Lecture 5: 19 April, 2021

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
April–July 2021

Categorical vs numeric attributes

- So far, all attributes have been categorical
- What age groups make up young, middle, old?
- How are these boundaries defined?
- How do we query numerical attributes?
  - Height, weight, length, income,

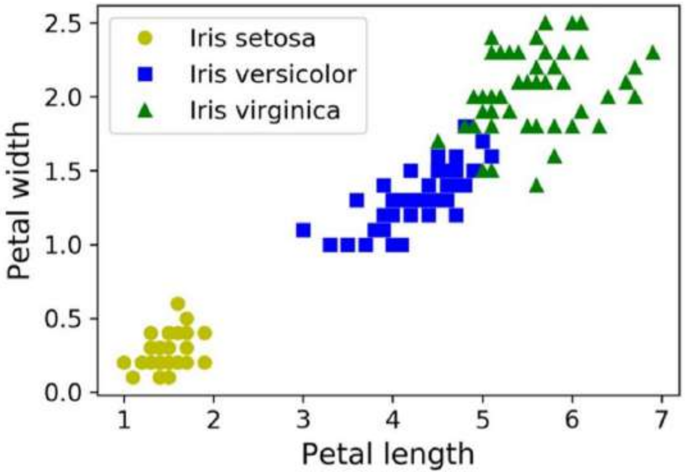
ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Iris dataset

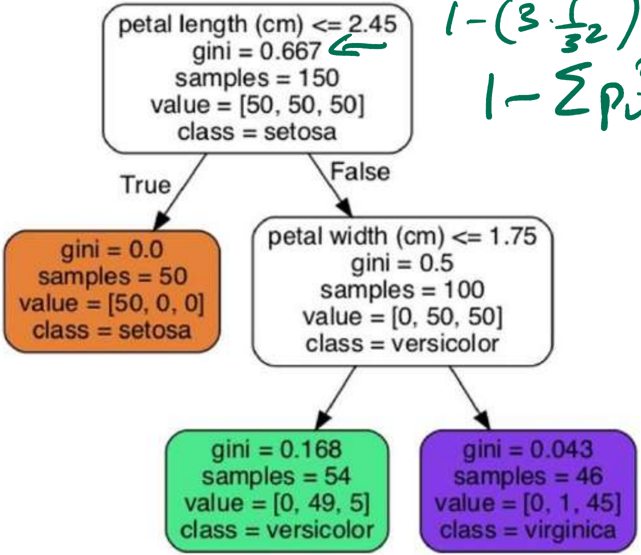
- Iris is a type of flower
- Three species: *iris setosa*, *iris versicolor*, *iris virginica*
- Dataset has sepal length and width and petal length and width for 150 flowers

Iris dataset

- Iris is a type of flower
- Three species: *iris setosa*, *iris versicolor*, *iris virginica*
- Dataset has sepal length and width and petal length and width for 150 flowers
- Scatter plot for two attributes, petal length and petal width



- Iris is a type of flower
- Three species: *iris setosa*, *iris versicolor*, *iris virginica*
- Dataset has sepal length and width and petal length and width for 150 flowers
- Scatter plot for two attributes, petal length and petal width
- Decision tree for this data set

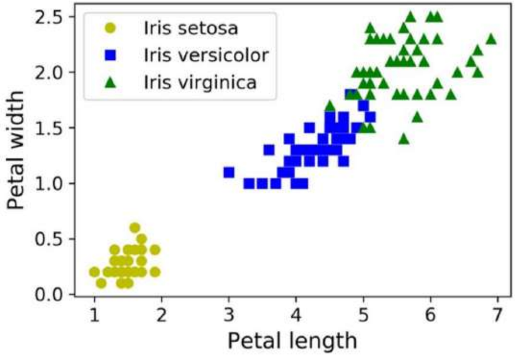


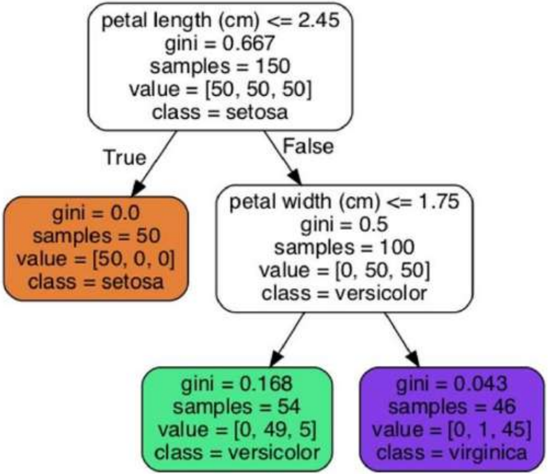
$1-(3\cdot\frac{1}{3^2})$

$1-\sum p_i^2$

Decision tree for iris dataset

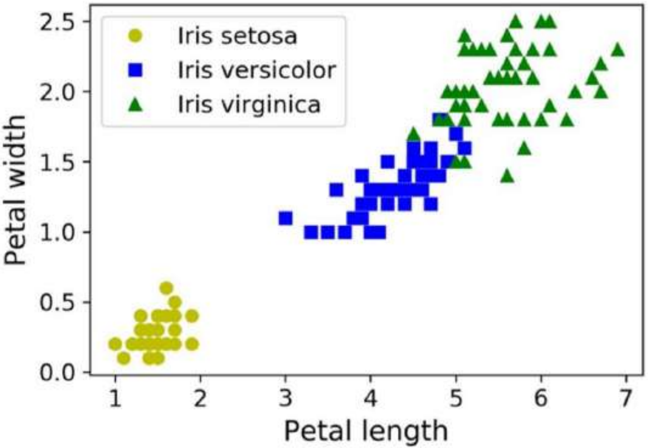
- Queries compare numerical attribute against a value
- How do we find these query values?





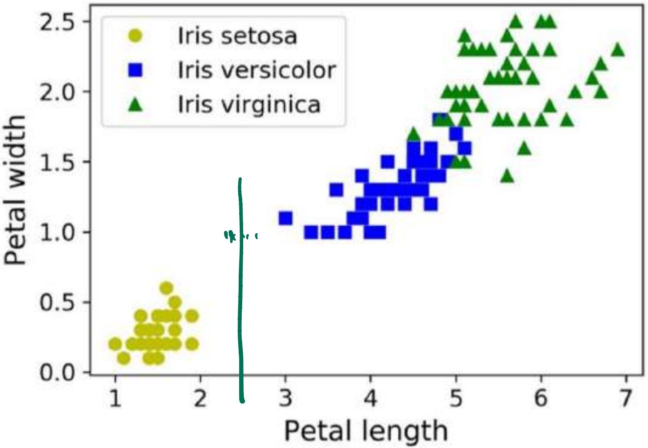
Querying numerical attributes

- Numerical attribute takes values in a range $[L, U]$
  - Petal length : $[1, 7]$
  - Petal width : $[0, 2.5]$



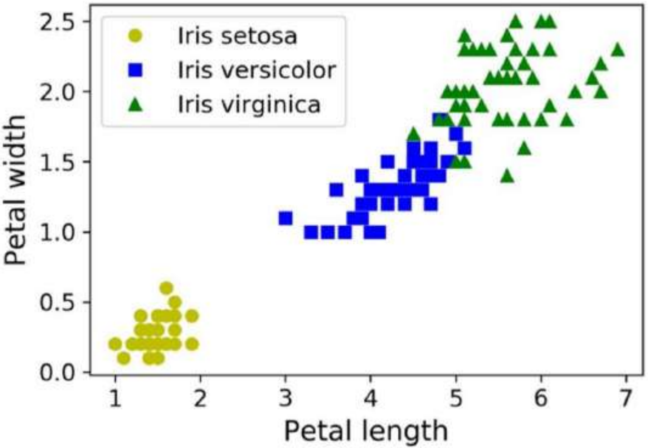
Querying numerical attributes

- Numerical attribute takes values in a range $[L, U]$
  - Petal length : $[1, 7]$
  - Petal width : $[0, 2.5]$
- Pick a value v in the range and check if $A \leq \underline{v}$ Threshold



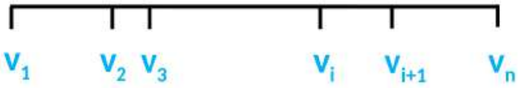
Querying numerical attributes

- Numerical attribute takes values in a range $[L, U]$
  - Petal length : $[1, 7]$
  - Petal width : $[0, 2.5]$
- Pick a value v in the range and check if $A \leq v$
- Infinitely many choices for v
- How do we pick a sensible one?



Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$

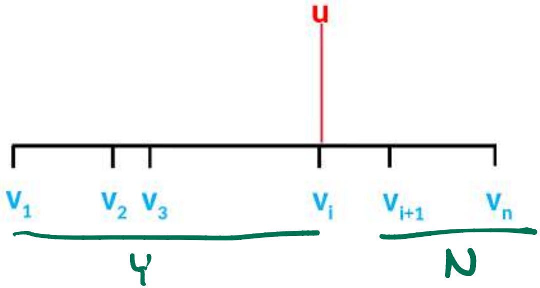


Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$
- Consider interval $[v_i, v_{i+1}]$

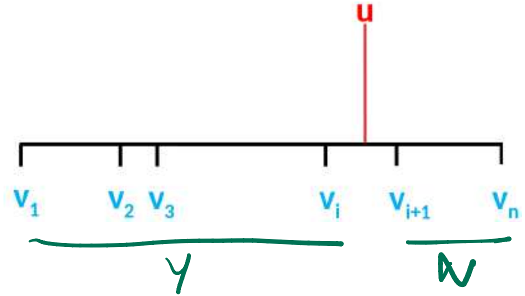
Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$
- Consider interval $[v_i, v_{i+1}]$
- For each $v_i \leq u < v_{i+1}$, query $A \leq u$ gives the same answer



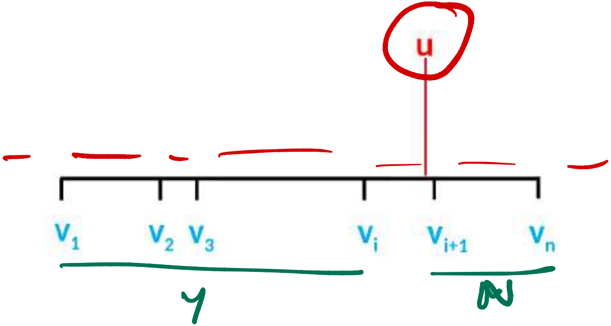
Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$
- Consider interval $[v_i, v_{i+1}]$
- For each $v_i \leq u < v_{i+1}$, query $A \leq u$ gives the same answer



Querying numerical attributes

- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$
- Consider interval $[v_i, v_{i+1}]$
- For each $v_i \leq u < v_{i+1}$, query $A \leq u$ gives the same answer
- Only $n-1$ useful intervals to check



Querying numerical attributes

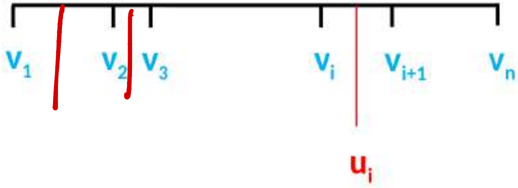
- Only n values for A in training data
  - Sort as $v_1 < v_2 < \cdots < v_n$
- Consider interval $[v_i, v_{i+1}]$
- For each $v_i \leq u < v_{i+1}$, query $A \leq u$ gives the same answer
- Only $n-1$ useful intervals to check
- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval

$A \leq u_1$

$A \leq u_2$

$\vdots$

$A \leq u_{n-1}$



Querying numerical attributes

- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval
- Each query $A \leq u_i$ partitions training data

Querying numerical attributes

- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval
- Each query $A \leq u_i$ partitions training data
- Choose the query $A \leq u_i$ with maximum information gain
- Assign this as the information gain for this attribute

Querying numerical attributes

- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval
- Each query $A \leq u_i$ partitions training data
- Choose the query $A \leq u_i$ with maximum information gain
- Assign this as the information gain for this attribute
- Compare across all attributes and choose best one

Attributes $A_1, \ldots, A_k$

Question for each A_k

Pick A_j whose split is best

Querying numerical attributes

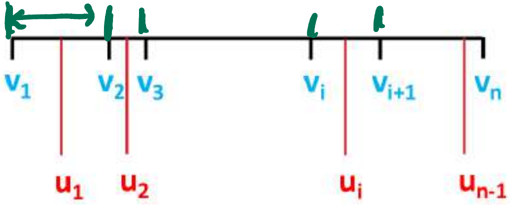
- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval
- Each query $A \leq u_i$ partitions training data
- Choose the query $A \leq u_i$ with maximum information gain
- Assign this as the information gain for this attribute
- Compare across all attributes and choose best one

- Any point within an interval can be used

Querying numerical attributes

- Pick midpoint $u_i = (v_i + v_{i+1})/2$ as query value for each interval
- Each query $A \leq u_i$ partitions training data
- Choose the query $A \leq u_i$ with maximum information gain
- Assign this as the information gain for this attribute
- Compare across all attributes and choose best one

$A \leq u_i$



- Any point within an interval can be used
- May prefer endpoints — midpoints may not be meaningful values

Building a decision tree

- For each numerical attribute, choose query $A \leq v$ with maximum information gain

Building a decision tree

- For each numerical attribute, choose query $A \leq v$ with maximum information gain
- Across all categorical and numerical attributes, choose the one with best information gain

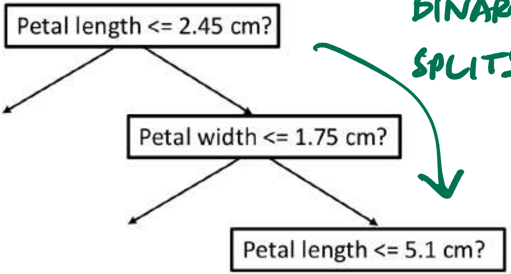
Building a decision tree

- For each numerical attribute, choose query $A \leq v$ with maximum information gain
- Across all categorical and numerical attributes, choose the one with best information gain
- Categorical attrbutes can be queried only once on a path
- Numerical attributes can be queried repeatedly — interval to query keeps shrinking

CART – Breiman

classification regression

BINARY SPLITS



$2.45 < l \leq 5.1$

Testing a supervised learning model

- How do we validate software?
  - Test suite of carefully selected inputs
  - Compare output with expected answers

Testing a supervised learning model

- How do we validate software?
  - Test suite of carefully selected inputs
  - Compare output with expected answers
- What about classification models?
  - By definition, deploy on data where the outcome is unknown
  - If expected answer available, have a deterministic solution, model not needed!

Testing a supervised learning model

- How do we validate software?
  - Test suite of carefully selected inputs
  - Compare output with expected answers
- What about classification models?
  - By definition, deploy on data where the outcome is unknown
  - If expected answer available, have a deterministic solution, model not needed!
- On what basis can we evaluate a supervised learning model?

Creating a test set

- Training data is labelled
  - No other source of inputs with expected answers

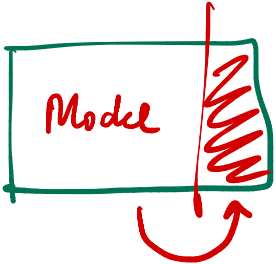
Training data

↓ Generalization

Unseen data

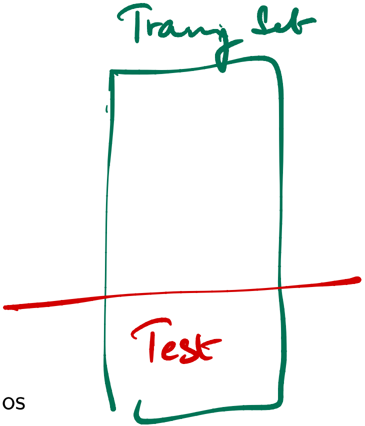
Creating a test set

- Training data is labelled
  - No other source of inputs with expected answers
- Segregate some training data for testing
  - Terminology: training set and test set
  - Build model using training set, evaluate on test set



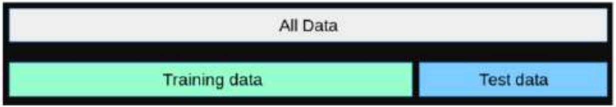
Creating a test set

- Training data is labelled
  - No other source of inputs with expected answers
- Segregate some training data for testing
  - Terminology: training set and test set
  - Build model using training set, evaluate on test set
- Creating the test set
  - Need to choose a random sample
  - Can further use stratified sampling, preserve relative ratios (e.g., age wise distribution)
  - ML libraries can do this automatically



Creating a test set

- How large should the test set be?
  - Typically 20-30% of labelled data
- Depends on labelled data available
  - Need enough training data to build the model

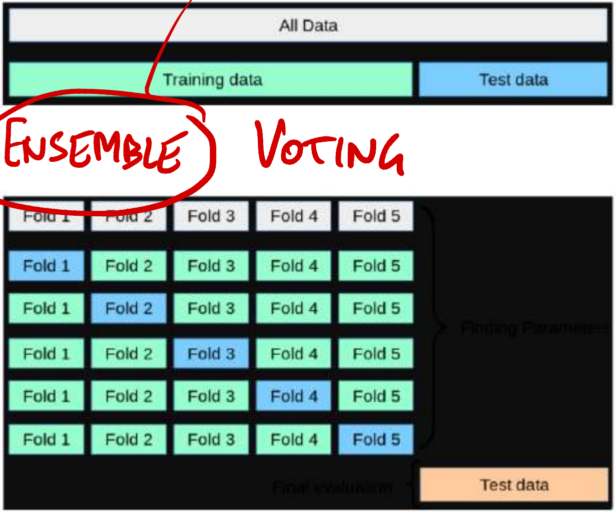


Creating a test set

- How large should the test set be?
  - Typically 20-30% of labelled data
- Depends on labelled data available
  - Need enough training data to build the model

GROUP_SET

ENSEMBLE VOTING



Cross validation

- Partition labelled data into k chunks
- Hold out one chunk at a time
- Build k models, using $k-1$ chunks for training, 1 for testing
- Useful if labelled data is scarce

What are we measuring?

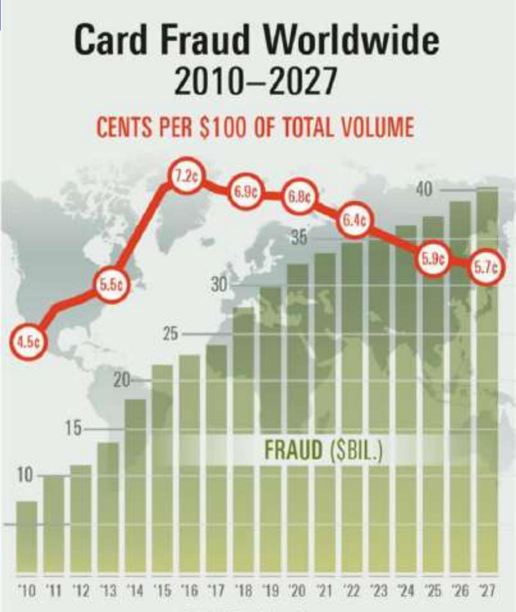
- Accuracy is an obvious measure
  - Fraction of inputs where classification is correct

What are we measuring?

- Accuracy is an obvious measure
  - Fraction of inputs where classification is correct
- Classifiers are often used in asymmetric situations
  - Less than 1% of credit card transactions are fraud

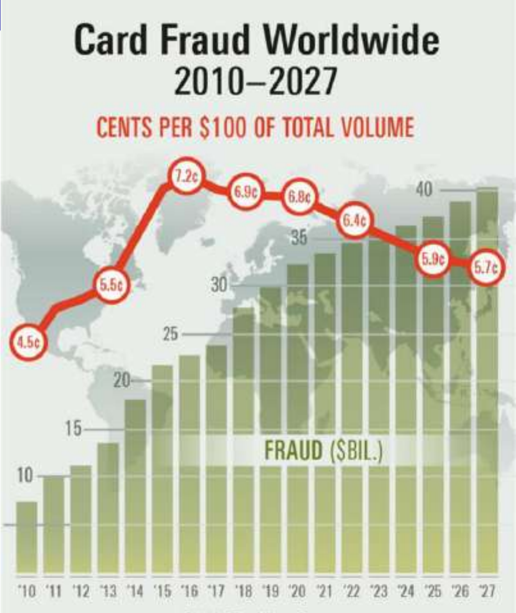
$$\frac{7}{1000}$$

$$\frac{100}{10000}$$

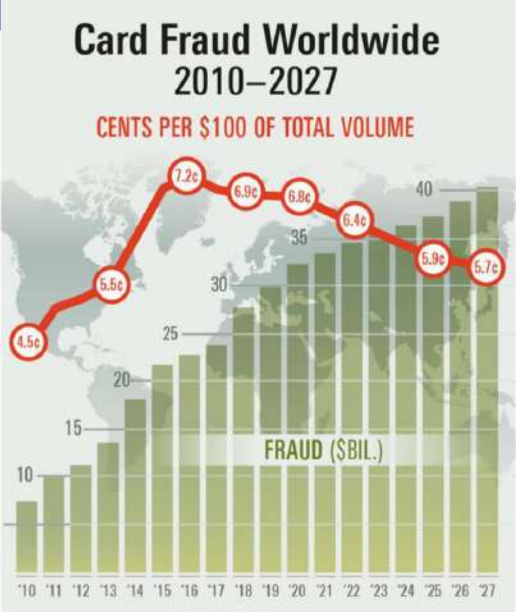


What are we measuring?

- Accuracy is an obvious measure
  - Fraction of inputs where classification is correct
- Classifiers are often used in asymmetric situations
  - Less than 1% of credit card transactions are fraud
- "Is this transaction a fraud?"
  - Trivial classifier — always answer "No"
  - More than 99% accurate, but useless!

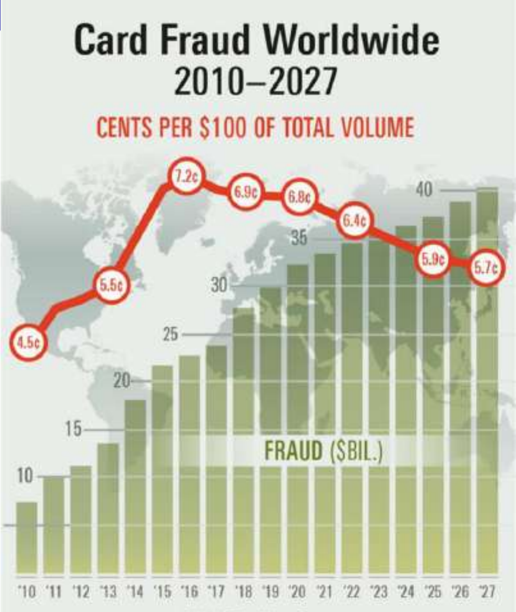


- The minority case is the useful case
  - Assume question is phrased so that minority answer is "Yes"
  - Want to flag as many "Yes" cases as possible



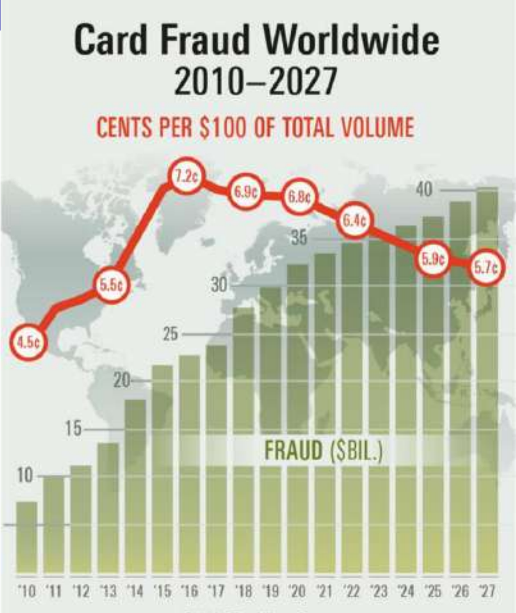
Catching the minority case

- The minority case is the useful case
  - Assume question is phrased so that minority answer is "Yes"
  - Want to flag as many "Yes" cases as possible
- Aggressive classifier
  - Marks borderline "No" as "Yes"
  - False positives



Catching the minority case

- The minority case is the useful case
  - Assume question is phrased so that minority answer is "Yes"
  - Want to flag as many "Yes" cases as possible
- Aggressive classifier
  - Marks borderline "No" as "Yes"
  - False positives
- Cautious classifier
  - Marks borderline "Yes" as "No"
  - False negatives



Confusion matrix

- Four possible combinations
  - Actual answer: Yes / No
  - Prediction: Yes / No

Confusion matrix

- Four possible combinations
  - Actual answer: Yes / No
  - Prediction: Yes / No
- Record all four possibilities in confusion matrix
  - Correct answers
    - True positives, true negatives
  - Wrong answers
    - False positives, false negatives

Actual Ans

Model

	Classified positive	Classified negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

Correct or not

Column = Model verdict

Performance measures

Precision

- What percentage of positive predictions are correct?

$$\frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall

- What percentage of actual positive cases are discovered?

$$\frac{\text{TP}}{\text{TP} + \text{FN}}$$

	Classified positive	Classified negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

$$\text{Accuracy} = \frac{\text{Right}}{\text{Total}} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$$

Performance measures

- Precision 1, Recall 0.01

	Classified positive	Classified negative
Actual positive	1	99
Actual negative	0	900

Performance measures

- Precision 1, Recall 0.01
- Recall up to 0.4, but precision down to 0.29

	Classified positive	Classified negative
Actual positive	40	60
Actual negative	100	800

Performance measures

- Precision 1, Recall 0.01
- Recall up to 0.4, but precision down to 0.29
- Recall up to 0.99, but precision down to 0.165

	Classified positive	Classified negative
Actual positive	99	1
Actual negative	500	400

Performance measures

- Precision 1, Recall 0.01
- Recall up to 0.4, but precision down to 0.29
- Recall up to 0.99, but precision down to 0.165
- Precision-recall tradeoff
  - Strict classifiers : fewer false positives (high precision), miss more actual positives (low recall)
  - Permissive classifiers : catch more actual positives (high recall) but more false positives (low precision)

	Classified positive	Classified negative
Actual positive	99	1
Actual negative	500	400

Performance measures

- Which measure is more useful?
  - Depends on situation
- Hiring
  - Screening test: high recall
  - Interview: high precision
- Medical diagnosis
  - Immunization: high recall
  - Critical illness diagnosis: high precision

	Classified positive	Classified negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

Performance measures

Other measures, terminology

- Recall is also called sensitivity
- Accuracy: (TP+TN)/(TP+TN+FP+FN)
- Specificity: TN/(TN+FP)
- Threat score: TP/(TP+FP+FN)
  - TN usually majority, ignore, not useful

	Classified positive	Classified negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

Performance measures

Other measures, terminology

- Recall is also called sensitivity
- Accuracy: (TP+TN)/(TP+TN+FP+FN)
- Specificity: TN/(TN+FP)
- Threat score: TP/(TP+FP+FN)
  - TN usually majority, ignore, not useful

	Classified positive	Classified negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

F Score

- A single combined score
- Harmonic mean of precision, recall

$$\frac{2pr}{p+r}$$

$$\left[\frac{\frac{1}{r}+\frac{1}{p}}{2}\right]$$