IMPaCT Intelligence

Workplace Violence (WPV) Prediction Test Plan

Prepared by: Anthony Ung
Version: 1.0
Date: 04/14/2024
Sponsors: James Corbett (IMPaCT Intelligence) and Arend Van der Veen (WBAT Safety)

Table of Contents

1 INTRODUCTION

1.1 Document Purpose

The Test Protocol document contains the Test Procedures to be conducted according to the corresponding Validation Plan.

1.2 Document Organization

The remainder of this document is organized as follows:

- Section 2, Validation Strategy, contains the test levels and objectives, hardware and software required for validation, Resource and training requirements
- Section 3 contains a summary of the completed System Testing
- Section 4, Validation Methodology, explains the scope of validation, the organization and sequencing of validation procedures.
- Section 5, Validation Procedures, contains the actual validation procedure and the test script.

1.3 Project Background

This document contains the Test Protocol for the Workplace Violence Prediction API. The testing is done to ensure that all functional and nonfunctional requirements are satisfied.

1.4 Related Documents

Document Name	Version #	Location
Workplace Violence Prediction Requirements Document	N/A	
Workplace Violence Prediction Design and Architecture Document	N/A	

2 Validation Strategy

2.1 Testing Levels and Objectives

2.1.1 Testing Objectives

This protocol is written to accomplish the following testing objectives to ensure:
- The Machine Learning Model functions as intended
- The automated eMail alert system works as intended
- The necessary infrastructure is in place for both of these systems to work as intended.

2.1.2 Testing Levels

There are various levels of testing being done. The Developers Script includes various Unit and Integration tests. The Administration and Security Script System Testing.

2.1.2.1 Developers

The Developers Script (Section 5.2), which reflects the roles of the IT staff, includes various unit and integration tests for the following functionality:
- System Initialization
- Bearer Token
- Data Generation
- Testing and Automation
- Visualization
- eMail Notification
- Data Storage
- Performance
- Security

2.1.2.2 Client Script

The Client Script (Section 5.3) provides the sole system test identified in the validation plan.

2.2 System Requirements

This section describes the validation environment. The Validation environment should represent the eventual production environment as closely as possible. The descriptions that follow should indicate any deviations in the Validation environment from the expected production environment.

2.2.1 Hardware Configuration

The particular laptop chosen to do the testing is Anthony Ung's Lenovo X270 (with a dual-core 6th Generation i7) because this laptop is significantly older than the computers expected to be found in the production environment.
Some form of internet connectivity is needed.

2.2.2 Software Configuration

In order to execute the test/validation procedures, you need a computer with the capability to run the following software

- PyCharm in order to build the API and install the dependencies listed in the requirements file.
- Docker in order to run the API containers.
- Postman in order to test the API routes.

2.2.3 Source Code

Python 3.12 is required to run the WPV API.

2.2.4 Libraries/Directories

All requirements are identified in the requirements.txt file.
The pip install -r requirements.txt command will automatically install all required libraries for you.

2.2.5 Migration of Software and Data

This section describes procedures for migrating software and data into the validation environment.

The run configurations needed to launch the Docker container are found in ./idea/runConfigurations/.

Also, in order to set up connections to the database in the validation environment, you will need to update the following fields in the config.toml file:

user = "anthony"
password = "ungant67"
host = "ssh.bobbitt.dev"

bearer = "2a7b611b6571b19140fffeb8c20a83907ee927cd"

This is the same database which will be used in the production environment.

2.3 Resource Types and Training Needs

These are the following people who are needed to successfully validate the Workplace Violence Prevention API.
The specific personnel and their training needs are documented below:

Resource	Location	Training Needs
Anthony Ung	Rowan University	N/A
James Corbett	IMPaCT Intelligence	N/A
Arend Van der Veen	WBAT Safety	N/A

3 System Test Evaluation

The system test is primarily driven by unit testing of various pieces of functionality in the Workplace Violence API.

Test Scripts 5.1.1 and 5.1.2 initialize the system and establish that the tester has the credentials they should have. Test Scripts 5.1.3 through 5.1.8 are unit and integration tests which determine that the most important functional and nonfunctional requirements are fulfilled.

Test Script 5.2 is a system test on only the parts of the WPV API that the security and hospital administration are able to meaningfully interact with.

4 Validation Methodology

4.1 General Information

The methodology used in the validation efforts is mostly unit testing on the riskiest modules identified in the corresponding validation plan.

4.2 Organization of Validation Procedures

I have provided test scripts 5.1.1 and 5.1.2 to verify that (1) the WPV API can be initialized, and (2) the credentials work as intended. The other tests in the Developers Script are unit tests.

5 Validation Procedures

This section contains the test cases and scripts needed to verify that all functional and nonfunctional requirements are fulfilled. A test case describes what is being tested and a test script is the detailed procedures for executing a test case. See the forms that follow this page for the test cases/scripts.

5.1 Initialization Script

The Initialization Script is provided as a courtesy to the Phase 2 Developers.

These three test scripts do not test any functional or nonfunctional requirements identified in the corresponding Validation Plan.

5.1.1 System Initialization Script

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
N/A	1		Open Pycharm	Pycharm Opens				
	2	Listed in 2.2.6	Populate the config.toml file with the correct credentials					
			Select the “Docker” run configuration at the top of the window next to the Green Play Button.					
			Click the Green Play Button.	The Docker Image containing the WPV API and the data generation API will be created. This will take a few minutes.				
	3		Open Docker	Docker Opens				
			Click on “Containers”	The Docker container that was created in Step 2 opens.				
			Click on “workplace-violence-prevention-api”	The console output from “wpv-api” and “dummy-api” is visible.				
	3A		If you have already built a Docker Image and have not made any changes, you can navigate back to the corresponding Docker Container in Step 3 to relaunch the API.	The console output from “wpv-api” and “dummy-api” is visible.				

5.1.2 Bearer Token Script

This test script is provided so that developers who are new to the team can get their credentials.

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
N/A	1		In Postman, create a GET request to 127.0.0.1:8000/api/token					
	2	anthony ungant67	In the “Authorization” tab, enter the username and password which were given to you by the developer who manages the credentials.	A JSON document will be returned with your key.				

5.1.3 Login Script

This test script is provided so that developers can view the current version of the webpage.

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
N/A	1		At the login page, enter the following: Username: “anthony” Password: “ungant67”					
			Click the blue “Log In” button.	You will be logged in and taken to the home page.				

5.2 Developers Script

These test cases validate the functionality of the system initialization, the data generation, the machine learning model predictions, and the eMail Notification System.

These developer scripts assume that the developer is able to run the test scripts identified in 5.1.1 and 5.1.3.

5.2.1 Data Generation Script

This test script is a unit test which determines whether (1) data is being generated, (2) the data is in the correct format which the machine learning model can understand, and (3) the data has consistent values.

This test script assumes that 5.1.1 and 5.1.2 have passed.

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
2	1		In Postman, create a GET request to 127.0.0.1:8000/api/data/latest					
			Under the “Authorization” tab, paste the bearer token that you obtained from 5.1.2.					
			Click the blue “SEND” button.	A JSON object is returned with the latest HospitalData object that was created.				
	2		Check the invariants for the generated data.	percentBedsFull should be between 0 and 1. timeOfDay should be formatted as “HH:MM:SS”.				

5.2.2 Testing and Automation Script

This test script is provided to assist the developers in inspecting the actual Machine Learning Model itself and ensure that it has the desired accuracy.
This test script assumes that 5.1.1 and 5.1.2 have passed.

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
3	1		Scroll to the top of the output generated by “wpv-api”.					
			Find the line that says “Forest model complete!”					
			Find the number that was printed above.	The value should be equal to or greater than 0.9 (representing the desired 90% accuracy stated in the requirements document)				

5.2.3 Visualization Script

This test script is provided to assist the developers in inspecting the User Interface itself.

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
4	1		Navigate to 127.0.0.1:8000 in your web browser.	The webpage for the WPV API is is displayed with the current WPV risk level. It refreshes each time a prediction is made (Every 10 seconds)				
	2		Click the “Log Incident” button	Your browser takes you to the incident logging page.				
	3		Click the “Go Back” button.	Your browser takes you back to the home page.				
	4		Click the blue “Manage Emails” button.	Your browser takes you to the eMail management page.				
	5		Click the “Go to Home Page” hyperlink.	Your browser takes you back to the home page.				

5.2.4 Email Notification Script

This test script provides unit testing for the eMail Notification System itself.

${NAME}: _____

${EMAIL}: _____

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
5	1		On the home page (127.0.0.1:8000), click on the blue “Manage eMails” button.	The browser takes you to the “Manage Emails” page.				
	2		Write your name and eMail address on the lines provided.					
		${NAME} ${EMAIL}	Enter your name and eMail addresses in the provided boxes. Then click the blue “Add e-mail” button.	Your e-mail address appears in the list below.				
	3		Click the yellow “Send Email” button.	An eMail is sent from the WPV API to the eMail addresses in the list.				

5.2.5 Performance Requirements Script

This test script is an integration test to ensure that the machine learning model meets the performance non-functional requirement.

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
13	1		In Postman, create a GET request to 127.0.0.1:8000/api/data/latest					
			Under the “Authorization” tab, paste the bearer token that you obtained from 5.1.2.					
			Click the blue “SEND” button.	A JSON object is returned with the latest HospitalData object that was created.				
	2		Inspect the output lines generated	The following lines of output should be generated: - “get_data” executed successfully - The row number along with the WPV risk and percentage. - “predict” executed successfully For the performance requirement to be fulfilled, the second and third lines should be printed before the next “get_data” line is.				

5.2.6 Security Script

This integration test ensures that nobody can inspect the data without having the proper credentials.

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
14	1		Navigate to http://localhost:8000/. If you are logged in (from running the previous test scripts), click the white “Log Out” button.	Your browser takes you to the login page.				
	2		At the login page, enter the following: Username: “jack” Password: “myersjac”					
			Click the blue “Log In” button.	You will be denied access with a red message.				
	3		At the login page, enter the following: Username: “anthony” Password: “ungant67”					
			Click the blue “Log In” button.	You will be logged in and taken to the home page.				

5.3 Client Script

This test script provides the only system-level test identified by the Validation Plan.

5.3.1 Client Script

${DATE}: _____

${TIME}: _____

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
4	1		Navigate to http://127.0.0.1:8000/ in the web browser.	The main web page appears with the dial of the latest prediction. It updates every 10 seconds.				
	2		Click on the “Log Incident” button	The web page for logging incidents appears.				
	3		Right-click anywhere on the page. Click on “Inspect”. Navigate to “Console”.					
			Enter the following fields: Type: “Test” Date/Time: “09-06-2018 18:45:00” Affected People: “Anthony” Description: “STOPit”	N/A				
			Click on the blue “Submit Incident” button	You should get a 404 Error Message.				

ID	Step	Input or Prequisites	Procedure	Expected results	Actual Results	Initials & Date/Time	Attachment Ref.	Pass/ Fail
	4		Write the current Date and Time on the lines provided.					
			Enter the following fields: Type: “N/A” Date/Time: [Click the calendar button. Find the date and time closest to ${DATE} and ${TIME}] Affected People: “N/A” Description: “N/A”					
			Click on the blue “Submit Incident” button.	In the console, you should get an “Incident Reported.” message.				
	5		Click on the browser’s “Back” button.	You return to the main webpage.				
	6		Click on the “Manage Emails” webpage.					
	7		Enter your eMail address in the box and click the blue “Add Email” button.	Your Email address is added to the database. The next time a WPV incident is predicted, you automatically get a notification.				