Classification

Basic algorithms

Basic classification algorithms

- Task:
  - Build a model by using known data
    (a classifier for classifying new "unseen" examples)
  - The data that we used for building our model is called the **TRAINING SET**
- Supervised learning:
  - the class for the training set examples is known
- You will learn about the following classifiers:
  - ZeroR (zero rules = no rules)
  - OneR (one rule)
  - Naïve Bayes

Again, you will learn about …

- ZeroR (0R, zero rule or "no rules")
- OneR (1R, one rule)
- Naïve Bayes

ZeroR

- **The ZeroR algorithm**:
  1. Count the examples for each class value
  2. Find the most frequent class value
  3. Predict the majority class
- In simpler terms:
  – Always predict the most frequent/majority class
- **Error**: 1 – P(majority class)
- Example:
  – Weather forecasting (prediction):
    - Given: data about weather for the previous year – mostly cloudy
    - Always predict cloudy weather

ZeroR – the "weather" data set

Outlook	**Temp**	**Humidity**	**Windy**	**Play**
Sunny	Hot	High	False	**No**
Sunny	Hot	High	True	**No**
Overcast	Hot	High	False	**Yes**
Rainy	Mild	High	False	**Yes**
Rainy	Cool	Normal	False	**Yes**
Rainy	Cool	Normal	True	**No**
Overcast	Cool	Normal	True	**Yes**
Sunny	Mild	High	False	**No**
Sunny	Cool	Normal	False	**Yes**
Rainy	Mild	Normal	False	**Yes**
Sunny	Mild	Normal	True	**Yes**
Overcast	Mild	High	True	**Yes**
Overcast	Hot	Normal	False	**Yes**
Rainy	Mild	High	True	**No**

Use of the ZeroR classifier

- ZeroR classifier:

  Majority class = **Yes**

- Error:

  9 correct, **5** incorrect classifications

  accuracy = 9/14 ≈ **64.3%** (error = 5/14 ≈ **35.7%**)

- Classify:

Sunny	Hot	High	False	Yes

Rainy	Cool	Low	True	Yes

Tornado	Freezing	100%	True	Yes

A slightly different data set …

I	D	A	B	E	F	C
438	**12.03.2040**	5	3.49	14	good	y
450	24.04.1934	3	58.48	32	bad	z
461	05.01.1989	5	47.23	12	bad	y
466	07.08.1945	1	31.40	21	good	y
467	**21.07.2028**	5	79.60	20	bad	y
469	30.04.1966	3	19.88	3	bad	w
485	28.02.2015	5	59.13	4	bad	w
514	19.03.2033	3	27.05	2	bad	x
522	13.03.2022	2	80.14	16	good	y
529	28.07.2037	4	65.02	20	bad	z
534	05.10.1986	2	99.17	13	good	z
566	20.04.1982	4	43.97	24	good	y
578	15.05.2012	2	13.02	2	good	y
600	30.11.1943	1	32.43	10	bad	y

… why did ZeroR chose y?

Class	Frequency
w	2
x	1
y	**5**
z	3

- ZeroR classifier:

  Majority class = **y**

- Error:

  6/11 ≈ **54.55%**

OneR

- ZeroR doesn't take into account any attribute
- **OneR** classifies based on just one attribute

- The OneR algorithm builds a one-level decision tree
- How?
  - Build a one-level decision tree for each attribute
  - Calculate the error of each decision tree
  - Choose the one decision tree with lowest error

OneR – procedure

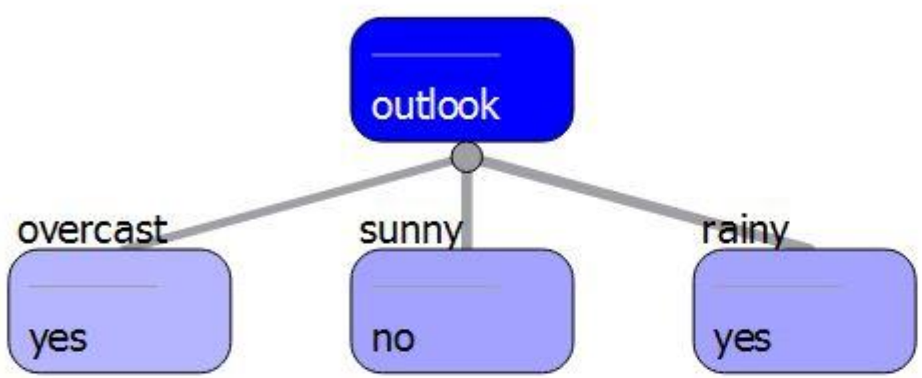
- For each attribute:
  - For each attribute value:
    - Count the class frequencies
    - Determine the most frequent class value
    - Make a rule predicting the most frequent class value for the current attribute
  - Calculate the error
  - Sum up all the errors for the current attribute
- Choose the attributes with the lowest total error

OneR – the "weather" data set

Outlook	**Temp**	**Humidity**	**Windy**	**Play**
Sunny	Hot	High	False	**No**
Sunny	Hot	High	True	**No**
Overcast	Hot	High	False	**Yes**
Rainy	Mild	High	False	**Yes**
Rainy	Cool	Normal	False	**Yes**
Rainy	Cool	Normal	True	**No**
Overcast	Cool	Normal	True	**Yes**
Sunny	Mild	High	False	**No**
Sunny	Cool	Normal	False	**Yes**
Rainy	Mild	Normal	False	**Yes**
Sunny	Mild	Normal	True	**Yes**
Overcast	Mild	High	True	**Yes**
Overcast	Hot	Normal	False	**Yes**
Rainy	Mild	High	True	**No**

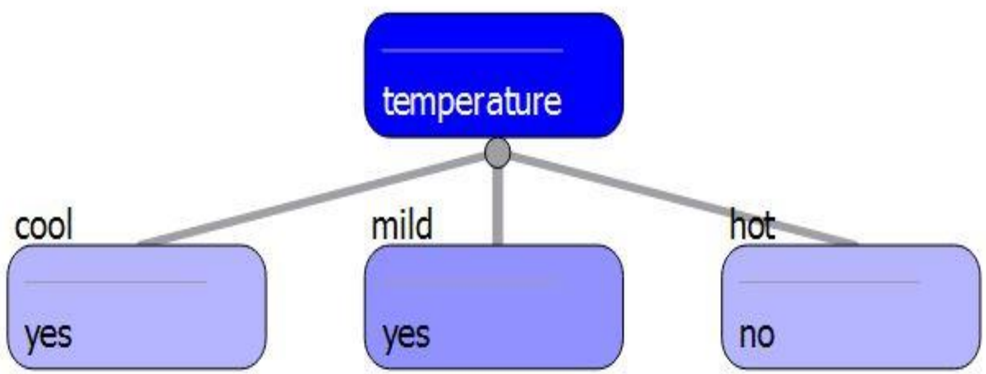
OneR – the "Outlook" attribute

Outlook \ Play	Yes	No	Error
Sunny	2	**3**	2
Overcast	**4**	0	0
Rainy	**3**	2	2
		Total error:	**4** = 4/14 ≈ **28.6%**



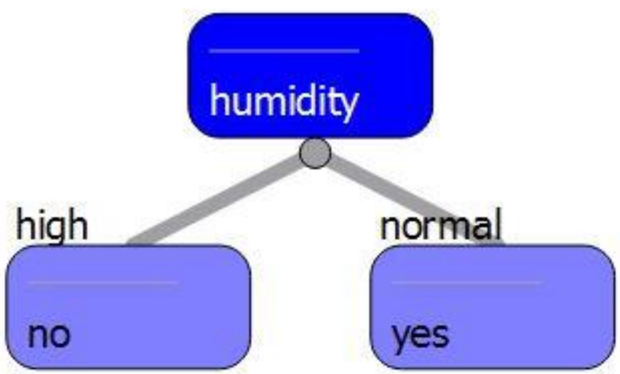
OneR – the "Temperature" attribute

Temperature \ Play	Yes	No	Error
Hot	2	**2**	2
Mild	**4**	2	2
Cool	**3**	1	1
		Total error:	**5** = 5/14 ≈ **35.7%**



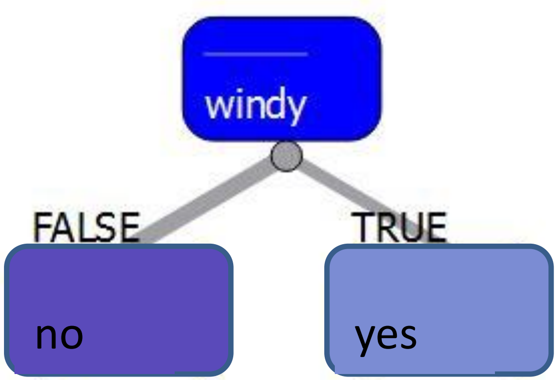
OneR – the "Humidity" attribute

Humidity \ Play	**Yes**	**No**	**Error**
High	3	**4**	3
Normal	**6**	1	1
		Total error:	**4** = 4/14 ≈ **28.6%**



OneR – the "Windy" attribute

Windy \ Play	**Yes**	**No**	**Error**
True	**6**	2	2
False	3	**3**	3
		Total error:	**5** = 5/14 ≈ **35.7%**



OneR – making predictions

- Predict the class value for these examples:
  - We have chosen **Outlook** as our "best" attribute

Sunny	Hot	High	False	**No**

Rainy	Cool	Low	True	**Yes**

Overcast	Freezing	100%	True	**Yes**

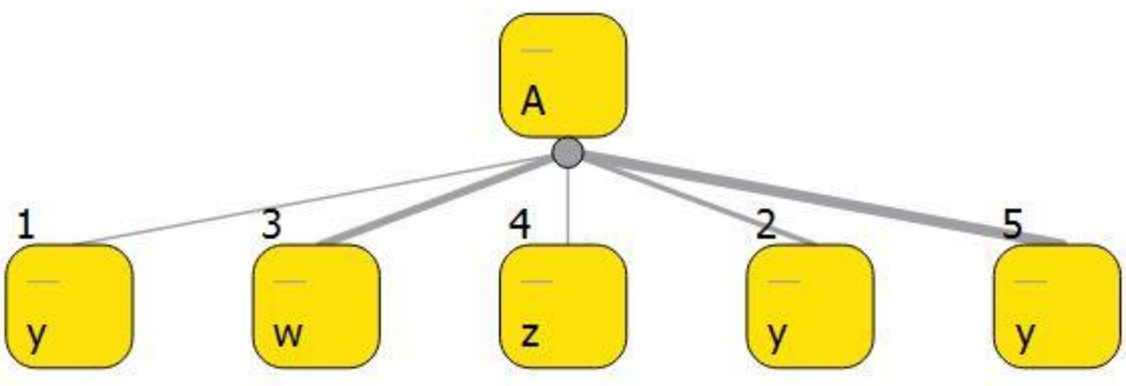
A slightly different data set again …

I	D	A	B	E	F	C
438	**12.03.2040**	5	3.49	14	good	**y**
450	24.04.1934	3	58.48	32	bad	**z**
461	05.01.1989	5	47.23	12	bad	**y**
466	07.08.1945	1	31.40	21	good	**y**
467	**21.07.2028**	5	79.60	20	bad	**y**
469	30.04.1966	3	19.88	3	bad	**w**
485	28.02.2015	5	59.13	4	bad	**w**
514	19.03.2033	3	27.05	2	bad	**x**
522	13.03.2022	2	80.14	16	good	**y**
529	28.07.2037	4	65.02	20	bad	**z**
534	05.10.1986	2	99.17	13	good	**z**

OneR – the "A" attribute

A \ C	w	x	y	z	Error
1	0	0	**1**	0	0
2	0	0	**1**	1	1
3	**1**	1	0	1	2
4	0	0	0	**1**	0
5	1	0	**3**	0	1

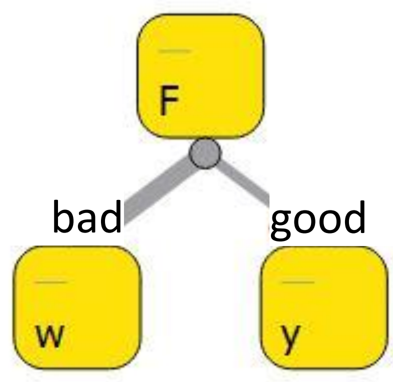
4 / 11 ≈ **36.36%**



OneR – the "F" attribute

F \ C	w	x	y	z	Error
good	0	0	**3**	1	1
bad	**2**	1	2	2	5

6 / 11 ≈ **54.55%**



OneR – making predictions

- For numeric attributes WEKA uses class-dependent discretisation
  - in our example we simply "ignored" them
- Classify the following examples (use OneR):

I	D	A	B	E	F	C
566	20.04.1982	4	43.97	24	good	z
578	15.05.2012	2	13.02	2	good	y
600	30.11.1943	1	32.43	10	bad	y

Naïve Bayes

- Uses all the attributes
  - That is not always a good choice …
    - Example: 1,000,000 attributes

- Naïve, because of its over-simplified "looking at things".

  It assumes that:
  - All attributes are "equaly important"
  - All attributes are pairwise independent

The Bayes rule

$$\Pr[H \mid E] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]}$$

H = class
E = attributes
Pr[H|E] = probability of class, given the attributes
…
Pr[E|H] = probability of attributes, given the class
Pr[H] = "a priori" probability of the class (without knowing the attributes)
Pr[E] = probability of the attributes (without knowing the class)

$$\Pr[yes \mid sunny, cool, normal, true] = \frac{\Pr[sunny, cool, normal, true \mid yes]\Pr[yes]}{\Pr[sunny, cool, normal, true]}$$

Naïveness …

- Pr[E|H] can be written as …

  $$\Pr[E|H] = \Pr[E_1|H]\Pr[E_2|H] \ldots \Pr[E_n|H]$$

- It follows that …

$$\Pr[sunny, cool, normal, true \mid yes] = \Pr[sunny \mid yes] x \Pr[cool \mid yes] x \Pr[normal \mid yes] x \Pr[true \mid yes]$$

- This, we can compute …
  - Pr[sunny|yes] … probability of sunny, while we are playing
    - 9 times we played, 2 times it was sunny → 2/9
  - Pr[cool|yes] … probability of cool, while we are playing
    - 9 times we played, 3 times it was cool → 3/9
  - …

The Bayes rule again …

… assuming the attributes are pairwise independent (a "naïve" assumption)

$$\Pr[H \mid E] = \frac{\Pr[E_1 \mid H]\Pr[E_2 \mid H]\ldots\Pr[E_n \mid H]\Pr[H]}{\Pr[E]}$$

Naïve Bayes – the "weather" data

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

… build the frequency/probability table

Outlook			Temperature			Humidity			Windy			Play	
	Yes	***No***		***Yes***	***No***		***Yes***	***No***		***Yes***	***No***	***Yes***	***No***
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Classify a new day:

Sunny	Hot	High	False	

Likelihoods:
P("**Yes**") = 2/9 x 2/9 x 3/9 x 6/9 x 9/14 ≈ **0.007**
P("**No**") = 3/5 x 2/5 x 4/5 x 2/5 x 5/14 ≈ 0.027

(Normalized) probabilities:
P("**Yes**") = 0.007 / (0.007 + 0.027) ≈ **20.5%**
P("**No**") = 0.027 / (0.007 + 0.027) ≈ **79.5%** → Play = "**No**"

… what about this day?

Overcast	Hot	High	False	

Likelihoods:
P("**Yes**") = 4/9 x 2/9 x 3/9 x 6/9 x 9/14 ≈ **0.014**
P("**No**") = 0/5 x 2/5 x 4/5 x 2/5 x 5/14 = **0**

(Normalized) probabilities:
P("**Yes**") = 0.014 / (0.014 + 0.0) = **100%** → Play = "**Yes**"
P("**No**") = 0.0 / (0.014 + 0.0) = **0%**

- Does this make sense?
  - one attribute "overrules" all the others …
  - we can handly this with the **Laplace estimate**
- Laplace estimate:
  - Add 1 to each frequency count
  - Again, compute the probabilities

… with the Laplace estimate

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	3	4	Hot	3	3	High	4	5	False	7	3	10	6
Overcast	5	1	Mild	5	3	Normal	7	2	True	4	4		
Rainy	4	3	Cool	4	2								
Sunny	3/12	4/8	Hot	3/12	3/8	High	4/11	5/7	False	7/11	3/7	10/16	6/16
Overcast	5/12	1/8	Mild	5/12	3/8	Normal	7/11	2/7	True	4/11	4/7		
Rainy	4/12	3/8	Cool	4/12	2/8								

Classify a new day:

Overcast	Hot	High	False	

Likelihoods:

P("**Yes**") = 5/12 x 3/12 x 4/11 x 7/11 x 10/16 ≈ **0.015**

P("**No**") = 1/8 x 3/8 x 5/7 x 3/7 x 6/16 ≈ **0.005**

(Normalized) probabilities:

P("**Yes**") = 0.015 / (0.015 + 0.05) ≈ **75%** → Play = "**Yes**"

P("**No**") = 0.05 / (0.015 + 0.05) ≈ **25%**

A slightly different data set again …

A	F	C
5	good	y
3	bad	z
5	bad	y
1	good	y
5	bad	y
3	bad	w
5	bad	w
3	bad	x
2	good	y
4	bad	z
2	good	z

… build the frequency/probability tables

A \ C	w	x	y	z
1	1	1	2	1
2	1	1	2	2
3	2	2	1	2
4	1	1	1	2
5	2	1	4	1

F \ C	w	x	y	z
good	1	1	4	2
bad	3	2	3	3

A \ C	w	x	y	z
1	1/7	1/6	2/10	1/8
2	1/7	1/6	2/10	2/8
3	2/7	2/6	1/10	2/8
4	1/7	1/6	1/10	2/8
5	2/7	1/6	4/10	1/8

F \ C	w	x	y	z
good	1/4	1/3	4/7	2/5
bad	3/4	2/3	3/7	3/5

C	w	x	y	z
	3	2	6	4

C	w	x	y	z
	3/15	2/15	6/15	4/15

… classify the following example

A	F	C
2	bad	?

Compute the likelihoods:
P("**w**") = 1/7 x 3/4 x 3/15 ≈ **0.021**
P("**x**") = 1/6 x 2/3 x 2/15 ≈ **0.015**
P("**y**")= 2/10 x 3/7 x 6/15 ≈ **0.034**
P("**z**") = 2/8 x 3/5 x 4/15 ≈ **0.04**

w	x	y	z
0.021	0.015	0.034	0.04

Derive the (normalized) probabilities:

19%	13.6%	30.9%	**36.4%**

Choose the highest probability and classify the example in class **z**.

What about numeric attributes?

- We have 2 options:

1. Discretize the attribute

2. Compute the mean and standard deviation
   - For each new example, compute the **probability density**
   - Assuming, the attribute values are "normally" distributed

Numeric attributes – computation

- Usual assumption: attributes have a *normal* or *Gaussian* probability distribution (given the class)
- The *probability density function* for the normal distribution is defined by two parameters:
  - *Sample mean* $\boldsymbol{\mu}$

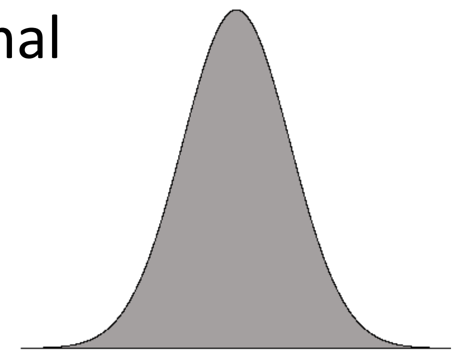
$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

  - *Standard deviation* $\boldsymbol{\sigma}$

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu)^2$$

  - Then the probability density function *f(x)* is:

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



Karl Gauss, 1777-1855

great German mathematician

Naïve Bayes – problems

- Multiple copies of the same attribute
- Dependence between the attributes

Problem: multiple attribute copies

- Assuming,
  all the attributes are equally important
- If an attribute has multiple copies, it "gets to vote" multiple times!
- Example:
  temperature in °C and in °K
  (these "total" dependencies count as copies of the attribute)

Problem: the XOR dependency

X	Y	C
0	0	False
0	1	True
1	0	True
1	1	False

X \ C	True	False
0	1/2	1/2
1	1/2	1/2

Y \ C	True	False
0	1/2	1/2
1	1/2	1/2

The probability of predicting a new example will (always) be random:

P("**true**") = 1/2 x 1/2 x 2/4 = 0.125 → **50%**

P("**false**") = 1/2 x 1/2 x 2/4 = 0.125 → **50%**

Missing values

- Naïve Bayes is not affected by missing values – it simply "leaves them out" of the calculations

Classify the new day:

?	Hot	High	False	

Likelihoods:
P("**Yes**") = ~~5/12~~ x 3/12 x 4/11 x 7/11 x 10/16 ≈ ~~**0.015**~~ ≈ **0.036**
P("**No**") = ~~1/8~~ x 3/8 x 5/7 x 3/7 x 6/16 ≈ ~~**0.005**~~ ≈ **0.043**

(Normalized) probabilities:
P("**Yes**") = 0.036 / (0.036 + 0.043) ≈ **46%**
P("**No**") = 0.043 / (0.036 + 0.043) ≈ **54%** → Play = "**No**"

What have you learned?

- ZeroR
- OneR
- Naïve Bayes