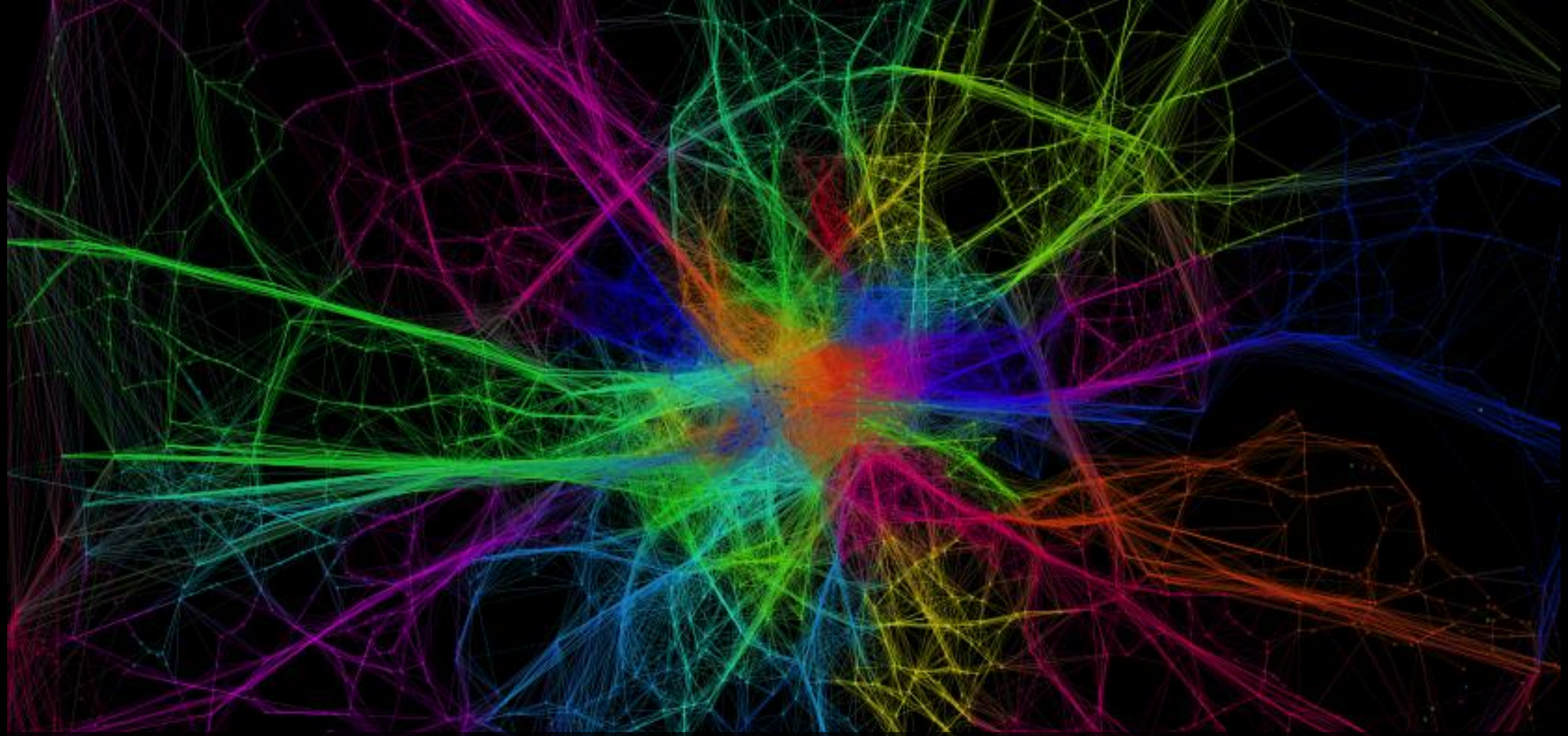


Supervised machine learning

CASA0006: Data Science for Spatial Systems

Huanfa Chen

CASA0006

1. Introduction to Module
2. Supervised Machine Learning
3. Tree-based Methods
4. Artificial Neural Networks
5. Analysis Workflow
6. Panel Regression
7. Difference in Difference
8. Regression Discontinuity
9. Dimensionality Reduction
10. Spatial Clustering

Objectives

- Learn the basics of supervised learning
- Develop an intuition of supervised learning
- Understand the analysis workflow of supervised learning
- Be able to choose the metrics for regression and classification

Outline

1. Supervised machine learning (SML)
2. The analysis workflow of SML
3. Cross validation
4. Metrics
5. Randomness in model training

Supervised learning

Supervised learning

- We gave the algorithm a data set where a "right answer" was provided, build a model to predict the output (y) given the input variables (x)

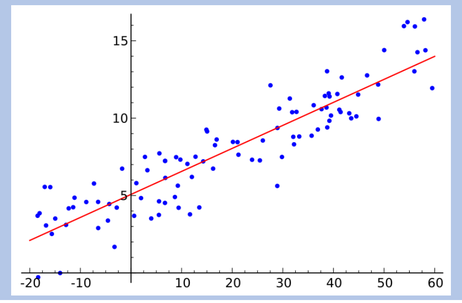
What is the type of Y?

Continuous (integer/float)

Discrete (categorical)

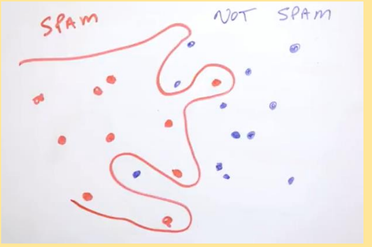
Regression

- Predicting house price
- Methods include linear regression, tree-based models, neural nets, etc.



Classification

- Email -> spam or non-spam
- Methods include logistic regression, tree-based models, neural nets, etc.
- Intuition: *find a boundary between classes*



Many methods can be used for both tasks

Challenge of supervised learning: overfitting

- ML is aimed at making accurate predictions for new unseen data.
- The model overfits the data, when the model is fitting the training data too well (accuracy of 100%) but does not generalise to new unseen data
- Overfitting is unavoidable: all machine learning methods tend to overfit, especially the complicated models.
- How to recognise overfitting? Use train-test split
- How to mitigate overfitting? Tune the hyperparameters carefully
- Workflow is crucial!

Example of spam recognition (Is this a spam?)

My name is Aisha Al-Qaddafi the only biological Daughter of late
Libyan President (Late Colonel Muammar Al-Qaddafi). i am a Widow with three Children, I'm currently living in Muscat Oman under political asylum protection by the Oman Government. Considering my present condition, i have chosen to contact you after my prayers and I believe you will not betray my trust, but rather take me as your own sister or daughter.

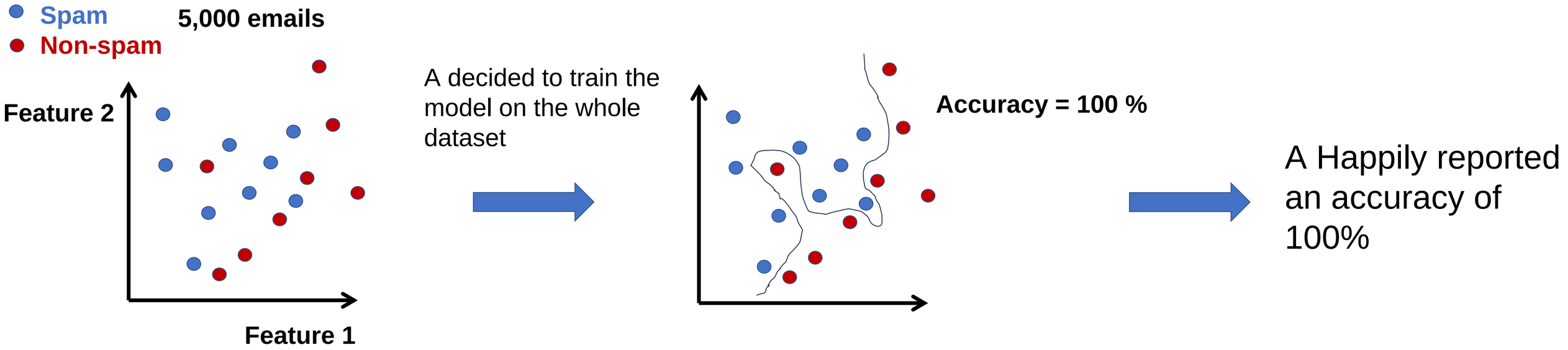
I have investment funds worth $27,500,000.00 USD "Twenty Seven Million Five Hundred Thousand United State Dollars" which i want to entrust on you for investment project in your country because of my current refugee status here in Oman. If you are willing to handle this project on my behalf, kindly reply urgently to enable me provide you more information about the investment funds.

…… My husband late Dr. Joe Bronson who died in the year 2020, was a minor at Kruger gold company and trades on gold. **When he was alive he deposited the sum of $4,800,000 million in a bank in Manama the capital city of Bahrain in Southern Asia**. The money I am willing to give you was income from long years of hard work of my late husband…… I will be going in for an operation surgery soon and I need your urgent answer to know if you will be able to execute this project, and I will give you more information on how the fund will be transferred to your bank account

Example of spam recognition (Is this a spam?)

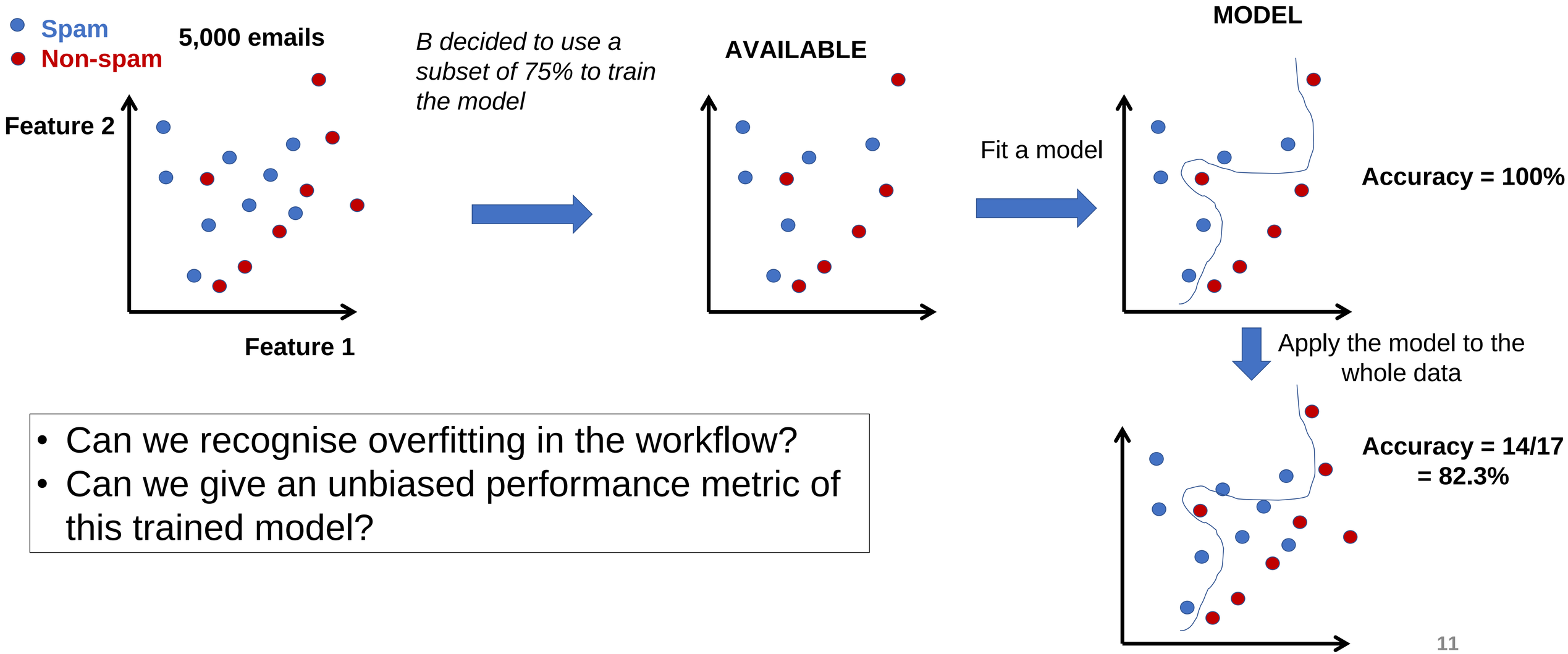
- A data scientist team is asked to develop a spam recognition application using a powerful ML method and 5,000 emails (labelled as spam/non-spam, with two features). The hope is that this application will be used daily in the company.
- Three team members (A, B, C) proposed three different workflow
- You are the team leader, so please think about the following questions
  - Can we recognise overfitting in the workflow?
  - Can we give an unbiased performance metric of this trained model (to the unseen data)?

Workflow of A



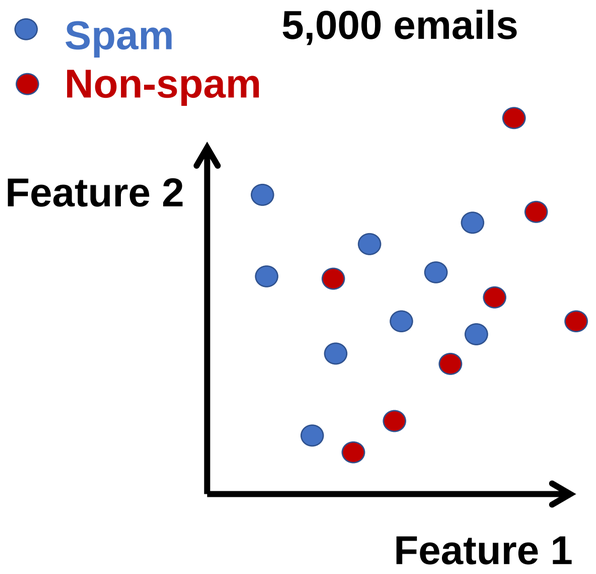
- Can we recognise overfitting in the workflow?
- Can we give an unbiased performance metric of this trained model?

Workflow of B



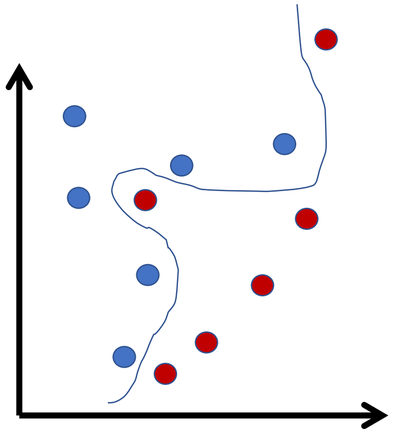
- Can we recognise overfitting in the workflow?
- Can we give an unbiased performance metric of this trained model?

Workflow of C



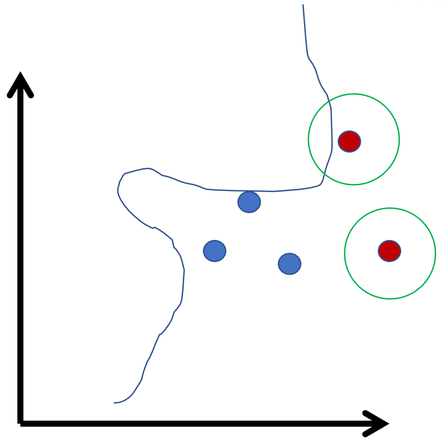
C randomly split the data into 75% and 25%, used 75% subset to train the model

Fit a model



Accuracy = 100%

Apply the model to the rest 25%



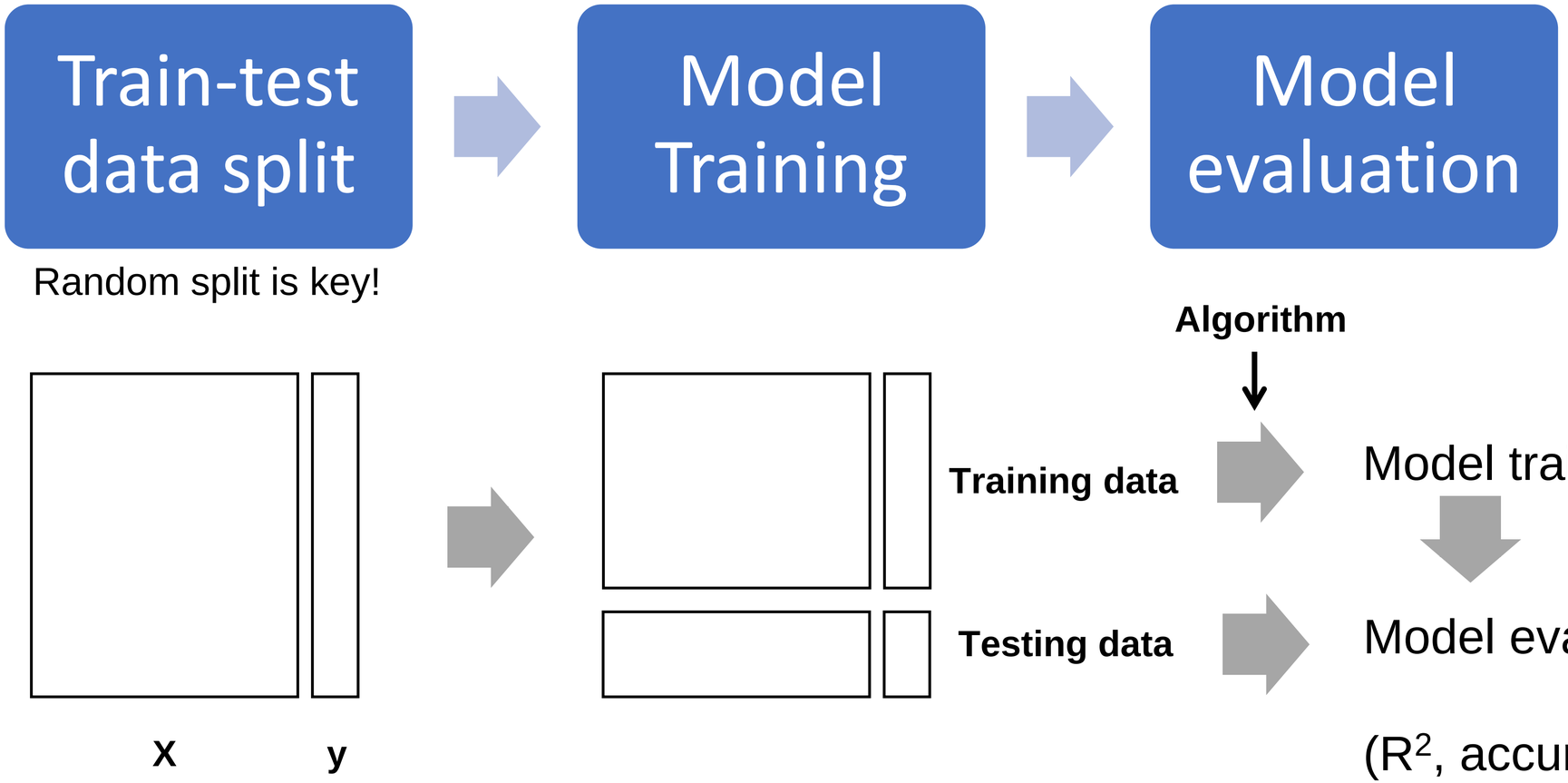
Accuracy = 40%

- Can we recognise overfitting in the workflow?
- Can we give an unbiased performance metric of this trained model?

Summary – which workflow is the best?

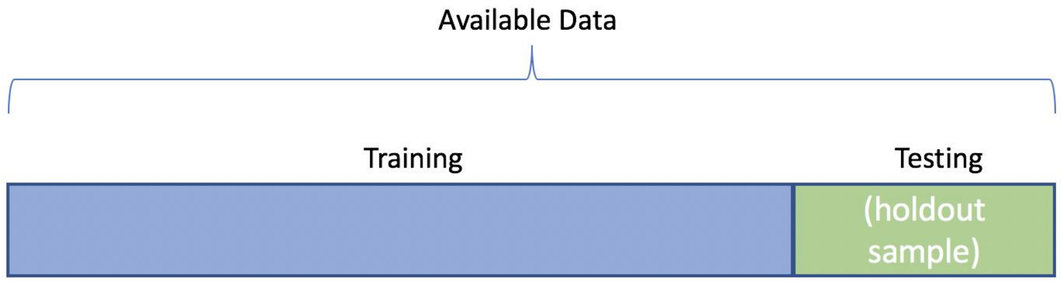
Analysis workflow of supervised learning

Analysis workflow (v1; train-test split)



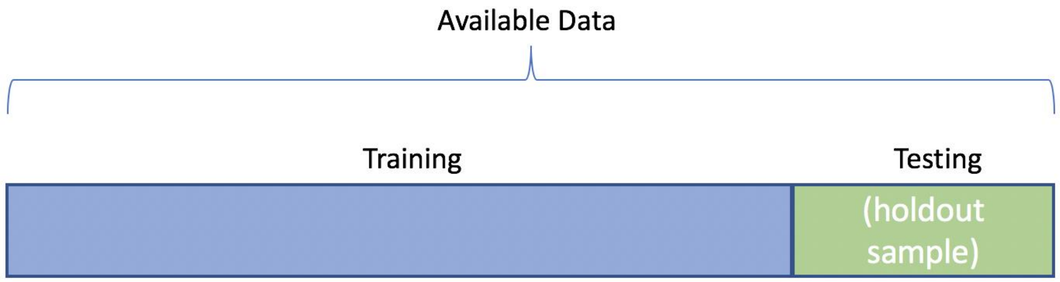
Train-test data split

- Usually, 75% for training and 25% for testing
- A random split: the data is randomly shuffled before split. This is to avoid selection bias and guarantee that the training and testing data are independently and identically distributed (i.i.d)
- But, the random split introduces randomness in model training and evaluation – will discuss randomness later.
- This technique is called hold-out validation in statistics.



Train-test data split

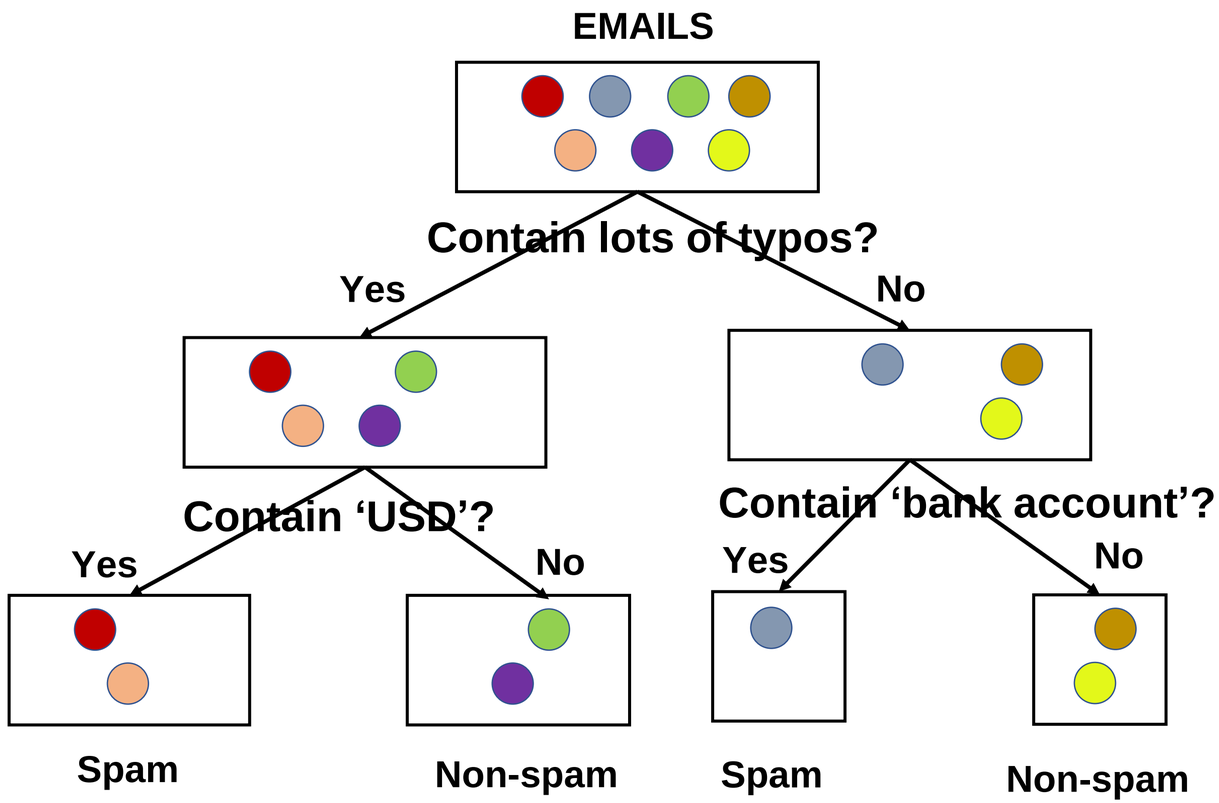
- The testing data should be not used in any stages of model training or fitting.
- If the testing data is used in model training, this is called data leakage.
- Sometimes, data leakage is subtle and not easy to identify. Will give more examples later.



A decision tree for spam detection

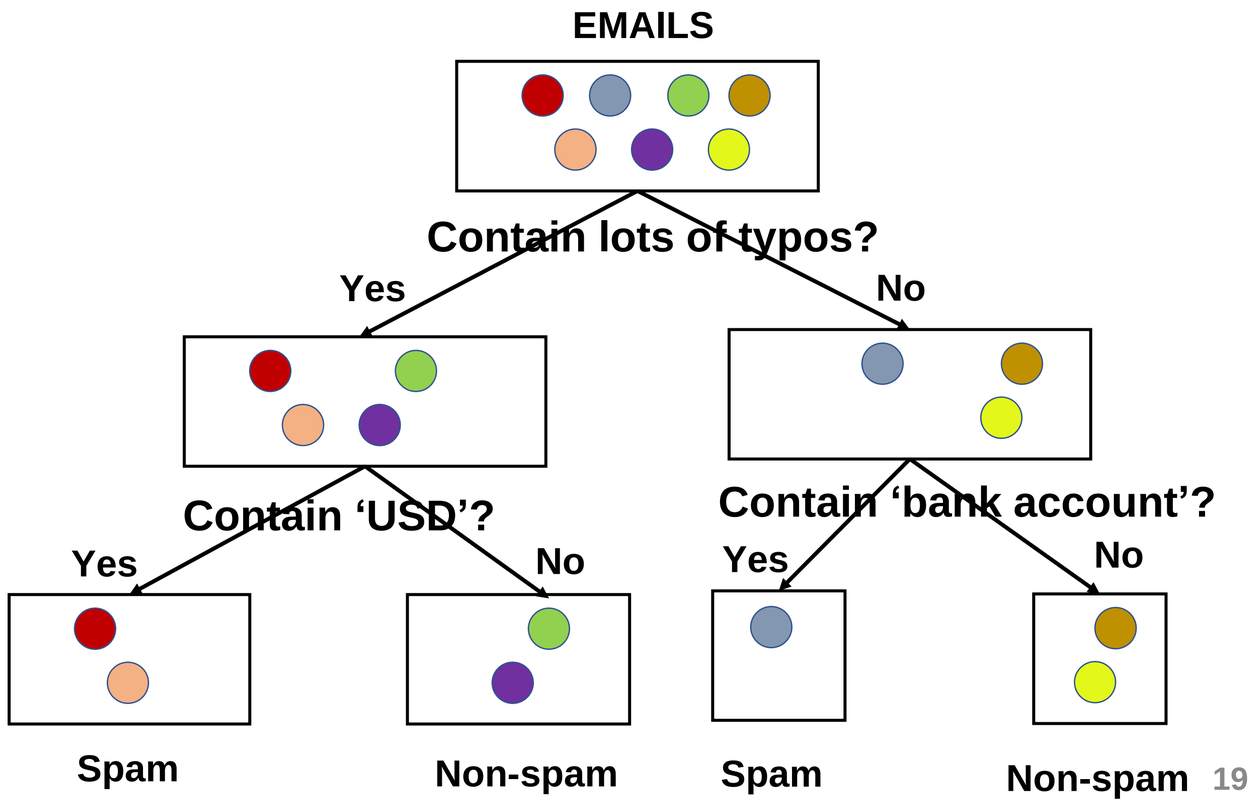
- Decision tree = a flow diagram or a ‘tree’ of decisions about the x variables of a dataset

- Algorithm: the idea of decision tree
- Model training: to find the optimal ‘criteria’ to split spam and non-spam
- Prediction: given a new sample, is it spam?



Is this workflow perfect?

- Problem solved
  - Giving an unbiased evaluation of model performance
  - Recognising overfitting (overfitting occurs, if training accuracy >> testing accuracy)
- Problem unsolved
  - Mitigating overfitting?
  - Hyperparameter tuning?

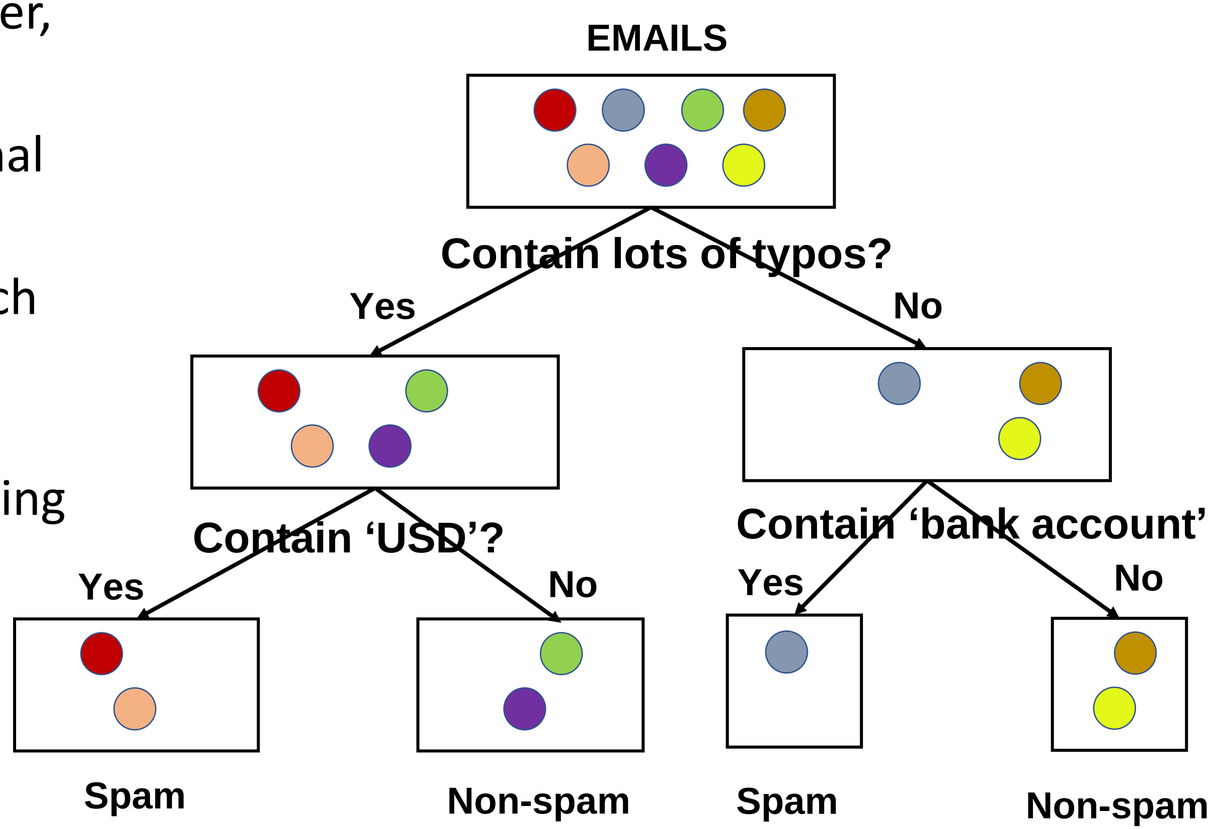


Hyperparameter tuning

- **Hyperparameters**: the settings predefined by the user, e.g. max height of the tree (range: 5,10,15)
- **Hyperparameter tuning**: How can we find the optimal hyperparameters for the model?
- A common approach is to predefine the range of each hyperparameter, and empirically find the optimal combination of these values.
- For linear or logistic regression, hyperparameter tuning is not needed (as there are no hyperparameters)

Questions

- Can we use the training data for this purpose? (We can't)
- Can we use the testing data for this purpose? (We can't)

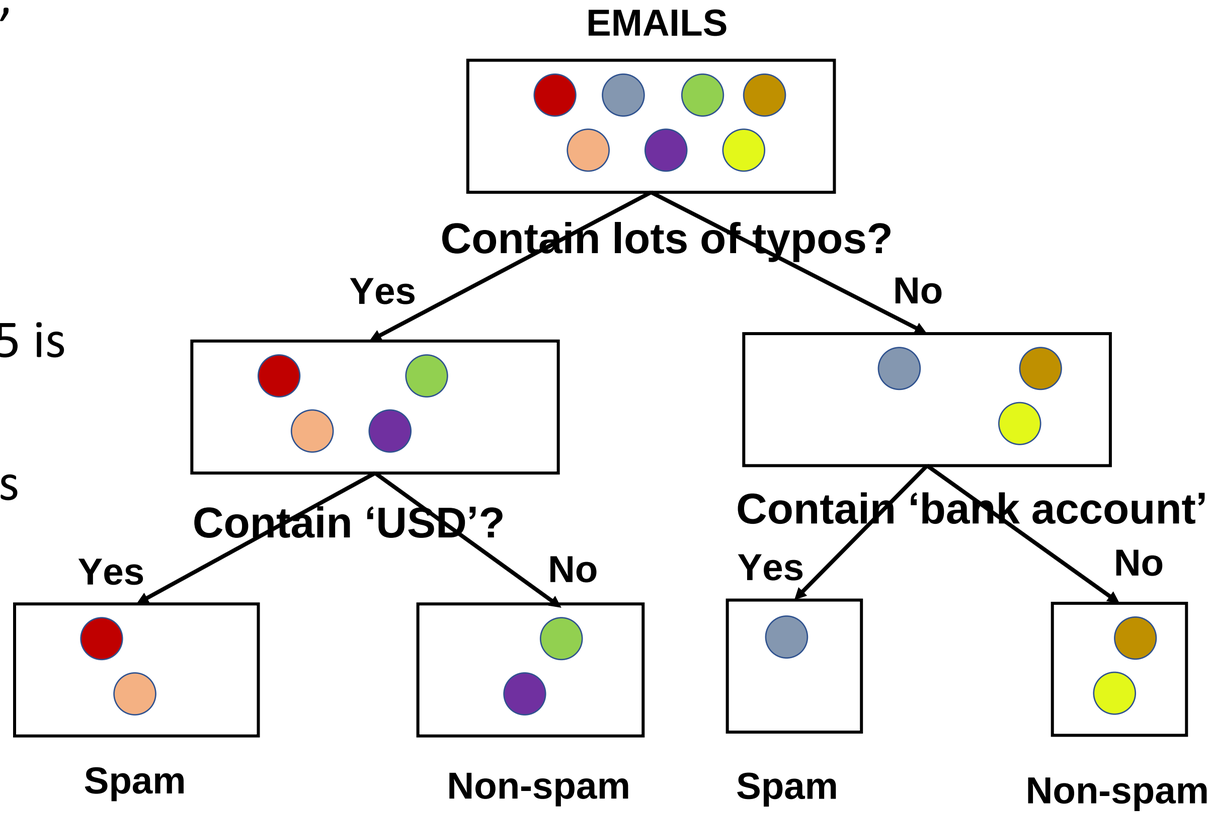


Hyperparameter tuning

Example: to find the optimal tree heights (range: 5, 10, 15), we split the data into training and testing sets and trained three models with height=5,10,15. Results as below.

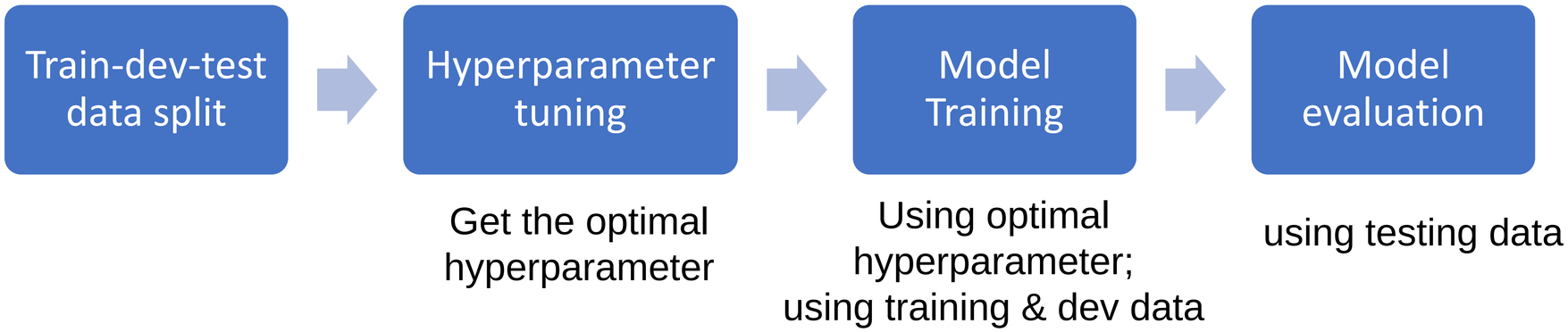
Questions

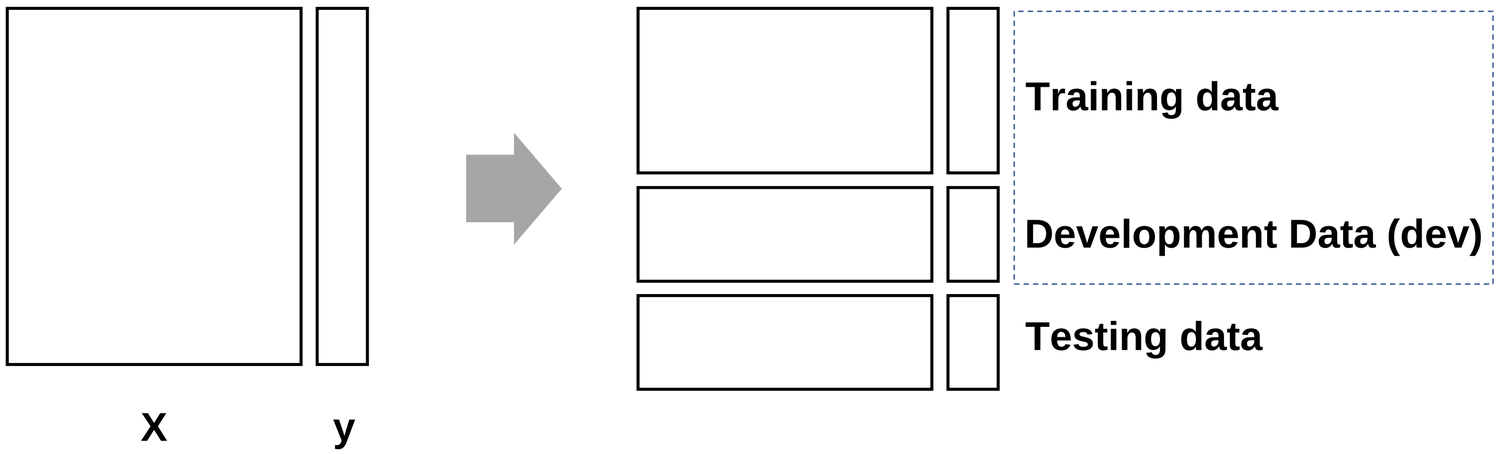
- Can we use the training data and conclude height=15 is the best? (We can't)
- Can we use the testing data and conclude height=5 is the best? (We can't)



Tree heights	5	10	15
Accuracy on training data	90%	92%	95%
Accuracy on testing data	80%	75%	66%

Analysis workflow (v2)

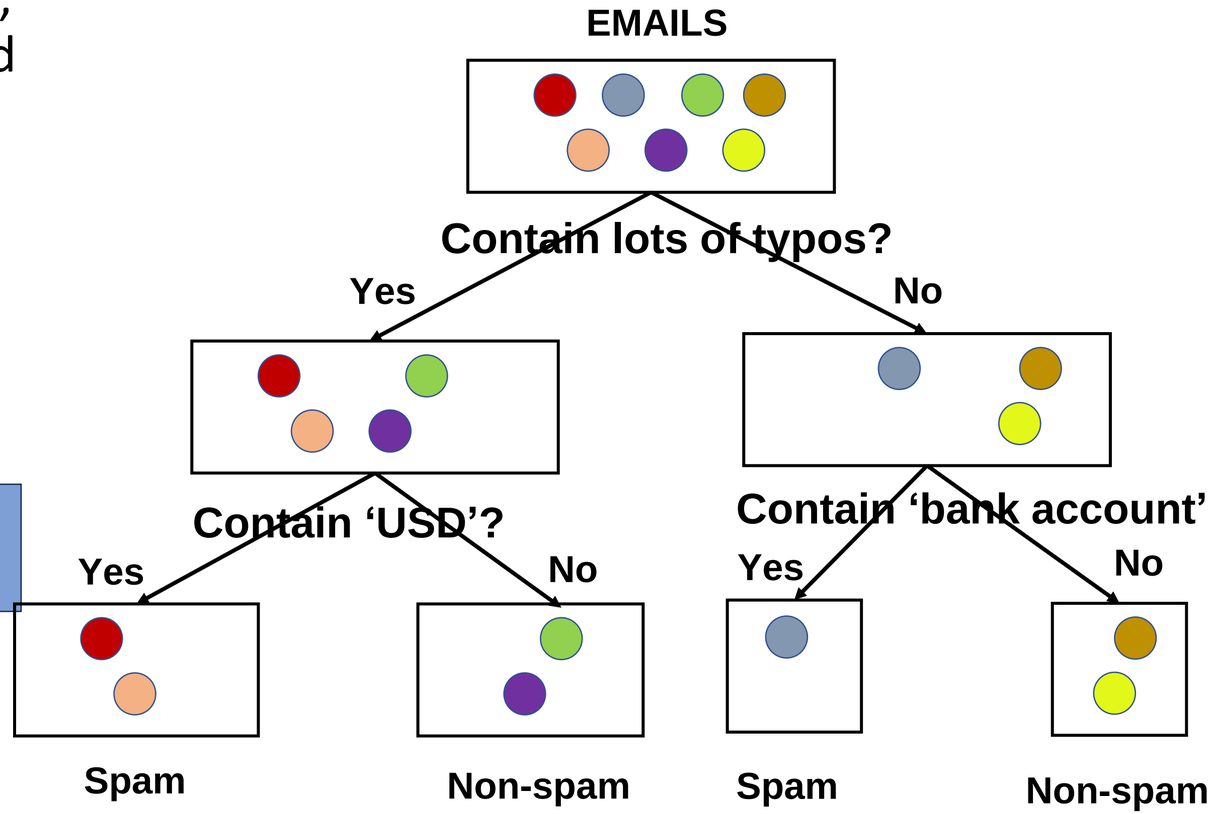




Hyperparameter tuning

Example: to find the optimal tree heights (range: 5, 10, 15), we split the data into training/dev/testing sets and trained three model (using only training data) with height=5,10,15. Results as below.

Model	1	2	3
Tree heights	5	10	15
Accuracy on training data	90%	92%	95%
Accuracy on dev data	83%	87%	70%
Accuracy on testing data	80%	75%	66%



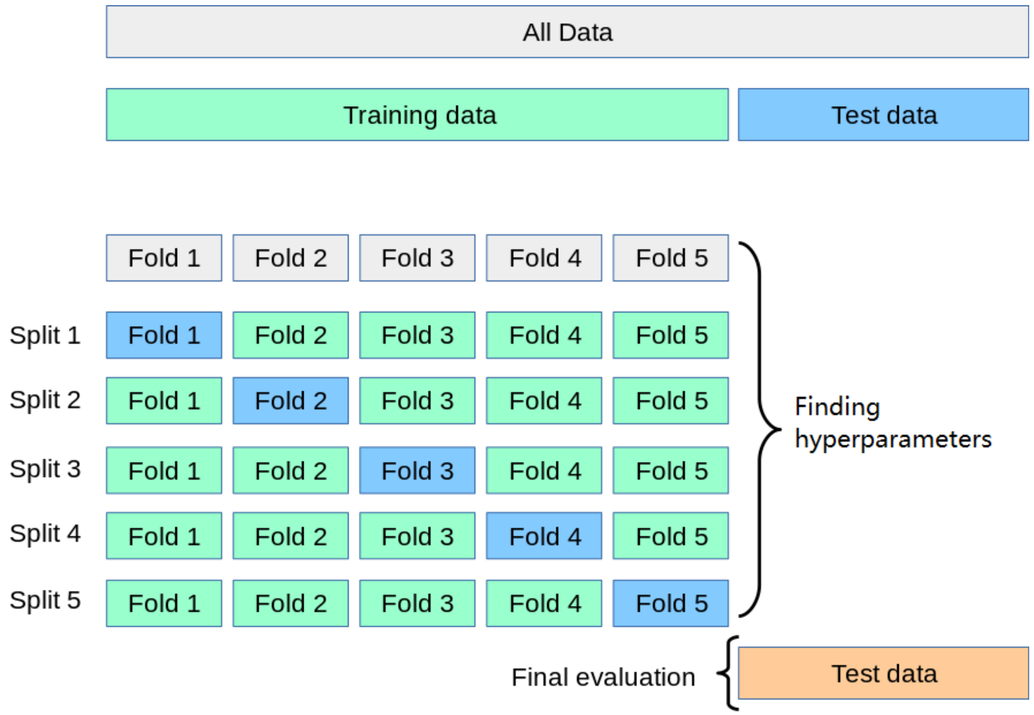
Tree height = 10 is the optimal. So, the model #2 is adopted and the testing accuracy of model #2 is reported (75%)

Cross validation

Problem of workflow v2

Problem with Train-dev-test split: only limited data is used for model training, which is a problem when data are sparse; subject to the randomness of split between train and dev data

Solution - Cross validation: more sufficient use of data for model training



(Amended from scikit-learn.org)

Example: to find the optimal tree heights (TH, range: 5, 10, 15) using 5-fold CV.

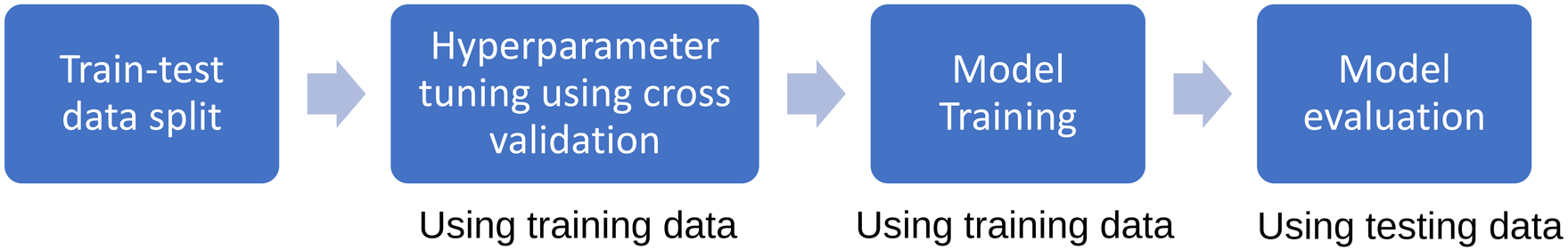
- To get performance of TH=5: 5 models are trained.

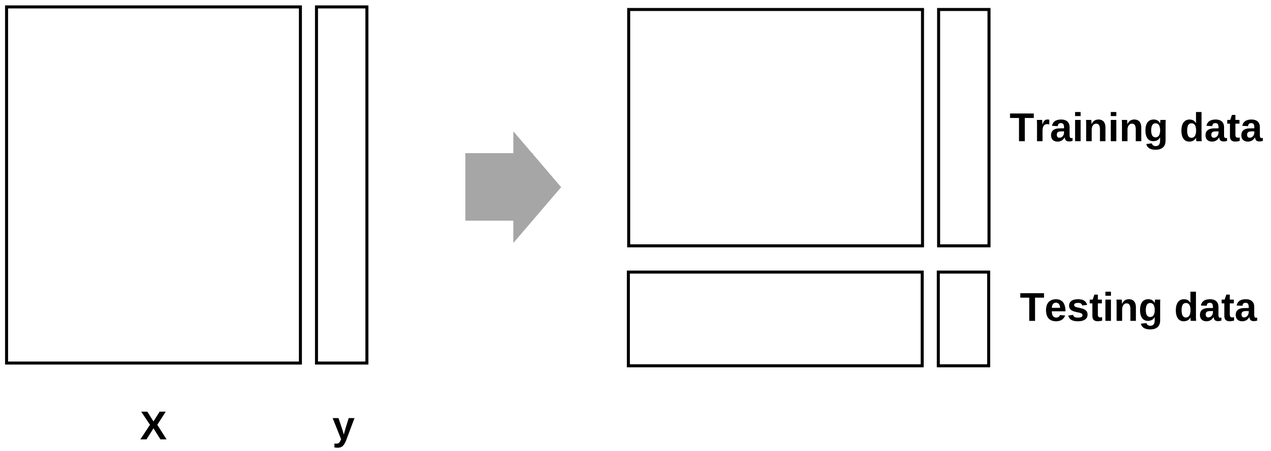
Model	1	2	3	4	5
Training	Fold 2/3/4/5	Fold 1/3/4/5	Fold 1/2/4/5	Fold 1/2/3/5	Fold 1/2/3/4
Evaluation	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Performance of TH=5: average of these five models

Pick up TH with best performance, then train a model using this TH on the whole training data

Analysis workflow (v3)



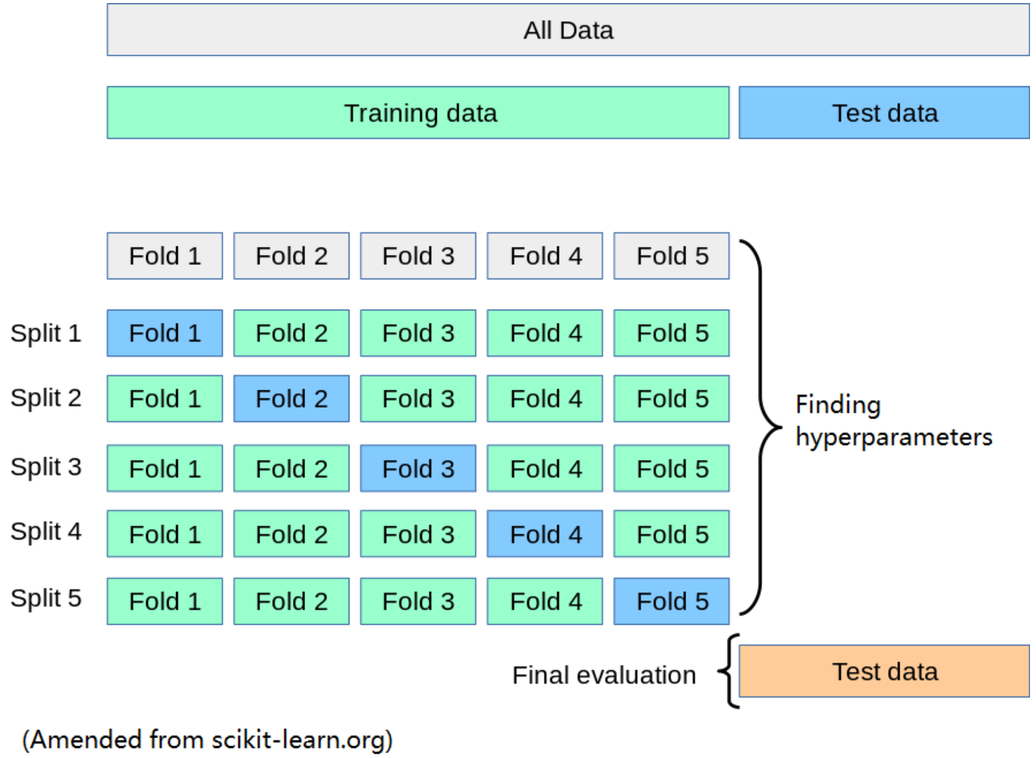


Cross validation

Cross validation is recommended for most ML tasks.

- Pros: sufficient use of training data; less subject to random split (using average performance)
- Cons: time-consuming

Common: 5-fold or 10-fold CV.



(Amended from scikit-learn.org)

Metrics

Metrics for regression

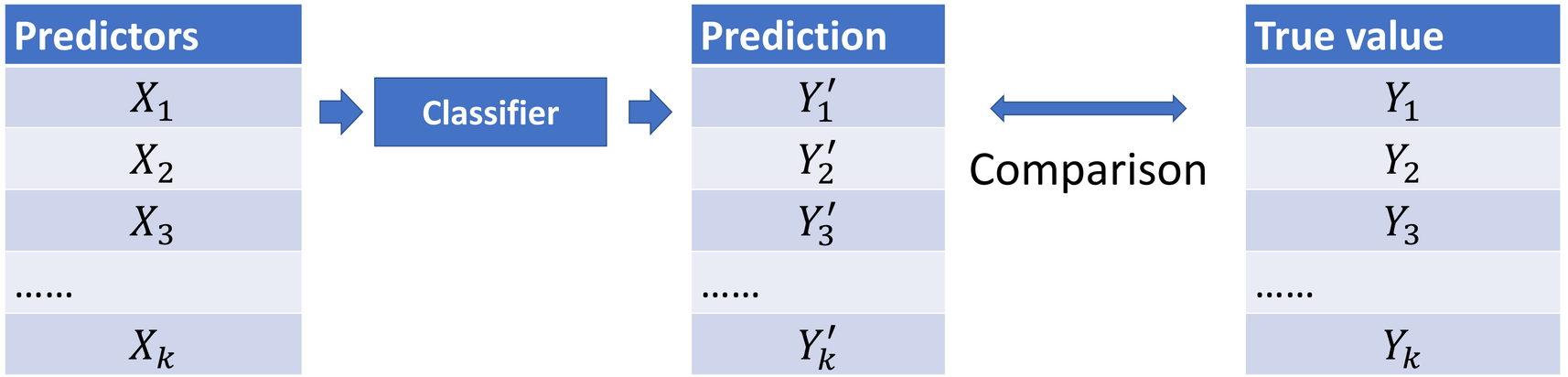
	Definition	Range	Trend
R Squared	$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$	(-Inf, 1]	The higher R^2, the higher accuracy
RMSE	$\frac{\sqrt{\sum(y_i - \hat{y}_i)^2}}{n}$	[0, Inf)	The smaller RMSE, the higher accuracy

1. By definition, R^2 is not the square of a value.
2. Two possibilities of a negative R^2: the prediction algorithm is not suitable or not well tuned; the input variables are not related to y variable.

3. A special case of R^2 is that linear regression has a R^2 in the range of [0,1].

Metrics for classification

Example: given travel survey data, predict travel modes as one of four modes



How accurate are the predictions?

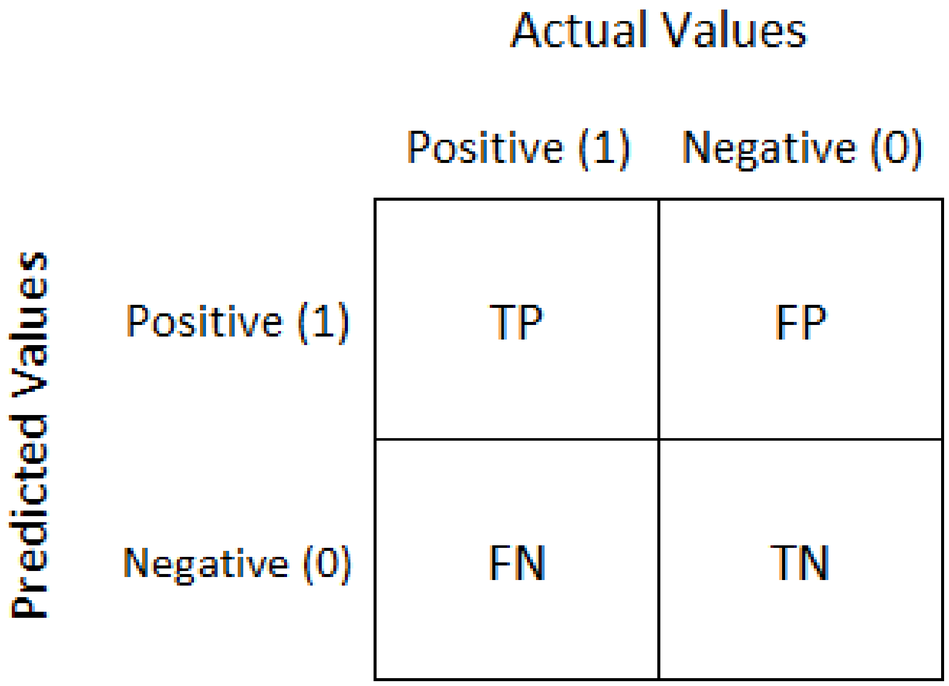
- For a single record: the prediction is TRUE or FALSE
- For many records: confusion matrix

https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Performance metrics

Confusion Matrix: comparing predicted against observed classes

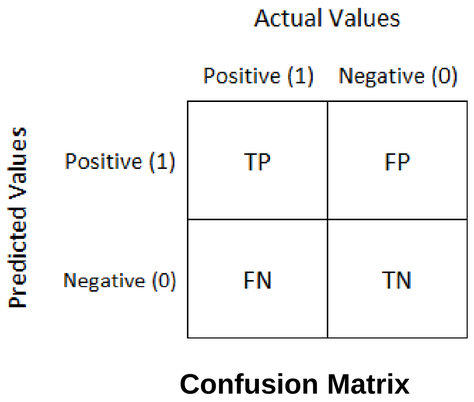
Two-class (e.g. Driving vs. Not-driving)



https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Performance metrics

Comparing predicted against observed classes



Confusion Matrix

Classification Accuracy

Proportion correctly classified

$$\frac{Correct}{Correct + Incorrect} = \frac{\#tp + \#tn}{\#tp + \#tn + \#fp + \#fn}$$

Range of [0,1]

Precision

How many positive predictions are correctly classified?

$$\frac{\#tp}{\#tp + \#fp}$$

Range of [0,1]

Recall

How many positive classes are correctly classified?

$$\frac{\#tp}{\#tp + \#fn}$$

Range of [0,1]

F1

A balance between precision and recall, takes beta attribute which weights precision or recall (usually beta = 1)

$$(1 + \beta^2)\frac{precision * recall}{\beta^2 precision + recall}$$

Range of [0,1]

Performance metrics

Q1: What is the conflict between precision and recall?

A: In many cases, improving one of them would lead to degrading of the other.

S1: predict most as 'negative' – maximise prec

Predicted

Actual		Positive	Negative
	Positive	1	99
	Negative	0	100

Precision = 1
Recall = 0.01

S2: predict most as 'positive' – maximise recall

Predicted

Actual		Positive	Negative
	Positive	100	0
	Negative	99	1

Precision = 0.5
Recall = 1

Performance metrics

Q2: why is F1 score a tradeoff between prec and recall?

A: F1 score combines these two metrics into a single metric; It has a value between min(prec, recall) and max(prec, recall).

S1: predict most as 'negative' – maximise prec

Predicted

Actual		Positive	Negative
	Positive	1	99
	Negative	0	100

Precision = 1
Recall = 0.01
F1 = 0.02

S2: predict most as 'positive' – maximise recall

Predicted

Actual		Positive	Negative
	Positive	100	0
	Negative	99	1

Precision = 0.5
Recall = 1
F1 = 0.67

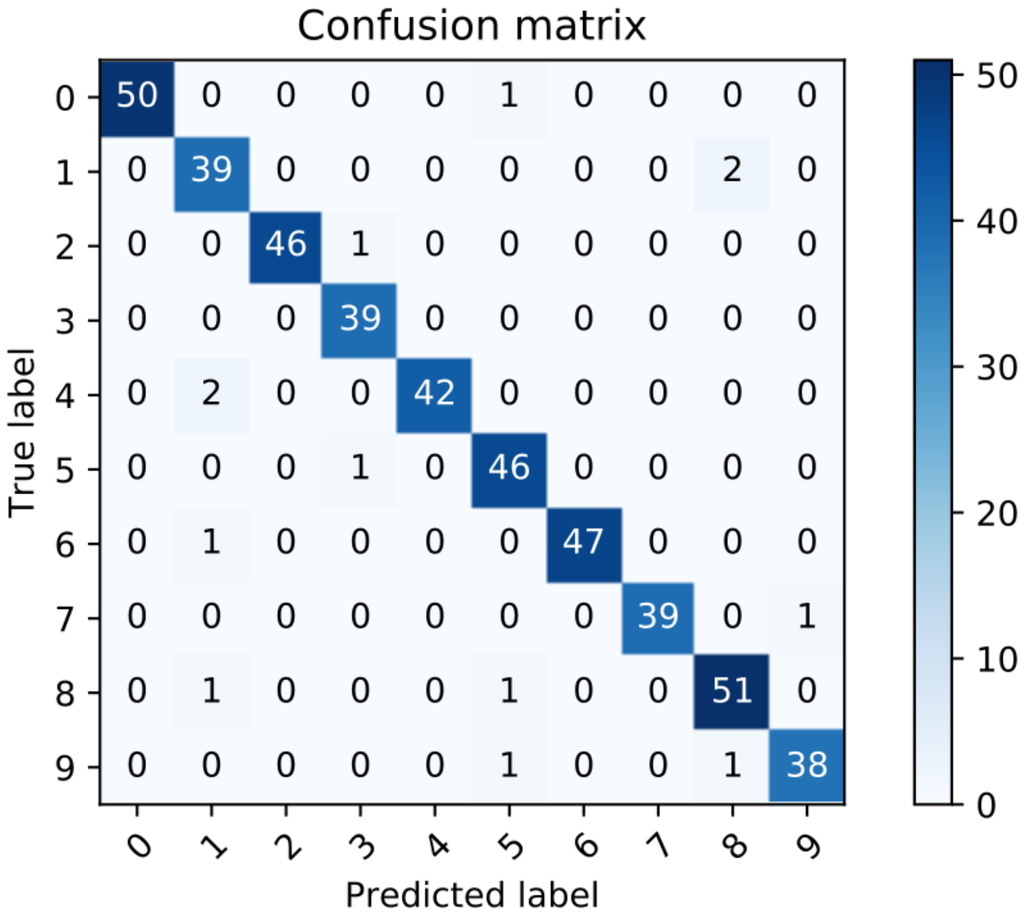
Performance metrics

Q3: What is the problem of accuracy (or why are other metrics needed?)

Accuracy paradox: Accuracy might be not useful when the class distribution is highly imbalanced. For example, when predicting mode with actual 99% driving and 1% cycling, a ‘trivial’ predictor that simply predicts ‘driving’ will have very high accuracy, but this predictor is not useful.

Suggestion: when presenting classification results, you could present both accuracy and F1 score.

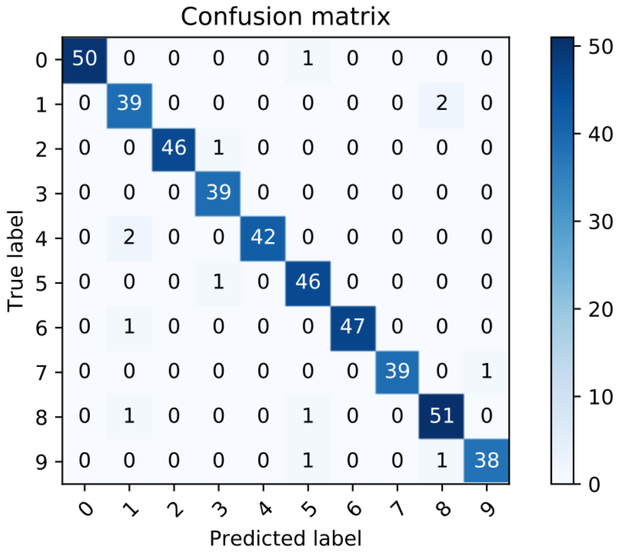
Performance metrics



- Numbers on the diagonal line are the TP for each class
- n_{ij} is the number of instances with actual class i that are classified as class j.
- When reading a confusion matrix, it is important to know meaning of rows and columns (rows as true or predicted label?)

Performance metrics

Multiple-class problem (K classes)
→ K confusion matrices



Classification Accuracy $\frac{\sum_i n_{ii}}{\sum_i \sum_j n_{ij}}$

Precision* average$(Prec_1, Prec_2, \ldots, Prec_K)$

Recall* average$(Rec_1, Rec_2, \ldots, Rec_K)$

F1* average$(F1_1, F1_2, \ldots, F1_K)$

This is called macro average of precision/recall/F1.

Other ways of calculating these scores include ‘micro’ and ‘weighted’. See here.

Randomness in machine learning

Randomness

There are multiple sources of randomness in model training and testing.

- Random data split
- Random seeds used by models (almost all modes using random seeds)
- Model optimisation (may be subject to local optimum)

Ways to mitigate randomness

- Get sufficient data (>50 data samples, preferably > 100)
- Use robust methods
- Multiple runs using different random seeds and report average and standard deviation of performance

Summary

- Basics of supervised learning: regression, classification, model generalisation
- The analysis workflow of supervised learning: train-test split, cross validation
- Metrics for regression: R^2, RMSE
- Metrics for classification: accuracy, F1 score
- Randomness in machine learning

Workshop

- Weekly quiz on Moodle: please finish them before the workshop and we will discuss the quiz in the workshop
- Python notebooks for workshop: will be ready by 5pm Thursday.
- See you in the workshop on Friday 1-3pm