IMPaCT Intelligence

Workplace Violence Prediction

Requirements Document

Aiden Touhill

Date: 02/19/2024

Sponsored by James Corbett and Arend van der Veen

Preface

This document has been prepared as an overview of the functionality and requirements for the Workplace Violence (WPV) Prediction Software and is thus expected to be reviewed by the product sponsors. Though the goal of Team Orangutan is to deliver on their initial project outline, the requirements and functionality of the system may shift overtime. Thus, a sufficient portion of the outlined requirements are considered tentative.

Glossary

Term/Acronym	Definition
Model	Refers (broadly) to the system being used to evaluate hospital conditions. Data will be formatted and fed into an AI prediction model that will use a predetermined set of rules to assess potential stress in a department.
GUI	“Graphical User Interface”. Acts as a means for visualizing the output of the model and potentially allowing user input.
Database	A place for storing, formatting, and potentially returning collected data. This aspect of the system is invisible to the user but is crucial to its functionality.
“Null” marker	The “null” marker is meant to denote that a piece of data does not currently exist in a database.

Introduction

The purpose of the WPV Prediction API is to assess the likelihood that a violent incident will occur within a professional environment. The system will be tailored for integration into a hospital setting and may likely utilize real-time analysis of hospital resources, staff availability, and other contributing factors to create an intuitive display of estimated stress levels for each department. This display will work in conjunction with an automated alert system to notify hospital administration and security personnel of possible risks of a workplace violence incident.

Though the WPV Prediction Model is unlikely to be 100 percent accurate, it will serve as a quick and reliable visual tool for assessing the conditions of each department and will allow healthcare workers a greater deal of awareness and synchrony, increasing safety for both medical workers and the patients they treat.

**Note that a simple diagram of the described product functionality can be found at the bottom of this document.*

Functional Product Requirements

Team Orangutan prioritizes the development of a model that is fast and easy to implement. In order to achieve this, there are a series of tasks directly related to the development of the software that must be completed:

1. Model Development
    a. Team Orangutan began this project with a serious discussion regarding the appropriate machine learning model for processing input data. They studied various machine learning algorithms in order to determine the best possible model for accurately predicting workplace violence.
    b. The greatest candidate for this project is the Random Forest model. This model consists of a group of decision trees that work together to process data and make predictions.
2. Data Generation
    a. In the event that hospital systems are not accessible to the Scrum Team, the team must be able to test the capabilities of the WPV Prediction model using their own set of data. This will be accomplished using a simple data generation tool built with Python.
    b. Generated data will only approximate data which will be found in the workplace to the best of the team's knowledge. This data will not be pushed into the database during the testing phase–instead being fed directly into the model for easy testing. In later phases of the project, this data will be sanitized and stored.

3. Testing and Automation
    a. After the initial stages of model design have been completed, the developers have the capability to test the model's responses to various forms of data. Ideally, the model should be tested under a variety of input data from multiple departments.
    b. SciKit Learn's *accuracy_score* function provides information on the model's responses and numerical representations of its accuracy. This gives developers a definitive measurement of prediction quality and identifies areas which need improvement.
    c. After the model has been tested thoroughly, the team needs a feature that can autofill generated data into the system to simulate polling data from a live hospital environment. The team also needs mechanisms to format and sanitize the data to ensure that it is in a format which the model can understand. This data will then be fed into the model and processed.
4. Visualization
    a. The team creates a user interface to display a visual representation of the model output.
        i. This is dependent on having an acceptably accurate prediction model, a functional database, and mechanisms to both sanitize the data and get the data in and out of the model.
    b. Hospital workers need to be able to access all required functions from the UI. It will be split into three windows: the *department view*, the *email entry screen*, and the *logging screen*.

i. The department view displays a visual representation of the estimated stress levels for each department. This visualization will most likely come in the form of a vector dial and will be used as the primary bridge between model output and users. If possible, the team will design a “master dial” that averages the stress levels of each department into one overall visual.

ii. The logging screen will be used whenever a violent incident is predicted and confirmed. From here, hospital staff enter details about the incident and save it into a temporary storage log for the model to reference in the future. This feature of the GUI is described in more detail in the nonfunctional requirements section.

iii. The email entry screen allows users to opt in and out of the automated email notification system. The email system sends alerts whenever a department is under high stress.

1. The automated email system is not meant to be a call to action, but rather a means of keeping hospital staff informed of potentially stressful areas and conditions, thus increasing the level of overall situational awareness within the hospital.

5. Email Notification
    a. The system automatically sends email notifications to registered administrators and security personnel when high stress is detected in a given department.
    b. As mentioned, hospital staff have the capability to opt in and out of receiving such emails.

Non-functional Product Requirements

Although the primary focus of this project is the prediction model, it is necessary to acknowledge associated nonfunctional requirements that ensure the necessary infrastructure is implemented for the model to make its predictions and then communicate its predictions to the end user. Such requirements include:

11. Data Storage
    a. The development team needs a mechanism to store (1) training data for the machine learning model, (2) data which is either polled directly from the hospital environment or simulated to be polled from such an environment, and (3) past model predictions along with the associated polled data.
    b. The model connects to a MariaDB database by connecting to a remote server.
    c. Team members are given their own credentials and granted access to query the database. Members can connect to the MySQL database by entering the host IP, port, and their personal credentials.
    d. Sanitization and formatting of data is crucial for the accuracy of the model. Null values in particular will hinder model accuracy and pollute the database. The team has opted to resolve this by sanitizing and formatting data so that it does not feed any *null* values into the model.
12. Data Insertion/Retrieval
    a. HTTP routes provided by Django REST are created to link the data generation and model predictions to the backend.
    b. JSON objects are the primary mechanism by which data is moved in and out of the database.

13. User Interface Requirements

a. The GUI will be designed as simply as possible while still communicating all the data necessary for hospital staff to take action whenever risks of workplace violence are identified. It should be visually appealing and easy to navigate without prior experience.

    i. This will be achieved through the addition of labeled buttons and the appropriate distribution of GUI functions (keeping the appropriate components in the appropriate window).

b. Metering will be simple and uninterpretable. Individuals should be able to read and properly understand the data presented, even in the absence of technical skill. This can be accomplished by designing the vector dial with a color gradient. The gradient will be green on the far left, but will blur from yellow to red as the dial moves further to the right. A stock example of such dial can be found below:

14. Performance Requirements

Hospital workers need prediction software that can manage a large stream of data and display readings quickly. The model must be able to continuously interpret data and update graphics within 10 seconds of that data being received in order to keep up with potentially volatile working conditions

.

15. Security

    a. As of now, team Orangutan has decided on HTTP token authorization to protect the model from third-party requests.

    b. Staff of the hospital will be able to access the primary interface by entering their company credentials.

Note that additional functional and nonfunctional requirements may be appended as the scope of the project shifts. Further requirements will be prioritized based on their projected impact on product quality.