AI Powered Test Failure Prediction Dashboard

Introduction

This project presents an intelligent test analytics system that combines automation testing and data science. The system analyzes historical automation test execution data and predicts the probability of failure for future test runs. It provides a web based dashboard to help quality assurance teams make data driven decisions and optimize regression testing cycles.

Objective

The main objective of this project is to build a predictive model that identifies high risk test cases using historical execution data. The system aims to reduce regression testing time, improve release confidence, and support proactive defect detection.

Problem Statement

In large software applications, regression suites contain hundreds or thousands of test cases. Executing the entire suite for every build consumes significant time and resources. Traditional automation frameworks execute all test cases without understanding which ones are more likely to fail. There is a need for a predictive system that can analyze past results and prioritize high risk test cases.

Proposed Solution

The proposed solution is a Streamlit based dashboard that performs the following tasks:

1. Upload historical automation test results in CSV format
2. Preprocess and clean the data
3. Train a machine learning model using historical records
4. Predict failure probability for each test case
5. Generate a summary report including execution statistics and risk analysis
6. Provide downloadable detailed results for stakeholders

Technology Stack

Programming Language: Python
Frontend Framework: Streamlit
Machine Learning Library: Scikit learn
Data Processing: Pandas and NumPy
Visualization: Matplotlib and Seaborn
Automation Data Source: Selenium or Playwright test results

Methodology

The system collects historical test execution data containing information such as module name, execution time, previous failure count, build number, and last run status.

Data preprocessing includes encoding categorical values and preparing feature variables. A Random Forest classification algorithm is trained using the processed data.

The model predicts whether a test case is likely to pass or fail. Model performance is evaluated using accuracy score, confusion matrix, precision, and recall.

System Workflow

Automation test execution generates result data.
The result data is uploaded into the dashboard.
The system trains the machine learning model.
Predictions are generated for each test case.
A summary report is displayed showing overall statistics and module wise risk levels.

Key Features

The dashboard provides total number of test cases, actual failures, predicted failures, and failure rate percentage.
It identifies high risk test cases.
It performs module wise risk analysis.
It displays model performance metrics.
It allows downloading of the detailed report in CSV format.

Results

The system successfully predicts potential failures based on historical data. High risk test cases can be identified before execution, allowing teams to prioritize regression testing efforts. The dashboard provides clear insights into unstable modules and overall test health.

Business Impact

This project helps reduce regression testing effort by focusing on high risk areas. It improves test planning and supports faster release cycles. The solution promotes data driven quality engineering and proactive defect management.

Conclusion

The AI Powered Test Failure Prediction Dashboard demonstrates the integration of automation testing and data science. It enhances traditional automation frameworks by adding predictive intelligence. The project shows how machine learning can be applied in quality assurance to improve efficiency, reduce risk, and support better decision making.