Data Mining

INFS 4203/7203

Miao Xu

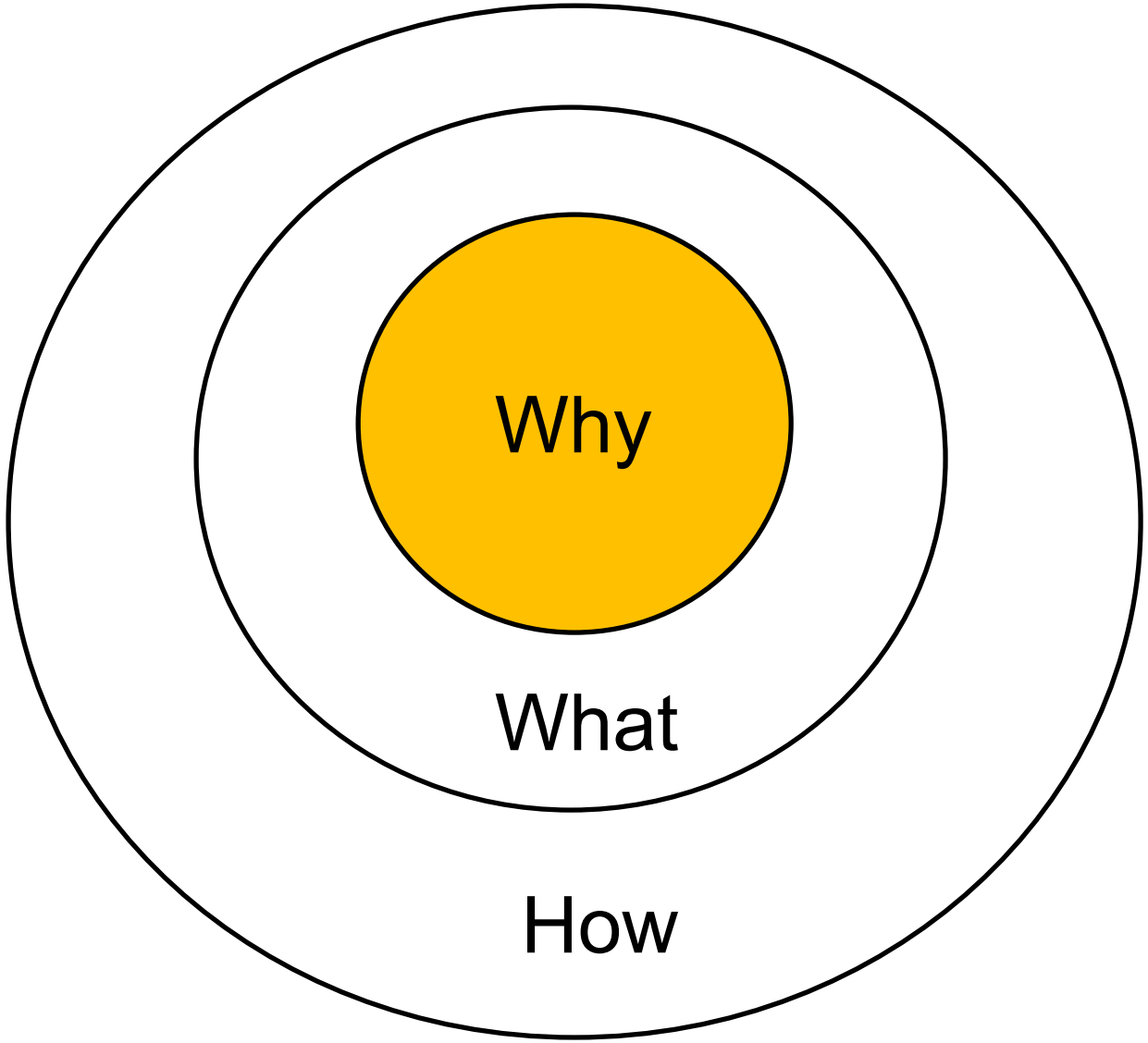
miao.xu@uq.edu.au

The University of Queensland, 2020 Semester 2

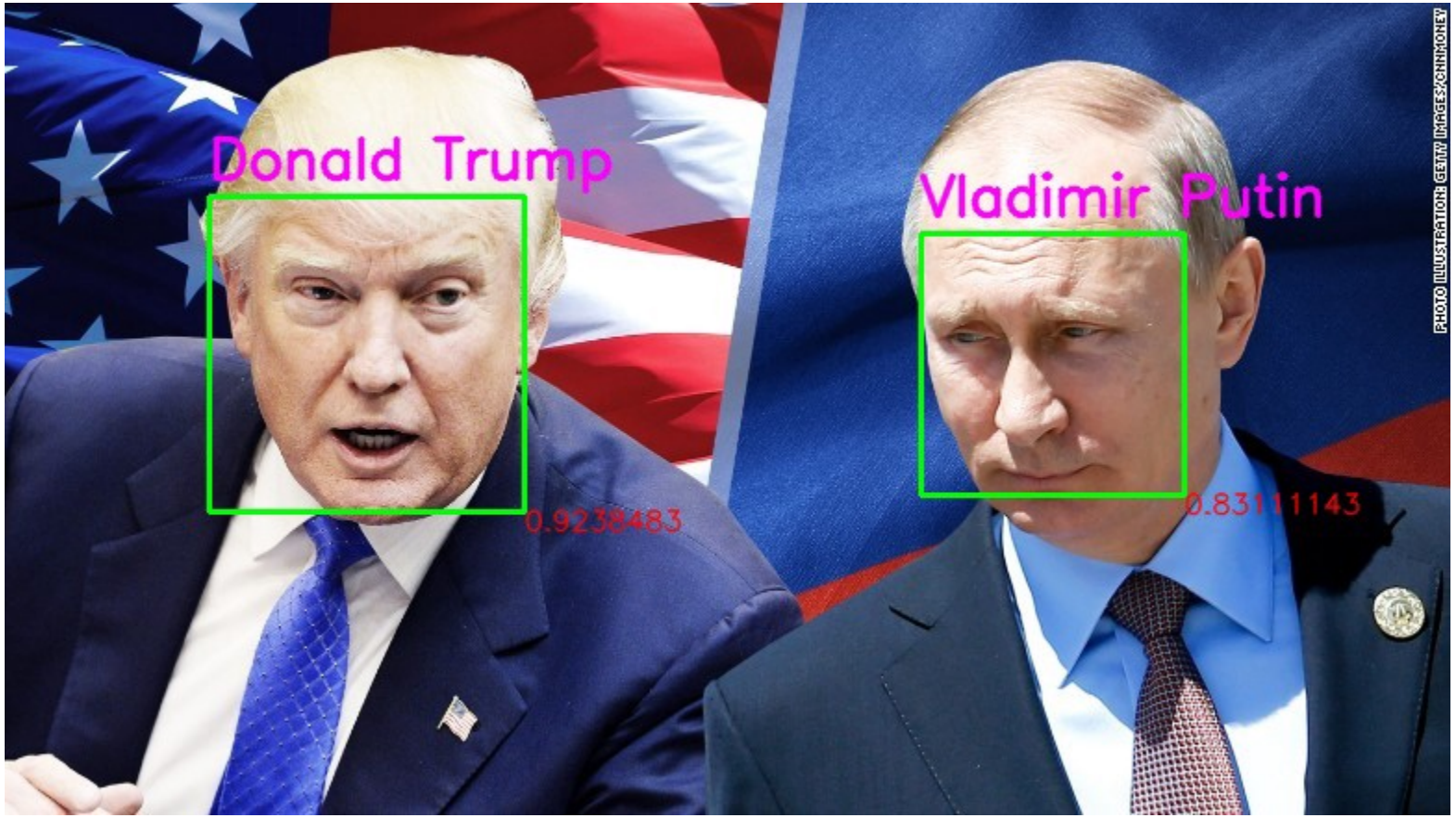
- Mid-term: under marking
- Final-term: scheduled on Nov. ~~3rd~~ 19th
- Form: BB quiz, invigilated
- What we will do in preparing:
  - Summarize contents that are important at the end of each lecture
  - Provide BB quiz for practicing calculation problems before each tutorial (not marked)
  - A practice exam in the same form
  - A review at the end of the course

Definition of classification

- Given a collection of records (or called samples/training set)
- Each record (or sample) contains:
  - An object described by a set of attributes (or features/input variables)
  - A label (or class/output variable)
- Find a model, such that the label is a function of the values of all other attributes


Why
What
How

Face recognition



- Collect training data

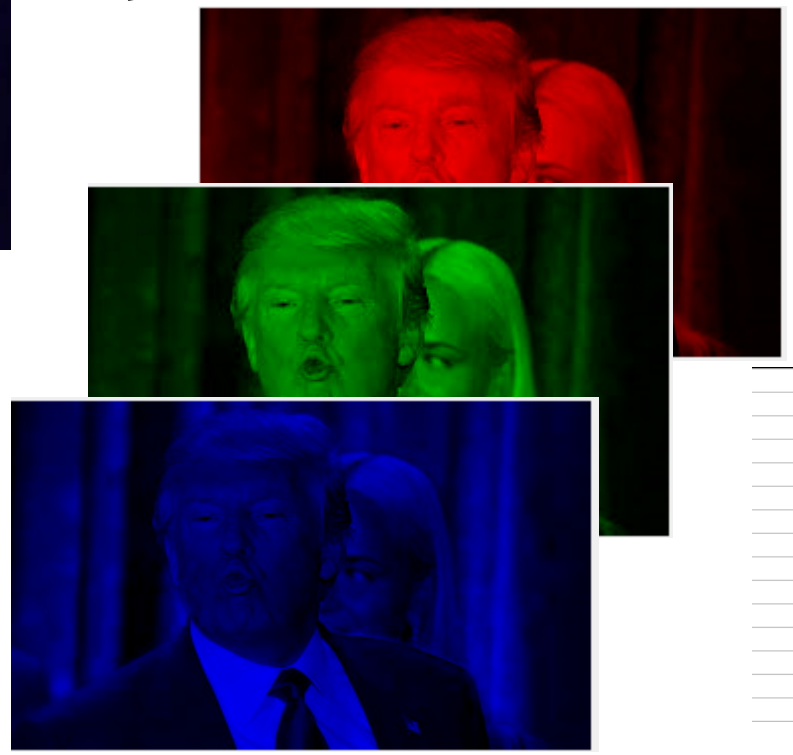
President Trump

President Putin

President Trump

290*174 Pixels

Label: Trump



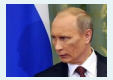
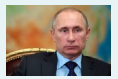
					7	14	25	38	48	55	55	54
					7	14	27	38	48	55	55	54
					7	14	26	39	49	55	55	55
					9	15	26	39	49	55	57	55
					9	15	28	39	49	56	57	55
					10	15	28	39	49	56	58	56
					10	17	28	41	51	56	58	56
				1	4	9	13	17	19	21	23	60	58
				1	4	8	13	17	19	21	23	62	60
				1	4	7	12	16	19	21	23	63	61
				0	3	7	12	16	19	21	23	64	62
				0	3	7	12	16	18	21	23	65	63
				0	3	7	12	16	18	20	22	66	65
				0	2	7	11	15	18	20	22	67	65
				0	2	7	12	16	20	22	24	68	66
1	5	12	19	25	29	35	38	21	23				
1	5	12	19	25	29	35	38	22	24				
1	5	11	19	25	29	35	38	23	25				
1	5	11	19	25	29	35	38	24	26				
1	5	12	19	25	29	35	38	25	26				
1	5	12	19	25	29	35	38	24	26				
1	5	12	19	25	29	35	38	25	27				
1	5	12	20	26	31	37	40						
1	5	12	20	27	31	37	40						
2	6	13	21	27	32	38	41						
2	7	14	22	28	33	39	42						
3	8	15	23	29	34	40	43						
4	9	16	23	30	35	41	44						
5	9	16	24	31	35	41	44						
3	8	15	25	31	36	42	45						

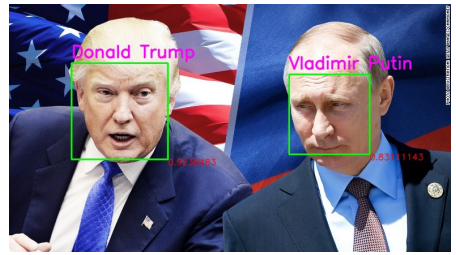
Feature extraction

$$\left(7, 14, \ldots, 36, 42, 45 \right)$$

Attributes

training set

	Attribute1		Attribute d	Label
	1	…	23	Trump
	3	…	45	Trump
	3	…	33	Trump
	1	…	67	Trump
	7	…	45	Trump
	3	…	92	Trump
	1	…	12	Putin
	5	…	23	Putin
	3	…	32	Putin
	3	…	33	Putin
	1	…	33	Putin
	3	…	92	Putin

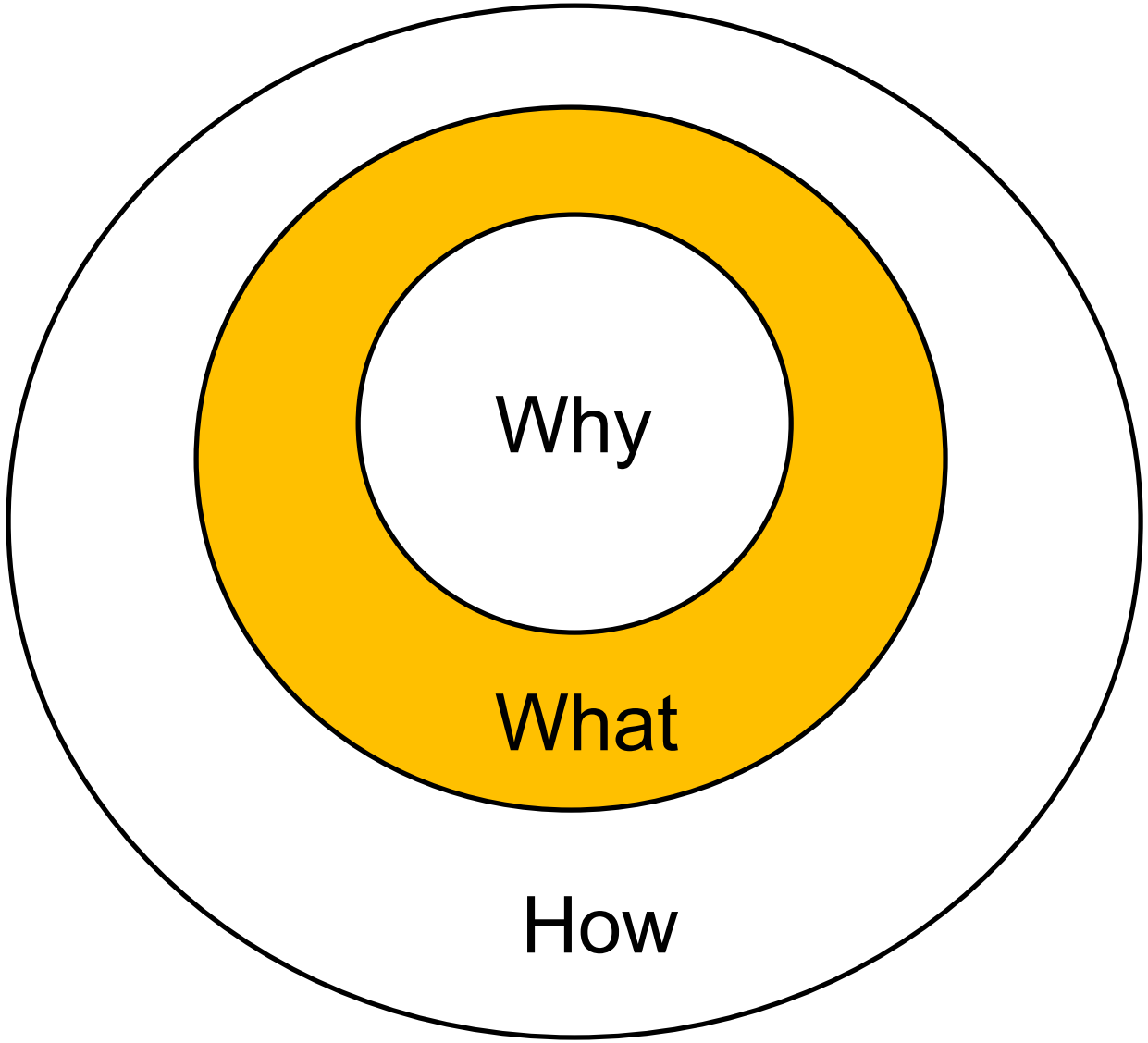


Emotion recognition

	Attribute1		Attribute d	Label
	1	…	23	Sadness
	3	…	45	Silent
	3	…	33	Happiness
	1	…	67	Happiness
	7	…	45	Anger
	3	…	92	Anger
…	…	…	…	…

Many other applications

- Image recognition
- Speech recognition
- Medical diagnosis
- Weather prediction
- Spam email recognition
- ……


Why
What
How

More on definition

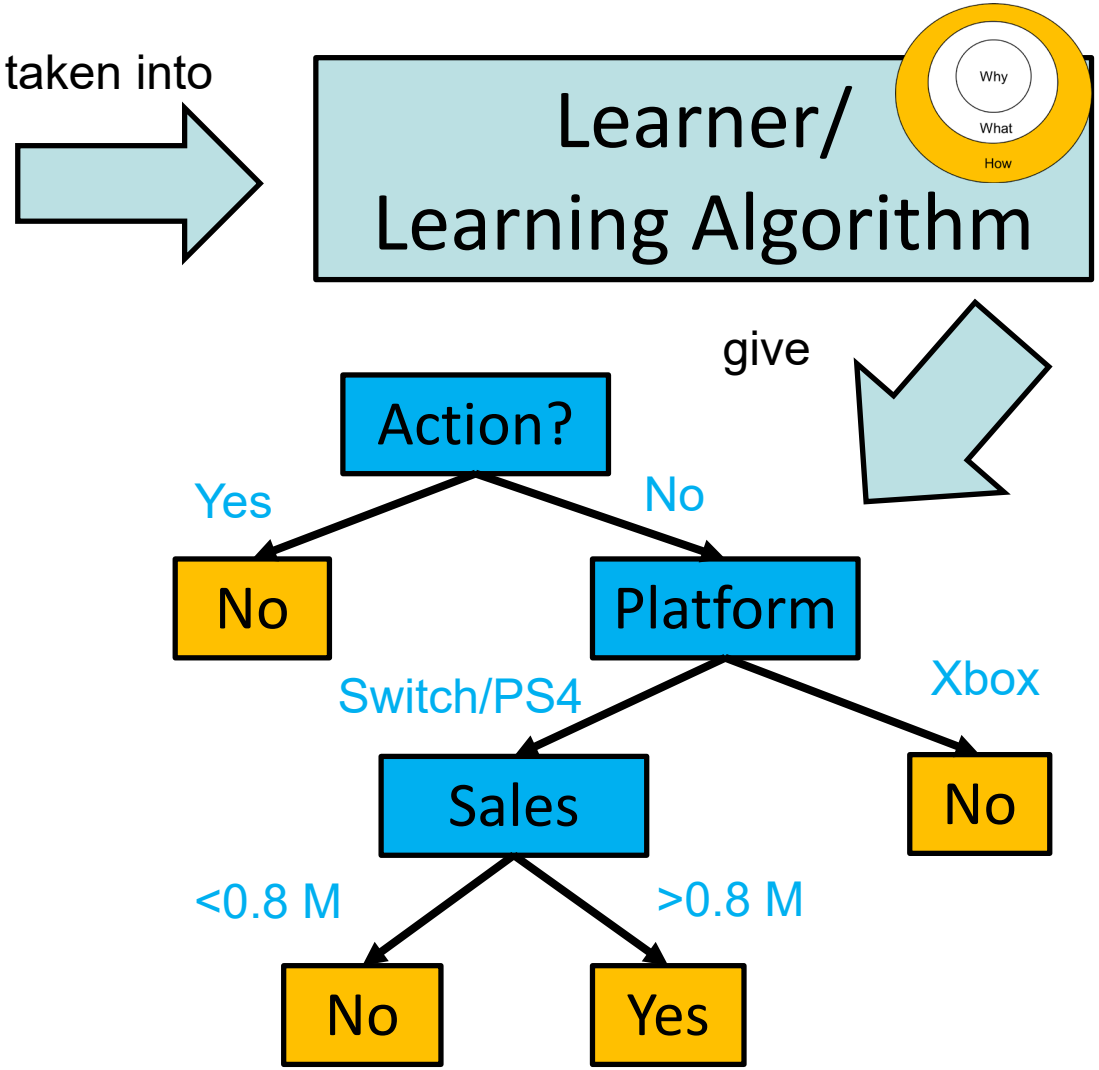
- Training data/set: a collection of records (instances)
  - Objects described by attributes: denote one record by ***x***
  - Labels for these objects: denote one label by *y*
- Learner/training algorithm: an algorithm, which takes in all the training data (trained on all training data), and gives a classifier (classification model)
- Classifier/classification model: a model, which is a mapping from ***x*** to *y*

Objective: the model can assign a class to unseen records as accurate as possible

Framework: an example

TID	Action?	Platform	Sales	Buy?
1	Yes	Switch	1.25M	No
2	No	Xbox	1M	No
3	No	Switch	0.7M	No
4	Yes	Xbox	1.2M	No
5	No	PS4	0.95M	Yes
6	No	Xbox	0.6M	No
7	Yes	PS4	2.2M	No
8	No	Switch	0.85M	Yes
9	No	Xbox	0.75M	No
10	No	Switch	0.9M	Yes

x y



Training data/set: $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), (\boldsymbol{x}_4, y_4), (\boldsymbol{x}_5, y_5), (\boldsymbol{x}_6, y_6), (\boldsymbol{x}_7, y_7), (\boldsymbol{x}_8, y_8), (\boldsymbol{x}_9, y_9), (\boldsymbol{x}_{10}, y_{10})\}$, each $(\boldsymbol{x}_i, y_i)$ is called a training instance

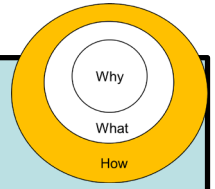
Framework: an example

	Action?	Platform	Sales	?
1	Yes	Switch	1.25M	2
2	No	Xbox	1M	No
3	No	Switch	0.7M	No
4	Yes	Xbox	1.2M	No
5	No	PS4	0.95M	Yes
6	No	Xbox	0.6M	No
7	Yes	PS4	2.2M	No
8	No	Switch	0.85M	Yes
9	No	Xbox	0.75M	No
10	No	Switch	0.9M	Yes

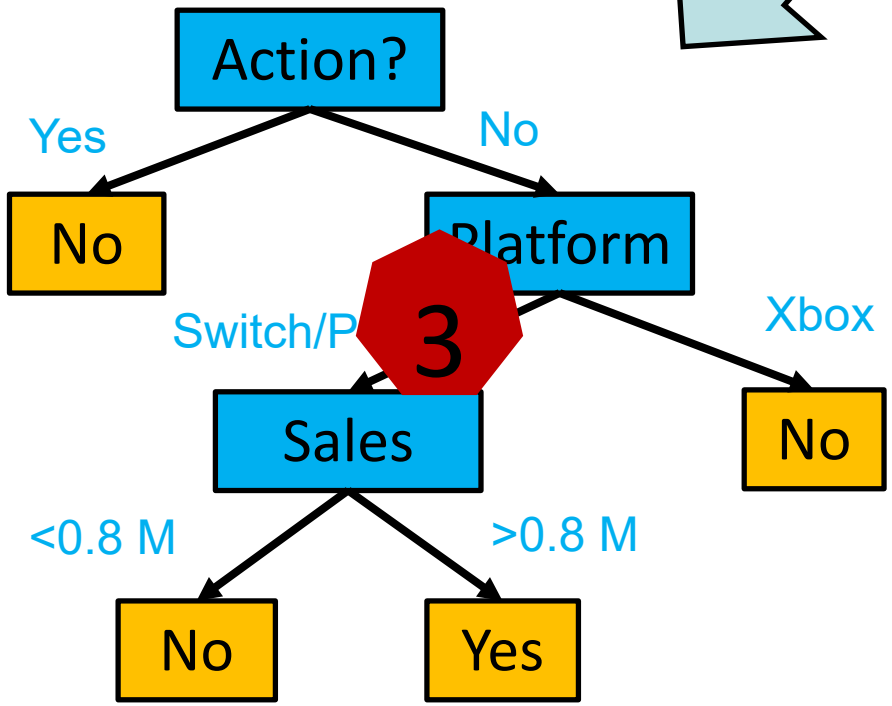
x y

taken into

Learner/
Learning Algorithm



give



1. Input variables (or called features/attributes):
   - Missing values
   - Normalization
   - Curse of dimensionality
2. Output variable (or labels/supervised information)
   - Number of classes
3. Classification model/classifier
   - Generalization and overfitting
   - Cross-validation
   - Evaluation

Input variables: missing values 1

- Some of the values are not known in the input variables:
  - broken equipment, privacy…
- Recommended way to deal with missing values:
  - Ignore all data with missing values
    - If the remaining data become too small, not good!
  - Imputation of missing values by the attribute's "all" values
  - Imputation of the missing values by the attribute's "class-specific" values

More advanced methods: https://sci2s.ugr.es/MVDM

Input variables: missing values 2.1

Imputation of missing values by the attribute's all values

- Nominal/Ordinal feature: most common attribute value
- Numerical feature: average value

TID	Action?	Platform	Sales
1	Yes	Switch	1.25M
2	No	Xbox	1M
3	No	Switch	0.7M
4	Yes	?	1.2M
5	No	PS4	0.95M
6	No	Xbox	0.6M
7	Yes	PS4	2.2M
8	No	Switch	0.85M
9	No	Xbox	0.75M
10	No	Switch	0.9M

What will be the imputation of "?" by attribute's all values?

1. Attribute "Platform" is a nominal feature
2. For the other 9 instances, Switch 4 times, Xbox 3 times, PS4 2 times
3. Most common attribute value: Switch
4. Imputation "?" as Switch

Input variables: missing values 2.2

Imputation of missing values by the attribute's all values

- Nominal/Ordinal feature: Most Common Attribute Value
- Numerical feature: Average Value

TID	Action?	Platform	Sales
1	Yes	Switch	1.25M
2	No	Xbox	1M
3	No	Switch	0.7M
4	Yes	Xbox	?
5	No	PS4	0.95M
6	No	Xbox	0.6M
7	Yes	PS4	2.2M
8	No	Switch	0.85M
9	No	Xbox	0.75M
10	No	Switch	0.9M

What will be the imputation of "?" by attribute's all values?

1. Attribute "Sales" is a numerical feature
2. For the other 9 instances, the average of them is 1.02M
3. Imputation of "?" is 1.02M

Input variables: missing values 3.1

Imputation of missing values by the attribute's class-specific values

- Nominal/Ordinal feature: most common class-specific attribute value
- Numerical feature: class-specific average value

TID	Action?	Platform	Sales	Buy?
1	Yes	Switch	1.25M	**No**
2	No	Xbox	1M	**No**
3	No	Switch	0.7M	**No**
4	Yes	**?**	1.2M	**No**
5	No	PS4	0.95M	Yes
6	No	Xbox	0.6M	**No**
7	Yes	PS4	2.2M	**No**
8	No	Switch	0.85M	Yes
9	No	Xbox	0.75M	**No**
10	No	Switch	0.9M	Yes

What will be the imputation of "?" by attribute's class-specific values?

1. Attribute "Platform" is a nominal feature
2. For the other 9 instances, 6 of them share the same class "No"
3. Among these 6 instances, Switch 2 times, Xbox 3 times, PS4 1 times
4. Most common class-specific attribute value: Xbox
5. Imputation of "?" is Xbox

Input variables: missing values 3.2

Imputation of missing values by the attribute's class-specific values

- Nominal/Ordinal feature: most common class-specific attribute value
- Numerical feature: class-specific average value

TID	Action?	Platform	Sales	Buy?
1	Yes	Switch	1.25M	**No**
2	No	Xbox	1M	**No**
3	No	Switch	0.7M	**No**
4	Yes	Xbox	**?**	**No**
5	No	PS4	0.95M	Yes
6	No	Xbox	0.6M	**No**
7	Yes	PS4	2.2M	**No**
8	No	Switch	0.85M	Yes
9	No	Xbox	0.75M	**No**
10	No	Switch	0.9M	Yes

What will be the imputation of "?" by attribute's class-specific values?

1. Attribute "Sales" is a numerical feature
2. For the other 9 instances, 6 of them share the same class "No"
3. Among these 6 instances, the average is 1.08M
4. Imputation of "?" is 1.08M

More advanced methods: https://sci2s.ugr.es/MVDM

The input variables: normalization

- Different attributes can have different degrees of magnitude
- Some methods can be highly impacted
- Two ways to do normalization:

TID	Mode Age (year)	Price (AUD)	Sales
1	24	25	1.25M
2	32	90	1M
3	15	0	10M
4	22	30	0.8M
5	48	100	0.5M

Max-Min normalization:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Standardization:

$$x' = \frac{x - \mu}{\sigma}$$

The input variables: max-min normalization

TID	Mode Age (year)	Price (AUD)	Sales
1	24	25	1.25M
2	32	90	1M
3	15	0	10M
4	22	30	0.8M
5	48	100	0.5M

Max-Min normalization:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

For Mode Age: max 48, min 15

1. (24-15)/(48-15) = 0.27
2. (32-15)/(48-15) = 0.51
3. (15-15)/(48-15) = 0
4. (22-15)/(48-15) = 0.21
5. (48-15)/(48-15) = 1

For Price: max 100, min 0

1. (25-0)/(100-0) = 0.25
2. (90-0)/(100-0) = 0.9
3. (0-0)/(100-0) = 0
4. (30-0)/(100-0) = 0.3
5. (100-0)/(100-0) = 1

The input variables: standardization

TID	Mode Age (year)	Price (AUD)	Sales
1	24	25	1.25M
2	32	90	1M
3	15	0	10M
4	22	30	0.8M
5	48	100	0.5M

Standardization:

$$x' = \frac{x - \mu}{\sigma}$$

TID	Mode Age (year)	Price (AUD)	Sales
1	-0.33	-0.55	-0.36
2	0.30	0.94	-0.42
3	-1.05	-1.12	1.78
4	-0.49	-0.45	-0.47
5	1.57	1.17	-0.54

The input variables: normalization

- Max-Min normalization is good when you know that the distribution of your data **does not** follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like *k*-NN.
- Standardization can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Standardization does not have a bounding range of [0, 1].

Finally, the choice of using max-min normalization or standardization will depend on your problem and the algorithm you are using.

The input variables: curse of dimensionality (1)

- **Dimensionality**: number of attributes
- When the dimensionality increases, the available data, which is sufficient to learn a satisfiable classifier in low-dimensionality case, become sparse;
- To learn a satisfiable classifier, the amount of data needed often grows exponentially with the dimensionality.

We need a lot of training data to learn if the data is high-dimensional

The input variables: curse of dimensionality (2)

For methods replying on the similarity between objects

- In high-dimensional case, calculate the distance is more difficult
- In high-dimensional case, all objects appear to be sparse and dissimilar—methods relying on the similarity between objects could fail

Distance is hard to calculate. Data points can be sparse and dissimilar in high-dimensional space.

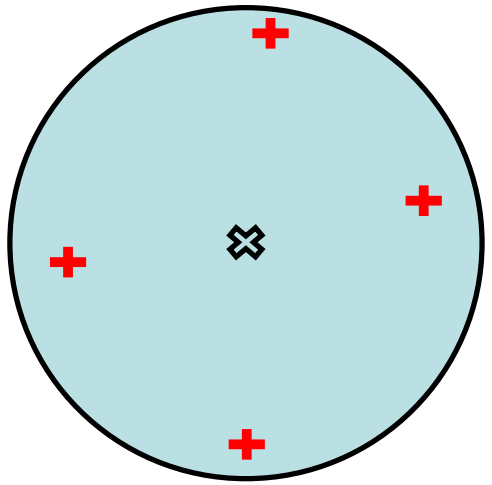
The input variables: curse of dimensionality (3)

Consider *N* data points uniformly distributed in a *p*-dimensional **unit ball** centred at the origin. The **median** distance from the origin to the closest data point is given by

$$distance = \left(1 - 2^{-\frac{1}{N}}\right)^{\frac{1}{p}}$$

Given $N = 500$, $p = 2$: 0.04

$p = 10$: 0.52, $p = 100$: 0.94



When p becomes large, most points are distributed far away from the origin. Actually, they are also far away from each other.

See https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12_toc.pdf Sec. 2.5 for more details

The output variables: number of classes (1)

Binary classification vs. multi-class classification

- Binary classification: classifying the objects into **2 classes** (yes/no, cat/dog, Trump/Putin…)
- Multi-class classification: classifying the objects into **multiple classes** (cat/dog/horse, happiness/sadness/anger…)

The output variables: number of classes (2)

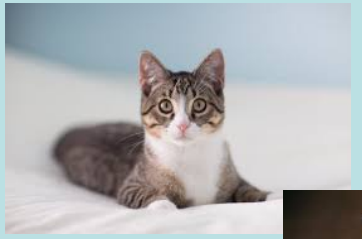
Binary classification





Multi-class classification




The output variables: number of classes (3)

How to transfer a multi-class classification into multiple binary classification?

- **One vs. all**: learn one binary classifier for each class;
  - Horse/not horse, dog/not dog, cat/not cat
  - Decide by confidence (a real value from the classifier)
- **One vs. one**: learn one binary classifier for each pair of classes
  - Horse/dog, dog/cat, cat/horse
  - Decide by vote

For both of these two methods, ambiguity may exist.

The model: generalization and overfitting (1)

- **Objective of classification:** the model can assign a class to unseen records as accurate as possible
- Generalization: the model's ability to classify unseen data
- Formally, assume the training data $\{(\boldsymbol{x}_i, y_i)\}\ (i = 1 \ldots n)$ are sampled from Distribution *D*, and there is a metric $\ell(f(\boldsymbol{x}), y)$ measuring the classification error of model $f(.)$
- Generalization error of the model is:

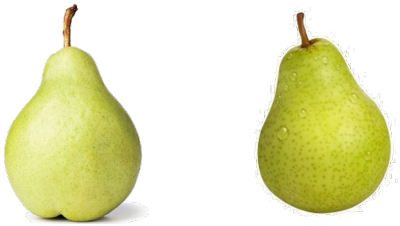
$$\ell(f) = E_{(\boldsymbol{x},y)\sim D}[\ell(f(\boldsymbol{x}), y)]$$

The model: generalization and overfitting (2)

- The good models we are looking for are those which good generalization, i.e. performing well on unseen new objects.
- Such models should be learned from the training samples.
- Overfitting: the phenomenon that a **model**, which has learned some **special characteristics** of the training data, does well on the training samples but not well on unseen new samples, resulting in reduced generalization.

The model: generalization and overfitting (3)

Example: a classifier has learned from the following data what is an apple:



Pear

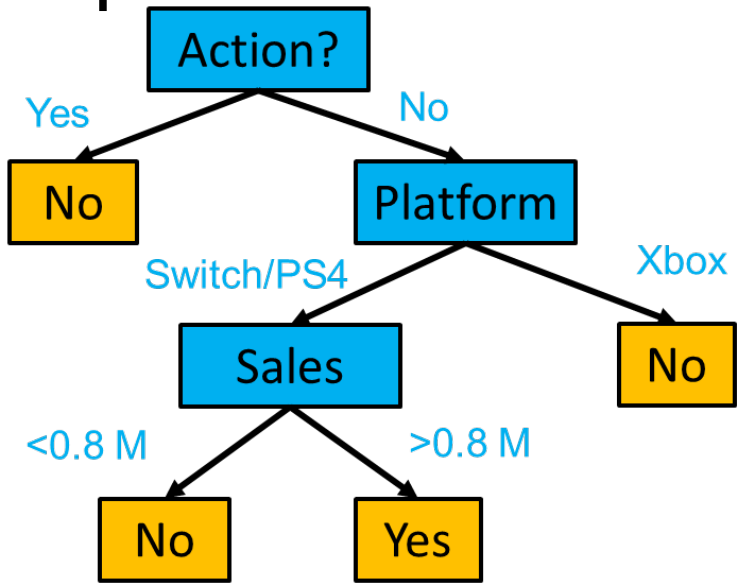
Apple

An overfitted classifier may think this as an apple

The model: cross-validation (1)

$$\ell(f) = E_{(x,y)\sim D}[\ell(f(\boldsymbol{x}), y)]$$

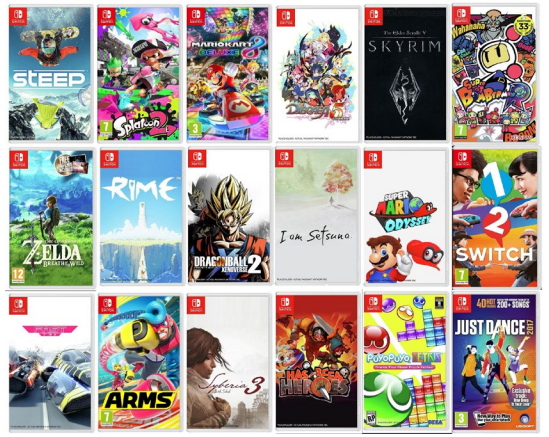
- How to calculate the “expectation on unseen data” in real life?
- Sometimes we are offered “test data”, and we want the performance of the model to be good on test data



Action? No

Platform: Switch

Sales: 2.32M



The model: cross-validation (2)

- How about if we do not have test data?
- We use **cross-validation (CV)** to evaluate the model!
- For *K*-folder cross validation:
  - For a training dataset D, we **randomly** divided it into *K* folders with almost equal number of instances in each folder: $\mathrm{D}_1, \ldots, \mathrm{D}_K$
  - Every time, D_k is used as testing data, and all others are used as training data
  - All the K test results are averaged to evaluate the performance

  Most of the time, K is set as 5, 10, 30…

The model: cross-validation (3)

- An example of 10-CV:

TID	Action?	Platform	Sales	Buy?
1	Yes	Switch	1.25M	No
2	No	Xbox	1M	No
3	No	Switch	0.7M	No
4	Yes	Xbox	1.2M	No
5	No	PS4	0.95M	Yes
6	No	Xbox	0.6M	No
7	Yes	PS4	2.2M	No
8	No	Switch	0.85M	Yes
9	No	Xbox	0.75M	No
10	No	Switch	0.9M	Yes

1. Train a classifier on TID 1-9,. And test on TID 10
2. Train a classifier on TID 1-8, 10,. And test on TID 9
3. Train a classifier on TID 1-7, 9-10,. And test on TID 8
4. Train a classifier on TID 1-6, 8-10,. And test on TID 7
5. Train a classifier on TID 1-5, 7-10,. And test on TID 6
6. Train a classifier on TID 1-4, 6-10,. And test on TID 5
7. Train a classifier on TID 1-3, 5-10,. And test on TID 4
8. Train a classifier on TID 1-2, 4-10,. And test on TID 3
9. Train a classifier on TID 1, 3-10,. And test on TID 2
10. Train a classifier on TID 2-10,. And test on TID 1

Each of the classifier will have its test results on its specific test data.
The overall performance is evaluated as the average test results of these ten trials.

In this example, because each folder only contains one instance, sometimes called LOO (leave-one-out).

The model: cross-validation (4)

- Sometimes, CV is also used to determine the hyper-parameters of the learner
- **Hyper-parameter**: parameters need to be determined for training before the training starts
  - For a training dataset D, we randomly divided it into *K* folders with almost equal number of instances in each folder: $\mathrm{D}_1, \ldots, \mathrm{D}_K$
  - Every time, D_k is used as **validation** data, and all others are used as training data
  - Different hyper-parameters are used to train a classifier, and the model are then evaluated on the validation data
  - For each hyper- parameter, all the K validation results are averaged
  - The hyper-parameter with the best average validation result is picked

The model: evaluation

- Compare the true labels (called "**ground truth**"), and the predicted labels by the classifier
- A metric is used to evaluate the difference
- Most commonly used one: accuracy

Accuracy and zero-one error

Buy?	Buy?	
No	Yes	✗
No	Yes	✗
No	Yes	✗
No	Yes	✗
Yes	Yes	✓
No	No	✓
No	No	✓
Yes	No	✗
No	No	✓
Yes	No	✗

True labels

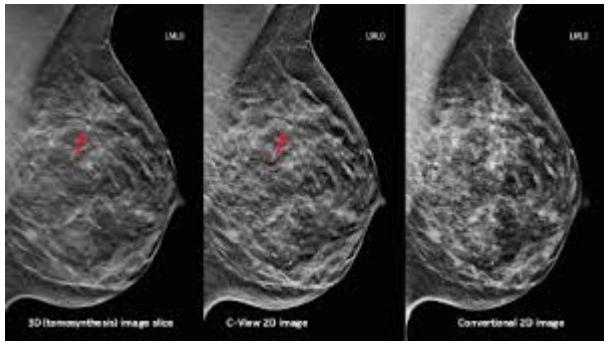
Predicted labels

- Accuracy: the rate of correct prediction:
- zero-one error: 1- accuracy

Accuracy: 0.4 = 40%
zero-one error: 0.6

Limitation of accuracy

- Assume a classification task of breast cancer detection from medical images



- Among 10,000 people detected, only 5 are diagnosed as breast cancer
- If a classifier treating all people as “not cancer”, what is the accuracy?

99.95%

Other metrics are needed to evaluate the classifier

Confusion matrix (1)

- In binary classification, we usually call one class "positive" and another class "negative"
  - E.g. cancer (positive), not cancer (negative)
  - E.g. Yes(positive), No (negative)
- A confusion matrix: measure the number of tested examples in

	Predicted positive	Predicted negative
Ground truth positive	TP (true positive)	FN (false negative)
Ground truth negative	FP (false positive)	TN (true negative)

Confusion matrix (2)

- Consider the breast cancer example
  - Among 10000 people, 5 are breast cancer (positive)
  - An algorithm predict all as “negative”

	Predicted positive	Predicted negative
Ground truth positive	0 (true positive)	5 (false negative)
Ground truth negative	0 (false positive)	9995 (true negative)

Precision and recall

- Precision: in all predicted as positive samples, the rate of those that are actually “positive”

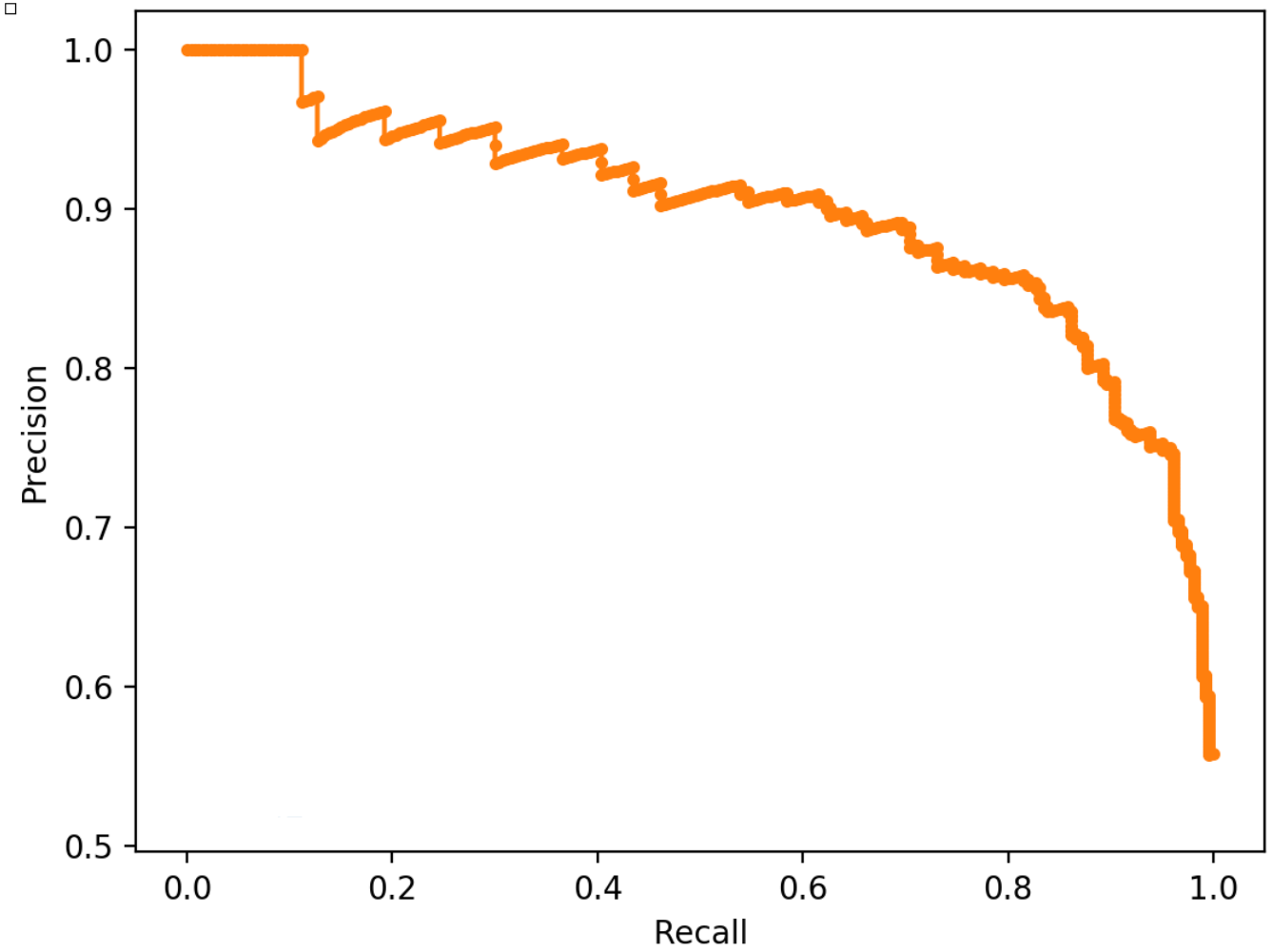
$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall: in all positive samples, the rate of those predicted as “positive”

$$\text{Recall} = \frac{TP}{TP + FN}$$

The relationship between precision and recall

- An illustration of precision-recall curve



The F1 measurement

- The harmonic mean of precision and recall (the higher the better, max value: ?) 1

$$\mathrm{F1} = \frac{2}{\frac{1}{\mathrm{Precision}} + \frac{1}{\mathrm{Recall}}} = \frac{2\mathrm{TP}}{n + TP - TN}$$

Exercise: calculate precision, recall and F1

Buy?	Buy?	
No	Yes	✗
No	Yes	✗
No	Yes	✗
No	Yes	✗
Yes	Yes	✓
No	No	✓
No	No	✓
Yes	No	✗
No	No	✓
Yes	No	✗

True labels Predicted labels

Accuracy: 0.4 = 40%
zero-one error: 0.6

Confusion matrix:

	Predicted positive	Predicted negative
Ground truth positive	1	2
Ground truth negative	4	3

Precision: 1/(1+4) = 0.2

Recall: 1/(1+2) = 0.3

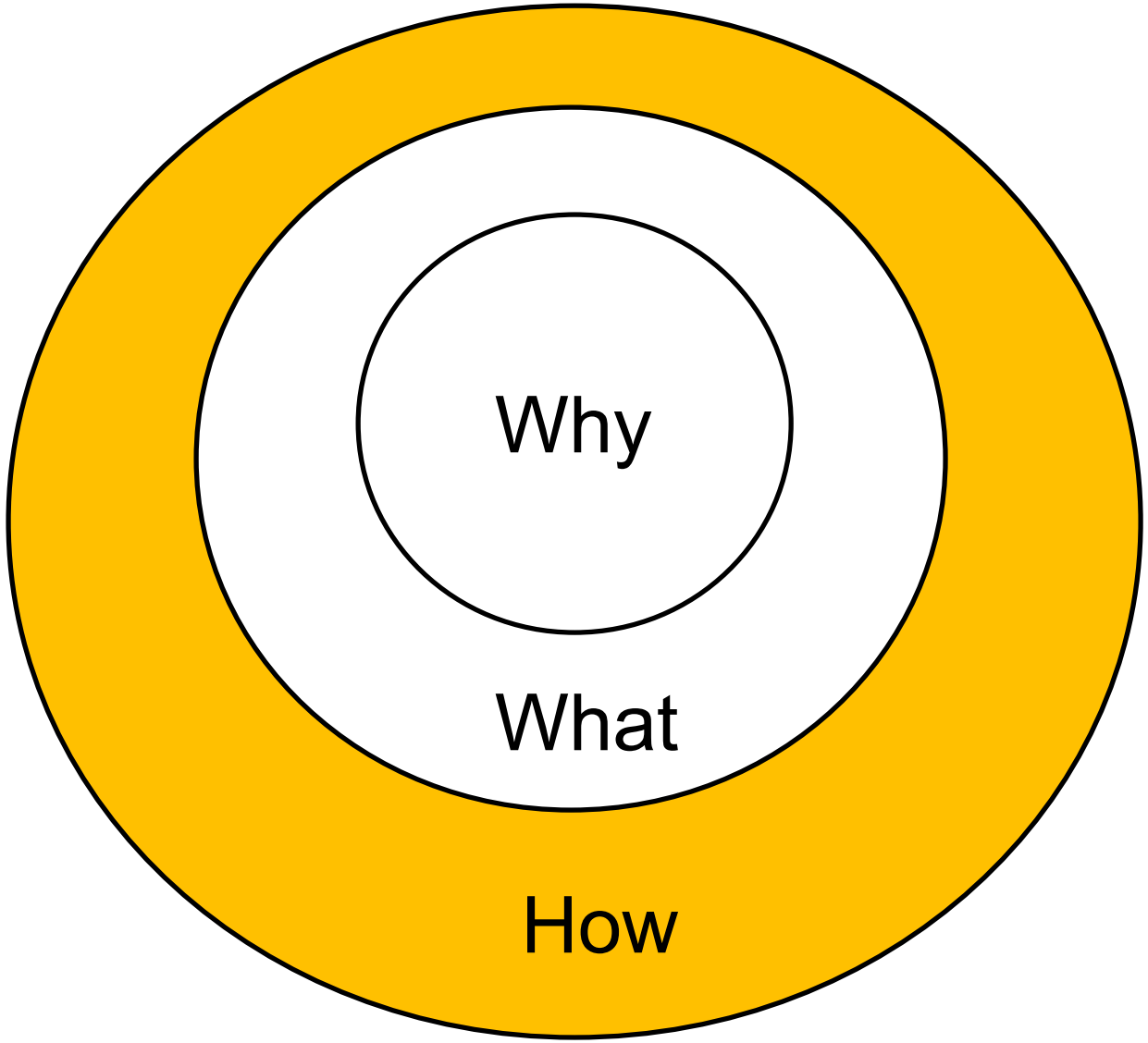
F1:

$$\frac{2}{\frac{1}{0.2}+\frac{1}{0.3}} = 0.25$$

Summary

- The concept:
  - Classification
  - Curse of dimensionality
  - Generalization
  - Overfitting
  - Cross validation
- The calculation and the concept:
  - Accuracy, recall, precision, F1

Next week



Recommended reading

- There is no text book reading recommended this week.

- Recommended readings are provided inside the slides.