



The Data Science Process

Jan 6th 2020

Today's Lesson

Learning Objectives

- Describe the phases of the CRISP-DM data science process
- Use the CRISP-DM framework to analyze and plan out data science projects

Activities

- Reflecting on Your Data Science Experience
- Data Science Process Overview
- Revisiting Your Data Science Experience
- Reviewing Your Projects
- Exit Ticket



Your Data Science Experience

Think about a time when you have personally experienced a data science application in real life.

Task: Take turns with a neighbor, answering the questions below:

- What was the experience?
- What about it made you think it was “data science”?
- What, if anything, do you know about the model that was used?
- How did the experience affect you or others?

With your partner, pick one experience to share with the class.



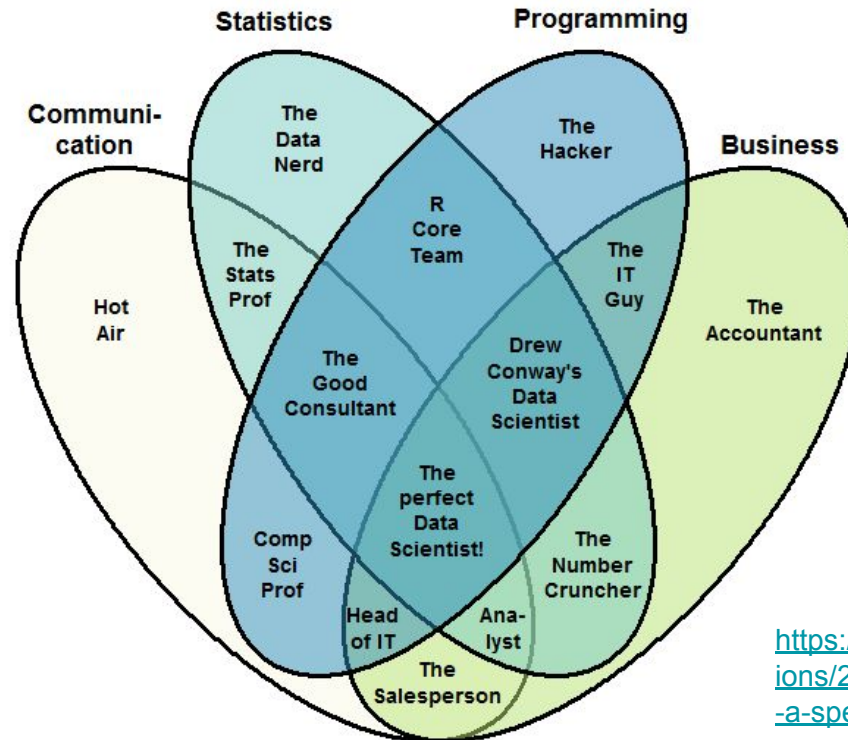
Why bother defining a “data science process”?

- Common language for data scientists and business colleagues
- Project planning
- Evaluate your and others’ work
- Ensure you remember the big picture



Defining “Data Scientist”

The Data Scientist Venn Diagram



<https://datascience.stackexchange.com/questions/2403/data-science-without-knowledge-of-a-specific-topic-is-it-worth-pursuing-as-a-ca>

Data Science Activities

Data Science **OSEMN** Model

Adapted from: KDNuggets

Obtain

- from other location
- Query from database or API
- Extract from another file
- Generate data (e.g. Sensors)



Scrub

- Filtering lines
- Extracting columns or words
- Replacing values
- Handling missing values
- Converting formats



Explore

- Understanding data
- Deriving statistics
- Creating visualization



Model

- Clustering
- Classification
- Regression
- Dimensionality reduction



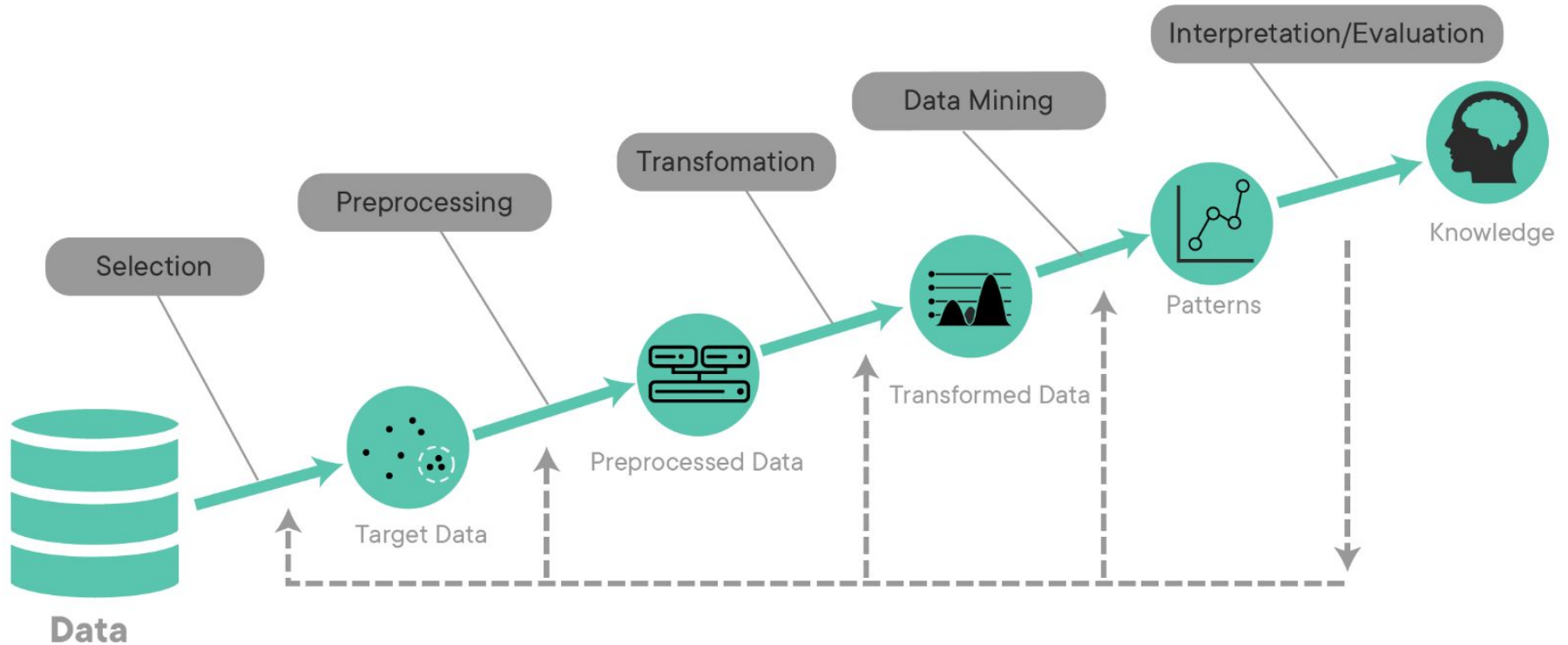
Interpret

- Drawing conclusion from data
- Evaluating meaning of results
- Communicating result



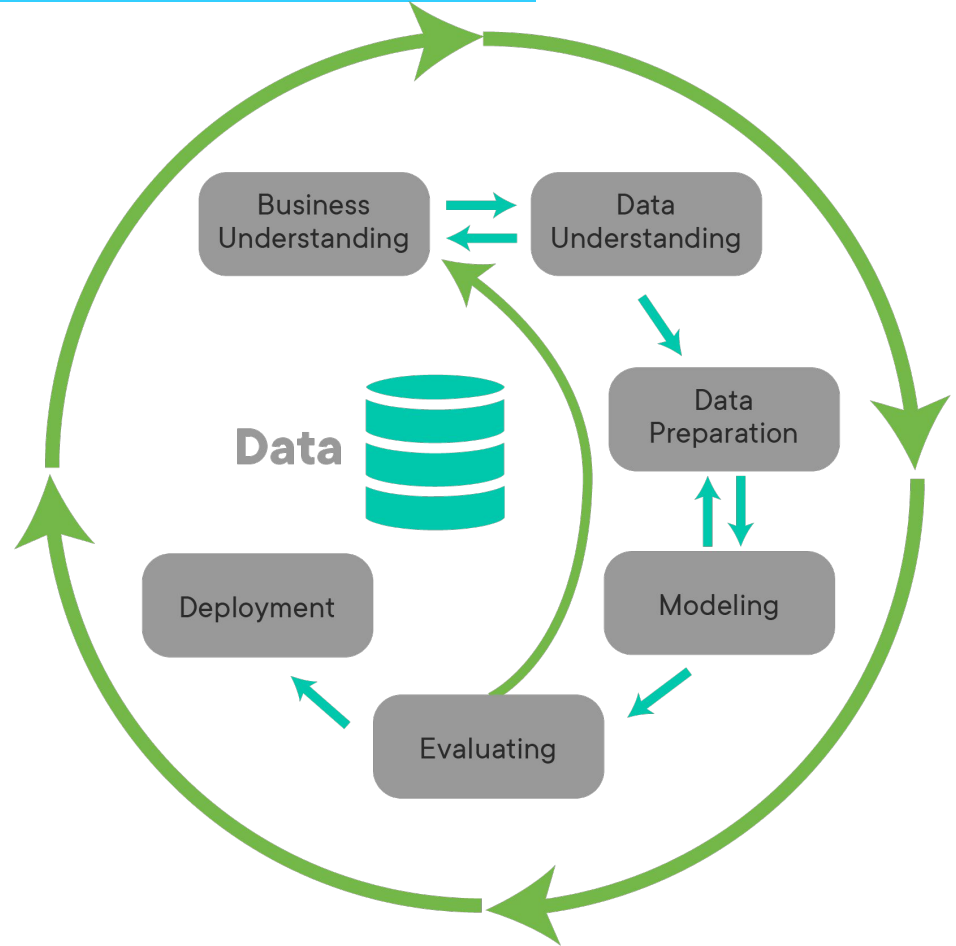
Data Science Decision Making

Adapted from: KDNuggets



Data Science's Cyclical Nature

Cross-Industry Standard Process for Data Mining (CRISP-DM)



<https://www.sv-europe.com/crisp-dm-methodology/>

http://cdn.intechopen.com/pdfs/5937/InTech-A_data_mining_amp_knowledge_discovery_process_model.pdf

Defining “Data Science”

Business understanding	Data understanding	Data preparation	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

Your Data Science Experience #2

Task: Work with your previous partner to revisit one experience you discussed with them.

Step into the shoes of the data scientist responsible for creating the experience you had. What do you imagine that person did or said during the Business Understanding or Evaluation phases?

What goal were they trying to achieve?

Do you think that they think they were successful?



Your Data Science Project

Task: With a new partner, take turns showing each other your regression project and discuss answers to the following questions:

- Which of the phases were strong for your project, and which were limited or didn't occur?
- Evaluate your model from a business perspective, rather than a statistical one. What real-world need might your project have served, and how well would it have done so?
- Imagine that you were going to take what you've learned and redo your project for a real client. What specific activities might you engage in to improve each phase of the process?

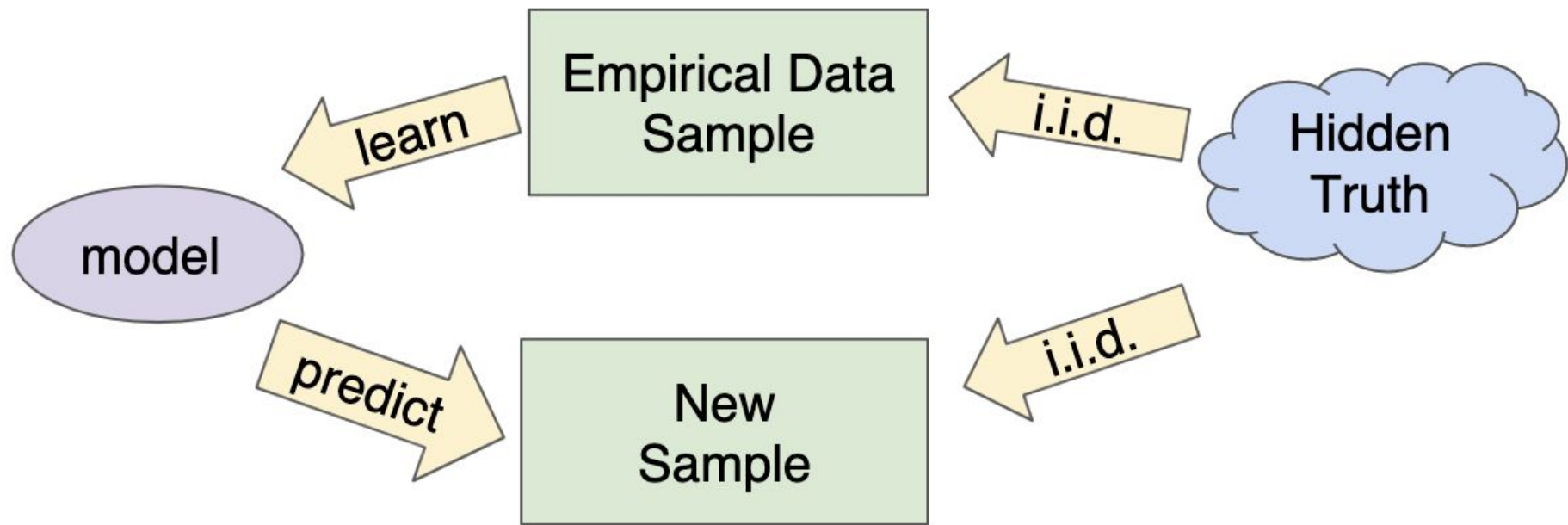


How do we know our model will work in real life?

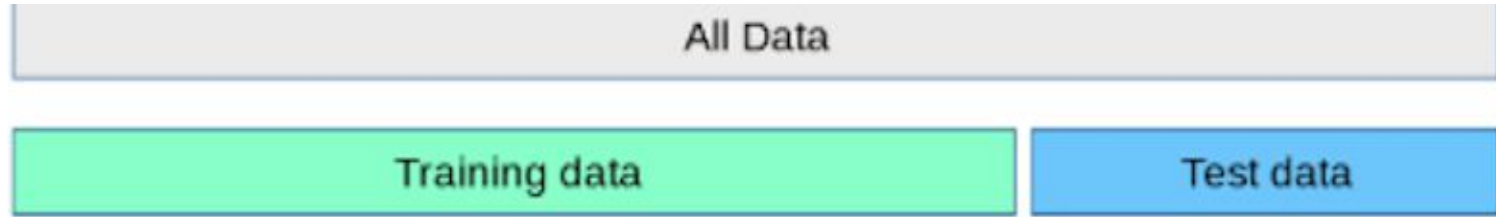
- Validation
- Bias / Variance tradeoff
- Underfitting vs Overfitting
- Feature Engineering vs Regularisation
- Delivering value



Validation



Train Test Split



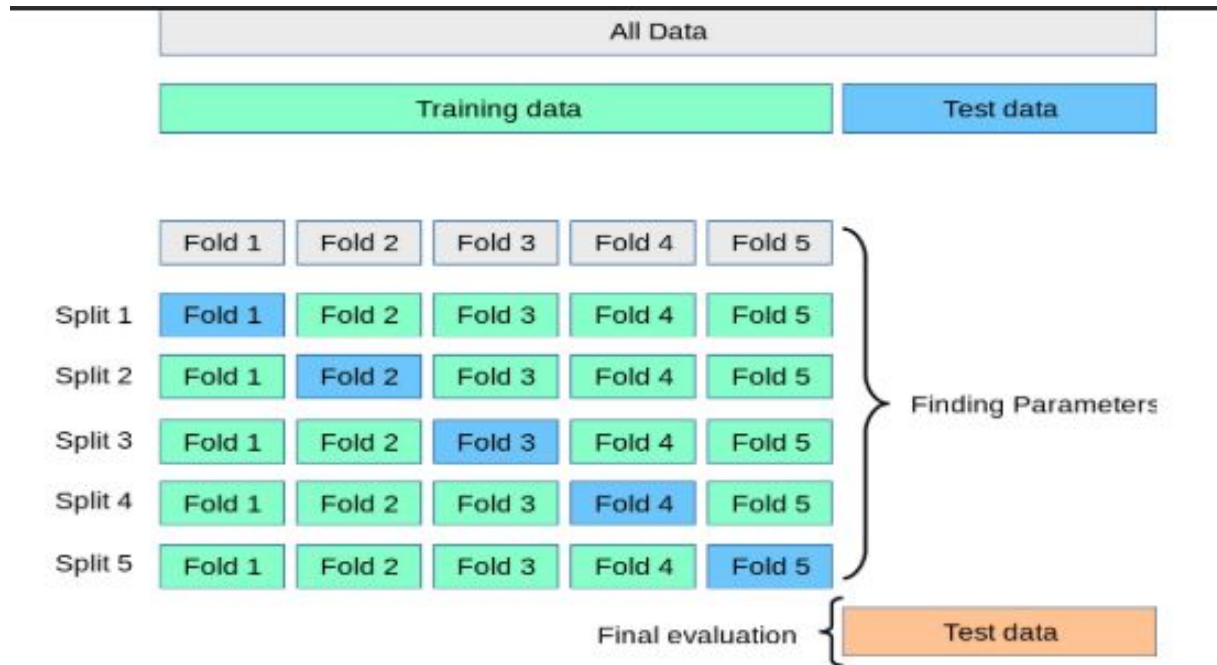
- Requirements:
 - Random sampling of stationary data from the same distribution
 - Large enough test dataset (80/20, min: 1000-3000 samples)
 - Refrain from using test data for model comparison and/or parameter optimisation
 - Use **only on your final model** to understand expected performance

K-fold Cross Validation (CV)

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

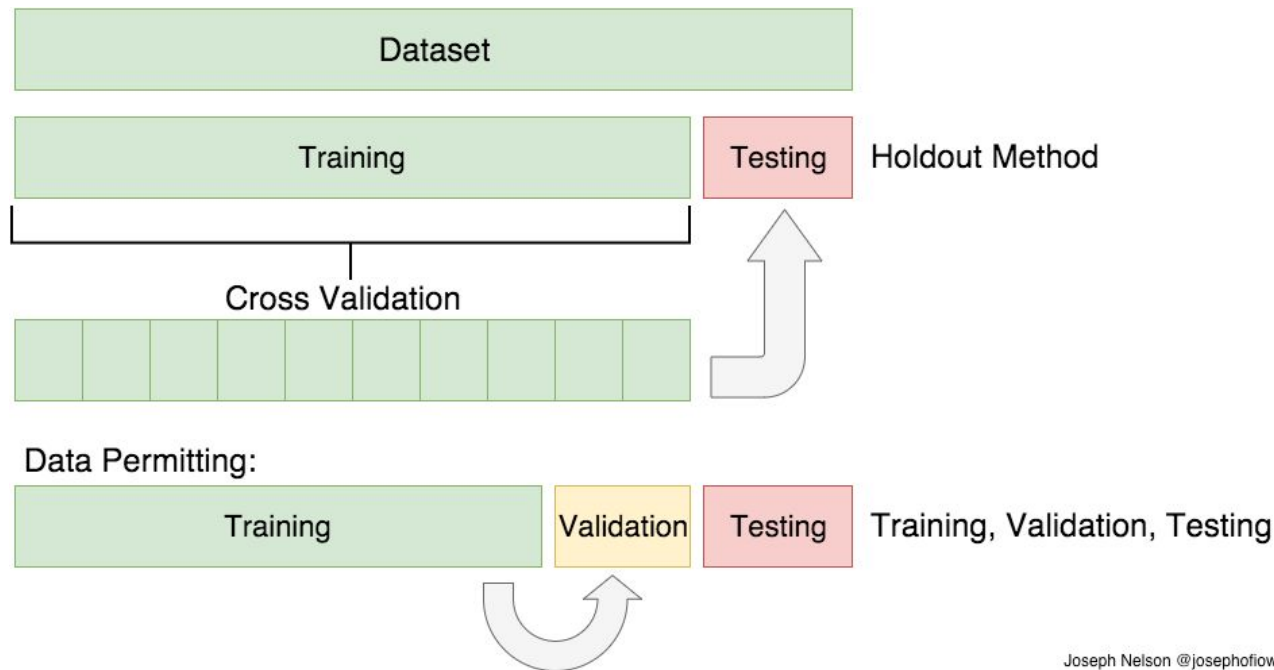
- Makes it possible to work with smaller dataset (higher variance)
- We expect the mean to be closer to the hidden truth than with TTS

Nested CV inside TTS



- By nesting CV inside TTS we can compare multiple models/parameters and select the best one without looking at the the test dataset
- The test dataset is **still only looked at for your final model**

Train / Validation / Test

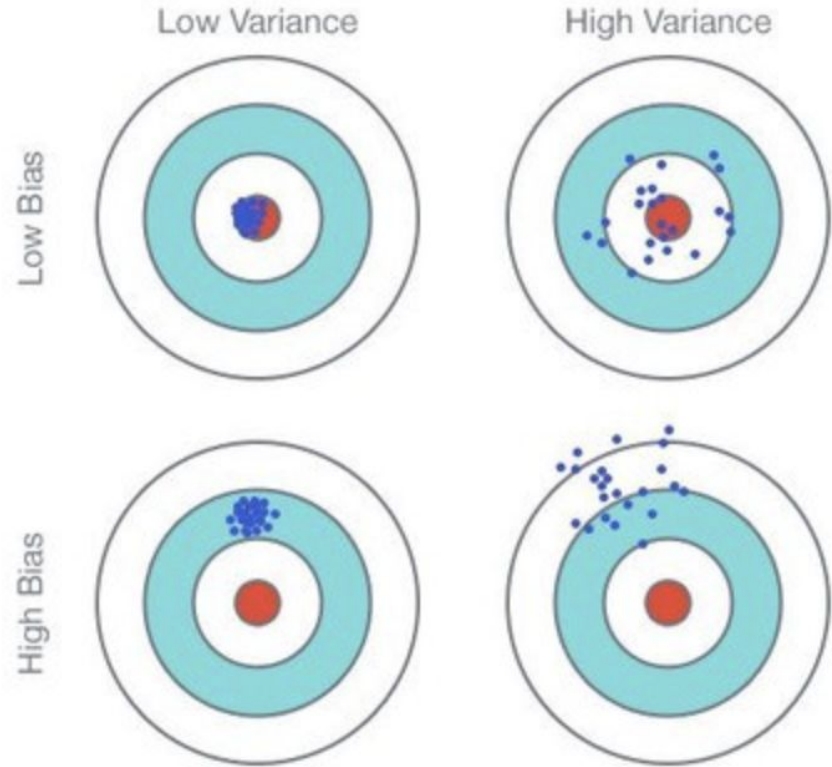


Joseph Nelson @josephofiowa

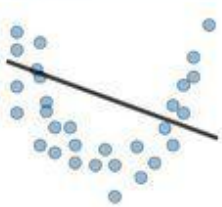


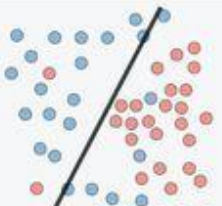
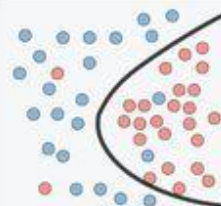
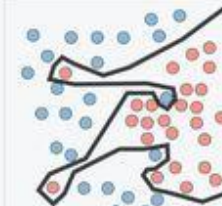


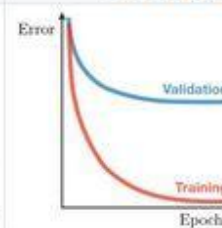
- By nesting TTS inside TTS we eliminate the need for training K times
- Make sure your validation dataset is big enough (same rules as test)
- The test dataset is **still only looked at for your final model**

Bias vs Variance

- The data your model will see after deployment will not be identical to the data it was trained on.
- If the magnitude of your errors change a lot depending on the sample, the model may not work well in real life.
- This may be due to your model learning the idiosyncrasies of the training data too well and expecting to find them in all other data (Overfitting)



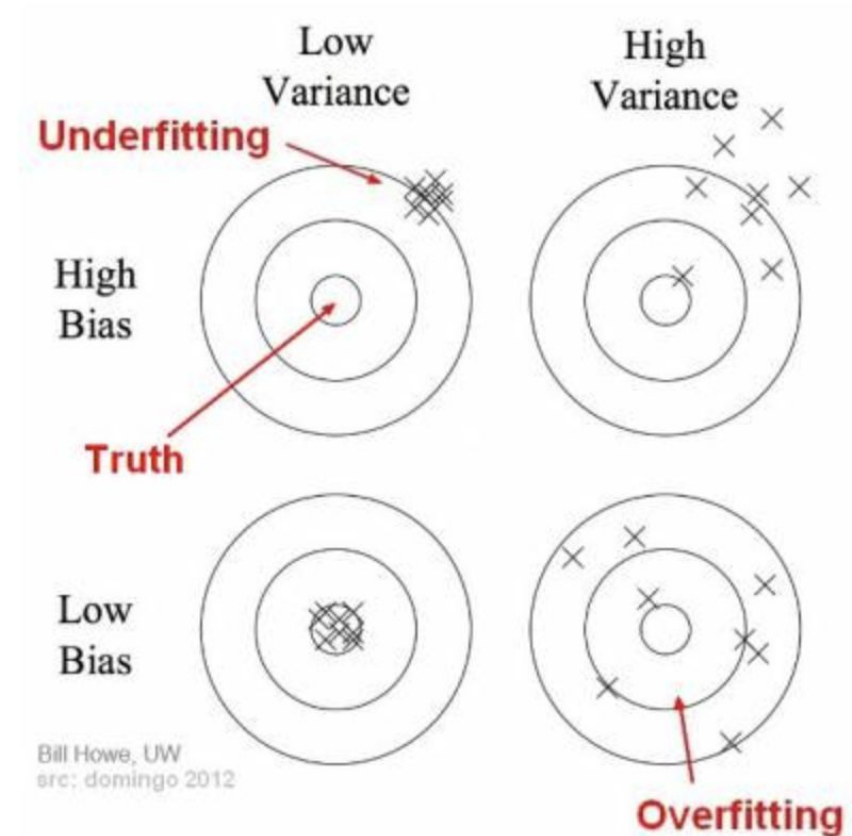
Underfitting vs Overfitting

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> • High training error • Training error close to test error • High bias 	<ul style="list-style-type: none"> • Training error slightly lower than test error 	<ul style="list-style-type: none"> • Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	<ul style="list-style-type: none"> • Complexify model • Add more features • Train longer 		<ul style="list-style-type: none"> • Perform regularization • Get more data

- Overfitting (high variance of error):
 - Very high training performance
 - Training performance much higher than validation
- Underfitting (high average error):
 - Poor performance in the training dataset
- Good fit:
 - Training performance just a bit over test performance
- Unknown fit (assumption violation):
 - Test performance higher than Train performance

Solving fitting issues

- When overfitting:
 - Increase the size of the test dataset (data variance)
 - Reduce the complexity of the model (model variance)
 - Feature elimination (drop redundant / irrelevant features)
- When underfitting:
 - Increase the size of the training dataset (data bias)
 - Increase the complexity of the model (model bias)
 - Feature engineering (add new relevant variables)



Golden Rules

Ockham's Razor

The simpler a model the more likely it will generalise

Golden Rules

Always ask yourself:

“Will the model be good with a new sample of data?”

Golden Rules

Always validate your model on an unseen dataset (test)
And only check it once!!

Treat the test dataset as data points that you will collect one by one in the future
For optimisation use a validation dataset or nested cross validation

Delivering Value

Your job is not to create high performance models

They pay you to **solve problems**

Summary + Exit Ticket

Presented by Dan Sanz