Machine Learning, a tutorial

Tairung Chen, KKLAB

Last Week

Part I

- What is machine learning?
  - The process and the roles
- Data type and feature engineering
  - a.k.a. "Real-world Data is Dirty"
- Your first model: Linear Regression
  - Logistic model
  - ….which is tricky…..

Last week

- Fill the missing values
- Encode categorical variable for linear/logistic models
- Deal with outlier, nonlinearity, feature interaction
- => Try between many methods, evaluate their performance
    - How?

Outline

Part II

- Model evaluation
    - For Classification
- Tree-based models
- Hyperparameter Tuning

Model Evaluation

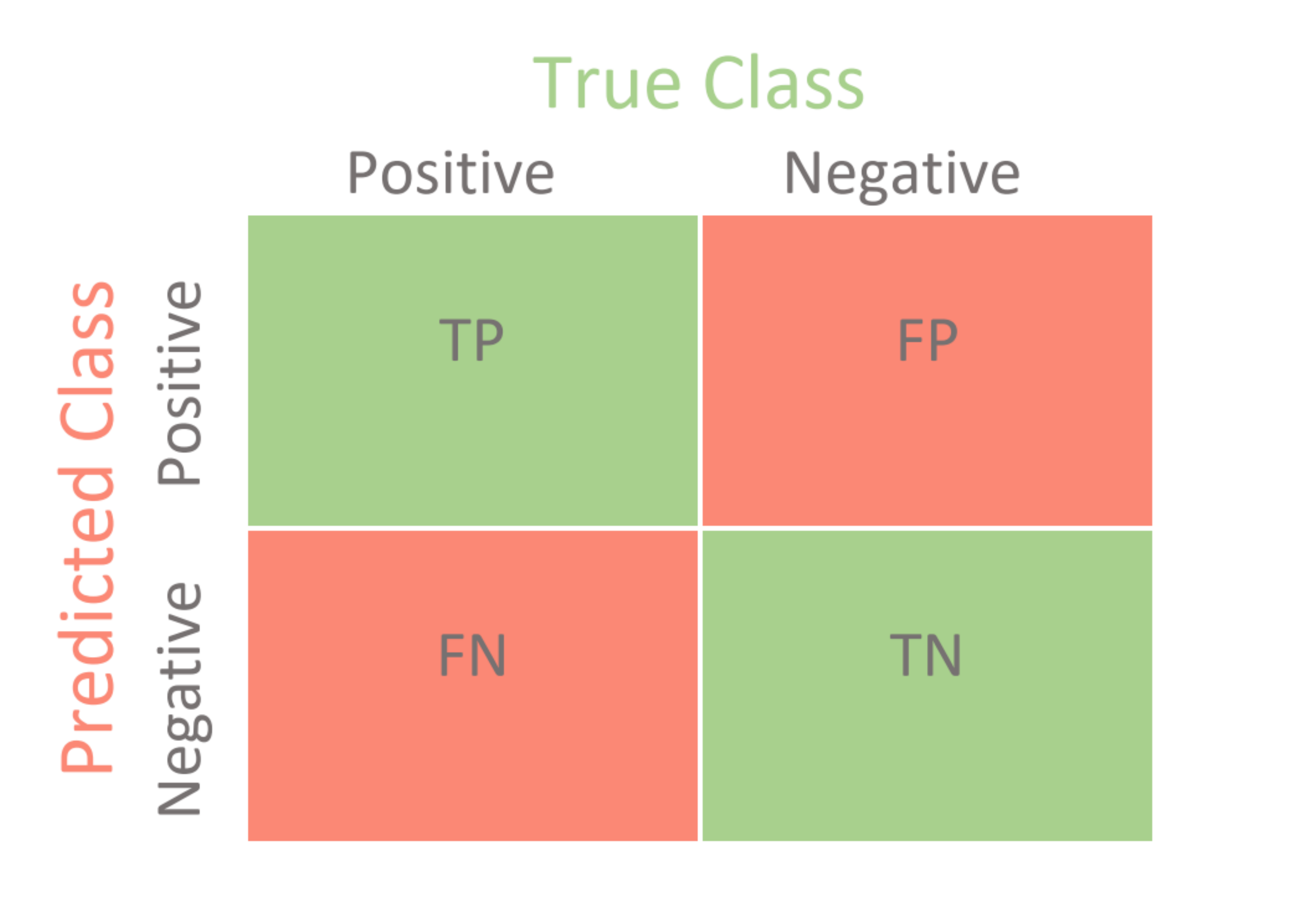
Classification

- Answer = true_value
- Prediction from model = pred_value,
- PERFORMANCE of the model is f(true_value, pred_value)
- There are many fs

Model Evaluation

Classification

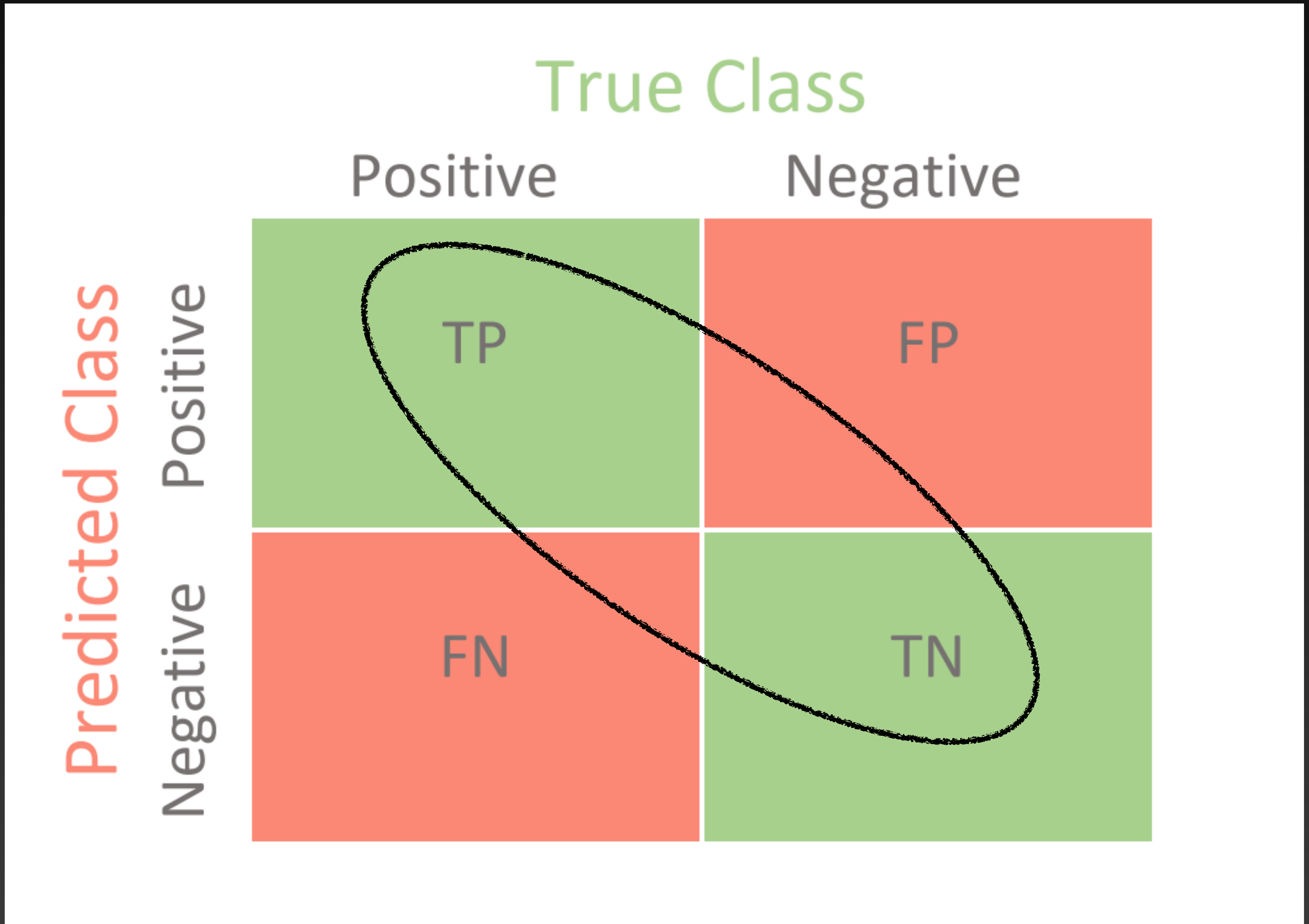
- In a binary classification setting:
  - True positive = 有病確診
  - True negative = 沒病回家
  - False postive = 冤枉被關
  - False negative = 出去害人
- Confusion matrix



Model Evaluation

Classification

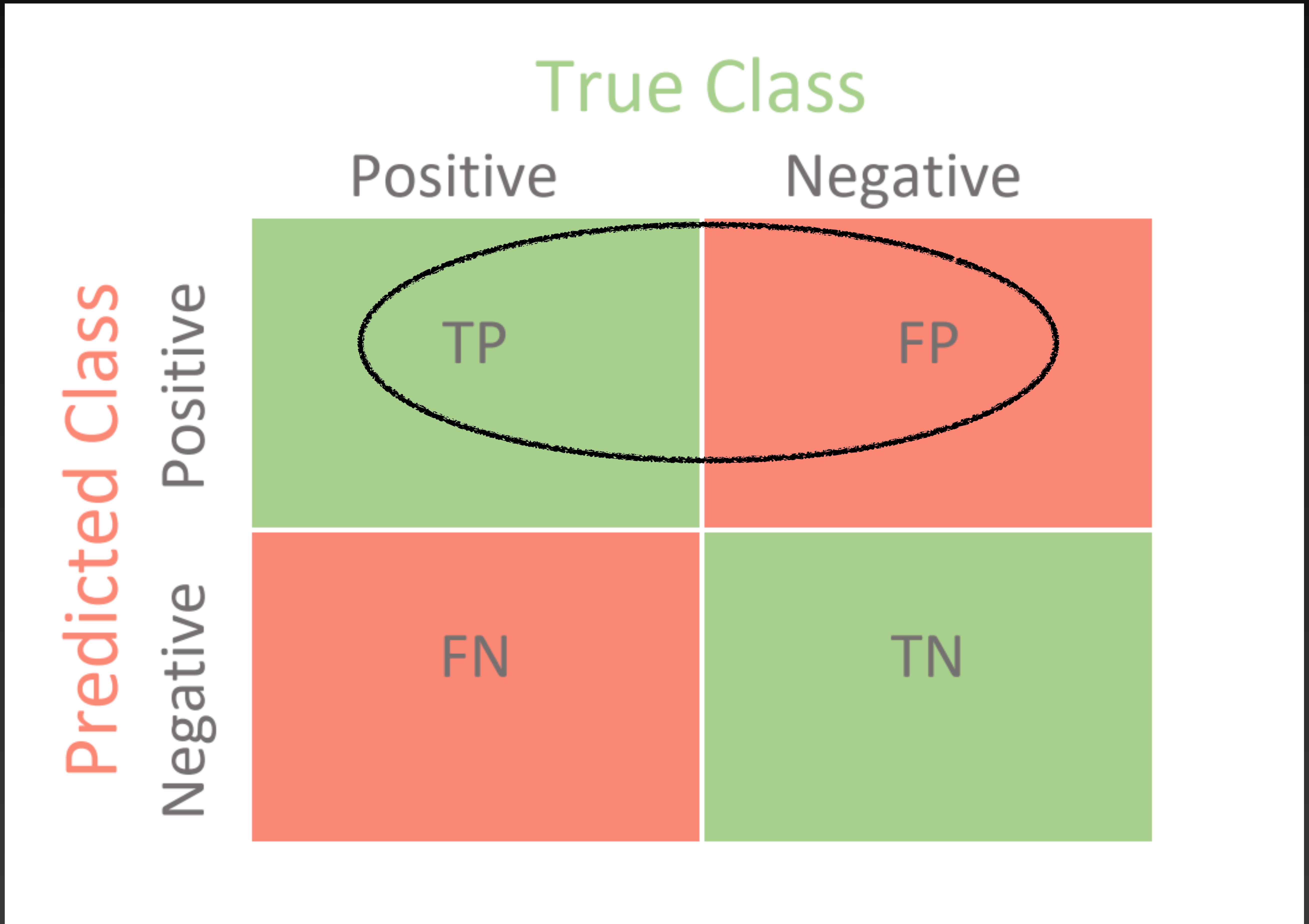
- Accuracy = (TP + TN) / ALL
- 「有預測正確的比例」
- The most common f
- 「如果真實答案有99%在一個類別，1%在另外一個」
- => 99% accuracy with a no-brain guess



Model Evaluation

Classification

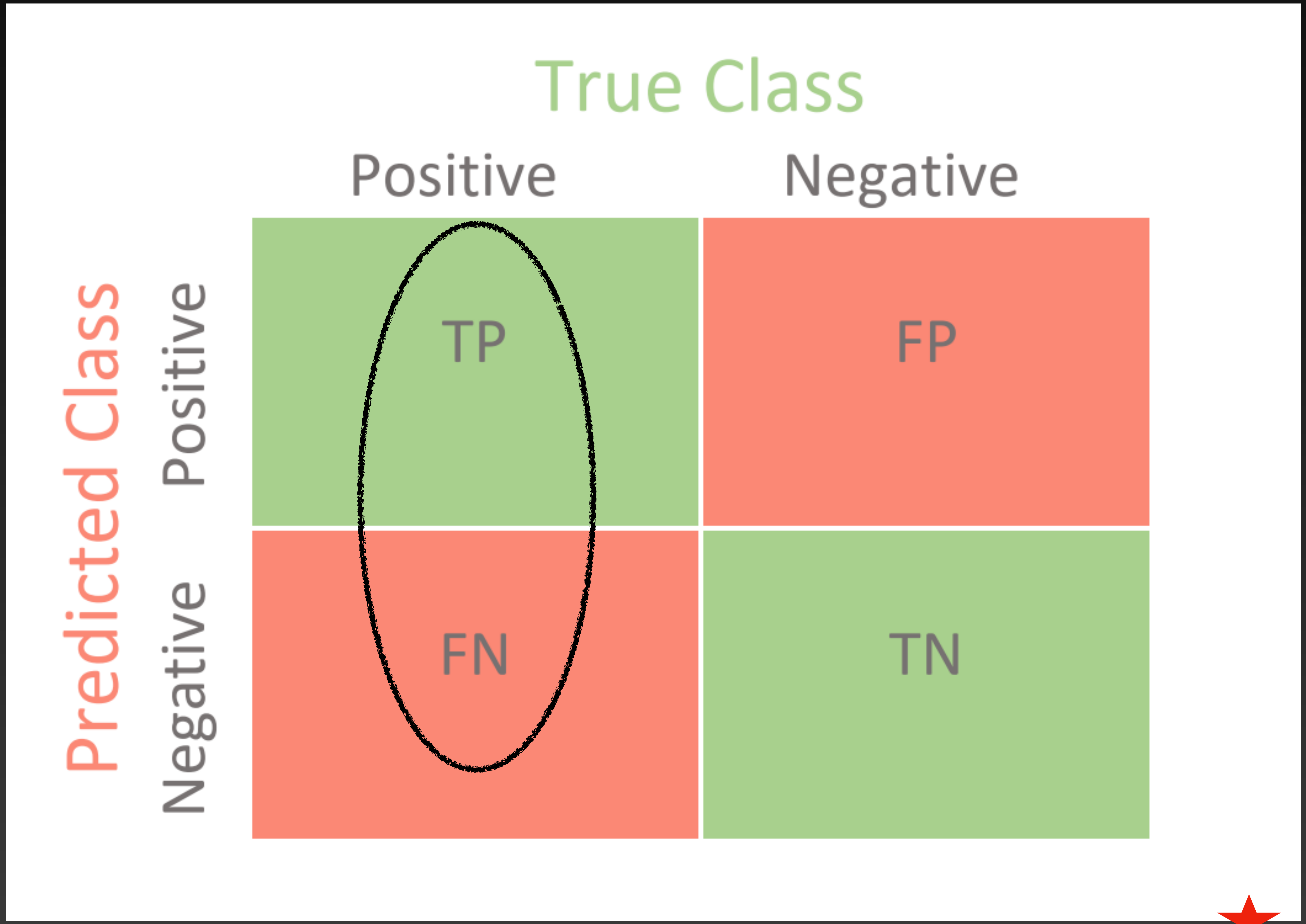
- Precision = TP / (TP + FP)
- 「預測成True的人裡面，實際是True的比例」
- i classes, i precision values
- 「快篩會不會害很多人白白被抓去隔離？」



Model Evaluation

Classification

- Recall = TP / (TP + FN)
- 「實際是True的人裡面，有預測成True的比例」
- => 跟Precision概念相反
- 哪一個好，端看業務需求
- also called Sensitivity



Model Evaluation

Classification

- Recall of True= Sensitivity = TP / (TP + FN) = True positive rate
    - 「實際是True的人裡面，有預測成True的比例」
- Recall of False = Specificity
    - 「實際是False的人裡面，有預測成False的比例」
- 1 - Specificity = FP / (TN + FP) = False positive rate
    - 「實際是False的人裡面，沒有預測成False的比例」

Model Evaluation

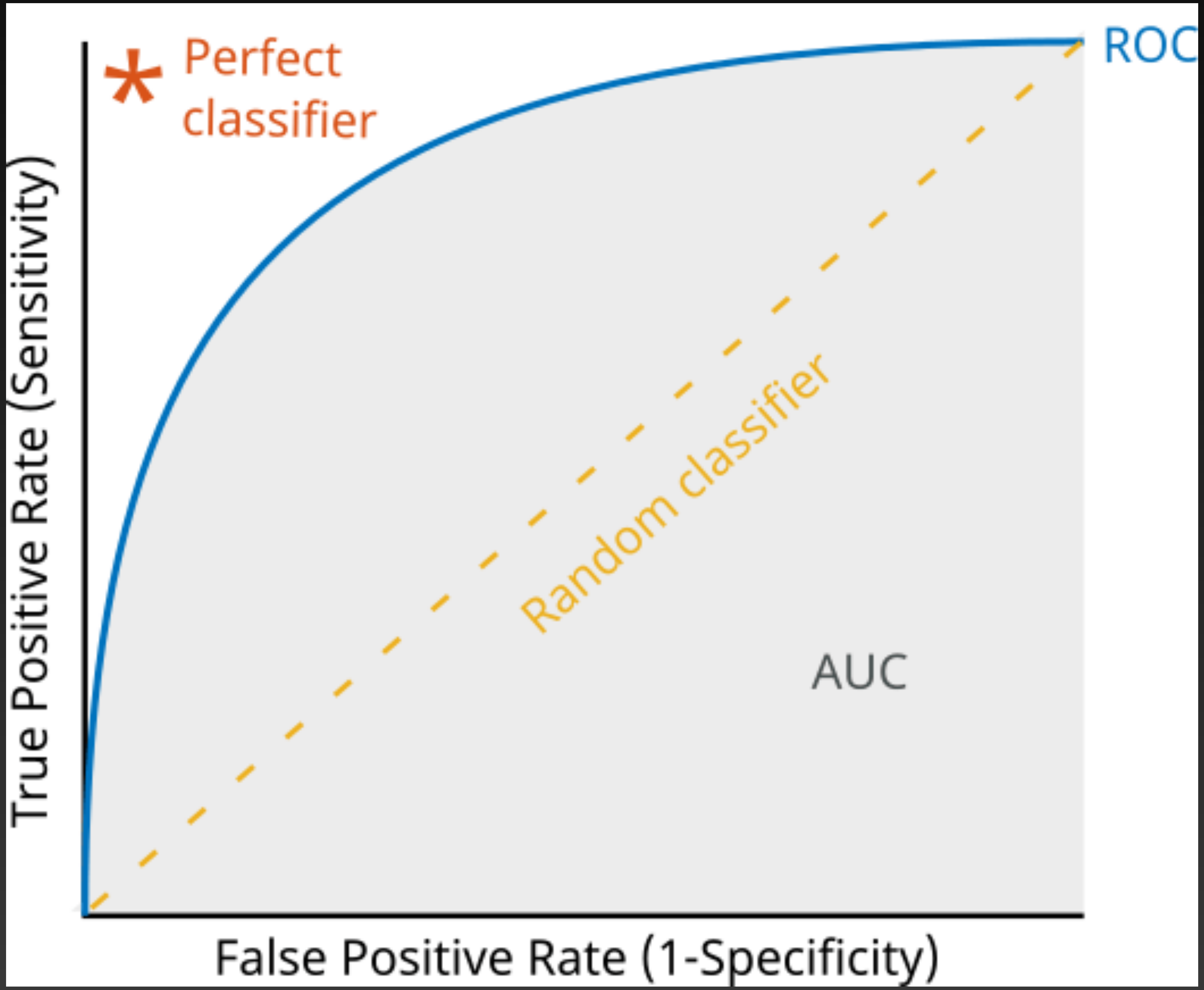
Classification

- IF pred_value > 0 THEN True
    - All predicted as True => FN = TN = 0, TPR = FPR = 1
- IF pred_value > 1 THEN True
    - All predicted as False => TP = FP = 0, TPR = FPR = 0
- IF pred_value > 0.5 THEN True
    - ?

Model Evaluation

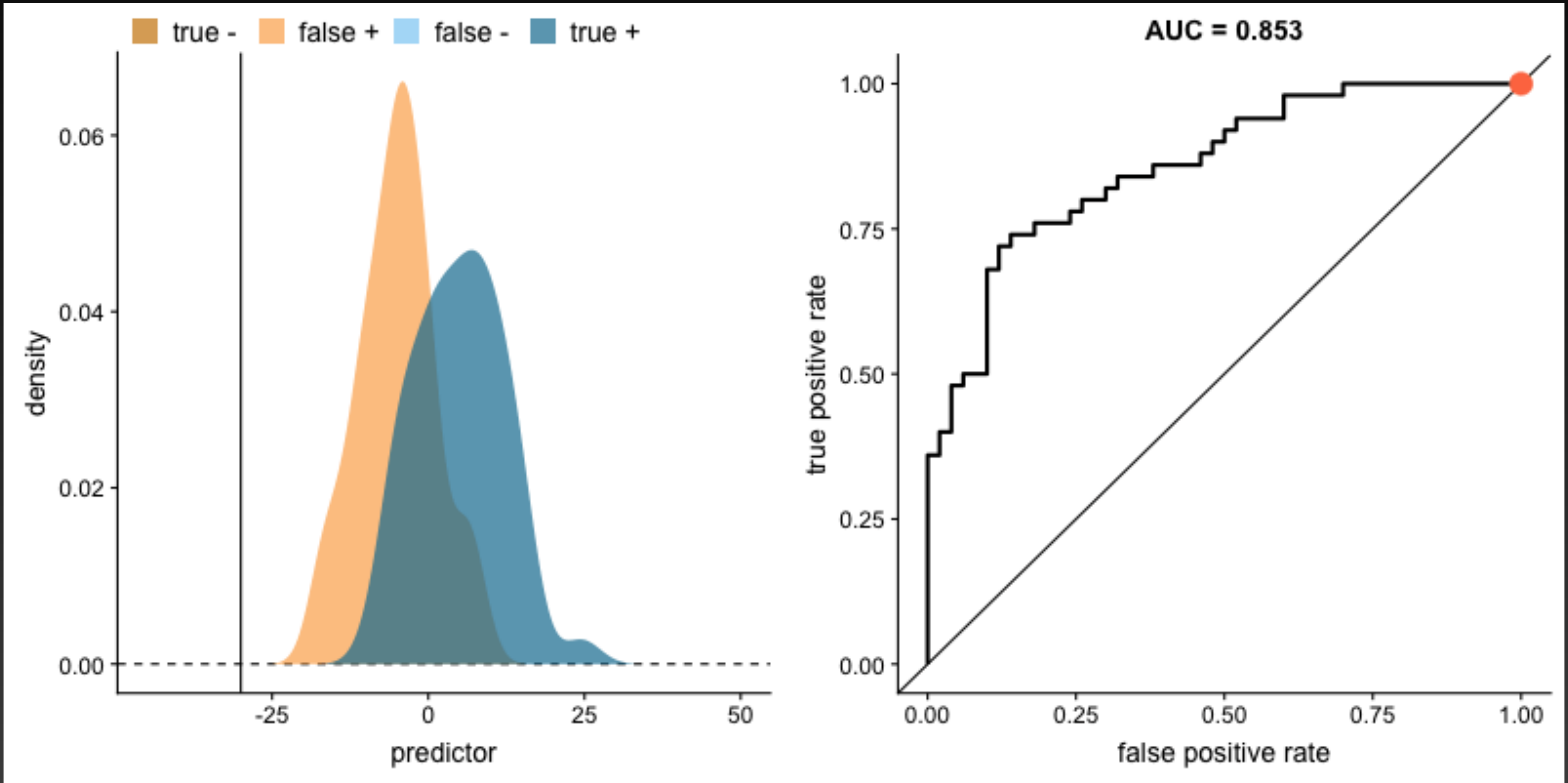
Classification

- X = FPR, Y = TPR. Go through all the threshold values, we will get a curve: receiver operating characteristic (ROC) curve
- AUC: area under the curve



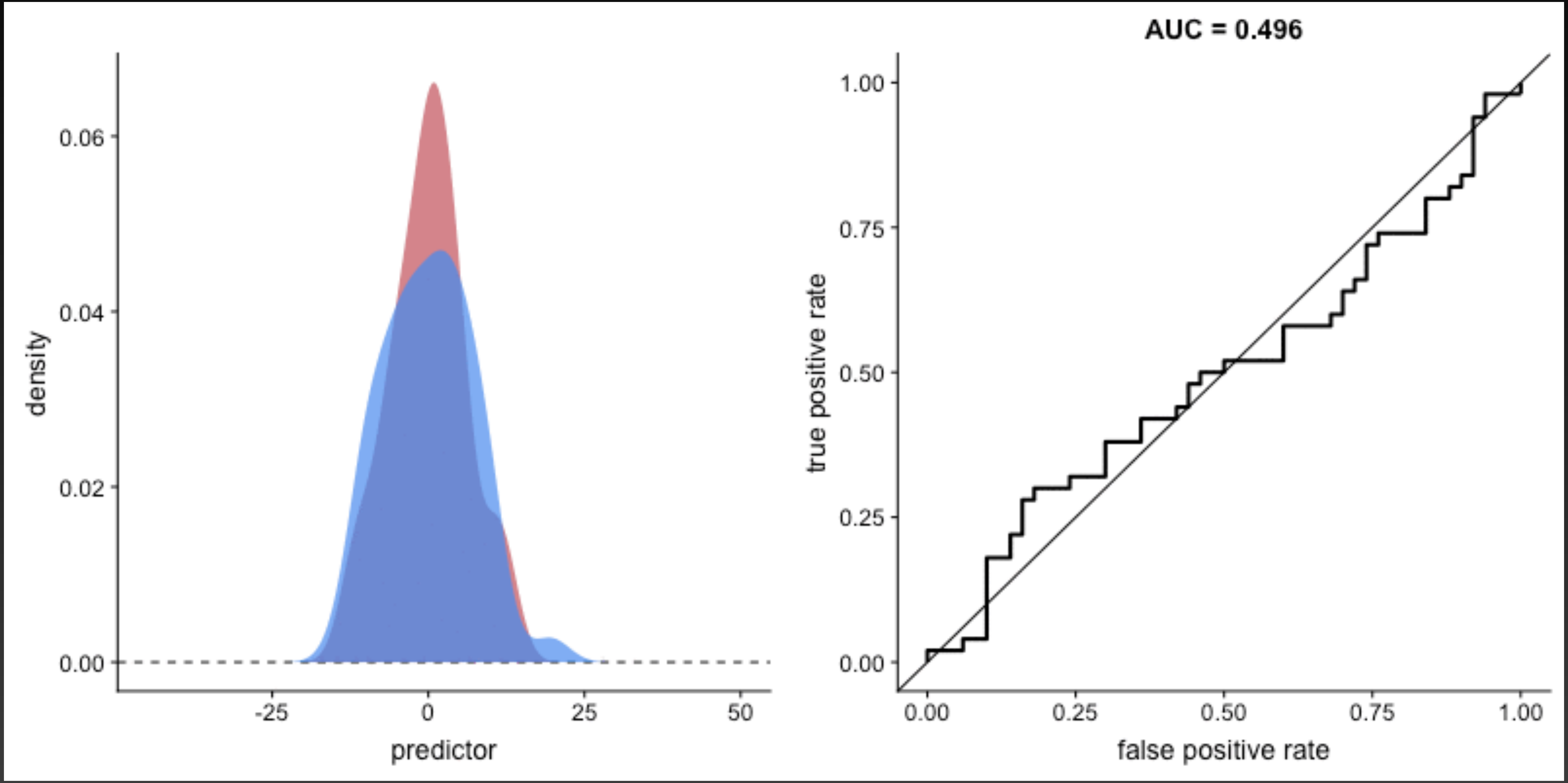
Model Evaluation

Classification



Model Evaluation

Classification



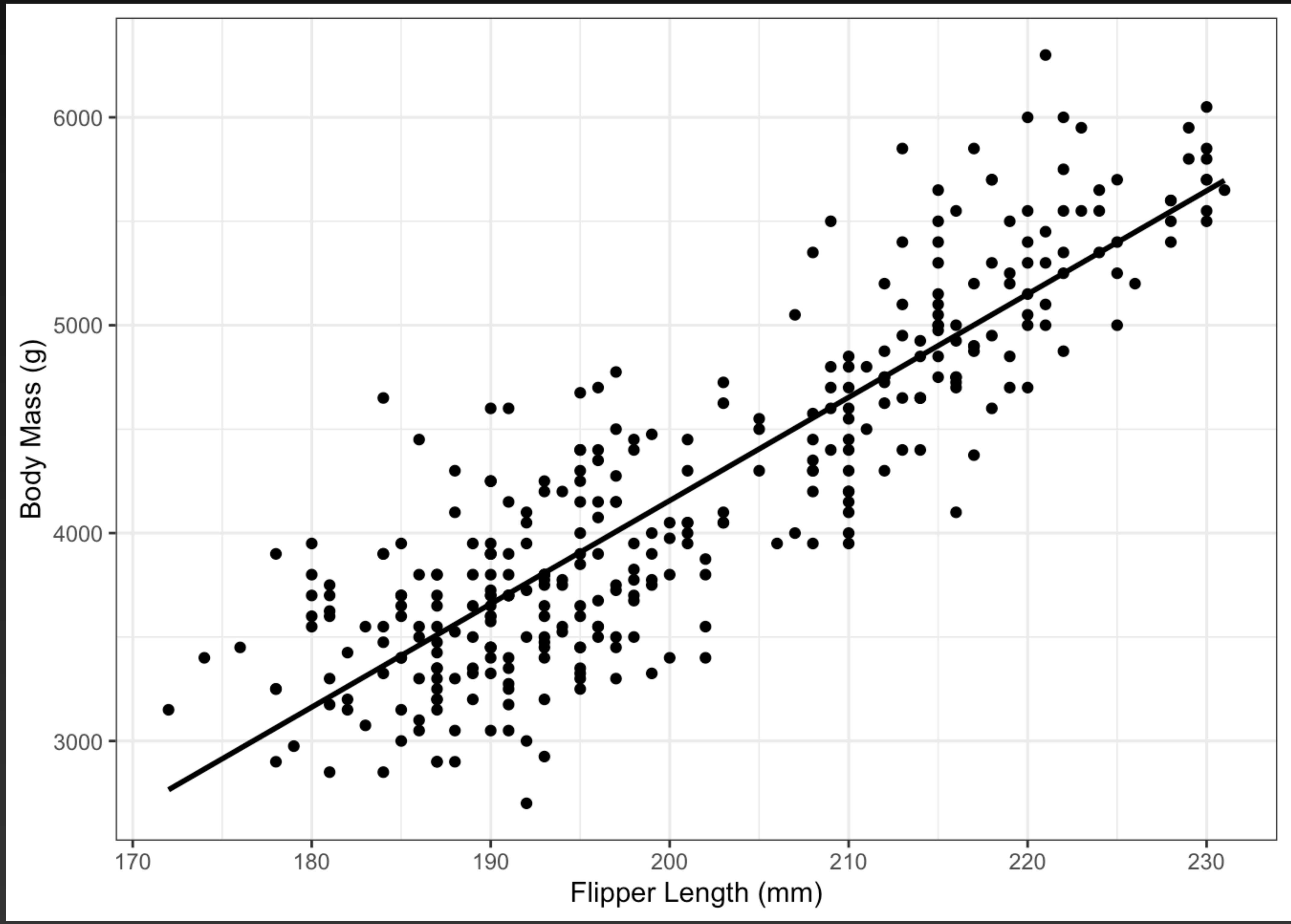
Any question?


他的戰鬥能力有
5000那麼多


我的戰鬥數值就是五十三萬

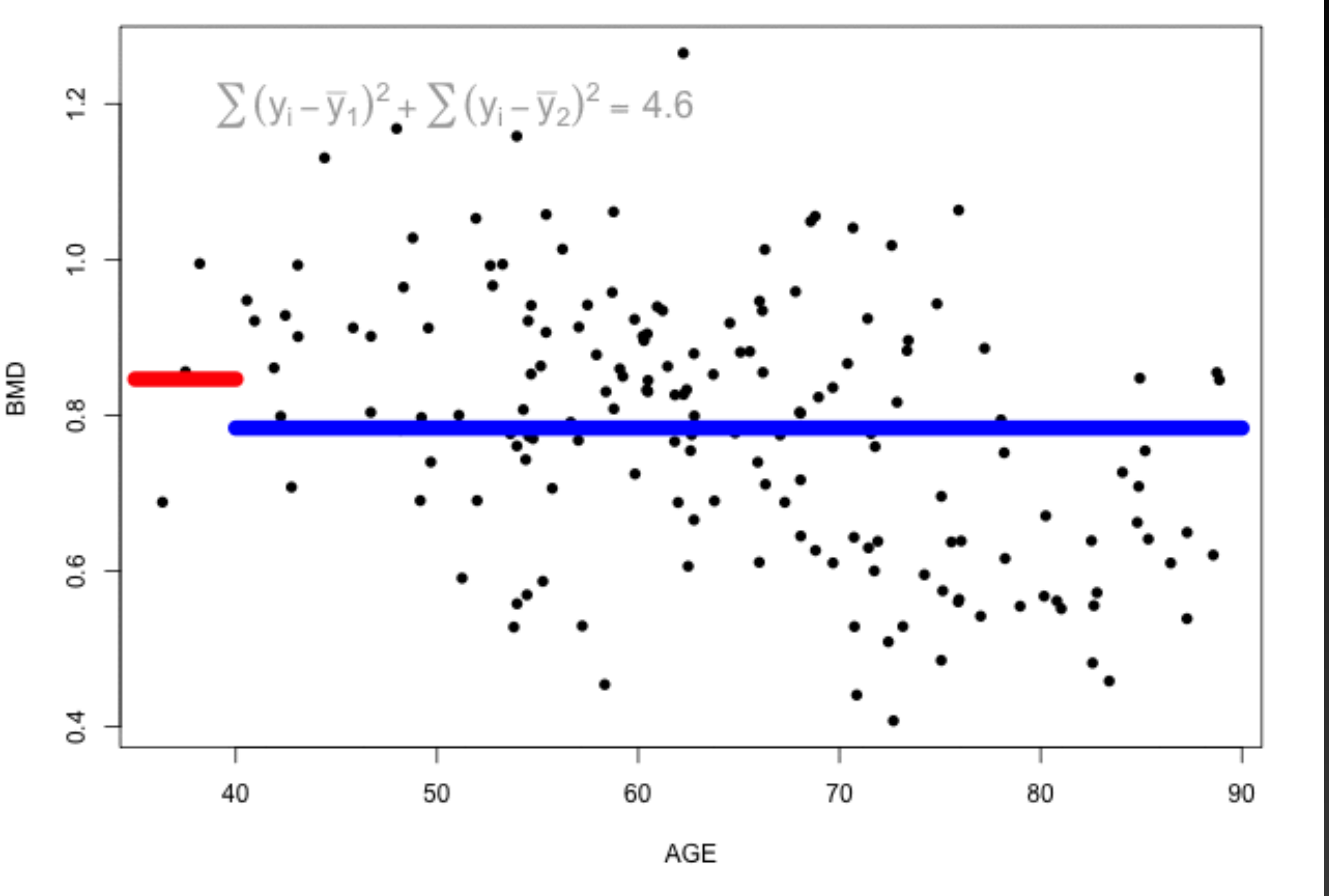
Tree-based models

- You know the power of tree-based models now
- What is prediction?
  - E(y | X)
- The key concept of linear regression (and variants):
  - Link E(y | X = 1), E(y | X = 2), E(y | X = 3)….. together, in a linear way

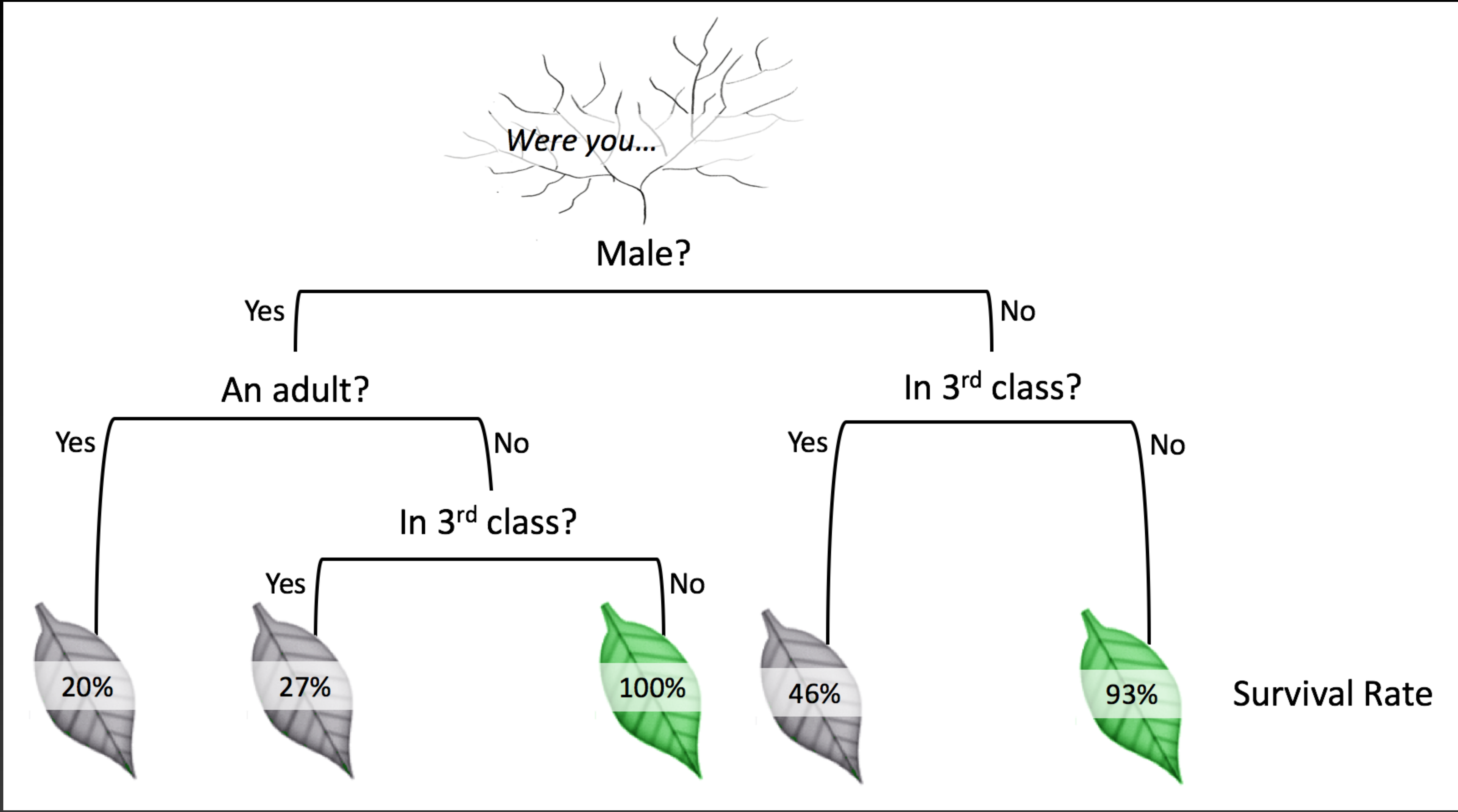


Tree-based models

- The key concept of decision tree
    - Find an optimal cut point “X = k” to maximize abs[E(y | X > k) - E(y | X <= k)]
    - Link all X >= k together, assign predicted value E(y | X >= k), and vice versa
    - Keep cutting until reach certain condition



$\sum(y_i - \bar{y}_1)^2 + \sum(y_i - \bar{y}_2)^2 = 4.6$
BMD
1.2
1.0
0.8
0.6
0.4
40
50
60
70
80
90
AGE

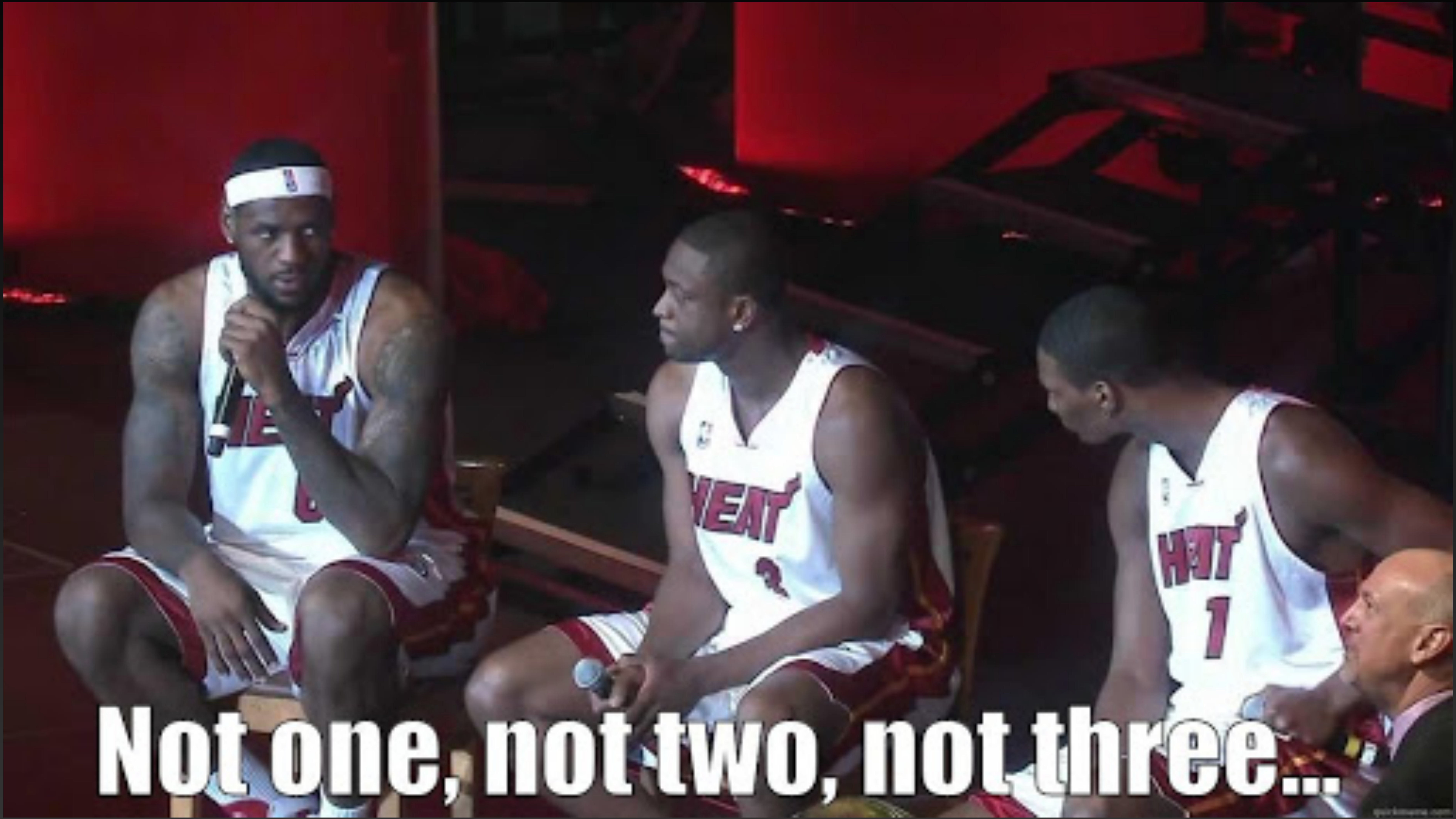

Were you...
Male?
Yes
No
An adult?
Yes
No
In 3rd class?
Yes
No
In 3rd class?
Yes
No
20%
27%
100%
46%
93%
Survival Rate

Decision Tree

- Recall all the problems we need to handle when working with linear model
    - Missing values
    - Categorical variables
    - Non-linearity & interaction
    - Outliers (!)
- In many cases, you don’t have to worry much about them with decision tree
- Why?

Decision Tree

- It sounds magic right?
  - Easy to understand, not much feature engineer work
- A decision tree has its own inherent problems
  - If we let the tree growth to its limit, it produce unstable and bumpy predictions
  - a.k.a Overfitting (Good when training but poor in test)
  - So it needs to be pruned …..even so……


Not one, not two, not three...

How about a forest?

- 三個臭皮匠，勝過一個諸葛亮
- Economics is all about division of labor
- 一個和尚挑水喝，兩個和尚抬水喝，三個和尚沒水喝


父親
間諜
母親
職業殺手
女兒
超能力者

How about a forest?

- Linear regression, decision tree are poorly performed in most cases
- Combining several poor models in a right way boosts performance!

- => Ensemble methods
- IN A RIGHT WAY
- $\sum tree = forest$

How about a forest?

- Random forest
- Having many many decision trees, average all the results
- "The definition of insanity is doing the same thing over and over again and expecting different results."
- NOT THE SAME
    - Different rows for each tree (Bootstrap sampling)
    - Different columns for each tree
- Bagging = **B**ootstrap **agg**regat**ing**

How about a forest?

- Except setting the features in the model, there is nothing we can manipulate in linear regression or decision tree
- However, for more advanced models, there are hyperparameters. A good set of hyperparameters gives you best prediction results
- To random forest:
    - The size of one tree, leaf……
    - The number of trees in forest
    - …..etc…

How about a forest?

- Remember that: a good prediction model doing well on unknown / future data, instead of training set
- So we have to split our data into two, to simulate "unknown / future"
- However, less data means poorer prediction

How about a forest?

- Random forest offers a estimation of “Accuracy on test set": oob score
- oob score enable us to skip the split and use more data in training!
- To determine a good set of hyperparameters, you can stick to the train/test splitting tradition or follow the job score

More than a forest

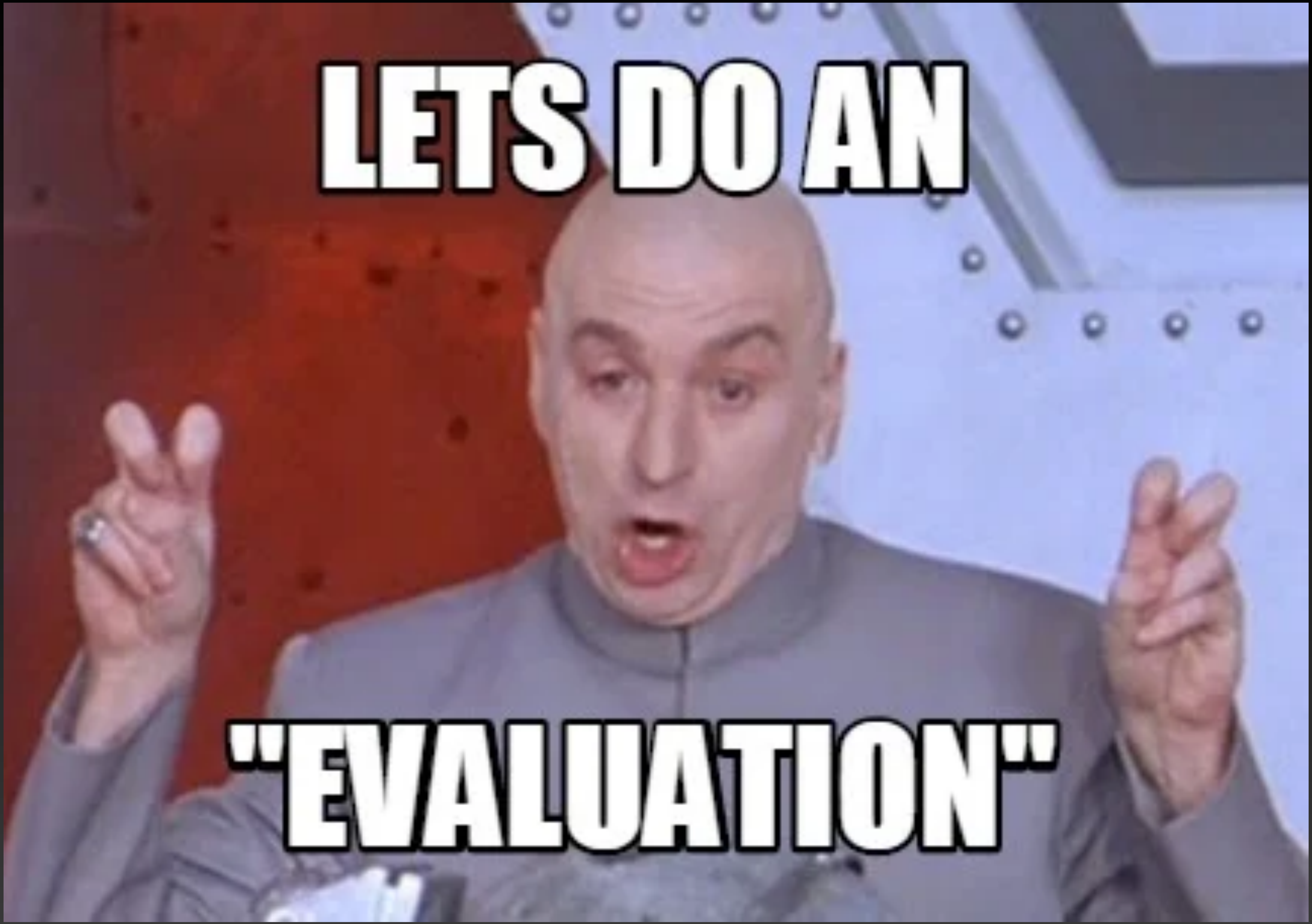
- “Predicted value” itself is valuable though
- There is more information for us when doing machine learning
    - “How is the relationship between age and being tranported?”
- With linear model?
- With tree-based model?

★

Any question?

I Can Predict The Past

- What we have so far:
  - Data: Handling with missing value, categorical variables, non-linearity, interaction, outliers…..
  - Models: Linear regression & Tree-based models
  - Evaluation: Accuracy, ROC AUC
    - oob score
- It’s already a simple but complete machine learning pipeline

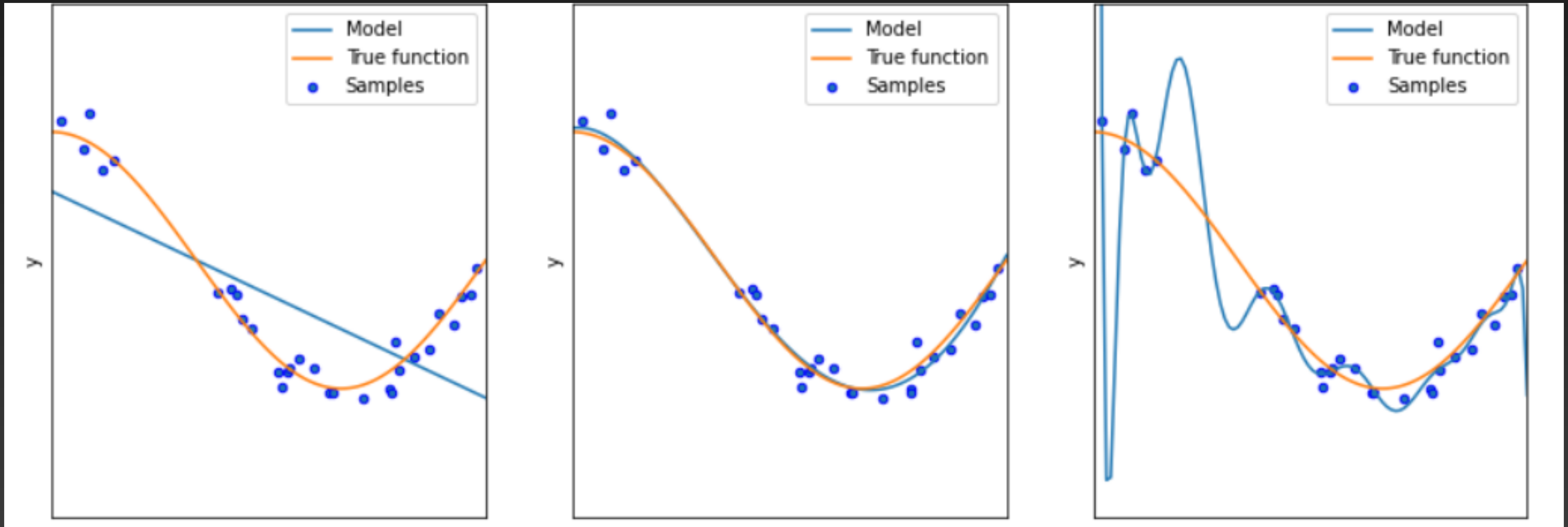

LETS DO AN
"EVALUATION"

I Can Predict The Past

- In the section of decision tree, I've mention the idea of "overfitting"
- Recall that: a good prediction model doing well on unknown / future data, instead of training set
- Data = Pattern + Noise
- The unknown/future data may be similar with training data. Not identical.
- Performance on unknown data: Generalization error

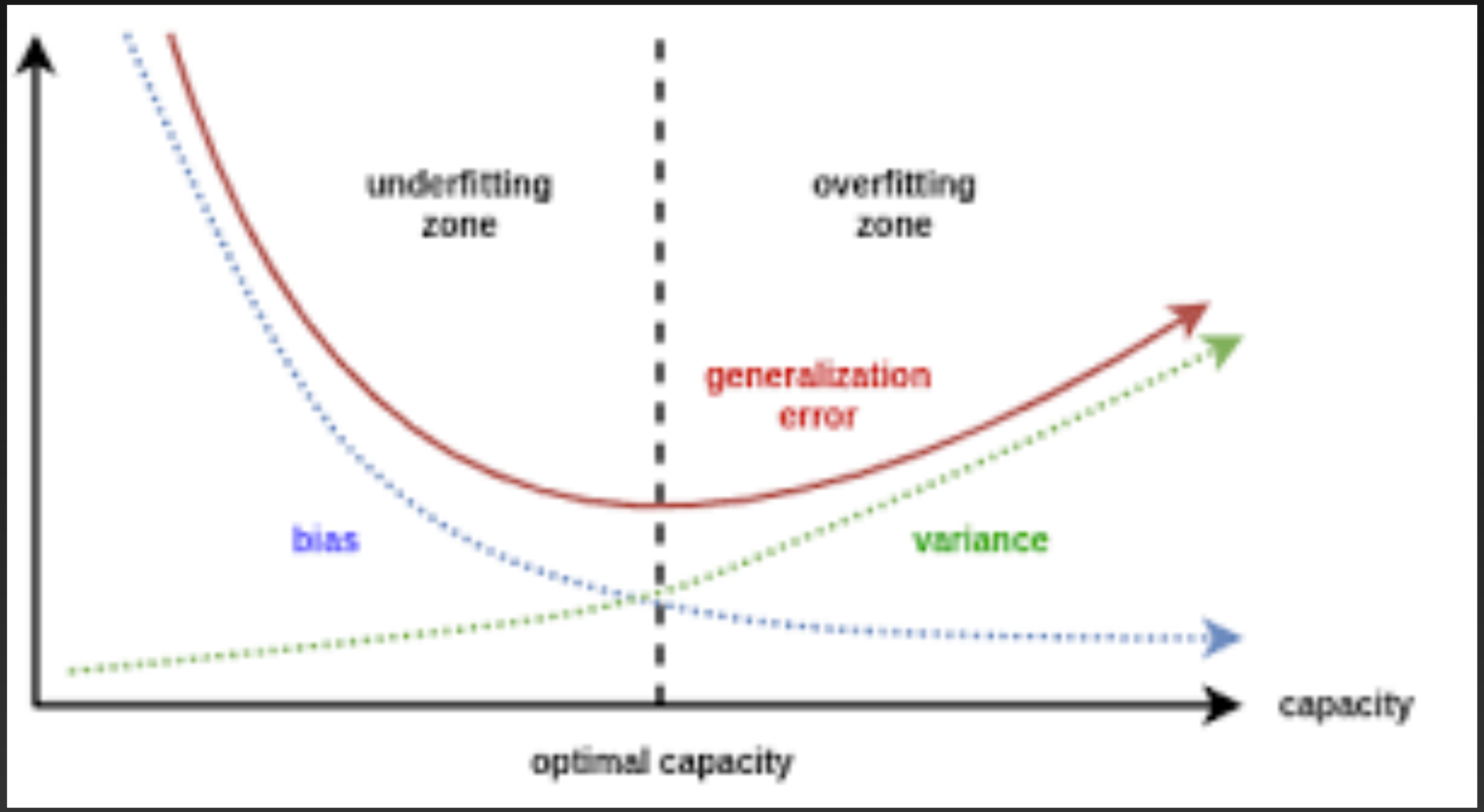
I Can Predict The Past

- Overfitting: Being obsessed with training data, even with some special cases or noisy ones
- Overfitting & Underfitting



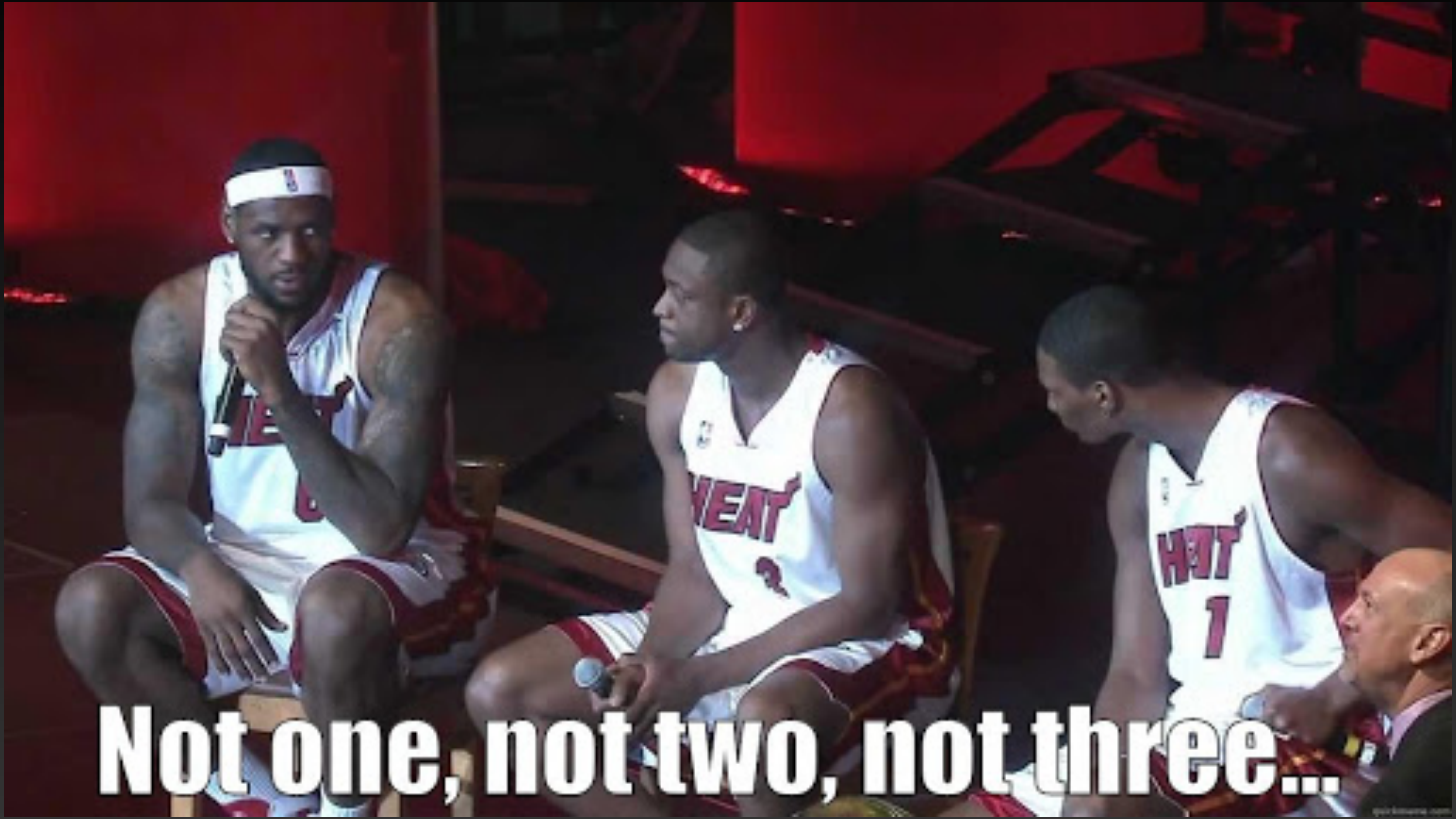
I Can Predict The Past

- Another way to understand this issue: Bias–variance tradeoff
- Bias: being “wrong”
- Variance: being “unstable”
- Overfitting: Low bias (in training), High variance (in testing)
- Underfitting: High bias, Low variance



I Can Predict The Past

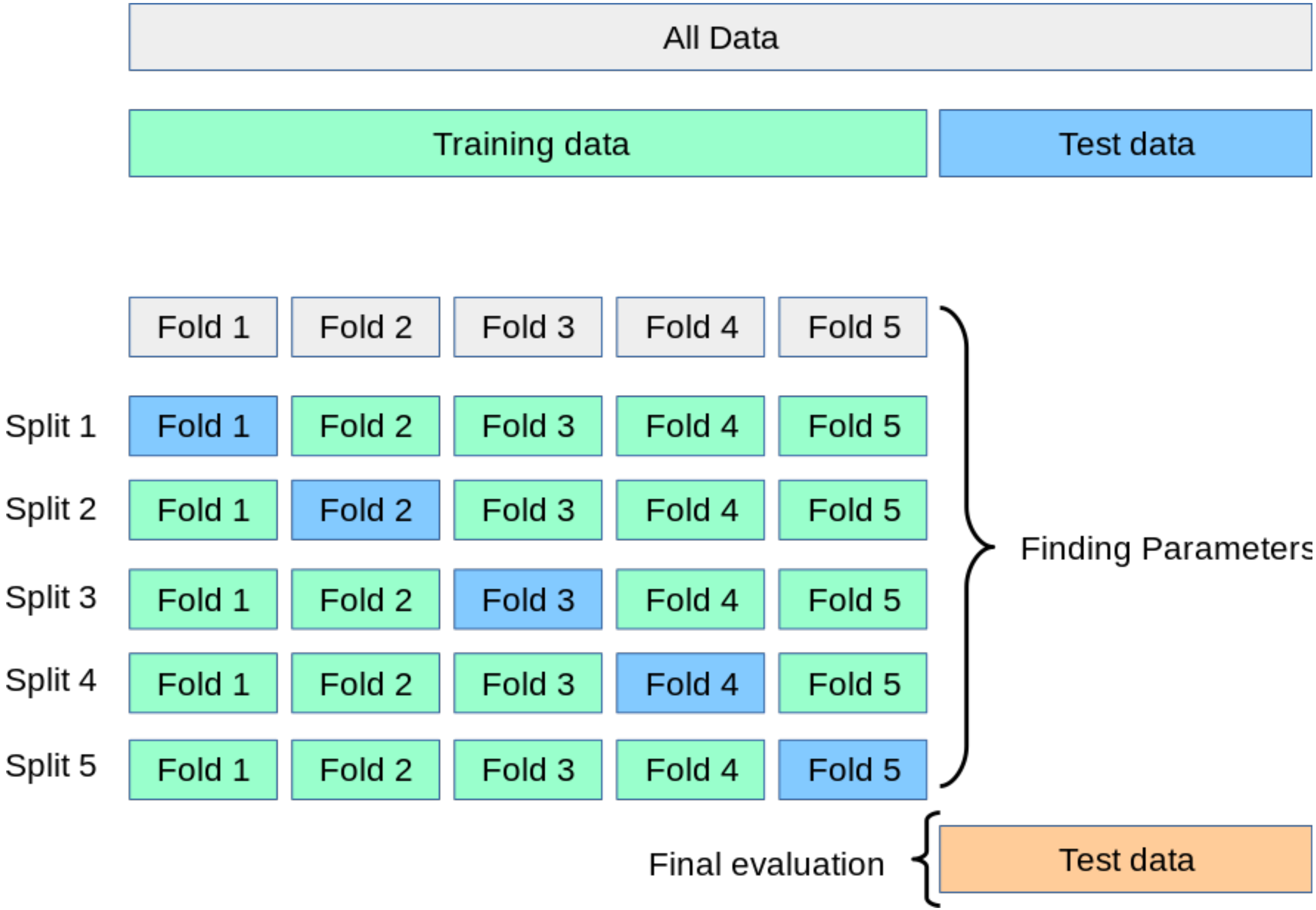
- What do we have to deal with it?
    - Train/test split
    - Evaluate performance with test data (= Imitation of generalized scene)
- It’s fine when we just want to compare between few models
- The number of models comes up?
    - Hyperparameter searching
- Now you’re obsessed with the test data!


HEAT
HEAT
HEAT
1
Not one, not two, not three...

I Can Predict The Past

- Before: Train with training set. Evaluate on test set
- After: Train with partial training set. Evaluation first on the remaining training set. Repeatedly. Evaluate on test set, finally

- Cross-Validation
- (In some application only) Out-of-bag score


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Parameters
Final evaluation
Test data

I Can Predict The Past

- Before: Train with training set. Evaluate on test set
- After: Train with partial training set. Evaluation first on the remaining training set. Repeatedly. Evaluate on test set, finally

- Cross-Validation
- (In some application only) Out-of-bag score