9th UCAAT User Conference on Advanced Automated Testing

JanIA: Intelligent Practices for Automated Testing

JanIA DIGITAL BUSINESS ASSURANCE

Presented by: Pablo Manso García-Mauriño

mtp DIGITAL BUSINESS ASSURANCE

14/09/2022

What is JanIA?

JanIA DIGITAL BUSINESS ASSURANCE Intelligent Decisions

JanIA Intelligent Decisions:

- It is an Artificial Intelligence (AI) solution applied in the field of quality assurance, control and quality engineering of information systems.
- It is based on the exploitation of all the information generated by the software processes and environments for the intelligent governance of the applications.

Which benefits does JanIA provide?

Reducing the number of incidents in production

Reduction of maintenance costs.

Reduced time-to-market of applications.

Avoiding "unnecessary" tests

Which capabilities could JanIA offer?

- Testing Optimization
  - Functional testing
  - Non-functional testing
  - Regression testing
  - Acceptance testing

- Recommend testing strategy
- Predict test results
- Prioritize regression tests

- Sofware Risks View
  - SW bugs
  - Improvement points

- Predict the risk level of a release
- Predict failure-prone SW components
- Correlate code quality with production issues

JanIA, how is it structured?

Static SW Metrics

KLOC
V(G)
…

SW Promotions

Dates
Incidents
…

Pre-Production testing

Test cases defined
Test cases executed
Defects encountered
…

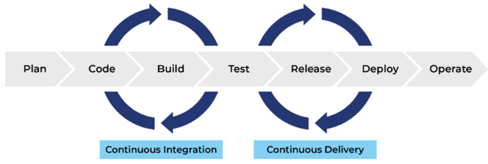
Historic data from +9 months

JanIA DIGITAL BUSINESS ASSURANCE
Intelligent decisions

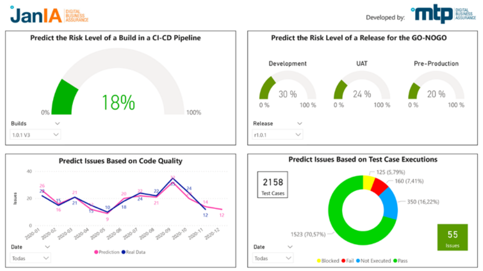
Machine Learning
- Spatial Clustering (DBSCAN)
- Clustering KMEANS
- Spectral Clustering
- Hierarchical Clustering
- Gaussian Classifier (Gaussian Naive Bayes -GNB)
- Support-Vector Machine (SVM)
- Decision Tree
- Linear Discriminant Analysis (LDA)
- Support Vector Regression (SVR)
- Linear Regression

Deep Learning
- Convolutional Neural Network (CNN)
- Long Short-Term Mermory Neural Network (LSTM)

CI/CD



Quality Gates (medium term)



User friendly DashBoard (deliverable)

Feasibility analysis to apply JanIA

Feasibility Analysis: Proof of Concept

- Objective:

Explore the feasibility of applying Predictive Modeling Techniques (based on Machine Learning) for Risk Assessment associated with different software applications in different aspects such as Quality Assurance or Control.

- **Data availability?**
  Representativeness, Volume, Variability, Consistency
- **Predictive capacity?**
  Target variables/ KPIs, Mathematical Models
- **Favorable results?**
  Confidence, Alignment, Improvement capacity

REAL USE CASE (POC)

- Context and motivations
- Use Case 1: Accesibility (ACCE) and Vulnerability (VULNE) Test Prediction
- Use Case 2: Performance Test Prediction

Real use case (POC)

Context and motivations:

- Project environments

Development Environment:

QA/QC Enviroment:

Production Environment:

- Large number of versions to promote monthly.
- Manual and expensive pre-exploitation tests to implement.
- Small testing team.

- Bottleneck between Pre-exploitation and Production environments.
- High time-to-market.
- Applications in production with errors.

Real use case (POC)

Use Case 1: ACCE and VULNE Test Predictions

- Objective: Predict the result of the Accessibility and Vulnerability Tests of a version of an application taking as input a set of data composed of information on the version itself and its previous versions.

Initial data

- Collecting data on the versions and applications to be analyzed

JanIA DIGITAL BUSINESS ASSURANCE

Supervised Learning Model

Expected result

- Test results
- Confidence in prediction
- Prioritization of the tests to be executed

Use Case 1: ACCE and VULNE Test Predictions

- Problem Formulation
  - Input Features $\boldsymbol{X}$: The feature vector $x^{(i)}$ is formed by *31* variables obtained from the data provided by a static code analysis tool together with data from previous versions of the applications themselves

Training set size: Number of versions promoted to Pre-Exploitation environment(~2500)

Rx[1]	Rx[2]	…	Rx[30]	Rx[31]
99	91	…	4	1
80	100	…	9	0

A version promoted to pre-exploitation corresponds to an example in the training set

Use Case 1: ACCE and VULNE Test Predictions

- Problem formulation
  - Output Labels Y:
    - Test Result ACCE and VULNE: These are discretized into binary classes, Test Accepted and Test Rejected which correspond respectively to classes 0 and 1 used in ML models.
    - Recommendation probability: Value between 0 and 1 that represents the exact probability offered by the model to assign one class or another.

Use Case 1: ACCE and VULNE Test Predictions

- Training dataset generation
  - For the generation of the train dataset, application data has been taken from 2016 to 2022. Among them are:

Health factors + Defects by characteristic and priority + Code Metrics + Other explanatory variables

  - Automatic labeling and preprocessing.

Information about version labeling and preprocessing:

As is well known in supervised learning models, input data must be labeled. In this case, this process is automatic, which is an advantage when applying the model, since it is not necessary to invest time on it.

Use Case 1: ACCE and VULNE Test Predictions

- Model training:

Items to have in mind

- Binary classification problem
- Unbalanced dataset
- Conservative vs Bold Strategy
- Feature importance

ML Models checked out

- Algorithm selection:
  - Logistic Regression
  - XGBoost
  - Random Forest
- Data Split (rebalanced):
  - Training on 85%
  - Validation on 15%

Real use case (POC)

Use Case 1: ACCE and VULNE Test Predictions

- Results

- Not prioritizing tests in 21.2% of the versions. Accessibility problems detected with a 91% accuracy.
- Prioritize tests in 52% of the versions. Accessibility problems detected with 86% accuracy.
- Moderate/Low confidence 26% of versions

21.2%
52%

ACCESSIBILITY TESTS

VULNERABILITY TESTS

28.7%
27.2%

- Not prioritizing tests in 28.7% of versions. Vulnerability problems detected with a 97% accuracy.
- Prioritize testing in 27.2% of versions. Vulnerability problems detected with 82% accuracy.
- Moderate/Low confidence 44.1% of versions

Real use case (POC)

Use Case 1: ACCE and VULNE Test Predictions

- Results

Tests:		Vulnerability		Accesibility	
Apps:	Version:	Recomendation:	Decision:	Recomendation:	Decision:
PRPC	02	NO	PDTE	NO	PDTE
CAPP	04.04.01.01	NO	PDTE	YES	PDTE
CVIM	04.04	YES	NO	NO	YES
PRPI	04.00.00.45	NO	PDTE	NO	PDTE
CIMP	01	NO	PDTE	NO	PDTE
CINI	01.10.00.01	NO	PDTE	NO	PDTE
ADGC	02.04	NO	PDTE	NO	PDTE
AGFE	03.01.21.04	NO	PDTE	YES	PDTE
AIFI	04.25	NO	PDTE	YES	PDTE
AIFO	06.02.00.03	NO	PDTE	YES	PDTE

- Low Priority
- Medium Priority
- High Priority
- Against the recommendation

Real use case (POC)

Use Case 2: Performance Test Prediction

- Objective: Predict the production behavior of a version taking as input a set of data composed of information on the version itself and the performance of its previous versions together with the defects detected in production.

Initial data

- Collecting data on the versions and applications to be analyzed

→ JanIA DIGITAL BUSINESS ASSURANCE

Supervised Learning Mode →

Expected result

- Production Performance
- Confidence in prediction
- Correlation with pre Performance tests

Use Case 2: Performance Test Prediction

- Problem Formulation
  - Input Features $\boldsymbol{X}$: The feature vector $x^{(i)}$ is formed by *33* variables obtained from the production data of previous versions related to execution errors and various metrics such as response time or CPU consumption.

Training set size: Number of versions promoted to Exploitation environment(~3000)

Rx[1]	Rx[2]	…	Rx[32]	Rx[33]
99	91	…	2,4	350
80	100	…	7,8	129

A production version corresponds to an example of the training set

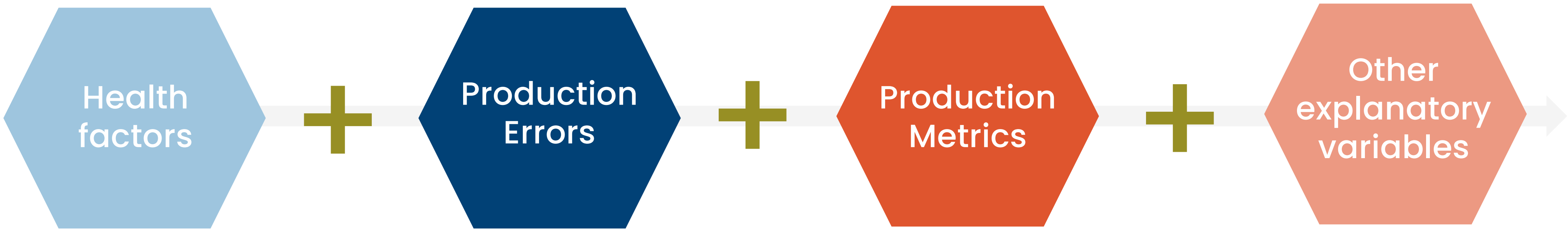
Use Case 2: Performance Test Prediction

- Problem Formulation
  - Output Labels Y:
    - Performance in production: This value is discretized into binary classes, Good Performance and Bad Performance that correspond respectively to classes 0 and 1 used in the ML models.
    - Recommendation probability: Value between 0 and 1 that represents the exact probability offered by the model to assign one class or another.

Real use case (POC)

Use Case 2: Performance Test Prediction

- Training dataset generation
  - For the generation of the train dataset, application data has been taken from 2016 to 2022. Among them are:



  - Automatic labeling and preprocessing.

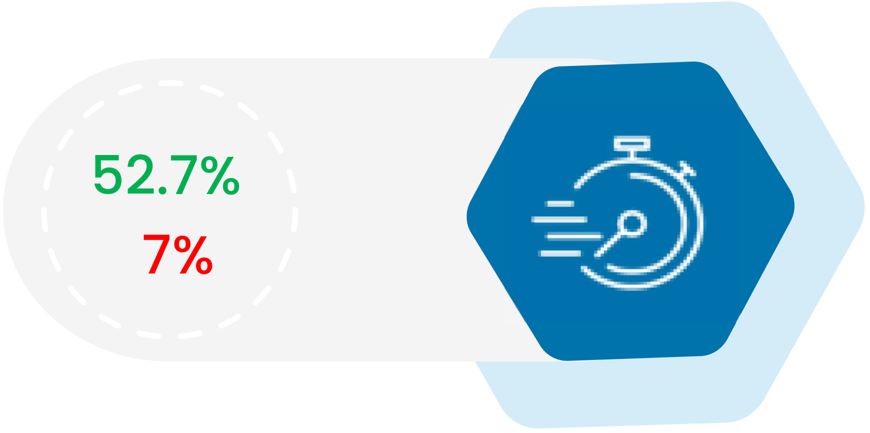
Information about version labeling and preprocessing:
For the labeling of the versions, thresholds calculated monthly from the metrics and errors of applications in production during the previous month have been taken into account.

Use Case 2: Performance Test Prediction

- Results

- 52.7% of the versions in the test data receive a "GOOD" PERFORMANCE prediction with a 98.5% accuracy.
- Regarding the prediction of versions with "BAD" PERFORMANCE, a 96.1% success rate is achieved and a range of 7% of the versions analyzed in the test data.
- Moderate/Low confidence 40% of versions

PERFORMANCE IN PRODUCTION



Conclusions

- The results obtained during the POC are good enough to put the solution into practice.
- The solution is currently in process of being integrated into the workflow between environments while continuing to be improved.
- The solution has been extended to other areas of the company where a new proof of concept is being developed.

Any further questions?

Contact me:
pablo.manso@mtp.es