**Auto-Defect-Validator**

In the Software Testing Life Cycle (STLC), the process of defect validation occurs after a defect has been identified and logged into the defect management tools. The Scrum Master, Defect Manager, or Business Analyst (BA) is responsible for reviewing and validating the defects to determine if they are valid or invalid issues.

When a defect is marked as invalid, the defect manager writes comments explaining the reasons for rejecting the defect. These reasons can vary depending on the nature of the defect and the specific circumstances. Some common reasons for marking a defect as invalid include:

**Duplicates**: If the same defect has already been reported by another tester or team member, it is considered a duplicate and marked as invalid. Duplicates can occur when multiple individuals identify and report the same issue separately.

**As per design:** If the reported behavior aligns with the intended design or specifications, it may be considered as per the expected functionality and marked as invalid. This means that the system is behaving correctly based on the design requirements.

**Test Data Issue:** Sometimes defects can be attributed to incorrect or inadequate test data. If the defect is caused by test data-related issues, such as invalid inputs or incomplete datasets, it may be marked as invalid. This suggests that the issue lies with the test data rather than a genuine flaw in the system.

**Test Environment Issue:** Defects can also arise due to problems in the testing environment, such as configuration issues, hardware limitations, or network problems. If the defect is found to be specific to the testing environment and does not impact the actual system under test, it may be marked as invalid.

**Requirement**: Given the list of defects with some relevant data along with their comments, Build a model which will automate defect manager’s job of validating the defects. A model should take the new defects as an input and should classify each defect either Accept / Reject (Valid / In-Valid) along with the confidence score for each

**Data**

Data consists of two files in json format:

1.Valid defects

2.Invalid / Cancelled defects

3.along with their comments and other relevant data

Data is attached in the doc. The data dictionary is given below.

1. Status: The status of the defect

2. Description: The user description related to the defect.

3. Summary: Summary related to the defect

4. Comments: The comments related to the defect.

5. Root cause: What is the root cause of the defect

6. Priority: The priority of the defect(High, Medium, Low )

7. Assignee: The developer assigned to the defect.

8. Application: The application name in which the defect was raise

Day-01

1. Understand the Problem
2. Load the JSON Data
3. Data Transformation
4. Data Exploration
5. Come up with the approach
6. Data Preprocessing

* Cleaning the data
* Handling missing values
* Standardizing the data