UNIVERSITÄT MANNHEIM

Data Mining I
Classification, Part 1

Heiko Paulheim

Outline

A Couple of Questions

- What is this?
- Why do you know?
- How have you come to that knowledge?

Introductory Example

- Learning a new concept, e.g., "Tree"

"tree"

"tree"

"tree"

"not a tree"

"not a tree"

"not a tree"

Introductory Example

- Example: learning a new concept, e.g., "Tree"
  - we look at (positive and negative) examples
  - ...and derive a *model*
    - e.g., "Trees are big, green plants"





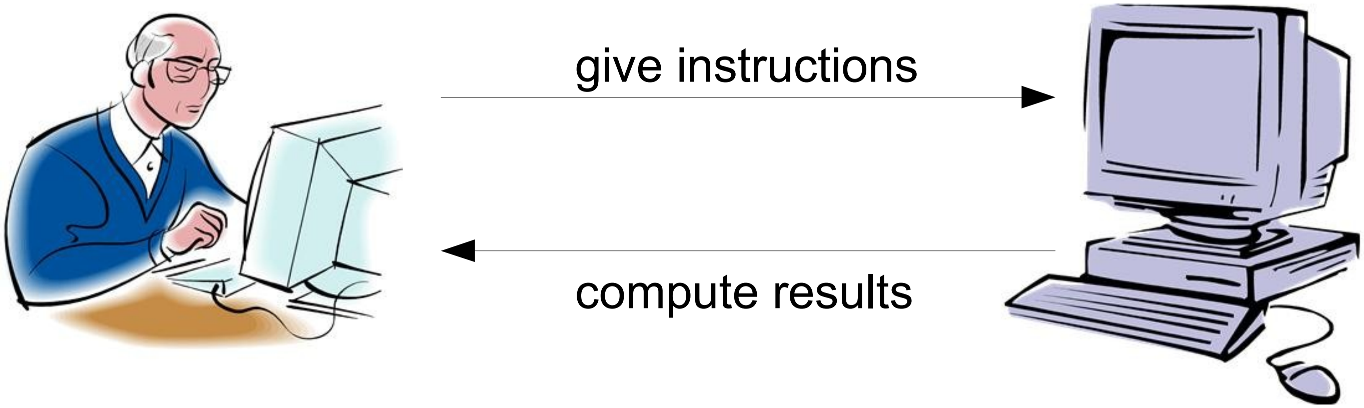

- Goal: Classification of new instances

"tree?"

Warning:
Models are only approximating examples!
Not guaranteed to be correct or complete!

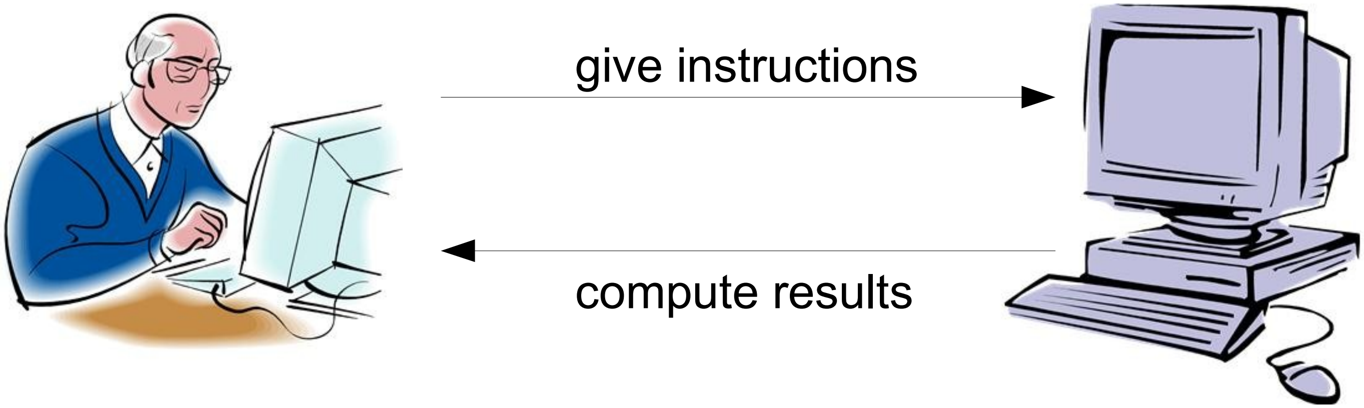
What is Classification?

- Classic programming:
  - `if more than 10 orders/year and more than $100k spent`
    `set customer.isPremiumCustomer = true`
- The prevalent style of programming computers
  - works well as long as we know the rules
  - e.g.: what makes a customer a premium customer?



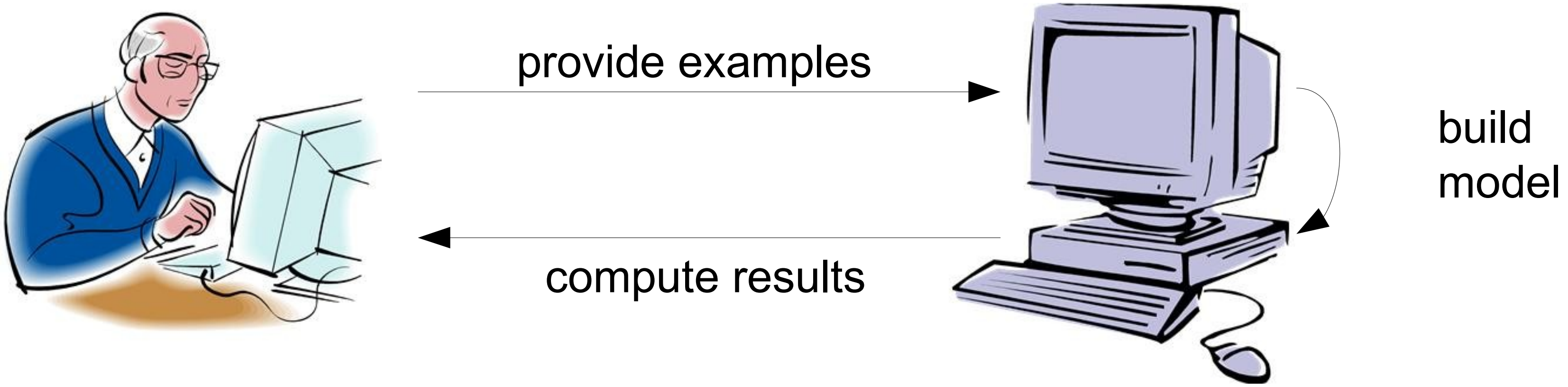
What is Classification?

- Sometimes, it's not so easy
- E.g., due to missing knowledge
  - `if customer is likely to order a new phone`
    `send advertisement for new phones`
- E.g., due to difficult formalization as an algorithm
  - `if customer review is angry`
    `send apology`



What is Classification?

- A different paradigm:
  - User provides computer with examples
  - Computer finds model by itself
  - Notion: the computer *learns* from examples (term: *machine learning)*
- Example
  - labeled examples of angry and non-angry customer reviews
  - computer finds model for telling if a customer is angry



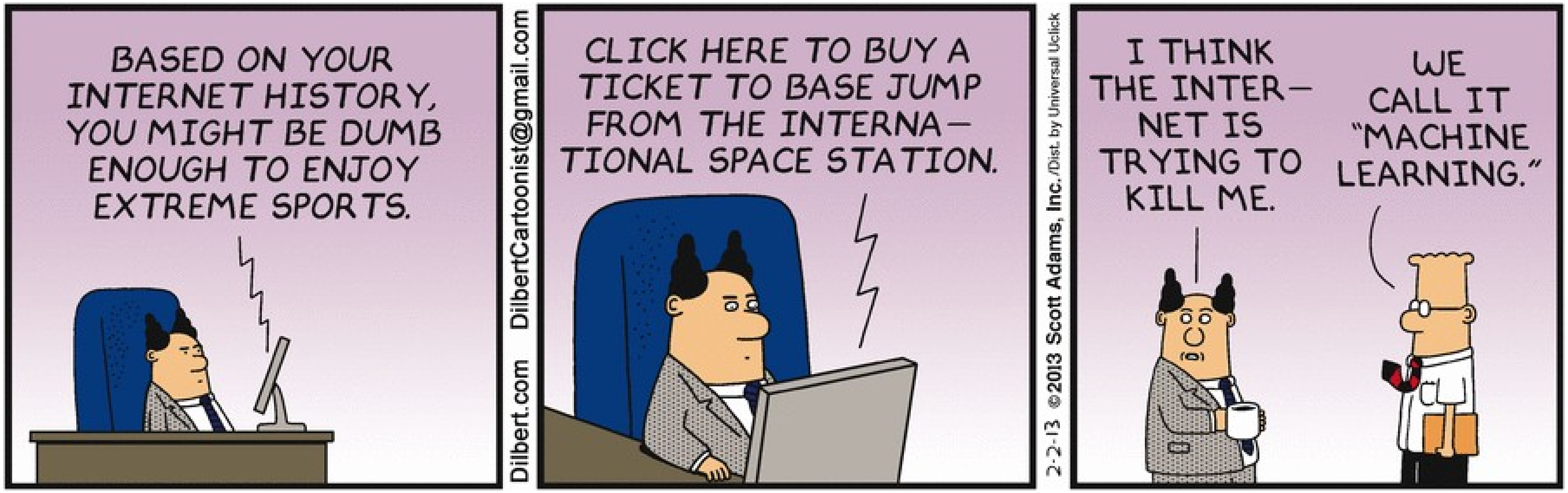
Classification: Formal Definition

- Given:
  - a set of labeled records, consisting of
    - data fields (a.k.a. attributes or features)
    - a class label (e.g., true/false)
- Generate
  - a function f(r)
    - input: a record
    - output: a class label
  - which can be used for classifying previously unseen records

- Variants:
  - single class problems (e.g., only true/false)
  - multi class problems
  - multi label problems (more than one class per record, not covered in this lecture)
  - hierarchical multi class/label problems (with class hierarchy, e.g., product categories)

What is Classification?

- Classification is a *supervised* learning problem
  - i.e., given labeled data, learn a prediction function for those labels



http://dilbert.com/strip/2013-02-02

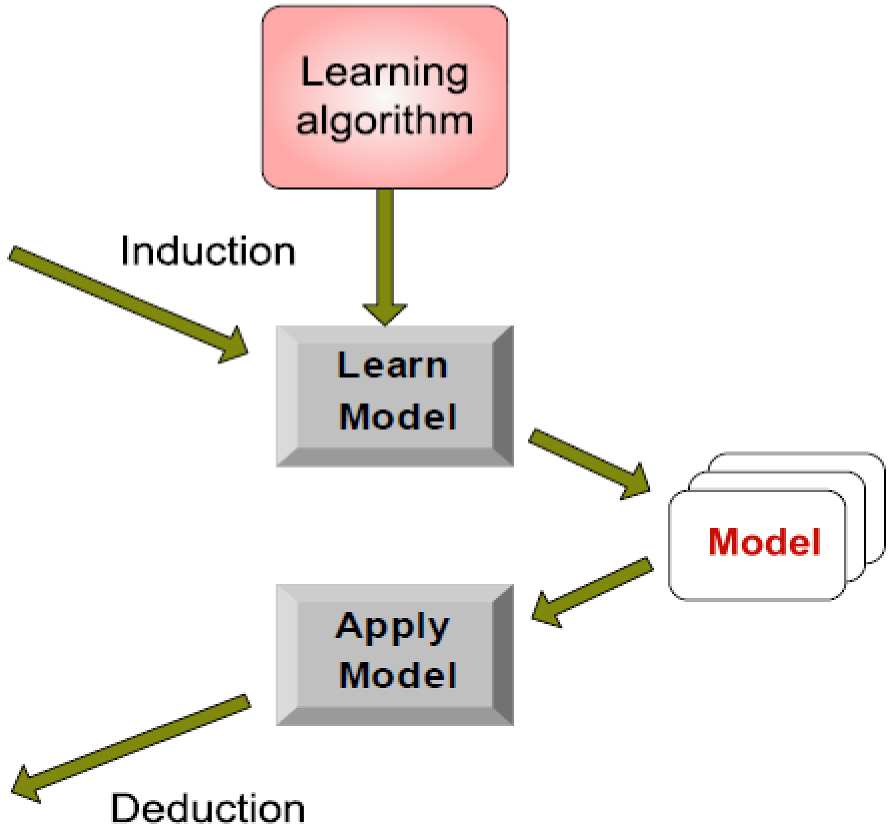
The Classification Workflow

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

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	?

Unseen Records



Classification Applications – Examples

- Attributes: a set of symptoms (headache, sore throat...)
  - class: does the patient suffer from disease X?
- Attributes: the values in your tax declaration
  - class: are you trying to cheat?
- Attributes: your age, income, debts, …
  - class: are you getting credit by your bank?
- Attributes: the countries you phoned with in the last 6 months
  - class: are you a terrorist?

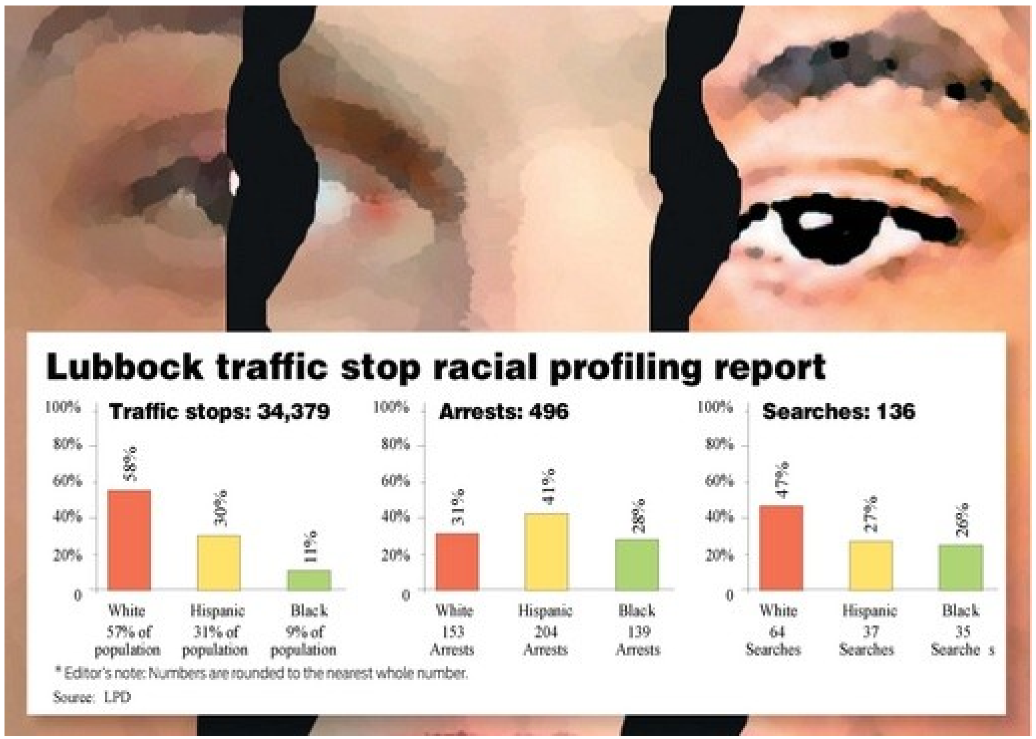
Classification Applications – Examples

- Attributes: words in a product review
  - Class: Is it a fake review?

- Attributes: words and header fields of an e-mail
  - Class: Is it a spam e-mail?

Classification Applications – Examples

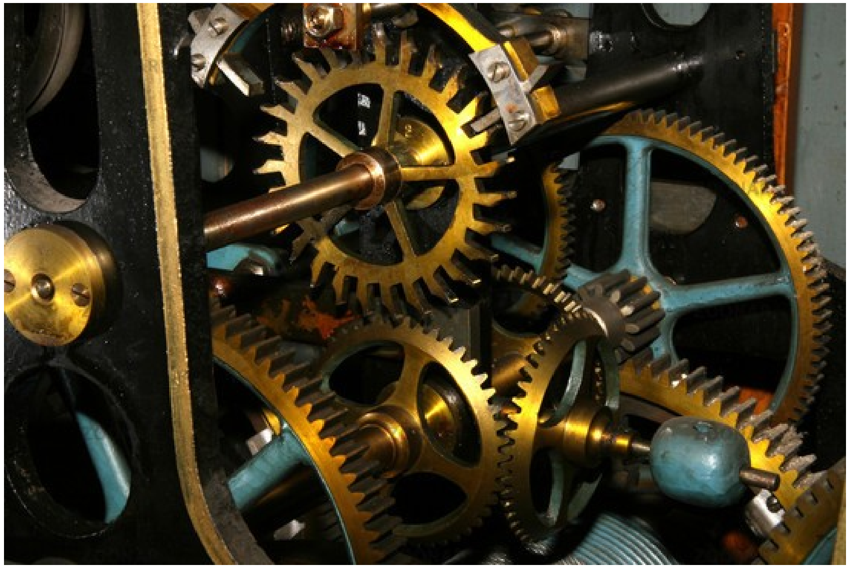
- A controversial example
  - Class: whether you are searched by the police
  - Class: whether you are selected at the airport for an extra check



http://lubbockonline.com/stories/030609/loc_405504016.shtml

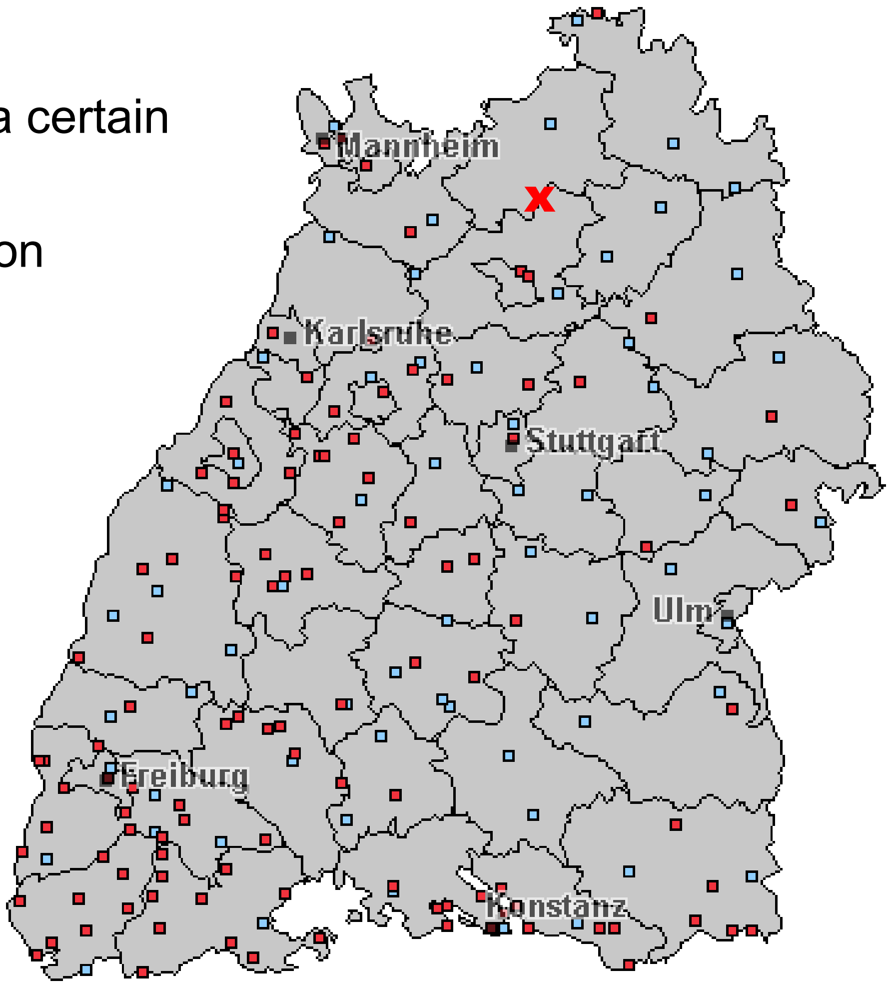
Classification Algorithms

- Recap:
  - we give the computer a set of labeled examples
  - the computer learns to classify new (unlabeled) examples
- How does that work?



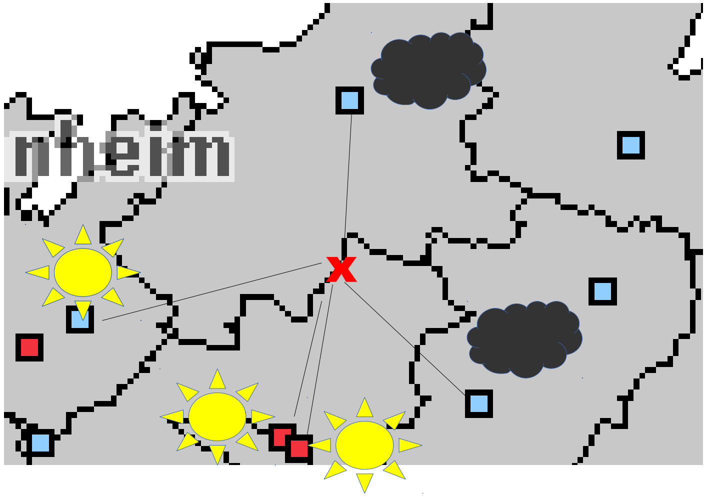
k Nearest Neighbors

- Problem
  - find out what the weather is in a certain place
  - where there is no weather station
  - how could you do that?



k Nearest Neighbors

- Idea: use the average of the nearest stations
- Example:
  - 3x sunny
  - 2x cloudy
  - result: sunny
- Approach is called
  - “k nearest neighbors”
  - where k is the number of neighbors to consider
  - in the example: k=5
  - in the example: “near” denotes geographical proximity



k Nearest Neighbors

- Further examples:
- Is a customer going to buy a product?

  → have similar customers bought that product?
- What party are you going to vote for?

  → what party do your (closest) friends/family members vote for?
- Is a film going to win an oscar?

  → have similar films won an oscar?

- and so on...

Experiment

- Trying to predict: do you want to watch "Ad Astra" (coming to cinemas tomorrow)?

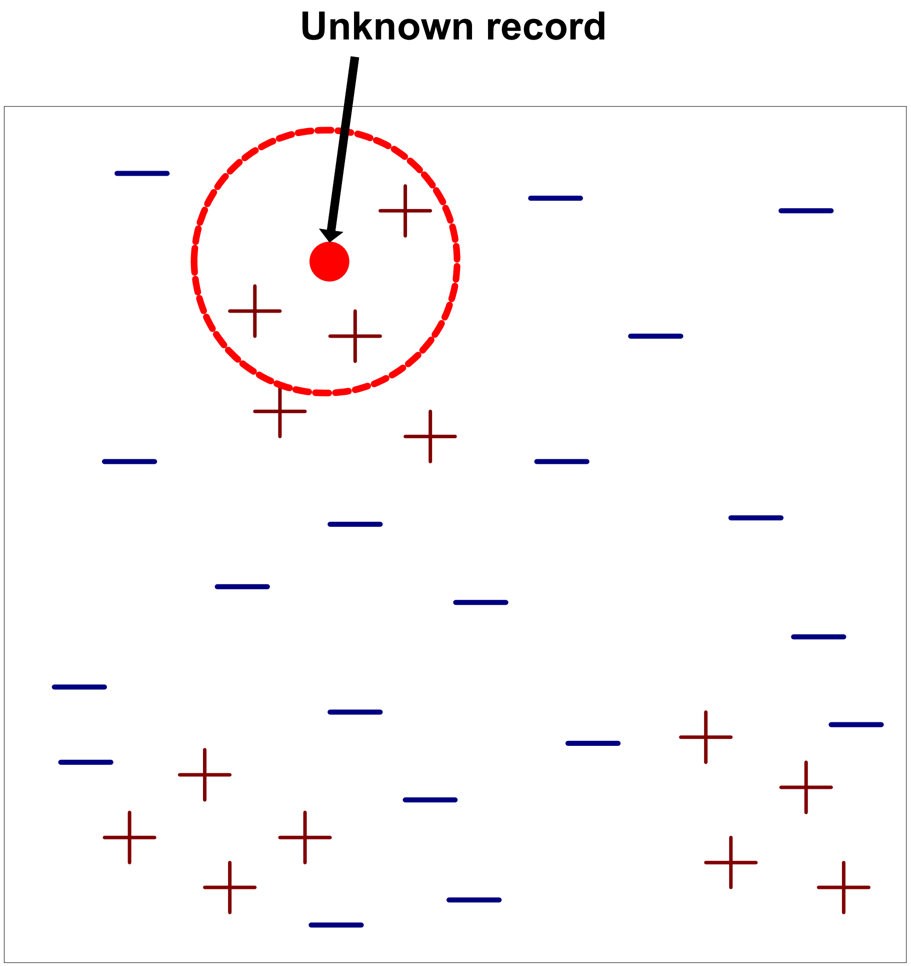
- Binary attributes: have you watched these 2019 films?
  1) Replicas
  2) Lego Movie 2
  3) Captain Marvel
  4) The Kid
  5) Shazam!
  6) Long Shot
  7) Dark Phoenix
  8) Secret Life of Pets 2
  9) Angry Birds Movie 2

Recap: Similarity and Distance

- k *Nearest* Neighbors
  - requires a notion of similarity (i.e., what is "near"?)
- Review: similarity measures for clustering
  - similarity of individual data values
  - similarity of data points
- Think about scales and normalization!

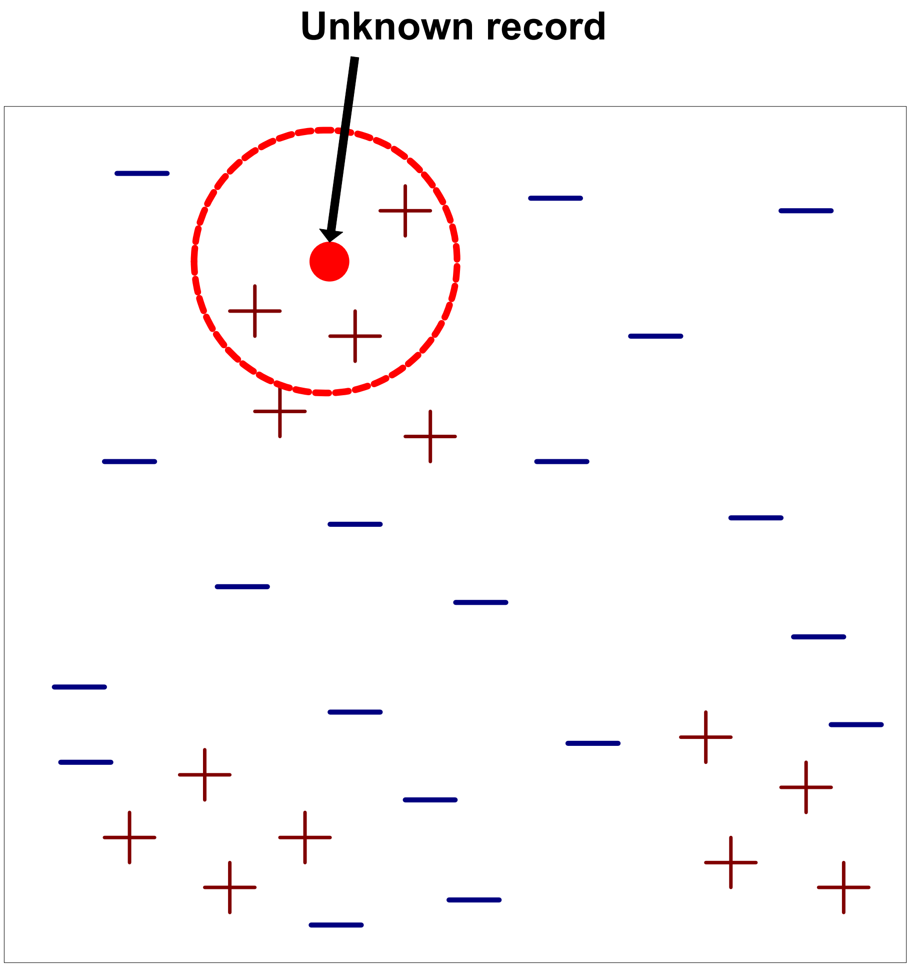
- Which similarity measure was used in our experiment?
  - we could have used different ones
  - probably with different outcomes

Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - A distance metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve

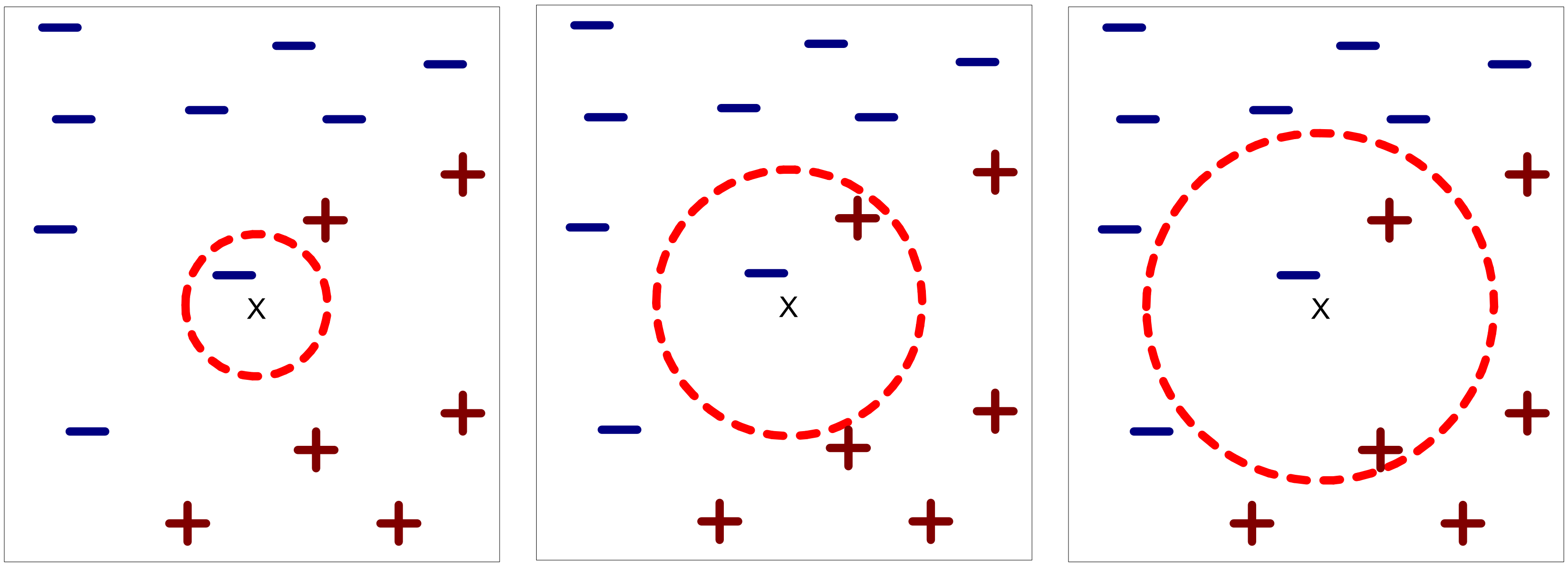
Nearest-Neighbor Classifiers



- To classify an unknown record:
  - Compute distance to each training record
  - Identify k nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record
    - by taking majority vote
    - by weighing the vote according to distance

Definition of the k Nearest Neighbors

The k nearest neighbors of a record x are data points that have the k smallest distance to x.

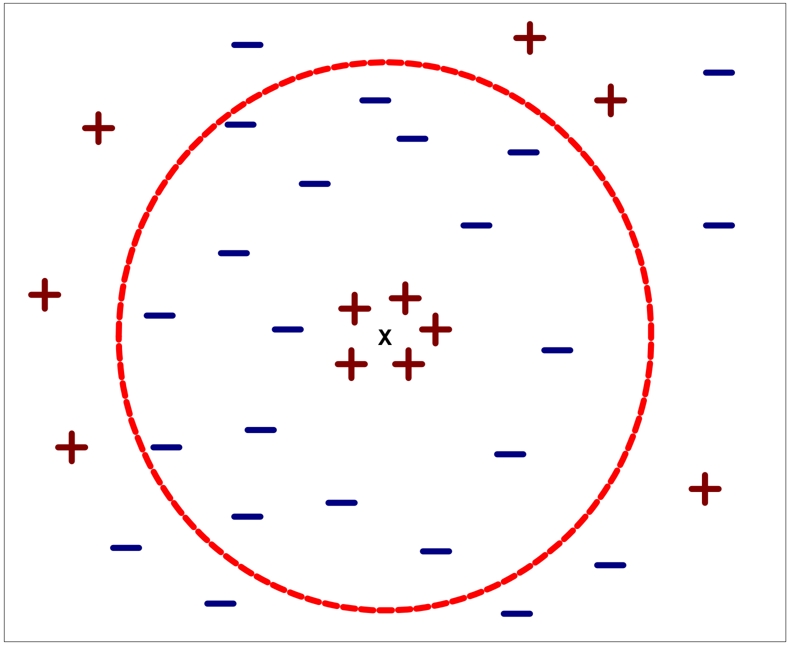


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Choosing a Good Value for k

– If k is too small, sensitive to noise points

– If k is too large, neighborhood may include points from other classes



– Rule of thumb: Test k values between 1 and 10.

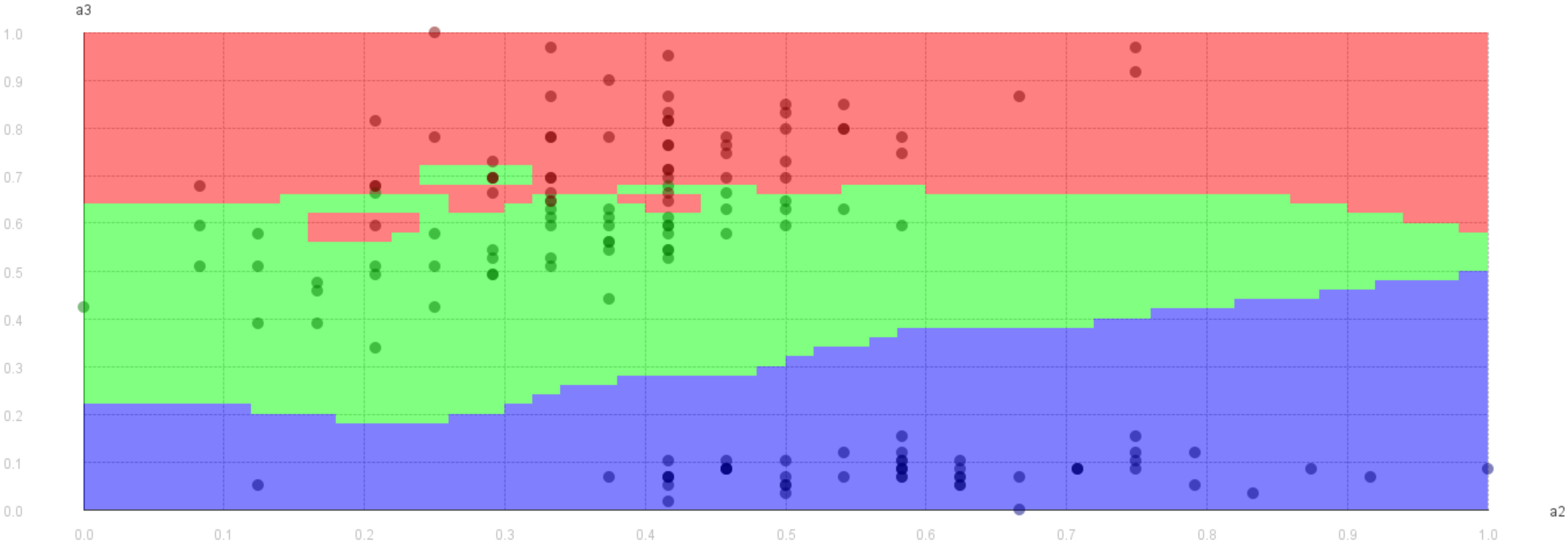
Discussion of K-Nearest Neighbor

- Often very accurate
- … but slow as training data needs to be searched

- Can handle decision boundaries which are not parallel to the axes
- Assumes all attributes are equally important
  - Remedy: Attribute selection or using attribute weights

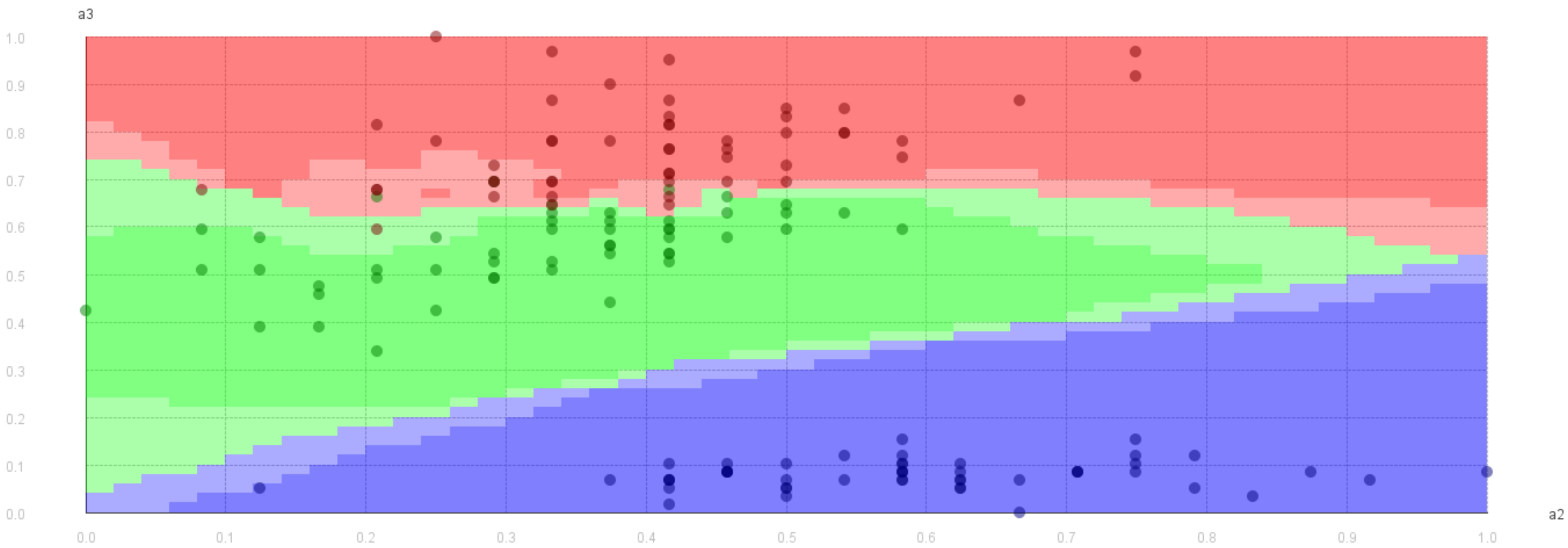
Decision Boundaries of a k-NN Classifier

- k=1
- Single noise points have influence on model

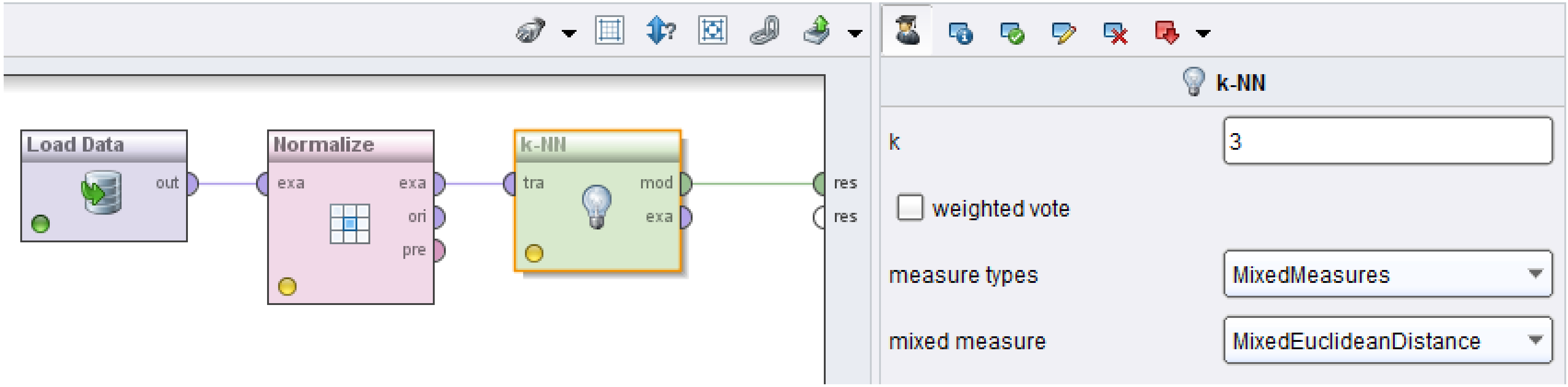


Decision Boundaries of a k-NN Classifier

- k=3
- Boundaries become smoother
- Influence of noise points is reduced

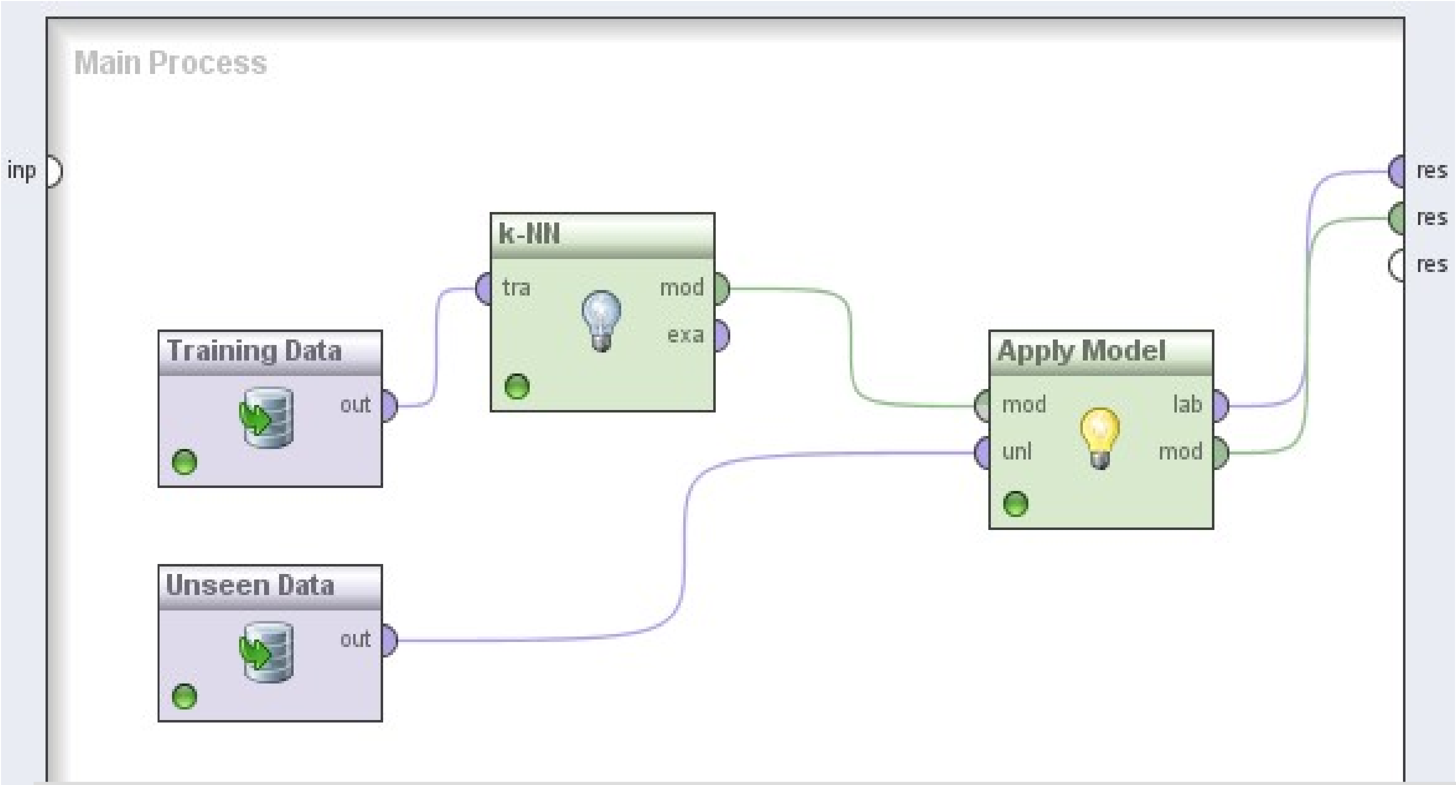


KNN in RapidMiner & Python




```
scaler = MinMaxScaler()
features_norm = scaler.fit_transform(features)
model = KNeighborsClassifier(n_neighbors=3)
model.fit(features_norm,label)
```

Applying the Model




```
test_norm = scaler.transform(test)
model.predict(test_norm)
```

Contrast: Nearest Centroids

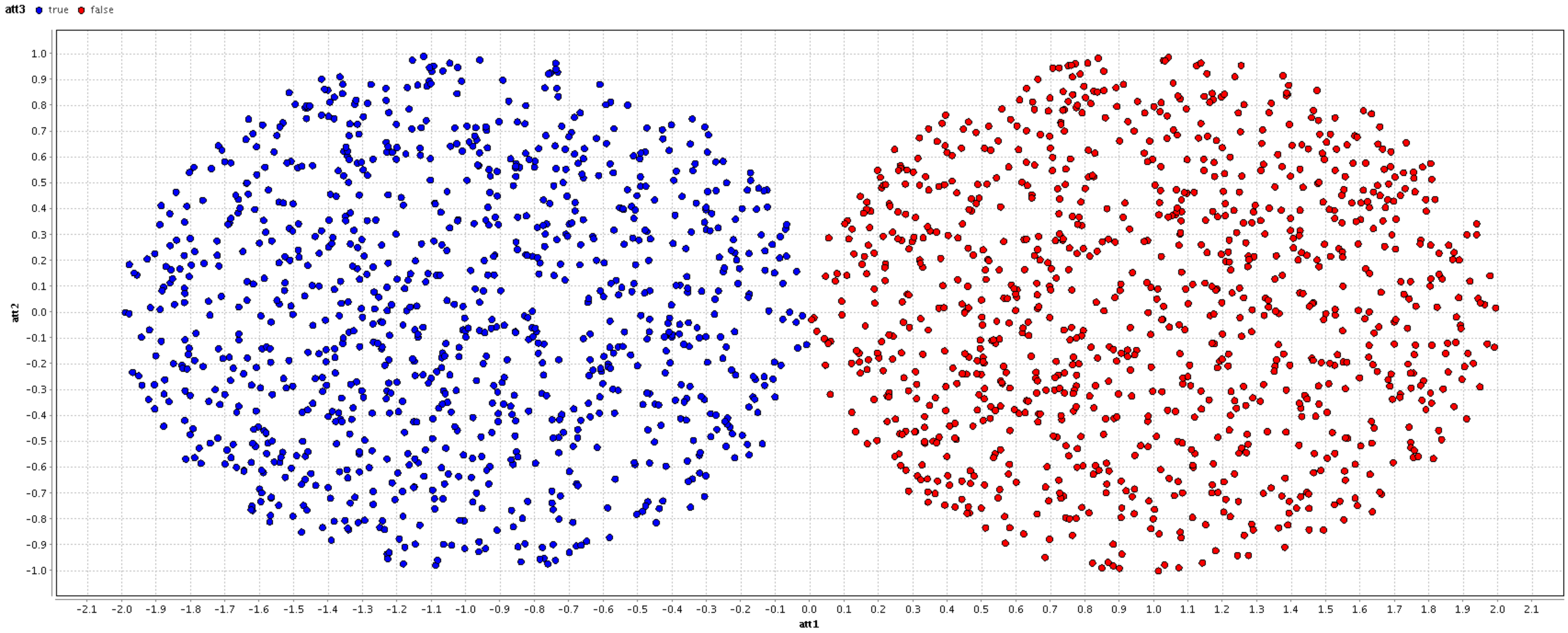
- a.k.a. Rocchio classifier
- Training: compute centroid for each class
  - center of all points of that class
  - like: centroid for a cluster
- Classification:
  - assign each data point to nearest centroid

Sounds pretty much just like k-NN, huh?

- RapidMiner:
  - available in Mannheim RapidMiner Toolbox Extension
- Python:
  - scikit_learn.neighbors.NearestCentroid

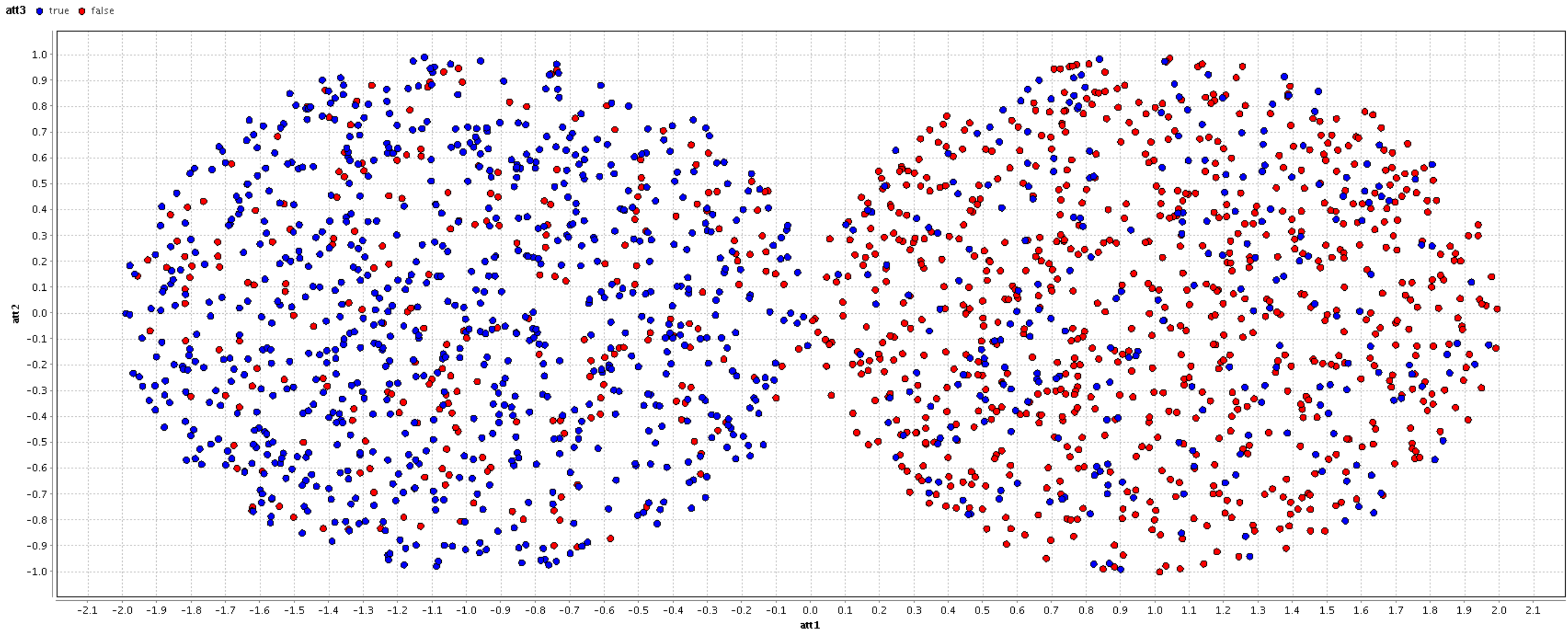
k-NN vs. Nearest Centroid

- Basic problem: two circles
  - Both k-NN and Nearest Centroid are rather perfect



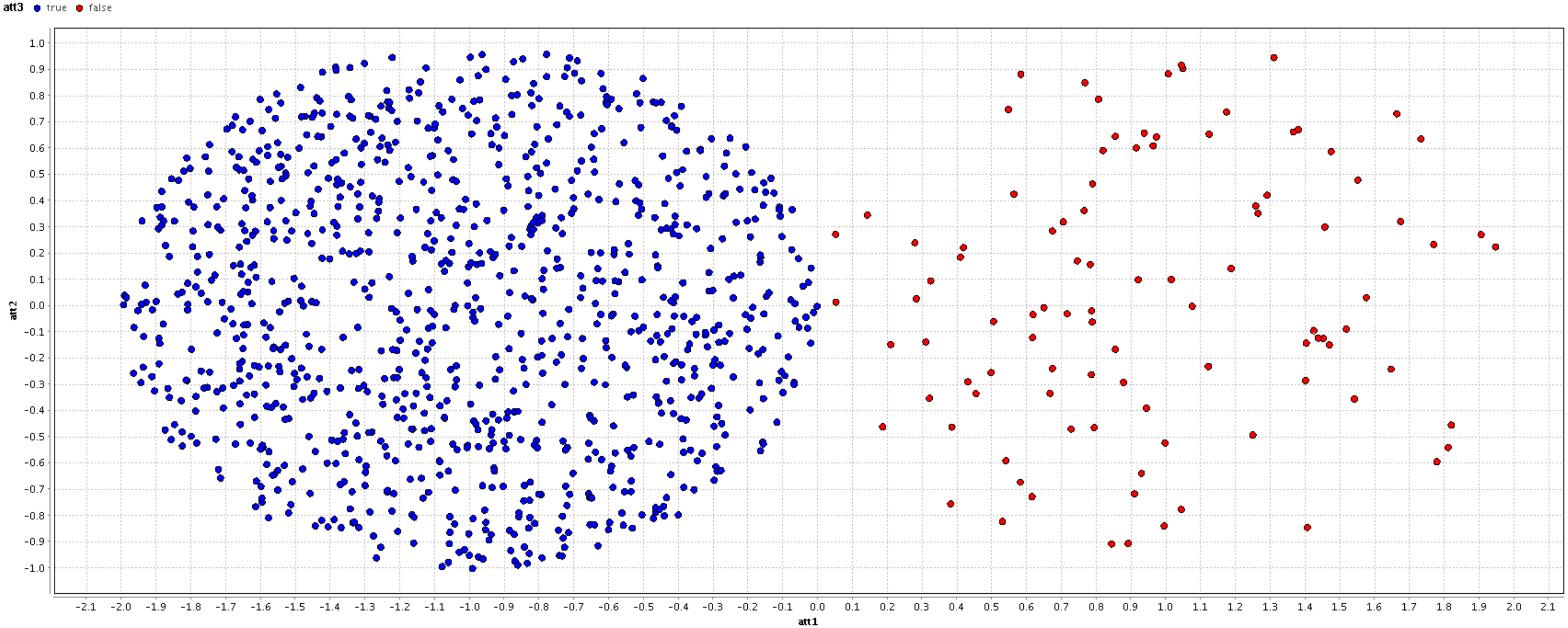
k-NN vs. Nearest Centroid

- Some data points are mislabeled
  - k-NN loses performance
  - Nearest Centroid is stable



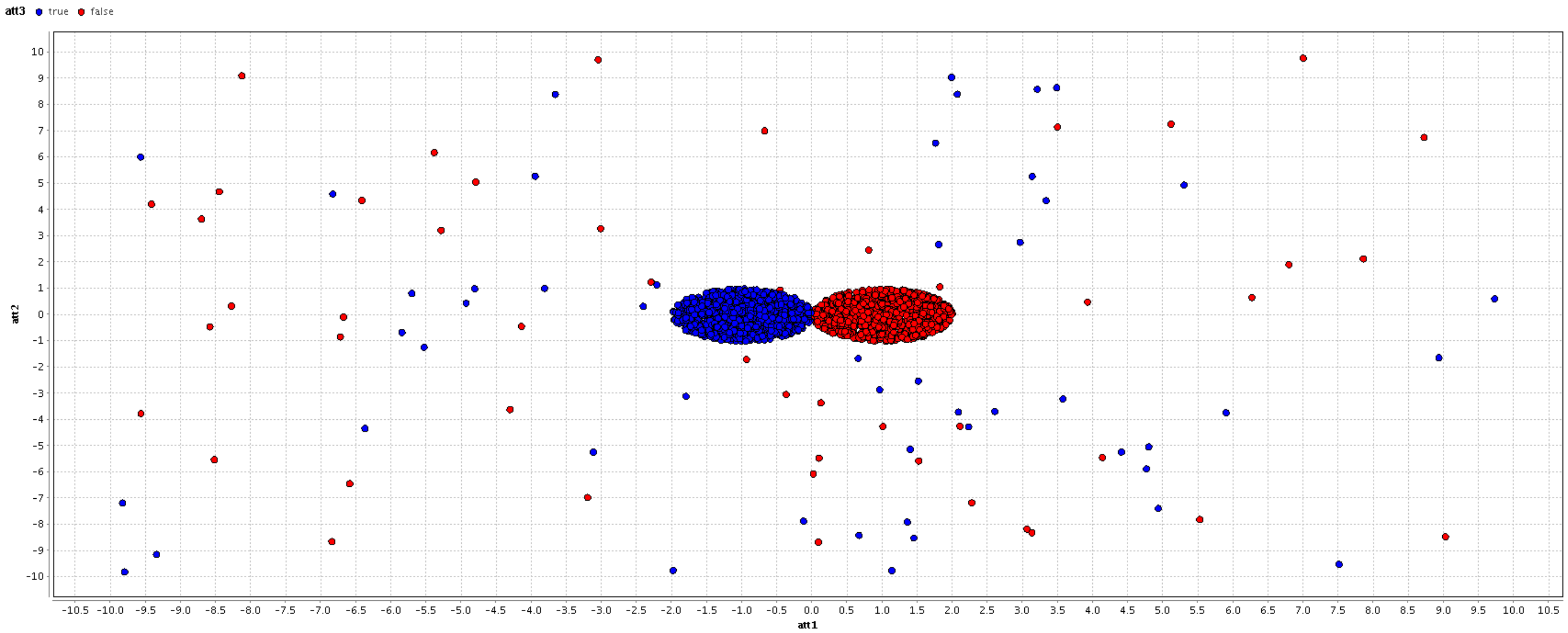
k-NN vs. Nearest Centroid

- One class is significantly smaller than the other
  - k-NN loses performance
  - Nearest Centroid is stable



k-NN vs. Nearest Centroid

- Outliers are contained in the dataset
  - k-NN is stable
  - Nearest Centroid loses performance



k-NN vs. Nearest Centroid

- k-NN
  - slow at classification time (linear in number of data points)
  - requires much memory (storing all data points)
  - robust to outliers
- Nearest Centroid
  - fast at classification time (linear in number of classes)
  - requires only little memory (storing only the centroids)
  - robust to label noise
  - robust to class imbalance

- Which classifier is better?
  - that strongly depends on the problem at hand!

Bayes Classifier

- Based on Bayes Theorem
- Thomas Bayes (1701-1761)
  - British mathematician and priest
  - tried to formally prove the existence of God
- Bayes Theorem
  - important theorem in probability theory
  - was only published after Bayes' death

Conditional Probability and Bayes Theorem

- A probabilistic framework for solving classification problems
- Conditional Probability:

$$P(C|A) = \frac{P(A,C)}{P(A)}$$

$$P(A|C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem:

$$P(C|A) = \frac{P(A|C)P(C)}{P(A)}$$

Conditional Probability and Bayes Theorem

- Bayes Theorem
  - Computes one conditional probability P(C|A) out of another P(A|C)
  - given that the base probabilities P(A) and P(C) are known

- Useful in situations where P(C|A) is unknown
  - while P(A|C), P(A) and P(C) are known or easy to determine/estimate

- Example:
  - Given a symptom, what's the probability that I have a certain disease?

Example of Bayes Theorem

- ELISA Test
  - the most common test for HIV
- Numbers:
  - If you're infected, ELISA shows a positive result with p=99.9%
  - If you're not infected, ELISA shows a negative result with p=99.5%

- Assume you have a positive test
  - What's the probability that you're infected with HIV?
- Make a guess!

Example of Bayes Theorem

- We want to know P(HIV|pos)
  - Bayes theorem:

$$P(HIV|pos)=\frac{P(pos|HIV)P(HIV)}{P(pos)}$$

0.1% in Germany

- We still need P(pos)
  - the probability of a positive test

$$\begin{aligned}P(pos)&=P(pos|HIV\vee\neg HIV)\\&=P(pos|HIV)\cdot P(HIV)+P(pos|\neg HIV)\cdot P(\neg HIV)\end{aligned}$$

- Putting the pieces together:

$$P(HIV|pos)=\frac{P(pos|HIV)P(HIV)}{P(pos|HIV)\cdot P(HIV)+P(pos|\neg HIV)\cdot P(\neg HIV)}$$

Example of Bayes Theorem

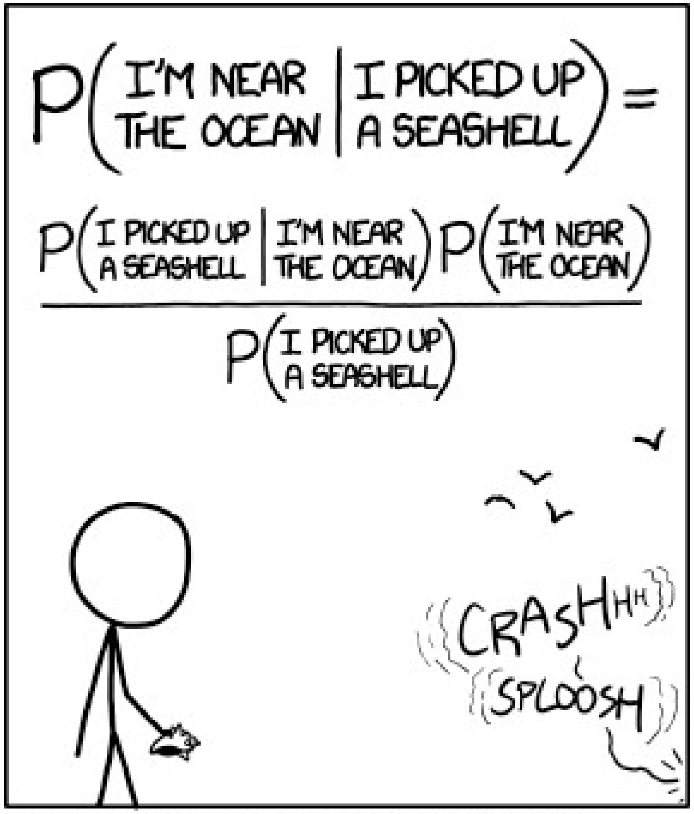
- Now: numbers

$$P(HIV \mid pos) = \frac{P(pos \mid HIV)P(HIV)}{P(pos \mid HIV) \cdot P(HIV) + P(pos \mid \neg HIV) \cdot P(\neg HIV)}$$

$$= \frac{0.999 \cdot 0.001}{0.999 \cdot 0.001 + 0.005 \cdot 0.999} = 0.167$$

- That means:
  - at more than 80% probability, you are still healthy, given a positive test!
- Reason:
  - low overall apriori probability of being HIV positive

Example of Bayes' Theorem



STATISTICALLY SPEAKING, IF YOU PICK UP A SEASHELL AND *DON'T* HOLD IT TO YOUR EAR, YOU CAN PROBABLY HEAR THE OCEAN.

http://xkcd.com/1236/

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes P(C| A_1, A_2, …,A_n)
- Can we estimate P(C| A_1, A_2,…,A_n) directly from the data?

Bayesian Classifiers

- Approach:
  - compute the probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

$$\boxed{P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}}$$

  - Choose value of C that maximizes $P(C \mid A_1, A_2, \ldots, A_n)$
  - Equivalent to choosing value of C that maximizes $P(A_1, A_2, \ldots, A_n|C)\ P(C)$
- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n | C_j) = P(A_1| C_j) P(A_2| C_j)\ldots P(A_n| C_j)$
  - Can estimate $P(A_i| C_j)$ for all A_i and C_j
  - New point is classified to C_j if $P(C_j) \Pi P(A_i| C_j)$ is maximal

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
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

- Class: $P(C) = N_c/N$
  - e.g., $P(No) = 7/10$, $P(Yes) = 3/10$
- For discrete attributes:

  $P(A_i \mid C_k) = |A_{ik}|/ N_c$

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k
  - Examples:

    P(Status=Married|No) = 4/7
    P(Refund=Yes|Yes)=0

How to Estimate Probabilities from Data?

- For continuous attributes:
  - Discretize the range into bins
    - one binary attribute per bin
    - violates independence assumption
  - Two-way split: (A < v) or (A > v)
    - choose only one of the two splits as new attribute
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
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

- Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

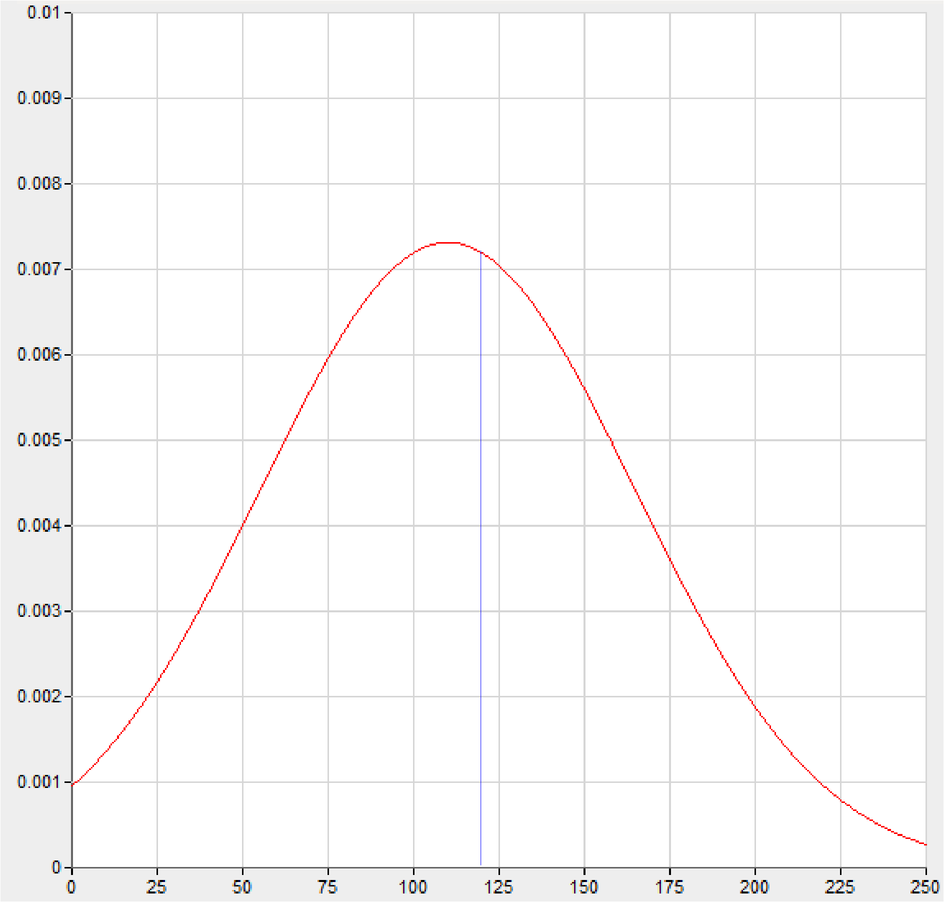
  – One for each (A_i,c_i) pair

- For (Income, Class=No):
  – If Class=No
    - sample mean = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

How to Estimate Probabilities from Data?

- Example visualization:
  - normal distribution
  - mean = 110
  - variance = 2975
- P(Income=120|No) = 0.0072



Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No) $\times$ P(Married| Class=No) $\times$ P(Income=120K| Class=No) = 4/7 $\times$ 4/7 $\times$ 0.0072 = 0.0024
- P(X|Class=Yes) = P(Refund=No| Class=Yes) $\times$ P(Married| Class=Yes) $\times$ P(Income=120K| Class=Yes) = 1 $\times$ 0 $\times$ (1.2 $\times$ 10^{-9}) = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)
=> Class = No

Handling missing values

- Missing values may occur in training and classification examples
- Training: Instance is not included in frequency count for attribute value-class combination.
- Classification: Attribute will be omitted from calculation.
- Example:

Tid	Refund	Marital Status	Taxable Income	Evade
15	No	?	120k	?

Likelihood of "yes" = $1 * (1.2 * 10^{-9}) = 1.2 * 10^{-9}$

Likelihood of "no" = $4/7 * 0.0072 = 0.0041$

From Likelihoods to Probabilities

- A person can either evade or not
  - so why do the likelihoods not add up to 1?
- Recap: $$\boxed{P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}}$$

  We have ignored the denominator so far!
  - however, it is the same for all classes
  - so we can simply normalize to 1:

Likelihood of "yes" = 1 * (1.2 * 10^{-9}) = 1.2 * 10^{-9}

Likelihood of "no" = 4/7 * 0.0072 = 0.0041

P("yes") = 1.2 * 10^{-9} / (1.2 * 10^{-9} + 0.0041) = 0.0000003

P("no") = 0.0041 / (1.2 * 10^{-9} + 0.0041) = 0.9999997

Zero Frequency Problem

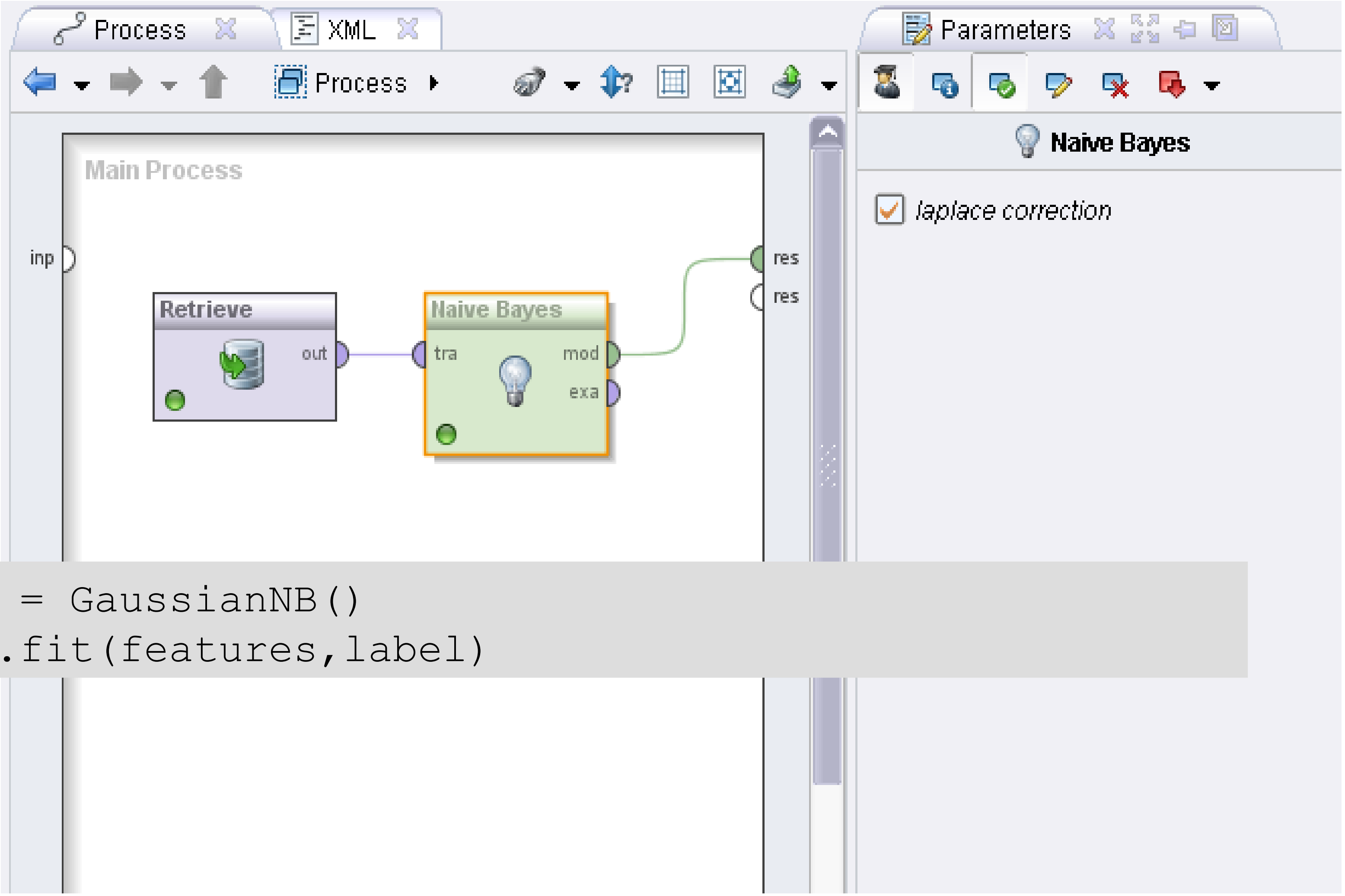
- If one of the conditional probabilities is zero, then the entire expression becomes zero
- And it is not unlikely that an exactly same data point has not yet been observed
- Probability estimation:

$$\text{Original: } P(A_i|C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace: } P(A_i|C) = \frac{N_{ic}+1}{N_c+c}$$

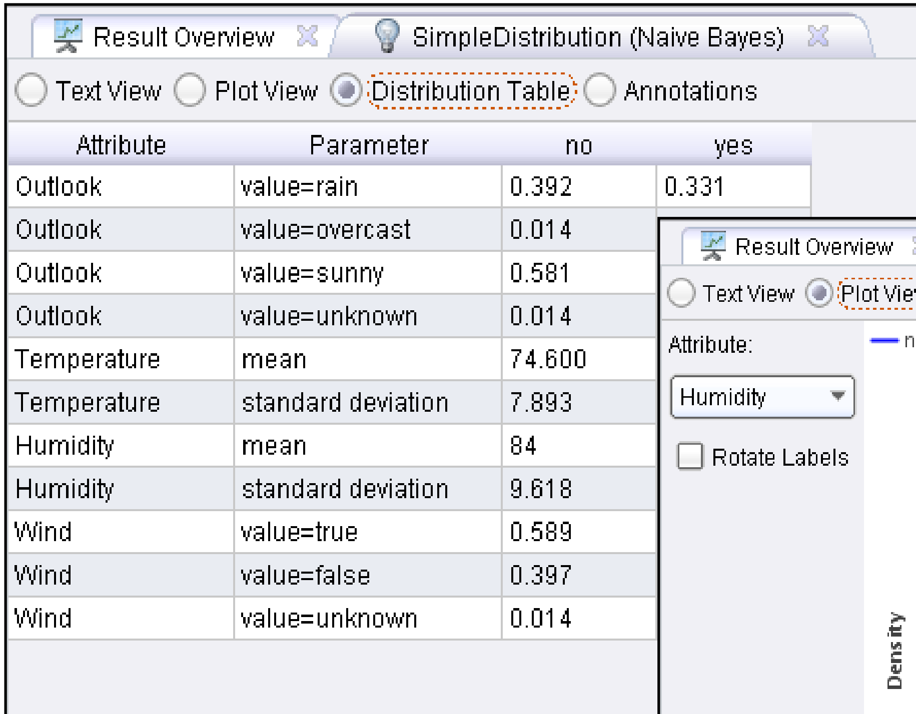
c: number of classes

Naïve Bayes in RapidMiner & Python

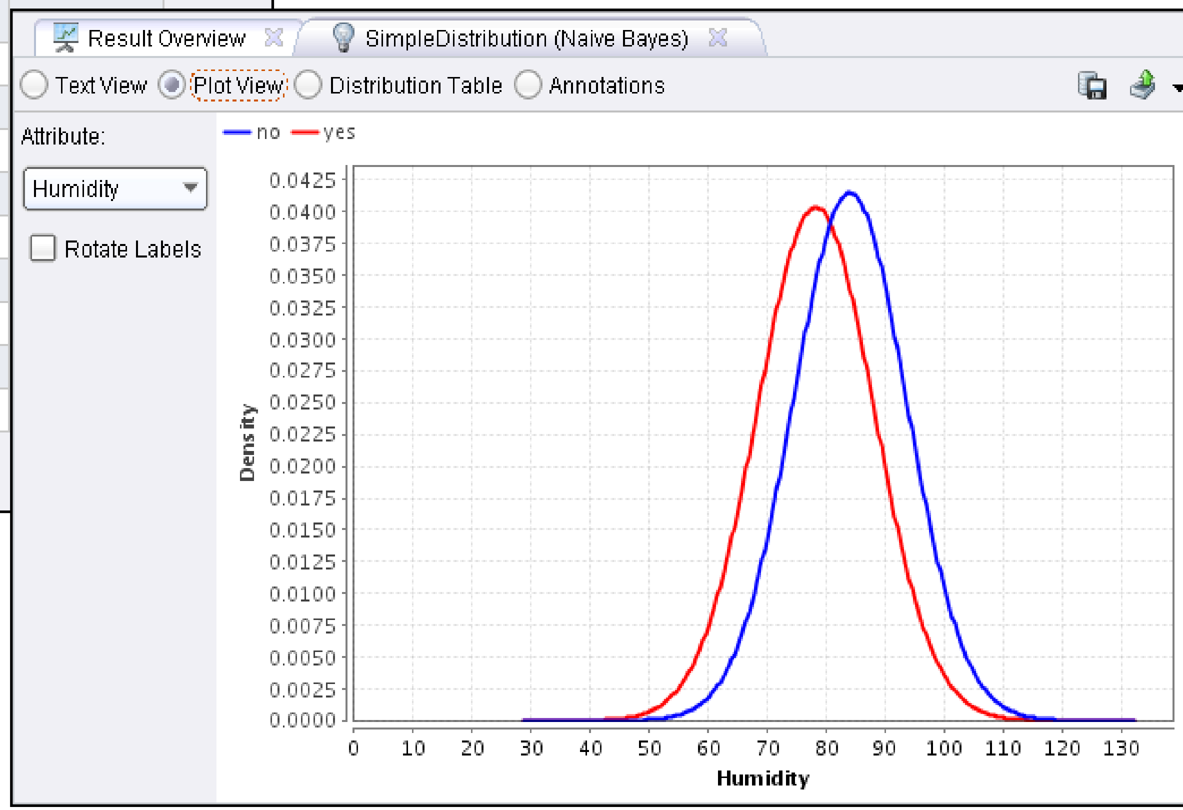



```
model = GaussianNB()
model.fit(features,label)
```

Anatomy of a Naïve Bayes Model



Attribute	Parameter	no	yes
Outlook	value=rain	0.392	0.331
Outlook	value=overcast	0.014	
Outlook	value=sunny	0.581	
Outlook	value=unknown	0.014	
Temperature	mean	74.600	
Temperature	standard deviation	7.893	
Humidity	mean	84	
Humidity	standard deviation	9.618	
Wind	value=true	0.589	
Wind	value=false	0.397	
Wind	value=unknown	0.014	



Using Conditional Probabilities for Naïve Bayes

Result Overview | ExampleSet (Retrieve Golf-Testset)

Data View | Meta Data View | Plot View | Advanced Charts | Annotations

ExampleSet (14 examples, 4 special attributes, 4 regular attributes) View Filter (14 / 14):

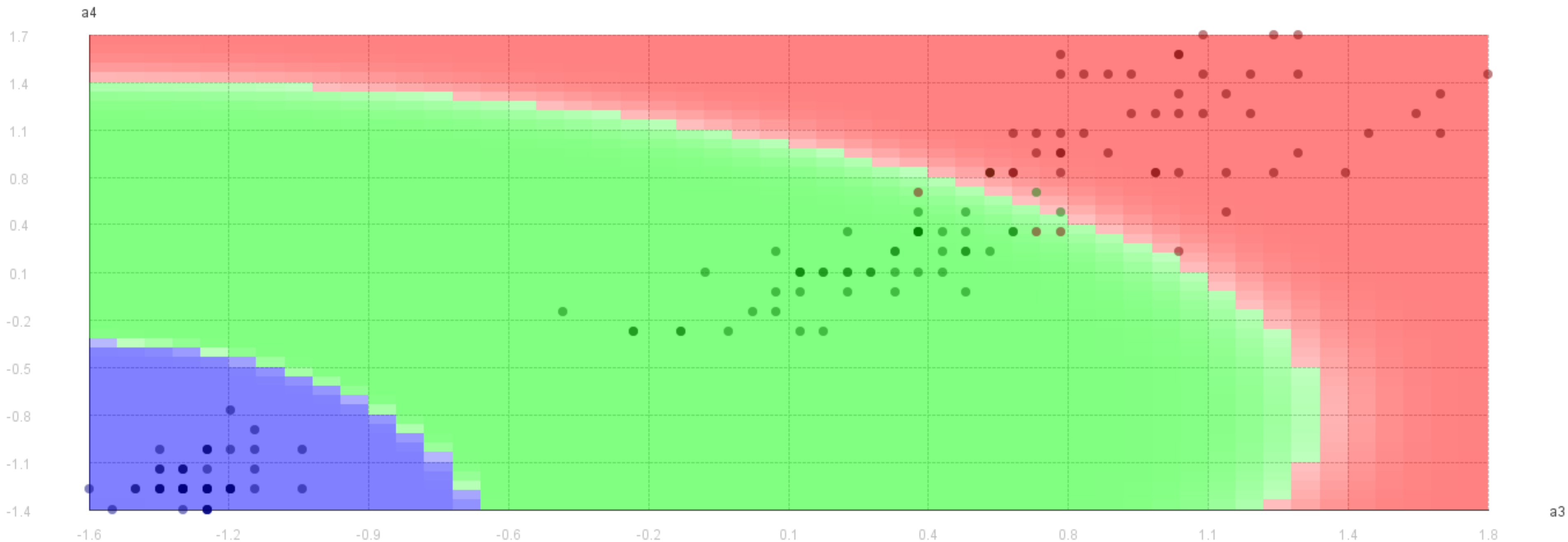
Row No.	Play	confidence(no)	confidence(yes)	prediction(Play)	Outlook	Temperature	Humidity	Wind
1	yes	0.711	0.289	no	sunny	85	85	false
2	no	0.058	0.942	yes	overcast	80	90	true
3	yes	0.014	0.986	yes	overcast	83	78	false
4	yes	0.412	0.588	yes	rain	70	96	false
5	yes	0.460	0.540	yes	rain	68	80	true
6	no	0.336	0.664	yes	rain	65	70	true
7	yes	0.010	0.990	[illegible]	[illegible]	[illegible]	[illegible]	true
8	no	0.596	0.404	no	[illegible]	[illegible]	[illegible]	false
9	yes	0.248	0.752	yes	sunny	69	70	false
10	no	0.407	0.593	yes	sunny	75	80	false
11	yes	0.496	0.504	[illegible]	[illegible]	[illegible]	[illegible]	true
12	yes	0.038	0.962	yes	[illegible]	[illegible]	[illegible]	true
13	no	0.027	0.973	yes	overcast	81	75	true
14	yes	0.453	0.547	yes	rain	71	80	true

classifier is quite sure

classifier is not sure

Decision Boundary of Naive Bayes Classifier

- Usually larger coherent areas
- Soft margins with uncertain regions
- Arbitrary (often curved) shapes



Naïve Bayes (Summary)

- Robust to isolated noise points
  - they have a small impact on the probabilities
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)

Why *Naïve* Bayes?

- Recap:
  - we assume that all the attributes are independent

- This does not hold for many real world datasets
  - e.g., persons: sex, weight, height
  - e.g., cars: weight, fuel consumption
  - e.g., countries: population, area, GDP
  - e.g., food: ingredients
  - e.g., text: word occurrences (“Donald”, “Trump”, “Duck”)
  - ...

Naïve Bayes Discussion

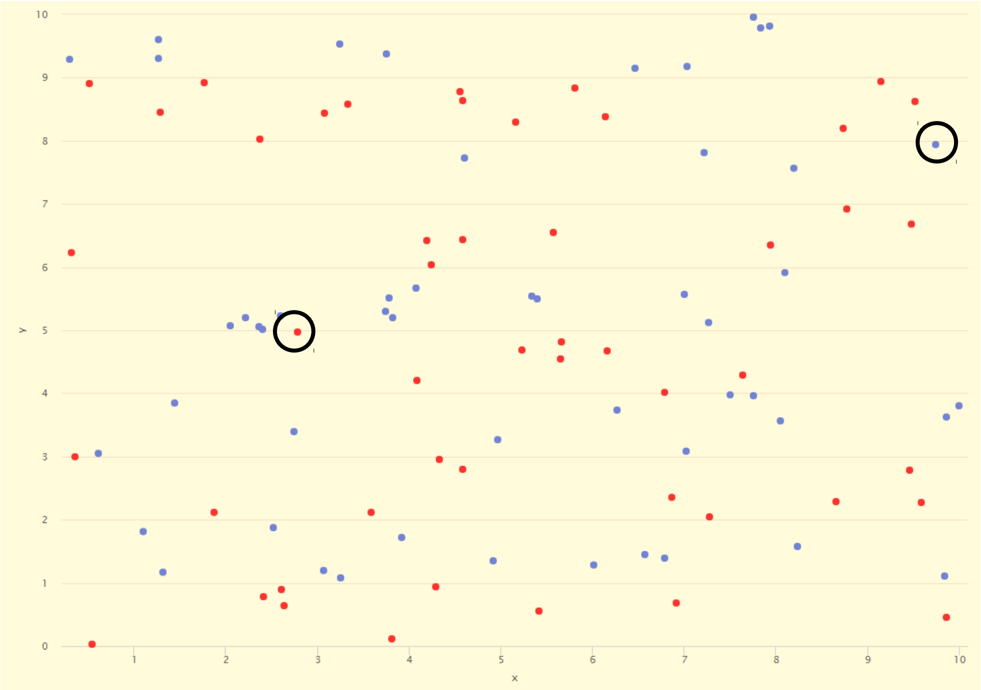
- Naïve Bayes works surprisingly well.
  - even if independence assumption is clearly violated
  - Classification doesn't require accurate probability estimates as long as maximum probability is assigned to correct class
- However: Adding too many redundant attributes will cause problems
  - Solution: Select attribute subset as Naïve Bayes often works as well or better with just a fraction of all attributes.
- Technical advantages:
  - Learning Naïve Bayes classifiers is computationally cheap as probabilities can be estimated doing one pass over the training data
  - Storing the probabilities does not require a lot of memory

Redundant Variables

- Consider two variables which are perfectly correlated
  - i.e., one is redundant
  - e.g.: a measurement in different units
- Violate independence assumption in Naive Bayes
  - Can, at large scale, skew the result
  - Consider, e.g., a price attribute in 20 currencies
    $\rightarrow$ price variable gets 20 times more influence
- May also skew the distance measures in k-NN
  - But the effect is not as drastic
  - Depends on the distance measure used

Irrelevant Variables

- Consider a random variable x, and two classes A and B
  - For Naive Bayes: p(x=v|A) = p(x=v|B) for any value v
  - Since it is random, it does not depend on the class variable
  - The overall result does not change
- For kNN:



Comparison kNN and Naïve Bayes

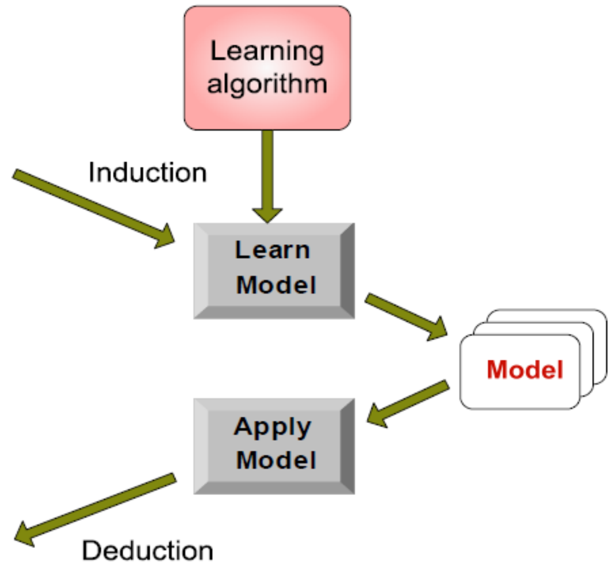
- Computation
  - Naïve Bayes is often faster
- Naïve Bayes uses *all* data points
  - Naive Bayes is less sensitive to label noise
  - k-NN is less sensitive to outliers
- *Redundant* attributes
  - are less problematic for kNN
- *Irrelevant* attributes
  - are less problematic for Naïve Bayes
  - attribute values equally distributed across classes
    → same factor for each class
- In both cases
  - attribute pre-selection makes sense (see Data Mining II)

Lazy vs. Eager Learning

- k-NN, and Naïve Bayes are all “lazy” methods
- They do not build an explicit model!
  - “learning” is only performed on demand for unseen records
- Nearest Centroid is a simple “eager” method

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



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	?

Unseen Records

Lazy vs. Eager Learning

- We have seen three of the most common techniques for lazy learning
  - k nearest neighbors
  - Naïve Bayes
- ...and a very simple technique for eager learning
  - Nearest Centroids
- We will see more eager learning in the next lectures
  - where explicit models are built
  - e.g., decision trees
  - e.g., rule sets

Questions?