Classification of Research Articles Using NLP and Machine learning

**Software Requirements Specification**

Version 1.0



**Group Id: S2302A8683 (MC220204248)**

**Supervisor Name : Saad Ahmed**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 24/6/2023 | 1.0 | Introduction of the project | MC220204248 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope of Project](#scope)
2. [Functional and Non-functional Requirements](#FRNFR)
3. [Use Case Diagram](#UCD)
4. [Usage Scenarios](#UCS)
5. [Work Plan](#Gantt)

**SRS Document**

**Scope of Project**

**Project justification:** In this project, research articles will be classified into six categories (Computer Science, Physics, Mathematics, Statistics, Quantitative Biology, Quantitative Finance) based on their title and abstract. Four Machine Learning (ML) algorithms will be used to train the Natural Language Processing (NLP) model. By recognizing key features of the technical jargon, the model will be able to classify any research paper into most suitable category. This will make it easier to classify large volumes of research articles without having to read each one separately.

**Project goals/ deliverables:** The goal of the project is to train and deploy a Natural Language Processing model for classification of research articles. It has the following deliverables:

1. Analyze and preprocess data: The dataset will be imported for training and testing of the NLP model. In order to select high quality data, Exploratory Data Analysis (EDA) and preprocessing will be performed to analyze patterns of each research category.
2. Train NLP model: The NLP model will be trained using four supervised machine learning techniques and two optional deep learning techniques.
3. Test the trained NLP model: The trained model will be evaluated using test data and the trained model will be saved for future use.
4. Make user interface to interact with the trained NLP model: A user interface window will be developed and the trained NLP model will be integrated with user interface. This will allow the user to input research article data. This data will be used by the NLP model to predict the research article category of the new research article.

**Functional and non-functional Requirements:**

**Functional requirements:**

* Import the dataset: The dataset of research articles will be imported into the system. A large dataset will be used to differentiate between unique terms of each research category.
* Perform Exploratory Data Analysis (EDA): The system will display the summary statistics, trends, patterns, and insights on the data visually by performing Exploratory Data Analysis.
* Preprocess the data: The dataset will be cleaned to identify missing values, inconsistencies and duplicates. Any punctuation and stop words will also be removed. Then, the words will then be converted into lower case for tokenization and lemmatization. Hence, the data will be transformed to find distinguishing attributes of each research category.
* Split the data into train and test: The dataset will be divided into test and train subsets. The training subset should be large enough to make the distinguishing features of research category very prominent.
* Use supervised machine learning and deep learning algorithms to train the data: The system will use four machine learning algorithms (logistic regression, random forest, Naïve Bayes (NB), Support Vector Machine (SVM) and K-nearest neighbor (KNN), AdaBoost, Quadratic Discriminant Analysis (QDA) etc.) and two optional deep learning algorithms (Bidirectional Encoder Representations from Transformers (BERT), Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN) etc.) for training.
* Evaluate the trained model on the test data. The test subset will be used to test the trained model. If the model is overfitted/ underfitted, it will not be able to classify new research articles correctly.
* Save the trained model: The trained model will be saved for future use. Hence, the user will be able to use the intelligent model for classification of new research articles.
* Develop User Interface window: The interface window will be created using GUI Python Library or a Python web framework (Django, Flask, FastApi, StreamLite etc.) to allow the user to communicate with the trained NLP model.
* Integrate the trained model with the user interface: The interfacing of trained NLP model will the user interface will allow the user to easily input test data and analyze classification results.
* Test user interface: The user will be provided an option to interact with the system, by first entering, and predicting the category of research article.

**Non-functional requirements:**

* Accuracy: The NLP model will be considered accurate if the ratio of correct predictions to the total number of tests is high.
* Adaptability: The NLP model will be considered adaptable if it can make accurate predictions for different but related research articles.
* Completeness: The NLP model will be considered complete if it is trained using a comprehensive set of research article data, and it can address all the information requirements of each research category.
* Consistency: The NLP model will be considered consistent if a series of classification tests of the same research article using the same method produce similar results.
* Efficiency: The NLP model will be considered efficient if it can perform the classification task in minimal time and effort.
* Interpretability: The NLP model will be considered interpretable if it is easy to extract classification results from the model concerning test research articles.
* Interoperability: The NLP model will be considered interoperable if it can communicate effectively with the user interface to receive test data input, and output classification results.
* Maintainability: The NLP model will be considered maintainable if it can be easily modified to correct faults, improve performance or learn new knowledge.
* Performance: The NLP model will be considered to have high performance if it can perform classification tasks within defined time or memory constraints.
* Portability: The NLP model will be considered portable if it can be easily transferred from one environment to another.
* Reliability: The NLP model will be considered reliable if it has a high probability of performing without failure for a large number of research articles.
* Reusability: The NLP model will be considered reusable if model can be used for similar but different classifications.
* Safety: The NLP model will be considered safe if it is free from failures that can cause the system to crash.
* Testability: The NLP model will be considered testable if it supports testing by input data and ensures the visibility of results and failures.
* Usability: The NLP model will be considered usable if users can effectively learn and use the system.

**Use Case Diagram**

The use case diagram of the research article classification system is shown in Figure 1 below. It explains the sequence of events to train, test and deploy the NLP classification model There are nine use cases in total.

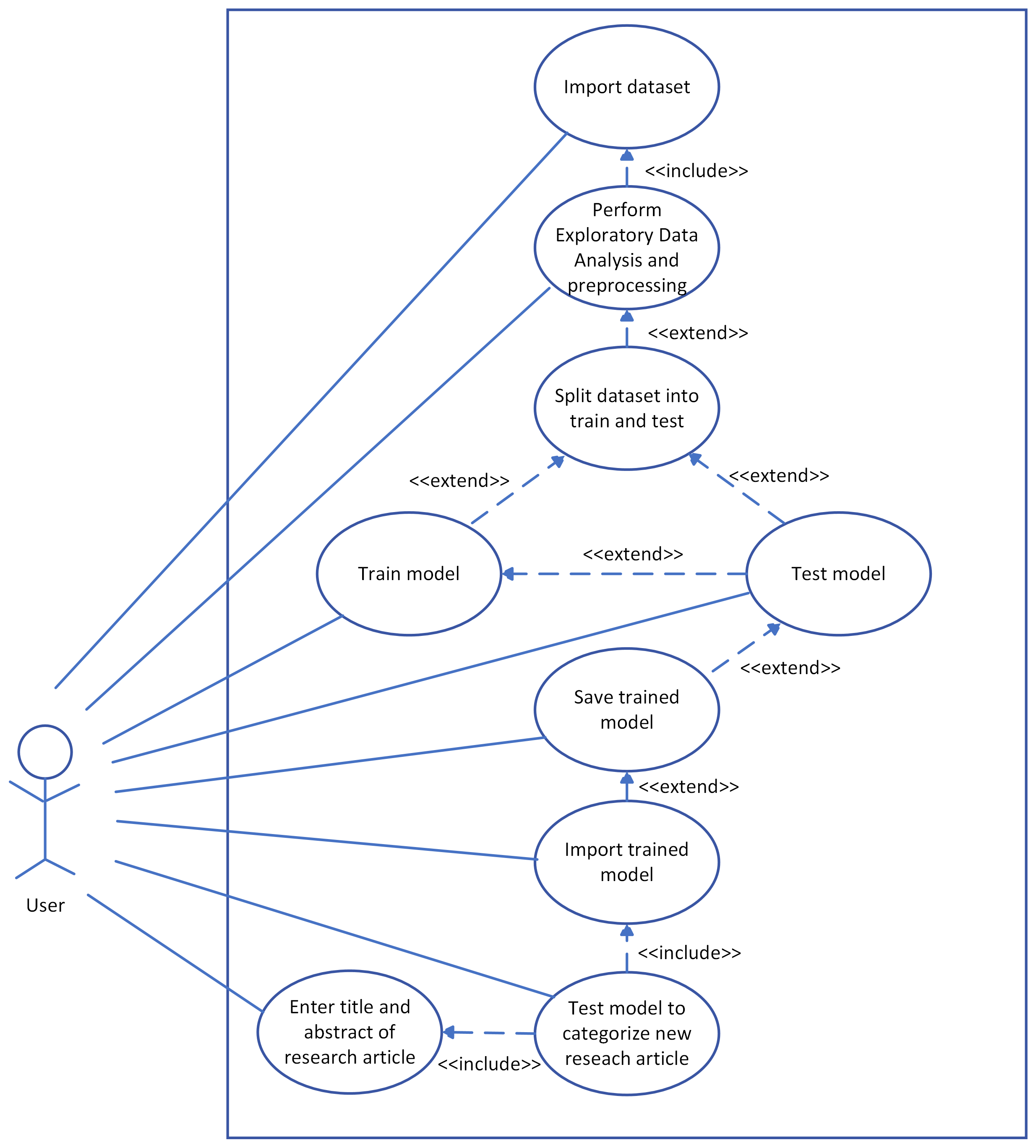


Figure 1: Use case diagram.

Each use case is explained in detail in the next section.

**Usage Scenarios:** The nine usage scenarios shown in the Figure 1 are explained in this section.

1. Import dataset: This use case involves importing dataset for training and testing of the research article classification model

|  |  |
| --- | --- |
| **Use case title** | Import dataset |
| **Use case ID** | 1 |
| **Description:** The admin will import the dataset for training and testing the research article classification model. | |
| **Pre-conditions:**   1. The data file must be properly formatted so that the system can read and parse the content of file. 2. The data file must be placed in an appropriate location so that it is accessible to the system. | |
| **Task sequence** | **Exceptions** |
| 1. Place the data file in the workspace directory. | If the file address is incorrect, the system will be unable to read the data file. |
| 1. Import data file in the program. | The file reader may fail to read file if the file is corrupt. |
| 1. Parse the content of file to read the article ID, title, abstract and category. | The parser may fail to parse the content if the data is not correctly formatted. |
| **Post-conditions:** The data of research articles will be successfully imported in the system. Hence, the ID, title, abstract and category of each research article will be stored successfully in the program. The system can move on to perform exploratory data analysis use case. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

1. Perform Exploratory Data Analysis and preprocessing: This use case involves performing Exploratory Data Analysis (EDA) to display the summary statistics, trends, patterns, and insights on the dataset.

|  |  |
| --- | --- |
| **Use case title** | Perform Exploratory Data Analysis and preprocessing |
| **Use case ID** | 2 |
| **Description:** The system will perform Exploratory Data Analysis to display the summary statistics, trends, patterns, and insights on the data. | |
| **Pre-conditions:**   1. The import dataset use case must be completed; hence the data file must be imported in the system. | |
| **Task sequence** | **Exceptions** |
| 1. Perform Exploratory Data Analysis to understand properties of data e.g., statistical properties, missing values, correlation and occurrences of technical words in research articles of same category etc. | If the training dataset is not of high quality, the document classification model will not be able to categorize the research articles efficiently. It is difficult to find unique features of each research article category. |
| 1. Display the summary statistics of Exploratory Data Analysis using graphs and charts to list the unique features of each research article category. | A lot of data analysis is needed to understand distinguishing features of different categories of research articles. If the Exploratory Data Analysis is not properly done, the model will not perform well. |
| 1. The dataset will be processed to identify missing values, inconsistencies and duplicates. Any punctuation and stop words will also be removed. Then, the words will then be converted into lower case for tokenization and lemmatization. | If stemming and lemmatization is not done properly, the |
| **Post-conditions:** The system can move on to split dataset into train and test use case. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

1. Split dataset into test and train: This use case involves splitting the dataset into test and train subsets.

|  |  |
| --- | --- |
| **Use case title** | Split dataset into test and train |
| **Use case ID** | 3 |
| **Description:** The system will split the dataset into test and train datasets. | |
| **Pre-conditions:**   1. The perform Exploratory Data Analysis use case must be completed; hence high-quality data must be available for testing and training of the model. | |
| **Task sequence** | **Exceptions** |
| 1. The system will split the dataset into two categories: train and test. It is important to have sufficient data for training so that the classifier can distinguish unique features of each research article category. | A lot of data is needed for training of the model. The training data must be of high quality so that the research articles can be classified efficiently. |
| **Post-conditions:** The system can move on to train model using four algorithms use case. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

1. Train model: This use case involves training of the research article classification model using four machine learning algorithms and two optional deep learning algorithms.

|  |  |
| --- | --- |
| **Use case title** | Train model |
| **Use case ID** | 4 |
| **Description:** The system will train the research article classification model using four machine learning algorithms. | |
| **Pre-conditions:**   1. The split dataset into train and test use case must be completed; hence high-quality data must be available for training of the model. | |
| **Task sequence** | **Exceptions** |
| 1. The system will train the research article classification model using four machine learning algorithms (logistic regression, random forest, Naïve Bayes (NB), Support Vector Machine (SVM) and K-nearest neighbor (KNN), AdaBoost, Quadratic Discriminant Analysis (QDA) etc.) and two optional deep learning models (Bidirectional Encoder Representations from Transformers (BERT), Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN) etc.). The best algorithm will give the most accurate predictions of research article category. | A lot of training is needed for accurate classification of research articles. The unique features of one research article category must not be strongly correlated to another category. If some features are common among multiple categories or if the training data size is small, the model may give incorrect results. |
| **Post-conditions:** The system can move on to save trained model use case. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

1. Save trained model: This use case involves saving the trained research article classification model.

|  |  |
| --- | --- |
| **Use case title** | Save trained model |
| **Use case ID** | 5 |
| **Description:** The system will save the trained research article classification model. | |
| **Pre-conditions:**   1. The train model using four algorithms use case must be completed; hence the model must have high prediction accuracy of research article classification. | |
| **Task sequence** | **Exceptions** |
| 1. The system will save the trained model for future use. The model must be trained using multiple algorithms and the best model must be saved. If several models give accurate results, they can all be saved. | If the accuracy of some model is very high, it may be due to overfitting. The model will perform very well for train dataset but it will fail to fit additional data or make reliable future predictions. |
| **Post-conditions:** The system can move on to test model using testing data use case. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

1. Test model: This use case involves testing the trained research article classification model using test dataset.

|  |  |
| --- | --- |
| **Use case title** | Test model |
| **Use case ID** | 6 |
| **Description:** The system will test the trained research article classification model using test dataset. | |
| **Pre-conditions:**   1. The save trained model use case must be completed; hence the most accurate model must be available for research article classification. | |
| **Task sequence** | **Exceptions** |
| 1. The system will test the trained model using test dataset. If the model is highly trained, it will be able to accurately predict the category of new research articles. 2. The accuracy of each model will be evaluated using statistical metrics and scores. | 1. If the model is overfitted/ underfitted, the model will fail to fit test data and make inaccurate predictions. 2. If the test data includes research articles belonging to a category other than the six research categories, the model will not be able to classify the research article correctly. |
| **Post-conditions:** The system can move on to test model using testing data use case. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

1. Import trained model: This use case involves importing the trained research article classification model to the user interface.

|  |  |
| --- | --- |
| **Use case title** | Import trained model |
| **Use case ID** | 7 |
| **Description:** The system will import the trained research article classification model to the user interface. | |
| **Pre-conditions:**   1. The test model using training data use case must be completed; hence the most accurate model must be available for research article classification. | |
| **Task sequence** | **Exceptions** |
| 1. The system will import the trained model into the user interface module. The interface window will be created using GUI Python Library or a Python web framework (Django, Flask, FastApi, StreamLite, etc.). 2. The interface will be designed to make it easy to send input data to the trained model and retrieve classification data results. Hence, the user will be able to communicate easily with the model. | If the model is not properly saved or the file is corrupt, the system will not be able to import the trained model file. |
| **Post-conditions:** The user can move on to enter title and abstract of research article use case. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

1. Enter research article title and abstract: This use case involves entry of the research article title and abstract using the user interface window.

|  |  |
| --- | --- |
| **Use case title** | Enter research article title and abstract |
| **Use case ID** | 8 |
| **Description:** The user will enter the research article title and abstract using the user interface window. | |
| **Pre-conditions:**   1. The import trained model use case must be completed; hence the system must be ready to classify the research article using the most accurate model must be available for research article classification. | |
| **Task sequence** | **Exceptions** |
| 1. The user will enter the research article title and abstract using the user interface window. The system will parse the input text and pass it to the trained model for prediction of research article category. | 1. If the user inputs improperly formatted data, the system will not be able to understand it. 2. If the user inputs text in an unfamiliar language, the system will not be able to process the content of the text. |
| **Post-conditions:** The system can move on to test model to categorize new document use case. | |
| **Authority:** User | |
| **Author:** S2302A8683 (MC220204248) | |

1. Test model to categorize new document: This use case involves testing the trained model to categorize new research article using the information provided by the user.

|  |  |
| --- | --- |
| **Use case title** | Test model to categorize new document |
| **Use case ID** | 9 |
| **Description:** The system will test the trained model to categorize new research article using the information provided by the user. | |
| **Pre-conditions:**   1. The import trained model use case and enter research article title and abstract must be completed; hence the system must be ready to classify the research article using the most accurate model and the model must understand the research article title and abstract data inputted by the user. | |
| **Task sequence** | **Exceptions** |
| 1. The system will test the trained model to categorize new research article using the information provided by the user. By using the features of the tittle and abstract, the trained model will classify the research paper into one of the six available categories. The result of the processing will be sown to the user through the user interface window. | 1. If the research article does not belong to one of the available research article categories, the system may not be able to classify it. 2. If the research article title and abstract has very few of the identified distinguishing features, the model will not be able to categorize the research article accurately. |
| **Post-conditions:** The system will show the results of the research article classification using the user interface window. | |
| **Authority:** Administrator | |
| **Author:** S2302A8683 (MC220204248) | |

**Adopted Methodology**

The model of the adopted project methodology is shown in Figure 2 below. It is a combination of waterfall and spiral methodology. The radial dimension represents the cumulative time and the angular dimension represents the progress of the project. Each phase begins by determining objectives of that phase and performing risk analysis. Because of the spiral nature of development, it is easy to judge how much to test and there is no distinction between development and maintenance.

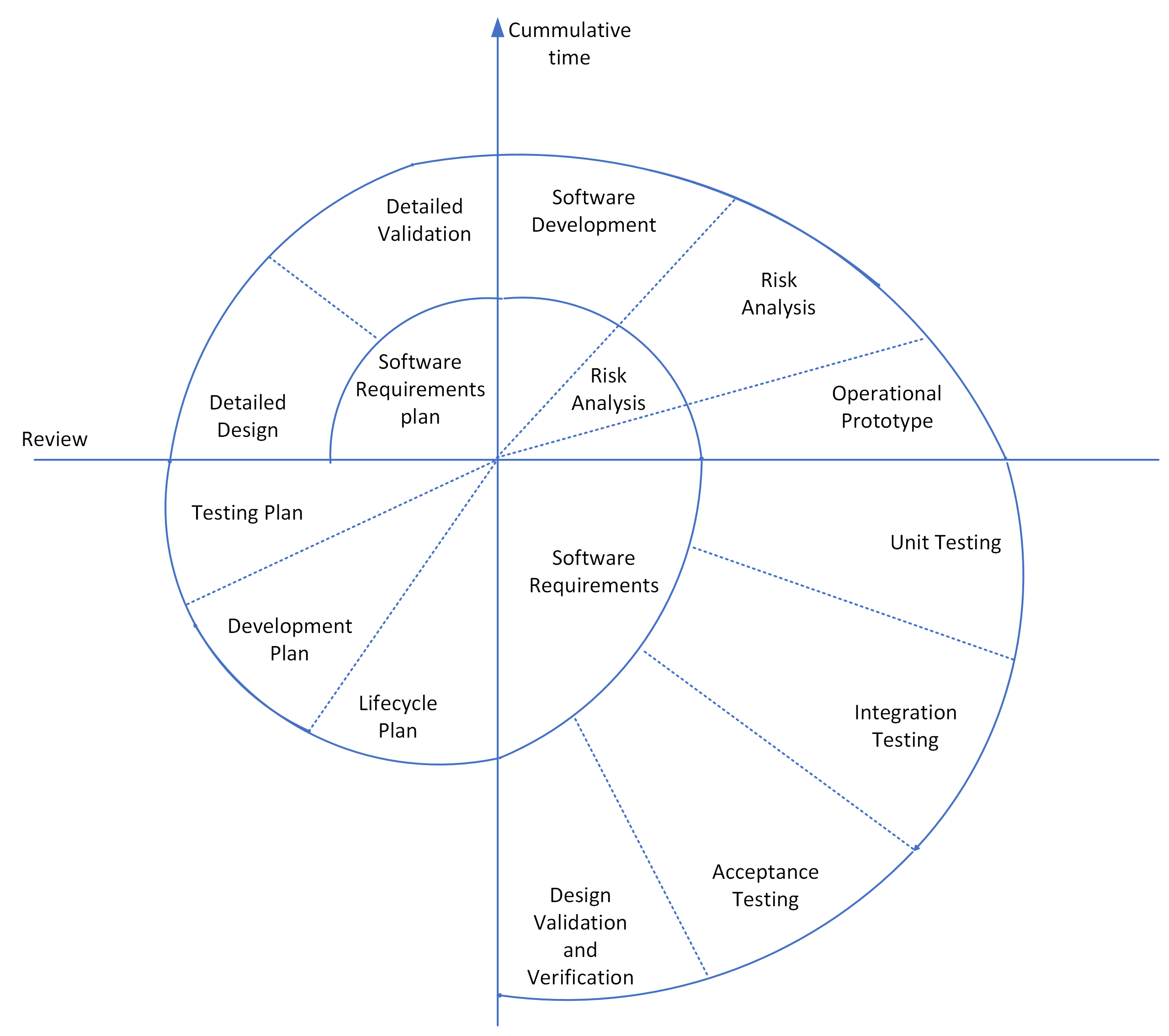


Figure 2: Adopted project methodology.

**Work Plan**

The Gantt chart of the project work plan is shown in Figure 3 below. The project is divided into four phases:

1. Software requirements specification phase (31 May, 2023 - 14 Jul, 2023): This phase includes research and planning to prepare the Software Requirements Specification (SRS) document.
2. Software design phase (15 Jul, 2023 - 13 Sep, 2023): This phase includes performing Exploratory Data Analysis of dataset, training of NLP model and design of user interface window.
3. Prototype phase (14 Sep, 2023 - 4 Oct, 2023): This phase includes the testing of user interface and integrated NLP model for prototype phase viva.
4. Final Deliverable (5 Oct, 2023 - 4 Dec, 2023): This phase includes the testing of complete system and preparation of project report for final submission.

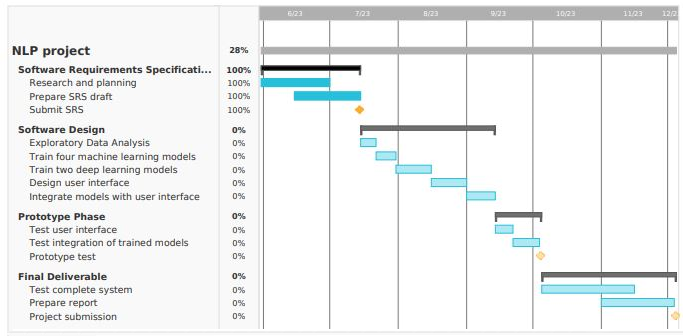


Figure 3: Gantt chart of complete project work plan.