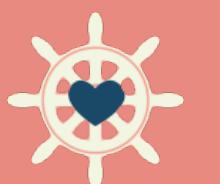




# DECISION TREE



IKM LAB





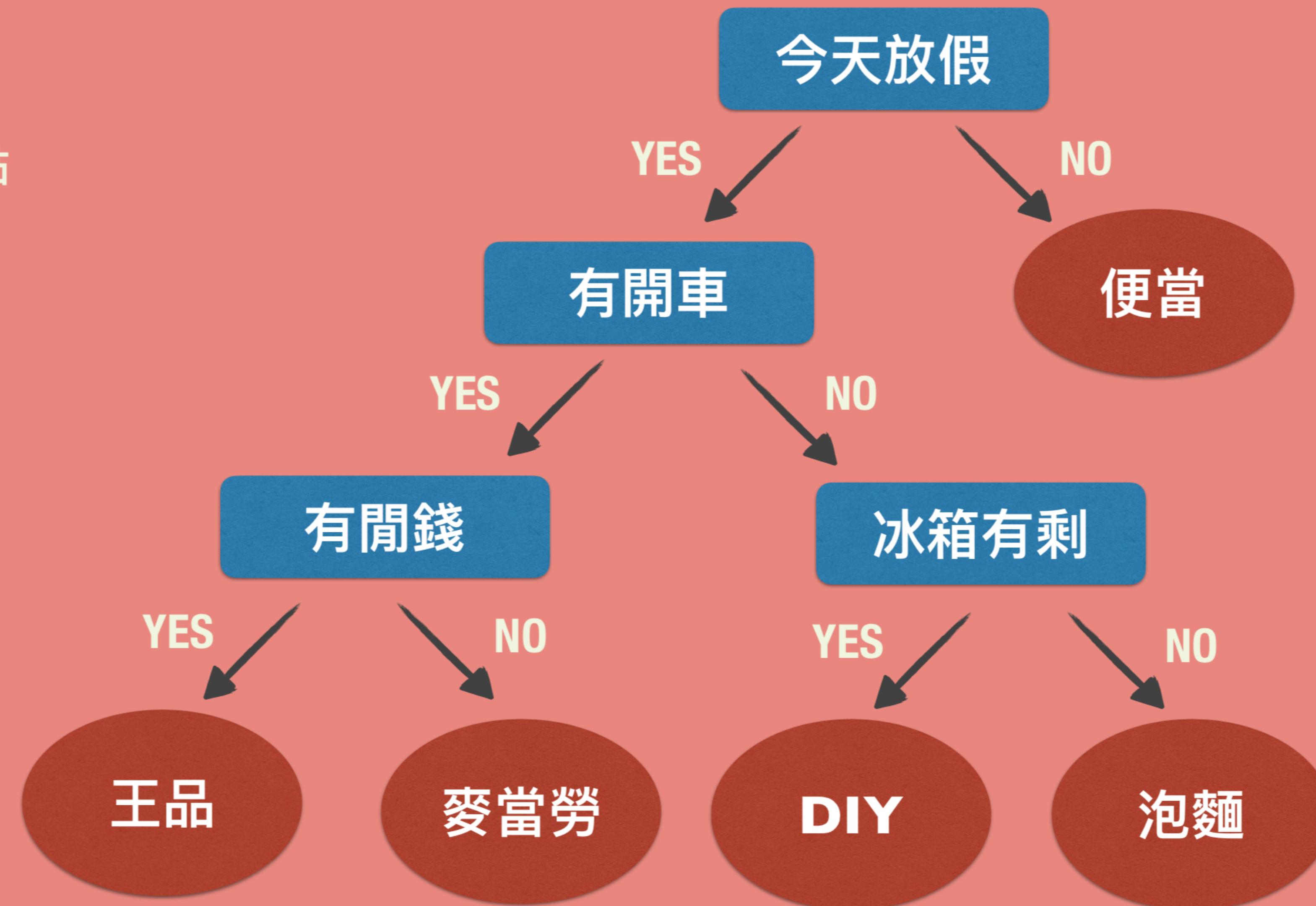
---

# OUTLINE

- **What is Decision Tree?**
- **How can we use it?**
- **How can we know which attribute is important?**
- **Building my decision tree**
- **Practice time!**

# **WHAT IS DECISION TREE ?**

- 藍色方形：決策點
- 紅色橢圓：結果



**HOW CAN WE  
USE  
DECISION TREE ?**



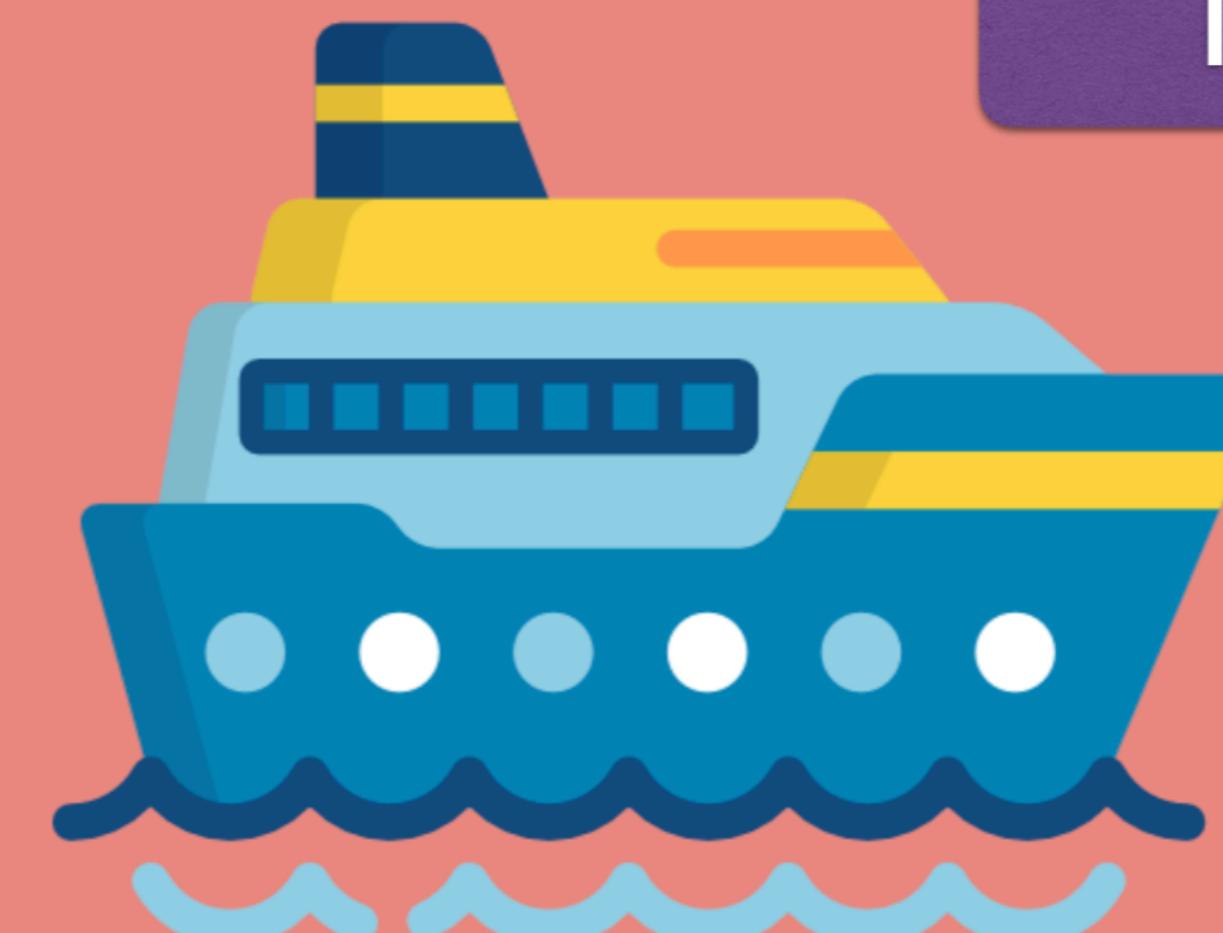
Passenger 1

35-year-old  
Ticket Class = 2nd  
Parents. Wife



Passenger 2

7-month-old  
Ticket Class = 3rd  
Parents



Titanic



Passenger 3

62-year-old  
Ticket Class = 1st  
Husband. son

# TRAINING DATA

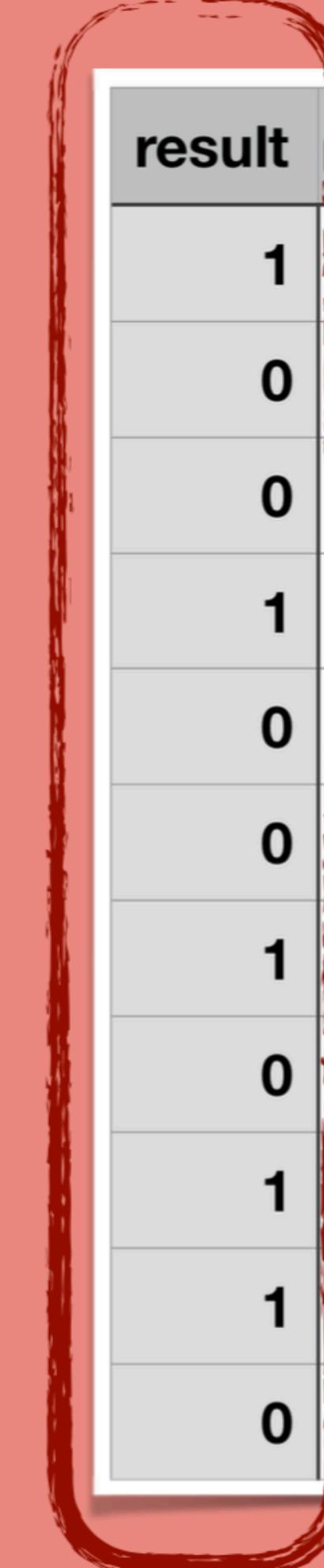
- PClass : 艙等 (1~3)
- Sex : 性別
- Age : 年齡
- SibSp : 兄弟姐妹、丈夫（妻子）人數
- Parch : 父母、小孩人數
- Fare : 票價
- Result : 是否存活 (1:存活，0:死亡)

result	Pclass	Sex	Age	SibSp	Parch	Fare
0	3	0	22	1	0	7.25
1	1	1	38	1	0	71.2833
1	3	1	26	0	0	7.925
1	1	1	35	1	0	53.1
0	3	0	35	0	0	8.05
0	1	0	54	0	0	51.8625
0	3	0	2	3	1	21.075
1	3	1	27	0	2	11.1333
1	2	1	14	1	0	30.0708
1	3	1	4	1	1	16.7
1	1	1	58	0	0	26.55
0	3	0	20	0	0	8.05

from : Kaggle

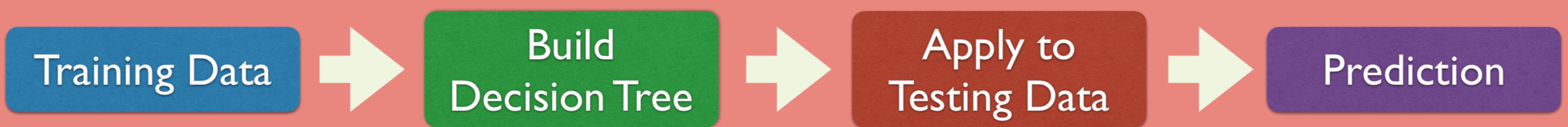
# TESTING DATA

- 現在已知以下資訊：
  - PClass : 艙等 (1~3)
  - Sex : 性別
  - Age : 年齡
  - SibSp : 兄弟姐妹、丈夫（妻子）人數
  - Parch : 父母、小孩人數
  - Fare : 票價
- 希望猜出該乘客是否存活

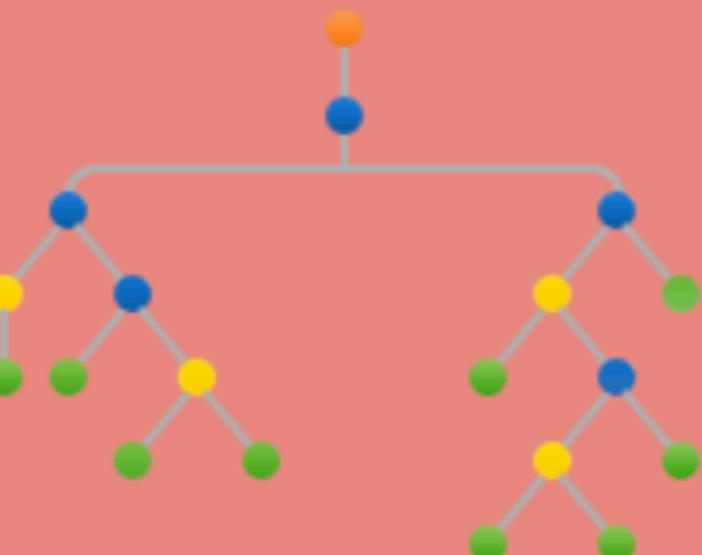


result	Pclass	Sex	Age	SibSp	Parch	Fare
1	2	1	40	1	1	39
0	1	0	31	1	0	52
0	2	0	70	0	0	10.5
1	2	0	31	0	0	13
0	3	0	18	0	0	7.775
0	3	0	24.5	0	0	8.05
1	3	1	18	0	0	9.8417
0	3	1	43	1	6	46.9
1	1	0	36	0	1	512.3292
1	1	0	27	0	0	76.7292
0	3	0	20	0	0	9.225

# FLOWCHART



result	Pclass	Sex	Age	SibSp	Parch	Fare
0	3	0	22	1	0	7.25
1	1	1	38	1	0	71.2833
1	3	1	26	0	0	7.925
1	1	1	35	1	0	53.1
0	3	0	35	0	0	8.05
0	1	0	54	0	0	51.8625
0	3	0	2	3	1	21.075
1	3	1	27	0	2	11.1333
1	2	1	14	1	0	30.0708
1	3	1	4	1	1	16.7
1	1	1	58	0	0	26.55
0	3	0	20	0	0	8.05



result	Pclass	Sex	Age	SibSp	Parch	Fare
1	2	1	40	1	1	39
0	1	0	31	1	0	52
0	2	0	70	0	0	10.5
1	2	0	31	0	0	13
0	3	0	18	0	0	7.775
0	3	0	24.5	0	0	8.05
1	3	1	18	0	0	9.8417
0	3	1	43	1	6	46.9
1	1	0	36	0	1	512.3292
1	1	0	27	0	0	76.7292
0	3	0	20	0	0	9.225



**HOW CAN WE KNOW  
WHICH ATTRIBUTE IS  
IMPORTANT?**

資料集	人數
存活	100
死亡	200

### Attribute 1

年齡 > 60

YES

NO

	人數		人數
存活	50	存活	50
死亡	50	死亡	150

### Attribute 2

艙等 = 1st

YES

NO

	人數		人數
存活	99	存活	1
死亡	1	死亡	199

資料集	人數
存活	100
死亡	200

## Attribute 1

年齡 > 60

YES

NO

	人數
存活	50
死亡	50

	人數
存活	50
死亡	150

## Attribute 2

艙等 = 1st



YES

NO

	人數
存活	99
死亡	1

	人數
存活	1
死亡	199

# GINI

- Gini impurity score is a measure of how the data would be incorrectly labeled.
- To compute Gini impurity for a set of items with  $J$  classes, suppose  $i \in \{1, 2, \dots, J\}$ , and  $p_i$  be the fraction of items labeled with class  $i$  in the set.
- Gini = 0.0 -> best
  - when we use this attribute to classify the dataset, it can be clearly categorized.
- Gini = 0.5 -> worst
  - this attribute is not an important feature to make a decision.

$$1 - \sum_{i=1}^J p_i^2$$

$J = \# \text{ of classes} = 2$  {

Training Data	人數
存活	100
死亡	200

i=1

i=2

$$\frac{1 - (100/300)^2 - (200/300)^2}{=0.44}$$



年齡 > 60

YES

NO

	人數
存活	50
死亡	50

	人數
存活	50
死亡	150

$$1 - (50/100)^2 - (50/100)^2 = 0.5$$

$$1 - (50/200)^2 - (150/200)^2 = 0.375$$

艙等 = 1st

YES

NO

	人數
存活	99
死亡	1

	人數
存活	1
死亡	199

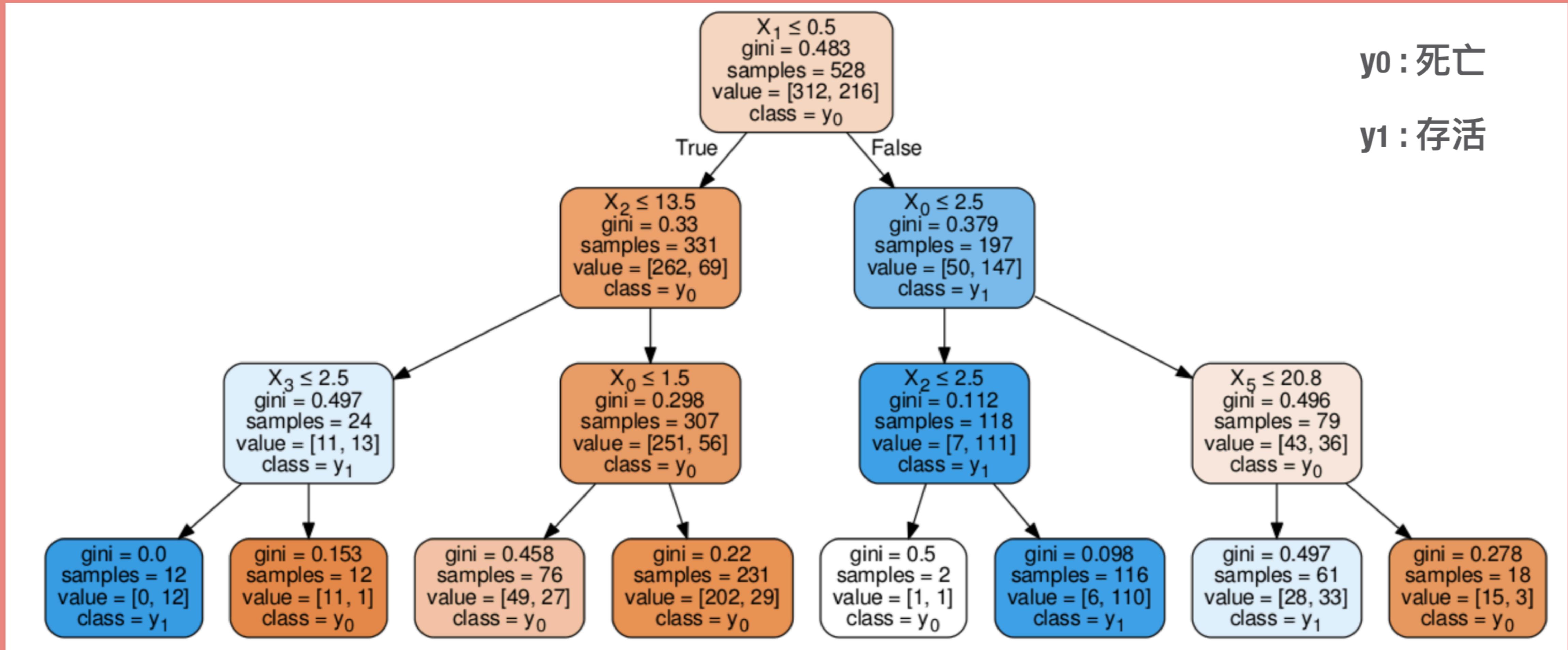
$$1 - (99/100)^2 - (1/100)^2 = 0.02$$

$$1 - (1/200)^2 - (199/200)^2 = 0.01$$

# TREE.PDF

$X_0 : \text{PClass}$        $X_3 : \text{SibSp}$   
 $X_1 : \text{Sex}$        $X_4 : \text{Parch}$   
 $X_2 : \text{Age}$        $X_5 : \text{Fare}$

$y_0 : \text{死亡}$   
 $y_1 : \text{存活}$



# EVALUATION

result	Pclass	Sex	Age	SibSp	Parch	Fare
	2	1	40	1	1	39
	1	0	31	1	0	52
	2	0	70	0	0	10.5
	2	0	31	0	0	13
	3	0	18	0	0	7.775
	3	0	24.5	0	0	8.05
	3	1	18	0	0	9.8417
	3	1	43	1	6	46.9
	1	0	36	0	1	512.3292
	1	0	27	0	0	76.7292
	3	0	20	0	0	9.225

## TESTING DATA

將答案遮住

使用剛剛的TREE

來預測每位乘客是否存活

accuracy : 0.8532608695652174

# PRACTICE TIME!

- Dataset link: <https://www.kaggle.com/c/titanic>
- Github link: <https://github.com/IKMLab/FDA-practice-Decision-Tree>

# SCIKIT-LEARN



- Scikit-learn (`sklearn`) is a well-known machine learning library for Python.
- It features various classification, regression and clustering algorithms including decision tree, random forests, k-means and DBSCAN..., and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

# **THANK YOU**

