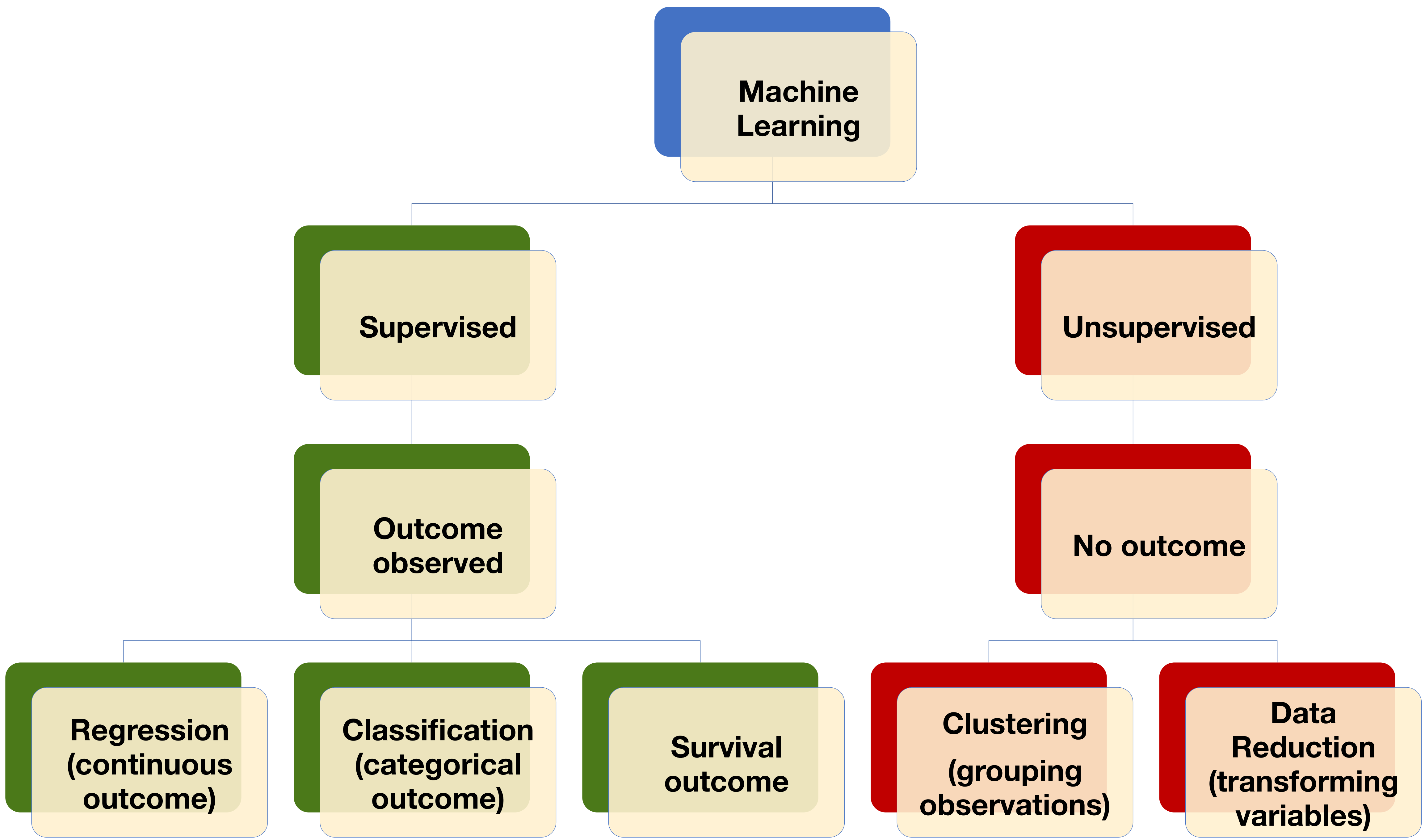
Machine Learning in R

Lecture 2

Jean Feng – Epidemiology and Biostatistics

Last lecture

1. What is Machine Learning?
2. Supervised vs Unsupervised Learning
3. Supervised learning:
    1. Notation
    2. Evaluating models using a loss function
    3. Parametric and non-parametric models
    4. Model selection
    5. Bias-Variance Trade-off
    6. Multivariate models


Machine Learning
Supervised
Unsupervised
Outcome observed
No outcome
Regression (continuous outcome)
Classification (categorical outcome)
Survival outcome
Clustering (grouping observations)
Data Reduction (transforming variables)

Today’s lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Notation
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. K-nearest neighbors
    6. Receiver Operating Characteristic (ROC)

Recall: Supervised Learning

- **Response**: what we want to predict, also known as the outcome, target, or dependent variable

$$Y$$

- **Predictors**: inputs, regressors, covariates, features, or variables

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$$

- **Dataset**:

$$\left\{(x_i, y_i) : i = 1,\ldots,n\right\}$$

- **Model**: A function that predicts Y at point $X = x$

Linear regression

The true linear model is defined as

$$\min_{\beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}} \boxed{\mathbb{E}} \left[\left(Y - \underbrace{(X^\top \beta + \beta_0)}_{\text{Prediction } f(X)} \right)^2 \right]$$

Ordinary least squares fits a model by solving the following optimization problem:

$$\min_{\beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}} \boxed{\frac{1}{n} \sum_{i=1}^{n}} \left(y_i - (x_i^\top \beta + \beta_0) \right)^2$$

Linear regression

If we define ℓ as the **squared error loss**, then

$$\min_{\beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (x_i^\top \beta + \beta_0) \right)^2$$

is equivalent to

$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell \left(y_i, f_\beta(x_i) \right).$$

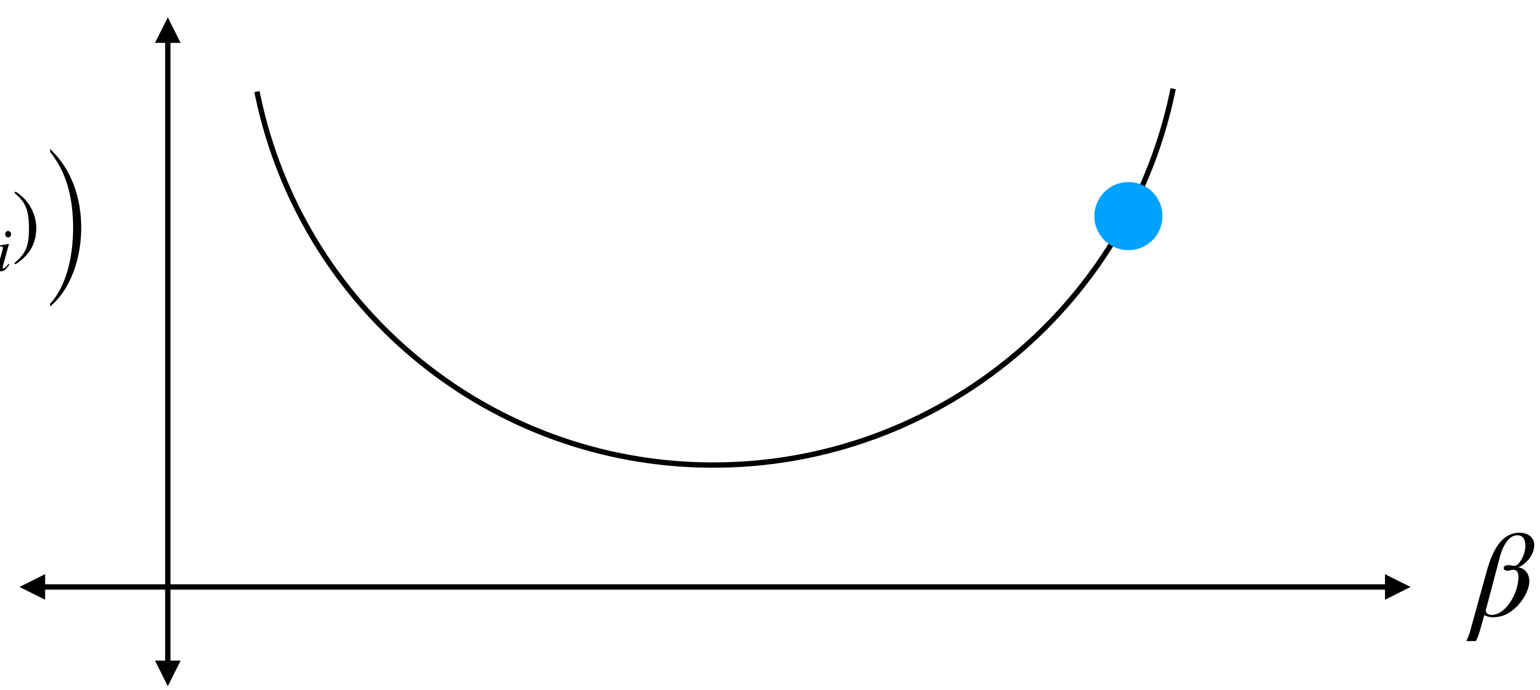
Many machine learning algorithms can be written in a similar way, where it is defined as the minimizer of the average loss.

Solving the optimization problem

- How do you find a β that solves the optimization problem
$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)?$$

- You could use a general optimization algorithm like gradient descent:

$$\frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$$

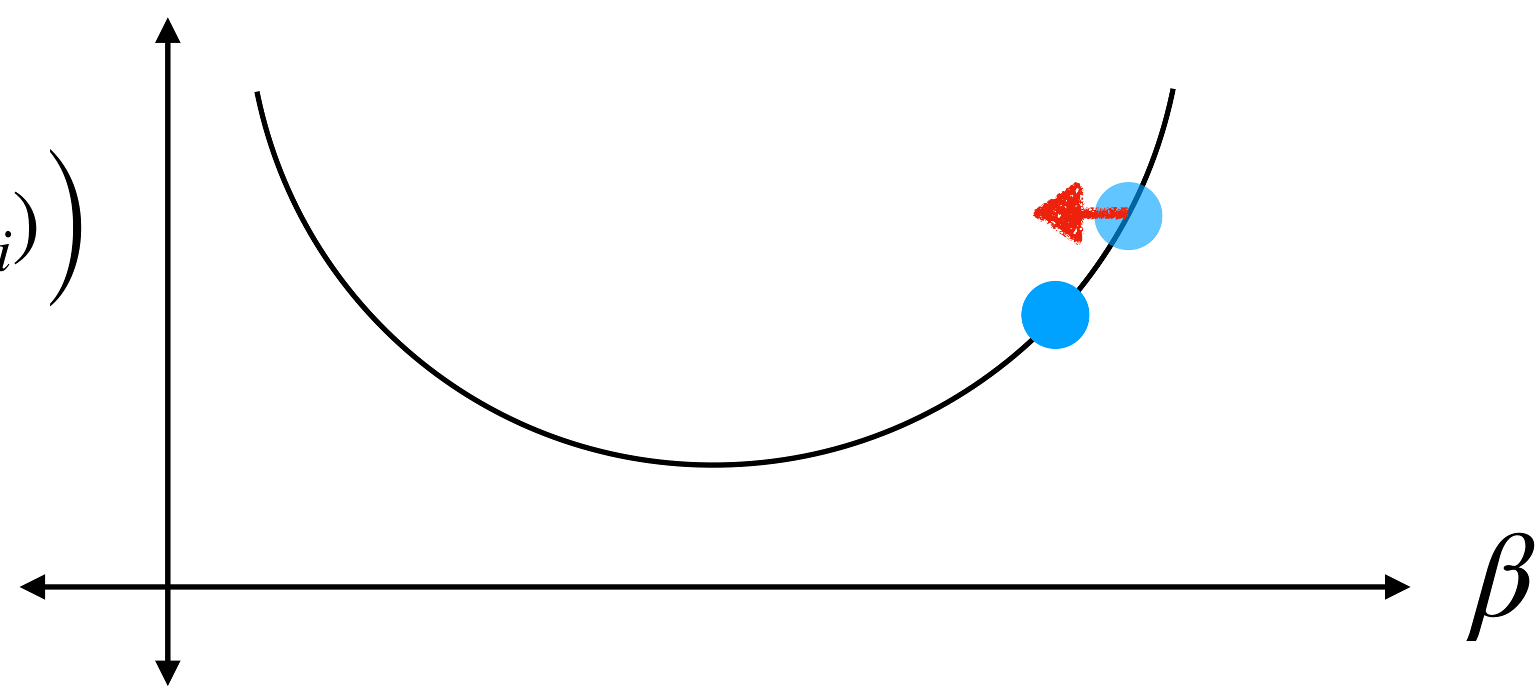


Solving the optimization problem

- How do you find a β that solves the optimization problem
$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)?$$

- You could use a general optimization algorithm like gradient descent:

$$\frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$$

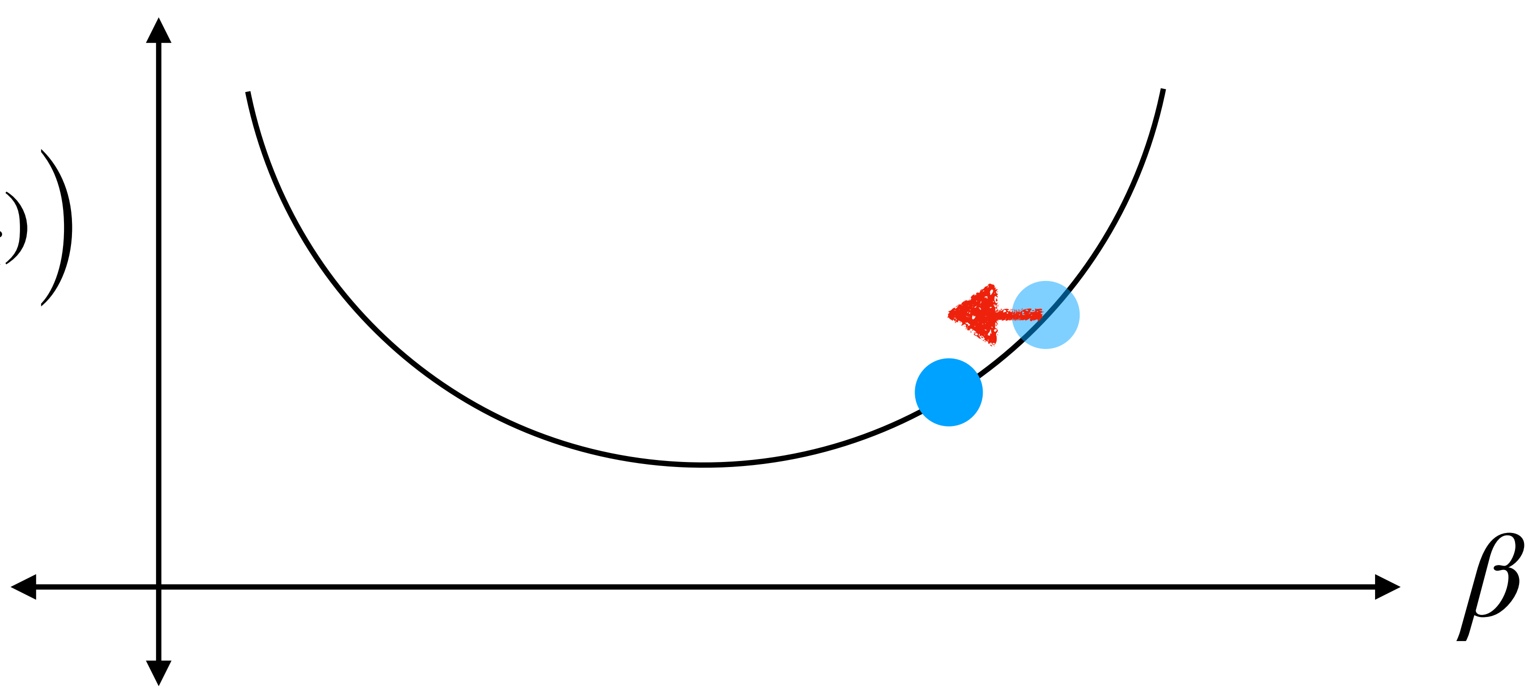


Solving the optimization problem

- How do you find a β that solves the optimization problem
$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)?$$

- You could use a general optimization algorithm like gradient descent:

$$\frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$$

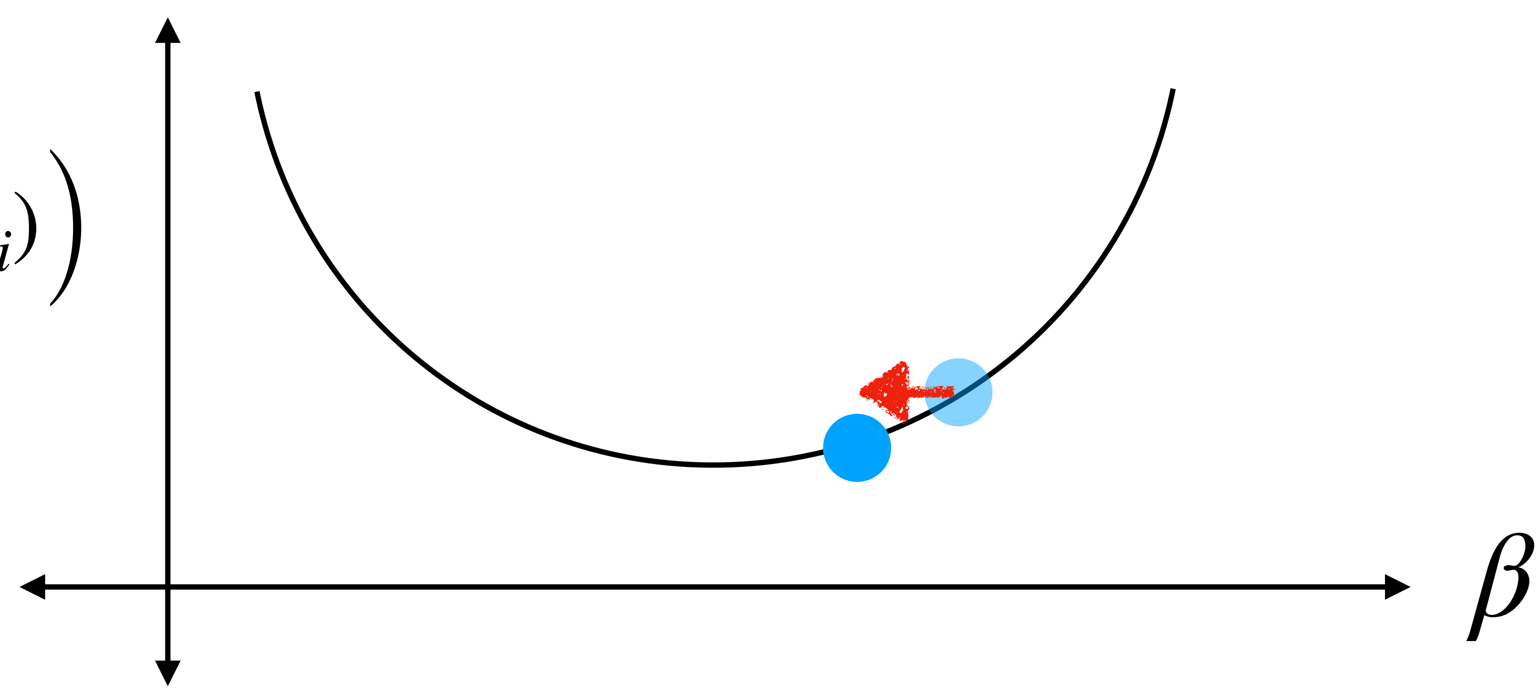


Solving the optimization problem

- How do you find a β that solves the optimization problem $\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$?

- You could use a general optimization algorithm like gradient descent:

$$\frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$$

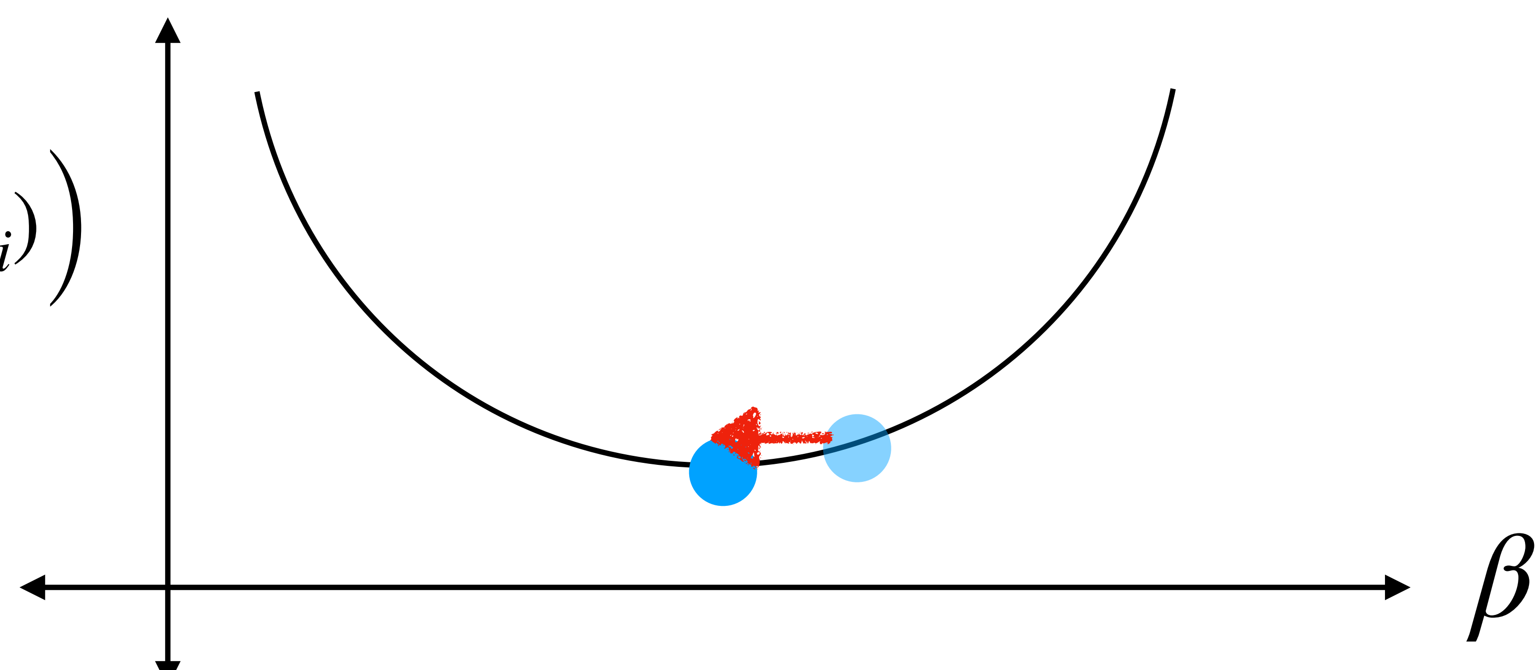


Solving the optimization problem

- How do you find a β that solves the optimization problem $\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$?

- You could use a general optimization algorithm like gradient descent:

$$\frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$$

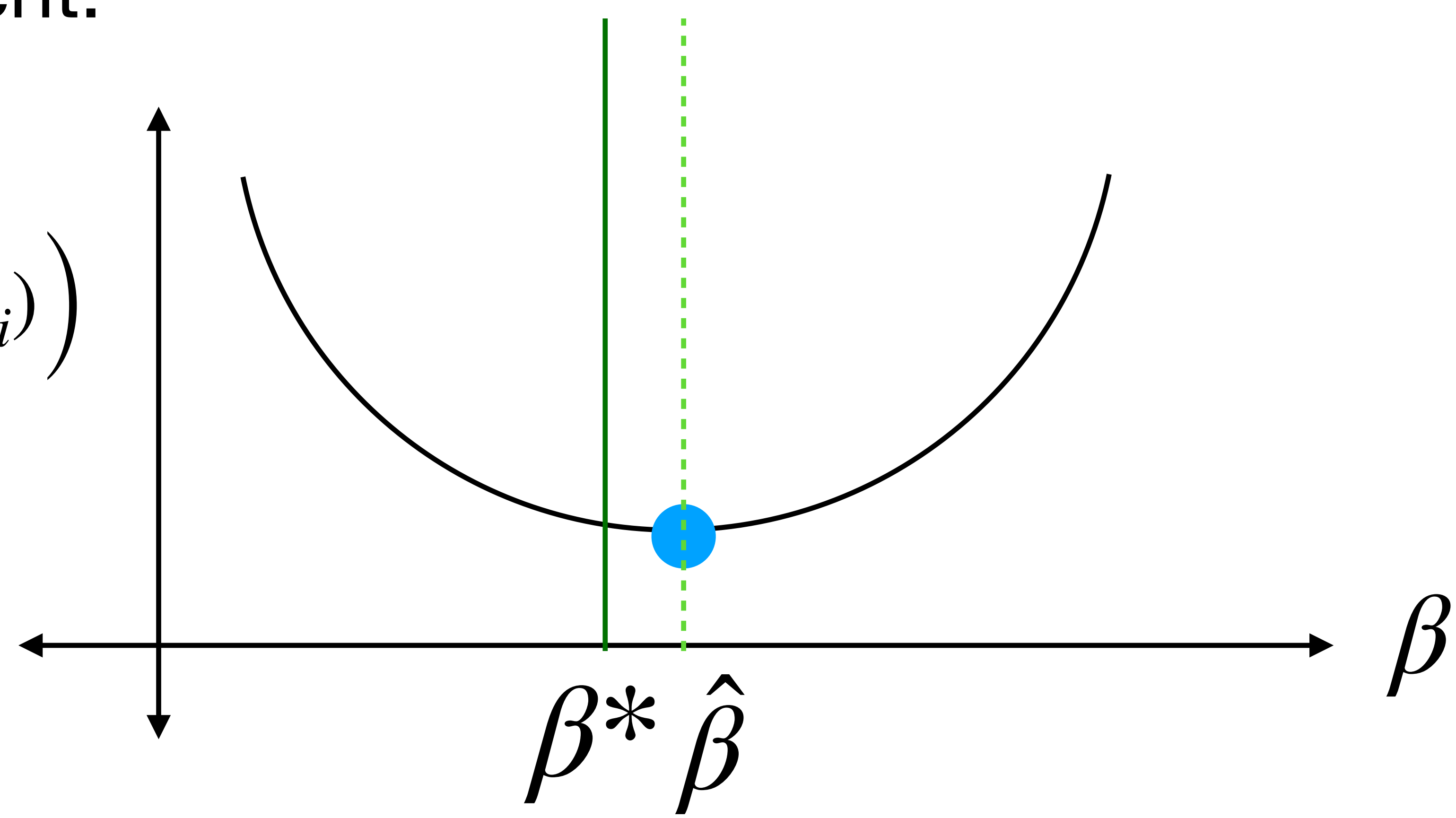


Solving the optimization problem

- How do you find a β that solves the optimization problem
$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell \left(y_i, f_\beta(x_i) \right)?$$

- You could use a general optimization algorithm like gradient descent:

$$\frac{1}{n} \sum_{i=1}^{n} \ell \left(y_i, f_\beta(x_i) \right)$$



Solving the optimization problem

- How do you find a β that solves the optimization problem $\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f_\beta(x_i)\right)$?

- You could use a general optimization algorithm like gradient descent.

Many machine learning algorithms can be written as the minimizer of the average loss and solved using general-purpose optimization algorithms.

Solving the optimization problem

- Okay, for linear regression, there is actually a closed-form solution. We won’t go over the equations here.

- Using a closed-form solution means you can fit a linear model very quickly, much faster than using a general optimization procedure.

Today's lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Notation
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. K-nearest neighbors
    6. Receiver Operating Characteristic (ROC)

Classification

- Supervised learning where the outcome is a class variable
- Unordered class variable such as:
  - `dx` $\in$ {`cancer`, `benign`}
  - `blood_type` $\in$ {`typeA`, `typeB`, `typeAB`, `typeO`}
- Given a set of predictors X and a set of possible classification outcomes $\mathscr{C}$, the objective is to find a function/algorithm $f(X)$ that takes X as input and predicts the outcome Y or the probability of each outcome.

Classification

- **Q**: Can we treat this as a regression problem?
- **A**: You probably don't want to do that for a number of reasons:
    - If you make this a regression problem, you'll need to order the outcome values. This might not make sense for things that are not naturally ordered.
    - Even if things are ordered, it can be hard to assign numerical values, because that will imply how far apart the outcomes are. For example, should we actually code cancer stage using the original numbers?
    - Regression models can predict values outside the range of possible values.

Classification loss functions

- Options:
  - **0-1 loss**: Is the predicted outcome not the same as the observed outcome? $\ell(y, \hat{y}) = 1\{\hat{y} \neq y\}$
  - **Negative log likelihood (cross-entropy)**: Is the observed outcome highly unlikely under the given probability model?
    $\ell(y, \hat{p}) = -y \log \hat{p} - (1-y) \log(1-\hat{p})$

Today’s lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Loss functions
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. k-Nearest Neighbors
    6. Receiver Operating Characteristics (ROC)

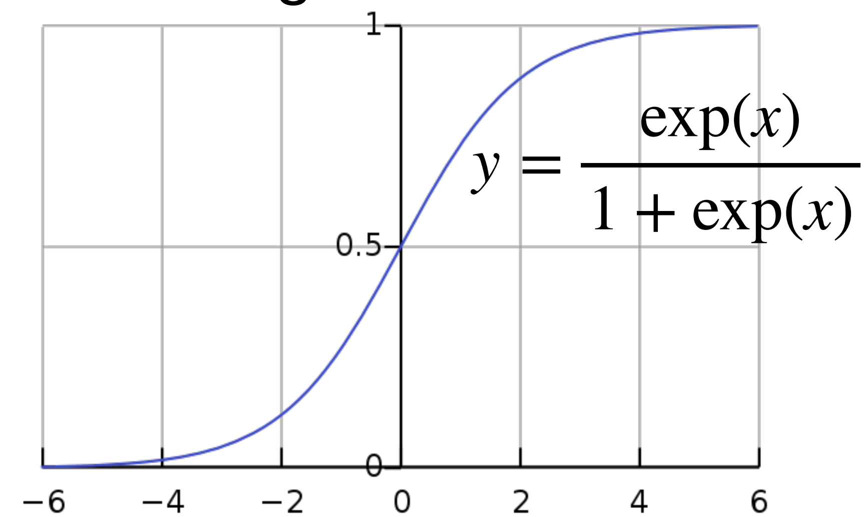
Logistic regression

- For binary classification settings where $Y \in \{0,1\}$

- Logistic regression assumes that the probability that $Y = 1$ for variable X has the form

$$\Pr(Y = 1 \mid X) = \underbrace{\frac{\exp(X^\top \beta + \beta_0)}{1 + \exp(X^\top \beta + \beta_0)}}_{\text{This value is always between 0 and 1 for any value of } \beta_0 \text{ and } \beta}.$$

Logistic function



- Recall that the parameters β can be interpreted as the log odds ratio.

Logistic regression as an optimization problem

To perform logistic regression, we do **maximum likelihood estimation,** which is equivalent to minimizing the negative log likelihood:

$$\min_{\beta_0\in\mathbb{R},\beta\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} \ell\left(y_i, f_{\beta_0,\beta}(x_i)\right)$$

Find a probabilistic model such that the observed data is highly probable

Negative log likelihood:
How unlikely is it to observe this data
given this probability model?

Predicted probability

where

$$\ell(y, f(x)) = -\,y\log f(x) - (1-y)\log(1-f(x))$$

and $f_{\beta_0,\beta}(x)$ is the probability of $Y = 1$ for input x:

$$f_{\beta_0,\beta}(x) = \frac{\exp(x^\top\beta + \beta_0)}{1+\exp(x^\top\beta+\beta_0)}.$$

Example: Predicting cause of death

- UCSF's POST-SCD study has autopsied every incident out-of-hospital sudden death within San Francisco.

- Can we identify biomarkers in the heart that predict whether a patient died of a sudden arrhythmic death versus trauma?

- $Y \in \{\underbrace{\text{trauma}}_{Y=0}, \underbrace{\text{sudden arrhythmic death}}_{Y=1}\}$

- $X =$ standardized heart weight

Can heart weight predict cause of death?

```r
logistic_m <- glm(y ~ x, family = "binomial")
summary(logistic_m)
```

```
Call:
glm(formula = y ~ x, family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9761  -0.9510  -0.5801   1.0430   1.9822

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.2626     0.2222  -1.182  0.23728
x             0.9688     0.2576   3.760  0.00017 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Confidence intervals

- We can construct confidence intervals to quantify our uncertainty about the log odds ratio:

```
> confint(logistic_m)
Waiting for profiling to be done...
                 2.5 %    97.5 %
(Intercept) -0.7043626 0.1715319
x            0.4993661 1.5173068
```

- Interpretation: “A one unit increase in the standardized heart weight is associated with an increase in the log odds by 0.97 (CI: 0.50, 1.51).”

Getting predictions from the model

- What are the probability that the cause of death was sudden arrhythmic death for standardized heart weight of 1?

```
> predict(logistic_m, newdata = data.frame(x=c(1)), type = "response")
        1
0.6682795
```

Today’s lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Loss functions
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. k-Nearest Neighbors
    6. Receiver Operating Characteristics (ROC)

Multinomial regression

- Sometimes the outcome can belong to one of K classes. Instead of doing logistic regression, we can do multinomial regression.

- The model is indexed by parameters $(\beta_{0,k}, \beta_k)$ for $k = 1,\ldots,K$ the form. The probability of the kth class for input x is defined as

$$\Pr(Y = k \mid X = x) = \frac{\exp(x^\top \beta_k + \beta_{0,k})}{\sum_{l=1}^{K} \exp(x^\top \beta_l + \beta_{0,l})}.$$

(also known as the soft-max function)

(value between 0 and 1)

- To perform multinomial regression, we also perform maximum likelihood estimation, where we replace the logistic model with the multinomial model.

Little quiz!

- **Q**: How do logistic and multinomial regression try to model the data?

- **A**: They try to model Y as a function of X

- **B**: They try to model X as a function of Y

Today’s lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Loss functions
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. k-Nearest Neighbors
    6. Receiver Operating Characteristics (ROC)

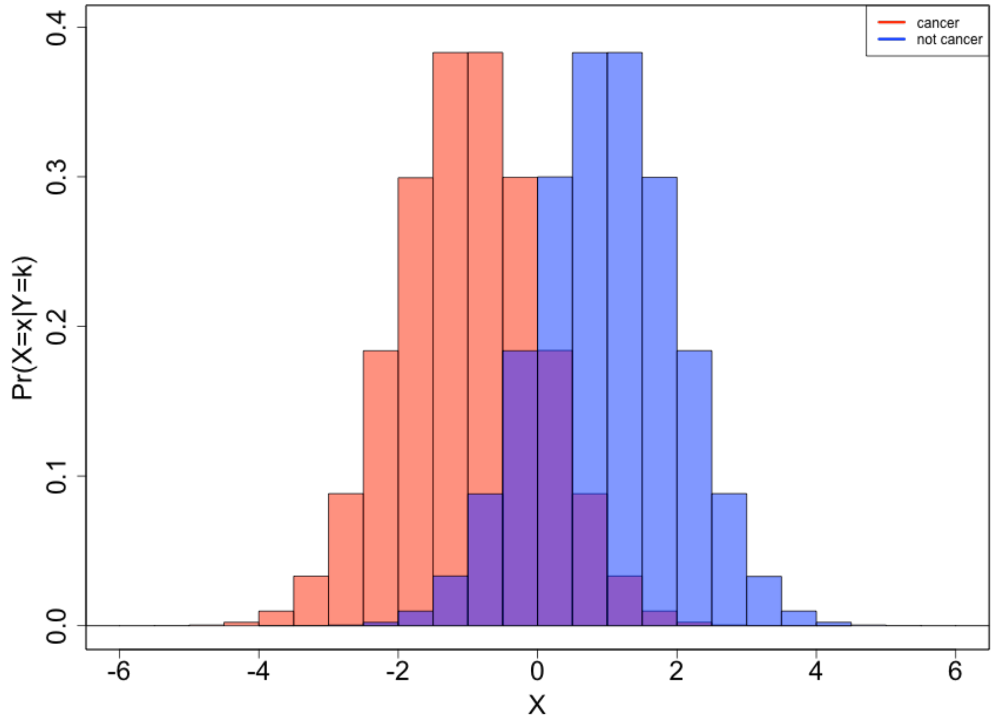
Modeling X|Y instead of Y|X

- Up to now, we've been modeling $\Pr(Y = y \mid X = x)$.
- How about switching things around and modeling $\Pr(X = x \mid Y = y)$?
- **Q**: When do we want to model X|Y vs Y|X?

Modeling X|Y instead of Y|X

- **A**: Possible reasons:

1. You don't have a lot of data, but you have a good understanding of X|Y. For example, you know that X is normally distributed within each category.



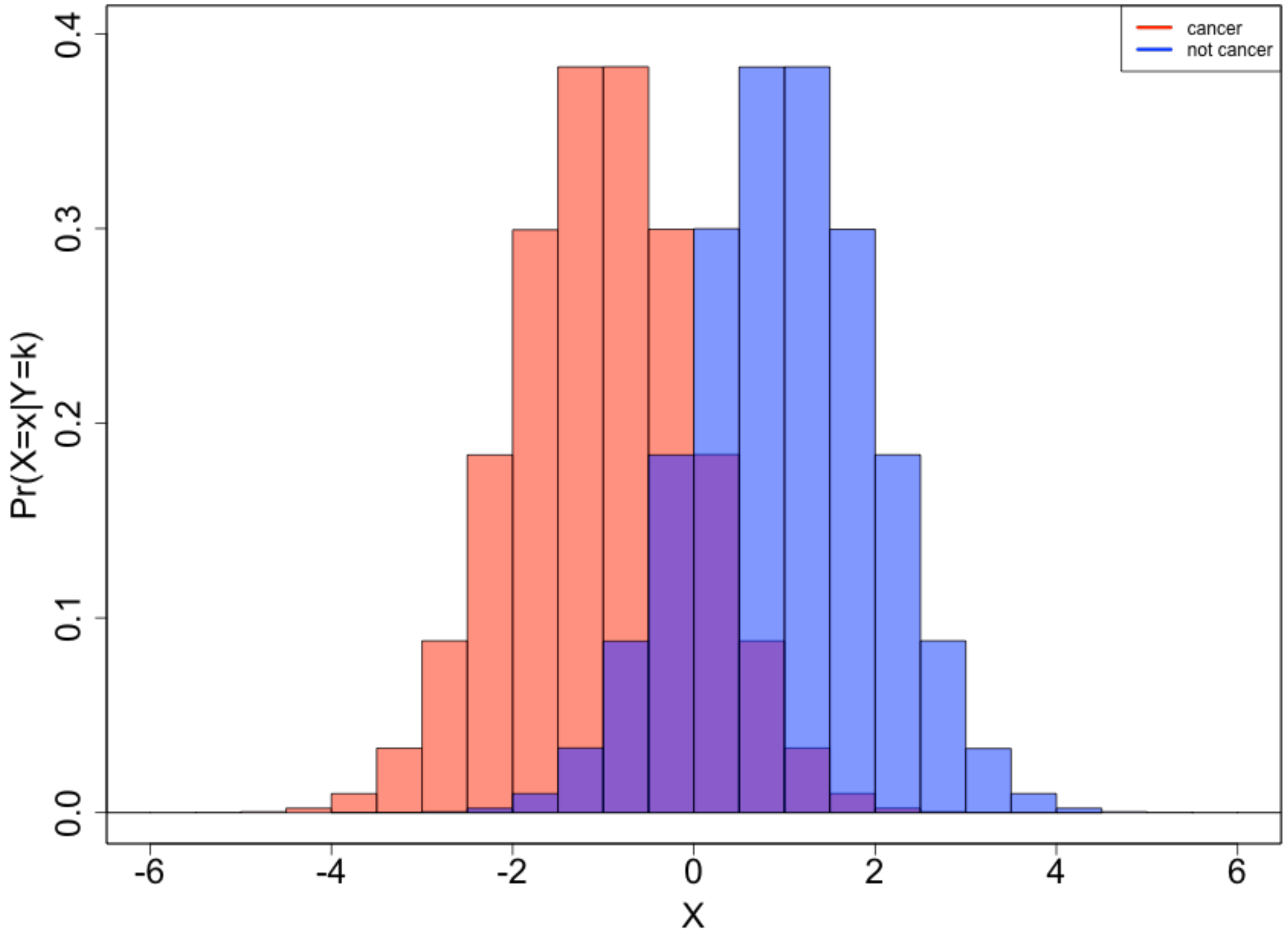
2. You want to generate examples of X|Y. For example, faces of men versus women.

Linear discriminant analysis (LDA)

- The primary motivation for using LDA is that you already have a model in mind for $\Pr(X = x \mid Y = y)$.

(Highly) Parametric model!

- In LDA, $X \mid Y$ is assumed to be a normal distribution:

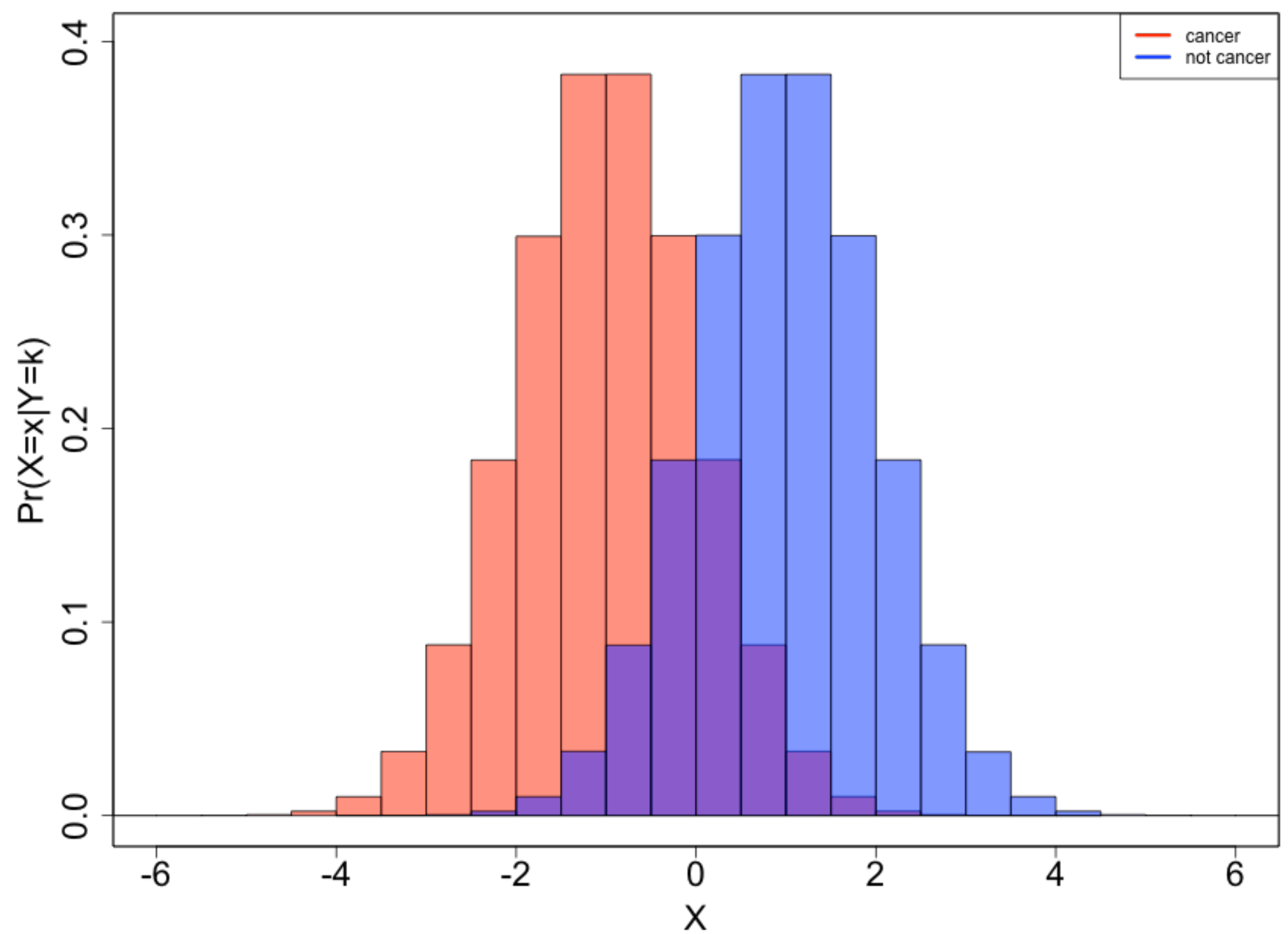


Linear discriminant analysis

To model $\Pr(X = x \mid Y = y)$, LDA estimates:

- The mean of X within each class

- The variance of X, which is usually assumed to be the same across all classes.



- Also, we model the probability of each class $\Pr(Y = y)$

Bayes' Theorem

- How do you predict Y given X if you only have a model of X given Y?

- Bayes' theorem states that

$$\Pr(Y = y \mid X = x) = \frac{\Pr(X = x, Y = y)}{\Pr(X = x)}$$

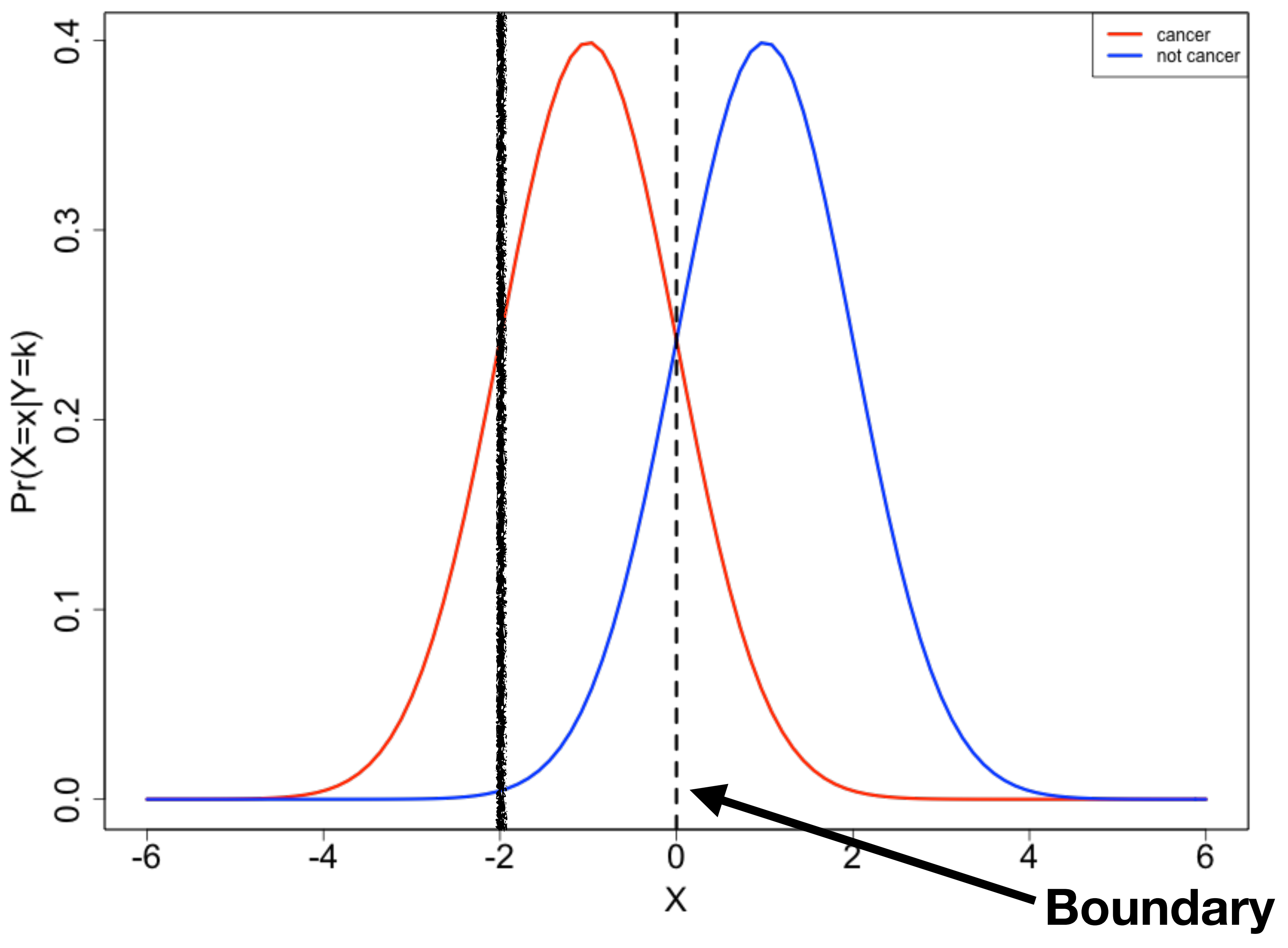
$$\propto \Pr(X = x, Y = y)$$

$$\propto \underbrace{\Pr(X = x \mid Y = y)}_{\text{Model of X|Y}} \, \underbrace{\Pr(Y = y)}_{\text{Model for the prevalence of Y}}$$

Linear discriminant analysis

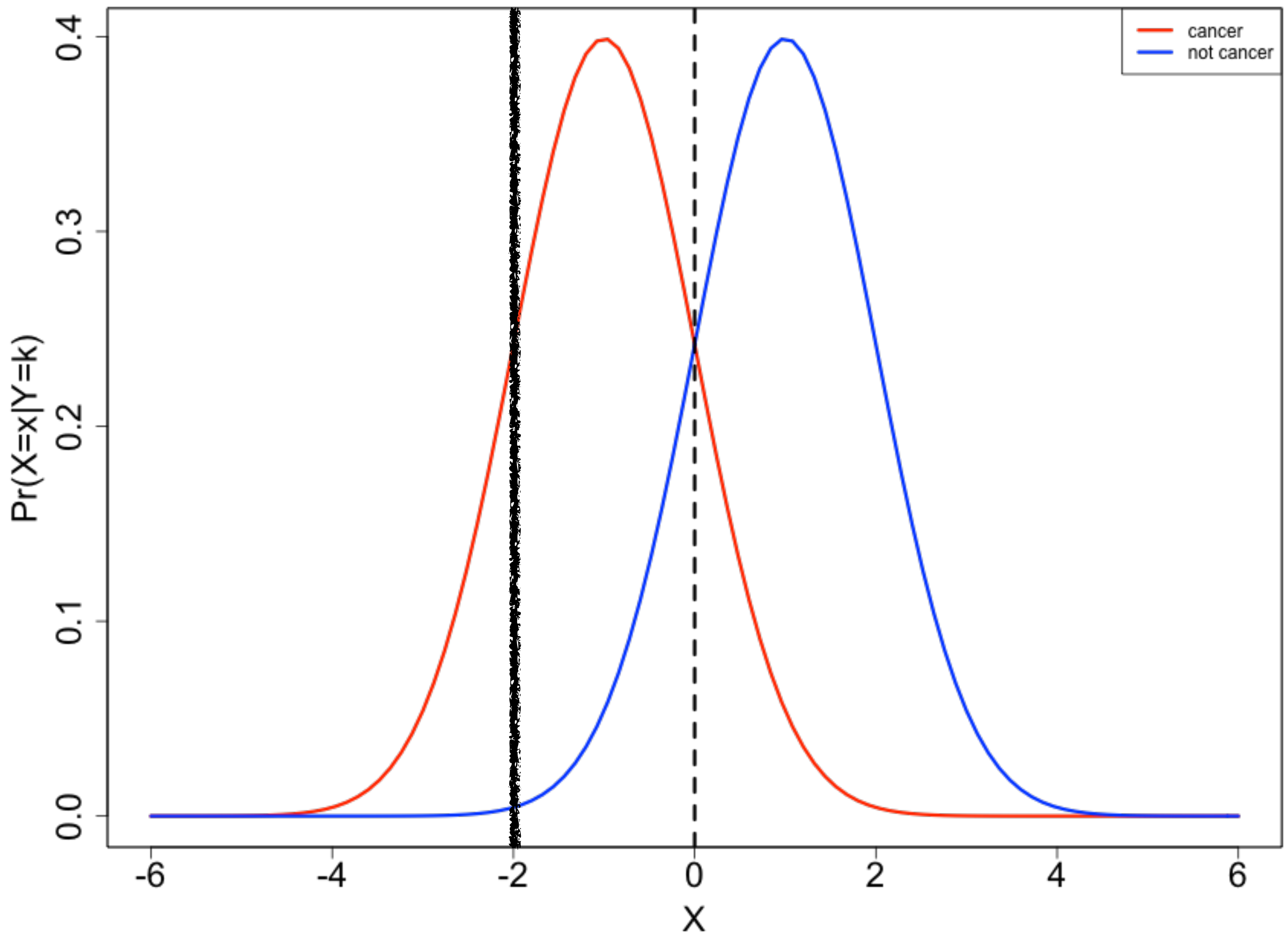
- If $\Pr(Y=1) = \Pr(Y=0)$, which class is more likely for a subject with $X = -2$?

Cancer is more likely.



Linear discriminant analysis

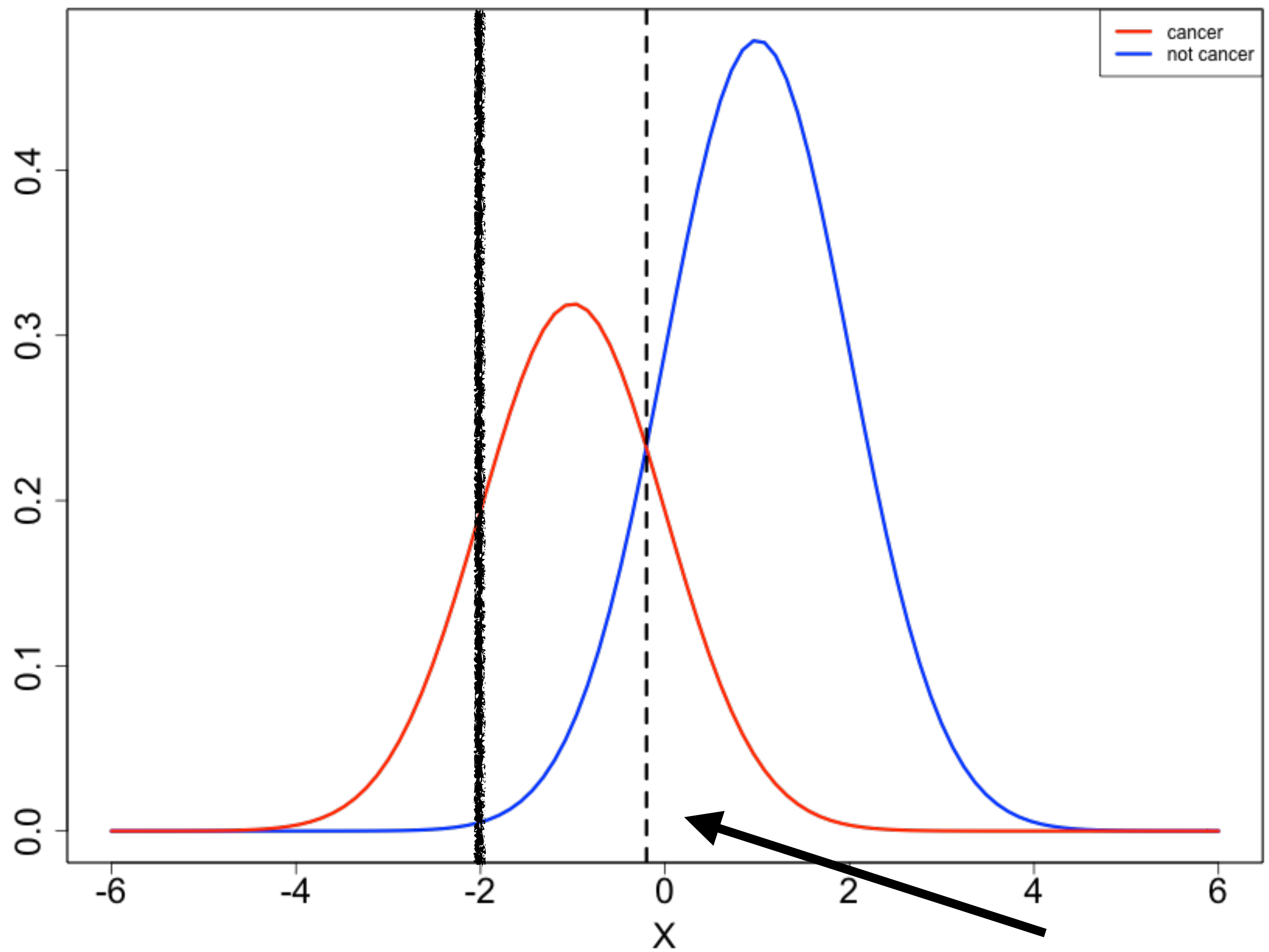
- If $\Pr(Y=0)=0.2$ and $\Pr(Y=1)=0.8$, which class is more likely for a subject with $X=-2$?



We need the prevalence information as well.

Linear discriminant analysis

- If $\Pr(Y = 0) = 0.2$ and $\Pr(Y = 1) = 0.8$, which class is more likely for a subject with $X = -2$?



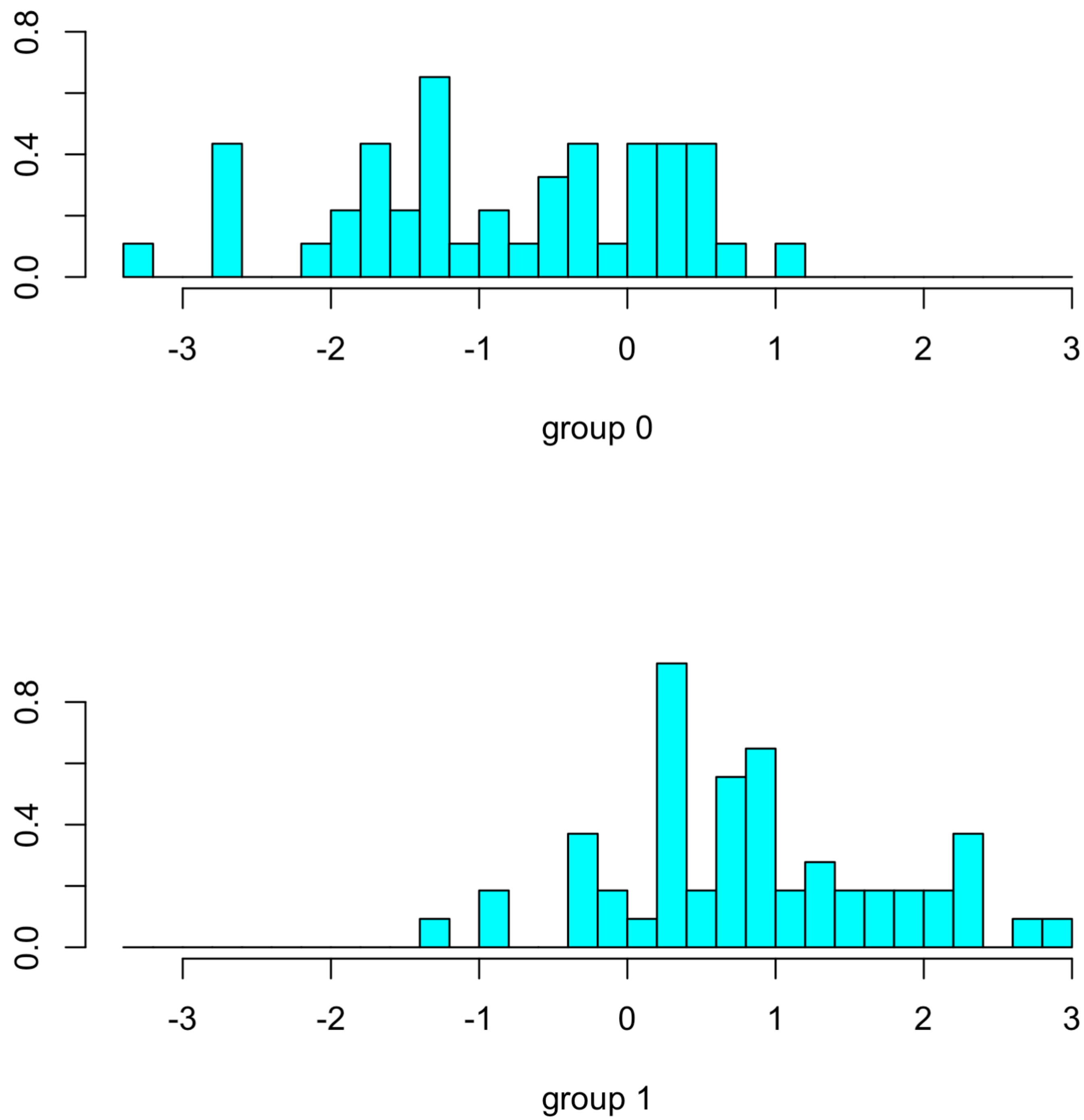
Notice that the boundary shifted a bit!

Heart weight example, continued.

```
> lda_m <- lda(y ~ x)
> lda_m
Call:
lda(y ~ x)

Prior probabilities of groups:
   0    1
0.46 0.54

Group means:
           x
0 -0.7883936
1  0.5240000

Coefficients of linear discriminants:
       LD1
x 1.279236
```

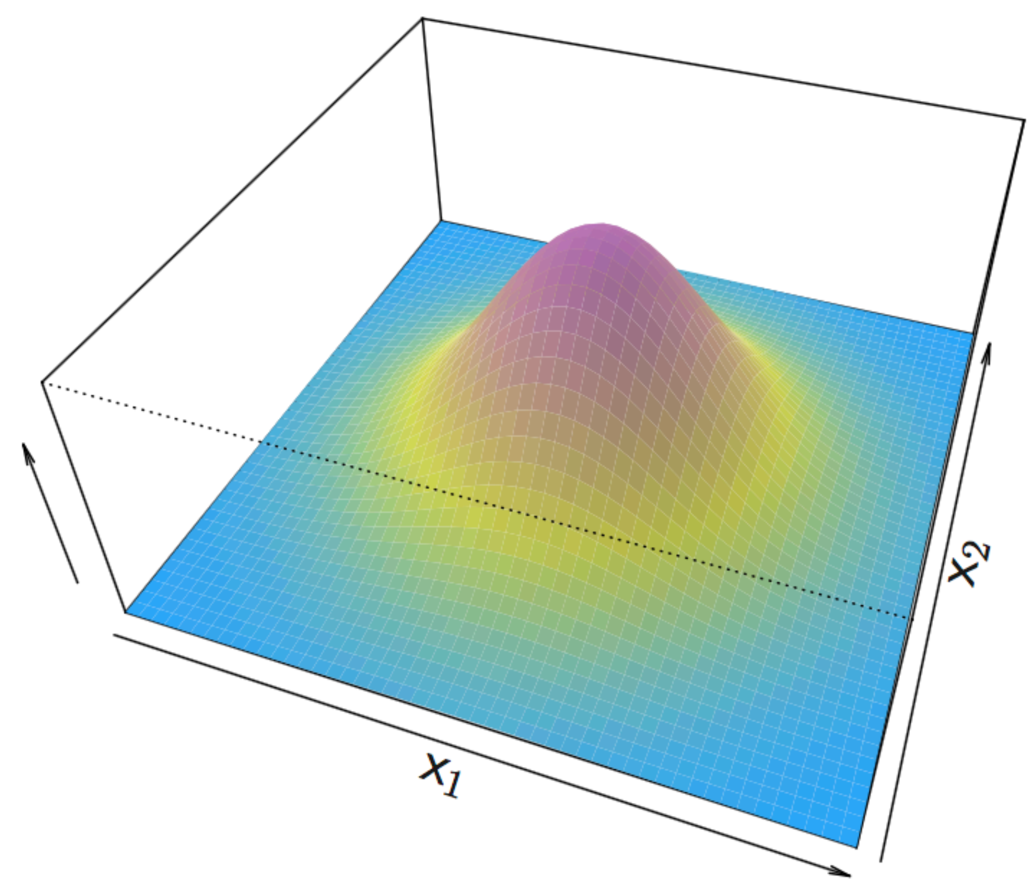
Plotting LDA

```
> plot(lda_m)
```



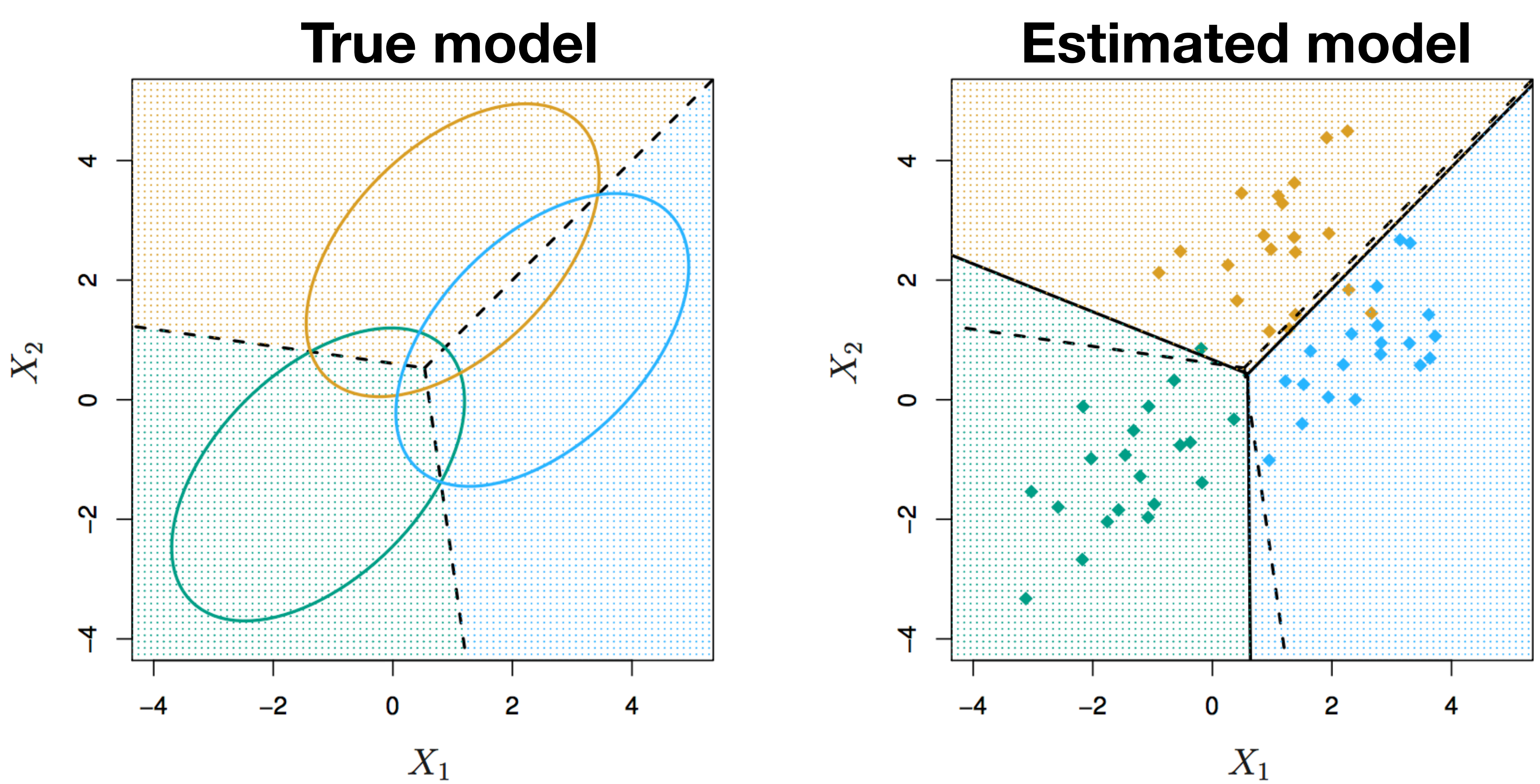
Multiple variables in LDA

- When there are multiple predictors, LDA models it using a multivariate Gaussian distribution.
- Note that it again assumes that the shape of the Gaussian distributions are the same across all classes.



Multiple classes in LDA

- When there are multiple classes, LDA fits a Gaussian distribution for each class and separates the classes according to a “Bayes decision boundary”. The boundaries are linear, hence the name of the method.



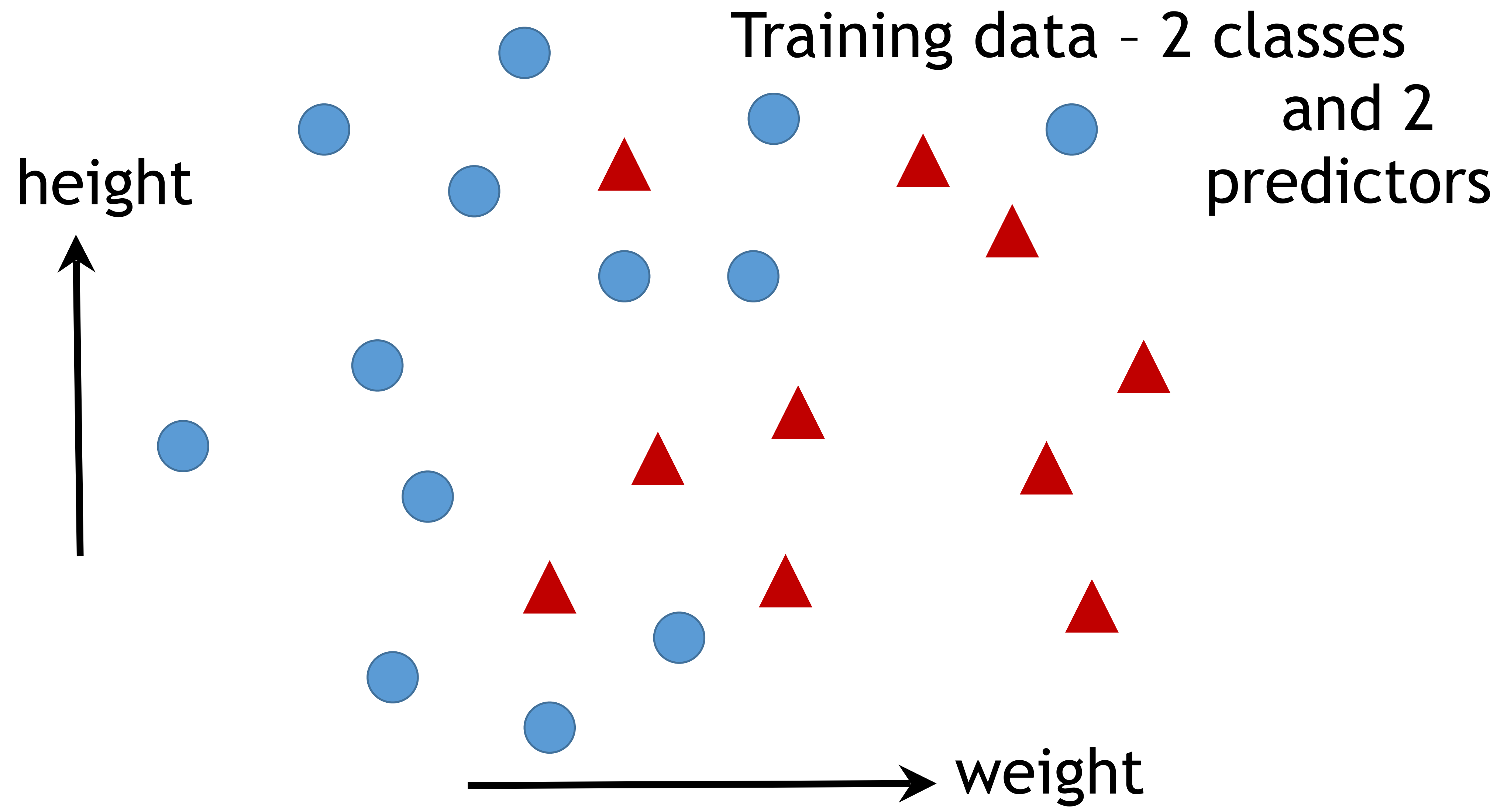
Today's lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Loss functions
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. k-Nearest Neighbors
    6. Receiver Operating Characteristics (ROC)

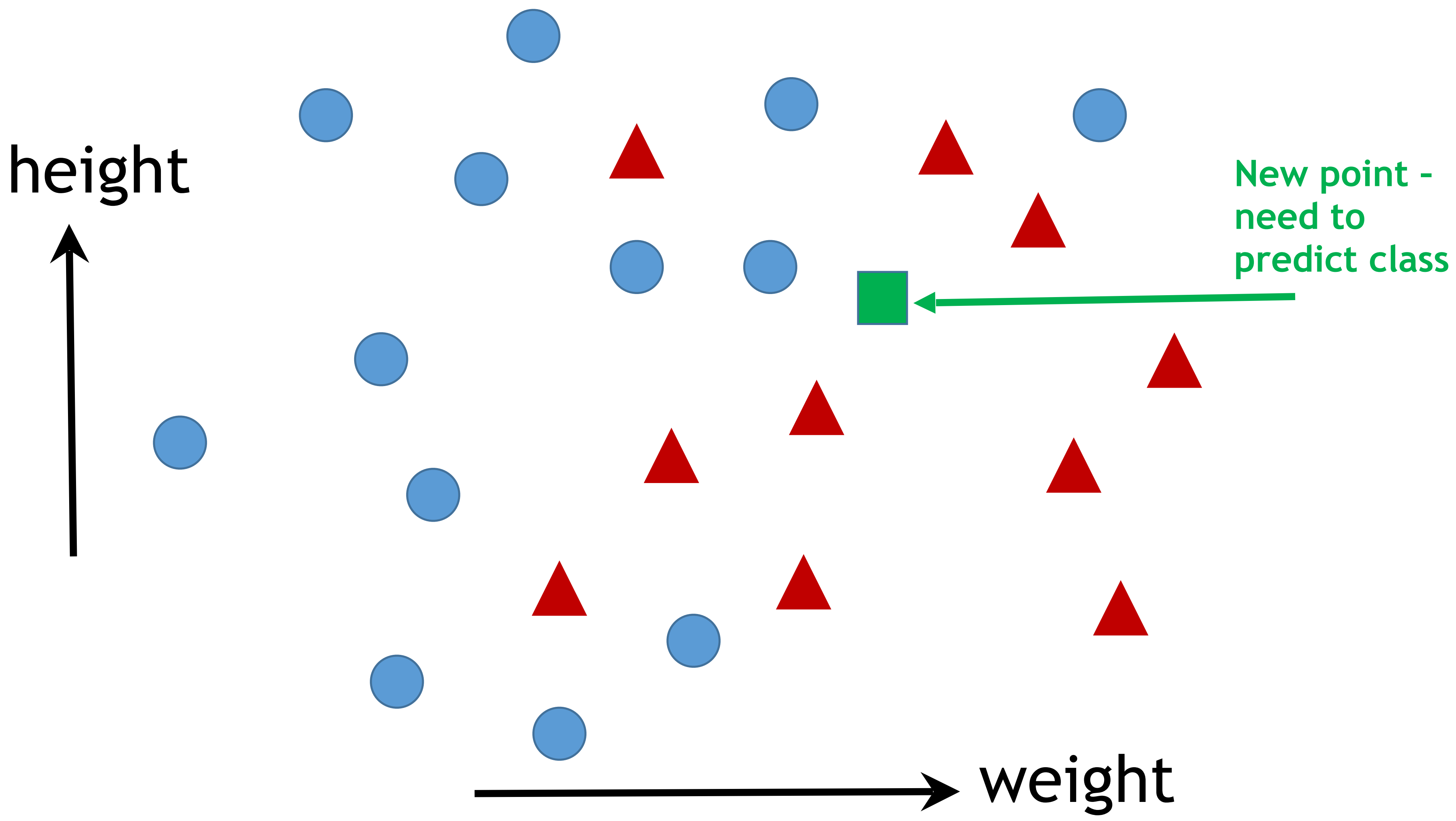
Parametric

Non-parametric

K-nearest neighbors

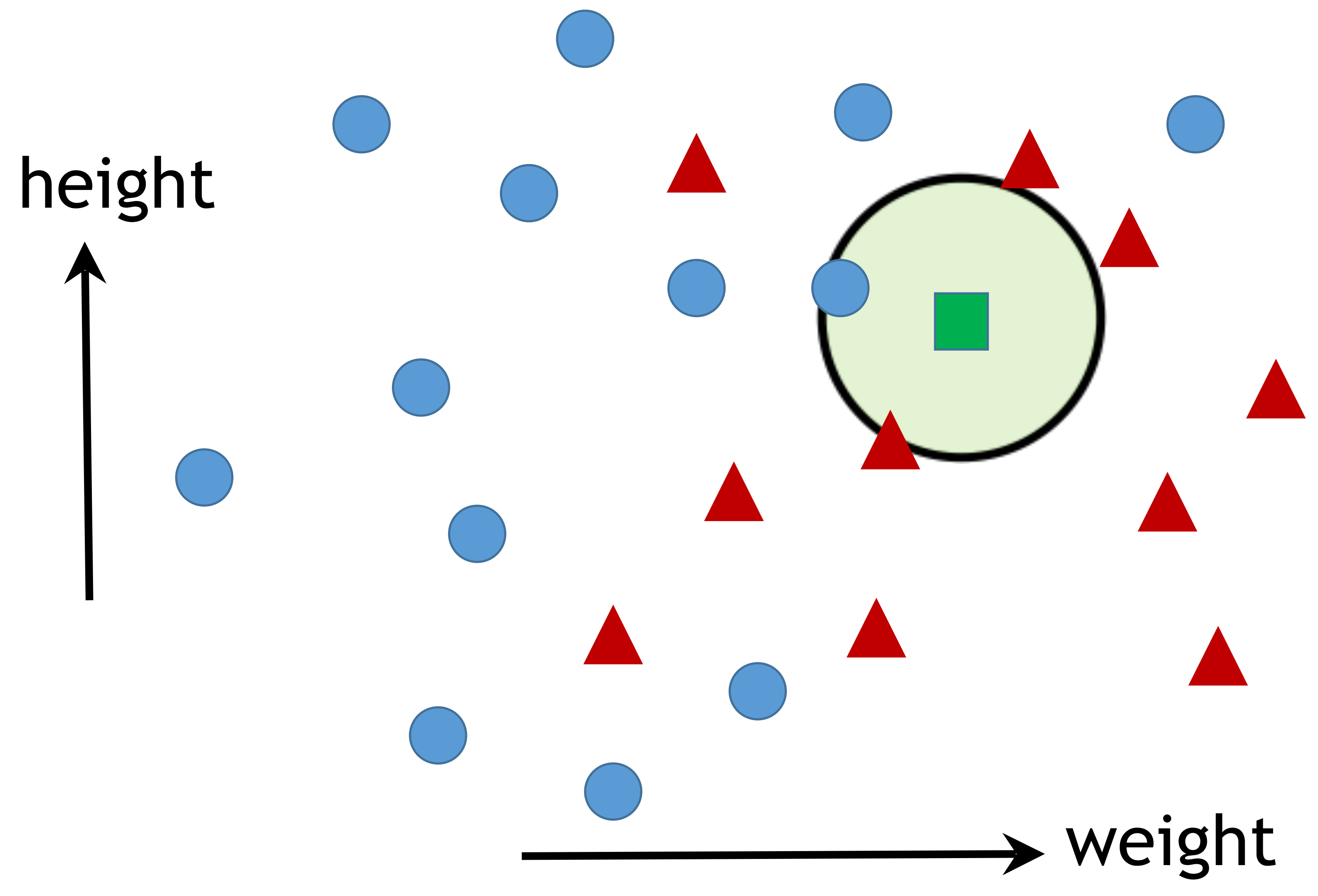


K-nearest neighbors



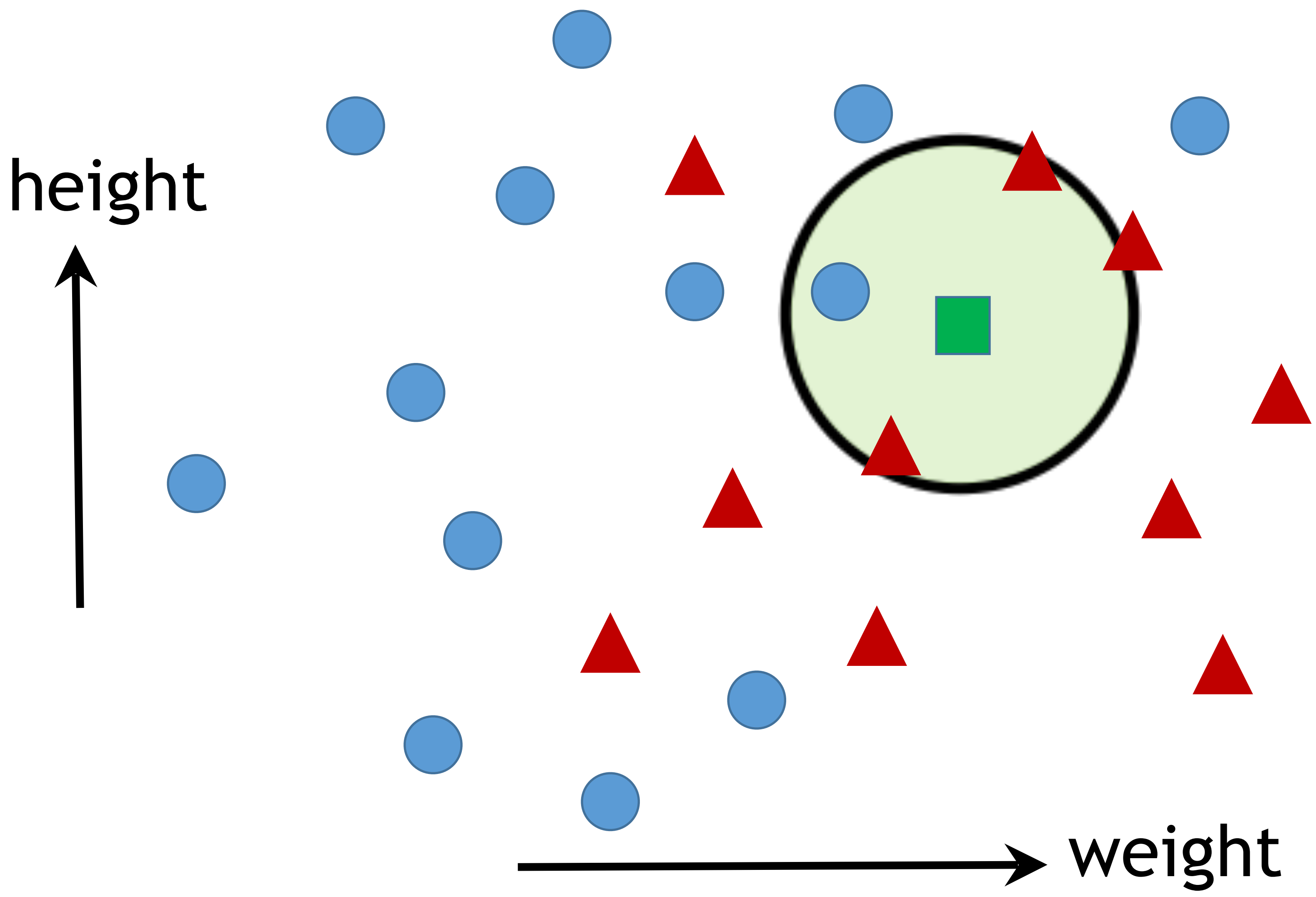
K-nearest neighbors, K=1

Closest neighbor is blue.



K-nearest neighbors, K=3

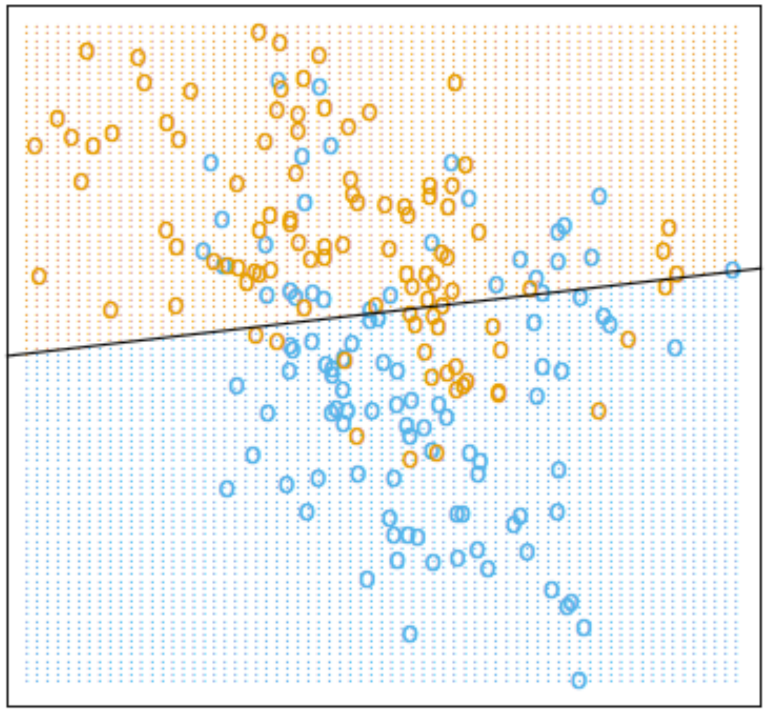
2/3 closest neighbors are red.



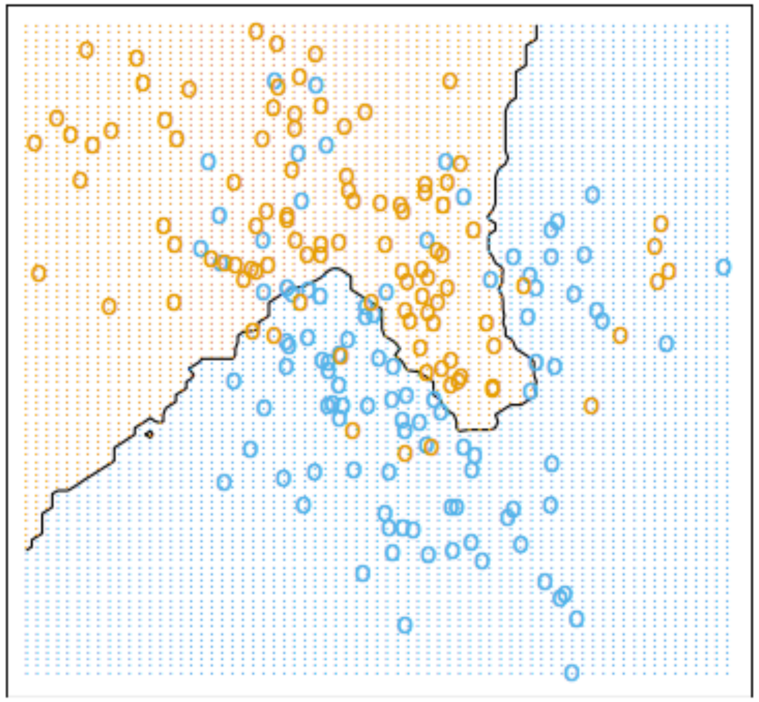
K-Nearest Neighbors

- Take majority vote among k-nearest neighbors
- Ties are broken at random (e.g. 1 red neighbor and 1 blue neighbor when k=2 – choose red or blue at random).
- Works for problems with more than 2 classes
- You'll need to define a distance metric. You'll probably want to standardize predictors.
- Interpretable model and easy to fit!

Bias-variance tradeoff



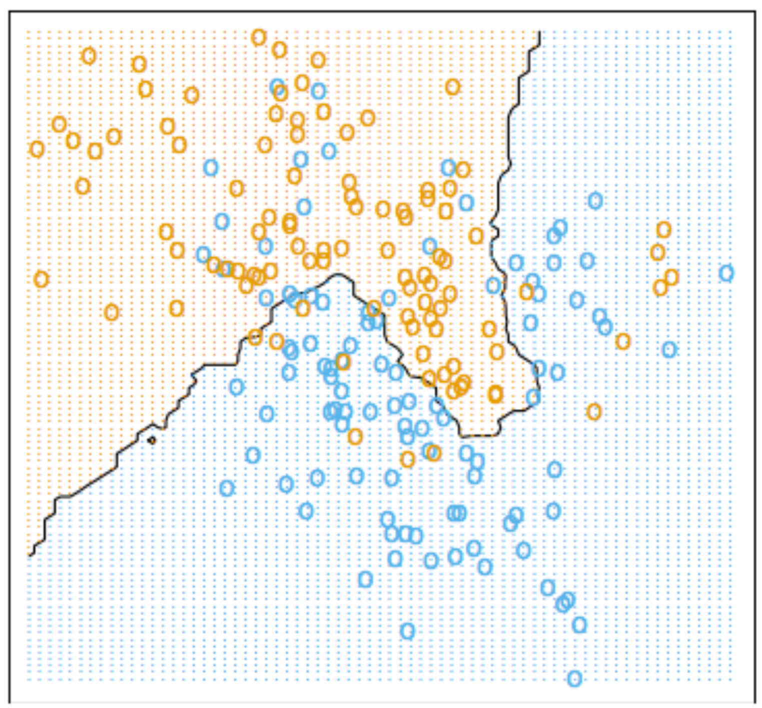
LDA



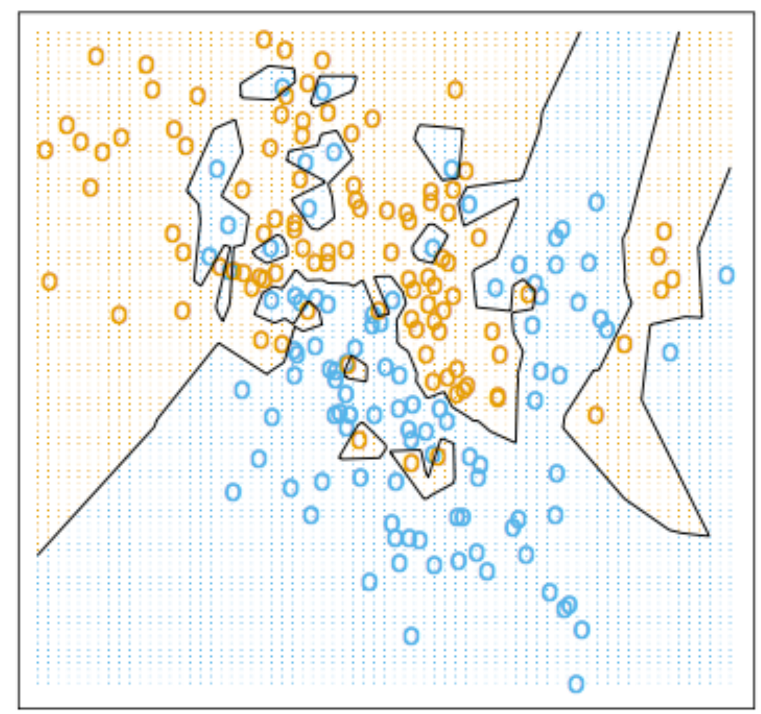
15-NN

Bias-variance tradeoff

As K decreases, the model is more likely to overfit:



15-NN



1-NN

Q: How do we pick K?

A: Training-test split. Evaluate the test error for different values of K, and pick the K with the smallest test error.

We'll talk about more sophisticated ways for selecting K next lecture.

Today's lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Loss functions
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. k-Nearest Neighbors
    6. Receiver Operating Characteristics (ROC)

Another way to evaluate a classification model…

- We’ve talked about 0-1 loss and the negative log likelihood.

- In addition, people often use sensitivity and specificity as two separate metrics.

Sensitivity and Specificity

Truth

Test

	Disease Free	**Diseased**	Total
Negative	True Negative TN	False Negative FN	FN+TN
Positive	False Positive FP	True Positive TP	TP+FP
Total	FP+TN	TP+FN	N

Specificity = $\dfrac{TN}{TN + FP}$ =

Proportion of negatives correctly identified as negative

Sensitivity = $\dfrac{TP}{TP + FN}$ =

Proportion of positives correctly identified as positive

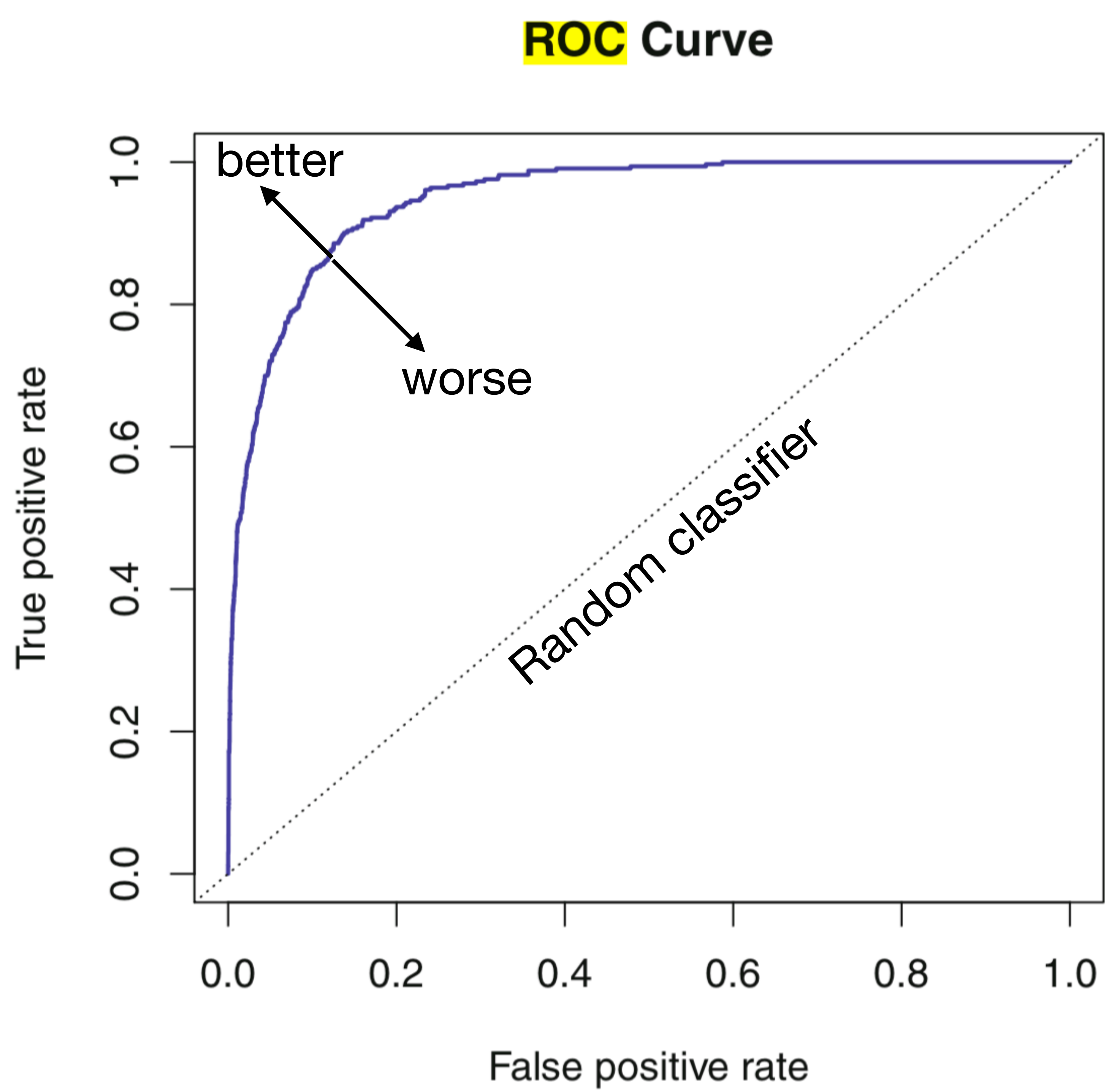
Thresholding predictions

- In practice, many classification models output a continuous value, such as a probability.

- In this case, you need to pick a threshold for deciding positive versus negative tests. Only then can you calculate sensitivity and specificity.

- Rather than calculating sensitivity and specificity for a single threshold, we could plot the two values *for all possible thresholds*.

Receiver operating characteristic (ROC)



(Sensitivity)

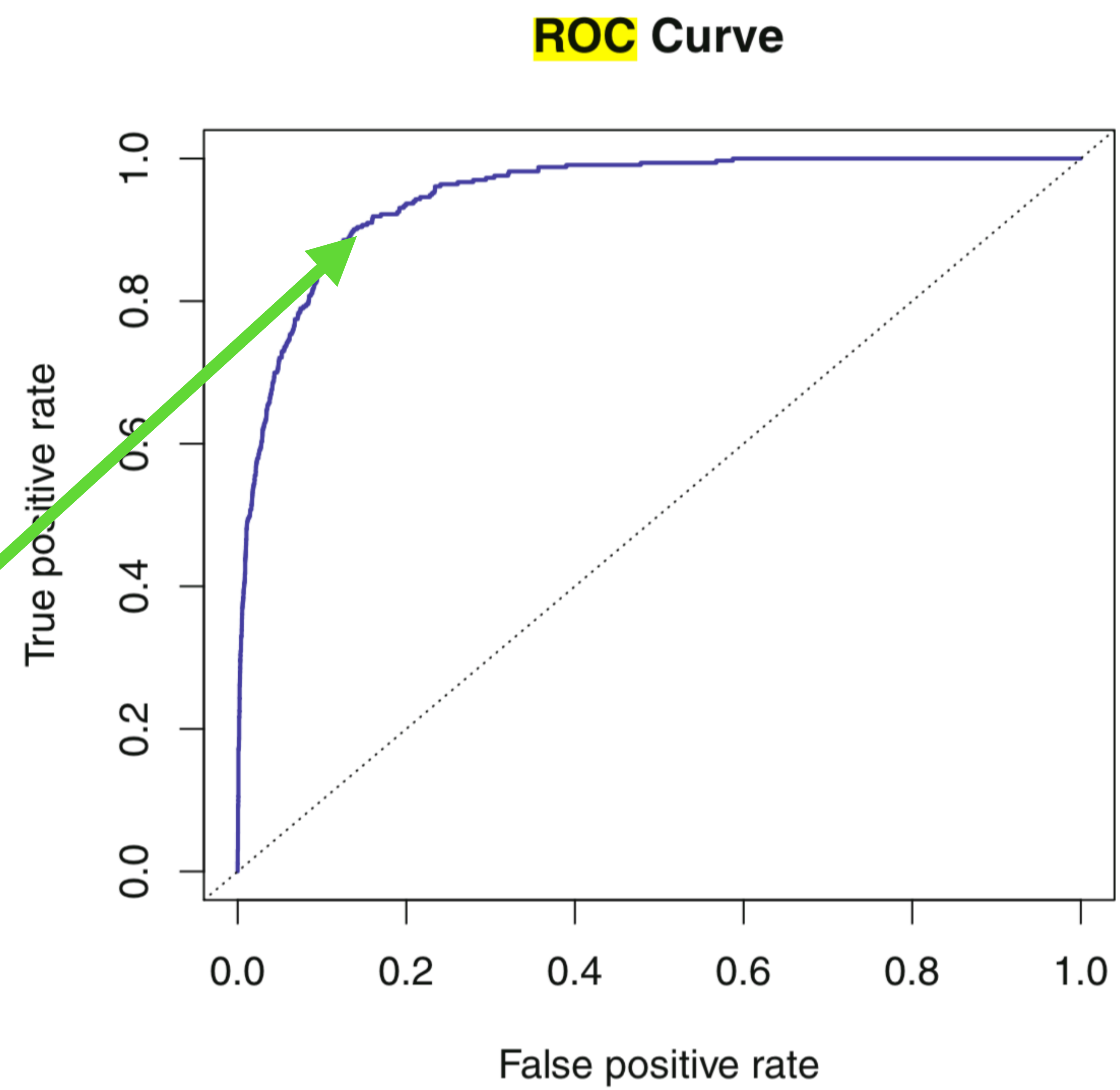
(1 - Specificity)

Picking a threshold

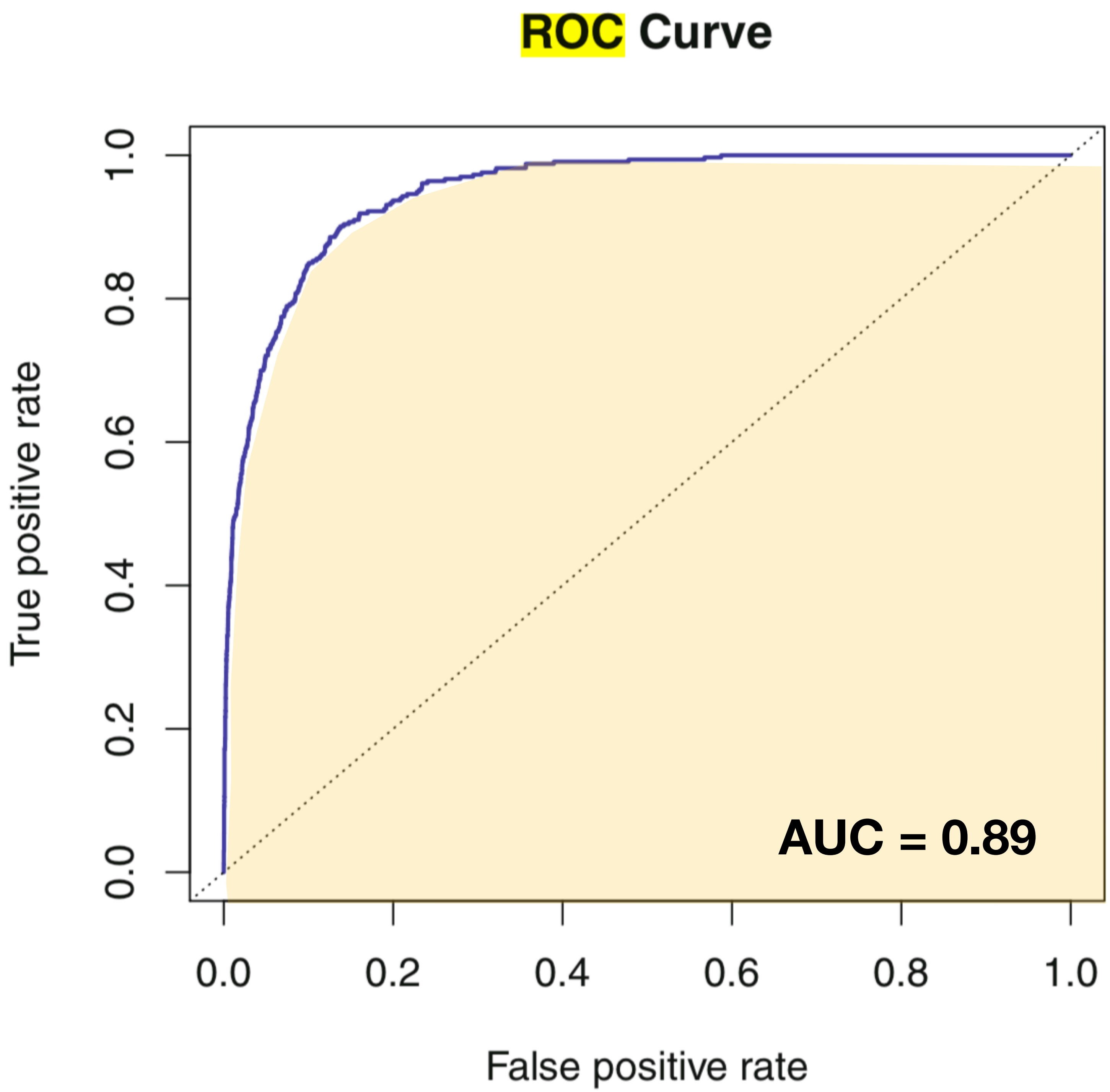
The ROC can help you pick a threshold for a binary classifier. For example:

- Pick the threshold that achieves the largest TPR given a maximum FPR of 0.2

- Pick the elbow (the farthest point from the diagonal line).



Area under the Receiver operating characteristic (AUROC or AUC)



- AUC of a random classifier is 0.5.
- Ideal AUC is 1.
- AUC characterizes the discriminatory capability of the model.
- Describes the behavior of a model independent of disease prevalence.

Today’s lecture

1. Linear regression as an optimization procedure
2. Classification
    1. Loss functions
    2. Logistic Regression
    3. Multinomial Regression
    4. Linear Discriminant Analysis
    5. k-Nearest Neighbors
    6. Receiver Operating Characteristics (ROC)