# Classification

Richard Lui

# Logistic Regression

# Classification: Purchase or Not Purchase?

Features (Age, Salary (1000))

Label (Purchase?)

| Customer ID | Age | Salary (1000) | Purchase? |
|---|---|---|---|
| 1 | 18 | 8 | 0 |
| 2 | 25 | 40 | 0 |
| 3 | 30 | 30 | 0 |
| 4 | 40 | 10 | 0 |
| 5 | 45 | 33 | 1 |
| 6 | 41 | 48 | 1 |
| 7 | 25 | 60 | 1 |
| 8 | 35 | 10 | 1 |
| 9 | 15 | 0 | 0 |
| 10 | 40 | 70 | 1 |

Not buy

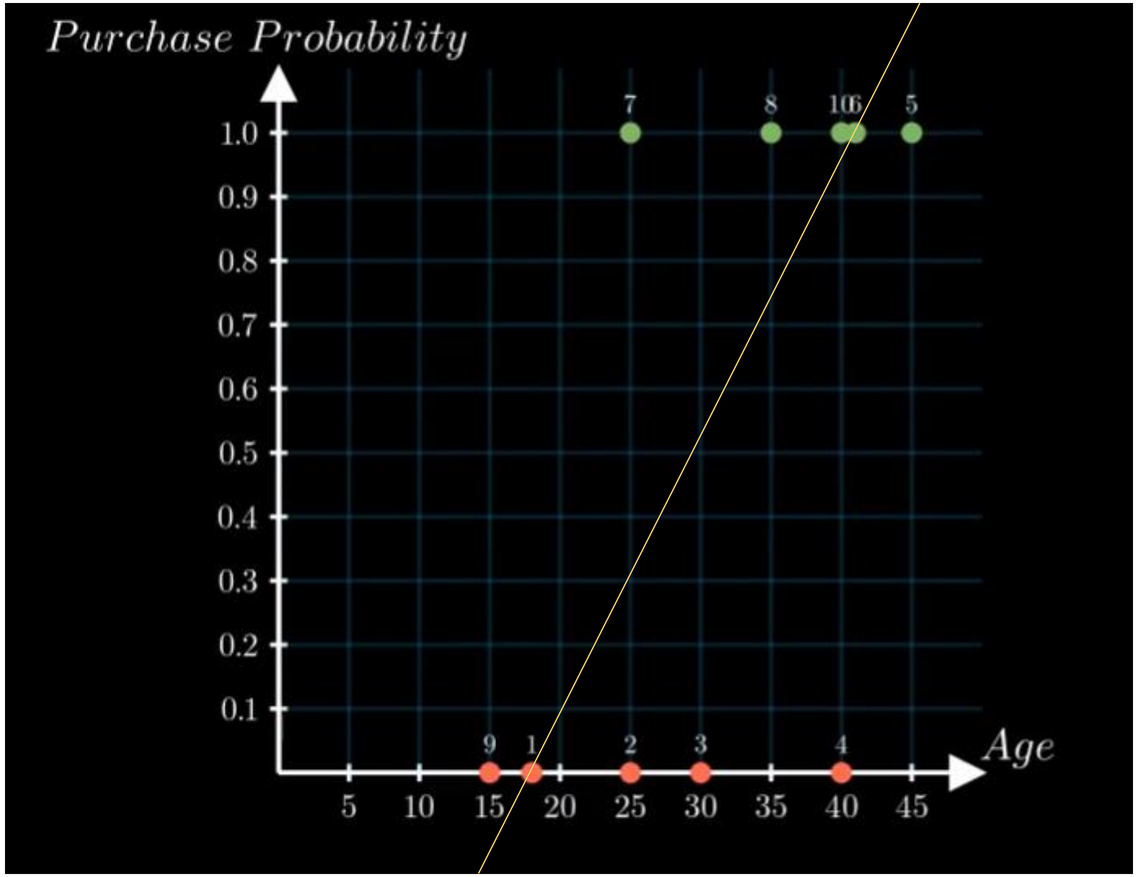
Buy

Let's consider models with only one predictor: age.

# Predict probability of purchase

$$p = b + m\,Age$$

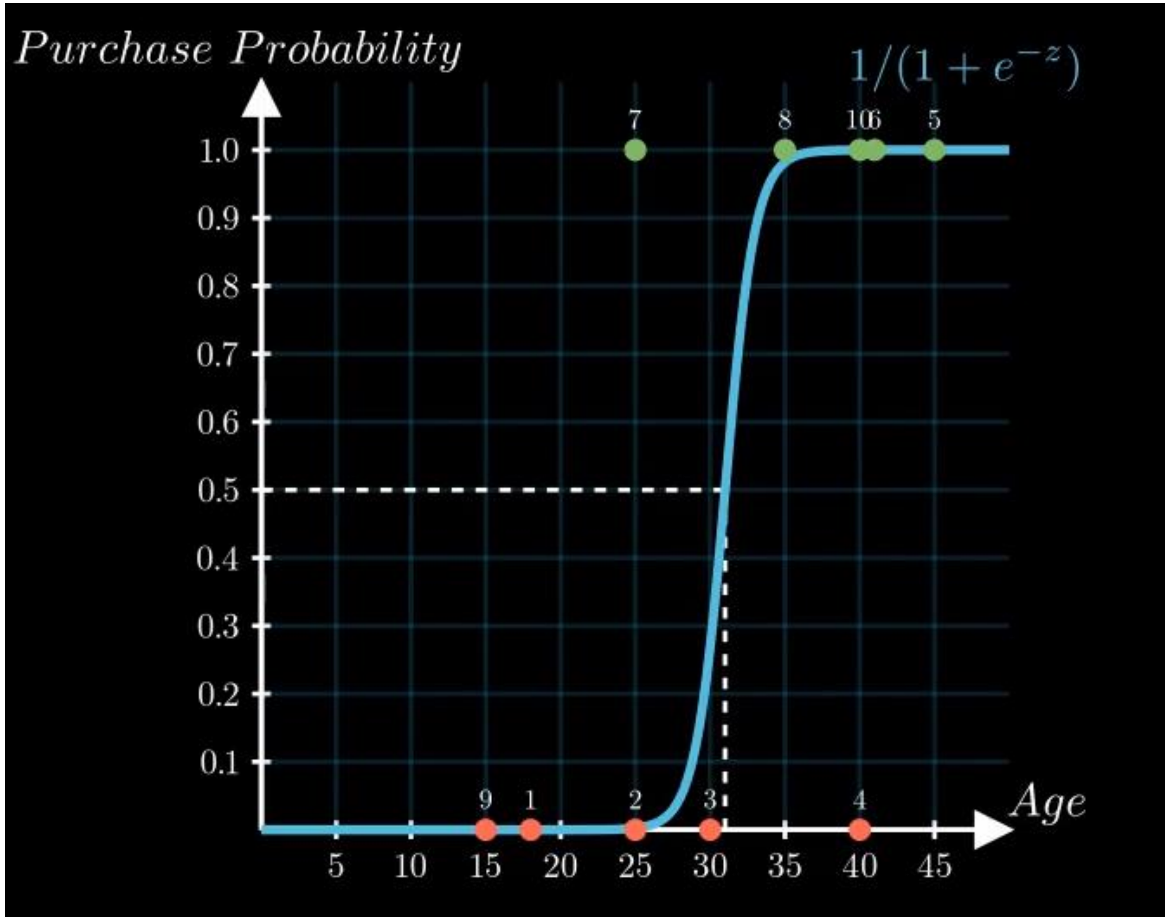
- We want to predict the probability that a customer (with certain age) will purchase.
- Let $p$ be the probability that a customer will purchase the product
- Can we use linear regression to predict the probability $p$ a customer will purchase the product




**Probabilities should be bounded by $0 \leq p \leq 1$!**

# Fitting an S-shape curve

- We want a function $p = f(Age)$ such that
  - $p$ must always be positive ($p \geq 0$)
  - $p$ must be less than 1 ($p \leq 1$)

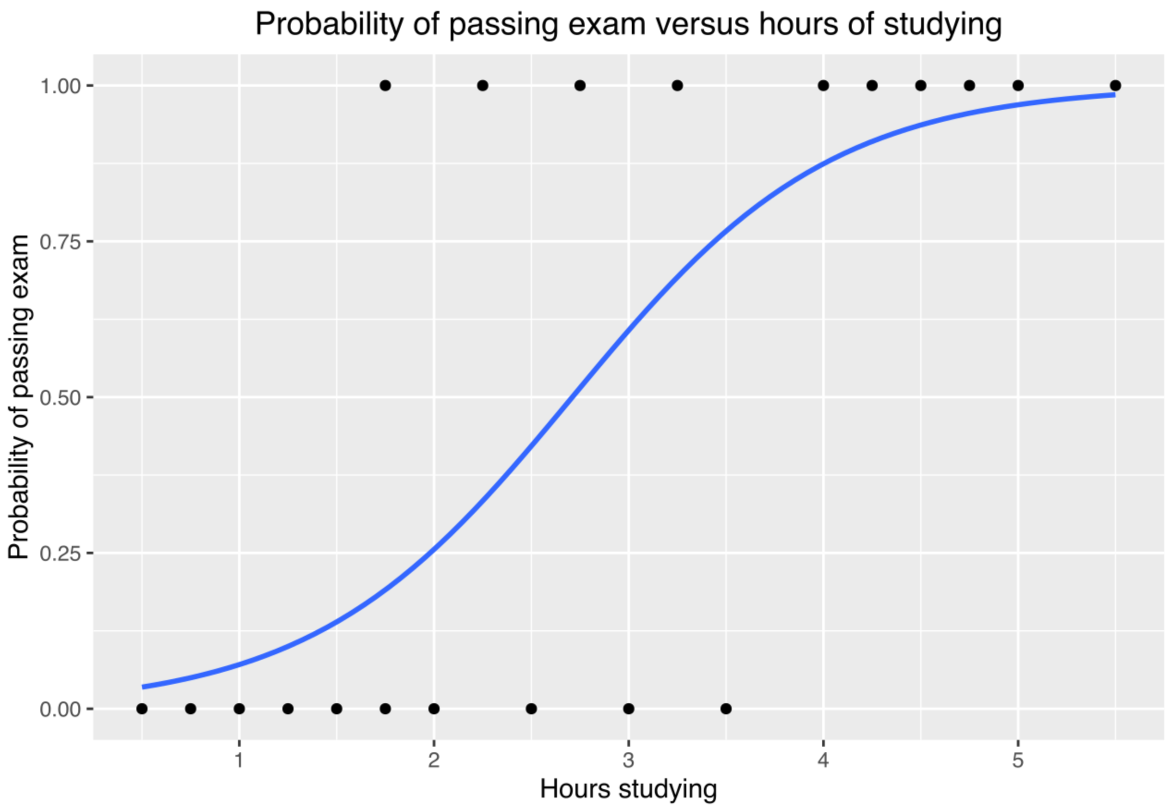


# Logistic Function

- Technique originated from statistics
- Logistic function/sigmoid function
  - "S" shape

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

*e is a special mathematic constant (~2.71828)*




https://en.wikipedia.org/wiki/Logistic_regression

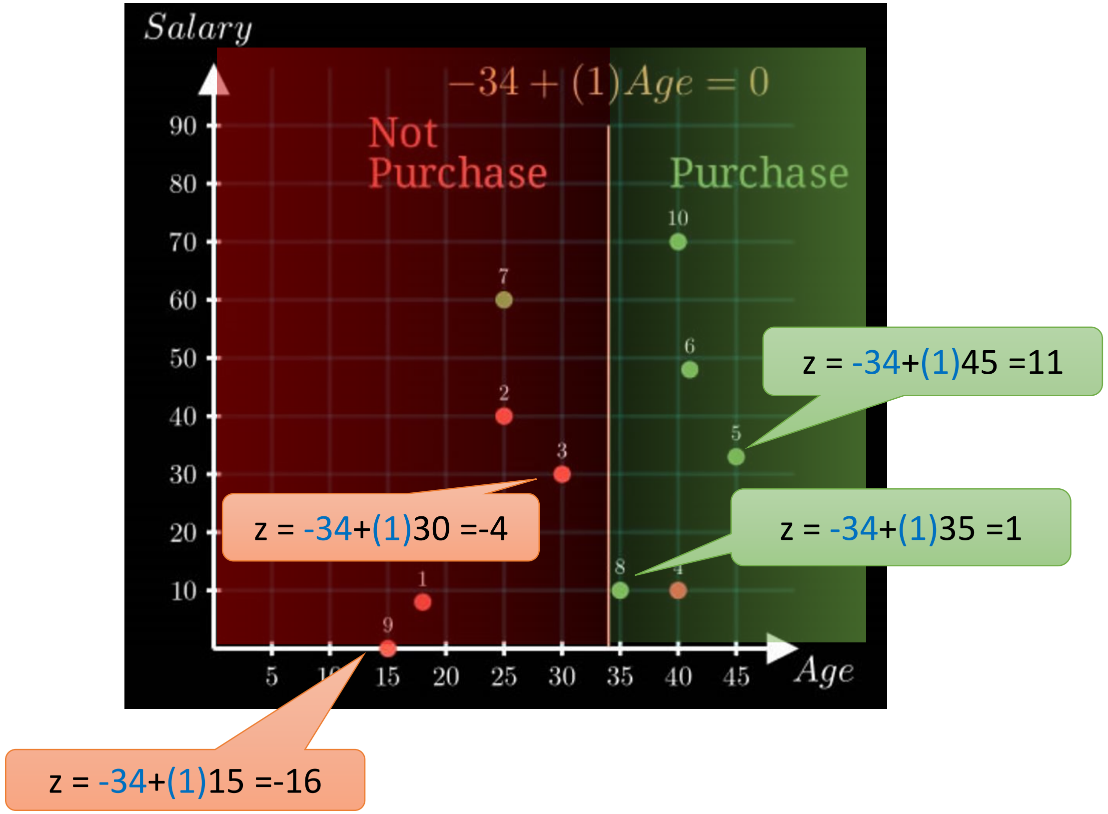
# Decision boundary

Our model's parameters

$$z = b + w\,(Age)$$

- Suppose $b = -34$ and $w = 1$.
- Corresponds to the line (decision boundary)

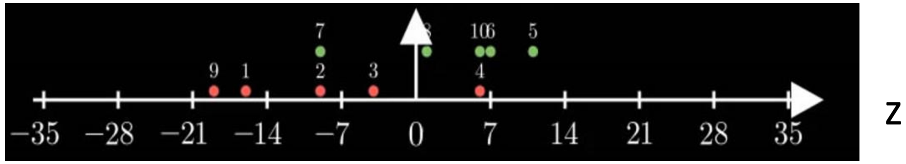
$$-34 + (1)Age = 0$$

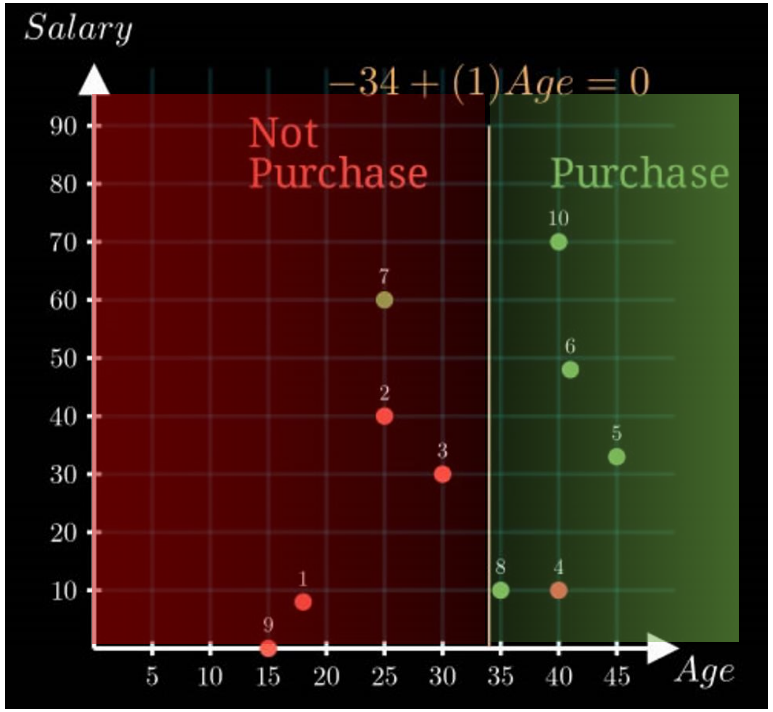


# Decision boundary

$$z = b + w\ (Age)$$

- The further the data points are from the left of the line, the less likely they will purchase
- The further the data points are from the right of  the line, the less likely they will purchase
- At the decision boundary (z=0)
  - 50% purchase, 50% not purchase




As z decreases, p decreases

As z increases, p increases

z=0, p =0.5

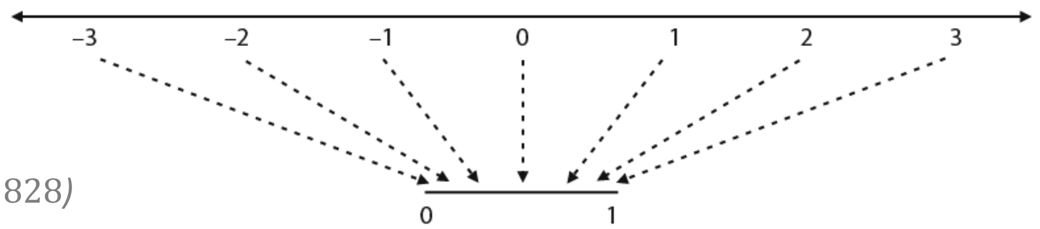
# How to map z to a probability?

- z ranges from $-\infty$ to $+\infty$
- We want a function to map z to the range of probability (between 0 and 1)

$$z = b + w\ (Age)$$

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

*e is a special mathematic constant (~2.71828)*




If z =-100 ,

$$\sigma(\mathrm{z}) = \frac{1}{1+2.71828^{-(-100)}} \approx 0$$

~0% chance of purchase

If z =0 ,

$$\sigma(\mathrm{z}) = \frac{1}{1+2.71828^{-(0)}} \approx 0.5$$

50-50 chance of purchase

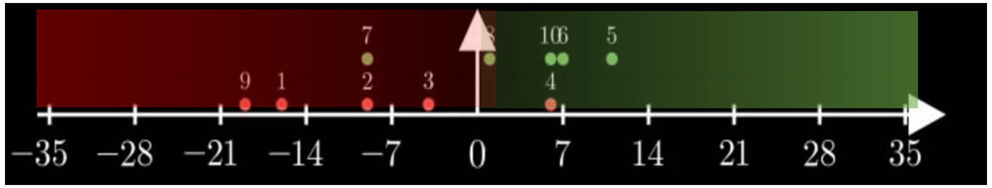
If z =100 ,

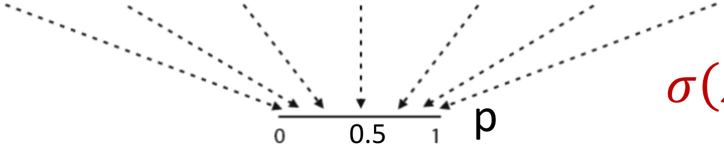
$$\sigma(\mathrm{z}) = \frac{1}{1+2.71828^{-(100)}} \approx 1$$

~ 100% chance of purchase

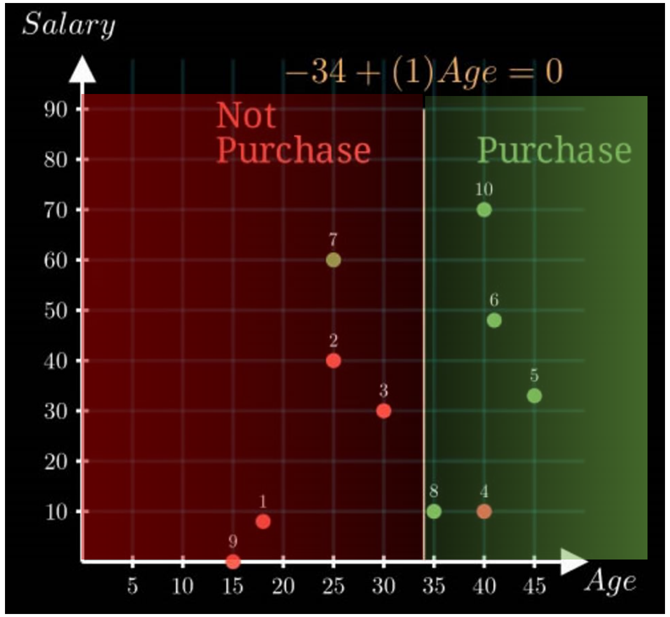
| Customer ID | Age | Purchase? |
|---|---|---|
| 1 | 18 | 0 |
| 2 | 25 | 0 |
| 3 | 30 | 0 |
| 4 | 40 | 0 |
| 5 | 45 | 1 |
| 6 | 41 | 1 |
| 7 | 25 | 1 |
| 8 | 35 | 1 |
| 9 | 15 | 0 |
| 10 | 40 | 1 |

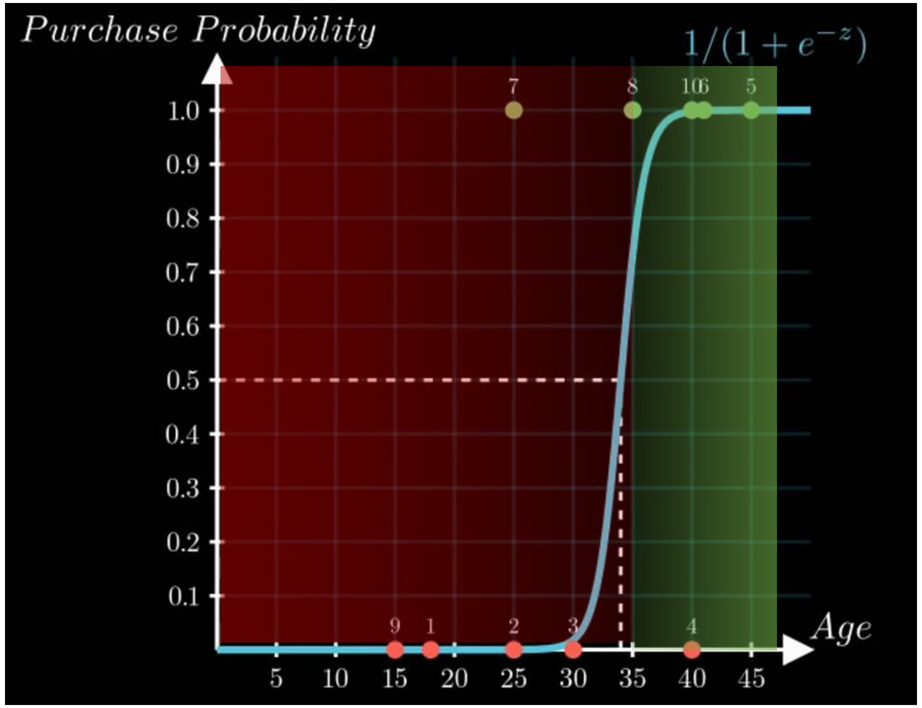
$$z = -34 + (1)\,Age$$



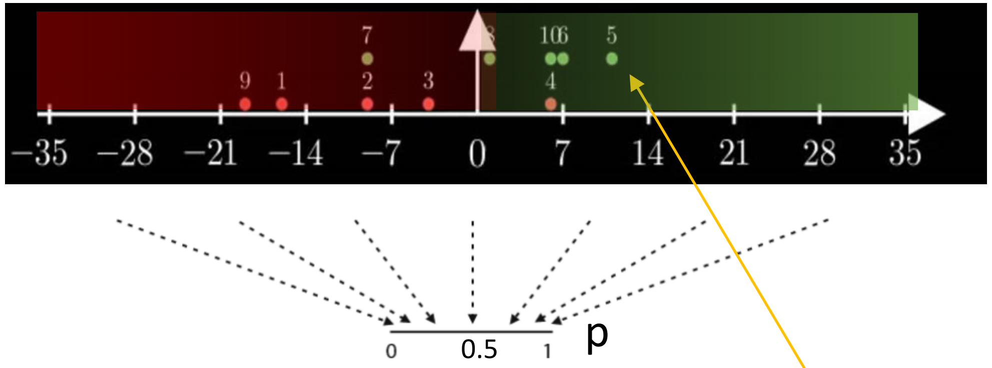

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

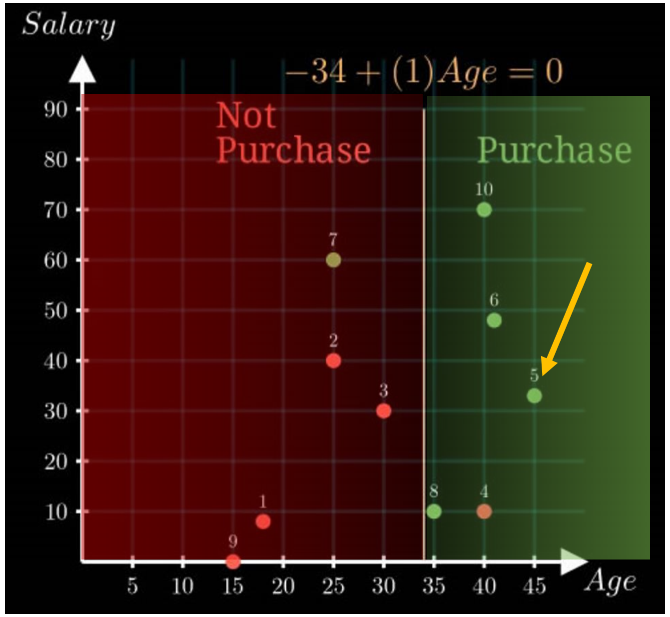


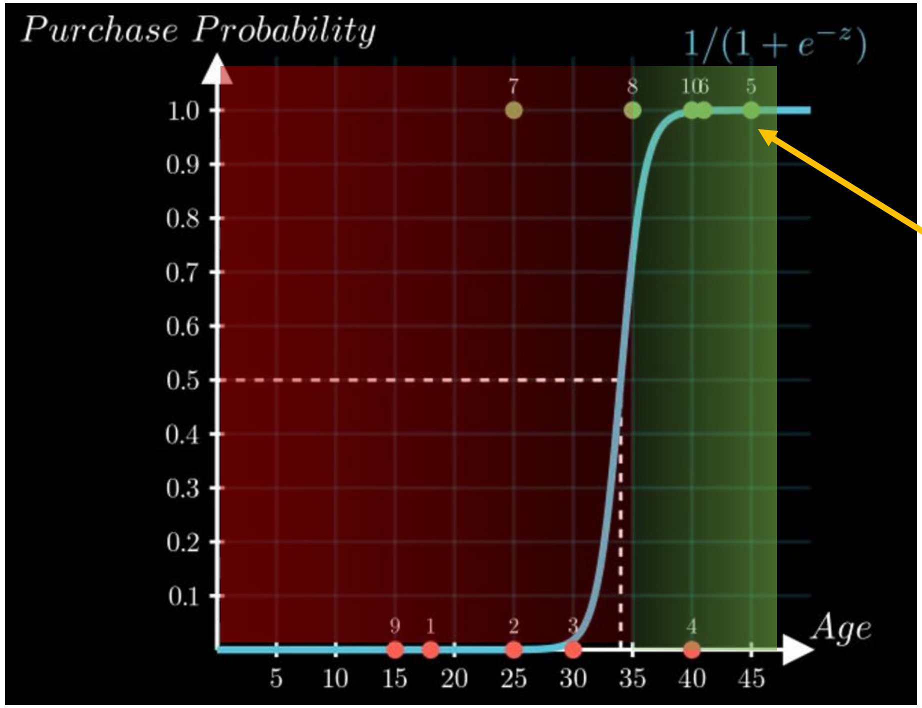



| Customer ID | Age | Purchase? |
|---|---|---|
| 1 | 18 | 0 |
| 2 | 25 | 0 |
| 3 | 30 | 0 |
| 4 | 40 | 0 |
| **5** | **45** | **1** |
| 6 | 41 | 1 |
| 7 | 25 | 1 |
| 8 | 35 | 1 |
| 9 | 15 | 0 |
| 10 | 40 | 1 |

$$z = -34 + (1)\ Age$$




$$p = \frac{1}{1+e^{-z}}$$

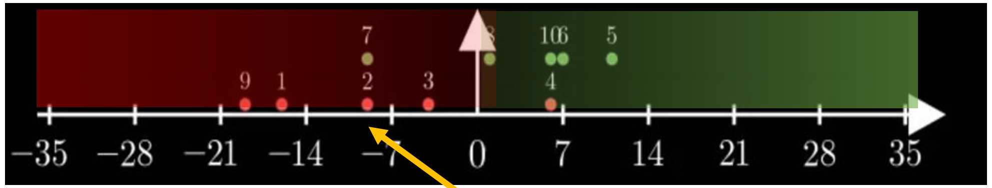
$$= \frac{1}{1+e^{-(-34+1(45))}}$$

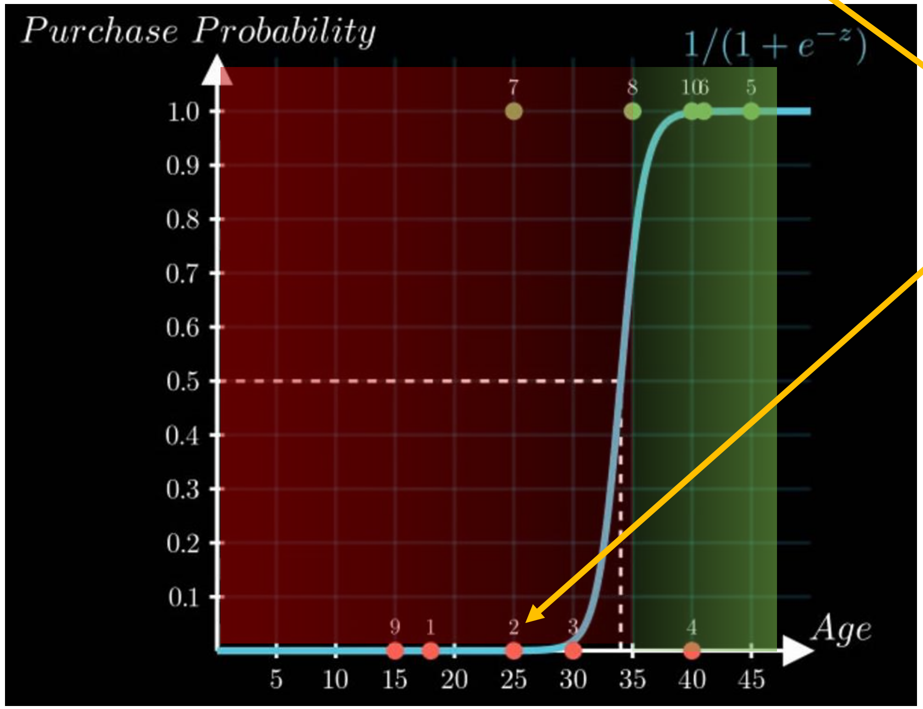
$$= 0.99998$$

This customer is very likely to purchase!

| Customer ID | Age | Purchase? |
|---|---|---|
| 1 | 18 | 0 |
| 2 | 25 | 0 |
| 3 | 30 | 0 |
| 4 | 40 | 0 |
| 5 | 45 | 1 |
| 6 | 41 | 1 |
| 7 | 25 | 1 |
| 8 | 35 | 1 |
| 9 | 15 | 0 |
| 10 | 40 | 1 |

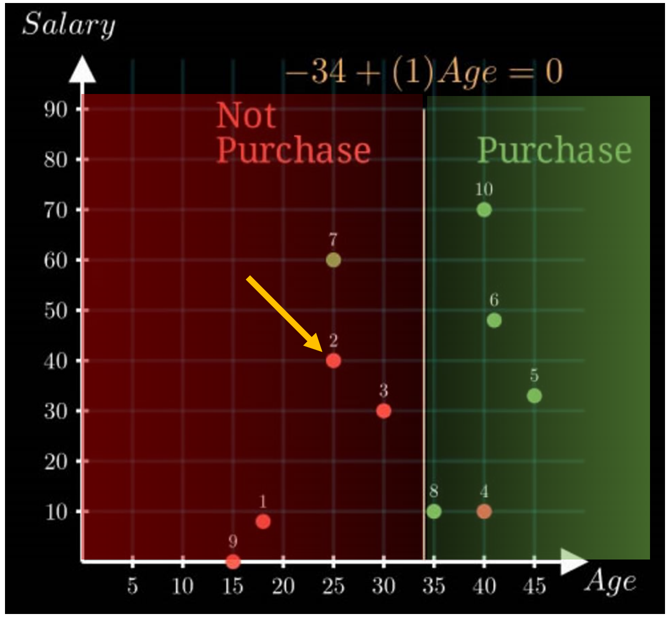
$$z = -34 + (1)\ Age$$



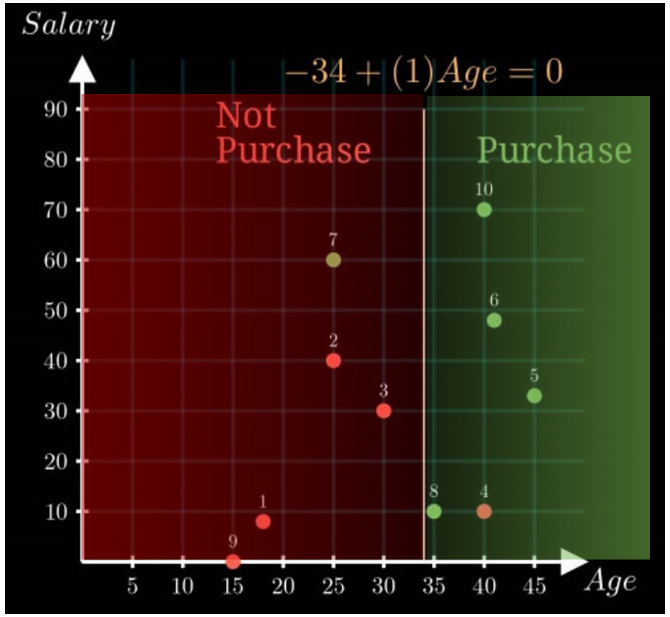



$$p = \frac{1}{1+e^{-z}}$$
$$= \frac{1}{1+e^{-(-34+1(25))}}$$
$$= 0.000123$$

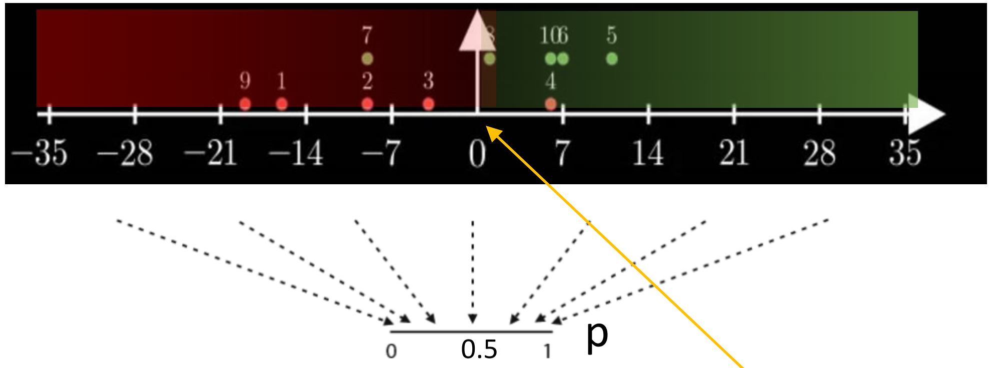
This customer is very unlikely to purchase!

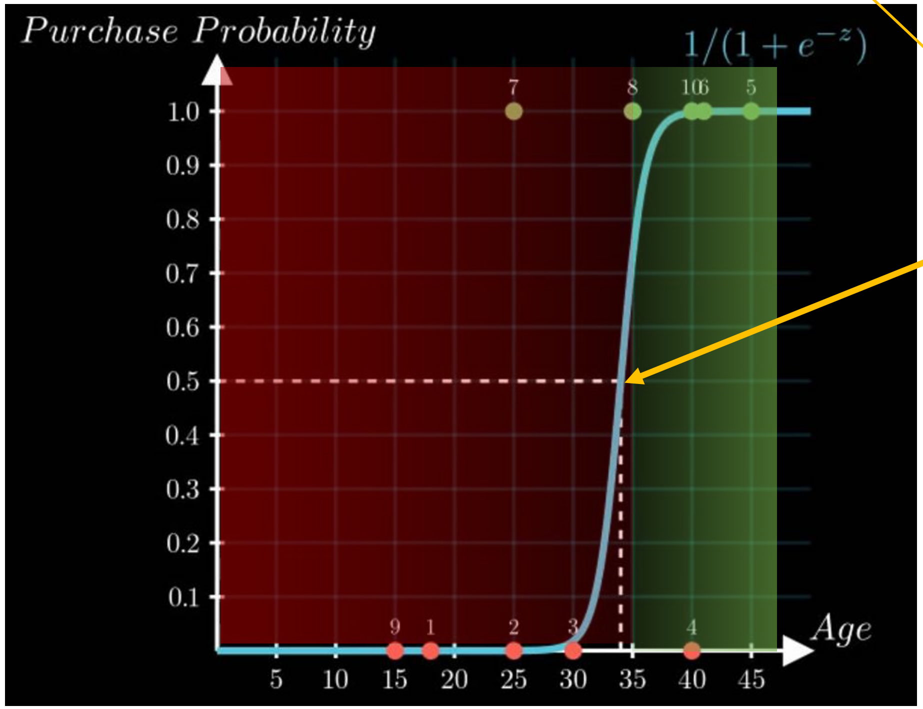


| Customer ID | Age | Purchase? |
|---|---|---|
| 1 | 18 | 0 |
| 2 | 25 | 0 |
| 3 | 30 | 0 |
| 4 | 40 | 0 |
| 5 | 45 | 1 |
| 6 | 41 | 1 |
| 7 | 25 | 1 |
| 8 | 35 | 1 |
| 9 | 15 | 0 |
| 10 | 40 | 1 |




$$z = -34 + (1)\ Age$$




$$p = \frac{1}{1+e^{-z}}$$
$$= \frac{1}{1+e^{-(34+1(34))}}$$
$$= 0.5$$

A customer with age 34 has 50% chance of purchase.

# Prediction

$$p = \frac{1}{1 + e^{-(b + w\,(Age))}}$$

Our model's parameters

Our model's parameters

$$z = -34 + 1\,(Age)$$

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

$Age$

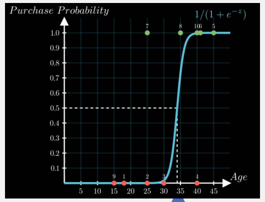
$w\ (1)$

b (-34)

z=1

$\Sigma$

Sigmoid function

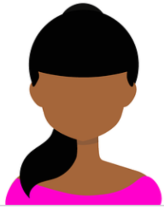



p=0.731

Probability p that label is 1

## ***Prediction***

| p<0.5 | p>=0.5 |
| --- | --- |
| Customer will not purchase | Customer will purchase |



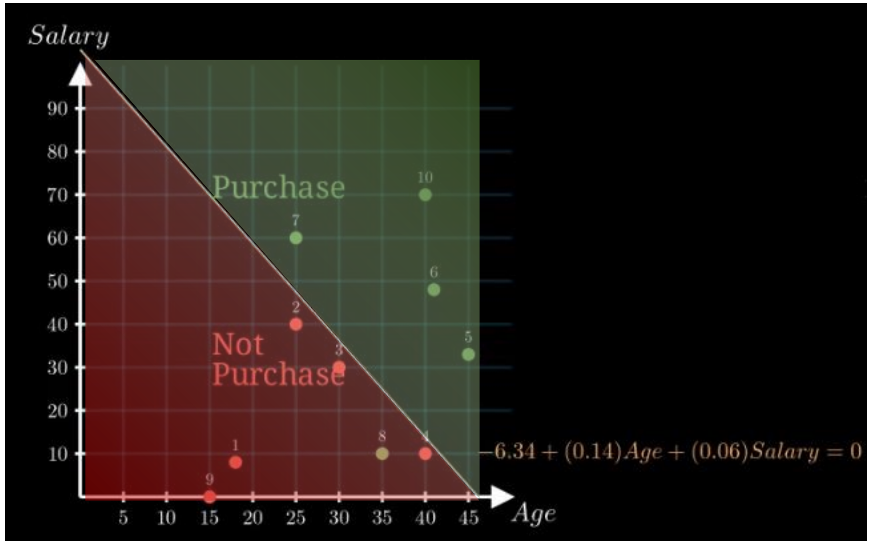
Age=35

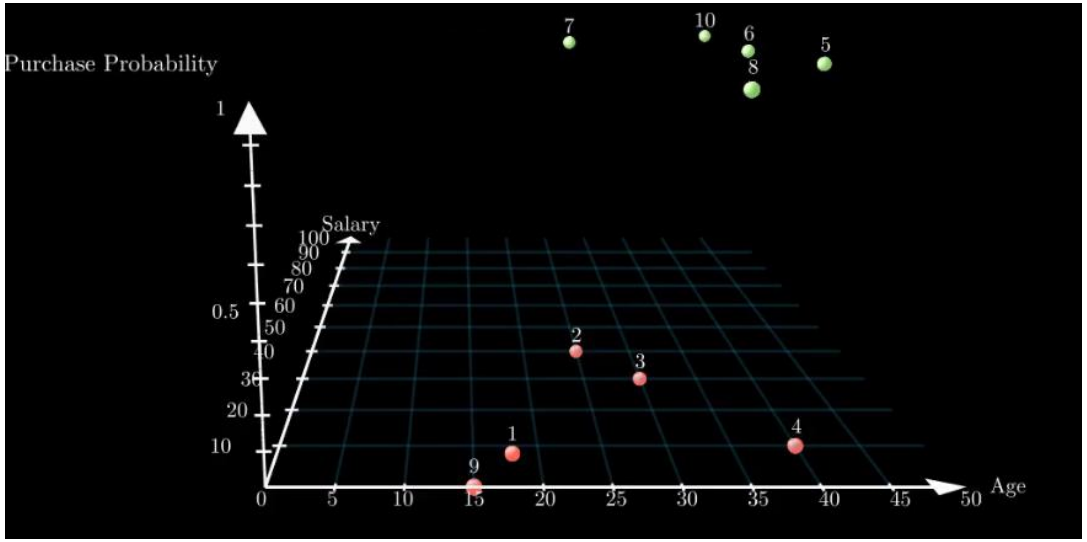
# Using age and salary as predictors

| Customer ID | Age | Salary (1000) | Purchase? |
|---|---|---|---|
| 1 | 18 | 8 | 0 |
| 2 | 25 | 40 | 0 |
| 3 | 30 | 30 | 0 |
| 4 | 40 | 10 | 0 |
| 5 | 45 | 33 | 1 |
| 6 | 41 | 48 | 1 |
| 7 | 25 | 60 | 1 |
| 8 | 35 | 10 | 1 |
| 9 | 15 | 0 | 0 |
| 10 | 40 | 70 | 1 |

$$z = b + w_1\,(Age) + w_2(Salary)$$

$$z = -6.34 + 0.14\,(Age) + 0.06(Salary)$$

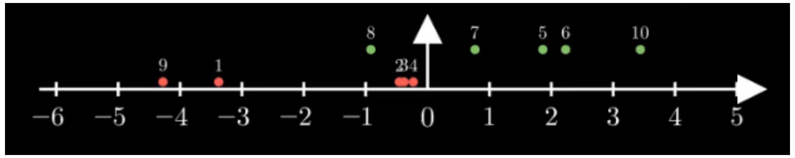


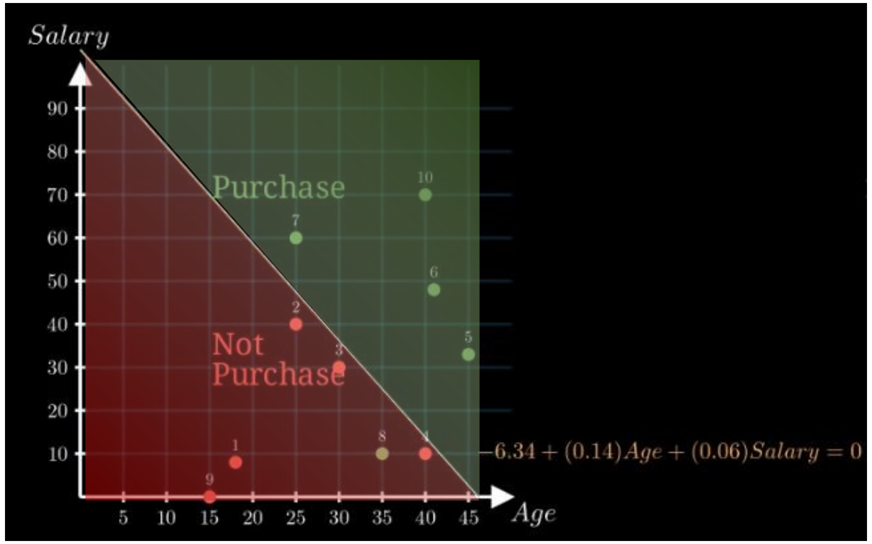
# Using age and salary as predictors

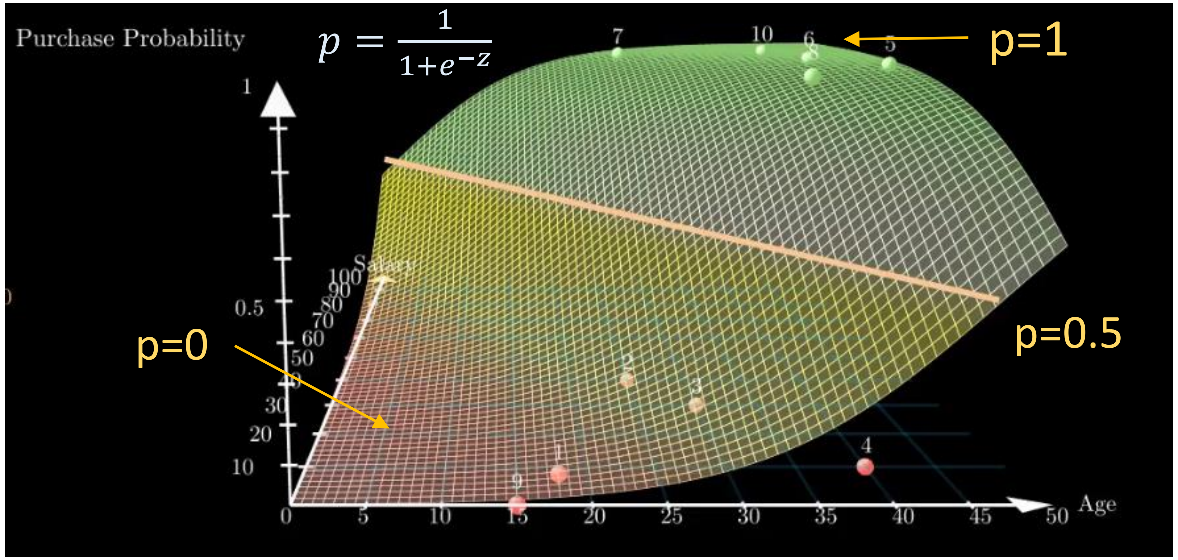
| Customer ID | Age | Salary (1000) | Purchase? |
|---|---|---|---|
| 1 | 18 | 8 | 0 |
| 2 | 25 | 40 | 0 |
| 3 | 30 | 30 | 0 |
| 4 | 40 | 10 | 0 |
| 5 | 45 | 33 | 1 |
| 6 | 41 | 48 | 1 |
| 7 | 25 | 60 | 1 |
| 8 | 35 | 10 | 1 |
| 9 | 15 | 0 | 0 |
| 10 | 40 | 70 | 1 |

$$z = b + w_1\,(Age) + w_2(Salary)$$



$$z = -6.34 + 0.14\,(Age) + 0.06(Salary)$$



# Prediction (2 predictors)

Our model's parameters

$z = -6.34 + 0.14\ (Age) + 0.06(Salary)$

$$p = \frac{1}{1 + e^{-z}}$$

$Age$ $\quad w_1\ (0.14)$

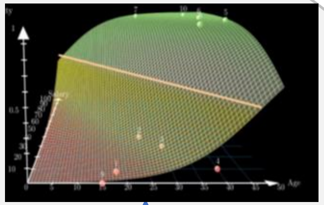
$Salary$ $\quad w_2\ (0.06)$

**b (-6.34)**

1

$\Sigma$

Sigmoid function



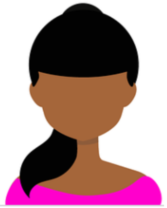
Probability p that label is 1

## *Prediction*

| p<0.5 | p>=0 |
|---|---|
| Customer will not purchase | Customer will purchase |

**p=0.30153**

Age=35
Salary=10



# Example

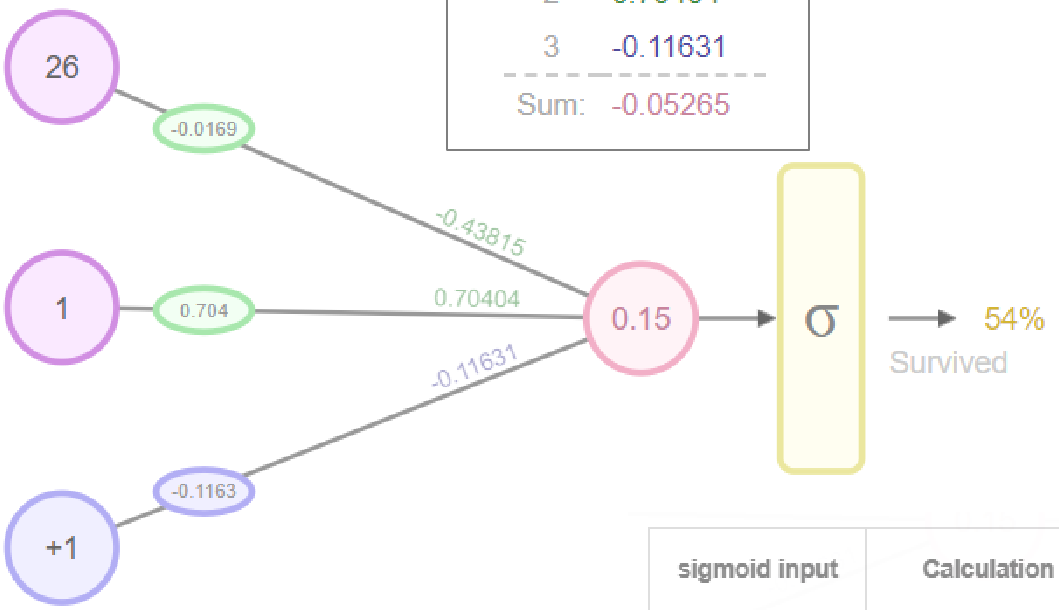
| PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |

......

https://www.kaggle.com/c/titanic

| Age | Sex | Survived |
|---|---|---|
| 22 | 0 | 0 |
| 38 | 1 | 1 |
| 26 | 1 | 1 |
| 35 | 1 | 1 |
| 35 | 0 | 0 |
| 14 | 1 | 0 |
| 25 | 0 | 0 |
| 54 | 0 | 0 |

| # | input value |
|---|---|
| 1 | -0.64038 |
| 2 | 0.70404 |
| 3 | -0.11631 |
| Sum: | -0.05265 |





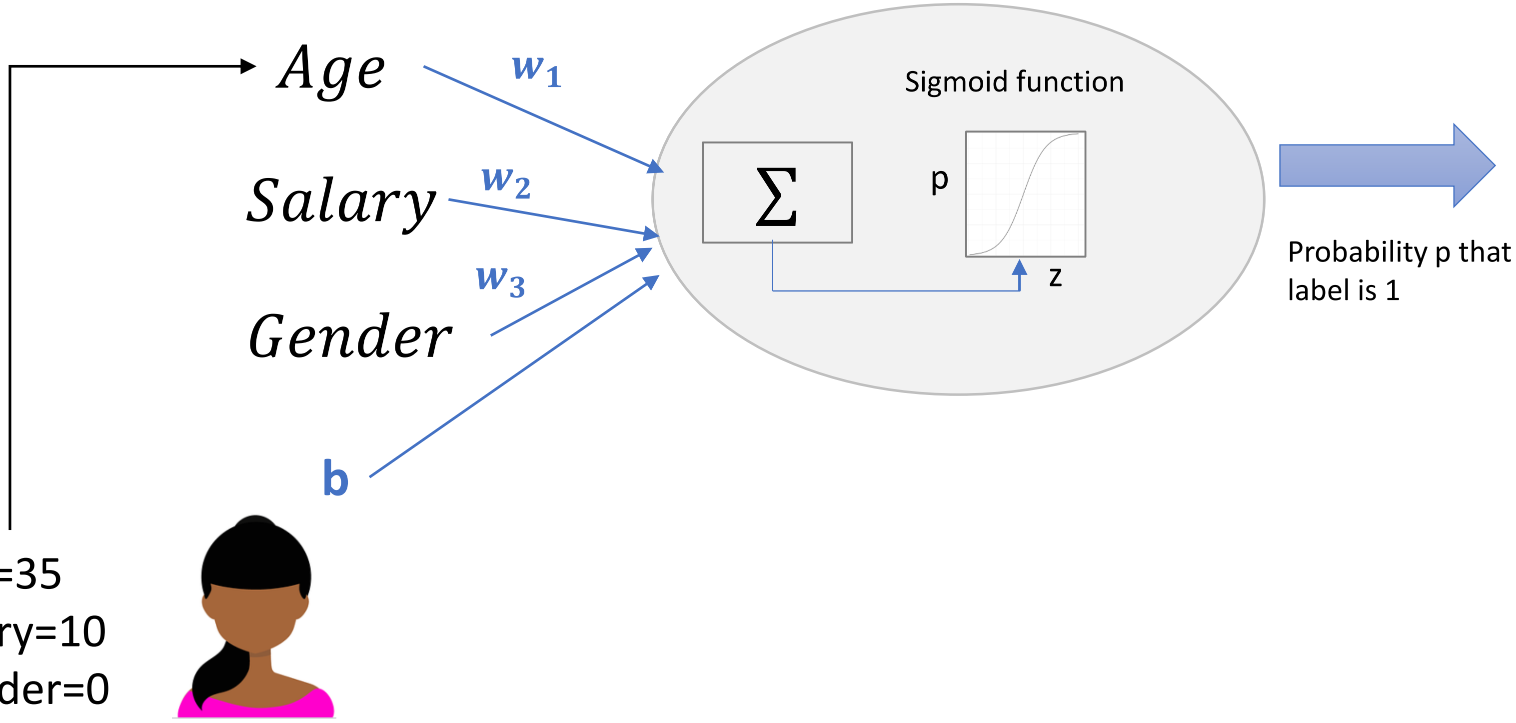
| sigmoid input | Calculation | sigmoid output |
|---|---|---|
| 0.14958 | $\frac{1}{1+e^{-(0.14958)}}$ | 0.54 |

# Include more predictors

Our model's parameters

$$z = b + w_1\,(Age) + w_2(Salary) + w_3\,(Gender)$$

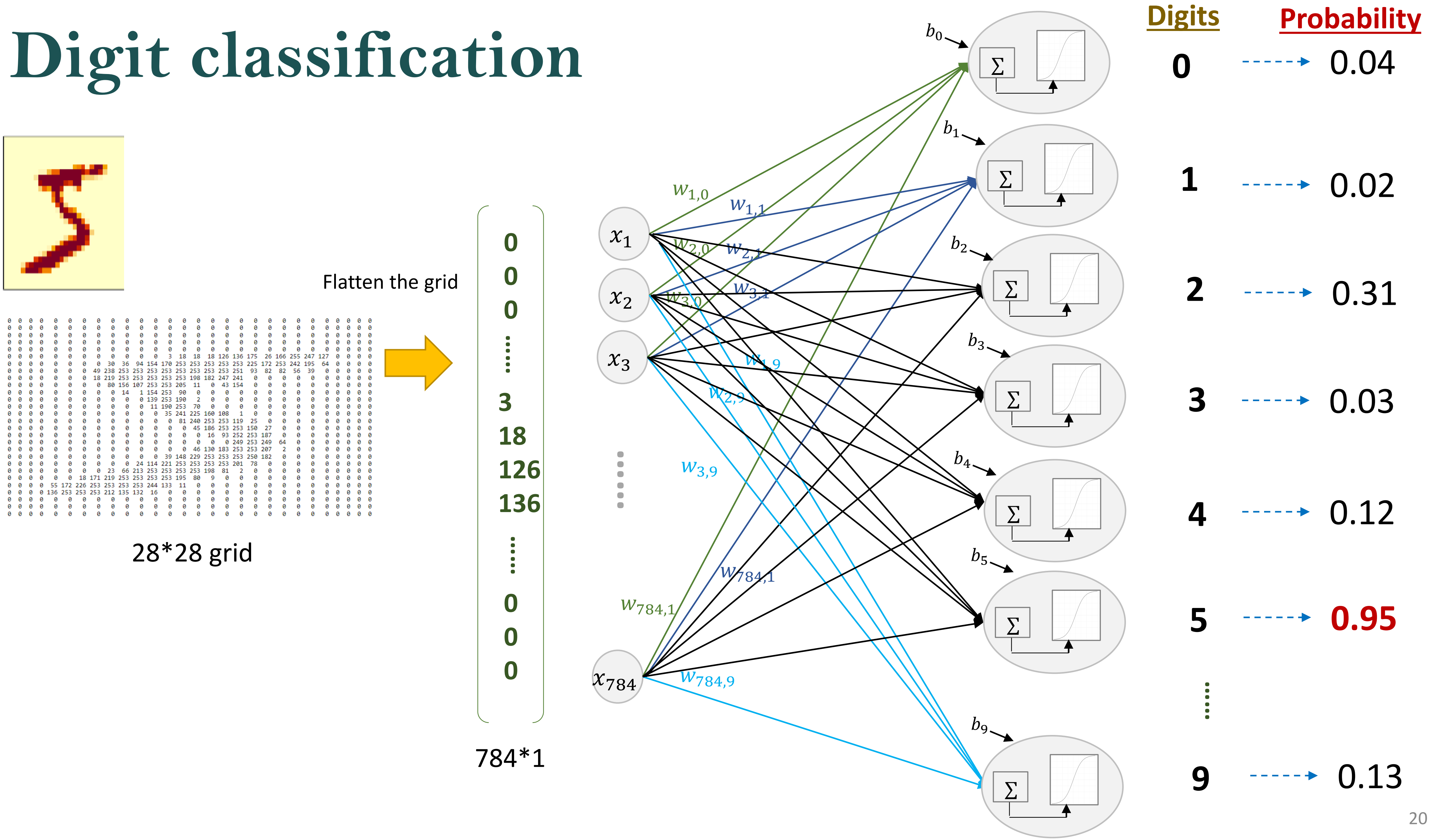
$$p = \frac{1}{1 + e^{-z}}$$




***Prediction***

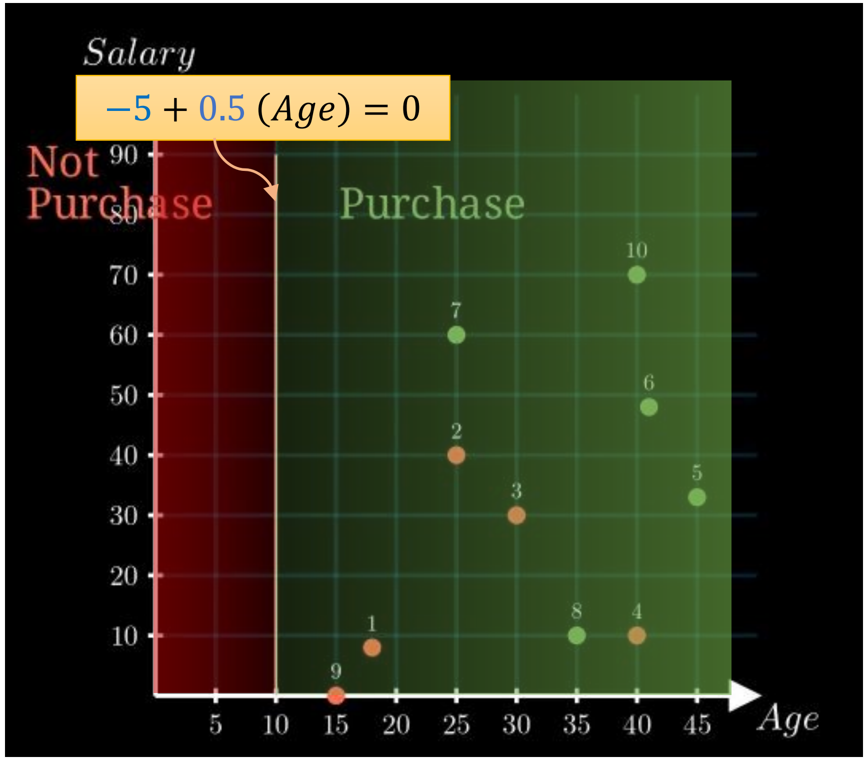
| | |
|---|---|
| p<0.5 | p>=0 |
| Customer will not purchase | Customer will purchase |

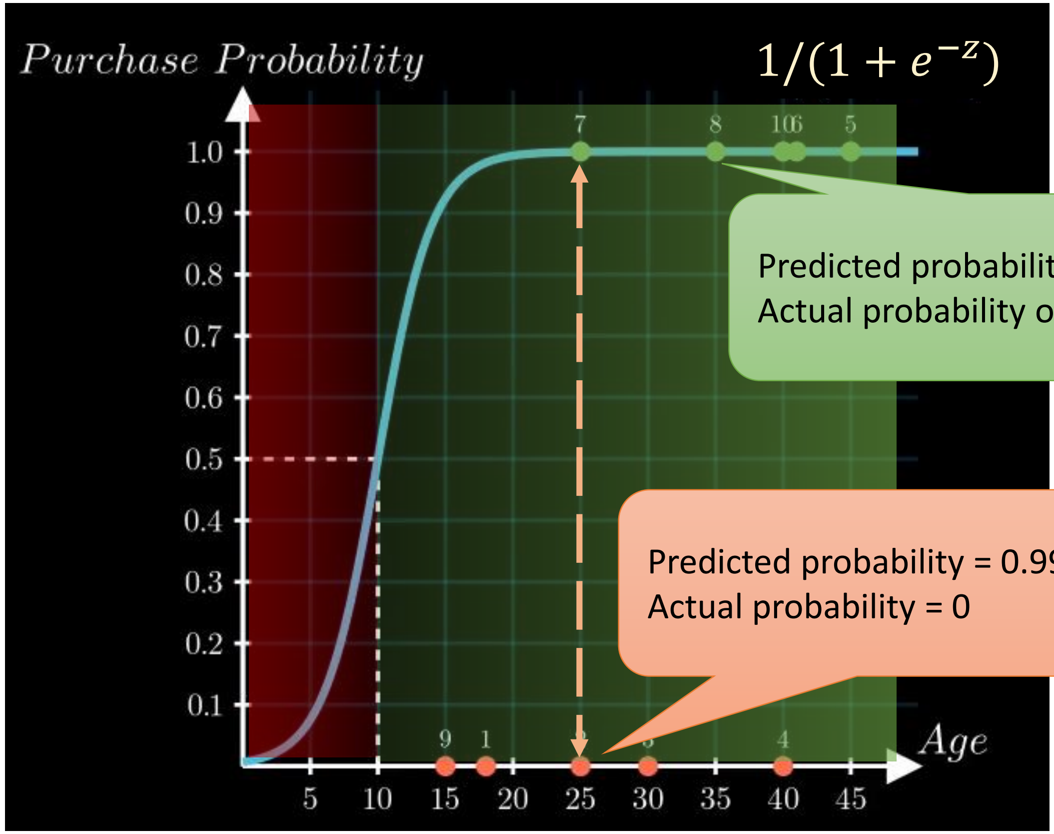
# Digit classification



# Prediction Error

$z = -5 + 0.5\,(Age)$




Predicted probability = 0.99999
Actual probability of purchase= 1

Predicted probability = 0.9994
Actual probability = 0

$$Prediction\ Error = Actual - Predicted$$

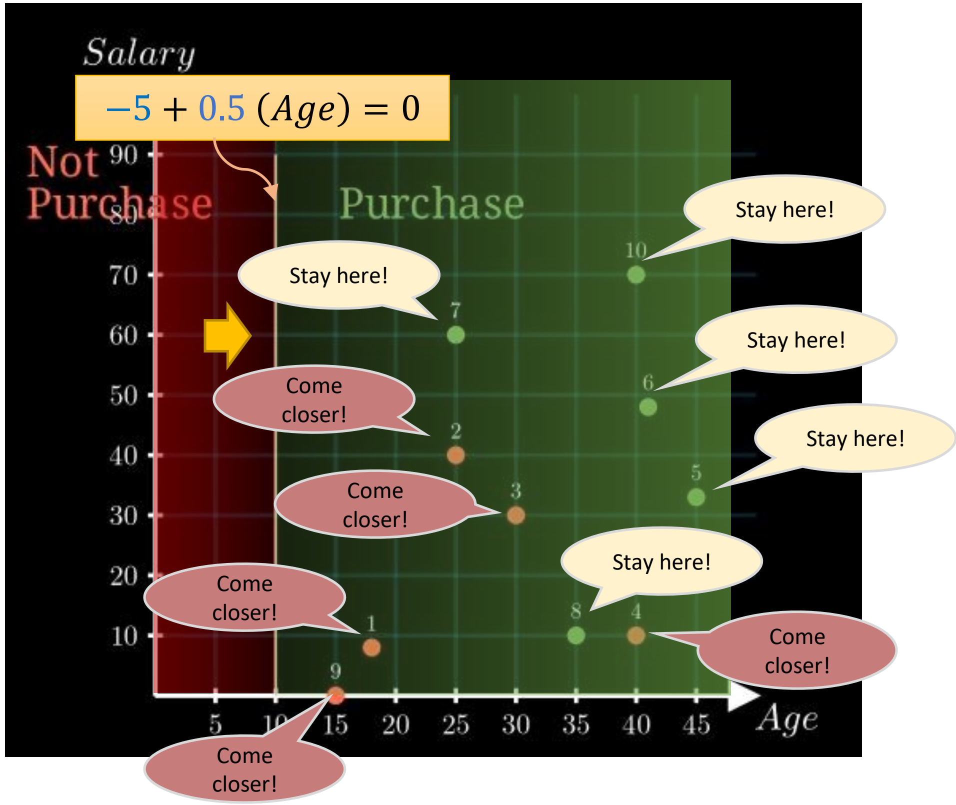
This model has large error!

# How to reduce prediction error?

- If the point is incorrectly classified, the decision boundary should be moved closer to the point

Slightly better model:

$$z = -5.01 + 0.48\,(Age)$$

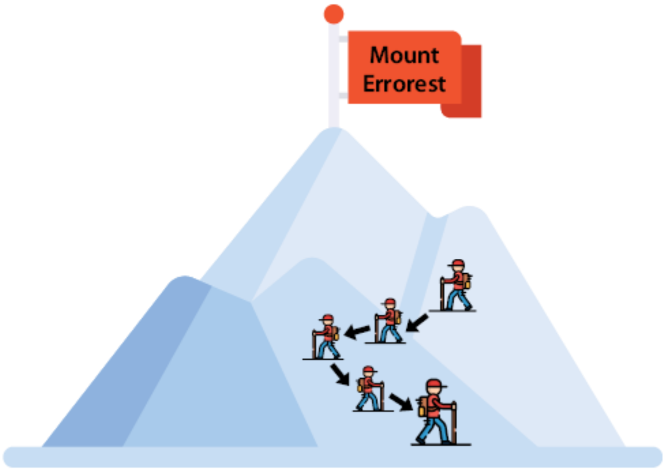


# Gradient Descent

How do we find the best b and w that best fits the data?

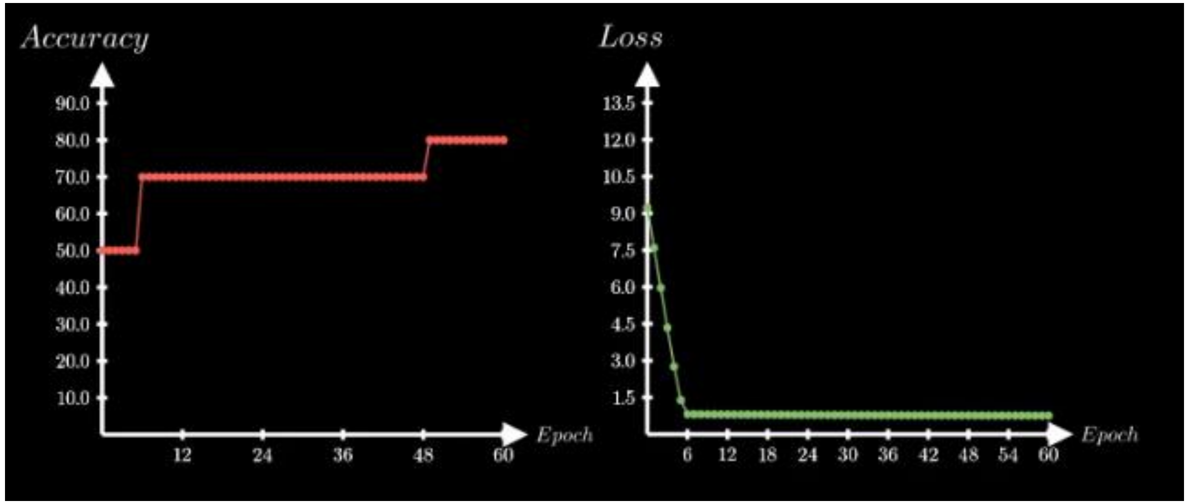
$$p = \frac{1}{1 + e^{-(b + w\ (Age))}}$$

1. We start with random **w** and **b**
2. Find the best direction to take one small step, in the direction of greatest descent. Take this small step to update b and w
3. Repeat the step  many times



# When should we stop the training?

- Terminate after a defined maximum number of iterations
- When the error function stop improving
  - converges to the point with minimum error

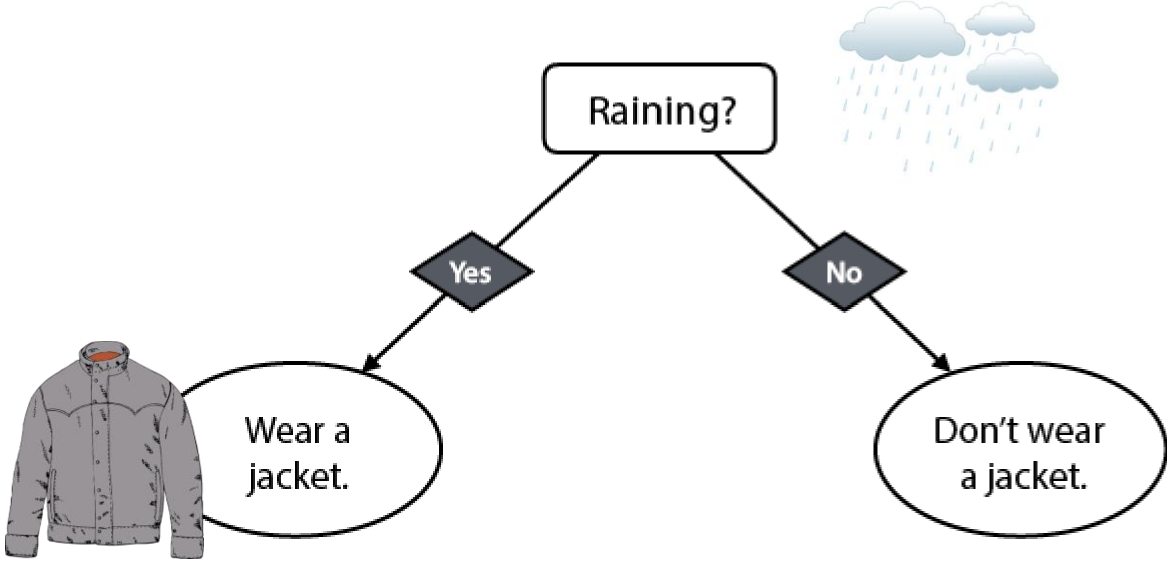


# Decision Tree

# Picking a good first question

Which questions are useful?

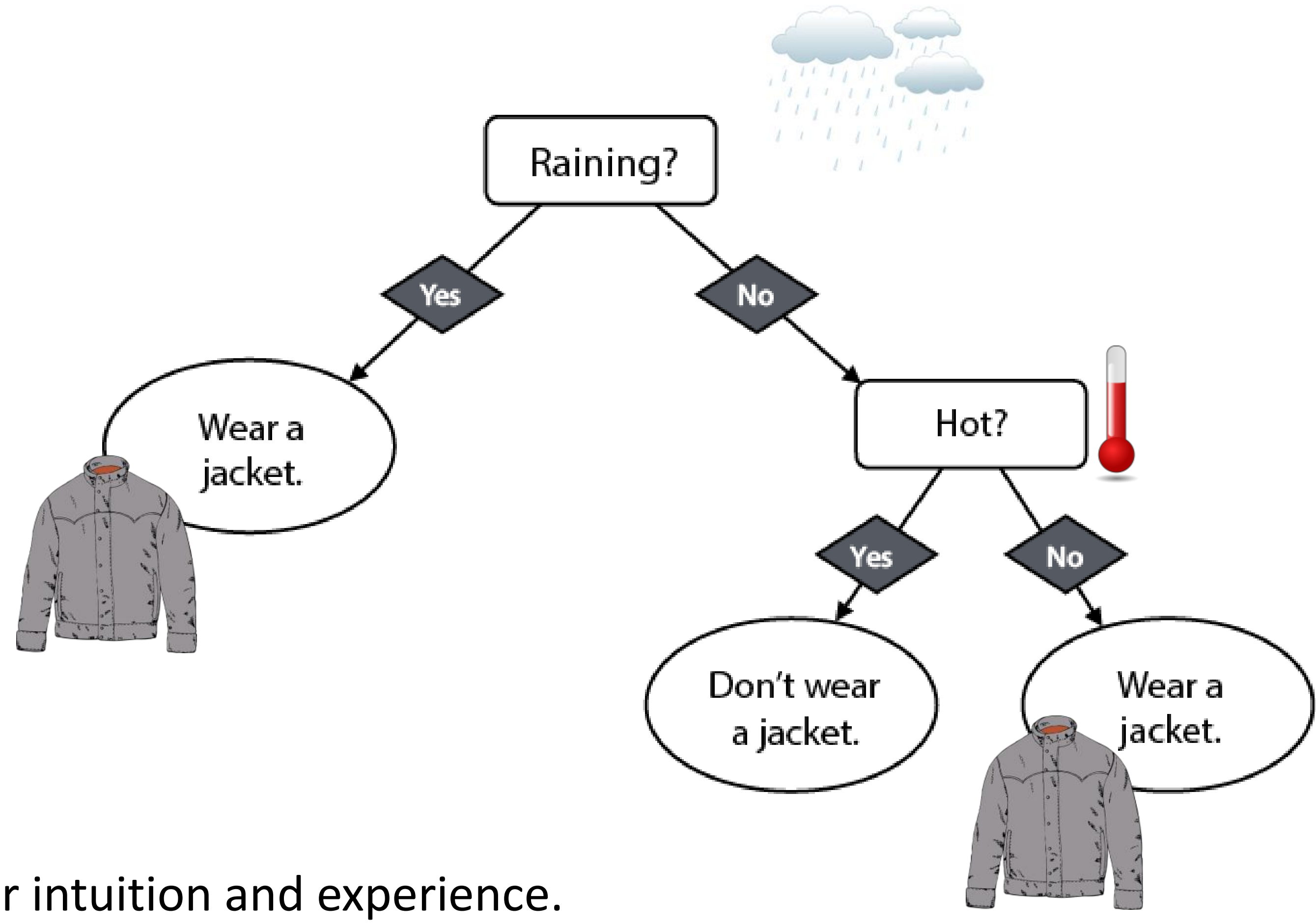
- Is it raining?
- Is it cold outside?
- Am I hungry?
- Is there a red car outside?
- Is it Monday?




When it rains, wearing a jacket is always the correct decision

However, there are days on which it doesn't rain, and not wearing a jacket is not the correct decision.

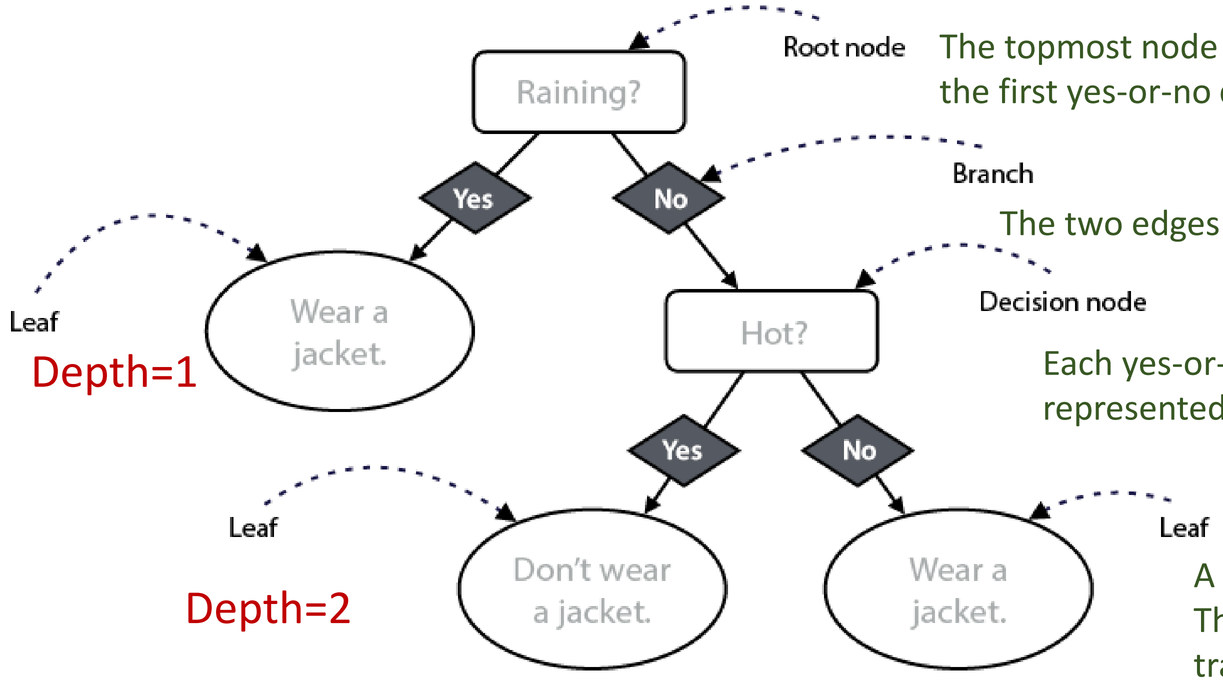
# Picking the next best question




- In this example, we made our decisions using our intuition and experience.
- Let's see how we may build these trees based on data

# What is a decision tree?

- A machine learning model based on a set of questions




Root node: The topmost node of the tree. It contains the first yes-or-no question.

Branch: The two edges emanating from each decision node

Decision node: Each yes-or-no question in our model is represented by a decision node.

Leaf: A node that has no branches emanating from it. These represent the decisions we make after traversing the tree

Depth=1

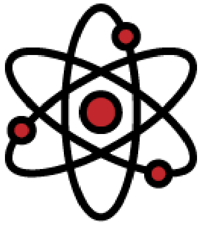
Depth=2

Height=2

# Recommend apps to users

- Consider the task of recommending to users which app to download
  - **Atom Count**: an app that counts the number of atoms in your body
  - **Beehive Finder**: an app that maps your location and finds the closest beehives
  - **Check Mate Mate**: an app for finding Australian chess players

| Platform | Age | App |
|---|---|---|
| iPhone | Young | Atom Count |
| iPhone | Adult | Check Mate Mate |
| Android | Adult | Beehive Finder |
| iPhone | Adult | Check Mate Mate |
| Android | Young | Atom Count |
| Android | Young | Atom Count |



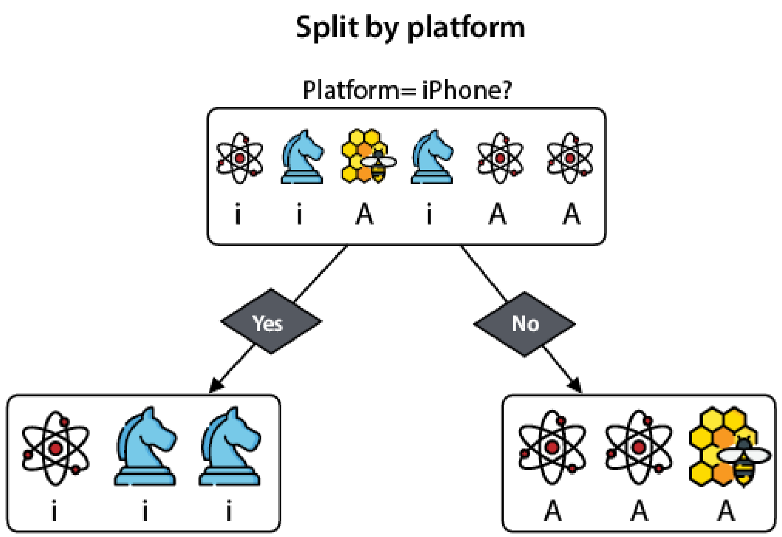
Atom Count

Beehive Finder

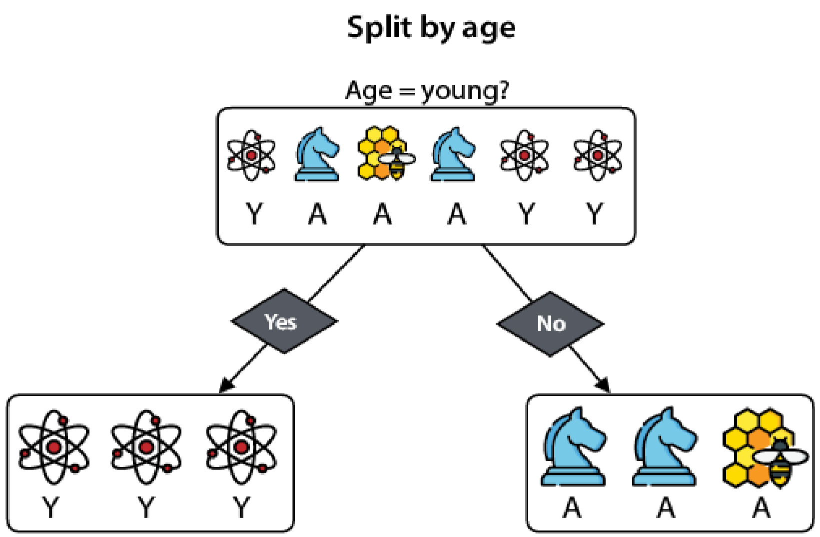
Check Mate Mate

# Which question is the best to ask first?

| Platform | Age | App |
|---|---|---|
| iPhone | Young | Atom Count |
| iPhone | Adult | Check Mate Mate |
| Android | Adult | Beehive Finder |
| iPhone | Adult | Check Mate Mate |
| Android | Young | Atom Count |
| Android | Young | Atom Count |

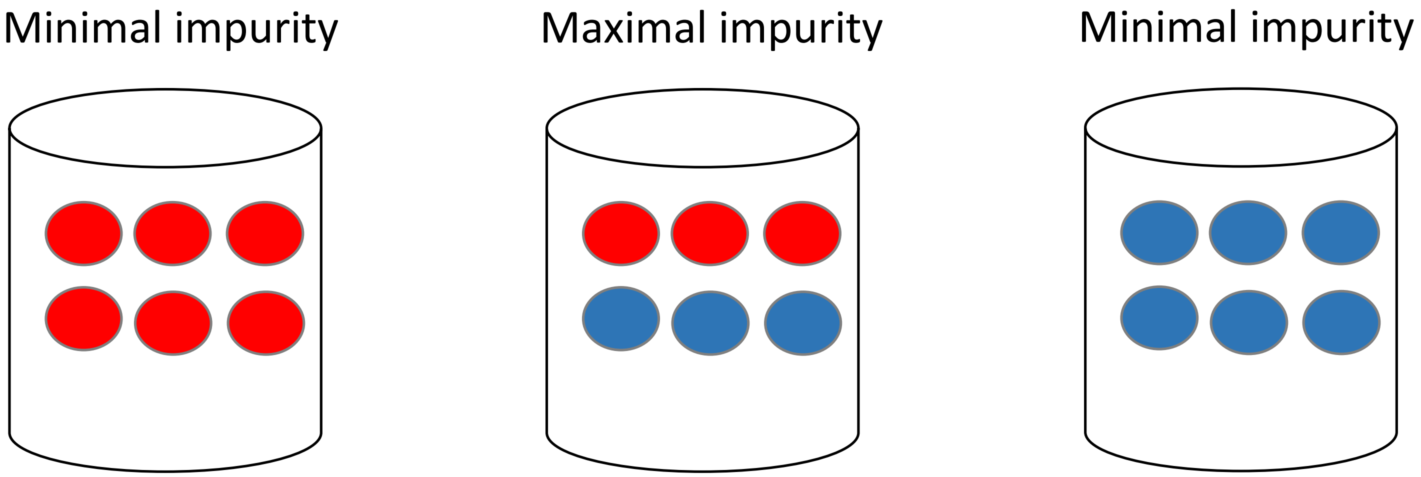



VS.



# Measuring impurity

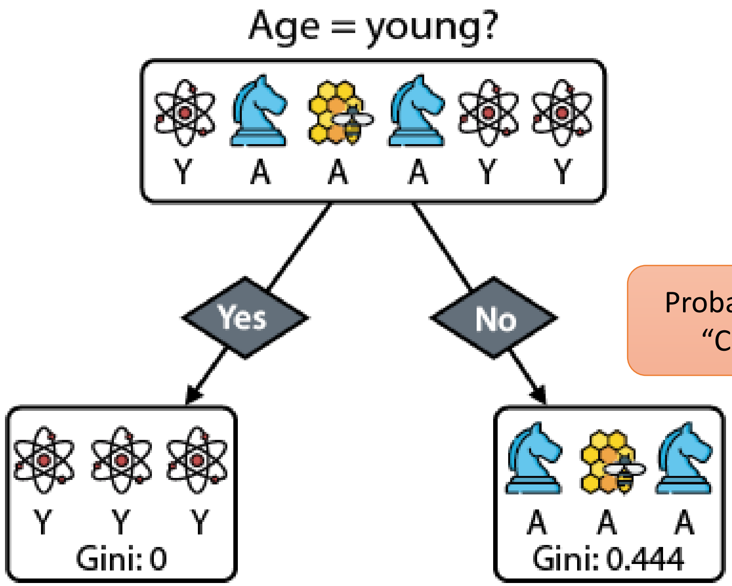
- In order to answer the splitting decision, one needs to define the concept of impurity or chaos.
- Decision trees will aim at minimizing the impurity in the data.



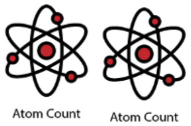
# Gini Index

- Gini Index measures how diverse our data is
- If we pick two random elements from each of the node, what is the probability that they belong to different classes?

$$\text{Weighted Gini} = \frac{3}{6}(0) + \frac{3}{6}(0.44) = \mathbf{0.22}$$



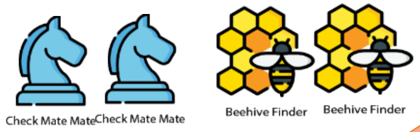

Probability of picking two "Atom Count"




$$Gini = 1 - \frac{3}{3} * \frac{3}{3}$$
$$= \mathbf{0}$$

Prob. of picking two elements of different class is 0

Probability of picking two "Check Mate Mate"



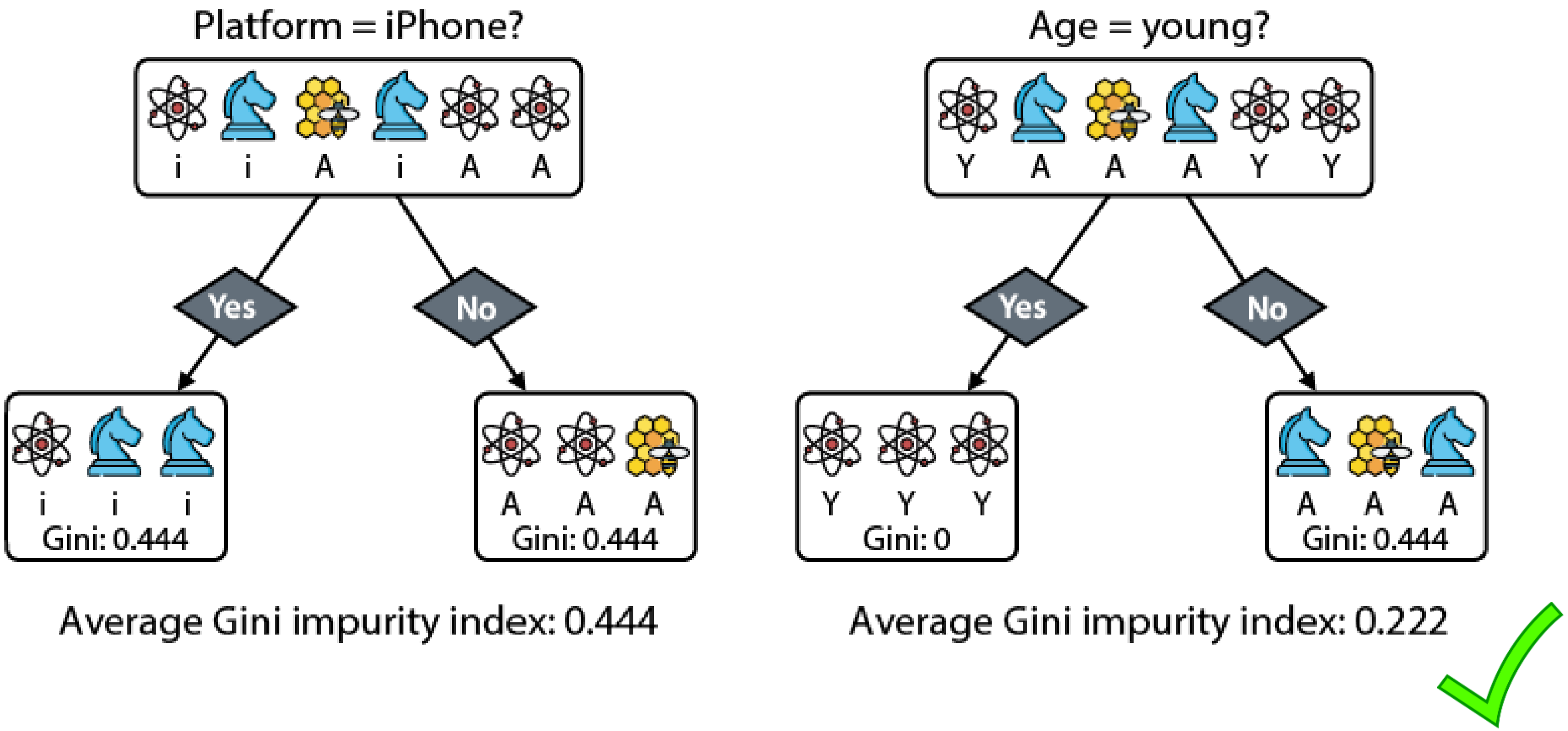

Probability of picking two "Beehive Finder"

$$Gini = 1 - \frac{2}{3} * \frac{2}{3} - \frac{1}{3} * \frac{1}{3}$$
$$= \mathbf{0.444}$$

Prob. of picking two elements of different class is 0.444

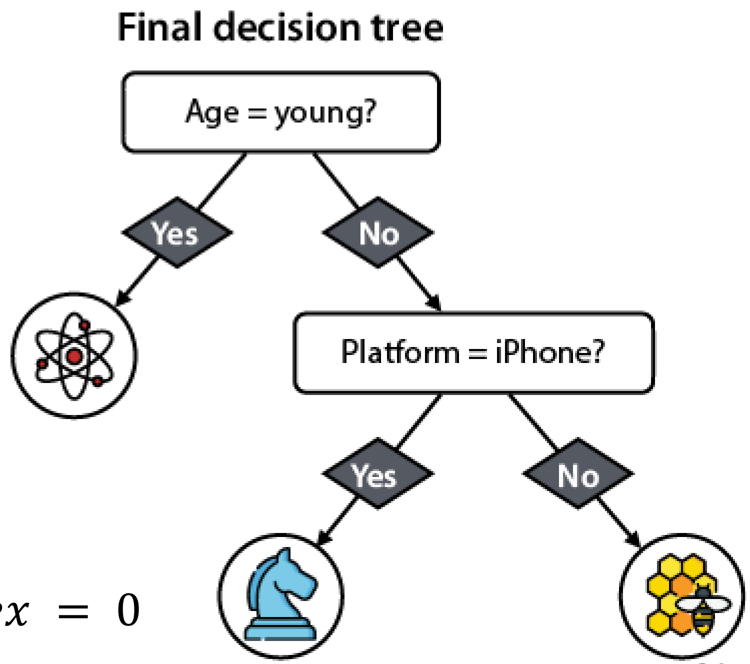
# Picking the better split

- **Low** Gini Index: if we have a set in which all the elements are similar  (lower impurity)
- **Large** Gini Index: if all the elements are different (higher impurity)



# Continue our split

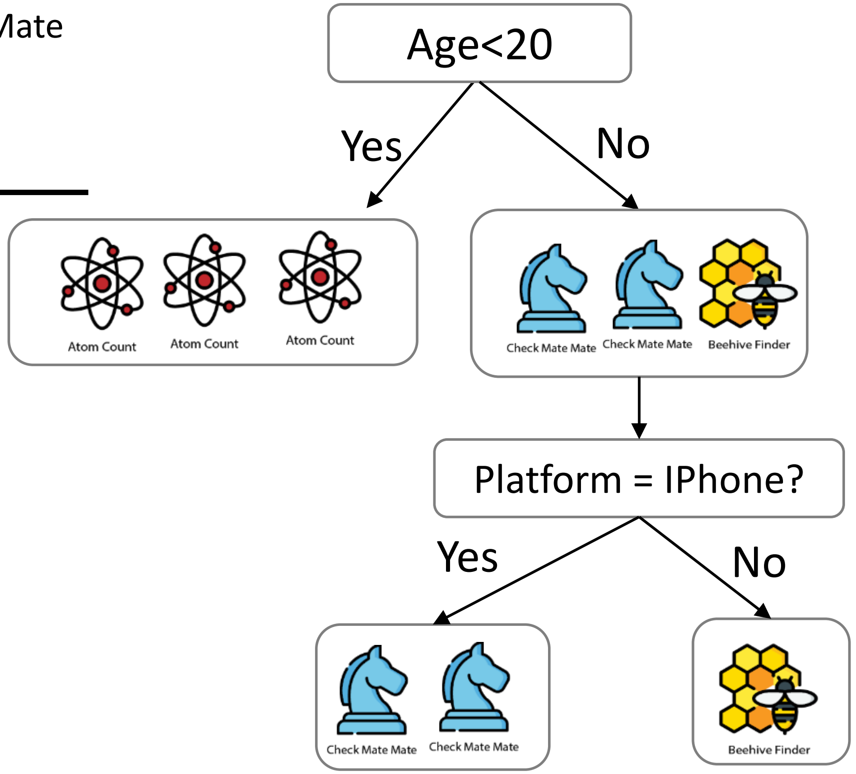
- The dataset on the right can still be divided, because it has two labels: "Beehive Count" and "Check Mate Mate."
- We've used the **age** feature already, so let's try using the **platform** feature.

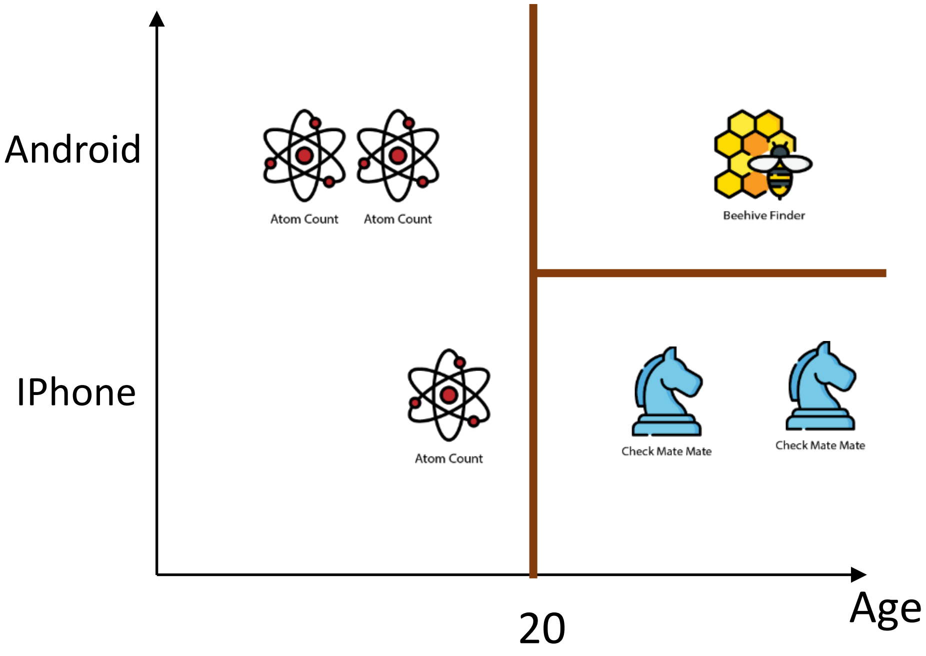



$$Average\ Gini\ Impurity\ Index\ =\ 0$$

# Splitting the data using continuous features

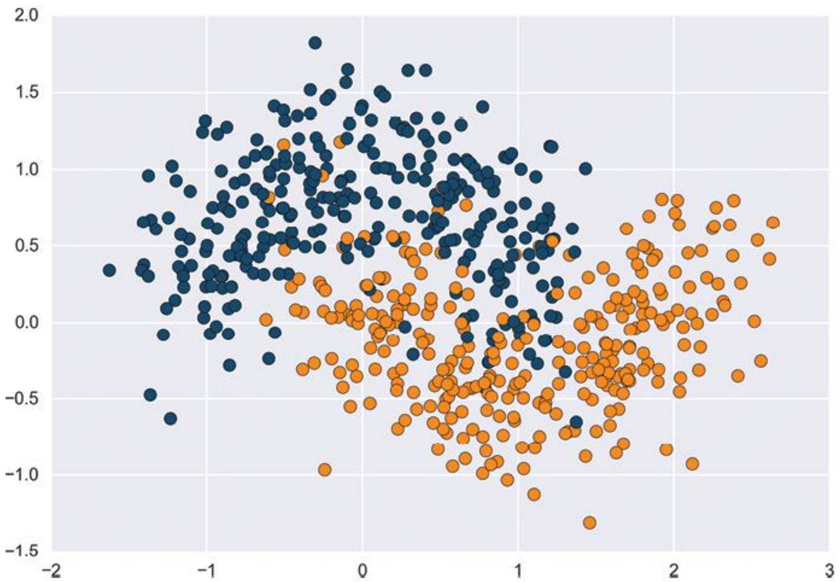
| Platform | Age | App |
|---|---|---|
| iPhone | 15 | Atom Count |
| iPhone | 25 | Check Mate Mate |
| Android | 32 | Beehive Finder |
| iPhone | 35 | Check Mate Mate |
| Android | 12 | Atom Count |
| Android | 14 | Atom Count |

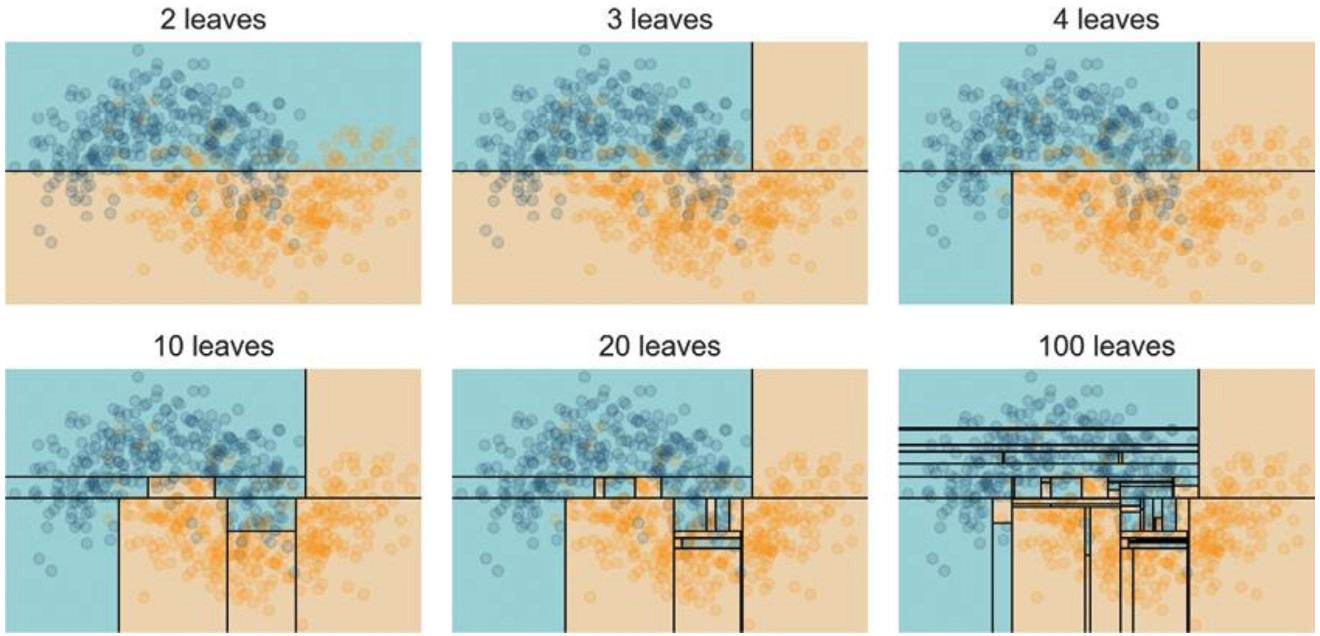



Decision tree always results in Axis-parallel splits.

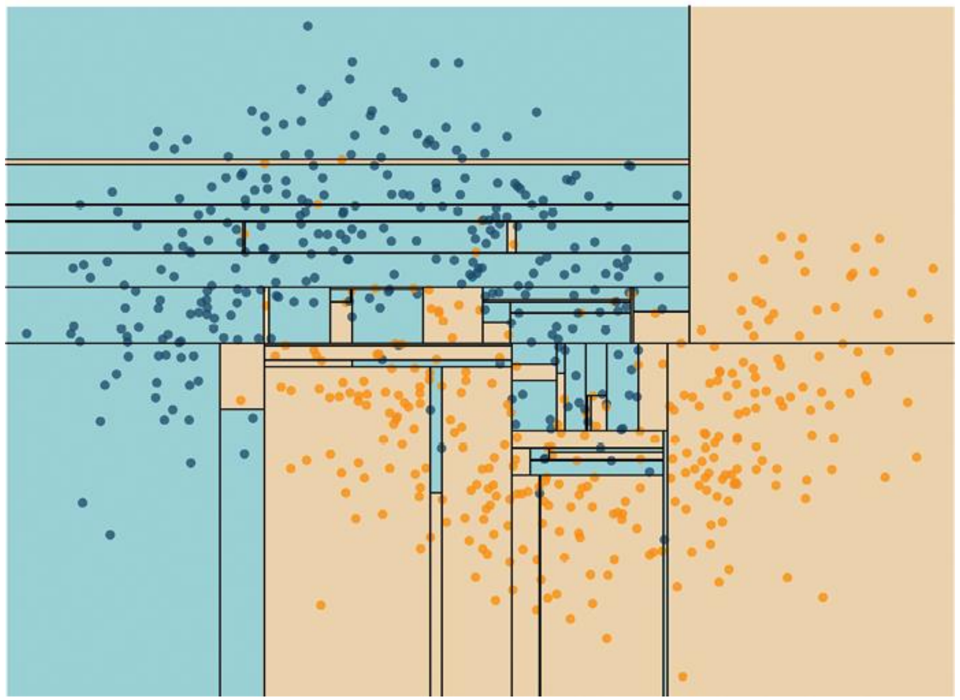
# Let's fit this data with decision tree



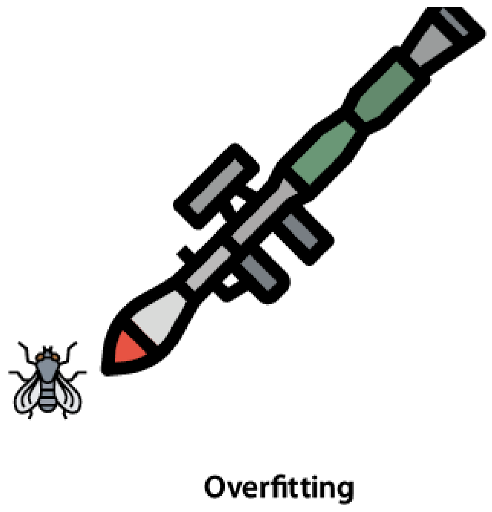
# A decision tree with 100 leave nodes

- We can achieve 100% training accuracy with 100 leaves to correctly classify all the points in the training data!
- What's wrong with the model?



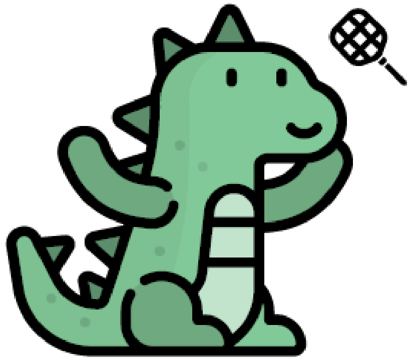
# Overfitting

- *Overfitting* looks a lot like memorizing the entire textbook instead of studying for the exam.
- It happens when we try to train a model that is too complex, and it memorizes the data instead of learning it well.



# Underfitting

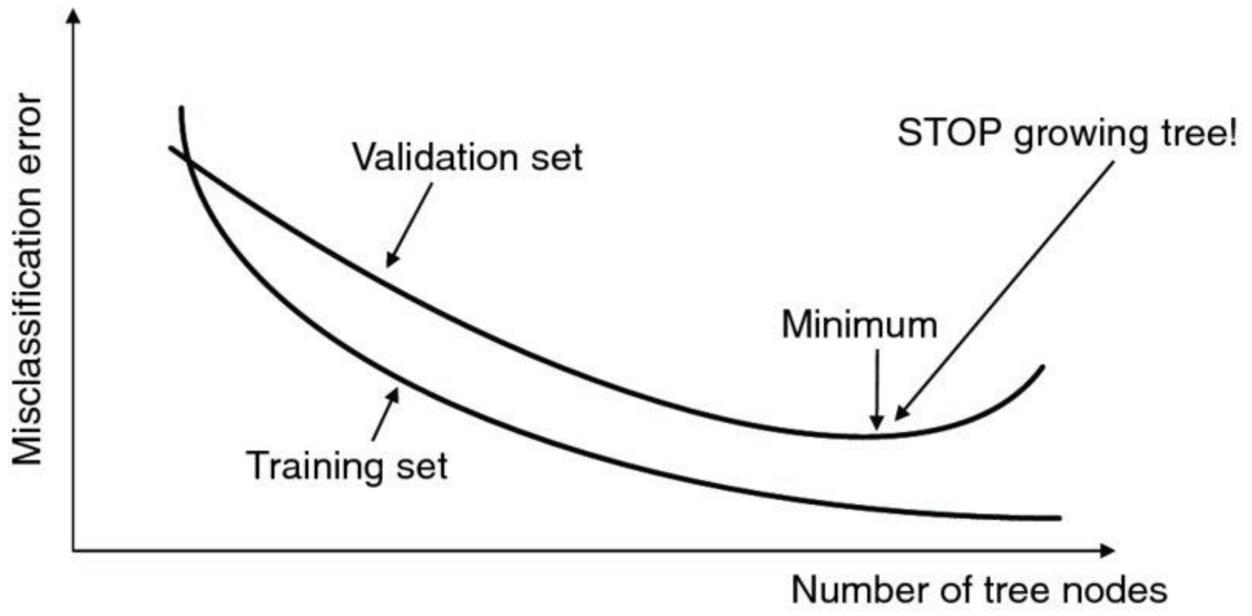
- In machine learning, *underfitting* looks a lot like not having studied enough for an exam.
- It happens when we try to train a model that is too simple, and it is unable to learn the data.



Underfitting

# Using a validation set to stop growing a decision tree

- A good model is one that neither underfits nor overfits
- Learns the data properly and can make good predictions on new data that it hasn't seen.



# When to stop building the tree?

- Don't split a node if the change in accuracy/Gini index is below some threshold.
- Don't split a node if it has less than a certain number of samples.
- Split a node only if both of the resulting leaves contain at least a certain number of samples.
- Stop building the tree after you reach a certain depth.