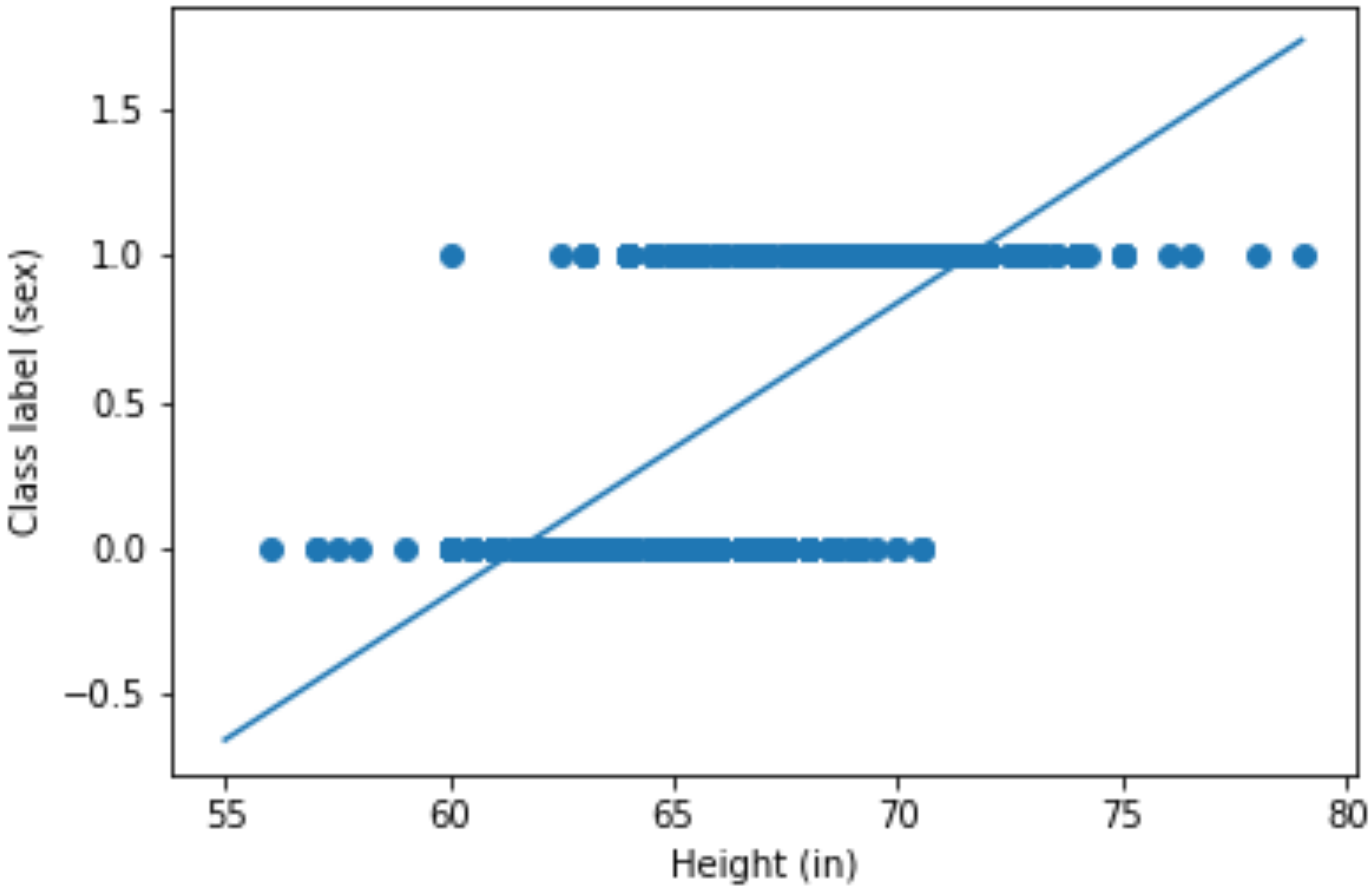
# DATA221 Intro Machine Learning
# 07 logistic regression

William Trimble

Winter 2023

THE UNIVERSITY OF CHICAGO

# Linear regression on Galton height data




Linear regression on indicator variables..
not the best we can do

Really? Why not?
Linear discriminator functions select a special dimension out of $R^D$ which is the direction the difference lies..
leaves a $R^{D-1}$ dimensional hyperplane dividing the vector space of features in half.

# Linear classifier

Prediction(**X**, **w**) = g($w_0$ + $\mathbf{w}^T$ **X** )

Nonlinear function; “activation function”

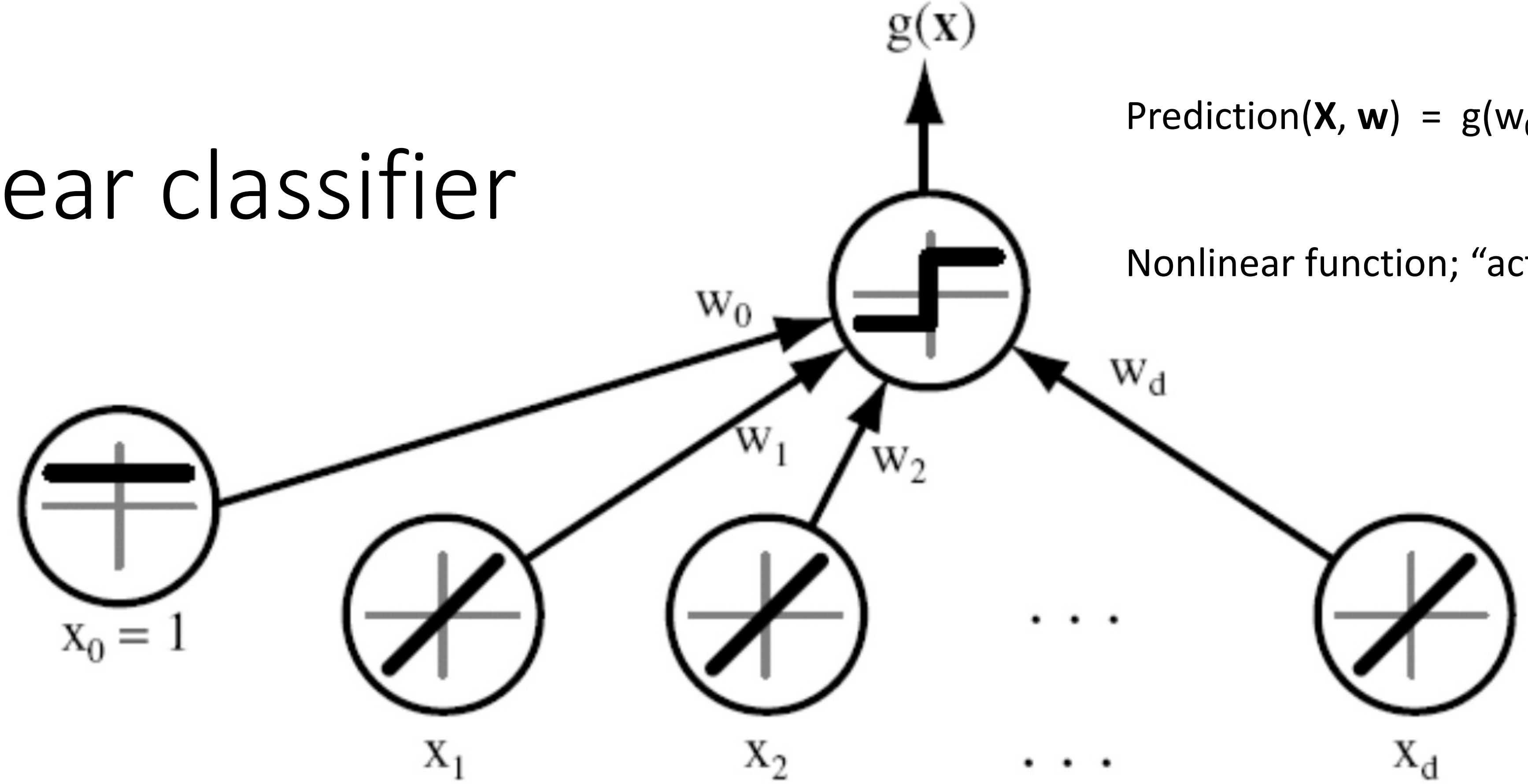



Figure 5.1: A simple linear classifier having $d$ input units, each corresponding to the values of the components of an input vector. Each input feature value $x_i$ is multiplied by its corresponding weight $w_i$; the output unit sums all these products and emits a $+1$ if $\mathbf{w}^t\mathbf{x} + w_0 > 0$ or a $-1$ otherwise.

Duda, Pattern Classification

# Wait, not so fast, this is one of those equations that hides a lot of complexity

- Prediction($\mathbf{X}$, $\mathbf{w}$) = $g(w_0 + \mathbf{w}^T \mathbf{X})$

- $\mathbf{y} = g(w_0 + \mathbf{w}^T \mathbf{X})$ n observations

k dimensions
in y

r features for each
observation in x

( _ x _ )

# Wait, not so fast, this is one of those equations that hides a lot of complexity

- Prediction(**X**, **w**) = $g(w_0 + \mathbf{w}^T \mathbf{X})$

- $\mathbf{y} = g(w_0 + \mathbf{w}^T \mathbf{X})$ n observations

( _ x _ ) ( _ x _ ) ( _ x _ ) ( _ x _ )

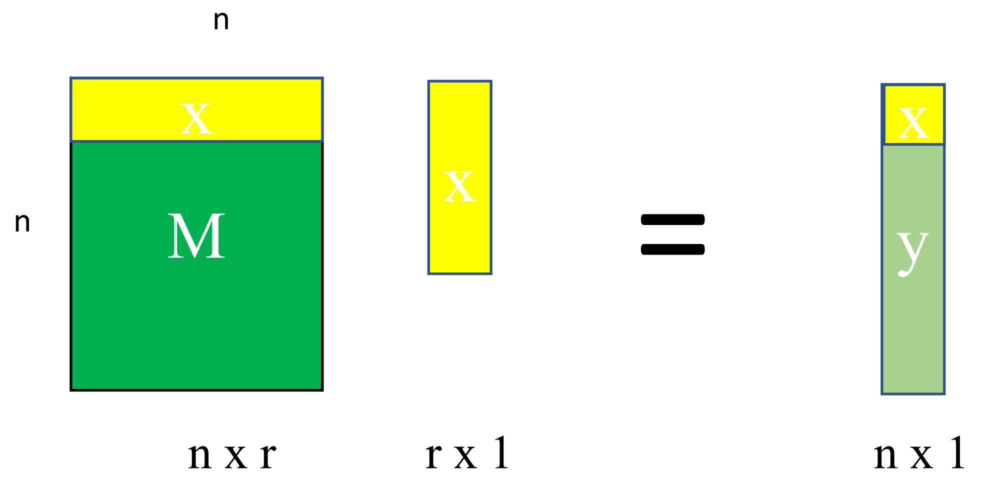
r features for each observation in x

k dimensions in y

# Non-square matrix multiplication....

$$\mathbf{M}\,\mathbf{x} = \mathrm{y}$$

$$\Sigma M_{ij}\, x_j = y_i$$

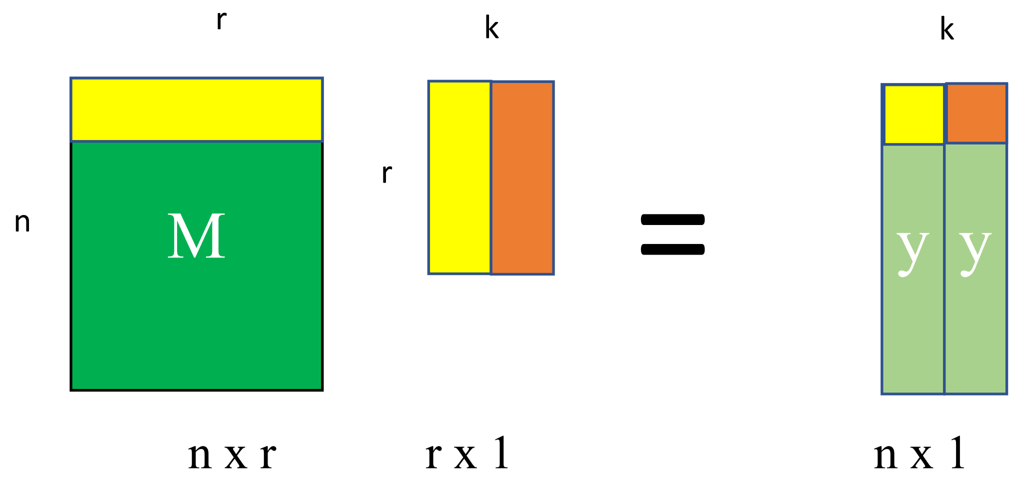


# Non-square matrix multiplication....

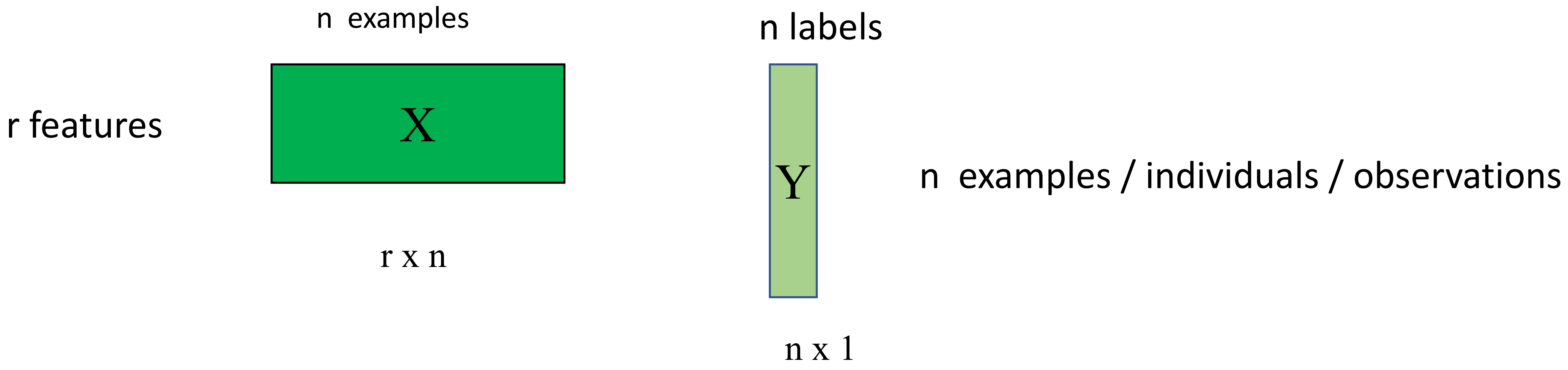
$$\mathbf{M}\,\mathbf{w} = \mathrm{y}$$

$$\Sigma M_{ij}\, w_{jk} = y_{ik}$$

square
matrix

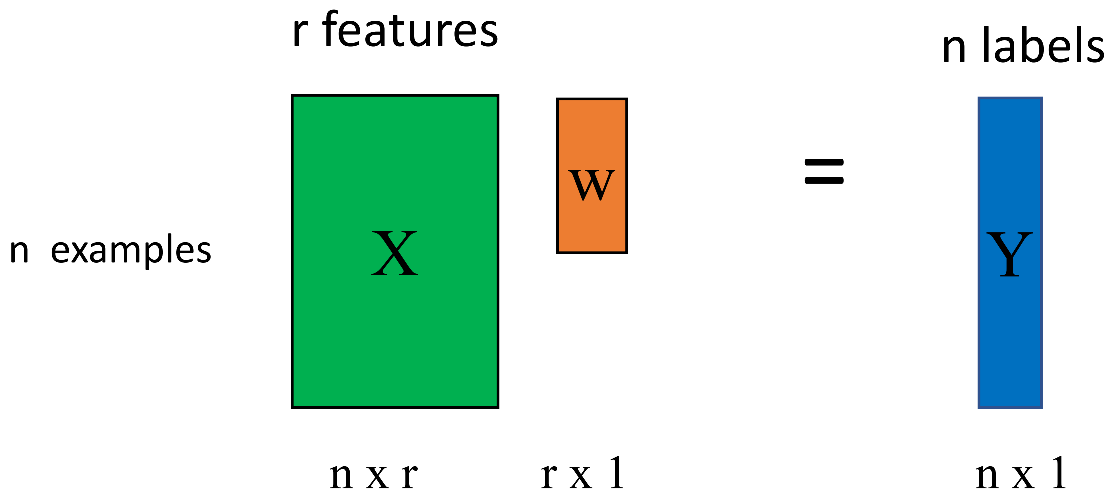


# Features and labels as matrices…



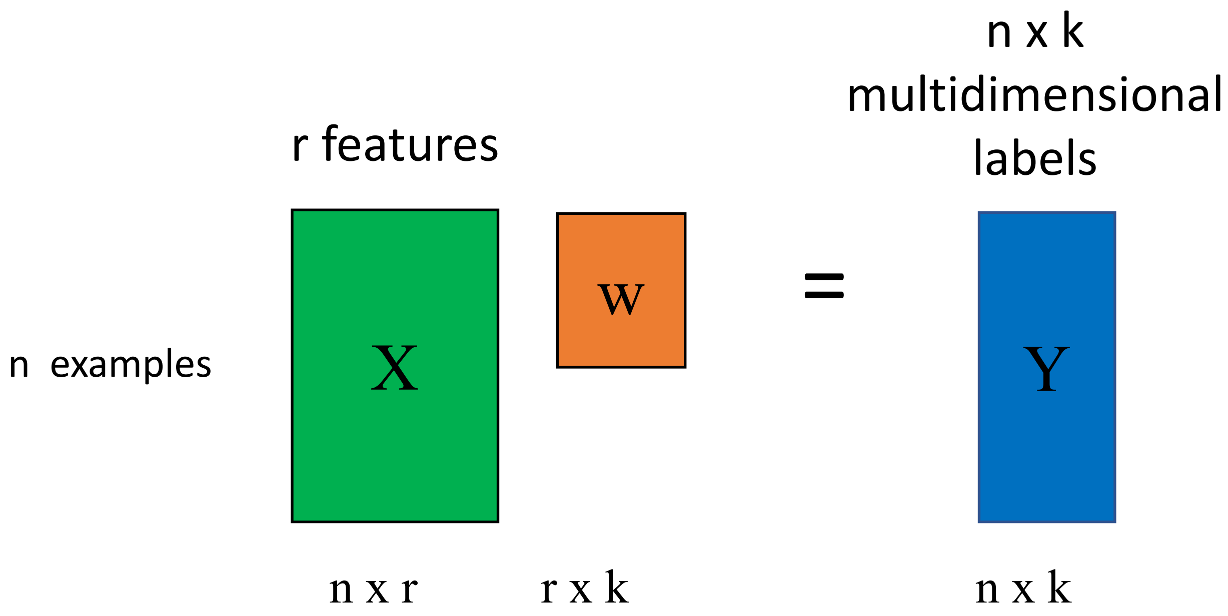

This is the form a lot of our library functions will expect.

# Features and labels as matrices…



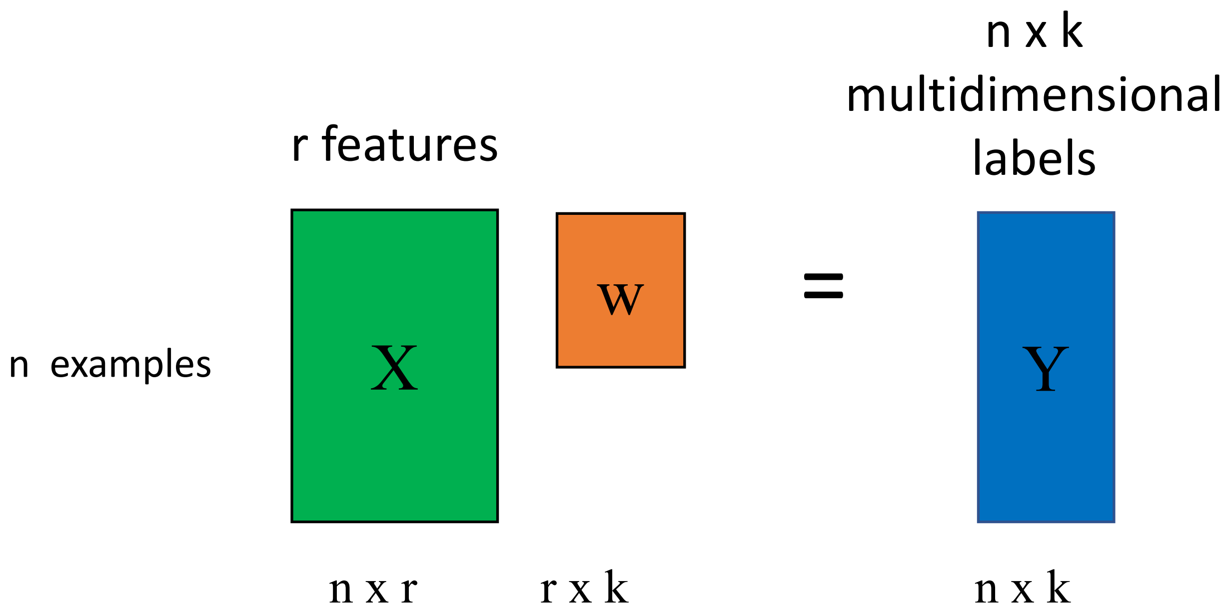

n examples / individuals / observations
r features ( in X ) (predictors)
1 outcome variable (Y)

# Features and labels as matrices…




n examples / individuals / observations
r features ( in X ) (predictors)
k outcome variables (Y)

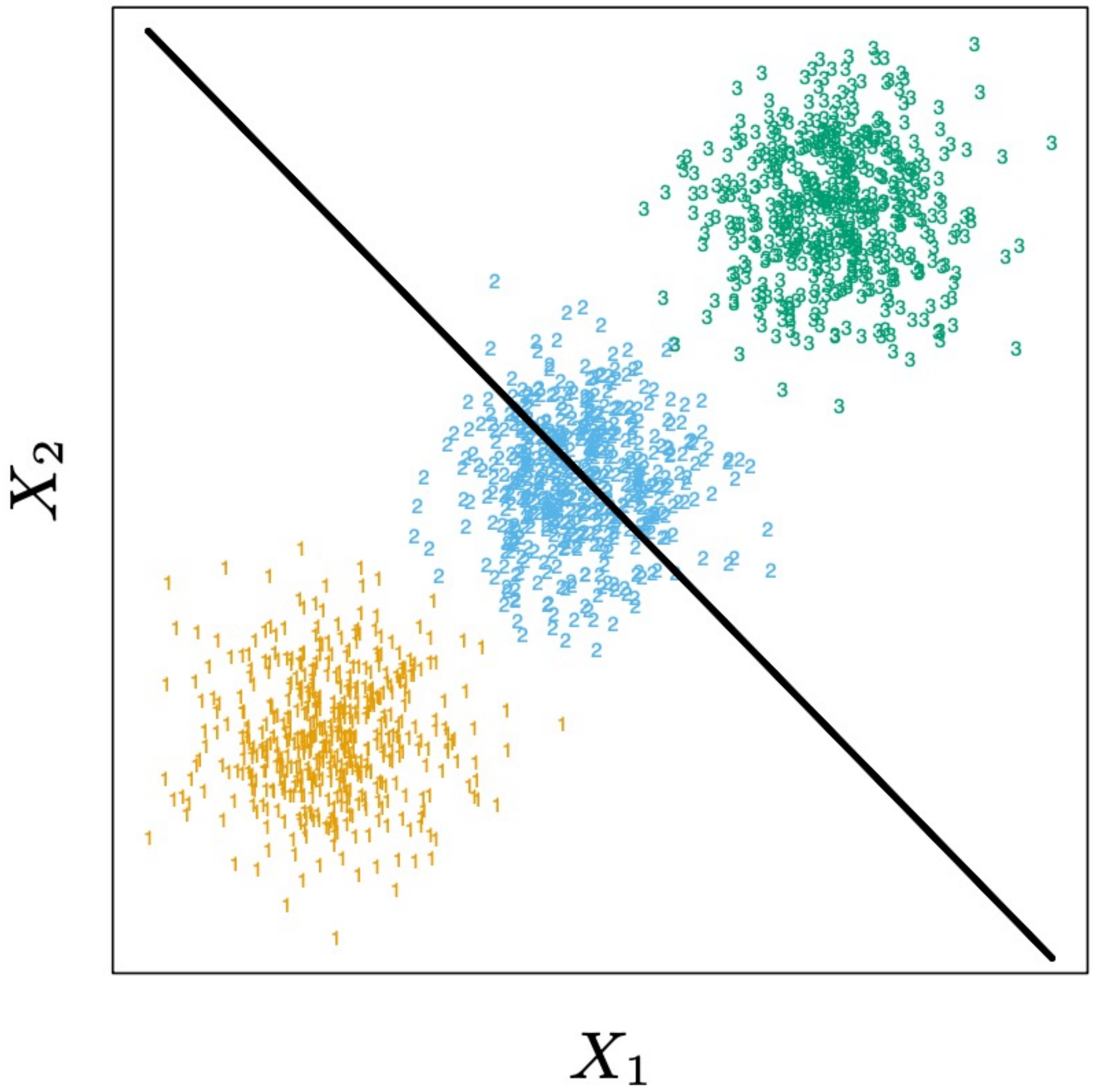
# Features and labels as matrices…



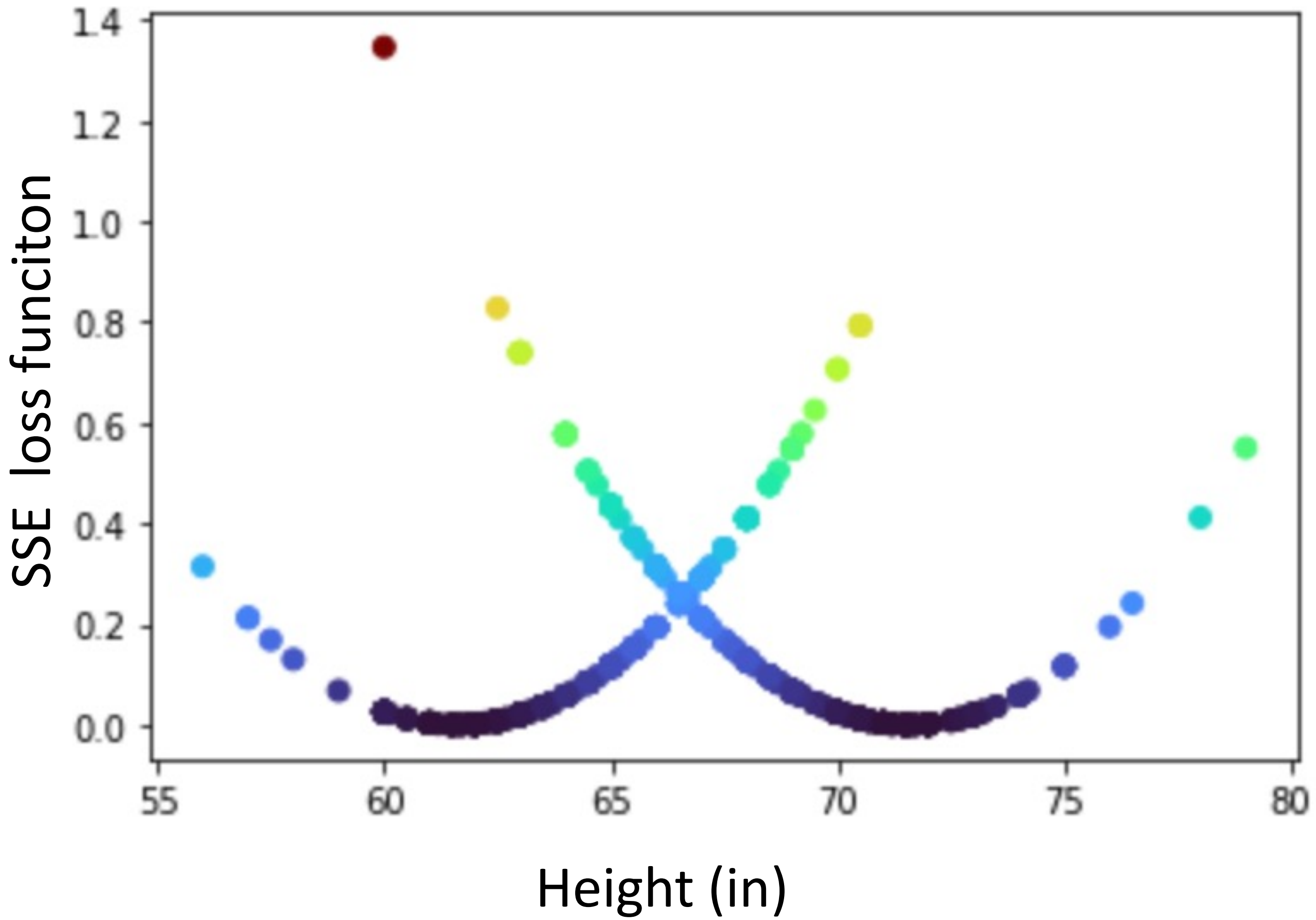

n examples / individuals / observations
r features ( in X ) (predictors)
k outcome variables (Y)

For our penguins:
n is the number of penguins ( about 300)
r is the number of features measured (4), and
k is the number of predictor variables of interest ( 3 species, 1 sex)

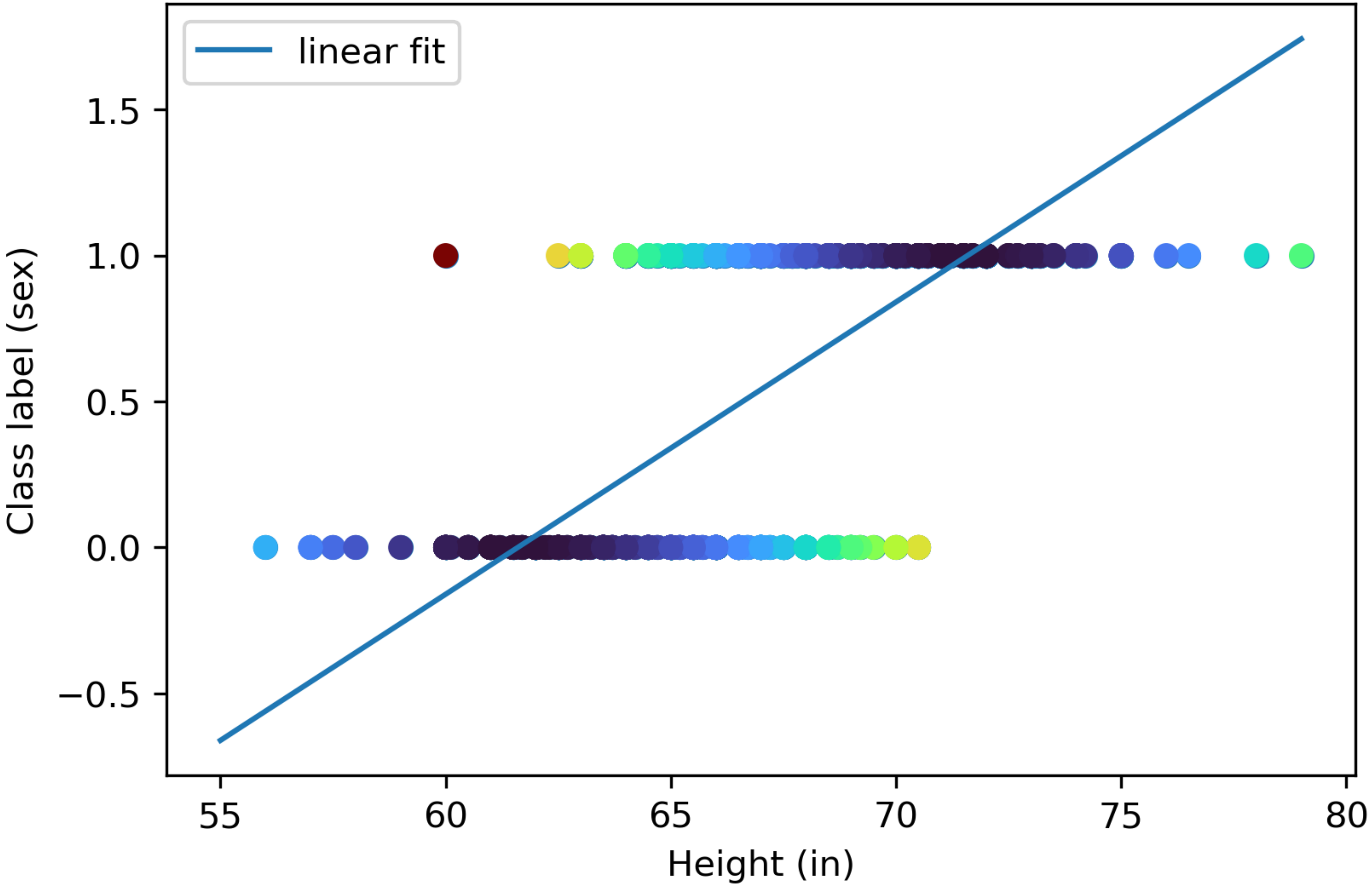
# Linear Regression



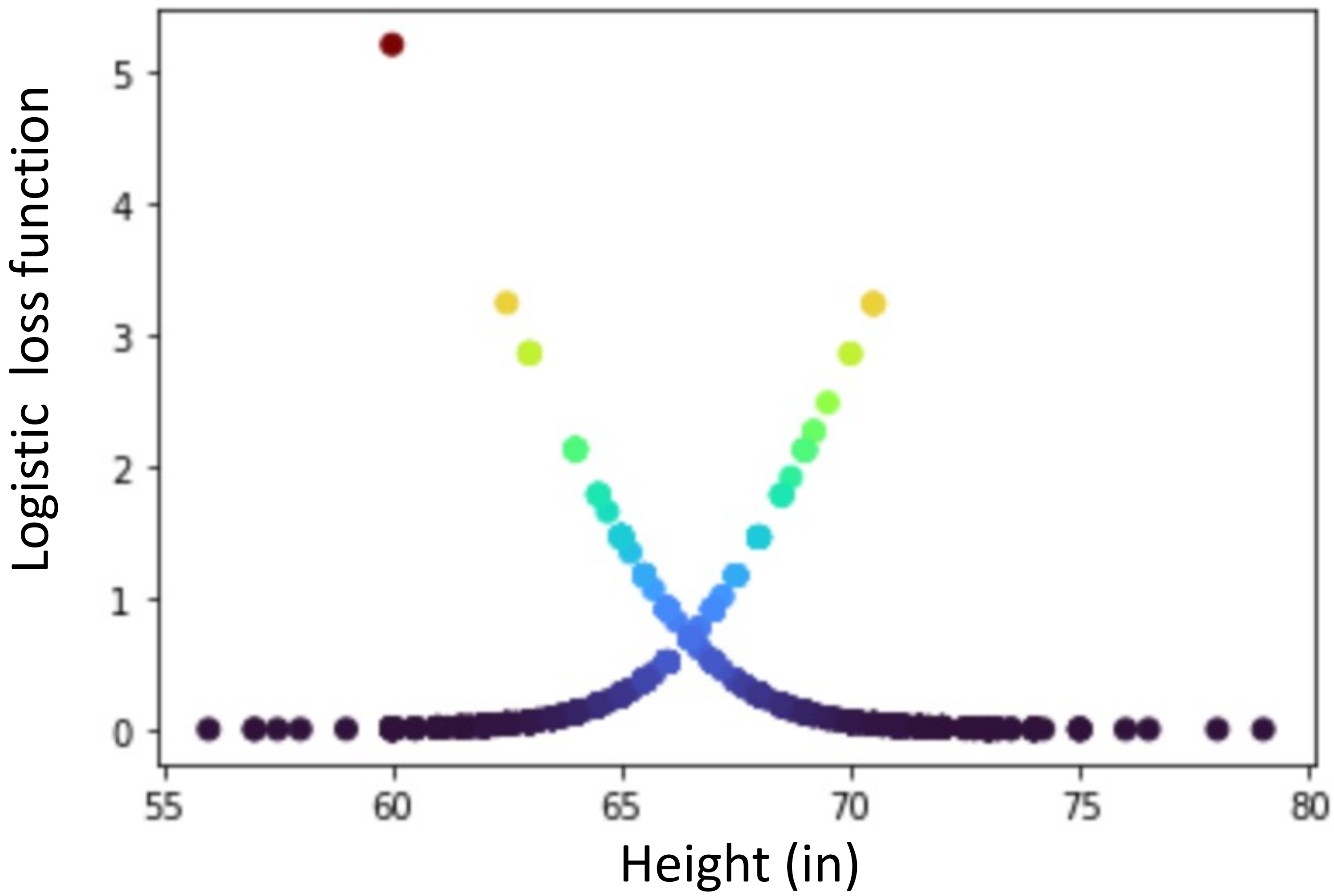
# SSE loss function for Galton height data



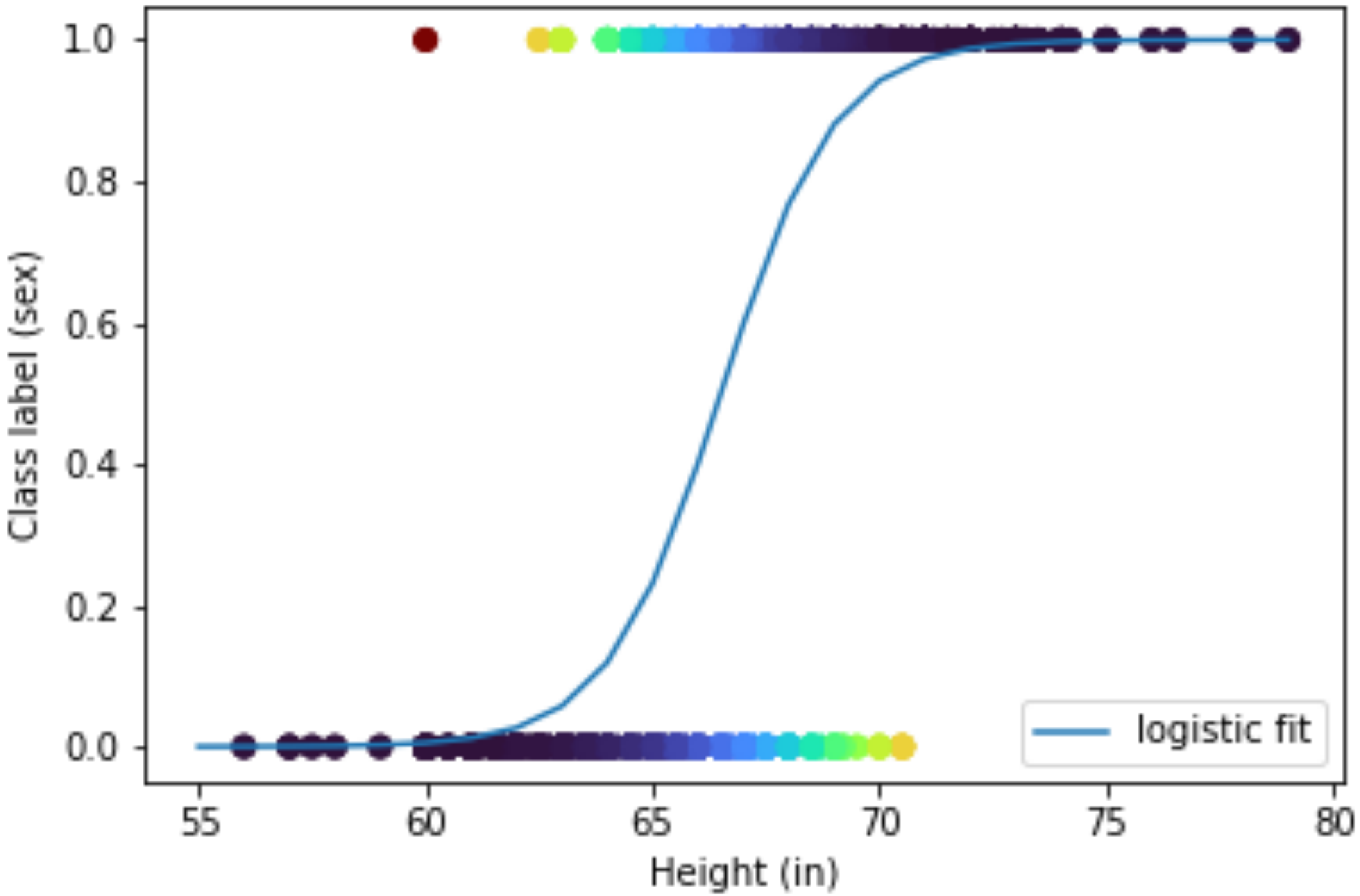
# Linear regression on class label



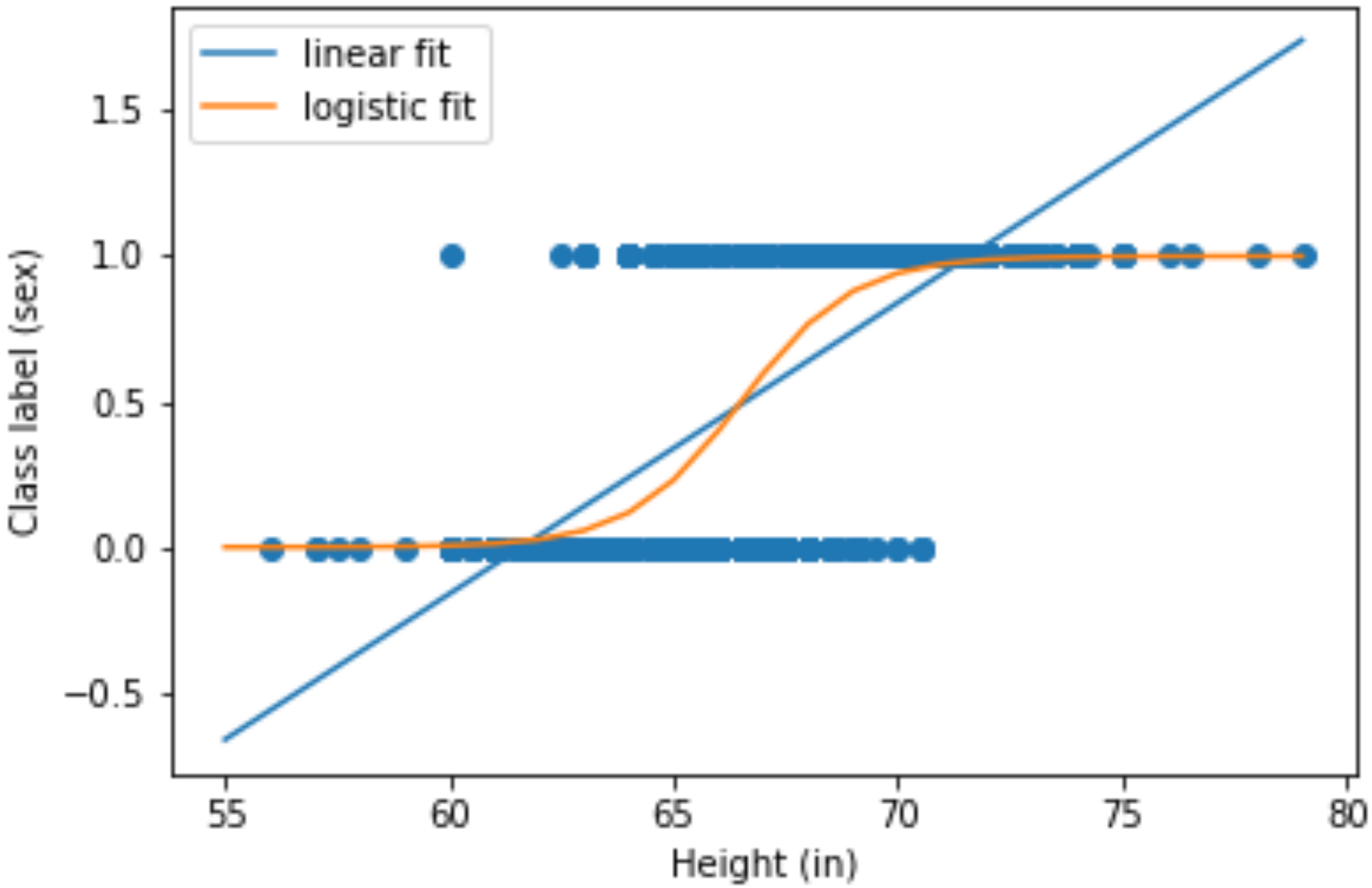
# Logistic loss function on Galton height data



# Logistic regression on class label



# Logistic regression on Galton height data



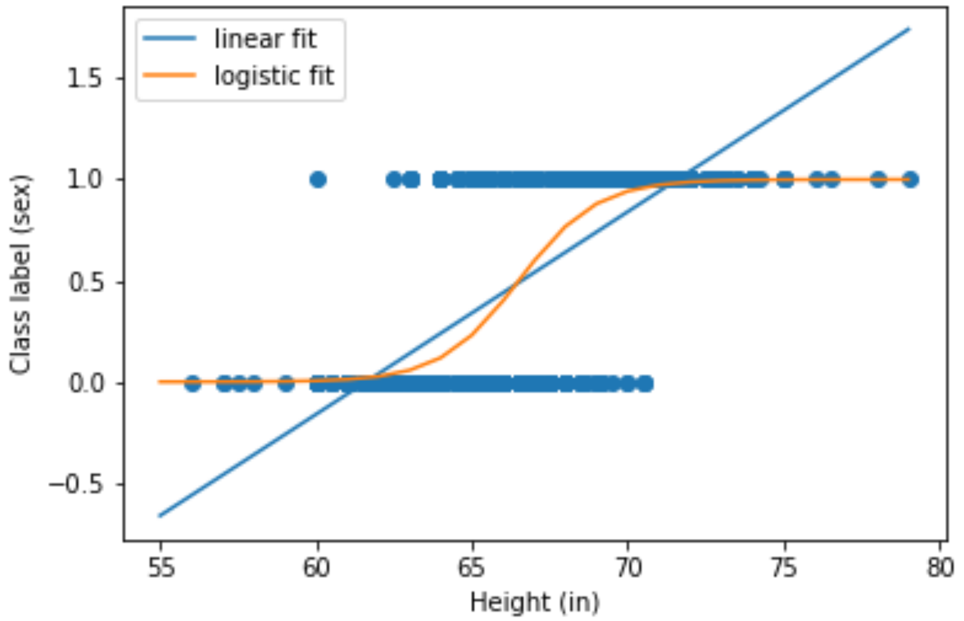
# Logistic regression - ingredients

$$\hat{y}_1 = w_{01} + \mathbf{w}_1^T \mathbf{X}$$

Linear function of X

$$p_1 = \frac{\exp(\hat{y}_1)}{1+\exp(\hat{y}_1)}$$

What are all these?

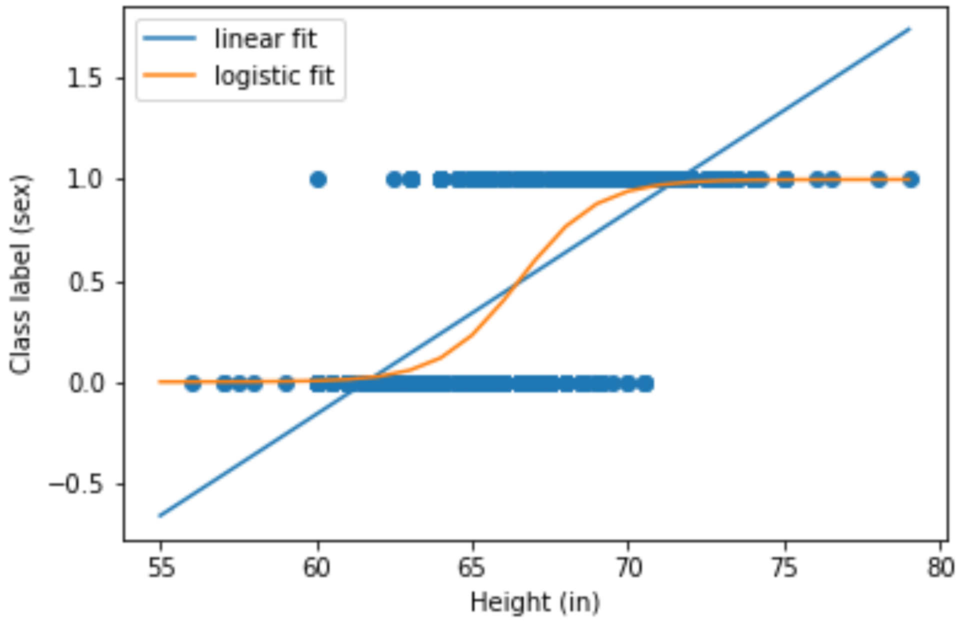


# Logistic regression - ingredients

$\hat{y}_1 = w_{01} + w_1^T X$ Linear function of X

$p_1 = \frac{\exp(\hat{y}_1)}{1+\exp(\hat{y}_1)}$ Limited to the range 0-1

*We can interpret as a probability.*




The logistic function is your function for compressing the real line to the range (0,1).

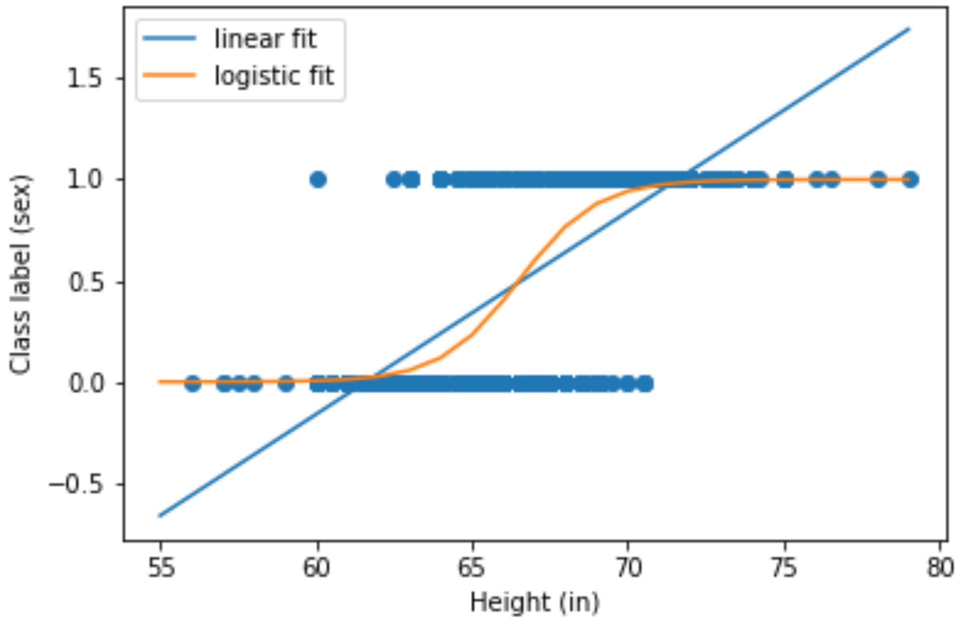
# Logistic regression - ingredients

$$\hat{y}_1 = w_{01} + w_1^T X$$

*Interpreted as a probability.*

$$p_1 = \frac{\exp(\hat{y}_1)}{1+\exp(\hat{y}_1)}$$

If y is true, log likelihood should be log(p)
If y is false, log likelihood should be log (1-p)




Logistic loss function

$$\text{LLF}(y, x) = \sum y_i \log(p_i(x)) + (1-y_i)\log(1-p_i(x))$$

Here y represents my data (the true labels), and p is my logistic predictions

# Odds interpretation of logistic function

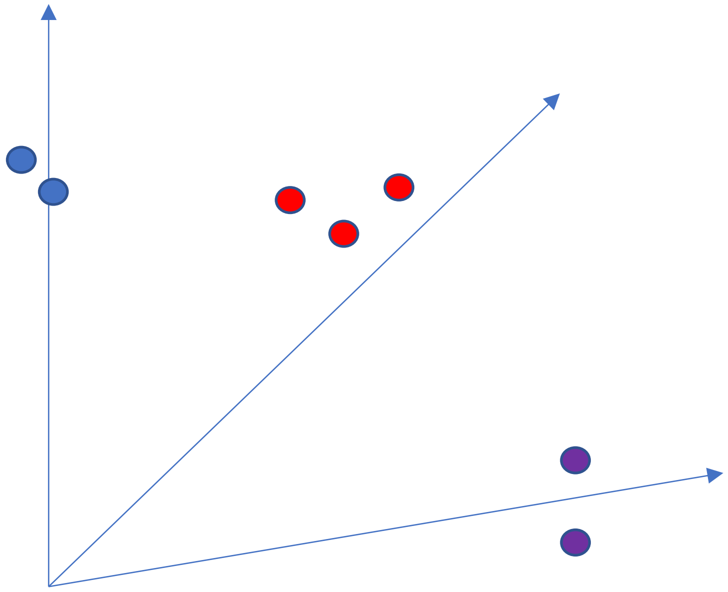
$$w_{01} \quad + \quad w_1^T X \quad = \quad \log \frac{p(x)}{1-p(x)}$$

linear function of (multivariate) X

odds in favor of x

Why might someone like accounting in log-odds terms ?

# Generalize to estimate N probabilities



$$\hat{y}_1 = w_{01} + w_1^T X$$

$$\hat{y}_2 = w_{02} + w_2^T X$$

$$\hat{y}_3 = w_{03} + w_3^T X$$

$$p_j = \frac{\exp(\hat{y}_j)}{\sum_i \exp(\hat{y}_i)}$$

# Generalize to estimate N probabilities

The class with the highest probability is chosen as the predicted class label.

Since these are linear functions of X, the decision boundaries are linear.

The loss term steers the coefficients to values that discriminate.

$$\hat{y}_1 = w_{01} + w_1^T X$$

$$\hat{y}_2 = w_{02} + w_2^T X$$

$$\hat{y}_3 = w_{03} + w_3^T X$$

$$p_j = \frac{\exp(\hat{y}_j)}{\sum_i \exp(\hat{y}_i)}$$

# Naïve Bayesian Classifier

For classification (categories A, B, C given data D)

When you have a **probability model** for observations.

(Probability model is jargon for a set of outcomes and numbers for all their probabilities)

P(D|A) , P(D|B), and P(D|C) are known.

$x \in \{1, . . . , K\}$ out of D kinds of observation

$y = \{1, . . . , C\}$ out of C classes

$\theta_c$ = (prior) probaiblities of classes

- $p(x|y = c, \theta) = \prod p(x_j |y = c, \theta_{jc})$

# Naïve Bayesian Classifier

Called naïve because the probabilities of the features are assumed to be independent.

What would correlated features generally do?

This is very much like our dice problem, and lends itself to simple bag-of-words models.

$x \in \{1, . . . , K\}$ out of D kinds of observation

$y = \{1, . . . , C\}$ out of C classes

$\theta_c$ = (prior) probaiblities of classes

- $p(x|y = c, \theta) = \prod \; p(x_j \,|y = c, \theta_c)$

# HW2

- 1 SPAM
- 2a PENGUINS linear regression, decision boundary
- 2b PENGUINS logisitic regression, decision boundary
- 2c PENGUINS quadratic decision boundary
- 2d PENGUINS classification/ confusion (any of the three above)