CASA0006: Data Science for Spatial Systems (20/21)

Analysis Workflow

Huanfa Chen

Some slides courtesy of Kira Kempinska

1. Introduction to Databases
2. Introduction to SQL
3. Advanced SQL
4. Data Munging
5. Advanced Clustering
6. Advanced Regression
7. Data Classification
8. Dimensionality Reduction
9. Handling Unstructured Data
10. Analysis Workflow

Recap

What we already know

We can handle data

Using a database accessed through SQL, and tools such as Pandas we can take raw data through to something useful

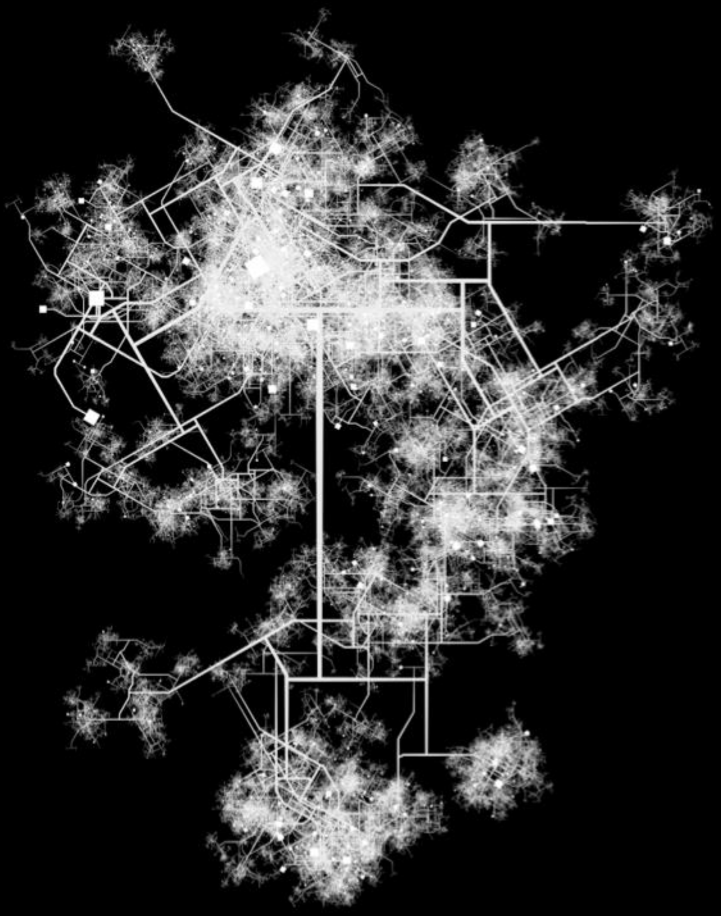
We can analyse data

Clustering, Regression, Classification, Dimensionality Reduction

We can handle unstructured data

But we lack engineering techniques for diagnosing and improving a model

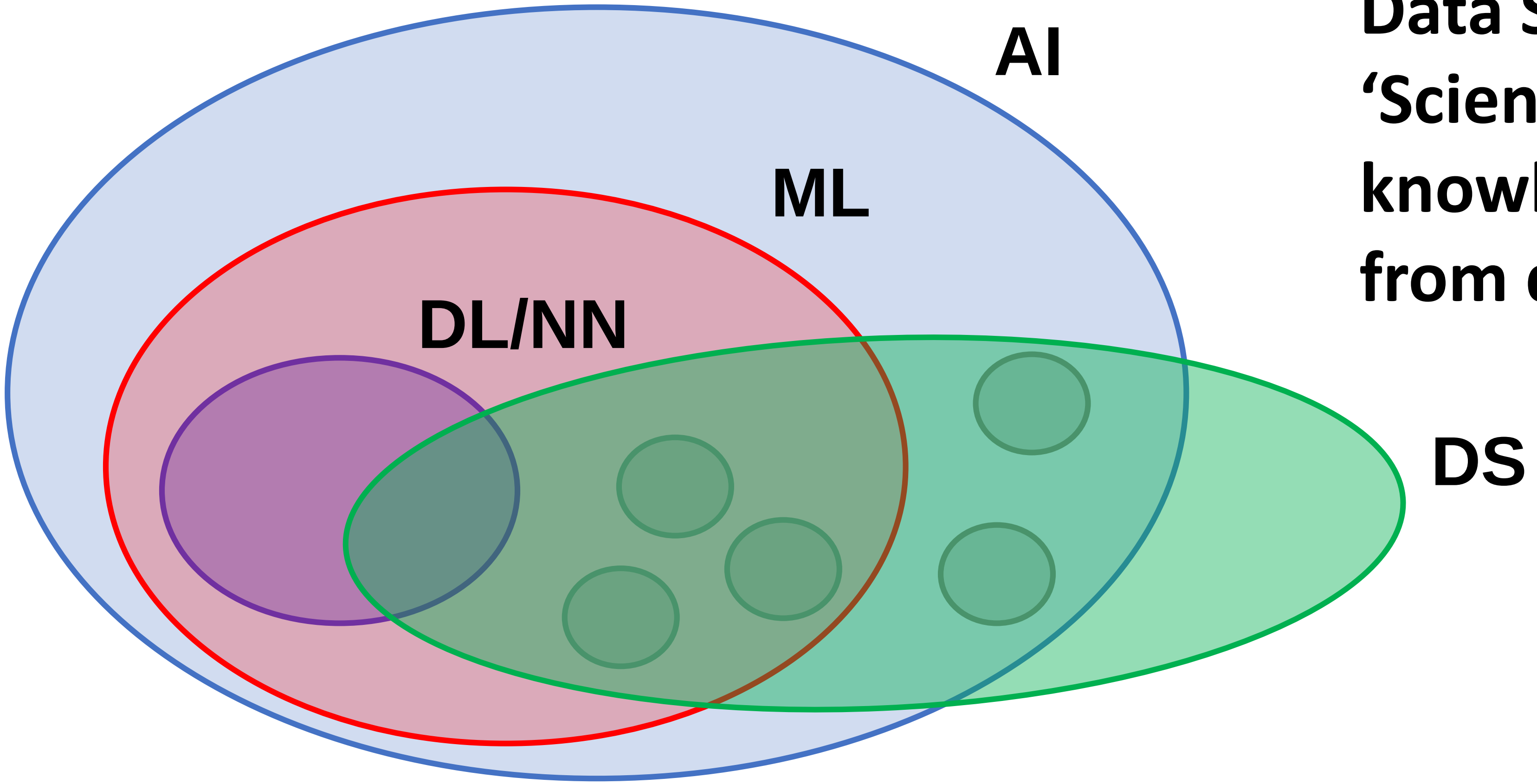
Outline



1. Machine learning recap
2. Setting up development and test sets
3. Basic error analysis
4. Machine learning workflow

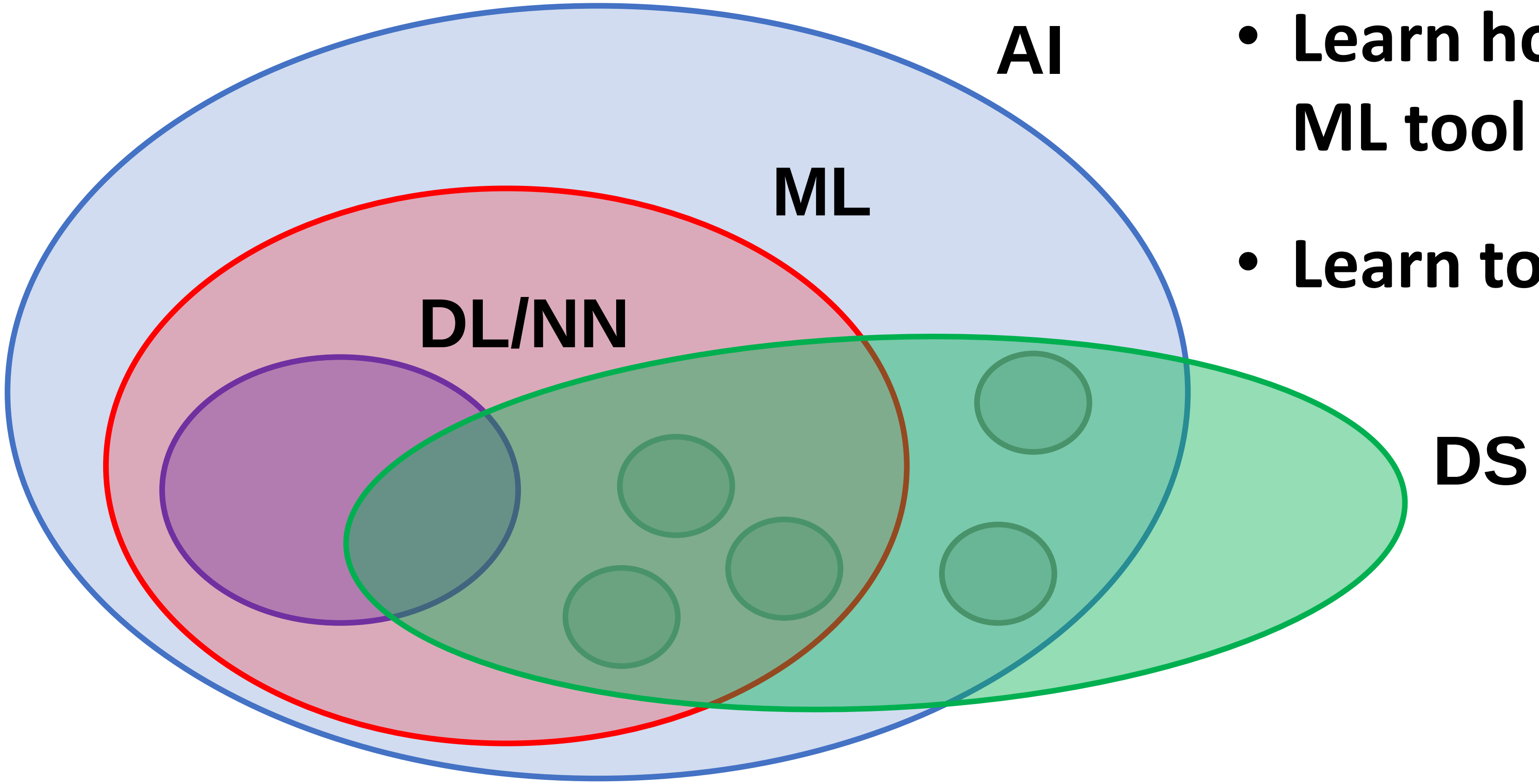
Machine learning recap

Recap: What is Data Science?



Data Science:
'Science of extracting knowledge and insights from data'

Recap: What is Data Science?



- **Learn how to pick the right ML tool for the job.**
- **Learn to use ML effectively.**

Types of machine learning

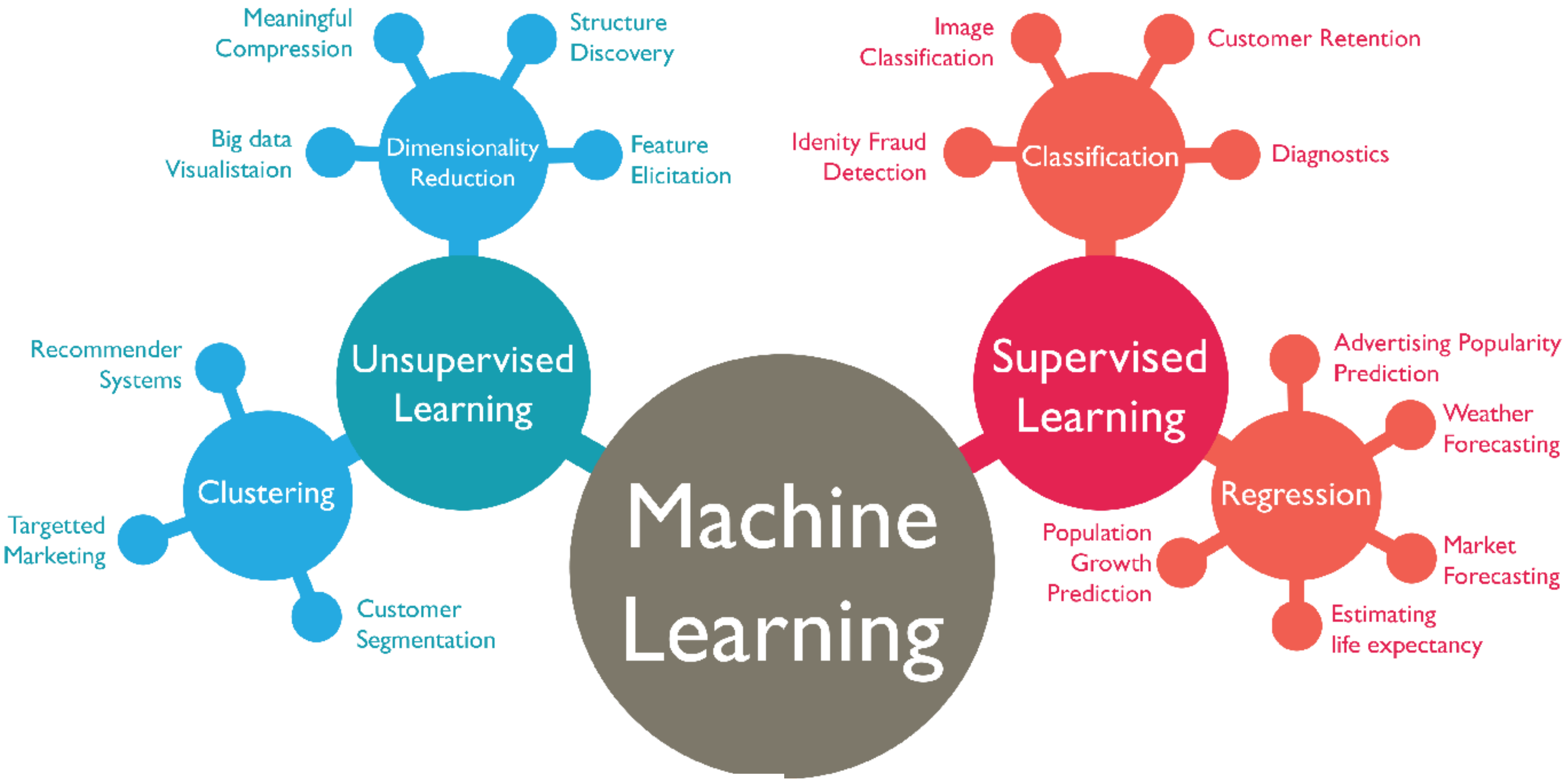


Image Source: https://www.slideshare.net/awahid/big-data-and-machine-learning-for-businesses

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	regression

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	regression
image, radar info	→	position of other cars	

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	regression
image, radar info	→	position of other cars	regression

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	regression
image, radar info	→	position of other cars	regression
ad, user info	→	click? (0/1)	

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	regression
image, radar info	→	position of other cars	regression
ad, user info	→	click? (0/1)	classification

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	regression
image, radar info	→	position of other cars	regression
ad, user info	→	click? (0/1)	classification
age, gender	→	Covid-19 risk	

Input (A)		Output (B)	ML Type
email	→	spam? (0/1)	classification
image	→	cat or dog?	classification
house properties	→	house price	regression
image, radar info	→	position of other cars	regression
ad, user info	→	click? (0/1)	classification
age, gender	→	Covid-19 risk	regression

Input (A)	Output (B)	ML Type
customer profiles →	customer segmentation	

Input (A)	Output (B)	ML Type
customer profiles →	customer segmentation	clustering

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	clustering

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	clustering
patient health	→	2D visualisation	

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	clustering
patient health	→	2D visualisation	dim. reduction

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	clustering
patient health	→	2D visualisation	dim. reduction
company sales	→	company benchmarking	

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	clustering
patient health	→	2D visualisation	dim. reduction
company sales	→	company benchmarking	clustering

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	clustering
patient health	→	2D visualisation	dim. reduction
company sales	→	company benchmarking	clustering
email content	→	email features for spam classification	

Input (A)		Output (B)	ML Type
customer profiles	→	customer segmentation	clustering
pizza orders	→	delivery zones	clustering
patient health	→	2D visualisation	dim. reduction
company sales	→	company benchmarking	clustering
email content	→	email features for spam classification	dim. reduction

Choosing the right tool

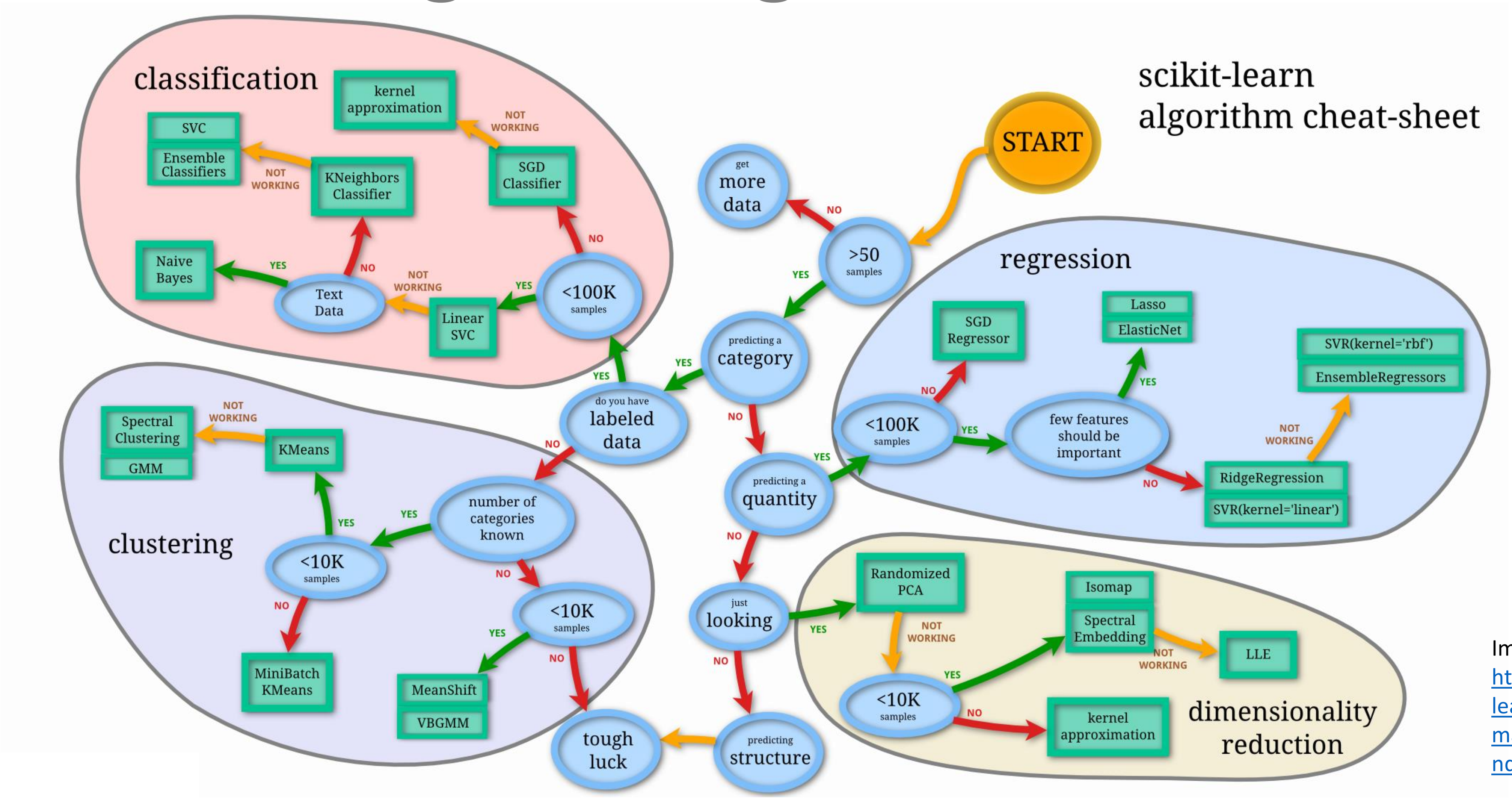


Image source: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Tools for supervised learning

1. Simple models (linear regression)
    - Mainly used for explaining relationship between data
2. Machine learning models (e.g. random forest, XGBoost)
    - Well-suited for relatively small data
3. Deep learning models (e.g. neural nets)
    - Well-suited for high-dim data and large dataset
    - Well-suited for unstructured data


Machine learning strategy

Why machine learning strategy?

Example: Building a cat picture startup

You use a **neural network** for detecting cats in pctures.
But tragically, your algorithm's accuracy is not yet good enough. What do you do?

- Get more data?
- Collect more diverse training set?
- Train the algorithm longer?
- Try smaller/bigger neural network? …

Why machine learning strategy?

Example: Predicting London house prices

You use a **linear regression** model to predict house prices given number of bedrooms and location. Your model's R^2 score is only 0.56. What do you do next?

- Get more data?
- Try a decision tree or deep learning model instead?
- Stop?

- Most ML problems leave clues that tell you what’s useful to try, and what’s a waste of time.
- Learning to read those clues will save you weeks or months of development time.
- **Let’s learn basic workflow to out your ML project in the right direction.**

Setting up development and test sets

Your development and test sets

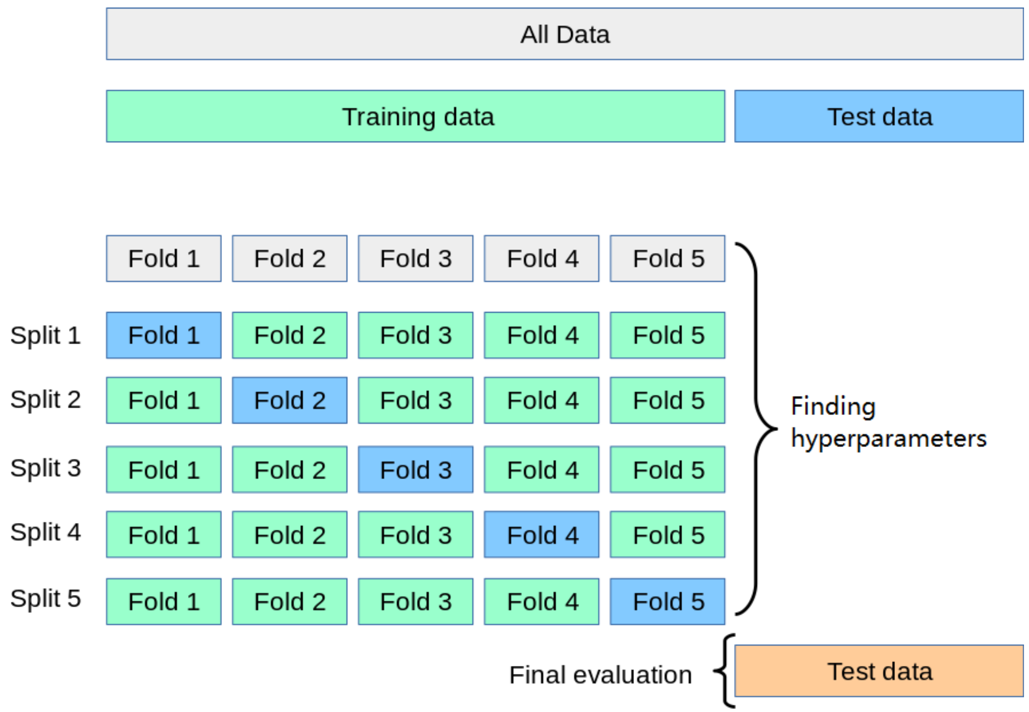
- **Training set** – which you run your algorithm on.

- **Dev (development set or validation set)** - which you use to tune hyperparameters, select features, and make other decisions regarding the learning algorithm. This dataset is your “problem specification”.

- **Test set -** which you use to evaluate the performance of the algorithm, but not to make any decisions regarding what learning algorithm or parameters to use.

Cross validation

- Cross validation is a special case of splitting training and dev set.
- If you use normal training and dev set split, the dev set is static and the validation score is calculated in one run.
- If you use cross validation, the dev set is dynamic and the cross-validation set is the average of multiple runs.



(Amended from scikit-learn.org)

Score = Mean(Split 1, Split 2, etc.)

Dev and test sets (1):linear regression

- **Training set** – use it to train linear regression.
- **Dev set** – not necessary if you only consider linear regression since there are no hyperparameters to tune in linear regression.
- **Test set –** use it to measure the accuracy of your model using the R^2 statistic.

Dev and test sets (2): random forest

- **Training set** – use it to train (fit) random forest.
- **Dev set** – use it to optimise the number of trees and the depth of trees in random forest.
- **Test set –** use it to measure the accuracy of your final model.

Dev and test sets (3): deep learning

- **Training set** – use it to train neural network.

- **Dev set** – use it find the most suitable neural network hyperparameters.

- **Test set –** use it to measure the final accuracy of your model.

Your dev and test sets should come from the same data distribution

Development and test sets should reflect data you expect to get in the future and want to dwell on.

In other words, development and test sets should represent the data distrbution that you want your model to perform well on.

Checking your understanding (1)

Example: a cat picture startup

- Your start-up wants to build a **mobile app** that detects cats in uploaded photos.
- You get a large dataset by downloading pictures of cats (positive) and non-cats (negative) off of different websites.
- You split the dataset 70%/30% into training and test sets.
- Using this data, you build a classifier that works well on the training and test sets.

Checking your understanding (1)

Example: a cat picture startup

- But when you deploy this classifier into the mobile app, you find that the performance is really poor!
- **What happened?**

Checking your understanding (1)

Example: a cat picture startup

- But when you deploy this classifier into the mobile app, you find that the performance is really poor!
- **What happened?**
- Mobile phone images tend to be lower resolution and blurrier than the webiste images that you collected.

Checking your understanding (2)

Example: predicting London house prices

- You collect house price data from different London Borough websites.
- You pick Camden as your development set, Waltham Forest as your test set, and the rest as training data.
- You train a decision tree classifier and pick the optiomal classifier parameters according to the dev set.
- You achieve almost 99% accuracy on the dev set.

Checking your understanding (2)

Example: predicting London house prices

- But when you predict house prices on the test set (Waltham Forest), the predictions are too high!
- **What is the problem?**

Avg house price in March 2020:
Camden: ~881 K
Waltham Forest: ~437 K

dev set (Camden)

test set (Waltham Forest)

https://www.statista.com/statistics/1029250/average-house-prices-in-london-united-kingdom-by-borough/

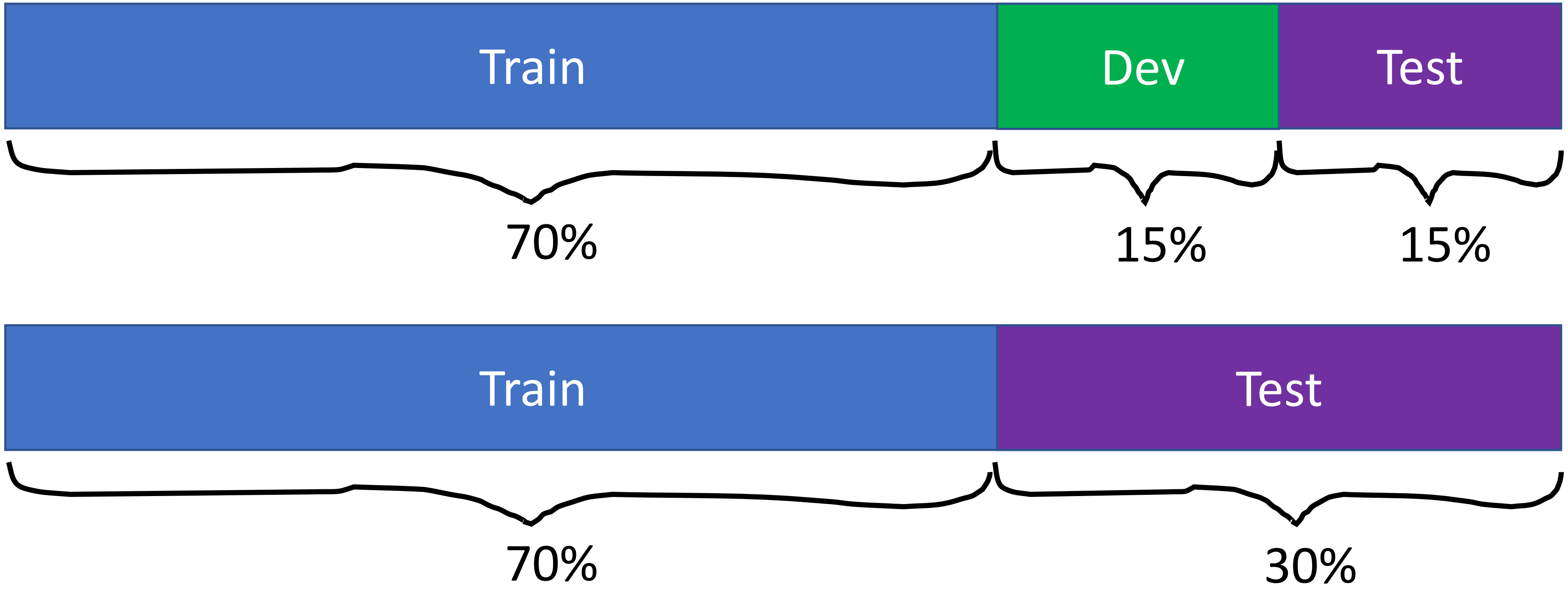
Checking your understanding (2)

Example: predicting London house prices

- But when you predict house prices on the test set (Waltham Forest), the predictions are too high!
- **What is the problem?**
- You’ve tuned your model parameters to Camden, where house prices are much higher than in Waltham Forest.

How large should the data sets be?

If you have between 100 and 10,000 examples:



If you have >10,000 examples, you can reduce % of Dev and Test sets.

Establish a single evaluation metric

Regression: R^2 score

Classification: accuracy, F1 score

Classifier	Precision	Recall	F1 Score
A	95%	90%	**92.4%**
B	98%	85%	**90.1%**

Clustering: Silhouette Coefficient

For a more thorough list of evaluation metrics, please refer to:
https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Basic error analysis

Evaluate performance on both training and dev sets

Use your single evaluation metric (e.g. accuracy) to measure your model performance on both training and dev sets.

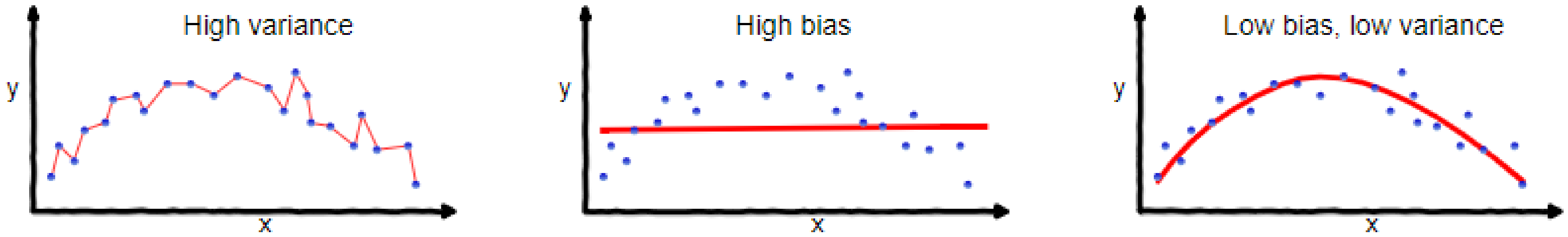
- **Training error 15%** (85% accuracy)
- **Dev error 21%** (79% accuracy)

Bias and Variance – the two big sources of error

It is unlikely that dev error is smaller than training error.

- Training error 15% ==> **Bias** of your algorithm.
- Dev error - training error (21% - 15%) ==> **Variance** of your algorithm.

Task: predicting y using x (one-dimensional example). The line represents the learned relationship



overfitting

underfitting

Good balance

Examples of bias and variance

Consider again our cat classification task.

Training error	1%	15%	15%	0.5%
Dev error	11%	16%	30%	1%

Examples of bias and variance

Consider again our cat classification task.

Training error	1%	15%	15%	0.5%
Dev error	11%	16%	30%	1%

High Variance

Examples of bias and variance

Consider again our cat classification task.

Training error	1%	15%	15%	0.5%
Dev error	11%	16%	30%	1%
	High Variance	High Bias		

Examples of bias and variance

Consider again our cat classification task.

Training error	1%	15%	15%	0.5%
Dev error	11%	16%	30%	1%
	High Variance	**High Bias**	**High Bias** **High Variance**	

Examples of bias and variance

Consider again our cat classification task.

				You're done!
Training error	1%	15%	15%	0.5%
Dev error	11%	16%	30%	1%
	High Variance	High Bias	High Bias High Variance	

Low-high combination of bias and variance

Variance \ Bias	Low	High
Low	Good balance	Underfitting
High	Overfitting	*Improvement needed*

Machine learning workflow

Machine Learning Workflow

Training error high

Dev error high

Machine Learning Workflow

Training error high —**Yes**→ **Bigger model**

Train longer

High Bias

Dev error high

Machine Learning Workflow

Training error high Yes → **Bigger model**

Train longer

High Bias

Dev error high

Machine Learning Workflow

Training error high — **Yes** → **Bigger model**

Train longer

High Bias

No ↓

Dev error high

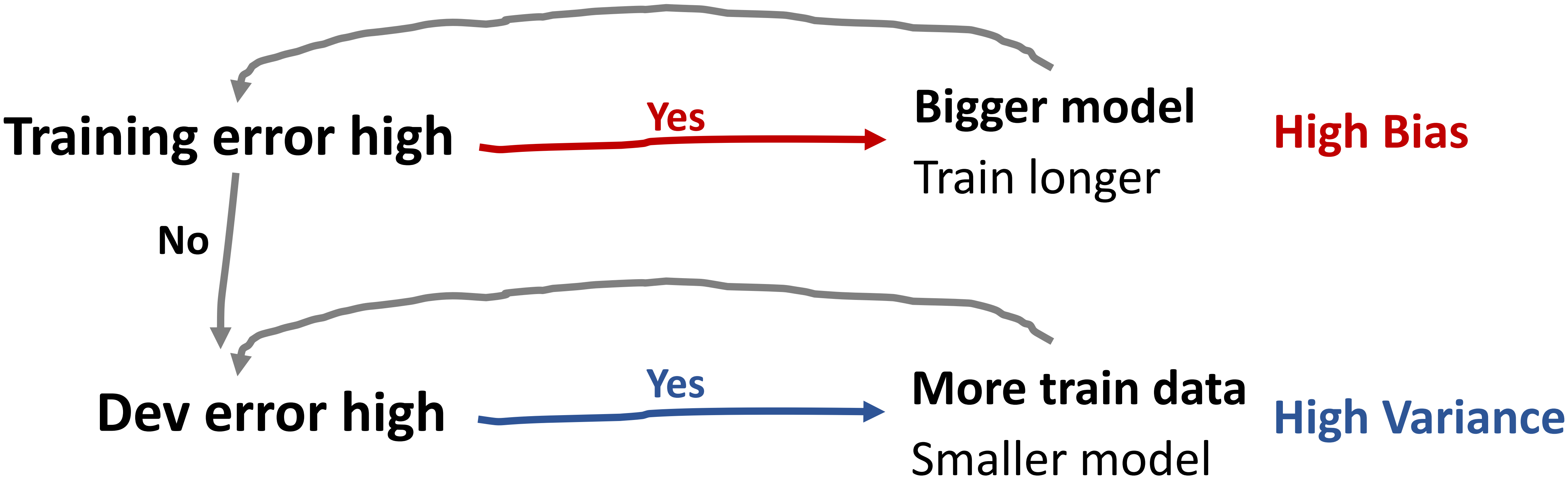
Machine Learning Workflow

Training error high — **Yes** → **Bigger model** / Train longer — **High Bias**

No

Dev error high — **Yes** → **More train data** / Smaller model — **High Variance**

Machine Learning Workflow



Machine Learning Workflow

Training error high — **Yes** → **Bigger model**, Train longer — **High Bias**

No ↓

Dev error high — **Yes** → **More train data**, Smaller model — **High Variance**

No ↓

You're Done!

Techniques for reducing bias

Increase model size

- Replace a simple linear regression model with a more flexible model, such as random forest or deep learning.
- Add more neurons or layers in a deep learning model.

Modify input features

- Inspect your training data to understand which examples your model is not doing well on. See if you can modify data features to eliminate these errors.

Add more training data

- This technique helps with variance problems, but it usually has no significant effect on bias.

Techniques for reducing variance

Add more training data

- This is the simplest and the most reliable way to address variance, so long as you have access to significantly more data.

Reduce model size/complexity

- Replace a large neural network with a random forest.
- Add regularization to your neural network.
- Decrease neural network size.

Feature selection to decrease number/type of input features

- This technique might help with variance problems, but it might also incerase bias.
- With modern deep learning, there has been a shift away from feature selection, and we are now more likely to give all the data to the algorithm and let the algorithm sort out which ones to use.

Benchmarking with baseline model

Let’s consider again our house price prediction problem.

You want to build a regression model that predicts house price given the number of bedrooms and the location.

Which regression model do you pick:

(a) Linear regression
(b) ML/DL regression

Benchmark with baseline model

Let's consider again our house price prediction problem.

You want to build a regression model that predicts house price given the number of bedrooms and the location.

Which regression model do you start with:

(a) Linear regression
(b) ML/DL regression

If in doubt, always start with a simple model. It is quicker to build and test. If it suffers from high bias, then try a more complicated model.

Wrapping Up

Today you've learnt how to:

- select the right ML tool for your problem (regression, clustering, …)
- prepare your train, dev and test sets
- diagnose bias and variance problems with your model
- reduce bias and variance

Well done! You now have all the tricks you need to build successful ML project.

Any Questions?

Huanfa Chen

huanfa.chen@ucl.ac.uk