

Test Plan for DeepSeek Email Classification and OCR Solution

1. Objective: The objective of this test plan is to validate the functionality of the DeepSeek Email Classification and OCR solution by testing it with various sample emails in .eml and .msg formats. The solution should accurately classify emails, extract key details such as amount, date, and deal name, and generate a structured output.

2. Test Environment:

- Operating System: Windows 10/11 or Linux
- Python Version: 3.8+
- Dependencies Installed:
 - torch
 - transformers
 - datasets
 - fastapi
 - uvicorn
 - extract_msg
 - eml_parser
 - pytesseract
 - pdf2image
 - pillow

3. Test Scenarios:

Test Case ID	Test Scenario	Input	Expected Output
TC-001	Classify .eml email	Lending service request .eml	Request type and extracted details
TC-002	Classify .msg email	Lending service request .msg	Request type and extracted details

Test Case ID	Test Scenario	Input	Expected Output
TC-003	Extract details from an email containing structured text	Email with amount, date, and deal name	Correctly extracted details
TC-004	Handle email with OCR attachment	Email with image/pdf attachment containing text	Extracted text from attachment
TC-005	Handle unsupported file formats	Random file uploaded	Error message

4. Steps to Execute Tests:

1. Start the API using the command: `uvicorn api:app --reload`
2. Upload test .eml and .msg files to the `/classify-email` endpoint using a tool like Postman.
3. Verify the JSON response for classification and extracted details.
4. Test with various email formats to confirm robustness.
5. Log all results into a CSV file.

5. Expected Results:

- The API should classify emails accurately based on predefined categories.
- Extracted details should match the expected values in test emails.
- OCR should correctly extract text from images and PDFs.
- The system should return an error for unsupported file formats.

6. Test Completion Criteria:

- All test scenarios must pass successfully.
- Any issues encountered must be documented and resolved before final deployment.

7. Deliverables:

- A CSV file containing test results.
- A summary of any issues encountered and resolutions.

8. Test Execution Logs: Test execution results will be documented in `test_results.csv` with fields:
| Test Case ID | Input File | Expected Output | Actual Output | Status (Pass/Fail) |