IST407/707 Applied Machine Learning

Decision Trees

Syracuse University
School of Information Studies

Agenda

Model Evaluation Techniques

Class Project Overview

Model Evaluation

Topics:

- Review model development process
- Model overfitting
- Model evaluation methods and metrics
- Model comparison and selection

Model Development Process Review

The Automated Classification Process

Two steps

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Learning algorithm

Induction

Learn Model

Model

Apply Model

Deduction

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

The first step is induction. In this process we use a training data set and a learning algorithm to induce, or develop a learning model.

The second step is deduction, where we apply the learned model to test data, in which the decisions are unknown

An Example of Decision Tree

Problem: to label each person as to whether they will cheat IRS

categorical, categorical, continuous, class

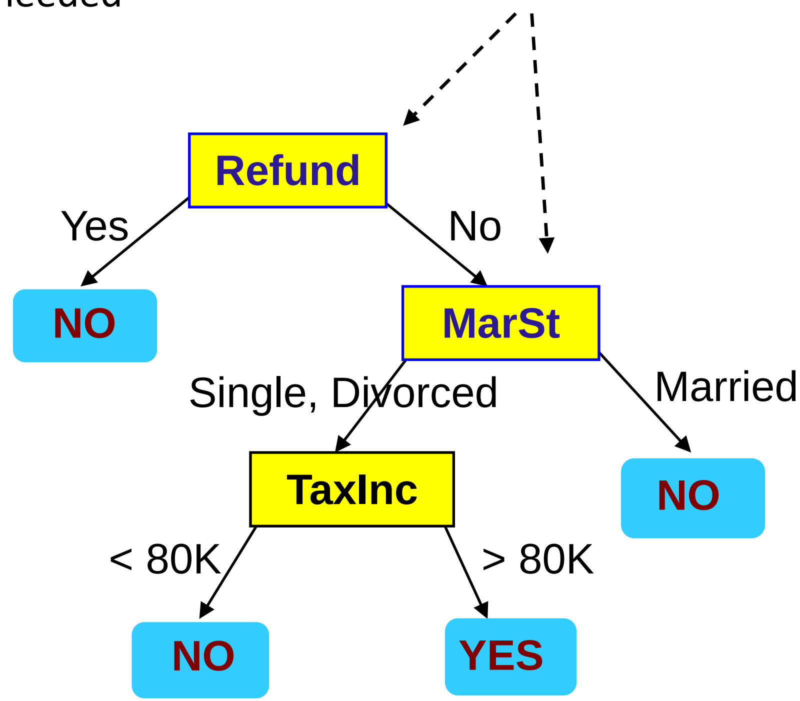
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Training Data

MarSt = Married is pure. No additional split needed

Married = Single / Divorced is not pure. Additional split needed

Splitting Attributes



Model: Decision Tree

Summary of Decision Trees

Strengths of decision trees are that they:

- Fast in prediction
- Interpretable patterns
- Robust to noise

Weaknesses of decision trees are that they:

- Tend to overfit (pruning helps)
- Are error prone with too many classes
- Are computationally expensive in training (compared to the low cost in prediction)

Model Overfitting

Model Overfitting

Overfitting means a model fits the training data very well but generalizes to unseen data poorly.

Therefore, if the test error is much higher than training error, the model is more likely to be overfitting

Model Overfitting

Two fundamental concepts

- **Training error**: train a model (e.g. a decision tree) on a training set, then test the model on the same training set.
  - The error rate is called "training error", which measures how well the model fits the training data.

- **Test error**: test the model on a test set that is different from the training set.
  - The error rate is called "test error", which measures how well the model generalizes to new, unseen data.

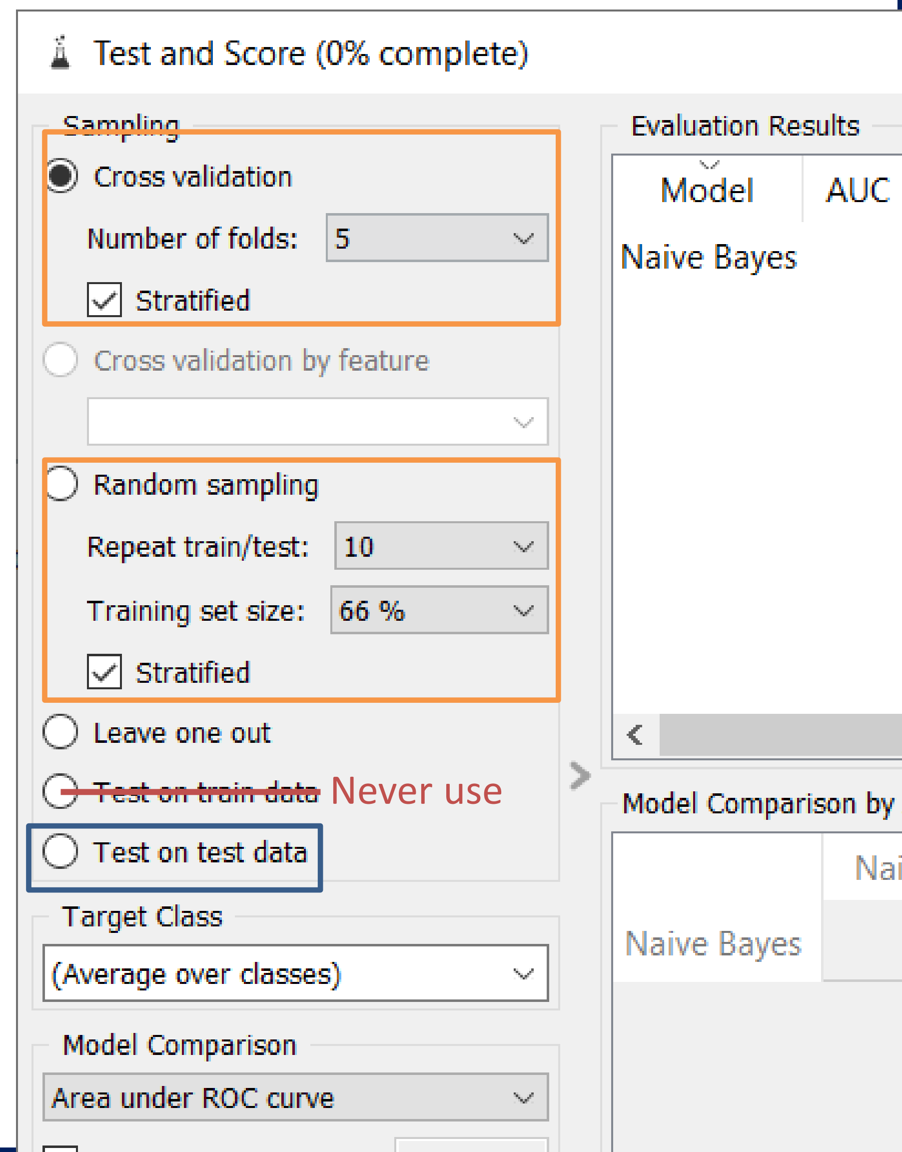
Training error vs. test error

Training error using cross validation

Using random sampling

Test error

Test and Score



Model Complexity and Overfitting

Complex models are more likely to overfit than simple models

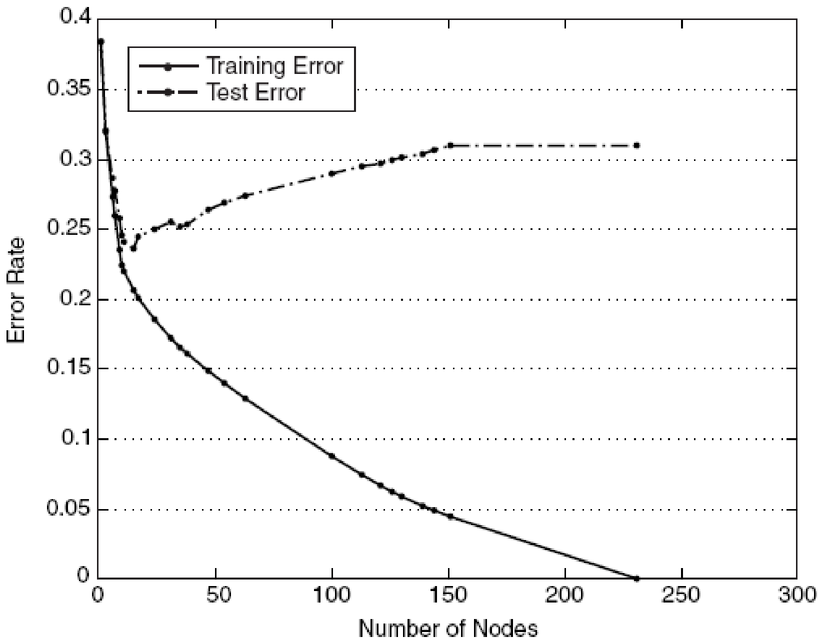
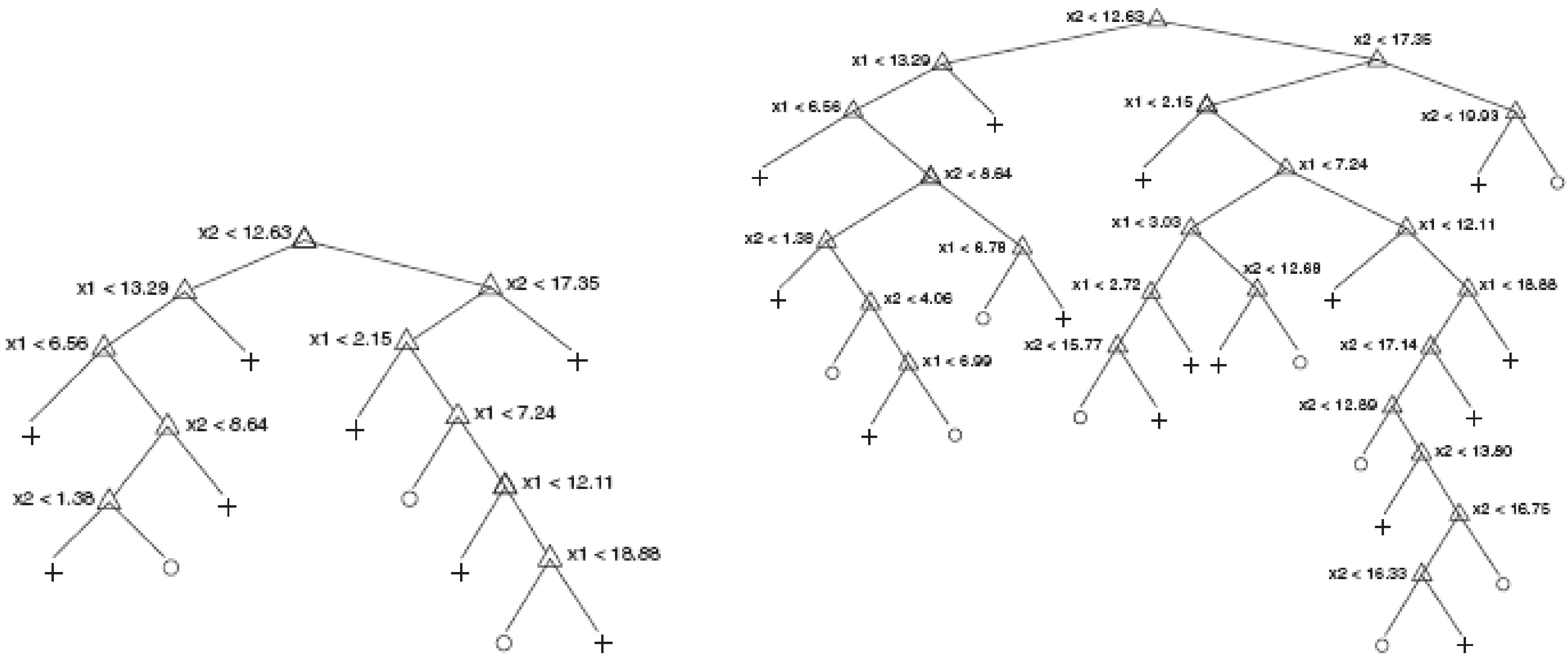


Figure 4.23. Training and test error rates.

For decision tree, #nodes indicates model complexity.

more #nodes ->
higher model complexity ->
lower training error, and higher test error



(a) Decision tree with 11 leaf nodes.

(b) Decision tree with 24 leaf nodes.

Figure 4.24. Decision trees with different model complexities.

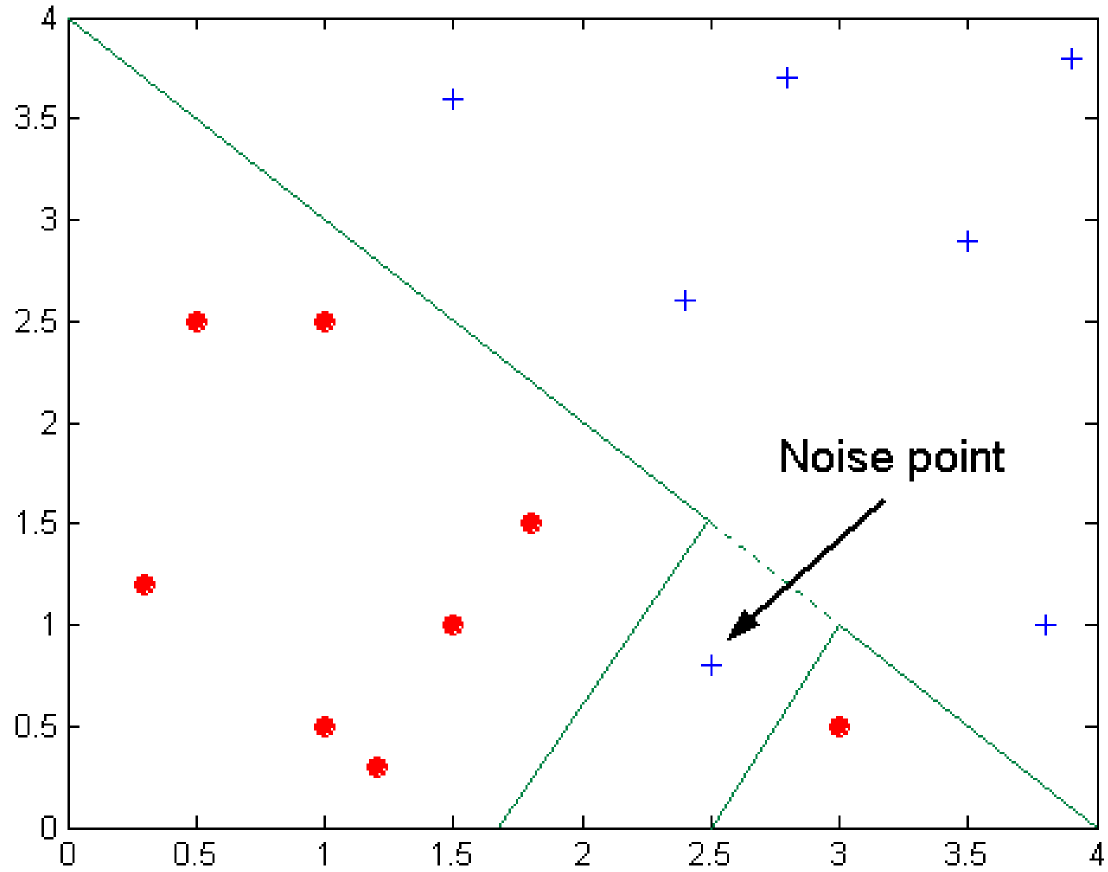
Syracuse University
School of Information Studies

Main reasons for model overfitting

Overfitting due to noise

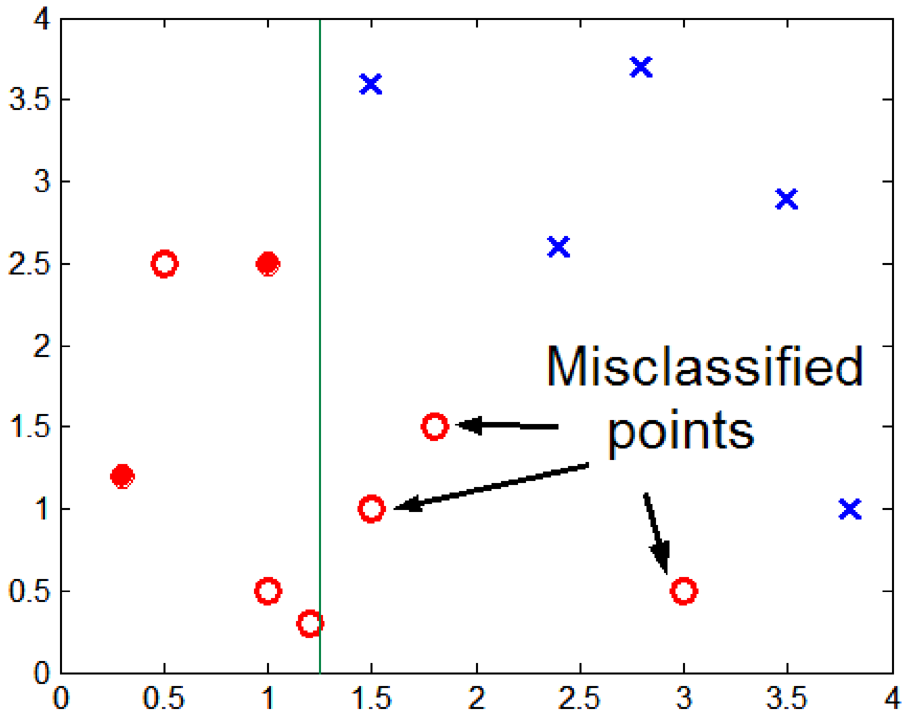
Overfitting due to insufficient samples

Overfitting due to Noise



The decision boundary (supposedly a straight line) is distorted by the noise point. The overfitted decision boundary is the solid blue lines.

Overfitting due to Insufficient Examples



Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels in that region.

Blue crosses and solid red dots are training data.

Red circles are test data.

The green, vertical line is the decision boundary created by a simple decision tree (if x>1.25, label=blue; otherwise, label=read).

Occam's Razor

Given two models of similar generalization errors, the simpler model is preferred over the more complex model

For complex models, there is a greater chance that it was overfit accidentally by errors in data or data imbalance.

Therefore, model complexity should be considered when evaluating a model

Model evaluation methods and metrics

Model evaluation methods

What methods can measure model fitness before using it in real predictions?

Some evaluation methods have been designed to test the model on training data while controlling model overfitting.

- Hold-out test
- Cross validation

Hold-out test

Hold-out test

- split the training data to two subsets, using one subset for training, and the other for testing.
- The splitting ratio is determined by the training set size in that both subsets cannot be too small.
- 50/50 or 2:1 are common splitting ratios.

Advantage

- Fast

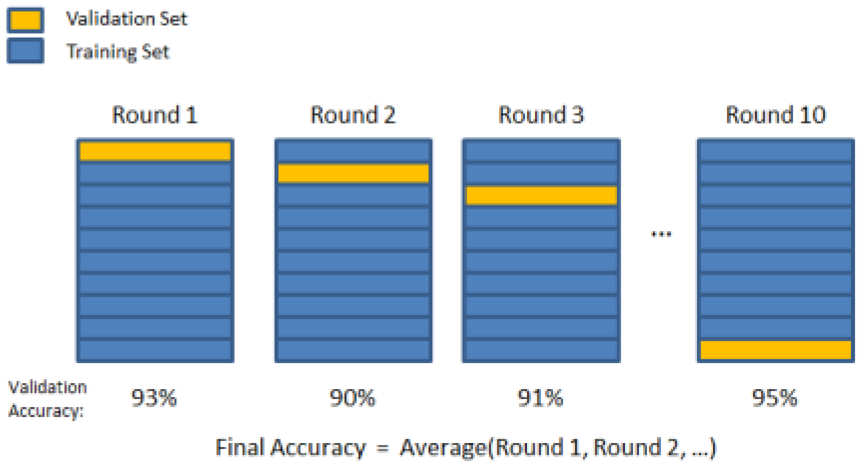
Shortcoming

- When the split changes, the test result changes too. High variability in the test result.

Cross Validation (CV)

N-fold Cross validation (CV)

- N is determined by the training set size. The larger the N, the longer it takes to run the experiment.
- 5 or 10 are common choices for N.



https://scikit-learn.org/stable/modules/cross_validation.html

Leave One Out

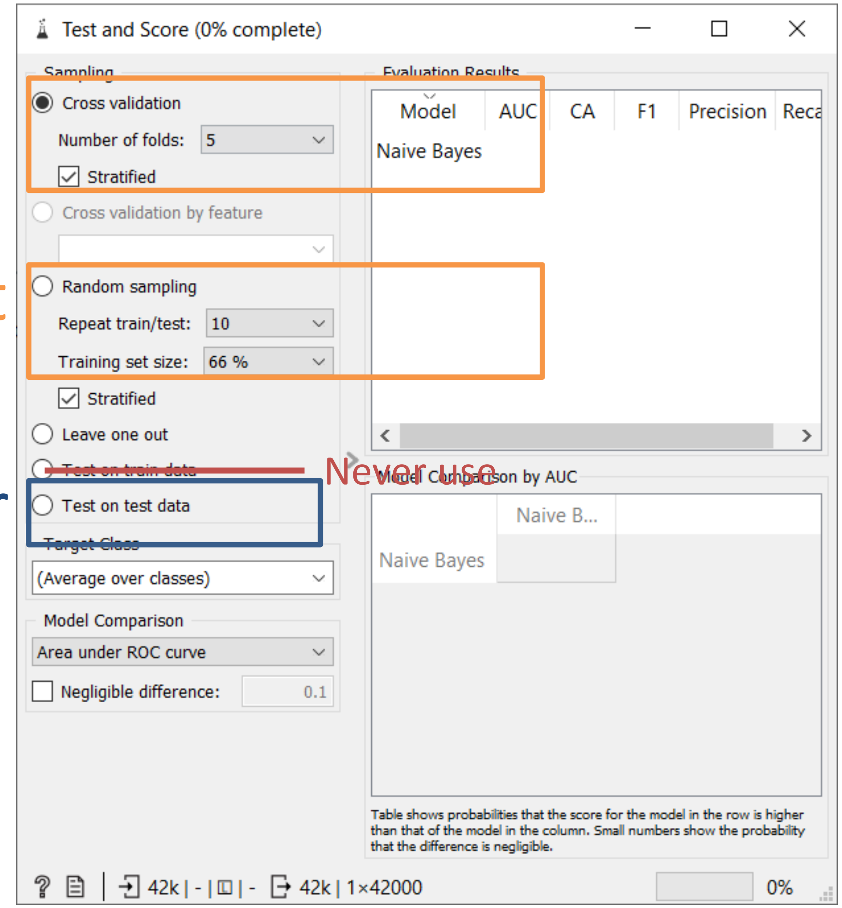
- An extreme case of cross validation
  - N equals the training set size S.
- Advantage
  - No variability in the test result (always get the same result)
- Problems
  - The most time-consuming method
  - Usually used on very small data sets

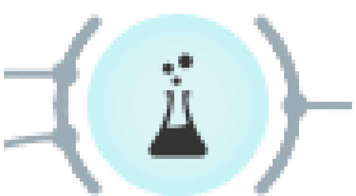
Hold-out Test vs. Cross Validation

Cross validation

Hold-out test

Test error





Test and Score

Hold-out Test vs. Cross Validation

Hold-out test

- Pros: fast
- Cons: high variability in the result depending on the split

Cross validation

- Pros: less variability and thus more reliable error estimation
- Cons: takes longer time

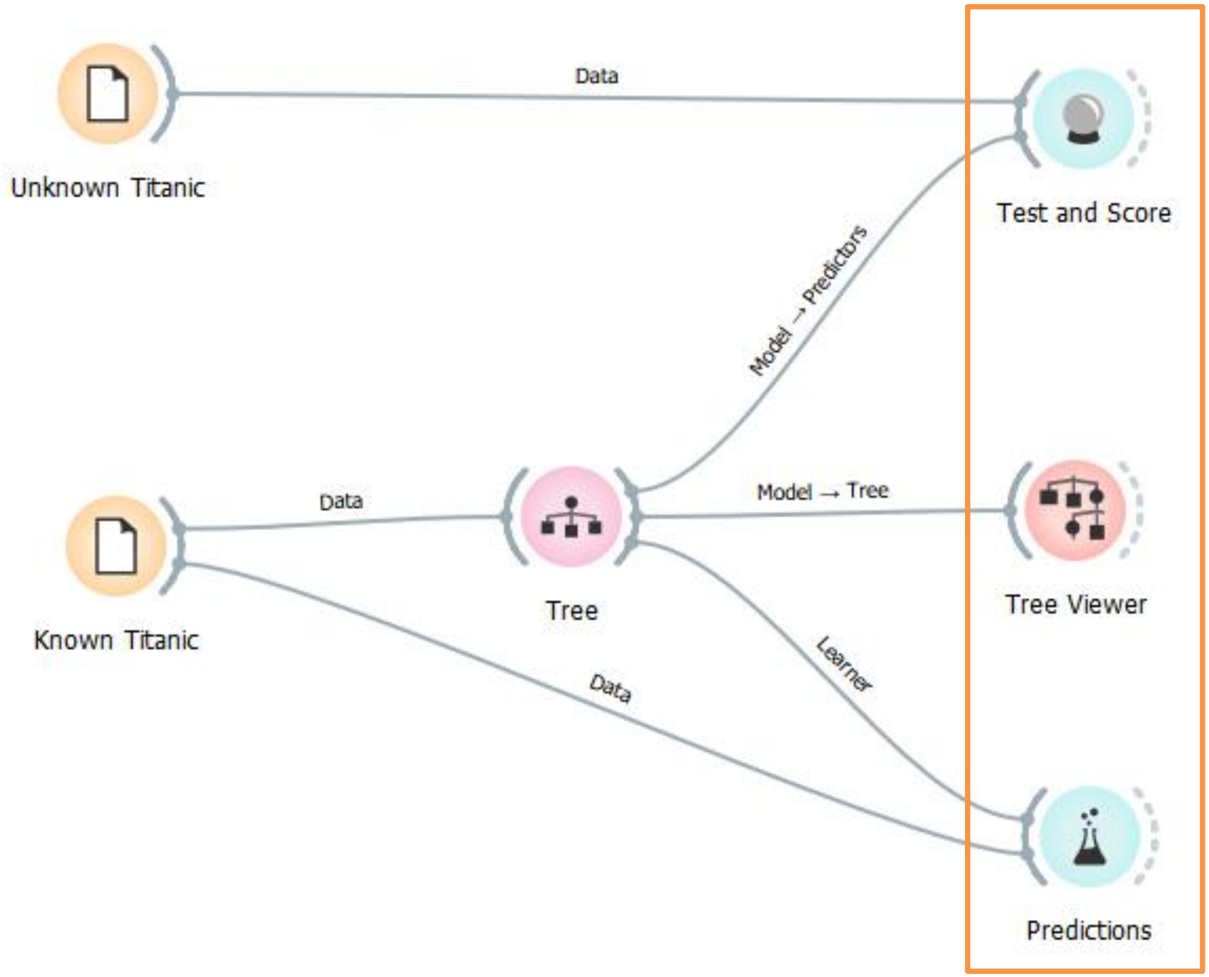
Which model evaluation methods to choose?

CV is the standard method

When data set is huge, hold-out test can save time

When data set is small, leave-one-out can be considered

Model Evaluation in Orange — Decision Trees

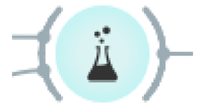


Evaluate model

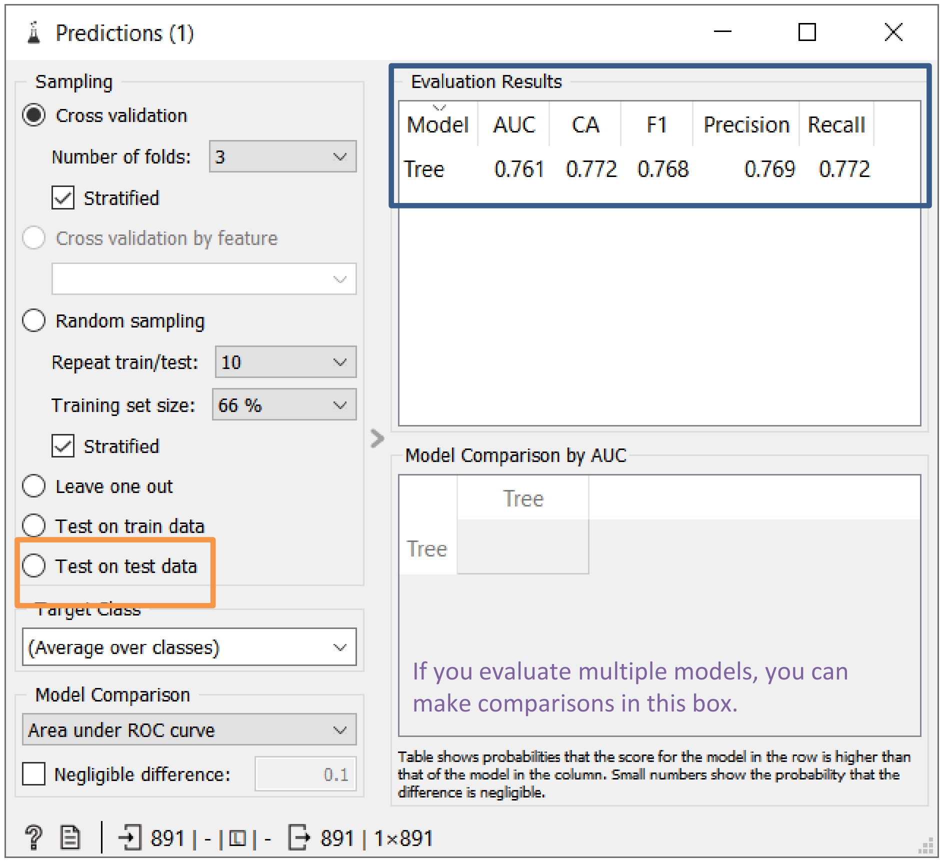
and make

predictions

Evaluation using



Input: Data and Model(s) (decision tree)



This option requires both the test *and* training data.

The test data *must include ground truth* (supervised learning).

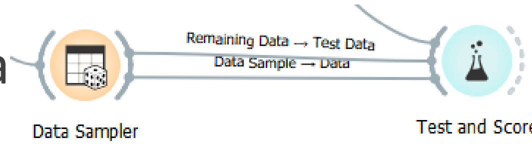
Evaluation metrics (classification):

- **A**rea **U**nder **C**urve
- **C**lassification **A**ccuracy
- **F1**-measure
- **Precision**
- **Recall**

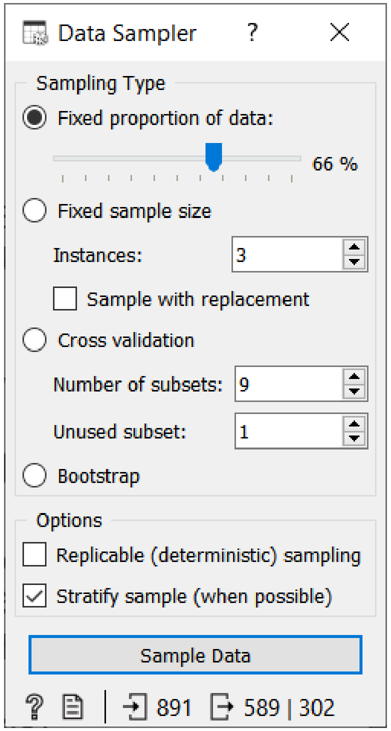
Broadly, the larger these numbers, the better the model.

Exercise: compare variability in CV and hold-out test results

- Build a decision tree model using the Titanic da



- Using the "data sampler" and "test and score" modules *(see images)*, click the "Sample Data" button to conduct a 5-fold CV.
    - Do this five times. Record.
    - *How does the data change as you change the sample?*
- Then do 5 ”random sample" tests with an 80-20 split. Record.
- Now compare the variability by calculating average accuracy and standard deviation
    - hypothesis: hold-out test results have higher variability
    - Does your result support this hypothesis?



Metrics for model performance

Metrics for model performance

Accuracy is the most common measure, but it has limitations, especially on skewed data set.

Data set with similar number of examples in each category is "balanced", otherwise "unbalanced" or "skewed"

Titanic training data set is skewed with more negative examples than positive ones

- 549 "0": did not survive
- 342 "1": survived

Syracuse University School of Information Studies

Problem with accuracy measure

We need to learn some fundamental concepts first:

- Confusion matrix for two classes (can be extended to multiple classes)

	PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Accuracy definition based on confusion matrix

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	A (TP)	B (FN)
	Class=No	C (FP)	D (TN)

Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

Consider a 2-class problem

- Number of Class 0 examples = 9990
- Number of Class 1 examples = 10

If a model predicts every test example as “0”, the model’s accuracy is 9990/10000 = 99.9 %

- Accuracy is misleading because the trivial model does not detect any class 1 example

Two types of error

Market analysis: to predict if a student is going to buy new computer or not.

Prediction result in a confusion matrix:

	predictions		
Ground truth	buy_computer = yes	buy_computer = no	total
buy_computer = yes	6000	1000	7000
buy_computer = no	500	2500	3000
total	6500	3500	10000

False positive: wrong targets

False negative: missed customers

Which type of error matters more?

For a company, one type of error might be more costly than the other.

- Eg. One would rather send out more coupons than missing a potential buyer.
- E.g. one would rather tolerate some junk mail in inbox than risking misclassify a regular mail to junk.

The accuracy measure does not differentiate these two types of errors

Precision and recall measures will.

Precision and recall

Concepts borrowed from the information retrieval field.

Define precision and recall on each category

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	A (TP)	B (FN)
	Class=No	C (FP)	D (TN)

Precision

$$\text{Precision}_{\text{class=yes}} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

Meaning: among all positive predictions, how many are correct?

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	A (TP)	B (FN)
	Class=No	C (FP)	D (TN)

Recall

$$\text{Recall}_{\text{class=yes}} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

Meaning: among all positive examples, how many are correctly predicted?

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	A (TP)	B (FN)
	Class=No	C (FP)	D (TN)

Example: calculate precision and recall

	predictions			
Ground Truth	buy_computer = yes	buy_computer = no	total	recall(%)
buy_computer = yes	6000	1000	7000	
buy_computer = no	500	2500	3000	
total	6500	3500	10000	
Precision (%)				

Example: calculate precision and recall

	predictions			
Ground Truth	buy_computer = yes	buy_computer = no	total	recall(%)
buy_computer = yes	6000	1000	7000	6000/7000
buy_computer = no	500	2500	3000	2500/3000
total	6500	3500	10000	
Precision (%)	6000/6500	2500/3500		

F-measure

An ideal model would achieve high precision and recall on all categories

But in reality precision and recall are like the two sides of a see-saw: if one goes up, the other might go down

F-measure is a weighted average of precision and recall

$$F_{\text{class=yes}} = \frac{2 \times precision \times recall}{precision + recall}$$

Exercise: calculate F-measure

Calculate the F-measures for the following confusion matrix.

	predictions			
Ground Truth	buy_computer = yes	buy_computer = no	total	recall(%)
buy_computer = yes	6000	1000	7000	6000/7000
buy_computer = no	500	2500	3000	2500/3000
total	6500	3500	10000	
Precision (%)	6000/6500	2500/3500		

Baselines for Model Evaluation

If your classification model reached 80% accuracy, is it "good enough"?

Two common baselines for comparison

- **Random guess**: if there are two categories, a model based on random guess would result in 50% accuracy.
- **Majority vote**: if the data set is skewed, a trivial model would assign all test data to the larger category.
  - In the Titanic training data set, the majority vote model would result in 549/891=62% accuracy.

Your model is expected to outperform the common baselines

Majority vote baseline

	predictions			
Ground Truth	buy_computer = yes	buy_computer = no	total	recall(%)
buy_computer = yes	7000	0	7000	1
buy_computer = no	3000	0	3000	0
total	10000	0	10000	
Precision (%)	.70	na		

Fair comparison

When comparing the performance of two models, e.g. an unpruned tree vs. a pruned tree, make sure the comparison is fair, meaning the test data should be exactly the same.

Common mistakes:

- run hold-out test on one model, but cross validation on another model
- Set up different numbers of folds for the two models when using cross validation
- Set up different split ratio for the two models when using hold-out test

Other aspects of evaluation

Speed

- time to construct model (training time)
- time to use the model (classification/prediction time)

Robustness

- handling noise and missing values

Scalability

- the data set size keeps increasing

Interpretability

- understanding the insight provided by the model

Is the model good enough?

There is always room for improvement for non-trivial prediction tasks.

Evaluation from system perspective

Evaluation from user perspective

Exericse: model comparison

Are you satisfied with your email spam filter? Use terms like accuracy, precision and recall to explain the strength and weakness of the email spam filter that you are using. Rank the strength and weakness aspects based on their importance to you.

TRAINING DATA SET SIZE

Training data size affects accuracy

Larger training data set usually helps improve the model, but not always

- Data saturation
- Noise in data

How many is "enough"?

- Depends on many factors, e.g. data availability, cost to obtain data, data quality

Training data size affects accuracy

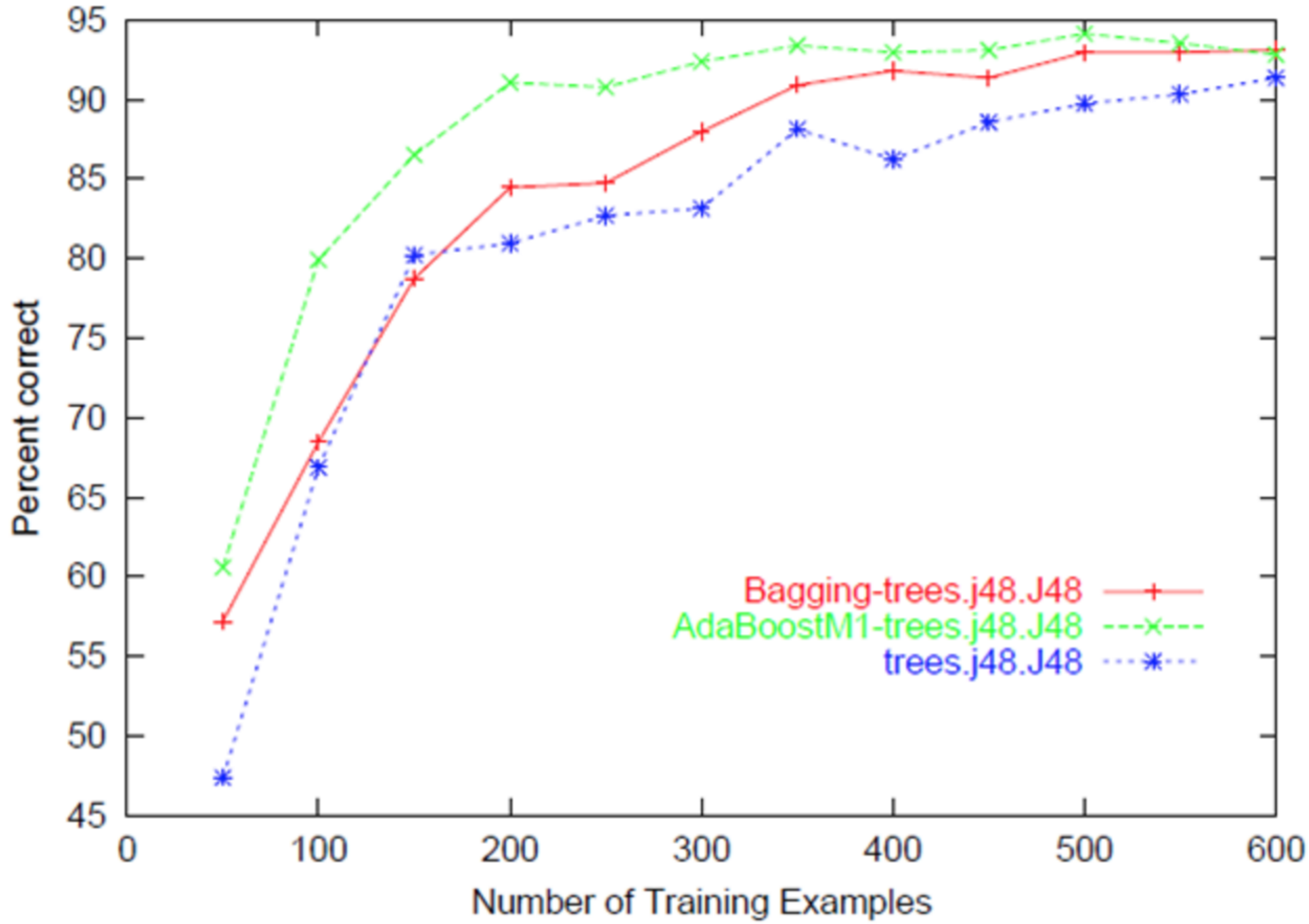
Larger training data set usually helps improve the model, but not always

- Data saturation
- Noise in data

How many is "enough"?

- Depends on many factors, e.g. data availability, cost to obtain data, data quality

Learning curve



http://stackoverflow.com/questions/4617365/what-is-a-learning-curve-in-machine-learning

Exercise: training set size

Titanic: increase the percentage split from 10% to 20%, 30%, 40%, 50%, …, 90%

Does the accuracy increase?

Note this is not a precise learning curve because the test data also changed each round

TRAINING DATA ACQUISITION

Not enough data?

Semi-supervised learning

Active learning

Crowdsourcing

Semi-supervised learning

Utilize the strength of current model

Assume the most confident predictions are highly accurate

Process

- Build model on current training data
- Apply model to test data
- Rank test data by prediction confidence.
- Add the most confident ones into training data

Active Learning

Goal: adding data to reduce current model's weakness

Also rank test data by prediction confidence

Choose the least confident ones

Confirm these predictions with human experts

Add them to training data

Crowdsourcing

Divide and conquer

- ask many people to each label a few examples for you

Amazon Mechanical Turk

How trustworthy is human annotation?

Reliability test

- If asking 2 or more people to mark the sentiment of a collection of tweets, to what extent will they agree with each other?

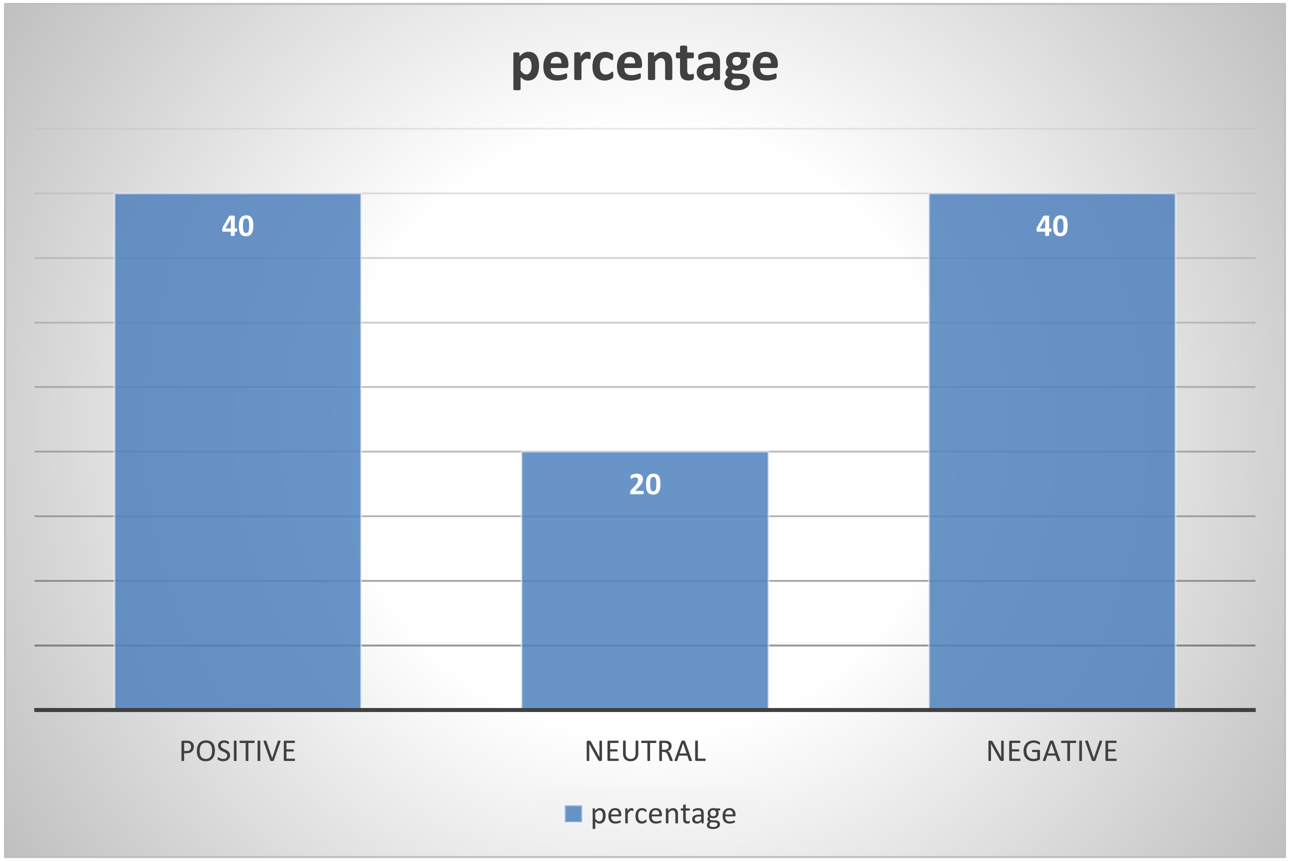
Subjectivity in Classification

Some classification tasks involve certain level of subjectivity in decision.

Whether a tweet is positive or neutral can be subjective decision.

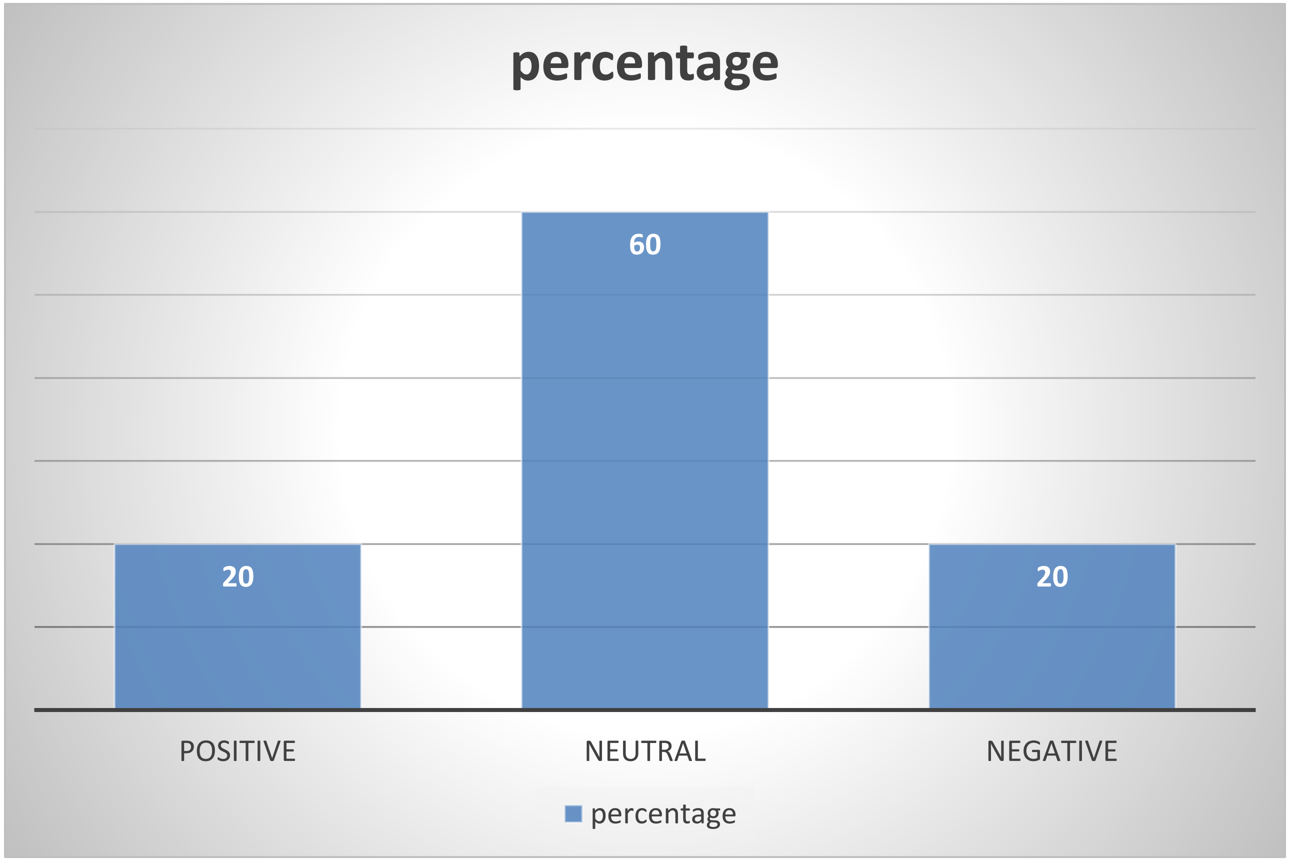
Different people may annotate the same tweet with different labels, e.g. “positive”, “neutral”

A “polarized” coder



Syracuse University
School of Information Studies

A “neutral” coder



Inter-coder agreement

Measures to evaluate the reliability of human annotation

- Percentage of agreement
- Cohen's Kappa

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

Po: observed agreement
Pe: chance of agreement

Inter-coder Agreement

Raw agreement:

- a=count(agreed_items)/total_items

Problem with raw agreement:

- Skewed categories: 90% raw agreement in both tables

	Coder A	
Coder B	Positive	negative
positive	45	5
negative	5	45

	Coder A	
Coder B	Positive	negative
positive	90	10
negative	0	0

Cohen's kappa

a=raw_agreement

c=chance_agreement

K=(a-c)/(1-c)

	Coder A		
Coder B		Positive	negative
	positive	45	5
	negative	5	45

	Coder A		
Coder B		Positive	negative
	positive	90	10
	negative	0	0

Cohen's kappa

a=raw_agreement

c=chance_agreement

$K=(a-c)/(1-c)$

K=0.8

	Coder A		
Coder B		Positive	negative
	positive	45	5
	negative	5	45

K=0

	Coder A		
Coder B		Positive	negative
	positive	90	10
	negative	0	0

How to calculate kappa?

Given a confusion matrix of two coders

		Coder A	
Coder B		Positive	negative
	positive	45	5
	negative	5	45

How to calculate kappa?

Calculate marginal distribution

	Coder A			
Coder B		Positive	negative	
	positive	45	5	50%
	negative	5	45	50%
		50%	50%	

How to calculate kappa?

Calculate raw agreement (a=0.9)

Calculate

- P(both A and B gives “positive” label) = 0.25
- P(both A and B gives “negative” label) = 0.25
- Chance_agreement: c=0.25+0.25=0.5
- Kappa=(a-c)/(1-c)=(0.9-0.5)/(1-0.5)=0.4/0.5=0.8

Tools to calculate kappa

Online tool

- http://vassarstats.net/kappa.html

Exercise: calculate kappa agreement?

	Coder A	
Coder B		
	Positive	negative
positive	89	9
negative	1	1

Reproducible research

Reproducible research is a cornerstone of scientific research

Report your data mining approach and results in a reproducible way

Use tools like RMD to document the process

If possible, open data access

CLASS PROJECT OVERVIEW