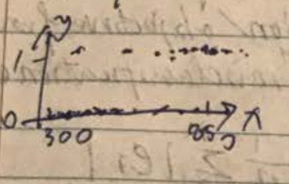
Back to the loan example where $y \in \{0, 1\}$
Let's say we have $P=1$ feature, the credit score:
$x \in [300, 850]$. So your training data looks like

$$D = \langle \vec{x}, \vec{y} \rangle = \left\langle \begin{bmatrix} 813 \\ 360 \\ 350 \\ \vdots \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ \vdots \end{bmatrix} \right\rangle \updownarrow n$$

Let's plot the data:



what is the "null model" g_0, which is the model if you didn't have any x's whatsoever.

$g_0 = \text{Mode}[\vec{y}]$
what is the simplest possible candidate space $\mathcal{H}$?
$\mathcal{H} = \{\mathbb{1}_{x \geq \theta} : \theta \in \mathcal{X}\}$ e.g. $g(x) = \mathbb{1}_{x > 600}$

2/8/2021

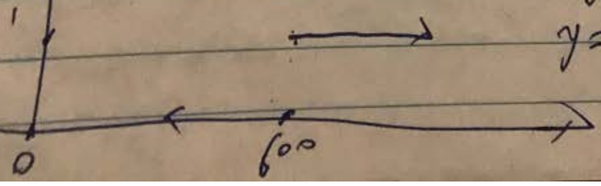
null model $g_0 = \text{Mode}[\vec{y}]$ if you don't have any x's
$g_0 = \text{Mode}[\vec{y}]$

The simplest possible candidate space $\mathcal{H}$?
$\mathcal{H} = \{\mathbb{1}_{x \geq \theta} : \theta \in \mathcal{X}\}$ e.g. $g(x) = \mathbb{1}_{x > 600}$

$\mathcal{H} = \{\mathbb{1}_{x \geq \theta} : \theta \in \Theta\}$

θ: model parameter
Θ: parameter space.

g



$\hat{y} = g(\vec{x})$
$y = g(\vec{x}) + e = \hat{y} + e$
$= \hat{y} + (y - \hat{y})$

The algorithm A produces g since g is fully specified by ~~data~~ Θ, the algorithm selects/fits Θ

Let's create an algorithm. A bad algorithm will have high estimation error.

$$\begin{array}{c|cc} & \multicolumn{2}{c}{\hat{y}} \\ y_i & 0 & 1 \\ \hline 0 & 0 & -1 \\ 1 & 1 & 0 \end{array} \quad e \; (y - \hat{y})$$

Let's define an overall error function/objective function called "misclassification error" (ME)

$$ME = \frac{1}{n}\sum_{i=1}^{1} \mathbb{1}_{g(\vec{x}_i) \neq y_i} = \frac{1}{n}\sum_{i=1}^{n} |e_i|$$

or accuracy (ACC) as.

goal is to minimize ME

$$ACC = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{g(\vec{x}_i) = y_i} = 1 - ME$$

To do so, we check every possible $\theta \in \Theta$ and keep track of the ME (theta) and then return the model with lowest ME

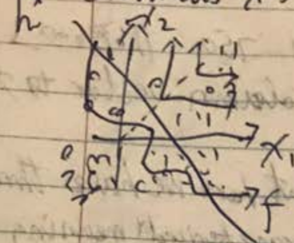
How to define parameter space? It must be finite bc we need to check (ie compute ME) each element.

grid up $[300, 850]$ e.g. $\{351, \ldots, 850\}$.

But it's more convenient to check the unique value of x

$$A \text{ produce } g(x) = \mathbb{1}_{x \geq \underset{\theta \in \text{unique}(\vec{x})}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\mathbb{1}_{x_i \geq 0} \neq y_i} \right\}}$$

Let's make an loan model with two Continuous x's i.e. x_1, x_2 ($p=2$)



$\dim[\Theta] = 2 = p$

A two dimensional threshold model extending what we have before has candidate set

$$\mathcal{H} = \{\mathbb{1}_{x_1 \geq \theta_1}, \mathbb{1}_{x_2 \geq \theta_2} : \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix} \in \Theta$$

This candidate set is very restrictive! which means we will have high misspecification error.

Another hypothesis set = all lines

$$\mathcal{H} = \{\mathbb{1}_{x_2 \geq \underbrace{a}_{\text{interception}} + \underbrace{b}_{\text{slope}} x} : a \in \mathbb{R}, b \in \mathbb{R}\}$$

The slope and interception provide you with enough "degree of freedom" to specify and seperating line. We need an algorithm to find g i.e. to specify a and b. This is a hard promblem

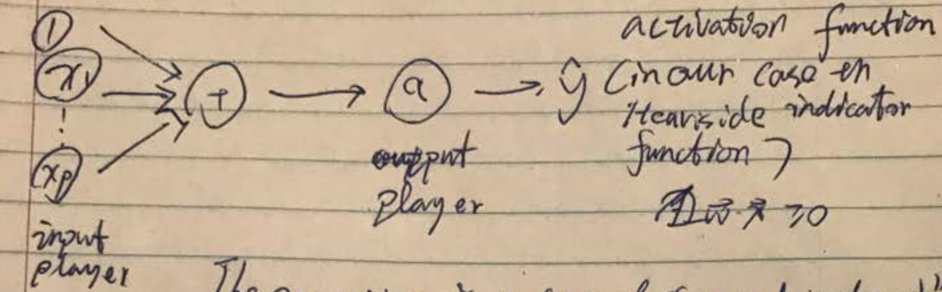
We will first reparameterize the hypothesis space to be

$$\mathcal{H} = \{\mathbb{1}_{w_0 + w_1 x_1 + w_2 x_2 \geq 0} : w_0 \in \mathbb{R}, w_1 \in \mathbb{R}, w_2 \in \mathbb{R}\}.$$

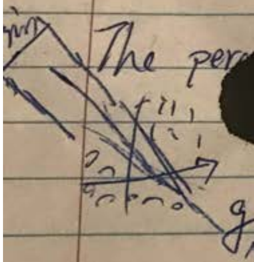
weight of the first feature, second feature.

intercept term, ~~bias~~s bias

The perception is proved to converged for linearly separable data set but for non-linearly separable datasets, anything can happen so it may fail



The perception is a type of "neural network" model. So are deep learning models. They're called neurons since they kind of act like neurons!



The per[illegible]tion has infinite solutions.
[illegible]l possible ~~values~~ solutions which vary based on starting values.

best model divides the margin evenly