Applied Machine Learning

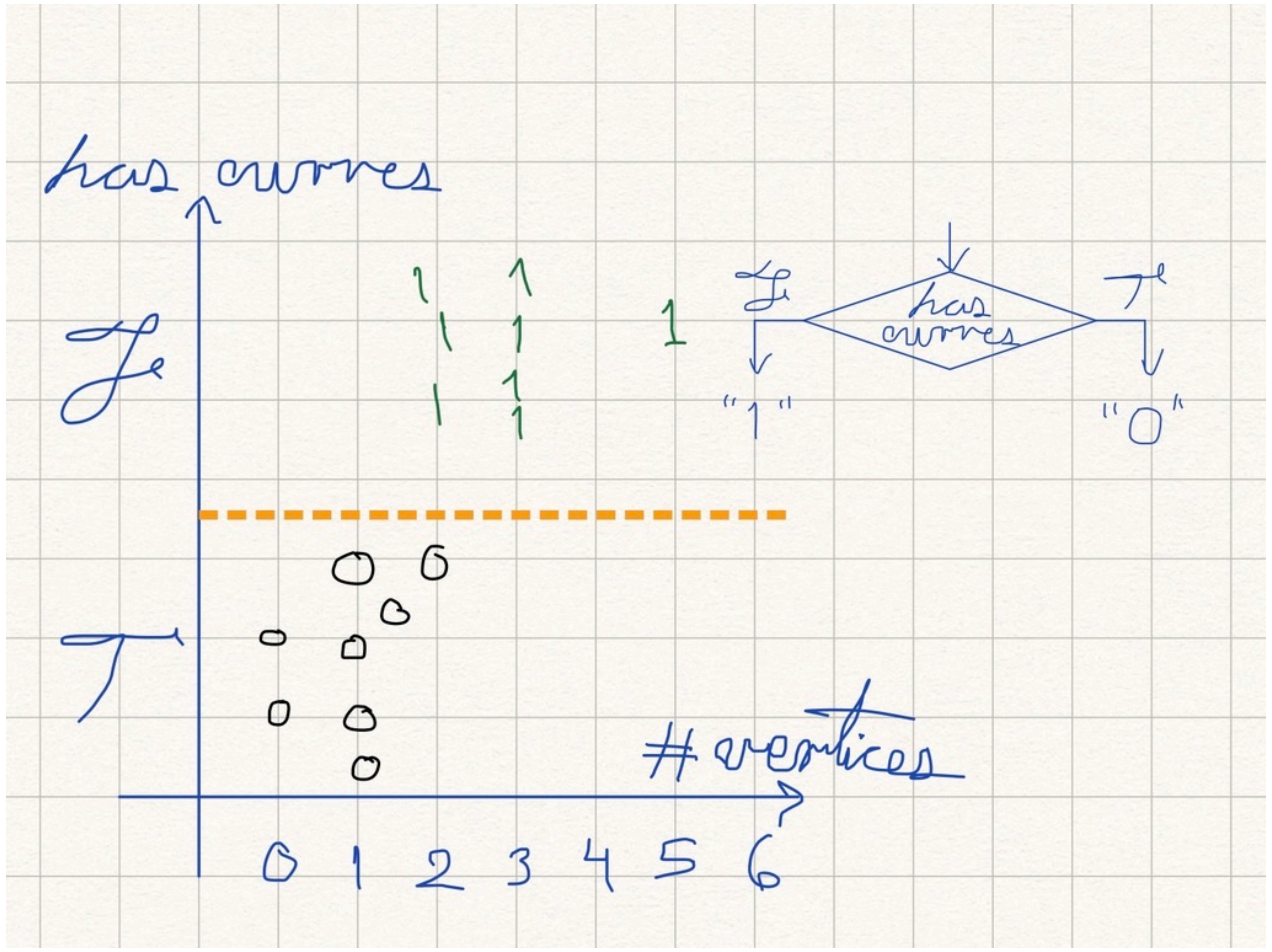
Classification - Random Forests

Random Forests

- Decision Trees
- Limitations of Decision Trees
- Random Forests

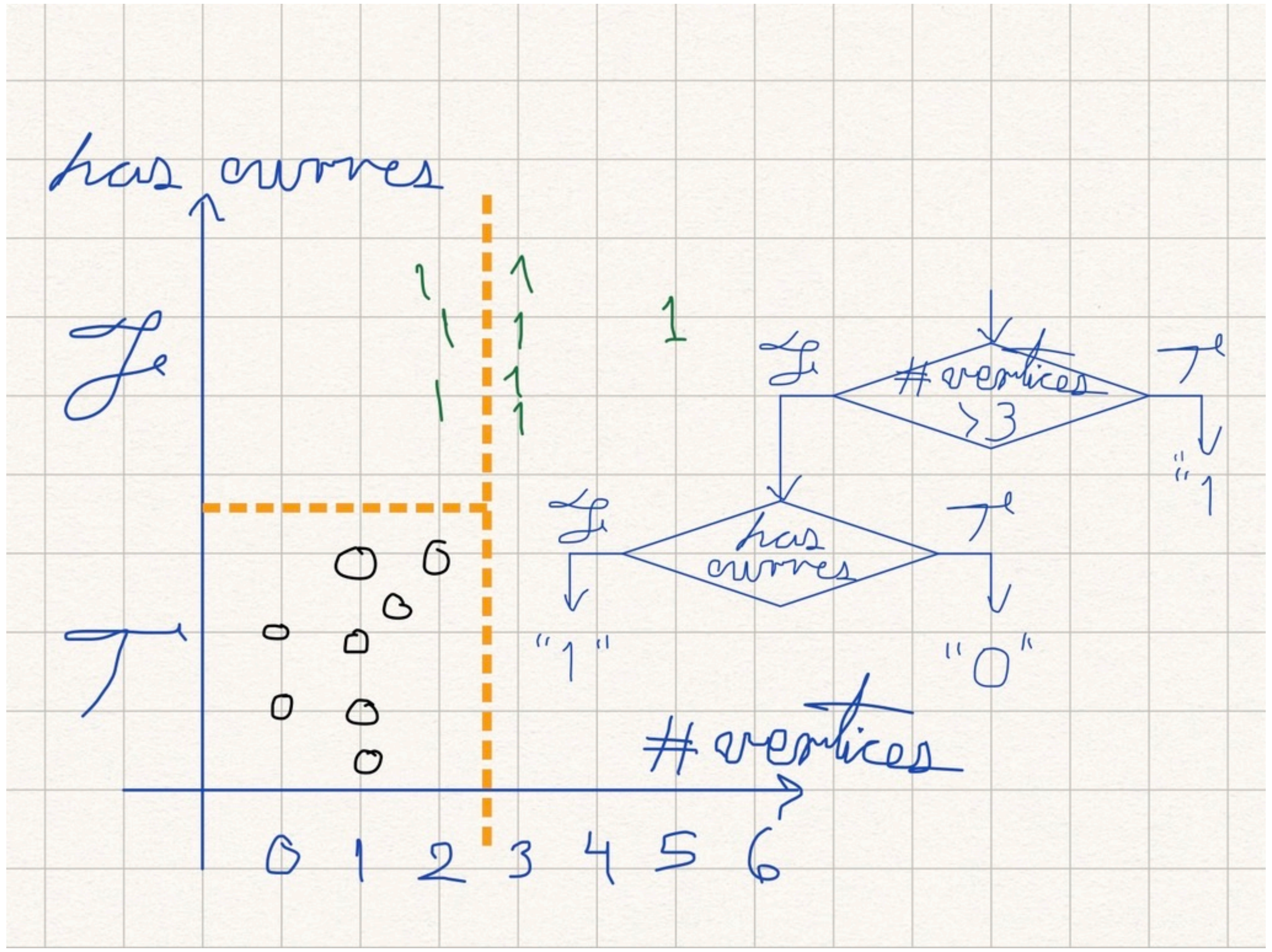
Random Forests and Decision Trees

- Based on the construction of Decision Trees
- A decision tree represents a classification function
- Tree data structure
- Not unique



Decision Trees

- Each node is a test on some input feature
- The result of the test indicates what branch to take
- Each leave represents the resulting class



Decision Trees - Features

- if-then-else rules:

  if (# vertices > 3) then

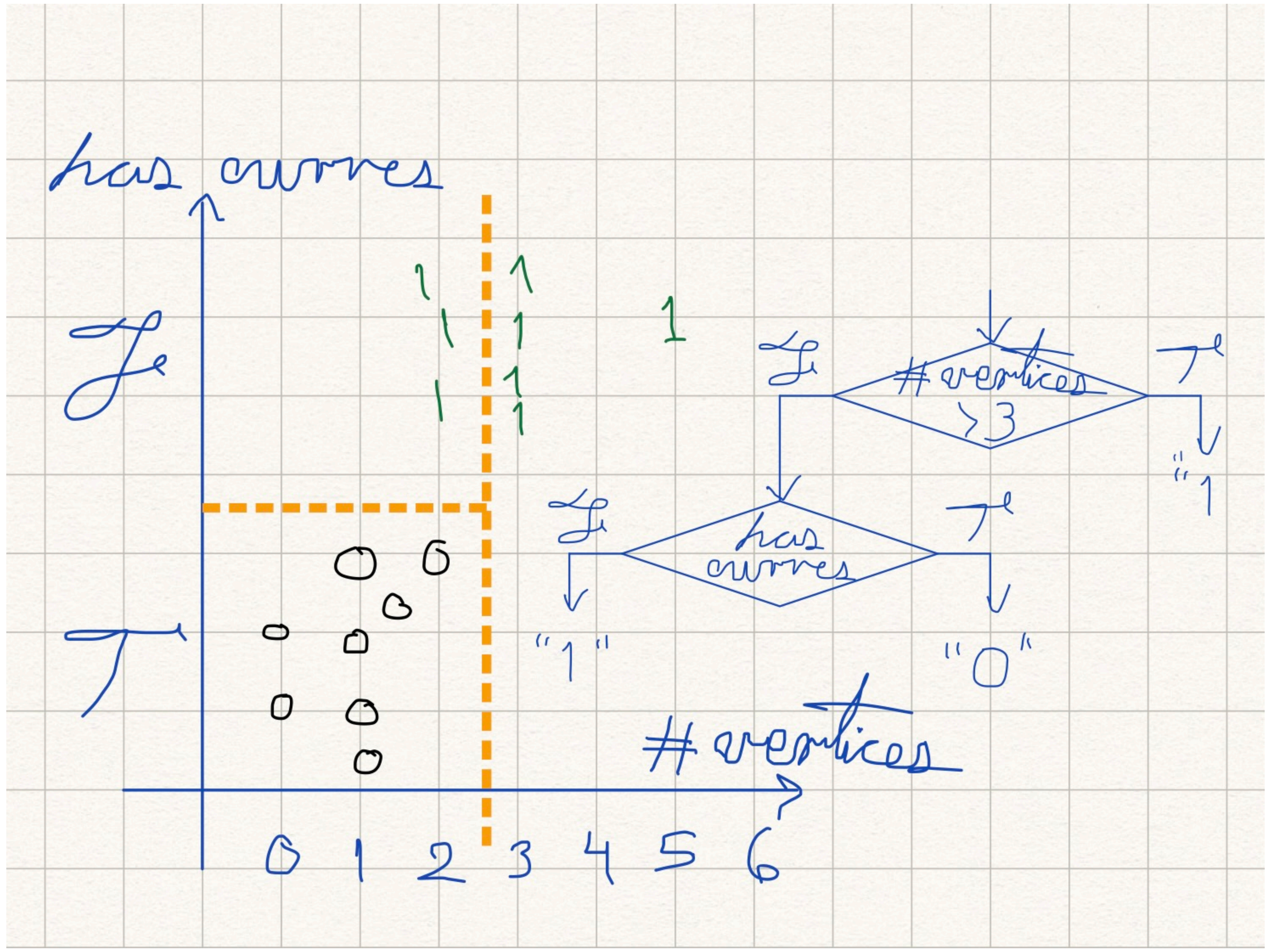
    class = “1”

  else if (has curves) then

    class = “0”

  else

    class = “1”



- "1" $= (\#\text{vertices} > 3) \vee (\#\text{vertices} \leq 3 \wedge \neg\text{has curves})$
- "0" $= (\#\text{vertices} \leq 3 \wedge \text{has curves})$

Decision Trees - Construction

- DecisionTreeExpand (branch, dataset)

  stop when

    depth(branch_node) >= max_depth

    size(dataset) <= min_leave_size

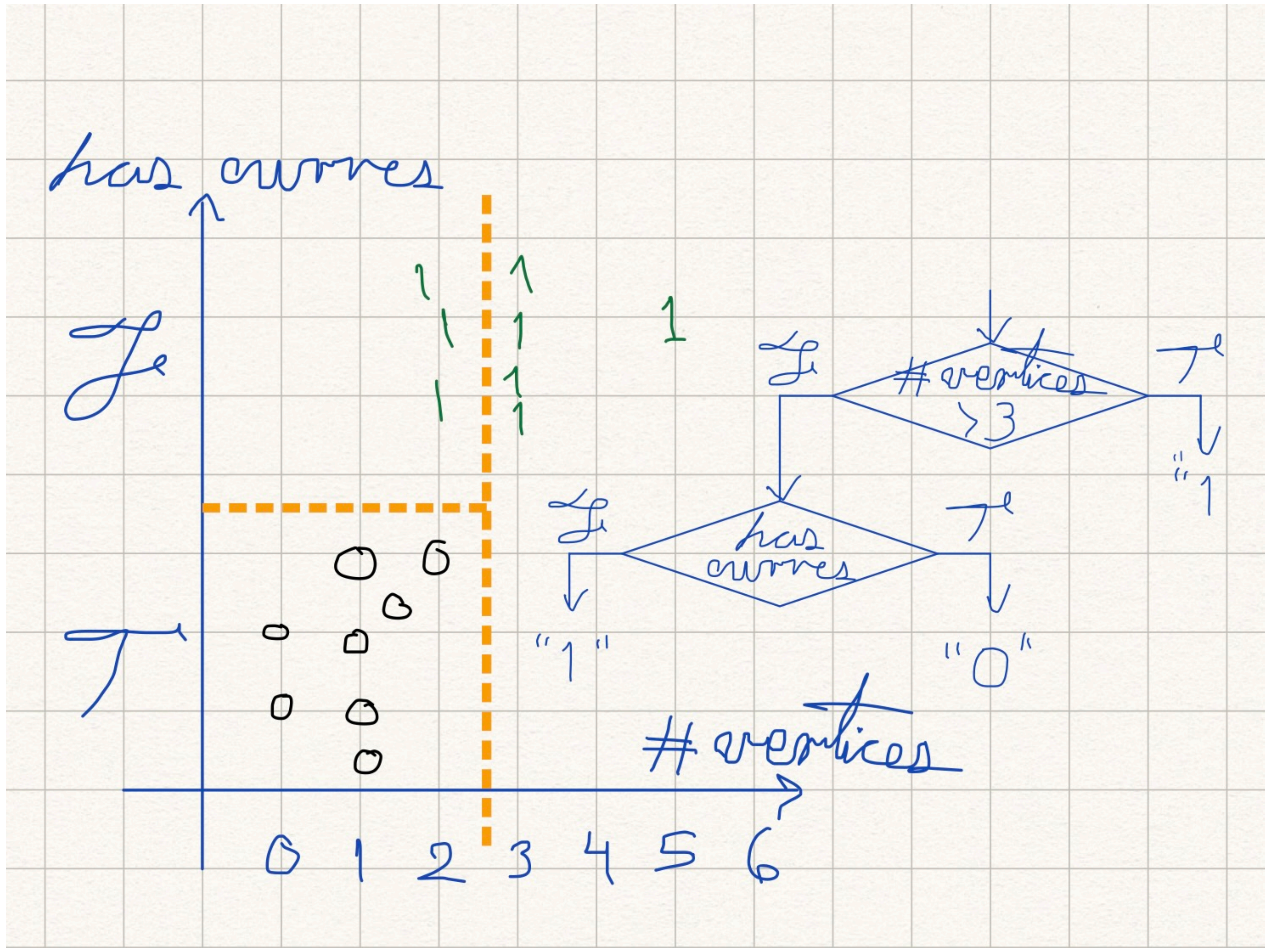
    all elements in dataset in same class

  (subset_l, subset_r, test) = best_split(dataset)

  (child_r,child_r) = new_branch(branch, test, split_l, split_r)

  DecisionTreeExpand(child_l,subset_r)

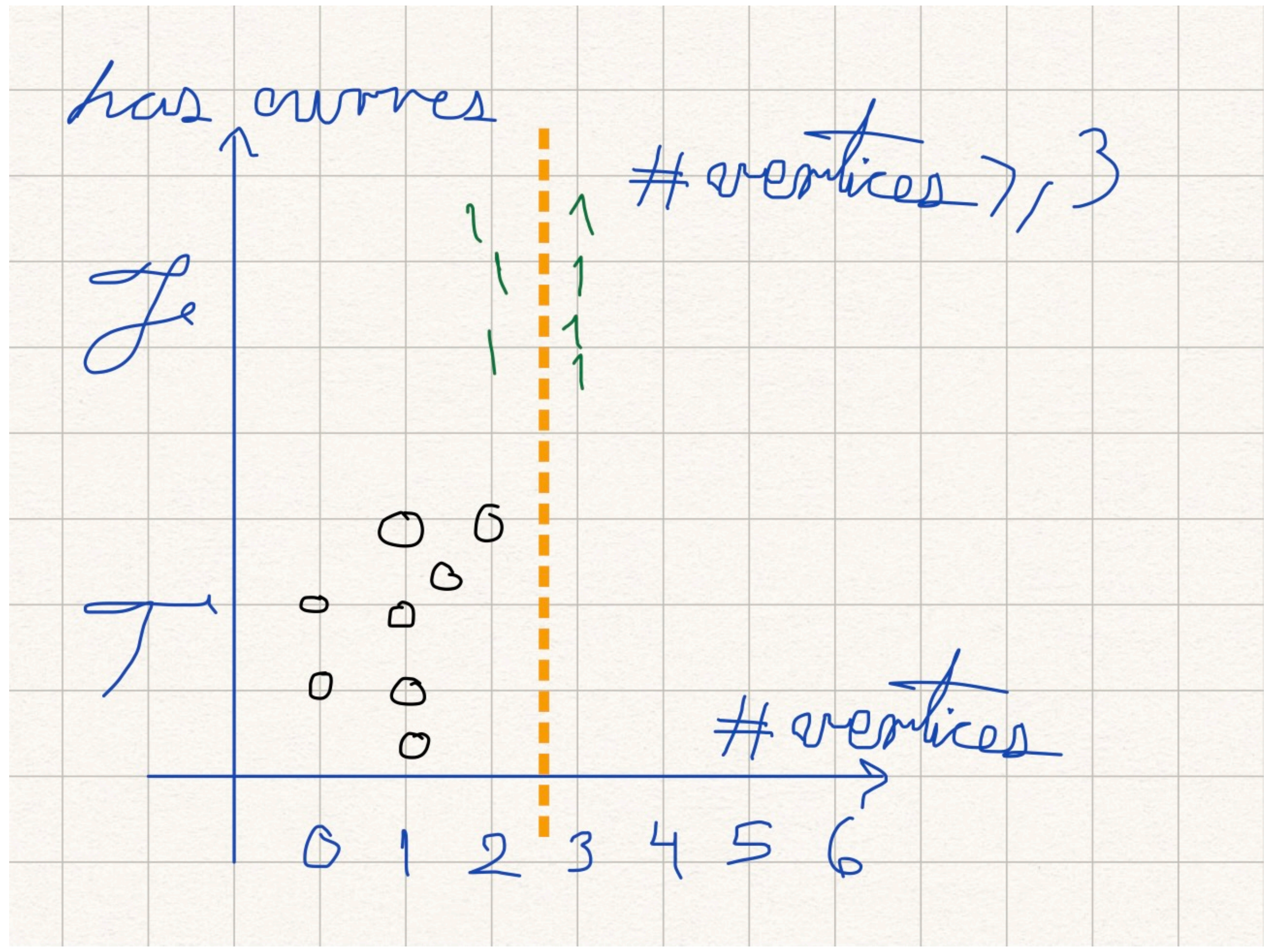
  DecisionTreeExpand(child_r,subset_r)

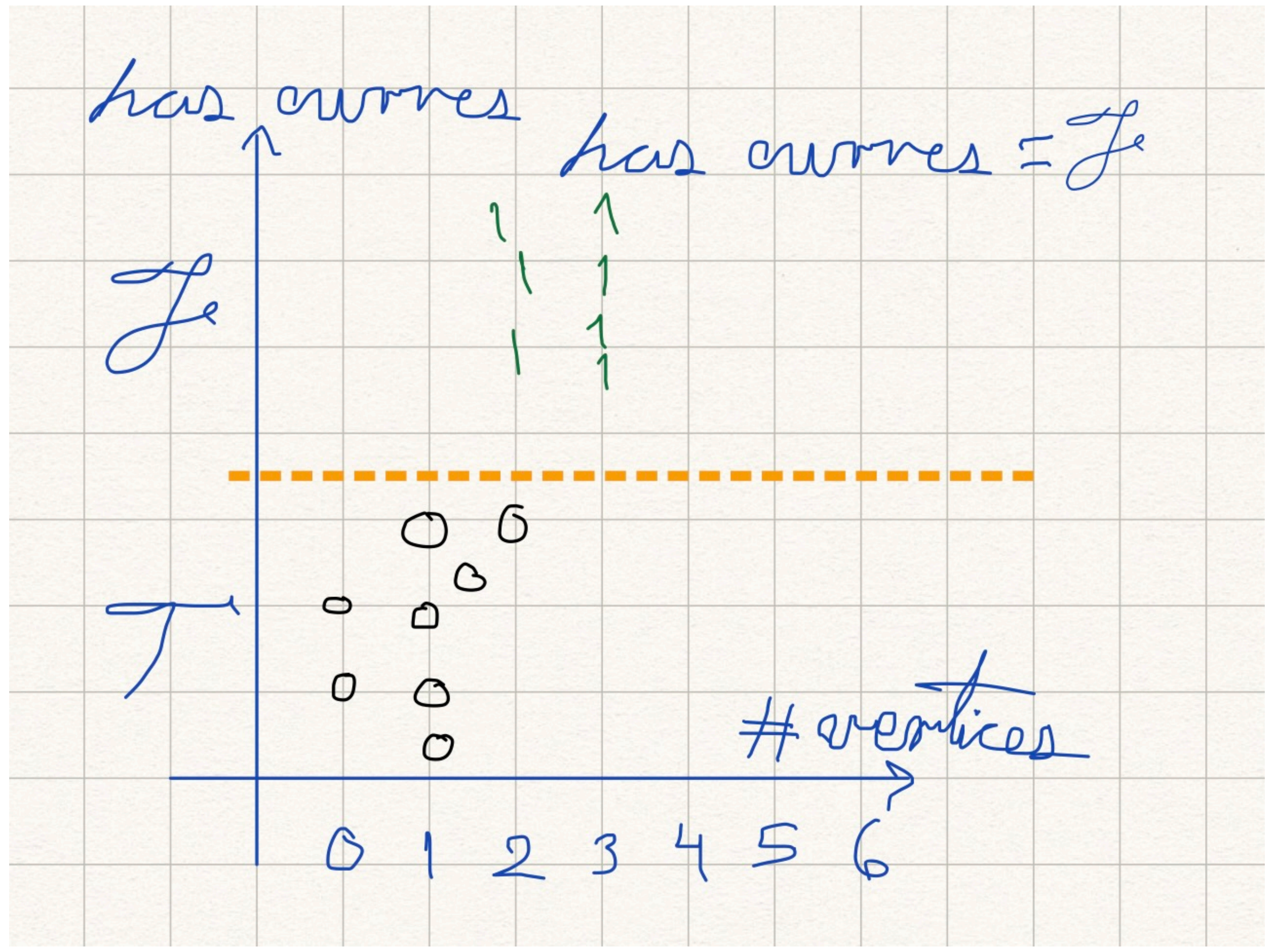


Decision Trees - Best Split

(subset_l, subset_r, test) = best_split(dataset)

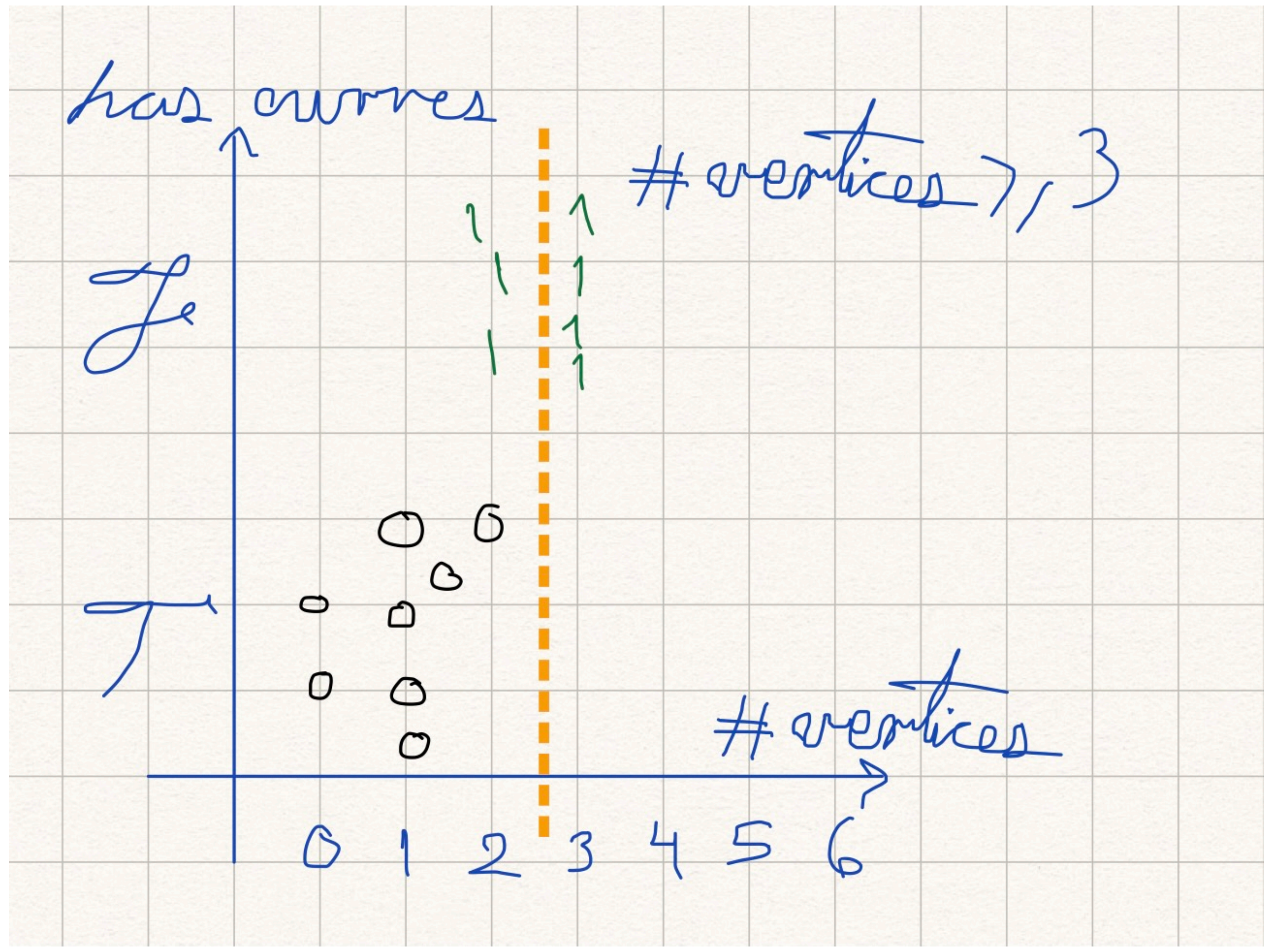
- identify potential subsets of the dataset subset_l and subset_r and a test to classify on values of feature $x^{(i)}$
- Measure Information Gain for each potential split
- Choose (subset_l, subset_r, test) that produce the highest Information Gain

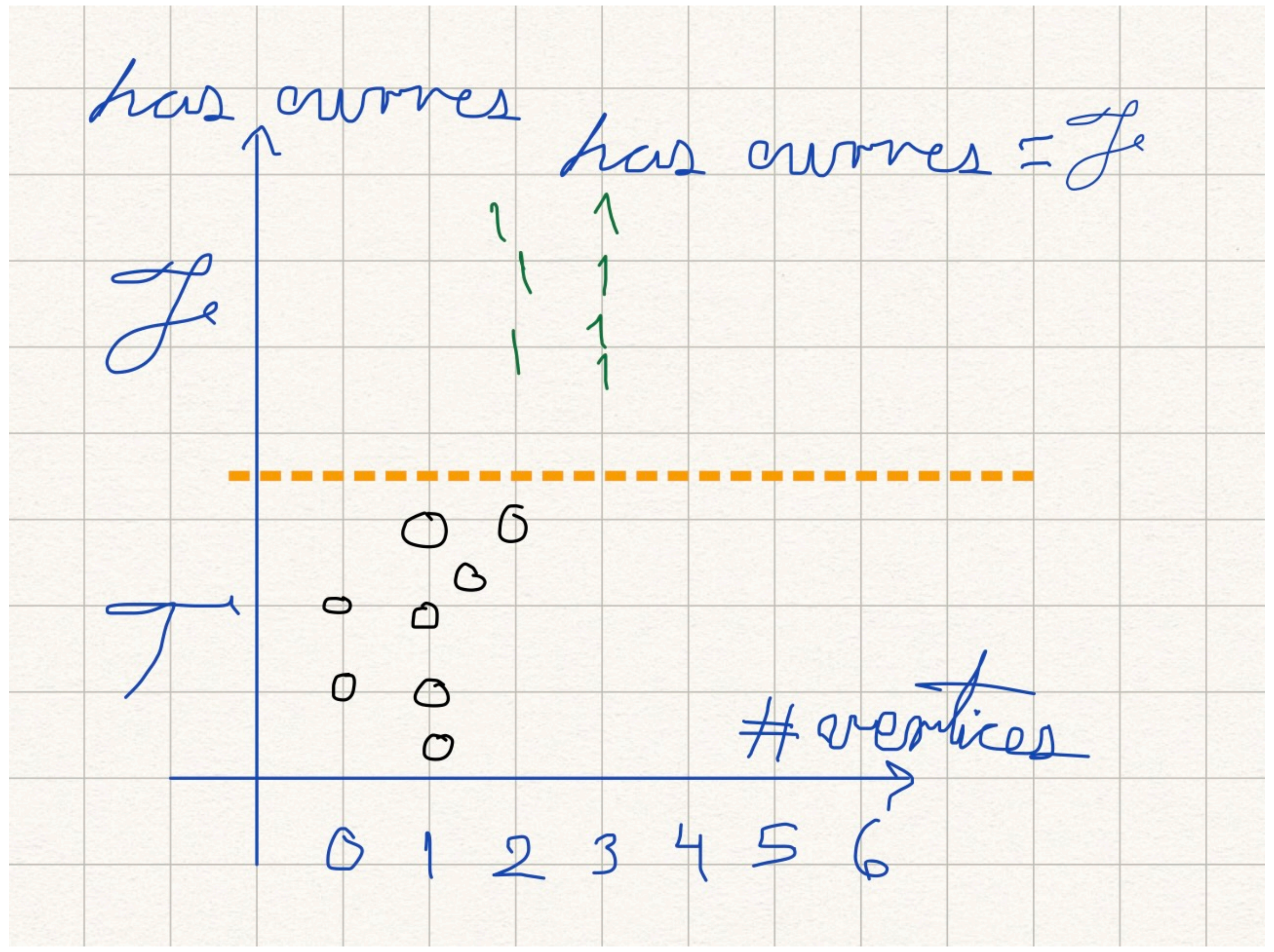




Decision Trees - Candidate Splits for a Branch

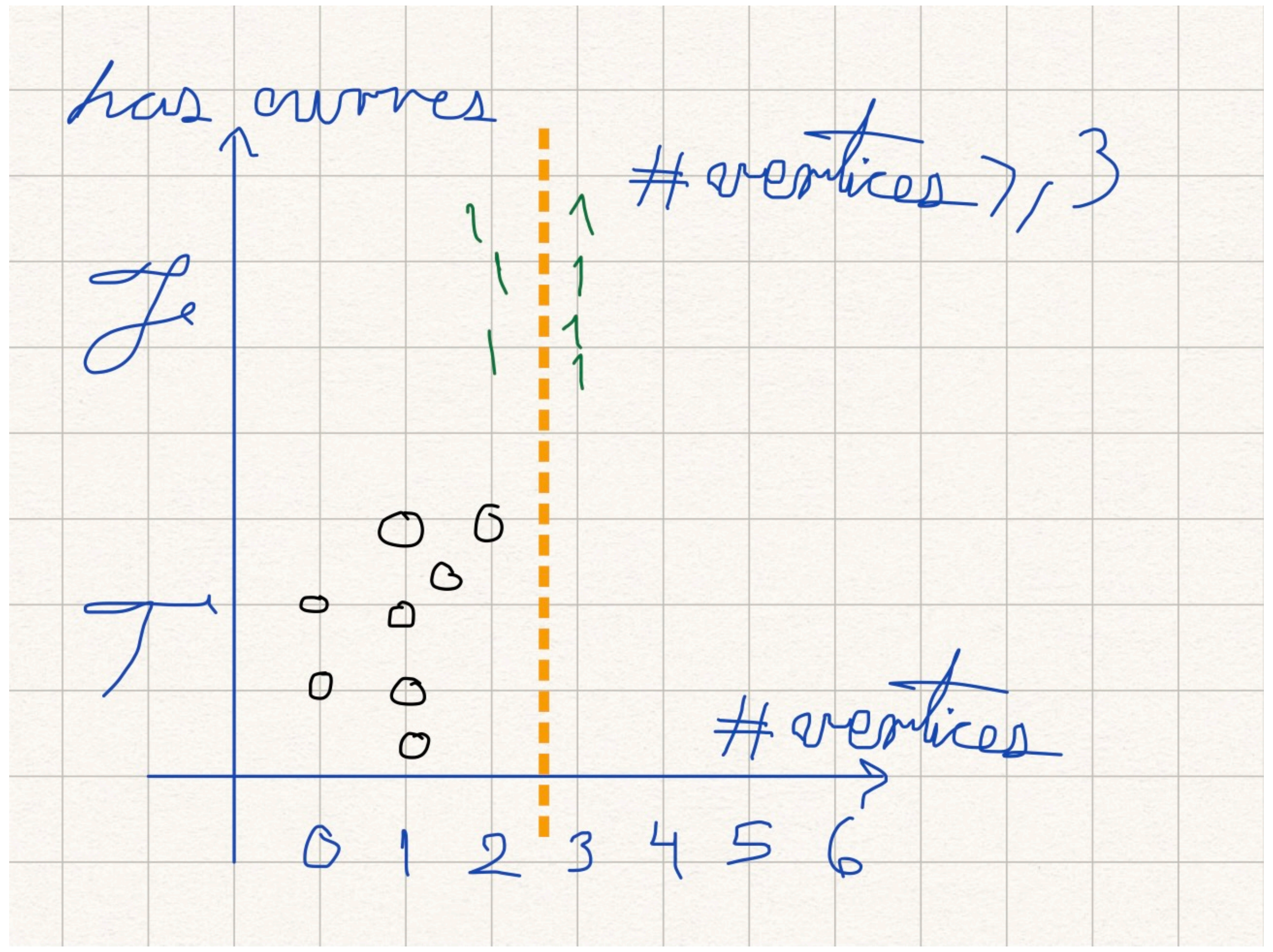
- Choose at random $m = \sqrt{|x|}$ features $x^{(i)}$
- Identify candidate splits of branch set on $x^{(i)}$
  - If feature $x^{(i)}$ can't be sorted
    - small number
      - Iterate per feature value, matching items assigned to one subset, non-matching items assigned to the other one
    - large number
      - iterate per feature value assigning with 50% probability to each subset

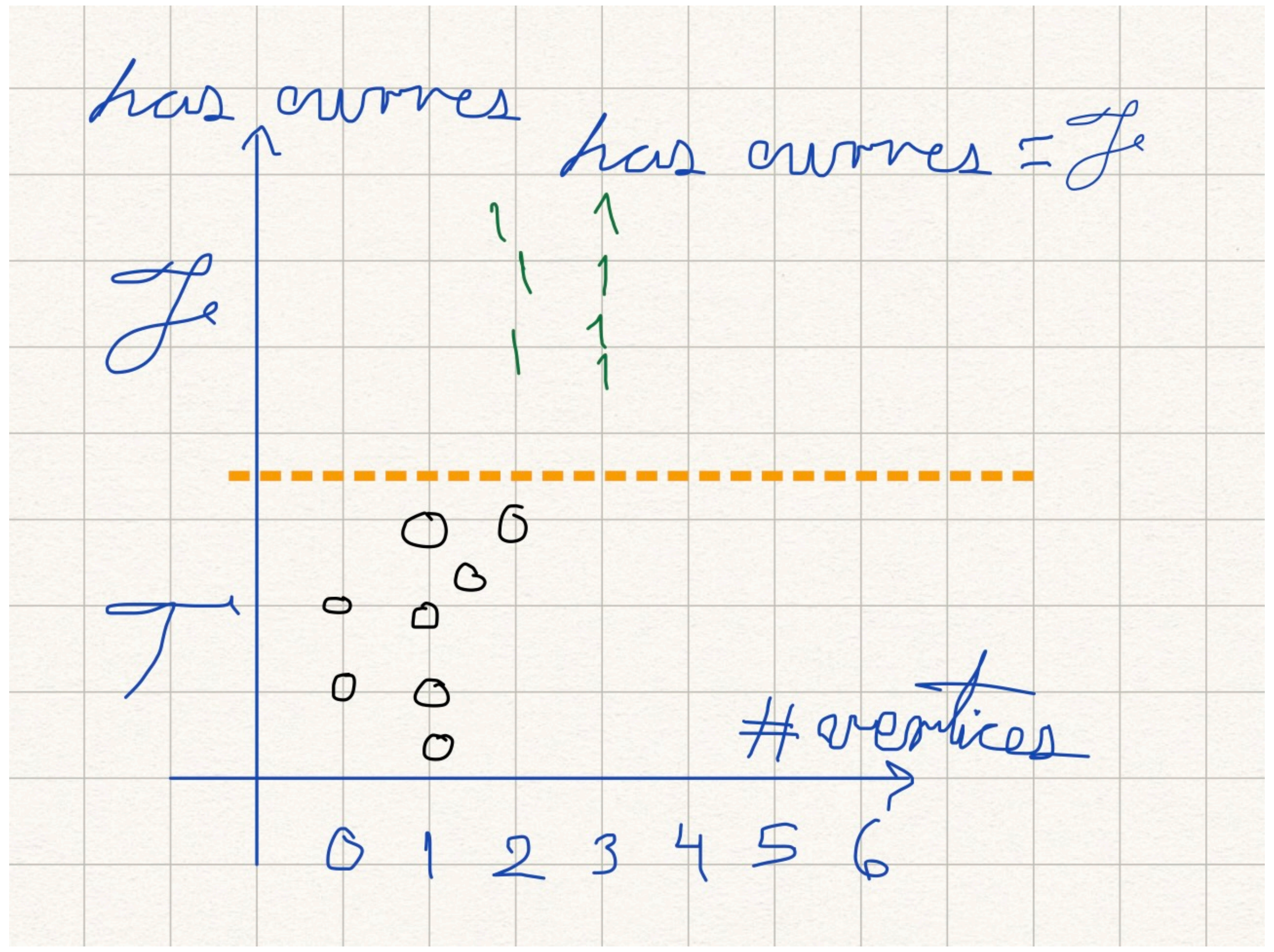




Decision Trees - Candidate Splits for a Branch

- Choose at random $m = \sqrt{|x|}$ features $x^{(i)}$
- Identify candidate splits of branch set on $x^{(i)}$
  - Feature $x^{(i)}$ can be sorted
    - class boundaries as thresholds
      - sort data items according to feature value
      - adjacent pairs $(item_0, item_1)$ in different class
      - threshold is midway between $item_0$ and $item_1$
    - randomly select k thresholds





Decision Tree Learning

- Versions for
    - Categorical features
    - Continuous features
- Robust to errors
- Robust to missing feature values
- Construction of decision trees favors small trees

Decision Tree Learning - Limitations

- Many different trees can lead to similar classifications
- The algorithm to build a decision tree grows each branch just deeply enough to perfectly classify the training examples
    - potential overfit
- Randomness in identification of splits: m features, k thresholds
    - better splits may have not been considered
- Addressed through Random Forests

Random Forests

- Build many decision trees
- Use randomness in identification of splits: m features, k thresholds
- Classification
  - Each tree votes for a class
    - one vote per tree on its classification
    - N_i votes per tree i on its classification. N_i is the number of items in the leave that determines the class in tree i

Building Random Forests - Simple Strategy

- Separate dataset into Training Set and Test Set
  - Train multiple decision trees on Training Set using random splits
  - Evaluate with Test Set

Building Random Forests - Bagging

- Bagging
  - For each tree in the forest
    - Build a **bag**
      - Random subsample of Training Set with replacement
      - Same size as Training Set
    - Train tree with its **bag**
    - Evaluate tree with its **out-of-bag** examples
  - Average **out-of-bag** errors for all trees

Random Forests

- Decision Trees
- Limitations of Decision Trees
- Random Forests

Applied Machine Learning

Classification - Random Forests