Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Test Name | Purpose of Test | Expected Result | Actual Result | Actions Required |
| 1 | Running in incorrect order | Provide error identification and allow the implementation of try catch methods for handling to ensure errors are formatted with constructive. This should assist in making the program more functional. | Program should stop and provide a message to the user with instructions on missing previous step required for the process. | **Import Libraries -** Working as expected.  **Fetch Dataset -** Working as expected.  **Datasetup -** Working as expected.  **Data Visualisation -** Method outside of try catch errored and wasn’t caught.  **Model Setup -** Working as expected.  **Train Models –** Working as expected.  **Save Trained Models** – Working as expected.  **Fetch Trained Models** – No error handling methods for broken links  **Load models** – Missing ”Import Library” check.  **Test models** – Working as expected.  **Model history visualisation** – Try Catch missing on error.  **Ensembling** – Try Catch missing on error. | **Data Visualisation –** Contain all runtime methods inside trycatch.  **Fetch Trained Models** – Implement method to clearly inform user of fetch issues.  **Model history visualisation** – check that “Load Models” was executed  **Ensembling** – check that “Load Models” was executed |
| 2 | Check try catches | Similar reasons as test 1, but ensure all messages are clear and accurate and helpful message are provided when system is run incorrectly. | Clearly explained messages pointing to correct cell for running. | **Import Libraries –** No message required  **Fetch Dataset -** No message required  **Datasetup –** Clear instruction.  **Data Visualisation -** Clear instruction. Method outside of try catch errored and wasn’t caught.  **Model Setup -** Clear instruction.  **Train Models –** Clear instruction.  **Save Trained Models** – Catch message asks of model has been trained, rather than telling user to execute train model’s cell.  **Fetch Trained Models** – No try catches present for getFile methods.  **Load models** – Missing instruction to run “Import Library”  **Test models** – Message for dataSetup is present, but no check to see if models are present.  **Model history visualisation** – missing try catch messages, no check for required data.  **Ensembling** – No try catch for block, not customised error for used. | **Data Visualisation –** Put all executables inside the catch.  **Save Trained Models** – Provide a clearer message for the user.  **Fetch Trained Models** – Make bad fetches cleared for user, point them to train manually.  **Load models** – amend try catch message to also instruct user to run “Import Library”.  **Test models** – add a check for “load models”, provide instruction for user to run “load models” cell.  **Model history visualisation** – add a check for “load models”, provide instruction for user to run “load models” cell.  **Ensembling** – add a check for “load models”, provide instruction for user to run “load models” cell. |
| 3 | Missing Models | Make sure the system is still functional in the case of a missing model, and that relative information is passed to the user to acquire the missing model with the methods provided. | Inform the user with which model is missing, and how to train new if needed | No check for individual models being fetched and system errors when missing, issue with variable modelsAreFetched having no condition for activation. | Provide conditions for “ModelsAreFetched” variable to activate after checking fetched models were 100% fetched. Otherwise have user train new models if this isn’t possible. |
| 4 | Pneumonia image  classification | Check that the program is classifying new pneumonia images correctly. | Positive for classification of Pneumonia. | Model classified with Pneumonia. | Working as expected. |
| 5 | Healthy image  classification | Check that the program is classifying new images without pneumonia correctly. | Negative for classification of Pneumonia. | Model classified with Pneumonia. | Classified incorrectly over 20 randomly selected images. Possible bias issue needing further investigation. |
| 6 | Alternative disease image  classification | Check how the program handles lung images of other diseases. | Program should return a classification without error. | Model classified with Pneumonia. | Classified Pneumonia over 5 randomly selected images. Possible bias issue needing further investigation. |
| 7 | None-relevant image classification | Check what the system does when provided an image outside the scope of what is expected, for example a cat or a dog. | Program should return a classification without error. | Model classified with Pneumonia. | Classified Pneumonia over 5 randomly selected images. Possible bias issue needing further investigation. |
| 8 | Incorrect Size of image used | Ensure the system can process the inputted image | System should reshape image regardless of size | Shaped all tests with no problems | Working as expected. |
| 9 | Validation test | Check system is running expected overall after bias fix | System should complete all tests without errors | Functions without errors | Issue with VGG download path in fetch models. Location key and file name require correction. |
| 10 | Further testing of image classification | Retest image classification after possible bias issues discovered from test 4-8. | Images should classify correctly same amount as each model’s accuracy. | Models should classify pneumonia and no pneumonia present images. | Normal Classification appearing correctly more often than previous test. |