<u>Exercise 4.</u> Naïve Bayes for data with nominal attributes

Given the training data in the table below (*Buy Computer* data), predict the class of the following new example using Naïve Bayes classification: age<=30, income=medium, student=yes, credit-rating=fair

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Solution:

E= age<=30, income=medium, student=yes, credit-rating=fair

E_1 is age<=30, E2 is income=medium, student=yes, E4 is credit-rating=fair

We need to compute P(yes|E) and P(no|E) and compare them.

$$P(yes \mid E) = \frac{P(E_1 \mid yes)\,P(E_2 \mid yes)\,P(E_3 \mid yes)\,P(E_4 \mid yes)\,P(yes)}{P(E)}$$

P(yes)=9/14=0.643 P(no)=5/14=0.357

P(E1|yes)=2/9=0.222 P(E1|no)=3/5=0.6

P(E2|yes)=4/9=0.444 P(E2|no)=2/5=0.4

P(E3|yes)=6/9=0.667 P(E3|no)=1/5=0.2

P(E4|yes)=6/9=0.667 P(E4|no)=2/5=0.4

$$P(yes \mid E) = \frac{0.222\ 0.444\ 0.667\ 0.668\ 0.443}{P(E)} = \frac{0.028}{P(E)} \qquad P(no \mid E) = \frac{0.6\ 0.4\ 0.2\ 0.4\ 0.357}{P(E)} = \frac{0.007}{P(E)}$$

Hence, the Naïve Bayes classifier predicts buys_computer=yes for the new example.

<u>Exercise 5.</u> Applying Naïve Bayes to data with numerical attributes and using the Laplace correction <u>(to be done at your own time, not in class)</u>

Given the training data in the table below (*Tennis* data with some numerical attributes), predict the class of the following new example using Naïve Bayes classification:

outlook=overcast, temperature=60, humidity=62, windy=false.

Tip. You can use Excel or Matlab for the calculations of logarithm, mean and standard deviation. Matlab is installed on our undergraduate machines. The following Matlab functions can be used: `log2` – logarithm with base 2, `mean` – mean value, `std` – standard deviation. Type `help` <function name> (`e.g. help mean`) for help on how to use the functions and examples.

outlook	temperature	humidity	windy	play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

Solution:

First, we need to calculate the mean μ and standard deviation σ values for the numerical attributes.
X_i, i=1..n – the i-th measurement, n-number of measurements

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

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

μ_temp_yes=73, σ_temp_yes=6.2; μ_temp_no=74.6, σ_temp_no=8.0

μ_hum_yes=79.1, σ_temp_yes=10.2; μ_hum_no=86.2, σ_temp_no=9.7

Second, to calculate f(temperature=60|yes), f(temperature=60|no), f(humidity=62|yes) and f(humidity=62|no) using the probability density function for the normal distribution:

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

$$f(temperature=60 \mid yes)=\frac{1}{6.2\sqrt{2\pi}}e^{-\frac{(60-73)^2}{2\ (6.2)^2}}=0.071$$

$$f(temperature=60 \mid no)=\frac{1}{8\sqrt{2\pi}}e^{-\frac{(60-74.6)^2}{2\ 8^2}}=0.0094$$

$$f(humidity=62 \mid yes)=\frac{1}{10.2\sqrt{2\pi}}e^{-\frac{(62-79.1)^2}{2\ (10.2)^2}}=0.0096$$

$$f(humidity=62 \mid no)=\frac{1}{9.7\sqrt{2\pi}}e^{-\frac{(62-86.2)^2}{2\ (9.7)^2}}=0.0018$$

Third, we can calculate the probabilities for the nominal attributes:
P(yes)=9/14=0.643 P(no)=5/14=0.357

P(outlook=overcast|yes)=4/14=0.286 P(outlook=overcast|no)=0/5=0
P(windy=false|yes)=6/9=0.667 P(windy=false|no)=2/5=0.4

As P(outlook=overcast|no)=0, we need to use a Laplace estimator for the attribute outlook. We assume that the three values (sunny, overcast, rainy) are equally probable and set μ=3:

$$P(outlook=overcast \mid yes)=\frac{4+1}{9+3}=\frac{5}{12}=0.4167$$

$$P(outlook=overcast \mid no)=\frac{0+1}{5+3}=\frac{1}{8}=0.125$$

Fourth, we can calculate the final probabilities:

$$P(yes \mid E)=\frac{0.4167*0.0071*0.0096*0.667*0.643}{P(E)}=\frac{1.22*10^{-5}}{P(E)}$$

$$P(no \mid E)=\frac{0.125*0.0094*0.0018*0.4*0.357}{P(E)}=\frac{3.02*10^{-7}}{P(E)}$$

Therefore, the Naïve Bayes classifier predicts play=yes for the new example.

~~Exercise 6. Using Weka (to be done at your own time, not in class)~~
~~Load iris data (iris.arff). Choose 10-fold cross validation. Run the Naïve Bayes and Multi-layer percepton (trained with the backpropagation algorithm) classifiers and compare their performance. Which classifier produced the most accurate classification? Which one learns faster?~~

Exercise 7. k-Nearest neighbours
Given the training data in Exercise 4 (*Buy Computer* data), predict the class of the following new example using k-Nearest Neighbour for k=5: age<=30, income=medium, student=yes, credit-rating=fair. For similarity measure use a simple match of attribute values: Similarity(A,B)=

$\sum_{i=1}^{4} w_i * \partial(a_i, b_i) \Big/ 4$ where $\partial(a_i, b_i)$ is 1 if a_i equals b_i and 0 otherwise. a_i and b_i are either *age, income, student* or *credit_rating*. Weights are all 1 except for income it is 2.

Solution:

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

RID	Class	Distance to New
1	No	(1+0+0+1)/4=0.5
2	No	(1+0+0+0)/4=0.25
3	Yes	(0+0+0+1)/4=0.25
4	**Yes**	**(0+2+0+1)/4=0.75**
5	Yes	(0+0+1+1)/4=0.5
6	No	(0+0+1+0)/4=0.25
7	Yes	(0+0+1+0)/4=0.25
8	**No**	**(1+2+0+1)/4=1**
9	**Yes**	**(1+0+1+1)/4=0.75**
10	**Yes**	**(0+2+1+1)/4=1**
11	**Yes**	**(1+2+1+0)/4=1**
12	Yes	(0+2+0+0)/4=0.5
13	Yes	(0+0+1+1)/4=0.5
14	No	(0+2+0+0)/4=0.5

Among the five nearest neighbours four are from class *Yes* and one from class *No*. Hence, the k-NN classifier predicts buys_computer=yes for the new example.

Exercise 8. Decision trees

Given the training data in Exercise 4 (*Buy Computer* data), build a decision tree and predict the class of the following new example: age<=30, income=medium, student=yes, credit-rating=fair.

Solution:

First check which attribute provides the highest Information Gain in order to split the training set based on that attribute. We need to calculate the expected information to classify the set and the entropy of each attribute. The information gain is this mutual information minus the entropy:

The mutual information of the two classes $I(S_{Yes}, S_{No}) = I(9,5) = -9/14 \log_2(9/14) – 5/14 \log_2(5/14) = 0.94$

- For Age we have three values $age_{<=30}$ (2 yes and 3 no), $age_{31..40}$ (4 yes and 0 no) and $age_{>40}$ (3 yes 2 no)

Entropy(age) = 5/14 (-2/5 log(2/5)-3/5log(3/5)) + 4/14 (0) + 5/14 (-3/5log(3/5)-2/5log(2/5))
= 5/14(0.9709) + 0 + 5/14(0.9709)
= 0.6935
Gain(age) = 0.94 – 0.6935 = 0.2465

- For Income we have three values $income_{high}$ (2 yes and 2 no), $income_{medium}$ (4 yes and 2 no) and $income_{low}$ (3 yes 1 no)

Entropy(income) = 4/14(-2/4log(2/4)-2/4log(2/4)) + 6/14 (-4/6log(4/6)-2/6log(2/6))
+ 4/14 (-3/4log(3/4)-1/4log(1/4))
= 4/14 (1) + 6/14 (0.918) + 4/14 (0.811)
= 0.285714 + 0.393428 + 0.231714 = 0.9108

Gain(income) = 0.94 – 0.9108 = 0.0292

- For Student we have two values $student_{yes}$ (6 yes and 1 no) and $student_{no}$ (3 yes 4 no)

Entropy(student) = 7/14(-6/7log(6/7)) + 7/14(-3/7log(3/7)-4/7log(4/7)
= 7/14(0.5916) + 7/14(0.9852)
= 0.2958 + 0.4926 = 0.7884

Gain (student) = 0.94 – 0.7884 = 0.1516

- For Credit_Rating we have two values $credit\_rating_{fair}$ (6 yes and 2 no) and $credit\_rating_{excellent}$ (3 yes 3 no)

Entropy(credit_rating) = 8/14(-6/8log(6/8)-2/8log(2/8)) + 6/14(-3/6log(3/6)-3/6log(3/6))
= 8/14(0.8112) + 6/14(1)
= 0.4635 + 0.4285 = 0.8920

Gain(credit_rating) = 0.94 – 0.8920 = 0.479

Since Age has the highest Information Gain we start splitting the dataset using the age attribute

age

<=30

Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

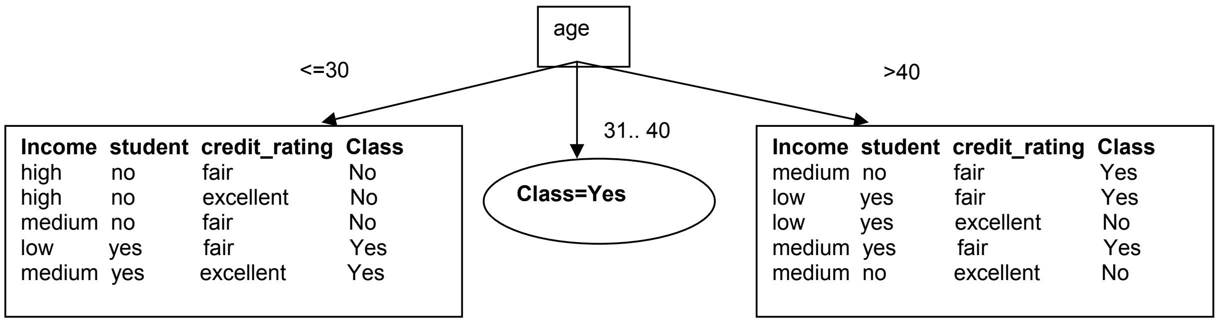
31.. 40

Income	student	credit_rating	Class
high	no	fair	Yes
low	yes	excellent	Yes
medium	no	excellent	Yes
high	yes	fair	Yes

>40

Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

Since all records under the branch $age_{31..40}$ are all of class Yes, we can replace the leaf with Class=Yes



The same process of splitting has to happen for the two remaining branches.
For branch $age_{<=30}$ we still have attributes income, student and credit_rating. Which one should be use to split the partition?

The mutual information is $I(S_{Yes}, S_{No}) = I(2,3) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5) = 0.97$

- For Income we have three values $income_{high}$ (0 yes and 2 no), $income_{medium}$ (1 yes and 1 no) and $income_{low}$ (1 yes and 0 no)

$$\text{Entropy(income)} = 2/5(0) + 2/5\,(-1/2\log(1/2)-1/2\log(1/2)) + 1/5\,(0)$$
$$= 2/5\,(1) = 0.4$$

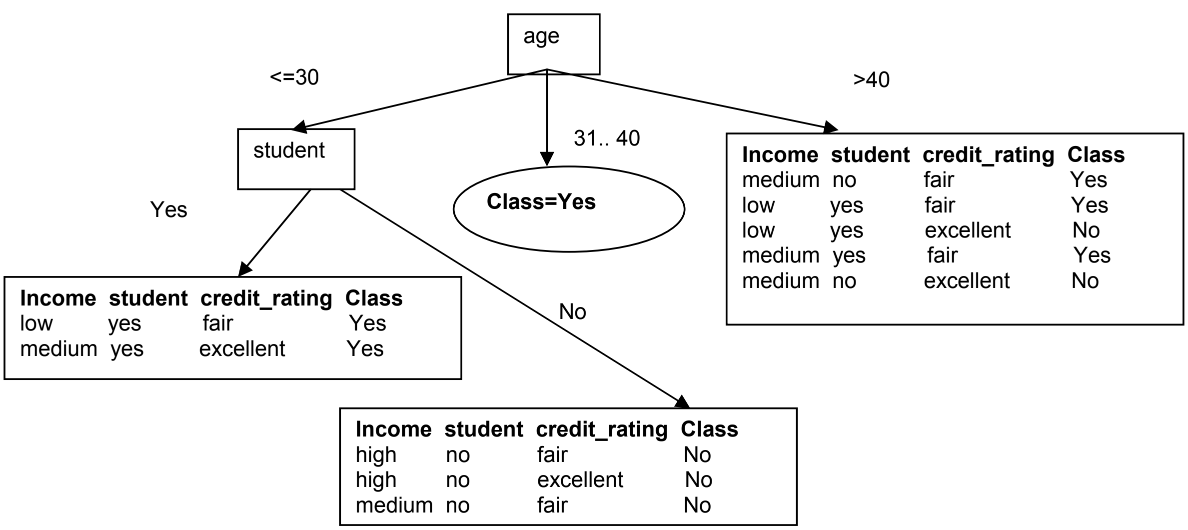
Gain(income) = 0.97 – 0.4 = 0.57

- For Student we have two values $student_{yes}$ (2 yes and 0 no) and $student_{no}$ (0 yes 3 no)

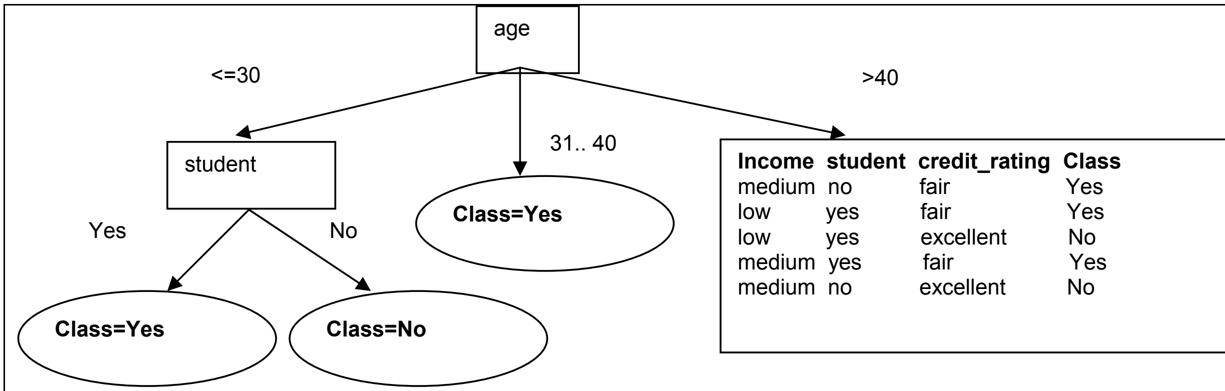
Entropy(student) = 2/5(0) + 3/5(0) = 0

Gain (student) = 0.97 – 0 = 0.97

We can then safely split on attribute student without checking the other attributes since the information gain is maximized.



Since these two new branches are from distinct classes, we make them into leaf nodes with their respective class as label:



Again the same process is needed for the other branch of age.

The mutual information is $I(S_{Yes}, S_{No}) = I(3,2) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0.97$

- For Income we have two values $income_{medium}$ (2 yes and 1 no) and $income_{low}$ (1 yes and 1 no)

$$Entropy(income) = 3/5(-2/3\log(2/3)-1/3\log(1/3)) + 2/5(-1/2\log(1/2)-1/2\log(1/2))$$
$$= 3/5(0.9182)+2/5(1) = 0.55+0.4 = 0.95$$

Gain(income) = 0.97 – 0.95 = 0.02

- For Student we have two values $student_{yes}$ (2 yes and 1 no) and $student_{no}$ (1 yes and 1 no)

$$Entropy(student) = 3/5(-2/3\log(2/3)-1/3\log(1/3)) + 2/5(-1/2\log(1/2)-1/2\log(1/2)) = 0.95$$

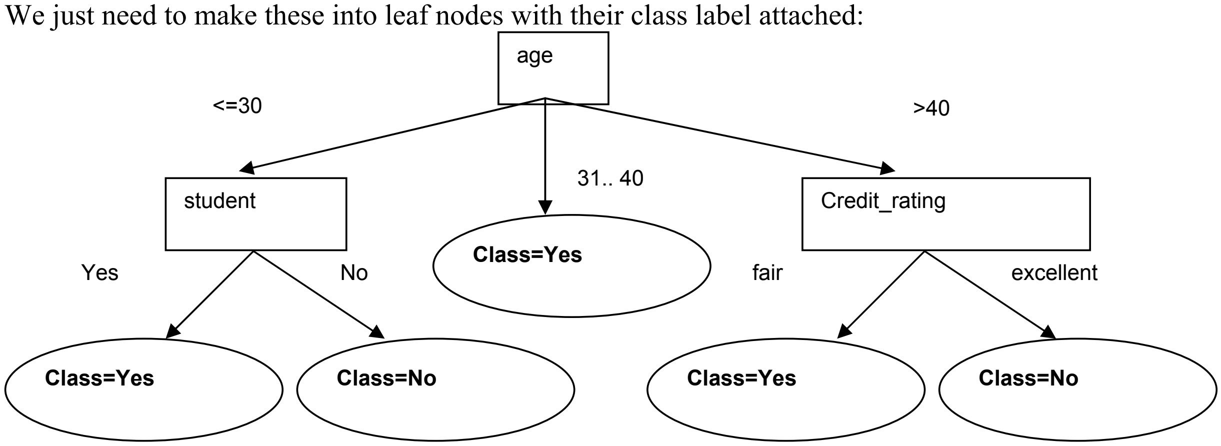
Gain (student) = 0.97 – 0.95 = 0.02

- For Credit_Rating we have two values $credit\_rating_{fair}$ (3 yes and 0 no) and $credit\_rating_{excellent}$ (0 yes and 2 no)

Entropy(credit_rating) = 0

Gain(credit_rating) = 0.97 – 0 = 0.97

We then split based on credit_rating. These splits give partitions each with records from the same class. We just need to make these into leaf nodes with their class label attached:



New example: age<=30, income=medium, student=yes, credit-rating=fair
Follow branch(age<=30) then student=yes we predict Class=yes ➔ Buys_computer = yes