LECTURE 03

08-10-2024

APPLICATION OF INFORMATION THEORY TO CLASSIFIERS

VECTOR FEATURES CLASS

CLASSIFIER

TRAINING SET

ALPHABETS

CLASSIFIER: GOAL 1 AND GOAL 2

TREE CLASSIFIERS: FEATURE TESTS AND LEAFS

INFORMATION GAIN RATIO

ID3 - ITERATIVE DICHOTOMISER

CATEGORICAL VS. NUMERICAL

STOPPING CRITERIA

EXAMPLE

SHAPE OF DECISION REGIONS

TREE ENSEMBLES AND RANDOM FORESTS

ASSIGNMENT 1

VECTOR, FEATURES, CLASS

$$\underline{v} = (x_1, \ldots, x_i, \ldots, x_m)$$

$\underline{v}$ → VECTOR (INSTANCE)

$(x_1, \ldots, x_i, \ldots, x_m)$ → FEATURES (VARIABLES)

$$F(\underline{v}) = C$$

C → CLASS (CATEGORY)

CLASSIFIER

IT IS AN ALGORITHM THAT TRIES TO ASSOCIATE THE CORRECT CLASS TO EACH VECTOR

IT'S BUILT STARTING FROM A TRAINING SET

≡ SET OF VECTORS FOR WHICH THE EXACT CLASS IS KNOWN.

$\underline{v} = (x_1, x_2)$

EXAMPLE (CATEGORICAL)

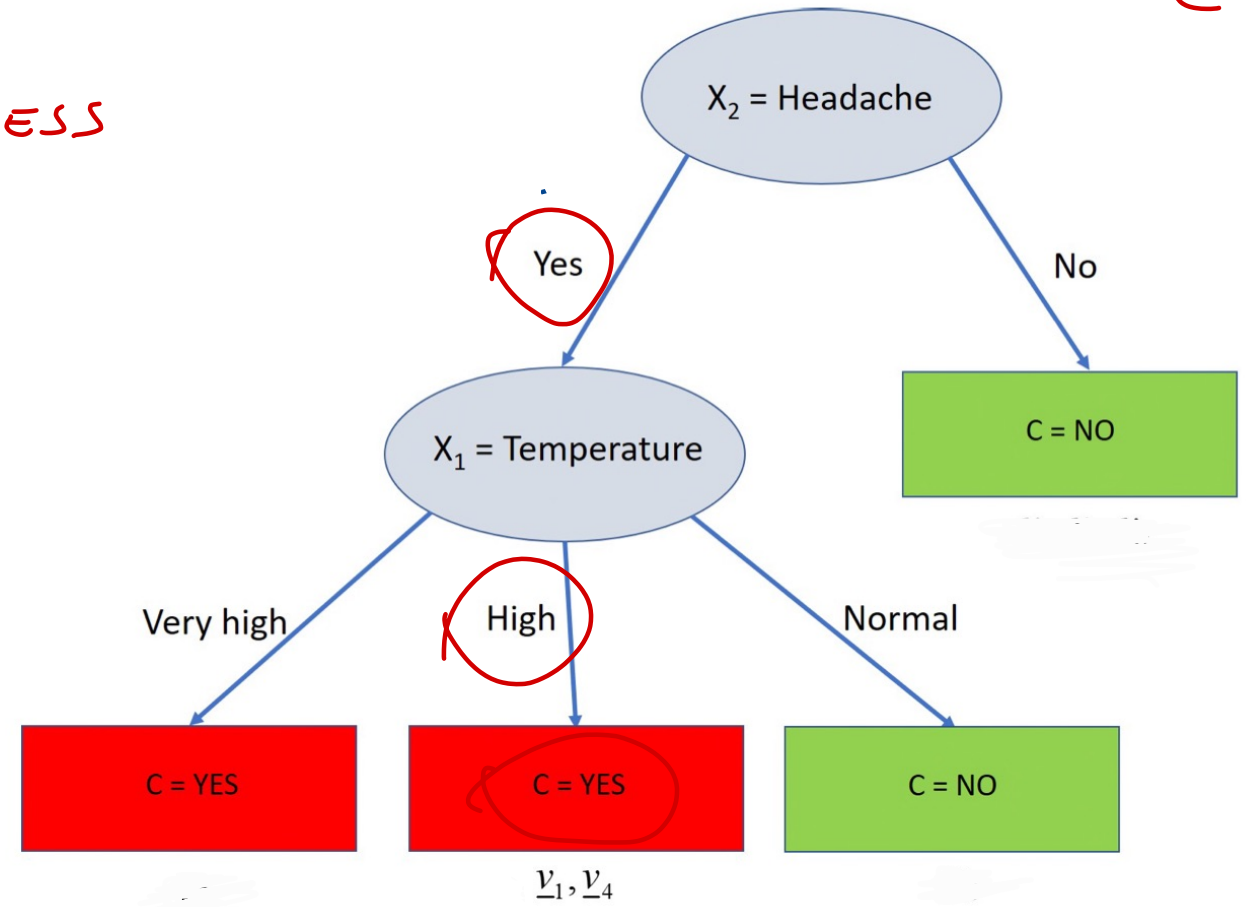
$\underline{v}$ = (HIGH, YES)

x_1 = TEMPERATURE

x_2 = HEADACHE

C = YES

CLASS C IS THE ILLNESS

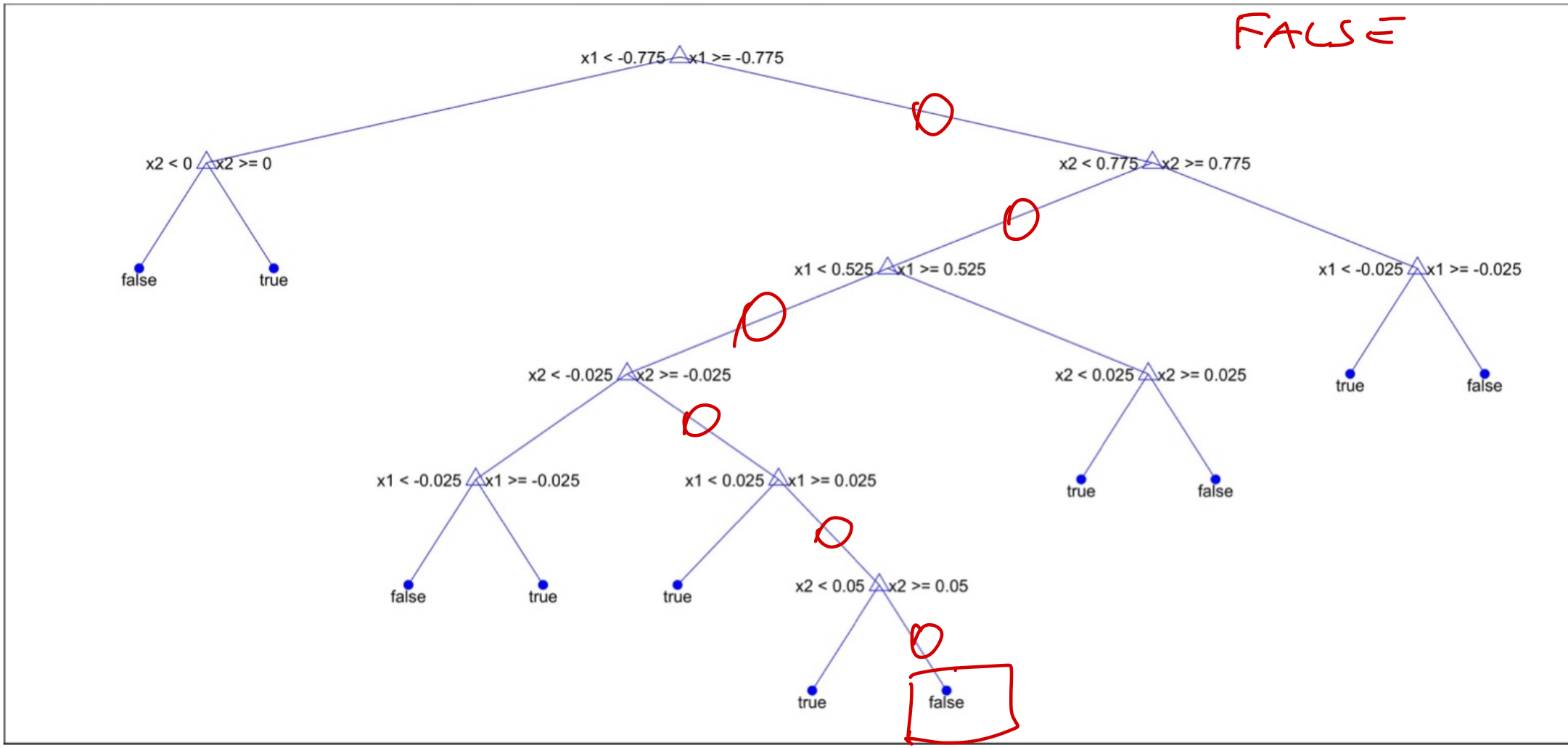


C = YES

$\underline{v} = (x_1 \quad x_2)$

EXAMPLE (NUMERICAL)

$\underline{v} = (0.2 \quad 0.5)$



FALSE

TRAINING SET

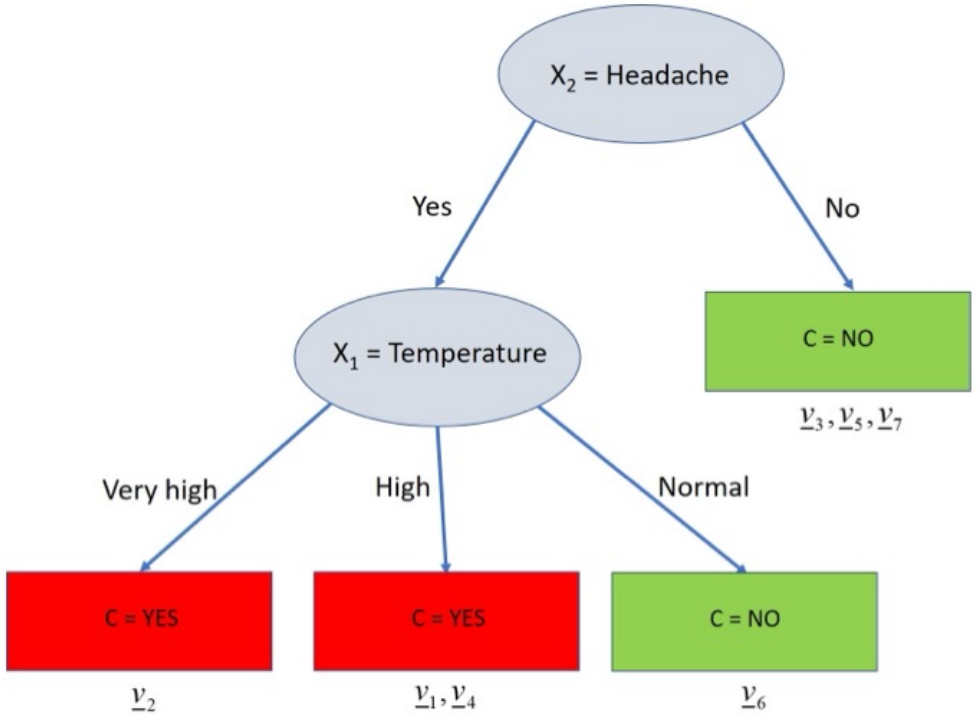
A SET OF VECTORS FOR WHICH THE CLASS IS KNOWN

$\underline{v} = (x_1 \quad x_2 \quad x_3) \quad c$

	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	normal	no	no	no
4	high	yes	yes	yes
5	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no

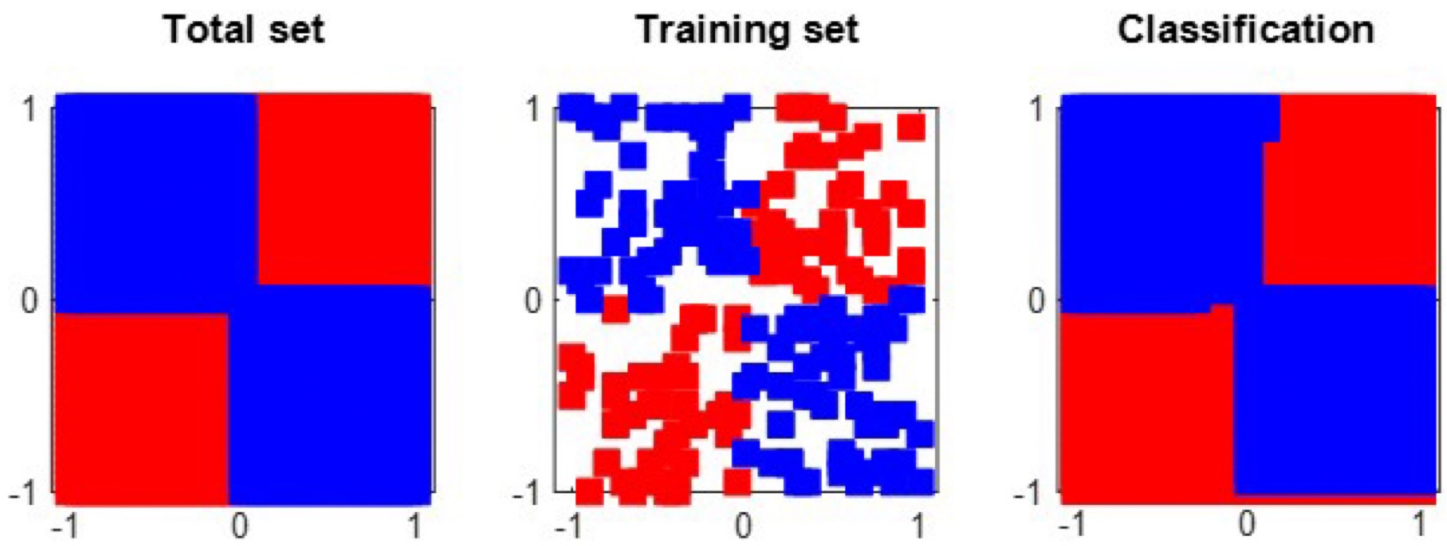
WE USE THE TRAINING SET TO BUILD THE CLASSIFIER

	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	normal	no	no	no
4	high	yes	yes	yes
5	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no



$\underline{x} = (x_1, x_2)$ $\quad -1 \le x_1 \le +1 \quad -1 \le x_2 \le +1$

$C = \{ \text{RED}, \text{BLUE} \}$



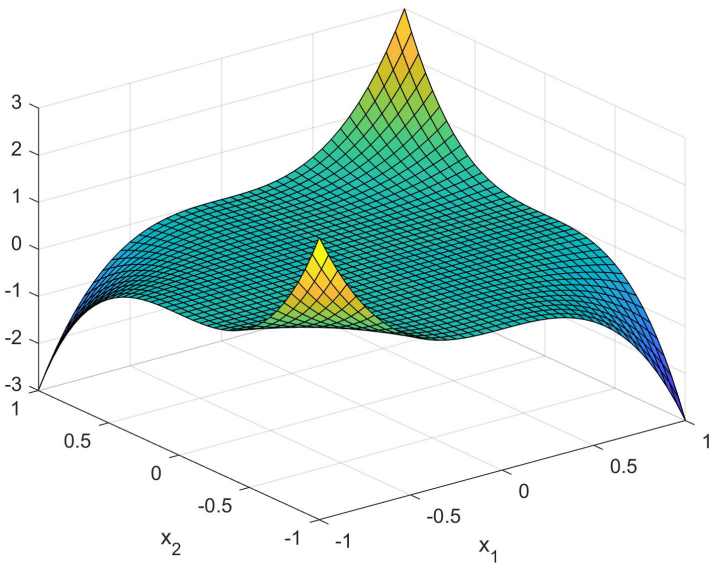
↑

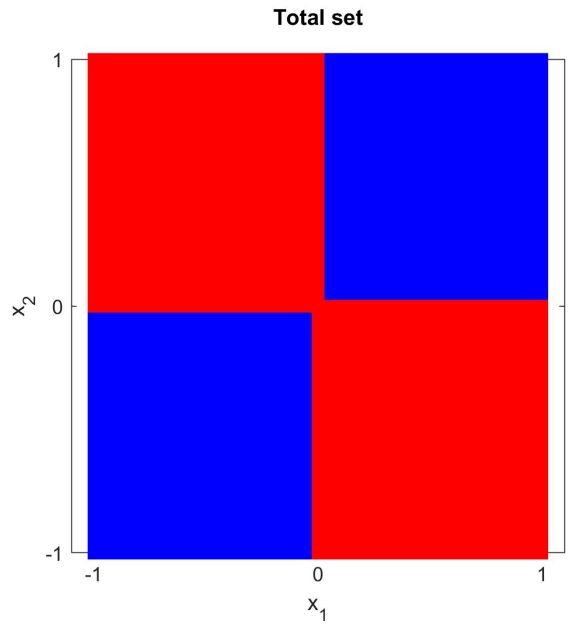
SOME CLASSIFICATIONS MAY BE WRONG

Example

$$\underline{v} = (x_1, x_2)$$

$$c = f(\underline{v}) = \text{sign}\left(3x_1^3 x_2^3\right)$$



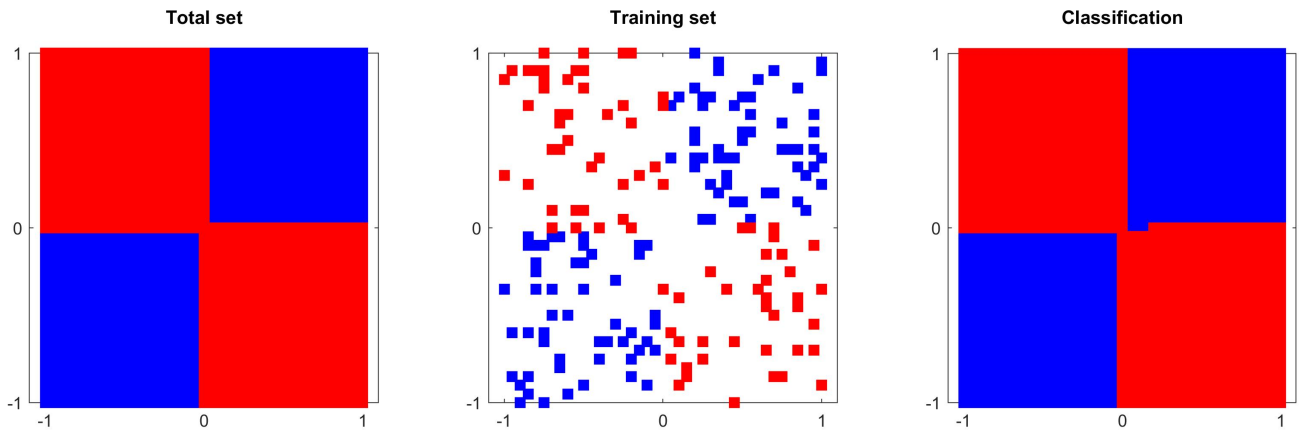


Example 2: training sets 1

$$x_1 = -1 : 0.05 : 1 \quad x_2 = -1 : 0.05 : 1;$$

$$N_v = 1681$$

$$N_{TS} = 300$$



THE QUALITY OF THE CLASSIFIER DEPENDS ON THE TRAINING SET

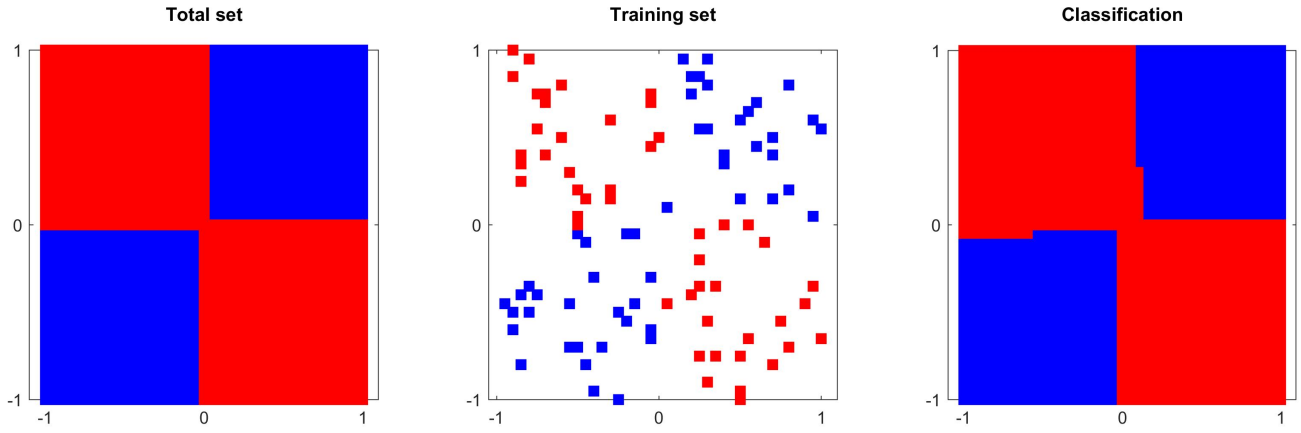
number of errors: 3/1681

Example 2: training sets 2

$$x_1 = -1 : 0.05 : 1 \quad x_2 = -1 : 0.05 : 1;$$

$$N_v = 1681$$

$$N_{TS} = 100$$



number of errors: 35/1681

Statistical classification has many different approaches:

- Linear Discriminant Analysis
- Naive Bayes Classifiers
- Nearest neighbor
- Support Vector Machine
- **Decision Trees**
- Neural Networks

VECTOR, FEATURES, CLASS, ALPHABETS

$$\underline{V} = (x_1, \ldots, x_i, \ldots, x_M)$$

A VECTOR $\underline{V}$ IS A SET OF M RANDOM VARIABLES CALLED FEATURES

$x_i \in \Omega_i$ ALPHABET FOR x_i

ALPHABET FOR $\underline{V}$

$$\Omega_V = \Omega_1 \times \cdots \times \Omega_i \times \cdots \times \Omega_M$$

TO EACH VECTOR $\underline{v}$ IS ASSOCIATED

A CLASS

$$C \in \Omega_C$$

$$F : \Omega_V \to \Omega_C$$

$$\underline{v} \to C$$

TRAINING SET

A TRAINING SET IS A SUBSET OF TOTAL ALPHABET Σ_V

$$\Sigma_T \subseteq \Sigma_V$$

FOR EACH VECTOR $\underline{w} \in \Sigma_T$ THE CLASS IS KNOWN

$$F(\underline{w}) = c$$

	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	normal	no	no	no
4	high	yes	yes	yes
5	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no

$\Omega_C = \{ YES, NO \}$

$\Omega_1 = \{ HIGH, VERY\_HIGH, NORMAL \}$

$\Omega_2 = \{ YES, NO \}$

$\Omega_3 = \{ YES, NO \}$

CLASSIFIER g

STARTING FROM THE TRAINING SET

WE BUILD A CLASSIFIER g

$$g : \Omega_V \to \Omega_C$$

$$\underline{v} \to c'$$

$$c' = g(\underline{v})$$

GOAL 1

$g = f$ FOR EACH VECTOR OF THE TS

$\forall \underline{w} \in \Omega_T \qquad f(\underline{w}) = g(\underline{w})$

NOT ALWAYS POSSIBLE

(CONFUSION MATRIX → ASSIGNMENT 1)

GOAL 2

$g = f$ FOR EACH VECTOR $\underline{v} \in \mathbb{R}_V$

$\forall \underline{v} \in \mathbb{R}_V : \quad f(\underline{v}) = g(\underline{v})$

NOT ALWAYS VERIFIED

TREE CLASSIFIER: FEATURE TEST, LEAF

THE FUNCTION g IS IMPLEMENTED BY A FLOW CHART WITH 2 BLOCKS

FEATURE TEST

WE CHECK A FEATURE AND WE CHOOSE A DIRECTION

LEAF

WE ASSOCIATE THE CLASS TO THE VECTOR

$g(\underline{v}) = C$

IMPORTANT: INFORMATION GAIN
= REDUCTION OF UNCERTAINTY

HOW DO WE SELECT THE FEATURE TO TEST

$$\underline{V} = (x_1 \quad x_i \quad x_n)$$

IDEA: SELECT THE FEATURE THAT MAXIMIZES THE INFORMATION GAIN RATIO

$$\overline{I}(C; x_i) = \frac{I(C; x_i)}{H(x_i)}$$

$$I(C; x_i) = H(C) - H(C|x_i)$$

INFORMATION GAIN RATIO

WHY ? BECAUSE OBSERVING X_i

MAXIMIZES THE

REDUCTION OF UNCERTAINTY

ABOUT CLASS C

$\equiv$

"IT CARRIES MORE INFO.

ABOUT C"

ID3 · ITERATIVE DICHOTOMISER (Ross Quinlan)

- START FROM THE TRAINING SET
- COMPUTE $\overline{I}(C, x_i)$ FOR EACH FEATURE
- SELECT THE FEATURE WITH LARGEST $\overline{I}(C; x_i)$
- IMPLEMENT THIS FEATURE NODE
- THIS DIVIDES YOUR TRAINING SET INTO SUBSETS
- REPEAT ITERATIVELY FOR EACH SUBSET

IF THE FEATURES ARE CATEGORICAL THEY CAN BE TESTED ONLY ONCE

WHAT ABOUT NUMERICAL FEATURES

FOR EACH NUMERICAL FEATURE WE CONSIDER ALL POSSIBLE THRESHOLDS THAT DIVIDE THE TRAINING SET VALUES IN TWO.

FOR EACH THRESHOLD WE COMPUTE IGR AND WE KEEP THE THRESHOLD WITH MAXIMUN IGR

Ex

$V = (\ldots, 20)$

$V = (\ldots, 40)$

$V = (\ldots, 60)$

$t_1 = 30$ $(0-30/31-100)$

$t_2 = 50$ $(0-50/51-100)$

WE REPEAT FOR EACH NUMERICAL FEATURE AND WE SELECT THE ONE WITH MAX IGR

DIFFERENCES FROM CATEGORICAL FEATURES

NUMERICAL FEATURES CAN BE TESTED MULTIPLE TIMES

STOPPING CRITERIA

BEST SCENARIO:

WE OBTAIN A SUBSET MADE BY VECTORS WITH SAME CLASS

→ NO DOUBTS → WE ASSIGN THIS CLASS

WE OBTAIN A SUBSET

MADE BY VECTORS WITH

DIFFERENT CLASSES

BUT (CATEGORICAL FEATURES)

ALL FEATURES HAVE

ALREADY BEEN ANALYTED

→ ASSIGN THE CLASS

WITH MORE VECTORS

→ WE MAKE A MISTAKE FOR

SOME $\underline{w} \in \Omega_T$ $f(\underline{w}) \neq g(\underline{w})$

Example

$\underline{v} = (x_1 \quad x_2 \quad x_3) \quad C$

	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	normal	no	no	no
4	high	yes	yes	yes
5	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no

$-\sum p_i \log_2 p_i$

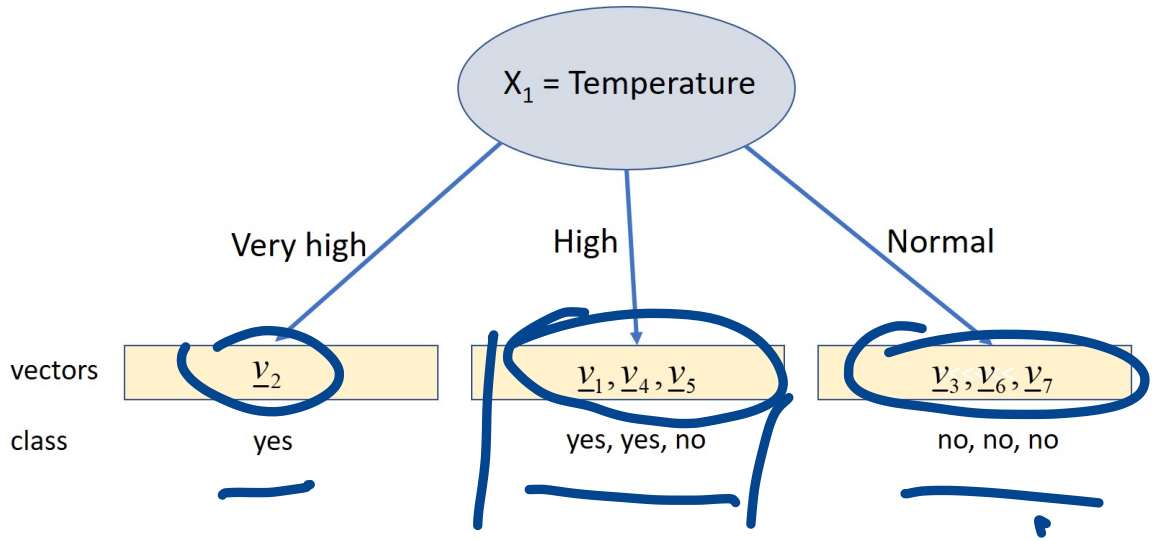
Let us first compute the entropy of the class:

$$P(c = \text{yes}) = \frac{3}{7} \quad P(c = \text{no}) = \frac{4}{7}$$

$$H(C) = 0.9852$$

Example

Let us consider the first feature:



$$H(C) = 0.9852$$

$$H(C|X_1 = \text{Very High}) = 0 \quad 1/2$$

$$H(C|X_1 = \text{High}) = 0.9183 \quad 3/7$$

$$H(C|X_1 = \text{Normal}) = 0 \quad 3/7$$

$$H(C|X_1) = 0.3936$$

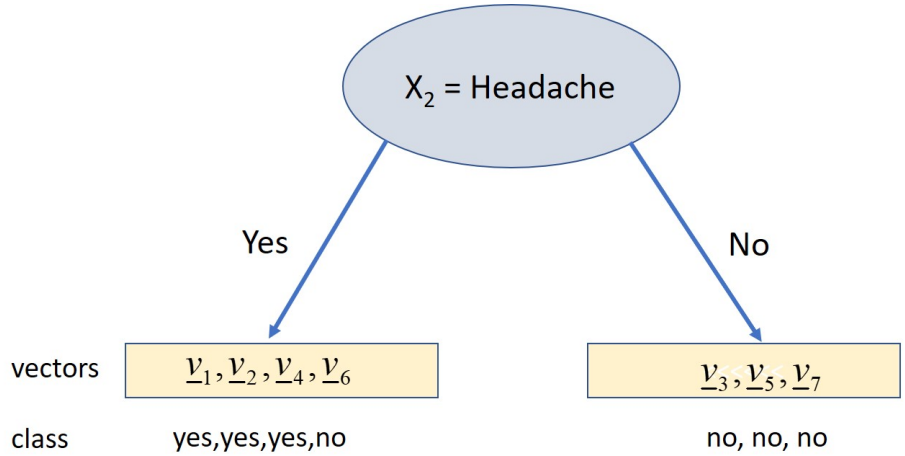
$$I(C; X_1) = 0.5916 = H(C) - H(C|X_1)$$

$$H(X_1) = 1.4488$$

$$IGR(C, X_1) = 0.4083 = I(C; X_1) / H(X_1)$$

Example

Let us consider the second feature:



$$H(C) = 0.9852$$

$$H(C|X_2 = \text{Yes}) = 0.8113 \qquad 4/7$$

$$H(C|X_2 = \text{No}) = 0 \qquad 3/7$$

$$\rightarrow \quad H(C|X_2) = 0.4636$$

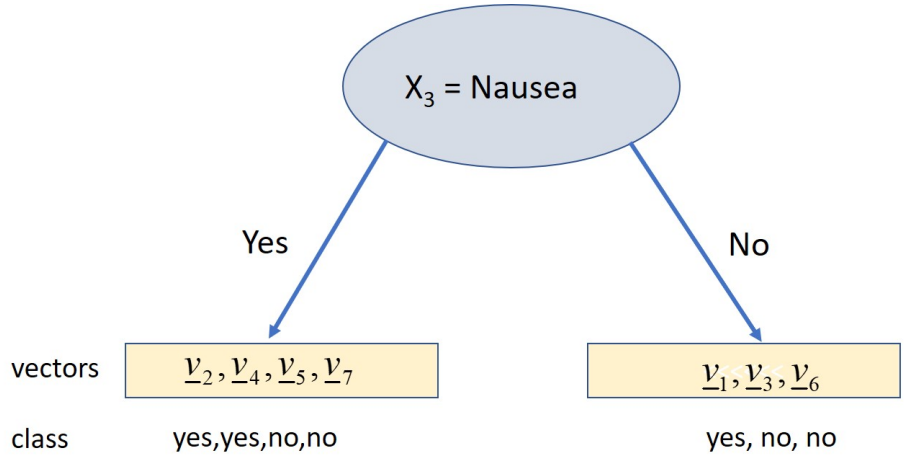
$$\rightarrow \quad I(C; X_2) = 0.5216 \qquad = H(C) - H(C|X_2)$$

$$\rightarrow \quad H(X_2) = 0.9852$$

$$\rightarrow \quad IGR(C, X_2) = 0.5294$$

Example

Let us consider the third feature:



$$H(C) = 0.9852$$

$$H(C|X_3 = \text{Yes}) = 1 \qquad 4/7$$

$$H(C|X_3 = \text{No}) = 0.9183 \qquad 3/7$$

$$H(C|X_3) = 0.9650$$

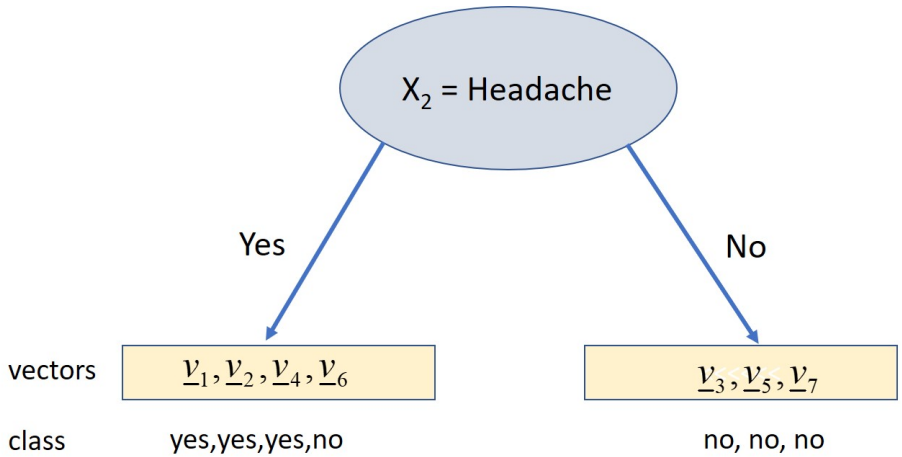
$$I(C; X_3) = 0.0202 \qquad = H(C) - H(C|X_3)$$

$$H(X_3) = 0.9852$$

$$IGR(C, X_3) = 0.0205 \qquad = I(C; X_3) / H(X_3)$$

Example

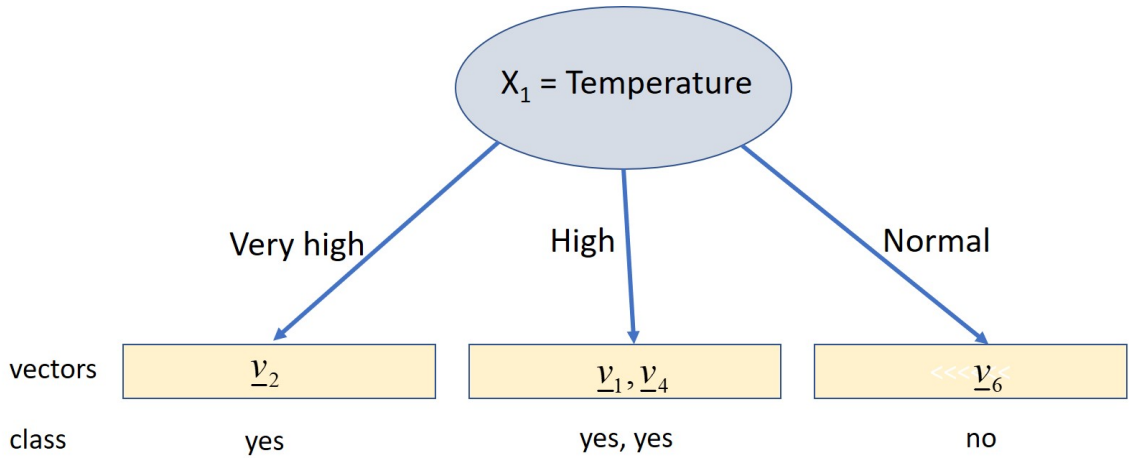
Based on the best Information Gain Ratio, we choose the second feature:



Now we apply the same procedure to the subset $\underline{v_1}, \underline{v_2}, \underline{v_4}, \underline{v_6}$. We consider the remaining features: x_1 =Temperature and x_3 = Nausea.

Example

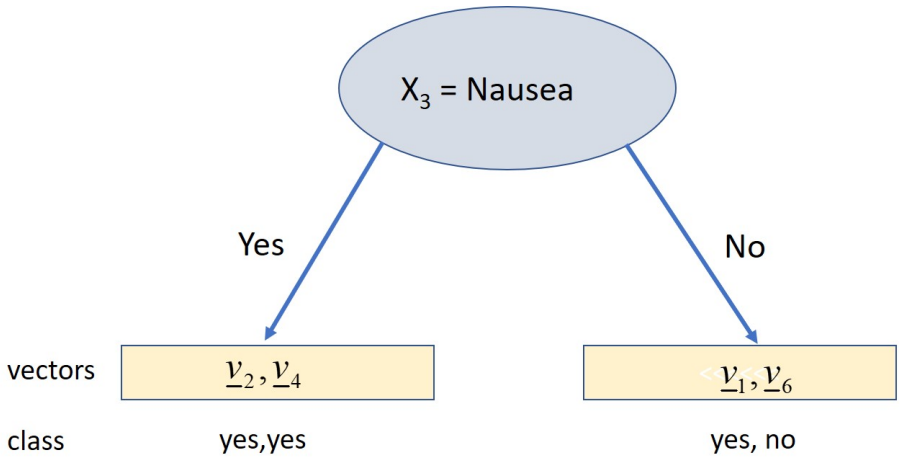
Let us consider the first feature x_1 =Temperature:



$$H(C) = 0.8113$$
$$H(C|X_1 = \text{Very High}) = 0$$
$$H(C|X_1 = \text{High}) = 0$$
$$H(C|X_1 = \text{Normal}) = 0$$
$$H(C|X_1) = 0$$
$$I(C; X_1) = 0.8113$$
$$H(X_1) = 1.5$$
$$IGR(C, X_1) = 0.5408$$

Example

Let us consider the third feature x_3 =Nausea:



$$H(C) = 0.8113$$

$$H(C|X_3 = \text{Yes}) = 0$$

$$H(C|X_3 = \text{No}) = 1$$

$$H(C|X_3) = 0.5$$

$$I(C; X_3) = 0.3113$$

$$H(X_3) = 0.9852$$

$$IGR(C, X_3) = 0.3160$$

Example

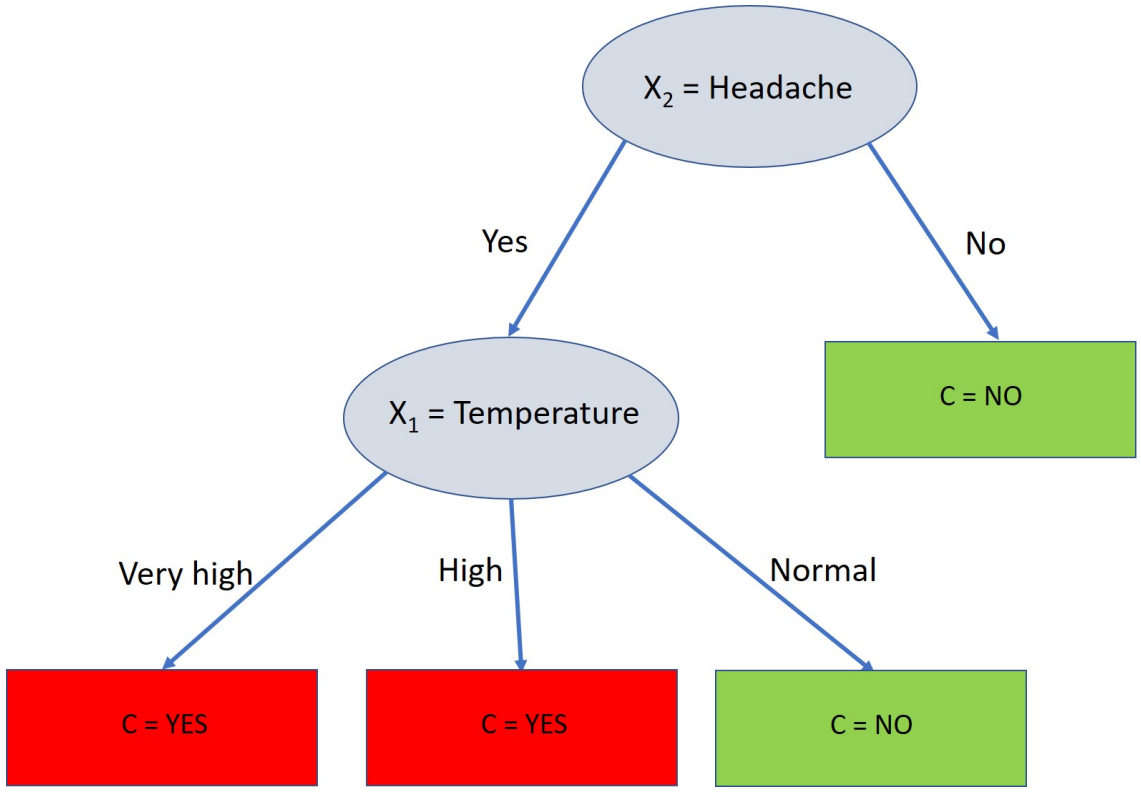
For the other subset it is useless to make a choice because all the vectors already belong to the same class:

$\underline{v}_3, \underline{v}_5, \underline{v}_7$

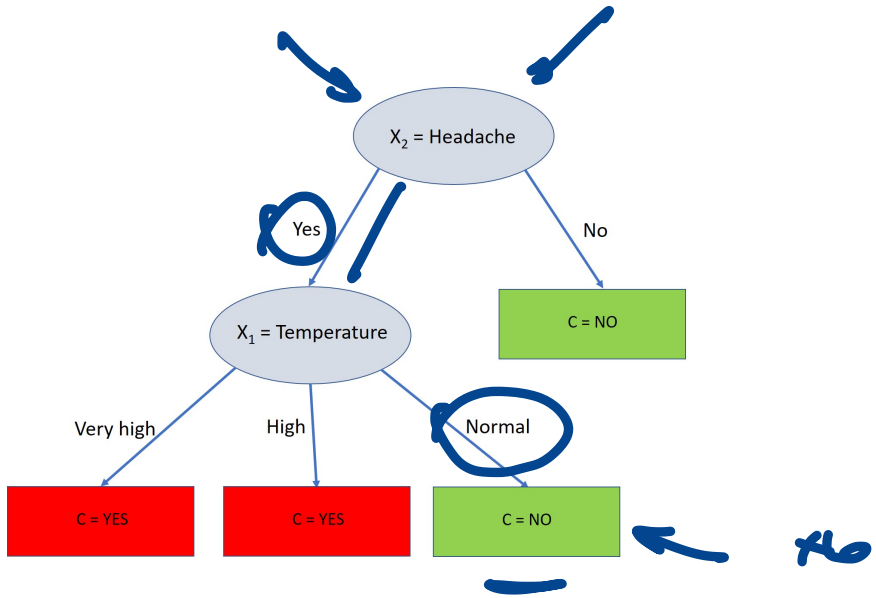
no, no, no

Example

Finally we have our decision tree:



Example



Now we can use the decision tree as a classifier. If we observe a new vector, we apply the rule to compute the class it belongs to. As an example

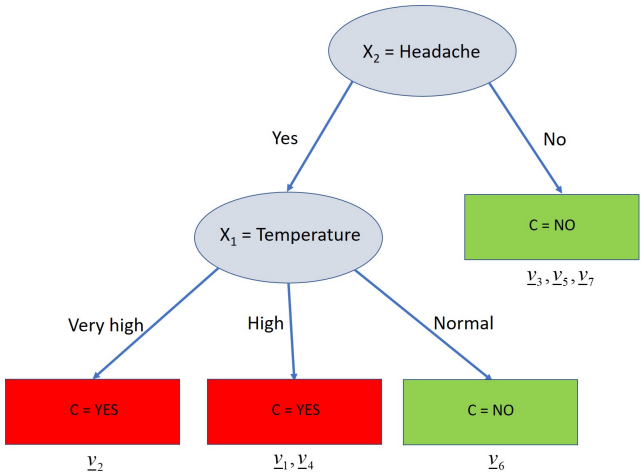
$$\underline{v} = (\text{normal}, \text{yes}, \text{yes})$$

is mapped into class NO

$$\underline{v} = (\text{veryhigh}, \text{yes}, \text{no})$$

is mapped into class YES

Example



Note that we can translate the tree classifier into these logical rules:

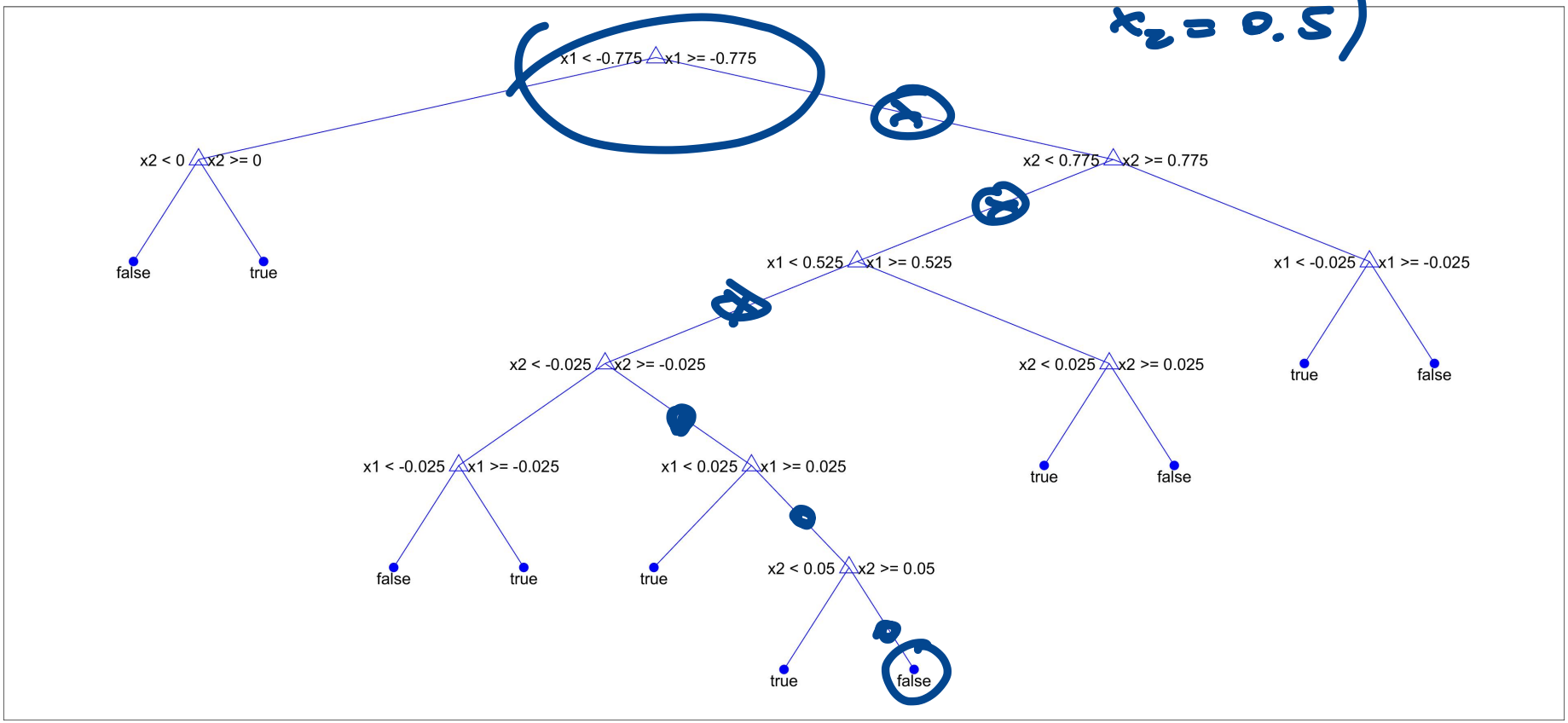
$$(\text{Headache} = \text{yes}) \quad \text{AND}\, (\text{Temperature} = \text{veryhigh}) \longrightarrow (\text{Class} = \text{YES})$$

$$(\text{Headache} = \text{yes}) \quad \text{AND}\, (\text{Temperature} = \text{high}) \longrightarrow (\text{Class} = \text{YES})$$

$$(\text{Headache} = \text{yes}) \quad \text{AND}\, (\text{Temperature} = \text{normal}) \longrightarrow (\text{Class} = \text{NO})$$

$$(\text{Headache} = \text{no}) \longrightarrow (\text{Class} = \text{NO})$$

Example 1


x1 < -0.775
x1 >= -0.775
x2 < 0
x2 >= 0
false
true
x2 < 0.775
x2 >= 0.775
x1 < 0.525
x1 >= 0.525
x1 < -0.025
x1 >= -0.025
true
false
x2 < -0.025
x2 >= -0.025
x2 < 0.025
x2 >= 0.025
true
false
x1 < -0.025
x1 >= -0.025
x1 < 0.025
x1 >= 0.025
true
false
false
true
true
x2 < 0.05
x2 >= 0.05
true
false

SHAPE OF DECISION REGIONS

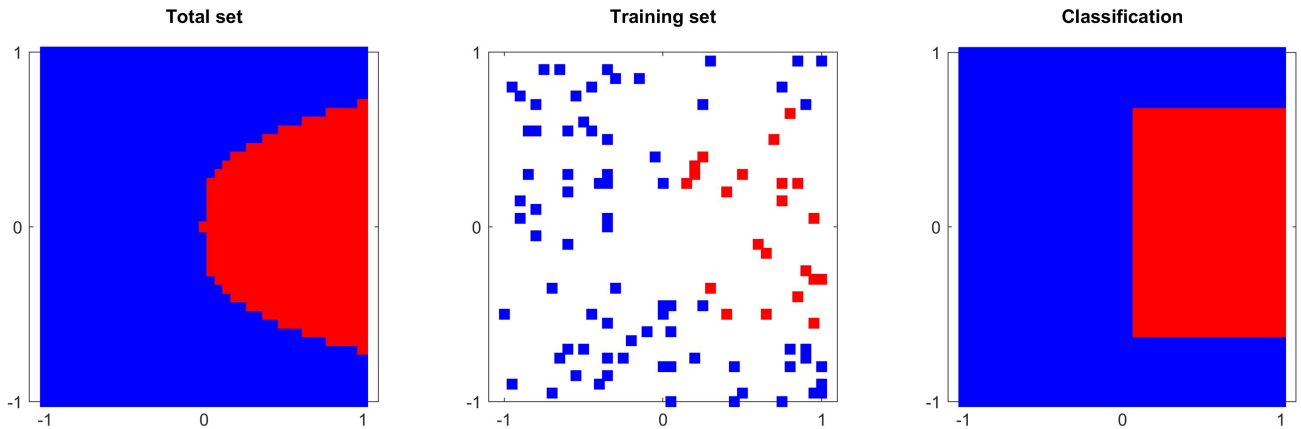
Example 3: training set 1

$$\underline{v} = (x_1, x_2)$$

$$c = f(\underline{v}) = \text{sign}\left(-2\sqrt[3]{x_1^2} + 4x_2^2\right)$$

$$x_1 = -1 : 0.05 : 1 \quad x_2 = -1 : 0.05 : 1;$$

$$N_v = 1681 \quad N_{TS} = 100$$



number of errors: 97/1681

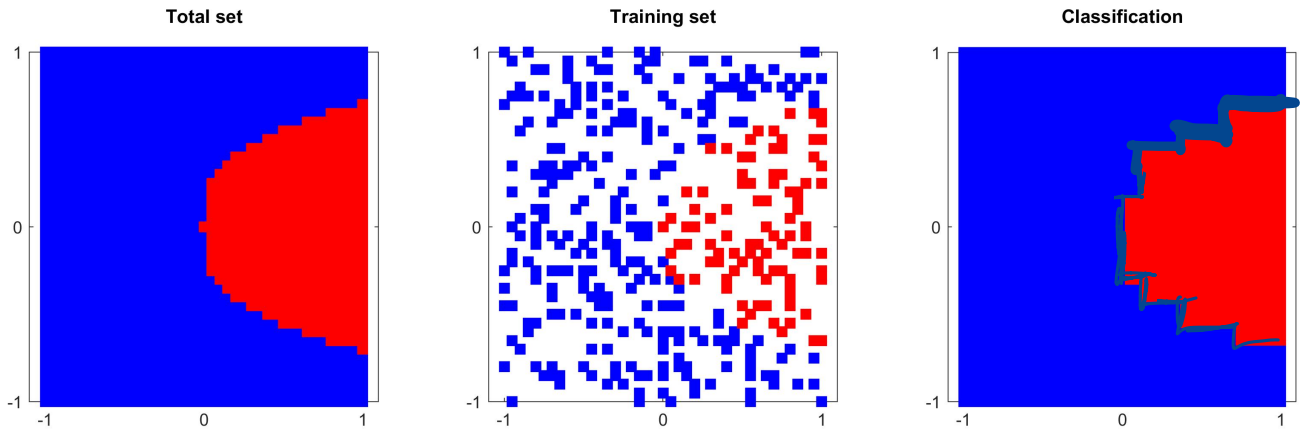
Example 3: training set 2

$$\underline{v} = (x_1, x_2)$$

$$c = f(\underline{v}) = \text{sign}\left(-2\sqrt[3]{x_1^2} + 4x_2^2\right)$$

$$x_1 = -1 : 0.05 : 1 \quad x_2 = -1 : 0.05 : 1;$$

$$N_v = 1681 \quad N_{TS} = 400$$



number of errors: 26/1681

Example 4: training set 1

$$\underline{v} = (x_1, x_2)$$

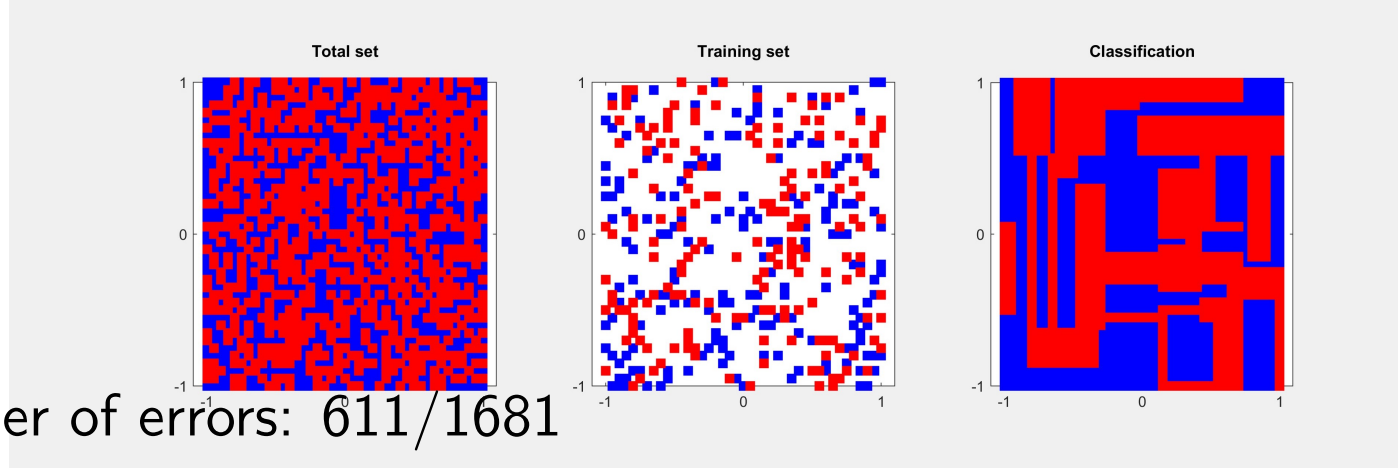
$$c = f\left(\underline{v}\right) = \operatorname{sign}\left(3x_1^3 x_2^3 h(x_1, x_2)\right)$$

where $h(x_1, x_2)$ is a 2-dimensional normal random variable with pdf

$$f_h(x_1, x_2) = \frac{1}{2\pi} \exp\left(\frac{-x^2 - y^2}{2}\right)$$

$$x_1 = -1 : 0.05 : 1 \quad x_2 = -1 : 0.05 : 1;$$

$$N_v = 1681 \quad N_{TS} = 400$$



number of errors: 611/1681

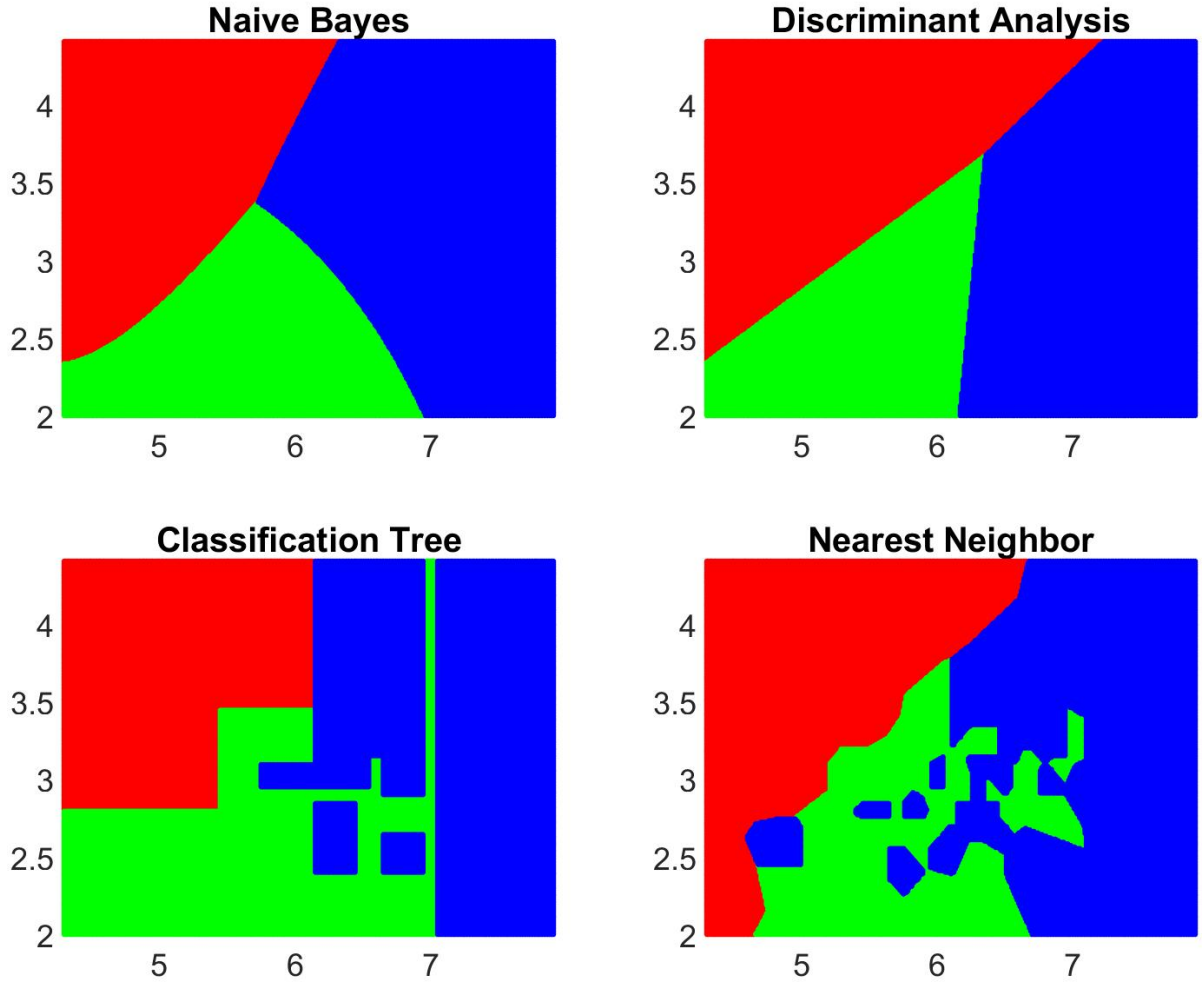
Problem

For numerical features, this Classifier divides the space $A_v = \mathbb{R}^M$ into **hyper-rectangles** of dimension M defined as

$x_{1,inf} < x_1 < x_{1,sup}, x_{2,inf} < x_2 < x_{2,sup} \ldots x_{M,inf} < x_M < x_{M,sup}$.

For each of these hyperrectangles, there is one and only one class. As a consequence, the set of vectors which is mapped by the classification function g into a given class is always the union of hyper-rectangles. If the true set mapped by the original function f into the same class has a different shape we can only approximate it.

Comparison


Naive Bayes
Discriminant Analysis
Classification Tree
Nearest Neighbor
4
3.5
3
2.5
2
5
6
7

Decision Tree Ensembles

To improve the performance of tree classifiers: Tree Ensembles

Main idea: given the training data-set, instead of building a single classification tree we build many and we combine their decisions.

Random forest approach

Bagging: starting from the training data-set T_S we build N bootstrapped data-sets made by randomly extracted vectors of T_S.

Random feature selection: For each bootstrapped data-set, we build a classification tree. When doing this, each time we process a subset to build a node, we only use $M' \leq M$ randomly extracted features.

Decision: given a new vector, we process it with all the N tree classifiers and we assign the most popular class among the N results.

(To decide the number M' of features, we usually start from a value $M' \simeq \sqrt{M}$. Then we test the accuracy of the forest by using the out-of-bag vectors (the vectors which are outside the different bootstrapped data-sets). We repeat the procedure by increasing and decreasing the values of M' looking for the value with best accuracy.)

Information Theory for Data Science

Assignment 1

Introduction to Information Theory and application to Classifiers

Final version 1.0

Exercises:

1. Entropy of a random variable with 3 outcomes (pt. 3)
2. Entropy of a random variable from a data series (pt. 4)
3. Application of the principle of maximum entropy (pt. 4)

4. Exercise on Information Gain Ratio (pt. 4)
5. Kullback Leibler divergence from empirical distribution (pt. 3)
6. Permutation entropy for anomaly detection (pt. 5)

7. Information Gain Ratio and Classifier (pt. 7)

Exercise 1 - Entropy of a random variable with 3 outcomes (pt. 3)

Exercise 1 - Entropy of a random variable with 3 outcomes

1. Given a random variable with 3 outcomes, write a program to plot the entropy as a function of all possible probability vectors
2. Start with a probability vector where one of the elements is significantly higher than the others. Apply an iterative averaging procedure (for example, replace each element with the average of itself and its neighbors, followed by normalization). For each updated vector, compute the entropy and plot its value on the figure generated in step 1. Show that as the probability distribution approaches the uniform distribution, the entropy approaches its maximum value. Finally, discuss the results.

Exercise 2 - Entropy of a random variable from a data series (pt. 4)

Exercise 2 - Entropy of a random variable from a data series

1. Identify a data series and estimate the probabilities of the outcomes based on their occurrences, updating the probabilities at each time step.
2. At each time step, compute the entropy, plot its behavior, and discuss the results

Note: In the presentation, include a link to the source of the data series

Exercise 3 - Application of the principle of maximum entropy (pt. 4)

Exercise 3.a

1. Invent an exercise where you have a random variable X with an alphabet Ω_X with 2 outcomes with integer values.
2. Show some examples of the the probability distribution P(X) for different values of the mean value.
3. Discuss the results

Exercise 3.b

1. Invent an exercise where you have a random variable X with alphabet Ω_X with at least 4 outcomes, where each outcome has an inteher value (“cost”).
2. Fix the mean value bigger than the arithmetic average of the costs, and apply the principle of maximum entropy to find the probability distribution P(X)
3. Plot P(X)
4. Repeat with a mean value equal to the arithmetic average and plot the result
5. Repeat with other values of the mean value and plot the results
6. Comment the results

You must numerically solve the equation generated by the Lagrange optimization.

As an example , for Matlab you can use

```
syms x
eqn = ( . . .  )*mu == ( . . . );
V = vpasolve(eqn,x,[0 10])
```

Exercise 4 – Exercise on Information Gain Ratio (pt. 4)

Exercise 4 – Exercise on Information Gain Ratio

1. Invent an exercise based on a data set like this one.

Athlete	Training Hours	Rest Hours	Gym Workouts	Performance
A	High	Low	Low	Lose
B	Medium	High	Medium	Win
C	Low	Medium	High	Lose
D	High	Medium	Medium	Win
E	Medium	High	Low	Win
F	Low	Low	High	Lose
G	High	Low	Medium	Win
H	Medium	Low	Low	Lose
I	Low	High	Medium	Win
J	High	High	High	Win

The last column represents your target variable X.

2. Compute the Information Gain Ratio with respect to all the other variables and select the one that provides more information about X.

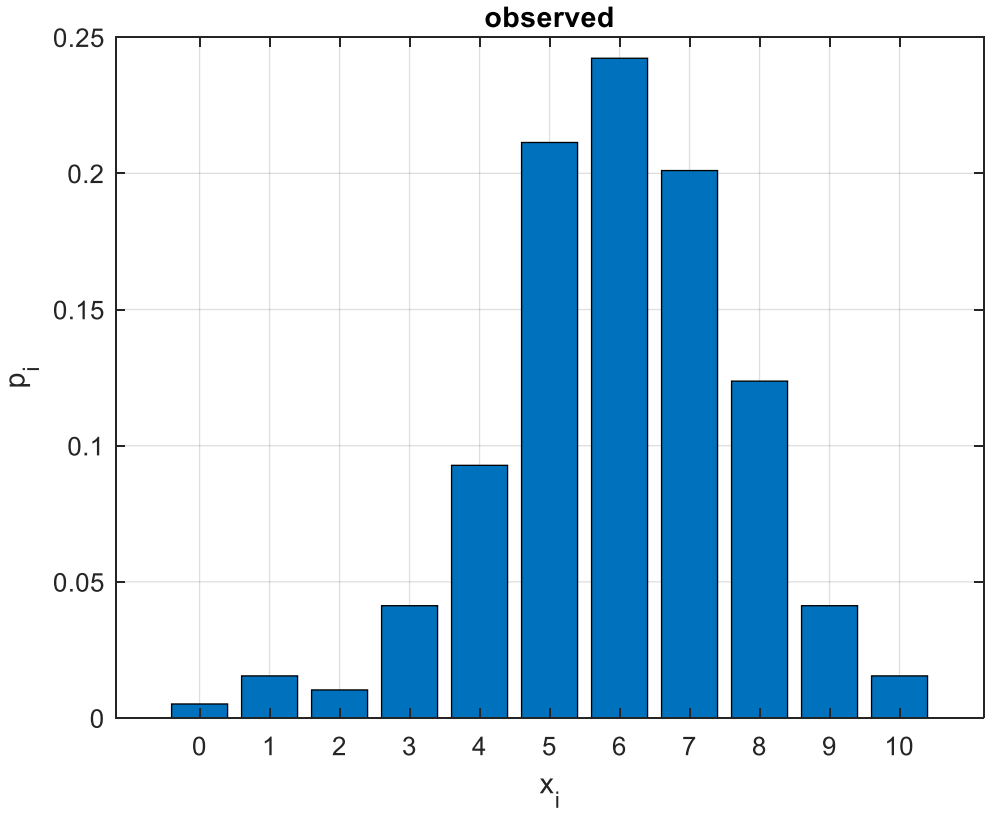
Exercise 5 - Kullback-Leibler distance from empirical distribution (pt. 3)

Exercise 5 - Kullback-Leibler distance from empirical distribution

Consider the observed data

$$x_i = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}$$

Number of observed x_i = [1 3 2 8 18 41 47 39 24 8 3]


observed
0.25
0.2
0.15
0.1
0.05
0
p_i
0
1
2
3
4
5
6
7
8
9
10
x_i

Compare with uniform pmf

1. Plot the two observed and the uniform pmfs
2. In the title write the KL divergence value

Compare with binomial pmf with $0 < p < 1$ (step=0.001)

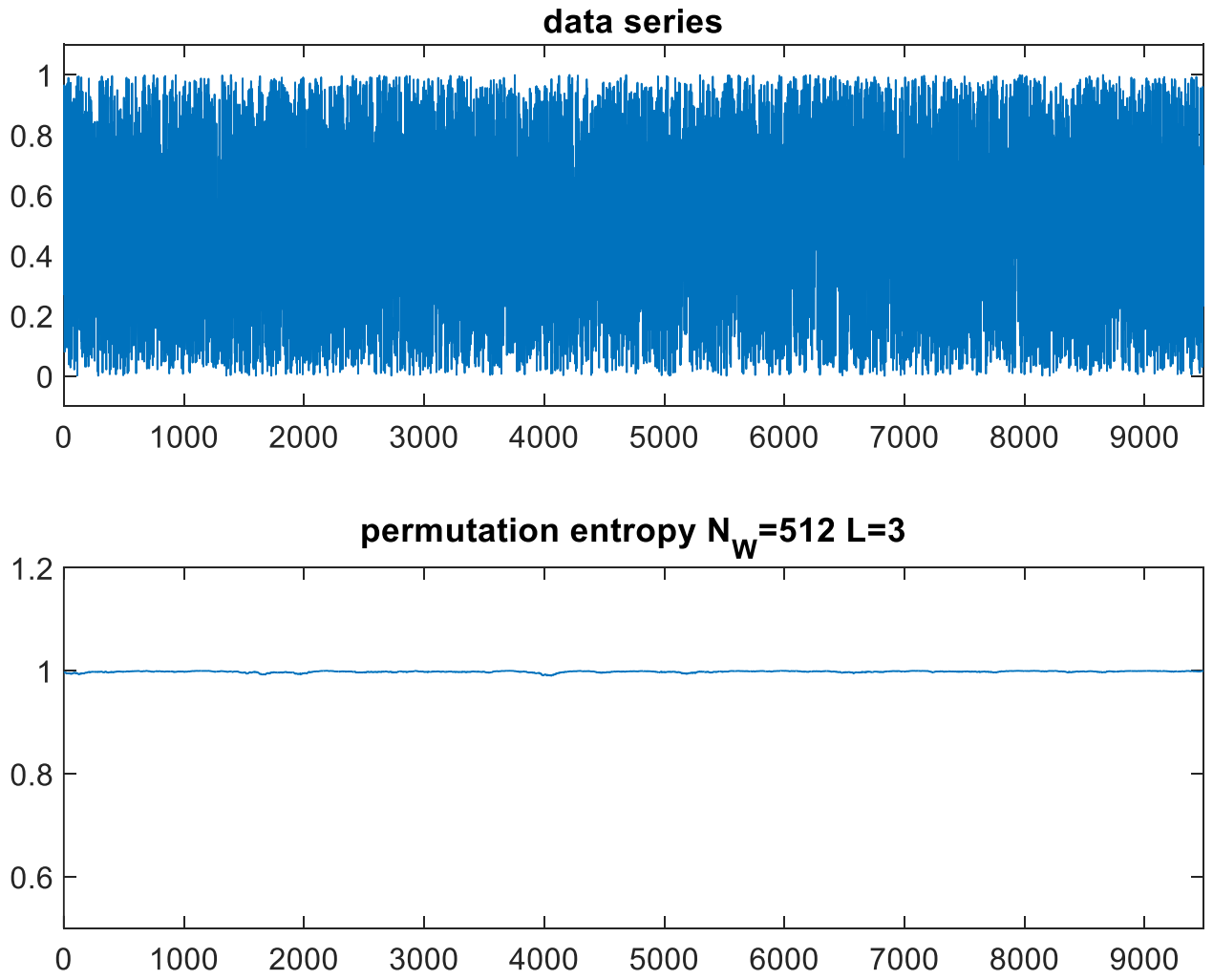
3. Identify the pmf at minimum KL divergence
4. Plot the observed and the best binomial pmfs
5. In the title write the value of p and KL divergence

Exercise 6 - Permutation Entropy for Time Series Anomaly Detection (pt. 5)

Exercise 6 - Permutation Entropy for Time Series Anomaly Detection

Write a program that:

- Generates a data series made by 10,000 random symbols.
- Computes the permutation entropy with a sliding window of $N_W = 512$ symbols. (For example, choose an order m = 3.)
- Plot the data and the entropy.


data series
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
permutation entropy N_W=512 L=3
1.2
1
0.8
0.6
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

- Insert a pattern between 3000 and 4000 made by correlated data (with about the same mean value and variance of the original random data).

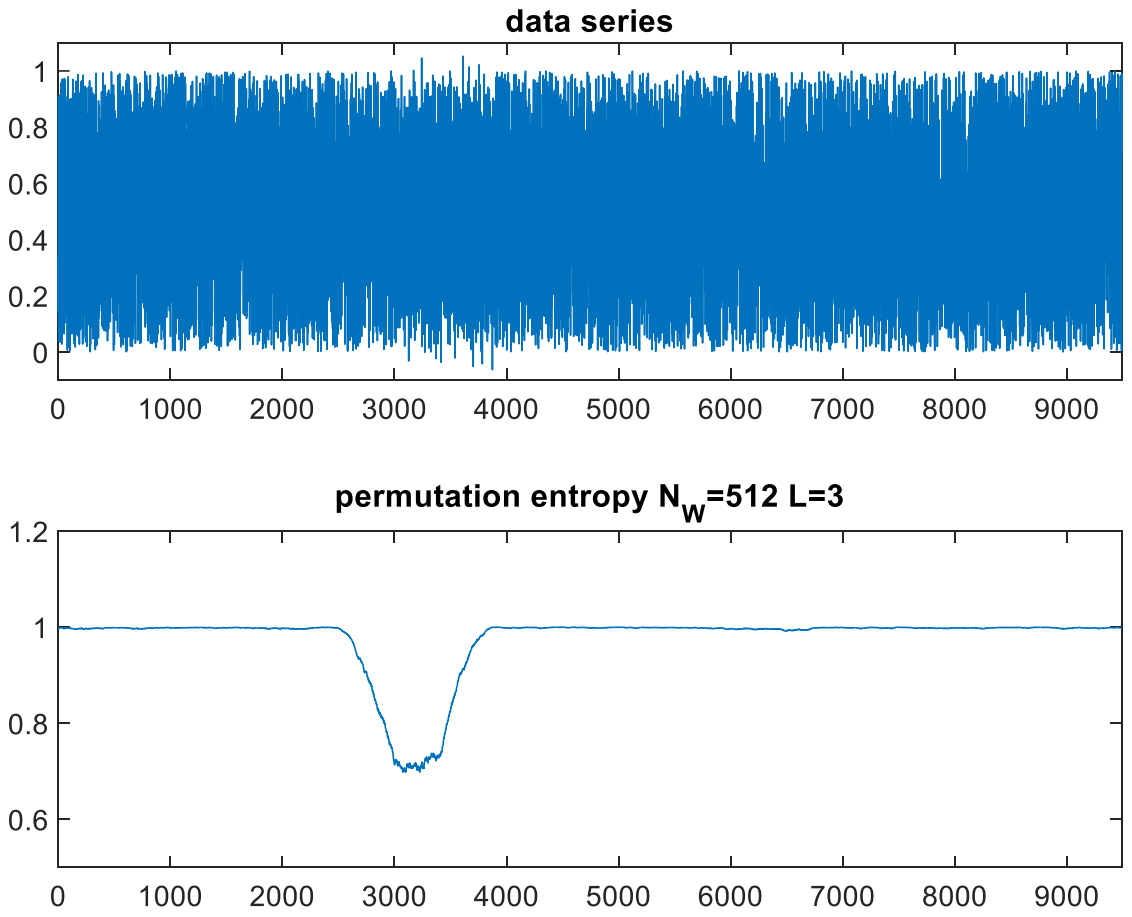
- **Describe how you generated the correlated data**

NORMALIZED (MAX = 1)

- Compute the permutation entropy with the same N_W and m used before.

- Plot the data and the entropy.

- Comment the results


data series
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
permutation entropy N_W=512 L=3
1.2
1
0.8
0.6
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

Some Matlab functions that might be useful

```
sort
perms
```

Exercise 7 - Information Gain Ratio and Classifier (pt. 7)

$\underline{V} = (x_1 \; x_2 \; x_3)$

Given this Training Set

v = [20, 0, 10;
20, 0, 70;
20, 1, 20;
20, 1, 80;
40, 0, 40;
40, 1, 60;
40, 1, 50;
20, 0, 60;
60, 1, 60];

CLASS c = [0; 0; 1; 0; 1; 1; 1; 0; 1];

SET THRESHOLDS FOR EACH FEATURE

E.G.

x_1 : 30, 50

x_2 : 0.5

x_3 : 15, 30, 45, 55, 65, 75

Builds a Decision Tree Classifier:
Write a program (no predefined classifiers) to implement a decision tree classifier based on the **Information Gain Ratio**. You don't need to create a generalized solution for any training set; it's sufficient if it works for the given training set with the specified three features.

Accepts Input:
The program should accept an input vector $v=(x_1,x_2,x_3)$, either from the keyboard, or a graphical interface, and outputs the corresponding predicted class.

Report:
•In the report, for each feature selection, write the computed **information gain ratios** for the considered features (without these numbers the exercise will not be counted).

•**Plot the tree classifier** generated.

•Verify if all the vectors of the training set are correctly classified by the tree.
•Explain what is the confusion matrix of a classifier.

•Present at least **four examples** of class predictions for four input vectors not belonging to the training set.

Important

Final version assigned on 08/10/2024

Delivery by

- 27/10/2024, 11.59 PM: +2 points
- 03/11/2024, 11.59 PM: +1 point
- 10/11/2024, 11.59 PM: 0 points
- **Later: not accepted**