FIT3163 Project Proposal

HEART DISEASE PREDICTIVE MODEL

Luke Wilson (20643292), Utkarsh Patel (29143926),
and Joshua Fehring (28802691)

1. Introduction

Coronary Heart Disease (CDH) kills more people annually than any other illness, but it is relatively difficult to diagnose, typically requiring expensive and invasive tests such as an electrocardiogram or coronary angiography. In this project, we aim to develop a predictive model to diagnose the presence of CDH in a patient based on data about the patients health, building upon previous work in this area that has aimed to develop similar models. We plan to develop a predictive model and implement it in an application that will allow for the input of patient data and the output of the models diagnosis for use by physicians and health professionals to obtain more accurate and convenient diagnoses for patients in order to refer them on to further treatment and testing. A number of models will be developed, tested and refined until a final model is chosen with the best outcomes. The development and testing of this model and application will take place over semester 2 of this year ending in November as detailed in this document.

2. Literature Review

2.1. Introduction

Coronary heart disease (CHD) is the leading cause of death globally according to the World Health organisation (WHO). CHD has no direct cure, but can be controlled by reducing risk factors and taking drugs to reduce symptoms. Early diagnosis and subsequent treatment of CHD improves outcomes and reduces mortality in patients, and so plays a vital role in reducing CHD deaths worldwide (Kramer, Schlößler, Träger, & Donner-Banzhoff, 2012). Diagnosis of heart disease can be highly complex and currently the primary and most effective method for obtaining a definitive diagnosis is through a coronary angiogram -- a highly invasive and costly procedure, requiring experienced technicians to perform. Such restrictions limit the efficacy of the procedure, as isolated and exiguous communities without access to specialised equipment and expertise are neglected, and the cost makes it prohibitive to routine use for diagnosis in low-risk patients. Due to this, development of fast, cheap, and non-invasive methods of diagnosing CHD has become an area of increasing interest. One of the most promising methods of diagnosis investigated in recent years has been in the use of predictive models derived from large bodies of existing data on the symptoms, measurements, and laboratory readings of patients to determine the likelihood of an individual presenting with CHD (Krishnaiah, Narsimha, & Chandra, 2016). This review will aim to summarise and compare a number of the leading approaches towards developing a predictive classification model for the purpose of diagnosing heart disease and critically compare them in order to shed light on the future development of improved models.

Predictive models are an attractive option for the diagnosis of CHD as they can be used to quickly and accurately generate diagnoses based on readily available information from the patient. The increased speed, affordability, and convenience that predictive models could offer could greatly improve quality of life and health outcomes for patients with CHD, especially if the affliction is detected early. In order to ensure that patients who need treatment for CHD receive it and to reduce the number of treatments and testing when unnecessary, highly accurate models are needed. Unfortunately, there is no perfect model capable of diagnosing patients perfectly every time, so compromises are necessary when training a model to be effective. In order to attain a desirable level of accuracy, indicative of a reliable, effective, and robust model that may be generalised for widespread use, researchers have attempted numerous machine learning methods and modelling algorithms (Dwivedi, 2018).

Common models employed to diagnose patients with CHD include Random Forests (Jabbar et al., 2016), Logistic Regression (Dwivedi, 2018; Kim & Kang, 2017), Artificial Neural Networks (ANN) (Alizadehsani et al., 2013; Dwivedi, 2018), Naive Bayes (Dwivedi, 2018) and Support Vector Machines (SVM) (Alizadehsani et al., 2013; Ciecholewski, 2013). Each of these methods have benefits and drawbacks that will be explored in this review.

2.2. Feature Selection Algorithms

Feature selection algorithms identify features of the data that would be considered statistically or probabilistically significant to the model's predictive capabilities, and exclude those extraneous or redundant features that could introduce error or noise into the model. Since excluding important features can lead to the loss of important inferential information in the model, and including too many features in the model may introduce noise or bias in the data, reducing the reliability and predictive power of any model derived from it, feature selection is therefore a vital step in any model development. There are many methods to select or extract subsets of features, and no one is universally optimal. Most are primarily suited to a particular model or types of data, but they generally fall into 3 major classes; Filter, Wrapper, and Embedded. Hybrid approaches of filter and wrapper methods may also be used (Labani, Moradi, Ahmadizar, & Jalili, 2018).

2.2.1. Filter Methods:

Filter methods start with the entire dataset of features and iteratively reduce the subset by selecting features based on properties of the data. They generally weight features based on some predetermined criteria, such as information gain, variation, or correlation. Filter methods fall into two further categories of univariate or multivariate methods. Univariate filter methods assess each feature independently. For instance, those features that exhibit low variance would not provide much predictive power to a model as they would be treated as effectively constant, while multivariate methods take into account multiple features and their interactions, and are therefore more capable than univariate methods at accounting for irrelevant or redundant features. An example of multivariate filtering would be the consideration of pairwise correlations between features. Those features that are found to be highly correlated (or highly negatively correlated) would be effectively redundant, as their cumulative effects could be derived from one feature. In such a case, the feature that is more correlated with the target variable would be retained and the other discarded. Filter methods are generally much simpler and not as computationally intensive as wrapper methods, however multivariate filters can scale poorly as the size of the dataset increases.

2.2.2. Wrapper Methods:

Wrapper methods select subsets of features based on their cumulative effects on the performance of the specific model that the features are being selected for. Because they take into account the specific model and model dynamics, wrapper methods are generally better at finding the optimal mix of features for a given model, as well as take into account interactions between features (Saeys, Inza, & Larrañaga, 2007). Improved specificity however comes at a higher processing cost, and a greater risk of overfitting to the data.

2.2.3. Embedded Methods:

Embedded methods are feature selection methods that are built into or functionally required in the development of specific models. Such models that contain embedded feature selection include decision tree-based models such as Random Forest that automatically generate feature importance weightings, and some implementations of weighted naive Bayesian models (Jiang, Zhang, Yu, & Wang, 2019). Much like wrappers, embedded methods can capture interactions between variables as well as develop and optimise the feature selections for a specific model, however they are limited in that they can only be used with the models that they are embedded in.

The choice in features to use will vary based on the data, the model being used, and the purpose of the model, but in all cases selecting a feature selection method is a vital step in the development of a model.

2.3. Summary of the Candidate Models

2.3.1. Random Forest

A Random Forest is an ensemble model that creates multiple decision trees and calculates values using the votes of these trees. Each tree predicts using a randomly selected subset of all features within the data, then classifies the data and returns its result. Every tree in the Random Forest then provides one vote and the ensemble result is derived via some centrality measure. Due to the use of decision trees as predictors, Random Forests can capture some non-linear relationships within the data, but conversely, they can be weak at capturing linear relationships between features. Random Forest models can be classification or regression models, and can handle both continuous and categorical data (Cutler, Cutler, & Stevens, 2012). Major benefits of Random Forest models include their relatively impressive accuracy when compared with other models and the simplicity of their execution, as well as an effective embedded method for feature selection that can be used to identify and quantify important features within the data. They are also relatively resistant to anomalies in the data such as missing values or outliers. These improvements in performance over standard decision tree models come at the cost of an increased computational complexity, especially for larger datasets, as well as a complete loss of interpretability of the resultant model's logical decision structure.

2.3.2. Artificial Neural Network

Artificial Neural Networks (ANNs) are an abstraction of the biological neural networks from which they take their name. They consist of structured layers of "neurons", where every neuron processes a continuous signal level according to weights along the connecting links ("axon"), biases within each neuron, and a decision function within each neuron. The uppermost layer consists of a single neuron for each input feature -- having been normalised to a signal between 0 and 1 -- and a layer of output neurons for each output variable is at the other end. Categorical data must therefore be translated into a series of "dummy variable" binary switches. In between, any user defined number of "hidden layers" of neurons exist between the input and output layers. The greater the number of processing layers between input and output, the greater the ability to fit a model to the data. However, this also comes at the expense of a larger computational load, and an increased risk of overfitting the model to the data that would make the model unfit for generalisation.

2.3.3. Support Vector Machine

Support Vector Machines (SVMs) are models that classify objects by splitting the data into groups based on the features of the data along N dimensional hyperplanes where N is the number of features in the data (Noble, 2006). These hyperplanes bound groups of objects that share the same classification within the data, so new objects can be classified based on which hyperplane partition they fall into. The distance measure or metric used to determine the n-dimensional "closeness" of nodes, and whether they are separated by such a hyperplane, depends on the selected kernel, and the structure and form of the data. The key to SVM models is therefore the selection of the hyperplane which partitions the data, as well as any necessary transformations required to move the data into higher dimensional space so as to be more effectively separated by the hyperplane. Various forms of this boundary can affect the model's classification accuracy, as well as its classification biases. SVMs are very good at handling highly complex or reticulate data, as well as data with large feature sets, and can therefore be some of the top performing predictive models when optimised accordingly. The primary downsides to SVMs are that they are limited to classification models, as well as the lack of interpretability of the model, which can make optimisation of the model extremely challenging.

2.3.4. Naive Bayes

Naive Bayesian classification models are based on Bayes' theorem of conditional probability, but are dependent on the statistical assumption that all features in the model are independent. Bayes' theorem allows the calculation of the probability of a particular state conditional upon another, or as it is applied in the Naive Bayesian model, the probable classification of an object conditionally upon its features (Miettinen, Steurer, & Hofman, 2019). Furthermore, the Naive Bayesian model can be used to model both continuous and binary data simultaneously by altering the likelihood distributions -- Gaussian for continuous, Bernoulli for binary -- and is easily updated with new data by updating the posterior distribution as the prior. While the "naive" assumption of independence does not often translate well into reality, it does generally provide a decent model for classification (Saritas, & Yasar, 2019). Naive Bayes is easy to train and implement, however its assumption that all predictors are independent is usually unrealistic, and can therefore limit its effectiveness as a reliable and unbiased predictor. Further, it is highly sensitive to the form of the data that it is conditioned upon.

2.3.5. Logistic Regression

Logistic Regression models classify data based on a logistic equation derives the marginal log likelihood for each feature as a coefficient in order to probabilistically evaluate the classification of an object. These coefficients are calculated using the maximum likelihood estimator of how well each feature in the training data predicts the class of that object. Logistic regression is relatively easy to train and refine through tweaking of the parameters, however it is prone to overfitting and is highly sensitive to noise and outliers in the data. Furthermore, it assumes linear relationships between features in the data, which can often prove unrealistic.

2.4. Comparison of the models

Many models for the diagnosis of CDH have been proposed using these techniques, and we have reviewed 5 papers proposing a total of 13 models for comparison in order to determine the most promising candidates for future model development (Table 1). To compare the models, we examined the accuracy of their predictions, as well as the specificity and sensitivity of the model. When evaluating the most suitable model, all three of these measures will be taken into account, although sensitivity will be weighted more heavily than specificity due to the disproportionately negative effect of not detecting a patient who has heart disease and failing to refer them to further testing when compared to the impact of falsely diagnosing someone as having heart disease and causing them to undergo further testing unnecessarily.

Of the models examined, the best performing in both accuracy and sensitivity was a SVM model developed by Alizadehsani et al. (2013) that utilised Sequential Minimal Optimisation (SMO) to train the SVM as well as used bagging as a cross-validation to improve the performance of the model. This model had an accuracy of 89.43% ± 6.78%, a sensitivity of 91.67%, and a specificity of 83.91% when all features from the data were included. In the same study, an ANN model was also shown to perform well, obtaining comparable values to the SVM but falling just behind in all 3 metrics. A Naive Bayes model was also proposed, however this model performed much worse than the others with an accuracy of 47.84% ± 6.35%. However, Alizadehsani et al. were able to generate three additional features that were specialised to recognise blockages in the three major coronary arteries. Using these,

TABLE 1. Comparison of predictive models and studies

Paper Title	*First Author, Year*	*Location and Date*	*Contextual Variable*	*Relevant Model*	*Results Accuracy?*	*Sensitivity*	*Specificity*
A data mining approach for diagnosis of coronary artery disease	*Alizadehsani, 2013*	*Tehran, Iran 20 September 2012*	*"CAD" (Coronary Artery Disease) or "Normal"*	*Support Vector Machine (with SMO and Bagging)*	89.43% ± 6.78%	91.67%	83.91%
				Naive Bayes	47.84% ± 6.35%	28.7%	95.42%
				Artificial Neural Network	85.43% ± 7.02%	90.28%	73.56%
Performance evaluation of different machine learning techniques for prediction of heart disease	*Dwivedi, 2018*	*Bhopal, India 17 September 2016*	*"Absence" or "Presence" of Heart Disease*	*Artificial Neural Network*	84%	87%	79%
				Support Vector Machine	82%	77%	89%
				Logistic Regression	85%	89%	81%
				Naive Bayes	83%	85%	80%
				Classification Tree	77%	79%	73%
Ischemic heart disease detection using selected machine learning methods	*Ciecholewski, 2012*	*Krakow, Poland 27 November 2012*	*Classification of SPECT cardio imagery*	*Support Vector Machine*	84.19% (*avg.*)	71.67% (*avg.*)	87.81% (*avg.*)
				Artificial Neural Network (1 hidden layer)	79.15% (*avg.*)	68.33% (*avg.*)	82.31% (*avg.*)
Neural Network-Based	*Kim, 2017*	*Incheon, Republic of*	*"High" or "Low" risk of CHD*	*Logistic Regression*	80.32%	87.53%	83.63%

Coronary Heart Disease Risk Prediction Using Feature Correlation Analysis		*Korea*		*Artificial Neural Network*	81.09%	67.55%	85.08%
Prediction of Heart Disease Using Random Forest and Feature Subset Selection	*M.A Jabbar, 2016*	*Hyderabad, India 15 December 2015*	*Diagnosis of heart disease*	*Random Forest*	*83.7%*	*85.8*	*82.3%*

the researchers were able to increase the metrics of their models, though only marginally for SMO and ANN, it saw improvements in excess of 20% for Naive Bayes for both accuracy and sensitivity. The generation of new features may be beyond the scope of this project though.

While this initially suggests that Naive Bayes does not perform as well as ANN and SVM models, another study by Dwivedi (2018) examined the performance of Naive Bayes models as well as Logistic Regression, ANN, Classification Tree, and SVM and found them comparable. Notably, Naive Bayes was found to have an accuracy of 83%, which far outstrips the Naive Bayes model proposed by Alizadehsani *et al*. Dwivedi also found that the logistic regression model performed best out of the models tested with an accuracy of 85% and a sensitivity of 89%, while both ANN and SVM models did slightly worse with accuracies of 84% and 82% respectively. The differences in findings between these two studies is likely largely due to differences in the development of the models resulting in different models with different prediction accuracies. The ANN and SVM proposed by Alizadehsani *et al.* generally outperformed the ones proposed by Dwivedi, although they had lower specificity, while the Naive Bayes model proposed by Dwivedi performed far better than that proposed by Alizadehsani *et al.* This highlights the importance of model development as models of the same type can perform very differently based on their construction.

Feature selection was also likely a factor in the differences in performance of the models in these studies. Dwivedi selected 13 parameters as predictors for their models, while Alizadehsani *et al.* included all features from the dataset in the models we compared. Alizadehsani *et al.* also compared the accuracy of their models when using only features selected by a feature selection algorithm and found that this improved the accuracy of every model they tested by roughly 1-5%. These findings highlight the importance of feature selection in the creation of predictive models.

Neither of the papers looked at Random Forests as a predictive model for CDH, although Dwivedi (2018) did test the accuracy of a single classification tree and found that it had lower overall accuracy than any of the other models. A paper by Jabbar, Deekshatulu, & Chandra (2016) proposed a Random Forest model for the diagnosis of heart disease and found that it had a classification accuracy of 83.70% which is not as high as the highest performing models previously discussed, but still cements the Random Forest model as a possible candidate for a diagnostic model given its similar performance to other methods. The study by Jabbar, Deekshatulu, & Chandra (2016) also compared the accuracy of their model on different data sets and obtained varying results for each. The model scored 83.70% accuracy on the Cleveland dataset, which is a commonly used dataset for training machine learning models for the diagnosis of heart disease, but scored 100% accuracy when tested on data from hospitals in Hyderabad, India obtained by the researchers which they called the T.S dataset. The models were initially trained on this T.S dataset which could account for some of the increase in accuracy, however the fact that there were no errors made by the model does raise some doubt about the validity of this figure, possibly indicating issues with the size of the dataset, the testing method, or overfitting by the model that does not generalise well to other datasets. The findings of this paper are still valid for the Cleveland dataset, although it does show that validating a model on different datasets can be informative about the accuracy of the model.

The range of models we compared from various studies in table 1 shows that while a wide variety of models can provide a high level of accuracy, SVM and ANN models generally perform well when compared with other models, however the performance of every model is subject to significant fluctuation which suggests that further refinement and optimisation may be possible for each of them.

2.5. Recent Advances

In a 2020 paper by Reddy *et al.,* researchers proposed a new feature selection algorithm for diagnostic models for heart disease that employs a hybrid method of feature selection along with an implementation of a genetic algorithm fuzzy logic to improve feature selection over previous models. Fuzzy logic involves the “fuzzification” of data into one of multiple non-binary classes based on the values in the data. It is used in the paper to convert numerical values into classifications such as “high”, “medium”, and “low”, then to use these classifications to refine classification and selection of the features within the data. Using fuzzy logic the feature selection model was able to outperform other similar hybrid genetic models, showing that employing fuzzy logic could lead to improved performance of feature selection algorithms. Use of fuzzy logic to improve model performance has been a popular approach recently, being used in multiple studies (Iancu, 2018; P & Acharjya, 2019). This provides a good basis for possible improvement on previous work using fuzzy logic. Feature selection using genetic algorithms has also been an approach that has seen recent success in improving the performance of predictive models (Gokulnath & Shantharajah, 2019; Jasuja, 2020; Mafarja & Mirjalili, 2018). Genetic algorithms are heuristic algorithms based on the theory of natural selection that select features by creating groups of features, then selecting subsets of those groups and “breeding” the best performing ones together in order to move towards more optimal algorithms. Development of these algorithms is promising for feature selection in general, and could be used in heart disease diagnostic models in order to improve performance.

2.6. Conclusion

Based on our review of CDH diagnosis using predictive models, we found that many different models have had varying success and shown promising results, however ANN and SVM models consistently give good results when compared with other models and are very good candidates for the development of future models. Feature selection was also shown to be extremely important to model performance, and proper testing of models was shown to be vital to proper evaluation of a models accuracy. Going forward, research should be directed towards further optimisation of these algorithms using new feature selection tools such as genetic algorithms, as well as using fuzzy logic to improve the predictive accuracy of the models by making feature interpretation more flexible. Less used models that have performed well could also be better characterised by undergoing these optimisation techniques, and may perform comparably to ANN and SVM models.

3. Project Management Plan

3.1. Project Overview

Heart disease is one of the leading causes of preventable death around the world and so the value of an effective method to catch early signs of CHD in high risk individuals cannot be overstated. No members of the team are medically trained, and so we do not intend for this system to perform diagnoses, or replace medically trained professionals. Instead, we will be using statistical and probabilistic analysis to merely flag those high risk individuals in order for them to be further investigated by those who are qualified to be making a diagnosis. For this reason, given the nature of the system and the potentially devastating consequences of a false negative, we intend to prioritise sensitivity over specificity. That is to say, we believe that a higher True Positive Rate (TPR) is more desirable than a lower True Negative Rate (TNR). If our system happens to flag that someone has high risk, and after further investigation they turn out to be fine, then that is good news. Yet the converse is not so true.

The resulting models will be developed from training data sets from the Alizadehsani 2012 study, consisting of 303 observations of 55 features and a target variable for each observation -- “CHD” or “Normal”. This data is already tidied, and therefore does not require extensive preprocessing, except for model specific transformations such as the normalisation and factorisation required for Artificial Neural Network Models. A secondary “processed” data set from a University of California study called the Cleveland dataset, also consisting of 303 observations, but with 75 features, has all been obtained and can be used for the same purposes. Additional datasets from the same UCI repository are available, however they are not cleaned and contain several missing values, so they may not prove very useful for our purposes.

Before the models can be developed however, extensive feature selection must be performed to reduce the dimensionality and complexity of the data into a form that is more manageable for our models. Specifically, with feature sets this large it is almost inevitable that a subset of the features are irrelevant, redundant, or superfluous. For instance, it is already apparent before processing has even begun that statistical analysis performed by OLS regression would produce biased estimates, as features for “Height”, “Weight”, and “BMI”, are included, as well as a binary flag for “Obesity” when “BMI” is greater than or equal to 25. Given that a necessary assumption for OLS regression to produce Best Linear Unbiased Estimates (BLUE) is that there exist no linear dependencies between features, and that BMI is a linear combination of Height and Weight, any statistical results from OLS methods would produce biased estimates and are therefore ineffectual.

The models will be developed and trained by k-fold cross validation, where the training data set will be split k times, with $\frac{1}{k}$ of the data being reserved for testing, and the remaining $(1-\frac{1}{k})\%$ of the data being used to train the models themselves. Testing will consist of unit testing sections of code during development, as well as testing the performance measures of each model in order to determine which, if any, or even an ensemble of the resulting models will produce the most sensitive classifications. To this end, Receiver Operating

Characteristics (ROC) will be calculated, as well as confusion matrices to determine the sensitivity and specificity of the models, and Gain and Lift to determine the performance of our models against if they weren't used.

Finally, a user interface (UI) will be developed for the deployable model(s) via a website application, where the intended audience of healthcare providers (HCP) can enter non-identifiable patient data for the models to perform analysis and produce an output that is interpretable to the HCP. Currently, the output is expected to be probabilistic, or distributional, such that the HCP can judge whether the risk is great enough to justify further investigation.

Any resultant models should be able to be updated with new data. Subsequently, the team should consider the possibility of either requiring email sign up to use the model, allowing for a follow up on high risk classifications to determine if they were diagnosed. Such data could be vital to the further development and fine-tuning of the system.

3.2. Project Scope

This project is about training an ensemble of Heart-disease classification models using the dataset(s). For a given number of observations and features, the models aim to test and quantify the risk that a given observation has CHD. Given sufficient time and progress, it is the intention of the team members to develop UI to show the statistical data, risk, or probability of a patient with the provided features as presenting with CHD. The specific techniques and processes for the advancement of the project's milestones can change slightly depending upon the data analysis and subsequent results. The project will be running until November 2020. Costs for the project aren't fully determined at present, as situations during COVID-19 are unresolved, unique, and unfolding, so additional yet necessary charges for internet, hardware, and software, if any, are not yet known. There are three students in the group who will collaborate during research, development, testing, and deployment phases, including decisions for the algorithms, tools and techniques required for the completion of the project. The group also has the supervisor Afsaneh who will support and mentor the team during the lifecycle of the project.

3.2.1. Project Deliverables

Project Management Deliverables are fluid and always subject to change. However, at this stage of the project, the team has:

- The above Scope Outline Statement
- Requirements Traceability Matrix
- Risk Register
- Work Breakdown Structure
- Project Timeline/Gantt Chart

For Product Deliverables, the team expects to have at the completion of the project:

- A deployable classification model that can be used to estimate the risk of a patient presenting with CHD, given the provided features.
- Development and testing code.

- End user instructional documentation for the system.
- A website application through which end users can access the model, input feature data, and receive the results of the model's analysis.

3.2.2. Product characteristics and requirements

See appendix 1 for Requirements Traceability matrix

3.2.3. Product user acceptance criteria

The HCP end user will be able to enter the minimal amount of (non-identifiable) data for a patient in order to derive a risk assessment for the data's likelihood to present with CHD. The system does not and can not diagnose, but instead offers a probabilistic output for the HCP user to apply in conjunction with their own skills and expertise to make healthcare decisions in the interest of the patient.

3.3. Project Organisation

3.3.1. Process Model

The team has elected to adopt a predictive life cycle model for the project, as all three members are largely inexperienced with models and project development. Furthermore, the expected requirements of the system, and the data itself, are not expected to change during the lifecycle of the project, so short term adaptivity is not a necessary requirement of the project team. However, it is expected that decisions to be made on which models and features to focus on will be required during the process, as well as algorithms to format output. The project therefore cannot be entirely planned for in advance, but a majority of the uncertainty can hopefully be accounted for by sufficient foresight during the planning phases.

3.3.2. Project Responsibilities

While no one team member is solely responsible for the completion and workload of a single subsection of the project, we as a team have designated "leaders" for certain project functions, in that the function leader is the one responsible for leading the completion of a subsection of the project, including scheduling, communication, and documentation within that specified domain. Actual workload within each function will be shared, but leadership will be shared. The designated functions and leaders are as follows:

Joshua Fehring:	**Luke Wilson:**	**Utkarsh Patel:**
Documentation, research, and pre-processing	*Feature selection, and model development*	*Testing, and evaluation*

When the above classification model is successfully developed to a sufficient standard of performance then the team is to embark on the next stage of the project in planning and developing a UI interface for the system.

3.4. Management Process

3.4.1. Risk Management

See appendix 2 for Risk Register

3.4.2. Monitoring and Controlling Mechanisms

The project team has scheduled regular weekly meetings, as well as establishing a list of minimum necessary requirements for a deliverable system. By maintaining a schedule of checkpoints and milestones, the team intends to hold each other accountable for consistency during the development stage(s) of the project until the minimum requirements are met, and the team may then reassess remaining time in the project life against additional desirable features.

3.4.3. Communication and Reporting Plan

The project team has scheduled weekly meetings, though we are not employing an agile project methodology. These meetings are therefore intended as a means to maintain consistent communication about where the project is at, which milestones are yet to be achieved, and how the project is progressing. These cannot be considered scrum meetings as there will be no short sprints, but some elements of agile development will be incorporated; primarily the regular meetings and status reports. Each meeting will have a planned topic or challenge to address, defined at the end of the previous week’s meeting, or during the week’s work between meetings if a new unforeseen challenge arises. Meetings will be led by the team member who is “leader” for that function, as defined under Project Responsibilities, and minutes will be taken for each meeting. Meetings will have very similar structure as scrum meetings in that each will address the three questions:

1. What have we accomplished since the last meeting?
2. What are we working on now?
3. What is the biggest challenge we are facing towards completion of our current work?

However these meetings will only occur weekly. The structure of the next six months are unsure as the COVID-19 pandemic response continues, and restrictions are eased in steps. It would be helpful to have meetings in person but it is still uncertain when or if this can happen during the second semester. Communication between meetings will be maintained via a shared Facebook Messenger group chat, and it is expected that most, if not all, meetings will be held digitally via Zoom.

3.4.4. Review and Audit Mechanisms

For development code, the team will be using versioning control via GIT functionality in RStudio IDE. This will allow simple versioning processes, as well as to maintain a single backup repository for source code. A secondary redundancy backup will be maintained via Google Drive that automatically synchronises with the root directory for RStudio document files. Finally, documentation will be maintained within a shared Google Drive Folder. Such documents will include the prioritised minimum requirement feature list, schedule, meeting minutes, and any further documentation required as necessary. Quality control will be maintained by using well established and peer reviewed R packages where possible, reducing development overhead and room for introduction of errors.

3.5. Schedule and Resource Requirements

3.5.1. Schedule

The provided Gantt chart (APPENDIX 3) contains useful information on how the team would like to approach the project duration in a number of steps. The process runs through a series of dependencies where one task needs to be completed before the next can begin, excepting a few tasks that overlap contiguously. Considering the project proposal is the part of the preliminary process of the project which will be followed by implementations of further tasks. The first task is pre-processing and to get the data into clean form by removing all the noise and inconsistency through a range of algorithms and the team believes this task of data pre-processing should be given a decent amount of time to understand and explore the data in various forms. Since the team believes pre-processing and data management is the most important part of the predictive model implementation process, it should be given a fair amount of time. There will be suggestions (of models for the feature selection and models to be implemented) from one of the team members while the comparison between all the models research journals is underway.

The next task is to implement the models from the list of models being selected as part of literature journals comparisons. It is another important factor in the project process so 60 days are allocated roughly to work on the models and during the same period, feature selection process will also be carried out in which the suggested feature selection model will be used to remove unnecessary features and to include only the important features in the models to be implemented. Therefore, there will be a range of models implemented with the use of the same important features in the dataset in this task. Due to the longer process, the time is allocated higher than most of the other tasks. Once the model is implemented, the next task is to test the model. Hence, the testing on each model will be performed to compare how accurate each model is behaving as compared to others. The comparison between models will be mainly based on sensitivity and accuracy, though other factors such as lift, gain, and information gain may be employed. By doing all the performance, it will end up with one best model and testing is not expected to take much time since the team will already have access to the dataset. The final evaluation of a single best model or an ensemble will determine the output of the system, the final task of the project is to develop the web platform (User Interaction) in a way such that HCP user(s) may easily access the page to use the system to check a patient’s risk factor of presenting with CAD.

However, developing the user interaction will be the last part of the team project followed by the final presentation including the model the team has come up with in the end. The above is just a rough preliminary outline of the process in terms of the schedule as it is subject to be changed according to the specific university submissions and deadlines.

WBS schedule breakdown:

Like it is mentioned before that one process needs to be finished before the next one starts. The process is divided into a series of steps. See appendix-4 for the work breakdown.

First of all, we have the first process in the project where we find the datasets relevant to our project and modify the data into the form it is required to process further. This process is expected to run during the mid-year break and will be finishing in the beginning period of semester-2 this year. The datasets relevant to the project have already been found and the exploration process is in progress. As part of this process, the team needs to communicate data using graphs available and do the wrangling process. Here, all the steps are closely related to each other and we cannot proceed to the next one before finishing the previous step. The allocation for Gathering data and exploring it to find a useful combination was originally 1 month which is roughly in the last week of May. During the last week of May, it was planned to start working on researching the literature and finding the useful algorithms to implement models and features. Again, we cannot proceed straight to model implementation without finding the results from literature. Therefore, we allocated the research and comparison of journals till 12th June to find the useful results. Now, the useful models are kept aside for further processing and the next step is to import the dataset and preprocessing related tasks. This will be the most important step to manage the data and the allocation is to finish it by 30th June. The process will be done on the features of the data to find out which features are the most useful and the most contributing to the data and these will be used in the development of the model. And it will be followed by splitting the dataset into training and testing parts for the model to be implemented followed by being tested which will take one whole month and furthermore in the start of the semester. But the target is to finish this process before 15th August.

Next, we have the process of model implementation which relies on the data modified and also on features being selected in the previous process. The steps are to process the suggested models and using the important features found, the models will be implemented, and this process should take until 15th September. There are a range of models as mentioned above to be trained on the training dataset as part of this process. They will be explored further by using a variety of their property attributes, but the process should be finished by the allocated date before it’s sent further for testing.

The implemented models will be explored further as they are tested on the testing dataset to compare how accurate they are as compared to each other. First, all the confusion matrix, sensitive values are calculated, ROC curves are plotted for each model. Until this part, roughly one week will be good enough (15th-22nd September) as the team believes they will need to see how each model performs to the data. There will be the discussion in the next week regarding the values found as part of the testing process. The decision will be taken based on whichever model performs best on the dataset overall. Alternatively, the best model can also be the embedded model (combination of two or more models).

Therefore, the team will have the best model(s) decided by 30th September. The next and last part of the project is how to output the best model to the target audience. It will include designing the web page to ensure end users can access the model smoothly to predict the heart-disease result for patients. Considering the team is still learning about the UI and design of the webpage, the allocated time for this process is 50 days [1st Oct - 20th Nov]. The features required will be on the design page and it will be developed by HTML, CSS or JAVASCRIPT language. By 20th November, it will be developed fully which can predict the heart-disease for one patient.

3.5.2. Resource Requirements

Schedule for development phase(s) of project: 180 days

Three team members, and one mentor.

Require research articles and materials regarding candidate feature selection algorithms, and classification models, for comparative purposes, and forewarning of model-specific issues that may have to be overcome.

Standard research and development of algorithms and models: minimum 8 hours per week, per team member.

Project meeting: 2 hours per week

Peer learning session in workshop: 2 hours per week

Hardware: Require one workstation per team member, including microphone for digital meetings, and a webcam is desirable but not necessary.

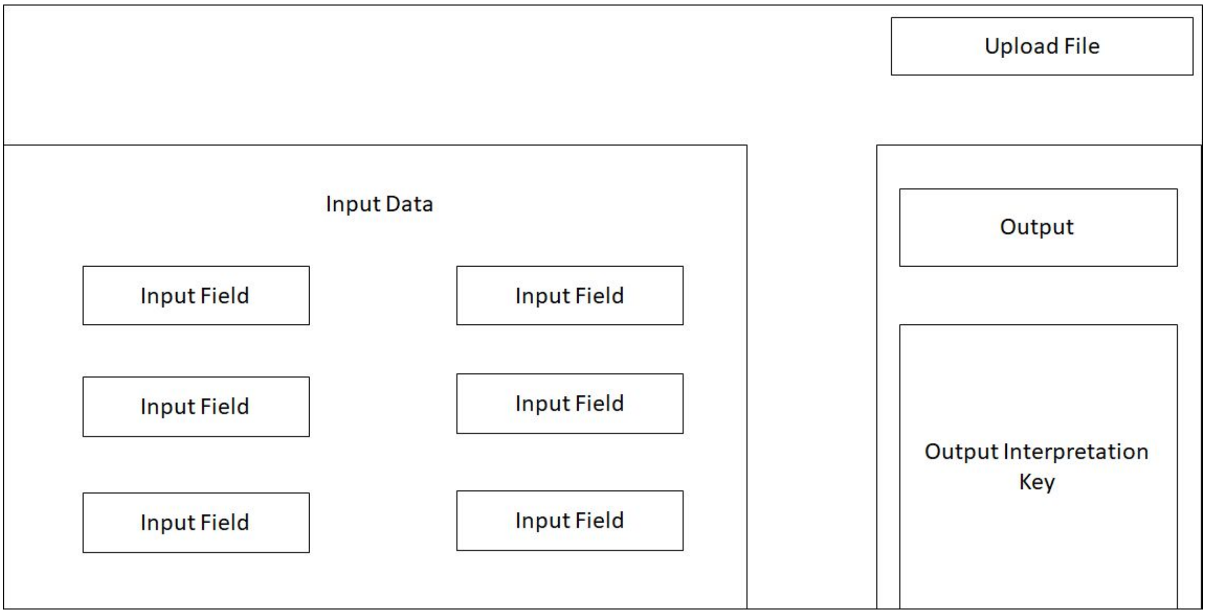
Software: Latest versions of R, Python, HTML, CSS, PHP, JavaScript. RStudio and Anaconda will be the primary IDEs during development. Data wrangling and processing to be developed in R through RStudio. IDEs and requirements for UI components will be reassessed at later phases, but it is expected that at least PHPStorm IDE will be used. Any process related diagrams will be generated through Lucidchart. Zoom software will be required for meetings, as well as active Facebook and Google accounts for access to Facebook Messenger, and Google Drive and Document accounts.

4.0 External Design

The initial training data set consists of 303 observations with 55 features and a target variable. Various feature selection algorithms will be performed on the data to reduce dimensionality and complexity, then K-fold Cross Validation will be performed to train the models on the data set, in order to reduce risk of overfitting to the training data.

Using the output of the model(s):

Since the end user only has access to a web page or application through which they can submit non-identifying feature data into the model, all that is necessary is a web portal with an ability to input a single patient's features or the ability to batch upload the features for multiple patients. Therefore, users can add data individually via input text fields/drop-down/radial inputs, or as a CSV/Excel file. The web page is the home page of the application, shown in the mock-up below:



The input fields are the patient's features which can be entered in the form it is shown on the page. It will ask to enter the entries on specific features, which have yet to be determined until feature selection is performed. Hence, the patient's features can be entered as an input. The optional way to enter the patient's data would be to enter the csv file consisting of the patient's features in a defined standard format. For this reason, entering data individually via the input fields will be the minimum required feature as it will be much simpler to standardise and control input features, and the batch upload will be an additional desirable feature that will likely take considerable time to ensure it can catch most errors. By entering the details necessary, the output of the model will be displayed on the screen with probable value in the output. The output value will either be constrained to the developed model if only one is preferred, or as a likelihood/risk value determined by output from ensemble models. Output will be displayed graphically as either a positive or negative diagnosis, or as a rating of the patients chance of having CDH from “Very Low” to “Very High”, possibly with accompanying probability. Output should be easily interpretable, or a visual key should be available for the end user to understand the output.

HTML/CSS/JAVASCRIPT will be required to implement the design of the page, PHP will be required to manage data flow between the client website and the models hosted on a server. Graphical output can be performed by R packages such as ggplot2.

5. Methodology

5.1. Required Packages

The model development algorithms would require using external packages and libraries for the corresponding model. Relevant R packages may include but are not limited to:

Decision Tree:	tree
Random Forest:	rf / randomForest
Logistic Regression:	ISLR
Naive Bayes:	e1071
Artificial Neural Network:	neuralnet / nnet
Support vector machine:	svm
ROC curve:	rocr
Machine Learning:	mlr
Data Manipulation:	Dplyr
Version Control:	GIT

5.2. Preprocessing

Since most of the datasets we have come across have been fairly well ordered with minimal missing values, not much cleaning of the data should be required to make it suitable for building models.

Normalisation of the data to values between 0 and 1 will likely improve the performance of all models, so the data will be normalised before building models. Similarly, some models will require factorisation of categorical data, which can be done by converting

Feature selection will be the major step involved in preprocessing the data in order to reduce the complexity of the data and remove noise that will negatively affect the performance of the model. Multiple feature selection algorithms and methods will be compared in order to obtain the best results.

5.3. Model Development

Pseudocode for the processing and development of the models:

dataset N: = (N1, N2, N3, ………., Nn)

Preprocessing:

Replacing Na’s with some values

or

Removing Na’s from the data: Na.omit(N)

Boundary and type testing, to ensure all feature inputs are valid:

If N.isNotType(x): catch error exception

If N.isNotInRange(): catch error exception

N$columns = scaling(N$columns)

Normalizing features with scaling

Important features:

Imp_attr = Using feature selection algorithm to find important features from desc_attr

Implementations: Performed k times

Train.N = $(1-\frac{1}{k})$ % of dataset N

Test.N = Remaining $\frac{1}{k}$ of dataset N

Decision Tree:

```
Ntree <- tree(target_attr ~ imp_attr, data= Train.N)
summary(Ntree)
```

Naïve Bayes:

```
NB <- naiveBayes(target_attr ~ imp_attr , data= Train.N)
```

Random Forest:

```
Nrf <- randomForest(target_attr ~ imp_attr, data= Train.N)
```

Artificial Neural Network:

N$non-numeric_columns = recoding them to binary values

```
Nneural <- neuralnet(target_attr ~ imp_attr, data= Train.N)
```

Logistic Regression:

```
Nlogistic <- glm(target_attr ~ imp_attr, data= Train.N)
```

Support vector machine:

```
Nsvm <- svm(target_attr ~ imp_attr, data= Train.N, kernel=”linear”)
```

5.4 Version Control

Team members will be expected to use the GIT functionality within the RStudio IDE to maintain version control. Commits will be regularly maintained, and a document will be maintained in the shared Google Drive with major update documentation details.

5.5 Overall Description

The final product will consist of a UI that allows the input of data, and the predictive model which receives input data from the UI input and displays its output prediction on the UI. The UI will take the user input in the form of values for various features of patient data, then pass this to the model where the data will be preprocessed and used to predict the patient’s outcome. This prediction will then be displayed graphically on the UI for the user to see.

6. Test Planning

6.1. Test Coverage

Preprocessing:
Ensuring no NA data, data is of the correct type and within the valid boundaries for a given feature, and is normalised for appropriate features.

Model development:
Appropriate libraries are installed, receive correct input, and output is appropriate.

Model selection:
Calculating ROC and confusion matrices for models.

Output and UI:
Users can enter data via a web portal, and that input is tested for validity. Output is clear, concise, and if in graphical format, a key is offered.

6.2. Test Methods

Testing will be largely unit tests, where each process will be treated as a black box. Given that R packages are robust and well established, we assume no errors or bugs unless already stated in package documentation.

The first training data set from the Alizadehsani study is already cleaned and tidy, so the only pre-processing required will be model specific, such as normalisation and factorisation for ANN model. Any subsequent data sets, and input data for the final deployed system, will require testing and validation of input data. This includes testing for empty/null values, incorrect data types, and data outside valid boundaries.

6.3. Test pseudo-code

```
#Testing algorithms:

Predicting the heart-disease for each observation of testing data using each model

Sensitivity: counted by true positive rate in the confusion matrix

Decision Tree:

T1 = Predict(Ntree, Test.N)

Confusion matrix(T1)

Sensitivity(T1)

ROC curve plotting (T1)


Naïve Bayes:

T2 = Predict(NB, Test.N)

Confusion matrix(T2)

Sensitivity(T2)

ROC curve plotting (T2)
```

Random Forest:

```
T3 = Predict(Nrf, Test.N)
Confusion matrix(T3)
Sensitivity(T3)
ROC curve plotting (T3)
```

Artificial Neural Network:

```
T4 = compute(Nneural, Test.N)
Confusion matrix(T4)
Sensitivity(T4)
ROC curve plotting (T4)
```

Logistic regression:

```
T5 = Predict(Nlogistic, Test.N)
Confusion matrix(T5)
Sensitivity(T5)
ROC curve plotting (T5)
```

Support Vector machine:

```
T6 = Predict(Nsvm, Test.N)
Confusion matrix(T6)
Sensitivity(T6)
ROC curve plotting (T6)
```

7. Conclusion

Heart disease is a serious and persistent problem around the world and people everywhere would benefit from an improved diagnostic method that is quicker, cheaper, less invasive, and more accessible than currently available methods. We have proposed a project researching an improved diagnostic model of heart disease, as well as the method and planning to achieve the goal of improving outcomes for patients with heart disease by developing a model to enable easier diagnosis of heart disease. This project will conclude with a presentation of the final model at the end of November 2020.

8. References

Alizadehsani, R., Habibi, J., Hosseini, M. J., Mashayekhi, H., Boghrati, R., Ghandeharioun, A., . . . Sani, Z. A. (2013). A data mining approach for diagnosis of coronary artery disease. *Computer Methods and Programs in Biomedicine, 111*(1), 52-61. doi:https://doi.org/10.1016/j.cmpb.2013.03.004

Ciecholewski, M. (2013). Ischemic heart disease detection using selected machine learning methods. *International Journal of Computer Mathematics, 90*(8), 1734-1759. doi:10.1080/00207160.2012.742189

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random Forests. In C. Zhang & Y. Ma (Eds.), *Ensemble Machine Learning: Methods and Applications* (pp. 157-175). Boston, MA: Springer US.

Dwivedi, A. K. (2018). Performance evaluation of different machine learning techniques for prediction of heart disease. *Neural Computing and Applications, 29*(10), 685-693. doi:10.1007/s00521-016-2604-1

Gokulnath, C. B., & Shantharajah, S. P. (2019). An optimized feature selection based on genetic approach and support vector machine for heart disease. *Cluster Computing, 22*(6), 14777-14787. doi:10.1007/s10586-018-2416-4

Iancu, I. (2018). Heart disease diagnosis based on mediative fuzzy logic. *Artificial Intelligence in Medicine, 89*, 51-60. doi:https://doi.org/10.1016/j.artmed.2018.05.004

Jabbar, M. A., Deekshatulu, B. L., & Chandra, P. (2016, 2016//). *Prediction of Heart Disease Using Random Forest and Feature Subset Selection.* Paper presented at the Innovations in Bio-Inspired Computing and Applications, Cham.

Jasuja, A. (2020). Feature Selection Using Diploid Genetic Algorithm. *Annals of Data Science, 7*(1), 33-43. doi:10.1007/s40745-019-00232-5

Jiang, L., Zhang, L., Yu, L., & Wang, D. (2019). Class-specific attribute weighted naive Bayes. *Pattern Recognition, 88*, 321-330. doi:https://doi.org/10.1016/j.patcog.2018.11.032

Kim, J. K., & Kang, S. (2017). Neural Network-Based Coronary Heart Disease Risk Prediction Using Feature Correlation Analysis. *Journal of healthcare engineering, 2017*, 2780501-2780501. doi:10.1155/2017/2780501

Kramer, L., Schlößler, K., Träger, S., & Donner-Banzhoff, N. (2012). Qualitative evaluation of a local coronary heart disease treatment pathway: practical implications and theoretical framework. *BMC Family Practice, 13*(1), 36. doi:10.1186/1471-2296-13-36

Krishnaiah, V., Narsimha, G., & Chandra, N. S. (2016). Heart disease prediction system using data mining techniques and intelligent fuzzy approach: a review. *International Journal of Computer Applications, 136*(2), 43-51.

Labani, M., Moradi, P., Ahmadizar, F., & Jalili, M. (2018). A novel multivariate filter method for feature selection in text classification problems. *Engineering Applications of Artificial Intelligence, 70*, 25-37. doi:https://doi.org/10.1016/j.engappai.2017.12.014

Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing, 62*, 441-453. doi:https://doi.org/10.1016/j.asoc.2017.11.006

Miettinen, O. S., Steurer, J., & Hofman, A. (2019). The Bayes' Theorem Framework for Diagnostic Research. In O. S. Miettinen, J. Steurer, & A. Hofman (Eds.), *Clinical Research Transformed* (pp. 109-114). Cham: Springer International Publishing.

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology, 24*(12), 1565-1567. doi:10.1038/nbt1206-1565

P, K. A., & Acharjya, D. P. (2019). A Hybrid Scheme for Heart Disease Diagnosis Using Rough Set and Cuckoo Search Technique. *Journal of Medical Systems, 44*(1), 27. doi:10.1007/s10916-019-1497-9

Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Rajput, D. S., Kaluri, R., & Srivastava, G. (2020). Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. *Evolutionary Intelligence, 13*(2), 185-196. doi:10.1007/s12065-019-00327-1

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics, 23*(19), 2507-2517. doi:10.1093/bioinformatics/btm344

Saritas, M., & Yasar, A. (2019). Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification. *International Journal of Intelligent Systems and Applications in Engineering*, 7(2), 88-91. https://doi.org/10.18201//ijisae.2019252786

WHO. (2016). 50 Facts: Global health situation and trends 1955-2025. https://www.who.int/whr/1998/media_centre/50facts/en/

9. Appendix

Appendix 1: Risk Register

Heart Disease Angina Predictive Model Risk Register

Prepared by: Joshua Fehring, Utkarsh Patel, Luke Wilson **Date:** 9/6/2020

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact	Risk Score	Status
1	1	Lack of communication between team members	If there is a lack of effective communication between members of the team, It may lead to the progress not managed properly and project may fall behind.	Communication	Poor communication management	Not proper coordination within the team on how to make decisions on certain things, no team work	Schedule regular weekly meetings and checkpoints to ensure all group members are aware of any issues or concerns of other members	Project team members	7	8	56	High (open)
2	4	Technical/security Risk in the website	Software risk can occur for uncertain events. It could end up with data leakage as privacy might be the issue.	Software	Executing website programming	Not enough coordination with functional management and underskilled indiviidual for the website development	It's important to start early and learn in advance how to extract output of the model into UI component	Team members as a group	6	7	42	Medium (open)
3	3	Project not complete by deadline	Barely to complete on time	Timeline Risk	Planning	Not enough monitoring group members on how the tasks have been handled resulting they lose track of time	Set regular achievable milestones to ensure that we stay on track for completion	Individual	5	9	45	Medium (open)
4	2	Model selected earlier than expected	Early completion of the model selection process	Positive Risks	Planning	Too much dedication at the planning stage and while developing	Model can be further refined and extended upon	Project team members	6	9	54	Medium (open)
5	6	Losing individual's work	If not managed properly, individual can lose their workstation proving costly for the team since it would end up losing a large amount of progress	Lost work Risk	Processing	Getting carried away with the work and not too worried about the backup	Team members need to make sure they are working under version controlled software or make sure they upload or save their work freuently to avoid this risk	Project team members	4	9	36	Medium (open)
6	7	Risk of demand	The model used in the output may be impressively accurate (with high sensitive value) such that it positively risks of being in demand	Positive Risk	Model Implementation	Model risks of having too much load after it passes tests on testing datasets and become successful further	Model can be further put into practice after appropriate testing procedures.	Project team members	4	7	28	Medium (open)

7	5	Individual skills risk	Having insufficient skills to work with models and UI platforms to end up spending too much time learning and risking deadlines	Resource Risk	Planning	Not having enough knowledge about the specific programming languages, libraries, models, etc.	The team needs to make sure beforehand about tools and techniques each member is familiar with and learn accordingly well in advanve if need be	Project team members	5	8	40	Medium (open)
8	8	Domain Specific Risk	The implemented model not being able to handle new dataset to adapt to the changes in model	Domain	pre-processing	Not enough variety into original dataset and model properties	While implementing models, make sure to exlplore data and model in thoroughly	Project team members	3	8	24	Low (open)
9	9	Hosting models on online organization platforms	Since the model is hosted on organization websites, it could risk of being overused by the users	Positive Risk	Outputting the tested model	Since model is put on the organisation site in the output process, it can become too popular	The UI or website part of the process can be revisited further to make it smooth and easy for users to access it	group of organization and team members	3	6	18	Low (open)
10	10	Model misdiagnoses people suffering from coronary heart disease as healthy	If the model is not sensitive enough to reliably diagnose patients with coronary heart disease, patients who erquire treatment could be overlooked with serious consequences for their health	User Risk	Insufficient refinement and testing of model	Diagnosis of patients who have been cleared by the model as suffering from heart disease	Avoid this risk by implementing rigorous testing of the model before implementing it in order to verify its suitability for use	Project team	5	8	40	Medium (Open)
11	11	Model misdiagnoses healthy people as having coronary heart disease	If the model cannot distinguish well enough between healthy and ill patients, healthy patients may be diagnosed as having heart disease and referred for uneccessary further testing	User Risk	Insufficient refinement and testing of model	Diagnosis of patients who have been diagnosed as having heart disease as healthy in later tests	Avoid this risk by implementing rigorous testing of the model before implementing it in order to verify its accuracy	Project team	3	6	18	Low (Open)

Appendix 2: Requirements Traceability Matrix

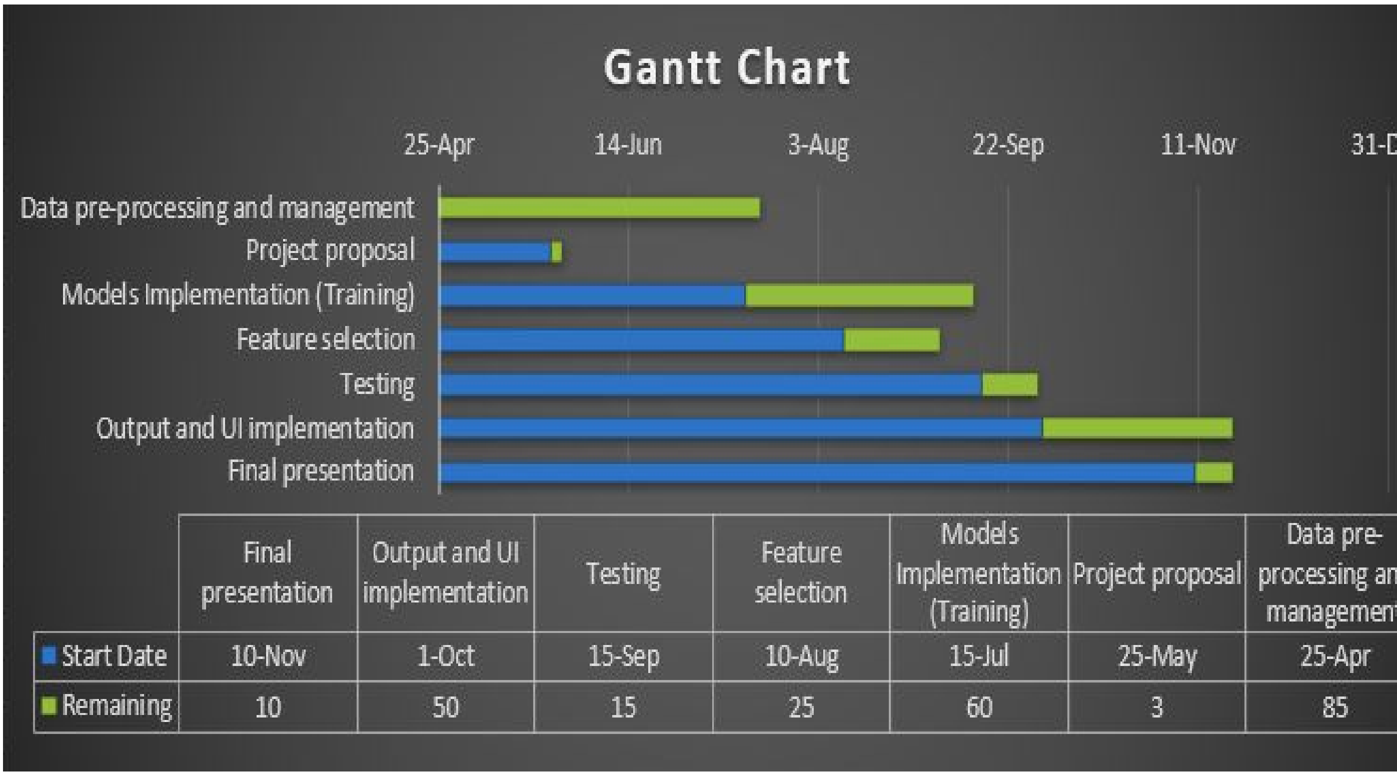
REQUIREMENTS TRACEABILITY MATRIX

Project Name:	Heart Diseas Angina
Project Manager Name:	FIT-3163 Staff
Project Description:	Aiming to develop a classification model that, given a number of observations and features, flags those observations that most confer the risk of angina and delivers a positive or negative diagnosis.

ID	***Requirements (Functional or Non-Functional)***	***Assumption(s) and/or Customer Need(s)***	***Category***	***Source***	**Status**
	Laptops	To share information with team members	Technical	Personal	Completed
1	Laptops	To share information with team members	Technical	Personal	Completed
2	Internet	To communicate with team members	Technical	Personal	Testing
3	Cleaned data	To easily train the model	Technical	Data preprocessing	Testing
4.	Imputation	To allow missing values to be replaced when it's necessary	Technical	Data preprocessing	Testing
4	Model with Interpretable outputs	To explore the model further for testing	Non-Technical	Training algorithm	In Progress
5	Model having the highest sensitivity value	To get the best model to be outputted	Technical	Testing algorithm	In Progress
6	Web Page / UI thread	To make it available for doctors to testing	Technical	Output part	In Progress

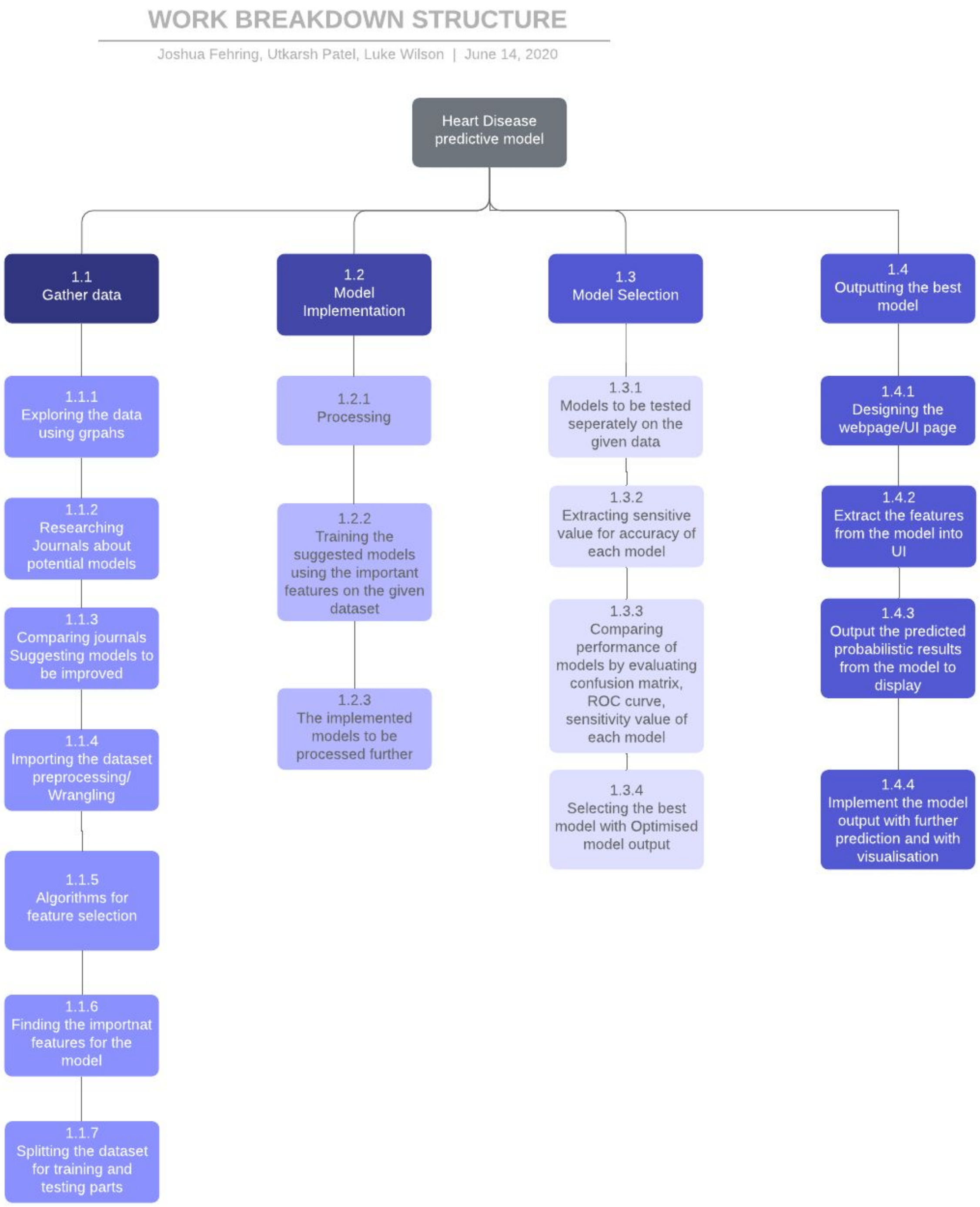
7	Visualisation	Models especially probabilistic outputs and performance evaluation may be more meaningful if they are visualised	Technical	Output	In Progress
8	*Extensibility*	System should be able to perform well on unseen data and adjust the model if data changes in any ways	Technical	Processing	Testing
9	Google drive	To collaborate team documents	Non-Technical	University module	Completed
10	Required programming languages on device	All team members are expected to have programming languages such as R, python, HTML, CSS, MySQL, etc on their devices to work with models smoothly and GIT to handle version control	Non-Technical	Personal	Testing

Appendix 3: Gantt Chart

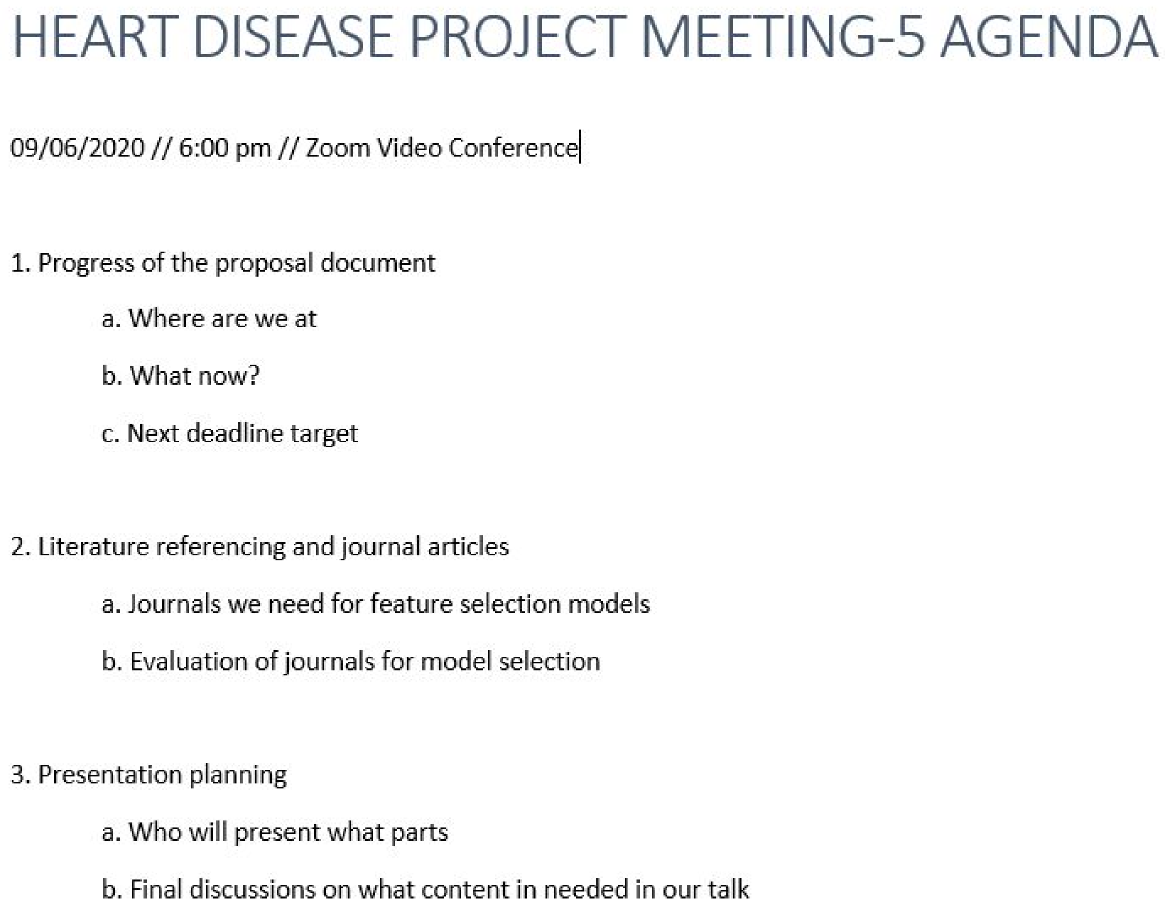


	Final presentation	Output and UI implementation	Testing	Feature selection	Models Implementation (Training)	Project proposal	Data pre-processing an managemen
🟦 Start Date	10-Nov	1-Oct	15-Sep	10-Aug	15-Jul	25-May	25-Apr
🟩 Remaining	10	50	15	25	60	3	85

Appendix 4: Work Breakdown Structure


WORK BREAKDOWN STRUCTURE
Joshua Fehring, Utkarsh Patel, Luke Wilson | June 14, 2020
Heart Disease predictive model
1.1 Gather data
1.1.1 Exploring the data using grpahs
1.1.2 Researching Journals about potential models
1.1.3 Comparing journals Suggesting models to be improved
1.1.4 Importing the dataset preprocessing/ Wrangling
1.1.5 Algorithms for feature selection
1.1.6 Finding the importnat features for the model
1.1.7 Splitting the dataset for training and testing parts
1.2 Model Implementation
1.2.1 Processing
1.2.2 Training the suggested models using the important features on the given dataset
1.2.3 The implemented models to be processed further
1.3 Model Selection
1.3.1 Models to be tested seperately on the given data
1.3.2 Extracting sensitive value for accuracy of each model
1.3.3 Comparing performance of models by evaluating confusion matrix, ROC curve, sensitivity value of each model
1.3.4 Selecting the best model with Optimised model output
1.4 Outputting the best model
1.4.1 Designing the webpage/UI page
1.4.2 Extract the features from the model into UI
1.4.3 Output the predicted probabilistic results from the model to display
1.4.4 Implement the model output with further prediction and with visualisation

Appendix 5: Sample Agenda



HEART DISEASE PROJECT MEETING-5 AGENDA

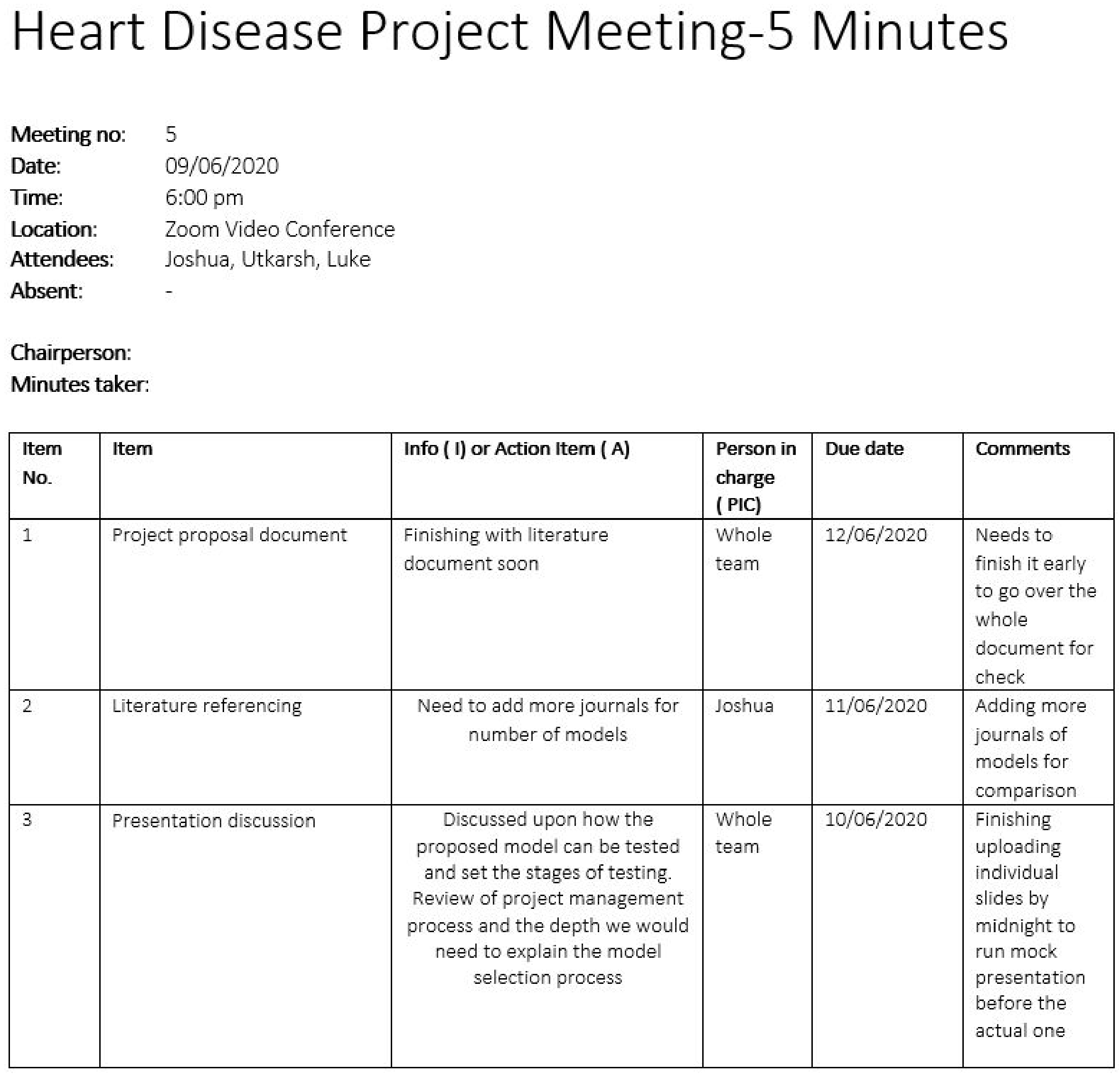
09/06/2020 // 6:00 pm // Zoom Video Conference

1. Progress of the proposal document
   a. Where are we at
   b. What now?
   c. Next deadline target

2. Literature referencing and journal articles
   a. Journals we need for feature selection models
   b. Evaluation of journals for model selection

3. Presentation planning
   a. Who will present what parts
   b. Final discussions on what content in needed in our talk

Appendix 6: Sample Minutes



Heart Disease Project Meeting-5 Minutes

Meeting no: 5
Date: 09/06/2020
Time: 6:00 pm
Location: Zoom Video Conference
Attendees: Joshua, Utkarsh, Luke
Absent: -

Chairperson:
Minutes taker:

Item No.	Item	Info (I) or Action Item (A)	Person in charge (PIC)	Due date	Comments
1	Project proposal document	Finishing with literature document soon	Whole team	12/06/2020	Needs to finish it early to go over the whole document for check
2	Literature referencing	Need to add more journals for number of models	Joshua	11/06/2020	Adding more journals of models for comparison
3	Presentation discussion	Discussed upon how the proposed model can be tested and set the stages of testing. Review of project management process and the depth we would need to explain the model selection process	Whole team	10/06/2020	Finishing uploading individual slides by midnight to run mock presentation before the actual one

Appendix 7: Team Member Contribution

Member	Contribution
Joshua Fehring	80% Introduction 70% Literature Review 10% Project Management Plan 20% External Design 60% Methodology 10% Test Planning 60% Conclusion 10% Editing
Utkarsh Patel	100% Front Matter 10% Literature Review 50% Project Management Plan 70% External Design 10% Methodology 10% Test Planning 10% Editing
Luke Wilson	20% Introduction 20% Literature Review 100% Project Overview 40% Project Management Plan 10% External Design 60% Methodology 80% Test Planning 20% Conclusion 80% Editing