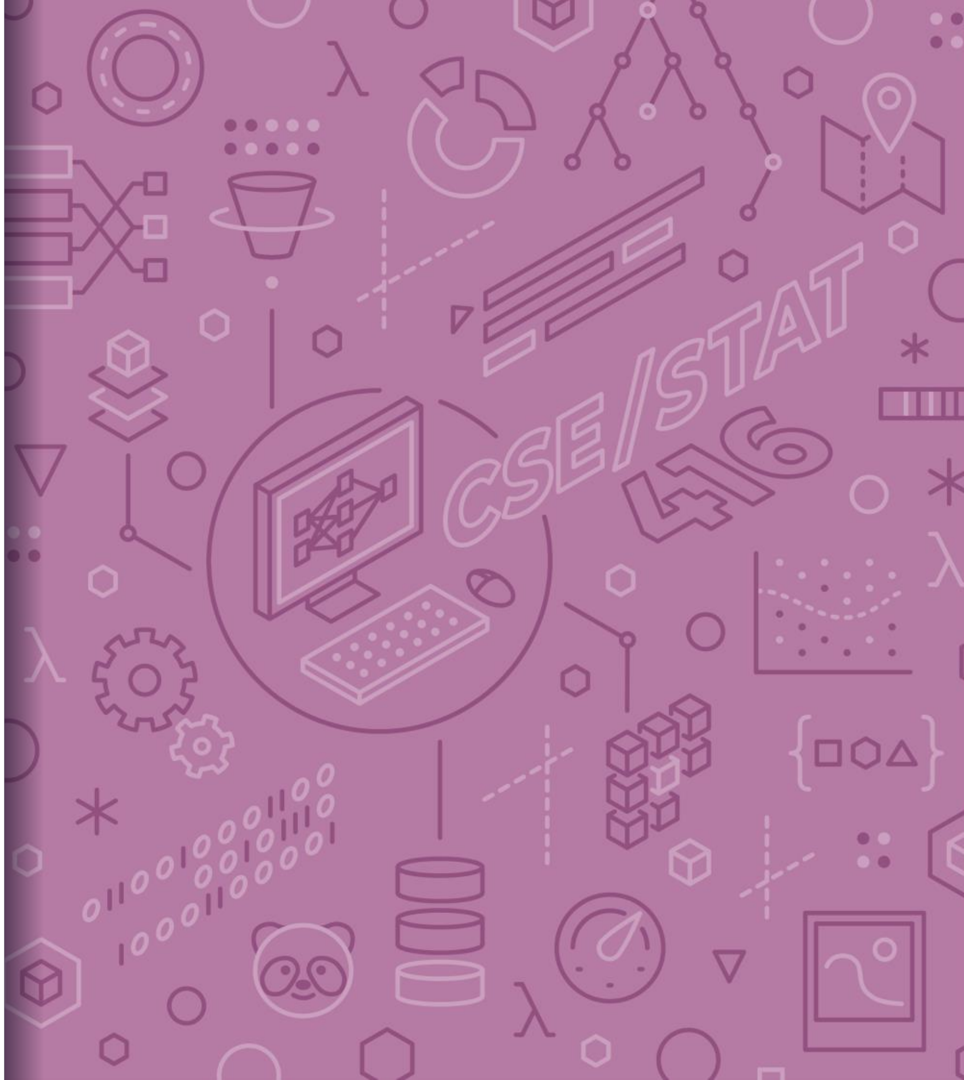
CSE/STAT 416

Non-Parametric Classification Methods

Amal Nanavati
University of Washington
July 18, 2022

Adapted from Hunter Schafer's slides



Administrivia

- Timeline:
  - **Last Week**: Intro to Classification, Logistic Regression
  - **This Week**: Non-Parametric Classification Methods, Ensemble Methods
  - **Next Week**: Neural Networks, Deep Learning
- Deadlines:
  - HW3 due tomorrow, 7/19 11:59PM
  - HW4 released Wed 7/20, due Tues 7/26 11:59PM
    - Programming component is on Kaggle, groups of ≤ 2
    - Partner assignments have been emailed to those seeking partners
    - Concept is still individual
  - Learning Reflection 5 due Fri, 7/22 11:59PM

Recap & Addressing LR Concerns

Poll Everywhere

Pair

~~Think~~

2 min

pollev.com/cs416

Suppose we trained a classifier and computed its confusion matrix on the training dataset. **Is there a class imbalance in the dataset and if so, which class has the highest representation?**

Think:
73% doggo
27% no imbalance

Pair:
55% doggo
45% no imbalance

Predicted Label

True Label	Pupper	Doggo	Woofer	
Pupper	2	27	4	} 33
Doggo	4	25	4	} 33
Woofer	1	30	2	} 33
Total # model predicted to be that class	7	82	10	

Total # in dataset with that label

Probability Classifier

$P(y=-1|x,w)$

Idea: Estimate probabilities $\hat{P}(y|x)$ and use those for prediction

$P(y=+1|x,w)$

review

Probability Classifier

Input x: Sentence from review

- Estimate class probability $\hat{P}(y=+1|x)$
- If $\hat{P}(y=+1|x) > 0.5$: ⟵ threshold
  - $\hat{y} = +1$
- Else:
  - $\hat{y} = -1$

$$P(y_i|x_i) = \begin{cases} \hat{P}(y_i=+1|x_i) & \text{if } y_i=+1 \\ \hat{P}(y_i=-1|x_i) & \text{else} \end{cases}$$

Notes:

- Estimating the probability improves **interpretability.**
  - Unclear how much better a score of 5 is from a score of 3. Clear how much better a probability of 0.75 is than a probability of 0.5

Connecting Score & Probability

$Score(x_i) = w^T h(x_i)$

$-\infty$ $\quad \hat{y}_i = -1 \quad$ 0 $\quad \hat{y}_i = +1 \quad$ ∞

Very sure
$\hat{y}_i = -1$

Not sure if
$\hat{y}_i = -1 \; or \; \hat{y}_i = +1$

Very sure
$\hat{y}_i = +1$

$\hat{P}(y_i = +1|x_i) = 0$ $\quad \hat{P}(y_i = +1|x_i) = 0.5$ $\quad \hat{P}(y_i = +1|x_i) = 1$

0 $\quad$ 1

$\hat{P}(y = +1\,|x)$

Logistic Regression Model

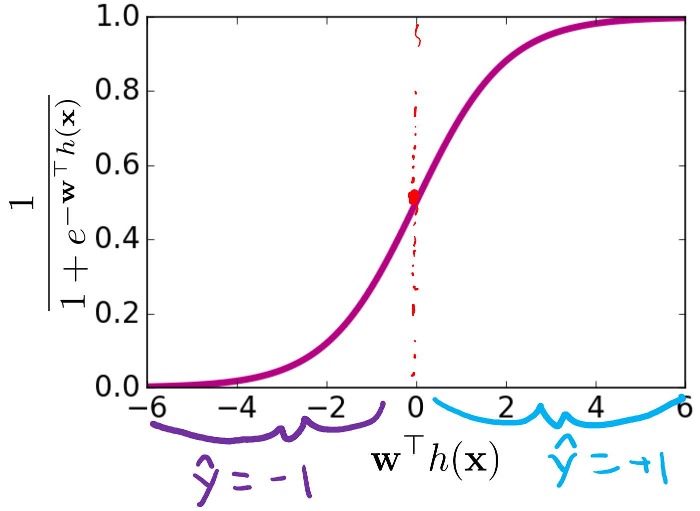
$Score(x_i) = \sum_{j=1}^{D} w_j h_j(x_i) = w^T h(x)$

$$P(y_i = +1 | x_i, w) = sigmoid(Score(x_i)) = \frac{1}{1 + e^{-w^T h(x_i)}}$$

Logistic Regression Classifier

Input x: Sentence from review

- Estimate class probability $\hat{P}(y = +1 | x, \hat{w}) = sigmoid(\hat{w}^T h(x_i))$
- If $\hat{P}(y = +1 | x, \hat{w}) > 0.5$:
  - $\hat{y} = +1$
- Else:
  - $\hat{y} = -1$



$\log(ab) = \log(a) + \log(b)$

Maximum Likelihood Estimate (MLE)

Find the w that maximizes the likelihood

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, \ell(w) = \underset{w}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i | x_i, w)$$

$\begin{cases} P(y_i = +1 | x_i, w) & \text{if } y_i = +1 \\ P(y_i = -1 | x_i, w) & \text{else} \end{cases}$

Generally, we maximize the log-likelihood which looks like

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, \ell(w) = \underset{w}{\operatorname{argmax}} \log(\ell(w)) = \underset{w}{\operatorname{argmax}} \sum_{i=1}^{n} \log(P(y_i | x_i, w))$$

Also commonly written by separating out positive/negative terms

$$\hat{w} = \underset{w}{\operatorname{argmax}} \sum_{i=1: y_i = +1}^{n} \underbrace{\ln\left(\frac{1}{1 + e^{-w^T h(x)}}\right)}_{\log P(y_i = +1 | x, w) \text{ for pos terms}} + \sum_{i=1: y_i = -1}^{n} \underbrace{\ln\left(1 - \frac{1}{1 + e^{-w^T h(x)}}\right)}_{\log P(y_i = -1 | x, w) \text{ for neg terms}}$$

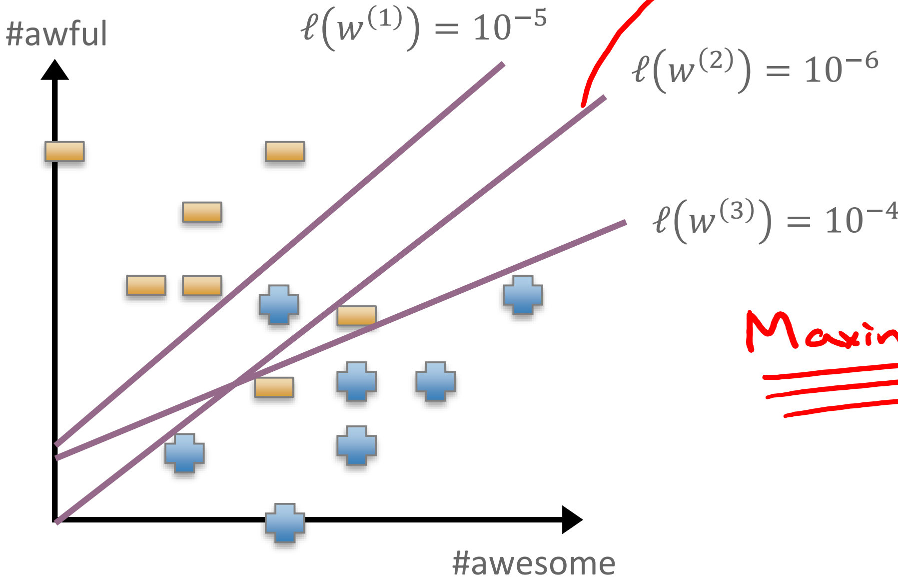
Poll Everywhere

Think

1.5 min

pollev.com/cs416

Which setting of w should we use?



each line represents a different learnt logistic regression model

Maximize likelihood

Gradient Ascent

Maximize ⇒ ascent
Minimize ⇒ descent

Gradient ascent is the same as gradient descent, but we go ”up the hill”.

```
start at some (random) point $w^{(0)}$ when $t = 0$
while we haven't converged
```

$$w^{(t+1)} \leftarrow w^{(t)} + \eta \nabla \underbrace{\ell(w^{(t)})}_{\text{Gradient of likelihood}}$$

$$t \leftarrow t + 1$$

learning rate

This is just describing going up the hill step by step.

η controls how big of steps we take, and picking it is crucial for how well the model you learn does!

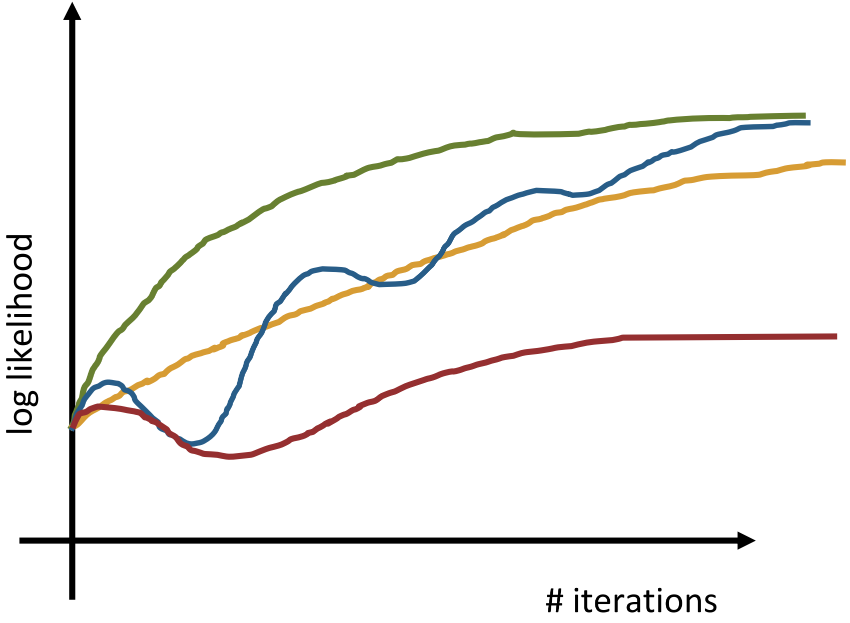
Poll Everywhere

Think

2 min

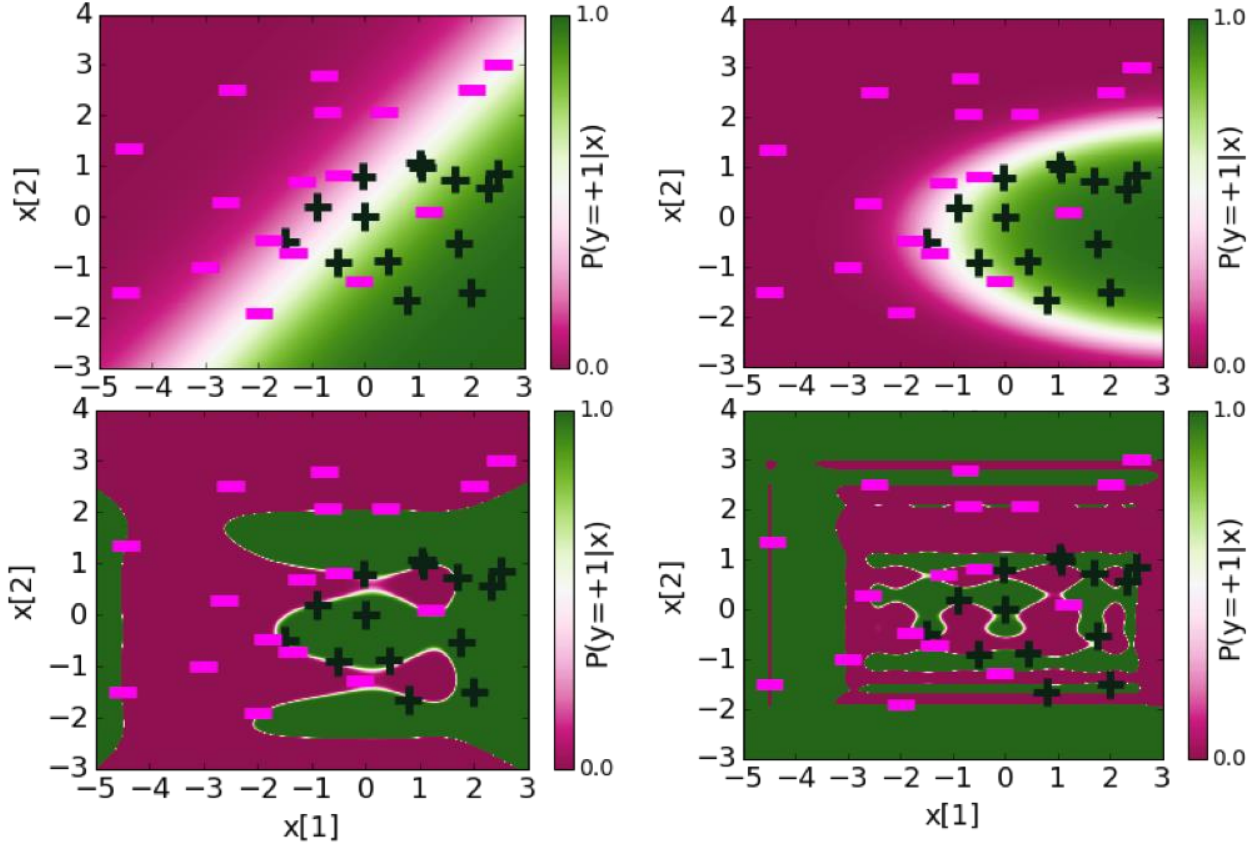
- Match the below lines to the following labels:
  - "Very High Learning Rate"
  - "High Learning Rate"
  - "Good Learning Rate"
  - "Low Learning Rate"

Find right learning rate with hyperparamete tuning



Plotting Probabilities

$$P(y=+1|x) = \frac{1}{1+e^{-\hat{w}^T h(x)}}$$



Some Details

Why do we subtract the L2 Norm?

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, \ell(w) - \lambda ||w||_2^2$$

subtract

How does λ impact the complexity of the model?

Same as Ridge

How do we pick λ?

Validation , CV

Parametric vs. Non-Parametric Methods

Parametric vs. Non-Parametric Methods

If you can write the learnt model using "simple" math, and you're learning weights/parameters in that equation, then the model is parametric.

Parametric Methods: make assumptions about the data distribution

- Linear Regression ⇒ assume the data is linear
- Logistic Regression ⇒ assume probability has the shape of of a logistic curve
- Those assumptions result in a ***parameterized*** function family. Our learning task is to learn the parameters.

Non-Parametric Methods: (mostly) don't make assumptions about the data distribution

- K-NN, Decision Trees
- We're still learning something, but not the parameters to a function family that we're assuming describes the data.
- Useful when you don't want to (or can't) make assumptions about the data distribution.

K-Nearest Neighbors

Poll Everywhere

Think

1 min

- Consider the below dataset. Consider a new patient, with a temperature of 99ºF and a runny nose. What illness would you predict for them?

Training Data

Temp	Runny Nose?	Illness
98.0	Y	Cold
98.6	N	Cold
98.9	Y	Flu
99	Y	Cold
99.5	Y	Flu
101.2	N	Flu

New Patient:

Temp	Runny Nose?
99.0	Y

Poll Everywhere

Group

1.~~5~~ min

- Consider the below dataset. Consider a new patient, with a temperature of 99ºF and a runny nose. What illness would you predict for them?

Cold?
↳ 1-NN

Temp	Runny Nose?	Illness
98.0	Y	Cold
98.6	N	Cold
98.9	Y	Flu
99	Y	Cold
99.5	Y	Flu
101.2	N	Flu

New Patient:

Temp	Runny Nose?
99.0	Y

Flu?
↳ 3-NN

Nearest Neighbors Overview

- **Big Idea**: Label a input point with with the class of its nearest point(s)!

- During **training**, just store the entire dataset $\{(x_i, y_i\ )\}_{i=1}^{n}$.
- When **predicting**, for every query example x_q
  - Find the nearest point(s) to x_q
  - Output the (majority) label of those point(s)

k-Nearest Neighbors Algorithm

Given a query point x_q:

- Compute its distance to every point in the training set
- Get the k points that are closest to x_q
- Count the instances of each label amongst the k closest points
- Output the majority label amongst the k closest points

Poll Everywhere

Think

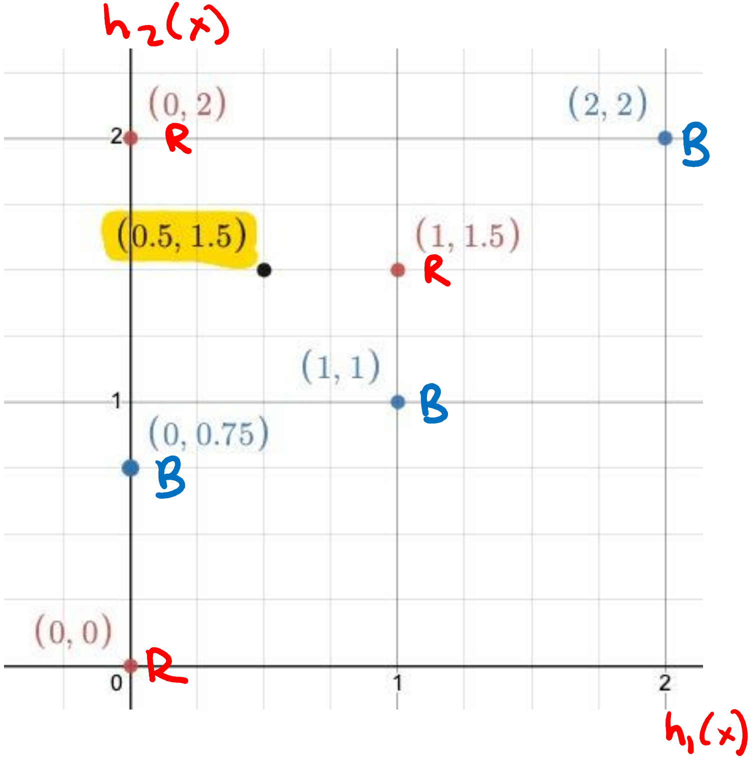
1.5 min

pollev.com/cs416

3-NN

- You are classifying the highlighted point using L2 (Euclidean) distance, and $k = 3$. What class do you classify it as?

straight-line distance



Poll Everywhere

Group

3 min

pollev.com/cs416

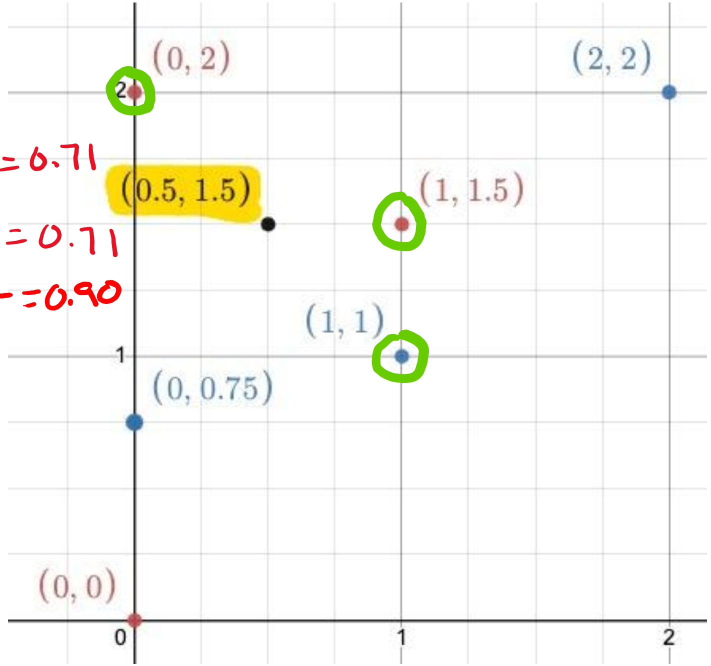
$$\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- You are classifying the highlighted point using L2 (Euclidean) distance, and $k = 3$. What class do you classify it as?

(x,y)	dist
(1, 1.5)	0.5
(0, 2)	$\sqrt{.5^2 + 0.5^2} = 0.71$
(1, 1)	$\sqrt{.5^2 + .5^2} = 0.71$
(0, 0.75)	$\sqrt{0.5^2 + 0.75^2} = 0.90$

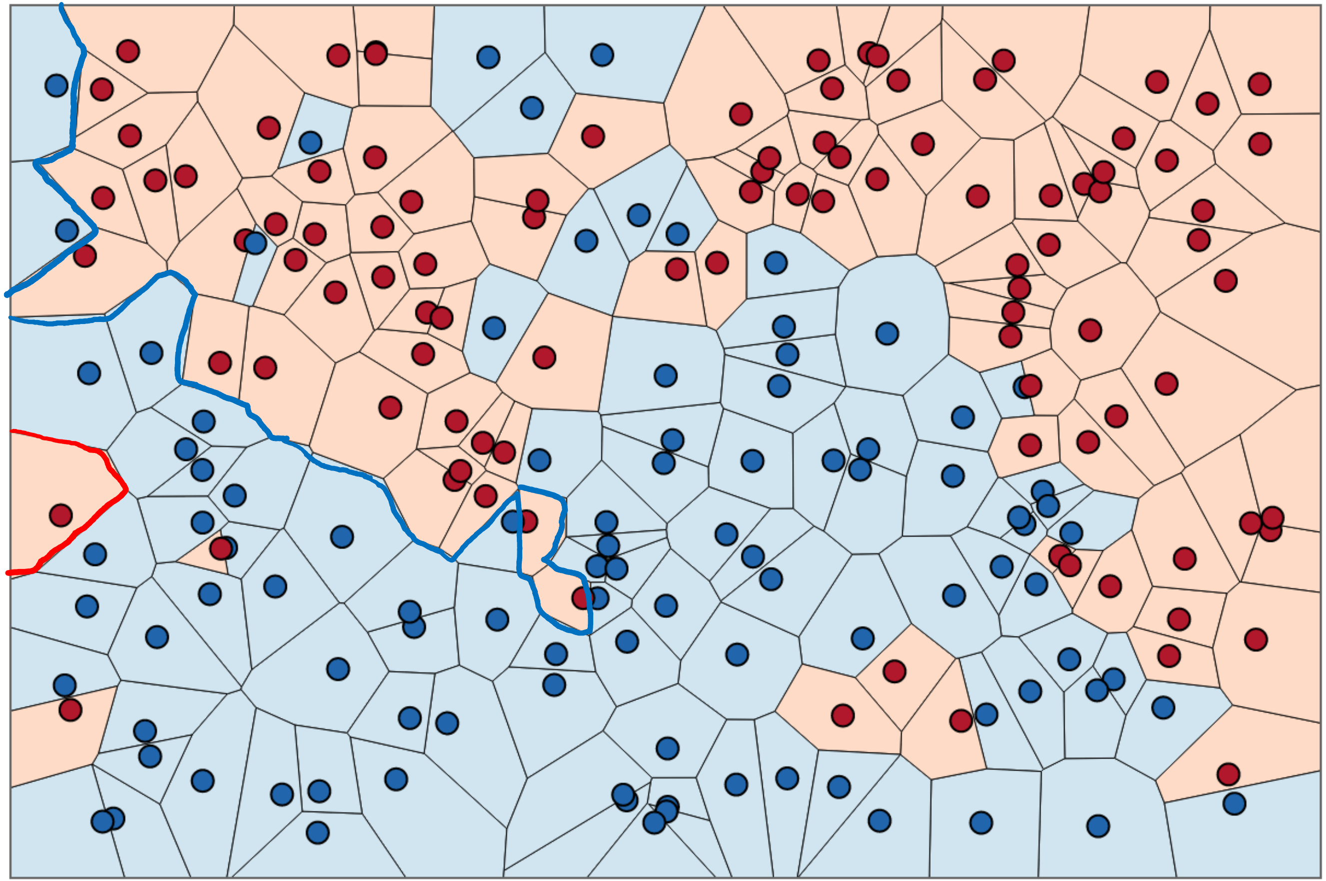
Red
Red
Blue

Majority: Red



Predict: Red

1-NN Decision Boundaries: Voronoi Diagram



Source: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

K-Nearest Neighbors: Hyper-paramters

- How do we define distance?
- What value of k should we use?

Types of Features

- **Numeric:** data described by a number (quantitative)
  - **Discrete:** cannot be subdivided
    - e.g., number of bedrooms
  - **Continuous:** can be subdivided
    - e.g., area of the house
  - **Tricky Case**: house price? (don't divide further than penny)
    - **Rule of Thumb**: if the discreteness is caused by units of measurement, as opposed to the quantity being measured, treat it as continuous!

- **Categorical**: data described by a category (qualitative)
  - **Ordinal**: has an order

    e.g., school quality (good / okay / poor)

    e.g., survey response (agree / neutral / disagree)
  - **Nominal**: doesn't have an order
    - e.g., nearest school type (public / private / charter)

Making Categorical Features Quantitative

Encodings

- All ML models we've learnt so far require input features to be numbers!
- **Ordinal**: Assign each value to a number:
  - e.g., good = 1, okay = 0, poor = -1
- **Nominal**: One-hot encoding, make each value its own binary feature!
  - In section, you saw a one-hot encoding of "County"

School	House Price
Public	$500K
Private	$750K
Charter	$600K
Public	$700K

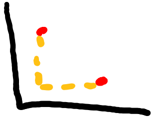
School - Public	School - Private	School - Charter	House Price
1	0	0	$500K
0	1	0	$750K
0	0	1	$600K
1	0	0	$700K

Distance

1. Featurize / Vectorize / Embed Encode the data
    - i.e., convert each row into a numeric vector
2. Use a common distance metric

- Euclidean Distance (e.g., L2 norm)
    - Gets the straight-line distance between two vectors

$$\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



- Manhattan Distance (e.g., L1 norm)
    - Gets the axis-aligned distance between two vectors

$$\text{dist}((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2|$$

- Cosine Similarity
    - Gets the cosine of the angle between two vectors
    - Common for featurized text data (e.g., Bag of Words)
    - distance = 1 – similarity

Choosing k

- What is the lowest value of k we can have?

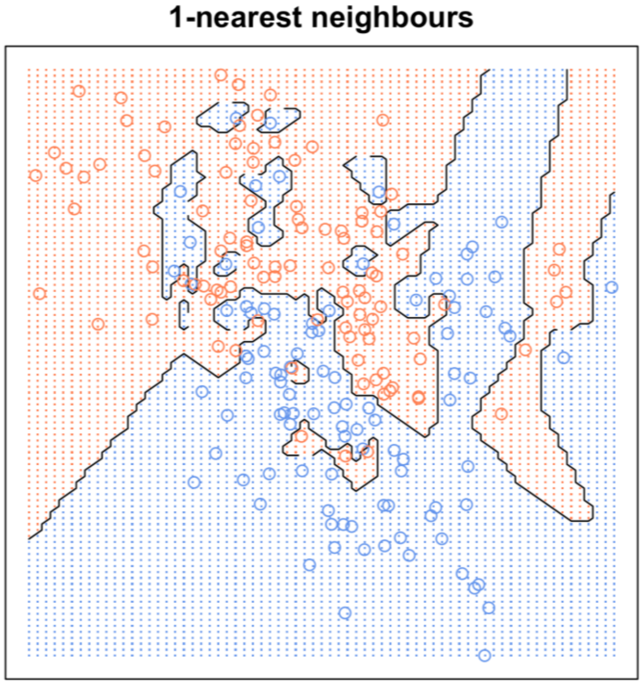
  k=1 - sensitive to noise
  - more likely to overfi

- What is the highest value of k we can have?

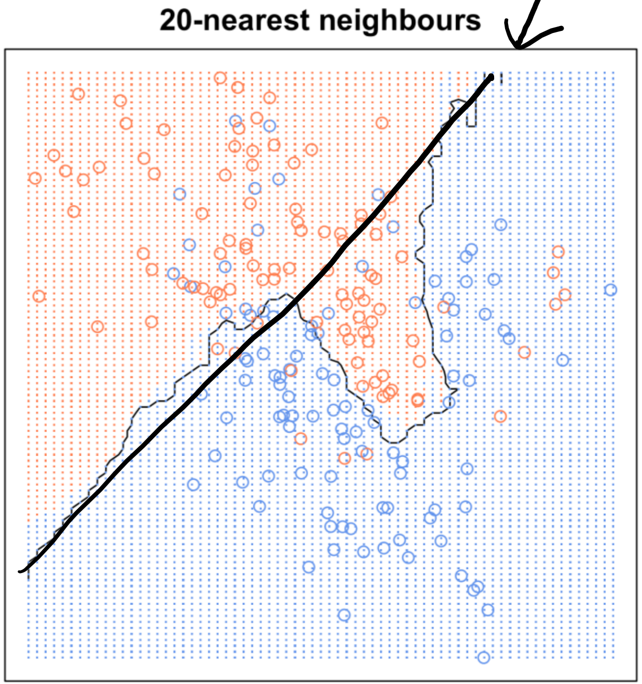
  k=n - Same as Majority Class Classifier

- How do we choose k? Validation set or cross-validation!

1-NN vs. k-NN



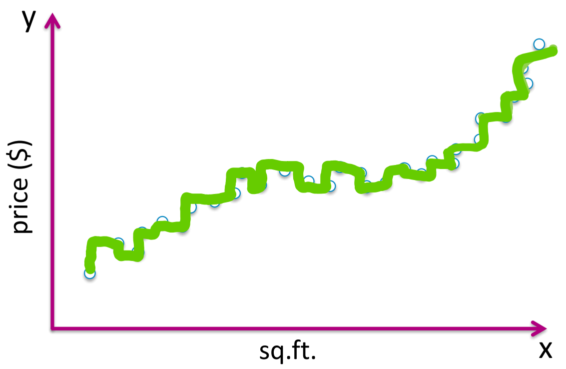
Logistic Regression decision boundary (without feature engineering)



Source: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

K-Nearest Neighbors Pros/Cons

- Pros:
  - Training is fast (don't need to compute anything, just store the dataset)
  - Doesn't make assumptions about the data distribution.
  - Can learn a non-linear decision boundary
  - Can readily do multi-class classification (unlike Logistic Regression)
- Cons:
  - Prediction is slow (must search through the whole dataset)
  - Large memory usage.
- NOTE: k-NN can be used for regression as well!



Brain Break

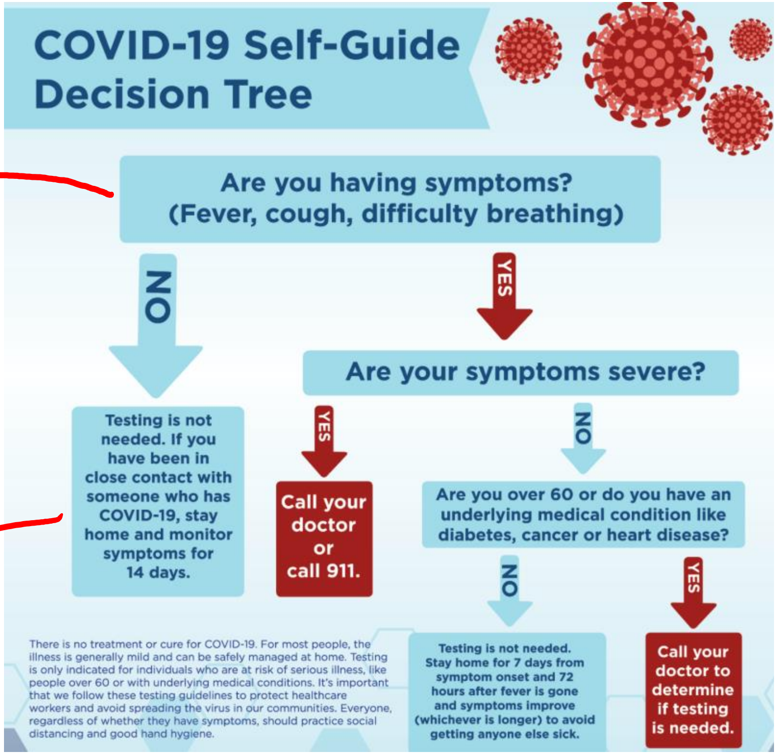
3:25

Decision Trees

How Do We Make Decisions?

Branch Nodes

Leaf nodes



https://www.holzer.org/coronavirus-covid-19-updates/

What makes a loan risky?

I want to buy a new house!

Loan Application

Credit History ★★★★

Income ★★★

Term ★★★★★

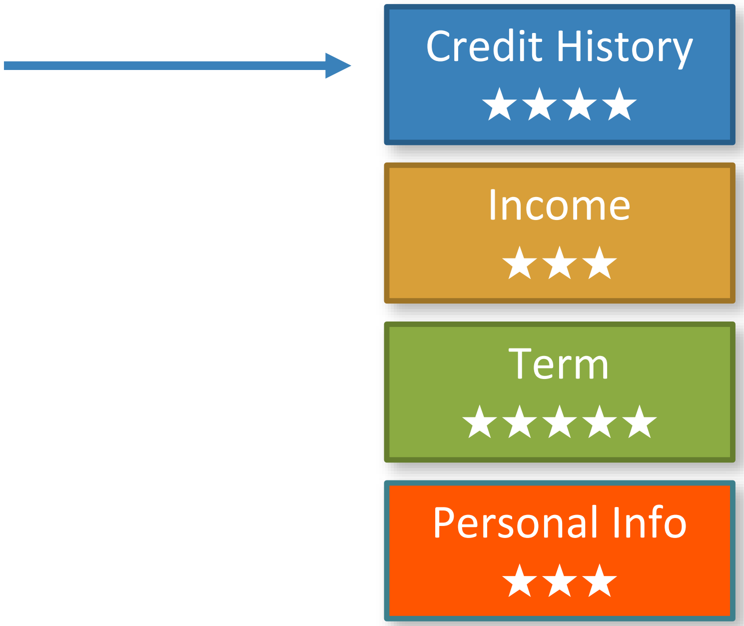
Personal Info ★★★

Credit history explained

Did I pay previous loans on time?

Example: excellent, good, or fair

Ordinal feature



Income

What’s my income?

Example:
$80K per year

Numeric feature

Credit History
★★★★

Income
★★★

Term
★★★★★

Personal Info
★★★

Loan terms

How soon do I need to pay the loan?

Example: 3 years, 5 years,

~~Numeric~~ Feature
Nominal

Credit History
★★★★

Income
★★★

Term
★★★★★

Personal Info
★★★

Personal information

Age, reason for the loan, marital status,…

Example: Home loan for a married couple

Credit History
★★★★

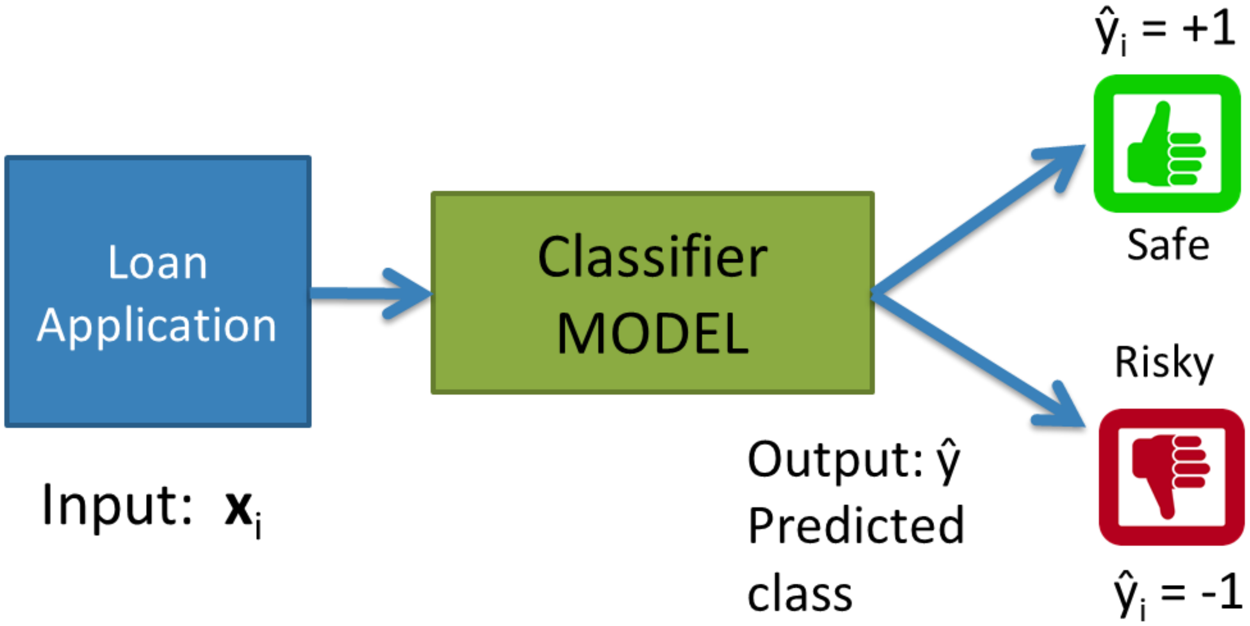
Income
★★★

Term
★★★★★

Personal Info
★★★

"Intelligent" application

Classifier review



N = 9 datapoints , D = 3 feature

Setup

Data (N observations, 3 features)

D

Label

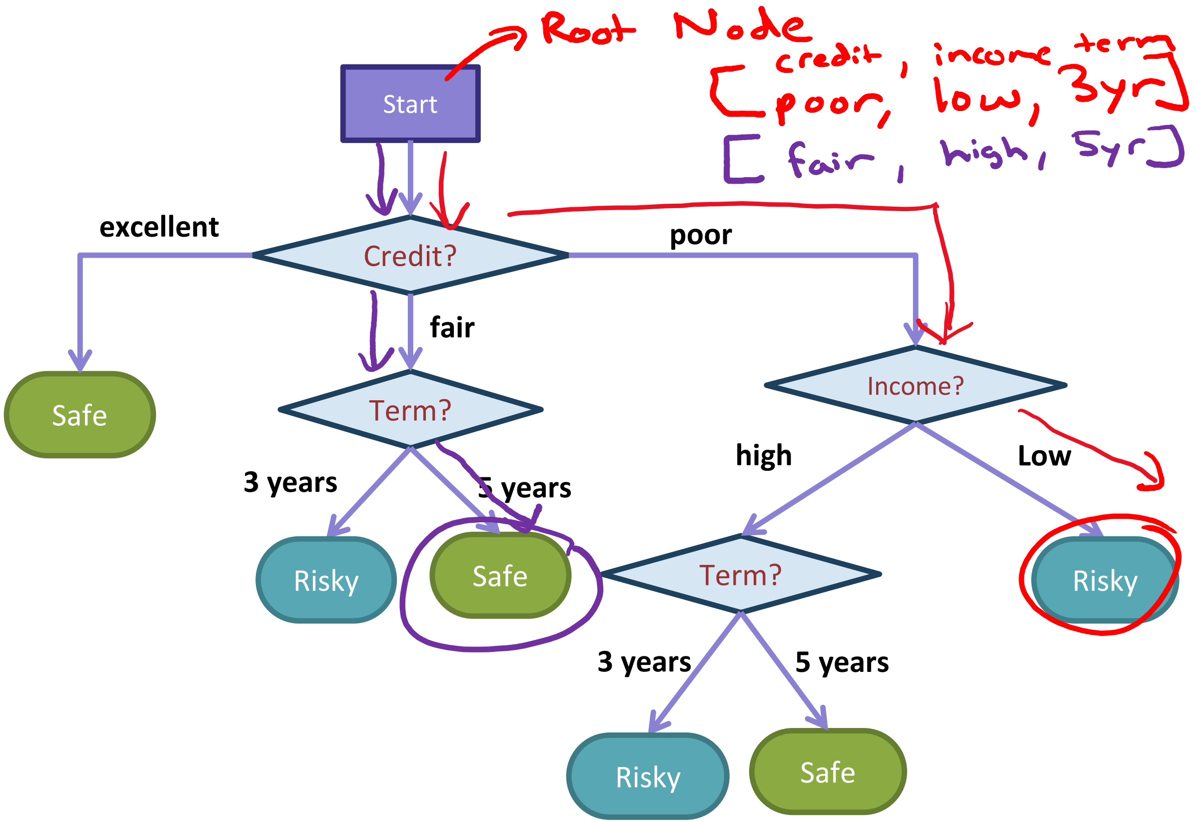
N

Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

Evaluation: classification error

Many possible decisions: number of trees grows exponentially!

Decision Trees



- **Branch/Internal node:** splits into possible values of a feature
- **Leaf node:** final decision (the class value)

Poll Everywhere

Think

1 min

- For each feature below, identify whether it is:
  - **numeric discrete**
  - **numeric continuous**
  - **categorical ordinal**
  - **categorical nominal**

Income	Highest Degree	Credit Score	Zip-Code	Num Dependents	Loan Risk
$110K	High School	Fair	98105	1	Safe
$50K	BS	Poor	97122	5	Risky
$75K	JD	Excellent	35012	2	Safe

Poll Everywhere

Group

2 min

- For each feature below, identify whether it is:
  - **numeric discrete**
  - **numeric continuous**
  - **categorical ordinal** → assign # based on order
  - **categorical nominal** → one-hot encoding

Continuous | Nominal | Ordinal | Nominal | Numeric Discrete | Nominal

Income	Highest Degree	Credit Score	Zip-Code	Num Dependents	Loan Risk
$110K	High School	Fair	98105	1	Safe
$50K	BS	Poor	97122	5	Risky
$75K	JD	Excellent	35012	2	Safe

Growing Trees

Visual Notation

Loan status: **Safe** **Risky**

Root

6 3

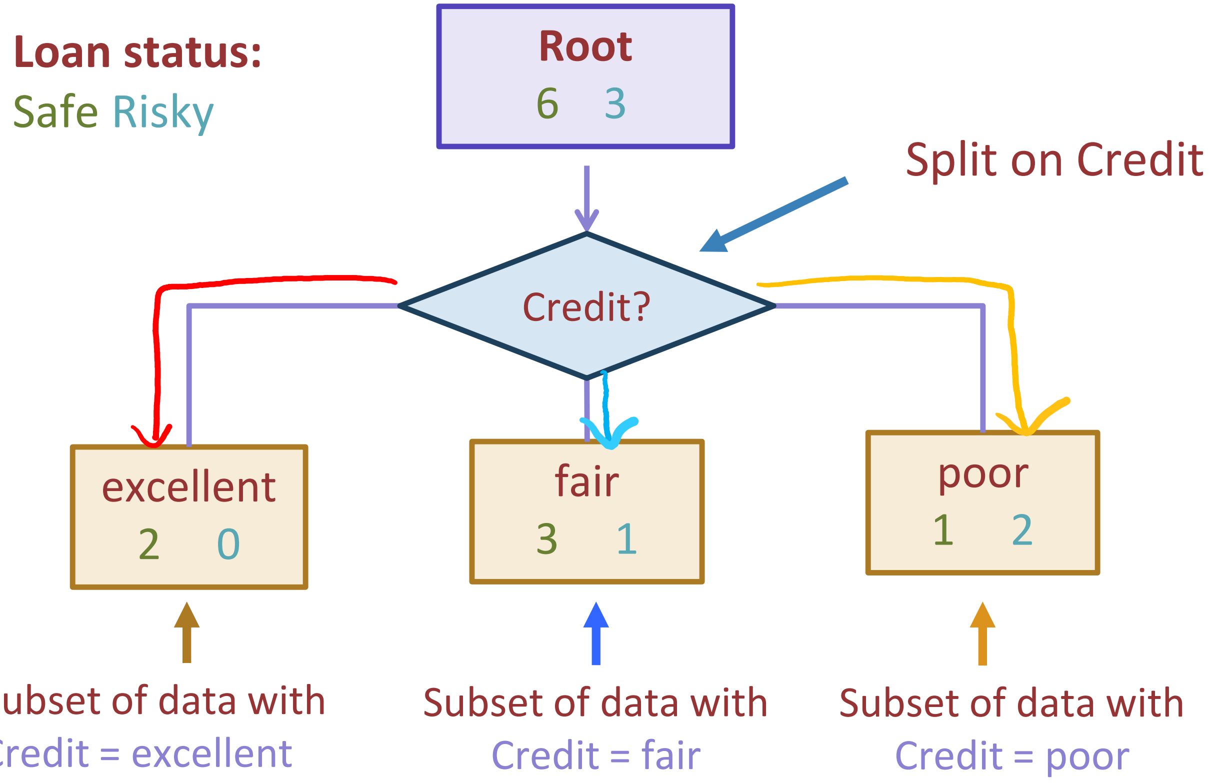
# of Risky loans

# of Safe loans

N = 9 examples

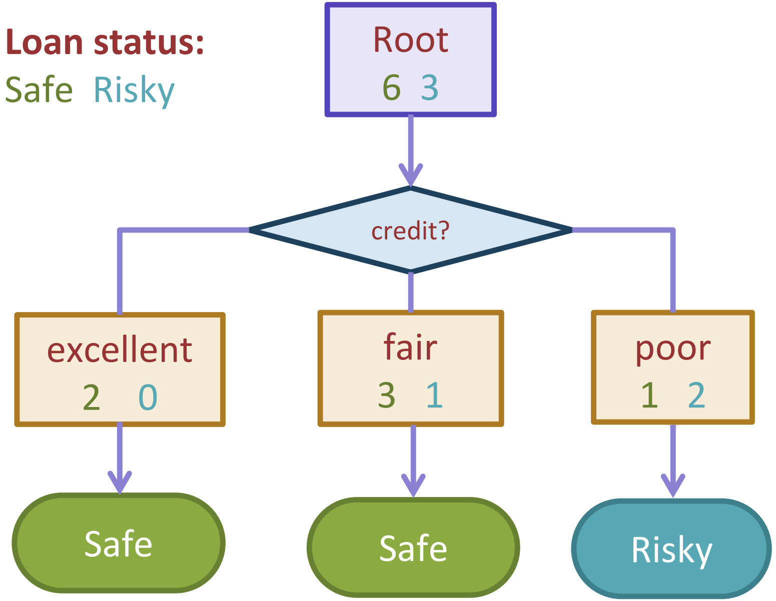
Decision stump: 1 level

Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe



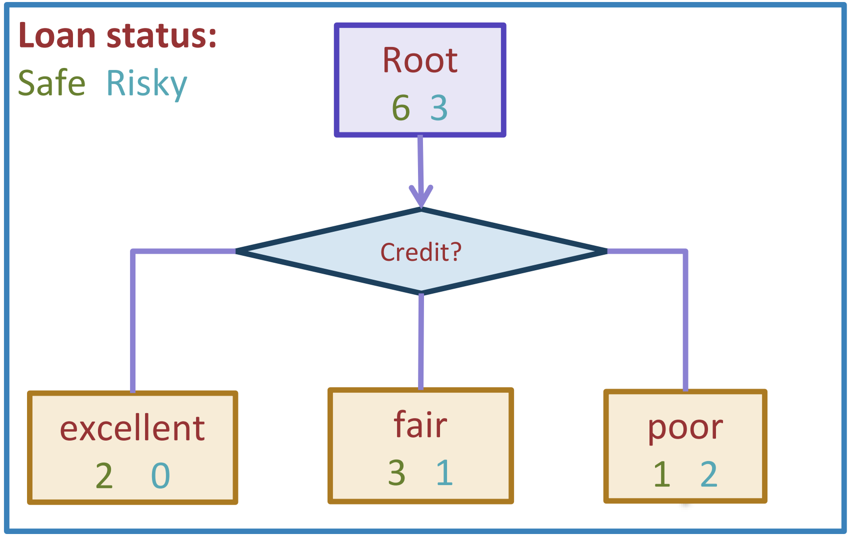
Making predictions

For each leaf node, set ŷ = majority value

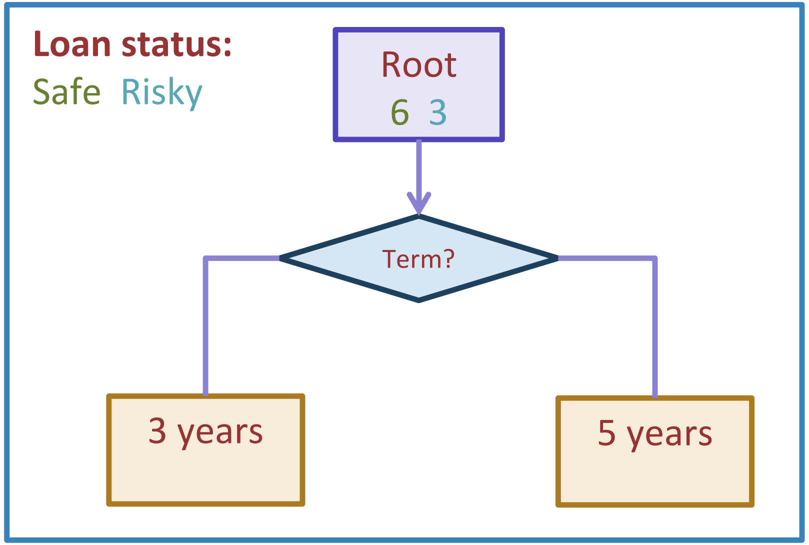


How do we select the best feature?

Choice 1: Split on Credit



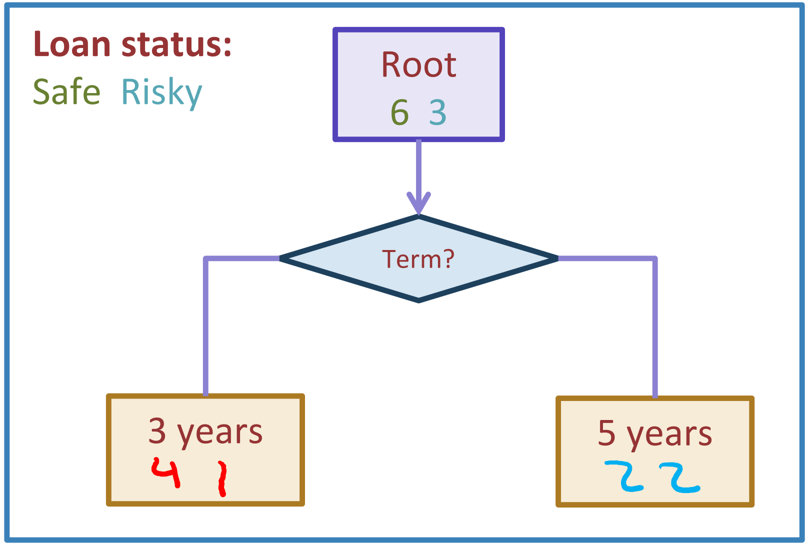
Choice 2: Split on Term



Calculate the node values.

Credit	Term	Income	y
excellent	3 yrs	high	safe
fair	5 yrs	low	risky
fair	3 yrs	high	safe
poor	5 yrs	high	risky
excellent	3 yrs	low	safe
fair	5 yrs	low	safe
poor	3 yrs	high	risky
poor	5 yrs	low	safe
fair	3 yrs	high	safe

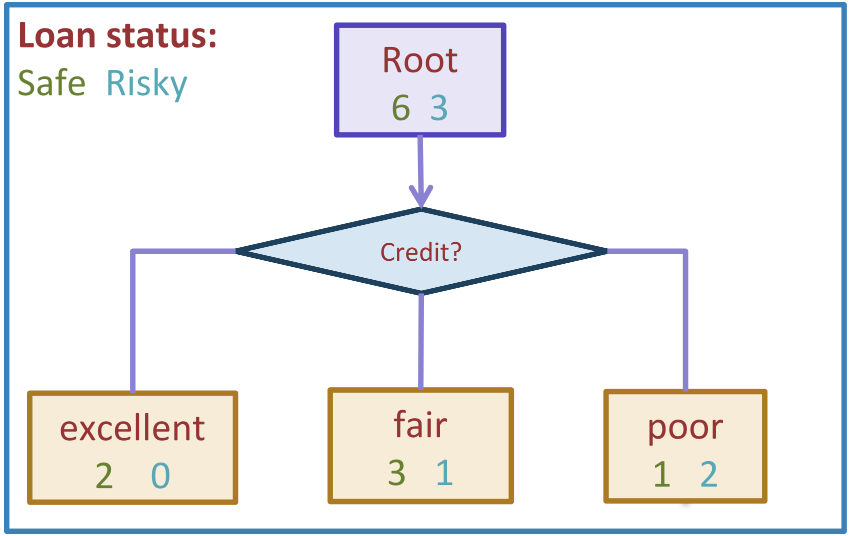
Choice 2: Split on Term



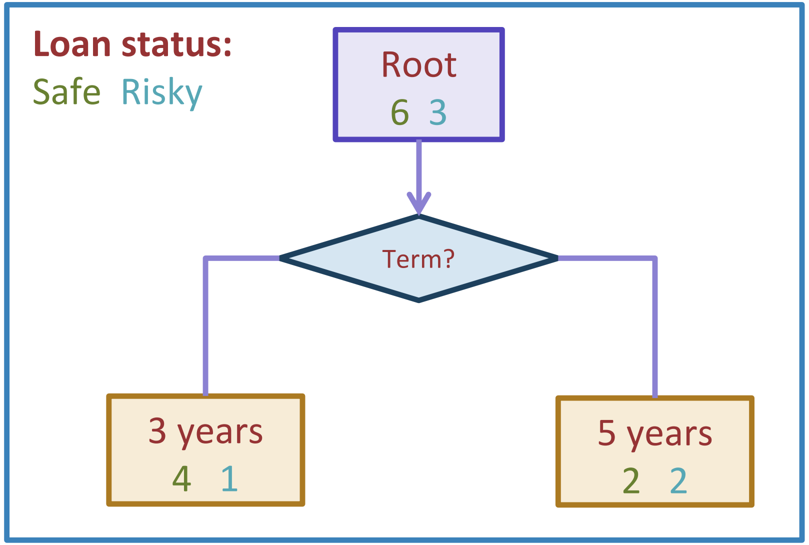
How do we select the best feature?

Select the split with lowest classification error

Choice 1: Split on Credit

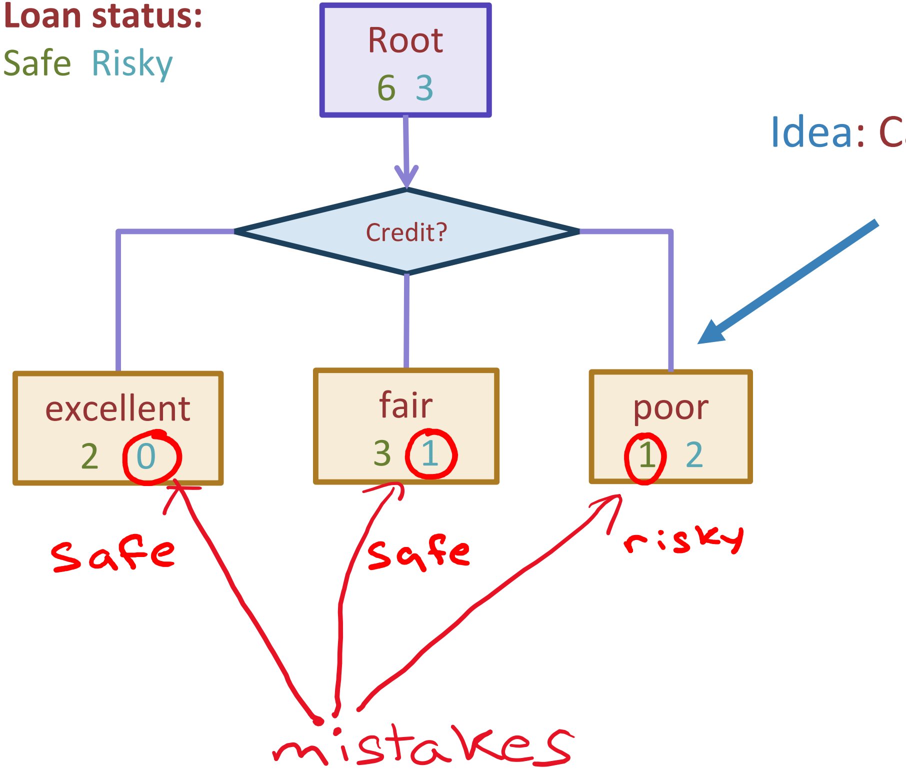


Choice 2: Split on Term



How do we measure effectiveness of a split?

Loan status:
Safe Risky



Idea: Calculate classification error of this decision stump

Error at a node

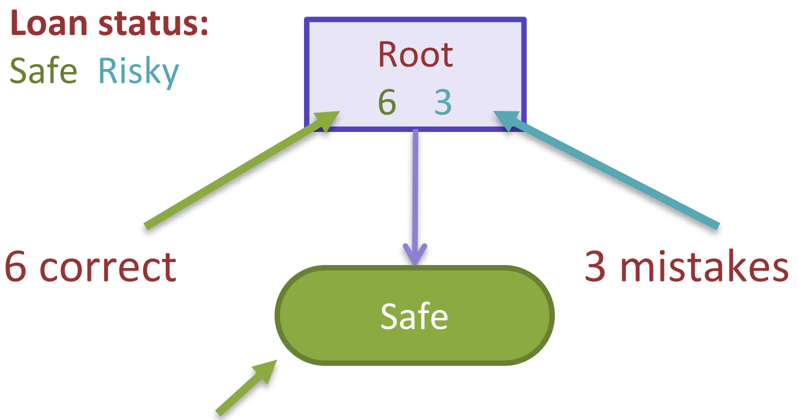
$$= \frac{\text{\# mistakes in each child node}}{\text{\# data points at the node}}$$

$$\frac{\text{\# mistakes}}{\text{\# examples}} = \frac{2}{9} \approx 0.22$$

Calculating classification error

Step 1: $\hat{y}$ = class of majority of data in node

Step 2: Calculate classification error of predicting $\hat{y}$ for this data



$\hat{y}$ = **majority class**

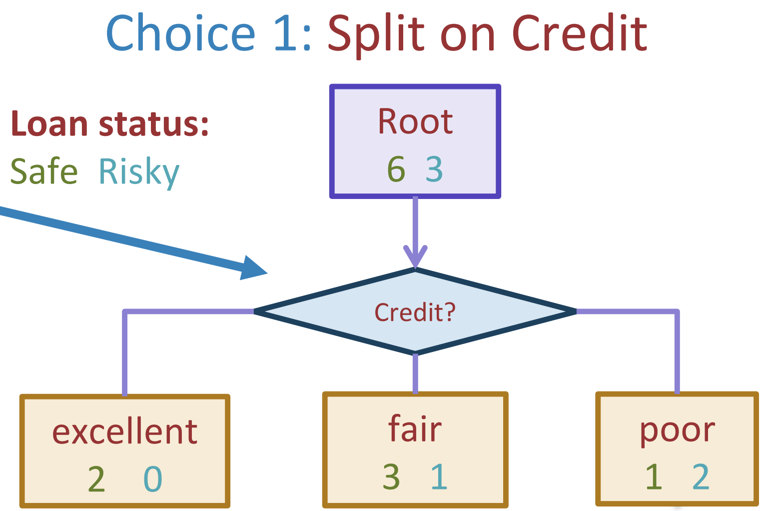
$$\text{Error} = \frac{3}{9}$$

$$= 0.33$$

Tree	Classification error
(root)	0.33

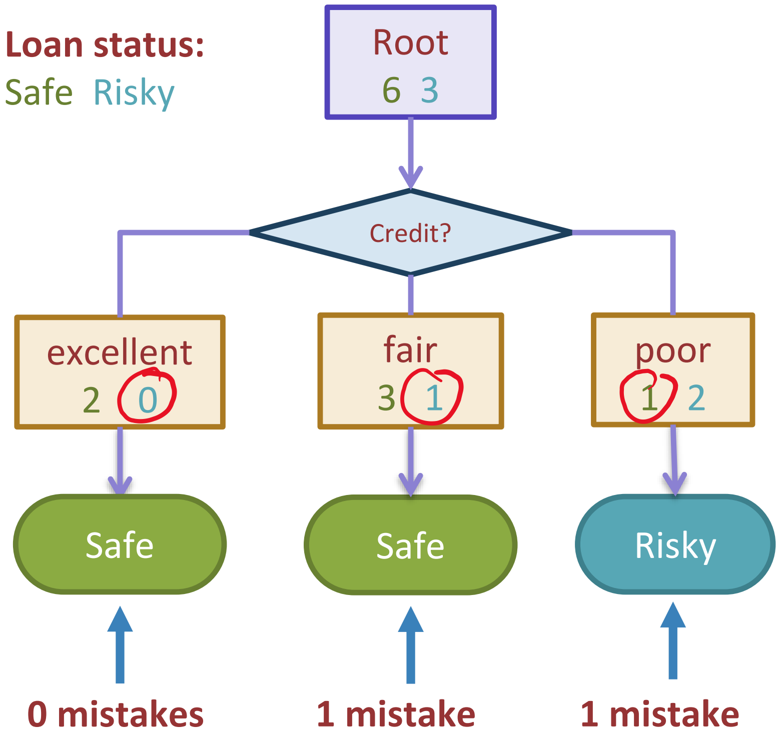
Choice 1: Split on Credit history?

Does a split on Credit reduce classification error below 0.33?



Split on Credit: Classification error

Choice 1: Split on Credit



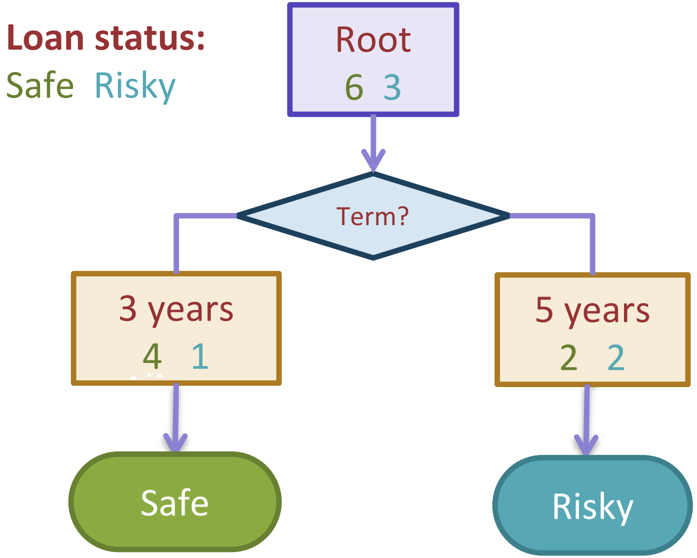
$$\text{Error} = \frac{2}{9}$$

$$= 0.22$$

Tree	Classification error
(root)	0.33
Split on credit	0.22

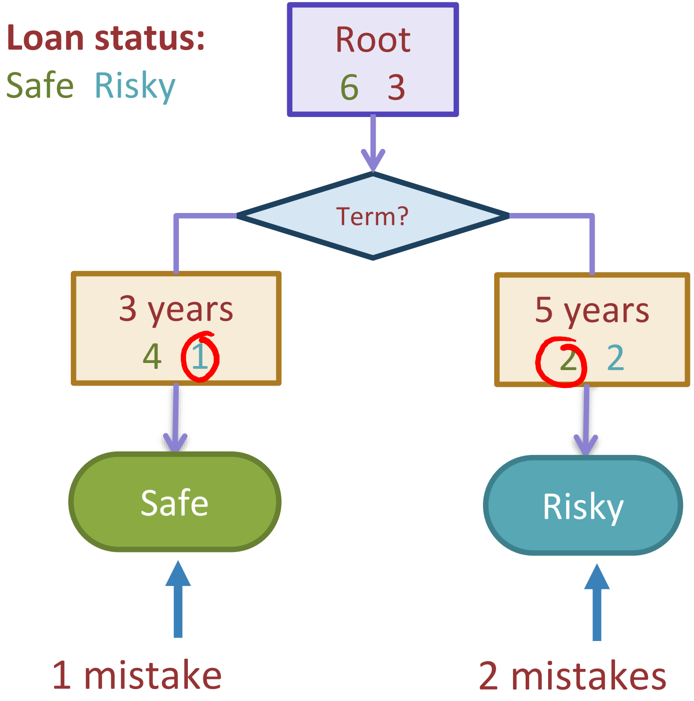
Choice 2: Split on Term?

Choice 2: Split on Term



Evaluating the split on Term

Choice 2: Split on Term



Error = $\frac{3}{9}$

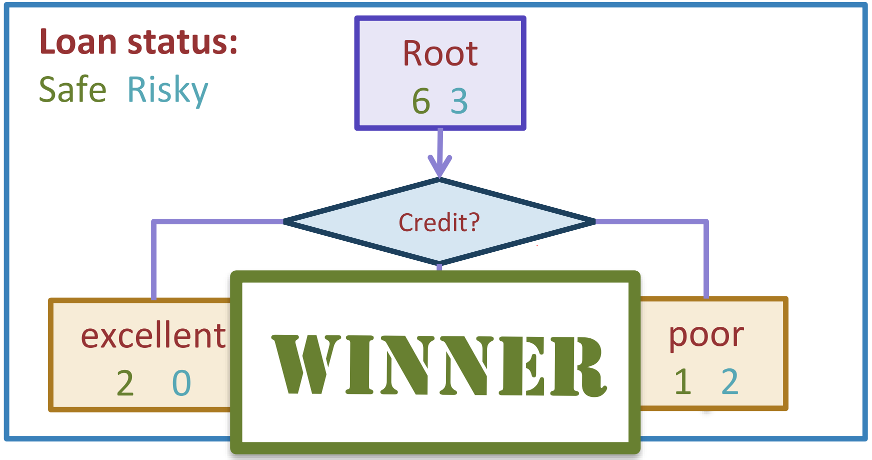
= 0.33

Tree	Classification error
(root)	0.33
Split on credit	0.22
Split on term	0.33

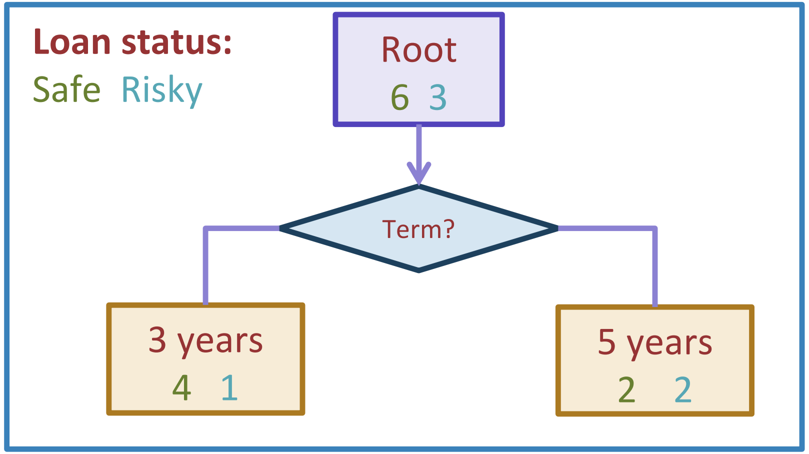
Choice 1 vs Choice 2: Comparing split on credit vs term

Tree	Classification error
(root)	0.33
split on credit	0.22
split on loan term	0.33

Choice 1: Split on Credit



Choice 2: Split on Term



Split Selection

Split(node)

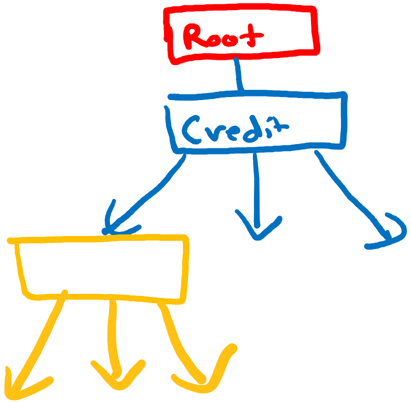
- Given M, the subset of training data at a node
- For each (remaining) feature $h_j(x)$:
  - Split data M on feature $h_j(x)$
  - Compute the classification error for the split
- Chose feature $h^*(x)$ with the lowest classification error

Greedy & Recursive Algorithm

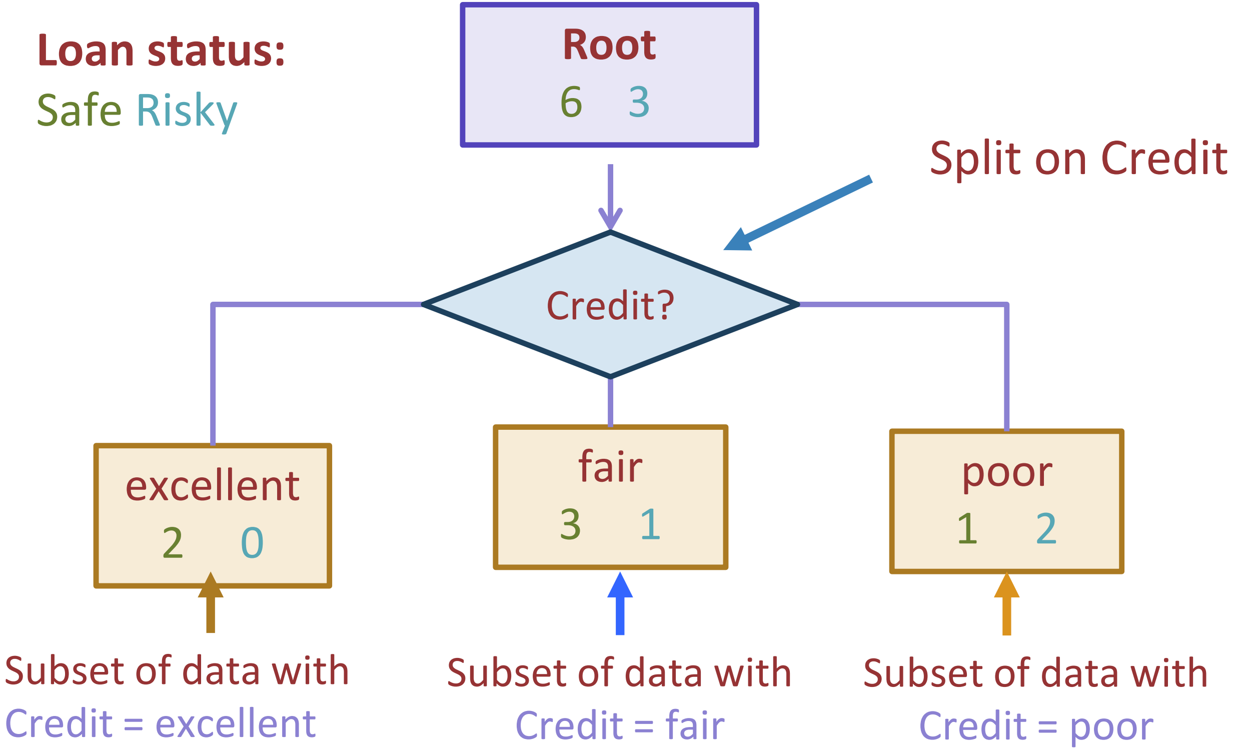
BuildTree(node)

- If termination criterion is met:
  - Stop
- Else:
  - Split(node)
  - For child in node:
    - BuildTree(child)

recursive

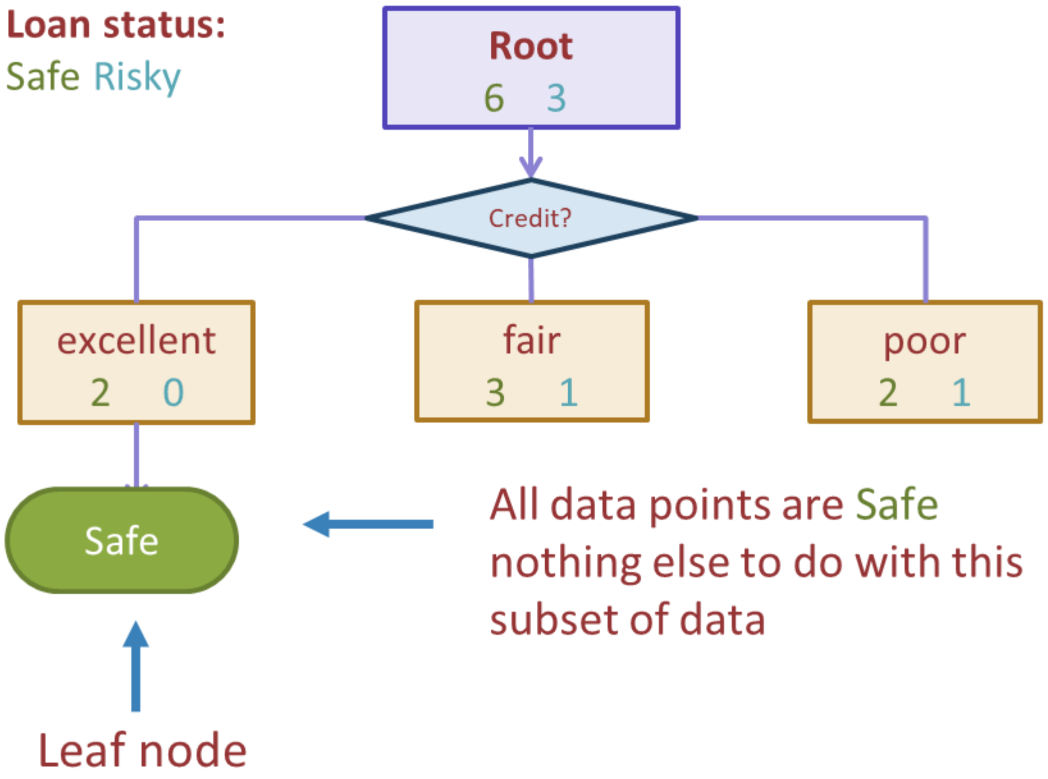


Decision tree expansion: 1 level

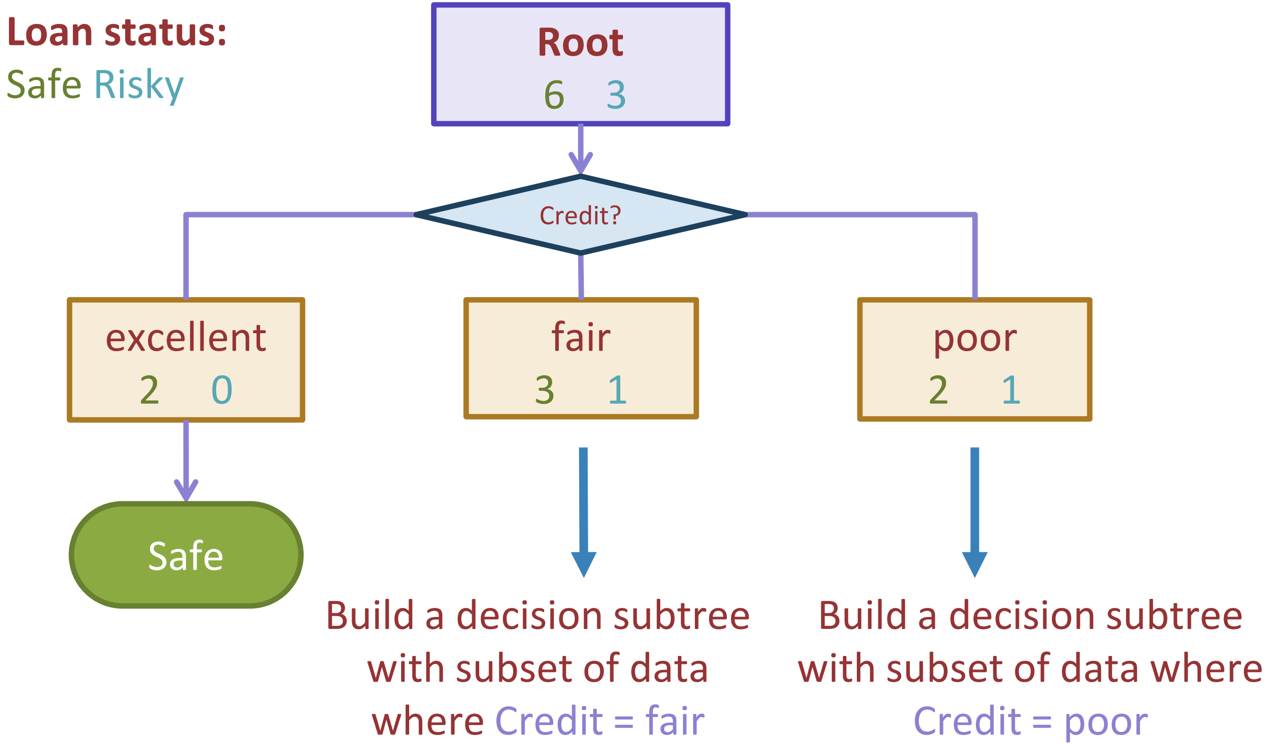


Termination Criteria

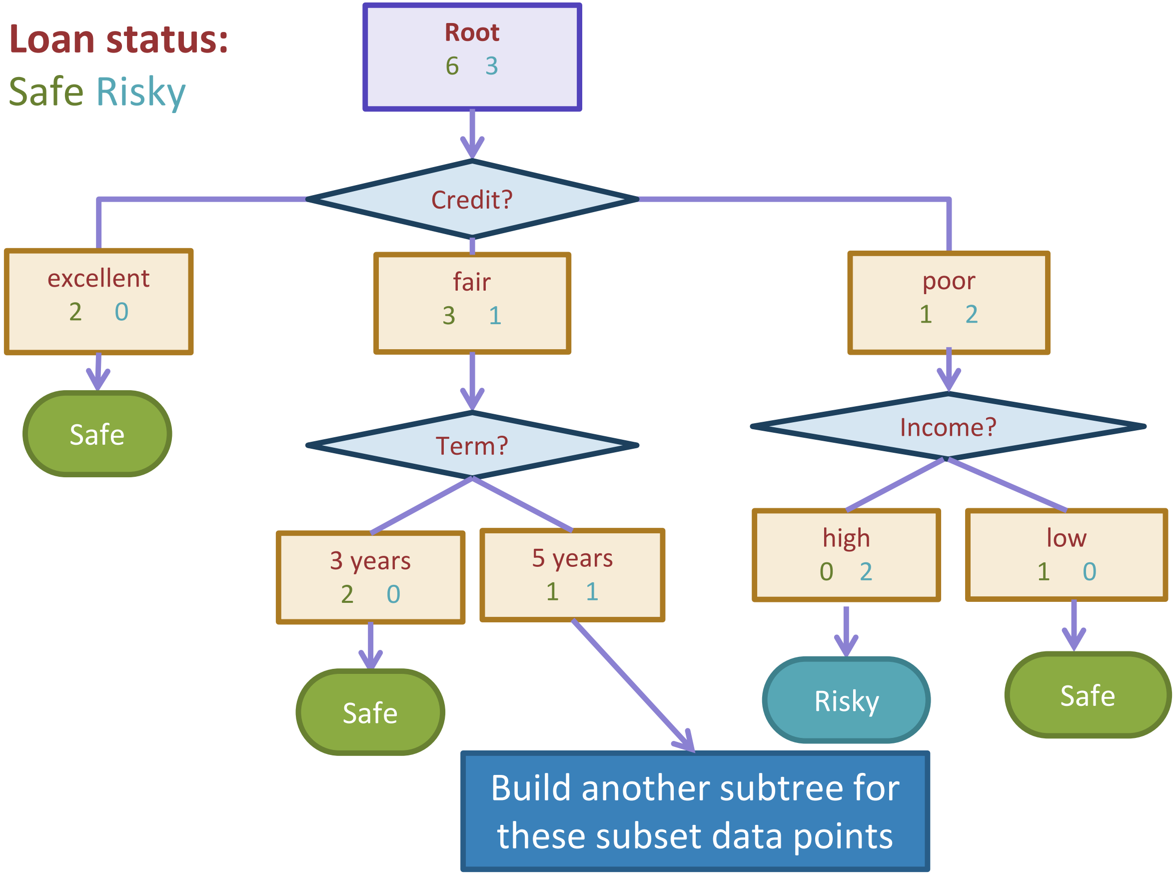
For now: Stop if all the points are in one class



Tree Learning = Recursive Stump Learning



Second level



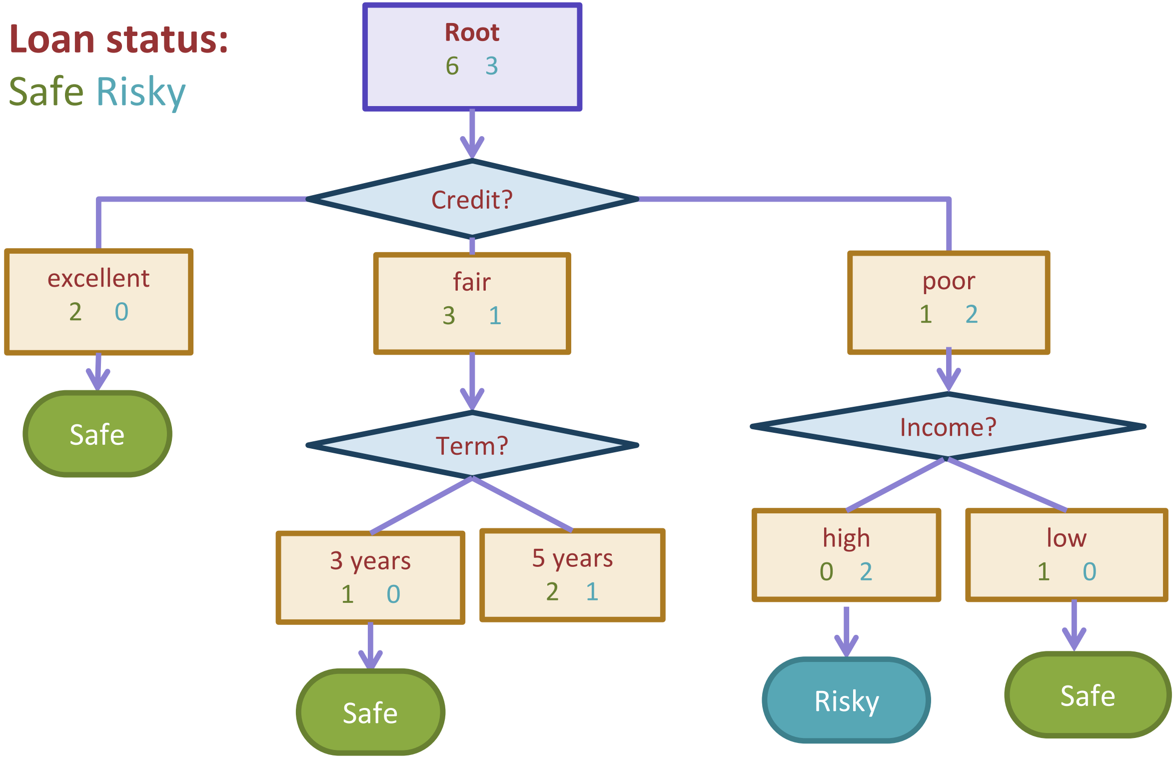
Poll Everywhere

Think

1 min

- What predictions **should** the below decision tree output for the following datapoints?

Credit	Term	Income
excellent	3 yrs	high
fair	5 yrs	low
poor	3 yrs	(missing)



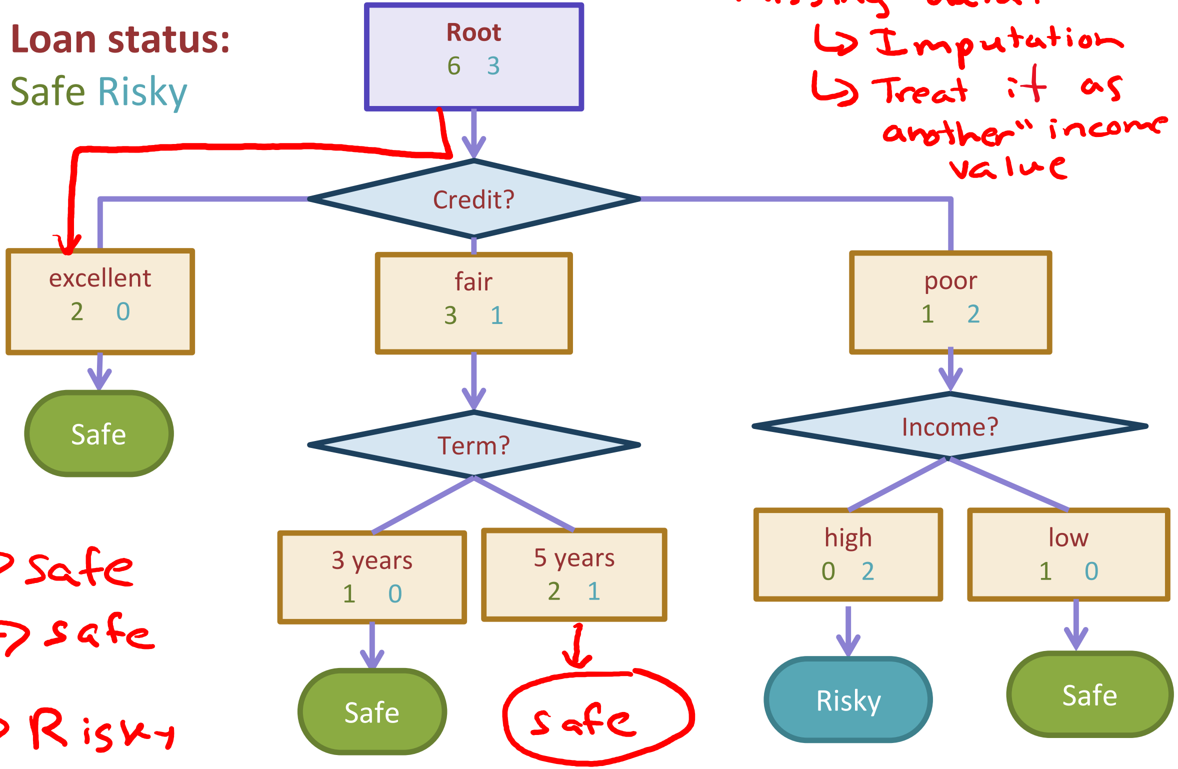
Poll Everywhere

Group

2 min

- What predictions **should** the below decision tree output for the following datapoints?

Two approaches for missing data:
↳ Imputation
↳ Treat it as another "income" value

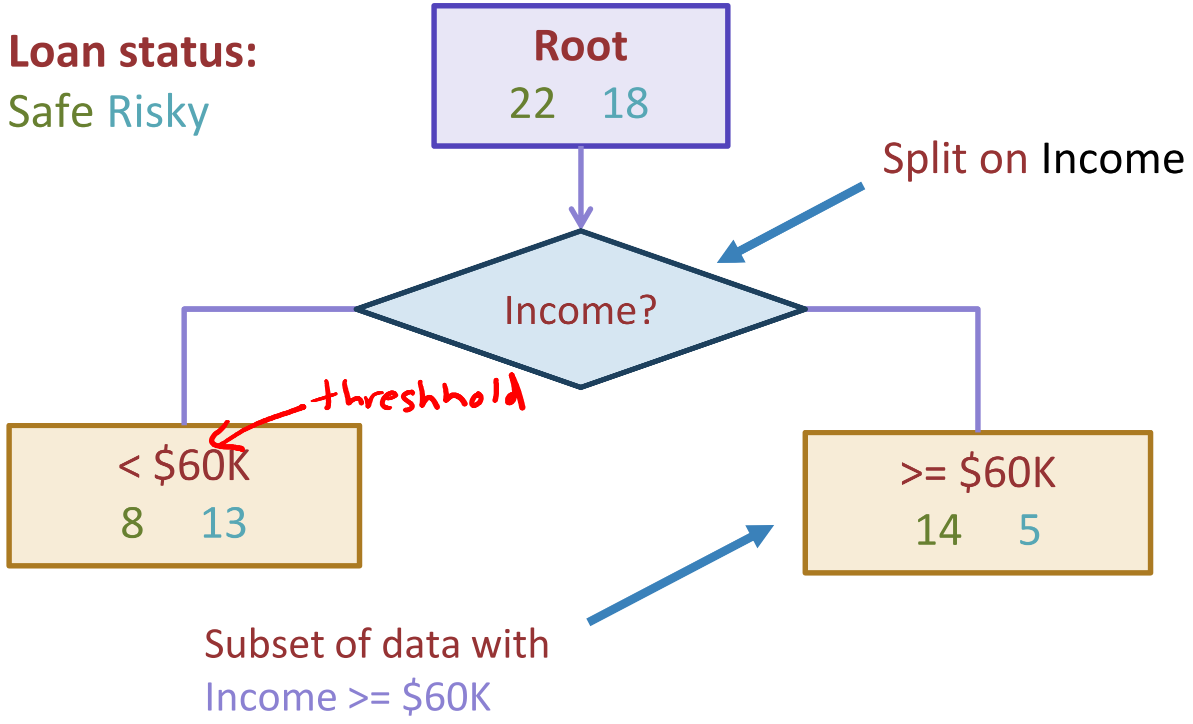


Credit	Term	Income	
excellent	3 yrs	high	→ safe
fair	5 yrs	low	→ safe
poor	3 yrs	(missing)	→ Risky

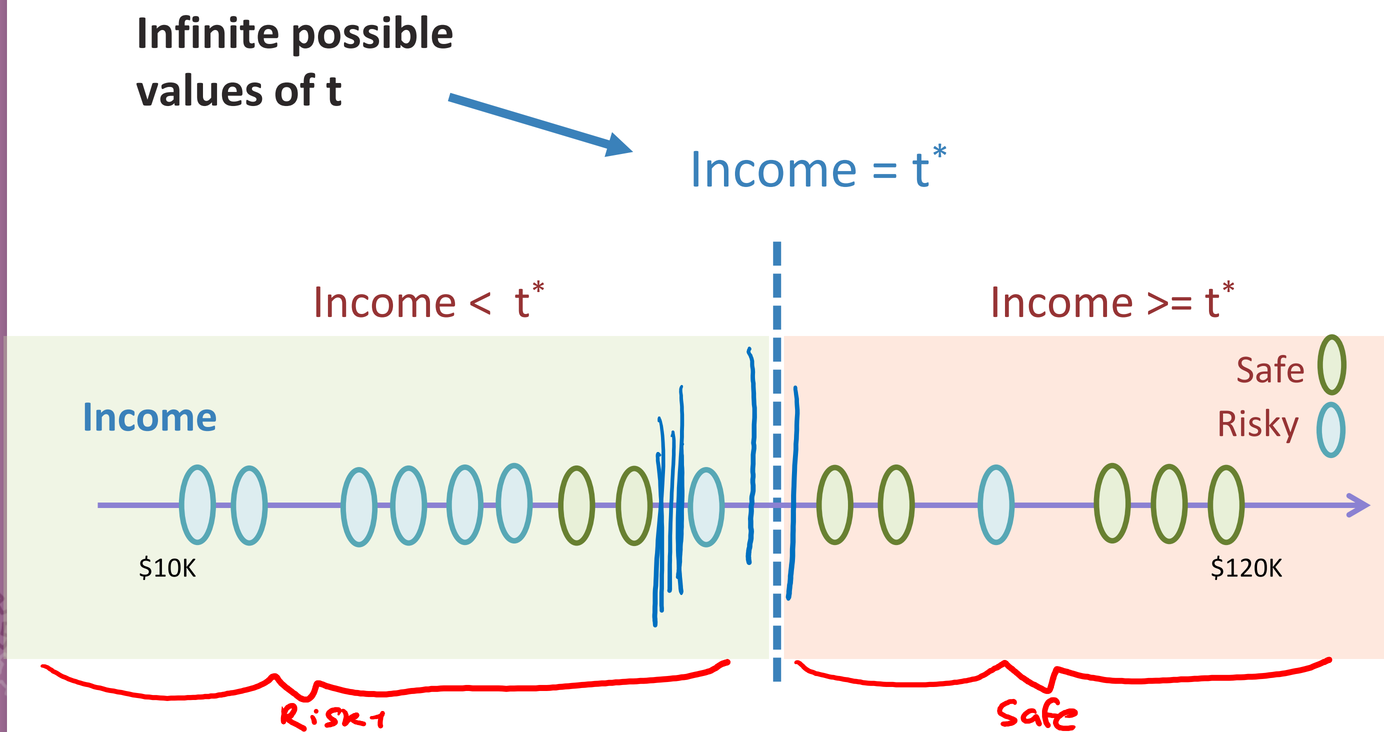
Splitting on Numeric Features

Income	Credit	Term	y
$105 K	excellent	3 yrs	Safe
$112 K	good	5 yrs	Risky
$73 K	fair	3 yrs	Safe
$69 K	excellent	5 yrs	Safe
$217 K	excellent	3 yrs	Risky
$120 K	good	5 yrs	Safe
$64 K	fair	3 yrs	Risky
$340 K	excellent	5 yrs	Safe
$60 K	good	3 yrs	Risky

Threshold Split for Numeric Features

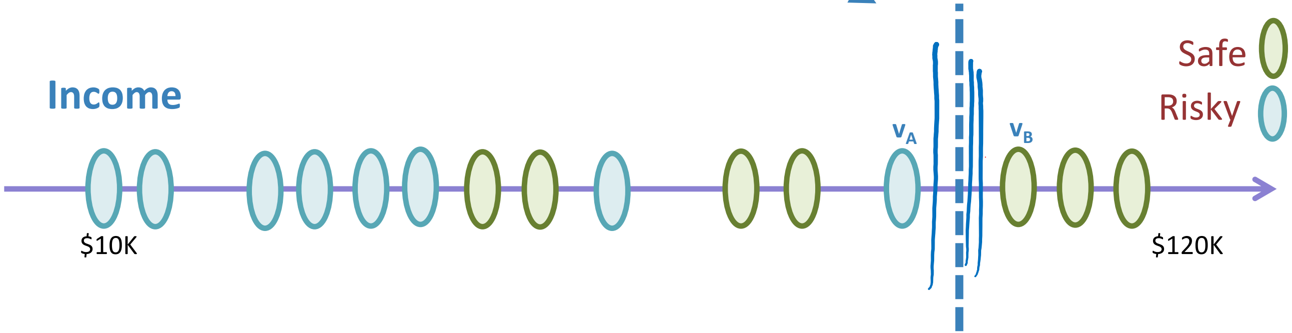


Best threshold?



Threshold Between Points

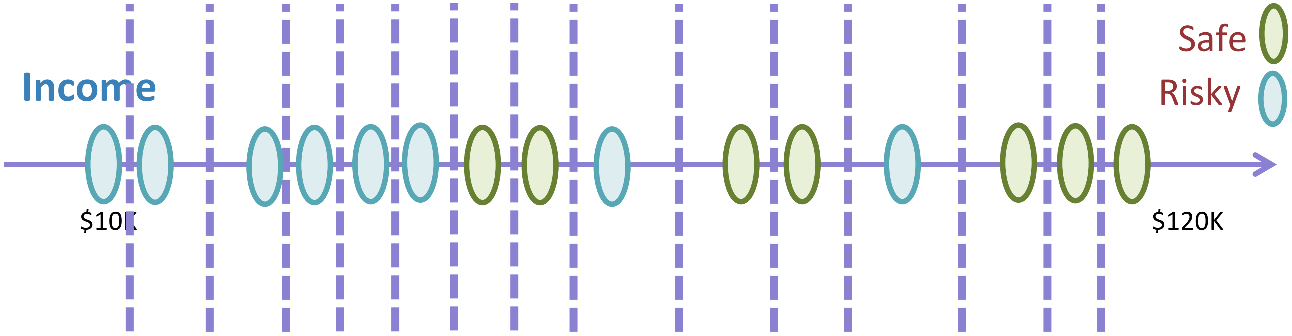
Same classification error for any threshold split between v_A and v_B



Only Need to Consider Mid-Points

splits to consider: n-1

Finite number of splits to consider



Splitting for Numeric Features

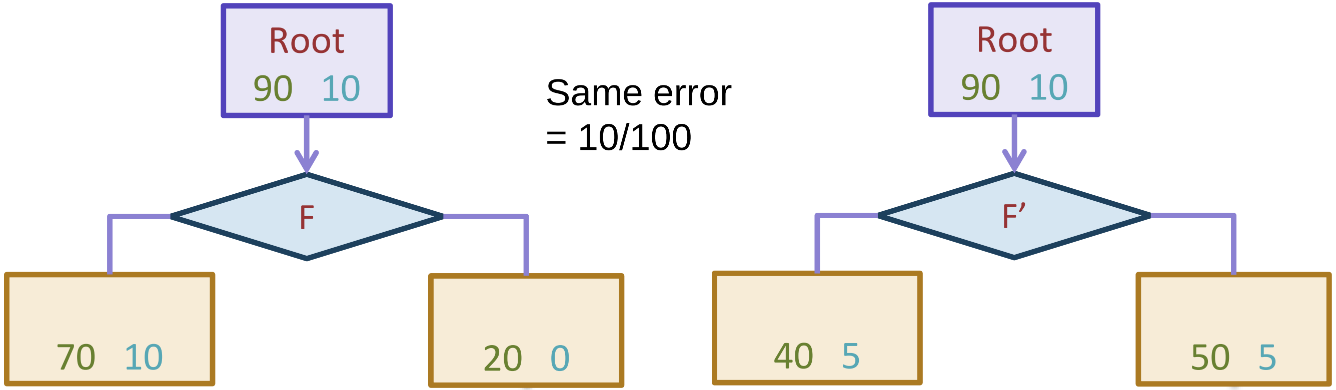
- Step 1: Sort the values of a feature $h_j(x)$:

  Let $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \ldots \mathbf{v_N}\}$ denote sorted values

- Step 2:
  - For i = 1 … N-1
    - Consider split $t_i = (\mathbf{v_i} + \mathbf{v_{i+1}}) / 2$
    - Compute classification error
  - Chose the $\mathbf{t^*}$ with the lowest classification error

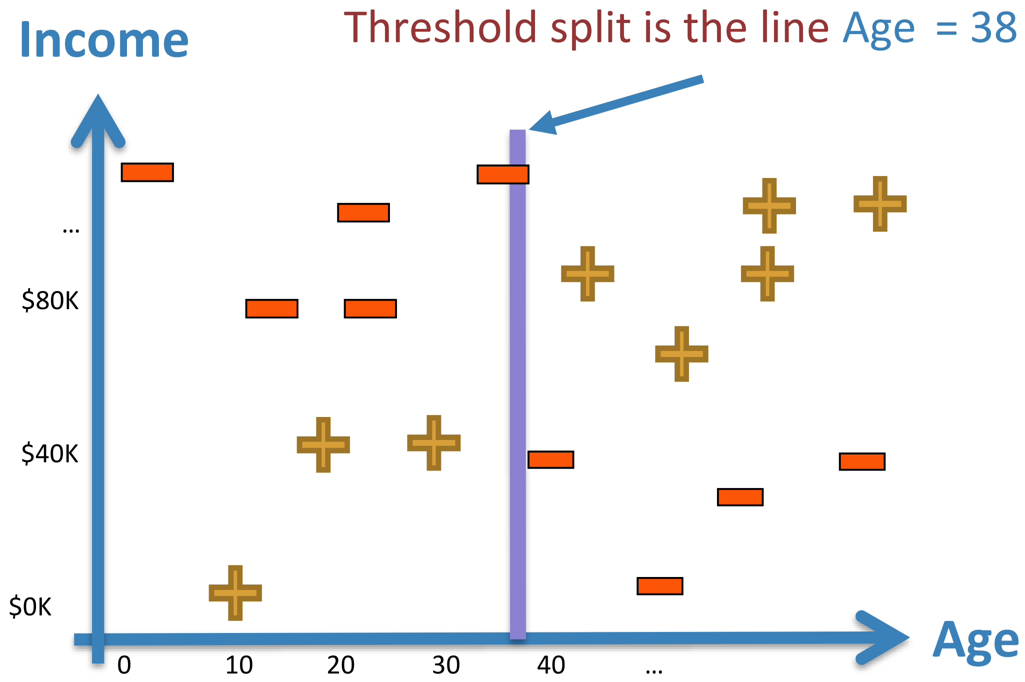
Issues With Using Classification Error for Splitting

- Is classification error sensitive enough to different splits?

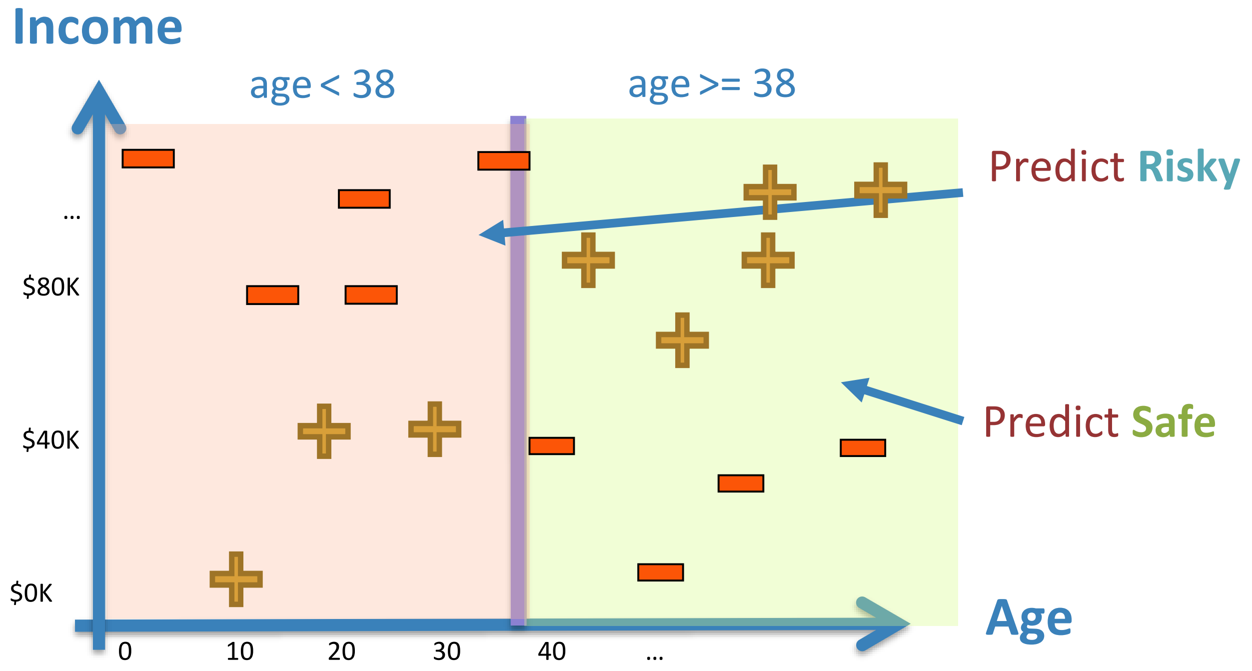


- In practice, people prefer using either Gini Impurity or Information Gain (not covered in this course) when computing the quality of a split.

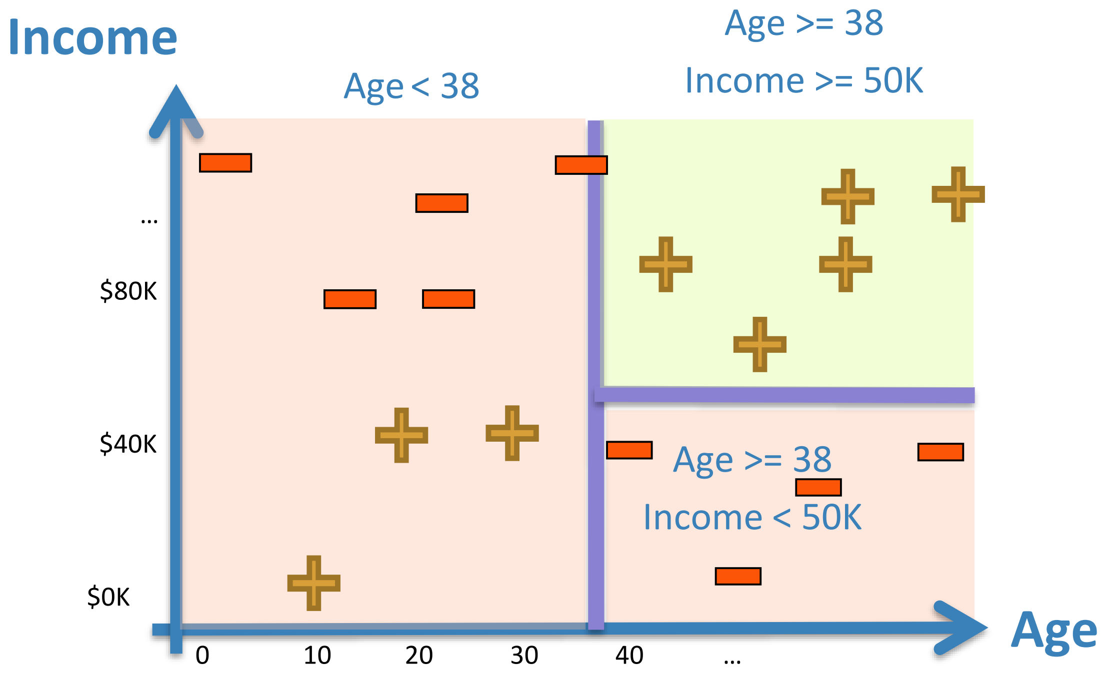
Visualizing the threshold split



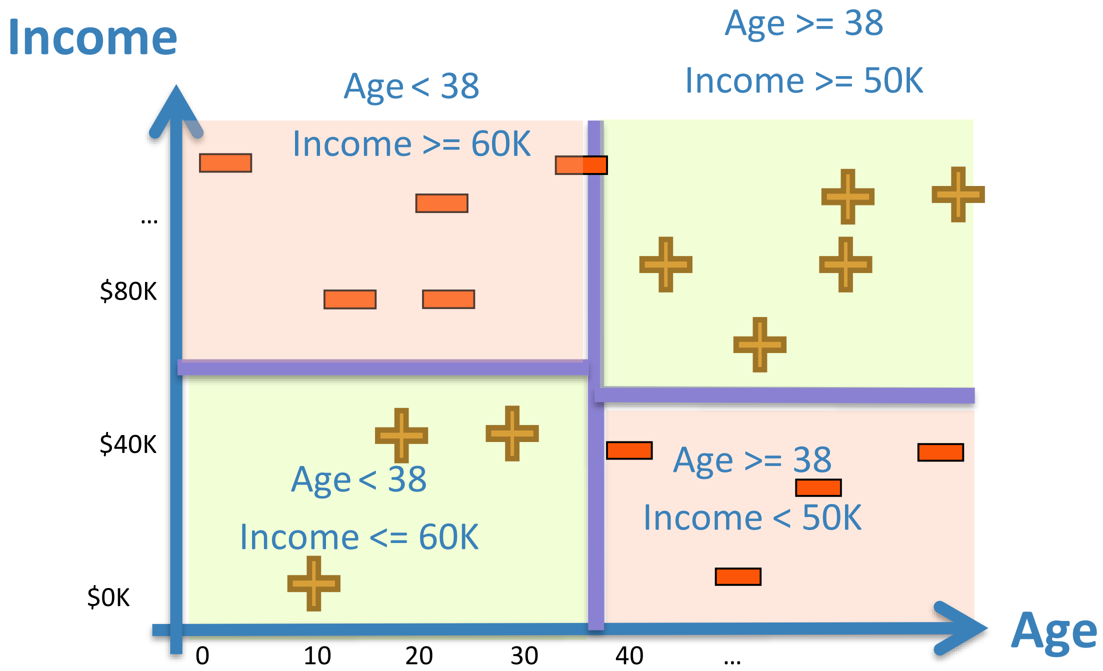
Split on Age >= 38



Each split partitions the 2-D space

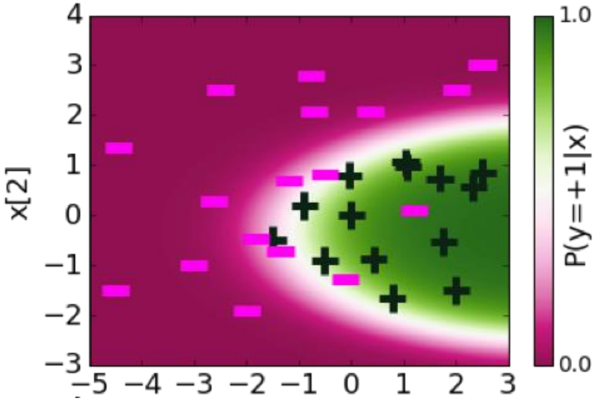


Each split partitions the 2-D space

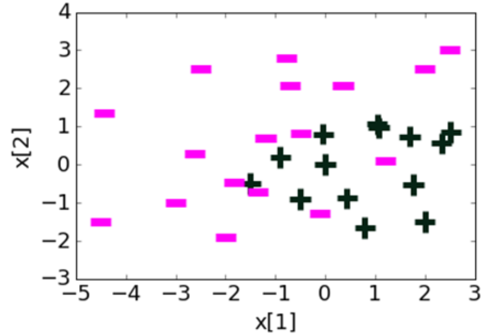


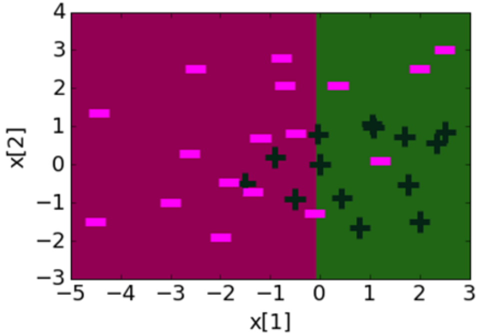
Will pick up from here on Wed

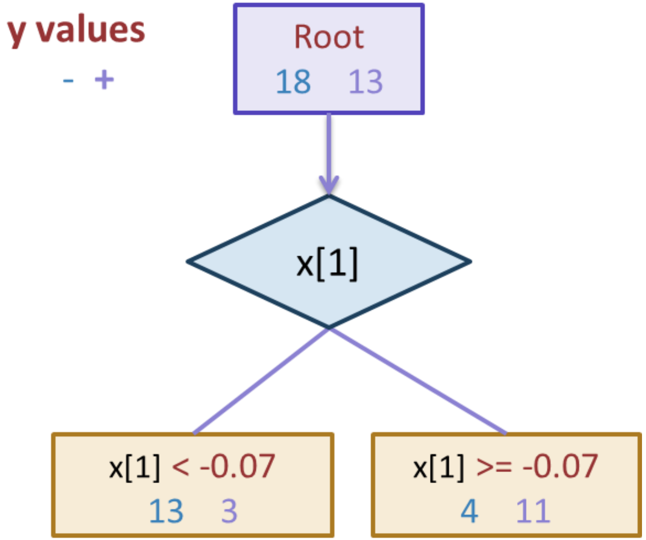
Recall This Example From Logistic Regression



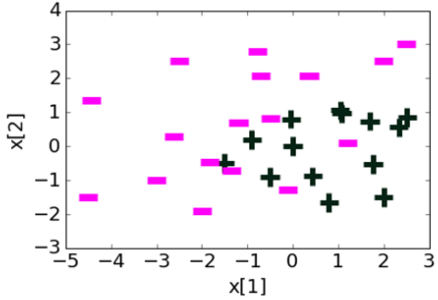
Depth 1: Split on x[1]



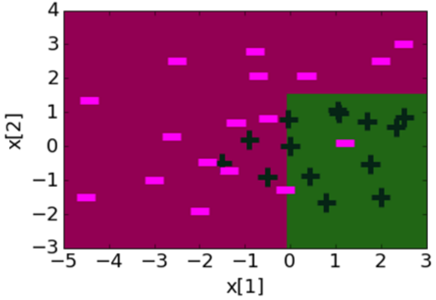




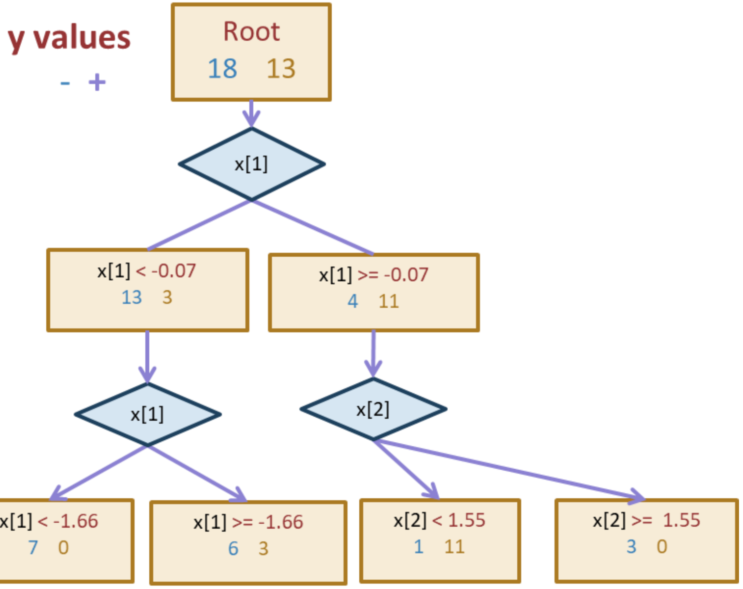
Depth 2



x[2]
x[1]



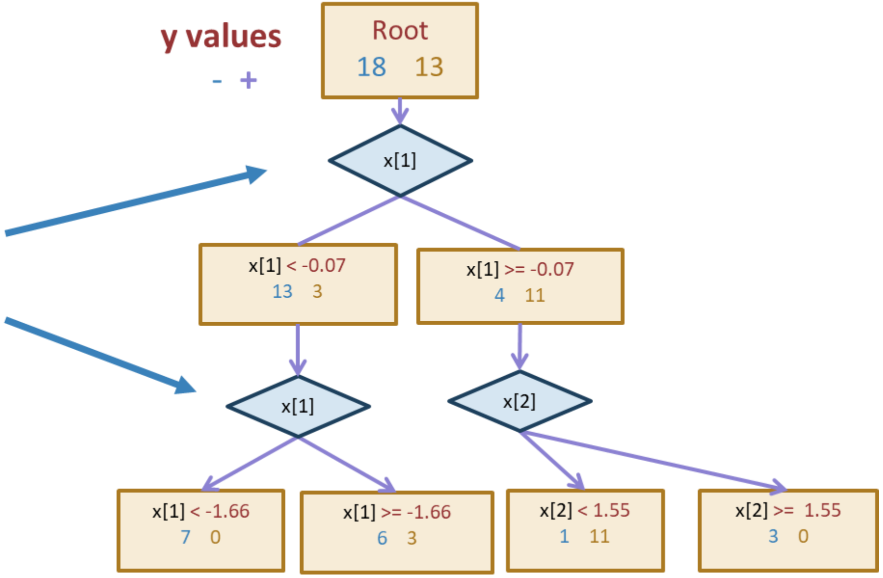
x[2]
x[1]



y values
- +
Root
18 13
x[1]
x[1] < -0.07
13 3
x[1] >= -0.07
4 11
x[1]
x[2]
x[1] < -1.66
7 0
x[1] >= -1.66
6 3
x[2] < 1.55
1 11
x[2] >= 1.55
3 0

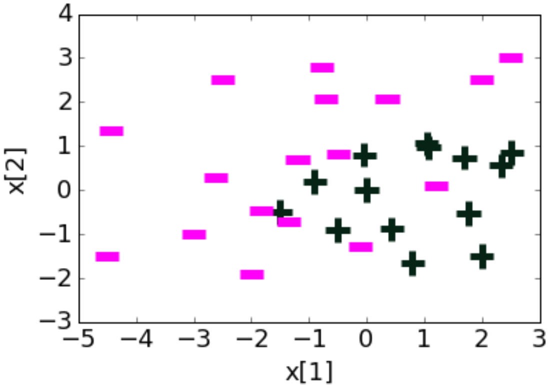
Same feature can be used to split multiple times

For threshold splits, same feature can be used multiple times

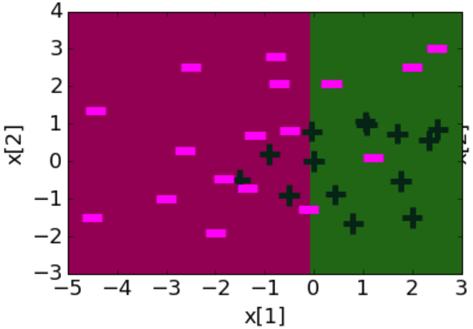


Decision boundaries at different depths

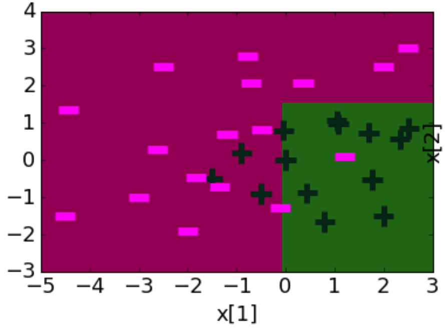
- Decision boundaries can be complex!



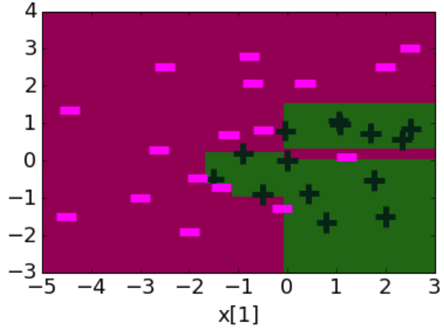
Depth 1



Depth 2



Depth 10



Overfitting Prevention (Termination Criteria Revisited)

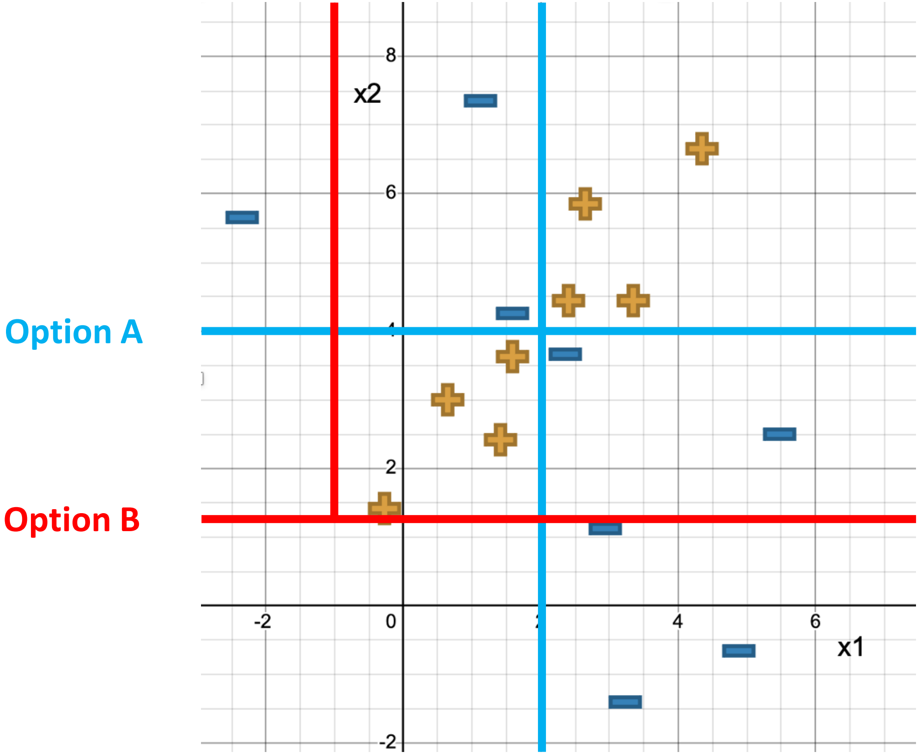
- Deep decision trees are prone to overfitting.
  - Decision boundaries are interpretable but not stable ().
  - Small changes in the dataset leads to big differences in the outcome.
- Overcome overfitting by stopping early:
  - Stop when the tree reaches a certain depth (e.g., 4 levels)
  - Stop when a leaf has ≤ some number of points (e.g., 20 pts)
    - Will use on HW
  - Stop if a split won’t decrease the classification error by more than some amount (e.g., 10%)
- Fine-tune hyperparameters using a validation set.
- Can also prune after growing a full-length tree

Poll Everywhere

Think

1 mins

- Which of the following decision boundaries will the algorithm we learnt output, if we stop growing the tree at a depth of two?
- **Hint**: compute classification errors.

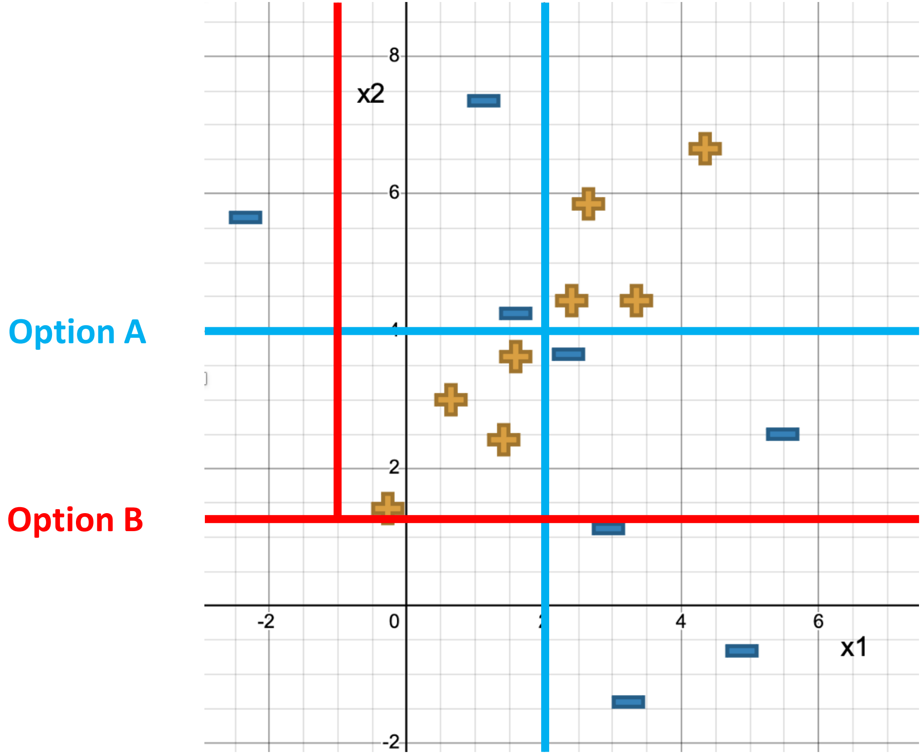


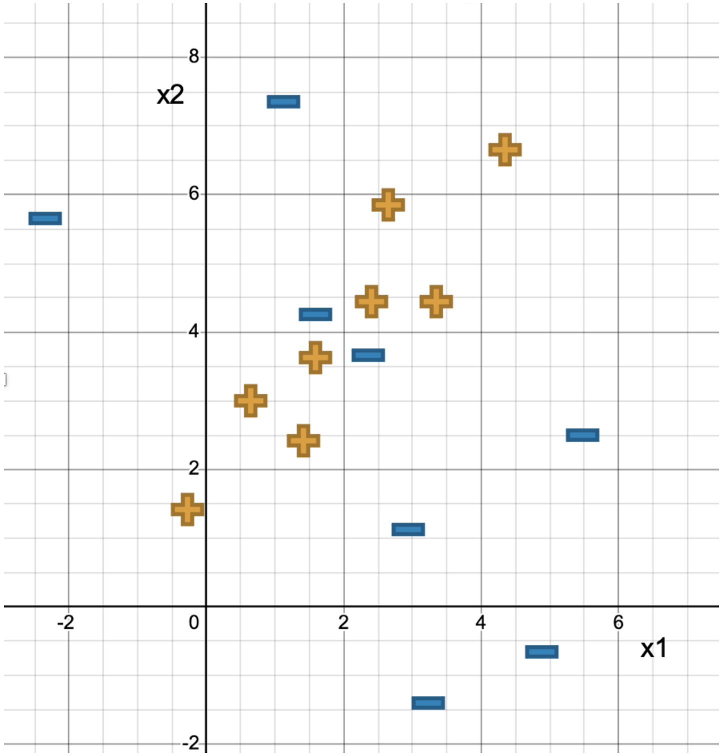
Poll Everywhere

Group

2 mins

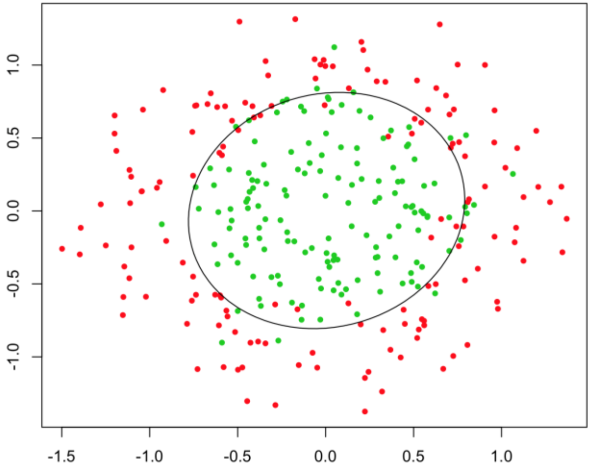
- Which of the following decision boundaries will the algorithm we learnt output, if we stop growing the tree at a depth of two?
- **Hint**: compute classification errors.




x2
8
6
4
2
-2
-2
0
2
4
6
x1

Pros/Cons Decision Tree

- Pros:
  - Easy to interpret
  - Handles numeric and categorical variables without preprocessing
  - No normalization required as it uses rule-based approach
  - Can create non-linear decision boundaries
  - Can readily do multi-class classification (unlike Logistic Regression)
- Cons:
  - Deep decision trees are prone to overfitting
  - Only allows axis-parallel decision boundaries



Recap

What you can do now:

- Understand parametric vs. non-parametric models
- Understand different data types and necessary preprocessing steps
- K-Nearest Neighbors:
  - Understand the k-Nearest Neighbor classifier
  - Make predictions using a k-Nearest Neighbors classifier
  - Understand pros and cons of a k-Nearest Neighbors classifier
- Decision Tree:
  - Define a decision tree classifier
  - Interpret the output of a decision trees
  - Learn a decision tree classifier using greedy & recursive algorithm
  - Advantages and Disadvantages of a decision tree
  - Ways to overcome overfitting in decision trees