



Quick Explainer

Artificial Intelligence: Glorified Statistics

Machine Learning: Glorified Statistics

Deep Learning: Glorified Bayesian Statistics

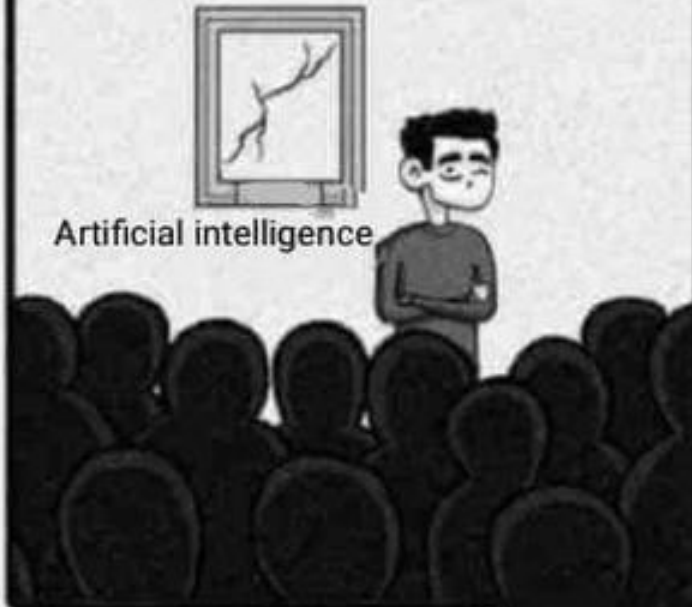
statistics



Machine Learning



Artificial intelligence



Better Explainer

Artificial Intelligence: algorithms that can change by learning from data

Machine Learning: building models that can change by learning from data

Learning? Yes. Models are trained using training data to identify improved generations of the model.

Let's see an example: marl/O

AI/ML Examples

Google Page Rank

Recommendation systems for Netflix or Amazon

Probability is a big part of it

Stochastic: Randomly determined

Non-Deterministic: doesn't always produce the same result

This is where Quantum comes in.

Transistors so small they can't produce a consistent result!

Non-deterministic computing paradigm: Inconsistent results are good, actually

Quantum Annealing: using Qbits to test & evaluate competing AI/ML algorithms/strategies/solutions. *"Find the ground truth."*

Deterministic Computation Paradigm

Perform a function



Agile: Gather
Information

Implies an
"Expected Result"

Non-Deterministic Computation Paradigm

Perform a function

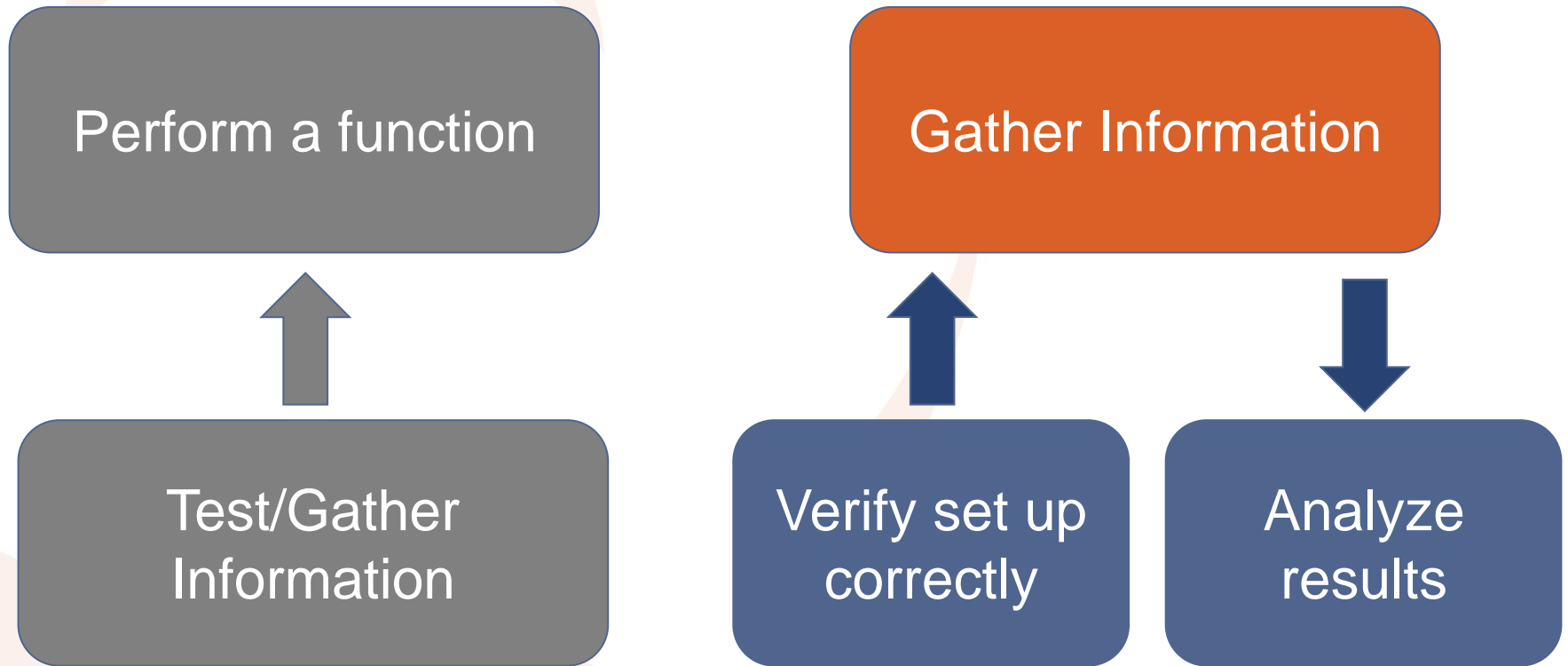


Test/Gather
Information

Implies an
"Expected Result"

"Machine learning systems are difficult to test because they are designed to provide an answer to a question for which no previous answer exists"

Non-Deterministic Computation Paradigm

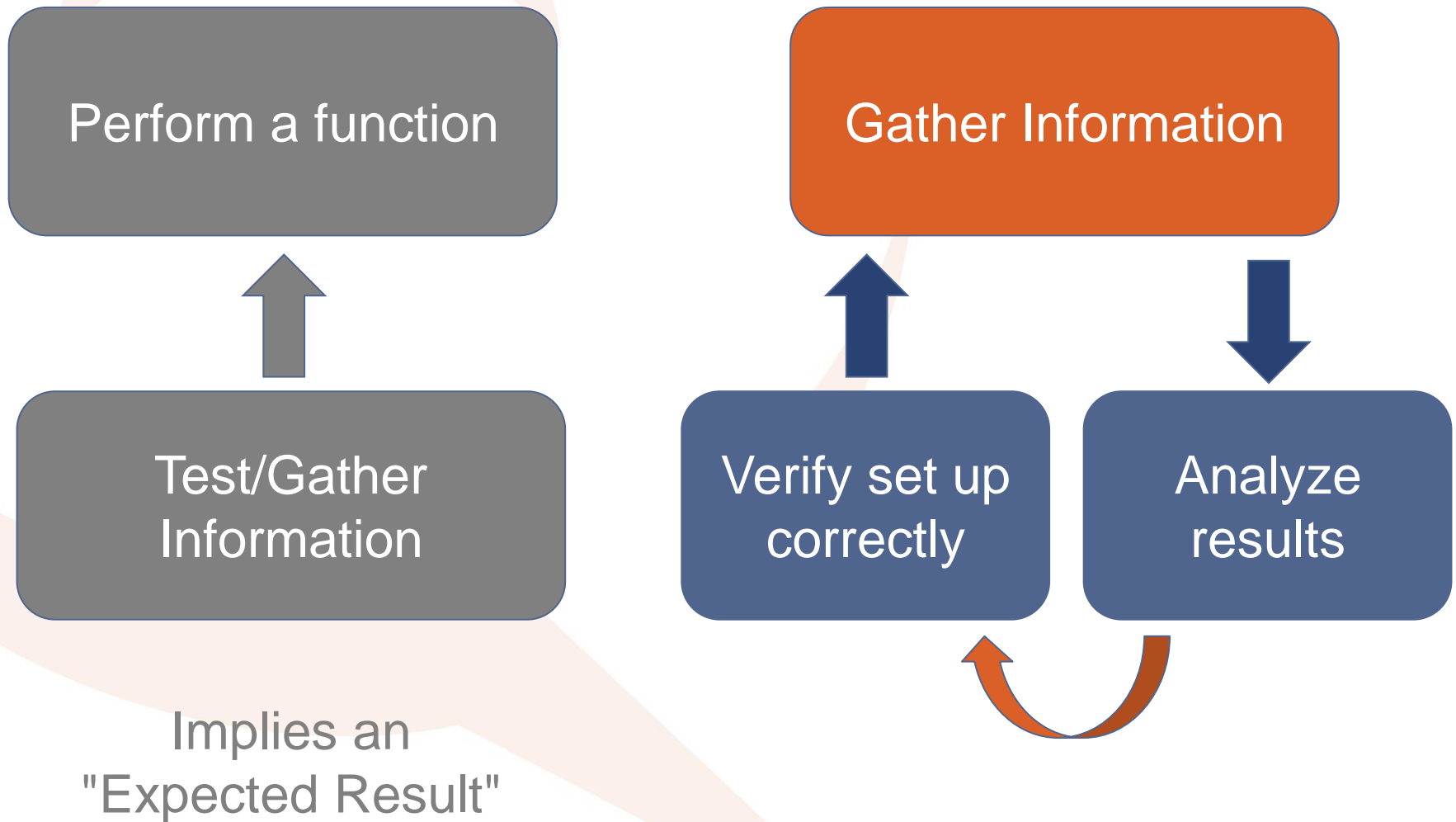


Implies an
"Expected Result"

The role of Test in AI/ML & Quantum

- AI/ML sorta ***is*** testing
- Building the ML models involves a lot of testing
- Doesn't mean there are no AI/ML tester roles, distinct from developer roles. BUT the skillset is based more on data science than epistemology

Non-Deterministic Computation Paradigm

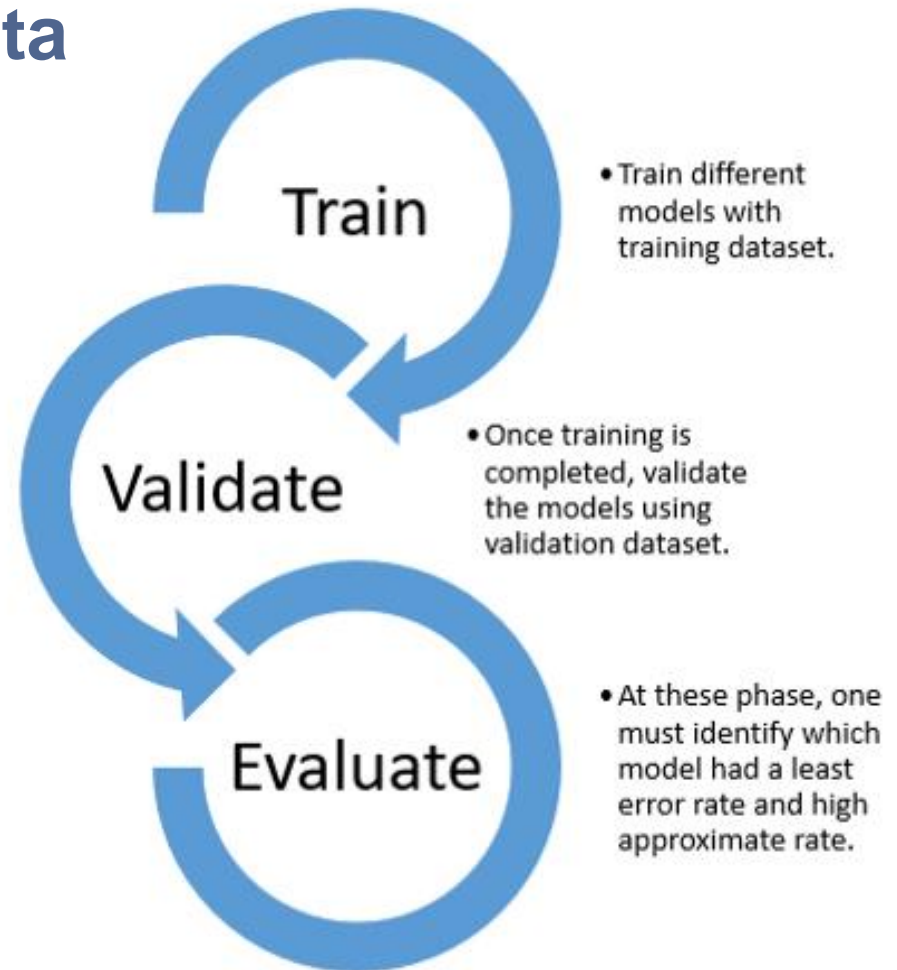


Testing ML things

Train the model using data

Get a dataset, split it into 3 chunks: **training** data (65%), **validation** data (20%), **test** data (15%). Then =====>

Think back to marl/O.
Random distribution of outcomes, but the ones with the strongest "fitness" were kept & built upon.



But what about bugs?

MarI/O found undiscovered bugs in Super Mario Bros. *These bugs had existed for decades and had not been found by humans.*

This is why ML will be useful as a testing technique itself to "find bugs," "assure quality," or more accurately, "gather information."

ML sorta *IS* testing

Training a model, and using it to make predictions, are essentially testing activities.

But you still hunt for bugs

Traditional Test activities in ML

- Validate data: Garbage in garbage out
- Validate model
 - Regression, cross-validation, Turing tests
- Evaluate results
 - Accuracy: true positives/negatives, false positives/negatives
 - Underfitting/Overfitting: trained too closely to training data, won't predict or perform well using non-training data
 - Identify spurious correlations

Less traditional ML test activities

- Understand the questions being asked
 - Success criteria
- Distinguish between Unexpected and Incorrect
- Know what problems testing can solve
 - How to generate inputs that trigger chunks of logic to uncover bugs
 - How to identify errors in the results
- And also what testing can't solve
 - Optimization problems
 - Sampling problems

Digital Twin

A model designed to represent something as closely as possible.

All the properties, characteristics and variables of the DT can be changed, then run through a simulation.

This way you can test different configurations for performance, optimization, correctness, robustness, etc.

This stuff takes a LOT of work

- Requirements analysis, asking the right questions, defining success criteria
- Generating data, validating data, Ideating data-driven tests
- Choosing the right algorithm, making predictions, evaluating results, choosing the best candidate model
- Evaluating "non-functionals" like performance, security, data privacy, fairness
- Simulation building, digital twin development

And this doesn't even count the coding of the learning program, algorithms selected, or ML framework. There is a non-developer role here but it requires highly trained and specialized skillsets.

Resources

AI – [StackExchange](#), [TesterStories](#), [Medium](#),

ML Models – [Tutorial](#), [Amazon](#), [Dummies](#), [Dummies](#), [Quora](#),

ML Testing – [ArXiv](#), [DeepXplore](#), [TesterStories](#), [Medium](#), [Google](#), [Wikipedia](#),
[Overfitting](#)

Digital Twin – [Wikipedia](#),

Quantum Annealing – [D-Wave](#), [Google](#), [Wikipedia](#)

Contact Us

Jacob Stevens – Project Management & Quality Control Consultant

206.734.6696

jacobs@quardev.com

