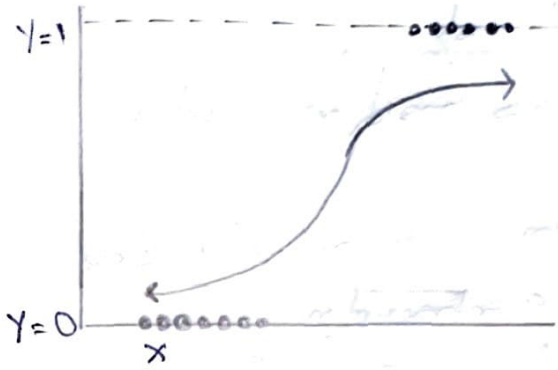
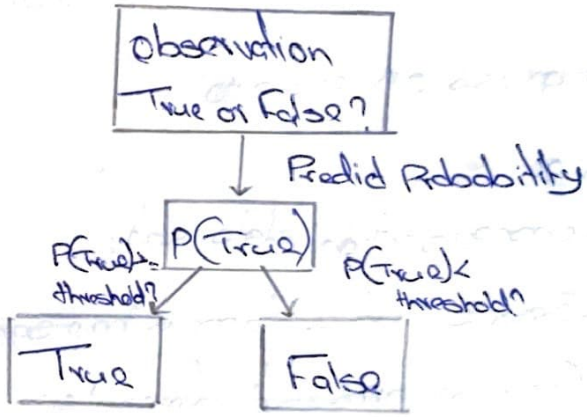
⊙ Logistic regression



- Predicts the probability of an observation belonging to one of two classes
- In ML, it refers to a binary classifier based on statistical logistic regression



Binary Classification

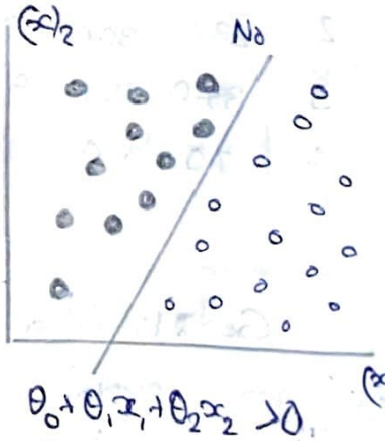
- When is logistic regression a good choice?

  - If the data is linearly separable, the decision boundary of logistic regression is a line, a plane, or a hyperplane

Example: $\theta_0 + \theta_1 x_1 + \theta_2 x_2 > 0$

  - To understand the impact of an independent feature.

Example: Select features based on model coefficient size or weights



- Logistic regression application
  - Predicting heart attack risk
  - Diagnosing patients based on a set of characteristics
  - Predicting whether a customer will purchase a product or half a subscription
  - Predicting product failure probability
  - Predicting mortgage default likelihood

- Logistic regression example

- Scenario

Telecommunication data set:

- Services that customers have signed up for
- Customer account information
- Demographic information
- Customers who have left in the last month

---

① Logistic regression example

Independent variables

Dependent variable

	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	Yes
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	Yes
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	NO
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	NO
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	?

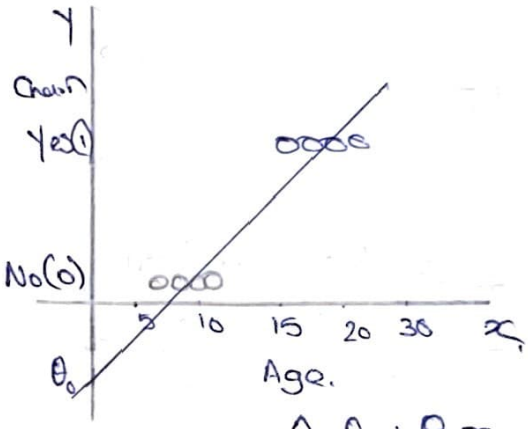
Binary Variable

Goal: Build a model to predict the class of each customer by considered the predicted probability that the customer will churn

- Predicting churn using linear regression

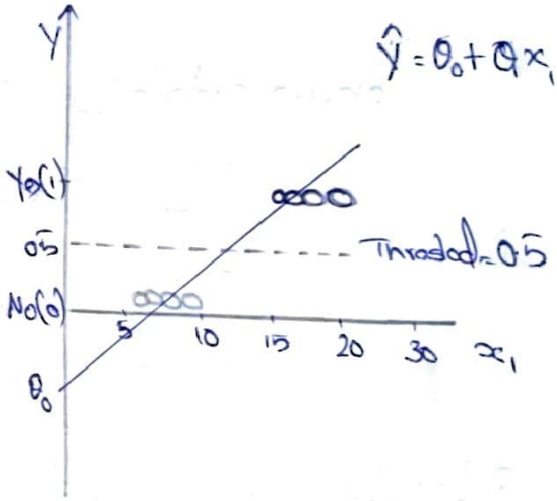
Y = Actual churn classes

$\hat{Y}$ = Predicted churn classes



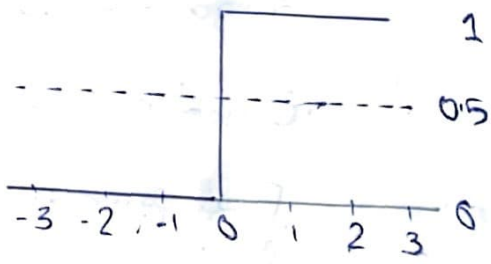
$\hat{Y} = \theta_0 + \theta_1 x_1$

$$\hat{Y} \rightarrow \begin{cases} 0 & \text{if } \hat{y} < 0.5 \\ 1 & \text{if } \hat{y} \geq 0.5 \end{cases}$$



- challenges of linear regression

$$\hat{Y} \rightarrow \begin{cases} 0 & \text{if } \hat{y} < 0.5 \\ 1 & \text{if } \hat{y} \geq 0.5 \end{cases}$$

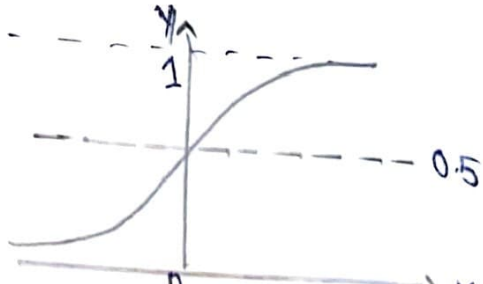


step function

- Towards probability

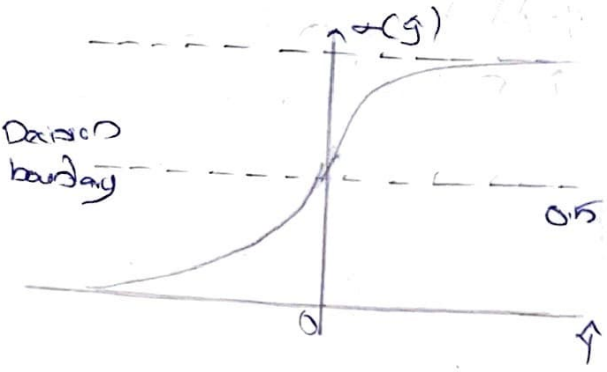
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Sigmoid function



- Probability to class prediction

$\hat{P} = \sigma(\vartheta) = \frac{1}{1+e^{-\vartheta}}$



Probability that class is 1

$\sigma(\vartheta) \rightarrow \begin{cases} 0 & \text{if } \sigma(\vartheta) < 0.5 \\ 1 & \text{if } \sigma(\vartheta) \geq 0.5 \end{cases}$

- Predicting customer churn

Churn probability: $P(Y=1|X)$

$P(Y=0|X) = 1 - P(Y=1|X)$

$P(\text{Churn} | \text{Income}, \text{age}) = 0.8$

$P(\text{Stay} | \text{Income}, \text{age}) = 1 - 0.8 = 0.2$

⊙ Logistic regression training

- Identify parameters that map input features to target outcomes
- Objective: Predict classes with minimal error
- Find parameters / that minimize cost function

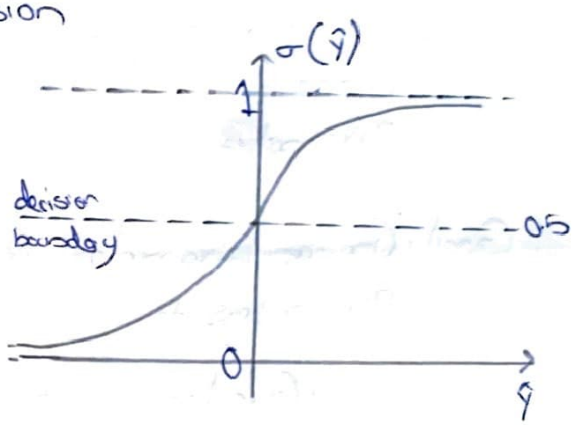
- Choose a set of parameters or θ
- Predict probability that class = 1
- Calculate prediction error (cost function)
- Update θ to reduce prediction error
- Repeat until:
  - Reach small log-loss value or
  - Targeted number of iterations

Optimal logistic regression

$$\hat{y} = \theta_0 + \theta_1 x_1$$

$$\hat{p} = \sigma(\hat{y}) = \frac{1}{1+e^{-\hat{y}}}$$

Probability that class of y is 1



$$\sigma(\hat{y}) \rightarrow \begin{cases} 0 & \text{if } \sigma(\hat{y}) < 0.5 \\ 1 & \text{if } \sigma(\hat{y}) \geq 0.5 \end{cases}$$

- Cost function or log-loss needs to be minimized
- Measures how well $(\hat{p}_i)$ matches y_i

$$\text{log-loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{p}_i) + (1-y_i)\log(1-\hat{p}_i)$$

- Confident & correct: Predicted probability of class 1 is high and correct $\Rightarrow$ log-loss is small
- Confident & incorrect: Predicted probability of class 0 is high and incorrect $\Rightarrow$ log-loss is large.

- Minimizing cost function with gradient descent
  - Stop training when log-loss is satisfactory
  - Use Gradient descent

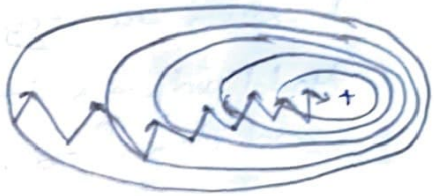
- Gradient descent
  - Iterative approach to finding the minimum of a function
  - Adjusts parameter values using log-loss derivative
  - Depends on a specified learning rate
  - Controls how far it's allowed to step the parameters

Goal: Change parameter values and find path to optimal parameters to minimize the cost function
  - Gradient descent path
  - Best parameters at minimum of cost function

- Calculated over the entire data set
- Large data set = slow descent
- Converge less likely as steps too big to settle minima
- Gradient can be approximated using a random subset

⊙ Stochastic gradient descent (SGD)

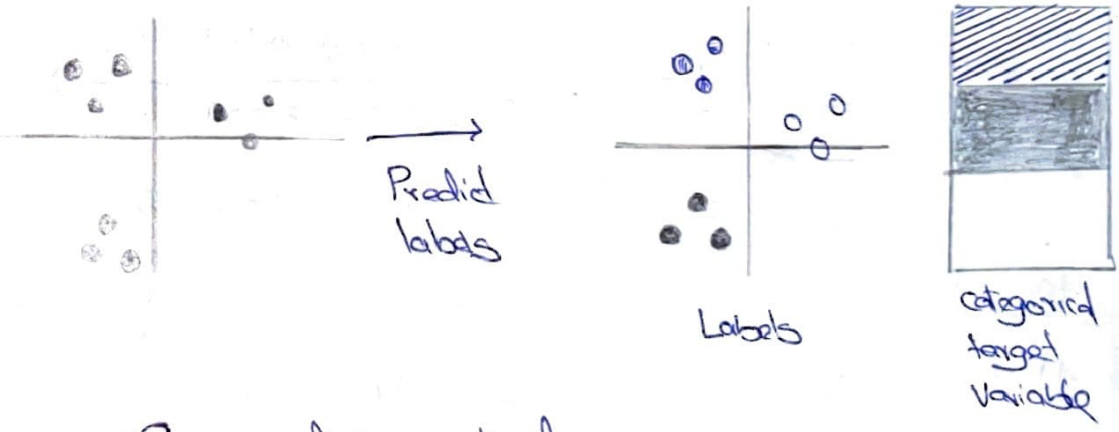
- Variation of the gradient descent
- Uses a random data subset and scales well
- Likely to overlook local and find global minima
- Converges quickly towards a global minimum



- Convergence can be improved by:
  - Decreasing learning rate
  - Gradually increasing sample size

Module 3

⊙ Classification



- Supervised ML method
- Uses fully trained models to predict labels on new data
- Labels from a categorical variable with discrete values.