Project Title: System Verification and Validation Plan for Natural Language Processing for Mental Health Risk Prediction

Team 13, The Cognitive Care Crew
Jessica Dawson
Michael Breau
Matthew Curtis
Benjamin Chinnery
Yaruo Tian

1/08/2024

Revision History

Date	Version	Notes
11/03/2023	1.0	Revision 0
1/08/2024	1.1	Verbiage Changes & adjustments to AI Guidelines
3/03/2024	1.2	Updated Tests to reflect updates to functional requirements
3/04/2024	1.3	Added more evaluation metrics and updated Tests to reflect updates to functional requirements
3/04/2024	1.4	Updated traceability table
3/06/2024	1.5	Updated task 3 information

Contents

List of Tables

1 Symbols, Abbreviations, and Acronyms

symbol	description
UQAM	Université du Québec à Montréal
RMSE	Root Mean Square Error
MZOE	Mean Zero-One Error
MAE	Mean Absolute Error
RS	Restraint Subscale
ECS	Eating Concern Subscale
SCS	Shape Concern Subscale
WCS	Weight Concern Subscale
RP	Reciprocal Rank
DCG	Discounted Cumulative Gain
IDCG	Ideal Discounted Cumulative Gain
NDCG	Normalized Discounted Cumulative Gain

2 General Information

2.1 Summary

The CLEF eRisk competition, better known as the Early Risk Prediction on the internet competition, is an organization that hosts a yearly event where numerous research teams come together to explore new strategies and applications of early detection technologies, particularly in the field of health and safety. The team at McMaster has partnered alongside the University of Quebec in Montreal to help leverage their existing research and work done in prior years of the eRisk competition to help design a new system entry for this year. The team will not be in charge of the system operations portion of the project, but instead will be focused on designing new strategies and implementations for the Natural Language Processing portion of the project.

The NLP team will be responsible for analyzing textual input from real user data, and will use its provided training data and algorithms to determine the probability of the user showing signs of a selected mental health issue. The primary goal of the project is to explore applications of machine learning in this area and it is unlikely a system suitable for clinical use will actually be developed.

The structure of the system will be built around the three tasks laid out by eRisk: search for symptoms of depression, a sentence ranking task; early detection of signs of anorexia, a task built around detecting signs of anorexia in as few posts as possible; and measuring the severity of the signs of eating disorders, a task built around filling out a self-report question for an individual.

More details about the eRisk tasks can be found in the project's SRS (Breau et al., 2023).

2.2 Objectives

The primary objective of VnV for this project is to ensure the system's prediction models outperform certain baselines in order to prove the models hold some form of predictive power and value.

Another objective of VnV for this project is to ensure the system takes input and produces output in forms that will allow it to compete in eRisk. This is to ensure the competition can be used to evaluate the system's performance against other competing models.

A final objective is to ensure the system follows guidelines surrounding data privacy and confidentiality. More details about this can be found in the project's SRS section E.3 (Breau et al., 2023).

Usability is a low priority for this project, the focus is on research and trying different approaches, not creating a piece of software ready for clinical use.

2.3 Relevant Documentation

- Problem Statement

The Problem statement document abstractly identifies the problem to be solved, characterizing it in terms of its inputs and outputs, which gives context to the related environment and stakeholders.

- Development Plan

The Development Plan contains the project development overview including team roles, team meeting and communication plans, git workflow and project schedule.

- SRS

The Software Requirements Specification identifies the various functional and non-functional requirements related to the project, while providing context for the benefits and usage scenarios of the project.

- Hazard Analysis

The Hazard Analysis identifies various hazards and obstacles to the project's development and operation, along with mitigation strategies to help deal with those potential hazards. It carries a focus on ensuring the project can be delivered to the standards of the eRisk competition while maintaining safety and privacy of user data.

- Module Guide

The Module Guide decomposes the eRisk NLP project into numerous modules based on the principle of information hiding and separating data structures, in order to help increase understandability of various project components.

- Module Interface Specification

The Module Interface Specification identifies modules found in the Module Guide and decomposes them into more primitive data types and functions, in order to better observe the module functionality and ensure it's behaving as intended.

- McMaster AI Guidelines

McMaster's Provisional Guidelines on the use of AI provides useful rulesets that helps the team ensure the safe and responsible use of AI practises.

- eRisk 2022 Paper

The prior efforts done by the UQAM team in the eRisk competition were instrumental in the team's foundation of understanding and served as a jumping off point for any efforts.

3 Plan

This section will outline the plan for verification and validation of a number of different components within this project (3.1). This section will first give an overview of the members that are considered part of the verification and validation team and what their roles and involvement will consist of. Next there will be subsections regarding verification of the project's SRS (3.2) along with the design plan (3.3) and the verification and validation plan itself (3.4). This section will then outline the verification plan that has been created for the implementation of the project and model (3.5). Automated Testing and verification tools that will be used throughout this project is then the next subsection that is outlined in 3.6 followed lastly by the validation plan for the software (3.7).

3.1 Verification and Validation Team

The verification and validation team will consist of the core team members (Matthew, Jessica, Ben, Yaruo and Michael) along with the Capstone professor, the team's TA, Marie-Jean and Diego from the Montreal team, and Professor Mosser. The core team will be responsible for creating test suites that ensure correctness in the solution along with catching possible bugs and issues that may arise. The team will be responsible for creating suitable edge cases to evaluate the correctness of the work along with general automated test suites that will be automatically deployed when new code is pulled.

The core team will be responsible for creating all test suites, along with executing them and documenting the results. The team will also be responsible for making any changes that are required after testing the code. All core team members will have a hand in all sections of testing but different team members will have different focused responsibilities. Firstly, all core team members will be responsible for documenting the results of the automated test suites when their code enters the repository through a pull request. More specifically, Yaruo and Michaels main responsibility will be creating a set of tests including edge cases for the NLP model that will ensure that our model functions as expected for a wide variety of input data. They will be required to create test suites along with automated test suites that will be run periodically when pull requests happen. Jessica, Ben and Matthew on the other hand will have the primary responsibility for training the data vs a training data set in order to determine the accuracy of the model. They will also have the responsibility to ensure code structure in the test suites that are created along with organizing the suites while Yaruo and Michaels main role is coming up with and creating the tests.

The team will meet with Professor Mosser and their TA as well periodically throughout each milestone in order to discuss the requirements for the project regarding the current milestone and to help solidify expectations. This is an opportunity for the team to ask any questions they may have as well. The team also will meet with Marie-Jean and Diego from the Montreal team periodically to help guide the team with regard to requirements of the project and the eRisk competition along with what validation means to them. They will help guide the team to find what the important things are to focus on within our project. Team 8 will also provide the team feedback for every milestone. Lastly, the actual competition itself will be the final validation step when testing the model against a new set of data and reporting how the model performs.

3.2 SRS Verification Plan

All of the members of the core team will be taking part in the SRS Verification Plan along with Group 8, Professor Mosser and the TA. The team will verify the contents of the SRS by comparing the requirements outlined by the eRisk competition along with our team in Montreal with the requirements stated in our SRS document. This will ensure that the team has covered everything from the SRS document and can see if there is anything new that must be added. Most of the SRS verification plan will involve inspections not only from our team, but also from the team in Montreal along with peer reviews and ad hoc feedback from group 8.

Since our project is unique in the fact that the team is submitting to a competition with a rigid output and guideline structure, the team will also have to compare the rubric for the SRS document with what has been done in the original SRS document to ensure that all the required checkpoints have been covered. On top of this, the team will verify the SRS document by going over the feedback given to on the SRS revision 0 from TA along with the feedback received from group 8 regarding the SRS revision 0. Lastly, when verifying the SRS document, the team will talk with Professor Mosser and ask him any questions that arise during the reviewing and verification process within the group.

These are some of the major areas the team would like to cover in the SRS verification plan to ensure a test exists for each of these components:

- Is a functional overview of the system provided in the SRS?
- Are high-level usage scenarios included?
- Has the software environment and all its elements been specified?
- Are the limitations and exclusions accurate and all encompassing of the true limitations and exclusions of the project?
- Are the inputs and outputs of the system specified?
- Is there any unnecessary overlap in our requirements?

- For every function that is outlined, are the inputs specified sufficient enough to perform the required functionality?
- Do any requirements conflict with each other or does each requirement avoid conflicts with other requirements?
- Are all detailed usage scenarios covered in the document and are completed in adequate detail or are there others we could add?
- Does each component and requirement have a priority?
- Are there any chances in the prioritization chart we should add due to changes in the code since the creation of the original SRS (revision 0)?
- Are the requirements clear enough that it would be possible to give to an independent team for implementation and they would be able to understand the requirements?
- Is every requirement testable?
- Is the verification and acceptance criteria specific enough? If not, what quantifiable metrics can be added to each point?
- Has every definition, abbreviation and acronym been defined in our Glossary?
- Are all imposed technical choices listed in the document?
- Are the risks that may be present in this project along with their mitigation analysis included in this document?
- Is the "Requirements Process and Reports" accurate to what is expected by each of the teams listed in the "Requirements Process and Reports" section?
- Does the Context and Overall Objectives section paint an accurate enough description of the project's context and objectives?

3.3 Design Verification Plan

The plan for the design verification will be to go through the SRS document and make sure that each of the outputs and inputs that were planned for are accounted for and included in our design planning and document. The team will also look at reviews given by Group 8 on the SRS document to consider any changes they may want to add or things to consider for the design. The team will go through the same process for feedback received from the TA on the SRS report. They will also utilize the guidance of Marie-Jean and Diego in order to verify that they are on the right path with the design. This assistance can help guide the design and verify that it checks all the boxes and functionalities that are required for the competition.

The following section will go over some design verification questions that will be asked regarding the functions of each of the major components in our design.

3.4 Verification and Validation Plan Verification Plan

For the Verification and Validation Plan Verification Plan the team will be making use of the reviews given for the VnV Plan from group 8 as well as from the TA. This will help the team gain some insight on areas that may be missing or need more work within the document. The team will also be going through the SRS document and ensuring they have created at least 1 test for each functional and nonfunctional requirement. It is important to ensure that they have sufficient tests for the requirements. The team would also like to do some mutation testing in the code to determine what changes will affect the tests created and in what way. It will also show that there is a possibility of missing some important tests if a mutation expected to result in a change or failure, results in no failed tests.

Below is a checklist of the type of questions and topics the team would like to cover as part of the Verification and Validation Plan Verification Plan.

- Is the brief summary of the software in the summary section enough to get an understanding of the software that is being tested and its general functions?

- Are the objectives for the VnV plan clearly outlined and is each objective covered at some point throughout the document?
- Is the planning section detailed with checklists for each section and steps that will be taken for each plan?
- Is there at least one test created for each requirement in the SRS document?
- Are all functions of each component being tested?
- Are there a wide variety of tests including edge cases?
- Do the non-functional requirements contain tangible quantifiable values?
- Is the traceability between test cases and requirements clearly shown in section 4.3?

3.5 Implementation Verification Plan

The tests for accuracy will be one of the primary ways we verify our implementation. As new code is added to our repository and as the team works on our model, we will be using implementation verification techniques like code walk-throughs and code inspections periodically. Every time a new pull request is made, a fellow teammate will be required to perform a code inspection for that pull request and give feedback. If there is something unclear, the two teammates will meet and the reviewer will receive a code walk-through from the reviewee. We will also perform code walk-throughs for new changes made to the model when we meet as a team. Lastly we plan to have continuous integration that will allow the team to run automated testing for every pull request that gets created as well as utilizing Fake9 as mentioned below to ensure our implementation follows proper coding standards.

3.6 Automated Testing and Verification Tools

There will be three main tools the team will be using for automated testing and verification. Firstly, we will be using Flake9 which is a fork of Flake8. This will help the team follow proper coding standards and prevent problems arising regarding topics such as syntax errors, typos, incorrect styling, bad formatting and more. This will also help save time for when we do peer code reviews as a team. We will also be creating test suites using Pytest since it is a free open-sourced, simple and scalable Python-based Test Automation Framework. We will be using GitHub Actions in order to implement continuous integration and continuous delivery into our project. This will allow us to automatically build, test, and deploy our pipeline. This will create a pipeline that allows the team to build and test every pull request when they happen and also when the team deploys and merges a pull request onto another branch.

3.7 Software Validation Plan

There are a multitude of methods and parts to our software validation plan. Firstly, our primary validation method will be the actual process of competing in the eRisk competition. During the competition they will take our model and test it with new data and observe the output. This will be the final and largest validation step for us that will help validate the correctness and precision of our design. The team will also have a set of training data that will be used throughout the design process to train against, the team can use this to test how close the results are to the expected results, validating how accurate the design is.

Another large aspect of the software validation plan will be our review sessions and meeting with Marie-Jean and Diego throughout the project. This will be a time for them to look at our software and run it themselves. A large checkpoint when they will have an opportunity to have our project demoed for them is around the Rev 0 demo. This will give them an opportunity to validate that our project passes all the requirements of the competition or if there is something else we need to add or change. This is an important time for us to take this feedback and use it in order to improve our project. This will also be a time for us to ask questions and for them to confirm the requirements are all being met in our design. These questions and requirements would largely surround the overall output of the design and the overall process (largely focusing on the training of the data and the prediction model created and used). It would also include validation of each of the program's components individually and as a collective (components being the Text Pre-Processing, Vectorization, Prediction and Output components).

Before that however, The team will meet with professor Mosser to discuss if he thinks that all the requirements documented line up with and are accurate to the requirements outlined for the course. The team will do this with Marie-Jean and Diego as well with respect to the requirements from the competition. The team will also go over the SRS document one last time to ensure that every requirement is met.

Regarding the demo with Professor Mosser, Marie-Jean and Diego the team will include the following questions:

- Is the output format what is expected by the competition?
- Are there any requirements not being met within the competition (question for Marie-Jean and Diego)?
- Are there any requirements that are not being met regarding the project rubric from McMaster (question for Professor Mosser)?
- Are there any areas of our code or processes we take that are not allowed in the competition or do not follow the competitions standards?
- What are your thoughts on our topic modeling approach? Do you have any suggestions?
- Are the metrics we use for determining accuracy appropriate in your opinion? If not, what form of metrics should we be using? What level of accuracy would you consider acceptable for this metric?
- In your opinion is the run time of our program to output the results acceptable? If not, what would be an acceptable time?

4 System Test Description

4.1 Tests for Functional Requirements

Our main requirements for the system are built around achieving an acceptable level of accuracy in our system's predictions.

4.1.1 Metrics

The following metrics will be used to evaluate the accuracy of the model for each task. These metrics were selected based on recommendations of the eRisk competition:

Task 1:

1. Reciprocal Rank (RP) is a measure used to evaluate the effectiveness of a ranked list of items. It is often used in the context of question answering or information retrieval systems.

$$RP = \frac{1}{\text{rank}}$$

2. Normalized Discounted Cumulative Gain (NDCG) is a metric used to evaluate the quality of a ranked list of items, considering both the relevance and the position of each item in the list:

$$DCG = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$IDCG = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$NDCG = \frac{DCG}{IDCG}$$

Task 2:

1. F-Score was used in the UQAM team's 2021 submission for the detection of problem gambling task (Maupomé et al., 2021). F-Score is a combination of two other measures, precision and recall, and is used to measure the accuracy of positive/negative tests while taking into account false positive and negatives:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$\text{F-score} = 2\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Task 3:

The details of the metrics used for task 3 can be found in the SRS: section S2 task 3. In light of this only a brief overview will be provided here.

1. MZOE evaluates what proportion of the predictions are incorrect, smaller is better.

$$MZOE(f, Q) = \frac{|\{q_i \in Q : R(q_i) \neq f(q_i)\}|}{|Q|}$$

2. MAE evaluates the average error in the predictions, smaller is better.

$$MAE(f, Q) = \frac{\Sigma_{q_i \in Q}|R(q_i) - f(q_i)|}{|Q|}$$

3. The restraint subscale measures specific ED symptoms by looking at specific questions in the questionnaire, smaller is better.

$$RMSE(f, U) = \sqrt{\frac{\Sigma_{u_i \in U}(R_{RS}(u_i) - f_{RS}(u_i))^2}{|U|}}$$

4. The eating concern subscale measures specific ED symptoms by looking at specific questions in the questionnaire, smaller is better.

$$RMSE(f, U) = \sqrt{\frac{\Sigma_{u_i \in U}(R_{ECS}(u_i) - f_{ECS}(u_i))^2}{|U|}}$$

5. The shape concern subscale measures specific ED symptoms by looking at specific questions in the questionnaire, smaller is better.

$$RMSE(f, U) = \sqrt{\frac{\Sigma_{u_i \in U}(R_{SCS}(u_i) - f_{SCS}(u_i))^2}{|U|}}$$

6. The weight concern subscale measures specific ED symptoms by looking at specific questions in the questionnaire, smaller is better.

$$RMSE(f, U) = \sqrt{\frac{\Sigma_{u_i \in U}(R_{WCS}(u_i) - f_{WCS}(u_i))^2}{|U|}}$$

7. The global ED is a combination of the previous four subscales that attempts to capture the overall ED symptomology in an individual, smaller is better.

$$RMSE(f, U) = \sqrt{\frac{\Sigma_{u_i \in U}(R_{GED}(u_i) - f_{GED}(u_i))^2}{|U|}}$$

4.1.1.1 Acceptable Error Metric Values

The bounds for which error measures are considered acceptable will be based on a combination of the results other teams achieved in past eRisk competitions and a comparison against certain baselines like guessing an average answer for every prediction. Our model should perform better than previous years and outperform baselines in at least one metric.

4.1.2 Methodology

Models need to be evaluated on data they have not "seen" yet. A model performing well on data it has been trained on does not mean it will perform well on the general problem it's attempting to solve, it may just be good at predicting on that data. As such it is necessary to reserve some of the data provided by eRisk for the models to be evaluated on after being trained on the rest of the data. This is known as a train/test split. Depending on the time it takes to train, a model may be trained and then evaluated on multiple splits in order to try and control for variations in accuracy metrics produced by differences in what data is in the train split and what data is in the test split.

4.1.3 Tests

4.1.3.1 Tests for Input

These tests check that the system can take in both xml and json data.

1. FRT-INP-1

   Control: Automatic

   Initial State: No models have been trained by the system yet, input data is available to the system.

   Input: Data matching the organizational structure of eRisk competition Data in xml formatting

   Output: The model will run to completion and return all required outputs.

   Test Case Derivation: The system should run all functions and statements without halting runtime or throwing any errors.

   How test will be performed: The appropriate xml data will be fed into the system, the system should complete run without throwing any errors.

2. FRT-INP-2

   Control: Automatic

   Initial State: No models have been trained by the system yet, input data is available to the system.

   Input: Data matching the organizational structure of eRisk competition Data in json formatting

   Output: The model will run to completion and return all required outputs.

   Test Case Derivation: The system should run all functions and statements without halting runtime or throwing any errors.

   How test will be performed: The appropriate jsonl data will be fed into the system, the system should complete run without throwing any errors.

4.1.3.2 Task 1: Search for Symptoms of Depression

These tests check that the results for task 1 are in a format suitable for eRisk and that the accuracy of the model improves on prior years.

1. FRT-T1-1

   Control: Manual

   Initial State: No models have been trained by the task 1 system yet, input data as jsonl files are available

   Input: The Task 1 Input data

   Output: The system will extract required sentences from the jsonl files and output data that is appropriate to be fed into Task 1.

   Test Case Derivation: The system will receive its input data, process then parse according to the Task's input format.

   How test will be performed: The tester will run the parser with the given jsonl files as input and compare the parsed data.

2. FRT-T1-2

   Control: Manual

   Initial State: No models have been trained by the task 1 system yet, past years input data, training data, and golden truth values are available

   Input: The Task 1 Training Data, Input data, and Golden Truth Values from a past year eRisk Competition

   Output: The system will output sentence ranking and diagnostic metrics based on the input data.

   Test Case Derivation: The system will receive its training data and input data corresponding to a prior year. It's analysis will output RP and NDCG accuracy metrics.

   How test will be performed: The system should derive its training data and feature sets based on correlation to 21 depression symptoms of beck depression index. The outputted metrics will be compared to golden truth data to ensure the accuracy is within the desired bounds and is improving on prior year implementation.

3. FRT-T1-3

   Control: Manual

   Initial State: No models have been trained by the task 1 system yet, input data is available

   Input: The Task 1 Input data

   Output: The system will output sentence ranking and diagnostic metrics based on the input data to a txt file.

   Test Case Derivation: The system will make an analysis based off the provided input data and return each prediction to a test file in a specified formatting.

   How test will be performed: The system's output txt file will be verified that each entry is on its own line in the format "{symptom_number}, Q0, {sentence-id}, {position_in_ranking}, {score}, {system_name}".

4.1.3.3 Task 2: Early Detection of Signs of Anorexia

These tests check that the results for task 2 are in a format suitable for eRisk.

1. FRT-T2-1

   Control: Manual

   Initial State: No models have been trained by the task 2 system yet, input data is available

   Input: The Task 2 Input data consisting of user posts

   Output: The system will output sentence ranking and diagnostic metrics based on the input data to a txt file.

   Test Case Derivation: The system will make an analysis based off the provided input data and return each prediction to a test file in a specified formatting.

How test will be performed: The system's output txt file will be verified that each entry is on its own line in the format "{Username} {prediction}".

2. FRT-T2-2

   Control: Manual

   Initial State: The task 2 system has received input data on a given individual and made a corresponding prediction.

   Input: More data corresponding to the same user analyzed in the input state.

   Output: The system will output an updated prediction taking into account all data provided.

   Test Case Derivation: The system should create a new prediction for the chosen user taking into account all data provided.

   How test will be performed: The system should provide a prediction based off the original data input, after receiving new data, the system should combine posts made by the same user to create a new prediction.

4.1.3.4 Task 3: Early Detection of Signs of Anorexia

These tests check that the results for task 3 are in a format suitable for eRisk and that the accuracy of the model outperforms a trio of simple baseline strategies.

1. FRT-T3-1

   Control: Manual

   Initial State: The task 3 model has been trained on a section of the training data provided by eRisk

   Input: A section of the training data that the model has not yet seen

   Output: The system will output predictions to a txt file where each line is of the format "{Username} {prediction for Q1} ... {prediction for Q28}"

Test Case Derivation: The system should output predictions in the format required for the eRisk competition

How test will be performed: A tester will run the system on the input data and verify that an output file is created and that it is of the correct format

2. FRT-T3-2

   Control: Manual

   Initial State: The task 3 model has been trained on a section of the training data provided by eRisk

   Input: A section of the training data that the model has not yet seen

   Output: The system will output predictions to a txt file where each prediction should be between 0 and 6

   Test Case Derivation: The system should output predictions that conform to the questions on the EDE-Q questionnaire

   How test will be performed: A tester will run the system on the input data and verify that none of the predictions in the output file are less than 0 or greater than 6

3. FRT-T3-3

   Control: Manual

   Initial State: The task 3 model has been trained on a section of the training data provided by eRisk and accuracy measures have been calculated for a baseline strategy where a score of 0 is guessed for every question

   Input: A section of the training data that the model has not yet seen

   Output: The system will print accuracy metrics for the model to the console where the model scores higher than the baseline strategy on MAE or ties MAE and improves on at least one other metric

   Test Case Derivation: If the system fails to outperform a simple guess 0 strategy on at least one metric it is highly likely it does not have any actual prediction power

How test will be performed: A tester will run the system on the input data and verify that the model outperforms the baseline

4. FRT-T3-4

   Control: Manual

   Initial State: The task 3 model has been trained on a section of the training data provided by eRisk and accuracy measures have been calculated for a baseline strategy where a score of 6 is guessed for every question

   Input: A section of the training data that the model has not yet seen

   Output: The system will print accuracy metrics for the model to the console where the model scores higher than the baseline strategy on MAE or ties MAE and improves on at least one other metric

   Test Case Derivation: If the system fails to outperform a simple guess 6 strategy on at least one metric it is highly likely it does not have any actual prediction power

   How test will be performed: A tester will run the system on the input data and verify that the model outperforms the baseline

5. FRT-T3-5

   Control: Manual

   Initial State: The task 3 model has been trained on a section of the training data provided by eRisk and accuracy measures have been calculated for a baseline strategy where the average score for a question based on the training data is guessed

   Input: A section of the training data that the model has not yet seen

   Output: The system will print accuracy metrics for the model to the console where the model scores higher than the baseline strategy on MAE or ties MAE and improves on at least one other metric

   Test Case Derivation: If the system fails to outperform a simple guess average strategy on at least one metric it is highly likely it does not have any actual prediction power

How test will be performed: A tester will run the system on the input data and verify that the model outperforms the baseline

4.2 Tests for Nonfunctional Requirements

4.2.1 Usability Requirements

1. NFRT-U-1

   Type: Dynamic, Manual

   Initial State: System is completed and trained

   Input/Condition: The system will be ran on a Windows desktop/laptop and macOS desktop/laptop device with the correct environment setup

   Output/Result: The system should generate an expected result

   How test will be performed: After setting up the environment on to both macOS and Windows machines, the system will be ran on both environment with the same command. Test will pass as long as an expected result is generated from both machines.

4.2.2 Safety and Security Requirements

1. NFRT-SS-1

   Type: Manual, Dynamic

   Initial State: The system has been trained and is awaiting data to form predictions on.

   Input/Condition: A set of data that the system can predict on.

   Output/Result: The resulting predictions, in a form where no sensitive data from the input is present in the output.

   Test Case Derivation: Sensitive in this case refers to the post history of the subjects eRisk provides as training and test data. If these posts can be reconstructed from any part of our system outputs it is possible the identity of the individual could be discovered. This represents an unacceptable breach of privacy and can not happen.

How test will be performed: After the system has produced it's predictions

4.2.3 Legal Requirements

1. NFRT-L-1

   Type: Static, Manual

   Initial State: The source code and documentation is prepared

   Input/Condition: The user reviews the entirety of the source code and related documentation

   Output/Result: Copyright licenses, appropriate credits and/or MIT license must attached

   How test will be performed: The testers will review the entirely of the project and check to see if appropriate copyright licenses and/or credits are included for resources that require them

4.3 Traceability Between Test Cases and Requirements

Requirement traceability from S2 of SRS to testing in this document.

Table 1: Traceability Between Functional Test Cases and Functional Requirements, T1FR-1 to T3FR-4

Test IDs	**Functional Requirement IDs**									
	T1FR-1	**T1FR-2**	**T1FR-3**	**T2FR-1**	**T2FR-2**	**T2FR-3**	**T2FR-4**	**T3FR-1**	**T3FR-2**	**T3FR-3**
FRT-INP-1	X			X				X		
FRT-INP-2	X			X				X		
FRT-T1-1	X									
FRT-T1-2		X								
FRT-T1-3			X							
FRT-T2-1					X		X			
FRT-T2-2						X				
FRT-T3-1									X	X
FRT-T3-2									X	X
FRT-T3-3										
FRT-T3-4										
FRT-T3-5										

Table 2: Traceability Between Functional Test Cases and Functional Requirements, T3FR-4 to T3FR-6

Test IDs	**Functional Requirement IDs**		
	T3FR-4	**T3FR-5**	**T3FR-6**
FRT-INP-1			
FRT-INP-2			
FRT-T1-1			
FRT-T1-2			
FRT-T1-3			
FRT-T2-1			
FRT-T2-2			
FRT-T2-1			
FRT-T3-2			
FRT-T3-3	X		
FRT-T3-4		X	
FRT-T3-5			X

Table 3: Traceability Between Non-Functional Test Cases and Non-Functional Requirements

Test IDs	**Non-Functional Requirement IDs**

	GR1	GR2	SR1	SR2
NFRT-U-1	X			
NFRT-SS-1				X
NFRT-L-1		X		

5 Unit Test Description

[This section should not be filled in until after the MIS (detailed design document) has been completed. —SS]

[Reference your MIS (detailed design document) and explain your overall philosophy for test case selection. —SS]

[To save space and time, it may be an option to provide less detail in this section. For the unit tests you can potentially layout your testing strategy here. That is, you can explain how tests will be selected for each module. For instance, your test building approach could be test cases for each access program, including one test for normal behaviour and as many tests as needed for edge cases. Rather than create the details of the input and output here, you could point to the unit testing code. For this to work, you code needs to be well-documented, with meaningful names for all of the tests. —SS]

5.1 Unit Testing Scope

[What modules are outside of the scope. If there are modules that are developed by someone else, then you would say here if you aren't planning on verifying them. There may also be modules that are part of your software, but have a lower priority for verification than others. If this is the case, explain your rationale for the ranking of module importance. —SS]

5.2 Tests for Functional Requirements

[Most of the verification will be through automated unit testing. If appropriate specific modules can be verified by a non-testing based technique. That can also be documented in this section. —SS]

5.2.1 Module 1

[Include a blurb here to explain why the subsections below cover the module. References to the MIS would be good. You will want tests from a black box perspective and from a white box perspective. Explain to the reader how the tests were selected. —SS]

1. test-id1

Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

2. test-id2

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input:

   Output: [The expected result for the given inputs —SS]

   Test Case Derivation: [Justify the expected value given in the Output field —SS]

   How test will be performed:

3. ...

5.2.2 Module 2

...

5.3 Tests for Nonfunctional Requirements

[If there is a module that needs to be independently assessed for performance, those test cases can go here. In some projects, planning for nonfunctional tests of units will not be that relevant. —SS]

[These tests may involve collecting performance data from previously mentioned functional tests. —SS]

5.3.1 Module ?

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

   Type: Automatic, Dynamic

   Initial State: The system has been trained and is awaiting data to form predictions on.

   Input/Condition: A set of data that the system can predict on.

   Output/Result: The resulting predictions, in a form where no sensitive data from the input is present in the output.

   Test Case Derivation: Sensitive in this case refers to the post history of the subjects eRisk provides as training and test data. If these posts can be reconstructed from any part of our system outputs it is possible the identity of the individual could be discovered. This represents an unacceptable breach of privacy and can not happen.

   How test will be performed: After the system has produced it's predictions a script will be run that takes all posts in the input data, forms a string out of all consecutive three word triplets (ie. "hello my good friend" forms "hello my good" and "my good friend"), both with and without processing (removal of stopwords, punctuation, etc.), and scans the output to ensure that none of these triples are present.

5.3.2 Module ?

...

5.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

References

Michael Breau, Matthew Curtis, Benjamin Chinnery, Jessica Dawson, and Yaruo Tian. Cognitive care crew system requirements specification, 2023. URL https://github.com/MichaelBreau/nlp-mentalhealth/blob/main/docs/SRS-Meyer/index.pdf.

Diego Maupomé, Maxime D. Armstrong, Fanny Rancourt, Thomas Soulas, and Marie-Jean Meurs. Early detection of signs of pathological gambling, self-harm and depression through topic extraction and neural networks. In *Conference and Labs of the Evaluation Forum*, 2021. URL https://api.semanticscholar.org/CorpusID:237298571.

6 Appendix

6.1 Symbolic Parameters

None.

6.2 Reflection

1. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage etc. You should look to identify at least one item for each team member.

   The following knowledge and skills will need to be acquired by the team to sucesfully complete the verification and validation for the project.

   1. Github Actions
   2. Pytest
   3. Mental Health Diagonostic Basics (For creating expected values)
   4. Python script for parsing and checking files
   5. System performance diagonistic skills

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

   1. Michael will acquire the skill/knowledge required for this skill due to being the owner of the repository. Approaches for this skill/knowledge can be to either research the topic on the github actions webpage or watch YouTube tutorials to find the required skillset.

   2. Jessica will acquire the skill/knowledge required for this skill due to being interested in the skill. Approaches for this skill/knowledge can be to either research the topic on the pytest documention webpage or watch YouTube tutorials to find the required skillset.

   3. Ben will acquire the skill/knowledge required for this skill due to being interested in the skill. Approaches for this skill/knowledge can

be to either research the topic with various papers or watch YouTube lectures to learn more about the required skillset.

4. Matthew will acquire the skill/knowledge required for this skill due to being interested in the skill. Approaches for this skill/knowledge can be to either research the skill using the documentation of some python libraries for parsing documents or watch YouTube tutorials to find the required skillset.

5. Yaruo will acquire the skill/knowledge required for this skill due to being interested in the skill. Approaches for this skill/knowledge can be to either research the topic on various papers or watch YouTube tutorials to find the required skillset.