Project report: Detecting heart disease in patients

Abrar Hamzah | 28551494

Tom Orlando | 28009592

Jian Tan | 28384695

Team 6 | 8,857 words

FIT3164 | Daniel Jitnah

Contents

Introduction

In today's society, cardiovascular diseases are the main cause of death around the world, according to the WHO which states that heart disease causes 17.9 million deaths each year. One of these cardiovascular disease is coronary artery disease (CAD), which leads to restricted blood flow as a result of blocked arteries from cholesterol and fatty deposits. The aim of this project is to design and implement various data mining and predictive modelling methods, in an attempt to predict the occurrence of CAD in patients, based on a particular set of parameters. This has been done through two key workstreams: 1) a predictive model and 2) an interactive user interface.

This document aims to build a complete and extensive project report for the use case described above. The report will include a project background section to set the scene for the project, a literature review to update readers on recent advancements in predictive machine learning methods, a project management plan describing relevant project processes undertaken, as well as discussions around methodology, project outcomes, and critical analyses of the final product.

Project background

Introduction & project context

With cardio vascular diseases being such an issue in today's world, this project aims to use data and state-of-the-art predictive models to help patients understand if they are at risk and to overall reduce CAD-related deaths around the world. To be clear, the aim of this project is not to diagnose someone from CAD; rather, we would like our product to be the "first step" for people to be aware of their heart condition whether they have a chance of CAD or not. Therefore, this product is aimed towards the general public who would like to know about their health condition. By implementing this project, we hope that our product will help reduce the mortality rate of coronary artery disease.

Although our product is fairly simple, our product has several functionalities and features that we believe will help achieve our main objective. The first good aspect of our product is it is aesthetically pleasing, interactive and user friendly. With the simple yet straightforward buttons and information shown, users will easily run through our product without any problems. The main functionality is the prediction process where first, the user can click a button that directs them to a form that needs to be filled. These values will then be used to predict CAD. The convenience that the form provides is that incorrect type of input will not be accepted and tooltips/pop-ups will be shown if a user is unsure about a specific feature. Therefore, users are assured that the product will run smoothly and correctly. Lastly, our product does not only give a statement on how likely you are to have a CAD but will also provide information on CAD such that users can know how to prevent it and what to do if they are likely to have CAD.

Updated literature review

On the literature review of the proposal, the main questions addressed were 1) what are the fundamental classification methods used in machine learning over the last decade? And 2) what are the leading classification methods that should be used to accurately predict heart disease in patients? Though these were successfully answered in our initial literature review, a new big question arose as the project was being implemented which was - can we improve the performance of the models suggested in our initial design and if so, how?

In this updated literature review, additional sections have been included, due to some changes along the course of the project. The first change, was implementing hyper-parameter optimisation, which allowed us to increase the performance of our models. The other, is that the final model for predicting CAD is based on an ensemble classifier which enabled us to receive better results and an increase of overall model performance.

Grid search

Machine learning models have parameters that can be set before training the model and those parameters can be optimised to get better models. Optimising the parameter of a function to get better performance is called hyper-parameter optimisation (HPO). As stated by Bergstra, Bardenet, Bengio and Kegl (2011), hyper-parameter optimisation is a compulsory step in the learning process. There are a lot of research papers that used hyper-parameter optimisation as their procedure when implementing their model. One study stated several advantages of using HPO. One advantage is that it improves the performance of the classification algorithm (Feurer & Hutter, 2018). Furthermore, Feurer and Hutter (2018) specified that another advantage is it “reduces the amount of human effort when implementing good machine learning”. Each machine learning algorithm has its own hyperparameters and applying HPO will reduce the time and complexity of finding which parameters are the best as well as what the best value is for each parameter. The main limitation of HPO that has been established early is that each approach will result differently depending on the datasets (Kohavi & John, 1995).

There are different approaches of HPO however, this review will explore a specific HPO, grid search. Grid Search CV is an approach which evaluates different sets of parameters from a given parameter grid to minimise loss or maximise accuracy. According to Feurer and Hutter (2018), a grid search approach is the most basic and easy to implement. Furthermore, grid search allows classifiers to have reproducible results (Young, Rose, Karnowski, Lim & Patton, 2015). This is due to the fact that the classifiers will always use the chosen parameters values. Another strength of grid search is it is a great tool for low dimensionality (Bergstra & Bengio, 2012). This approach has been used in many papers and showed with evidence that it holds its strength. A study by Wang, Gong, Li and Qiu (2019) on detecting epileptice seizures, showed that when applying grid search to random forest classifiers improved the accuracy from 88% to 96.7%. Not only accuracy, the study presented that other metrics also improved. They presented an AUC, after using grid search, increased to 99% which is almost perfect. In addition, another study performed the same approach on support vector machines (SVM) on different datasets and obtained improvement in performance and offered some computational improvements (Jiménez & Dorronsoro, 2007).

Although Feurer and Hutter (2018) stated that grid search is the most basic HPO method, they also mentioned that it has a computational problem or known as the “curse of dimensionality”. It is a problem that adding more parameters and values that need to be evaluated will exponentially increase the complexity of the algorithm. This statement has also been proven by our findings when applying grid search to an ensemble classification consisting of neural network, decision tree, SVM and logistic regression. Our algorithm did not halt at all meaning it took a long time when grid searching numerous hyperparameters for each classifier. Therefore, an alternative suggested by Bergstra and Bengio (2012) is using a random search approach. In the study, they showed that random search performed better in high dimensionality problems although not as much when dealing with low dimensionality.

Overall, from all studies mentioned above, it is clear that implementing hyper-parameter optimisation before training the model could improve significantly in performance. Especially, applying Grid Search CV on neural networks and SVM could improve the accuracy and reduce the error percentage.

Voting classifier

Ensemble Learning is known as the state-of-the-art of machine learning (Zhou, 2012). It is a method that is highly acknowledged for its performance and success. There are several ensemble methods however, the way ensemble methods work in general is by building a machine from “base learners” or multiple classifiers and combining them into one predictive model (Karlos, Kostopoulous & Kotsiantis, 2020). A variety of approaches to ensemble learning are bagging, boosting, stacking and voting.

A single learning algorithm can suffer from three problems such as high variance, high computational variance and high bias (Zhou, 2012). Therefore, through the combination ensemble method, the variance and bias may be reduced. As stated by Zhuo, this has been confirmed by many empirical studies. The combination ensemble method has different methods itself: simple average, weighted average, hard voting, plurality voting, and soft voting. However, this review will be focusing on hard and soft voting.

One type of voting method is soft voting which calculates the average probability of all base learners. However, there are limited studies that use this method as classifying health problems. Therefore, this review will explore the performance in different fields. One study on predicting the interfacial tension (IFT), as one of the transformer oil test parameters (Hassan & El-Hag, 2020), showed that soft voting technique outperformed one classifier. The accuracy of their proposed soft voting ensemble method achieved an accuracy of 87% whereas Naive Bayes, logistic regression and support vector machine achieved 83%, 83% and 75% respectively. Not only accuracy improvements, other metrics such as F1 score also proved had improved using soft voting. A study by Zhou, Zhang and Wu (2018) showed that the baseline of the F1 score is 0.599 and after utilising soft voting, it dramatically improved to a F1 score of 0.685.

Hard voting or known as majority voting is the most popular method. Every classifier has its own vote to predict a class label; the voting classifier will then combine it all and take the most voted class as its final output class label (Zhuo, 2012). One study on classifying diabetes by Bashir, Qamar, Khan and Javed (2014), showed that ensemble methods could improve the performance of the model significantly. In their study, they applied majority voting technique on three different decision trees. The result of their work was astounding; it outperformed Naive Bayes, SVM, K-NN, Random forest and single decision trees. The comparison is shown on the table below.

Technique	Accuracy (%)
Decision tree	41.26
SVM	74.8
K-NN	78
Naïve Bayes	74.61
Random forest	73.5
Ensemble method (majority voting)	89.33

Another study on a similar topic to our project on predicting a heart disease, showed that combining different classifiers will improve the performance for the final model (Raza, 2019). In his study, Raza showed that the ensemble method outperformed some fundamental learning methods on predicting heart disease with an accuracy of 88.88% compared to decision trees (73.79%) and logistic regression (85%). It was also mentioned that other metrics on evaluating a model improved. The proposed method was able to get 0.87 F1 score, 0.85 precision and 0.88 AUC.

In conclusion, as proved by the studies above, implementing an ensemble method could increase all metrics evaluation of a model significantly. The most common method used in those studies is voting classification.

Evaluation of literature & conclusion

To summarise the initial literature review, there are various classification methods used to predict and classify world problems specifically health problems. The methods discussed - which included logistic regression, naive Bayes, decision trees, SVMs and ANNs - answered the question, showing what the fundamental classification methods used in machine learning over the last decade. The other question posed on the initial literature review was which classification methods are the leading methods to be the benchmark of other classifiers. Logistic regression and naive Bayes classifiers will act as performance benchmarks to contrast and compare the three models, decision trees, SVMs and ANNs.

Furthermore, the information gathered above has answered the bigger question posed at the start of the review, which is how to improve the performance of the models. It has been stated through various studies that hyper-parameter optimisation and ensemble methods, specifically grid search and the voting classification ensemble method, can greatly improve the performance and reduce errors in the model.

Methodology

In order to achieve our aim, which is to help the general public to be aware of their heart condition by predicting the chance of CAD, the project has four phases that need to be done. The first phase - data pre-processing is where all data cleaning and normalisation are done. The next phase is done to understand more about the features available to classify positive or negative of CAD. Various feature selection algorithms were used to achieve the aim of the second phase. The third phase - classification is where the filtered data resulting from phase two is used for modelling to develop a machine learning that predicts CAD accurately and reliably. Finally, web application implementation and integrating the back-end and front-end are done in phase four using Django Framework.

Phase one - Data pre-processing

The first stage is to import the dataset and to pre-process the data. When pre-processing the data, we did it in two parts: data cleaning and data normalisation. Data cleaning is where we find missing values, unimportant features and also encode all categorical features except the target variable. When doing this, we found one feature 'CP' that is unimportant due to the fact that all records show 'N'. This is considered unimportant because a feature that has exact values across the dataset will not contribute much when predicting CAD meaning, it will not improve the performance of our model. Therefore, the first step is to discard all unimportant features. The next step is to find missing values. Fortunately, there are no missing values in our dataset. Furthermore, we encode all the categorical variables; specifically we one hot code all the categorical variables. One hot coding is a method of encoding categorical variables that transform a variable with x distinct values into x new binary variables representing the distinct variable. The reason being is because as stated by Potdar, Pardawala and Pai (2017), machine learning algorithms can only accept numeric values as input, otherwise it will discard the meaning of each category values.

The second part of our pre-processing stage is normalisation. Data normalisation is a process of scaling the data into a range of numbers typically zero to one range. It is a useful process because it could help or improve the prediction of our model (Patro & Sahu, 2015). The specific algorithm used for normalising the data is Min-Max Normalisation from scikit-learn package. Min-max normalisation converts the values on the original dataset into a range of values and we chose to normalise the dataset into 0 to 1. The steps mentioned above are compulsory in machine learning and modelling because real-world data are not perfect -- it contains noise and it may be incomplete (Singhal & Jena, 2013).

Phase two - Feature selection

Feature selection is also an important process when it comes to machine learning. Data has increased dramatically over the decade and it is expected to still increase over the upcoming years (Guyon & Elisseeff, 2003). With the increase of data, numbers of features also increased which resulted in several problems. Dash and Liu (1997) and Guyon and Elisseeff (2003) mentioned that applying feature selection helped improve the performance such as prediction accuracy. Furthermore they stated that reducing features used for prediction will reduce the complexity and run time of the machine learning model.

For our project, we used four feature selection algorithms as the base algorithm: correlation, stepwise, lasso and metaheuristic algorithm. We implemented a simple correlation algorithm which only finds features that are highly correlated with each other. The approach is to use a threshold to decide if two features are correlated. We tried different threshold values such as 0.7, 0.5 and 0.3. For values of 0.7 and 0.5, only a few features were correlated with each other. Therefore, we decided to reduce it to 0.3. After finding all the features that are correlated, we dropped it from the original dataset resulting in only 17 features left. The second algorithm for feature selection is stepwise regression. Implementing stepwise regression was easy, we only need the linear model built-in function from R. To know if the set of predictors are the best predictors, we see the value of AIC. The lower AIC the better the model. At the end, we were able to get an AIC of -460 with only 20 features that are important for predicting CAD.

Next is the Lasso method which is a form of penalised regression which uses an L1 norm as a regulariser and unlike Ridge regression, its norm regulariser drives parameters to zero, effectively deleting non-important variables. This is implemented using the sklearn package in Python. Lastly, we implemented a metaheuristic algorithm. The way metaheuristic algorithms work is based on the notion of natural selection where the fittest set will be used to create a new generation. This process will be done in loops until we find the best or fittest set of features and this can be done using any base classifier to evaluate the performance of fitness. Finally, all the features from all algorithms will be filtered to only 15, sorting by the most frequent features. The reason behind choosing top 15 most frequent features is that we want to keep our model fairly simple but still perform well. The features selected are listed below.

- typical_chest_pain
- Age
- DM
- HTN
- FH
- TG
- K
- region_rwma
- BBBN
- Tinversion
- FBS
- ESR
- EF.TTE
- DLPY
- diastolic_murmurY
- CAD_Yes

Phase three - Classification & evaluation

This phase will make use of different classifiers to accurately classify CAD positive or negative. These classifiers will then be combined using an ensemble method and used to compare all performances. Furthermore, we also applied grid search onto each classifier to improve the performance. The initial thought stated on the proposal was to get most classifiers to be above 75% accuracy. There is no specific justification as to why we put a threshold of 75% however, most studies have been able to show their classifiers above 80%.

We wanted to have a model that is solid and reliable especially on predicting a disease that has the highest rate of mortality. Therefore, there were several iterations of the models and from each iteration we were able to get performance improvements above 80% accuracy.

The first machine learning model is the decision tree. According to studies, it is one of the easiest methods to analyse data and known to perform well enough on predicting. Implementing decision trees only need the tree function from scikit-learn package. The performance of a simple decision tree was not overwhelming with only 75% accuracy. The next simple classifier that has been used many times in studies is naive Bayes. We used BernoulliNB and GaussianNB function from the same package scikit-learn and after evaluating it, BernnoulliNB performed better than the decision tree with an accuracy of 83% and 71% accuracy from GaussianNB. However, in the second iteration, we used cross-validation and it improved both models to 85% and 80% accuracy respectively. Other models are SVM, logistic regression and neural networks. SVM is a supervised learning model that analyses data given a set of training examples. Scikit-learn package provides a simple function called SVC that does the job of SVM. Like any other classifiers, we only need to initialise these models using its own function SVC(), logisticregression() and MLPClassifier for neural networks, and train it afterwards (note that these functions can be easily found on scikit-learn package). In the end, our classifiers performed better than our threshold ranging from 80% to 88%.

Finally, these classification models are used as a base learner and combined for our ensemble method, voting classifier. Voting classifier function can be found in scikit-learn package. The function itself allows two parameters: estimators and voting type. The way this function works is by taking a list of estimators or base learners and evaluating the performance based on the specified voting type. There are two voting types: hard and soft. Hard voting is where the algorithm takes the majority outputs as its final output whereas soft voting systems take consideration of the probability of each classifier in classifying CAD and average them as its final performance model. On the first iteration of our final model, we implemented it without any cross-validation and grid search. The performance of it was quite well with 84% accuracy and around 80% f1 score. As stated in some studies, using grid search will improve the performance of a model. Therefore, we applied grid search and cross validation into the model and trained the model, as a result, our ensemble method was able to perform solidly. Doing all these steps gave us a solid model with an average accuracy of 88%, an f1 score of 92% and AUC of 92%.

Phase four - Website implementation

In this phase, we implemented the website using Django Framework. Django is a framework that uses Python as the programming language. The reason behind using Django as our approach in creating a website is because to reduce complexity on integrating the back-end to the front-end. Since our model was implemented using Python, using Django framework will smooth the transition from back-end to front-end. To import the model to the front-end, we just need to run a one-line code from joblib and pickle packages that created a pickle file containing our final model. Another reason for using Django is that Django itself provides some data input security to some degrees. Some data security provided are:

- Only accept correct type of input
- Provide a drop down menu for categorical features
- Provide an error and tooltips for unsure features

In order to work on a web-app using Django, we need to create a folder for compiling different components of the Django architecture. This includes all various applications folders (note we only use one app - cloud_website folder) under the web folder, as well as all the backend files such as manage.py and urls.py. Most of the required code is used in the cloud_website folder, where three main folders are found, as well as the general Django admin files which are used to configure the web platform. The models file simply stores the saved model from the previous phase (specifically the voting classifier and scaler file to scale the input of the user). The static folder contains all the styling, image and js modules used for the website. The template is where all of the html files which underpin the content of the web-application are found.

As mentioned earlier, there are urls.py, models.py, forms.py, and views.py. The urls.py file is what allows users to control the available urls for the web-application. For our product Hearty, this includes four key pages and a few admin pages for back-end access. These are directly linked to the views.py file and will call functions stored there when specific urls are input into a browser. The forms.py and models.py files include all the information for the form input for the user and leverage Django's underlying features. Finally, views.py is the most important file that needs to be created. It is needed to connect the back-end with the front-end. All these files and components are essential to create our product Hearty, a user-friendly web-application that predicts CAD.

Resources used

There are only a handful of resources needed to achieve the aim and completion of this project. The only hardware used is our own computer and laptop. For implementing all algorithms and implementation, some of us used PyCharm and RStudio but we mainly used Visual Studio Code to share and update our code on the Github repository. We used Visual Studio Code because it provides convenience to run multiple programming languages. However, there is no obligation to use the software mentioned above; any other software that can run Python and R will suffice. R language will be used for the second stage of the development of the project which is Feature Selection. Whereas Python is our main programming language -- we use it on every step of the project.

There are a couple of reasons why we chose Python as the foundation of the project's programming language. First reason is that Python offers many packages for machine learning such as scikit-learn; it has all classifier algorithm functions. This resulted into the second reason, we found a function from the scikit-learn package called "votingClassifier". This function allows us to combine all classifiers/"base learner" to be evaluated together and outputs a new predictive model. This new model will be our final model used to predict Coronary Heart Disease. Furthermore, we used Django Framework which is written in Python. The output of our final model can be easily implemented onto the back-end of our website by dumping a file with a pickle extension. An advantage of this is the framework will automatically recognise the model. Overall, using Python and Django for the foundation of this project provides convenience, efficiency and an ease of project lifecycle.

Project management

Project overview

The aim of this project is to design and implement various data mining and predictive modelling methods, in an attempt to predict the occurrence of coronary artery disease (CAD) in patients, based on a particular set of parameters.

The final implementation of the product consists of two segments: 1) predictive model which uses machine learning and statistical methods 2) an interactive and responsive user interface in which the model is integrated. These two segments combined aim to provide an accurate and reliable tool, to be the "first step" for people to be aware of their heart condition and whether they are at risk of CAD.

Project management methodology

The project management method we have used throughout the project is the iterative method. This is an approach which allows feedback for unfinished work to improve and modify specific areas on an iterative and ongoing basis. This method stresses an approach of building "imperfect deliverables" first, and through feedback from stakeholders, gradually tweak and fix features until an adequate level is reached.

This method has been chosen based on the nature of this project being partly an inquisitive study to find algorithmic methods to best predict CAD. As there is no clear cut definition for the best model currently, there will be a lot of model testing with various combinations and feedback required and an iterative approach will prove most useful. This is similar with the website app deliverable, where an initial product was built and through feedback and testing, was tweaked to ensure the best performance and that all user acceptance criteria are adequately solved.

To adequately follow the methodology, we made sure to implement various processes to ensure the project was well managed and all processes were completed in line with the initial plan. The imperative tool used to achieve this, was our weekly communication through Zoom, with both team members and sometimes tutors. In these discussions, we would often bring up issues and try to understand how to solve them, drawing on various insights from different group members. There was often times when we would hit a road block and one of us had seen a problem solved in a similar way in previous unit or from a trusted resource. These meetings allowed us to continuously iterate through the current state of the project and to quickly solve issues. This was especially catalysed when experienced tutors would join the conversation and provided guidance and feedback based on what we had done. Though Zoom meetings were the primary tool used to follow the methodology, various other methods were also implemented such as the use of Github, Messenger and Google Drive.

Milestones reached in project execution

As shown in the WBS and Gantt chart in *Appendix 3* , the project had three key milestones, each with a designated role. The role needed in the first milestone is that of the data scientist. Their main task is to pre-process the data and provide ideas as to how we should approach the modelling based on their findings. In addition, the data scientist creates the model needed to predict heart disease and reviewed all code to test different algorithms. This has ensured that all algorithms run properly and will produce correct output or expected results on predicting CAD.

Our goal of the project is to create an interactive website. Therefore, the nature of the second milestone is around user interface and interactivity. The webpage was functional within a week to enable time to fully implement adequate testing and debugging. Lastly, the third milestone was to review our project through testing and monitoring. This stage was done throughout week 10 to make sure our software runs smoothly and correctly just before the submission date.

Project scope

Project deliverables

Project deliverables include everything that we've produced as a team for the project. These can be categorised into two succinct groups which include 1) user deliverables and 2) administrative deliverables. User deliverables include everything that has to do with the final product and that users will be able to see or use in some way. This will include the web application itself, pages with information on CAD included in the website, predictive functionalities (i.e. underlying predictive models) and accurate prediction outputs. On the administrative side, deliverables include all of the work that has allowed us to initially build the product and manage the project. These include a project charter, project schedule, literature review, WBS, scope statement, time and risk management tools, status reports, project proposal and this final report.

Product characteristics and requirements

After determining all deliverables, all the needs of the project have also been recorded. Product requirements define the values and purpose of the product, the characteristics of the final output and the necessary functions that the final deliverable must have in order to satisfy user needs. These include:

- the ability to compare different feature selection algorithms
- the ability to compare different classification algorithms
- the ability to accurately predict whether a patient has CAD based on their unique characteristics
- Ability to upload their personal data to the database through the UI
- Provide a clear and direct dashboard on UI for users

- Users can easily operate and use comprehensive functions to get their own results they want
- When an operation error occurs, an exception report is submitted and tips are given to correct

Product user acceptance criteria

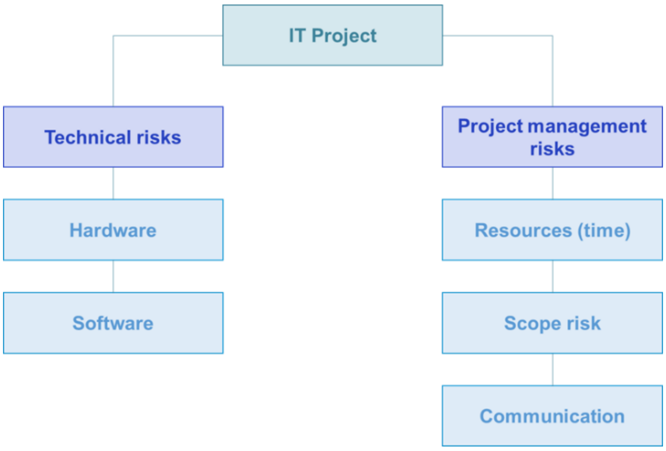
Product user acceptance criteria allows us to determine if we have achieved user objectives in the final product. This includes criteria from the user deliverables described above, ranging from the look and user experience of the website to the handling of errors and general functionality of the site. These have been described in more detail in the table below, with their corresponding completed quality rating and overall score.

Criterion	Completed quality	Score
Accurate and timely prediction	HIGH	86
Efficient and intuitive design	HIGH	85
Informative website	MEDIUM	75
Ability to use website functions for desired results	HIGH	90
Website errors are minimal and dealt with efficiently	HIGH	84

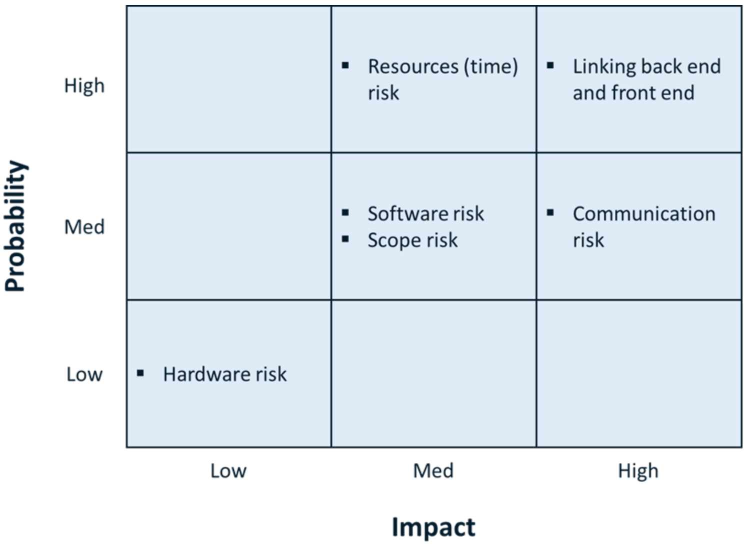
Risk management

Project risk management is the process of identifying, analysing, and responding to risk in the best interests of meeting project objectives, throughout the life of a project (Schwalbe, 2015). It is important to understand there are both positive and negative risks commonly arising throughout projects, both of which carry their own effects on meeting project objectives. This section aims to describe the risk management process for the project which includes identifying, analysing / prioritising and managing / controlling project risks.

The following *risk breakdown structure* (RBS) aims to identify the hierarchy of risk categories within the project, which allowed us to identify and categorise risks. These risks were initially identified through a *brainstorming* session, as a way to explore the potential risk landscape with a consensus amongst group members.



Once risks have been identified, further steps are taken to analyse and prioritise risks. This is initially done through the use of a *probability / impact matrix* to prioritise the most important risks and gauge the potential cost they might bring to the project (see below). Subsequently, a *Top Ten Item Tracking* method - built leveraging the impact matrix - is used to maintain an awareness of prioritised risks throughout the life of the project. At each iteration, the team made sure to list the current ranking, previous ranking, number of times the risk has appeared and a summary of the progress made in resolving the risk item. Note an in-depth risk register is also included in *Appendix 1.*



	Low	Med	High
High		▪ Resources (time) risk	▪ Linking back end and front end
Med		▪ Software risk ▪ Scope risk	▪ Communication risk
Low	▪ Hardware risk		

The complete risk management process described above was implemented as an iterative and dynamic process, with constant monitoring and ensuring that risk awareness is communicated within the team on an ongoing basis, through regular catch ups and discussions. This was done successfully through effective team management processes, which are summarised in the next section.

Team management summary

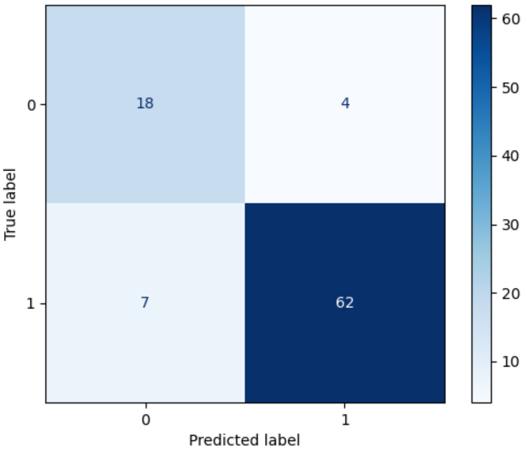
The overall team management of the project was deemed successful by all team members, as a result of careful planning, open communication and leverage of each other's unique skills and backgrounds. The success of our team management throughout the project was not just based on the how well we implemented the product, but also on how well the team was able to overcome issues and learn lessons from mistakes that were made. This was done particularly well and allowed the team to adapt to change and to be nimble in their approach to not only software development but also project management and teamwork.

When reflecting on the project as a team, we can see there are some minor points we would like to fix, or arising issues which seem clear now and we would all know how to change this in the future. One clear example and one of our main shortcomings (further described later in this report), was the decreased communication in the final stages of the project, which lead to some work being duplicated. After the initial issue was fixed, the team came together to understand why this might have occurred and how we can make sure this doesn't happen in the future. Overall the team is very pleased with both the outcome and the experienced gained in finalising this project.

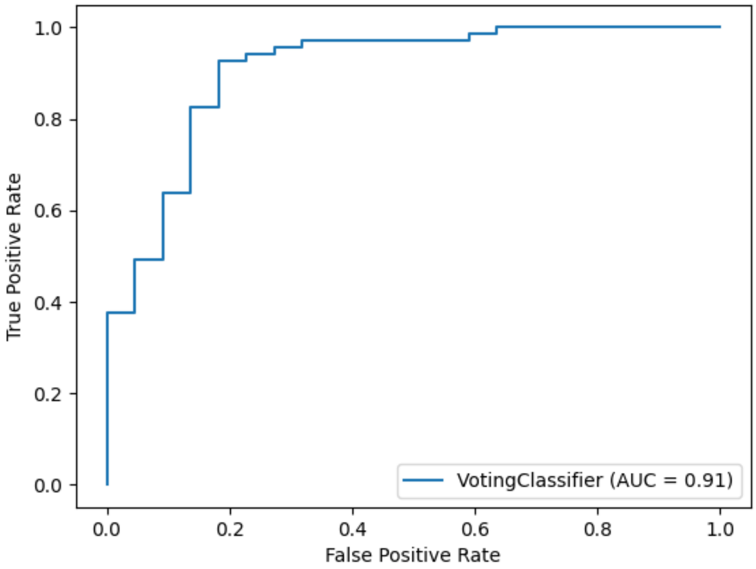
Project outcomes

Achievements & results

The overall project has turned out very well, with most of the refined deliverables (i.e. deliverables which were revisited throughout the project implementation) being successfully delivered. Our team was able to successfully implement all required feature selection algorithms, which were imperative in reducing the complexity of the classification models, as well as making the product more user friendly in the front-end.



Using these, the team was able to build classifiers with high prediction accuracy, sensitivity and F1-scores. All these were successfully implemented and optimised using a Gridsearch algorithm, allowing us to find the best combination of parameters resulting in highest accuracy. These were all then combined to make a final predictive model using a VotingClassifier and finally implemented into the back and front-end using Django. Gridsearch and VotingClassifer were used, as research found these are widely adopted methods of optimising classifiers and have the potential of greatly improving accuracy. The final model had its highest accuracy recorded at 0.93 with an average accuracy of 0.89. The F1-score for the model averaged at 0.92 whilst the AUC averaged around 0.9. Precision was highest with an average of 0.94 whilst recall average around 0.9 also. Obtaining a highly accurate predictive model was one of the team's best achievements throughout this project.



Another key achievement, was overcoming the risk of having little deployment and web development experience as a team. This was seen as one of the key flaws in our project plan and would take an extensive amount of work to overcome. However, through research and teamwork we were able to successfully implement a working product with an emphasis on the users interface and experience, which has proved delightful to use. The combination of a highly accurate classifier and a professional UI / UX design is what has made this project a success.

Achieving project requirements with software

The various requirements described at the start of the project, were iterated and changed as we started to better understand our abilities and limitations, as well as timing requirements to complete the project punctually. Notably, we were able to distil all requirements to two key elements 1) a highly accurate classifier for heart disease and 2) a delightful user interface and experience when using the product. These two overarching requirements were chosen as they represent fundamentally what all great products in today's society are made with, and what most of the top companies in the world strive to achieve.

As described in the results above, the team went above and beyond to build a highly accurate classifier for heart disease. Through the use of both feature selection and classification algorithms, the results of the model are classified as state-of-the-art when it comes to predictive accuracy. As for the user interface, extensive work was put in to the small details such as effective branding, website animations, 3D effects and memorable colour schemes. All useability features have also been rigorously tested to make sure no bugs or issues arise, dampening the user experience.

Though these implementations have all been successes, there are still some unforeseen shortcomings which emerged as the project developed. Mainly, the issue that the features required for the user to input in order to get a prediction can be quite hard to attain, if the user doesn't have specific medical information on hand. This might include features such as Potassium levels in the blood or whether a patient currently has symptoms like chest pain which are hard to properly diagnose. As the model and product require these inputs, it won't be able to make a prediction without something from the user.

A possible solution to this issue is to add in the option of users adding in "unsure" inputs, which will then permit the predictive model to ignore certain features and not use them in the prediction at all. Though this could cost predictive accuracy in the long run, it will be better than users being frustrated with and unable to use the product, without access to this medical information.

Overall however, the two key requirements have been successfully met by our product and the shortcomings described above could easily be addressed with more time and resources available for this project.

Summary of test report

Testing of our product was particularly important, in line with our key requirements to have a great front-end user experience. This meant rigorously testing how a potential user might use the product to ensure no input or any type of interaction would lead to errors. Based on this, we have chosen a manual approach to testing (rather than an automated testing algorithm), as a way to properly test out the carefully planned and thought-out user experience and to also add in new ideas and features based on discoveries from the test.

The test itself has been split into three key segments 1) unit testing 2) performance testing and 3) useability testing. The unit testing section aims to run specific tests on user inputs to see how the product will process edge cases like negative values, strings and a mix of both. In all the tests, a robust and clear response was returned, all of which leverage the underlying Django Forms infrastructure making it easy and reliable to make a robust product.

The performance testing section aims to see if model predictions are indeed accurate and whether or not the right outputs are provided to the user. This section also includes how the product itself performs when offline. All tests were successfully passed, however based on the product's design as a web application, it is unable to work offline.

The final useability section looks at how the product performs on various devices (mobile & tablet), different browsers and how the code base runs from a back-end perspective if the repository is downloaded. All tests were successfully passed. Overall the test report, reiterates the reliable and robust quality of our product, which in large part is due to the leverage of pre-built frameworks such as Django, to deliver the highest quality product possible to users.

Critical analysis of outcome

This section aims to explore and critically analyse the successes and failures which arose throughout the project, in particular in the methodology used, how the project was managed and the overall outcome of the project. This will provide a reference for better understanding what aspects of the project should be used versus what drawdowns must be abolished when undertaking a similar endeavour in the future.

Critical analysis of methodology

The methodology used in this project encompasses all the various methods, tools, and materials the team has used to build the product from scratch. This broadly includes items such as coding languages used, frameworks or packages leveraged, and algorithms implemented. Overall, the planning and implementation of our methodology has proved to be one of our strongest achievements in this project.

This success of our methodology is primarily due to two key factors 1) the careful planning of all methods designed and implemented and 2) the flexibility of all methods designed and implemented. Careful planning begun before the practical aspect of the project started and heavily leveraged information found in the literature review. The extensive research undertaken for the review allowed for the entire team to have an in-depth understanding in how these types of projects are designed and what makes them successful. This was particularly helpful in understanding how to implement various predictive models, how to accurately measure results, understanding flaws and managing implementation risks.

The flexibility of methods designed and manner of managing the project also played a large role in a successful methodology. This became particularly apparent when the team ran into some issue using the VotingClassifier, which is a module leveraging Python code and a few of our predictive models were written using R. We understood quickly we needed to pivot all our models to be built in Python to ensure we could successfully combine them as we planned prior to commencing the project. The ability to do this quickly and without any issues was in part due to project and team management, but mainly to the way we had designed our methodology plan. This included using simple and easy-to-use packages (like *sklearn*) for the individual models, with little tuning and optimisation as a way to streamline the process. Only once all models had successfully been implemented, we combined them using a VotingClassifier and only then started to optimise using Gridsearch and a range of other tuning methods. Overall the methodology of the project was implemented flawlessly with very little issues to provide an optimal product. Other aspects of the project however, were not done so successfully.

Critical analysis of project management

Project management includes all the various administrative and non-technical factors which play a large role in the overall success of the project. In particular the main sections of the project which played a critical role (both positive and negative) in the completion of the project include shortcomings in project scope and team management. Both of these will be critically analysed in the following segments.

One of the primary shortcomings in the management of the project was the diminishing size of the project scope as we came closer and closer to the due date for various deliverables. When initially set out in the project proposal document, the scope had been clear in what was to be included. However, as time went on and the project progressed, the team often found ourselves omitting parts of the project as a way to ensure all aspects of the scope being completed, were done at the best possible standards. This occurred when deciding to cut out one of the information pages on the web application. The initial page was supposed to include information as to how results were gathered and to build trust and credibility with users. As we arrived at a critical point in the timeline leading up to the end of the project, we made the decision to drop the extra page as little value would be extracted for the amount of work required. In future, such an issue could have been foreseen earlier and a discussion to perhaps share the workload of the new page could have taken place. As a team it was a learning experience to understand the importance of looking ahead and trying to see potential risks coming to the surface. A similar shortcoming was found in the general team management of the project.

Though our team management throughout the project was robust and well implemented, as deadlines got closer, it was clear some aspects of team management were starting to lose their efficiency. Notably, communication dropped to a level lower than it had been since the start of the project as the team had a lot of work starting to build up from other units and assignments. This led to a few different issues, but specifically an instance where work was duplicated and completed by two team-mates separately, leading to confusion, and the decision of having to choose between the two instances in an efficient and quick manner. Though this did not cost much in the overall completion of the project, it showed us the importance of maintaining open communication even when there is a lot of work and must be done better in subsequent projects. In the future, it is clear communication has to be at the forefront of a successful project and take utmost importance, with emphasis continuously updating team mates on recent task completions.

Critical analysis of project outcome

When looking at the overall project outcome, though there have been shortcomings in some useability factors of the product, it is clear the Hearty product is successful and the project team has accomplished what we initially set out to do. The minor shortcomings mentioned lie primarily around the useability of the product for users with limited access to medical information, as some of the features required for a prediction to be made can be relatively hard to come by unless having previously visited a medical professional.

This shortcoming was a product of the team being focused on details and losing sight of the big picture. This became more apparent, once the product was nearing finalisation as we were able to take a step back and understand exactly how our product could be used from a user's perspective. A mistake like this, could have also been foreseen in the planning stage and seen as one of the risks, or at least as soon as the list of 15 best features to use was gathered in the early stages of the project. There are however ways around this issue, and with more time and resources a fix could be made, such as adding in "unsure" inputs, which will then permit the predictive model to ignore certain features and not use them in the prediction at all. In the future, the team has understood the importance of taking a step back from the detail to take in the full picture, and make decisions accordingly. Small changes in attitude, in particular always trying to look at things from a user's perspective can make a large difference to software and is a skill that is widely sought out in professional circles.

Despite the shortcomings discussed, as a team we strongly believe the Hearty product and the overall project has been a success. Both on a tangible basis for the actual product delivered but also on a personal development level for all team members involved. We have learn many lessons, from the importance of working in a team and maintaining strong communication, to promoting ‘big picture’ thinking and looking at things from a user’s perspective. This, coupled with essential skills in project management, product development and report writing have made this project a formidable learning experience.

Conclusion

Throughout this project report we have described all the information used to effectively design, build, test and maintain our working product. First, we introduced the project and gave an update on recent research in the space through a literature review, where further research on new methods used that weren’t in the initial project proposal were added. Next we discussed the methodology including how everything from machine learning methods to front-end integration was implemented. We then discussed extensively how the project was managed through a detailed project management report, which provides insights into how we managed our process model, scope and requirements. This was followed by an understanding in what the project outcomes were in terms of performance and achievements. Finally, a critical analysis of the project was undertaken to better understand successes and shortcomings we made and the lessons learnt for the future.

In doing so, we are able to provide a complete project report, with the notion of describing the team’s success in bringing together the different expertise and backgrounds from our members, in an effort to build the Hearty product.

References

Bashir, S., Qamar, U., Khan, F. H., & Javed, M. Y. (2014). An Efficient Rule-Based Classification of Diabetes Using ID3, C4.5, & CART Ensembles. 2014 12th International Conference on Frontiers of Information Technology, 226-231. doi:10.1109/FIT.2014.50.

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, 282-305.

Bergstra, J., Bardenet, R., Bengio, Y., & Kegl, B. (2011). Algorithms for hyper-parameter optimization. NIPS'11: Proceedings of the 24th International Conference on Neural Information Processing Systems, 2546–2554.

Dash, M., & Liu, H. (1997). Feature Selection for Classification. Intelligent Data Analysis, 1(1-4), 131-156. Retrieved from https://doi.org/10.1016/S1088-467X(97)00008-

Feurer, M., & Hutter, F. (2018). Chapter 1 Hyperparameter Optimization. 3-4.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, 1157-1182. doi:10.1162/153244303322753616

Hassan, A. N., & El-Hag, A. (2020). Two-Layer Ensemble-Based Soft Voting Classifier for Transformer Oil Interfacial Tension Prediction. Energies, 13. doi:10.3390/en13071735

Jimenez, A. B., LAzaro, J. L., & Dorronsoro, J. R. (2007). Finding Optimal Model Parameters by Discrete Grid Search. Innovations in Hybrid Intelligent Systems, 120 127.

Karlos, S., Kostopoulos, G., & Kotsiantis, S. (2020). A Soft-Voting Ensemble Based C Training Scheme. Algorithms, 13(1). doi:10.3390/a13010026

Kohavi, R., & John, G. H. (1995). Automatic Parameter Selection by Minimizing Estimated Error. Machine Learning Proceedings, 304-312. Retrieved from https://doi.org/10.1016/B978-1-55860-377-6.50045-1

Raza, K. (2019). Chapter 8 - Improving the prediction accuracy of heart disease with ensemble learning and majority voting rule. U-Healthcare Monitoring Systems, 1, 179-196. Retrieved from https://doi.org/10.1016/B978-0-12-815370-3.00008-6

Singhal, S., & Jena, M. (2013). A Study on WEKA Tool for Data Preprocessing,. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2(6), 250-253.

Wang, X., Gong, G., Li, N., & Qiu, S. (2019). Detection Analysis of Epileptic EEG Using a Novel Random Forest Model Combined With Grid Search Optimization. Bioelectrical Technologies for Normal Motor Functions and Movement Disorders.

Young, S. R., Rose, D. C., Karnowski, T., Lim, S. H., & Patton, R. M. (2015). Optimizing deep learning hyper-parameters through an evolutionary algorithm. MLHPC '15:

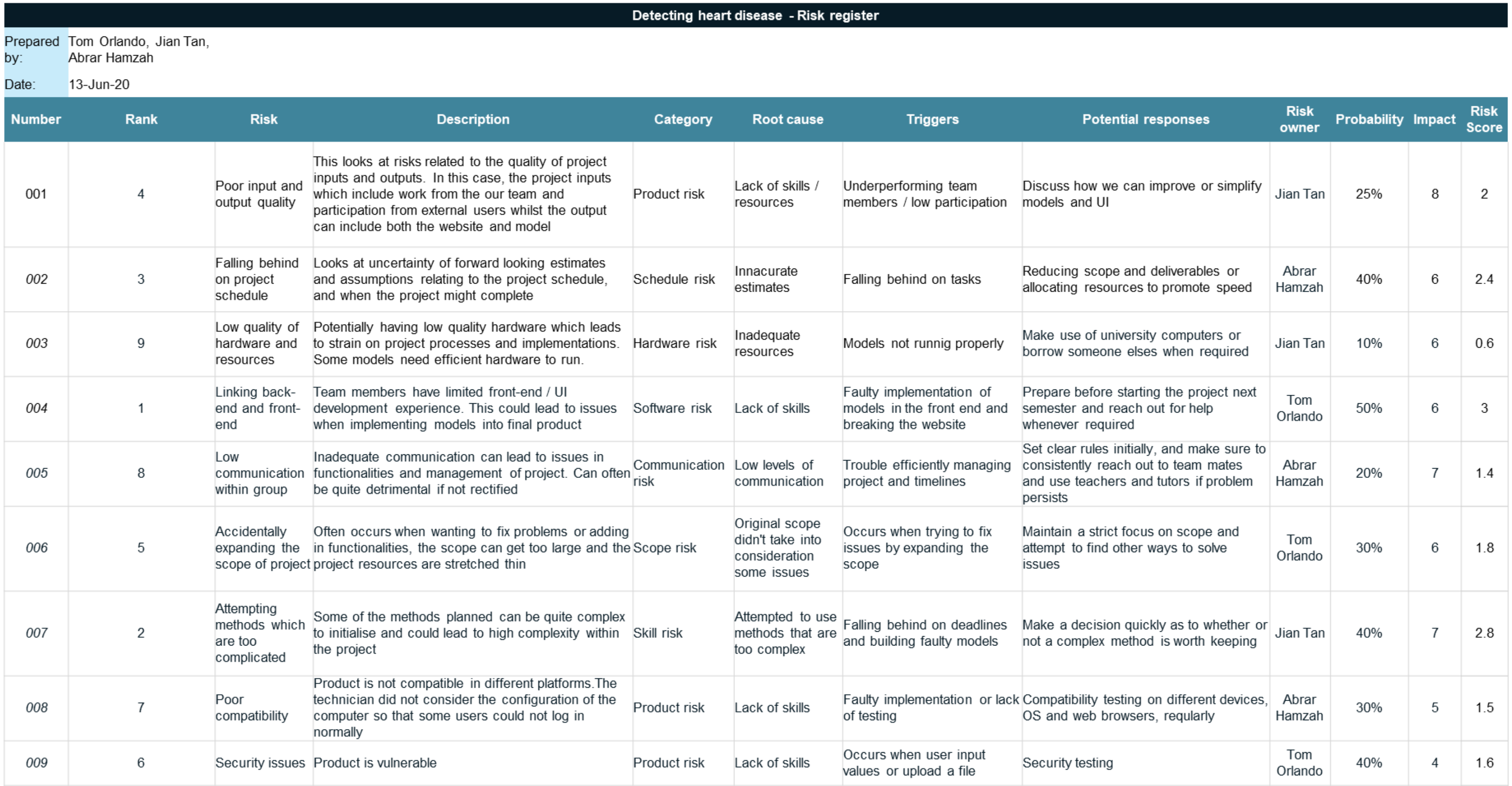
Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, 1-5.

Zhou, Q., Zhang, Z., & Wu, H. (2018). NLP at IEST 2018: BiLSTM-Attention and LSTM Attention via Soft. 189-194. doi:10.18653/v1/W18-6226

Zhou, Z. H. (2012). Ensemble Methods: Foundation and Algorithm. In Machine Learning & Pattern Recognition Series (pp. 67-77). Taylor & Francis Group, LLC.

Appendices

Appendix 1: Risk register



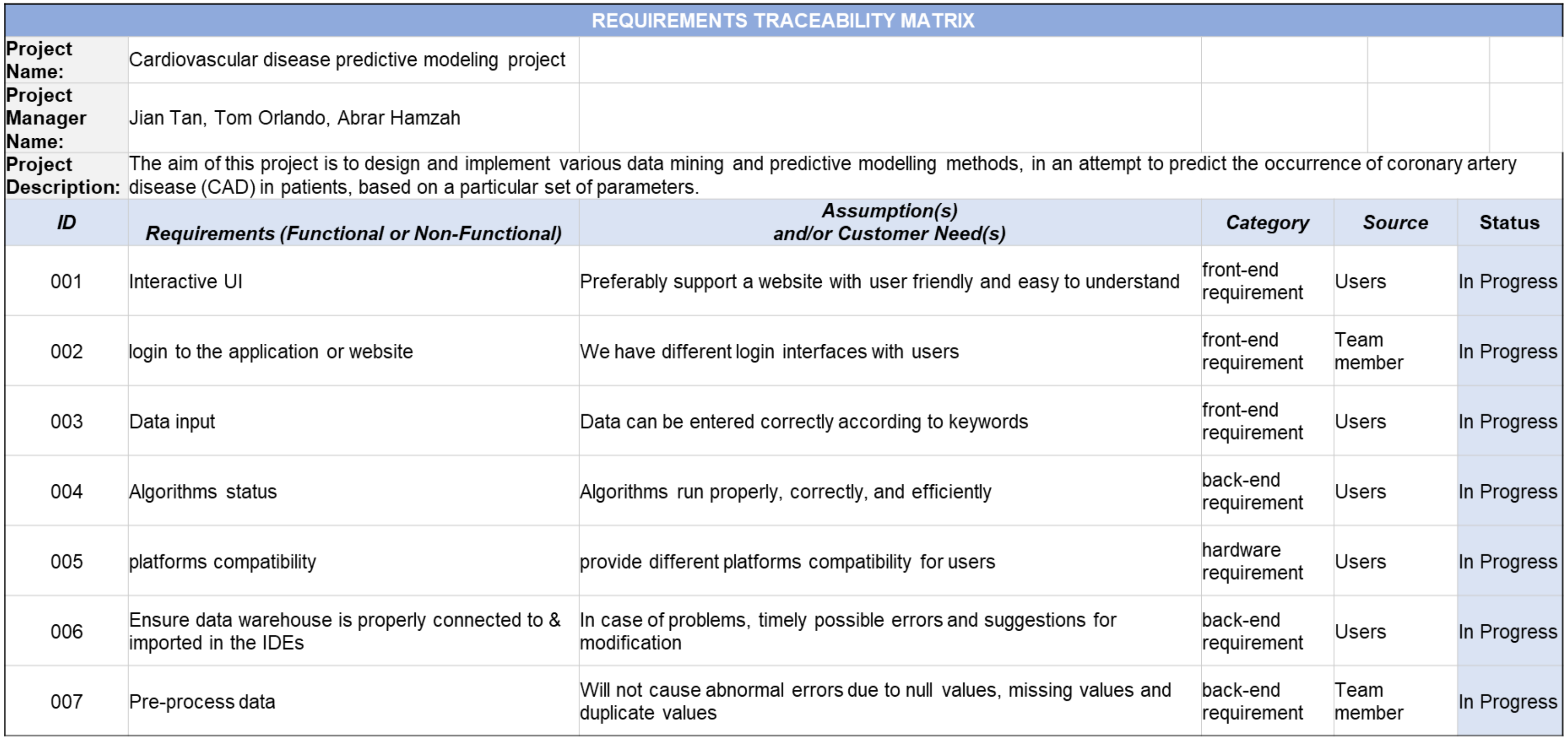
Detecting heart disease - Risk register

Prepared by: Tom Orlando, Jian Tan, Abrar Hamzah

Date: 13-Jun-20

Number	Rank	Risk	Description	Category	Root cause	Triggers	Potential responses	Risk owner	Probability	Impact	Risk Score
001	4	Poor input and output quality	This looks at risks related to the quality of project inputs and outputs. In this case, the project inputs which include work from the our team and participation from external users whilst the output can include both the website and model	Product risk	Lack of skills / resources	Underperforming team members / low participation	Discuss how we can improve or simplify models and UI	Jian Tan	25%	8	2
002	3	Falling behind on project schedule	Looks at uncertainty of forward looking estimates and assumptions relating to the project schedule, and when the project might complete	Schedule risk	Innacurate estimates	Falling behind on tasks	Reducing scope and deliverables or allocating resources to promote speed	Abrar Hamzah	40%	6	2.4
003	9	Low quality of hardware and resources	Potentially having low quality hardware which leads to strain on project processes and implementations. Some models need efficient hardware to run.	Hardware risk	Inadequate resources	Models not runnig properly	Make use of university computers or borrow someone elses when required	Jian Tan	10%	6	0.6
004	1	Linking back-end and front-end	Team members have limited front-end / UI development experience. This could lead to issues when implementing models into final product	Software risk	Lack of skills	Faulty implementation of models in the front end and breaking the website	Prepare before starting the project next semester and reach out for help whenever required	Tom Orlando	50%	6	3
005	8	Low communication within group	Inadequate communication can lead to issues in functionalities and management of project. Can often be quite detrimental if not rectified	Communication risk	Low levels of communication	Trouble efficiently managing project and timelines	Set clear rules initially, and make sure to consistently reach out to team mates and use teachers and tutors if problem persists	Abrar Hamzah	20%	7	1.4
006	5	Accidentally expanding the scope of project	Often occurs when wanting to fix problems or adding in functionalities, the scope can get too large and the project resources are stretched thin	Scope risk	Original scope didn't take into consideration some issues	Occurs when trying to fix issues by expanding the scope	Maintain a strict focus on scope and attempt to find other ways to solve issues	Tom Orlando	30%	6	1.8
007	2	Attempting methods which are too complicated	Some of the methods planned can be quite complex to initialise and could lead to high complexity within the project	Skill risk	Attempted to use methods that are too complex	Falling behind on deadlines and building faulty models	Make a decision quickly as to whether or not a complex method is worth keeping	Jian Tan	40%	7	2.8
008	7	Poor compatibility	Product is not compatible in different platforms.The technician did not consider the configuration of the computer so that some users could not log in normally	Product risk	Lack of skills	Faulty implementation or lack of testing	Compatibility testing on different devices, OS and web browsers, reqularly	Abrar Hamzah	30%	5	1.5
009	6	Security issues	Product is vulnerable	Product risk	Lack of skills	Occurs when user input values or upload a file	Security testing	Tom Orlando	40%	4	1.6

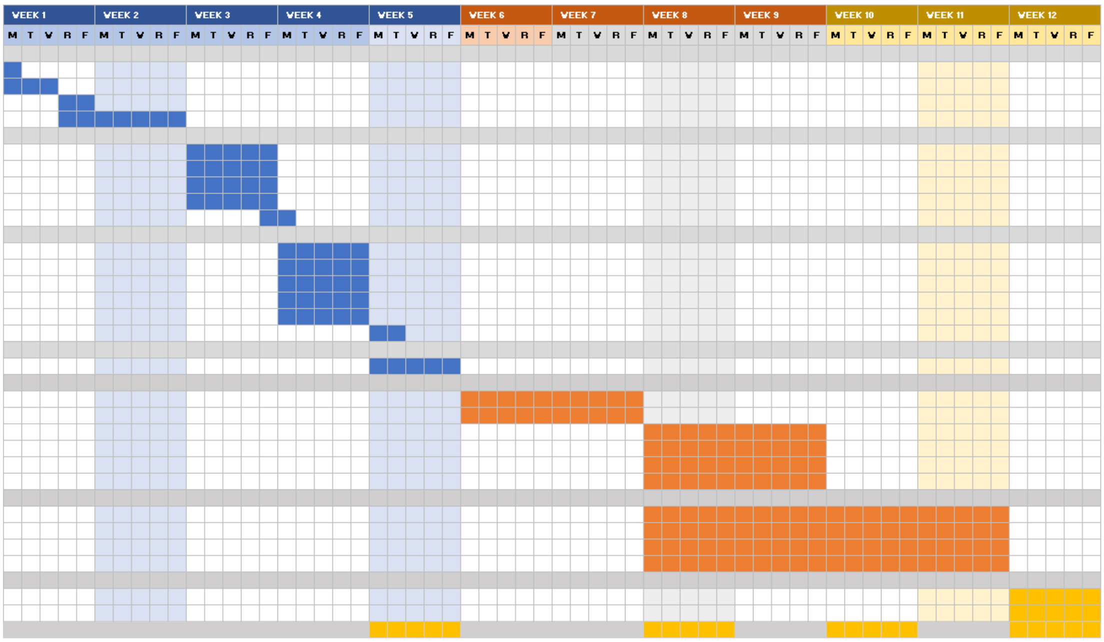
Appendix 2: Requirements traceability matrix

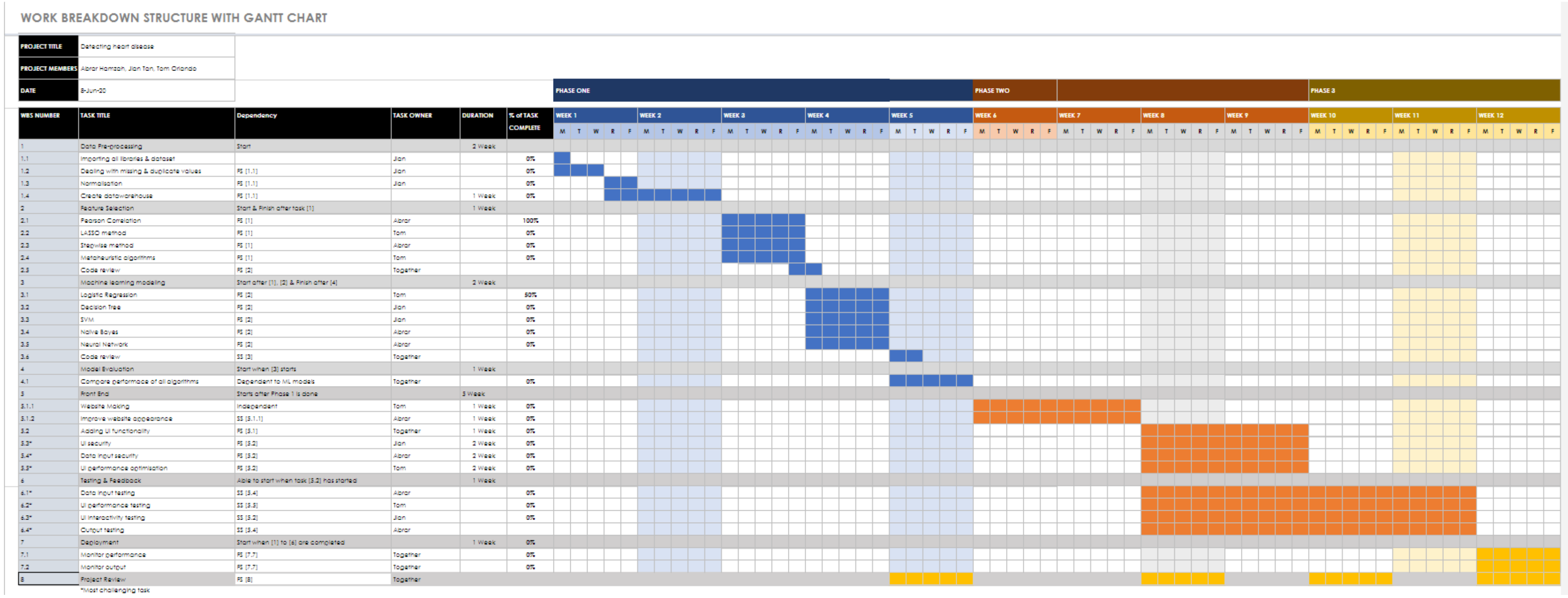


REQUIREMENTS TRACEABILITY MATRIX					
Project Name:	Cardiovascular disease predictive modeling project				
Project Manager Name:	Jian Tan, Tom Orlando, Abrar Hamzah				
Project Description:	The aim of this project is to design and implement various data mining and predictive modelling methods, in an attempt to predict the occurrence of coronary artery disease (CAD) in patients, based on a particular set of parameters.				
ID	***Requirements (Functional or Non-Functional)***	***Assumption(s) and/or Customer Need(s)***	***Category***	***Source***	**Status**
001	Interactive UI	Preferably support a website with user friendly and easy to understand	front-end requirement	Users	In Progress
002	login to the application or website	We have different login interfaces with users	front-end requirement	Team member	In Progress
003	Data input	Data can be entered correctly according to keywords	front-end requirement	Users	In Progress
004	Algorithms status	Algorithms run properly, correctly, and efficiently	back-end requirement	Users	In Progress
005	platforms compatibility	provide different platforms compatibility for users	hardware requirement	Users	In Progress
006	Ensure data warehouse is properly connected to & imported in the IDEs	In case of problems, timely possible errors and suggestions for modification	back-end requirement	Users	In Progress
007	Pre-process data	Will not cause abnormal errors due to null values, missing values and duplicate values	back-end requirement	Team member	In Progress

Appendix 3: Work breakdown structure

VBS NUMBER	TASK TITLE	Dependency	TASK OVNER	DURATION	% of TASK COMPLETE
1	Data Pre-processing	Start		2 Week	
1.1	Importing all libraries & dataset		Jian		0%
1.2	Dealing with missing & duplicate values	FS [1.1]	Jian		0%
1.3	Normalisation	FS [1.1]	Jian		0%
1.4	Create datawarehouse	FS [1.1]		1 Week	0%
2	Feature Selection	Start & Finish after task [1]		1 Week	
2.1	Pearson Correlation	FS [1]	Abrar		100%
2.2	LASSO method	FS [1]	Tom		0%
2.3	Stepwise method	FS [1]	Abrar		0%
2.4	Metaheuristic algorithms	FS [1]	Tom		0%
2.5	Code review	FS [2]	Together		
3	Machine learning modeling	Start after [1], [2] & Finish after [4]		2 Week	
3.1	Logistic Regression	FS [2]	Tom		50%
3.2	Decision Tree	FS [2]	Jian		0%
3.3	SVM	FS [2]	Jian		0%
3.4	Naïve Bayes	FS [2]	Abrar		0%
3.5	Neural Network	FS [2]	Abrar		0%
3.6	Code review	SS [3]	Together		
4	Model Evaluation	Start when [3] starts		1 Week	
4.1	Compare performace of all algorithms	Dependent to ML models	Together		0%
5	Front End	Starts after Phase 1 is done		5 Week	
5.1.1	Website Making	Independent	Tom	1 Week	0%
5.1.2	Improve website appearance	SS [5.1.1]	Abrar	1 Week	0%
5.2	Adding UI functionality	FS [5.1]	Together	1 Week	0%
5.3*	UI security	FS [5.2]	Jian	2 Week	0%
5.4*	Data input security	FS [5.2]	Abrar	2 Week	0%
5.5*	UI performance optimisation	FS [5.2]	Tom	2 Week	0%
6	Testing & Feedback	Able to start when task [5.2] has started		1 Week	
6.1*	Data input testing	SS [5.4]	Abrar		0%
6.2*	UI performance testing	SS [5.5]	Tom		0%
6.3*	UI interactivity testing	SS [5.2]	Jian		0%
6.4*	Output testing	SS [5.4]	Abrar		
7	Deployment	Start when [1] to [6] are completed		1 Week	0%
7.1	Monitor performance	FS [7.7]	Together		0%
7.2	Monitor output	FS [7.7]	Together		0%
8	Project Review	FS [8]	Together		





PHASE ONE
PHASE TWO
PHASE 3
WEEK 1
WEEK 2
WEEK 3
WEEK 4
WEEK 5
WEEK 6
WEEK 7
WEEK 8
WEEK 9
WEEK 10
WEEK 11
WEEK 12
M
T
W
R
F

WORK BREAKDOWN STRUCTURE WITH GANTT CHART

PROJECT TITLE	Detecting heart disease
PROJECT MEMBERS	Abrar Hamzah, Jian Tan, Tom Orlando
DATE	8-Jun-20

WBS NUMBER	TASK TITLE	Dependency	TASK OWNER	DURATION	% of TASK COMPLETE
1	Data Pre-processing	Start		2 Week	
1.1	Importing all libraries & dataset		Jian		0%
1.2	Dealing with missing & duplicate values	FS [1.1]	Jian		0%
1.3	Normalisation	FS [1.1]	Jian		0%
1.4	Create datawarehouse	FS [1.1]		1 Week	0%
2	Feature Selection	Start & Finish after task [1]		1 Week	
2.1	Pearson Correlation	FS [1]	Abrar		100%
2.2	LASSO method	FS [1]	Tom		0%
2.3	Stepwise method	FS [1]	Abrar		0%
2.4	Metaheuristic algorithms	FS [1]	Tom		0%
2.5	Code review	FS [2]	Together		
3	Machine learning modeling	Start after [1], [2] & Finish after [4]		2 Week	
3.1	Logistic Regression	FS [2]	Tom		50%
3.2	Decision Tree	FS [2]	Jian		0%
3.3	SVM	FS [2]	Jian		0%
3.4	Naive Bayes	FS [2]	Abrar		0%
3.5	Neural Network	FS [2]	Abrar		0%
3.6	Code review	SS [3]	Together		
4	Model Evaluation	Start when [3] starts		1 Week	
4.1	Compare performace of all algorithms	Dependent to ML models	Together		0%
5	Front End	Starts after Phase 1 is done		5 Week	
5.1.1	Website Making	Independent	Tom	1 Week	0%
5.1.2	Improve website appearance	SS [5.1.1]	Abrar	1 Week	0%
5.2	Adding UI functionality	FS [5.1]	Together	1 Week	0%
5.3*	UI security	FS [5.2]	Jian	2 Week	0%
5.4*	Data input security	FS [5.2]	Abrar	2 Week	0%
5.5*	UI performance optimisation	FS [5.2]	Tom	2 Week	0%
6	Testing & Feedback	Able to start when task [5.2] has started		1 Week	
6.1*	Data input testing	SS [5.4]	Abrar		0%
6.2*	UI performance testing	SS [5.5]	Tom		0%
6.3*	UI interactivity testing	SS [5.2]	Jian		0%
6.4*	Output testing	SS [5.4]	Abrar		
7	Deployment	Start when [1] to [6] are completed		1 Week	0%
7.1	Monitor performance	FS [7.7]	Together		0%
7.2	Monitor output	FS [7.7]	Together		0%
8	Project Review	FS [8]	Together		

*Most challenging task

Appendix 4: Team members’ contribution

Project background	
Abrar	100%
Project management	
Jian	100%
Methodology	
Abrar	100%
Project outcome	
Tom	100%
Critical analysis	
Tom	100%
Intro / conclusion	
Jian	100%