Systems Development Group Project – UFCF7S-30-2

Feeding Dashboard Project.

Jack McDaid	21023808
Charlie Taylor	21024059
Cosmo Elford	21028021
Lauren Pearce	22010935
Matthew Woodhall	21026348

Introduction:

In today's rapidly developing world, research suggests hospitals haven't been quick to adapt to the use of technology, especially in small and urban areas (Vadillo, et al. 2016). This project is to develop an application to bridge the gap between technology, and the healthcare sector, streamlining referrals for patients in critical care units.

By developing a widely compatible, efficient, easy-to-use, small form factor application to assist dietitian's, this could reduce the disparities of technology in the healthcare sector. (Wickramasinghe, et al. 2015)

- Firstly, the improved data capture and analysis. By facilitating the efficient recording and processing of relevant patient information, the application will provide a comprehensive profile on patients in the CCU. This enables healthcare professionals to track trends, identify patterns, and gain valuable insights into individual treatment, interventions, or crucial nutritional support.
- Secondly, the enhanced decision-making through technology. The application will explore the potential of integrating AI, such as algorithms and neural networks, to support decision-making. This involves developing predictive models that identify patients at higher risk of malnutrition, or complications associated with inadequate nutritional intake, thereby proactively prompting dietitian referrals.

By combining user-friendly design with advanced data analysis capabilities, this application has the potential to significantly improve the efficiency and accuracy of the dietitian referral processes in CCUs, ultimately contributing to better patient outcomes and better optimised resource allocation within hospitals. We envision it to be a valuable tool for healthcare professionals and dietitian's; allowing them to provide patients with the best possible care and contributing to an overall more efficient and effective healthcare system.

Aims & Objectives:

Aims:

- To build a system to accurately predict whether an CCU patient needs a referral to a dietician.
- To provide relevant data through an easy-to-use dashboard system.

Objectives:

- Develop the application in the given time period as a team, considering risks and literature studies.
- Learn how to implement a Machine Learning algorithm to the dataset.
- Learn how to create an appropriate GUI using tools such as Tkinter.
- Develop a management pipeline to keep the project on track.
- Communicate to ensure that all project members are kept up to date and that the project stays on track.
- Use learning from the literature review to select appropriate frameworks and models for the dashboard system.

Literature Review:

AI in the healthcare sector:

Machine learning is fast becoming a crucial part in modern healthcare, the most common application of which being precision medicine. It can predict what the correct protocols for treatment will be based on provided variables such as the patients' attributes and the treatment context. (Lee SI et al, 2018). In the case of medical use, a machine learning system would be required to be developed through supervised learning. This would ensure that the algorithm is able to accurately provide predictions and that it would adhere to SLEP guidelines.

Machine learning tools have been used previously in many aspects of healthcare, although they are only recently entering more common use as the technology advances. One such tool is Mia (mammography intelligent assessment) produced by Kheiron medical technologies and London imperial College. (NHS 2021) Mia is tool for analysis in breast cancer screening and is already being used within the NHS as part of a clinical trial across 15 hospitals. Mia is being developed in response to a critical issue, the lack of manpower in NHS, it is part of a two-step operation within the workflow, allowing a reduction in the manpower needed, which in turn allows a better overall service.

There have been many different algorithms applied to the healthcare sector. Many of them fall within standard classification formats such-as SVM or Forest models. There are a few however, that must use more unique approaches. In several cases forms of genetic algorithms have been used to more appropriately schedule staff (Du, G., Jiang, Z. and Yao, Y. 2013). In this case, the staff were able to carry out increased activity without having the workload increased. The same system can also allow for easier transmission of information to patients, giving an easy, automated schedule to everyone thus allowing increased efficiency in every area of the hospital.

In another example of the use of a genetic algorithm (Kramer, A,A. 2021) A genetic algorithm is used to identify patterns associated with mortally ill patients in an ICU. The algorithm was used as a classification device to identify the risk posed to each patient.

This is another example of how, whilst not able to take over any jobs from the healthcare professionals, the AI model can advise and structure treatments in a way to help improve workflow and verify doctors' results. In the case above (Kramer, A,A. 2021), the model was proved effective with each patient being correctly classified leading to either more effective treatment or, in many cases, a correct mortality prediction.

Whilst, as a tool, AI models have been proven to be effective, there are still many holdouts in the industry, especially on a more individual level, who either do not trust, or are unwilling to distance themselves from such critical decisions to be made with the use of AI modelling. A study published in 2022 (Kostopoulou, O., Arora, K. & Pálfi, B. 2022) showed that, when presented with the results of an appropriate model, even doctors who initially doubted the use of the model, would change their mind about 26% of the time. This was predominantly in the case where the doctor believed the patient to be healthy, a False Negative. For the use of AI modelling to be fully effective it needs to be adopted in a more widespread capacity and ethically tested with larger datasets.

Another challenge posed by using AI tools is the transition from evidence, to probability-based machines. This is generally regarded as a positive step; however, it raises challenges of an ethical nature (Aronson S, Rehm H. 2015). These challenges arise through the ethical implications of deciding through purely statistical means, as an AI tool would, as opposed to relying on the experience of trained professionals.

This suggests that, immediately at least, the best use of AI tools in healthcare be as an advisor or as one stage in a verification system, as was the case with Mia.

Ethics of AI in the healthcare sector:

As discussed earlier with Mia, ethics is an important point to consider when using an AI model to advise healthcare. The primary job of most healthcare involves making decisions on the treatment of patients, this can either be on an individual basis, or systemic. With individual decisions, AI can be used to help diagnose and chose effective treatment methods whereas with systemic issues the use of Ai is more in the scheduling and logistical infrastructure. Both aspects can raise ethical issues.

Currently, in the USA, diagnostic errors account for approximately 10% of all patient deaths (National Academies Press. 2015). All medical diagnoses come with a degree of uncertainty, the issue comes when using Ai to help with the diagnostic issues as, whilst it could improve diagnostic accuracy, any mistakes in diagnoses could be attributed to the Ai rather than mistakes or potential incompetence in the clinician. This raises serious ethical issues as discussed earlier.

Many of these issues are already mitigated by using AI modelling as a purely advisory tool, with all final decisions being made by trained healthcare professionals. This allows the healthcare system to benefit from the increased accuracy that Ai can provide whilst still providing an experienced human decision. This finally also allows for a greater level of accountability by having every decision made being traced back to the clinician responsible.

Machine learning model selection:

In the case study being explored the problem to be solved is one of classification. Fortunately, there is a wide array of classification algorithms available for such use (Pimendis, E. et al. 2021).

The first step when choosing a machine learning model to use is the decision between the primary types of algorithms: supervised, semi-supervised, unsupervised and reinforcement. The task is one of classification, and it is this that will enforce the decision as to which model type of algorithm is best to use.

Reinforcement, whilst possible in this case study, is generally unsuited as it relies on consistent improvement and large amounts of trial and error, this creates a system where it will reach the optimal result, but only after first, repeatedly, reaching incorrect results. This means that it would require more supervision, with input needed at several stages in the process and would still not necessarily reach the best solution, even if would create a reasonable one.

Unsupervised machine learning is best used when dealing with large amounts of raw data. It is not suitable for general classification requirements but can be useful when applied to predicting missing data within large datasets. As such, the only unsupervised methods used will be in the filling in of blanks within the datasets used in the system and not for the classification task itself.

Supervised machine learning techniques are generally considered to be the best for classification tasks as they can take input from the user into the weighting of different criteria and variables when carrying out a classification task. The issue is in selecting the correct supervised model to avoid several of the issues that might arise, such as overfitting, scalability and slow processing times.

Semi supervised machine learning models combine both supervised and unsupervised models to solve problems more accurately. This allows accurate classification even when there are large amounts of missing data in the dataset as the unsupervised method, such as KNN, can be used to predict the missing data, and then the supervised method can be more accurately used to carry out the classification aspect of the task. This is likely the best suited method for the case study as it covers both primary aspects the model will need, to predict the missing data and to accurately classify the datasets.

It is tricky to say which machine learning model will be the most accurate and efficient, fortunately it's relatively easy to test several within a short time frame. In this case all models will come from the Scikit-Learn library (Scikit-Learn, 2024). The models selected for the classification are: Random Forest Algorithm, Decision Tree, and various support vector machines with KNN being used to predict missing values in the dataset. In this case, the KNN will be tested with the intent that it predicts gaps in the data provided, and the other method will be tested with the completed dataset.

Each algorithm has its own positives and negatives and, whilst the testing is going to be used to decide the final model used, it is still beneficial to talk about the aspects of the different methods separately.

Random Forest Algorithm (Breiman, L. 2001):

- A supervised machine learning algorithm that utilises multiple trees based on subjects of a given dataset. The trees will then vote on a prediction with the average vote being taken as the answer.
- Low risk of overfitting.
- Low training time.
- High accuracy.
- Can be slow and inefficient.

Decision Tree (Song, Y.Y., Lu, Y. 2015):

- A supervised algorithm that utilises a single tree built from multiple subsets, iterating through these to arrive at a classification.
- Simple to implement.
- Good visual depiction of decision.
- Not overly robust
- Some risk of overfitting

Support Vector Machines (SVM's) (Guo, C. Chou. Y. 2020):

- Easily interpretable results.
- Very accurate when used within its scope.
- Can create unreasonable memory requirements.
- Parameter fitting can create or exacerbate runtime issues

KNN (K-Nearest Neighbours) (Mucherino, A. et al. 2009):

- An unsupervised algorithm that clusters data points into different sets based on similarity.
- Easy to implement
- Fast and efficient.
- Easily scalable.
- Potential memory issues

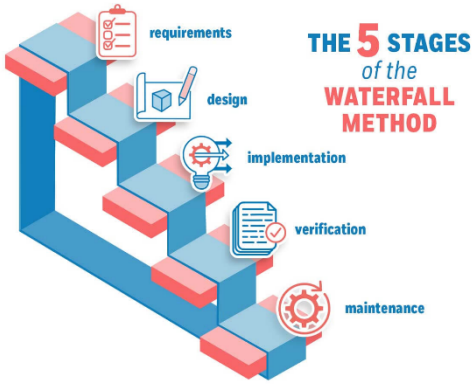
Project Plan:

Methodology:

When planning a large project such as this, it is important to ensure that there is a good framework in place for people to follow, and to ensure that all criteria are met. There are many such frameworks already developed for use in large scale development, these include methods such as Agile, SCRUM etc.

For this project, the primary system that will be used is waterfall. This is a basic system best explained with a visual aid. (see Fig1) (Management Library. 2024) This diagram shows the progression of the project through each sequential stage with every stage being necessary for the following one. If deemed necessary a model like this can be adapted into a more agile development system, with multiple cycles of verification, redesign, and redevelopment. For the scope of this project, however, this was deemed unnecessary as the focus is on the report and process, with the final product being relatively simple.

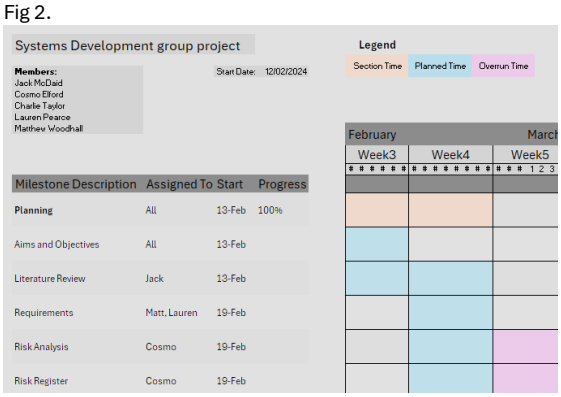
Fig1.



Group Roles:

The primary source of direction for the group was through assigning different tasks on a GANNT chart (see App 1, Fig 2). The tasks were selected as those deemed necessary for the completion of the task in the given time. The GANNT chart denotes all the tasks necessary for the completion of the project, with several of the tasks drawing directly from the objectives. The chart breaks down every major aspect of the project and states the assigned team member as well as the projected time the task will take. It can also consider when tasks overrun.

Fig 2.



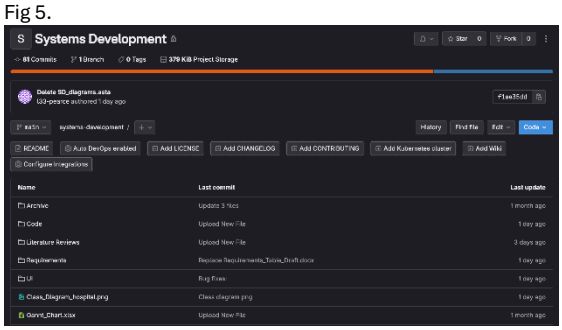
However, as the project team is relatively small, it may be tricky to meet all the requirements on the agenda at set deadlines. Roles matured towards the latter half of the project's development stage, finding a balance between contributions to design, planning and programming.

During non-contact hours, close contact was kept with our peers using WhatsApp and Discord messaging. This allowed quick effective communication of ides, creating an effective work environment that would be flexible around team member's needs.

Version Control:

In addition to the GANNT chart, our project had an information backup, and version control system setup in UWE's GitLab. Here, the team contributed by pushing updates to our private Git repository. Keeping our project securely backed up – this is an industry standard.

Fig 5.



Risk Management:

The team implemented tried and tested ways to manage risks within our project.

A risk matrix was referenced (see Fig 3.) to determine the likelihood of the impact of each risk to the project. A final risk score was calculated by multiplying the values, this would then correspond to the colours as well, giving a visual aid to the level of risks and what to prioritize.

Fig 3.

Likelihood

5	5	10	15	20	25
4	4	8	12	16	20
3	3	6	9	12	15
2	2	4	6	8	10
1	1	2	3	4	5
	1	**2**	**3**	**4**	**5**

Hazard

Additionally, how should the group effectively identify and collect actual potential risks? There are many ways risks can crop up, so there must be a place to collectively store them. Looking at one of the entries in the risk register; (see Fig 4.)

Fig 4.

As follows:

Description, Likelihood, Hazard, Total Risk, Mitigation, Mitigated Risk, Author

Technical risks, including programming, algorithms and the choices of technologies.	2	5	10	As we have multiple programmers working on our project, we need to stay consistent with programming practices, and the technologies we are using. By communicating within our group, we can mitigate this risk.	4	**Cosmo** *Created:* ***18th March 2024***

To view our entire Risk Register (see App 2.)
You can see there is a great overview of the content within that entry to the risk register. By using the risk matrix in combination, members can determine what level our risk is at, and what kind of threat it poses. Finally, there's a description of the mitigation methods and its respective new score after defining them.

Requirements:

Functional:

- Create an accessible user-friendly dashboard - The dashboard must be easy for user to use and understand. This will require use of software/libraries that allow for the creation of a UI, and it should be produced in a way that practicality is paramount.
- Show admitted patients - The relevant data of patients currently in the CCU should be obtained from the given CSV file and displayed to the user. This will require the interface to allow the information to be displayed in a suitable format.
- Show patients who should be referred - The dashboard must be able to filter out patients that should not be referred to a dietitian so their information is not displayed meaning the user can only see the information of patients who should get dietitian referral. This will require the system to be able to select the necessary patients from the given data and display that to the user.
- Flag patients in need of referral - The patients must be displayed in a way where those in need of referral are easily identifiable. This will require the system to know which patients fulfil this and flag them in a way that is convenient to the user.
- Show individual patient data - To fulfil this requirement, the user must be able to select a displayed patient from the patient list to view all individual data relevant to the user. This will require the system to retrieve the information of the patient selected by the user and display it in a visual format.
- Generate reports - When requested by the user, a report will be generated by the system by retrieving all necessary information from the system file and display it in an appropriate format using tables and graphs.

- Uploading User submitted files - To fulfil this, there must be a method for the user to submit CSV files to the system. Once uploaded, the system must be able to retrieve the given data from the file and display it on the dashboard.
- Determining patient referral - When uploading a CSV file, the algorithm API must read patient data to determine whether they need a dietitian referral or not. This new data should also be saved onto the local system.
- Creation of a suitable machine learning algorithm that can accurately predict the referral outcome of a patient based on the data provided in the dataset. Various suitable machine learning algorithms were discussed in the literature review each of these will be tested on the dataset to find the optimal model to implement.

Non-Functional:

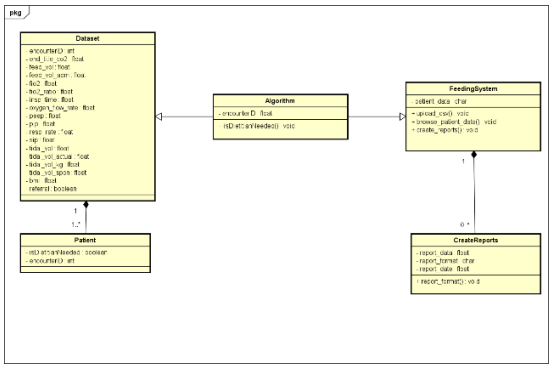
- Data Protection - All patient data should be protected for both ethical and legal reasons, according to The Data Protection Act 2018.
- Compatibility on multiple screen sizes - The system should be able to run smoothly on multiple screen sizes such as tablets, desktop computers and phones. This includes scaling and resizing, making it user friendly.
- Compatibility on multiple operating systems The system should be able to run effectively on different operating systems such as Windows, Linux or MacOS. This could be achieved by choosing an optimal programming interface to base the app on.
- Performance speed - The system should run efficiently with any task. It should take no longer than 60 seconds to display and filter patient data for referral.
- Reliability - The system should run efficiently with minimal faults or crashes. Additional non-functionalities:
- Allow for the user to create notes on patients - This will allow healthcare staff to build profiles on patients, by storing this in a local database, with options to store it in the cloud. This would require the use of different technologies implemented into the app, such as SQLite and pandas for data handling.
- Information graphics visualising data - This will allow healthcare staff to visualise the large data set. This can be in the form of pie charts, bar charts, scatter graphs, etc.

Design Plans:

Class Diagram (see Fig 6):

A class diagram was the obvious choice to visually represent the relationships our system should have. It shows the connections between different classes required to make the system run as desired, as well as any attributes and operations a class may have. An attribute of a class represents a characteristic associated with that class such as the columns included in the provided dataset whereas an operation is an action that class should perform. The algorithm, being a primary feature of the system, should take information from the dataset, process it, and send it to the Feeding System. The system can then generate reports using that data.

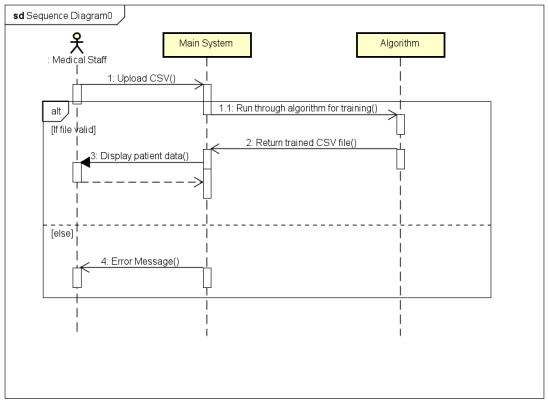
Fig 6.



CSV Uploader (see Fig 7):

One of the sequence diagrams runs through the process of uploading a CSV. Upon uploading, the system validates the file's format and content. If the CSV file is deemed valid, it proceeds to run the data through the algorithm for processing. Subsequently, the system returns the processed patient data, allowing medical staff to access and utilize the insights derived from the analysis. However, if the CSV file is invalid—for instance, if it contains formatting errors or lacks necessary data—the system displays an error message, indicating that the upload process cannot proceed until the issues are addressed.

Fig 7.

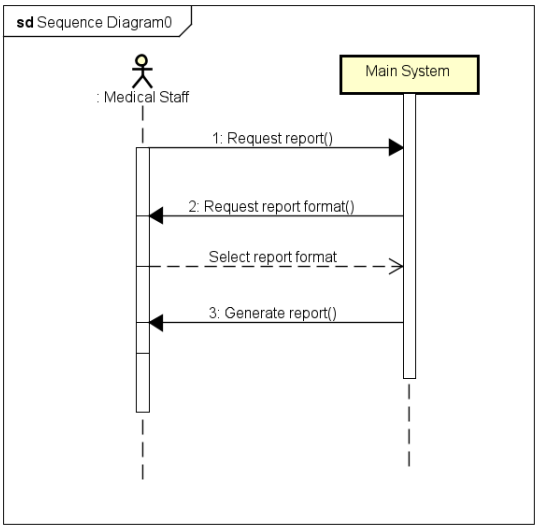


This sequence diagram served as a foundational blueprint for developing the system's functionality, offering a structured guide for implementing the upload process. By visually mapping out the sequence of interactions between users and the system, as well as the system's internal processes, the diagram facilitated a clear understanding of the upload workflow and its various checkpoints. This clarity was invaluable during the development phase, helping streamline the implementation of each step and ensuring that the final system aligns with the intended workflow.

Create Reports (see Fig 8):

Another function of our system that was initially mapped out with a sequence diagram is the generating reports function. The sequence diagram created for the generating reports function outlines the series of steps involved when medical staff request a report within the hospital system application. First, medical staff initiate the process by requesting a report. Upon making the request, they are prompted to select the desired format for the report. Once the format is selected, the system proceeds to generate the report according to the chosen specifications, incorporating relevant patient data and analytics. Finally, the generated report is presented to the medical staff for review or download, depending on the chosen format.

Fig 8.



This sequence diagram served as a valuable guide during the system development process, offering a clear depiction of the steps involved in report generation. By visually mapping out the interactions between users and the system, as well as the system's internal processes, the diagram provided developers with a comprehensive understanding of the report generation workflow. This understanding was crucial for implementing the functionality effectively, ensuring that the system behaves as expected and meets the requirements of medical staff.

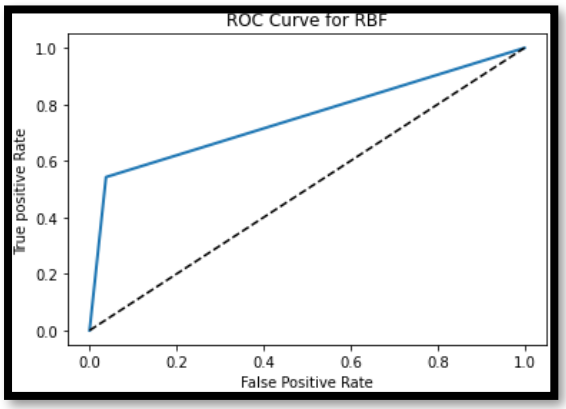
Machine Learning Model:

To select the best model for the machine learning aspect of the project a series of test must be carried out. These tests will sequentially go through a selection of different models as discussed in the Literature review section.

The success of a model will be determined based on the comparison of the confusion matrices of the models as well as the ROC (Receiver operating characteristic) curve. This is a representation that shows the performance of classification models using predefined threshold levels. A standard measurement for how effective a model is the area under the curve of the ROC. Another tool for analysis of the model is the classification report, this gives the same data as the confusion matrix as well as other variables such as the F1 score. These metrics allow a better understanding of the true accuracy of the models, leading to a better selection of which model should be used.

Support Vector Machine (SVM):

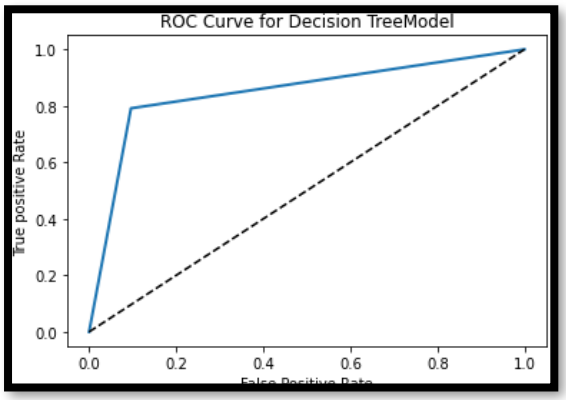
To ease the comparison between models a grid-search is used to compare which SVM model is most appropriate. Only the most accurate SVM will be used to generate the ROC curve and the confusion matrix.



The models tested were Linear, RBF and polynomial. Out of these the RBF (Radial Basis Function performed best, with the area under the curve being 0.7525.

Decision Tree:

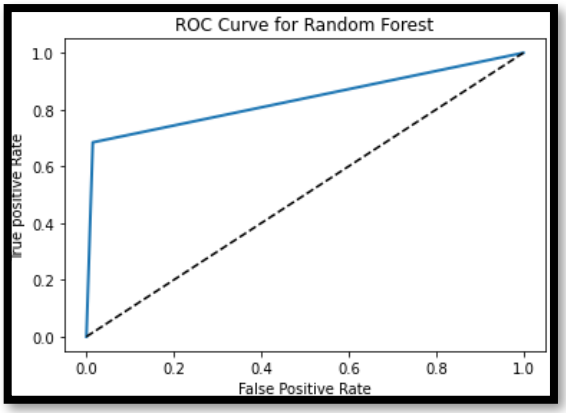
For the decision tree, there were no model variations to test, so the test was confined to a grid-search of the available variables that could be changed.



The ROC curve for the decision tree is decidedly better than that of the SVM with an area under the curve score of 0.8476.

Random Forest:

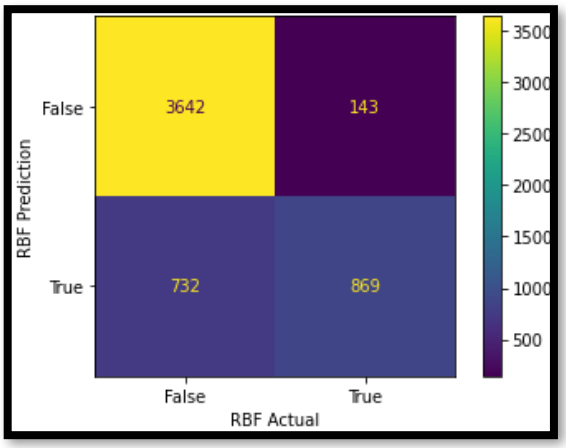
The final model for testing is the decision tree, this again only has the one model to test so the grid-search was applied for the other variables available.

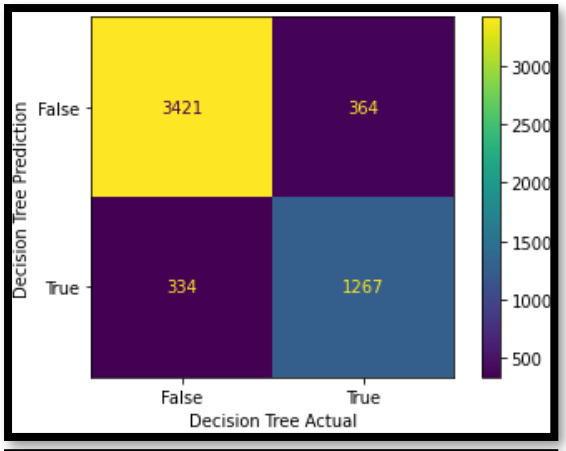


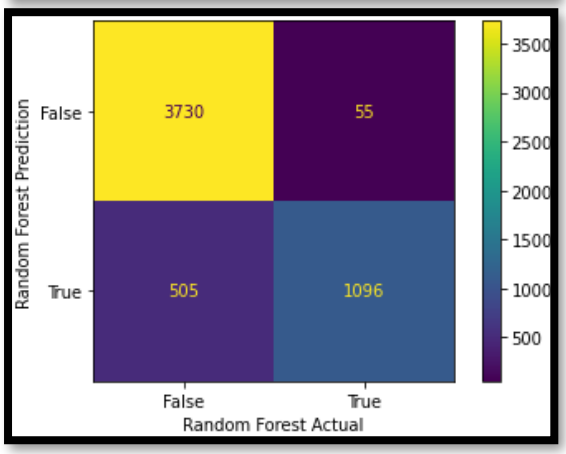
The random forest model gives a slightly worse area under the curve score than the decision tree with a score of 0.8350.

Confusion Matrix Analysis:

From the confusion matrices the key elements that can be viewed are True positives, True negatives, False negatives and False Positives. Of these the most important for analysis are the True Positives and Negatives, as they give the success rate of the model, and the False Negative. The False Negative is likely the most important metric here as, in this case, it shows the patients that would need a referral who would not get one through the algorithm.







The first thing to be gathered from these matrices is how the RBF model falls behind, having the least True results of all the models. The real comparison is between the decision tree and the random forest. The random forest has a slightly lower overall set of False results, although many more of its results are False Negatives. As False Negatives are the most important point here, this would lead the Decision Tree model to be the best to use.

Classification Report Analysis:

A classification report generates 4 key metrics for analysis of the models. These are:

- Precision - the percentage of correct predictions
- Recall - the percentage of correct positive predictions
- F1 - the score of the weighted mean of precision and recall
- Support - the number of occurrences of a class in the dataset

These metrics give further insight into the effectiveness of the models, with the F1 score being of particular importance and support being of none.

RBF:

	precision	recall	f1-score	support
0.0	0.83	0.96	0.89	3785
1.0	0.86	0.54	0.67	1601
accuracy			0.84	5386
macro avg	0.85	0.75	0.78	5386
weighted avg	0.84	0.84	0.83	5386

Decision Tree:

	precision	recall	f1-score	support
0.0	0.91	0.90	0.91	3785
1.0	0.78	0.79	0.78	1601
accuracy			0.87	5386
macro avg	0.84	0.85	0.85	5386
weighted avg	0.87	0.87	0.87	5386

Random Forest:

	precision	recall	f1-score	support
0.0	0.88	0.99	0.93	3785
1.0	0.95	0.68	0.80	1601
accuracy			0.90	5386
macro avg	0.92	0.84	0.86	5386
weighted avg	0.90	0.90	0.89	5386

From these reports again, the RBF model is inferior in every aspect to the other two models. The difference however comes when comparing the Decision Tree to the Random Forest. In this case the Random Forest beats the Decision Tree on every aspect when the averages are looked at.

Analysis Conclusion:

Having compared each of the models using a variety of methods the decision comes down to Random Forest and Decision Tree. The model selected for the system was, in the end, the Random Forest model. This is primarily due to its higher scores on the classification report as, despite the worse False Negative rate, it outperforms the Decision Tree model in almost every other aspect.

The KNN model is used to fill in the gaps in the data provided, it is not being compared to other models but there does need to be a comparison of the variables used for the process.

Test Plans:

Testing is important when developing a system to ensure all components run as desired. To test this system, both manual and automated methods of testing will be carried out. The test results can advise future developments and bug fixes as the project progresses.

For manual testing, a basic table (see Fig 10) will be set up on a document with columns used in standard software testing. These columns will be appropriately named, focusing on the predicted and actual outcomes of the components tested. The manual tests will be carried out on the UI and the user's experience of the system.

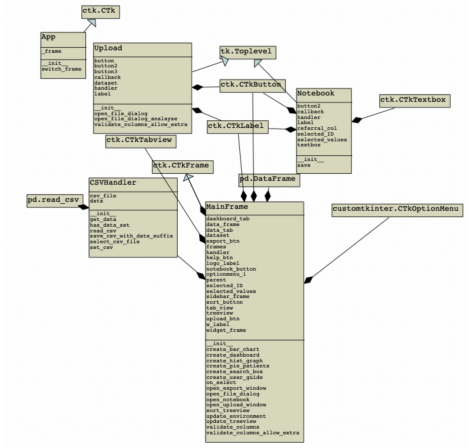
For automated testing, a method called Unit testing will be used. Unit testing is a fundamental practice in software development aimed at verifying the correctness and reliability of individual units of code, typically functions or methods. Unit tests ensure that the code functions and behaves as expected. The automated tests will focus on the back end of the system, ensuring the processes and algorithm run as expected.

Fig 10.

Software Development - Manual Test Plan

Test ID	Test Description	Predicted Outcome	Actual Outcome	Pass or Fail?	Notes
1.0	Main page - Viewing dashboard	When the application is initially run, a dashboard should display, showing multiple graphs based on the uploaded data.	When the application is opened, a dashboard is visually displayed, showing graphs based on uploaded data.	Pass	
1.1	Main page - Viewing all data	When the "All Data" tab is pressed, the application should display all data within the current CSV file.	When the "All Data" tab is pressed, the application displays all data contained in the current CSV file.	Pass	
1.2	Main page - Upload CSV file button	When pressed, the system's file explorer should open.	When pressed, a pop up opens with options for uploading data, one button for a new CSV file to	Pass	

Fig 11.



Testing & Implementation:

There are two main aspects of the project that need consolidating: the algorithm API and the UI/UX.

There are several aspects to consider when designing an app that meets the functional, and non-functional requirements whilst also following through with our aims & objectives.

In the planning phase, the team decided that Python would be most suitable to develop the application. Fulfilling the scope, a GUI (graphical user interface) needed to be created, which also needs to meet the functionality requirements outlined. For example, having a scalable interface and compatibility across all devices.

Following our development, the team plans break down big tasks into more manageable chunks; commonly known as decomposition.

Charlie & Jack led the development of our bespoke Python-based AI algorithm programming interface. This was a culmination of effort from both members, as outlined in the machine learning research section.

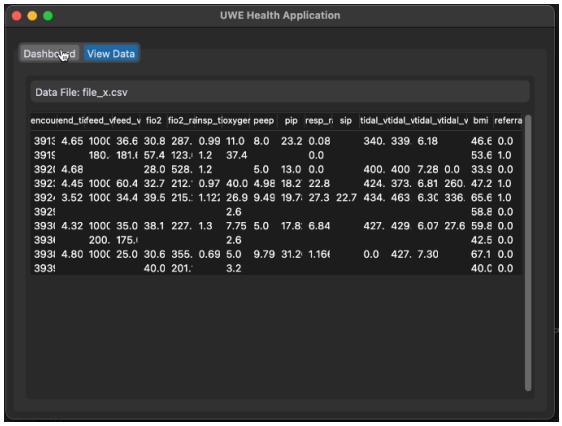
Cosmo led the development of the UI/UX, with help from all members involved – this was a consolidation of research, design and planning from the entire project. The final product must be something that was not only aesthetically pleasing for the user, but which also functioned well.

All Design Implementation images are in the appendix.

UX & UI Design:

Early stages of development:

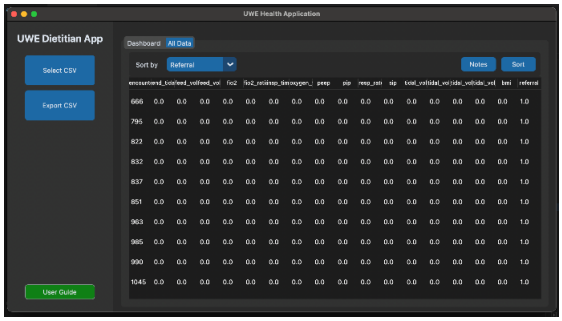
- One of our very first iterations of the project.
- Only has barebones functionalities
- Testing the Python Custom Tkinter UI elements, further noting how they display data and how to build the application around the tools planned to use.



Compared to our final product seen next, you can observe loads of positive design changes had taken place.

Latter stages of development:

- Improved accessibility in the form of a 'User Guide'
- Improved look and feel of the UI.
- Fully scalable on any device.
- Ease of use, big and clear design choices in the form of big buttons for example 'Select CSV'
- Bigger and more legible columns and rows, which can be selected and sorted through
- Functionality in both back, and front end, only achieved by great teamwork and communication.



Data Handling:

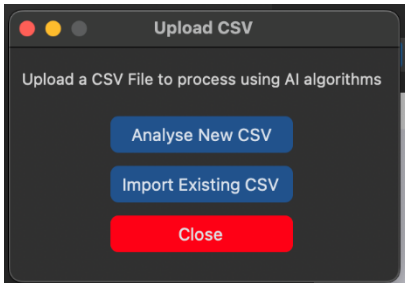
Efficient handling of data is one the key aspects of the product.

Breakdown some of the design in Python:

- Deviated from some of our original design plans, however, the developers implemented ways to handle CSV files.
- The App creates an instance of CSV handler to handle CSV files data across the product and all sections. Good programming practices have been followed.
- Using a SQLite3 local database to store some patient details.
- The final product uses a combination of SQLite3, and CSV data stored in a Pandas DataFrame.

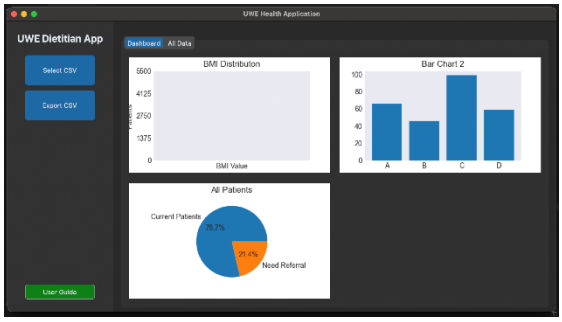
```
# CSV Handler class
class CSVHandler:
    def __init__(self):
        self.csv_file = None
        self.data = None

    def has_data_set(self):

    def save_csv_with_date_suffix(self, data, filename_prefix, directory="processed"):

    def get_data(self):

    def read_csv(self):

    def set_csv(self, new_csv):

    def select_csv_file(self):
```

Interact with the CSV Handler via the Upload Widget.



Data Visualisation:

The developers utilised Matplotlib to generate graphs for the user dashboard. This links back to our planning phase of the product; with visualising data being outlined in the scope of the project. From the data now pulled using the CSVHandler object, bespoke functions were built to take the data and plot graphs.



Although a non-functional requirement, an advantage of having Matplotlib baked into our frontend is that with Tkinter, it allows the window to be fully scalable with the graphs.

Algorithm API:

Consolidated efforts from the machine learning model section, a quick breakdown of the API:
(can be found at algo.py)

- It defines a class Algorithm with a method for performing machine learning tasks.
- The method loads a CSV file containing data, preprocesses it, splits it into training and testing sets.
- It then imputes missing values using K Nearest Neighbours and scales the data using StandardScaler.
- Random Forest Classifier is utilised for training and prediction.
- The method returns a DataFrame containing predicted referrals for a given dataset.

Testing Phases:

Alpha & Beta Testing

For the testing process, both manual and automated testing methods were executed to ensure sure the system is functioning as it should.

Manual tests were run on the system as the user would typically use it. These tests focus on the UI and the user experience with the system. Completing these tests helped outline any faults with the overall experience of the system, giving the development team feedback on which parts of the code needed fixing.

(To view Manual Tests document, see App3)

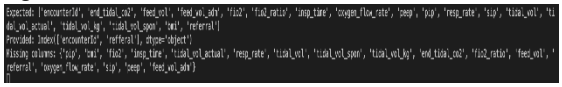
Automated Testing:

For automating testing, a process called unit testing was carried out. The primary goal of unit testing is to ensure that each unit of code behaves as expected under various conditions and test cases. The three key components we used to unit tests for were the algorithm, CSV uploader, and the app itself.

The unit test suite for the Algorithm class thoroughly evaluates the functionality of the algorithms implemented within the class. Each test case focuses on a specific aspect of the algorithm's behaviour, ensuring its correctness, robustness, and reliability. The first test case simulates the algorithm's execution with mock data, validating that it produces the expected output format - a DataFrame with the correct columns. This test ensures that the algorithm processes input data correctly and generates output in the expected format. The second test case verifies the process for missing values within the dataset. By testing with mock data containing missing values, it ensures that the algorithm appropriately handles and imputes missing data, preventing potential biases or inaccuracies in the analysis due to missing information. Overall, this unit test suite plays a critical role in ensuring the effectiveness, correctness, and reliability of the Algorithm class in making referrals to dietitian's based on patient data.

The unit test suite for uploading a CSV file comprehensively evaluates the functionality of CSV file handling within the hospital system application. The first test case verifies the ability of the system to select a CSV file using a file dialog. It ensures that the selected file path is returned as expected, validating the interaction between the application and the file system. The second test evaluates the system's capability to read a valid CSV file. By checking that the system can load data from a CSV file into a DataFrame and that the DataFrame is not empty, this test confirms the correct parsing and ingestion of CSV data. Conversely, the third test case examines the system's behaviour when attempting to read an invalid or non-existent CSV file. It ensures that the system handles such scenarios gracefully, returning False and maintaining data integrity by keeping the DataFrame empty (see Fig 7).

Fig7.



The unit test suite for the App class serves to thoroughly evaluate the functionality of the application’s main interface. The first test verifies the switch_frame method’s ability to transition between frames, ensuring smooth navigation and interaction across different application components. Meanwhile, the second test assesses the functionality of the open_upload_window method, crucial for enabling users to upload patient data via CSV files, thereby facilitating data analysis and management.

Unit testing has been instrumental in automating the verification process for key components of the hospital system application. Employing unit tests ensured the reliability and effectiveness of the Algorithm class, CSV uploader, and the main application interface. These unit tests have strengthened the application's reliability and effectiveness in managing patient data and facilitating seamless user interaction.

Conclusion:

Overall, the project has been successful, with the primary aims and objectives having all been met. The system makes mostly accurate predictions, with an accuracy rate of over 90% and displays the data through an easily accessible dashboard system. The functional requirements have been met with the non-functional requirements advising design decisions.

There is, however, still some room for improvement. The machine learning algorithm could be made to reach a higher accuracy, this would take a progressively longer time, however, so could be considered if the project was developed further. Another aspect that could be further developed going forward would be the versatility of the application. It currently displays all the information regarding the task well but could be adapted to be used as portal to view all the patients in a ward with relative ease.

Individual Contribution Reports:

Jack McDaid - 21023808

The project has overall, been successful. This is due not only to my own contributions, but those of the group as whole. The group managed to appropriately divide tasks and each member carried out their aspect of the task successfully. The greatest issues arose from a lack of hindsight, with some aspects being necessary for the task not being identified until later during the development process. This created issues as parts of the system and report that already had been created needed altering to fit these newly identified parameters.

During the project I didn't have one specific role but fulfilled several smaller ones and helped across most others. I helped identified the Aims and Objectives, oversaw the entirety of the literature review, created our methodology and Gantt chart, carried out the testing of machine learning models with Charlies help and, finally, carried out the editing and compiling of the group report. During this process issues arose in several aspects, all of which, however, were able to be remedied. The first real issue was in the definition of Aims and objective, the initial draft of which was much more like how the requirements ended up.

Most issues were down to time management, as despite a well-planned process involving the Gannt chart, most timelines were not followed, not only by myself, but by the whole of the group. This led to a large amount "crunch" work when nearing the deadline, this has led us to not be able to get as much feedback during the development process as I would like. This has meant that some issues that would be better remedied had more time been possible. The biggest instance where this was the case was with the literature review. It had initially been planned for the whole group to contribute to the review, unfortunately most aspects of the individual contributions weren't suitable. As this was the first major holdup in the project, I took it upon myself to finish the whole review so other aspects of the project could be undertaken.

Going forward the time management issues would be the best to resolve to achieve a better result. We were pressured throughout a lot of the development as we had left everything to late. Despite this I am still very happy with the results, almost every part of the project is to a good standard. I feel this is in part due to good communication and good structured meetings that we held regularly, often twice a week. During these meetings I tended to take the role of a team leader, ensuring that all team members knew their task or tasks, and that they would be able to undertake them during the time provided. For the rare times that a member needed help with a specific task, it was generally me who stepped in to help. Most tasks, however, were attributed to the team member best able to take them on.

The single largest aspect I worked on, after the literature review, was the Machine Learning algorithm. Whilst this section was headed by Charlie, I was able to help with it a lot as we had worked together on similar issues for a group task in our machine learning module. This enabled us to create a robust, powerful system that would reliably achieve good results. As part of this I completed the write up for the machine learning testing section. Where Charlie and I had systematically gone through all appropriate models to ascertain the best one to use for the final application. The biggest issue we encountered during this was deploying an already trained model. This was something that neither of us had any experience in. After some research, we couldn't find an easy solution to fully deploy the pre trained model, as such we incorporated the running of new data into the training and testing code, this produced results equivalent to if we had been able to it properly but was somewhat slower.

Overall, I believe I contributed a great deal to the project, having written around 4000 words of the report, lead the team in most aspects and helped with some of the other technical aspects. This is not to diminish the other's roles in the project, as everyone contributed a great deal to its completion with each person's contributions being necessary.

Lauren Pearce – 22010935

From the beginning of the project, I actively participated in meetings aimed at defining project goals, requirements, and timelines. We had two meetings a week on Mondays and Fridays and additionally met several other times both in person and via Discord to discuss our progress and plans for the future. These meetings allowed team members to share ideas, identify challenges, and find ways to address the project objectives. At the beginning of meetings, we would discuss what we have done so far, what else needs to be done, and split up tasks accordingly. Outside of meetings, we would still maintain regular communication and update each other on our own progress. By actively engaging in discussions and offering input, I was able to play a part in planning and development of our system. It was decided that Cosmo, Charlie, and Jack would take on much of the tasks relating to the development and coding of the main system, and Matthew and I would take on more roles around design and testing. Any other tasks were divided up between us.

Ethical considerations are important in any software project, especially one involving sensitive data such as patient information in a hospital system. One ethical concern is ensuring the privacy and security of patient data. We used strict privacy protocols to protect patient confidentiality. Another ethical consideration is the responsible use of algorithms, particularly in decision-making processes such as patient referrals. The algorithm was heavily tested to ensure that the algorithm's decisions are explainable and accountable. One legal risk is liability for errors or malfunctions in the system that result in harm to patients or healthcare providers. It’s important that, as the developers of the system, we do as much as we can to ensure the system's reliability, accuracy, and safety. This includes comprehensive testing and documenting known limitations and risks. By prioritizing ethics and legality in all aspects of the project, we can ensure that the hospital system not only delivers value to patients and healthcare providers but also upholds the highest standards of integrity and accountability.

One of my initial contributions to the project involved creating the initial diagrams that outlined key functionalities of the hospital system. These diagrams, including sequence diagrams for CSV uploading and report generation, and initial class and use-case diagrams, served as foundational blueprints for the system's design and development. These diagrams provided clarity and guidance to the development of the system. They helped ensure that the system's design aligned with project requirements and objectives.

I played a key role in the testing of our system. I helped develop the test plans for our system, which outlined the strategies and methodologies for validating the system's functionality, performance, and reliability. A significant aspect of my contribution to the project involved the implementation of automated testing. I developed and executed automated test suites to validate the system's functionality across various components and modules. These automated tests involved unit tests targeting critical functionalities such as algorithm processing, CSV uploading, and testing the main app itself. The automated tests I developed played a crucial role in ensuring the reliability and validity of the hospital system. By testing the system's behaviour under different conditions and scenarios, these tests helped identify and address any problems or inconsistencies that might arise.

While the project was overall a success, there are some areas we could have improved. Reflecting on the project, there are several areas where we could have improved to enhance the overall quality and effectiveness of our work. We could have improved on our time management, as well as had better planning from the beginning. One aspect where time management could have been enhanced is in sticking to deadlines and milestones.

Matthew Woodhall - 21026348

The group project as a whole was successful, despite the slight lack of communication between the group members and lack of attendance in meetings. As a group, tasks were split accordingly to each members' strengths and weaknesses, giving the optimal process for developing the system and completing the project.

At the start of the project, the group was fortunate enough for each member to already know the majority of the group, some having worked with some before too. This made assigning tasks to group members throughout the assignment easier as we could outline each other's strengths and weaknesses. After brief introductions from each member, our approach to the task was discussed and decided and multiple methods of communication were opened. These included a GitLab depository for sharing files between group members and group chats on Discord and WhatsApp for the group to communicate with each other remotely.

Once the GitLab was set up, the group began creating a project plan: which member would do which task, weekly meeting times, any deadlines for specific tasks. To help visualise the progress of the project, Jack created a Gantt chart, a common practice in software development.

As a group, with the help of the group chats and other online communication methods, held meetings throughout the past few months to keep track of progress between members, with a schedule of Monday and Friday mornings. However, due to multiple reasons from multiple members, there were many meetings held with at least 1 member absent. On the other hand, when every member was in attendance of these meetings, they would be very effective. The Monday meetings were a good time to plan out the upcoming week, and what work should be done for the Friday meetings. Unfortunately, the planned workflow was far different from the actual workflow.

To keep track of risks, a Risk Management table was created, filled with potential risks the group or project might have, such as group attendance lowering. The severity of each risk was calculated based on how likely they were to happen and how much of a hazard they would be to the assignment. With this Risk Management table, the group were able to determine the more affective risks and discuss solutions to overcome such risks.

Towards the start of the project, the group decided to split the written tasks, such as aims and objectives and requirements, rather than assign roles for these tasks. Each group member would come back to the group with the work they had achieved for the group to discuss and develop further if needed. The literature review required 8-10 references, so each member carried out research on 1 or 2 references relating to the system being produced and any methods to be potentially used. These were then brought together and edited by Jack to make the final literature review sound more coherent. For the development of the system itself however, roles were assigned, with Charlie, Cosmo and Jack on the coding team and Lauren and Matt on the testing team. While the coding team were creating the system and developing new versions, the testing team began writing test plans and carrying out basic tests.

The largest component of the project I worked on was testing the system. While the development team focused on developing the system, I executed multiple tests and gave feedback on any issues for the team to fix. However, due to lack of communication between the development and testing teams and delaying the assignment multiple times, this wasn't achieved until very late in the project.

On my part, there were delays in producing work when required, specifically in diagrams, causing the development team to design as they went along. Therefore, I do not see myself as an essential part of this project. Although I contributed on the written part of this project, attended most meetings and completed the manual testing of the system, I believe the group could have achieved just as well, if not better, without my input.

Charlie Taylor - 21024059

For the completion of the group project, it was necessary to produce a project plan to keep the team on track during development. The project plan included a Gannt chart which tracked progress on tasks and gave dates that work should be completed before. This increased productivity as each team member knew what to being working on at a given time.

The team interacted with in person meetings twice a week one at the start of the week and one at the end the meeting at the start of the week allowed for a overview of the progress of the project so tasks could be designated to members to be completed that week. This allowed the meeting at the ending of the week to be used for reviewing the work that had been completed that week to see if any changes or additions could be made to it. If a team member was for some reason absent from a meeting the general overview of what was discussed, and tasks allocated was posted into a team chatroom, so members were always up to date on what was going on regardless of if a session was missed. Online meetings where also used when necessary if certain areas of the project were progressing slowly.

At all points of the project team members were assigned roles which gave them responsibility for completing and managing a certain aspect of the project this meant team members could focus on one aspect of the project and make sure its completed to the best possible standard larger aspects would have multiple members working on it at time. As for personal contributions once the project progressed to planning proper my role was to research what methods and libraries, we can use to produce the dashboard. The research included finding what software could be used to generate an easy to navigate dashboard after reviewing different software's and libraries that could be used It was decided that using Tkinter and the CustomTkinter libraries as this allowed for python to be used for the code as that is a language all members are knowledgeable and proficient in. Tkinter has all the functionality that is required in the scope of the project and CustomTkinter allowed for the dashboard to be styled easily so that it was easy for a user to navigate. Once the libraries and language had been selected an example dashboard was produced without to show the rest of the team how it would look as well as commented code showing how the libraries are used.

The main contribution made to the group was the research and implantation of a machine learning algorithm to generate an accurate prediction on the given dataset. This required research into the different types of models and what would be most suitable for the project then testing the models to see which was the most effective on the dataset. To test the models, I had to learn how to train and test a machine learning algorithm this was aided by having a machine learning module which made me familiar with sklearn and gave me skills in analysing the effectiveness of modules using methods such as classification reports, confusion matrices and ROC curves. With these acquired skills I was able to test the models deemed most suitable from the research with grid searched parameters to see what would be the most effective. At the end of the testing, it was found that the random forest model had the highest accuracy on the dataset. Another skill developed during this task was the use of the K-Nearest Neighbours algorithm to impute missing data to a data set since the given dataset had a large amount of missing data which would need to be filled before a model was deployed on it.

In reflection of the project and my contribution I gained many skills in both working and operating as a group and on the deployment and evaluation of machine learning models. Reflecting on the use of sklearn to deploy the model it may have been more effective to use a different method as sklearn is more used for training a model on a dataset than deploying a trained model on an entire dataset. To mitigate this the model is trained each time it is deployed which increases the time and resources it takes to generate a prediction.

Another point of reflection is the use of accuracy score to evaluate the effectiveness as this prediction will be used to decide whether a patient is referred to dietitian. This means ethically more weight should be put on reducing false negatives as they mean someone in need of referral doesn't get the referral, they need which is worse than a patient that doesn't need referral getting one.

Cosmo Elford - 21018021

In this project, we developed an application to assist dietitian referrals in CCUs (critical care units) using research papers, modern programming tools and artificial intelligence. My role was to lead UI/UX implementation & design. The role mainly consisted of programming towards the latter stages of the project, however, I also played a big role in early stages of design and research into technologies which fit the project scope, and final product we wanted to deliver.

The approach taken by our group to deliver the product has been great. I feel like we worked very well. Honestly, at times we lacked motivation, but that's sometimes part of working in a team setting. I found it to be my first group endeavour, and a unique experience in which I have learnt a lot from. Working with other programmers, and having the project come together quite seamlessly was ultimately very fulfilling. Despite the early stages of development being slow, we picked up towards the end of term, delivering something that fully satisfied both our functionality requirements.

Moreover, the attendance at meetings have been regular enough, but where work could be done from home, we took that advantage. By setting up communication channels in various forms such as WhatsApp and Discord; we could stay organized, manage our time effectively and decompose tasks effectively. In terms of weekly planning, our GANNT chart was a blueprint for the layout of our working weeks. Mostly, we found that a slightly more relaxed approach through regular online meetings was great for our productivity dynamic.

During the early stages of planning, a lot of the team roles weren't so concrete. The planning, research and design were all a culmination of efforts from the group. However, during the latter stages of the project, we found that some members had different strengths and weaknesses, and we had to adapt to how we worked so thus we split up doing different tasks. Despite everyone taking an important role, the most prominent voice in our group I'd say was Jack, taking on a lot of leadership within our group. During implementation, Charlie Jack and I took on the programming aspect, and kept in regular contact with one another. We found this to be a good dynamic, as the other members of the group could write up a comprehensive testing report whilst they received app updates through the Git repository.

Considering LESP (legal, ethical, social & professional) issues, we had done a lot of research into AI sustainability and ethics. As the greatest functional feature of our project, we wanted to ensure that a lot of research had been done in that field, finding recent studies on efficacy of AI, namely in the use of technology in the health sector.

Throughout the project lifetime, I consistently contributed to early design planning, programming, risk management, aims, functionalities, and final documentation. I'd also say my greatest contribution was to the front-end interface, which you can see examples of in the documentation. By having to adopt Python libraries such as Tkinter and CustomTkinter, I was able to rapidly mock-up UI plans to test and adapt. After a recommendation from Charlie, I had to familiarise myself with their respective documentations; I then got to work programming a user-interface for our product.

I'd like to conclude by saying it was a great learning experience, I'm happy to be able to call it a group success. I picked up many skills working with other people, and it's something that will follow me into any professional setting in the future. If I could do it again, I would love to have more time developing non-functional features for the application. Where we have left it, it would be completely scalable, and I would like to see it deployed in a real setting; gathering data for training our prediction model and gathering feedback from actual hospital staff.

References:

Wickramasinghe, N., Arias, R., Wilgus, J. and Gonzalez, C. (2015) Reducing Healthcare Disparities with Technology. Encyclopaedia of Information Science and Technology, Third Edition [online]., p. 9. [Accessed 01 March 2024].

Vadillo, Patricia C. OTR/L; Rojo, Estrellita S. RN-C, CCRN; Garces, Adelaida RN-C; Checton, Maria G. PhD. Maximizing Healthcare Professionals' Use of New Computer Technologies in a Small, Urban Hospital's Critical Care Unit. Journal of Healthcare Management 61(5):p 352-362, September 2016. [Accessed 01 March 2024].

Lee SI, Celik S, Logsdon BA, Lundberg SM, Martins TJ, Oehler VG, Estey EH, Miller CP, Chien S, Dai J, Saxena A, Blau CA, Becker PS. (2018) A machine learning approach to integrate big data for precision medicine in acute myeloid leukaemia. Nat Commun 3;9(1):42. doi: 10.1038/s41467-017-02465-5. PMID: 29298978; PMCID: PMC5752671.

NHS – National Health Service (2021) Mia mammography intelligent assessment. Available from: https://transform.england.nhs.uk/ai-lab/explore-all-resources/understand-ai/mia-mammography-intelligent-assessment/ [accessed 28/02/2024]

Aronson SJ, Rehm HL. (2015) Building the foundation for genomics in precision medicine. Nature.15;526(7573):336-42. doi: 10.1038/nature15816. PMID: 26469044; PMCID: PMC5669797.

Du, G., Jiang, Z. and Yao, Y. (2013) Clinical Pathways Scheduling Using Hybrid Genetic Algorithm. [online]. J Med Syst 37 (9945) [Accessed 03 March 2024].

Kramer, A.A. (2021) Using Genetic Algorithms to Identify Deleterious Patterns of Physiologic Data For Near Real-time Prediction of Mortality in Critically Ill Patients. Informatics in Medicine Unlocked [online]. 26 (100754) [Accessed 03 March 2024].

Kostopoulou, O., Arora, K. & Pálfi, B. Using cancer risk algorithms to improve risk estimates and referral decisions. Commun Med 2, 2 (2022). https://doi.org/10.1038/s43856-021-00069-1

National Academies of Sciences, Engineering and Medicine. Improving diagnosis in healthcare. Washington DC: The National Academies Press (2015)

Pimenidis, E., Kapetanakis, S., & Polatidis, N. (2021). Recommender systems algorithm selection using machine learning. In Proceedings of the 22nd Engineering Applications of Neural Networks Conference (477-487). https://doi.org/10.1007/978-3-030-80568-5_39

Scikit-Learn (2024) Supervised learning. Available from: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning [accessed 8 April 2024]

Breiman, L. (2001) Random Forests. Machine Learning 45 [online]., p. 5–32.

Song, Y.Y. and Lu, Y. (2015) Decision Tree Methods: Applications For Classification and Prediction. Shanghai Arch Psychiatry. Shanghai Arch Psychiatry [online]. 25;27 (2), pp. 130-5.

Guo, C.-Y. & Chou, Y.-C. (2020) A novel machine learning strategy for model selections - Stepwise Support Vector Machine (StepSVM). PloS one. [Online] 15 (8), e0238384–e0238384.

Mucherino, A., Papajorgji, P.J. and Pardalos, P.M. (2009) K-nearest Neighbor Classification. In: Data Mining in Agriculture. New York, NY: Springer.

Management Library (2024) Available from https://management.org/waterfall-methodology [accessed 8 April 2024]