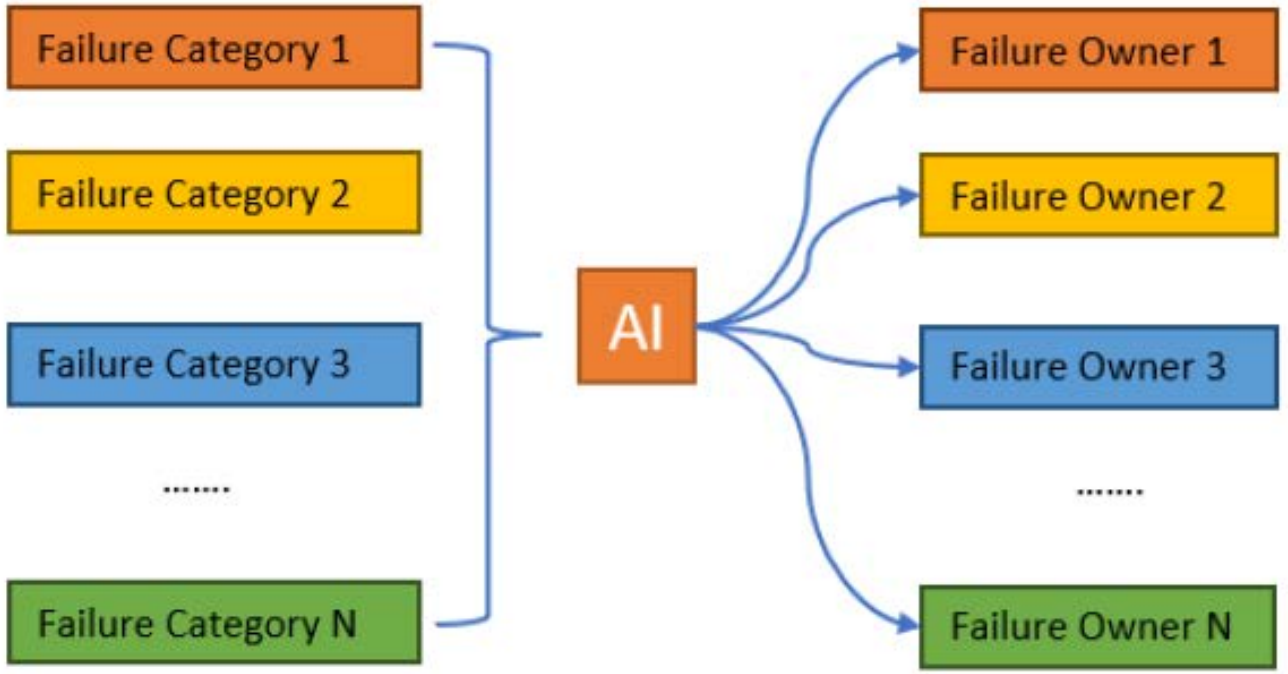
SSTA - Smart Signature Triage and Assignment

Automating Regression Triage and Reporting in Design Verification using AI-Based Random Forest Models

Lingkai Shi — Sr Manager @AMD MSIP team
Rohit Mathur — Director @AMD MSIP team

Introduction

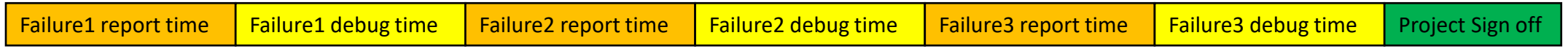
- Every IP/block generates 1000's of failures every week from Hardware Verification Regression Testing across testbenches and views
- Unique failure modes (signatures) need to be assigned to owner to root cause and fix the issue.



Objectives

- Improve accuracy of signature assignment with machine learning
- Eliminate engineer-time spent on repetitive and tedious signature assignment task.
- Assign the failure right after the failure show up, enabling faster bug fixing eventually resulting in faster design qualify sign off

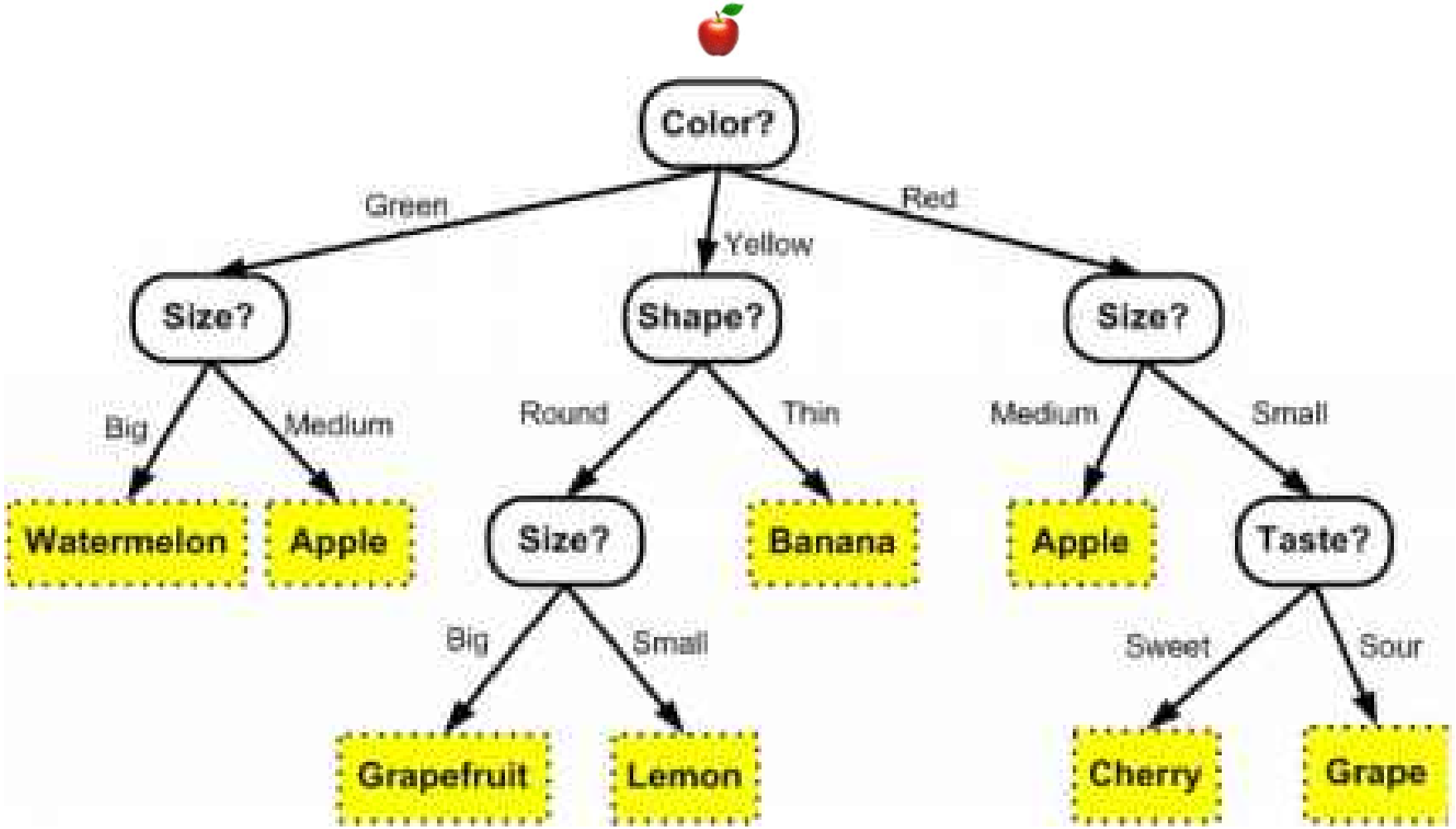
Old system



New system

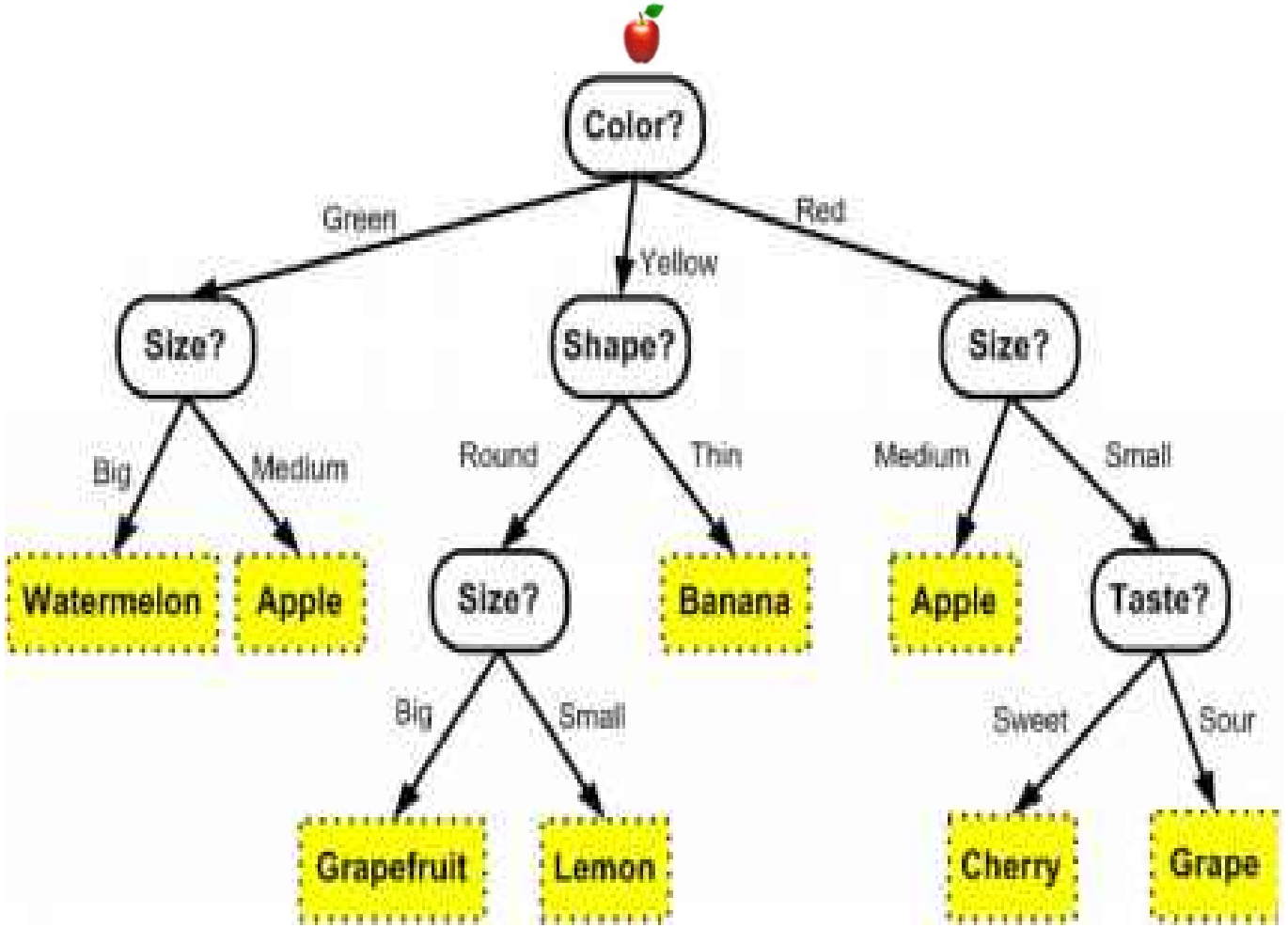
Introduction to Random Forest Model

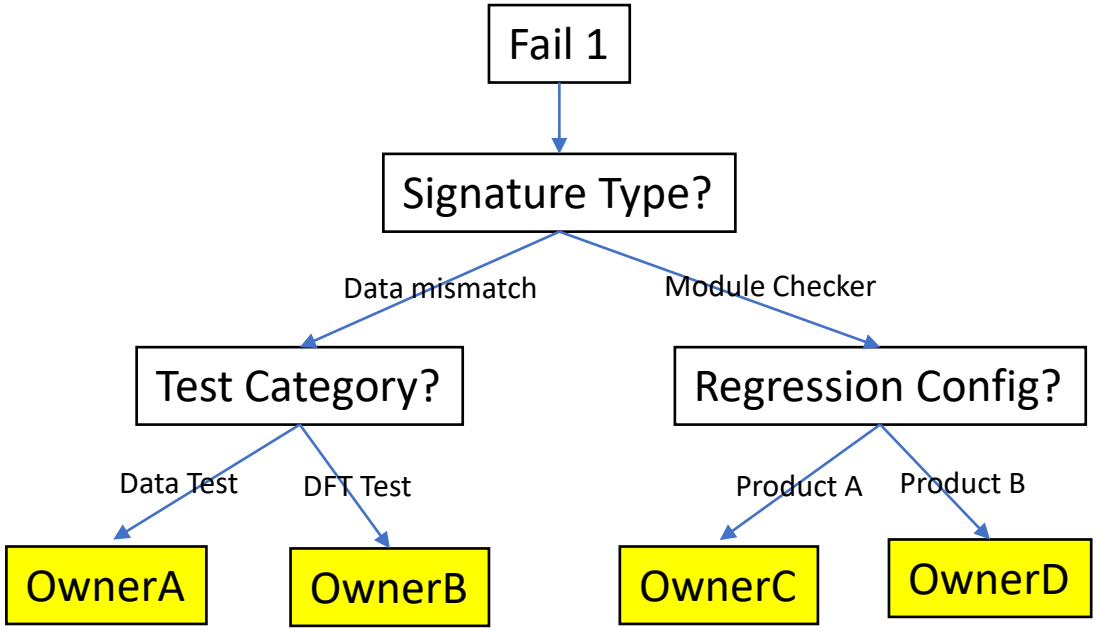
Decision Tree



Introduction to Random Forest Model

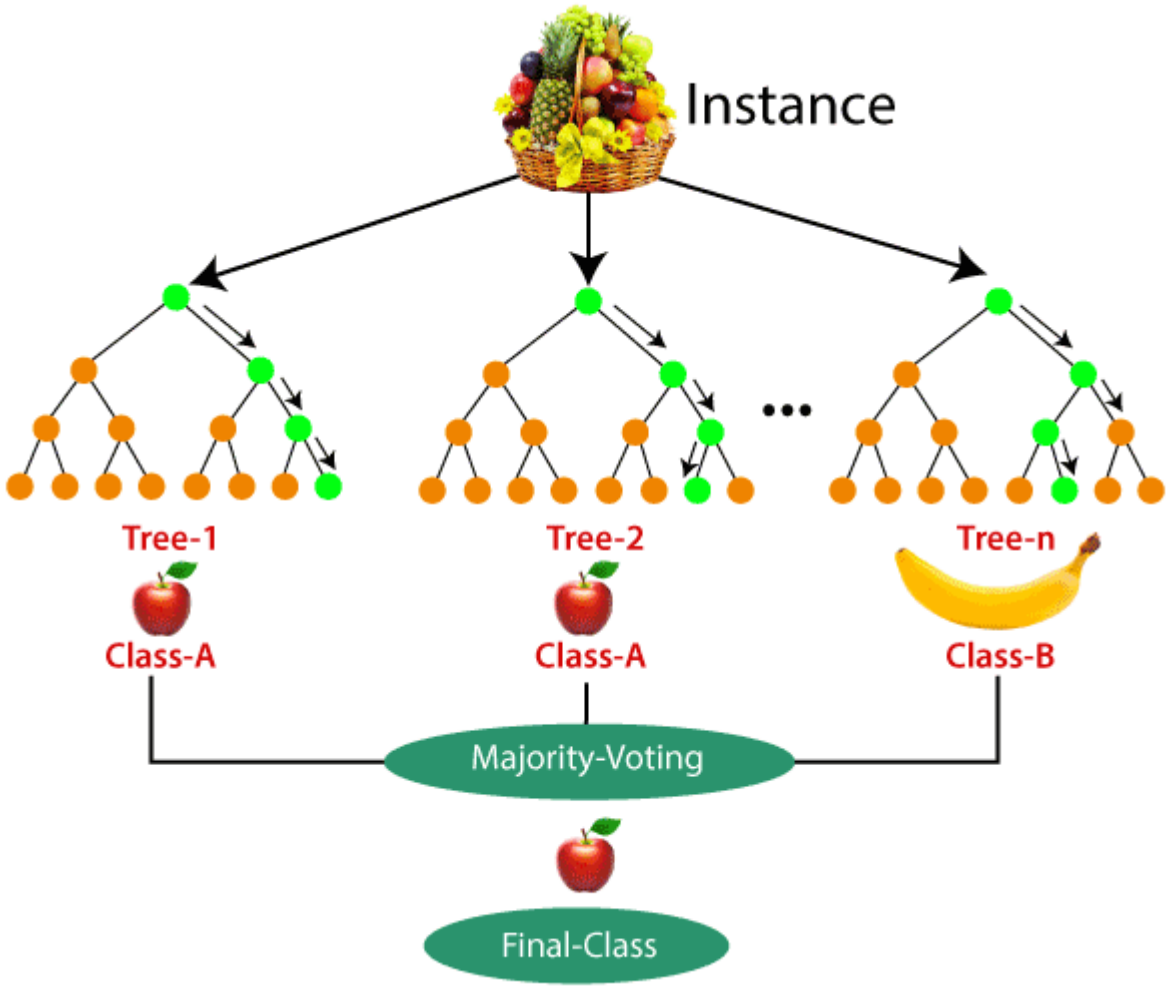
Decision Tree





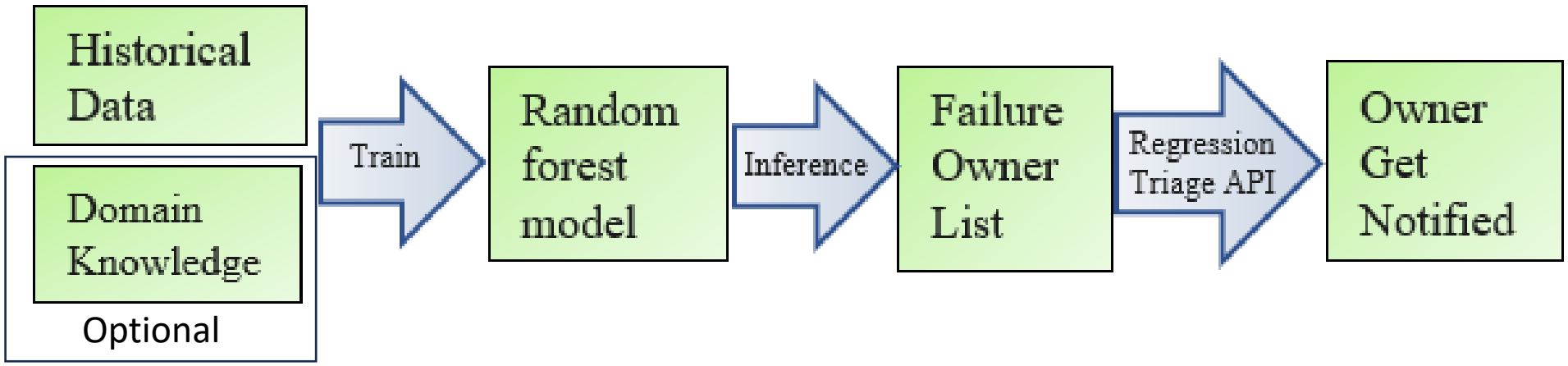
Introduction to Random Forest Model

Random Forest Example



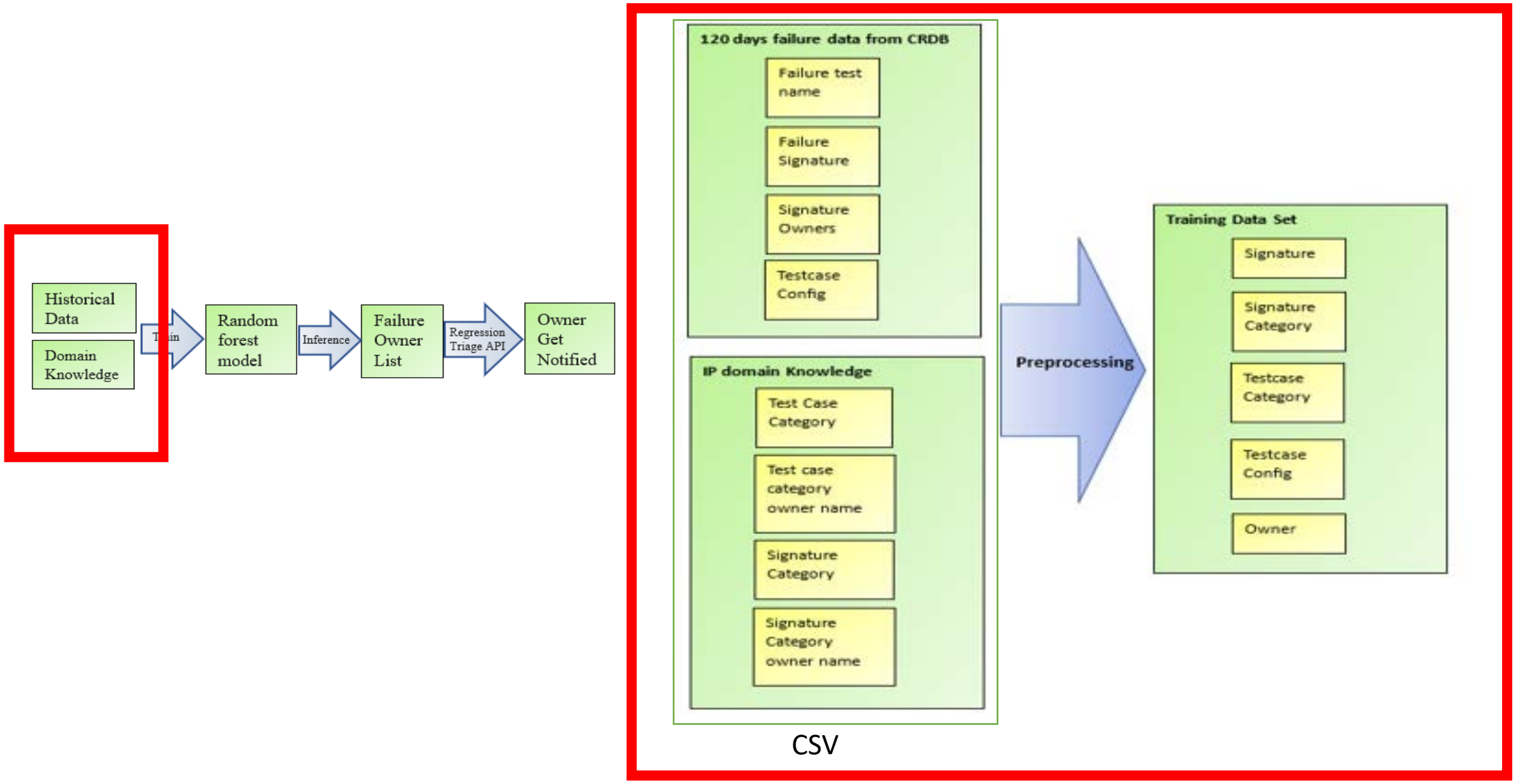
- Robustness
  - Overfit
  - Training Data Noise
- Versatility
  - Easy to remove or add property to existing model
  - Flexible on choice of property

SSTA Flow Diagram



Domain Knowledge is an optional input because it only get used to improve accuracy for new failure that never seen in the Historical data before. The new signature will become historical data afterwards.

Triage System Introduction

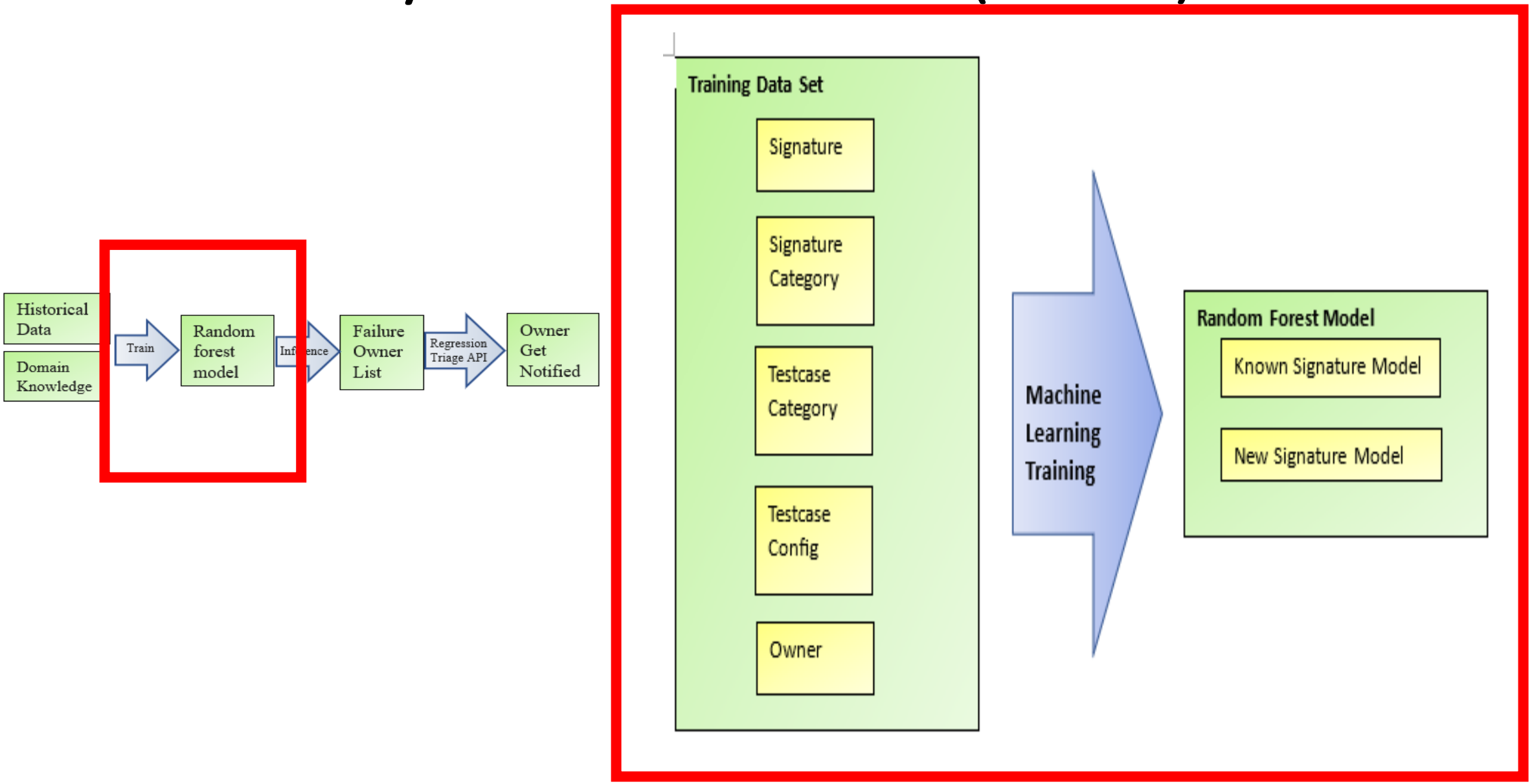


SSTA System Introduction (contd.)

Preprocessing

- Convert String categorical variables into numerical variables
- Filtering erroneous input
- Uniform the input format

SSTA System Introduction (contd.)

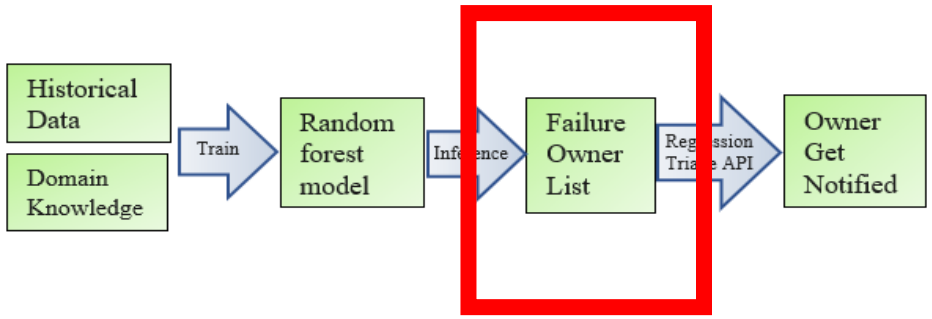


SSTA System Introduction (contd.)

Training

- CPU is powerful enough. No need for GPU. Training itself runs in 1 minute. (>30000 training set)
- Utilize Python Built-in Random forest model
- Dump the model into files to run anywhere and anytime.

SSTA System Introduction (contd.)



CSV format

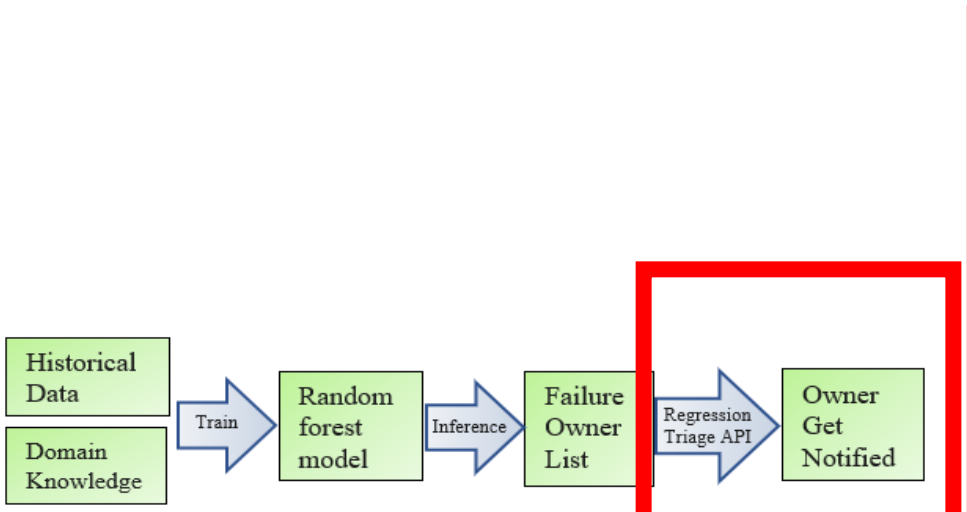
B	C	D	E	F	G	H	I
Signature	Signature	SIM ID	Testcase	Testcase_	Environment	Original Owner	Prediction Owner
[NOA] Null object access umc_rec_checker_gddr7.sv	rec	12072-0-79611	sanity::bank_state_self_che	generic_t	umc13_0__umc_	nan	
[NOA] Null object access umc_rec_checker_gddr7.sv	rec	12072-0-79904	beq_dfi_tb::dfi_beq_SramL	beq	umc13_0__umc_	nan	
[NOA] Null object access umc_rec_checker_gddr7.sv	rec	12072-0-80041	beq_dfi_tb::dfi_beq_SramL	beq	umc13_0__umc_	nan	
tb.umc_w_phy.umc_phy_if_inst_*_*.UMC_headsup	generic_s	12030-0-43592	dcqarb_real_phy_tb::real_t	generic_t	umc13_0__umc_	nan	
tb.umc_w_phy.umc_phy_if_inst_*_*.UMC_headsup	generic_s	12030-0-43397	mp_real_phy_tb::umc_mp_	generic_t	umc13_0__umc_	nan	
tb.umc_w_phy.umc_phy_if_inst_*_*.UMC_headsup	generic_s	12030-0-43942	mp_real_phy_tb::umc_mp_	generic_t	umc13_0__umc_	nan	
tb.umc_w_phy.umc_phy_if_inst_*_*.UMC_headsup	generic_s	12030-0-44134	mp_real_phy_tb::umc_mp_	generic_t	umc13_0__umc_	nan	
tb.umc_w_phy.umc_SmnTarg*_vif.axi_errs_rdata_x	generic_s	12072-0-79582	sanity::csr_hw_reset	generic_t	umc13_0__umc_	nan	
tb.umc_w_phy.umc_SmnTarg*_vif.axi_errs_rdata_x	generic_s	12072-0-79998	sanity::csr_hw_reset	generic_t	umc13_0__umc_	nan	
tb.umc_w_phy.umc_SmnTarg*_vif.axi_errs_rdata_x	generic_s	12072-0-80098	sanity::csr_hw_reset	generic_t	umc13_0__umc_	nan	
tb.umc_w_phy.umc_w_phy*.umc*.gfxumcctrl.gfxun	bist	12072-0-79923	bist_dfi_tb::bist_retention	bist	umc13_0__umc_	nan	

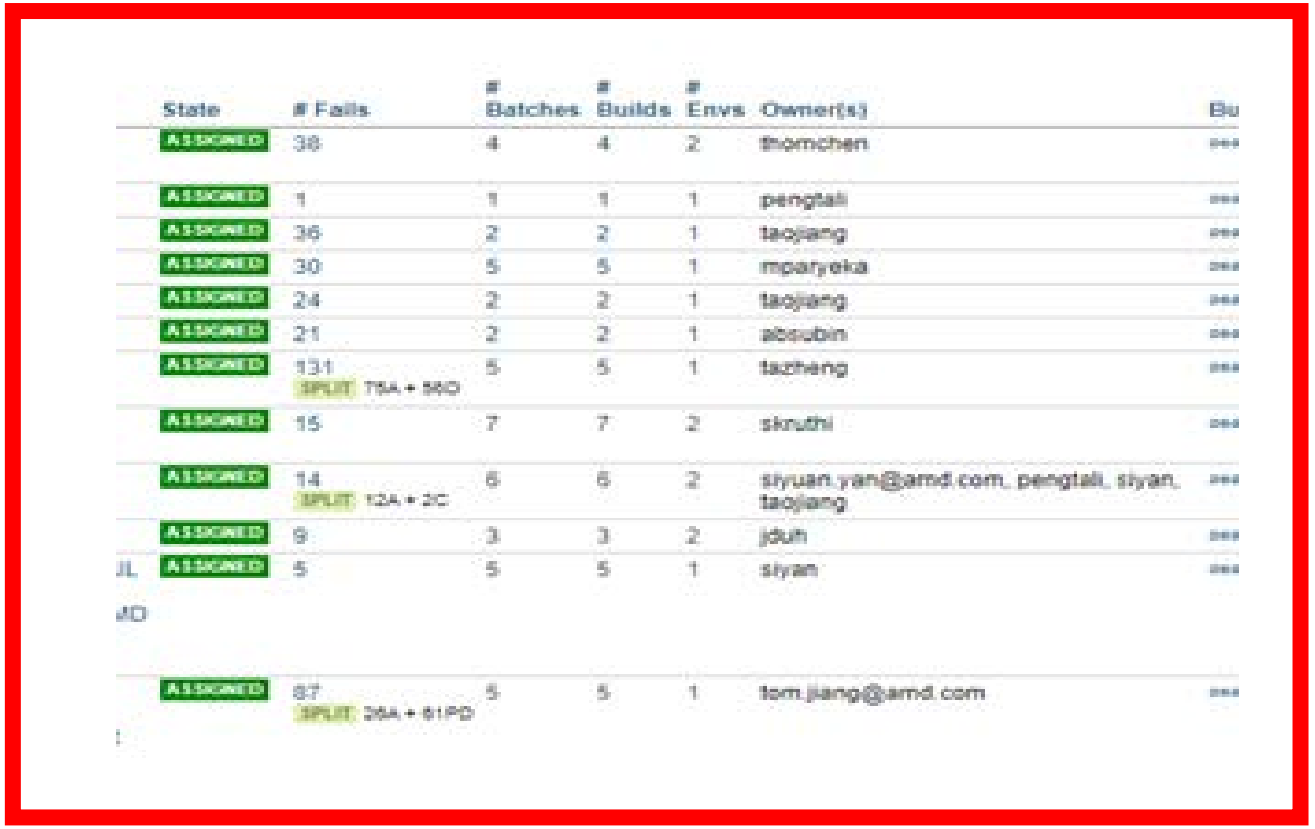
SSTA System Introduction (contd.)

Inferencing

- Running on a CPU. No GPU is required.
- Take the pending assignments list in CSV format.
- Output in CSV format containing all required info including predicted failure owner.

SSTA System Introduction (contd.)





State	# Fails	# Batches	# Builds	# Envs	Owner(s)	Bu
ASSIGNED	38	4	4	2	thomchen	
ASSIGNED	1	1	1	1	pengtali	
ASSIGNED	36	2	2	1	taojiang	
ASSIGNED	30	5	5	1	mparyeka	
ASSIGNED	24	2	2	1	taojiang	
ASSIGNED	21	2	2	1	absubin	
ASSIGNED	131 SPLIT: 75A + 56D	5	5	1	tazheng	
ASSIGNED	15	7	7	2	skruthi	
ASSIGNED	14 SPLIT: 12A + 2C	6	6	2	siyuan.yan@amd.com, pengtali, siyan, taojiang	
ASSIGNED	9	3	3	2	jduh	
ASSIGNED	5	5	5	1	siyan	
ASSIGNED	87 SPLIT: 26A + 61PD	5	5	1	tom.jiang@amd.com	

SSTA System Introduction (contd.)

Reporting

- Use Regression System API to report signature.
- The automation of triage system is transparent to the team. Deployment cost is minimal.

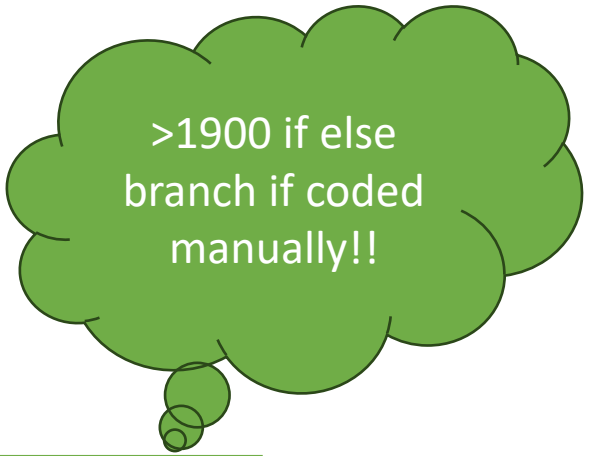
Model Details

- Deployed in multiple team in AMD with 100+ users for multiple project.

Model Details

	Training Data Metrics
No. Training Data Set	33460
No. Owner	67
No. Config	5
No. Unique Signature	523

	Performance (On Ryzen 5 CPU)
Training Time	1 minute
Inference Time	~5 seconds



	Existing Signature Model	New Model
Average No. of Decision Tree Branch	1256.84	698.54

Model Evaluation

Compare Model Updated data with manual assignment result

Variant 1 result (Around feature complete)

Week	Signature Count	Match Count	Test Count	Accuracy Rate	New Signature
1	83	76	1159	91.57%	11
2	112	106	1932	94.64%	42
3	85	82	1468	96.47%	10
4	80	74	1200	92.5%	13

Variant 2 result (2 weeks after DV regression start)

Week	Signature Count	Match Count	Test Count	Accuracy Rate	New Signature
1	75	71	2014	94.57%	5
2	75	71	1653	94.57%	10
3	62	65	1122	91.94%	15
4	68	57	503	95.58%	4

Model Evaluation (contd)

Regression Failure Category

Index	Content
1	Patterns observed in historical data.
2	Patterns are **not** observed in historical data but can be derived from domain knowledge.
3	Patterns were not observed in historical data, and the domain knowledge did not include such information during training.
4	Patterns are observed in historical data, but historical data is not applicable.

SSTA Highlights

- Fully automated process - eliminates ~99% of current tedious and repetitive manual workload.
- Prediction and reporting system are independent modules so it can be extended to other triage system.
- Resource (HW/SW) cost is negligible (no need of GPU!).

*Downloading current unassigned signatures is a manual process due to unavailability of API in CRDB. Solution is under development from CRDB team

Observations - ML

- ML works well in classification problem.
- The system can be extended to other flow easily as the metrics can be removed/added easily.
- 100% accuracy is hard to achieve, may not worth the effort.
- ChatGPT is very helpful during the tool development to learn AI domain knowledge

Thank you!

Any questions or suggestions?

Lingkai.Shi@amd.com