A QUALITY ENGINEERING INTRODUCTION TO AI+ML

1

ABOUT ME

TARIQ KING

FLORIDA TECH | FIU FLORIDA INTERNATIONAL UNIVERSITY | IBM | NDSU | Ultimate SOFTWARE | test.ai | epam

2

ABOUT: YOU

Agile + DevOps

PARTICIPANTS

3

COURSE GOALS

What you'll learn and how you can apply it

- Key challenges associated with developing, testing, and debugging AI/ML systems
- How autonomous and intelligent agents are modeled, trained, and evaluated
- Performance measures and methods for evaluating ML models prior to release
- Different types of AI/ML bias and how they impact software development risks
- Techniques for testing adaptive AI/ML systems online in production environments
- ML operations (MLOps) for continuous training, testing, integration, and deployment

4

COURSE GOALS

And you'll be able to:

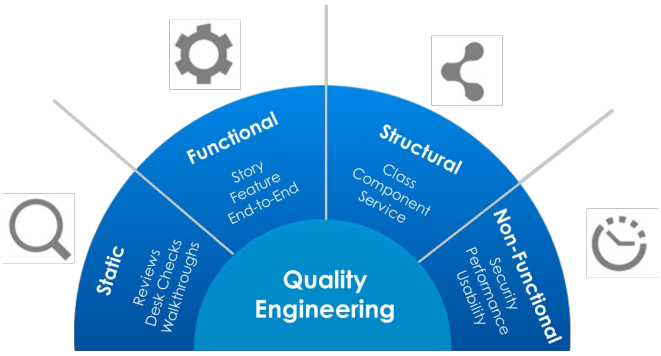
- Validate and evaluate the performance of ML models using popular Python libraries
- Use open source AI fairness toolkits to examine, report, and mitigate bias in ML models
- Develop and execute an end-to-end test plan for an ML-based system or component
- Create an automated ML pipeline for continuous model training, validation, and testing

5

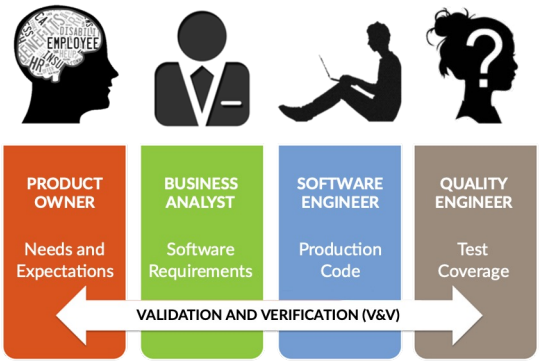
QUALITY ENGINEERING

6

QUALITY ENGINEERING (QE)



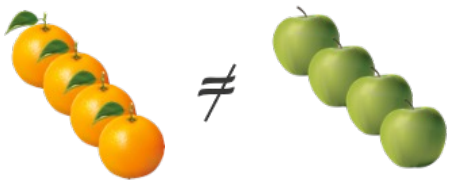
Focuses on building quality into the product and engineering process.



Validation and verification at every stage of the software development lifecycle.

QUALITY ENGINEERING (QE)

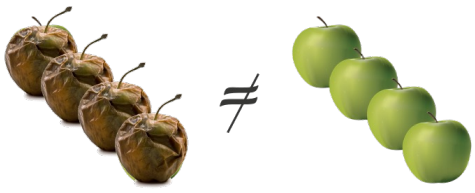
Starts with...



Building the Right Product

Validation

Continues with...



Building the Product Right

Verification

QUALITY ENGINEERING (QE)

Ends with...

OR...

Happy | Unhappy Customers

Customer Impact

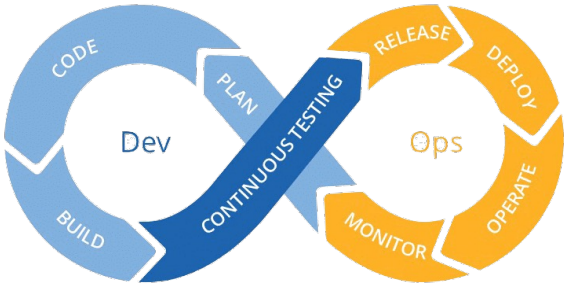
Which translates to...

OR...

Project Success | Failure

Business Impact

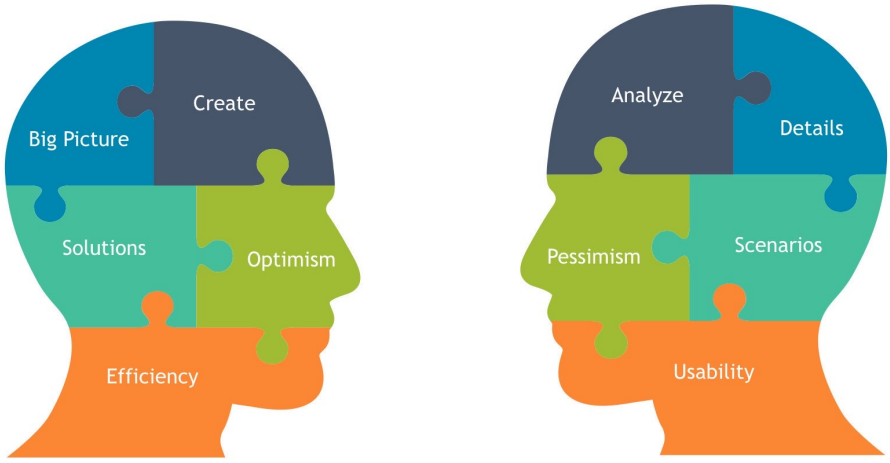
QUALITY ENGINEERING (QE)



Not just development.
But also, operations.

Less about quality practices and processes.
More about fostering a culture of quality.

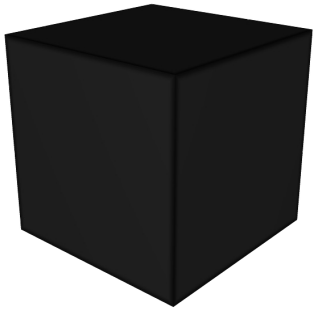
THE VALUE OF PERSPECTIVE



Developing Testing

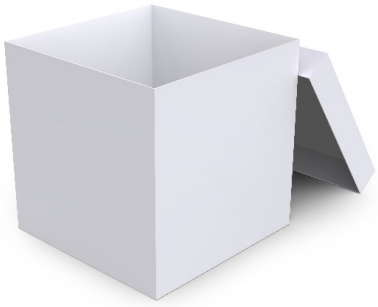
SOFTWARE TESTING

APPROACHES



Black Box

Gray Box

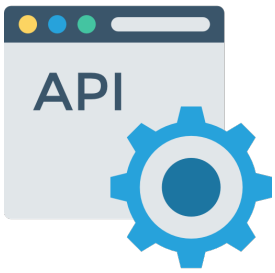


White Box

SOFTWARE TESTING

LEVELS

System Level
User Interface



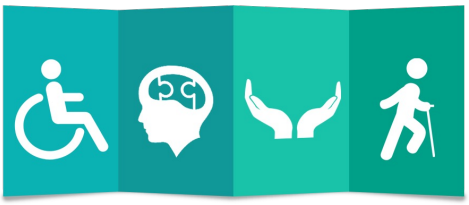
Integration Level
Service or API

Unit Level
Class or Microservice

SOFTWARE TESTING

QUALITY ATTRIBUTES

Performance

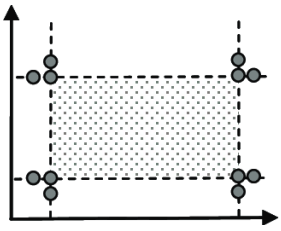


Accessibility

Security

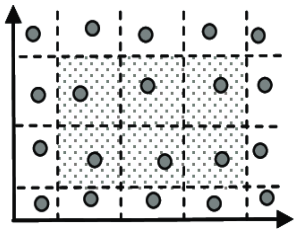
SOFTWARE TESTING

TECHNIQUES



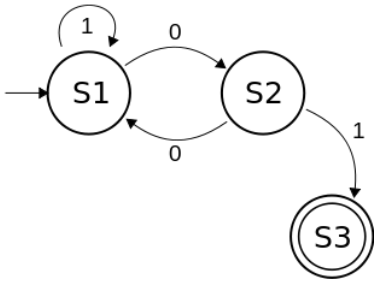
Error-Based

Off-By-One Errors
Extreme Conditions



Coverage-Based

Equivalence Class Testing
Pairwise Testing

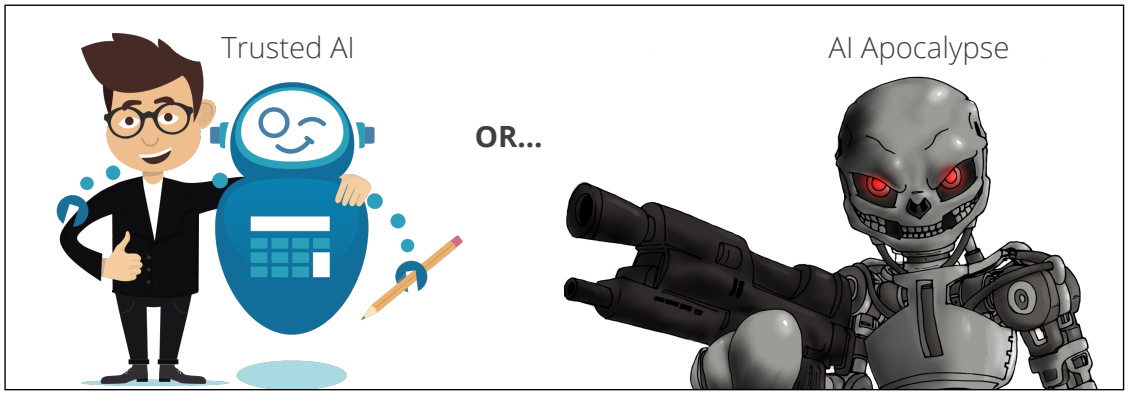


Model-Based

State Machine Testing
Decision Table Testing

QE AND TESTING IMPLICATIONS FOR AI

Just might be the difference between...

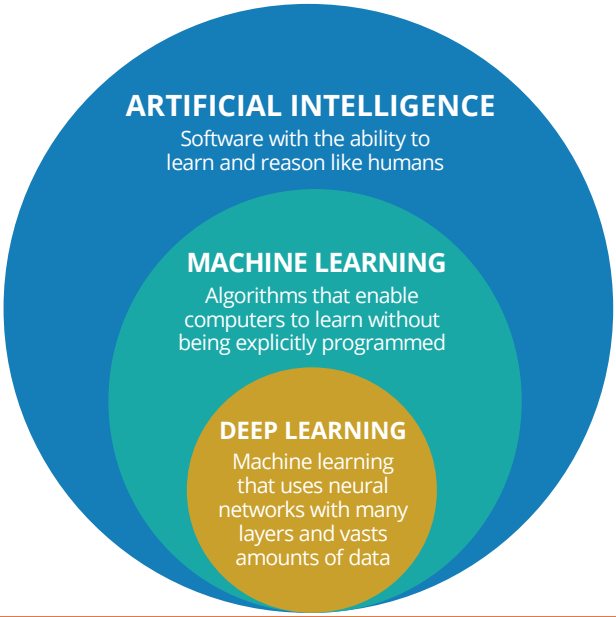


Potential Global Impact

AI & MACHINE LEARNING

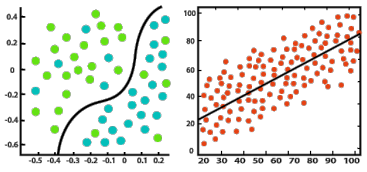
17

AI & ML

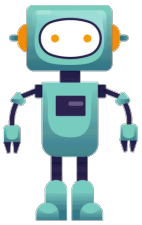


18

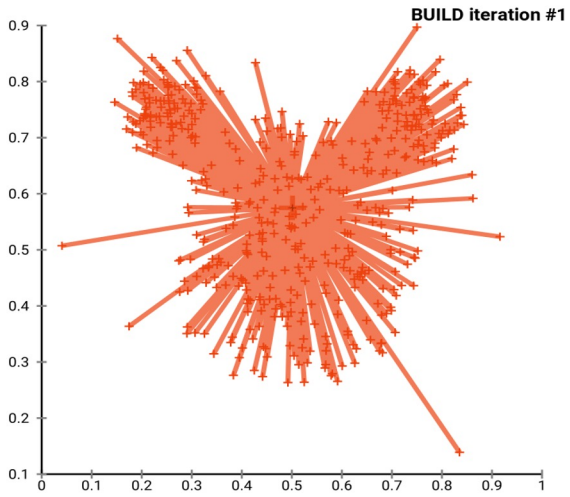
SUPERVISED LEARNING



Classification Regression

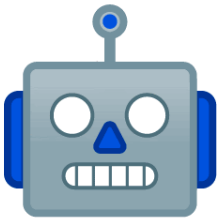


UNSUPERVISED LEARNING



REINFORCEMENT LEARNING

INTELLIGENT AGENTS: BOTS



Agent

Environment

INTELLIGENT AGENTS: TYPES OF BOTS

SIMPLE REFLEX AGENTS

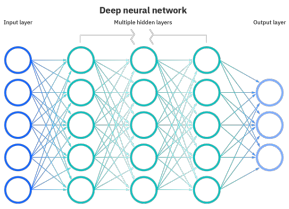
MODEL-BASED AGENTS

GOAL-BASED AGENTS

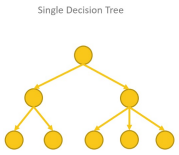
UTILITY-BASED AGENTS

LEARNING AGENTS

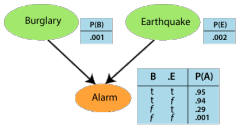
INTELLIGENT AGENTS: ML MODELS



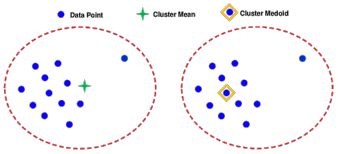
NEURAL NETWORKS



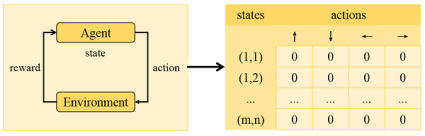
DECISION TREES AND RANDOM FORESTS



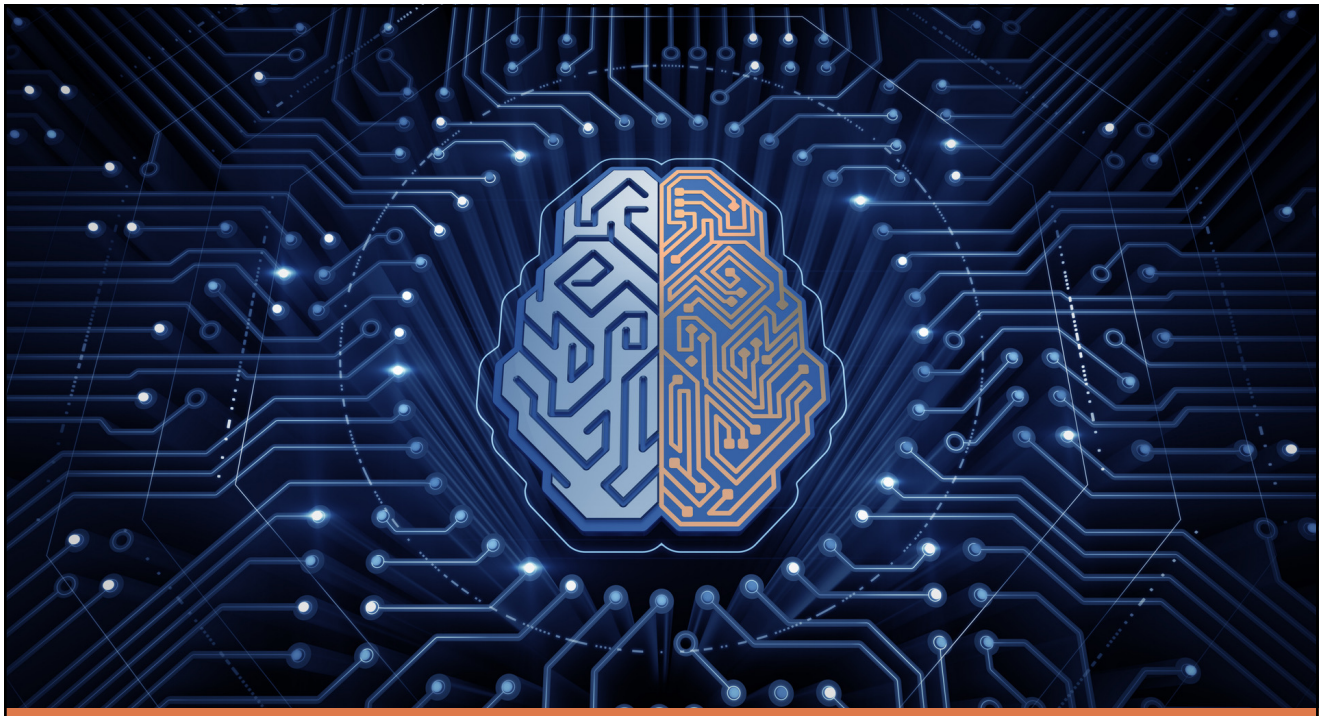
BAYESIAN NETWORKS



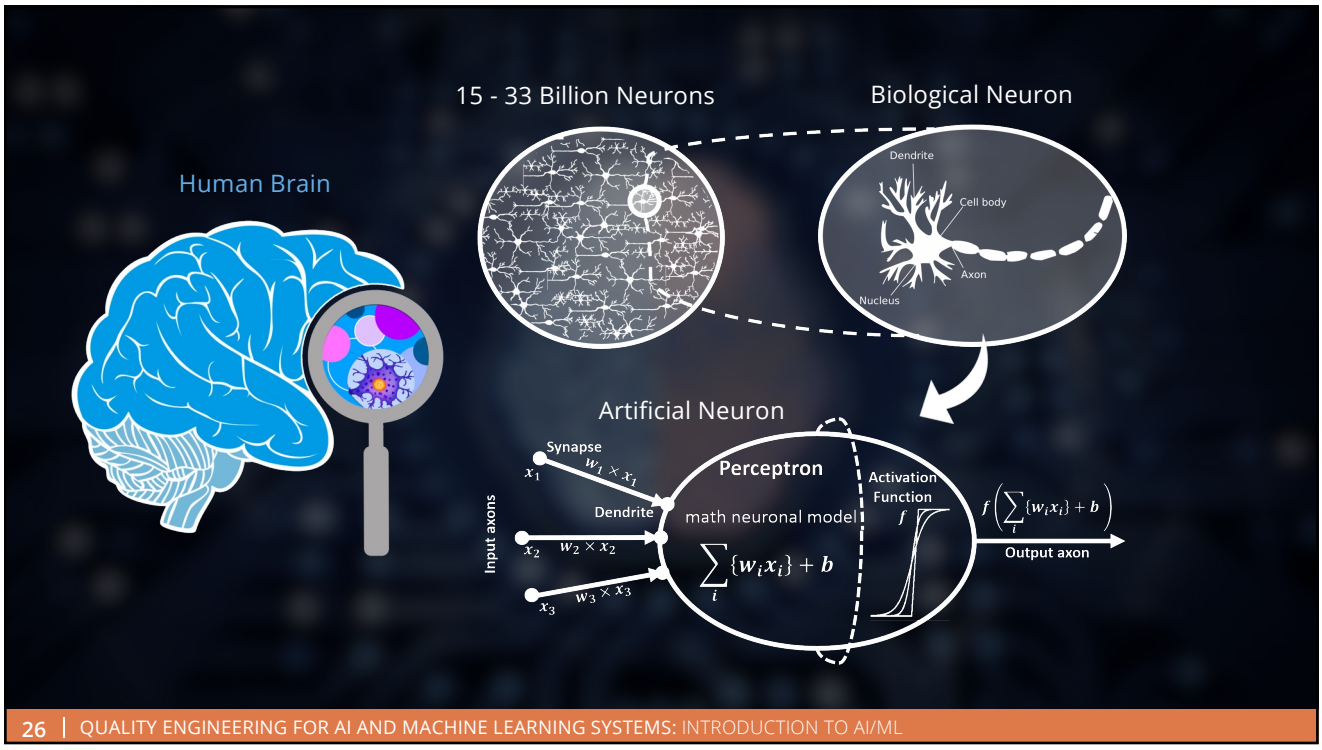
K-MEANS AND K-MEDOIDS CLUSTERING



Q-LEARNING

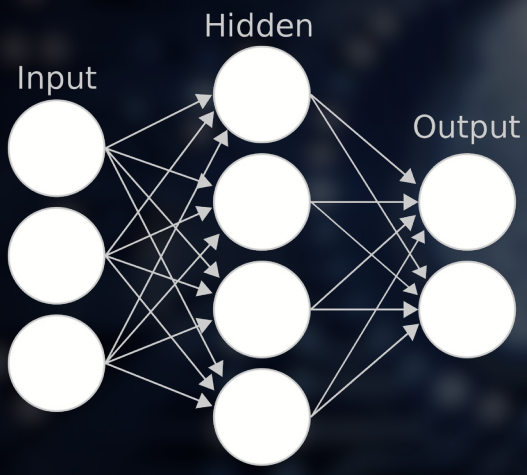


25


15 - 33 Billion Neurons
Biological Neuron
Human Brain
Dendrite
Cell body
Axon
Nucleus
Artificial Neuron
Synapse
x_1
$w_1 \times x_1$
Dendrite
x_2
$w_2 \times x_2$
x_3
$w_3 \times x_3$
Input axons
Perceptron
math neuronal model
$\sum_i \{w_i x_i\} + b$
Activation Function
f
$f\left(\sum_i \{w_i x_i\} + b\right)$
Output axon

26

ARTIFICIAL NEURAL NETWORKS

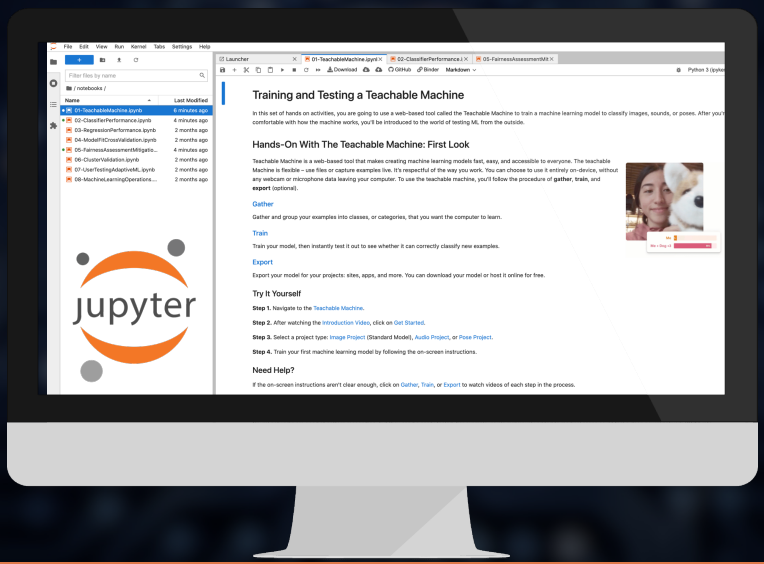


"A feed-forward network with a single layer is sufficient to represent any function…"

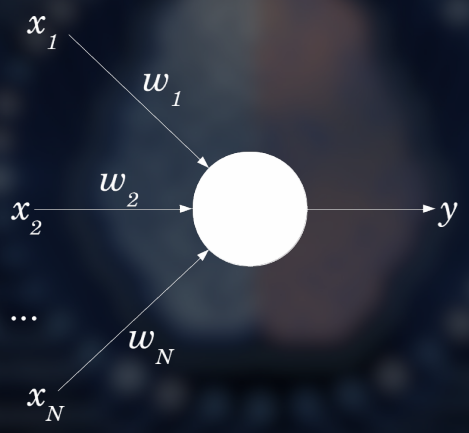
– Ian Goodfellow

ARTIFICIAL NEURAL NETWORKS

HANDS-ON: THE TEACHABLE MACHINE



ARTIFICIAL NEURAL NETWORKS: BUT HOW?

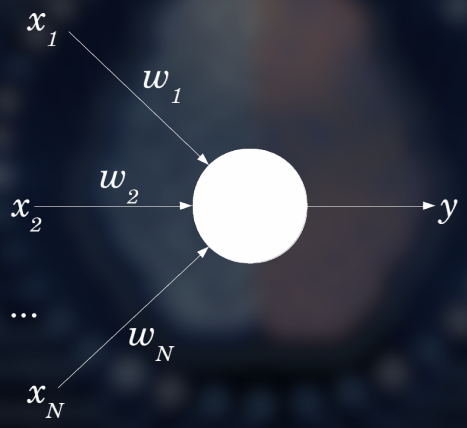


Each neuron is a function that maps inputs to outputs.

HOW NEURAL NETWORKS LEARN

Given examples, **find weights** that map inputs to outputs.

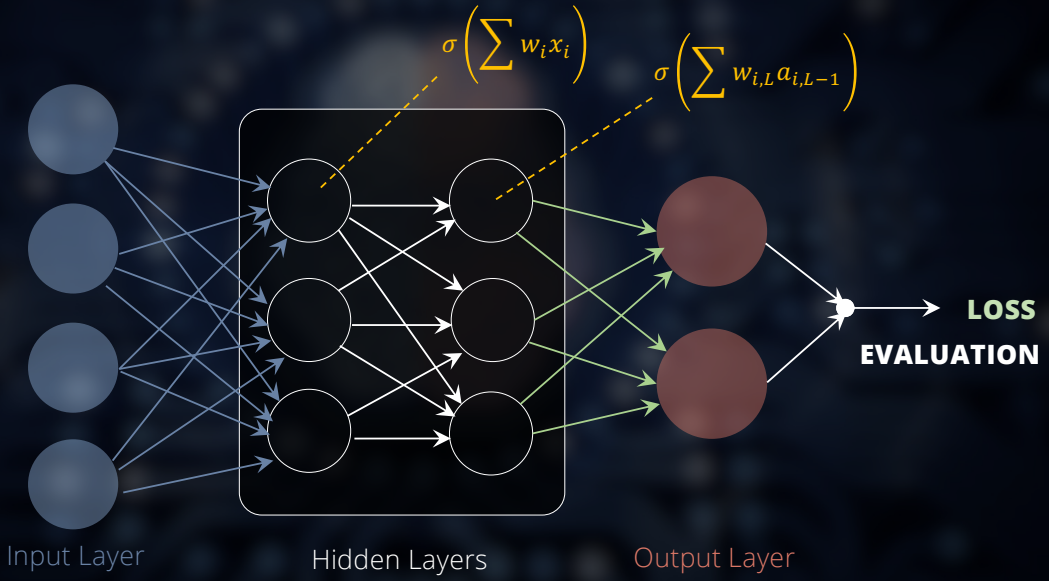
Initially, weights are **randomly** initialized.



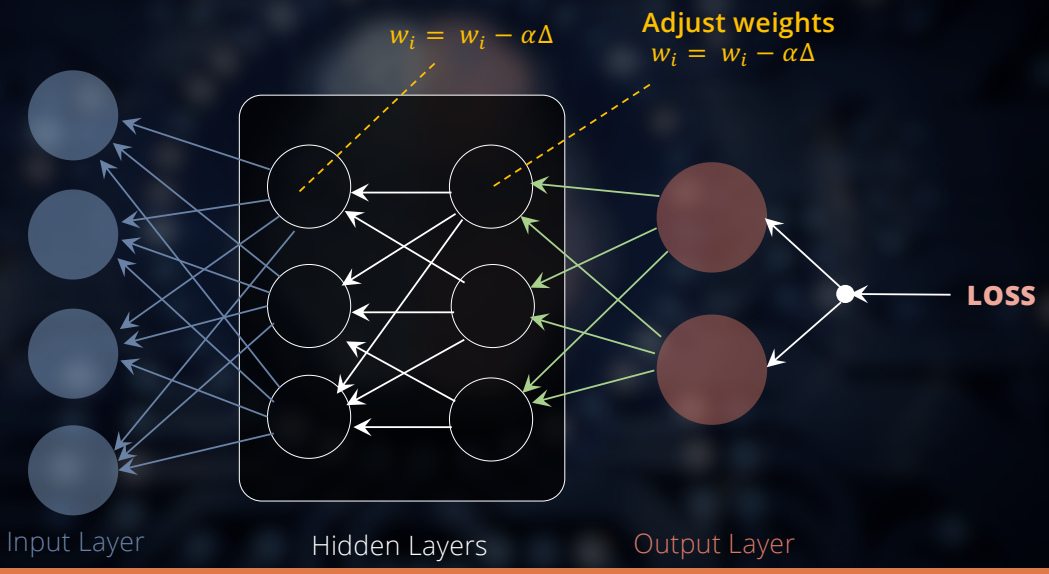
Value of **predicted output** $(\hat{y})$ may be different from the **actual output** (y).

A **LOSS** function is used to measure this difference.

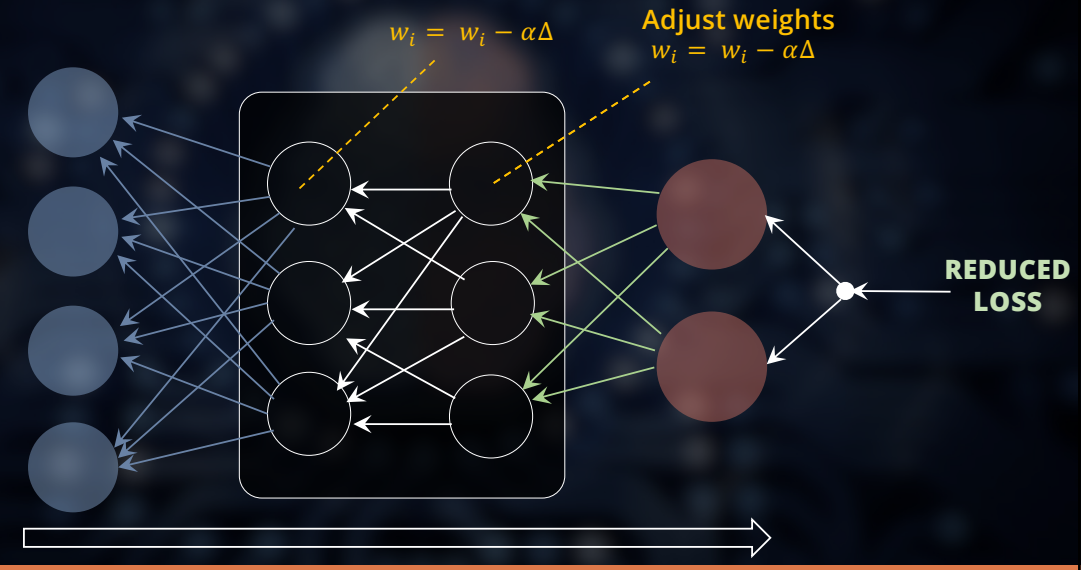
HOW NEURAL NETWORKS LEARN



HOW NEURAL NETWORKS LEARN



HOW NEURAL NETWORKS LEARN

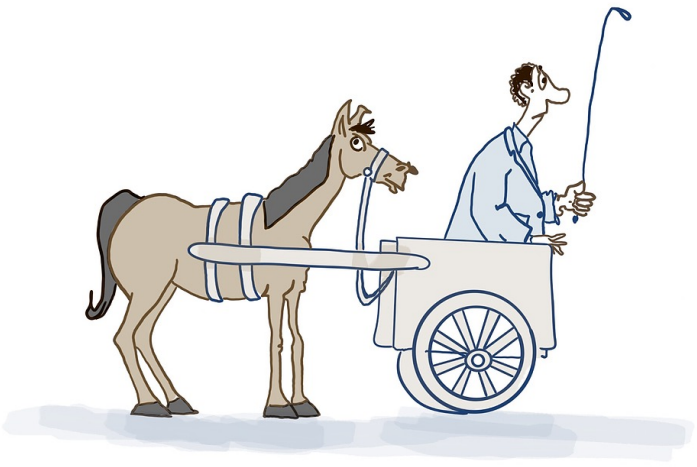


QE PRACTICES FOR AI/ML

35

QE FOR AI/ML: FOUNDATIONAL PRACTICES

AI PROBLEM FIRST, THEN SOLUTION



Define the problem

- What do want to accomplish?
- What are the desired outcomes?
- Do you have the data to infer those outcomes?

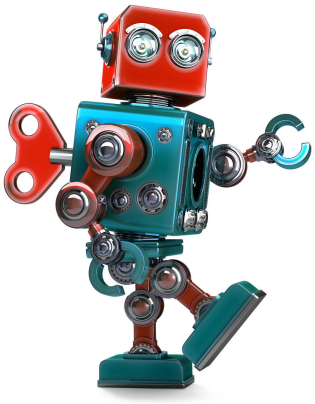
Consider the solutions

- Can the problem be solved by AI?
- Should it be solved by AI?
- Is there a simpler solution that does not require AI?

36

QE FOR AI/ML: FOUNDATIONAL PRACTICES

CHOOSE ML ALGORITHMS BASED ON NEEDS, NOT POPULARITY



Algorithms differ in…

- Kinds of problems they can solve
- Level of detail of the output
- Interpretability of the model/output
- Robustness to adversaries

Do not use as a shiny new toy

- Change algorithms to meet needs as the system evolves or the environment changes.

QE FOR AI/ML: FOUNDATIONAL PRACTICES

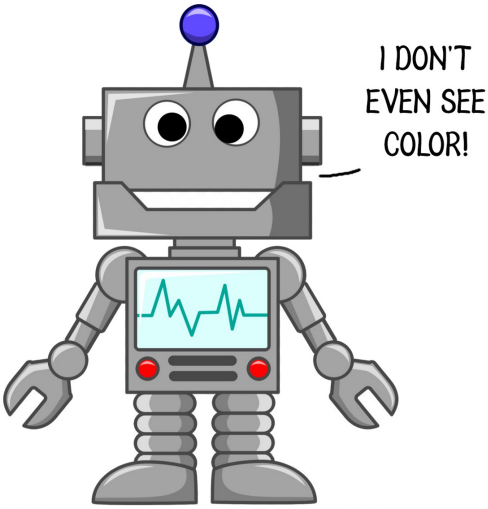
DESIGN, TEST, AND MONITOR AI FOR ETHICS

Account for organizational and societal values in all aspects of system…

- Ethics in data collection (privacy).
- Ethics in data representation (fairness)
- Ethics in decision making (fairness)
  - Training Process and model structure

Validate dataset and monitor system…

- Pre- and post-release
- Continuously where possible.



QE FOR AI/ML: FOUNDATIONAL PRACTICES

RULE YOUR DATA OR BE RULED BY IT

Data can impact quality directly...

- Output of an AI system is often tied to the data used to train it.

Data requires work...

- Ingestion, cleansing, protection
- Validation, testing, monitoring
- Overall management of data requires automation to scale.

QE FOR AI/ML: FOUNDATIONAL PRACTICES

IMPLEMENT FLEXIBLE, EXTENSIBLE SOLUTIONS

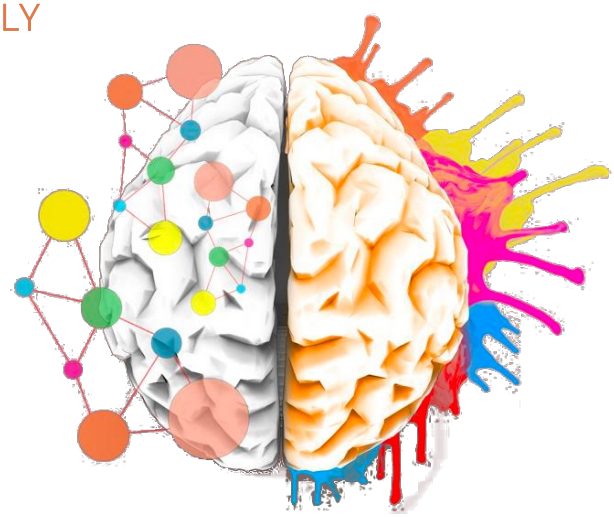
Boundaries of AI components are more sensitive than traditional software due to entanglement of data...

- Direct and indirect data dependencies can trigger changes in....
  - Functionality
  - Expected output
  - Supporting infrastructure
- Requires system design to be...
  - Loosely coupled
  - Highly extensible

QE FOR AI/ML: FOUNDATIONAL PRACTICES

THINK HOLISTICALLY

Science



Art

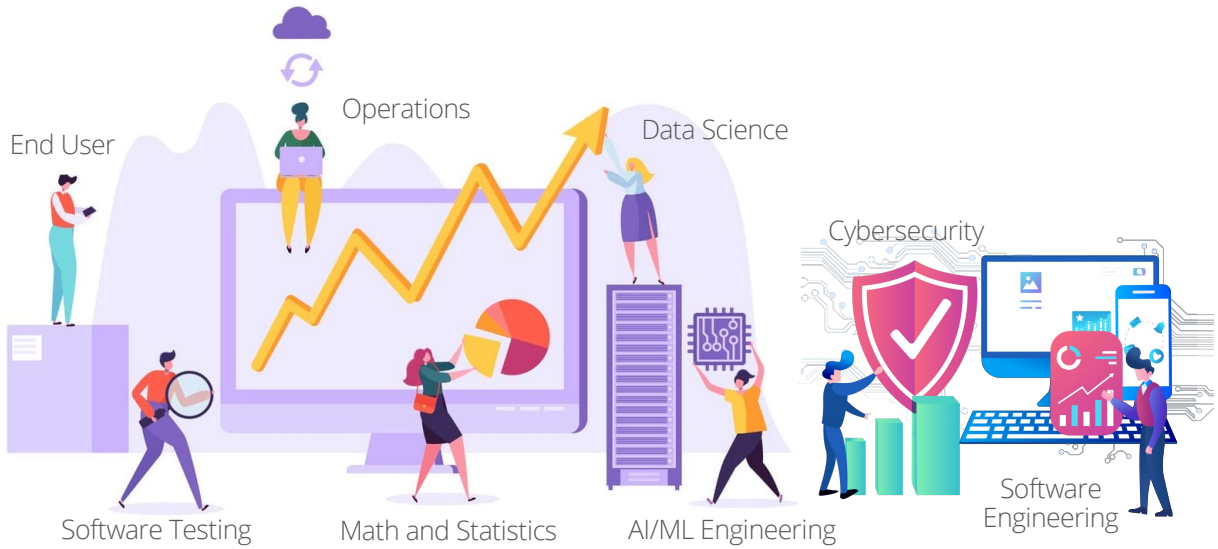
TOWARDS HOLISTIC QE FOR AI AND ML

CURRENT FOCUS: ACCURACY, PRECISION, RECALL

TOWARDS HOLISTIC QE FOR AI AND ML

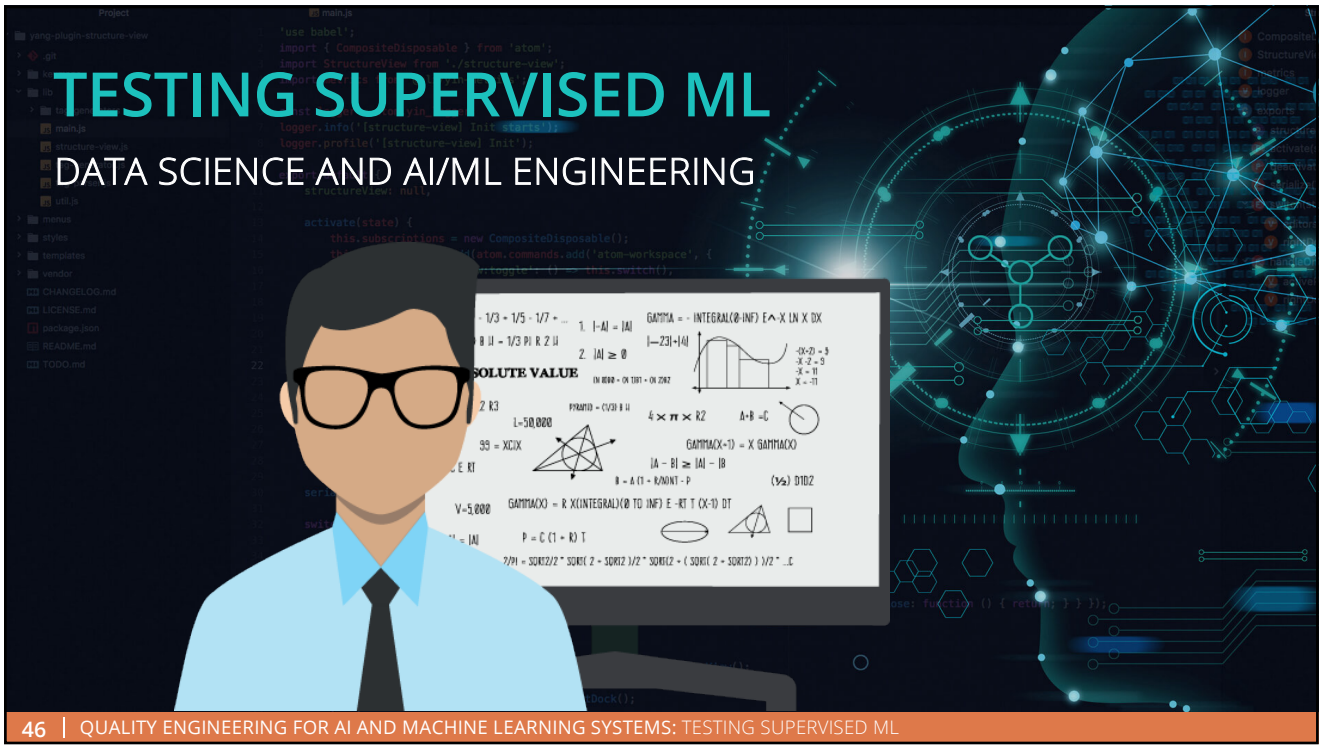
REMINDER: AI SYSTEMS ARE SOFTWARE-INTENSIVE

QE FOR AI AND ML: A HOLISTIC APPROACH



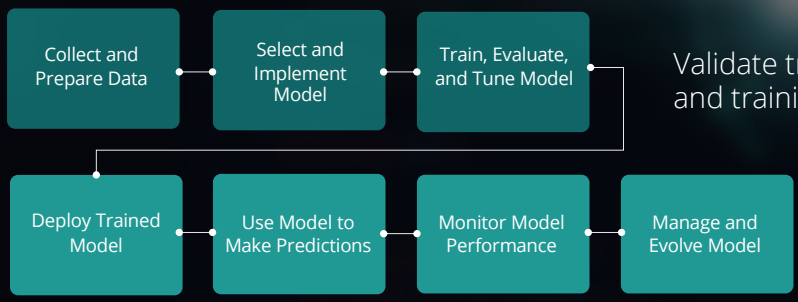
TESTING SUPERVISED ML

MODEL TRAINING & CLASSIFIER PERFORMANCE



ML ENGINEERING: FOUNDATION

7 STEPS OF MACHINE LEARNING



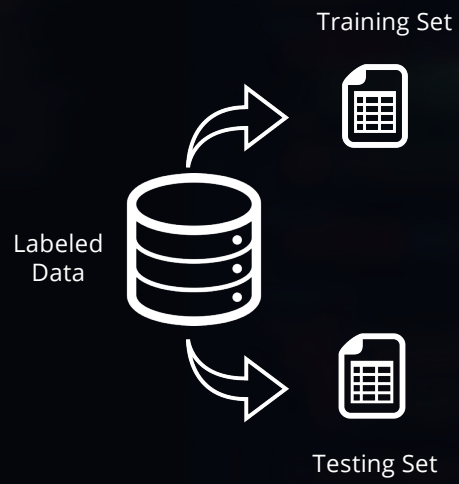
Validate training data, model selection, and training process.

Evaluate model performance pre- and post- deployment.

47

MODEL TRAINING

CLASSIFICATION



Features

Class

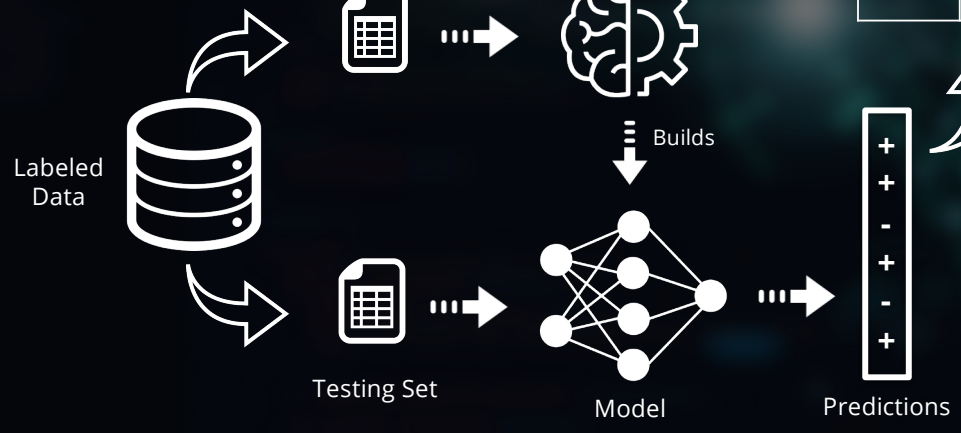
	Outlook	Temp	Humidity	Windy	Play Tennis
1	Rainy	Hot	High	False	No
2	Rainy	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Sunny	Mild	High	False	Yes
5	Sunny	Cool	Normal	True	Yes
6	Sunny	Cool	Normal	True	Yes
7	Overcast	Cool	Normal	True	Yes
8	Rainy	Mild	High	False	No

9	Rainy	Hot	High	False	?
10	Rainy	Hot	High	True	?
11	Overcast	Hot	High	False	?
12	Sunny	Mild	High	False	?

48

MODEL TRAINING

CLASSIFICATION



Confusion Matrix

		Predicted Class	
		Yes	No
Actual Class	Yes	4	1
	No	0	10

CLASSIFIER PERFORMANCE

ACCURACY AND ERROR

Accuracy = $\frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$

= 14 ÷ 15 = ~**93.3%**

Error = $\frac{\text{Number of Wrong Predictions}}{\text{Total Number of Predictions}}$

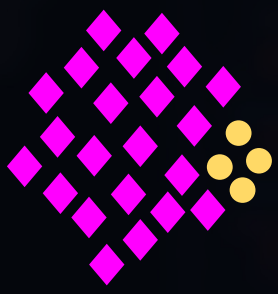
= 1 ÷ 15 = ~**6.7%**

Confusion Matrix

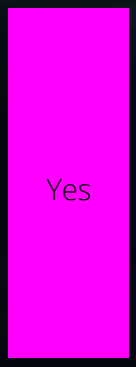
		Predicted Class	
		Yes	No
Actual Class	Yes	4	1
	No	0	10

CLASSIFIER PERFORMANCE

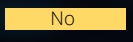
ACCURACY AND ERROR HAVE LIMITS



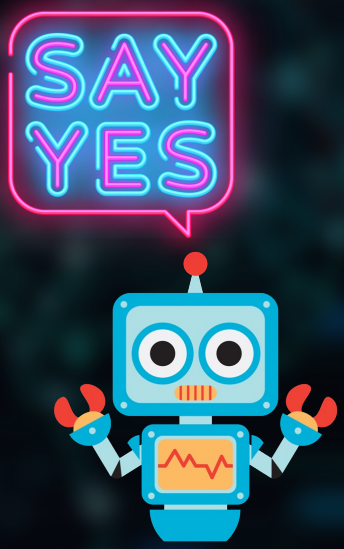
Class Imbalance



Majority Class



Minority Class



YesBot

CLASSIFIER PERFORMANCE

TRUE VERSUS FALSE PREDICTIONS

True Predictions

- TP: True Positives
  - Model Predicted Yes, and Actual Class is Yes
- TN: True Negatives
  - Model Predicted No, and Actual Class is No

False Predictions

- FP: False Positives
  - Model Predicted Yes, and Actual Class is No
- FN: False Negatives
  - Model Predicted No, and Actual Class is Yes

Confusion Matrix (Reloaded)

		Predicted Class: Yes	Predicted Class: No
Actual Class	Yes	TRUE POSITIVES	FALSE NEGATIVES
	No	FALSE POSITIVES	TRUE NEGATIVES

Predictions

Evaluation

CLASSIFIER PERFORMANCE

PRECISION, RECALL, F-SCORE

$$\textbf{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\textbf{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\textbf{F-Measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Confusion Matrix (Reloaded)

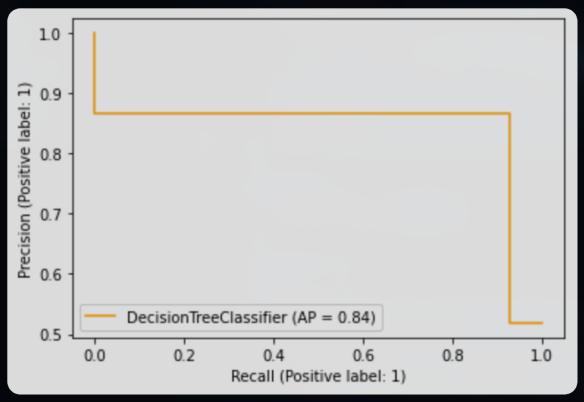
		Predicted Class: Yes	Predicted Class: No
Actual Class	Yes	TRUE POSITIVES	FALSE NEGATIVES
	No	FALSE POSITIVES	TRUE NEGATIVES

CLASSIFIER PERFORMANCE

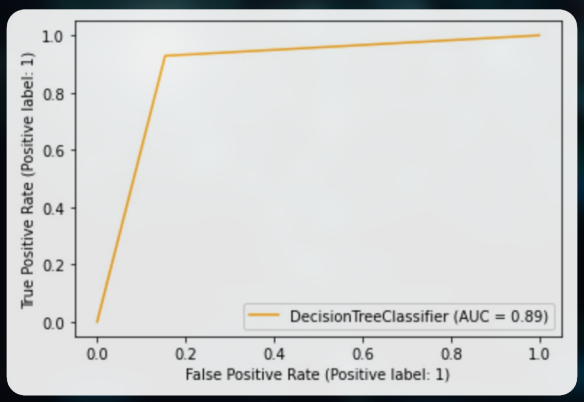
PROBABILISTIC FORECASTING

CLASSIFIER PERFORMANCE

PRECISION-RECALL CURVE, ROC CURVE, AND AUC-ROC SCORE



Precision-Recall Curve

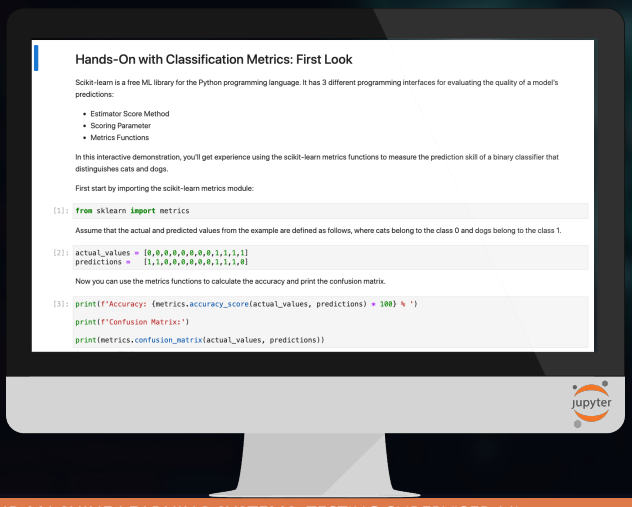


Receiver Operating Characteristic (ROC) Curve

55

HANDS-ON: CLASSIFICATION PERFORMANCE

FIRST LOOK, DEEP DIVE, AND EXERCISE



56

TESTING SUPERVISED ML

REGRESSION PERFORMANCE & MODEL VALIDATION

57

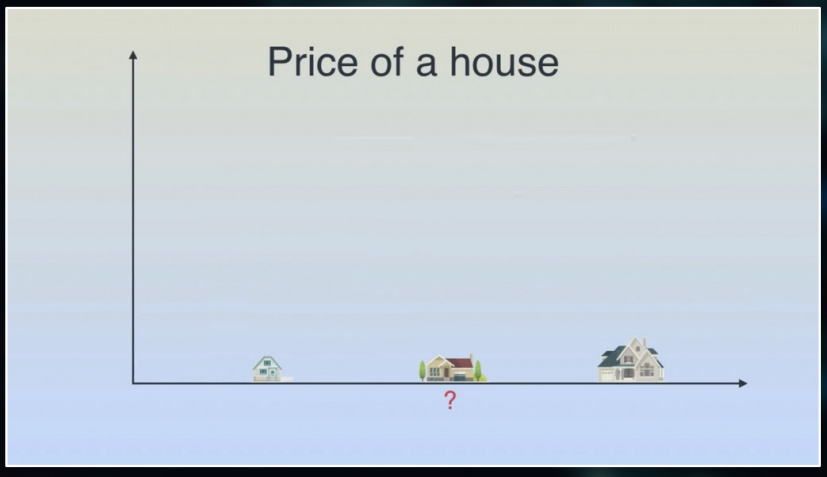
REGRESSION PERFORMANCE

INTRODUCTION

Predicts a Quantity
- Continuous values
- Amounts, sizes, prices, etc.

Example:
- Selling Price of a House

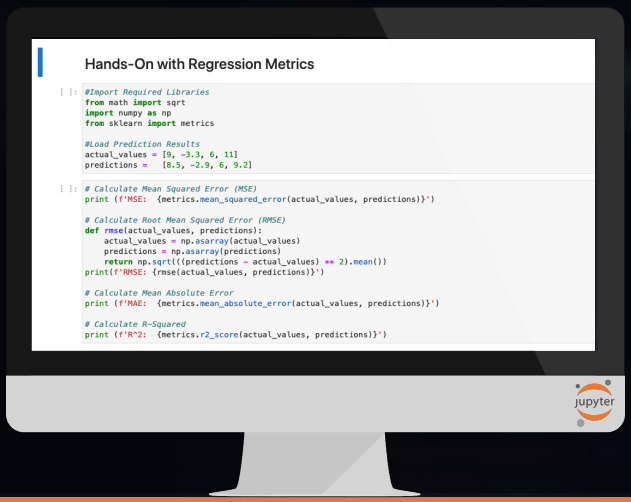
Question:
- How would you calculate the accuracy for this regression model?



58

HANDS ON: REGRESSION PERFORMANCE

ERROR (DISTANCE) METRICS



Mean Squared Error

Root Mean Squared Error

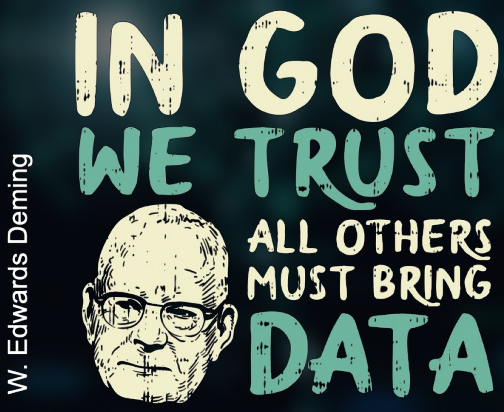
Mean Absolute Error

R-Squared

MODEL VALIDATION: MOTIVATION

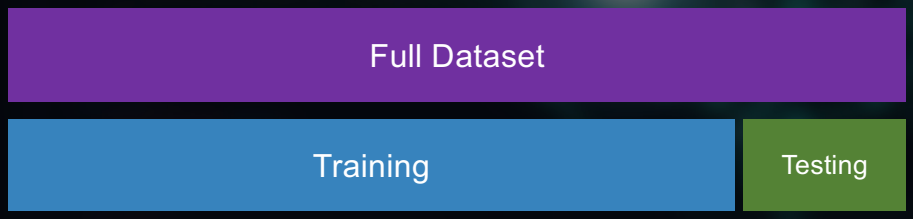
TWO SIMILAR QUALITY PHILOSOPHIES

Q=P



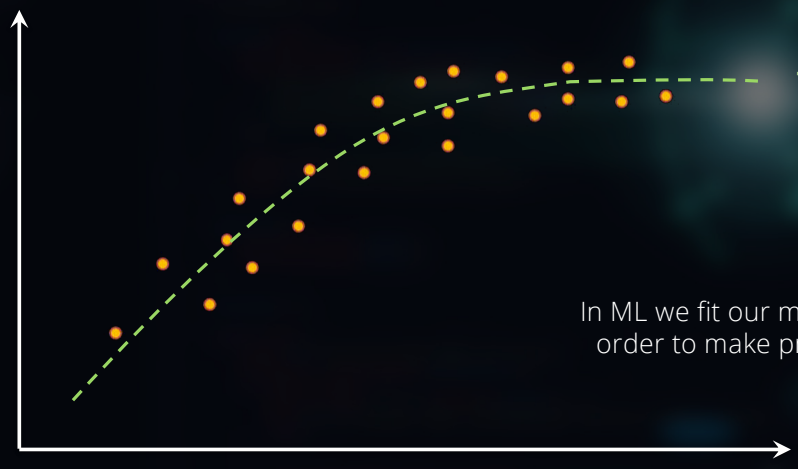
MODEL VALIDATION: INTRODUCTION

RECALL THE TRAIN-TEST SPLIT



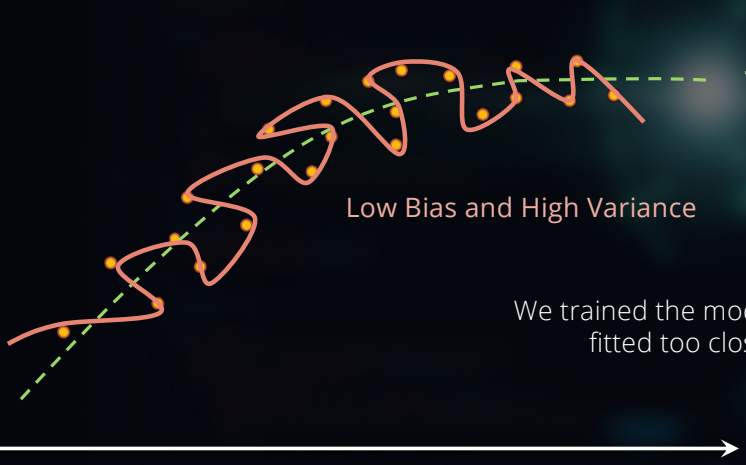
Typically this is an **80/20** or a **70/30** split.

MODEL VALIDATION: FITTING THE MODEL



In ML we fit our model on the training data in order to make predictions on the test data

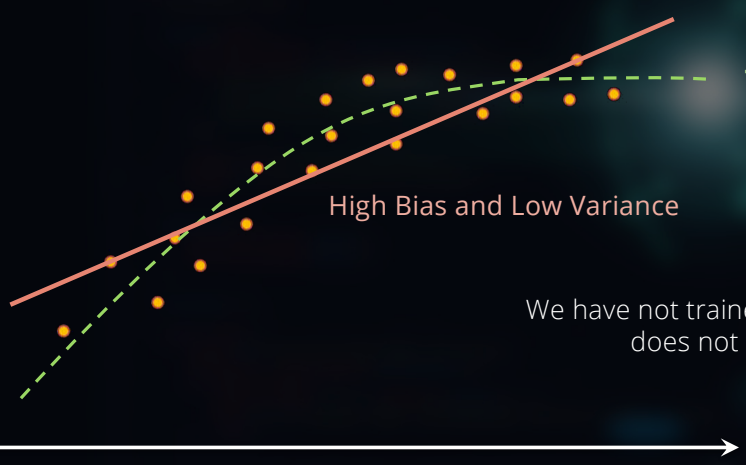
MODEL VALIDATION: OVERFITTING



Low Bias and High Variance

We trained the model "too well" and it has been fitted too closely to the training set.

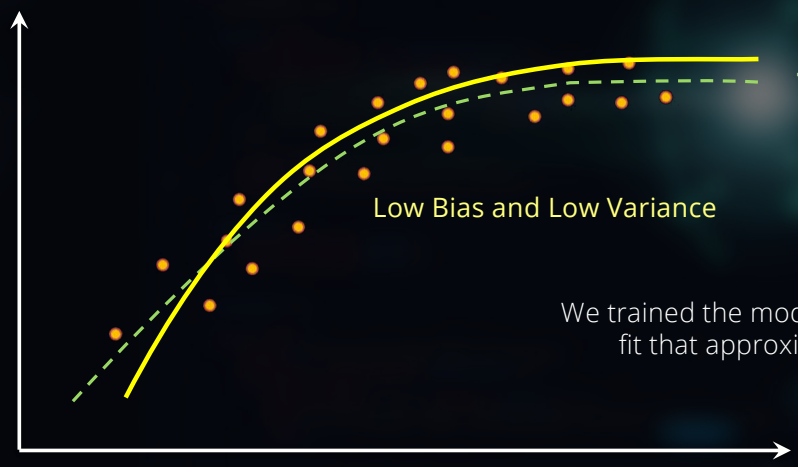
MODEL VALIDATION: UNDERFITTING



High Bias and Low Variance

We have not trained the model enough and it does not fit the training data

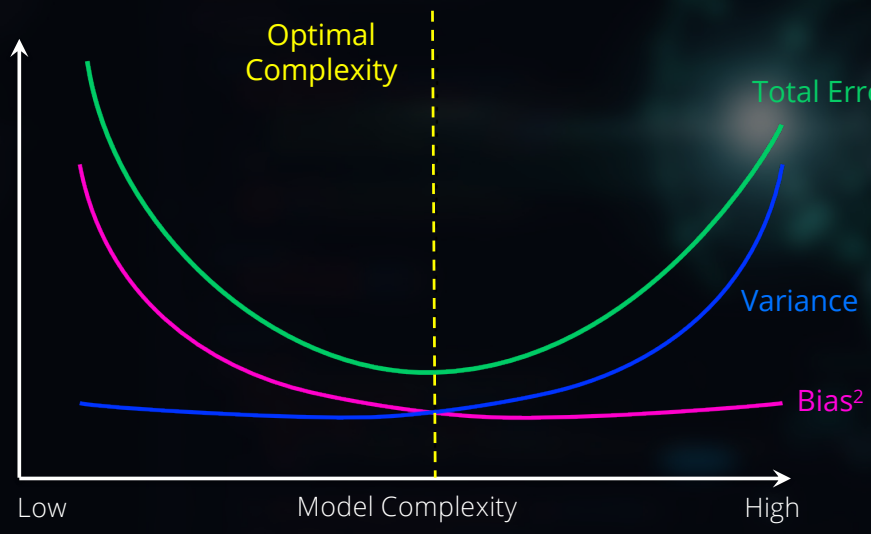
MODEL VALIDATION: BALANCED FITTING



We trained the model "just right" and it is a good fit that approximates the true function.

65

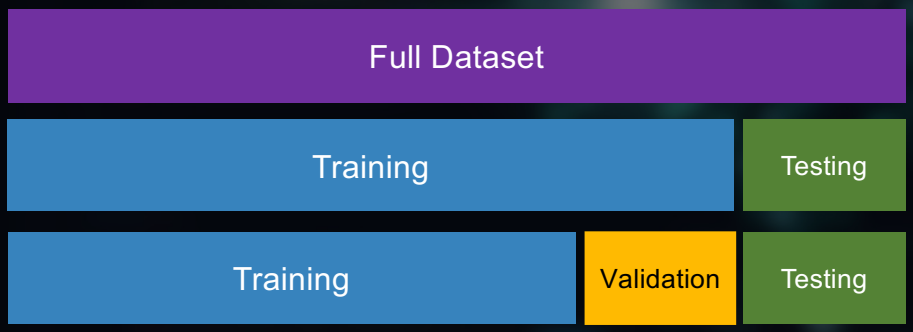
MODEL VALIDATION: BIAS-VARIANCE TRADE-OFF



66

MODEL VALIDATION: REVISITING THE SPLIT

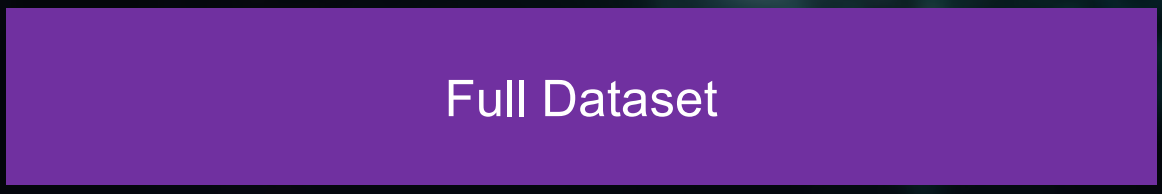
TRAINING-VALIDATION-TESTING



67

CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION



68

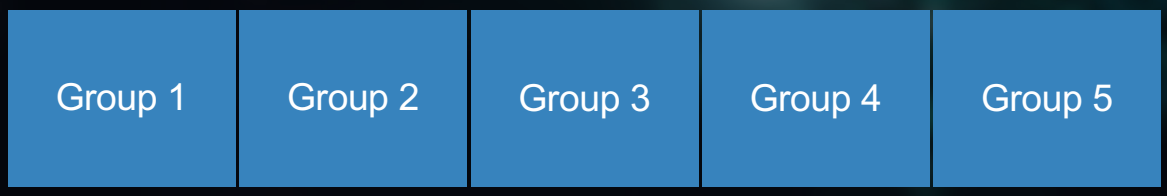
CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION

Shuffling the Dataset

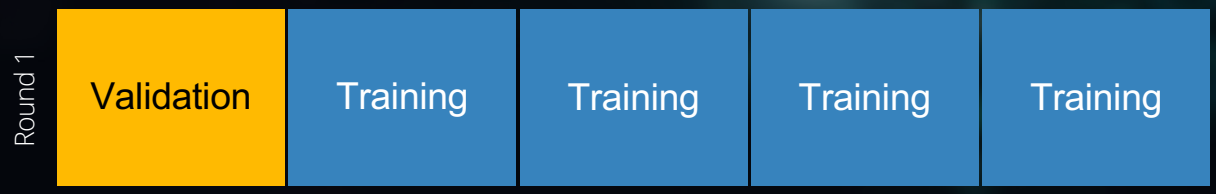
CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION (k=5)



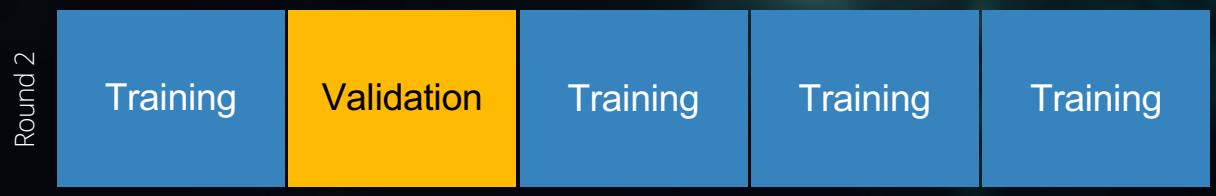
CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION (k=5)



CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION (k=5)

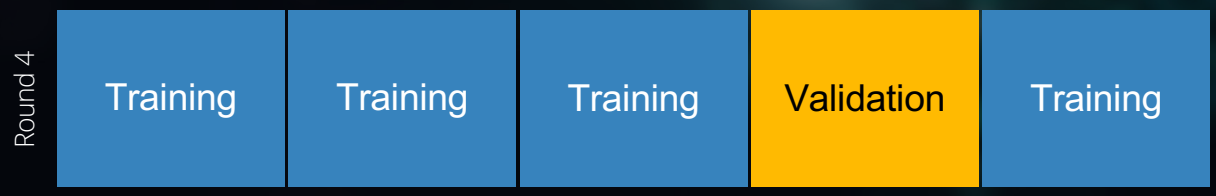


CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION (k=5)

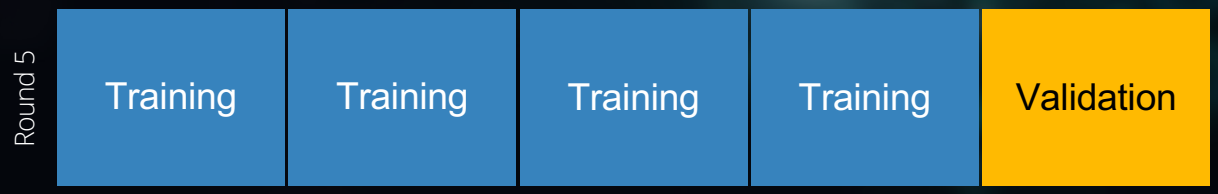
CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION (k=5)



CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION (k=5)



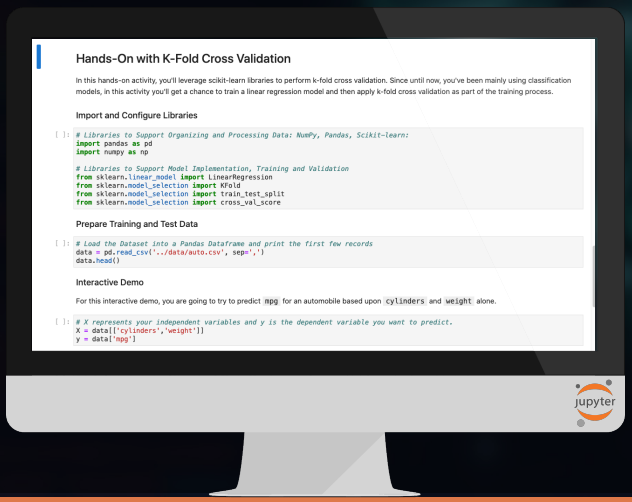
CROSS-VALIDATION TECHNIQUES

K-FOLD CROSS-VALIDATION (k=10)

Final Accuracy = Average (Round 1, Round 2, Round 3, ..., Round 10)

HANDS-ON: K-FOLD CROSS VALIDATION

TRY IT FOR YOURSELF



MORE CROSS-VALIDATION TECHNIQUES

SELF-PACED LEARNING

Leave One Out
Leave P-Out
Stratified
Repeated
Nested

TESTING SUPERVISED ML

AI FAIRNESS ASSESSMENT & RISK MITIGATION

TESTING THE TRAINING DATA

DATA SCIENCE AND ML MEETS TESTING

TESTING THE TRAINING DATA

POTENTIAL TRAINING DATA ISSUES

INSUFFICIENT DATA

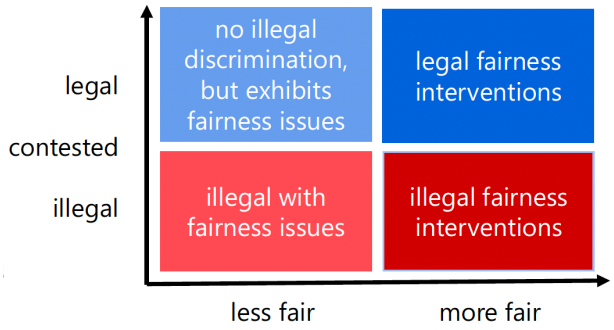
POOR QUALITY DATA

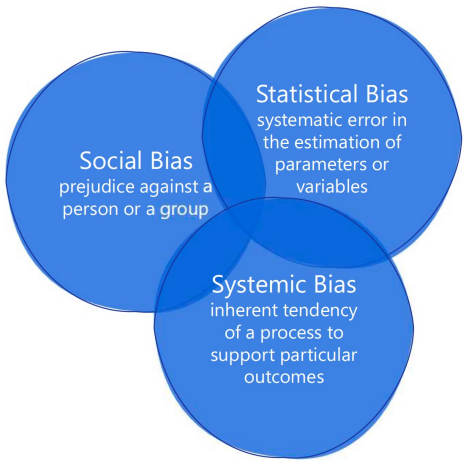
NON-REPRESENTATIVE DATA

TESTING AI FAIRNESS

TERMINOLOGY

FAIRNESS NOT LAW





IMPACTS & RISKS INSTEAD OF BIAS

TESTING AI FAIRNESS

TERMINOLOGY

TESTING AI FAIRNESS

GOAL

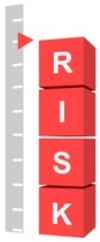
Are there **groups of people** who are **disproportionately**, **negatively impacted** by the system?

TESTING AI FAIRNESS

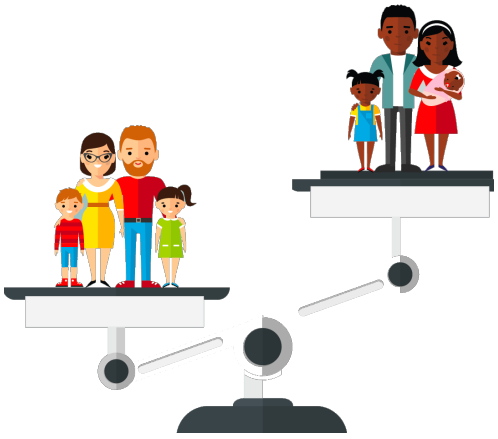
4 STEPS IN ASSESSING FAIRNESS

Identify Risks

Determine Impacted Groups



Quantify Risks

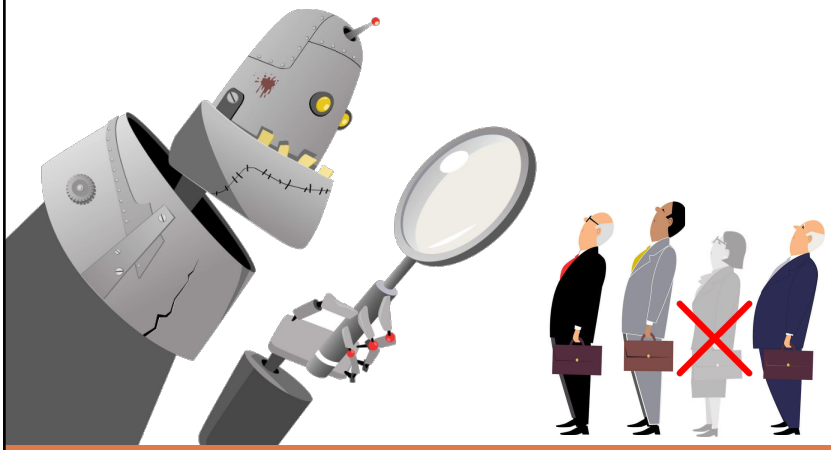


Compare Quantified Risks Across Groups

TYPES OF FAIRNESS RISKS

ALLOCATION RISK

Occurs when an AI system is used to assign opportunities or resources in ways that can have negative impacts on people's lives.



amazon

In 2014, a team of engineers at Amazon began working on a project to automate hiring at the company.

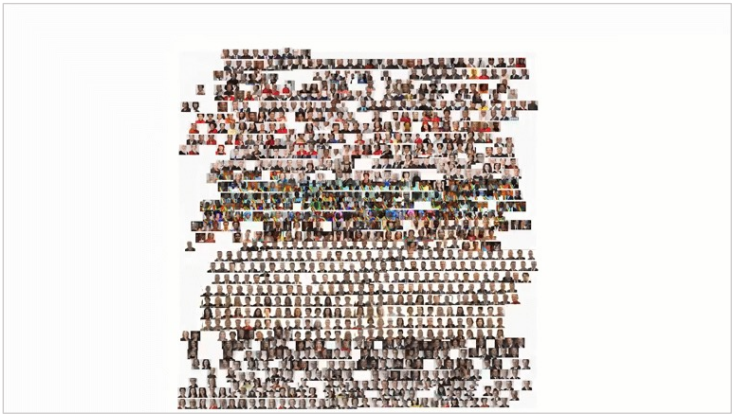
The goal was to build an ML-based algorithm that could review resumes and determine which applicants Amazon should bring on board.

The project was canned a year later, when it became clear that the tool systematically discriminated against women applying for technical jobs.

TYPES OF FAIRNESS RISKS

QUALITY OF SERVICE RISK

Occurs when an AI system does not work as well for one person as it does for another.



IBM Watson · Face++ 旷视 · Microsoft Cognitive Services

In 2018, researchers evaluated products from IBM, Face++, and Microsoft that all implement AI-based gender classification.

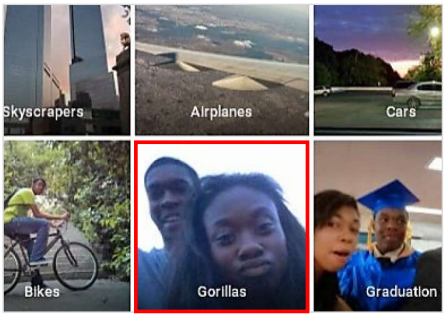
Although the highest rated product had an overall accuracy of 94%, all products performed better on males than females, and all products performed better on lighter-skinned subjects than darker-skinned.

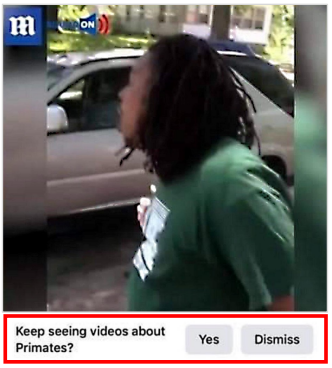
An analysis of intersectional groups revealed that all products performed worst on darker-skinned females.

TYPES OF FAIRNESS RISKS

REPRESENTATION RISK

Occurs when an AI system performs tasks that include stereotyping, denigration, and any form of over- or under-representation.





Google Photos · FACEBOOK AI

In 2015, an image recognition algorithm in Google's Photos app mistakenly tagged a black couple as being "gorillas". Nearly three years later, the algorithm was "fixed" by blocking it from identifying gorillas altogether.

In 2020, Facebook's AI asked users watching a video featuring Black men if they wanted to see more "videos about primates".

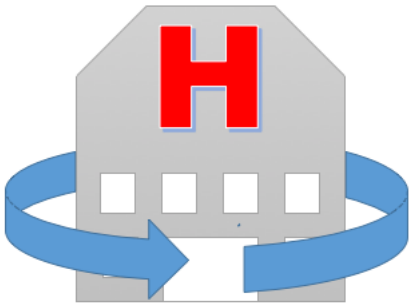
Both instances of mislabeling are harmful because such labels have a history of being used to demean Black people.

HANDS-ON: CONTEXT FOR ACTIVITIES

SCENARIO

Consider an automated system for **recommending patients for high-risk care management programs**.

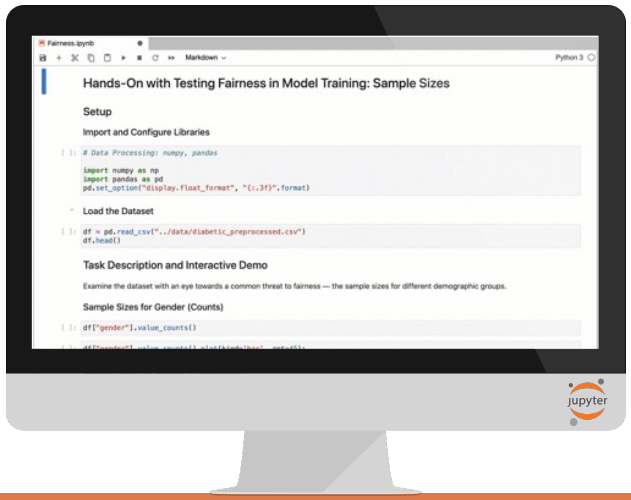
- Such a system may be implemented as an ML classifier that predicts whether a patient should be suggested for enrollment in the program.
- Historical information on patients who had to be readmitted to the hospital within 30 days after release, can be used as an indicator that a patient should be recommended to the program.



Readmit 30 days ≡ Care Program Recommendation

HANDS-ON: FAIRNESS IN TRAINING DATA

SAMPLE SIZES INTERACTIVE DEMO & EXERCISE



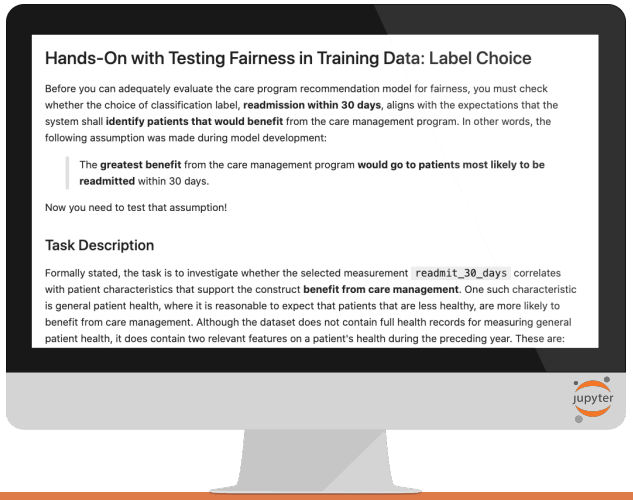
python™

pandas

NumPy

HANDS-ON: FAIRNESS IN TRAINING DATA

LABEL CHOICE INTERACTIVE DEMO & EXERCISE



python™

pandas

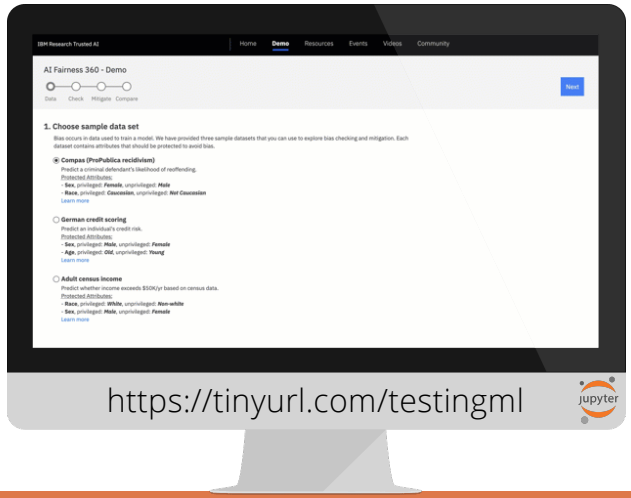
NumPy

seaborn

HANDS-ON: TESTING AI FAIRNESS

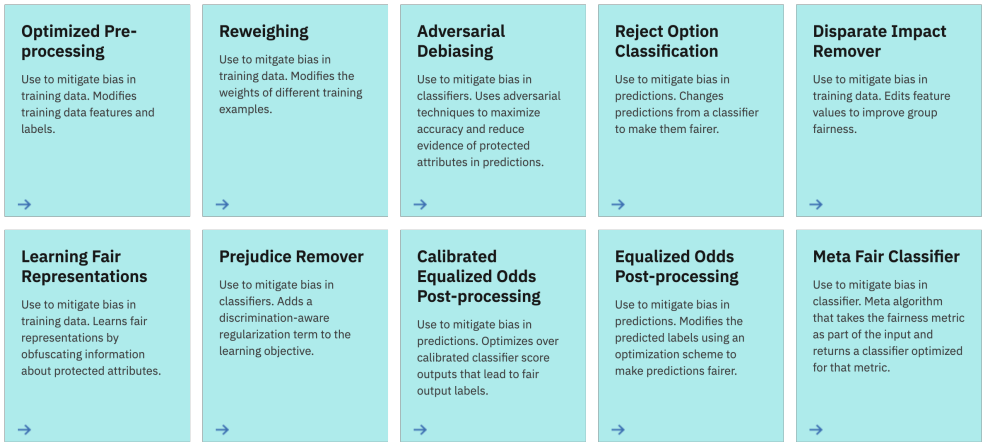
MITIGATING FAIRNESS RISKS WITH AI FAIRNESS 360

AI Fairness 360



TESTING AI FOR FAIRNESS

RISK MITIGATION DISCUSSION

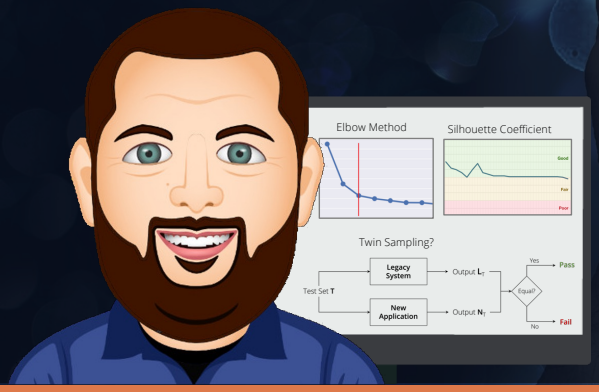


TESTING UNSUPERVISED ML

CLUSTERING VALIDATION

TESTING UNSUPERVISED ML

DATA SCIENCE, ML, TESTING CROSS-OVER



95

CLUSTER VALIDATION

INTRODUCTION

Affinity Propagation

Agglomerative Clustering

BIRCH

DBSCAN

K-Means

Mini-Batch K-Means

Mean Shift

OPTICS

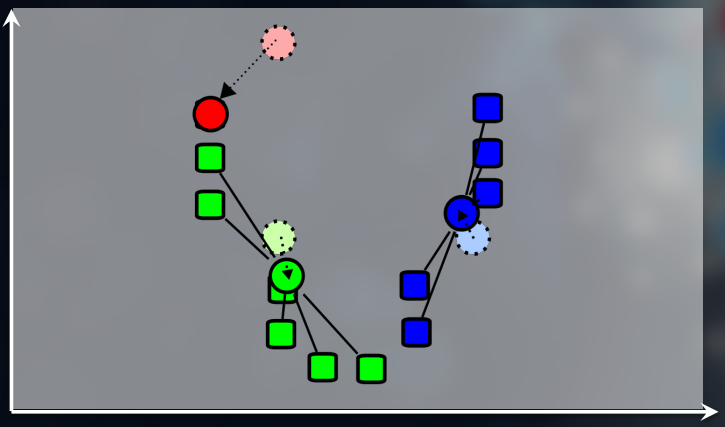
Spectral Clustering

Mixture of Gaussians

96

K-MEANS CLUSTERING

HOW IT WORKS



K-Means Algorithm

1. Randomly place K centroids for the initial clusters.
2. Assign each data point to their nearest centroid.
3. Update centroid locations based on the locations of the data points.
4. Repeat 2. and 3. until points don't move between clusters and centroids stabilize.

CLUSTERING: GOALS & CHALLENGES

Goal

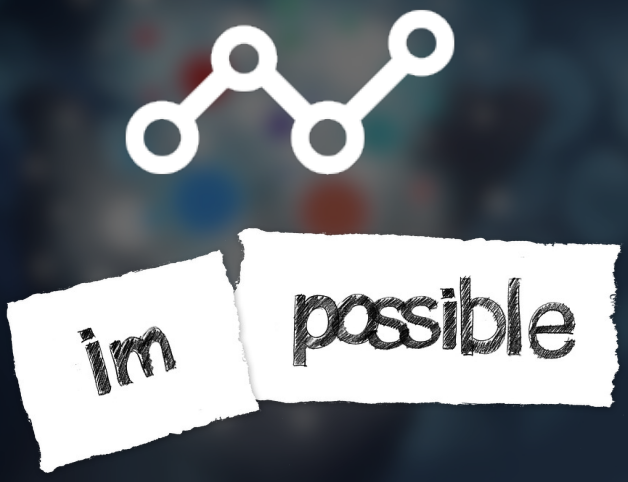
- Partition unlabeled data

Properties of Good Clustering

- Scale Invariance
- Consistency
- Richness or Wealth

Challenges

- Cannot be precisely defined
- Impossibility Theorem



VALIDATION GOALS AND CHALLENGES

Optimal Number of Clusters

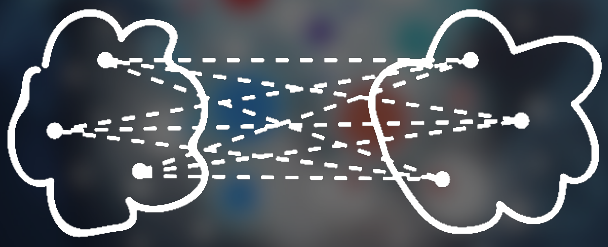
Comparing Sets of Clusters

Evaluating Results
(with or without oracle)

INTERNAL VALIDATION

COHESION AND SEPARATION

Cohesion (C_k)



Separation (C_j, C_k)

INTERNAL VALIDATION

COMBINED METRICS

- Silhouette Coefficient
- Calisnki-Harabasz Coefficient
- Dunn Index
- Xie-Beni Score
- Hartigan Index
- and more...

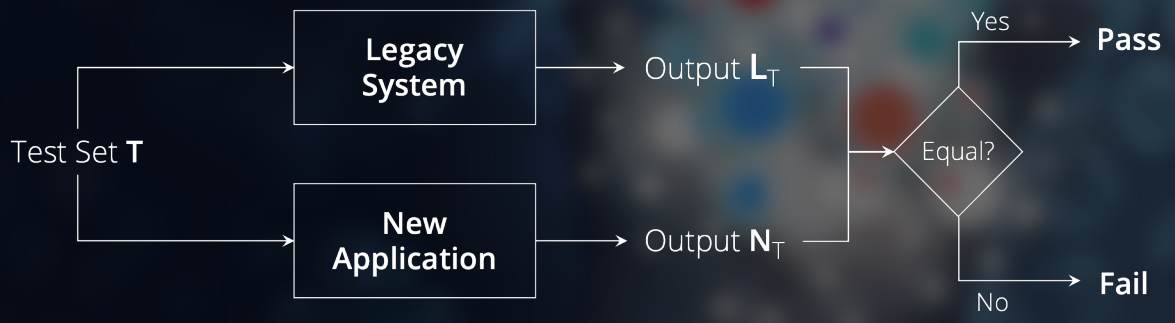
EXTERNAL VALIDATION

SUBJECT MATTER EXPERTS DEFINE TRUE LABELS

		Predicted Class	
		Yes	No
Actual Class	Yes	TRUE POSITIVES	FALSE NEGATIVES
	No	FALSE POSITIVES	TRUE NEGATIVES

CROSS-OVER VALIDATION

PARALLEL VALIDATION IN SOFTWARE TESTING



CROSS-OVER VALIDATION

TWIN VALIDATION IN TESTING ML

1. Create a twin-sample of the training data
2. Perform unsupervised learning on twin-sample
3. Import results for twin-sample from training set
4. Calculate similarity between the two sets of results


HANDS-ON: CLUSTER VALIDATION
Internal Validation for Clustering
+ 7 cells hidden
Hands-On with Clustering Performance: Silhouette Coefficient
+ 18 cells hidden
External Validation for Clustering
+ 5 cells hidden
Cross-Over Validation Techniques
+ 15 cells hidden
jupyter
105 | QUALITY ENGINEERING FOR AI AND MACHINE LEARNING SYSTEMS: TESTING UNSUPERVISED ML

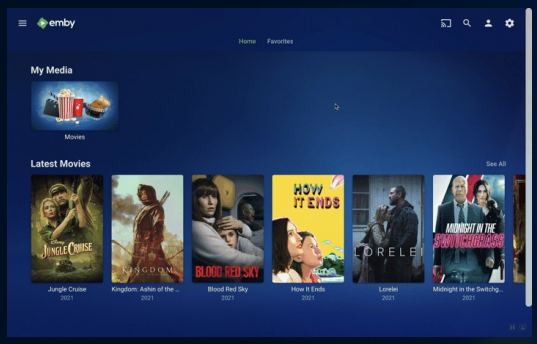
105


USER TESTING
TESTING ADAPTIVE ML
106 | EXPLORING THE INTERSECTION OF AI FOR TESTING: TESTING AI AND ML

106

SOFTWARE HAS BECOME MORE DYNAMIC

LEARN AND ADAPT IN REAL-TIME

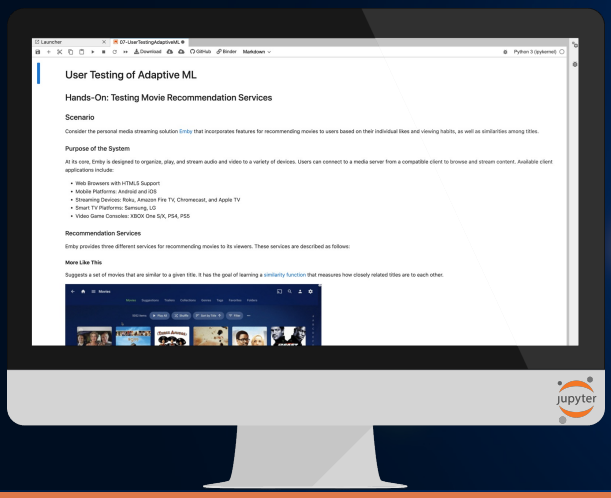


Recommendation Systems

Self-Driving Cars

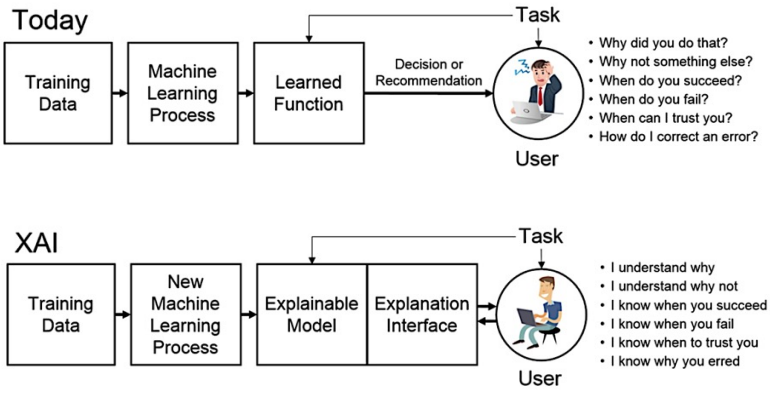
HANDS-ON: USER TESTING AI/ML

TESTING MOVIE RECOMMENDATIONS



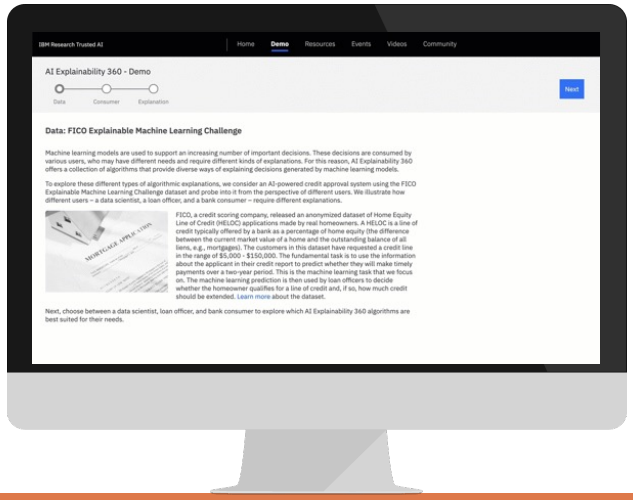
EXPLAINABLE AI: A DARPA INITIATIVE

TOWARDS TRUSTED AI AND MACHINE LEARNING



EXPLAINABLE AI: AI EXPLAINABILITY 360

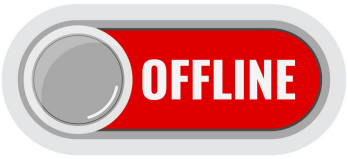
FICO EXPLAINABLE ML CHALLENGE



AI Explainability 360

USER TESTING AI/ML: KEY CONSIDERATIONS

SYSTEMS THAT LEARN AT RUNTIME, MUST BE TESTED IN PRODUCTION



Testing Algorithm in a Lab
Sandbox Environment
De Facto Standard: Split Validation
Academic Measures

ISOLATED AND INADEQUATE

Testing System in the Wild
Production Environment
De Facto Standard: A/B Testing
Realistic Measures

INTEGRATED AND ADEQUATE


FROM NOW WE TEST
IN PRODUCTION!

113


TIP ≠
INSUFFICIENT
PRE-PRODUCTION
TESTING

114

TIP CHALLENGES

Breaking Production Functionality

Loss or Corruption of Production Data

Interruption of Regular Operations

Narrowed Test Window

System Performance Degradation

Complicates Security Measures

Non-Compliance with Standards

TIP BENEFITS

Tests Mirror the Real World

Quick Feedback on Impact of Changes

Preventing Bad Deployments

Catching Issues Before Users Notice Them

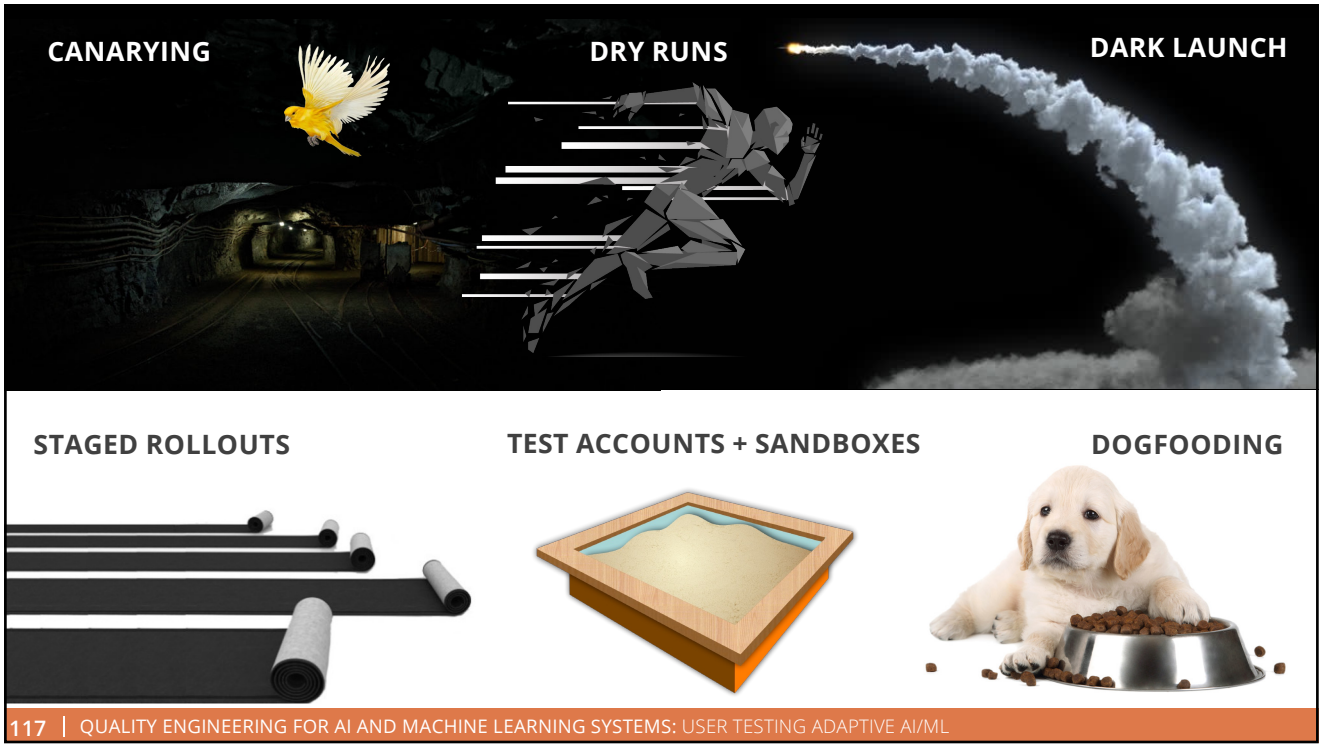
Some Bugs are Easier to Reproduce

Ensures Software is Production Ready

Helps in Designing Resilient Systems

APPROACHES TO

TIP


CANARYING
DRY RUNS
DARK LAUNCH
STAGED ROLLOUTS
TEST ACCOUNTS + SANDBOXES
DOGFOODING
117 | QUALITY ENGINEERING FOR AI AND MACHINE LEARNING SYSTEMS: USER TESTING ADAPTIVE AI/ML

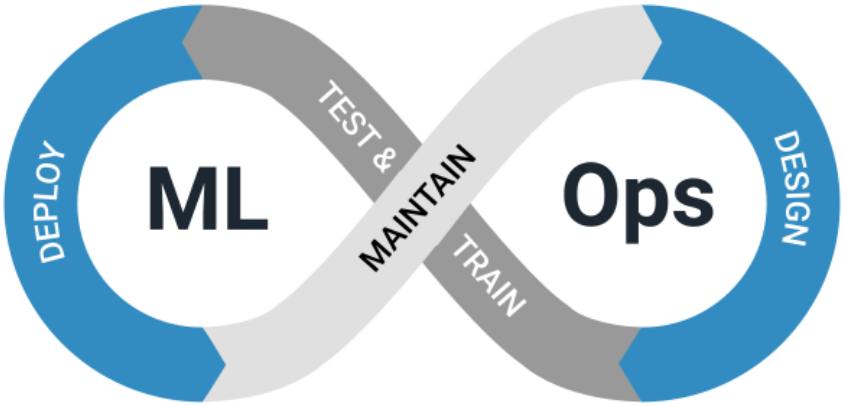
117


NORA JONES
Testing in Production
The Netflix Way
Testing in Production
The Netflix Way
-@nora_js-
NETFLIX
ChAP
118 | QUALITY ENGINEERING FOR AI AND MACHINE LEARNING SYSTEMS: USER TESTING ADAPTIVE AI/ML

118

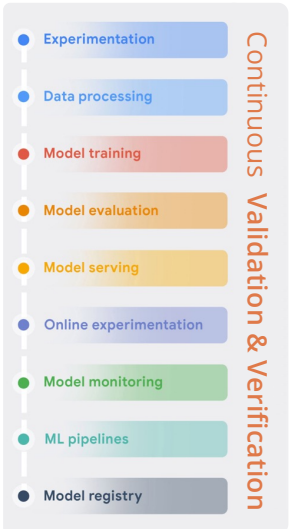
KEY TAKEAWAY

CONTINUOUS LEARNING REQUIRES CONTINUOUS TESTING



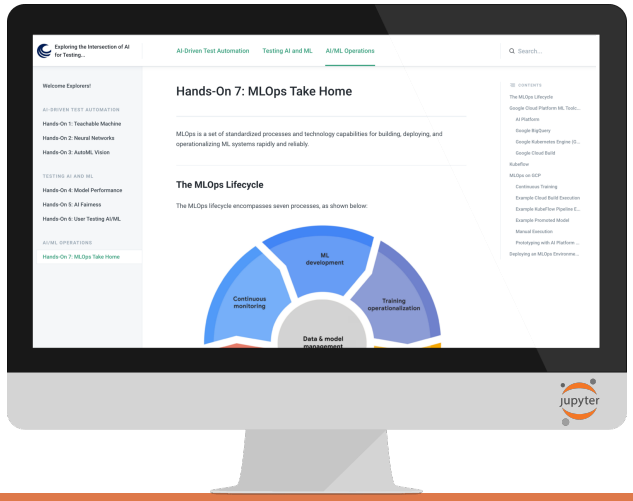
AI/ML OPERATIONS

ML OPERATIONS: LIFECYCLE & ACTIVITIES

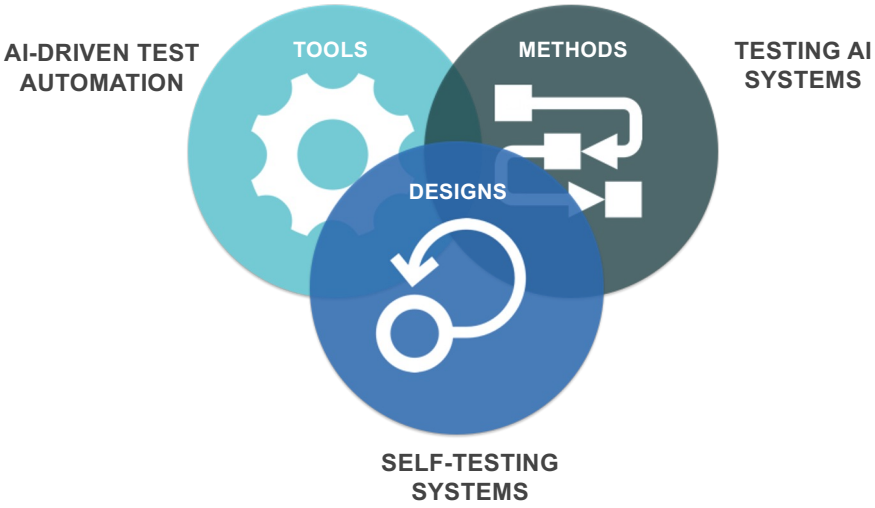


HANDS-ON: ML OPS TAKE HOME

GOOGLE AI PLATFORM

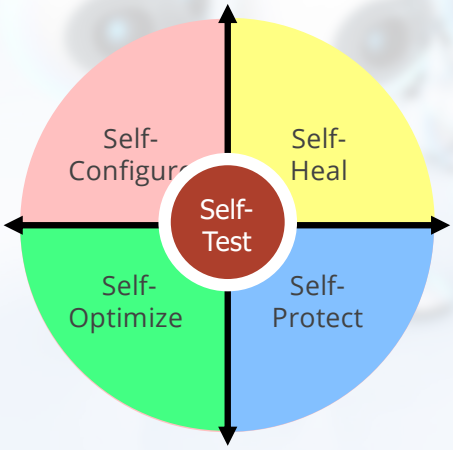


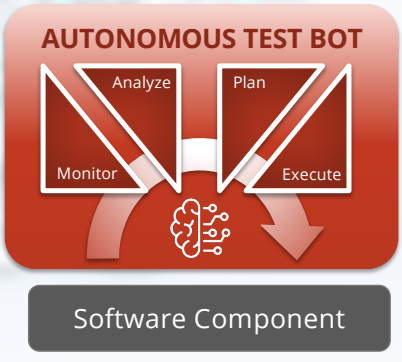
WRAP UP: AI FOR SOFTWARE TESTING



125

AUTONOMIC SELF-TESTING





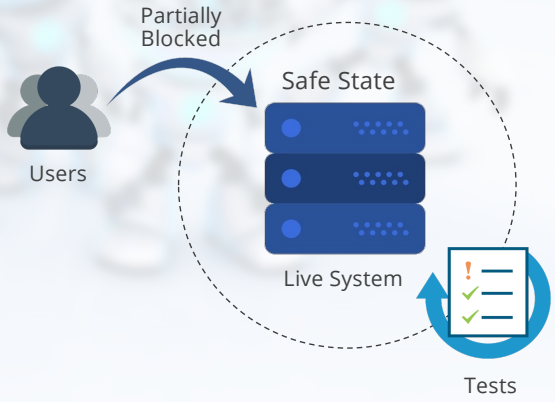
A Self-Testing Approach for Autonomic Software, Tariq M. King.
ProQuest Electronic Theses and Dissertations (2009).

126

SELF-TESTING APPROACHES

SAFE ADAPTATION WITH VALIDATION

1. Bring the system into a **safe state**, where some functions are partially blocked.
2. Implement changes and perform runtime **testing in-place** on the live system.
3. Once **testing completes**, the system returns to its mode of **full operation**.



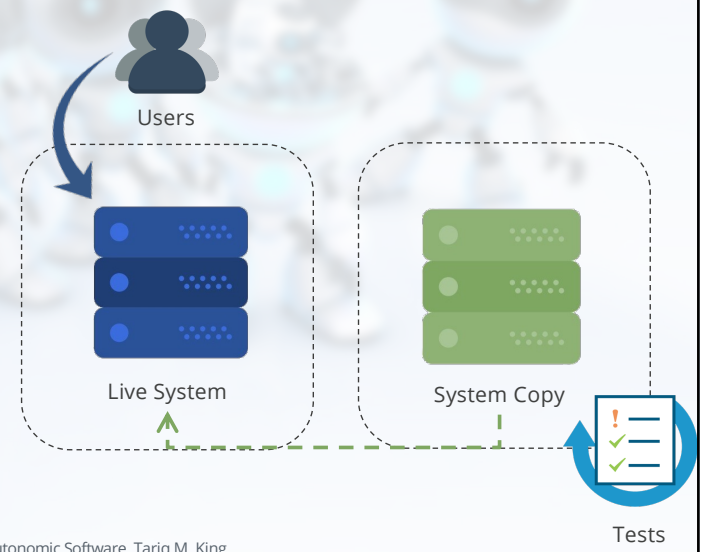
A Self-Testing Approach for Autonomic Software, Tariq M. King.
ProQuest Electronic Theses and Dissertations (2009).

127

SELF-TESTING APPROACHES

REPLICATION WITH VALIDATION

1. Create or maintain a **copy** of the system under test.
2. Implement any changes and perform runtime **testing on the copy** of the system.
3. Use the results of testing on the copy to decide whether to **adapt the live system**.

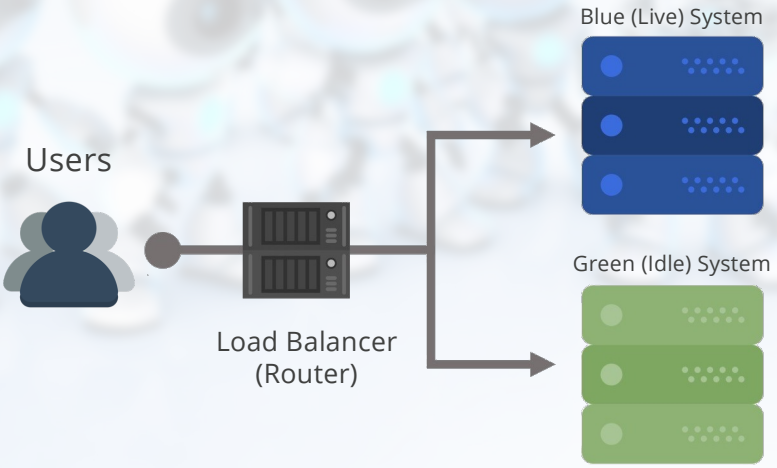


A Self-Testing Approach for Autonomic Software, Tariq M. King.
ProQuest Electronic Theses and Dissertations (2009).

128

BLUE-GREEN DEPLOYMENTS

ALSO KNOWN AS RED/BLACK DEPLOYMENTS



129

DEVOPS SUPPORTS SELF-TESTING

CI/CD REQUIRES CONTINUOUS TESTING

NEW CODE | EVERY BUILD | EVERY DEPLOY | IN PRODUCTION

SHIFT LEFT TESTING | SHIFT RIGHT TESTING

130

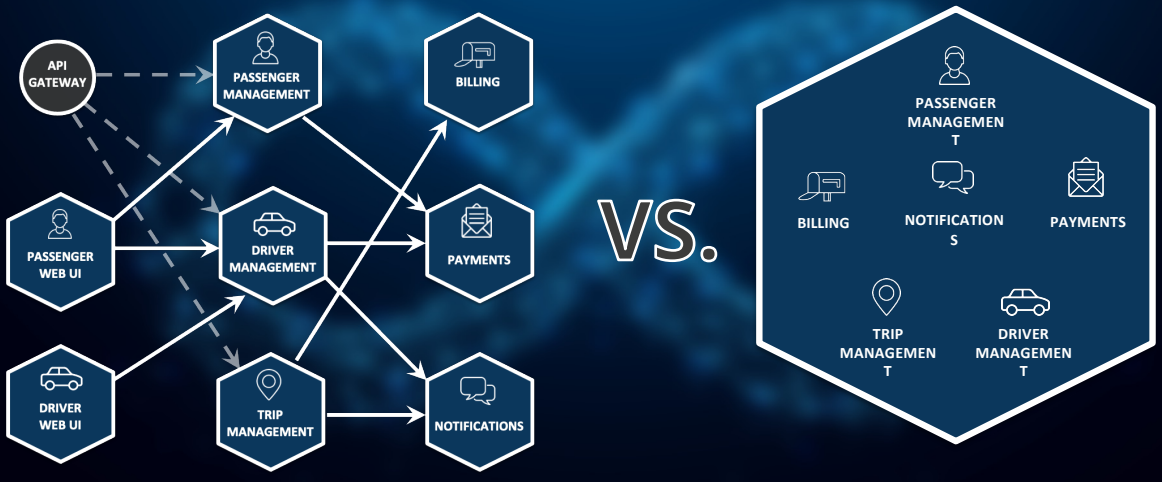
DEVOPS SUPPORTS SELF-TESTING

THE NEXT SHIFT IS INWARD

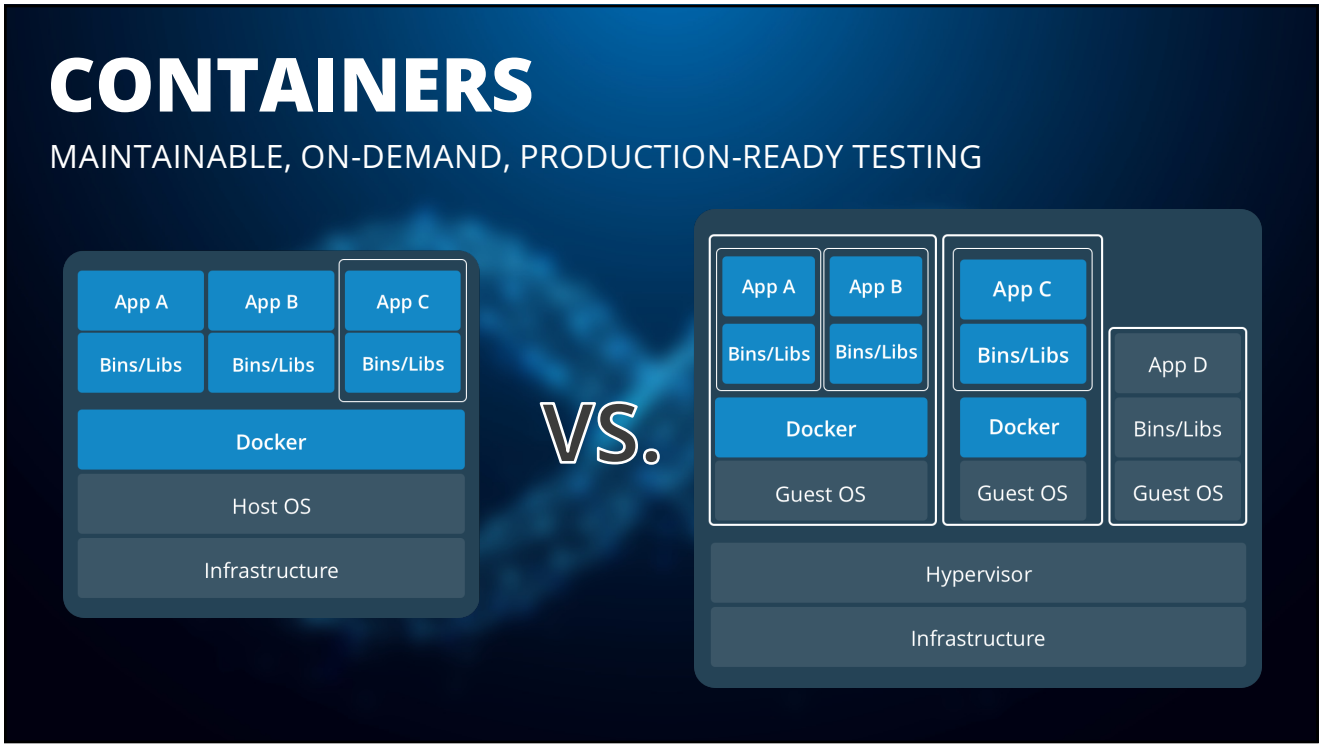
131

MICROSERVICES ARCHITECTURES

RESILIENT, CLUSTERED, MONITORED, SCALABLE TESTING



132


CONTAINERS
MAINTAINABLE, ON-DEMAND, PRODUCTION-READY TESTING
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Docker
Host OS
Infrastructure
VS.
App A
App B
Bins/Libs
Bins/Libs
Docker
Guest OS
App C
Bins/Libs
Docker
Guest OS
App D
Bins/Libs
Guest OS
Hypervisor
Infrastructure

133


AI SUPPORTS SELF-TESTING

134

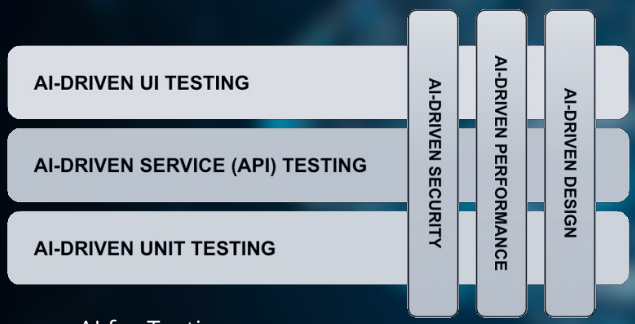
AI SUPPORTS SELF-TESTING

AUTOMATIC TEST GENERATION

135

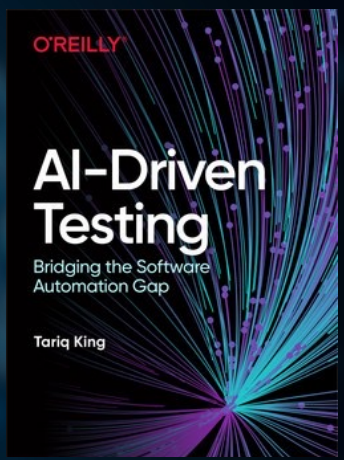
AI SUPPORTS SELF-TESTING

BRIDGING THE AUTOMATION GAP



AI for Testing
Core Concerns

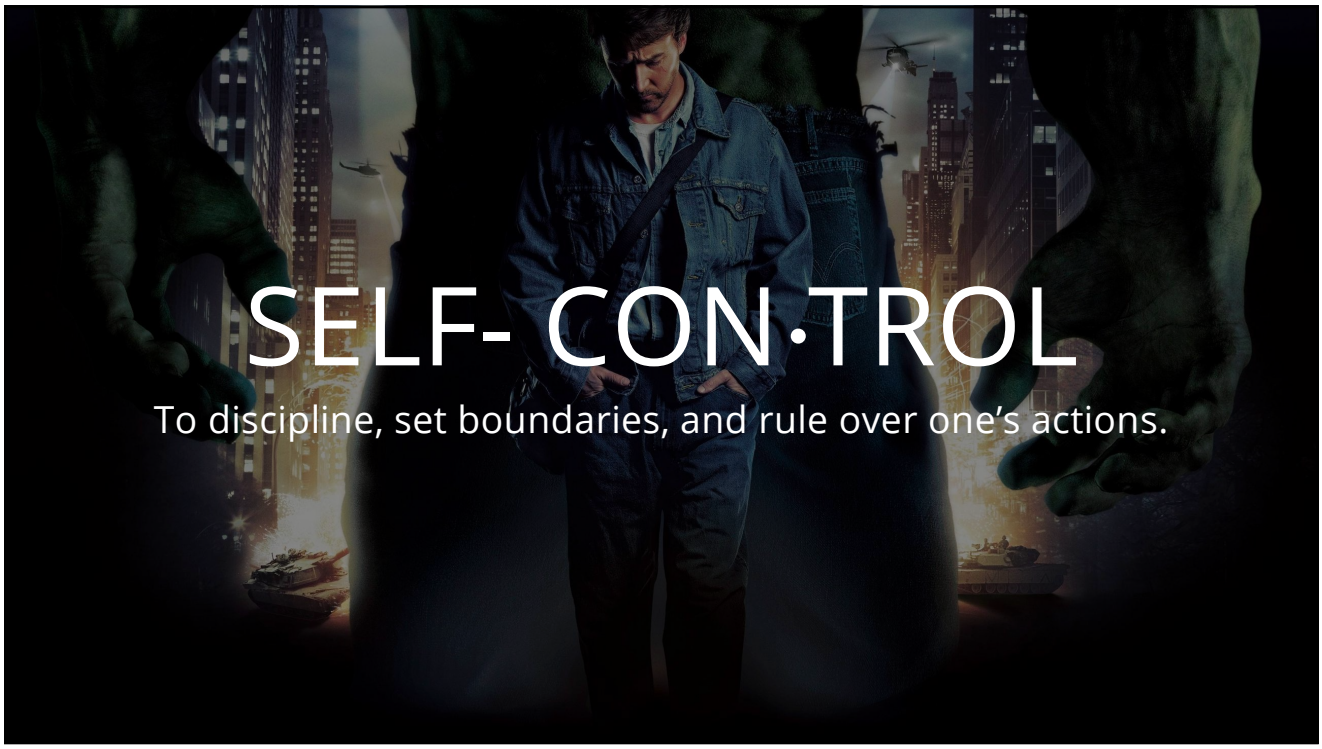
AI for Testing
Cross-Cutting Concerns



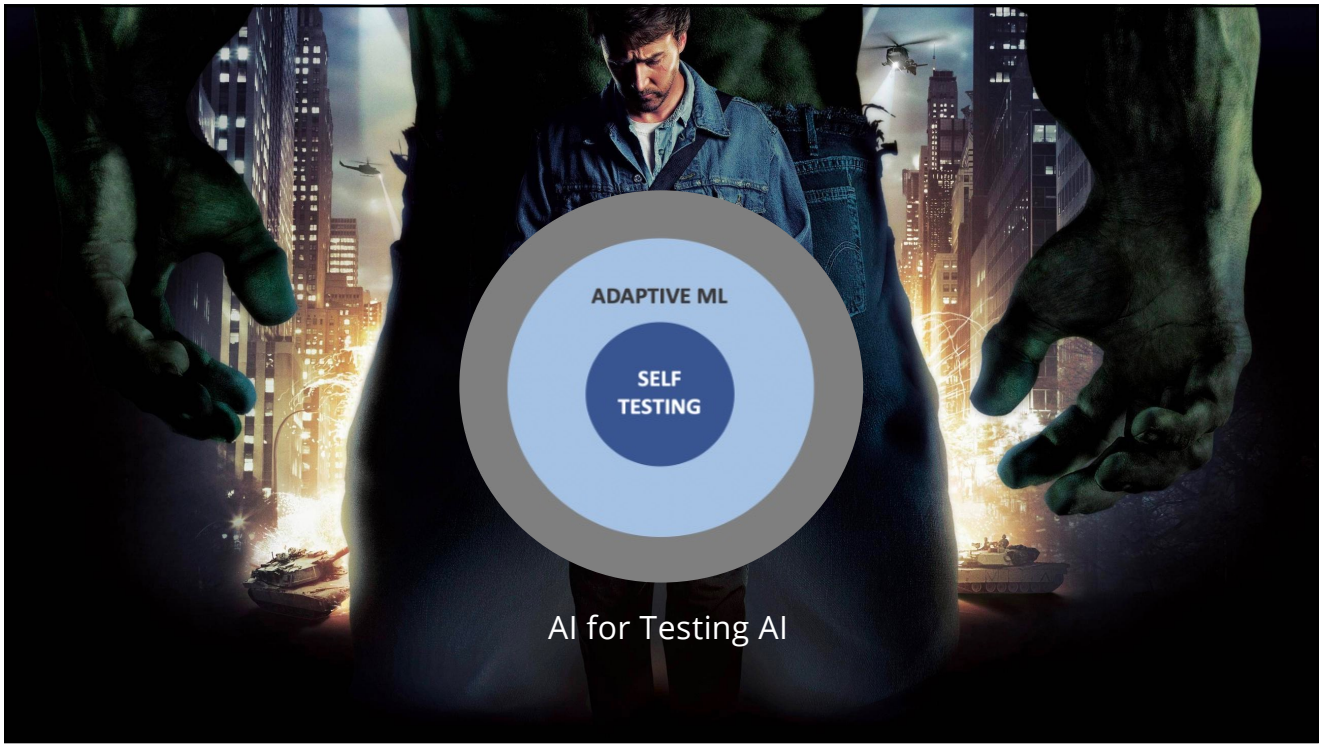
136


AI NEEDS SELF-TESTING

137


SELF- CON·TROL
To discipline, set boundaries, and rule over one's actions.

138



139

QUESTIONS?

THANK YOU

king.tariq@gmail.com

linkedin.com/in/tariqking

@tariq_king

140