Classification and Evaluation of Classifiers

Fundamentals of Machine Learning (G6061)

Dr. Benjamin Evans

University of Sussex

Recap of Previous Lecture

https://www.pollev.com/bdevans

Which of these is NOT a Machine Learning task?

- Classification — 0%
- Matrix Multiplication — 0%
- Regression — 0%
- Clustering — 0%
- None of the above — 0%

In ML, we use a model to solve a task by:

- Providing a framework for it to learn patterns from the data. — 0%
- Giving precise instructions on how to solve the task. — 0%
- Describing to the model the relationship between input variables. — 0%
- Manually setting the model parameters. — 0%
- None of the above — 0%

Which of these are (potentially) valid features for an ML algorithm?

Council Tax Band	0%
Wind Direction	0%
Age (in years)	0%
All of the above	0%
None of the above	0%

In ML, Regression tasks involve:

Predicting one of a discrete set of options given some input data. 0%

Recommending new things (e.g. products or films) based on previous interests. 0%

Predicting continuous values given some input data. 0%

Grouping unlabelled input data into clusters. 0%

None of the above. 0%

Models trained with labelled outputs for each input instance are called:

Semi-supervised	0%
Supervised	0%
Clustering	0%
Unsupervised	0%
None of the above	0%

What model would you use to predict whether a patient has a diease or not given their medical records?

Nobody has responded yet.

Hang tight! Responses are coming in.

Outline

- Cover the classification task in some detail, introducing terminology and notation.
- Consider what metrics can be used to assess performance.
- Talk about how binary classifiers can be used in multi-class problems.

Learning Outcomes for Today

- Understand the fundamentals of the classification task and supervised learning.
- Learn about how ML models are evaluated, understand several classification performance metrics.

Instances and Instance Spaces

- Consider data to consist of a set of instances
  - an instance represents an object of interest
- Set of all possible instances is the instance space:
  - e.g. set of all email messages written in English
  - or, set of human faces

Notation

- We will denote the instance space by $\mathcal{X}$
- An individual instance will be denoted by x
- $x \in \mathcal{X}$

Labels and Label Spaces

- In supervised problems, each instance is associated with a label
- Set of all labels for a task is called the label space:

  class labels $\mathcal{C}$ in <u>classification</u> tasks., e.g. $\mathcal{C} = \{\text{frogs}, \text{badgers}\}$
  real numbers $\subseteq \mathbb{R}$ in <u>regression</u> tasks. e.g. pixel coordinate.

Notation

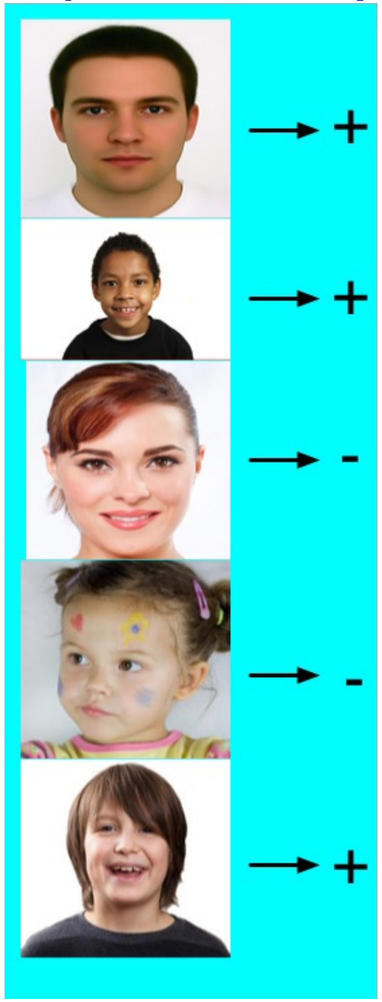
- We will denote arbitrary label space by $\mathcal{Y}$
- Labelling function $f : \mathcal{X} \to \mathcal{Y}$ maps from instances to labels
- Label associated with a given instance x denoted by $f(x)$

University of Sussex

Binary Classification

- In the simplest case, we just have two class labels
  - positive (+ or +1)
  - negative (- or -1)
- By convention the class of interest is labelled positive
- Binary classification task is to label instances with one or other of the class labels
- Can also be understood as concept identification
  - e.g. identifying faces in photographs

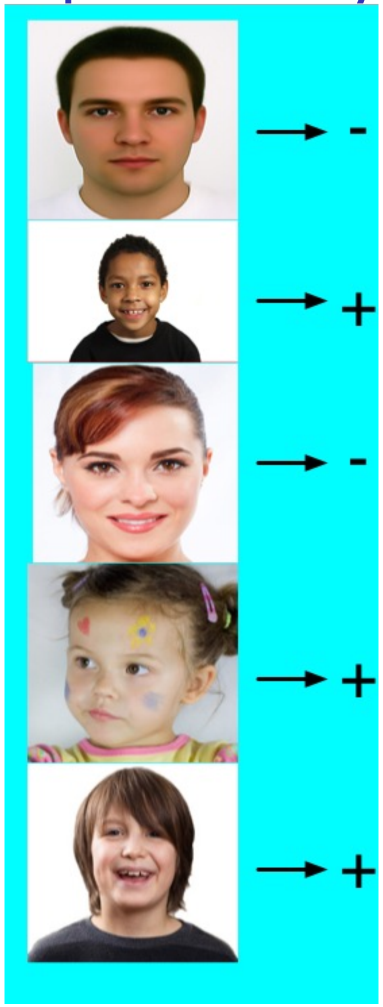
Example: Classifying Faces



- Binary classification of faces
- What concept does this correspond to?

University of Sussex

Example: Classifying Faces



- Binary classification of faces
- What concept does this correspond to?

University of Sussex

Learning a Classifier

- In practice, we do not know the true classification function f
- We have a training set of labelled instances, that is $(x, f(x))$
  - ▶ A set of manually annotated examples
- Use the examples to learn a model $\hat{f} : \mathcal{X} \rightarrow \mathcal{C}$
- $\hat{f}$ approximates the true classification function f
  - ▶ Should follow the training examples closely
  - ▶ Should generalise to whole of the instance space $\mathcal{X}$

Binary Classification: Assessing Performance

- For a learned binary classifier, $\hat{f}$ approximates the true classification function f
  - ► $\hat{f}$ will not exactly be the same function as f
  - ► $\hat{f}$ will make errors in assigning class labels to instances
- How can we assess the performance of a learned binary classifier?
  - ► What is a useful measurement to use?
    Hint: think about what we're going to ultimately do with the classifier.
    https://www.pollev.com/bdevans

What measures can we use to evaluate a classifier?

Nobody has responded yet.

Hang tight! Responses are coming in.

Binary Classification: Assessing Performance

- Use a contingency table (also known as confusion matrix)

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

Binary Classification: Accuracy and Error

- **Accuracy**: proportion of correctly predicted instances
- **Error**: proportion of incorrectly predicted instances

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

Accuracy on validation set (150 instances): (30 + 90) / 150 = 0.8 (80%)
Error rate on validation set: (10 + 20) / 150 = 0.2 (20%) = 1 – accuracy

University of Sussex

Binary Classifications: More useful metrics

Although the accuracy is important, it makes the unrealistic assumption that both classes are equally important to us.
Take the example of a face recognition system:

- Imagine the + class is one particular identity, so the - class is all others.
- We have a trade-off between always correctly identifying the + class, and mis-identifying the - class.
- If we err on the side of caution, i.e. always correctly find the + class, we will incorrectly find people from the -ve class.
- This can have major consequences! Examples of facial recognition use by the met police have not had positive responses.

Binary Classifications: More useful metrics

Although the accuracy is important, it makes the unrealistic assumption that both classes are equally important to us.

Take the example of a medical diagnosis system:

- Imagine: the + class is bad cancer (surgery), and
  the - class is benign (no surgery).
- We have a trade off between always correctly identifying the + class, and mis-identifying the - class.
- If we err on the side of caution, i.e. always correctly find the + class, we will incorrectly operate on people who would be fine otherwise, opening them up to unnecessary complications.
- Alternatively, we could be more optimistic and miss some malignant cases.
- The choice of metrics must be appropriate to the application.

Binary Classification: Per-Class Accuracy

- **True positive rate**:
  proportion of correctly predicted + instances
- **True negative rate**:
  proportion of correctly predicted - instances

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

TPR (**sensitivity**) on validation set: 30 / 50 = 0.6 (60%)
TNR (**specificity**) on validation set: 90 / 100 = 0.9 (90%)

Binary Classification: Per-Class Inaccuracy

- **False positive rate**:
  proportion of incorrectly predicted - instances
- **False negative rate**:
  proportion of incorrectly predicted + instances

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

FPR (1-TNR, **false alarms**) on validation set: 10 / 100 = 0.1 (10%)
FNR (1-TPR, **misses**) on validation set: 20 / 50 = 0.4 (40%)

Binary Classification: Precision and Recall

- **Precision**:
  proportion of + *predictions* that are correct
- **Recall**:
  proportion of + *instances* that are correctly predicted (TPR)

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

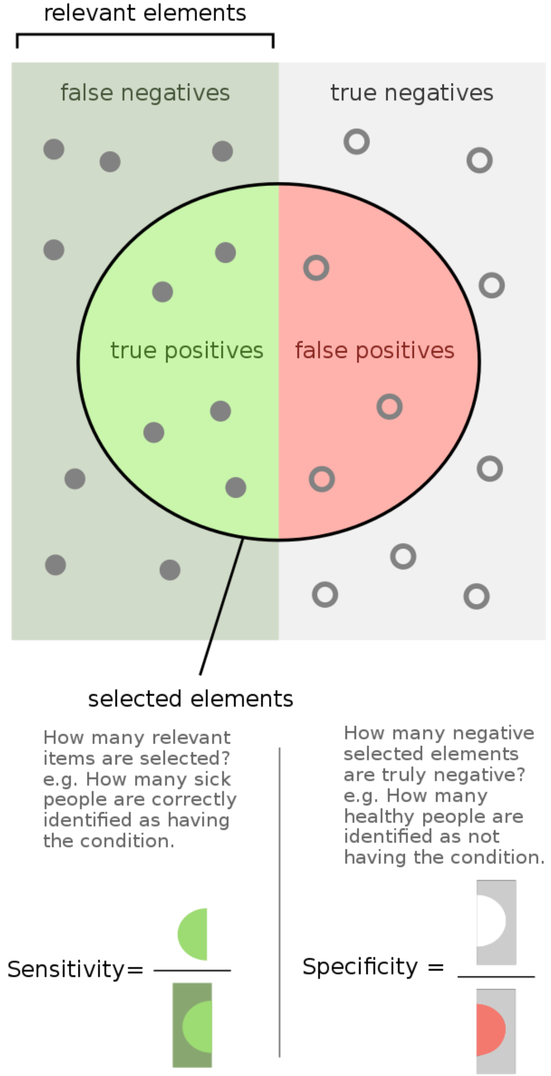
Precision on validation set: 30 / 40 = 0.75 (75%)

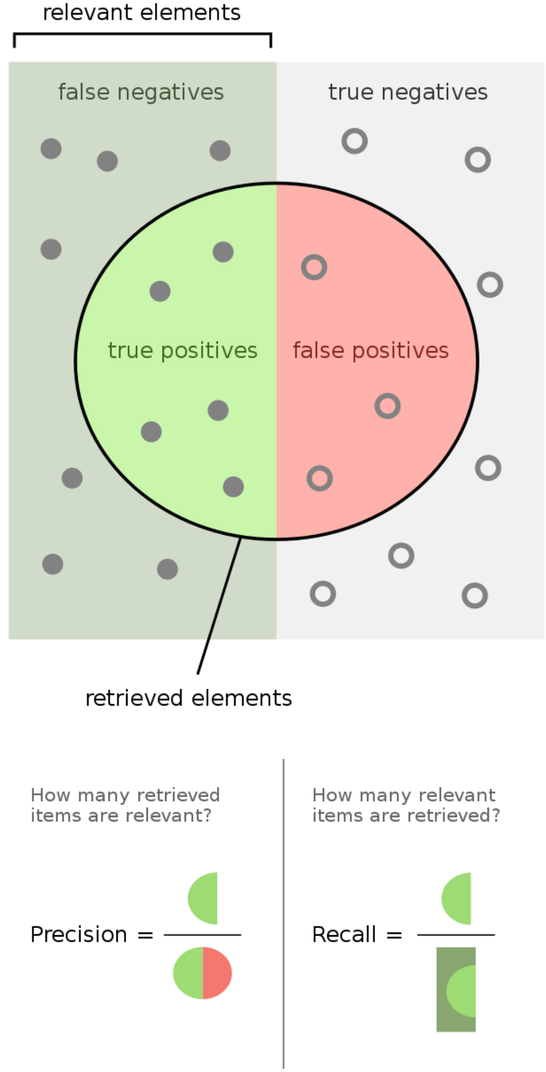
Note: normalise by *column* total

Recall on validation set: 30 / 50 = 0.6 (60%)

Note: normalise by *row* total

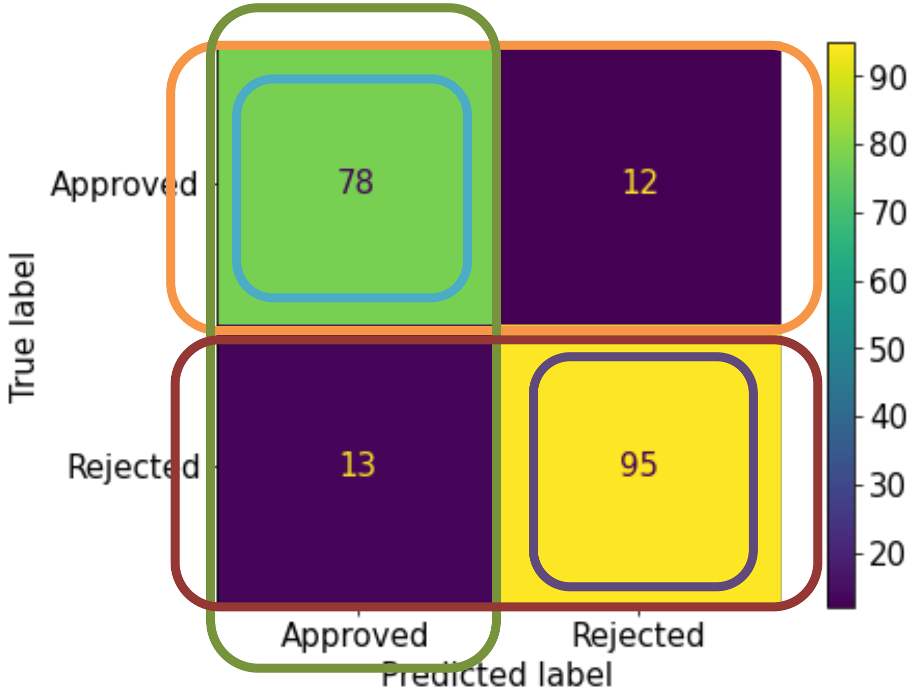
Binary Classification: Assessing Performance





💡 Tip: Remember the **+ve** class is conventionally the one of interest

Binary Classification: Assessing Performance



Also known as *Recall*

$$SENS = TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

$$= \frac{78}{(78 + 12)} = 0.867$$

$$SPEC = TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

$$= \frac{95}{(95 + 13)} = 0.880$$

$$Precision = \frac{TP}{TP + FP} = \frac{78}{(78 + 13)} = 0.857$$

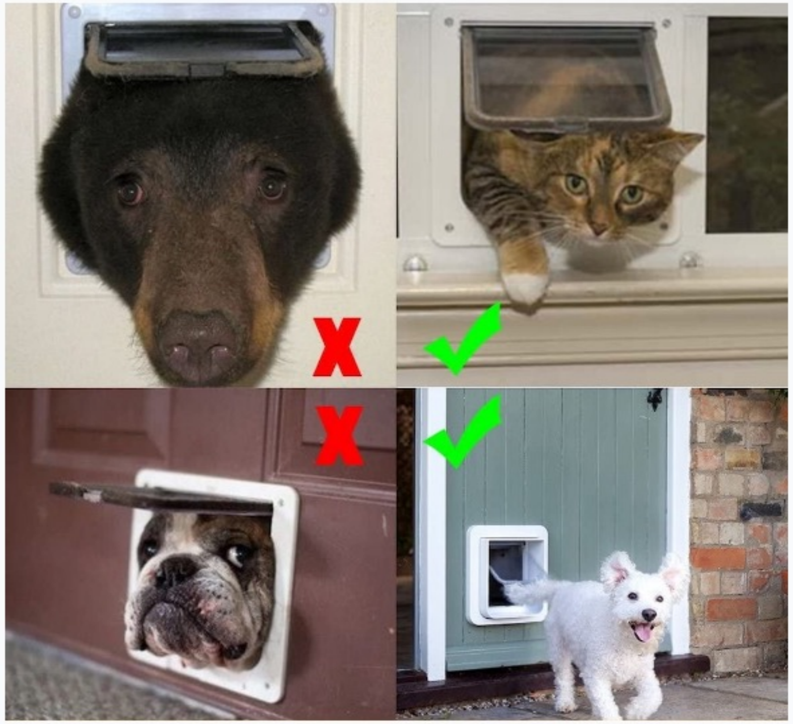
Quiz Time!

https://www.pollev.com/bdevans

In a dog 🐶 (+) / cat 🐱 (-) classifier, what does a false positive correspond to?

- A 🐶 incorrectly classified as a 🐱 — 0%
- A 🐱 incorrectly classified as a 🐶 — 0%
- A 🐶 correctly classified as a 🐶 — 0%
- A 🐱 correctly classified as a 🐱 — 0%
- A 🐱 classifying a 🐶 as problematic — 0%

Suppose you are building a "smart" pet-flap where your cat and dog are +ve instances and every other animal is -ve. What would happen if the system was tuned to be more specific than sensitive?



Your pets would get stuck outside in the rain sometimes

0%

Only doggos would be allowed in

0%

It would always open for cats

0%

Random animals would get in and urinate in your house

0%

What is the overall accuracy of this classifier?

		Predicted			
		Class 1	Class 2	Class 3	
Actual	Class 1	10	5	5	20
	Class 2	0	25	5	30
	Class 3	0	10	40	50
		10	40	50	100

0

0%

(10+0+0)/10

0%

(10+25+40)/100

0%

(10+5+5)/20

0%

(5+25+10)/40

0%

Confusion Matrix for a 3-class Problem

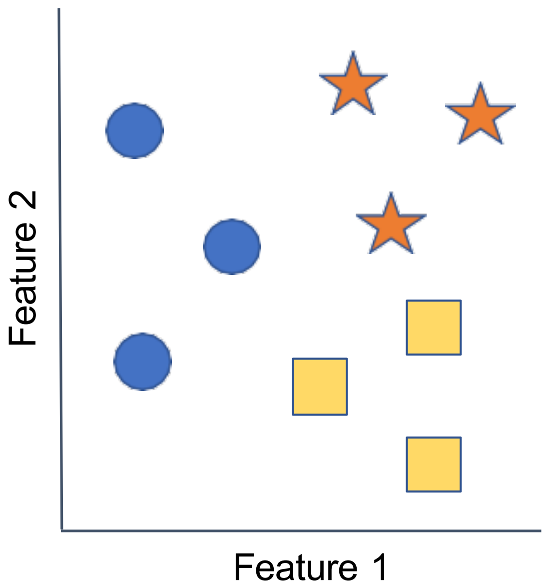
		Predicted			
		Class 1	Class 2	Class 3	
Actual	Class 1	10	5	5	20
	Class 2	0	25	5	30
	Class 3	0	10	40	50
		10	40	50	100

- What is the overall accuracy of the classifier?
  (10 + 25 + 40)/100 = 0.75

Multi-Class Classification

- Many classification tasks involve more than two classes:
  - ▶ Classifying newspaper article by topic
  - ▶ Recognising animal species from images
  - ▶ Predicting tomorrow's weather
- Some classifiers handle multiple classes naturally:
  - ▶ decision tree classifier and random forest
  - ▶ direct multi-class Support Vector Machine (SVM): Weston and Watkins (1999) and Cramer and Singer (2002) → however, typically one-versus-one SVM or one-versus-rest SVM is used in practice
  - ▶ multinomial logistic regression
- Other classifiers are fundamentally binary:
  - ▶ e.g. linear classifiers such as the perceptron classifier

Multi-Class Classification: OVR and OVO

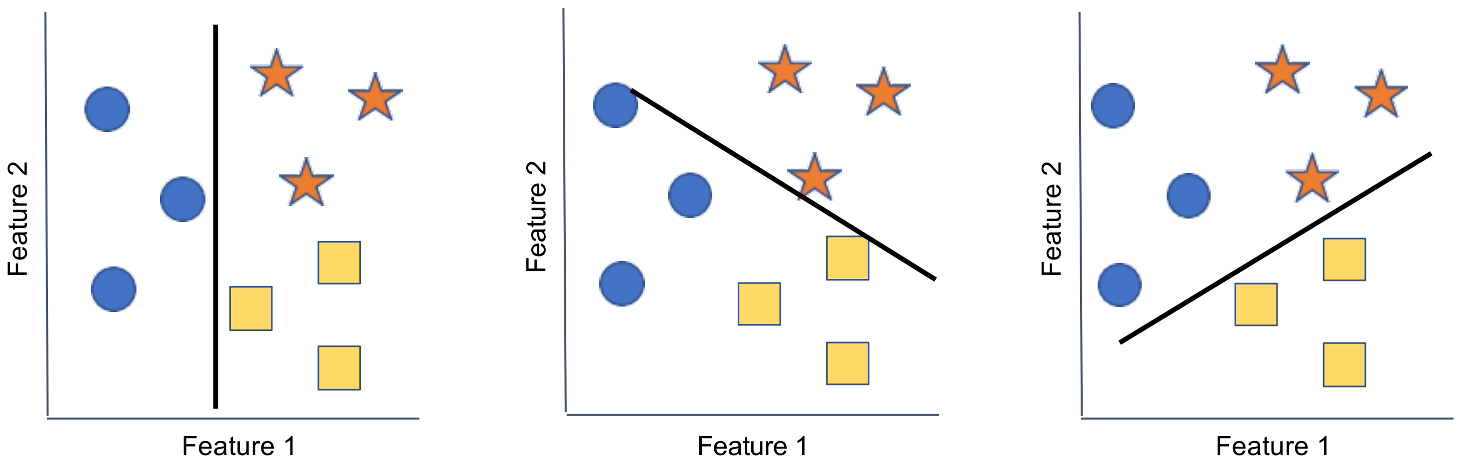


- How can we construct a multi-class classifier by combining several binary classifiers:
  - one-versus-rest (OVR) strategy
  - one-versus-one (OVO) strategy

Multi-Class Classification: OVR and OVO

- one-versus-rest: learn k different binary classifiers:
  - the first separates C_1 from $C_2, \ldots, C_k$
  - the second separates C_2 from $C_1, C_3, \ldots, C_k$, etc.
  - in general, the i-th classifier separates C_i from all the other classes

Multi-Class Classification: OVR Approach

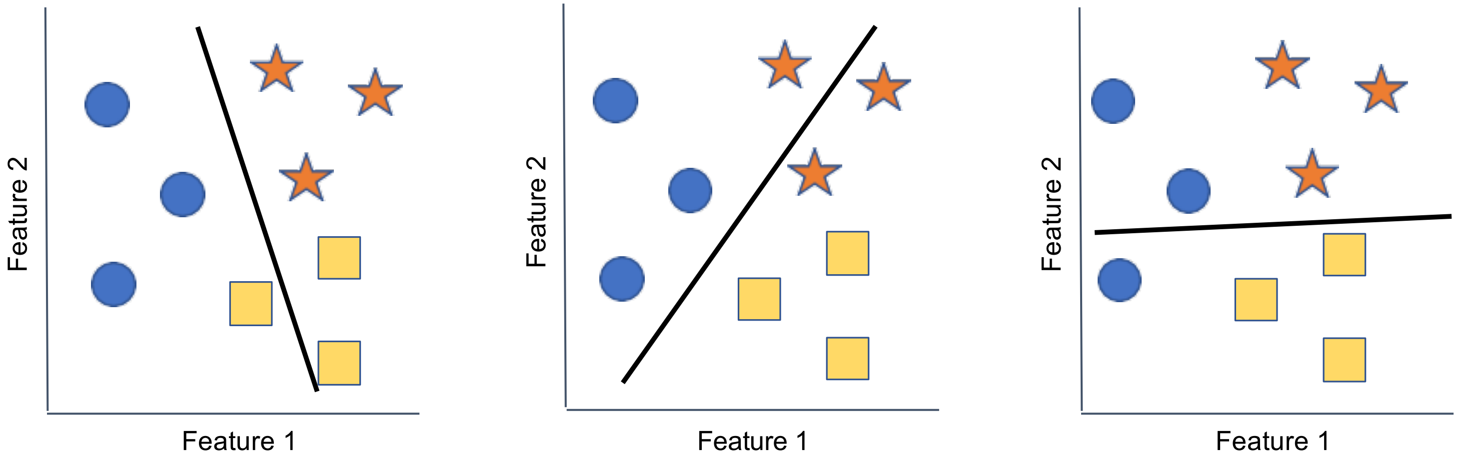


University of Sussex

Multi-Class Classification: OVR and OVO

- one-versus-rest: learn k different binary classifiers:
  - the first separates C_1 from $C_2, \ldots, C_k$
  - the second separates C_2 from $C_1, C_3, \ldots, C_k$, etc.
  - in general, the i-th classifier separates C_i from all of the other classes
- one-versus-one: learn $k(k-1)/2$ binary classifiers:
  - one classifier for each pair of different classes

Multi-Class Classification: OVR Approach



University of Sussex

Multi-Class Classification: OVR and OVO

- The OVR and OVO strategies provide a way of combining predictions of binary classifiers
  - ▶ OVR: relatively efficient as train just k classifiers
  - ▶ OVO: $k(k-1)/2$ classifiers, but less data used to train each
- BUT: still need to make a prediction!
  - ▶ basic idea: use binary predictions as votes for classes
  - ▶ class with most votes wins

Multi-Class Classification: OVO Example

- Suppose we have four classes: C_1, C_2, C_3, C_4
- For OVO there will be $4 \times (4-1)/2 = 6$ classifiers
- Use the classifiers to vote:

	BC_1	BC_2	BC_3	BC_4	BC_5	BC_6	Votes
C_1	+	−	−	0	0	0	1
C_2	−	0	0	−	−	0	0
C_3	0	+	0	+	0	+	3
C_4	0	0	+	0	+	−	2

University of Sussex

Multi-Class Classification: OVO Example

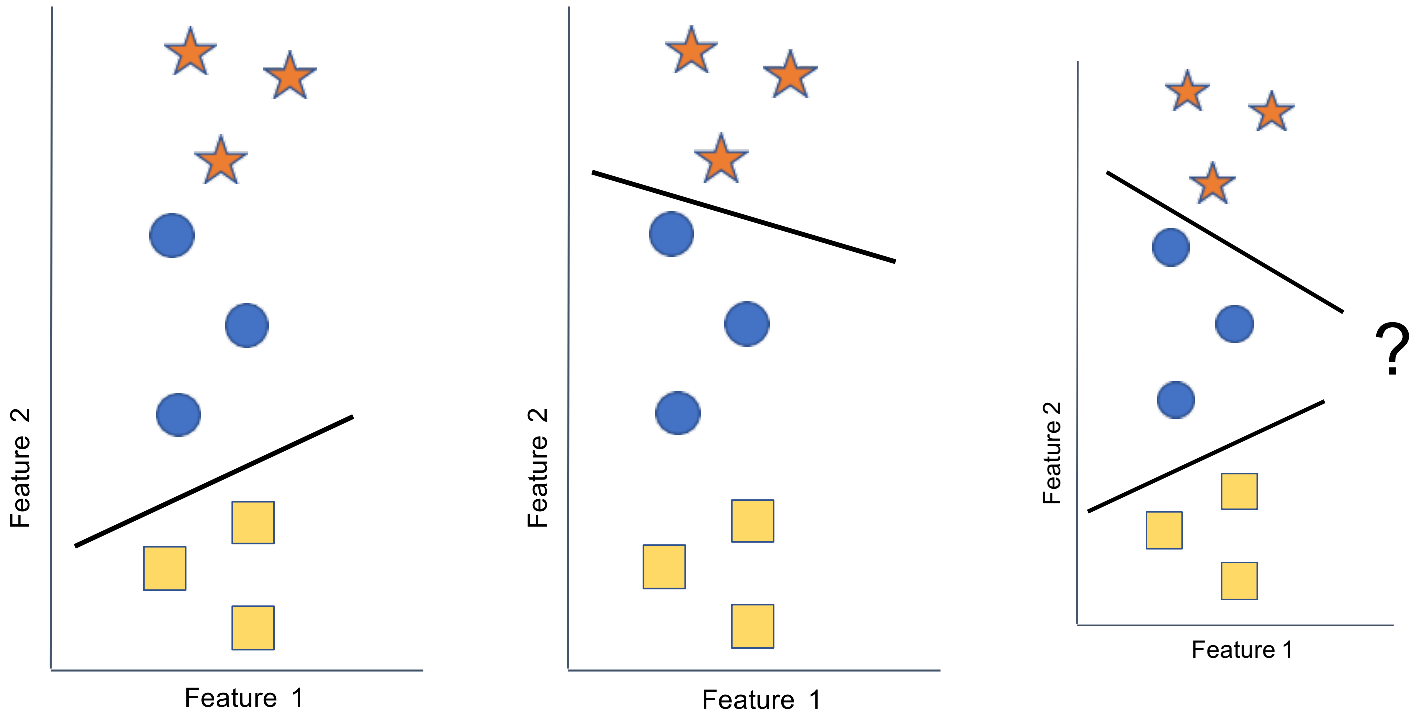
- Of course, things may not be this simple
  - two classes might get the same number of votes...

	BC_1	BC_2	BC_3	BC_4	BC_5	BC_6	Votes
C_1	+	+	–	0	0	0	2
C_2	–	0	0	–	–	0	0
C_3	0	–	0	+	0	+	2
C_4	0	0	+	0	+	–	2

  - If the classifier outputs scores or probabilities, could we take that into account?
    - loss-based decoding

Multi-Class Classification: OVR Difficult Example

OVR can also have limitations depending on the feature space:



US
University of Sussex

Multi-Class Classification with Binary Classifiers

- The choice is problem specific!
- For small numbers of classes, OVO may work better.
- For larger numbers of classes, OVR may be the only practical solution.
- Scikit-learn supports both! 😅

Labs

- Explore the basics of ML and classification in a quiz plus colaboratory worksheet.
- Introduction to scikit-learn.

What Will be on Next Week?

Recap on probability theory and probability density estimation

I'll see you at the labs next week, and lectures again in Week 5