Machine Learning Approaches to NLP

Part II

Sameer Maskey

Topics for Today

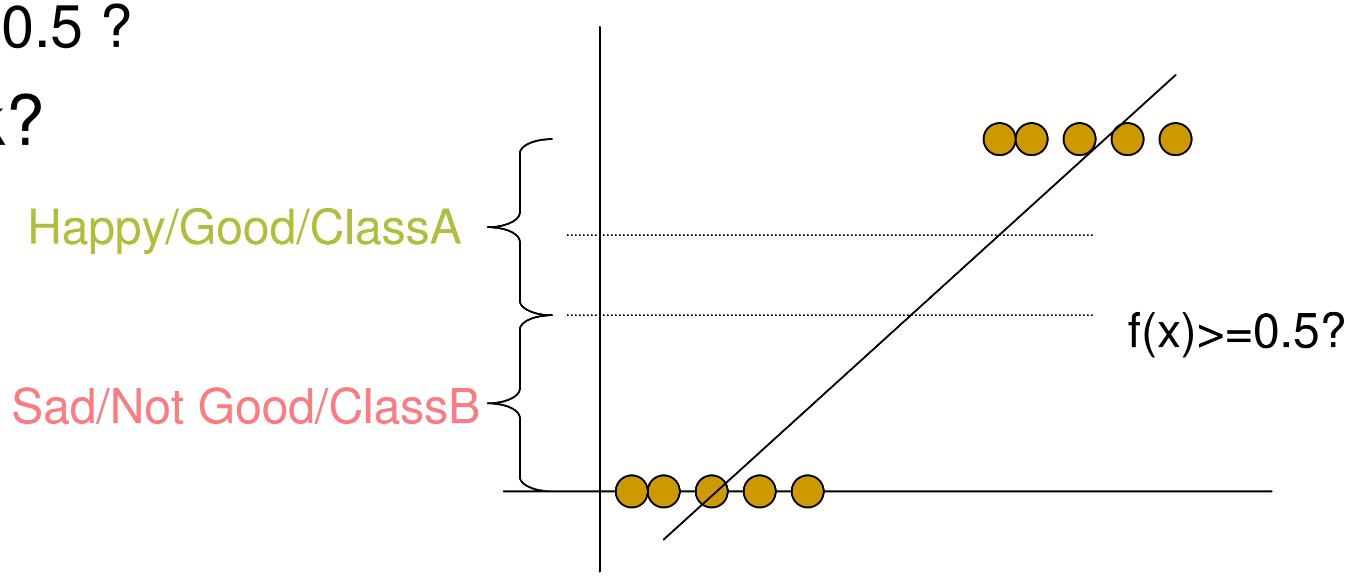
NLP -- ML

- Text Categorization

- Linear Methods of Classification
  - Perceptron
- Naïve Bayes
- Weka Tutorial

Regression to Classification

- Can we build a regression model to model binary classes?
- Train Regression and threshold the output
  - If f(x) >= 0.7 CLASS1
  - If f(x) < 0.7 CLASS2
  - f(x) >= 0.5 ?
- Is this ok?

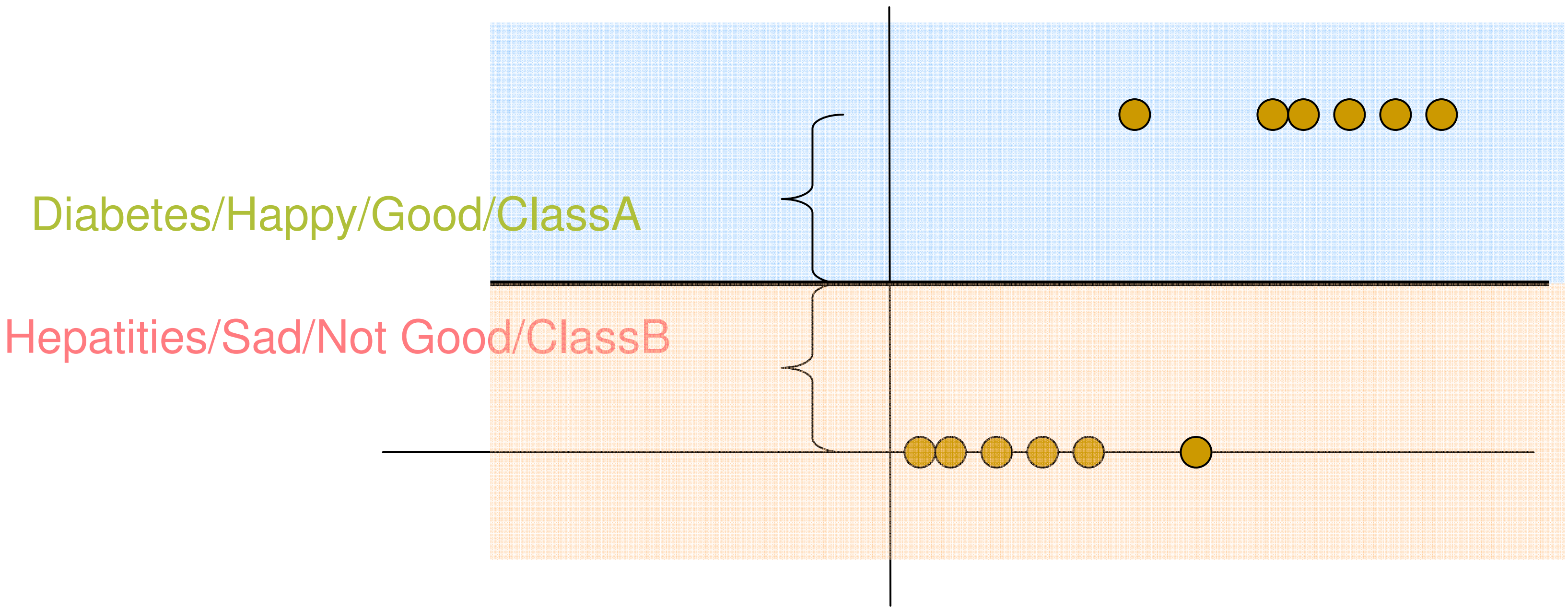


Linear Discrimination with a Hyperplane

- We looked at cosine similarity for text classification
- Besides cosine similarity there are many other ways for text classification
- Dimensionality reduction is one way of classification : Fisher's Linear Discriminant
- We can also try to find they discriminating hyperplane by reducing the total error in training
  - Perceptrons is one such algorithm

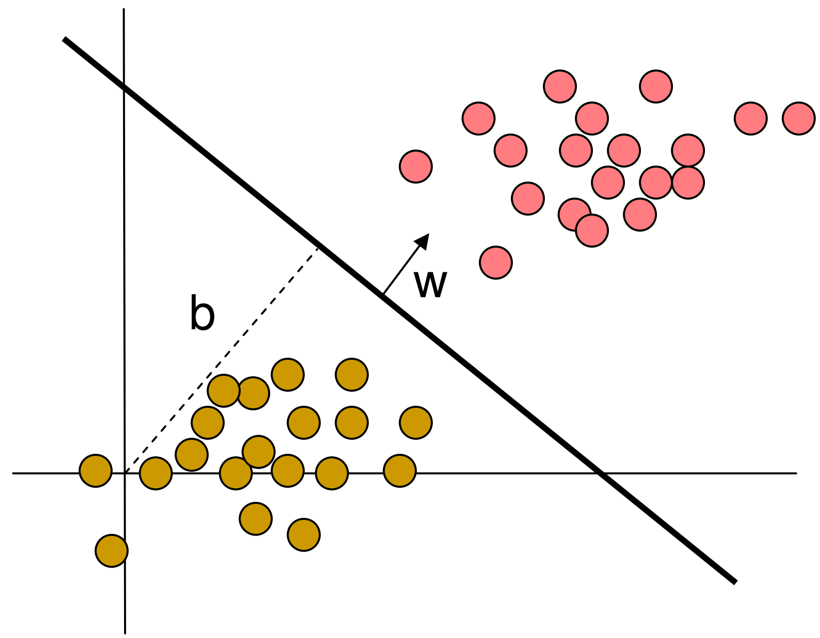
Half Plane and Half Spaces

- Half plane is a region on one side of an infinite long line, and does not contain any points from other side
- Half space n-dimensional space obtained by removing points on one side of hyperplane (n-1 dimension)
  - What would it look like for a 3 dimensional space



Discriminative Classification

$$f(x) = \mathbf{w}^T x + b$$



Perceptron for Text Classification

- We want to find a function that would produce least training error

$$R_n(w) = \frac{1}{n} \sum_{i=1}^{n} Loss(y_i, f(x_i; w))$$

Minimizing Training Error

Given training data $< (x_i, y_i) >$
We want to find w such that
$y_i(w.x_i) > 0$ if $y_i > 0$
$y_i(w.x_i) < 0$ if $y_i < 0$

- We can iterate over all points and adjust the parameters

$$w \leftarrow w + y_i x_i$$

$$\text{if } y \neq f(x_i; w)$$

- Parameters are updated only if the classifier makes a mistake

Perceptron Algorithm

We are given (x_i, y_i)

Initialize w

Do until converged

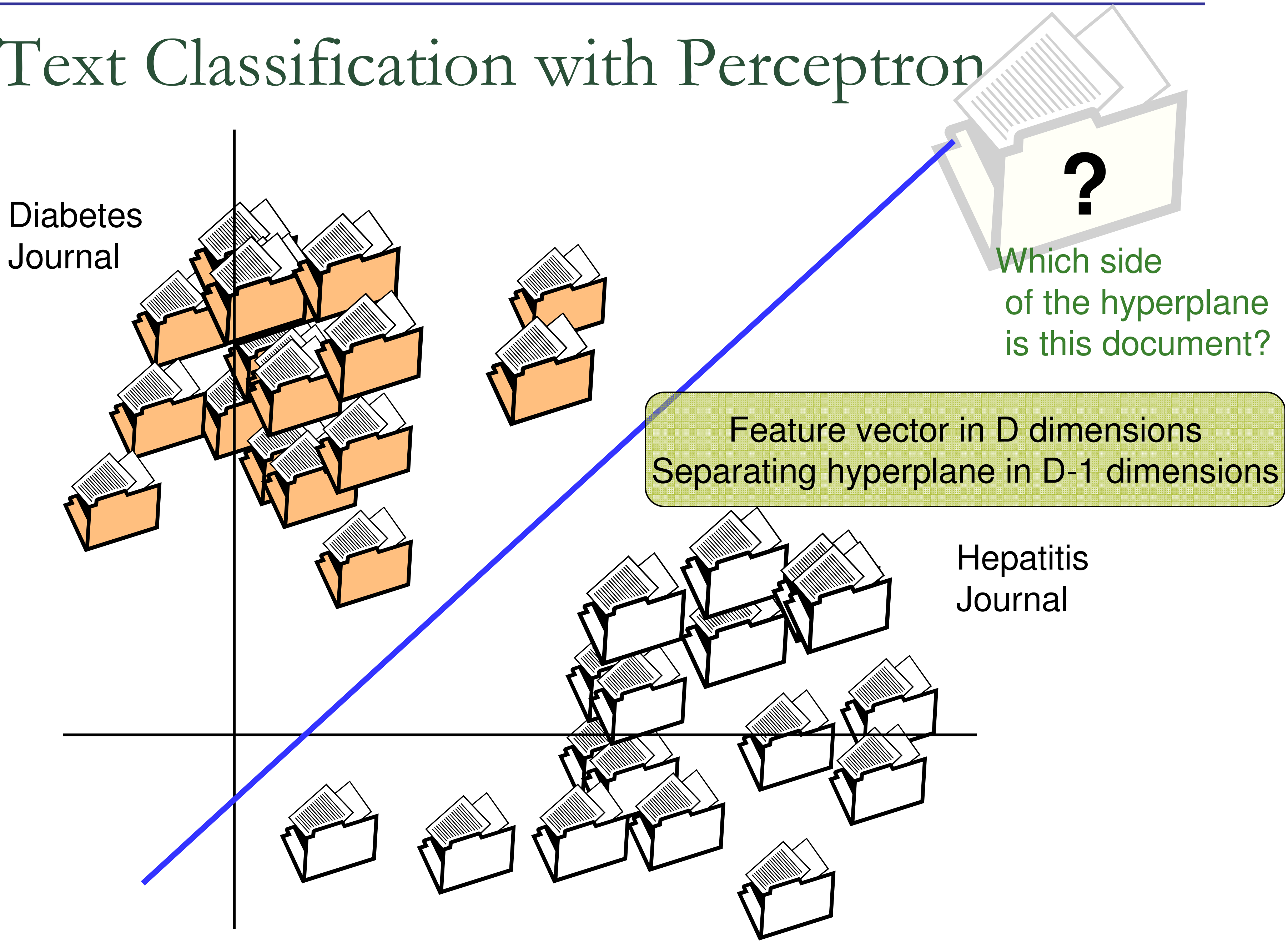
if error$((y_i, f(x_i, w)) == TRUE)$

$$w \leftarrow w + y_i x_i$$

end if

End do

Text Classification with Perceptron



Text Classification with Perceptron

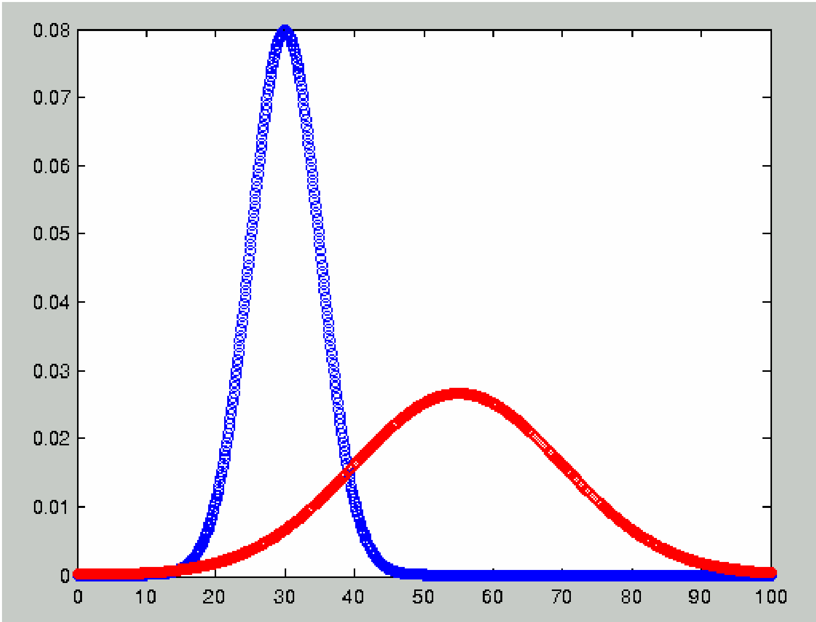
- Perceptron may not always converge
- Ok for two classes, not trivial to extend it to multiple classes
- Not the optimal hyperplane
  - Many hyperplanes that separates the class
  - Depends on random initialization

Generative vs. Discriminative

- Generative Classifier
  - Model joint probability p(x,y) where x are inputs and y are labels
  - Make prediction using Bayes rule to compute p(y|x)
- Discriminative Classifier
  - Try to predict output directly
  - Model p(y|x) directly

Generative Classifier

- We can model class conditional densities using Gaussian distributions
- If we know class conditional densities
  - p(x| y=C1)
  - p(x|y=C2)
- We can find a decision to classify the unseen example

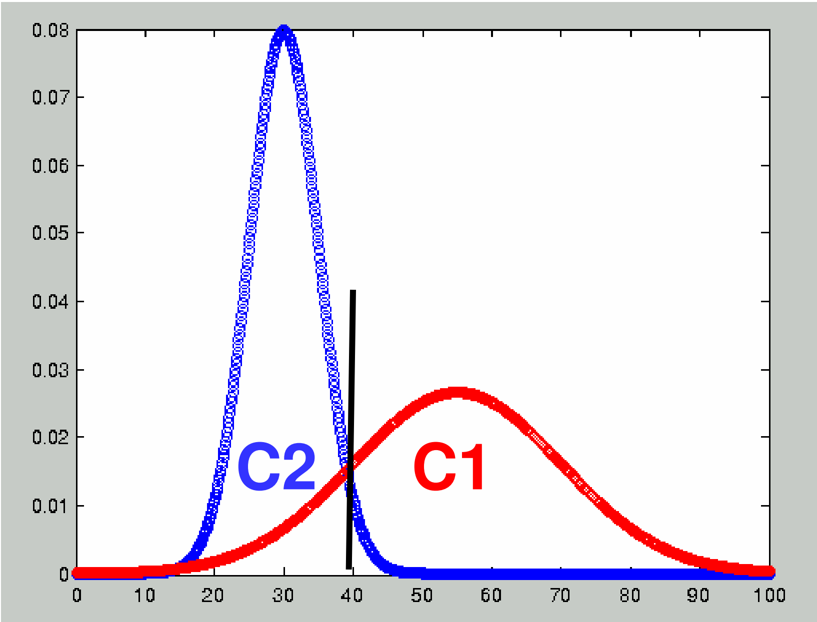


Bayes Rule

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

→So how would this rule help in classifying text in two different categories; Diabetes vs Hepatitis

→Think about distribution of count of the word diabetes for example



Generative Classifier

- If we have two classes C1 and C2
- We can estimate Gaussian distribution of the features for both classes
  - Let's say we have a feature x
    - x = length of a document
  - And class label (y)
    - y = 1 diabetes or 2 hepatitis

(30, 1)
(55, 2)
(24, 1)
(40, 1)
(35, 2)
…

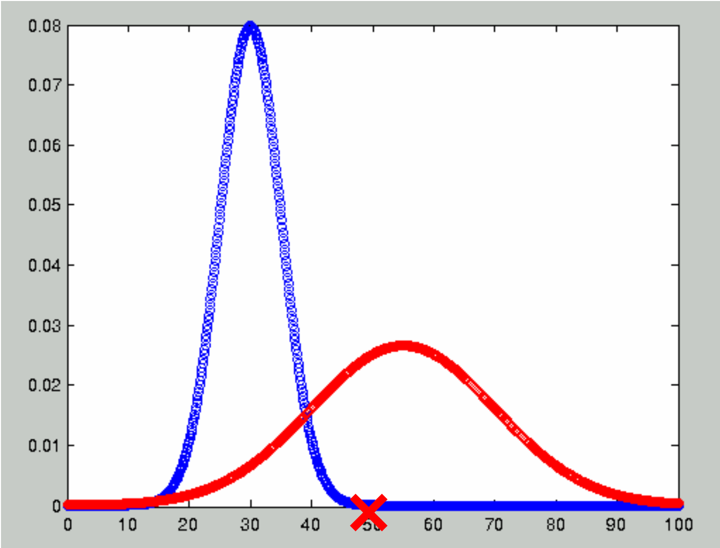
Find out μ_i and $\sum_i$ from data for both classes

$$N(\mu_1, \textstyle\sum_1)$$
$$N(\mu_2, \textstyle\sum_2)$$

Gaussian Distribution $$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Generative Classifier

- Given a new data point find out posterior probability from each class and take a log ratio
- If higher posterior probability for C1, it means new x better explained by the Gaussian distribution of C1



$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$$p(y=1|x) \propto p(x|\mu_1, \textstyle\sum_1)p(y=1)$$

Naïve Bayes Classifier

- Naïve Bayes Classifier a type of Generative classifier
  - Compute class-conditional distribution but with conditional independence assumption
- Shown to be very useful for text categorization

Conditional Independence

- Given random variables X, Y,Z, X is conditionally independent of Y given Z if and only if

$$P(X|Y,Z) = p(X|Z)$$

$$\begin{aligned} P(X|Y) &= P(X_1, X_2|Y) \\ &= P(X_1|X_2, Y)P(X_2|Y) \\ &= P(X_1|Y)P(X_2|Y) \end{aligned}$$

Conditional Independence

- For a feature vector with ‘n’ features we get

$$P(X_1, X_2, ..., X_N|Y) = \Pi_{i=1}^{N} P(X_i|Y)$$

N features are conditionally independent of one another given Y

Why would this assumption help?

Naïve Bayes Classifier for Text

$$P(Y_k, X_1, X_2, ..., X_N) = P(Y_k)\Pi_i P(X_i|Y_k)$$

Prior Probability of the Class

Conditional Probability of feature given the Class

Here N is the number of words, not to confuse with the total vocabulary size

Naïve Bayes Classifier for Text

$$P(Y = y_k | X_1, X_2, ..., X_N) = \frac{P(Y=y_k)P(X_1,X_2,..,X_N|Y=y_k)}{\sum_j P(Y=y_j)P(X_1,X_2,..,X_N|Y=y_j)}$$

$$= \frac{P(Y=y_k)\Pi_i P(X_i|Y=y_k)}{\sum_j P(Y=y_j)\Pi_i P(X_i|Y=y_j)}$$

$$Y \leftarrow argmax_{y_k} P(Y = y_k)\Pi_i P(X_i|Y = y_k)$$

Naïve Bayes Classifier for Text

- Given the training data what are the parameters to be estimated?

$P(y)$ $P(X|y_1)$ $P(X|y_2)$

$P(y)$	$P(X\|y_1)$	$P(X\|y_2)$
Diabetes : 0.8 Hepatitis : 0.2	the: 0.001 diabetic : 0.02 blood : 0.0015 sugar : 0.02 weight : 0.018 …	the: 0.001 diabetic : 0.0001 water : 0.0118 fever : 0.01 weight : 0.008 …

$$y \leftarrow argmax_{y_k} P(y = y_k)\Pi_i P(X_i|y = y_k)$$

Estimating Parameters

- Maximum Likelihood Estimates
  - Relative Frequency Counts
- For a new document
  - Find which one gives higher posterior probability
    - Log ratio
    - Thresholding
- Classify accordingly

Smoothing

- MLE for Naïve Bayes (relative frequency counts) may not generalize well
  - Zero counts
- Smoothing
  - With less evidence, believe in prior more
  - With more evidence, believe in data more

Laplace Smoothing

- Assume we have one more count for each element
- Zero counts become 1

$$P_{smooth}(w) = \frac{c_w + 1}{\sum_w \{c(w) + 1\}}$$

$$P_{smooth}(w) = \frac{c_w + 1}{N + V}$$

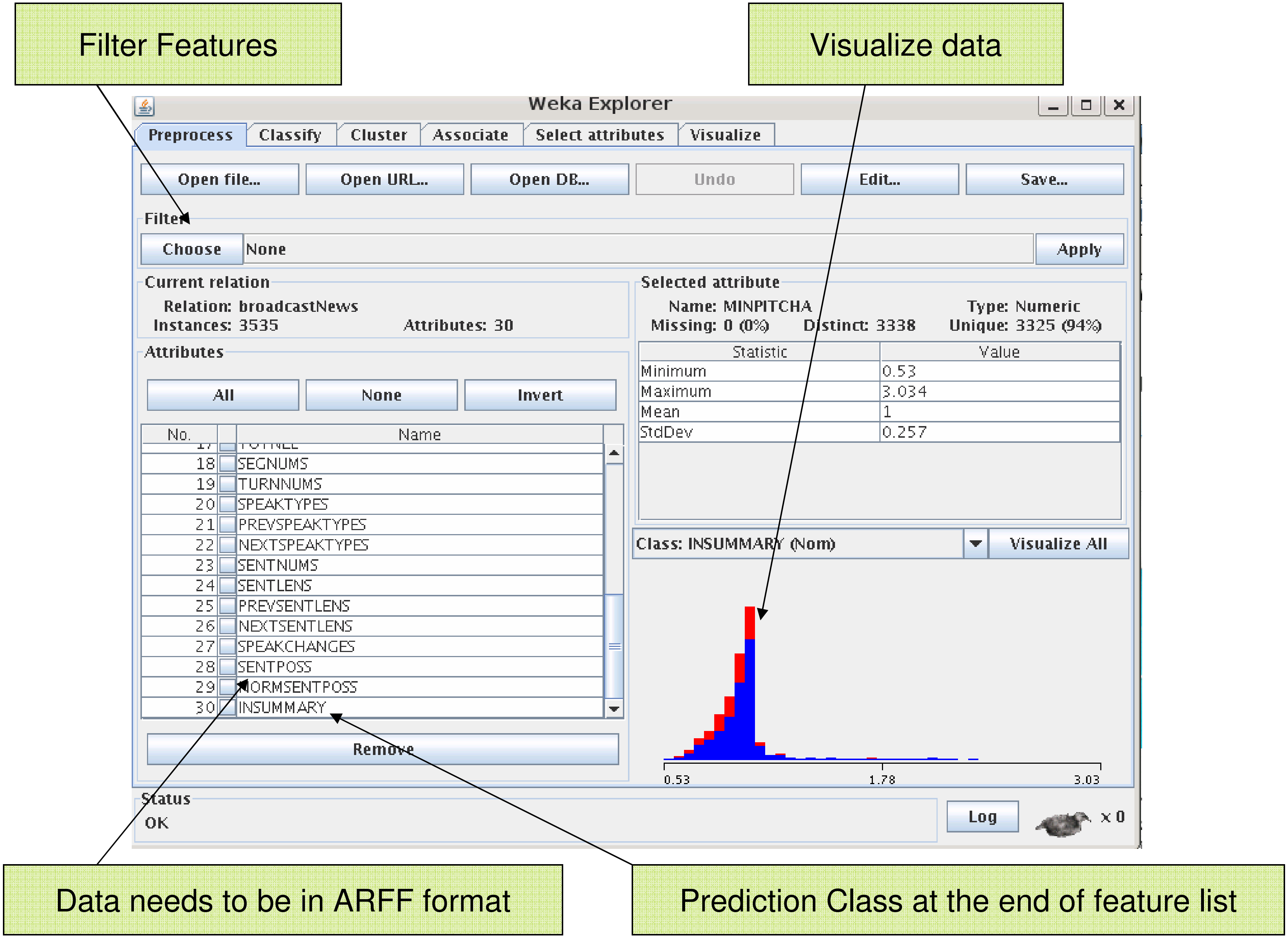
Vocab Size

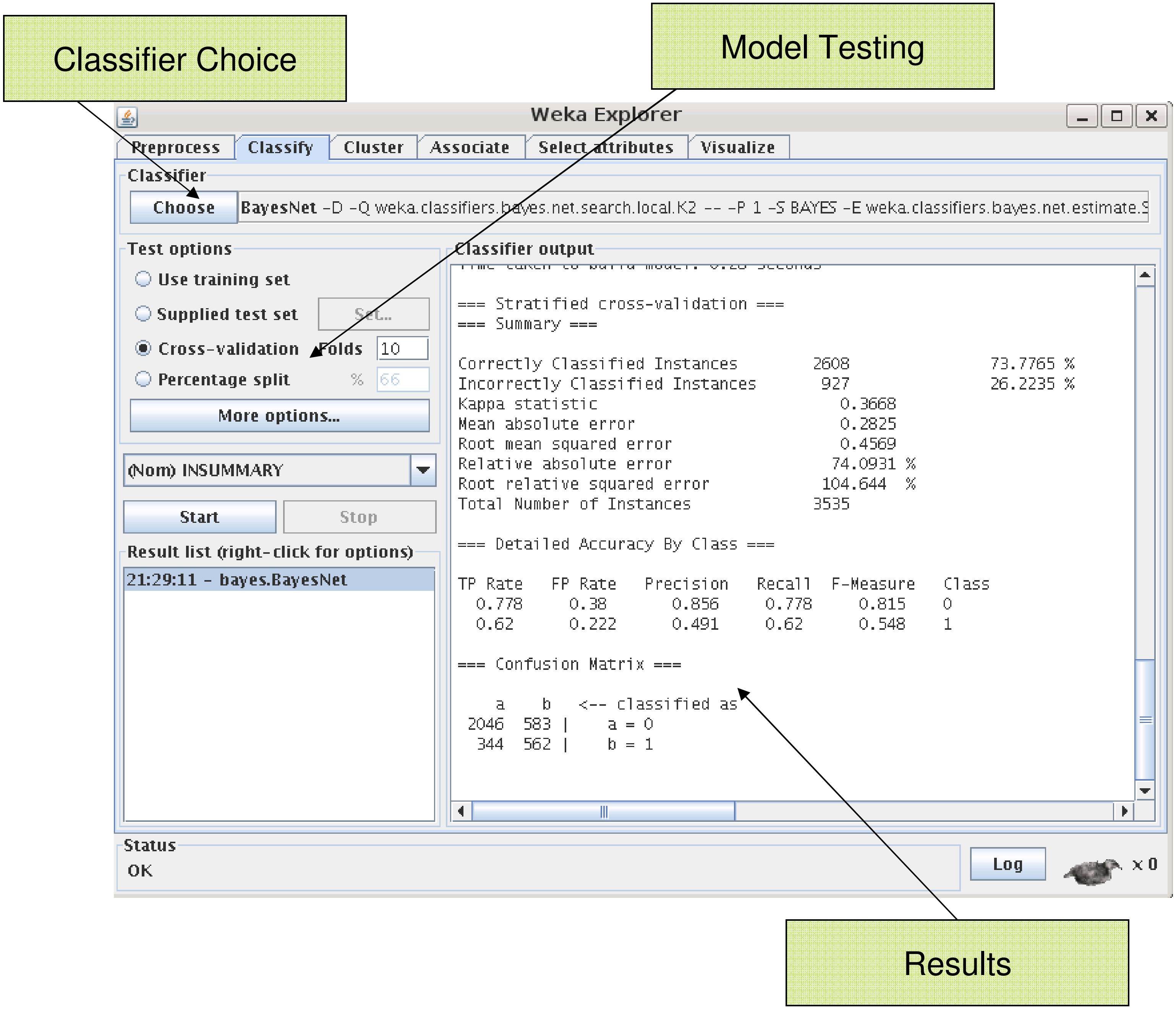
Weka

- Publicly available free software that includes many common ML algorithms that are used in Natural Language Processing
- GUI and Commandline Interface
- Feature Selection, ML algorithms, Data filtering, Visualization

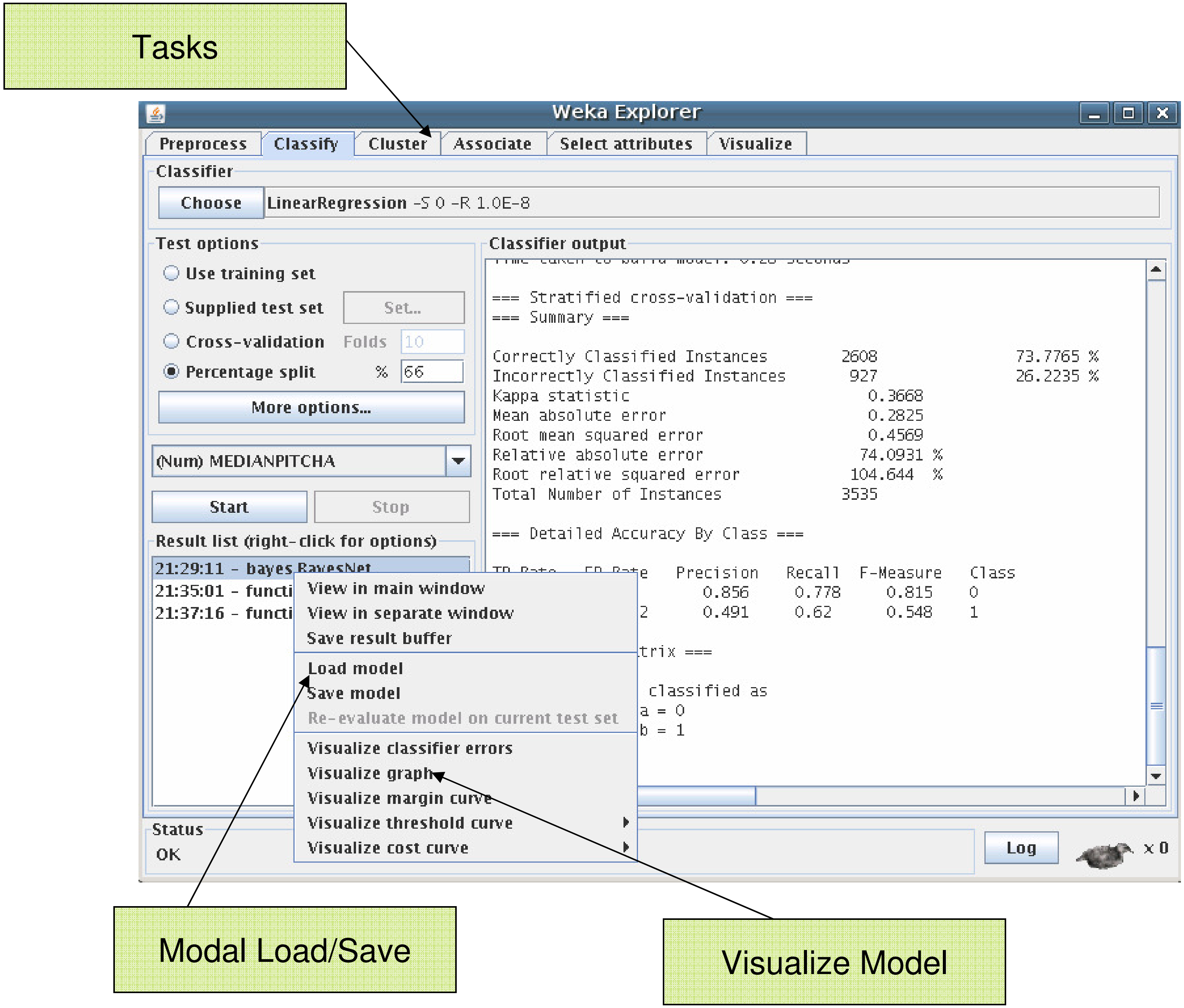
Weka Download and Setup

- http://sourceforge.net/projects/weka/files/weka-3-4/3.4.17/weka-3-4-17.zip/download
- >> unzip weka-3-4-17.zip
- >> java -jar weka-3-4-17/weka.jar
- >> Click on Explorer


Filter Features
Visualize data
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: broadcastNews
Instances: 3535
Attributes: 30
Selected attribute
Name: MINPITCHA
Type: Numeric
Missing: 0 (0%)
Distinct: 3338
Unique: 3325 (94%)
Attributes
All
None
Invert
No.
Name
18 SEGNUMS
19 TURNNUMS
20 SPEAKTYPES
21 PREVSPEAKTYPES
22 NEXTSPEAKTYPES
23 SENTNUMS
24 SENTLENS
25 PREVSENTLENS
26 NEXTSENTLENS
27 SPEAKCHANGES
28 SENTPOSS
29 NORMSENTPOSS
30 INSUMMARY
Remove
Statistic
Value
Minimum
0.53
Maximum
3.034
Mean
1
StdDev
0.257
Class: INSUMMARY (Nom)
Visualize All
0.53
1.78
3.03
Status
OK
Log
x 0
Data needs to be in ARFF format
Prediction Class at the end of feature list

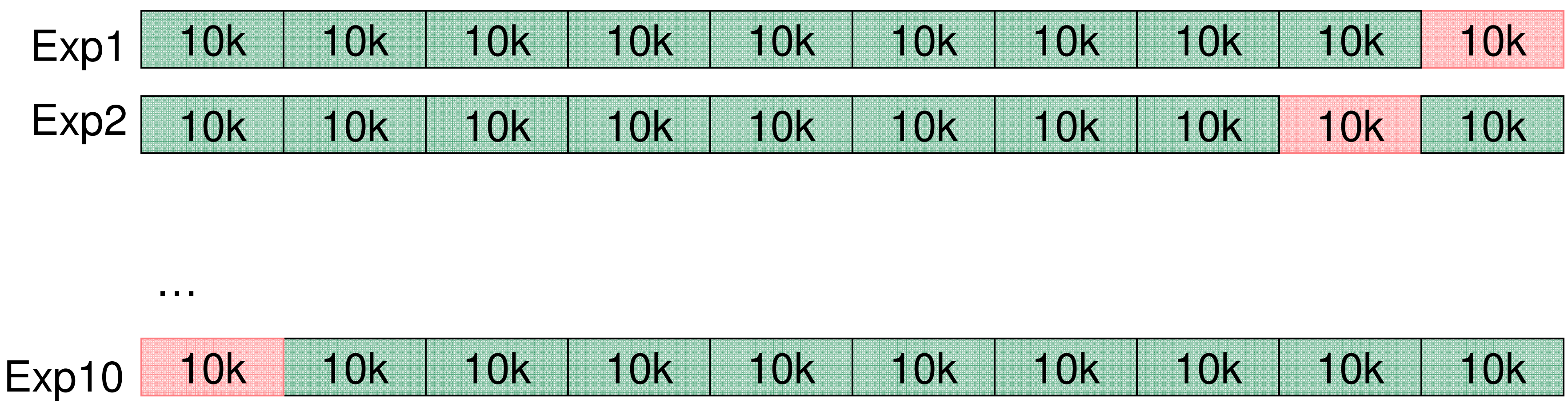


Classifier Choice
Model Testing
Results
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.S
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) INSUMMARY
Start
Stop
Result list (right-click for options)
21:29:11 - bayes.BayesNet
Classifier output
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 2608 73.7765 %
Incorrectly Classified Instances 927 26.2235 %
Kappa statistic 0.3668
Mean absolute error 0.2825
Root mean squared error 0.4569
Relative absolute error 74.0931 %
Root relative squared error 104.644 %
Total Number of Instances 3535
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure Class
0.778 0.38 0.856 0.778 0.815 0
0.62 0.222 0.491 0.62 0.548 1
=== Confusion Matrix ===
a b <-- classified as
2046 583 | a = 0
344 562 | b = 1
Status
OK
Log
x 0


Tasks
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
LinearRegression -S 0 -R 1.0E-8
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Num) MEDIANPITCHA
Start
Stop
Result list (right-click for options)
21:29:11 - bayes.BayesNet
21:35:01 - functi
21:37:16 - functi
Classifier output
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 2608 73.7765 %
Incorrectly Classified Instances 927 26.2235 %
Kappa statistic 0.3668
Mean absolute error 0.2825
Root mean squared error 0.4569
Relative absolute error 74.0931 %
Root relative squared error 104.644 %
Total Number of Instances 3535
=== Detailed Accuracy By Class ===
Precision Recall F-Measure Class
0.856 0.778 0.815 0
0.491 0.62 0.548 1
View in main window
View in separate window
Save result buffer
Load model
Save model
Re-evaluate model on current test set
Visualize classifier errors
Visualize graph
Visualize margin curve
Visualize threshold curve
Visualize cost curve
Status
OK
Log
x 0
Modal Load/Save
Visualize Model

10-fold Cross Validation

- 10 fold cross validation
  - Assuming we have 100K data points
    - Train on 90K (1 to 90,000)
    - Test on 10K (90,001 to 100,000)
  - But we can do this 10 times if we select different 10K of test data point each time



- 10 experiments, build model and test times with 10 different sets of training and test data
- Average the accuracy across 10 experiments
- We can do any N-fold cross validation to test our model

Interpreting Weka Results

	Actual	
Predicted	TP True Positive	FP False Positive
	FN False Negative	TN True Negative

Precision, Recall, F-Measure

Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F-Measure	$\frac{(1+beta^2) * Precision * Recall}{(beta^2*Precision + Recall)}$
Accuracy	(TP+TN)/(TP+TN+FP+FN)

Confusion Matrix

- Assume we are classifying text into two categories Hepatitis (H) and Others (B)
- Let's assume we had 1000 documents such that 500 are H and 500 are B
- Assume we got given predictions

Actual

		H	B
Predicted	H	400	200
	B	100	300

Precision	0.6667
Recall	0.8000
F-measure	0.7273
Accuracy	0.7000

Commandline for Weka

- Make sure CLASSPATH variable is setup; can also give the path explicitly using –cp parameter
  - >> export CLASSPATH=$CLASSPATH:/home/smaskey/soft/weka-3-4-17/weka.jar
- Try to see if java can access the classes for classifiers
  - >> java weka.classifiers.bayes.NaiveBayes
- Try to build a model from commandline
  - >>java weka.classifiers.trees.J48 -i -t data/weather.arff
- Try other examples from Weka wiki
  - >>java weka.classifiers.bayes.NaiveBayes -K -t soybean-train.arff -T soybean-test.arff -p 0

Weka Demo

- Text Classification with Weka
  - Classify documents into Hockey or Baseball
- 20 Newsgroup corpus
- Code and data will be available from the course website