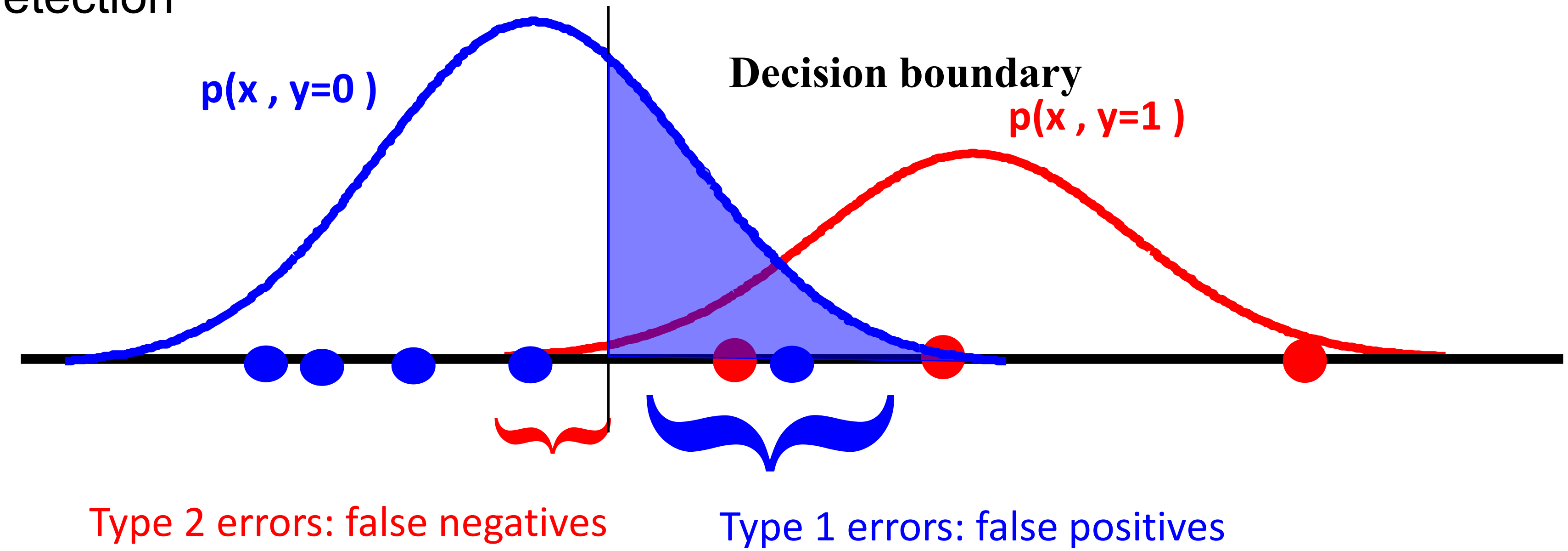
False negative rate: (# y=1, ŷ=0) / (#y=1)

# A Bayes classifier

Add multiplier alpha:

$$\alpha \ p(y=0,x) \ \begin{smallmatrix} < \\ > \end{smallmatrix} \ p(y=1,x)$$

- Decrease alpha: prefer class 1
- Cancer detection

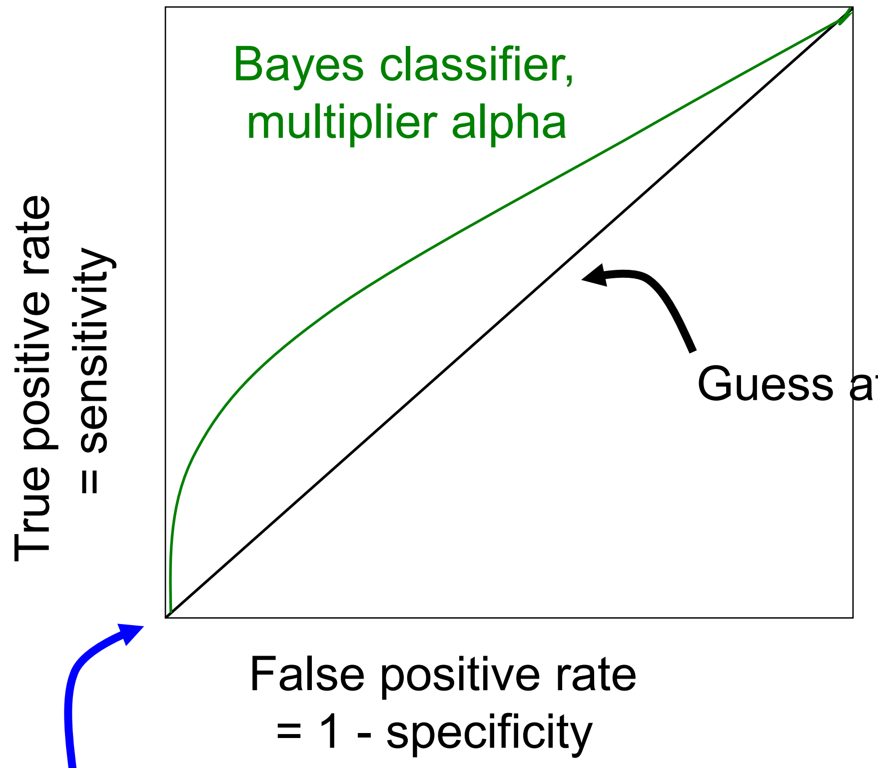



False positive rate: (# y=0, ŷ=1) / (#y=0)

False negative rate: (# y=1, ŷ=0) / (#y=1)

# ROC Curves

- Characterize performance as we vary the decision threshold?




Guess all 1

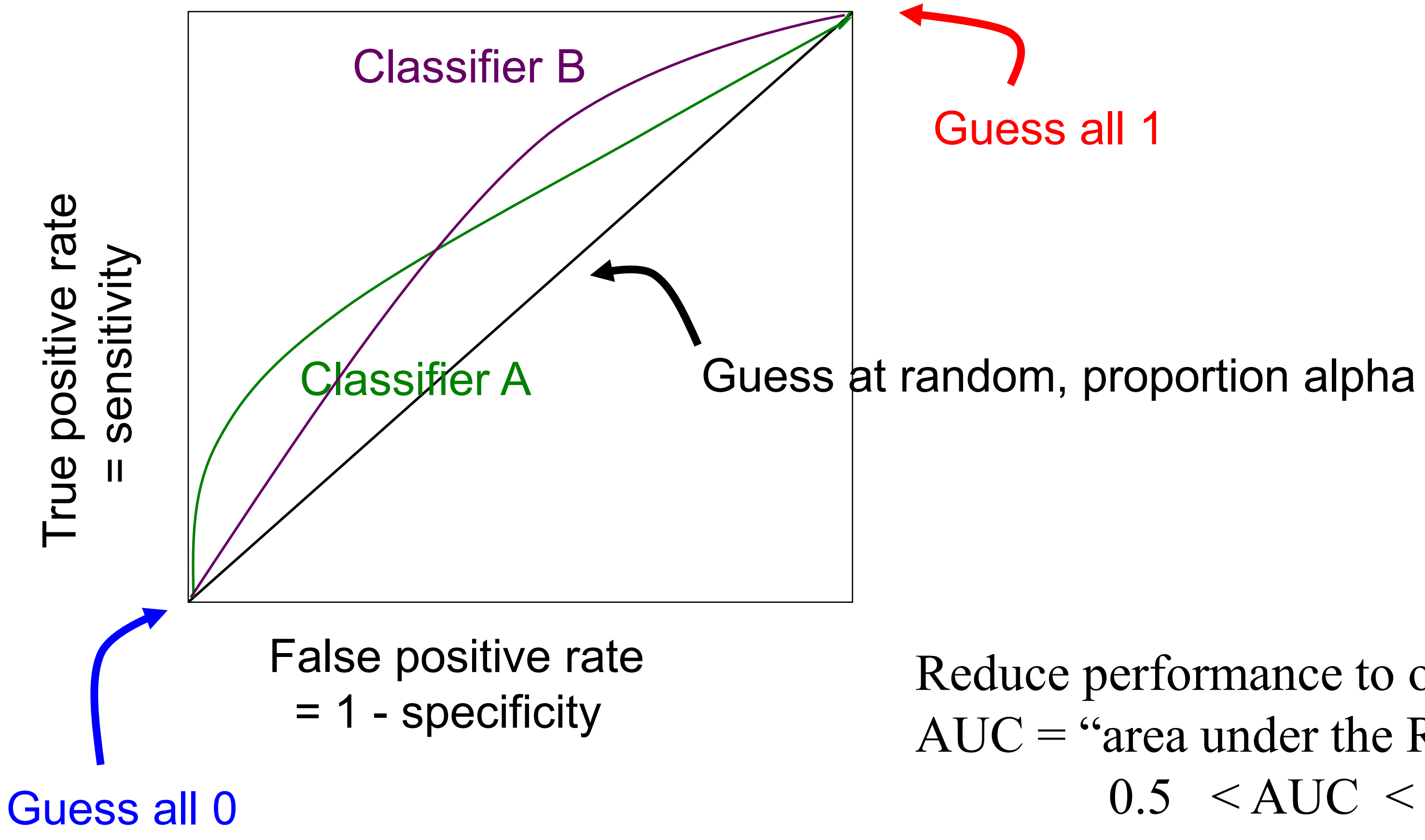
$$\alpha \ p(y=0,x) \ \underset{>}{<} \ p(y=1,x)$$

Guess at random, proportion alpha

Guess all 0

# ROC Curves

- Characterize performance as we vary the decision threshold?




Reduce performance to one number?

AUC = "area under the ROC curve"

$0.5 < \text{AUC} < 1$