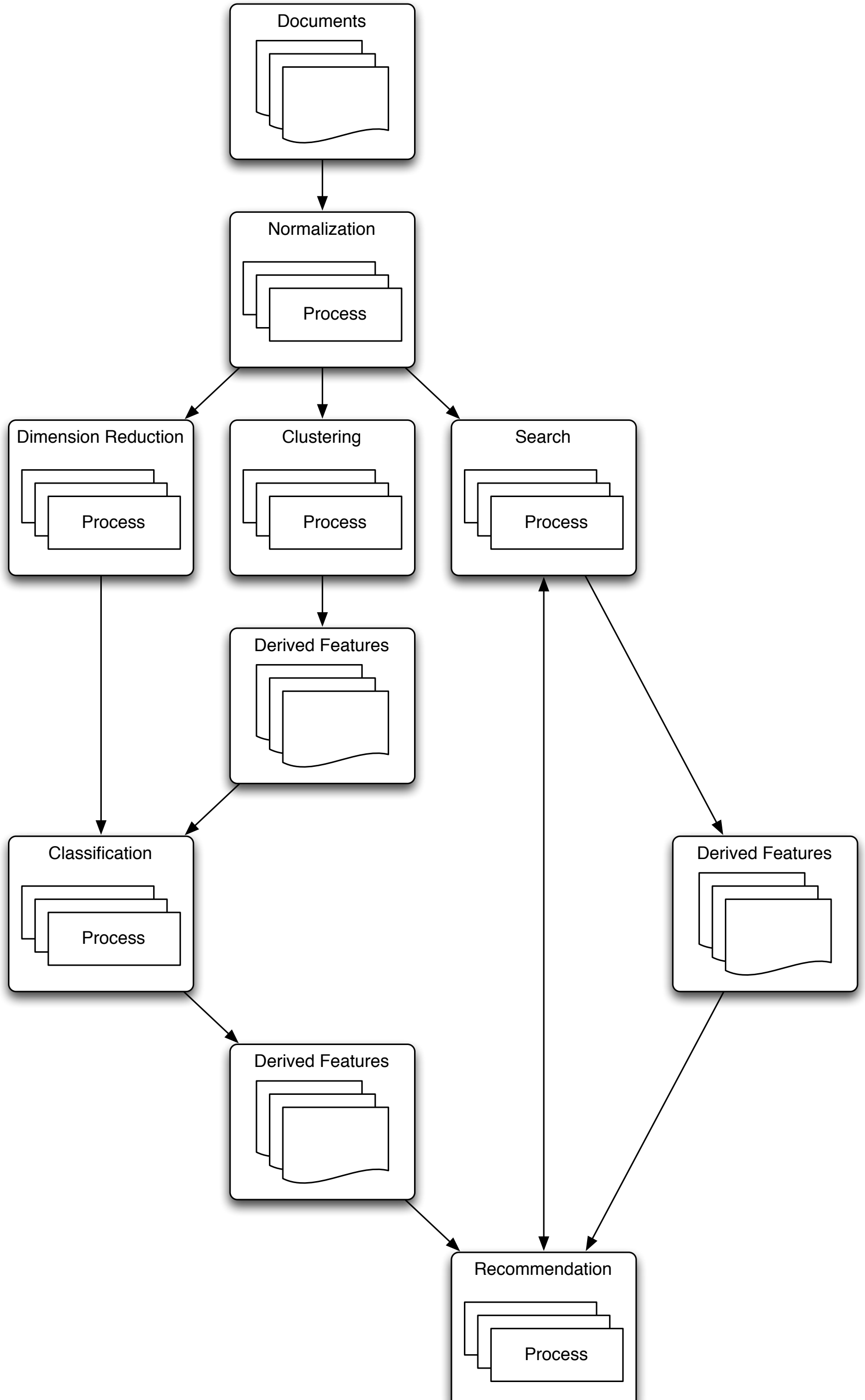
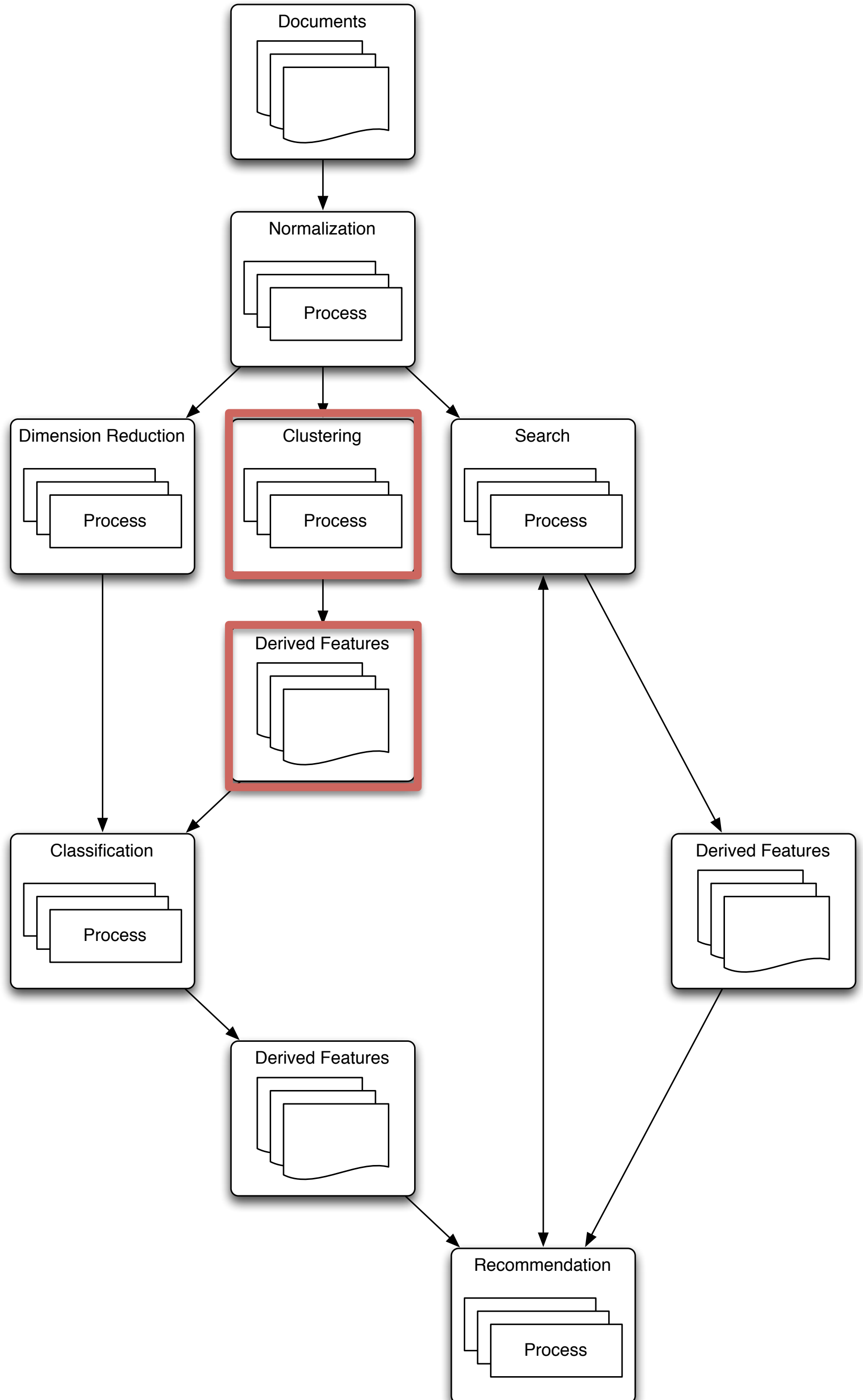
Mahout Workshop, Section 4

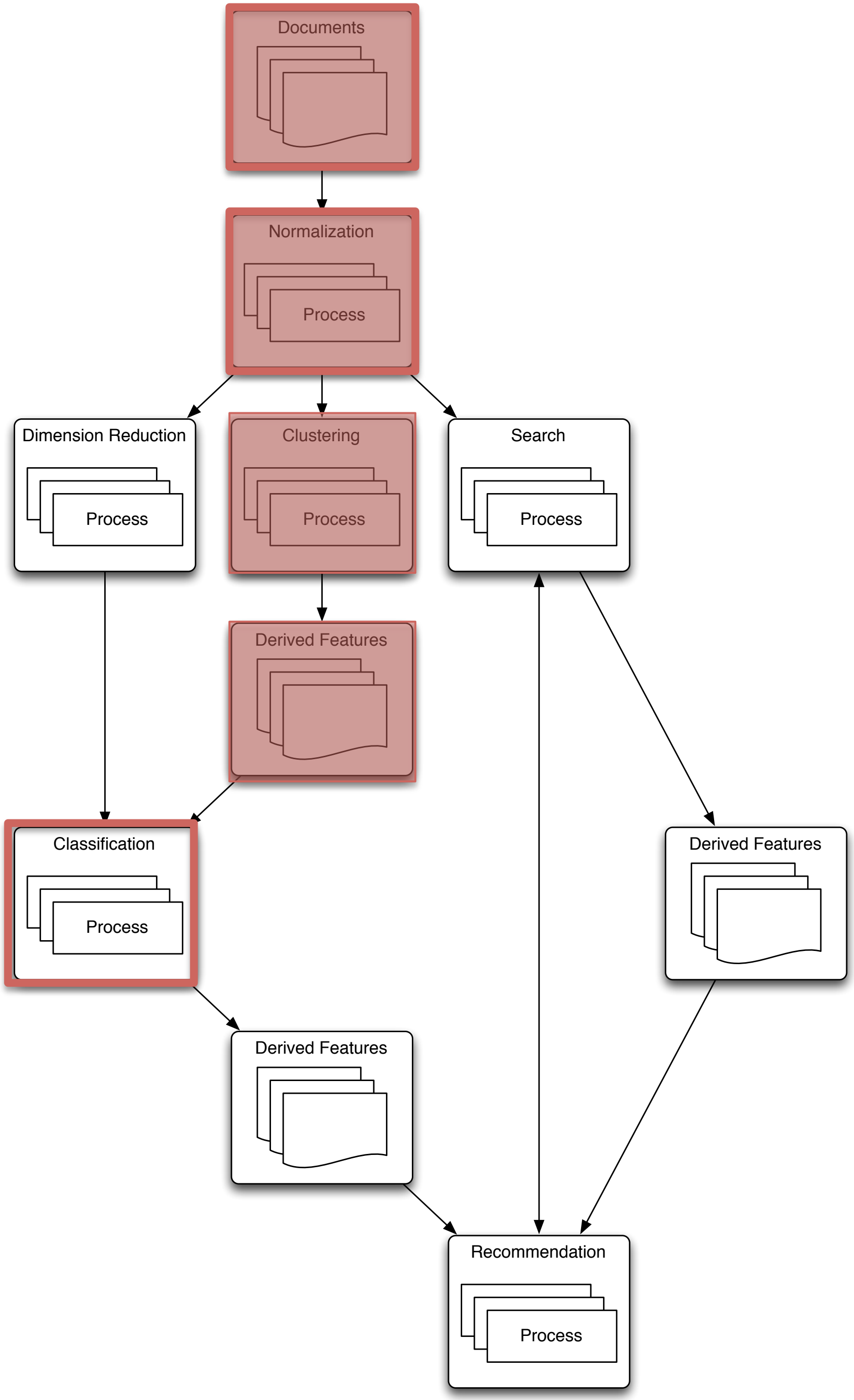
Allen Day, PhD
MapR Technologies


Documents
Normalization
Process
Dimension Reduction
Process
Clustering
Process
Search
Process
Derived Features
Classification
Process
Derived Features
Derived Features
Recommendation
Process

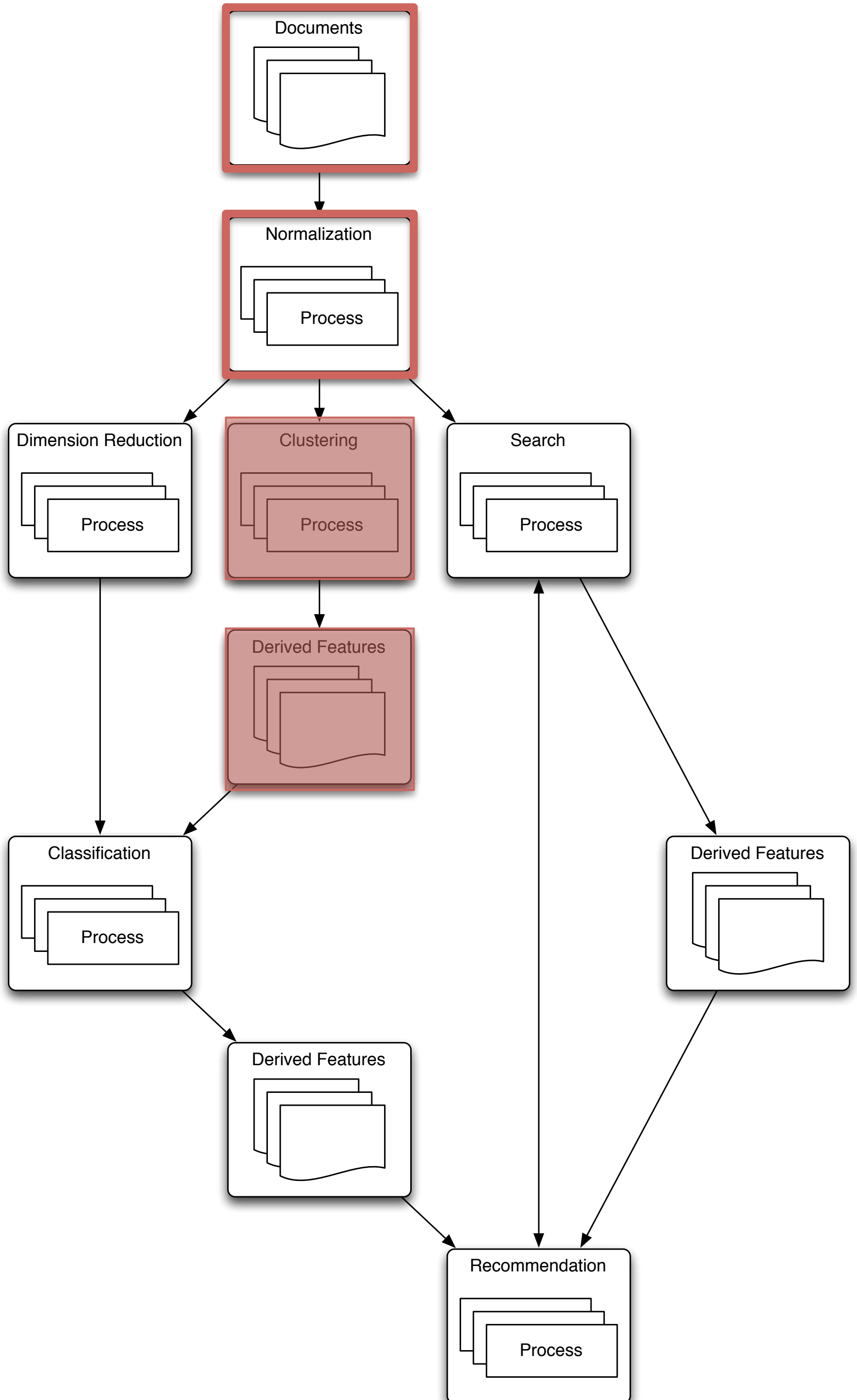
Section 3 (this one)



Section 4 (next one)

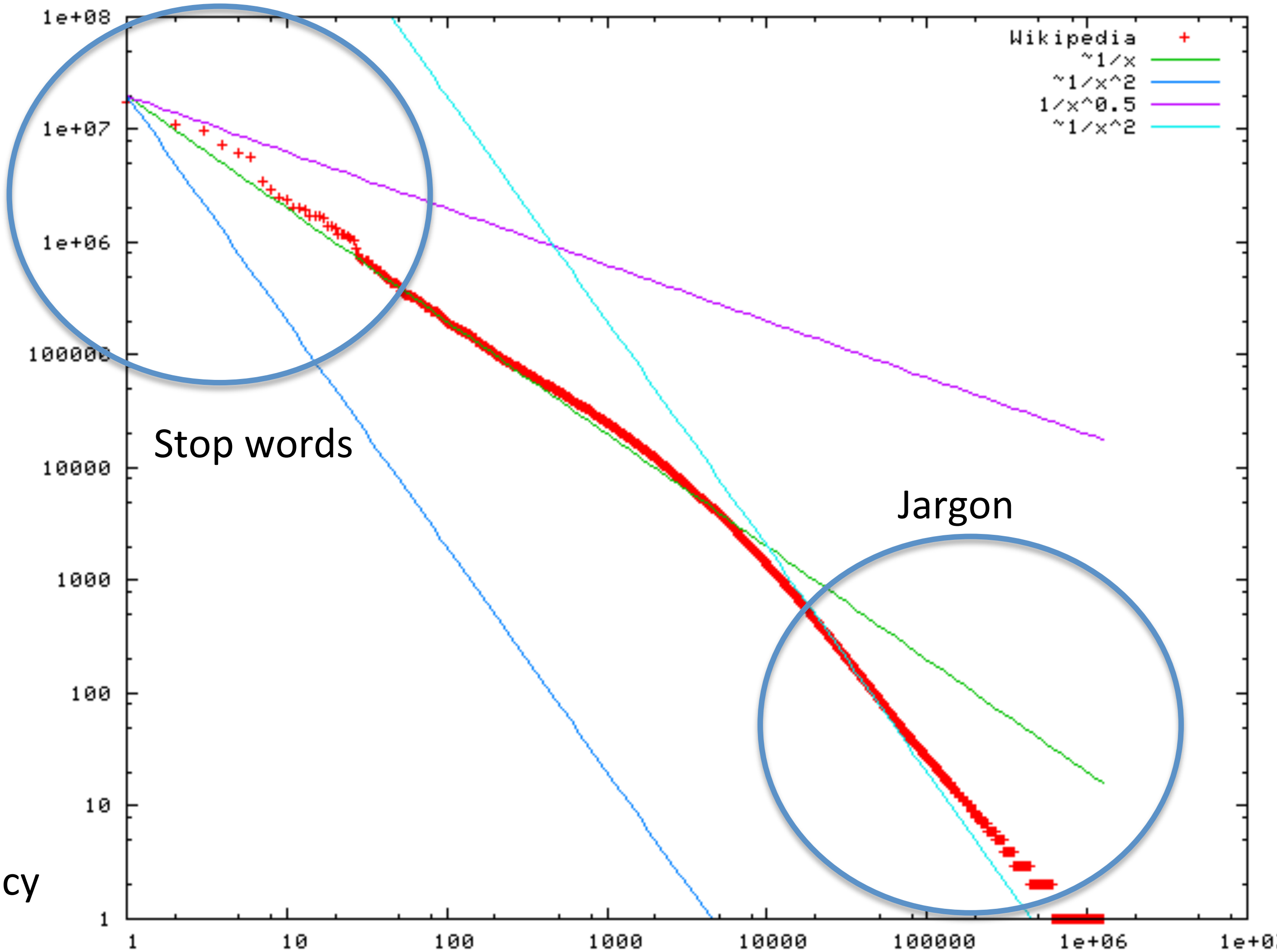


Section 4 (next one)



TF-IDF

Observation: Unequal Word Distributions



Frequency

Rank

http://en.wikipedia.org/wiki/Zipf's_law

Motivation: “Undo” Effect of Unequal Distribution

	doc1	doc2	doc3	TF (max)	DF(bool)
car	27	4	24	27	3
auto	3	33	0	33	2
insurance	0	33	29	33	2
best	14	0	17	17	2

	doc1	doc2	doc3	TF (max)	DF (bool)
car	27/27 * log(3/3)	4/27 * log(3/3)	24/27 * log(3/3)	27	3
auto	3/33 * log(3/2)	33/33 * log(3/2)	0/33 * log(3/2)	33	2
insurance	0/33 * log(3/2)	33/33 * log(3/2)	29/33 * log(3/2)	33	2
best	14/17 * log(3/2)	0/17 * log(3/2)	17/17 * log(3/2)	17	2

MAX

Motivation:
"Undo" Effect of Unequal Distribution

	doc1	doc2	doc3	TF (max)	DF (bool)
car	27/27 * log(3/3)	4/27 * log(3/3)	24/27 * log(3/3)	27	3
auto	3/33 * log(2/3)	33/33 * log(2/3)	0/33 * log(2/3)	33	2
insurance	0/33 * log(2/3)	33/33 * log(2/3)	29/33 * log(2/3)	33	2
best	14/17 * log(2/3)	0/17 * log(2/3)	17/17 * log(2/3)	17	2

	doc1	doc2	doc3	TF (max)	DF (bool)
car	0	0	0	27	3
auto	0.04	0.40	0	33	2
insurance	0	0.40	0.35	33	2
best	0.14	0	0.40	17	2

Note that I've used TF_{max} and DF_{bool} here. Other tf*idf variants are also valid.

Motivation: “Undo” Effect of Unequal Distribution

	doc1	doc2	doc3	TF (sum)	DF (bool)
car	27/55 * log(3/3)	4/55 * log(3/3)	24/55 * log(3/3)	55	3
auto	3/36 * log(3/2)	33/36 * log(3/2)	0/36 * log(3/2)	36	2
insurance	0/62 * log(3/2)	33/62 * log(3/2)	29/62 * log(3/2)	62	2
best	14/31 * log(3/2)	0/31 * log(3/2)	17/31 * log(3/2)	31	2

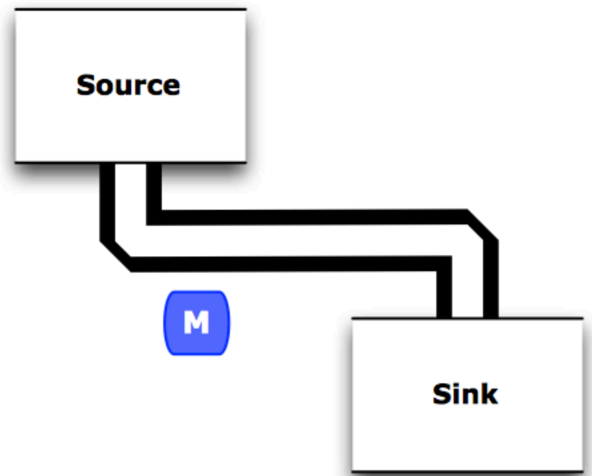
	doc1	doc2	doc3	TF (sum)	DF (bool)
car	0	0	0	27	3
auto	0.03	0.37	0	33	2
insurance	0	0.21	0.19	33	2
best	0.18	0	0.22	17	2

Note that I’ve used TF_{max} and DF_{bool} here. Other tf*idf variants are also valid.

TF-IDF IN CASCADING (ON HADOOP)

http://www.slideshare.net/pacoid/intro-to-data-science-for-enterprise-big-data

1: copy

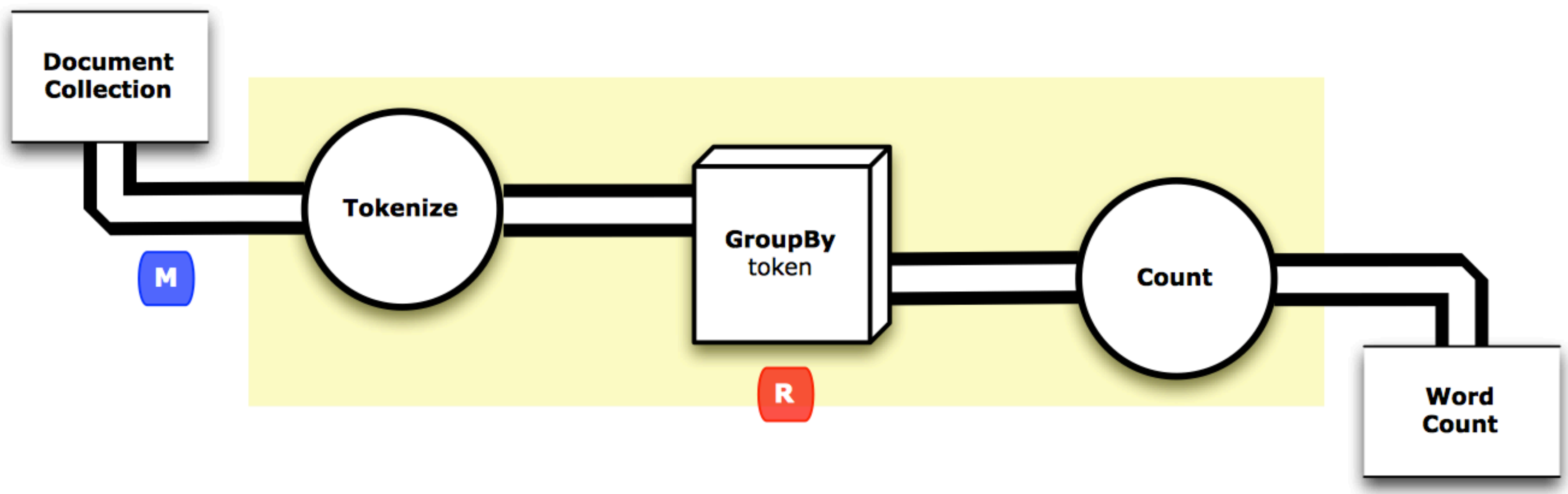



```
public class
  Main
  {
  public static void
  main( String[] args )
    {
    String inPath = args[ 0 ];
    String outPath = args[ 1 ];

    Properties props = new Properties();
    AppProps.setApplicationJarClass( props, Main.class );
    HadoopFlowConnector flowConnector = new HadoopFlowConnector( props
);

    // create the source tap
    Tap inTap = new Hfs( new TextDelimited( true, "\t" ), inPath );

    // create the sink tap
    Tap outTap = new Hfs( new TextDelimited( true, "\t" ), outPath );

    // specify a pipe to connect the taps
    Pipe copyPipe = new Pipe( "copy" );

    // connect the taps, pipes, etc., into a flow
    FlowDef flowDef = FlowDef.flowDef().setName( "copy" )
     .addSource( copyPipe, inTap )
     .addTailSink( copyPipe, outTap );

    // run the flow
    flowConnector.connect( flowDef ).complete();
    }
  }
```

1 mapper
0 reducers
10 lines code

http://www.slideshare.net/pacoid/intro-to-data-science-for-enterprise-big-data

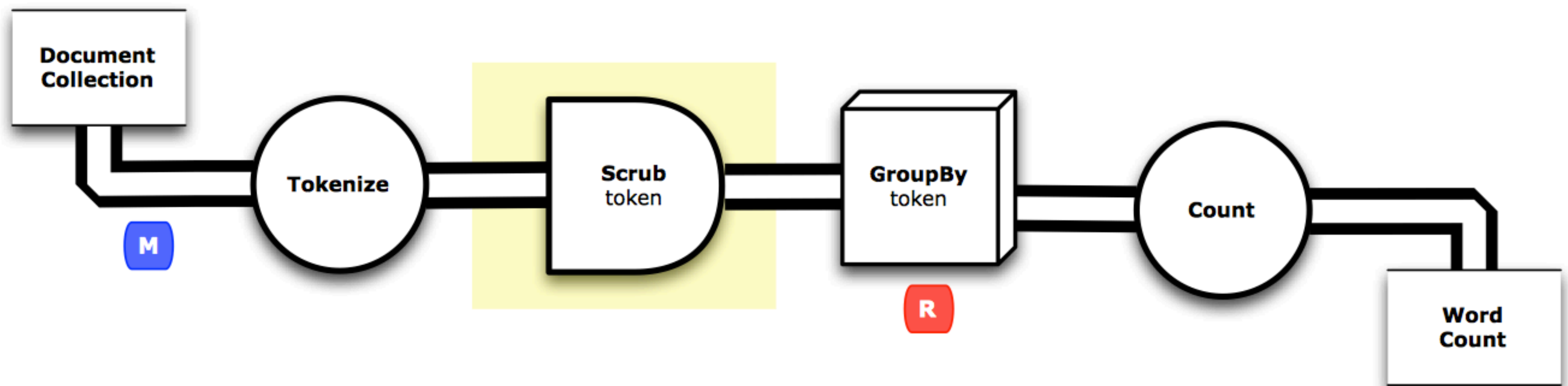
2: word count



1 mapper
1 reducer
18 lines code

http://www.slideshare.net/pacoid/intro-to-data-science-for-enterprise-big-data

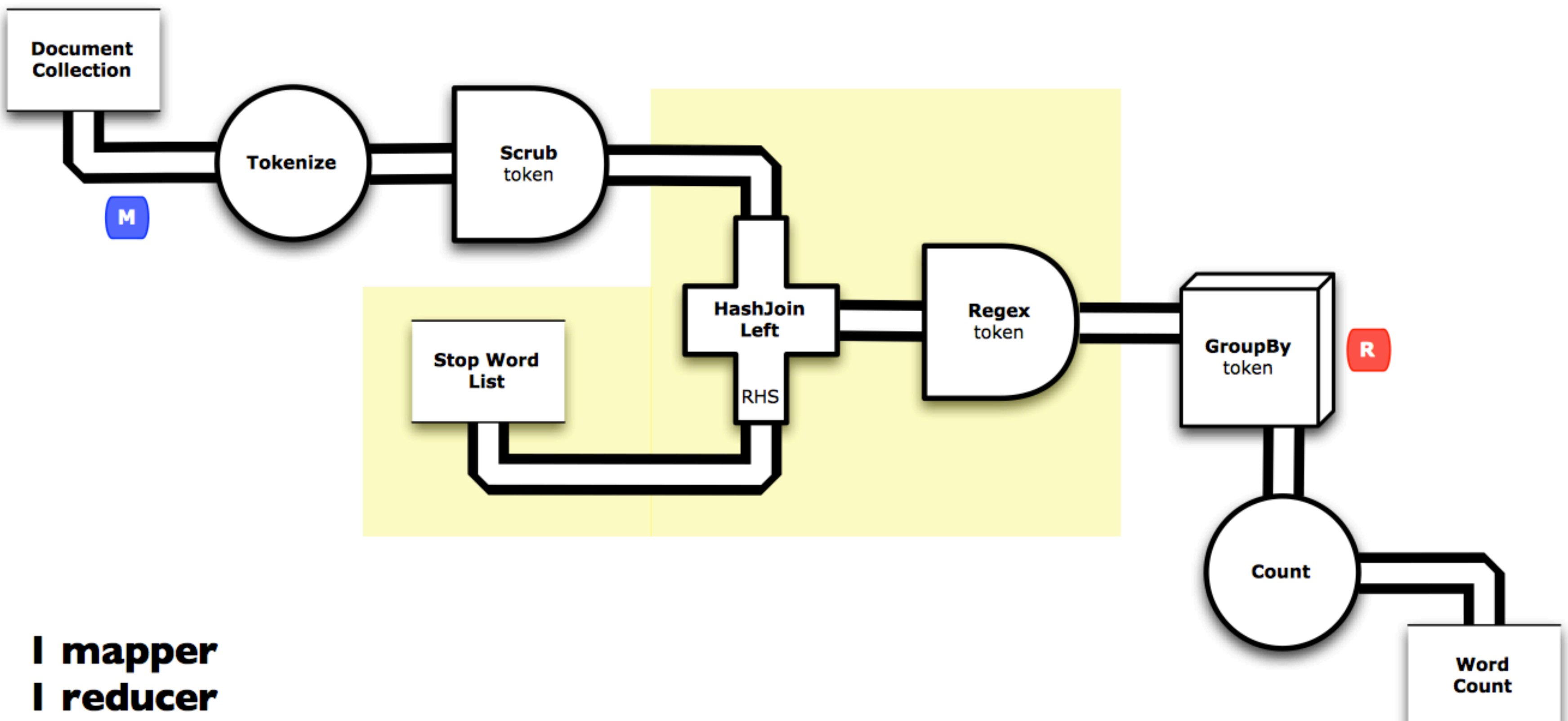
3: wc + scrub



1 mapper
1 reducer
22+10 lines code

http://www.slideshare.net/pacoid/intro-to-data-science-for-enterprise-big-data

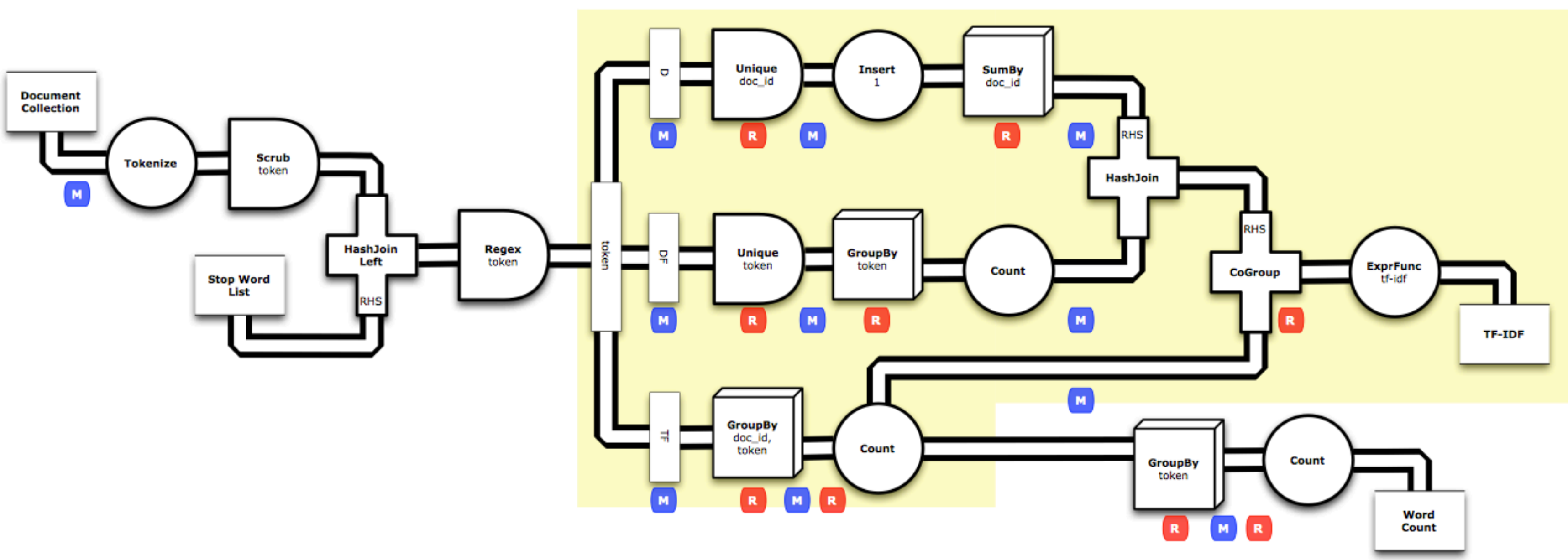
4: wc + scrub + stop words



1 mapper
1 reducer
28+10 lines code

http://www.slideshare.net/pacoid/intro-to-data-science-for-enterprise-big-data

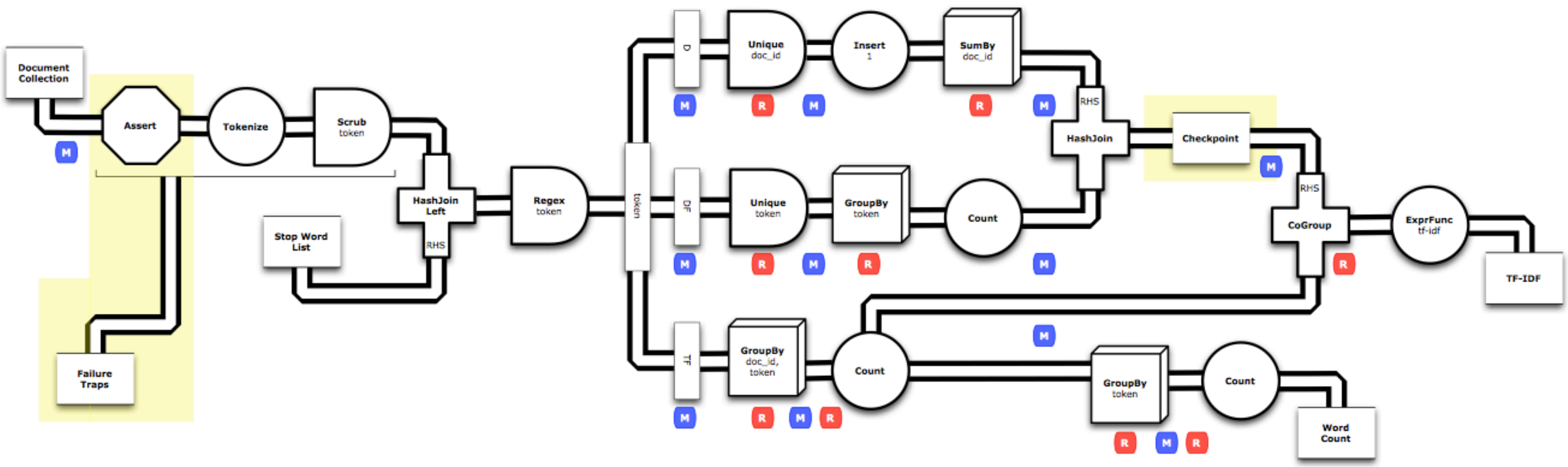
5: tf-idf



11 mappers
9 reducers
65+10 lines code

http://www.slideshare.net/pacoid/intro-to-data-science-for-enterprise-big-data

6: tf-idf + tdd



12 mappers
9 reducers
76+14 lines code

CLASSIFICATION INTRO

What is it?

- *D*-dimensional observations *X* (x_1, …, x_n)
- 2+ known classes *K* on some subset of *X*
- Find function f(x) to assign each *X* to a class *K*
  - Make it perform well
    - Precision vs Recall. Resource and time complexity

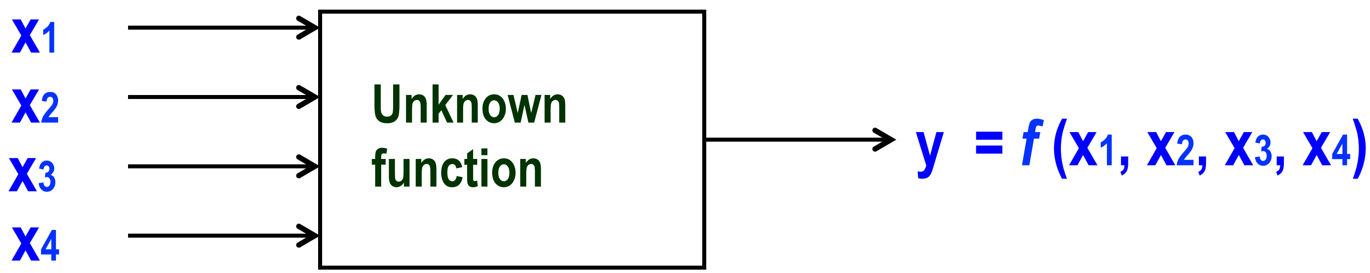
What is it?

- *D*-dimensional observations *X* (x_1, …, x_n)
- 2+ known classes *K* on some subset of *X*
- Find function f(x) to assign each *X* to a class *K*
  – Make it perform well
    - Precision vs Recall. Resource and time complexity

Applications

- Security – Does an observation fall outside of normal behavior boundaries?
- Marketing – Can a user's gender be inferred by browsing behavior? Search terms?
- Search – Does the user's query pattern imply she wants documents of a particular class?
- Health – Emergency Room triage. Does this patient need to be seen immediately?

Learning is impossible, unless…



Given:

Training examples $(x, f(x))$ of unknown function f

Find:

A good approximation to f

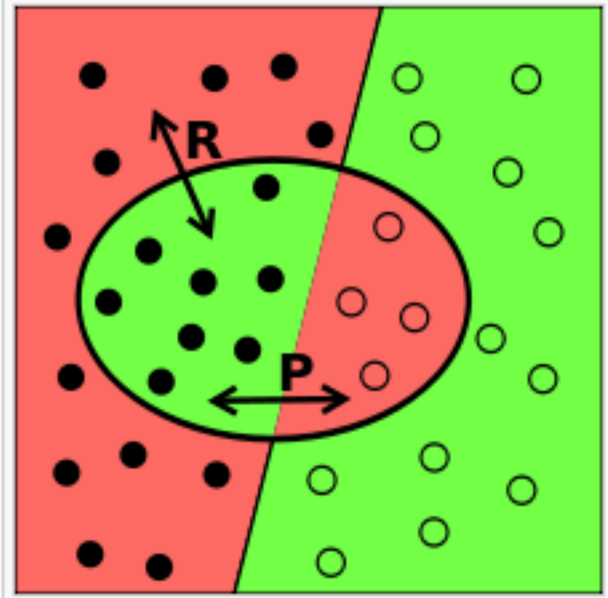
Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Precision vs. Recall
aka Sensitivity vs. Specificity

	actual class (observation)	
predicted class (expectation)	**tp** (true positive) Correct result	**fp** (false positive) Unexpected result
	fn (false negative) Missing result	**tn** (true negative) Correct absence of result

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

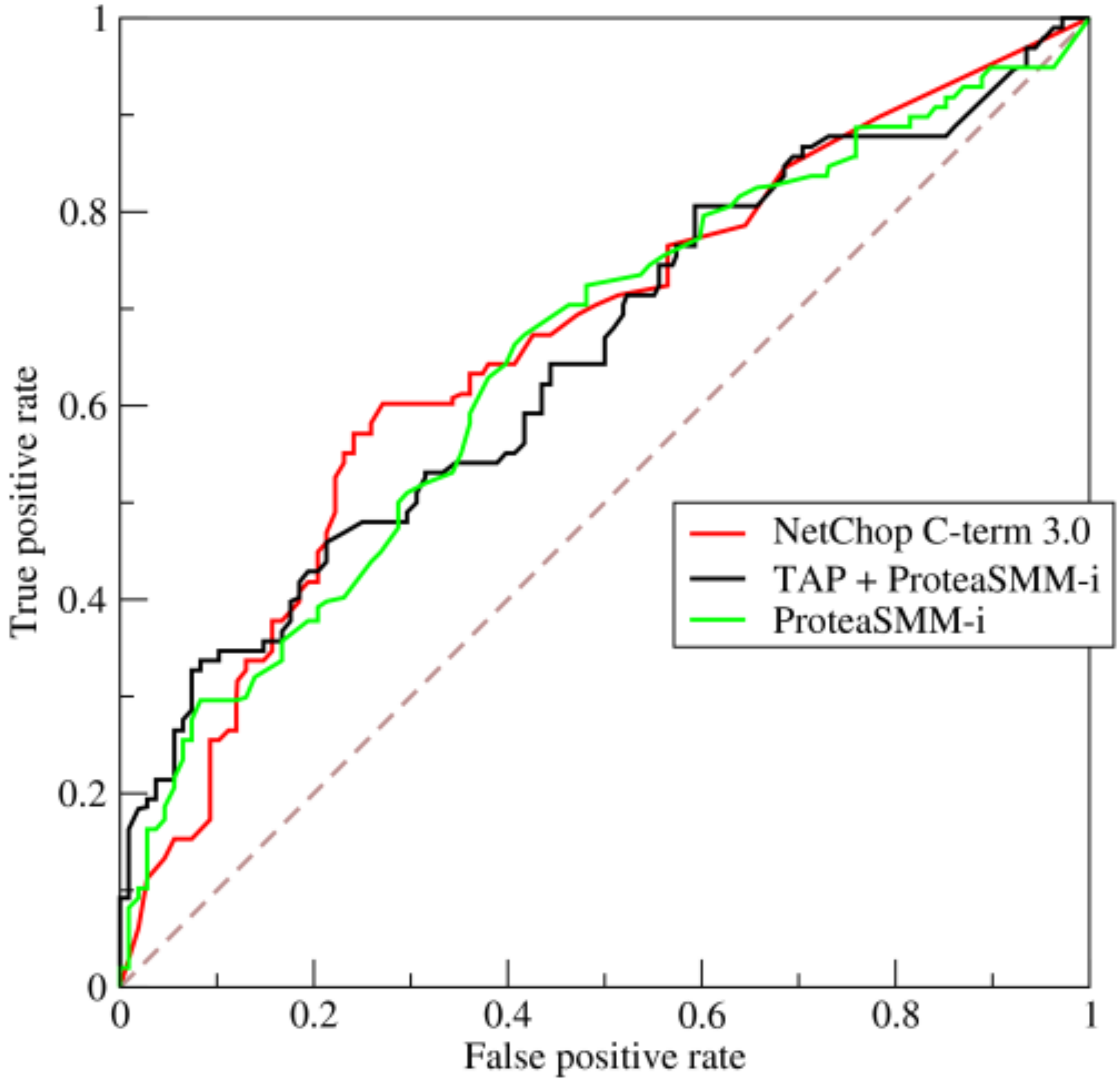


In this figure the relevant items are to the left of the straight line while the retrieved items are within the oval. The red regions represent errors. On the left these are the relevant items not retrieved (false negatives), while on the right they are the retrieved items that are not relevant (false positives).

http://en.wikipedia.org/wiki/Precision_and_recall

- *D*-dimensional observations *X* (x_1, …, x_n)
- 2+ known classes *K* on some subset of *X*
- Find function f(x) to assign each *X* to a class *K*
  - Make it perform well
    - Precision vs Recall. Resource and time complexity

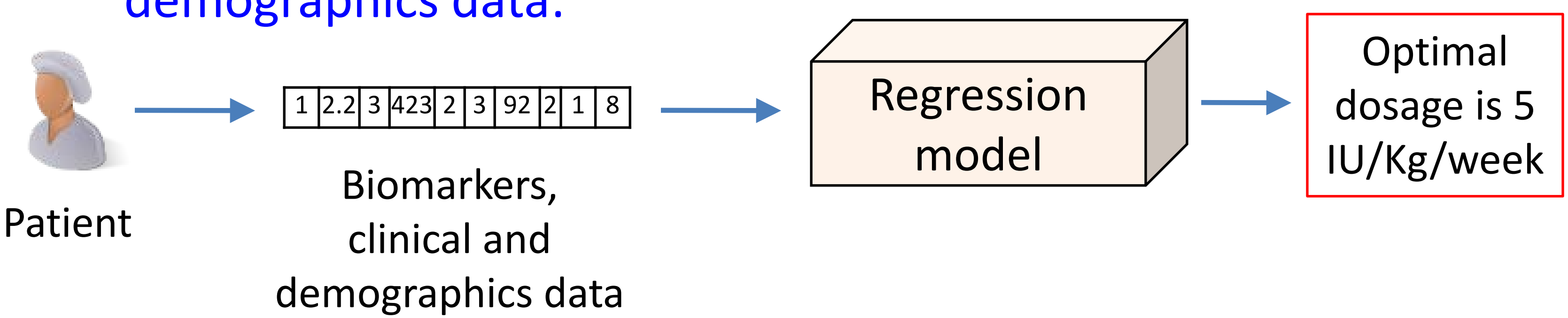
ROC Curve



http://en.wikipedia.org/wiki/Receiver_operating_characteristic

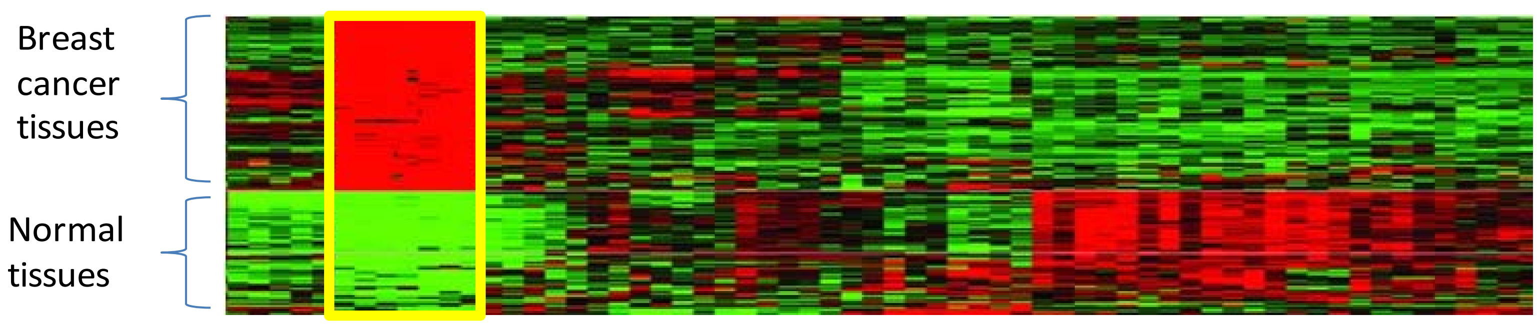
Data-analysis problems of interest

2. Build computational regression models to predict values of some continuous response variable or outcome.
   - Regression models can be used to predict survival, length of stay in the hospital, laboratory test values, etc.
   - E.g., build a decision-support system to predict optimal dosage of the drug to be administered to the patient. This dosage is determined by the values of patient biomarkers, and clinical and demographics data:



Data-analysis problems of interest

3. Out of all measured variables in the dataset, select the smallest subset of variables that is necessary for the most accurate prediction (classification or regression) of some variable of interest (e.g., phenotypic response variable).
   - E.g., find the most compact panel of breast cancer biomarkers from microarray gene expression data for 20,000 genes:

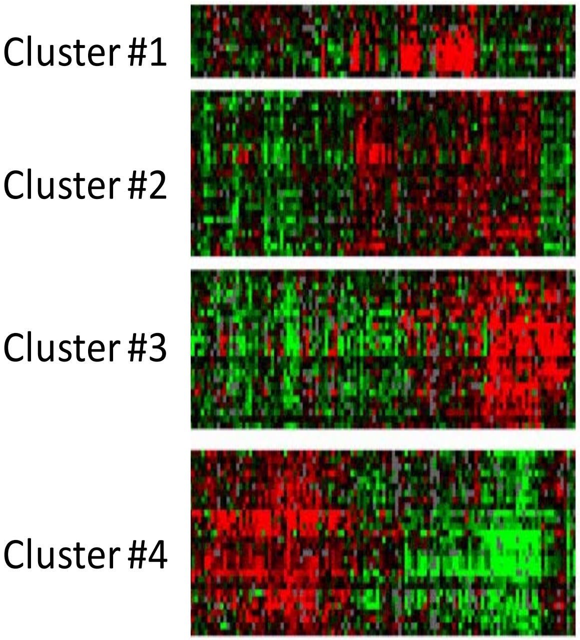


Data-analysis problems of interest

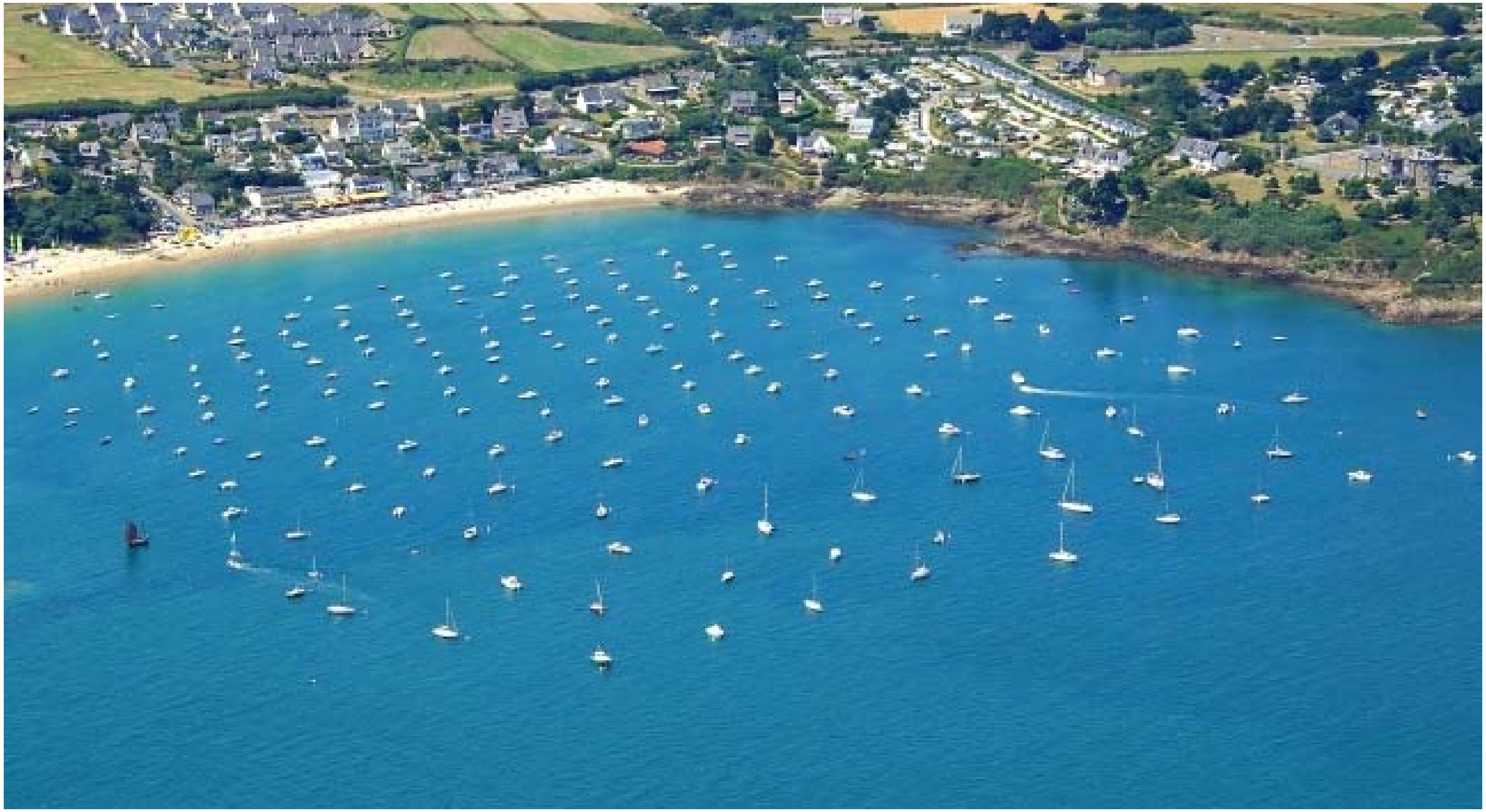
4. Build a computational model to identify novel or outlier patients/samples.
   - Such models can be used to discover deviations in sample handling protocol when doing quality control of assays, etc.
   - E.g., build a decision-support system to identify aliens.

Data-analysis problems of interest

5. Group patients/samples into several clusters based on their similarity.
   - These methods can be used to discovery disease sub-types and for other tasks.
   - E.g., consider clustering of brain tumor patients into 4 clusters based on their gene expression profiles. All patients have the same pathological sub-type of the disease, and clustering discovers new disease subtypes that happen to have different characteristics in terms of patient survival and time to recurrence after treatment.

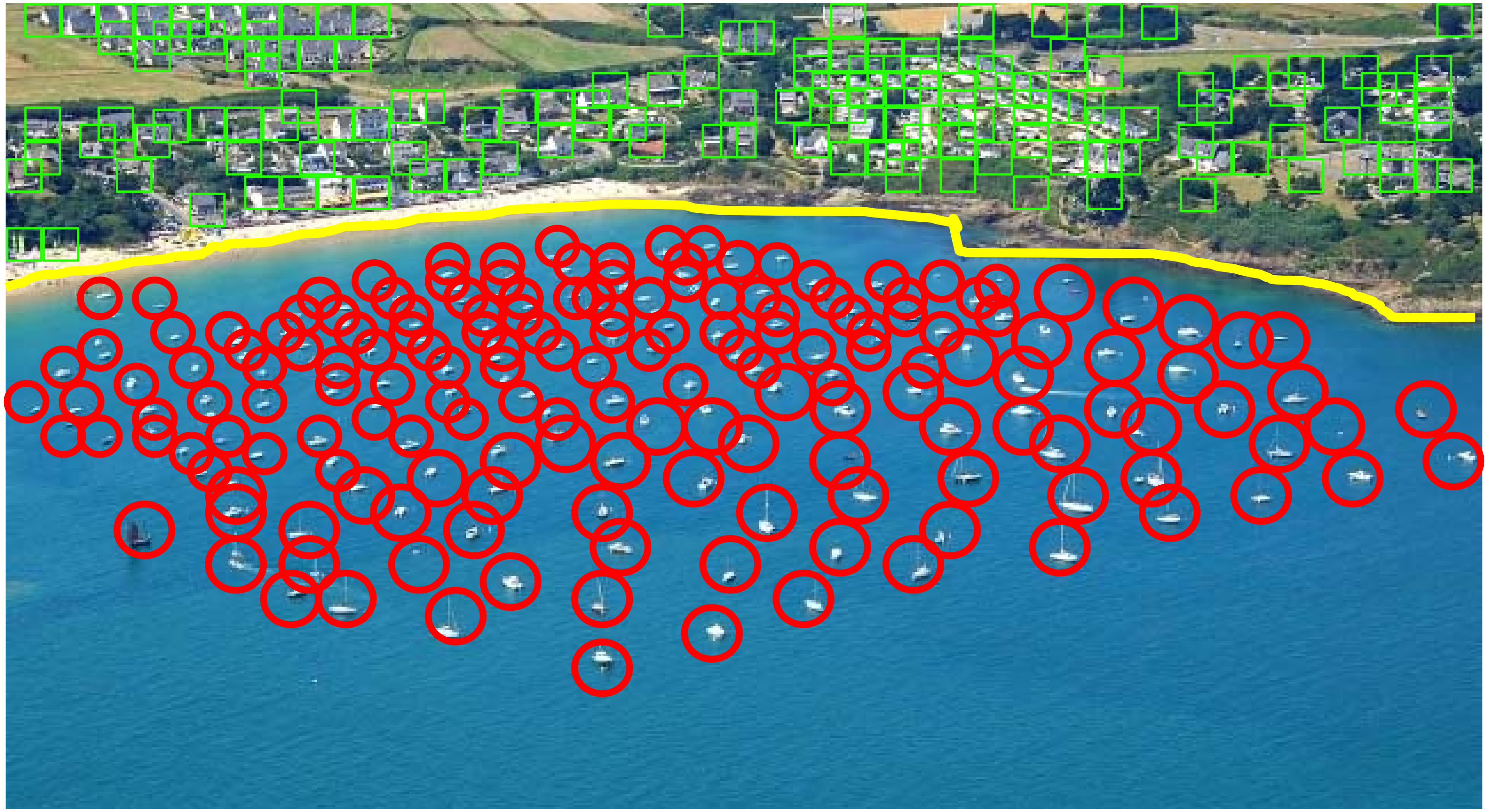


Basic principles of classification



- Want to classify objects as boats and houses.

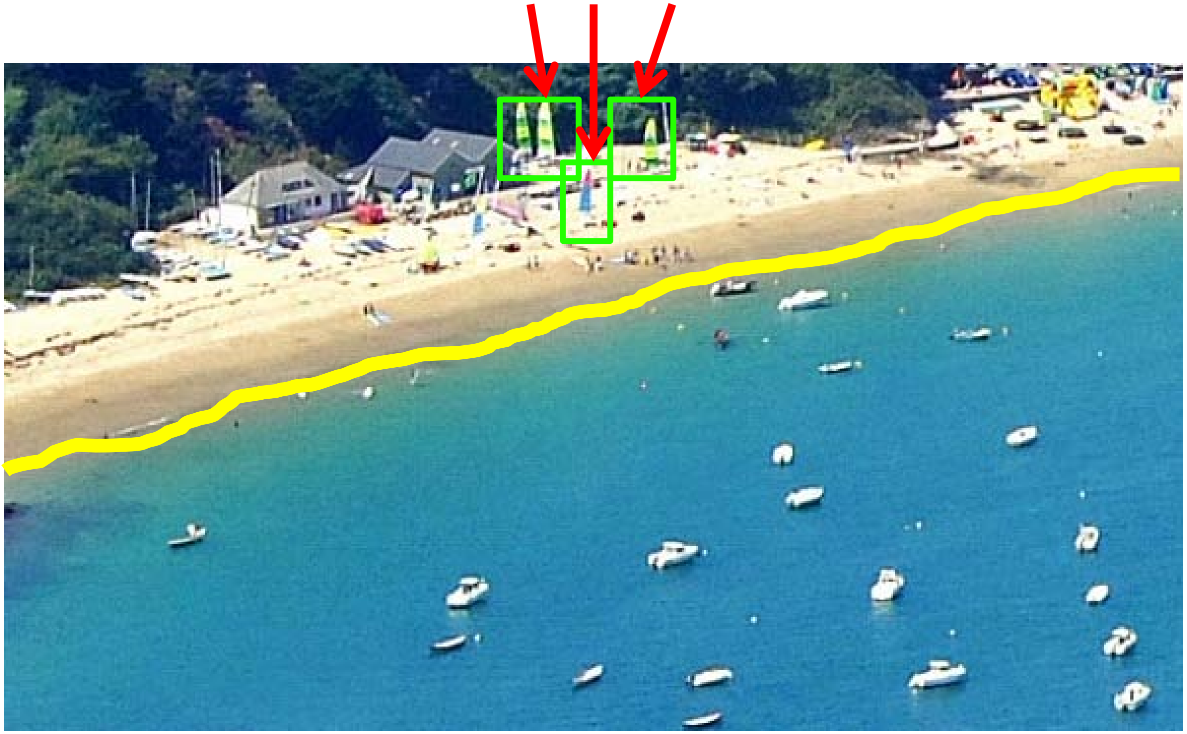
Basic principles of classification



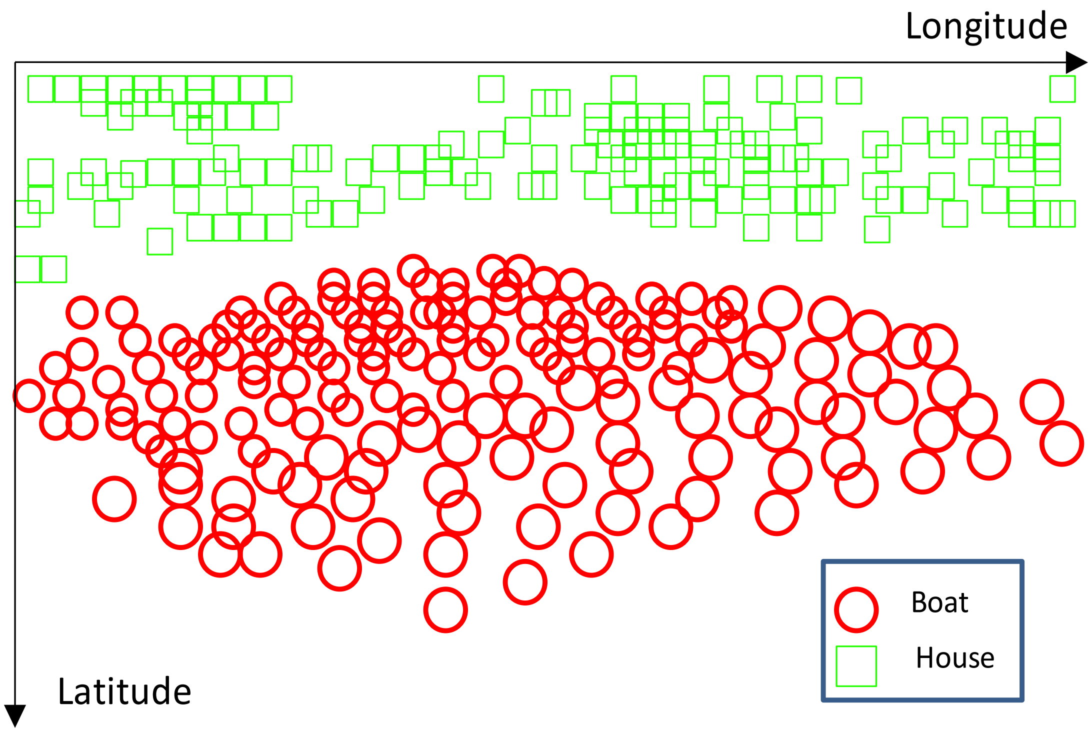
- All objects before the coast line are boats and all objects after the coast line are houses.
- Coast line serves as a *decision surface* that separates two classes.

Basic principles of classification

These boats will be misclassified as houses

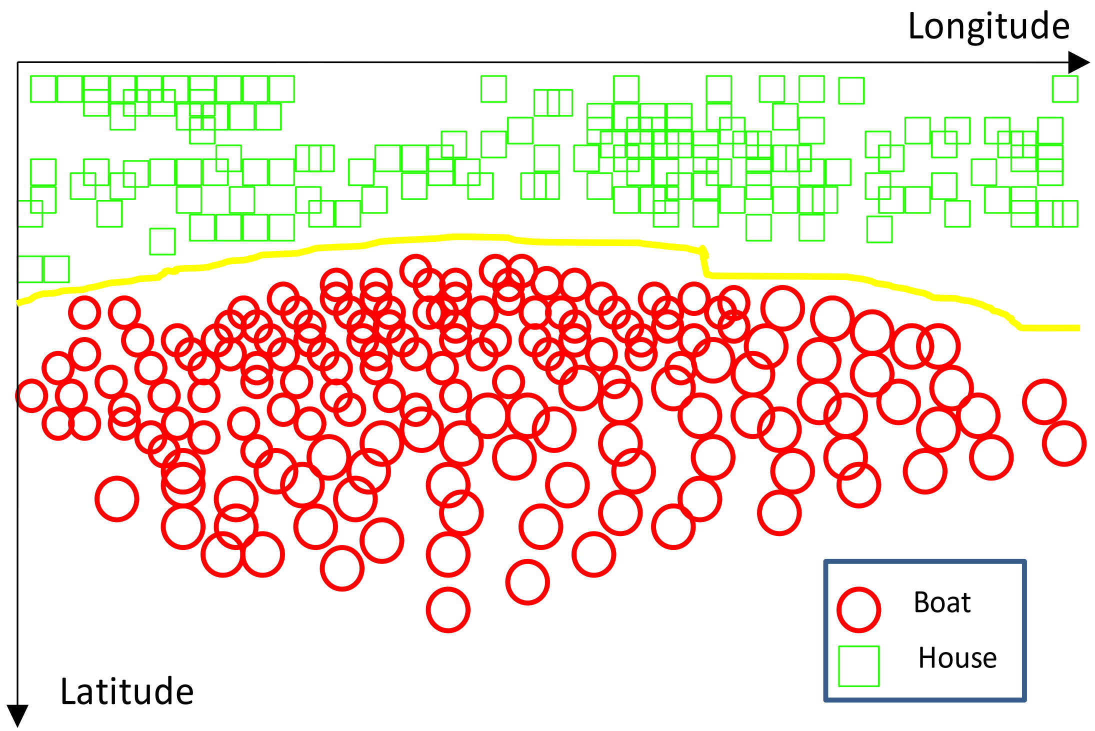


Basic principles of classification



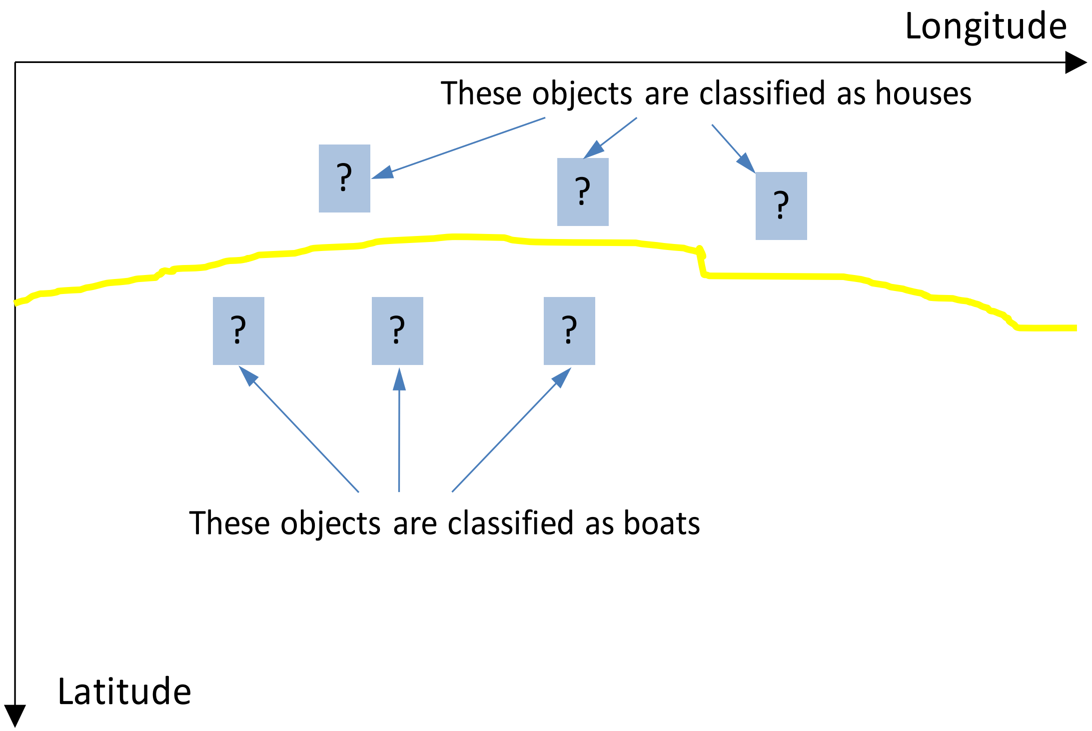
- The methods that build classification models (i.e., "*classification algorithms*") operate very similarly to the previous example.
- First all objects are represented geometrically.

Basic principles of classification



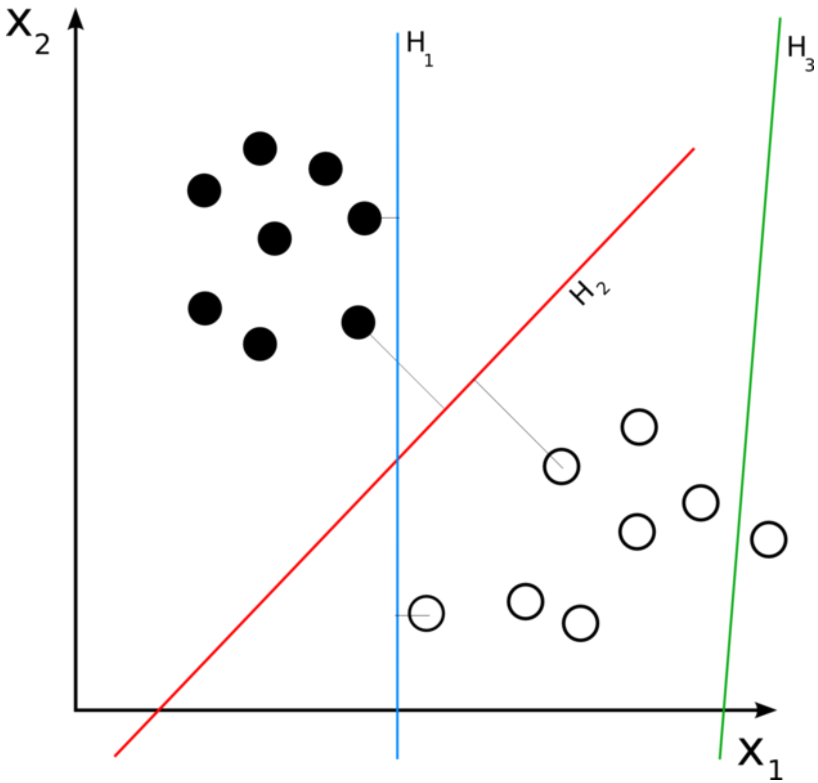
Then the algorithm seeks to find a decision surface that separates classes of objects

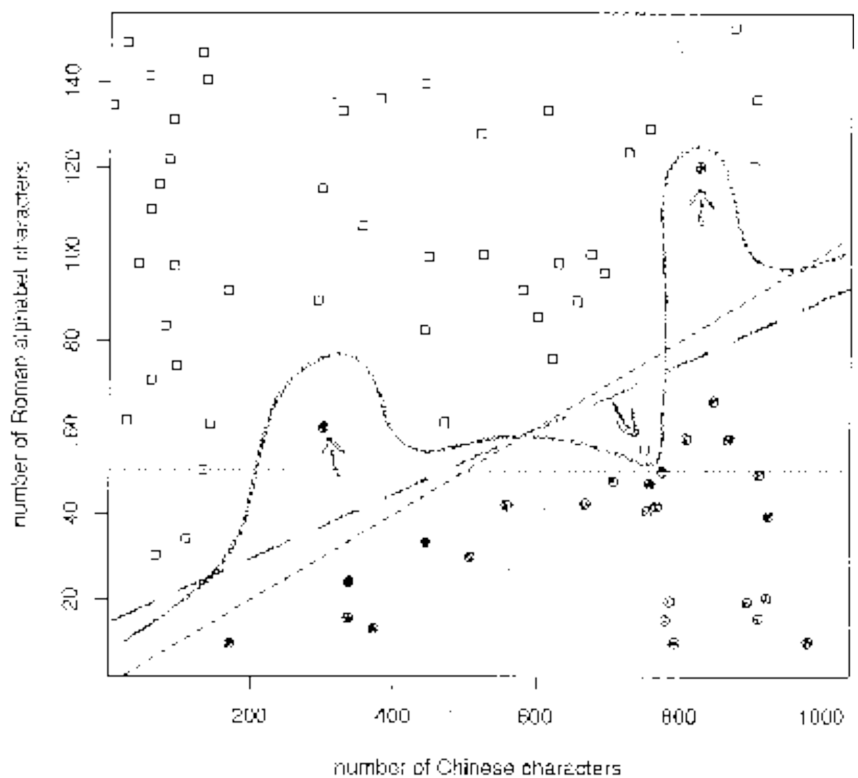
Basic principles of classification



Unseen (new) objects are classified as “boats” if they fall below the decision surface and as “houses” if the fall above it

Linear vs. Non-linear





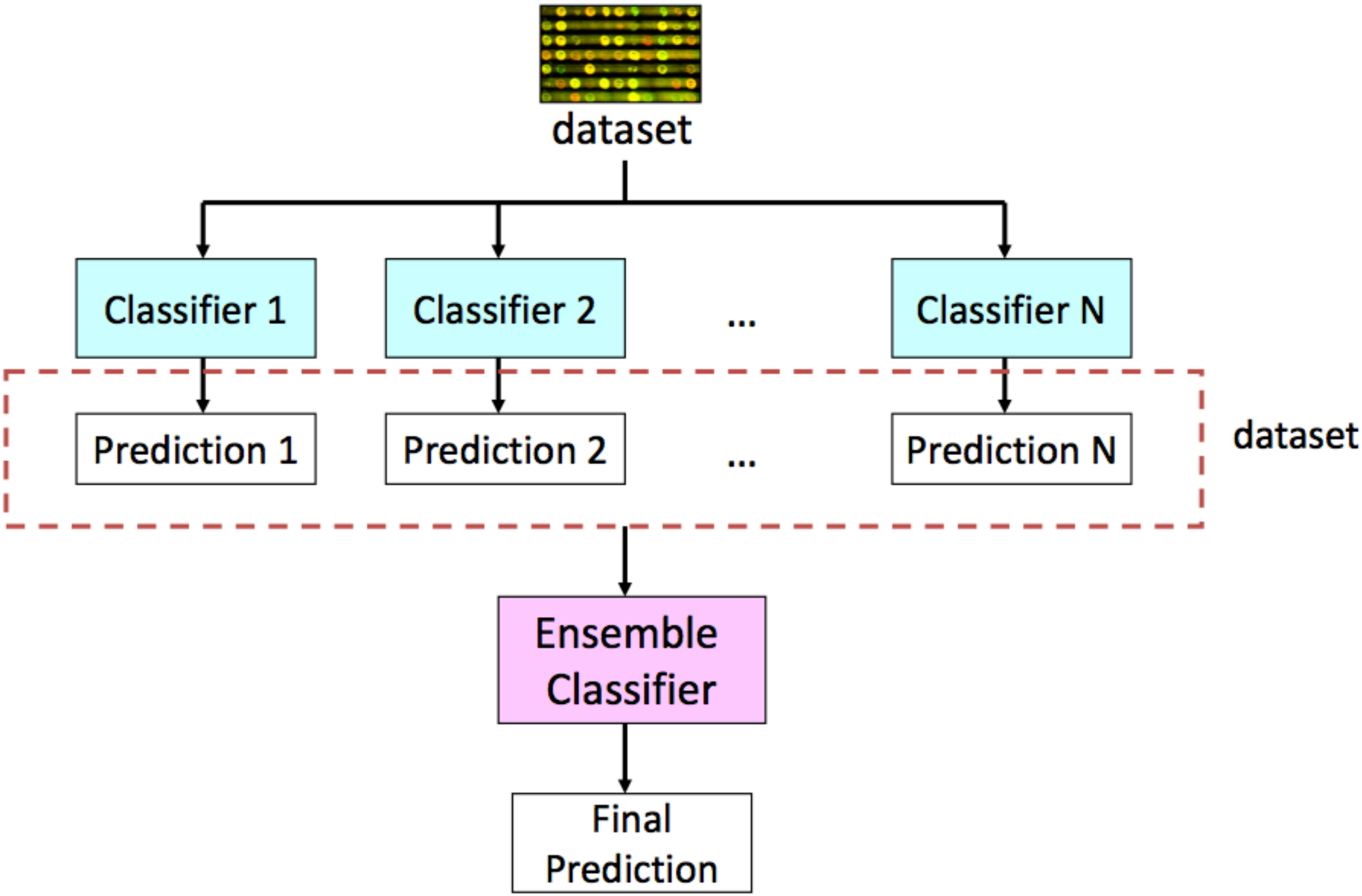
http://en.wikipedia.org/wiki/Linear_classifier

Classifiers

• K-Nearest Neighbors (KNN)	instance-based
• Backpropagation Neural Networks (NN)	neural networks
• Probabilistic Neural Networks (PNN)	
• Multi-Class SVM: One-Versus-Rest (OVR)	kernel-based
• Multi-Class SVM: One-Versus-One (OVO)	
• Multi-Class SVM: DAGSVM	
• Multi-Class SVM by Weston & Watkins (WW)	
• Multi-Class SVM by Crammer & Singer (CS)	
• Weighted Voting: One-Versus-Rest	voting
• Weighted Voting: One-Versus-One	
• Decision Trees: CART	decision trees

- Naïve Bayesian Classifiers
- Support Vector Machines

Ensemble classifiers



NAÏVE BAYES

What is it?

- A classifier that:

- Can train on a small # of observations
- Assumes independence between features
- Simple to implement

http://en.wikipedia.org/wiki/Naive_Bayes

Applications

- Typically K=2-class classifiers
  - K>2 possible
- E.g.
  - Spam vs. Non-spam
  - Female vs. Male

Prior vs. Posterior Probabilities

- Description vs. Prediction

How does it work?

- Given the prior probability of a class *k* in *K*
- …the prior probabilities of *N* object features
- …and their associations with *k*
- Calculate the posterior probability that a given object belongs to class *k*

- Easier to work through an example
  - http://bit.ly/10alkWY

http://en.wikipedia.org/wiki/Naive_Bayes

Making Best use of Data

- Train vs. Test
- Cross-validation
- ROC
  Ensembles

http://en.wikipedia.org/wiki/Naive_Bayes

HIERARCHICAL CLUSTERING