CPSC 4300/6300: Applied Data Science

2024

Classification

Dr. Aaron J. Masino
Associate Professor, School of Computing

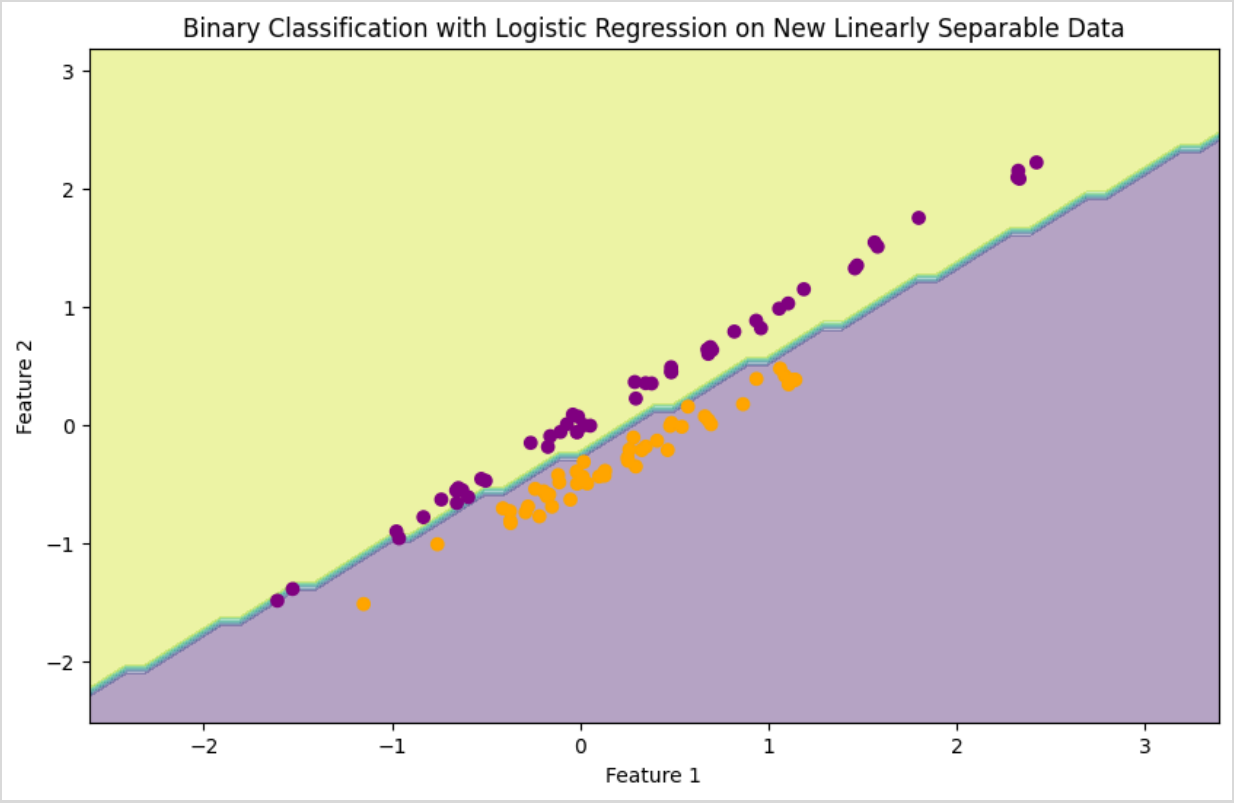
College of
ENGINEERING, COMPUTING
AND APPLIED SCIENCES

Lecture Outline

- Classification: Why not Linear Regression?
- Binary Response & Logistic Regression
- Estimating the Simple Logistic Model
- Classification using the Logistic Model
- Multiple Logistic Regression
- Classifier Performance

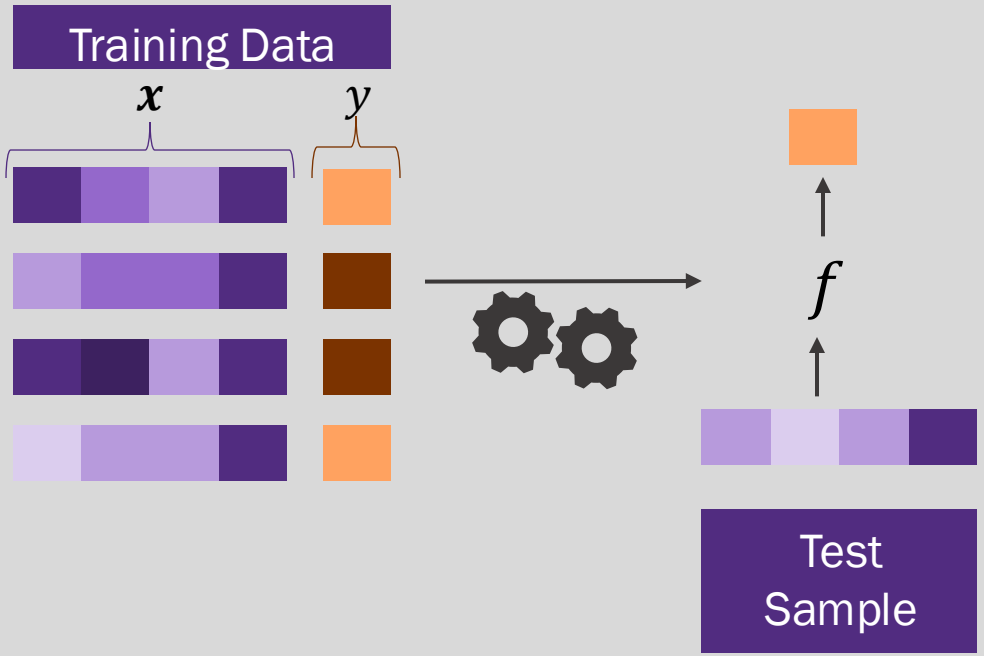
Classification

- Class – any attribute that partitions samples into distinct groups
- May be dichotomous (binary) or polytomous (multiclass)
- We usually assume that every sample belongs to exactly one class (there are variants that don't make this assumption)
- Objective: Construct a function, f, to estimate the class membership, Y, from input X
- Models will usually estimate the conditional probability, $p(Y|X)$ that the input is in class Y given the input, X.



Classification as a machine learning problem

- Supervised learning paradigm in which we have a dataset composed of pairs $(\boldsymbol{x}_i, y_i)$ where each $\boldsymbol{x}_i$ is a set of input feature values and y_i is the class label for the i^{th} sample
- Goal: Use the data to learn a mapping, $f(\boldsymbol{x}) \rightarrow y$
- **Key point**: y is a qualitative value, typically nominal, sometimes ordinal. Assigning numerical values to each of the possible y values and applying regression will generally perform poorly



Typical Classification Examples

The motivating examples for this lecture are mostly based on medical data sets. Classification problems are common (but by no means unique to) this domain:

- Diagnostic tests (pregnancy, cancer screening, infection screening, etc, …)
- Disease progression status based on treatment and other indicators
- Patient disease subtype assignment based on various genomic markers

Why not linear regression?

A categorical variable y could be encoded to be quantitative. For example, let's assume y represents a student's major and takes one of three values:

$$y = \begin{cases} 1 & if \text{ Computer Science (CS)} \\ 2 & if \text{ Statistics} \\ 3 & \text{otherwise} \end{cases} .$$

Given a dataset: $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_N, y_N)\}$

why not apply linear regression to predict the numerical value of y?

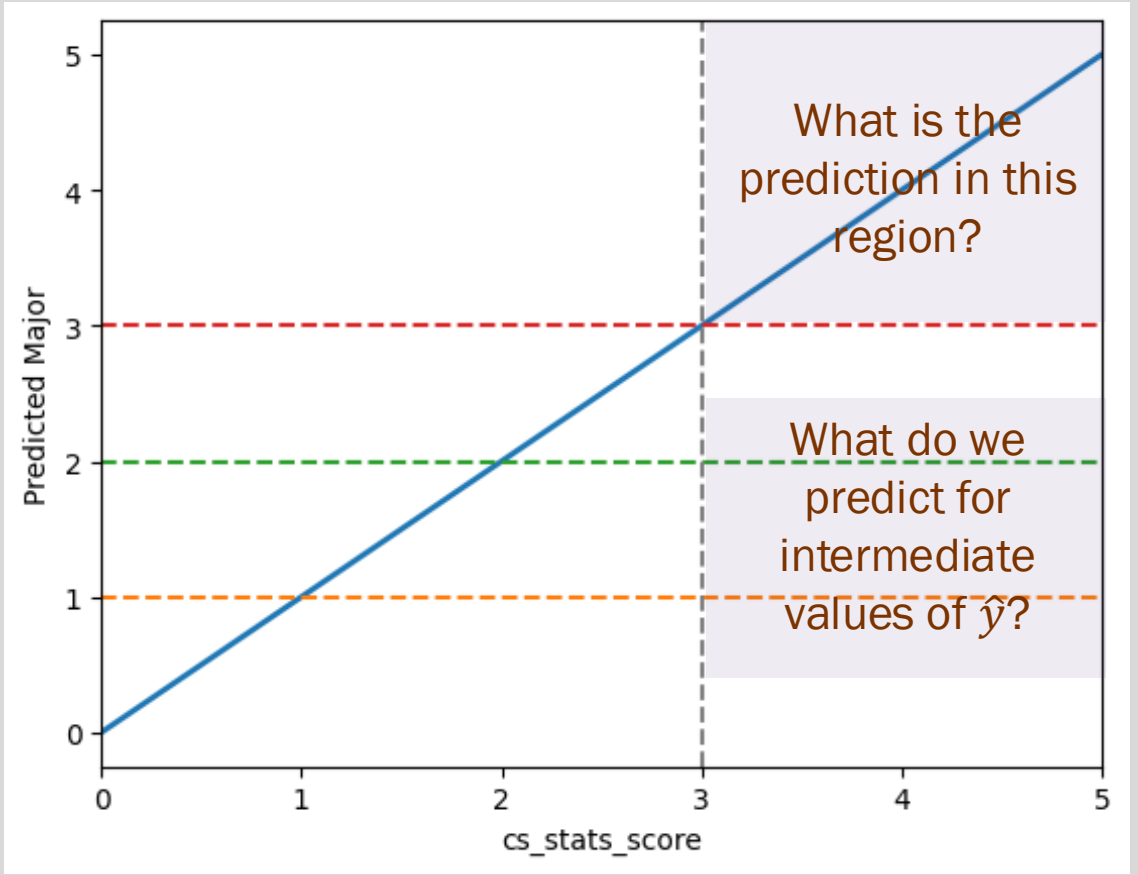
Linear regression **does not work well** in this setting. Why?

Why not linear regression?

Imagine we had a score, $s_{cs,stats} \ni [0,100]$ for every student and that we fit a linear regression model $\hat{y} = \beta_0 + \beta_1 s_{cs,stats}$ where y indicates student major

The model implies a specific ordering y and treats a one-unit change in y equivalently. Should the change from y = 1 to y = 2 (CS to Statistics) be interpreted the same as the change from y = 2 to y = 3 (Statistics to everyone else).

Reordering the response variables to would lead to a different model.



Simple Classification Example: Binary Response

Let's consider the situation when the outcome variable is binary, i.e., y is in one of two possible classes

Can we use linear regression for this? Technically yes, but it still creates problems.

Let's consider an example of predicting if someone has arteriosclerotic heart disease (ACS)

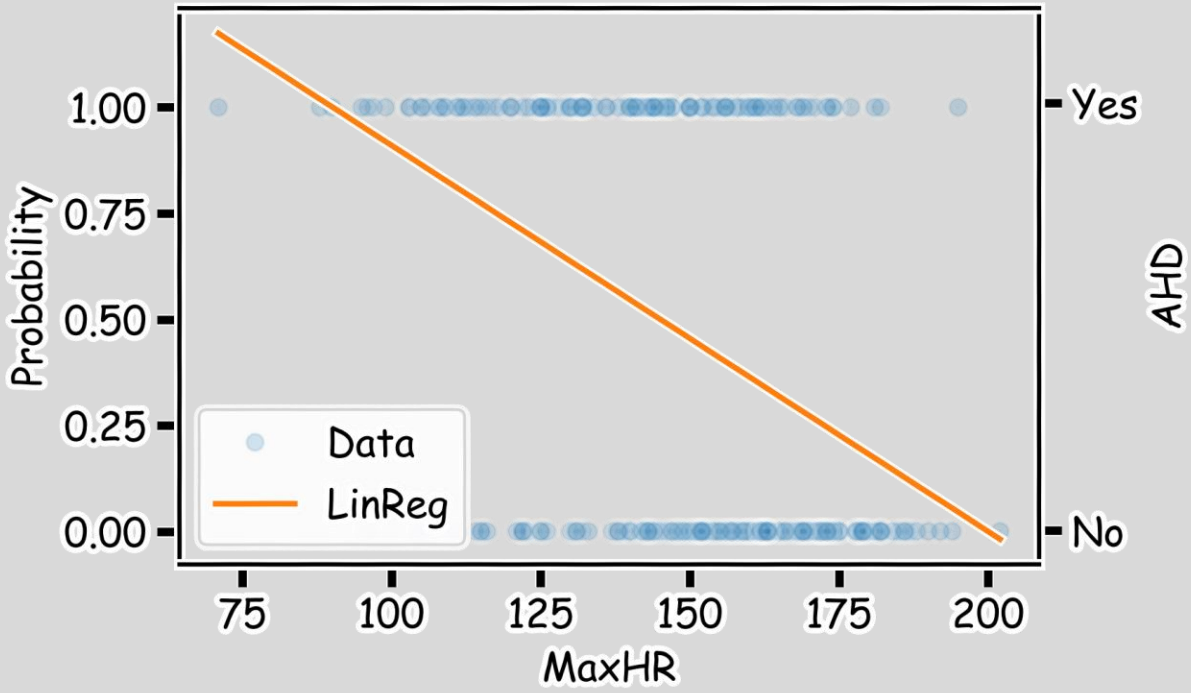
Our training data takes the form:

$$y = \begin{cases} 0 & no\ ACS \\ 1 & yes\ ACS \end{cases}$$

Linear regression could be used to predict y directly from a set of covariates and if $\hat{y} \geq 0.5$, we could predict *yes ACS* and *no ACS* if $\hat{y} < 0.5$.

Simple Classification Example: Binary Response

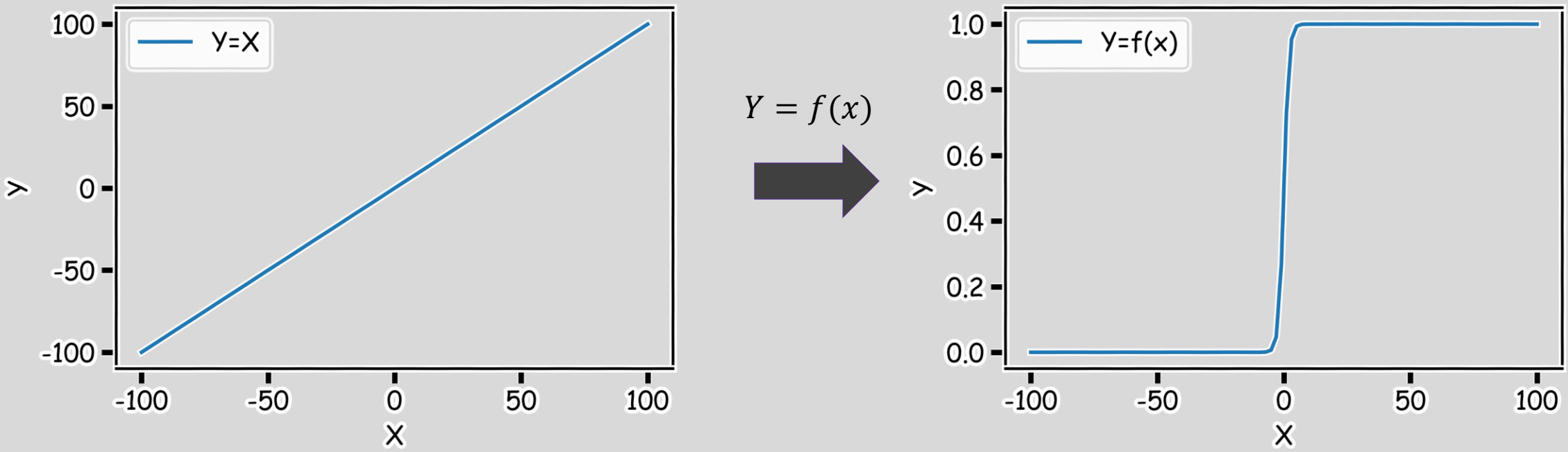
The main issue is you could get nonsensical values for $\hat{y}$. Since this is modeling $P(y = 1|hr)$, values for $\hat{y} < 0$ and $\hat{y} > 1$ would be at odds with the natural measure for y. Linear regression can lead to this issue.



Simple Classification Example: Binary Response

How can we address these issues?

What if we had a function that did this:



Lecture Outline

- Classification: Why not Linear Regression?
- Binary Response & Logistic Regression
- Estimating the Simple Logistic Model
- Classification using the Logistic Model
- Multiple Logistic Regression
- Classifier Performance

Logistic Regression with One Predictor Variable

Logistic Regression addresses the problem of constraining the probability estimate, $P(y = 1)$, to the range of [0,1].

The logistic regression model uses the ***logistic*** function, $\sigma(z) = {}^{1}/_{(1+e^{-z})}$ for this purpose

$$P(Y = 1) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} = \frac{1}{1 + e^{-(\beta_0+\beta_1 X)}}$$

Logistic Regression with One Predictor Variable

As a result, the model will predict $P(y = 1)$ with an S-shaped curve, which is the general shape of the logistic function.

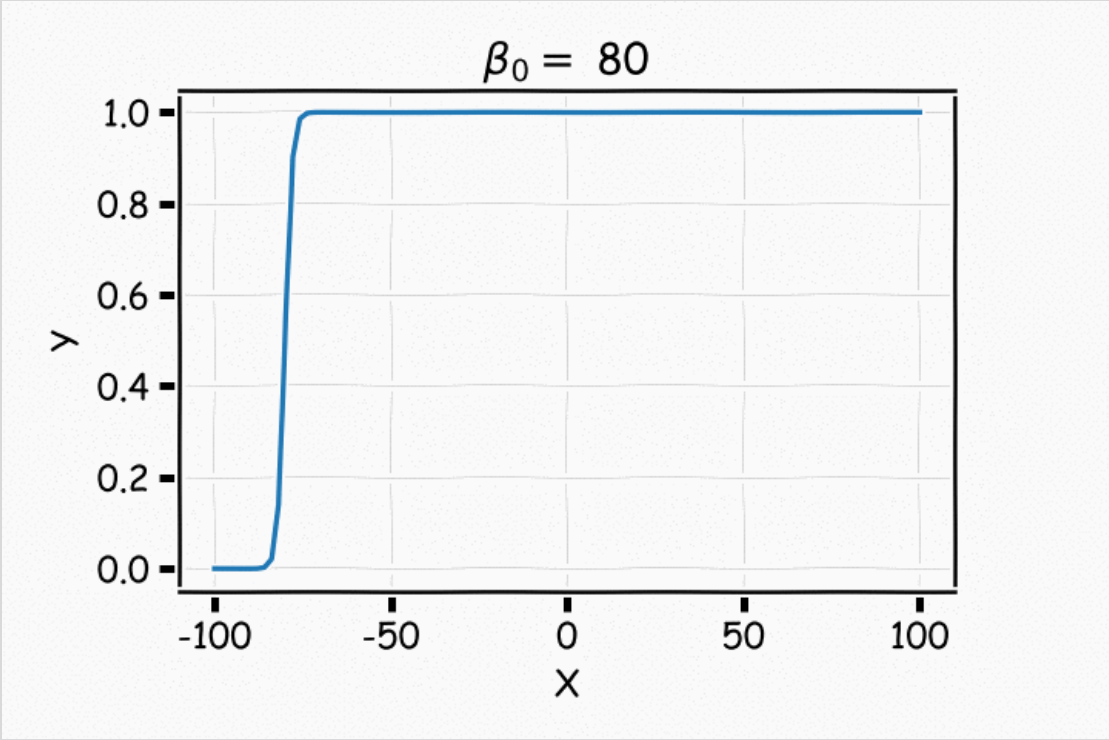
β_0 shifts the curve right or left by $c = -\frac{\beta_0}{\beta_1}$.

β_1 controls the steepness of the S-shaped curve is.

Note: if β_1 is positive, then the predicted $P(y = 1)$ increases as X increases and if β_1 is negative, then the $P(y = 1)$ decreases as X increases.

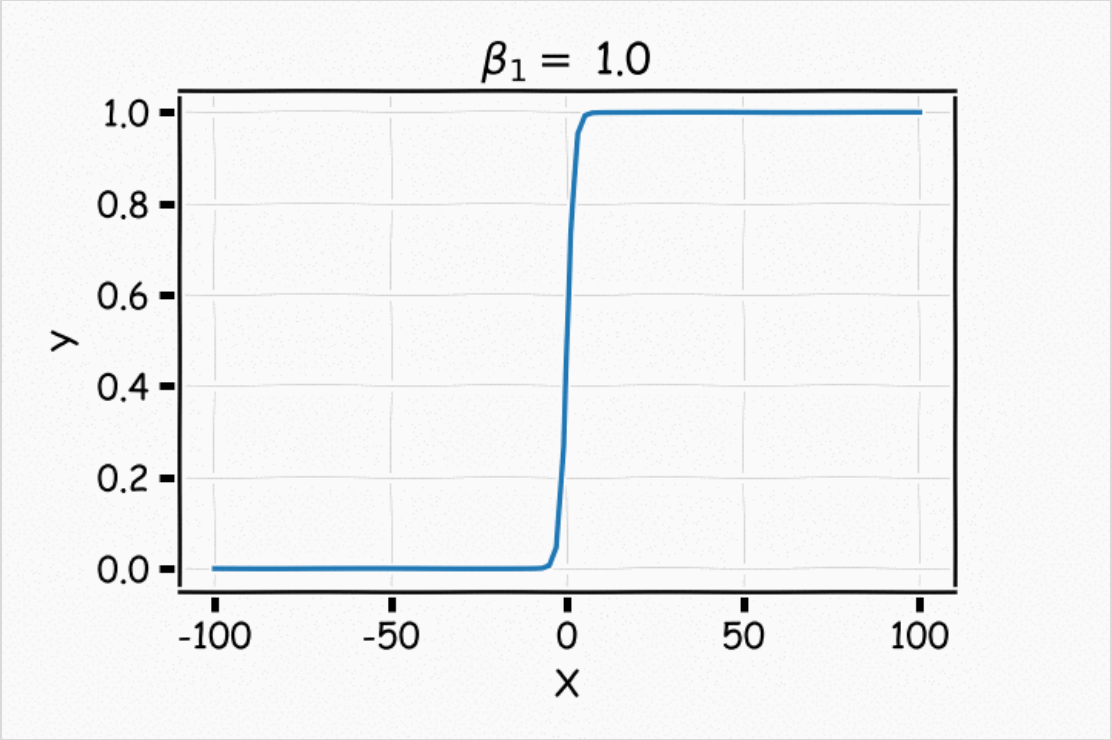
Logistic Regression with One Predictor Variable

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$



Logistic Regression with One Predictor Variable

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$



Logistic Regression with One Predictor Variable

With a little bit of algebraic work, the logistic model can be rewritten as:

$$\ln\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + \beta_1 X.$$

Logit

The value inside the natural log function $\frac{P(Y=1)}{1-P(Y=1)}$, is called the ***odds***, thus logistic regression is said to model the ***log-odds*** with a linear function of the predictors or features, X.

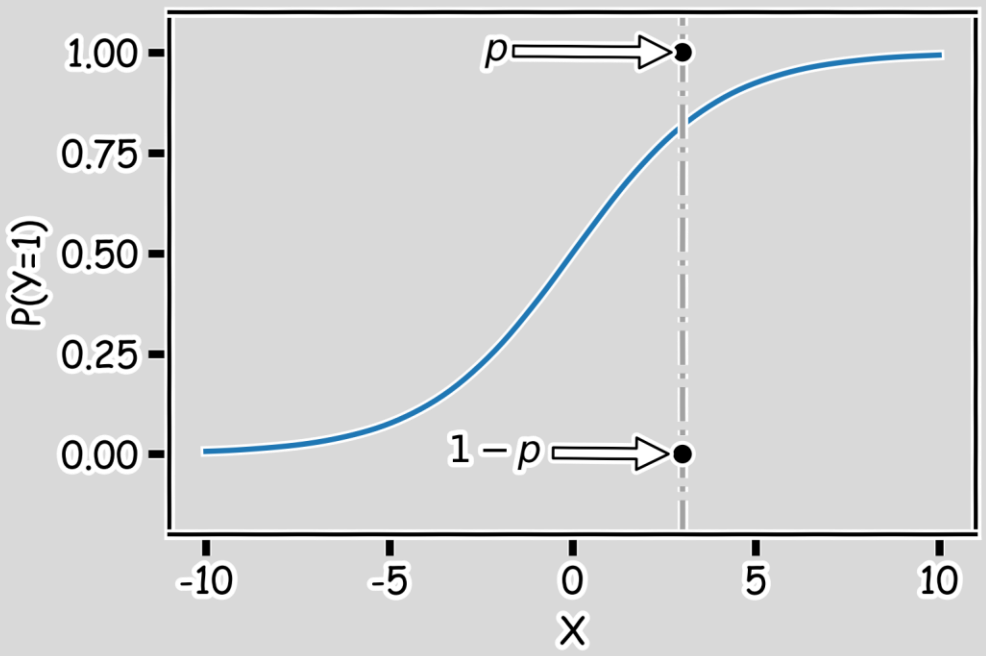
This gives a similar interpretation to linear regression

- A unit change in X is associated with a β_1 change in the **log-odds**
- A unit change in X is associated with an e^{β_1} change in the odds that $Y = 1$.

Lecture Outline

- Classification: Why not Linear Regression?
- Binary Response & Logistic Regression
- Estimating the Simple Logistic Model
- Classification using the Logistic Model
- Multiple Logistic Regression
- Classifier Performance

Estimation in Logistic Regression



Probability $Y = 1$: $p = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$

Probability $Y = 0$: $1 - p$

$$P(Y = y) = p^y(1-p)^{(1-y)}$$

where:

$p = P(Y = 1|X = x)$ and therefore p depends on X.

How do we pick the logistic regression coefficients? Maximum Likelihood

If we observe *n* samples, the likelihood of a **single observation** for *p* given *x and y* is:

$$L(p_i|Y_i) = P(Y_i = y_i) = p_i^{y_i}(1-p_i)^{1-y_i}$$

y_i is 0 or 1 for each *i*

Given the observations are independent, what is the likelihood function **for the entire dataset**?

The likelihood function is NOT a probability density function
It measures the *support* given by the observed data for possible values of β_0, β_1

$$L(p|Y) = \prod_{i=1}^{n} p(x_i)^{y_i}[1-p(x_i)]^{y_i-1}$$

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

$$l(p|Y) = -\log L(p|Y) = -\sum_{i=1}^{n} y_i \log p_i + (1-y_i)\log(1-p_i)$$

Likelihood as a Loss Function

$$l(p|Y) = -\sum_{i=1}^{n} y_i \log \frac{1}{1+e^{-\beta X_i}} + (1-y_i) \log\left(1 - \frac{1}{1+e^{-\beta X_i}}\right)$$

How do we minimize this?

Differentiate, equate to zero and solve for it!

But this looks messy! It will not necessarily have a closed form solution.

So how do we determine the parameter estimates? Through an iterative optimization approach.

Lecture Outline

- Classification: Why not Linear Regression?
- Binary Response & Logistic Regression
- Estimating the Simple Logistic Model
- Classification using the Logistic Model
- Multiple Logistic Regression
- Classifier Performance

Using Logistic Regression for Classification

How can we use a logistic regression model to perform classification?

That is, how can we predict when $Y = 1$ vs. when $Y = 0$?

We mentioned before, we can classify all observations for which $\hat{P}(Y = 1) \geq 0.5$ to be in the group associated with $Y = 1$ and then classify all observations for which $\hat{P}(Y = 0) < 0.5$ to be in the group associated with $Y = 0$.

Using such an approach is called the standard Bayes classifier.

The Bayes classifier takes the approach that assigns each observation to the most likely class, given its predictor values.

In general, we can select $0 < \gamma < 1$ as a threshold s.t. $\hat{P}(Y = 1) \geq \gamma$ implies $\hat{y} = 1$

Using Logistic Regression for Classification

The Bayes classifier minimizes the overall classification error rate. That is, it minimizes:

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

This has potential to be a good classifier if the predicted probabilities are well calibrated, that is if for all samples predicted to have Y=1 with probability *p,* then *100p%* will be in the class associated with Y=1

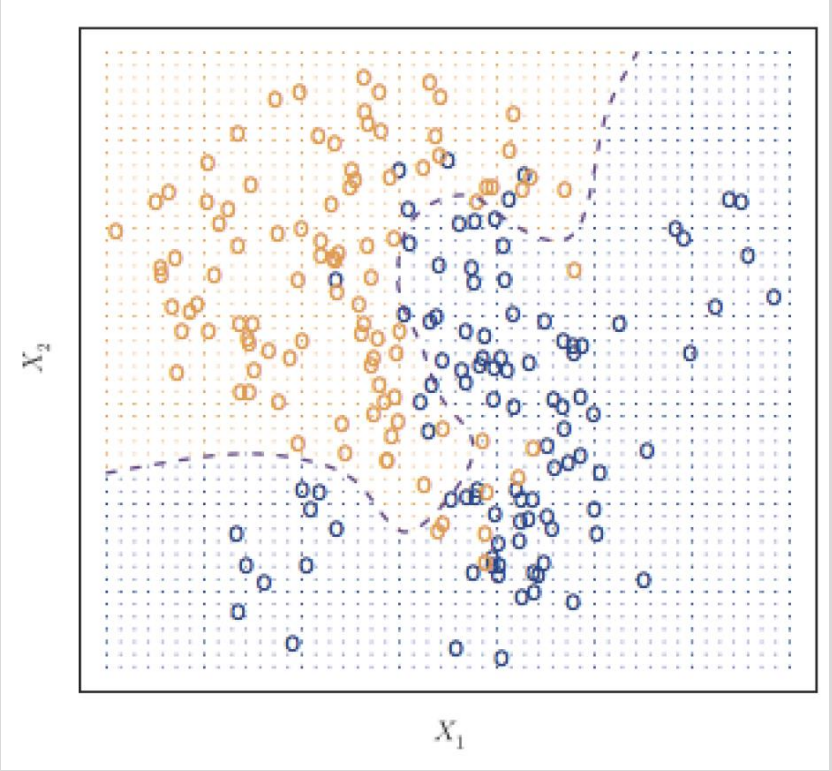
The Bayes classifier may be a poor indicator within a group, particularly if there is class imbalance

Lecture Outline

- Classification: Why not Linear Regression?
- Binary Response & Logistic Regression
- Estimating the Simple Logistic Model
- Classification using the Logistic Model
- Multiple Logistic Regression
- Classifier Performance

Classifier with Multiple Predictors

How can we estimate a classifier, based on logistic regression, for the following plot?



Multiple Logistic Regression

Multiple logistic regression is a generalization of the single predictor case to multiple predictors. More specifically we can define a multiple logistic regression model to predict $P(Y = 1)$ as such:

$$\log\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

$$P(y=1) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{p} \beta_i X_p)}}$$

Logistic regression with multiple classes

- Assume each sample is to be classified as belonging to 1 of K classes
- Apply a *one vs. all* approach:
  - Calculate the probability that the sample belongs to each of the K classes
  - Assign the sample to the class with maximum probability
- Formulate the individual probabilities that the sample belongs to class k using the *softmax* encoding

$$p(y = k|\boldsymbol{x}) = \frac{e^{z_k}}{\sum_{l=1}^{K} e^{z_l}}$$

Notice, the model for each class, k, has its own coefficient estimate $\boldsymbol{\beta}_k$

where $z_k = \boldsymbol{\beta}_k \cdot \boldsymbol{x}$

- The coefficients are estimated by maximizing the log likelihood function over the observed data as in binary logistic regression

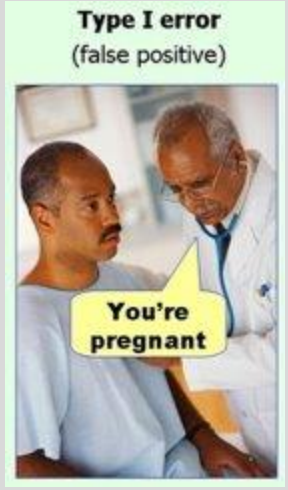
Lecture Outline

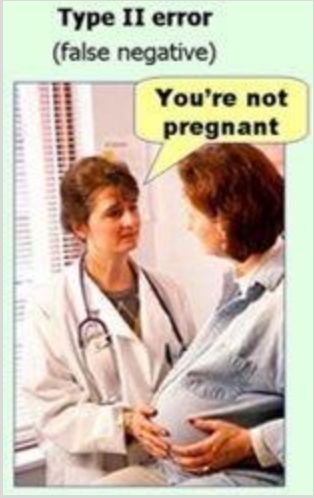
- Classification: Why not Linear Regression?
- Binary Response & Logistic Regression
- Estimating the Simple Logistic Model
- Classification using the Logistic Model
- Multiple Logistic Regression
- Classifier Performance

Classifier error types

All classifier performance metrics are derived from counts of TP, FP, TN, FN

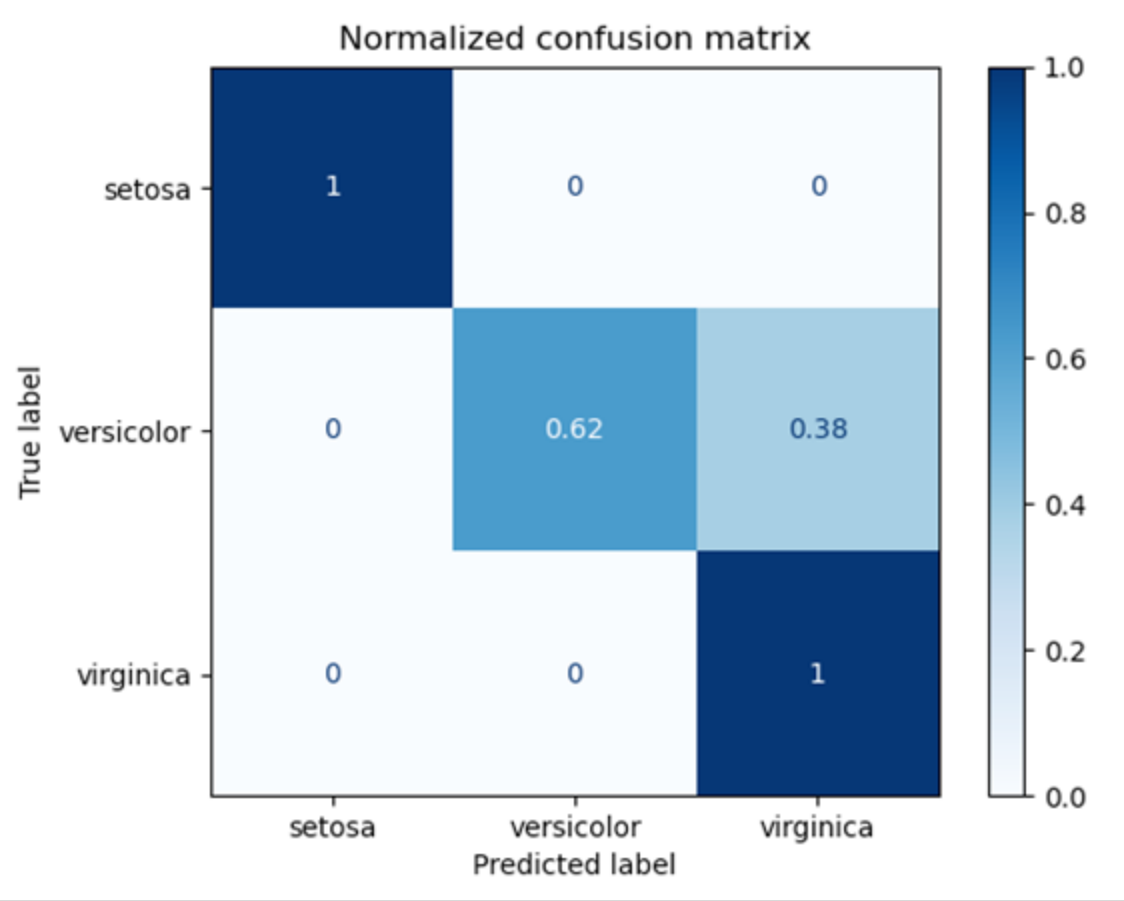
		Actual Class: Positive	Actual Class: Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP) *False Alarm* *Type I Error*
Predicted Class	Negative	False Negative (FN) *Type II Error*	True Negative (TN)





Confusion Matrix

We can identify how correctly classified were the observations by computing a Confusion Matrix that shows the correct and misclassified observations.

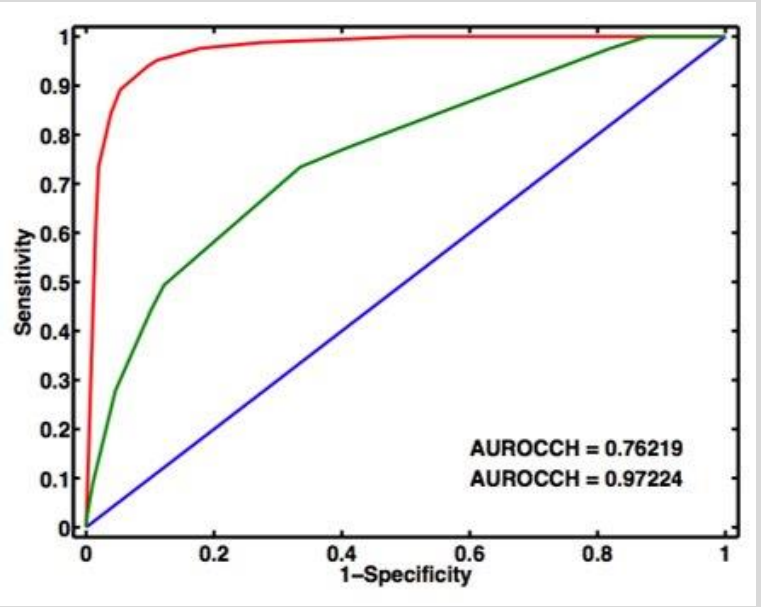


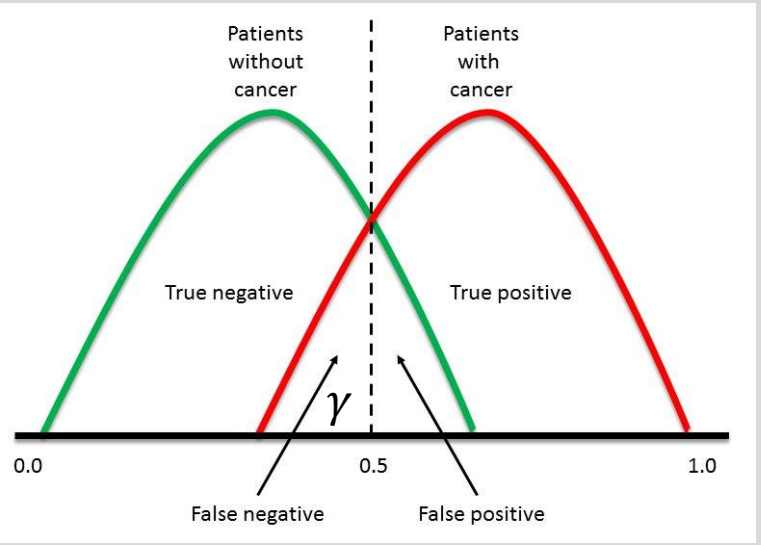
Classifier performance point metrics

- Accuracy = (TP+TN)/(TP+TN+FP+FN)
- Sensitivity (a.k.a Recall, True Positive Rate) = TP/(TP+FN) = TP/P
  - The fraction of actually positive samples predicted positive by the model
- Specificity (a.k.a. True Negative Rate) = TN/(TN+FP) = TN/N
  - The fraction of actually negative samples predicted negative by the model
- Precision (a.k.a. Positive Predictive Value) = TP/(TP+FP)
  - The fraction of samples predicted positive by the model that are actually positive
- False Positive Rate (a.k.a. Probability of false alarm) = FP/(FP+TN) = FP/N
- F1 Score = 2(Precision*Recall)/(Precision + Recall)
- Balanced Accuracy = (Sensitivity + Specificity)/2

Classifier performance with ROC and AUC

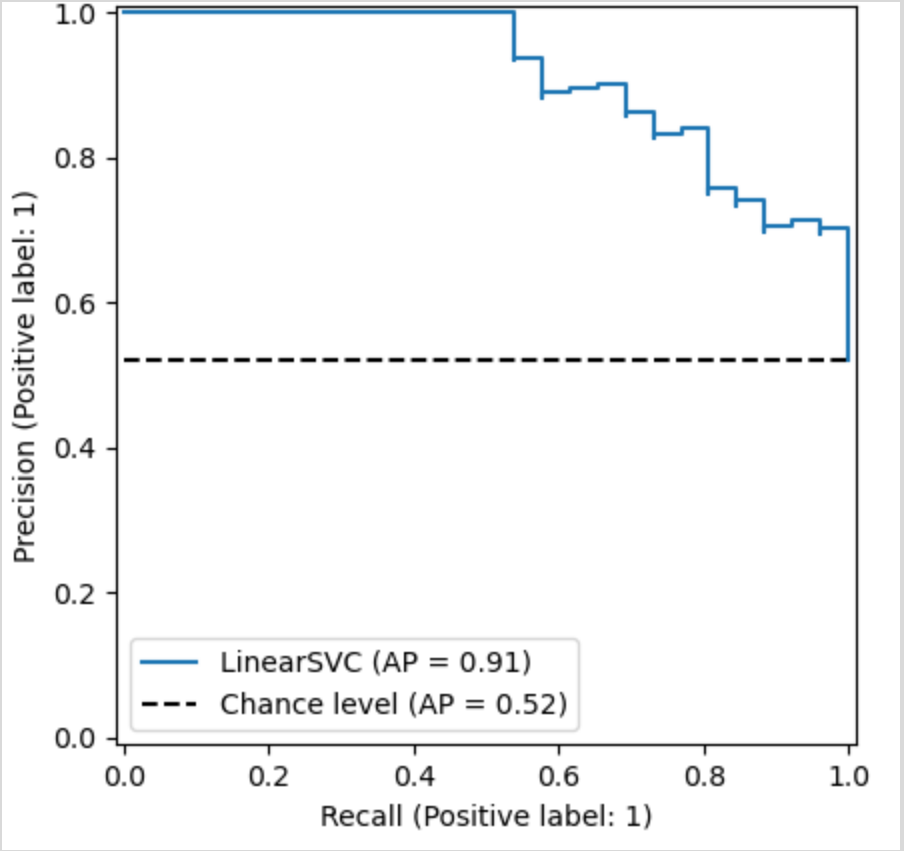
- Recall, we can select $0 < \gamma < 1$ as a threshold s.t. $\hat{P}(Y = 1) \geq \gamma$ implies $\hat{y} = 1$. What happens if we change γ?
- Receiver Operating Characteristic Curve (ROC)
  - Illustrates tradeoff between FPR and TPR of a binary classifier as the discrimination threshold, γ, is adjusted
  - Axes
    - 1 – Specificity (false positive rate) = 1 – TN/N
    - Sensitivity (true positive rate) = TP/P
- Area under the curve (AUC)
  - Summary statistic
  - AUC = 0.5 indicates no predictive value (random guessing)
  - AUC = 1.0 indicates perfect predictive value





Classifier performance with precision-recall curve

- Illustrates tradeoff between precision (PPV) and sensitivity of a binary classifier as the discrimination threshold, γ, is adjusted
- Particularly useful for highly imbalanced datasets (i.e., where one class is very rare). In those cases, it's possible to achieve high AUC with very low precision*
- Axes
  - Sensitivity (true positive rate) = TP/P
  - Precision (PPV) = TP/(TP+FP)
- Average precision – summary metric indicating the weighted average precision for each discrimination threshold



*Romero-Brufau S, Huddleston JM, Escobar GJ, Liebow M. Why the C-statistic is not informative to evaluate early warning scores and what metrics to use. Crit Care. 2015 Aug 13;19(1):285

Questions?