Data Mining of Cancer Data & Decision Support

UNIVERSITY OF Hull

LAMPADA FERENS

Paul Carter | Submitted for the BSc in Computer Science | May 5, 2016

Acknowledgements

I would like to thank my supervisor, Dr Chandra Kambhampati for his continued support during the course of this project, as well as his greatly appreciated attentiveness throughout the year.

Table of Contents

Abstract

The project explored the use of data mining to provide effective decision support to patients. The primary objectives of the project are to firstly use data mining as a tool for extracting useful information from legacy data from patients with expected and suspected bowel cancer. Patterns and trends from the data can be placed into rules, and these rules are then used inside a decision support tool. The decision support tool will aid clinicians in their decision to correctly classify patient risk, based on the symptoms selected in the user interface.

A methodical approach was taken to ensure the legacy data was fully cleaned and prepared for generating decision rules. The rules were then placed inside the decision support tool programmed to match selections to any imported rule. This report will go through the process and provide discussion on the various techniques used to create an effective system, as well as providing background material and technical analysis.

Data Mining of Cancer Data & Decision Support

Chapter 1. Introduction

This document will analyse the step-by-step process of development, as well as providing a detailed specification and discussion of the background material and a review of the management. The report will also review the technical development of the project, discussing the key stages and providing analysis with regards to the decision tool implementation and any testing of the application. A critical evaluation will be performed which will examine how accurately the system meets the objectives drawn up at the beginning of the year. Improvements towards the system will also be noted and explained.

The original specification had a primary focus towards the data mining element of the project and not the decision interface. However, after discussion with the project supervisor, the project will now focus on the decision interface element, with less of a focus on the data mining; however, both elements will be incorporated into the project. The project title has been changed to accommodate the different nature of the project deliverables (previously called 'Data Mining over Cancer Data'). The original project brief will be altered, discussed within section 1.1 to reflect the changes, as well as the reasons why they were altered.

1.1 Design Brief

The original project brief has been reworded to reflect what the project will actually be focusing on. The final specification reads as followed:

Decision support tools allow for further analysis and decisions based upon mined data and associated rules. Data mining allows general patterns to be retrieved from data sources such as data warehouses and databases. These patterns can then be used as an aid to making decisions by supplying decision rules. This project will make use of multiple Excel cancer databases supplied by clinicians at Castle Hill. Weka will be used with this data to generate decision rules for patient risk that can be mapped back onto the Excel data. These rules will be used to determine the risk to any patient. The decision tool will enable multiple selections to be made based upon patient symptomology. The interface will also be built using Visual Basic, that allows the user to select a magnitude of symptoms and view the risk assigned to any selected patient. A backend database will also be implemented to view previous patient selections and decisions. While it is expected that weka is used to do the data mining, the final decision process will be implemented in Visual Basic, using Excel as the supporting data warehouse.

The original specification was supplied in *Appendix C*, however further changes have been made since and revised in the full final project brief in *Appendix D*. The general change to the overall project is the main focus towards the development of the clinical decision support tool. The specific changes that have been made against the initial specification include the addition of a backend database and additional functionality for better rule allocation and user understanding. The project will also be programmed in Visual Basic due to its simplicity and ability to create 'easy to use' and clear interfaces.

1.2 Context

The decision support tool will be working within the medical domain, and more sensitively, with cancer data. Deliberation must be made about how the application will work within the domain, and whether these systems affect original clinician practices.

Clinical Decision Support Systems (CDSS) must provide affective medical aid to patients, as well as being intuitive with clinicians on the basis of any given symptoms (Sittig et al., 2007). Any decision rule must be able to assess the patient effectively in order to construct a reasonable allocation of risk. With this in mind, it's important to then deliver the best possible judgment, coinciding with the help of the clinician. It's apparent that for more than a decade, the need for technological clinical support has grown as there is a definite improvement of patient diagnosis (Berner,2009). It's also important to remember that although these systems are becoming far more intelligent, they are not there to replace conventional clinical practice. This improvement in patient diagnosis was made in a study by (Garg et al., 2005), which concluded that out of 97 clinical studies using a clinician support tool, 62 had improved practitioner performance, giving a percentile of 64%. In fact, out of all the various support tools available to clinicians within the study, they all had improved practitioner performance. Similarly, to this project, the study also concluded that diagnosis for acute bowel obstruction was 16x quicker than alternative means of diagnosing patients.

Taking this into account, the project will try to replicate the technological support for clinicians with regards to diagnosing patient risk. The project will also consider technical naivety, thus ensuring a user friendly applications to reduce the time to use the system, so not to impede the heavy workflow of the clinician.

1.3 Ethical Issues

The project uses real data from cancer patients, which is supplied by clinicians at Castle Hill Hospital in Cottingham, East Yorkshire. There are therefore issues that arise with regards to patient confidentiality. However, all data supplied is completely anonymous so no identifiable information is given, i.e. Names and Addresses. There are also ethical issues surrounding the system itself, whereby the accuracy of the predictions may not be satisfactory for the clinician. It is also possible that GP's may use the decision support tool more than they should, as they may believe the software is more accurate than their own diagnosis (Berner, 2009). Consequently, this reliance may then lead to misdiagnosis and then conclusively leading to a loss of life.

1.4 Aims and Objectives

The aim of this project is to take cancer patient details on an excel spreadsheet and make decisions and rules via the data mining methodology mentioned in section 2.3. The decisions and applicable rules will be visually represented in a decision interface, which allow the user to make decisions based on the mined data. The data supplied has attributes relating to patient symptoms and whether they have cancer. The data will have to be cleaned to be able to be mined effectively. The project will make use of weka software to provide a detailed analysis of the excel data. The detailed analysis provides the decision rules which can be used to discover if a patient has a high or low risk of cancer. A visual representation of the data will be stored in a user friendly, graphical user interface. The associated rules will be used and placed inside the user interface to provide a risk assessment after the user has selected the symptoms. The interface will also make use of a backend database which will store the selections made by the clinician. An objective conclusion will then be made based on recommendations on how to improve the way mined data is used and visually represented.

Project Deliverables:

1. To independently understand the importance of using data mining to provide a detailed analysis of clinical information.
2. To use Weka as the main tool for providing machine learning decisions based on the data supplied by Castle Hill.
3. To provide a graphical user interface in order for users to understand and analyse mined clinical data, linked with a backend database to store user selections.
4. To gain a deeper understanding of data mining techniques through practice and external research.
5. To objectively conclude and make recommendations on how to improve the way mined clinical data is used and visually represented.

Overall, the deliverables culminate in a decision support tool which will aid the clinician in their diagnosis of a patient. The decision support tool will also aid the reduction in time it will take to make that diagnosis.

1.5 Risk Analysis

Developing a clinical decision support tool creates a magnitude of risks when predicting whether a patient has a high or low risk of cancer. Typically, using software that uses inconsistent data, or a system which provides any kind of prediction, problems can happen, however it's gravely important to pacify those risks before other risks appear. The application that was designed will give the user the opportunity to enter in the symptoms of a patient and the system will calculate a prediction if a patient has a particular risk based upon the input in the decision support tool. It's therefore important that the clinician knows the system should not be used for a primary prediction, but only to be used to support *their* own prediction.

There are risks with this method of diagnosis and for the project. The first stage is finding particular patterns in the database, so it's important that cleaning the data is done at a comprehensive level, this will mean that there are no discernible errors which could perhaps contort or create a bias in the results, which would conclusively undermine a ruleset that is used to support a diagnosis. When the data is fully cleaned and no errors in the data exist, it can be put through predictive software, at this point it's important to be sure that the representation of false negatives are at a minimal. A False negative in this case is when low risk prediction is given to patients who have a high risk of cancer. If there is a risk of cancer, it's also possible the disease might become asymptomatic, this means that system could allocate low risk prediction to the patient, which is also a serious problem. A considerable amount of research must be allocated within this project in order to understand how the number of false negatives can be decreased, as the detrimental effect could be that a poor system may become dangerous to the health of patients. *Appendix A* illustrates the risks associate with this project under the following headings:

- **Risk** – The risk identifies what potential risks pose a threat to the development of the project.
- **Severity** – Severity is measured in High, Medium & Low and indicates a level of threat to the project development, if any should occur.
- **Likelihood** - The probability any given risk will occur during the development of the project, again measured in High, Medium or Low.
- **Mitigation** – How any risk can be either alleviated or made sure it will never happen during the development of the project.
- **Residual Impact** - If any risk does happen during the project and the risk has been mitigated against, residual impact assesses the leftover risk, measured in Low, Medium & High.

1.6 Report Structure

The remainder of the report is structured in the three following parts. The second chapter provides the background material and further research into data mining techniques and the fundamental criteria of this project. The third chapter discusses the technical development behind the construction of the clinical decision support system and how it was accomplished, accompanied with the data mining processes. The final chapter (Chapter Four) is a critical evaluation with reflection on how and what has been produced. The chapter will also evaluate how accurately the completed system matches the project objectives that were set out.

Chapter 2. Project Background

This chapter will discuss the technical aspects of the project that may be unfamiliar to many. The project includes medical background information, which includes many non-IT based topics. Section 2.1 will discuss the medical terminology used which aided the overall development of the project. Section 2.2 will discuss unfamiliar technical terminology used within the project.

2.1 Problem Context

As the decision tool is working within a medical domain, a lot of the preparation with regards to the project was understanding the technical medical terminology that was applied to the data set. It was also important to work out what conditions were more serious than others, in order to produce a better and more efficient DSS. For example, the dataset contained numerous abbreviations such as 'DD' which later was researched as meaning 'Diverticular Disease', which is a condition in the colon which causes disturbances in the bowel (Sheth et al., 2008). A lot of diseases which were unfamiliar at the beginning were also researched to analyse the various symptoms to see if there were patterns between the research and the dataset itself. Another thing that had to be researched was the average recordings of hemoglobin levels with each disease, to see if there was any correlation between them. The majority of the non-computational work was around researching medical terminology, medical statistics and symptomology.

2.1.1 Referral

The main idea behind the use of a decision support tool and the supporting data mining techniques, is to foremost minimize the number of incorrectly classified patients. Therefore, the decision support tool will allow for the reduction of falsely classified patients, or those who are thought to be low risk but are actually at a high risk of developing a condition. If a patient is of high risk, a referral can be made to a consultant which will look at their condition further.

2.1.2 Risk Outcomes

The data that that was supplied contained an array of medical conditions that the patients had under the attribute ‘Diagnosis’. Patients that had high risk conditions which were considered cancerous were generally given a ‘yes’ value to whether they had cancer. Those with conditions which were determined lower risk were subsequently given a ‘no’ value. The list below shows the high risk conditions with cancer diagnosis.

- Adenocarcinoma
- Carcinoid
- SCC (Squamous Cell Carcinoma)
- Various types of Polyp (e.g. Adenoma, Hyperplastic, Tubular Adenoma & Tubulovillous Adenoma)
- Sarcoma

The decision support tool does not indicate a specific condition; however, it does allocate a low or high risk to a patient. The conditions above can be grouped together as ‘cancer’, with less severe conditions also being grouped together. The reason why this has been done in this project and why it complements the decision process will be discussed further in section 3.3.7.1.

2.2 Technical Background

This section will explain the various technical topics relating to the project. Each topic will contain a description behind the technical process of the development of the decision support system.

2.2.1 Data Mining

Weka is a machine learning software with in-built algorithms for predictive modelling and data analysis (Jagtap, S. & Dr. Kodge B. G, 2013). This will be used within this project as it serves the fundamental component of the data mining by providing various tasks such as preprocessing, visualisation, regression and classification. These tools check a particular set of data that is fed into the software by detecting consistent patterns or relationships. The newly-found patterns are then visualised back to the user to check the relationship of the variables. There are many strengths to using this piece of software, as it allows a quick and effective way of providing knowledge of a dataset, which may not have been obvious beforehand. Depending on what classification (algorithm) is used, the knowledge supplied by Weka is visualised back by supplying the user a set of rules. These rules are identified via the algorithm by analysing consistent patterns in the data, these patterns are then mapped back, allowing the user to see the certain trends that appear in the dataset.

The raw data that was supplied must be cleaned before the data can be mined. There are various issues which could cause problems if the data is not cleansed; there can be a magnitude of errors which is called ‘noise’. Noise could involve data with missing values, or value types which are not consistent with the domain. Much of the data noise can be cleaned by removing the erroneous values, or replacing them with the mean of the remaining data values. Section 2.3 will outline the workflow that will be undertaken with regards to the data mining.

2.2.2 Knowledge Discovery

Knowledge discovery is the process of extracting useful information from databases, which otherwise, would not hold much use to anyone. The need for this extraction is growing rapidly as data becomes more digital, allowing for the new-found knowledge to help decision processes (Fayyad et al., 1997). This is an important part of the data mining methodology, as it's the way in which rules can be generated, and in particular, helping with clinical decision making with regards to this project. *Figure1* below illustrates the sequence in how to find knowledge in a database.

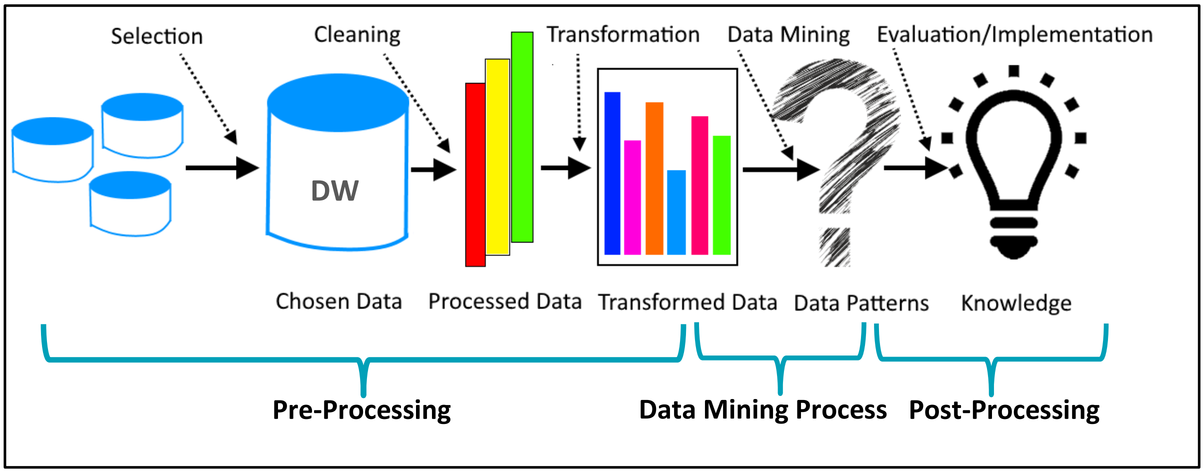


Figure1. Sequential view of the knowledge discovery timeline, showing the data mining processes.

2.2.3 Classification

Data mining classifiers use algorithms to classify data based on which attribute is stated as the target class (Giberta et al., 2010). Classifiers are used to learn various patterns in the dataset, allowing for knowledge discovery. The classifiers can produce decision rules based on those patterns, which can be used to find out useful information. An example of this, and what will be used in this project, are decision trees. Decision trees are used primarily for their easy-to-read modelling capabilities, as they produce a top-down tree, showing the process towards a particular rule. Although decision trees are easy to interpret, they do require the data to be fully impartial to exclude a bias towards a particular outcome. This is called class imbalance, and happens when an outcome is more prevalent than another. An explanation of the resolution to this problem will be discussed in section 2.2.4.

Once the classifier has been used, and the rules have been generated, issues can arise whereby rules can conflict with one another. A brief example would be if two classifiers were used which generated the same attribute values, however resulting in a different outcome. A full explanation will be discussed in section 2.2.5.

2.2.4 Class Imbalance

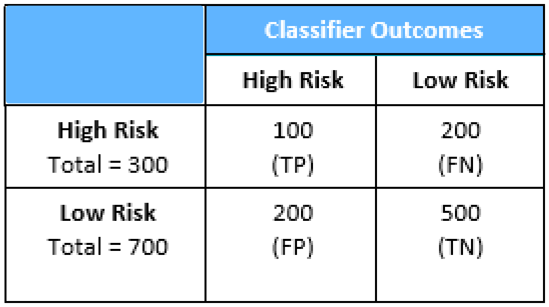
Often with medical datasets, class imbalance can and will occur as it's typical to see a far higher number of low risk patients than high risk (Li et al., 2010). Class Imbalance is typically when a particular outcome is far more common than the other. For example, the given project database had around 650 patients classed as not having cancer, whereas around 200 did, therefore there is a large disparity between the two different outcomes. This can have a negative effect by decreasing the generalization as there is a bias towards the majority outcome (Rahman & Davis, 2013). This poses as a threat with regards to clinical datasets as it's possible that classifiers will miss-classify high risk patients as being low risk due to imbalance of the outcomes. It's therefore useful to balance the classes to ensure there is no bias in the classifier when extracting rules.

2.2.4.1 Sampling

There are ways in which the problems with class imbalance can be alleviated to improve the accuracy of predicting the number of true positives e.g. the number of correctly classified high risk patients. Sampling is a general way of balancing the outcomes in a dataset by either removing or duplicating records of a particular outcome to match those of another outcome. Under-sampling is the process of reducing the majority outcome records to match that of the minority outcome. Whereas, over-sampling is the process of adding records to the minority class to match the majority. Random sampling can also be done, whereby a selection of random records in the minority class can be duplicated to balance outcomes. Contrastingly, random under-sampling involves deleting random records of the majority outcome. The practicality behind this will be discussed in chapter 3.2.5.

2.2.5 Classifier Performance

Once the classifier has been selected and has run through the data, the accuracy has to be checked before deploying the rule set and applying it to the DST. A confusion matrix defines the distribution of the data records to the different possible outcomes. The confusion matrix is split up into four categories which are: True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). With regards to a medical domain and the outcomes typically being High or Low Risk. True Positives define a patient that is correctly classified as having a high risk of cancer. Comparatively, True Negatives are applied to patients which are correctly classified as having a low risk of cancer. False Positives define low risk patients incorrectly classified as high risk, with false negatives defining high risk patients as having a low risk of cancer. *Figure2* illustrates a typical confusion matrix with example data.



	Classifier Outcomes	
	High Risk	Low Risk
High Risk Total = 300	100 (TP)	200 (FN)
Low Risk Total = 700	200 (FP)	500 (TN)

Figure2. An example of a confusion matrix with 1000 total data records

The example confusion matrix illustrates a certain scenario when it comes to working out the accuracy of the chosen classifier. The number of correctly classified high risk patients works out at (100/300) or 33.3% recurring, whereas the number of correctly classified low risk patients equates to 500/700 or 71.4%. It's also possible to work out the PPV (Positive Predictive Value) by the equation (TP/TP+FP) and the NPV (Negative Predictive Value) can be calculated by the equation (TN/TB+FN). This project will focus primarily on increasing the number of correctly classified high risk patients.

2.2.6 Data Warehouse

Data warehousing is one of the most essential elements of decision support. A data warehouse's purpose is a collection of data that is used in organizational decision making (Chaudhuri, S. & Dayal, U, 1997). Much of the preparation of the data e.g. cleaning, is done from within the original database for analytical processing, a pre-requisite before the data is put through a classifier. The project requires a data warehouse, as it's the fundamental part of the data management and pre-processing.

2.2.7 Decision Support Systems

Decision Support Systems (DSS) are information systems which provide support for decision making (Tripathi, 2001). There are various types of DSS ranging from executive information systems, online analytical processing and business intelligence systems (Arnott et al., 2004). They all offer the same fundamental principle, which is to provide decision support. Therefore, a clinical decision support system is just another type of decision support. The systems typically use rules from previous data mining processes, using the extraction from the mining of data to create decisions purely from the raw data itself. Extracting patterns from old data can then be transposed and used upon newer data. A DSS is typically an interactive computer based system which helps individuals, businesses or organisations to solve unstructured, sometimes complex, problems.

A common use for DSS is found in Tesco's clubcard loyalty system. The system monitors the purchases of products by customers and it will try and predict what the customer may want in the future, based on previous purchases, providing offers to entice the customer back into the store. With regards to clinical decision support systems, the goal is to provide a system which improves clinical diagnosis, by working side by side with clinicians. Any clinical decision support system must address issues before it can be integrated into the medical domain. Sittig et al. identified 10 grand challenges in order of importance, to be solved if the benefits of these systems were to be felt at their full potential. The provided challenges are critical if a clinical decision support system is to improve the quality, safety and efficiency of healthcare.

2.2.8 Conflict Resolution

It is likely that when developing a decision support system, several rules will contradict each other and provide a different outcome. Various methods exist when it comes to resolving conflicts. There is a simple method called specificity, which is to use the rule which provides more conditions e.g. a rule which uses more attributes to find an outcome. Another is using weighting, whereby the probability, improvement and confidence is measured in the rule. Conflict resolution is important to provide better accuracy in the rules, therefore providing a better decision support system (Bramer, 2007). The project will not be using conflict resolution as only one classifier will be used, therefore it's rare to find contradictions in rules which are from the same classifier.

Rule1: RectalBleeding=Yes & Constipation=Yes → Risk=Low
Rule2: RectalBleeding=Yes & Diarrhea=Yes → Risk=High

Figure3. An example of two rule conflicts with disaffirming outcomes

Note the two rules. The two rules belong in the same conflict set as the condition 'RectalBleeding' is the same in both rules. This particular example is called a disaffirming conflict, as the outcomes are different to one another.

2.2.8.1 Rule Improvement

Rule improvement is used to work out how strong a particular rule is, against comparing the same outcome with random chance. The support for the frequency of the conditions in the rule are calculated first. The confidence is then measured by dividing the support of the conditions and the outcome together by the support of the conditions.

Improvement can then be calculated by this equation P(items together)/ΠP(item). Using an example, if the condition 'Constipation' appeared 8/10 and 'Constipation' → High Risk' appeared 6/10, using the formula, this would equate to 0.8/(0.6*0.8) = 1.66. Therefore, the higher the improvement value, the greater the chance of those two attributes correlating towards either a high or low risk. Rules are then picked based on the frequency of a particular pattern being found. Any improvement value above 1, generally means a greater chance than just random.

2.3 Issues for Medical Data Mining

The challenges faced with medical data are unique considering the sheer complexities that exist in many medical databases. These include the privacy, sensitivity, data heterogeneity, ethical, security and legality (Ashwinkumar & Anandakumar, 2010). Researchers, doctors and data analysts have been using medical datasets for research for many years. Using these datasets have played a positive role in medical progress by acquiring new knowledge about particular conditions, and how those conditions directly affect people. It's therefore gravely important to make sure that the patient data being used has been authorized by the patient with their consent. Depending on the anonymity of the data, patients should ideally understand exactly what their data is being used for and who will be accessing the data records (Veen, 2008). Currently, the construction of reliable predictive models with regards to medicine require the integration of very complex data, which includes clinical, laboratory, genetic, genomic and proteomic data, running concurrently in a clinical decision support tool. At the moment however, these systems are not yet available as integrating this data, and making it work concurrently, is still an issue to be resolved (Bellazzi & Zupan, 2008).

2.4 Processes and Methodology

Data Mining can be a complex and can require a methodical process in order to complete a specific data mining task, especially if appropriate results are to be achieved. As data mining is a relatively new industry, methods in how adequate data mining results can be achieved vary from business to business, adopting different methodologies, so there is no universal approach. Data mining methodologies normally list the important steps from data analysis to the final step of deployment. The specific tasks that the methodology enlists is the acquirement of legacy data, i.e. data provided by a separate medium, which can then be used to discover unknown pattern and trends in the data to find meaningful information. The results can then be deployed into a decision support tool or a report as examples.

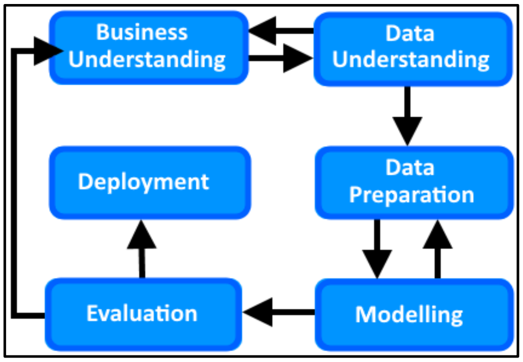


Figure4. CRISP-DM Model illustrating the process of data mining

CRISP-DM is a non-propriety data mining methodology and is one of the widely used methodologies for its preliminary understanding stages (Chapman et al., 2000). Another data mining methodology is SEMMA (Sample, Explore, Modify, Model & Assess) developed by SAS. SEMMA was created to be used as a universal model for data mining projects (Shafique et al., 2014). This project will use CRISP-DM for its simplicity and methodical approach in guiding through the specific tasks. The task list for this project can be viewed in *Appendix K.*

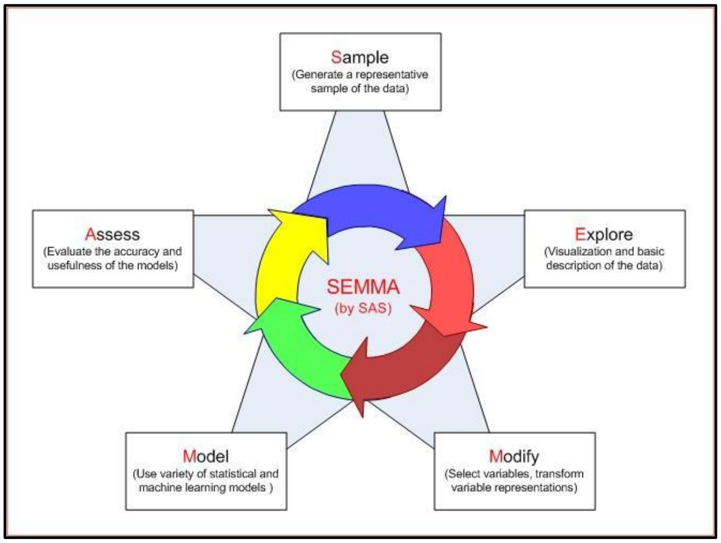

Sample
(Generate a representative sample of the data)
Assess
(Evaluate the accuracy and usefulness of the models)
SEMMA
(by SAS)
Explore
(Visualization and basic description of the data)
Model
(Use variety of statistical and machine learning models)
Modify
(Select variables, transform variable representations)

Figure5. Example SEMMA model showing an alternative process methodology (SIS, Processes in Data Mining)

Chapter 3. Technical Development

This chapter will explain the stages of the system's development, including the data handling, data mining and the software design principles. Every topic will be discussed in depth, with analysis covering each stage of the development. The system requirements will also be outlined.

3.1 Narrative Description

The project outlined many things that weren't learnt during the first two years of the degree program, which originally resulted in some worry as new techniques for the project had be undertaken with a large amount of self-learning. Therefore, each key feature that needed to be implemented from the specification, had to be learnt from accessing numerous sources of material, and of course learning new software in order to fulfil a particular set of goals. Fortunately, Data Mining & Decision Systems is a module that was undertaken in the third year which outlined the basics of data warehousing and the theory behind machine learning. This module helped greatly in building an understanding of how the different concepts work, and in particular, how it can be done at a practical level. This new knowledge therefore will enable me to adopt a platform to work from, and to build up a particular set of skills that will ensure a successfully completed project.

3.2 System Requirements

The development of the system required the use of several software packages. Software was used to carry out the data mining, the data manipulation and the software development phase. Database management software also used to store details of user selections.

3.2.1 Excel

The first stage of the project is the preliminary sorting of data before it gets put through Weka. This is the part that requires the most attention, as looking at the dataset can offer information about the type of classifier that could be used. Many tasks and issues are presented when a given dataset is provided. The dataset will most likely contain errors, of which none can be present if there is a requisite for rules as they may make any rules void. The errors which came with the dataset included attributes which contained values which offered little value to the project and the eventual outcome, so they were removed appropriately. Other errors existed such as unreadable values, these are values which are hard to understand, and therefore needed to be removed as well. It's not just errors which

need to be examined, it's also possible to do manual checks on the data. For example, looking at particular patterns manually which can't be done through Weka. For example, the standard deviation of high risk patients and low risk patients can be checked to see if there is a trend.
Excel allows for files to be stored as .csv (Comma Separated Values) files, which is the same file format in with Weka uses. Once all the errors are removed and the dataset is 'cleaned', as well doing manual checks, it's then acceptable enough to put through Weka to create rules

3.2.2 Weka

Weka will be used to for the data mining in the project. Weka software contains a number of algorithms and other useful tools. The software is easy to use, easily navigable and will be used to create decision rules that will be mapped back into Excel. The choice to use Weka in this project primarily stems from its ability to create graphical models of decision output. Using this software would be beneficial as is makes it easier to see how each rule was constructed. The main downfall which could occur with this technology, is if you had a very large dataset with a small amount of variability. Rules that will be produced from the dataset might make little practical sense, therefore creating rules which do not supply credible solutions. Another strength of using this piece of software for the project is that's free to download and typically does the job it's needed to do. Although Weka isn't an industry standard machine learning tool, it's sufficient enough for this project. It is essential then for this project to work with Weka, as a user interface will be built during the development process to allow a user to directly interpret data.

3.2.3 VisualBasic.Net

Having more experience in VB.Net, also being easy to learn, and knowing that good interfaces can be built using the language, it was decided to use VB.net for the development of the user interface. The user interface will be used to connect the user between the input from the data warehousing and the output created by Weka. This stage of the project binds all the technology together, to create an interface which uses the technology in the first half of the project to create a cancer diagnosis based on clinician input for the second half. The benefits of using VB.Net are mainly down to its sheer ease of use, from being able to insert objects that are pre-made, to the relative simplicity of the code. The drawbacks however are that if more complex extensions were made to the project, it's possible that the language will be too low-level to implement those changes, therefore Java may be a better option in that respect.

3.2.4 MySQL

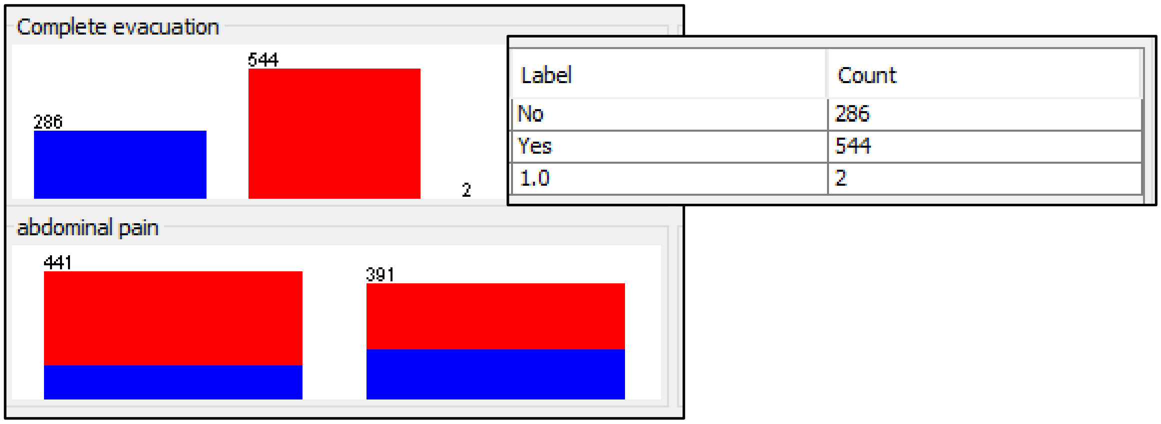
Whilst developing the project, it was realised that it would be quite appropriate if the user could see the selections made in the user interface, by storing them in a backend database. When the user selects from the different dropdown lists, any Boolean value that is picked is stored inside the database so the user can check the selections at a later time. The database can then be stored inside the user interface for easier access to the data. Some selections in the interface may not be backed up by a particular rule produced from Weka, so instead of the user interface just saying, "Sorry no rule supplied", it was therefore better to have the selections stored anyway. The selections can possibly be examined by the clinician, and then an opinion could still be made, even though the system couldn't directly come up with a diagnosis itself. It was firstly decided to use SQL, however it was then decided that the interface wouldn’t allow for direct manipulation of the data, and just to have it stored and able to be viewed, instead of changed. The benefits of using MySQL is down to its simplicity to learn and efficiency in database management (Letkowski, 2014). The software is also free to use and very powerful, enabling up to 50 million rows of data to be stored in each database (Inan et al., 2011). There are few disadvantages of using MySQL with this project, however the ones related to this project could the stability issues that it is infamously known to have; with regards to handling patient details, it could pose an eventual problem. Originally it was also decided to implement the database using access due to its familiarity, however it wouldn’t make any sense to use it as the file extension would have to be changed every time the software was being used on another computer. MySQL was also chosen based on the fact that it’s linked to a local server, and allows the user to access it on any computer using the interface.

3.3 Data Preparation

Data preparation is key to reducing noise before any data mining is able to be done. The data supplied accommodated many errors which needed to be amended before being processed through Weka to create any decision output. The original excel file contained two data worksheets called ‘JC_data1’ and ‘JC_data2’, which were combined to create a single worksheet called ‘BaseData’. BaseData contained numerous errors. These included data entry errors, many empty values and attributes which wouldn’t add any value to the decision making process. When putting the data through Weka as it is, the file will not load due to certain requisites and consistencies that Weka needs to be able to read the data. The data contained many instances of symbols such as hyphens and commas which Weka does not read at all. The data also contained both numerical and nominal values, which will need to be transformed to either one or the other when creating decision rules.

3.3.1 Data Exploration

Once the preparation of the data was completed and the data can be put through Weka, it’s now possible to explore the data that has been given, by using Weka’s various visualisation tools to find any data anomalies. This is an important step as it addresses problems that may have been missed out in the original preparation stage. *Figure6* shows what Weka’s visualisation tool looks like on the ‘BaseData’.



Label	Count
No	286
Yes	544
1.0	2

Figure6. Weka visualisation of attribute ‘Complete Evacuation’ with three distinct values

Figure6 is annotated to show the errors which can be visually checked. Complete Evacuation has 286 patients with a 'no' value with 'yes' being 544. However, there are two values which do not match the data taxonomy; it isn't clear what '1.0' could mean when it comes to whether a patient has symptoms of complete evacuation, so this error can be fixed by doing data transformation or by cleaning the values so that a classifier can check it more adequately.

Weka uses colour coordinated blue and red to show the outcomes of the classes. Complete Evacuation was chosen as the target class attribute. The attribute class 'abdominal pain' has blue and red split in each outcome. This is because it shows the distribution of the outcomes in the target class e.g. Complete Evacuation. A table is used to see the attributes and their transformations and stores all findings from the data cleaning stage. *Table1* shows an example data description table, highlighting two different attributes and their data types.

Attribute	Data Type	Value Range	Homonyms	Replacement	Missing Values	Importance
Age	Integer	20-96	None	N/A	None	Keep
Complete Evacuation	Nominal	Yes|No	1	Yes	None	Keep

Table1. Example data description table with two different attributes and data types

The 'Data Type' column shows the type of data that the attribute is. 'Value Range' shows the range of the values in the attribute. 'Homonyms' lists the errors and 'Replacement' lists the potential corrections. The full data description table for this problem can be found in *Appendix B*.

3.3.2 Cleaning Methods

The following methods were undertaken to clean the data before the data mining can be done. Various strategies exist in order to fully clean the data, and each topic will be discussed fully and illustrated appropriately.

3.3.3 Attribute Reduction

The first part of the data cleaning was to filter out the attributes which contained no useful data. There were some attributes in the dataset which contained a lack of data, and also some attributes which contained data which was not useful for the project. Due to those problems, the attributes were simply removed from the dataset as they could not be cleaned. An example of an attribute that had to be deleted was 'Frequency of Bleeding'. The attribute contained text values such as 'regularly' or 'daily', as well as containing many missing values. The attribute had no conformity and could compromise the predictive power of any classifier.

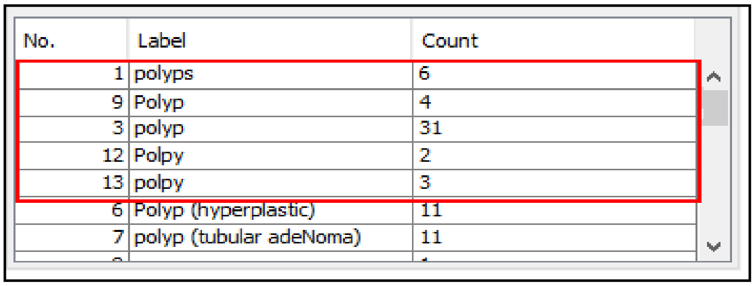
3.3.4 Data Cleaning

Removing unnecessary attributes isn't the only cleaning technique needed in order to have a fully clean and workable data set. Data can contain many human errors with regards to data entry, so the data set will need to be thoroughly checked for those abnormalities. Any errors that were found, were not deleted, but were copied over to a separate worksheet called ‘DirtyData’. The ‘DirtyData’ worksheet serves as a place where any data records with data entry errors can be stored, both for future reference and for good Data Mining practice. An example of errors that were found in the original data were invalid data entries. These entries included numerical values (13, 8.8) in the ‘Ex-Smoker’ attribute, when they should have been nominal ‘yes|no’ values. There is no way that these errors can go through the transformation process, as there’s no possible way of knowing if the patient is an ex-smoker, based on the given numerical value.

Other data entry errors included nominal values (Unknown, Normal) in the ‘Hemoglobin’ attribute where these were required to be numerical, as it measures the mathematical level of hemoglobin in the blood. There was also a data entry error in the attribute ‘CA Colon’ with a given value of ‘nn’. The attribute must be either yes or no, so ‘nn’ is not useful for prediction purposes, as it doesn’t relate to the data taxonomy. The attribute ‘Complete Evacuation’ had a transformation error, where a data record had a numerical value ‘1’ instead of nominal. This was cleaned by transforming the ‘1’ to a nominal ‘yes’ value. There were also missing values in the given data set e.g. in attributes ‘Change in Wt’, ‘Loss of Wt’, & ‘Loose Clothing’. The data records with the missing values were copied over to the ‘DirtyData’ worksheet.

3.3.5 Homonyms

Homonyms, in relation to data, are data entries that look and sound the same, but may be spelled differently, thus creating two entirely different values. There are many examples of this in the ‘BaseData’ worksheet. The data entry for the condition ‘DD’, also had a similar value of ‘D.D’, this creates two separate identities for what should be the same value. Another example includes ‘polyp’, ‘polyps’, ‘Polyp’, ‘Polpy’ & ‘polpy’. They should all be the same value, but are all registered as different because of the small differences in the spelling and casing. Weka also reads these as different values, because they are not spelled the same. This can cause problems because Weka will think these are new values, or more simply, different conditions entirely. The example below shows the homonyms in Weka.



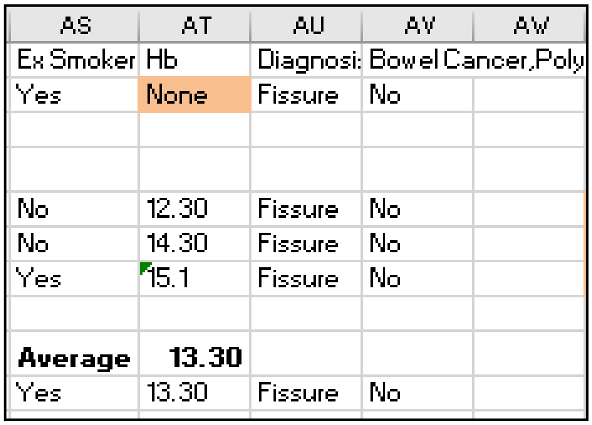
No.	Label	Count
1	polyps	6
9	Polyp	4
3	polyp	31
12	Polpy	2
13	polpy	3
6	Polyp (hyperplastic)	11
7	polyp (tubular adeNoma)	11

Figure7. Five instances of ‘polyp’ caused by spelling and case sensitive errors.

Figure7 shows the values which have been separated as different conditions because of the homonym. This problem can be alleviated by keeping the values consistent, spelled the same and in the same case, as Weka is case sensitive.

3.3.6 Haemoglobin Values

In the original data set, there were 271 missing, unknown and Null Hb values. This represents about a quarter of the data records. The easiest cleaning method is to take out these values and put them into the 'DirtyData' worksheet. There are ways in which all records can be cleaned effectively though. The missing, unknown and null hb values were temporarily placed into a separate worksheet to work on. The data records *with* hb values were separated into each diagnosis e.g. Normal, Polyp, Fissure, etc. into their own individual worksheets.



AS	AT	AU	AV	AW
Ex Smoker	Hb	Diagnosi:	Bowel Cancer,Poly	
Yes	None	Fissure	No	
No	12.30	Fissure	No	
No	14.30	Fissure	No	
Yes	15.1	Fissure	No	
Average	**13.30**			
Yes	13.30	Fissure	No	

Figure8. Working out the mean average of the 'fissure' diagnosis for the missing Hb value

The mean of the hb values in each diagnosis was calculated using the "=AVERAGE()" formula. The missing, unknown and null hb values were subsequently split into *their* diagnosis, and the mean was then added to them. The hb values were then put back into the original worksheet. The example below shows the working in excel. *Figure8* shows the working behind finding the average hemoglobin level in a particular diagnosis outcome. The average 'fissure' hb level was 13.3, so the average was then put into the 'dirty' hb data record.

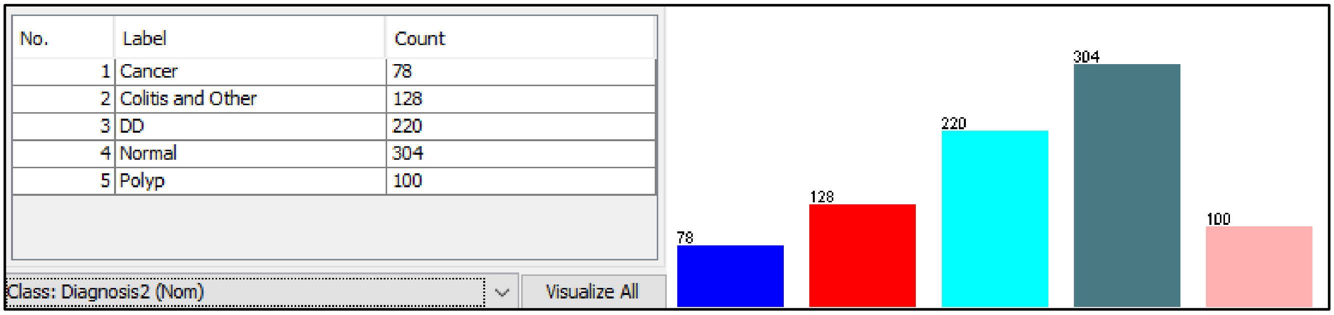
3.3.7 Data Outcomes

As the database is unbalanced with regards to the different outcomes, the diagnosis conditions can be grouped into more general categories, with the data records being distributed equally. The categories can then be used with under-sampling and over-sampling strategies to see if classifier performance improves. This experiment will measure whether over-sampling or under-sampling increases the performance of the classifier.

3.3.7.1 Diagnosis Grouping

The number of records in the attribute, 'Diagnoso1' had an uneven distribution of values in each diagnosis outcome. In order to alleviate this problem, a new attribute was created called 'Diagnosis2', this grouped similar conditions together with regards to its risk. The various diagnosis in the raw data were grouped together into five more general groups. The 'Normal' and 'DD' conditions stayed the same because of the large amount of patient records that had them diagnosis. There were five different types of polyp which were grouped together to make 'Polyp'. Any conditions which were cancerous were then grouped together and added into the 'Cancer' group. Amongst the remaining conditions, Colitis is the most severe as it causes inflammation of the colon (NHS choices, 2016), so it made sense to create a new grouping under 'Colitis'. The remaining conditions, e.g. Colitis, Fissure, Hemorrhoids etc. were grouped together in the 'Colitis' group, renamed as 'Colitis and Other'.

The image in *figure9* shows the new groupings in Weka, bringing the total number of conditions from 18, down to 5. By doing this, it's possible to reduce the bias towards a specific condition because of class imbalance. The next step is to balance the number of data records in each grouping, using both over-sampling and under-sampling.



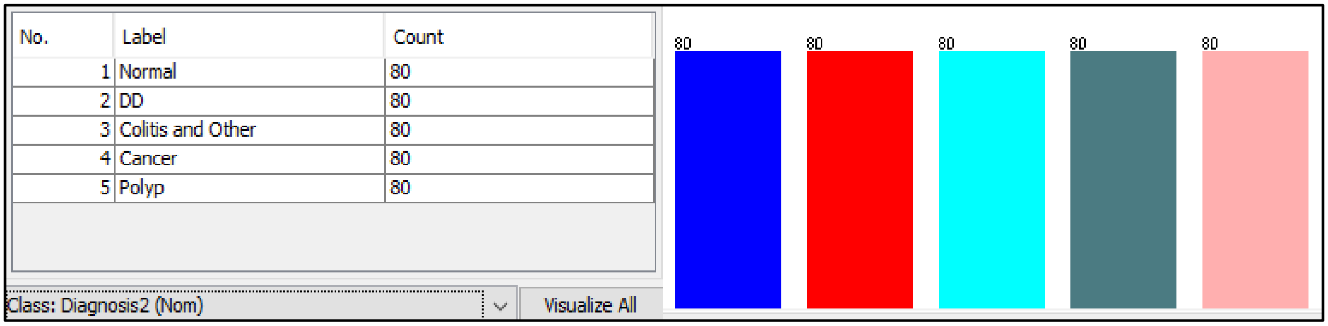
No.	Label	Count
1	Cancer	78
2	Colitis and Other	128
3	DD	220
4	Normal	304
5	Polyp	100

Class: Diagnosis2 (Nom)

Figure9. Screenshot of Weka output showing class imbalance in the diagnosis groupings

3.3.7.2 Under-Sampling

As the figure above shows, there is still a large imbalance in the data records for each diagnosis, especially so with 'DD' and 'Normal'. Firstly, the mean data records for the minority classes e.g Cancer, Colitis and Other & Polyp, were going to be calculated and the number of data records for the majority class would be removed to the calculated mean of 102. However, the eventual test was to bring all the diagnosis data values down to around 80, to match the level of 'Cancer', due to creating a more balanced data set. The records of each class (excluding Cancer) were randomly removed using a randomly generated number from the =RAND() excel function. The under-sampling of the data records in each diagnosis resulted in 48 Colitis and Other, 224 normal, 140 DD and 20 polyp data records being removed.



No.	Label	Count
1	Normal	80
2	DD	80
3	Colitis and Other	80
4	Cancer	80
5	Polyp	80

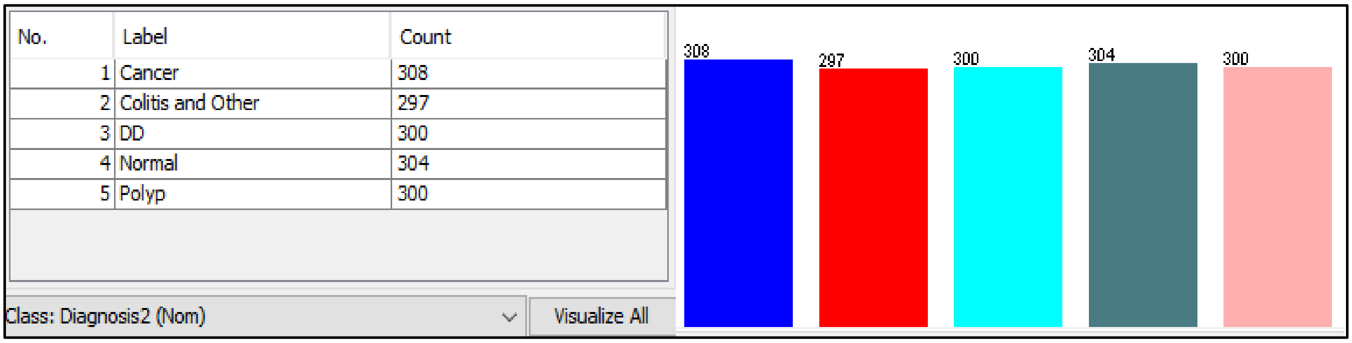
Class: Diagnosis2 (Nom)

Figure10. Using under-sampling to balance the data records in the diagnosis grouping.

Under-sampling allows for the balance of data outcomes, thus limiting a bias towards a specific majority outcome. Two separate worksheets were created for the diagnosis grouping (from chapter 3.2.7.1) and the data record reduction (under-sampling) to determine any significant difference in the accuracy of the classifier. Any rules which would run off this data should be far more accurate because of the balance; however, this wasn't the case. The results will be discussed in chapter 3.7.1.1.

3.3.7.3 Over-Sampling

In contrast to the under-sampling performed in chapter 3.2.7.2, it was decided to see if over-sampling would improve the accuracy of the classifier performance. From the groupings made in chapter 3.2.7.1, the condition 'normal' has 304 data records, making it the majority outcome. As opposed to under-sampling which brings the number of majority class records down to the minority, over-sampling works the opposite way around, bringing the minority up to the majority. It is therefore possible to duplicate the data records from each of the minority outcomes to balance with the majority outcome. To balance the records with 'Normal', 230 records were duplicated for 'Cancer', 169 for 'Colitis and Other', 80 for 'DD' & 200 for 'Polyp'. *Figure11* below shows the over-sampling in Weka.



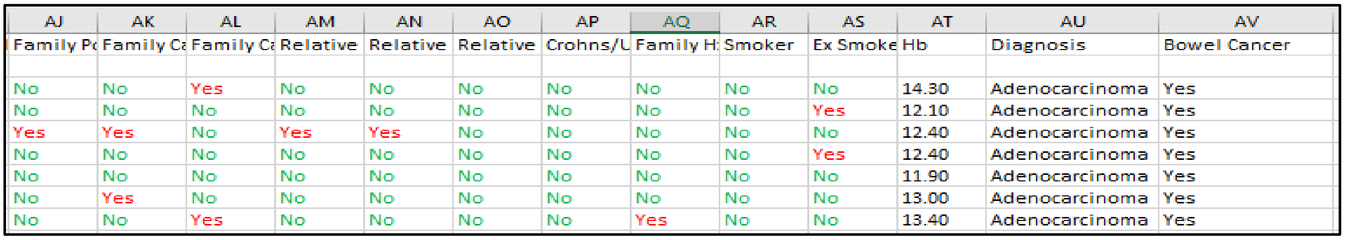
No.	Label	Count
1	Cancer	308
2	Colitis and Other	297
3	DD	300
4	Normal	304
5	Polyp	300

Figure11. Using under-sampling to balance the data records in the diagnosis grouping.

Much like under-sampling, over-sampling will exclude a bias with a particular outcome, again in this case, the majority outcome. All data records for each outcome now match the majority class, which should create more accurate rules. From Weka, it is apparent that the classification accuracy did significantly increase, with a far smaller number of False Negatives. The results will be discussed in chapter 3.7.1.2.

3.3.8 Attribute/Data Correlation

Another small test was carried out to determine if there were any similarities in the conditions of each patient in the separate diagnosis. For example, a selection of data records was randomly picked in the dataset in each diagnosis e.g. Normal and placed into a separate worksheet and examined to see if there were similarities in the conditions of each of the patients. It was realised after the test that the dataset had a large variability, and did not have very many similarities in each of the conditions which could produce important information. Some, however, did have similar results, such as that in *figure12,* illustrating that the patients suffering from Adenocarcinoma all had no relatives with cancer elsewhere, didn't suffer from Crohns and did not smoke.



AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV
Family P	Family C	Family C	Relative	Relative	Relative	Crohns/U	Family H	Smoker	Ex Smoke	Hb	Diagnosis	Bowel Cancer
No	No	Yes	No	No	No	No	No	No	No	14.30	Adenocarcinoma	Yes
No	No	No	No	No	No	No	No	No	Yes	12.10	Adenocarcinoma	Yes
Yes	Yes	No	Yes	Yes	No	No	No	No	No	12.40	Adenocarcinoma	Yes
No	No	No	No	No	No	No	No	No	Yes	12.40	Adenocarcinoma	Yes
No	No	No	No	No	No	No	No	No	No	11.90	Adenocarcinoma	Yes
No	Yes	No	No	No	No	No	No	No	No	13.00	Adenocarcinoma	Yes
No	No	Yes	No	No	No	No	Yes	No	No	13.40	Adenocarcinoma	Yes

Figure12. Checking if there are any similarities in the data under the conditions in each diagnosis

3.4 Mean/Standard Deviation on Haemoglobin Levels

Using Matlab, calculations were made to find out whether there were any major differences between the haemoglobin levels of patients with no cancer, and those who do have cancer. This was done to find whether there was any statistical patterns or differences in the two outcomes. When the calculations were made, it was worked out that the mean for cancer patients was 12.51, whereas the mean for non-cancer patients was 12.53. The standard deviation of the haemoglobin values were then calculated. The standard deviation of non-cancer patient Hb values was 1.73, with cancer being 1.79. This therefore concludes that there was no large statistical variation between the two outcomes with regards to the haemoglobin levels of patients. Therefore, there was no statistical evidence based on the given data values that cancer effects the Hb values of patients.

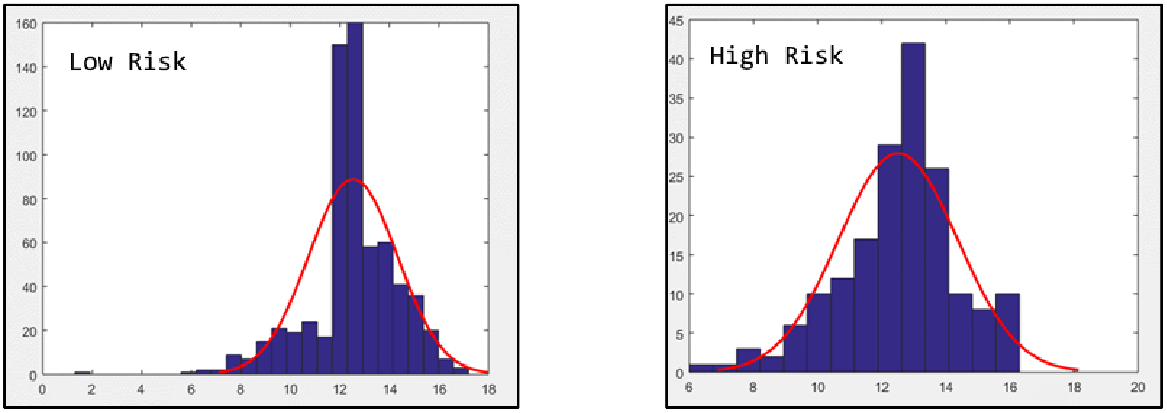


Figure13. Comparison figures of standard deviation of high and low risk hb values

3.5 Data Mining and Rule Generation

Prior to performing the data mining tasks, the selection of a classifier must be made and what rules this project will be using to create the DST. The data mining results, which include the generated rules and deployment will be discussed in chapter 3.5. After the entirety of the cleaning, the data needs to be put through Weka to create rules for the DST. At that point, any attributes which were not needed, can be removed before the data processing is done. An example of an attribute which always needs to be removed is ‘ID’. The reason for this is that ‘ID’ doesn’t add any facts to the decision process, and in particular, it doesn’t relate to a patient’s health; it will also be included in the rule output e.g. IF ID < 712 etc. It was decided that both attributes ‘Diagnosis1’ and ‘Diagnosis2’ will also be removed for this project. The decision support tool will work out the risk of the patient, and won’t be working out the specific condition that the patient has.

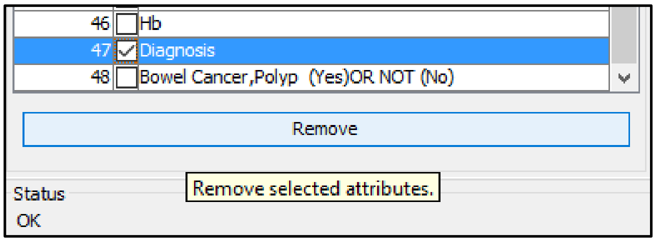

46 □ Hb
47 ☑ Diagnosis
48 □ Bowel Cancer,Polyp (Yes)OR NOT (No)
Remove
Remove selected attributes.
Status
OK

Figure14. Removing attributes which do not add any new facts to the classification of data

3.5.1 Chosen Classifier

It was decided that the project will work with only one classifier, with research into that classifier also being undertaken. The decision to only have one classifier comes from being able to keep the rule generation and decision process as simple as possible, with more focus being on the decision support system. J48 was chosen because of its simplicity in interpreting rules, and for its easily readable model output. All j48 rules will be implemented into the decision support tool.

3.5.2 j48-Decision trees

The classifier 'j48' can be used within Weka under the decision trees category. J48 is an implementation of the C4.5 algorithm (Yan et al., 2015). C4.5 is an algorithm which creates a decision tree based on a data set with labeled input data. The j48 decision can also be used to classify the data, with the dataset that is fed into it. A decision tree contains a top down modeled 'tree' which shows the list of conditions which makes up a particular rule. *Figure15* shows an example of the decision tree mechanism, whilst *Appendix H* shows real decision tree output from Weka. It shows that if the Hb Value is less than or equal to 12.4 and Rectal Bleeding is 'No', then the risk of cancer to a patient is typically low.

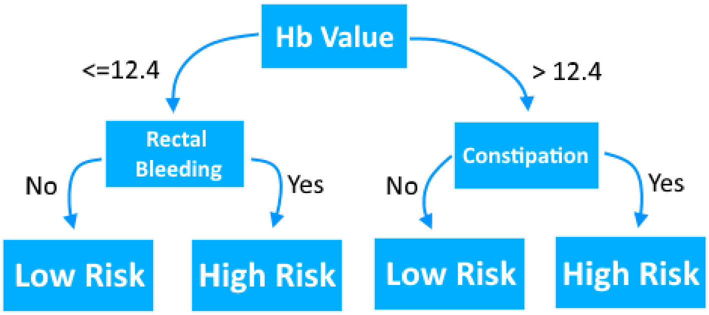


Figure15. Example model of the decision tree classifier, showing top down decision rules

3.6 Software Design

During the preliminary design for the decision support tool (found in *Appendix I)*, the original plan was to create a single form where the user can click from numerous selections, click the 'submit' button and then a second form will appear giving the outcome. The original designs didn't incorporate a backend database, whereby the clinician is able to store the selections and go back to them if need be, however the designs have been revised since to incorporate this feature. All user selections are made from combo boxes with fixed values being "Yes|No", relating to whether a patient has a symptom. Combo boxes were used because it alleviates any errors in user input, by giving the user fixed options to choose from instead of manually entering them in; this also saves the user a lot of time.

The main objective is to make the decision interface as easy to use as possible, so the original design for the user interface had to be rethought. A new design with four separate forms, instead of just one, each with different headings corresponding with the patient symptoms, was created. The original design had too many selections on one form, making the form look cluttered. Separating the symptoms into four categories made the decision interface far easier to use. Each form in the new design has an 'analyse' button, if the user selects this in any given form, the interface will work out the risk by the selections that the user has made. Previously, all 44 symptoms had to be selected in order for a risk to be worked out. However, the new design allows for any number of symptoms to be selected and the risk will be calculated. On first start up, a main menu screen greets the user with the various selections that can be made. *Appendix E* illustrates the flow of the entire program.

Additional features were also added to the interface. A backend database was developed, where the user can store any selections made in the user interface, the purpose of this was so that the clinician can look back on previous patient symptoms. The form 'Database Viewer' houses the datagridview which shows the user selections. The form also has a button called 'export', in which the user can press and the selections then go to a separate excel worksheet. The program also has an additional feature which shows the rule information. When the 'analyse' button gets clicked and a given rule shows the risk, a button can be clicked which takes the user to another form, showing the information used to make up that particular rule. This was done so that the clinician can decide whether the rule is worth following. The final designs meet the objectives set out at the start, which required a lot of testing and redesigns to finally make right, as mentioned in section 3.8.

3.6.1 Interface Forms

The final designs for the interface included four main forms; Clinician Support, Software Help, Database Viewer and Rule Information. The decision interface form is divided into four other 'children' forms, which can be accessed when the user presses the appropriate buttons. Each form will be discussed in detail, with the functionality and any previous design iterations that were made. A full guide on how to use the forms is found in the *Appendix M*.

3.6.1.1 'Welcome' Form

When the application is loaded, this form greets the user with four possible options that can be selected. These forms include; Clinician Support, Software Help, Database Viewer and Rule Information.

3.5.1.2 'Clinician Support' Form

Upon clicking the 'Clinical Support' button in the welcome form, the first clinical support form will load. The first form is called 'Bowel Habits', and houses the various symptoms relating to problems with the bowel. When the user selects the applicable symptoms, a button called 'analyse' can be pressed to check if any rule matches those selections. The form will also show the predicted outcome (Low or High risk), the actual rule used and whether the patient needs to be referred to a specialist. If a rule does appear, a button previously hidden, will be made visible. If the user selects this button, then a second form will appear called 'Rule Information', which will be discussed more clearly in section 3.5.1.5. The original design can be viewed in *Appendix I,* with the updated design in *Appendix J*.

The clinician support form will also contain a 'Summary' button to check the previous selections that have been made, instead of cycling through all the forms. If the user wants to make a new prediction, then the form has a button called 'New Prediction', this will close all background forms and take the user to step one, which is 'Bowel Habits'. If a rule hasn't been found in the first form, then a button called 'Further Questioning' can be selected which takes the user to step two called 'Other Problems'. The third step in the clinical support tool is called 'Effects', whereas the fourth step is called 'Patient/Family History'. When each step is selected, the previous step (form) will be hidden in the background, this is to avoid a cluttered screen. Each form has tabs at the top to make cycling through each of the steps much easier and quicker. If all the symptoms from each step have been selected, the user can then press the 'Add to Database' button. This will store all the user selections into the backend database. All the decision forms will subsequently close and the 'Database Viewer' form will load. The Database Viewer will be discussed in section 3.5.1.4.

3.6.1.3 'Software Help' Form

The software help form can be selected from the main menu, or internally in a dropdown menu from within the decision interface. The software help form will contain three buttons corresponding with the three other main forms in the application. When each button is selected, the user can scroll through the step by step help, relating to which button they pressed. The scrolling can be done using the 'track bar' tool in the design window.

3.6.1.4 'Database Viewer' Form

The database viewer form contains the 'datagridview' function tool. The data from the user selections can be viewed inside this tool, so that the clinician can view any previous patient data. The data is loaded using the 'Load Data' button, which will show the data inside the 'datagridview'. The user can also select an option called 'Export to Excel', whereby all the data stored in the database can be opened in a new formatted excel worksheet. This was developed because the clinician can now easily extract patient details and save them.

3.6.1.5 'Rule Information' Form

The rule information houses the decision tree used for the decision process in the clinician support tool. The button 'View Tree' can be selected which will show the full tree. The user can hover over the various rules and observe the information about that particular rule. As mentioned in section 3.5.1.2, a button can be clicked when a rule appears in the clinician support tool. This will take the user to the rule information form, where the rule which appeared in the DST, will be highlighted in this form with the corresponding information. The form can be viewed in *Appendix L*.

3.6.2 Rule Deployment

It was decided that the rules will be deployed using simple 'IF Else' statements, as time constrictions meant that a better approach could not be met. All the 26 rules which will be used are programmed into the decision support tool. If any of the selections match the criteria in each rule, then a risk will appear. Originally, the plan was to create a separate file with the rules inside it. The file can be read using the 'stream Reader' method which analyses strings (text). This method meant that even the clinician can add new rules inside the file and import it into the decision support tool. Unfortunately, due to time constrictions with other tasks, this idea was unfortunately not used.

3.7 System Implementation & Data Mining Results

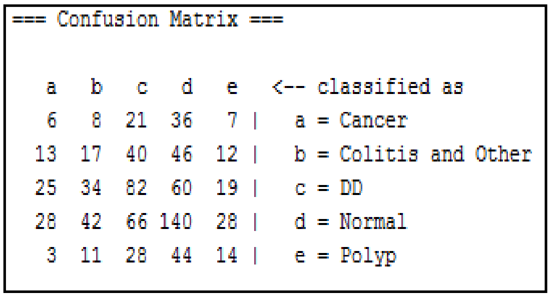
This section will discuss the implementation of the decision support system, the results of the data mining and the generated rules used. The code used to build the application will also be discussed, as well as any issues that arose during the implementation. This section will also analyse the chosen tree, with research behind the generated output.

3.7.1 Data Mining Results

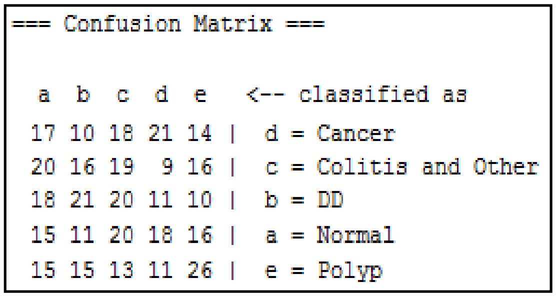
After data preparation was done to avoid a bias, the data can now be used in weka to compare the classifier output. As mentioned in section 3.2.7, under-sampling and over-sampling were done on the data to create balanced outcomes; this should in turn create better rules than compared to the original diagnosis grouping data set.

3.7.1.1 Under-Sampled Data Set

The decision tree was used on both the under-sampled dataset and the diagnosis grouping data set to work out the difference in classifier performance. The accuracy or ‘correctly classified instances’ of under-sampling were much lower than the grouping dataset. However, the number of patients correctly classified as having cancer was three times higher. Normally, in a medical domain, increasing the number of correctly classified cancer patients is necessary. However, as the accuracy at predicting the other conditions was poor. *Figure16* shows the comparison in matrices from each data set. Diagnosis grouping in on the left, under-sampling is on the right.



```
=== Confusion Matrix ===

  a   b   c   d   e   <-- classified as
  6   8  21  36   7 |   a = Cancer
 13  17  40  46  12 |   b = Colitis and Other
 25  34  82  60  19 |   c = DD
 28  42  66 140  28 |   d = Normal
  3  11  28  44  14 |   e = Polyp
```



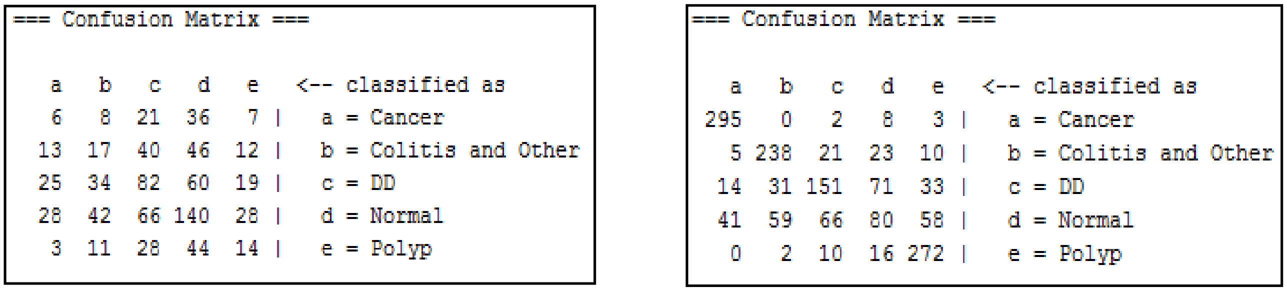
```
=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
 17 10 18 21 14 |  d = Cancer
 20 16 19  9 16 |  c = Colitis and Other
 18 21 20 11 10 |  b = DD
 15 11 20 18 16 |  a = Normal
 15 15 13 11 26 |  e = Polyp
```

Figure16. Comparison matrix between original diagnosis grouping (left) and under-sampling (right)

Under-sampling the data managed to improve the number of cancer patients correctly classified as having cancer, however the number of correctly classified instances as a whole, was poor (25%). Even though the number of correctly classified cancer patients was higher, there were still around 17/70 correct cancer classifications with under-sampling, or around 24% of the total number of instances. It was therefore decided not to use the rules from this particular output.

3.7.1.2 Over-Sampled Data Set

As mentioned in chapter 3.2.7.3, over-sampling was done to the data to check the classification accuracy and whether the number of correctly classified instances of ‘cancer’ had increased. The number of data records were increased to around 1500, with all conditions equal at around 300 data records each, matching the normal diagnosis, to stop any bias. The results of this test revealed that the number of correctly classified cancer patients increased remarkably and so did the accuracy compared to the grouping dataset (68%). The figures below show the groupings on the left compared to the oversampling on the right.



```
=== Confusion Matrix ===

  a   b   c   d   e   <-- classified as
  6   8  21  36   7 |   a = Cancer
 13  17  40  46  12 |   b = Colitis and Other
 25  34  82  60  19 |   c = DD
 28  42  66 140  28 |   d = Normal
  3  11  28  44  14 |   e = Polyp
```

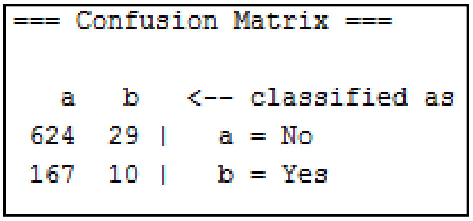
```
=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 295   0   2   8   3 |   a = Cancer
   5 238  21  23  10 |   b = Colitis and Other
  14  31 151  71  33 |   c = DD
  41  59  66  80  58 |   d = Normal
   0   2  10  16 272 |   e = Polyp
```

Figure17. Comparison matrix between original diagnosis grouping and over-sampling.

In conclusion, the number of correctly classified cancer patients increased from 6 to 295. The percentage of correctly classified cancer patients against the total number of cancer instances was 95%. In a real world scenario, this technique would have been chosen as the number of correctly classified cancer patients is very high. However, the decision tree produced 328 leaves (rules). Implementing all these rules would have taken a very long time, so therefore this project did not use the rule output from this test. The poor accuracy of classifier could stem from an asymmetric decision tree, which will be discussed in section 3.7.1.4.

3.7.1.3 Bowel Cancer Data Set

The original idea was to use a decision tree which predicted the condition of a patient, which will then be used to work out the risk. However, it was then decided to just predict the risk of a patient and neglect the prediction of a specific condition altogether. A new fully-cleaned worksheet titled 'Project Data' was created with just the symptom attributes and whether the patient has cancer or not. After putting the data set through weka, the classifier accuracy of this data was 76%, which was higher than both the under-sampling and over-sampling. The number of leaves (rules) was 26, and were enough for the implementation of the project. Conclusively, it was decided to use the rules from this data to implement into the DST. *Figure18* below shows the output in the confusion matrix.



```
=== Confusion Matrix ===

   a   b   <-- classified as
 624  29 |   a = No
 167  10 |   b = Yes
```

Figure18. Confusion matrix showing the classifier output for the target class 'Bowel Cancer'

3.7.1.4 Asymmetrical and Symmetrical Decision Trees

Decisions trees tend to capture asymmetries globally, instead of capturing them at a local level, this can cause the tree to exponentially grow, thus limiting the use and decreasing the predictive power of the classifier (Bielza & Shenoy, 1999). After analysing the chosen decision tree, it's quite clear that the conditions which make up the rules aren't effective enough to truly predict the risk of bowel cancer.

After experimenting with the data and the classifiers, the chosen tree had an accuracy of 76.3%. Even though the accuracy is relatively high compared to previous classification attempts, the conditions which make up the rules do not reflect true medical diagnosis.

Model output in Weka does not indicate the symmetry of a decision tree. For example, it models only one half of the tree, and doesn't visualise the opposite side. In many occasions, the tree could be unbalanced, or asymmetric, thus it's possible that the other side of the tree could be stronger and contain more relevant predictors of bowel cancer. This imbalance negatively effects the predictive power of the classifier, producing rules which aren't strong enough to support a specific outcome. A symmetric tree has decision paths and nodes which run parallel to each other, whereas an asymmetric tree does not. The decision paths then become unequal to each other, possibly creating an imbalance in the way the outcomes are classified. An asymmetric tree highlights the importance of the combined effect of two indicators in displaying the conditions of a particular outcome. A simple example shown in *figure19* demonstrates exactly what can happen if the tree is asymmetrical, and what information can be taken from it, and how the problems can be alleviated.

It is possible to transform an asymmetric tree to a symmetric tree by neglecting the opposite node and expanding the chosen node to create symmetry (Beroggi, 1999). Referring to the figure, image one shows an example of an asymmetric tree, with nodes that aren't parallel with one another, containing different conditions. In this instance, 'On Motion' can be neglected with the branch 'Y', with focus now turning to branch 'N' (illustrated in image two). The 'N' branch can then be expanded to match the conditions in the 'Y' branch, achieving symmetry as illustrated in image four. This then creates a balanced tree, which could create much better rules than the previous unbalanced decision tree.

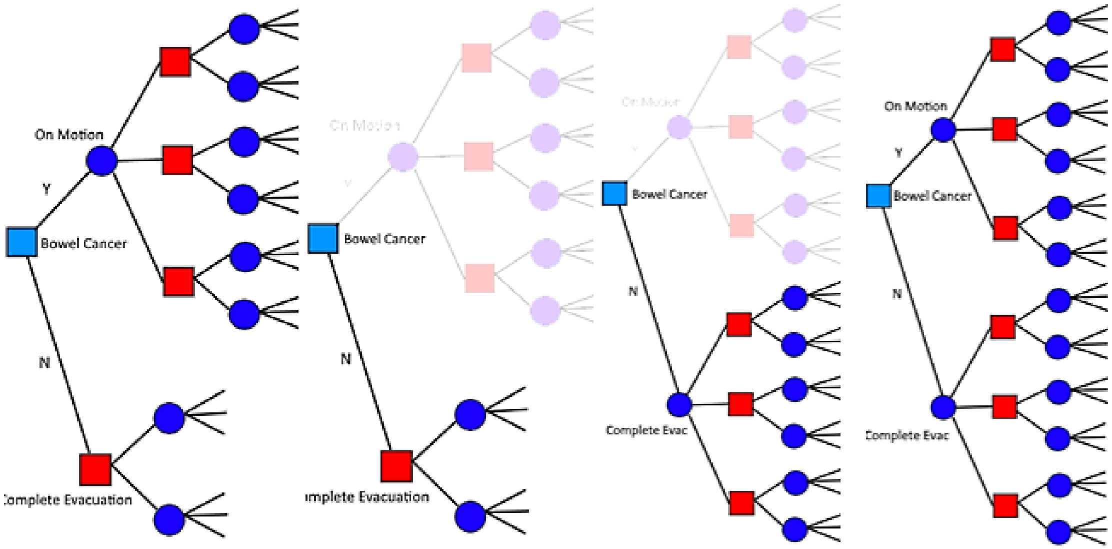


Figure19. Step-by-step process of solving an asymmetric tree problem from the class 'Bowel Cancer'

3.7.1.5 k-fold Cross-Validation

Changing cross-validation parameters (folds) was used in this project to find the best classifier accuracy. Cross-validation is a statistical method of splitting a data set into two distinct segments; the first segment is called the training set (Refaeilzadeh et al., 2008). The training set is used to discover predictive relationships of the patterns in the data set. A larger portion of analysing the data is used for the training set, with a smaller portion used for testing. The test set is used to test the strength of any pattern discovered in the training set. The data gets split into equally sized portions (folds), and each test set is tested against the rest of the training set (Refaeilzadeh et al., 2008).

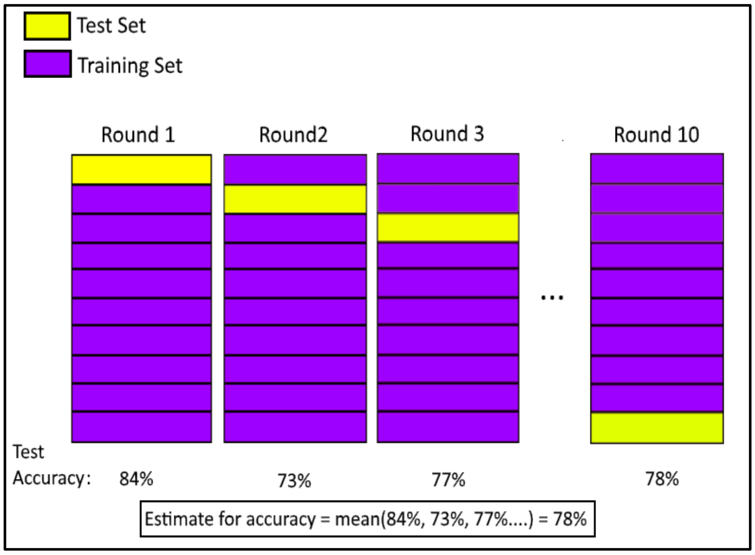


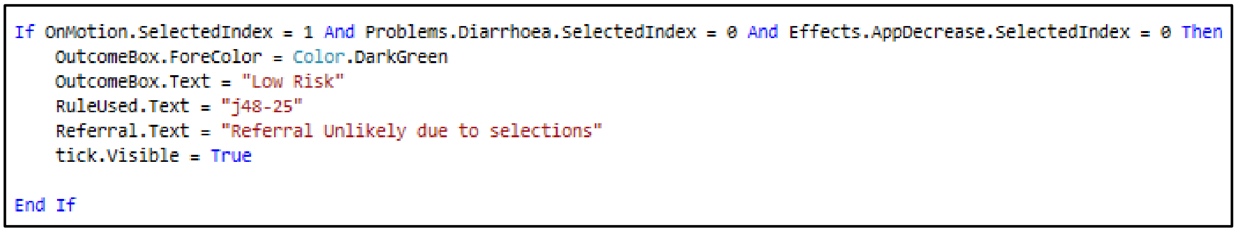
Figure20. Cross-validating data using folds and training set

For example, if the data gets separated into k~10, with one portion (10%) allocated to the test set, the training set will therefore be 9/10 or 90% of the data set. The test set tests each of those folds measuring the performance. The performance of the learning algorithm on each of these folds is determined by using a performance metric like accuracy. The accuracy of each correctly classified fold is measured, and the generalized mean of each fold accuracy is worked out (Bengio & Grandvalet, 2004). Reducing the number of folds to five (20% test set, 80% training set) reduced the accuracy of the classifier. This may be because less of the analysis is done through the training set which works out the statistical data patterns and trends. The number of folds used in this project was ten, as it produced the highest accuracy of correctly classified instances.

3.7.2 Decision Rule Integration

Initially, the project was going to be developed with rules which could be added by the user. For example, the user can add their own rules in a data file, the decision interface will read each line of the data file, with each attribute element being separated and analysed against the selections made in the decision interface. This technique would have allowed for any data file with any rules to be implemented without them being hardcoded; the system would simply recognize the new rules in the 'uploaded' file by the clinicians. This would have made the system more professional and robust. However, due to an inability to get the system working at a satisfactory level and resulting time restrictions, it was decided to hardcode the rules into the system.

The system now uses hardcoded ‘IF Else’ statements, with all the applicable rules used from the decision tree output. The IF statements contain the conditions of each rule, and if the user selections match those in any of the IF statements, the system will acknowledge this and produce a message telling the user which rule was used and its risk level. If the user selections do not match up to the conditions in each IF statement, the system will tell the user that no rule has been found for the selections that have been made. One of the IF statements used in the program is illustrated in *figure21* below.



```
If OnMotion.SelectedIndex = 1 And Problems.Diarrhoea.SelectedIndex = 0 And Effects.AppDecrease.SelectedIndex = 0 Then
    OutcomeBox.ForeColor = Color.DarkGreen
    OutcomeBox.Text = "Low Risk"
    RuleUsed.Text = "j48-25"
    Referral.Text = "Referral Unlikely due to selections"
    tick.Visible = True

End If
```

Figure21. IF statement used for rule-25, showing the conditions which make up that rule.

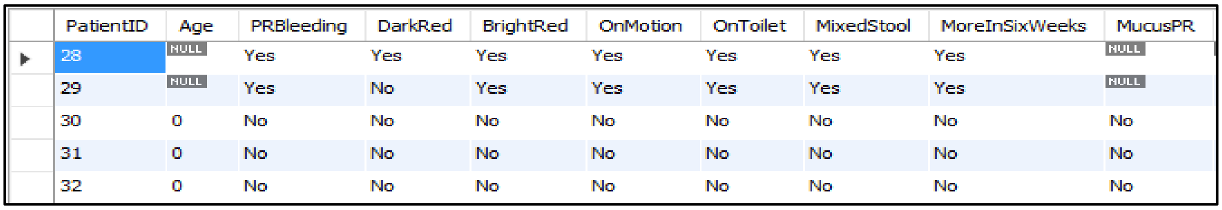
The above illustration shows an IF statement used in the system. This particular example is for rule ‘j48-25’, and lists the conditions used to make up that rule. For example, if all the conditions are met in this statement with the user selections in the decision support system, then an output is printed into text boxes which tells the user that this rule has been chosen. There are 26 individual IF statements, this ended up very time consuming with regards to integrating each of the statements. However, the functionality of the system works with any of the selections made in the decision support system.

3.7.3 Database Integration

As mentioned in part 3.1.4, later on in the design process it was decided to integrate a backend database to store all the user selections made in the decision support system. The integration of this was fairly simple, all it required was MySQL to house the database, and code to interact with the elements inside the decision support system. When the user has selected all the applicable symptoms, a button can be pressed which immediately stores all the selections into the backend database.

3.7.3.1 Using MySQL Database

The database used in MySQL is localhost only, meaning that it will only function on the computer where the database is stored. Localhost connection was used as opposed to connecting over a network because the project would benefit more if it was only on the clinician’s computer, due to security and patient confidentiality. *Figure22* below shows the database with five patients with the ‘Yes/No’ selections from the decision support system



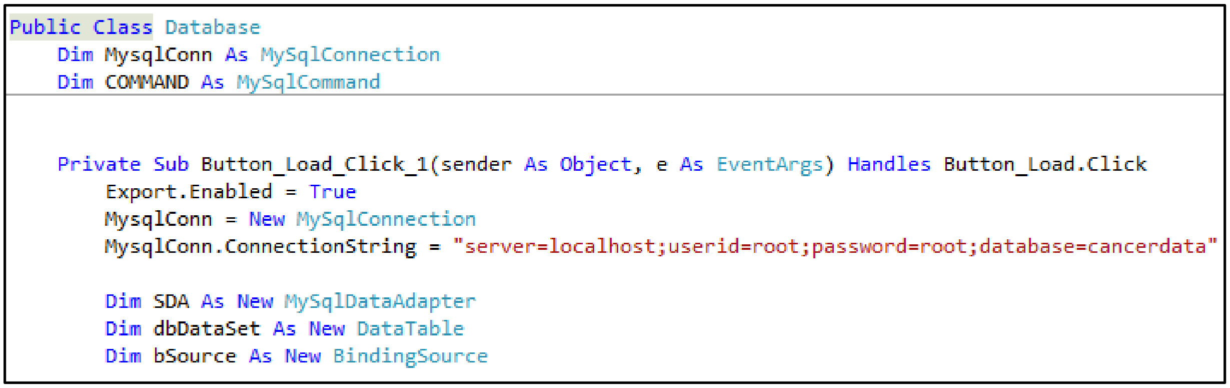
PatientID	Age	PRBleeding	DarkRed	BrightRed	OnMotion	OnToilet	MixedStool	MoreInSixWeeks	MucusPR
28	NULL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	NULL
29	NULL	Yes	No	Yes	Yes	Yes	Yes	Yes	NULL
30	0	No	No	No	No	No	No	No	No
31	0	No	No	No	No	No	No	No	No
32	0	No	No	No	No	No	No	No	No

Figure22. Using ‘DataGridView’ in VB.NET, showing the grid populated with patient symptoms.

The above illustrates how the database will look with the selections made in the application. Under each attribute, there will be a value of either ‘Yes’ or ‘No’, consistent with the data taxonomy in this project. The patient will be assigned an incremented Patient ID, with a value that is specific to each patient.

3.7.3.2 Data Flow Between Database and Application

Once the database had been created, the application now needed to link up with the application. A MySQL library had to be imported into Visual Studio in order to work with connection strings used to interact with an external database. The code used was implemented inside a ‘Button_Click’ private sub called ‘Load’, where the function used to create a link was activated with a button press. *Figure23* below shows the code used to create the connection.



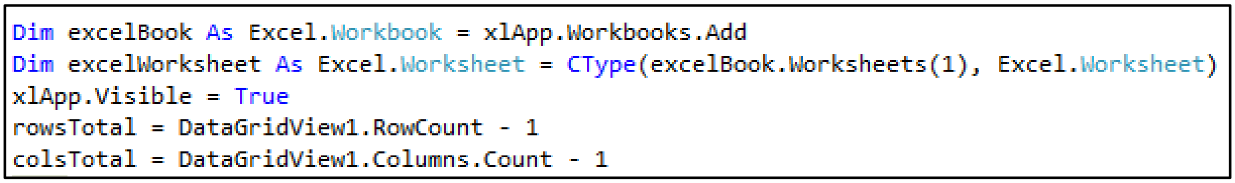
```
Public Class Database
    Dim MysqlConn As MySqlConnection
    Dim COMMAND As MySqlCommand


    Private Sub Button_Load_Click_1(sender As Object, e As EventArgs) Handles Button_Load.Click
        Export.Enabled = True
        MysqlConn = New MySqlConnection
        MysqlConn.ConnectionString = "server=localhost;userid=root;password=root;database=cancerdata"

        Dim SDA As New MySqlDataAdapter
        Dim dbDataSet As New DataTable
        Dim bSource As New BindingSource
```

Figure23. Code showing the connection between the database and the application.

Two variables were set with both the connection and the command, which will be used to open up the connection and close it again between the database and the application. On the button press, the connection will open with the ‘ConnectionString’ telling the system what database to connect to. The remaining three variables allow the data from the MySQL database to populate the ‘DataGridView’ element in the application.

3.7.3.3 Exporting Data to Excel

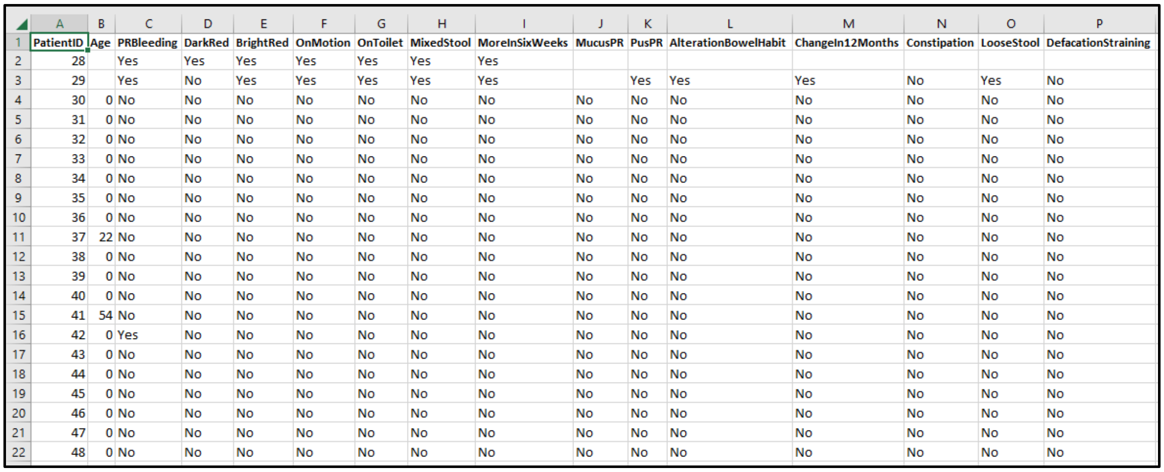
The application also allows for any data in the ‘DataGridView’ to be exported to a new excel worksheet for easier readability, and also to save patient details. When the ‘Export’ button is pressed, a new workbook will be created with the data from the DataGridView inside it. *Figure24* shows a section of the code which handles the creation of the workbook, and works out the number of columns from the DataGridView to add to the new workbook.



```
Dim excelBook As Excel.Workbook = xlApp.Workbooks.Add
Dim excelWorksheet As Excel.Worksheet = CType(excelBook.Worksheets(1), Excel.Worksheet)
xlApp.Visible = True
rowsTotal = DataGridView1.RowCount - 1
colsTotal = DataGridView1.Columns.Count - 1
```

Figure24. Code showing how to export the data to a new excel worksheet

The 'excelBook' variable is added to create a new workbook upon the button press. The 'excelWorksheet' variable was created to initialize a location to where the data will be going, in this case, the first worksheet in the workbook. The number of rows and columns are then counted and taken from the DataGridView and imported into the new worksheet.

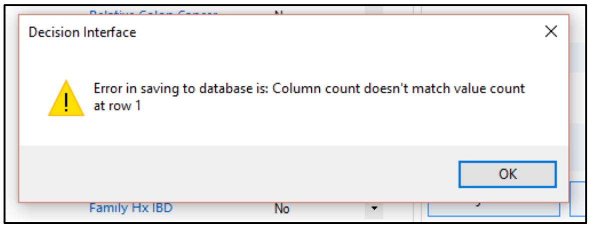


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	PatientID	Age	PRBleeding	DarkRed	BrightRed	OnMotion	OnToilet	MixedStool	MoreInSixWeeks	MucusPR	PusPR	AlterationBowelHabit	ChangeIn12Months	Constipation	LooseStool	DefacationStraining
2	28		Yes	Yes	Yes	Yes	Yes	Yes	Yes							
3	29		Yes	No	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	No	Yes	No
4	30	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
5	31	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
6	32	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
7	33	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
8	34	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
9	35	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
10	36	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
11	37	22	No	No	No	No	No	No	No	No	No	No	No	No	No	No
12	38	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
13	39	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
14	40	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
15	41	54	No	No	No	No	No	No	No	No	No	No	No	No	No	No
16	42	0	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No
17	43	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
18	44	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
19	45	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
20	46	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
21	47	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No
22	48	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No

Figure25. Illustration showing the workbook after all of the data has been exported to Excel

3.8 Software Testing

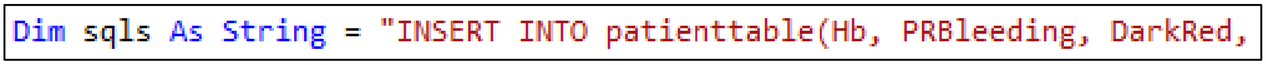
The decision support system required rigorous testing in order to provide a system which met the specified project objectives. As the development of the system progressed, each feature was tested to ensure it worked appropriately. Any feature which did not work correctly, was amended and subsequent tests were carried out. Arduous testing was carried out on the functionality of the database, testing whether there was the connection between the application and the database itself. On many occasions, setting the attributes to be exported to the database was not done correctly, so the transfer of data would fail. *Figure26* below shows one of exceptions which was thrown when an error occurred.



Error in saving to database is: Column count doesn't match value count at row 1

Figure26. Error message showing problems with adding the data to the database.

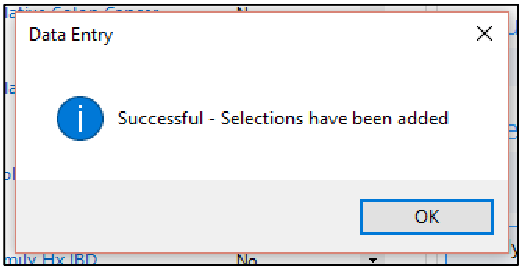
The error message shows that something is wrong with the column count in the first row, therefore there must be a problem with storing the data from the first column. The first column in the database is called 'Hb', and due to an implementation error, the Hb attribute was not initialized in the program, however the rest of them were done correctly.



```
Dim sqls As String = "INSERT INTO patienttable(Hb, PRBleeding, DarkRed,
```

Figure27. Using a string to initialize and connect the attributes in the MySQL database.

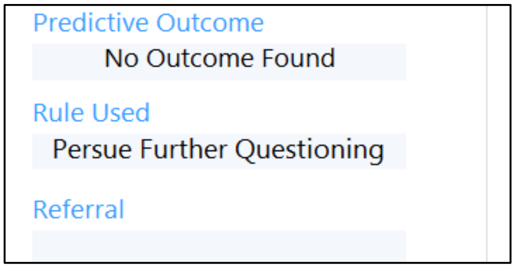
The string contains all the attributes used to store the data in them. If one of them is missing, an error will come up saying that the selection in the 'Hb' combo box couldn't be stored into the Hb column in the database. This is because it hasn't been told where the data is going to go, thus an error is thrown, resulting in a broken program. *Figure28* below shows what happens when all selections have been added successfully to the database.



Data Entry

Successful - Selections have been added

Figure28. Successful data input to database



Predictive Outcome

No Outcome Found

Rule Used

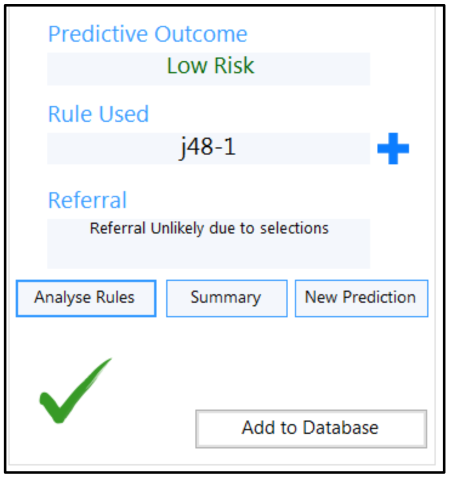
Persue Further Questioning

Referral

Figure29. No outcome found in decision tool

If all the attributes have been initialized correctly, a message box will appear telling the user that all the selections were added. When 'Ok' is clicked, the decision support tool will close and the 'Database Viewer' form will load with the previous patient details.

Another test that was performed was testing whether the decision support tool was working correctly. The main test was seeing whether the text boxes would get any output when the 'analyse' button was pressed. *Figure29* shows the testing on the decision support tool. The system is correct in saying that no outcome was found, as no IF statement matched the selections made in the decision support tool. Therefore, the test was successful and further implementation to the system can continue. A final thorough test was carried out on the decision support tool to see if any of the rules would appear, either low or high risk. The workings of this part of the decision support tool is critical, as it's the fundamental part of the software, and getting it to work perfectly is key. The first test, illustrated in *figure30,* indicates what happens when a low risk is given to the patient based on the input given in the decision support. Concurrently, the second test illustrated in *figure31* shows a high risk being allocated to a patient. A number of different patterns of Yes/No values were entered into the decision support, and it was proven that the decision support system meets the objectives, as each allocated rule showed correctly.



Predictive Outcome

Low Risk

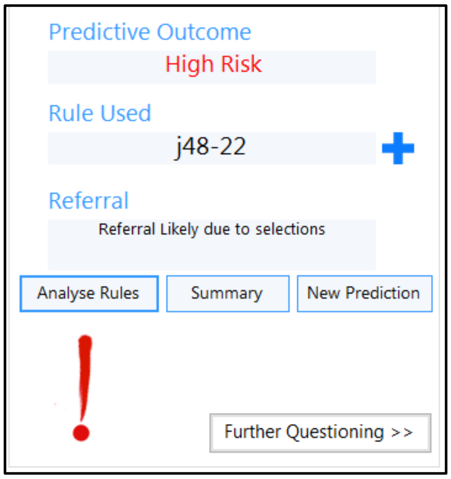
Rule Used

j48-1

Referral

Referral Unlikely due to selections

Figure30. Low risk prediction in the DSS



Predictive Outcome

High Risk

Rule Used

j48-22

Referral

Referral Likely due to selections

Figure31. High risk prediction in the DSS

Chapter 4. Critical Evaluation

This section will provide a personal reflection on how well the finished system meets the project objectives set out at the start. It will also provide an analysis on each key component that was developed, and how it can be improved. Conclusions will also be drawn on both the subject matter and the project itself.

4.1 Project Achievements

Relating to the project objectives drawn up before the development in chapter 1.4, this section will make an analysis on how accurate the final system meets the five numbered individual objectives.

1. The first objective was to independently understand the importance of using data mining to provide a detailed analysis of clinician information. This objective was fulfilled, as a lot of research was put in to understand certain medical terminology that is useful for the development of the project. A lot of time was put in to understanding the core principles of extracting useful knowledge from databases. Although there was knowledge in data mining due to previous activities, the project expanded upon that knowledge, allowing for detailed analysis of knowledge discovery and more advanced data mining techniques.
2. The second objective was to use Weka as the main data mining tool for providing decisions, based on the data supplied by Castle Hill. The project used Weka due to it being available for free, its familiarity and ease of use. The original plan was to use multiple classifiers to create the decision rules and pick the best ones for the decision support tool. However, the plans changed and the project now runs off just the decision tree output. The change still allows the goal to be met, as Weka as still used to provide the decision output for the application.
3. The third objective was to provide a graphical user interface for the decision rules to be implemented in, so that a clinician can make decisions based on patient symptoms. The decision interface was built using the output in Weka, containing multiple forms and other features to support the decisions made in the tool. This met the project objective as a fully functioning decision tool was built which allowed the user to check the risk of any patient. Also, the objective was to provide a backend database which stores the user selections from the decision support tool. This feature also allows for the data to be viewed inside the decision support tool, and also allows for the data to be exported back to excel. Originally, the plan was to just have the backend database with the data viewable in the application. However, further development was made to this feature to allow user to export all the patient details to a new excel worksheet.
4. The fourth objective was to gain a deeper understanding of data mining techniques through practice and external research. By doing this project, it meant that thorough research was

undertaken to find out various data mining techniques which were previously not understood or thought to even exist; this research was used to coherently back up the work that was done in this project.

5. The final objective was to conclude and make recommendations on how to improve the way clinical data is used and visually represented with regards to using a decision support tool. An important part of the project was to make sure that the decision interface is easy to use, therefore if there is a risk to a patient, it must be visually represented in a way that will be easy to view. The interface links also together in a linear way. For example, the user can go from the user interface to the backend database with just a few buttons. The same method was applied to the rest of the features, making each form interlink without any possible confusion.

Apart from these primary objectives, it was also discovered that due to the asymmetric tree, it's possible that the results which were gathered aren't as accurate as they could have been, as much of the data isn't being classified due to only half of the decision tree being modeled. Even though many data mining tasks were implemented throughout this project, it was difficult to discover a high enough accuracy which could have been used in a real world medical domain. It's possible that if further data mining tasks were implemented, the accuracy of the rules could increase. Other research discoveries such as folding and statistical correlation, helped make the process of data classification a lot easier to understand.

4.2 Alternative Solutions

Some alternative support systems do exist and are being used in the medical sphere. One example is Isabel, which also helps clinicians' decisions if a patient has a differential diagnosis, which means that a patient has symptoms but no outcome can be assigned from a GP (Ramnarayan et al., 2004). Isabel can be downloaded as an app and the website itself has the algorithms installed so that a patient can see their diagnosis online. Another organisation that uses online decision support is the NHS. The NHS symptom checker application allows the user to select their current symptoms, and an algorithm from mined data will give the user a diagnosis based on user input (Semigran et al., 2015). Therefore, it may be possible to build a clinician support tool and have a web application with a primary purpose for a wider, more general use. Many industrial clinician support tools like Isabel, also hone in clinician skills and help them increase their diagnostic skills by providing them concise information on how an outcome is provided from the data that was used. This would benefit greatly in the development of the project by providing a level of diagnostic education, as well providing an outcome.

4.3 Further Project Development

4.3.1 Data Preparation/Data Mining

There were further developments that this project could have undertaken which could have made the decision support tool a lot stronger. The first improvement would have been made to the data records which were removed during the cleaning process. Although the missing, null and unknown Hb values were 'restored', there were three other data records which could have been treated and put back into to the worksheet, 'CleanData'. Two data records had missing values in certain attributes, the modal average of the attribute could have been entered into the missing areas. Other advanced techniques of data mining could also have been looked at and further researched, however time restrictions due to other commitments meant that unfortunately this could not happen.

4.3.2 Code Improvements

Many code improvements could have been made which were previously outlined in this report. The decision support tool could have been made better by creating a tool which allows the user to import the rules on a worksheet, the tool would use these rules to analyse the selections made in the decision support, instead of using simple IF statements. Another concept that was brought up was creating a feature which would give the clinician the freedom to click from the previous patients that were diagnosed with a certain rule and risk value. Upon clicking the patient, the rule information and diagnosis would appear.

4.4 Personal Reflection

This project taught the important ethical issues of using clinical decision support tools to predict a patient risk, as well as teaching the fundamental principles of mining clinical data. As the project is larger than any that was undertaken, time management was an issue initially. However, as time went on, managing the time became easier as the project became more familiar. Looking back to the original time plan, it’s clear that there was some naivety towards the extent of project, thinking that some tasks would take longer or shorter than they actually were. Some features outlined in the initial report were slightly ambiguous, therefore they were not implemented into the final system.

4.5 Conclusion

The use of clinical decision support tools in the medical world is increasing as the technology behind the decision making becomes far more accurate. A number of issues still remain with regards to how far these tools can be used to support clinical diagnosis. This project has demonstrated that a methodical approach must be undertaken in order to ensure that the generated rules are accurate enough to predict a patient risk. Any data mining that is done in real world clinical decisions, has to exercise an advanced level of knowledge discovery.

Therefore, the discoveries that were made in this project conclude that even though it’s possible to adopt a methodical approach to data mining and discover new trends and patterns that were completely hidden, there are also other things which could hamper the results that were previously unforeseen. As previously mentioned, these include the asymmetric decision tree which skews classification accuracy as there is an unbalance. Using the right amount of folds to test the data is also important as using too many folds could cause over-parameterization, which leads to a weaker accuracy. Concurrently, using a small amount of folds could lead to the same result. It’s therefore important to decide that if classification of data is done to extract knowledge, particularly in real medical domain, other aspects which may impede results must be considered before release.

Word Count: 14,453

5 Bibliography

Arnott, D., Pervan, G., O'Donnell, P., Dodson, G. (2004). An Analysis of Decision Support Systems Research: Preliminary Results. Melbourne, Australia.

Ashwinkumar, U.M & Anandakumar, K.R. (2010). International Journal of Computer Applications. *Ethical and Legal Issues for Medical Data Mining*. 1 (28), p7-8.

Bellazzi, R & Zupan, B. (2008). International journal of medical informatics. *Predictive data mining in clinical medicine: Current issues and guidelines*. 77, p94.

Bengio, Y. & Grandvalet, Y. (2004). No Unbiased Estimator of the Variance of K-Fold Cross-Validation. Montreal, Canada.

Berner, E.S., 2009, Clinical Decision Support Systems: State of the Art, Rockyville, Maryland.

Beroggi, G. (1999). Decision Trees. In: *Decision Modelling in Policy Management*. Delft: Springer. 210-211

Bielza, C. & Shenoy, P.P. (1999). A Comparison of Graphical Techniques for Asymmetric Decision Problem. *Management Science*. 45 (11), 1554.

Bramer, M., 2007, Principles of Data Mining, London, Springer-Verlag London Limited.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. (2000). CRISP-DM 1.0. *Step-by-step data mining guide*. N/A (N/A), 1-6.

Chaudhuri, S. & Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology, Palo Alto, California.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1997) Good general Intro to Data Mining from Data Mining to Knowledge Discovery in Databases by, AI Magazine. PDF at: http://www.csd.uwo.ca/faculty/ling/cs435/fayyad.pdf

Garg, A.X., Adhikari, N.K.J, McDonald, H., Rosas-Arellano, M.P., Devereaux, P.J., Beyene, J., Sam, J., Haynes, R.B., 2005, Effects of Computerized Clinical Decision Support Systems on Practitioner Performance and Patient Outcomes, [online], Available: http://jama.jamanetwork.com/article.aspx?articleid=200503. [Accessed 02/02/2016]

Giberta, K., Sànchez-Marrèa, M., Codina, V. (2010). Choosing the Right Data Mining Technique: Classification of Methods and Intelligent Recommendation. *International Environmental Modelling and Software Society*. 1-3.

Inan, D.I., Juita, R. (2011). Analysis and Design Complex and Large Data Base using MySQL Workbench. *International Journal of Computer Science & Information Technology (IJCSIT)*. 3 (5), 174-175.

Jagtap, S. & Dr. Kodge B. G. (2013) *Census Data Mining and Data Analysis using WEKA. International Conference in Emerging Trends in Science, Technology and Management. Singapore. 2013, 35.*

Jerzy Letkowski. (2014). Doing database design with MySQL. *Journal of Technology Research*. 6 (1), 2-4.

Li, DC., Liu, CW., Hu, SC. (2010). A learning method for the class imbalance problem with medical data sets, Taiwan.

NHS Choices. *Ulcerative colitis.* Available: http://www.nhs.uk/Conditions/Ulcerative-colitis/Pages/Introduction.aspx. Last accessed 01/02/2016.

Rahman, M.M., & Davis, D.N. (2013), Addressing the Class Imbalance Problem in Medical Datasets, Denmark.

Ramnarayan, P., Kulkarni, G., Tomlinson, A. and Britto, J. (2004). ISABEL: a novel Internet-delivered clinical decision support system. *ISABEL Clinical Decision Support Systems*, 245-247.

Ranjit S. Veen. (2008). Introduction. In: *Medical Data Mining Issues and Experiments*. Washington D.C: American University. p1-3.

Refaeilzadeh, M., Tang. T., Liu, H. (2008). Cross-Validation. Arizona State University.

Semigran Hannah L., Linder Jeffrey A., Gidengil, Courtney. and Mehrotra Ateev. Evaluation of symptom checkers for self-diagnosis and triage: audit studyBMJ, 2015; 351:h3480

Shafique, U., Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *Innovative Space of Scientific Research Journals*. 12 (1), 217-220.

Sheth, A.A., Longo, W., and Floch, M.H. (2008). Diverticular Disease and Diverticulitis. *American Journal of Gastroenterology*. 103, 1550.

Sittig, D.F., Wright, A., Osheroff, J.A., Middleton, B., Teich, J.M., Ash, J.S., Cambell, E. and Bates, D.W., 2007, Grand challenges in clinical decision support, Boston, Elsevier Inc.

Tripathi, K.P. (2001). Decision support system is a tool for making better decisions in the organization. Kolhapur, India. 2 (1).

Yan, N.C., Ju, W., Fang, H. & Reika, S. (2015). Application of J48 Decision Tree Classifier in Emotion Recognition Based on Chaos Characteristics. Changchun University, China

SIS, 'Processes in Data Mining', SIS's Blog. Available online: http://sisbinus.blogspot.co.uk/2014/11/processes-in-data-mining.html [Accessed 04/05/2016].

Appendix A – Risk Analysis

Risk	Severity (L/M/H)	Likelihood (L/M/H)	How to Avoid/Mitigate	Residual Impact (L/M/H)
Data loss	H	M	Keep Backups	L
Loss of backups	H	L	Multiple Backups	L
Hard drive corruption	H	L	Make sure software is sent via SVN and backed up on external	L
Software corruption	H	M	Make sure software is sent via SVN	L
False Information	M	M	Thorough research is carried out	L
Cancer diagnosis changes	M	L	Use system flexibility	L
Wrong risk factor	H	M	Test to see if the application works properly	L
Poor data cleansing	L	L	Make sure data is cleansed to avoid homonyms	L
Database connection error	M	M	Make sure the data is stored correctly by testing the code and program.	L
Difficulty in learning Weka software	M	M	Methodically learn the core principles of using Weka	L

Table2. Risk analysis showing the potential risks to the project as a whole and the specific points in the development of the decision support

Appendix B – Data Description Table

Attribute	Data Type	Value Range	Homonyms	Replacement	Missing Values	Null Values	Importance
Age	Numeric	N/A	None	N/A	None	None	Keep
PR Bleeding	Categorical	20-96	None	N/A	None	None	Keep
Dark Red	Categorical	Yes|No	None	N/A	None	None	Keep
Bright Red	Categorical	Yes|No	None	N/A	None	None	Keep
On Motion	Categorical	Yes|No	None	N/A	None	None	Keep
On Toilet	Categorical	Yes|No	None	N/A	None	None	Keep
Mixed with Stool	Categorical	Yes|No	None	N/A	None	None	Keep
More than once in 6 weeks	Categorical	Yes|No	None	N/A	None	None	Keep
Frequency of Bleeding	Text	N/A	None	N/A	408	None	Delete
Mucous PR	Categorical	Yes|No	None	N/A	None	None	Keep
Frequency of Mucous	Text	N/A	None	N/A	430	None	Delete
Pus PR	Categorical	Yes|No	None	N/A	None	None	Keep
Frequency of Pus	Text	N/A	None	N/A	431	None	Delete
Alteration in Bowel Habit	Categorical	Yes|No	None	N/A	None	None	Keep
Frequency of B.O	Text	N/A	None	N/A	337	None	Delete
Change in 12 months	Categorical	Yes|No	None	N/A	None	None	Keep
Constipated	Categorical	Yes|No	None	N/A	None	None	Keep
Loose stool	Categorical	Yes|No	None	N/A	None	None	Keep
Diarrhoea	Categorical	Yes|No	None	N/A	None	None	Keep
Straining at defecation	Categorical	Yes|No	None	N/A	None	None	Keep
Complete evacuation	Categorical	Yes|No	1	Yes	None	None	Keep
Urgency	Categorical	Yes|No	None	N/A	None	None	Keep
Pain on defecation	Categorical	Yes|No	None	N/A	None	None	Keep
incontinence	Categorical	Yes|No	None	N/A	None	None	Keep
abdominal pain	Categorical	Yes|No	None	N/A	None	None	Keep
lethargy	Categorical	Yes|No	None	N/A	None	None	Keep
SOB on activities	Categorical	Yes|No	None	N/A	None	None	Keep
SOB on stairs	Categorical	Yes|No	None	N/A	None	None	Keep
Change in Wt	Categorical	Yes|No	None	N/A	None	None	Keep
Loss of Wt	Categorical	Yes|No	None	N/A	None	None	Keep
Loose clothing	Categorical	Yes|No	None	N/A	None	None	Keep
Inc Wt	Categorical	Yes|No	None	N/A	None	None	Keep
Appetite Inc	Categorical	Yes|No	None	N/A	None	None	Keep

Appetite Dec	Categorical	Yes|No	None	N/A	None	None	Keep
Aspirin	Categorical	Yes|No	None	N/A	None	None	Keep
Painkiller	Categorical	Yes|No	None	N/A	None	None	Keep
Polyps	Categorical	Yes|No	None	N/A	None	None	Keep
CA Colon	Categorical	Yes|No	None	N/A	None	None	Keep
Cancer elsewhere	Categorical	Yes|No	Nn	N/A	None	None	Keep
Family Polyp	Categorical	Yes|No	None	N/A	None	None	Keep
Family Ca Colon	Categorical	Yes|No	None	N/A	None	None	Keep
Family Ca Elsewhere	Categorical	Yes|No	None	N/A	None	None	Keep
Family - Who	Categorical	N/A	None	N/A	391	None	Delete
Relative Polyp	Categorical	Yes|No	None	N/A	None	None	Keep
Relative Ca Colon	Categorical	Yes|No	None	N/A	None	None	Keep
Relative Ca Elsewhere	Categorical	Yes|No	None	N/A	None	None	Keep
Relatives - Who	Text	N/A	None	N/A	408	None	Delete
Crohns/UC	Categorical	Yes|No	None	N/A	None	None	Keep
Family Hx IBD	Categorical	Yes|No	None	N/A	None	None	Keep
Smoker	Categorical	Yes|No	None	N/A	None	None	Keep
Ex Smoker	Categorical	Yes|No	None	N/A	None	None	Keep
Hb	Real	1.3-17.2	None	N/A	38	6	Keep
Diagnoso1	Categorical	N/A	Adenocarcinoma/adenocarcinoma	Adenocarcinoma	None	None	Specific Outcome
			Colitis/colitis	Colitis			
			D.D/D.D./DD	DD			
			Fissure/fissure	Fissure			
			Haemorrhoids/haemorrhoids	Haemorrhoids			
			Normal/normal	Normal			
			Polyp/ polyp/polyps/Polpy/polpy	Polyp			
			Polyp (hyperplastic)/Polpy (hyperplastic)/polyp (hyperplastic)	Polyp (hyperplastic)			
Bowel Cancer,Polyp (Yes)OR NOT (No)	Categorical	Yes|No	None	N/A	None	None	Outcome

Table3. Table showing the documented cleaning process of the individual attributes

Appendix C – Original Project Brief

Project Code: DND15
Title: Data Mining over Cancer Data
Specification:
Data mining allows general patterns to be retrieved from data sources such as data warehouses and databases. These patterns can then be used as an aid to making decisions. This project will make use of multiple Excel cancer databases supplied by clinicians at Castle Hill. A number of data mining applications will be used with this data to generate decision rules for patient risk that can be mapped back onto the Excel data. A decision interface will be built that allows a user to view the data and the risk assigned to any selected patient. While it is expected that weka is used to do the data mining, the final decision process could be implemented in Excel, VB or Prolog

Suitable Degree Programs:
Computer and Business Informatics
Computer Software Development
Computer Science
Computer Systems Engineering

System Environments and Hardware/Software requirements:
PC, weka, Excel, VB, Prolog

Ratings:

Research:	3
Analysis:	3
Design:	2
Implementation Volume:	2
Implementation Intensity:	2
Significant Element of Mathematical Work:	No

Appendix D – Revised Project Brief

Project Code: DND15
Title: Data Mining of Cancer Data & Decision Support
Specification:
Decision support tools allow for further analysis and decisions based upon mined data and associated rules. Data mining allows general patterns to be retrieved from data sources such as data warehouses and databases. These patterns can then be used as an aid to making decisions by supplying decision rules. This project will make use of multiple Excel cancer databases supplied by clinicians at Castle Hill. Weka will be used with this data to generate decision rules for patient risk that can be mapped back onto the Excel data. These rules will be used to determine the risk to any patient. The decision tool will enable multiple selections to be made based upon patient symptomology. The interface will also be built using Visual Basic, that allows the user to select a magnitude of symptoms and view the risk assigned to any selected patient. A backend database will also be implemented to view previous patient selections and decisions. While it is expected that weka is used to do the data mining, the final decision process will be implemented in Visual Basic, using Excel as the supporting data warehouse.

Suitable Degree Programs:
Computer and Business Informatics
Computer Software Development
Computer Science
Computer Systems Engineering

System Environments and Hardware/Software requirements:
PC, weka, Excel, VB

Ratings:

Research:	3
Analysis:	3
Design:	3
Implementation Volume:	2
Implementation Intensity:	2
Significant Element of Mathematical Work:	No

Appendix E – Activity Diagram

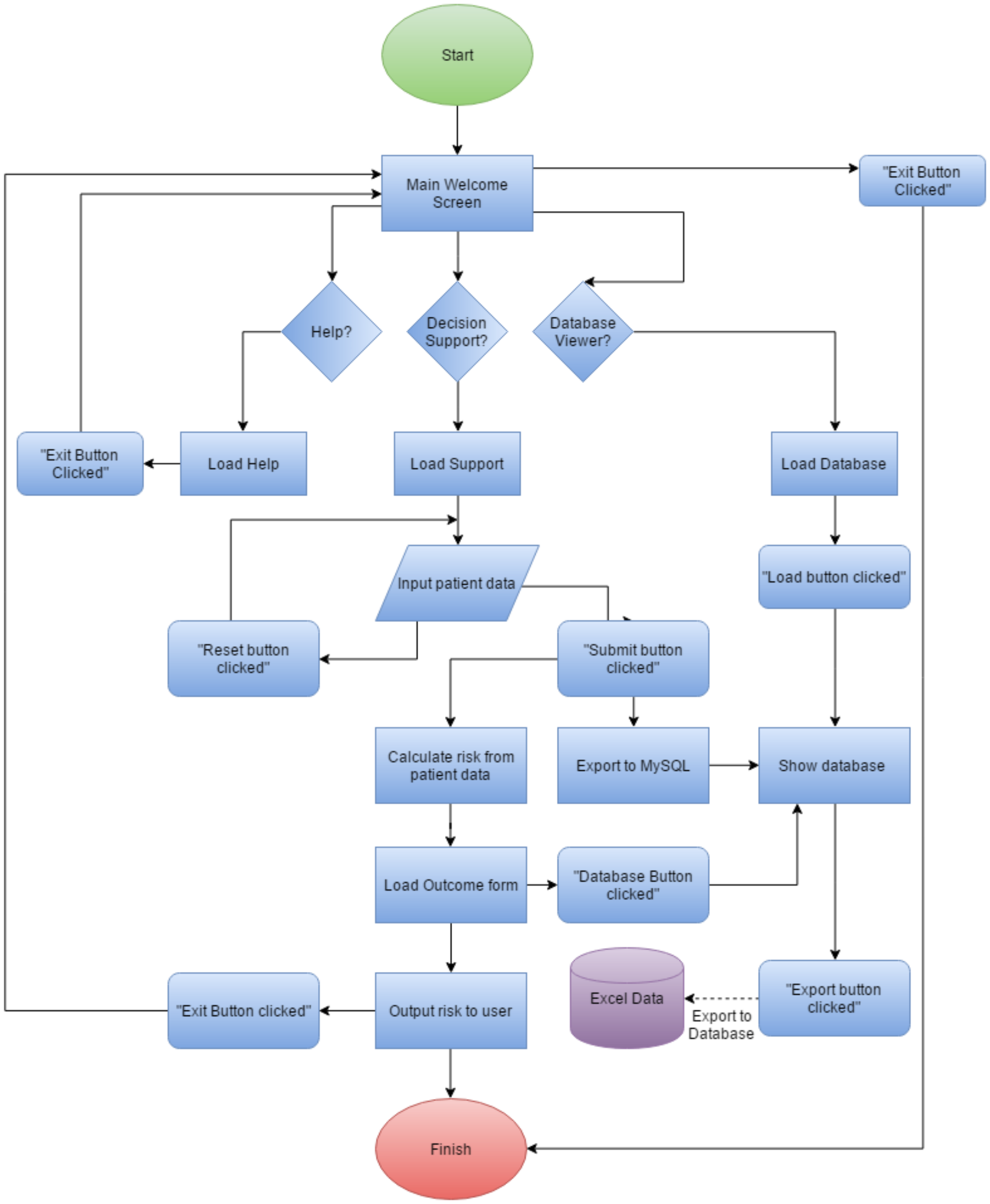


Figure32. Activity diagram showing the sequence of events from the difference elements in the software

Appendix F – Original Time Plan

Semester One

#	Task Name	Duration (Weeks)	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Xmas Wk1	Xmas Wk2	XmasWk3
		-	1/10	8/10	15/10	22/10	29/10	5/11	12/11	19/11	26/11	3/12	10/12	19/12	26/12	2/1	9/1
1	**Background Research**	Ongoing	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	Ongoing
2	**Initial Report**	2 weeks	🟦	End													
3	**Data Cleaning**	4 weeks	🟦	🟦	🟦	End											
4	**Choosing Classifier**	6 weeks					🟦	🟦	🟦	🟦	🟦	End					
5	**GUI Designs**	4 weeks	🟦	🟦	🟦	End											
6	**Software Designs**	5 weeks					🟦	🟦	🟦	🟦	End						
7	**Interim Report**	6 weeks										🟦	🟦	🟦	🟦	🟦	End
8	**Software Development**	Ongoing						🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	Ongoing
9	**Creating rule set**	3 weeks										🟦	🟦	End			
10	**Create database**	2 weeks													🟦	End	

Semester Two

#	Task Name	Duration (Weeks)	W16	W17	W18	W19	W20	W21	W22	W23	W24	W25	W26	W27	W28	W29	W30	W31
		-	16/1	23/1	30/1	6/2	13/2	20/2	27/2	5/3	12/3	19/3	26/3	2/4	9/4	16/4	23/4	30/4
1	**Background Research**	Ongoing	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	End				
2	**Final Report**	2 weeks	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	End			
3	**Hb Standard Deviation**	4 weeks					🟦	End										
4	**Over/Under-Sampling**	6 weeks					🟦	🟦	End									
5	**Final GUI Designs**	4 weeks									🟦	🟦	End					
6	**Final Software Designs**	5 weeks							🟦	🟦	🟦	🟦	🟦	End				
7	**Activity Diagram**	6 weeks										🟦	🟦	🟦	🟦	🟦	End	
8	**Software Development**	Ongoing	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	End	
10	**Software Testing**	2 weeks															🟦	End

Appendix G – Revised Time Plan

Semester One

#	Task Name	Duration (Weeks)	Wk1	Wk2	Wk3	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9	Wk10	Wk11	Wk12	Xmas Wk1	Xmas Wk2	XmasWk3
		-	1/10	8/10	15/10	22/10	29/10	5/11	12/11	19/11	26/11	3/12	10/12	19/12	26/12	2/1	9/1
1	Background Research	Ongoing	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	Ongoing
2	Initial Report	2 weeks	🟦	End													
3	Data Cleaning	4 weeks	🟦	🟦	🟦	End											
4	Choosing Classifier	6 weeks					🟦	🟦	🟦	🟦	🟦	End					
5	GUI Designs	4 weeks	🟦	🟦	🟦	End											
6	Software Designs	5 weeks					🟦	🟦	🟦	🟦	End						
7	Interim Report	6 weeks										🟦	🟦	🟦	🟦	🟦	End
8	Software Development	Ongoing						🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	Ongoing
9	Creating rule set	3 weeks										🟦	🟦	End			
10	Create database	2 weeks													🟦	End	

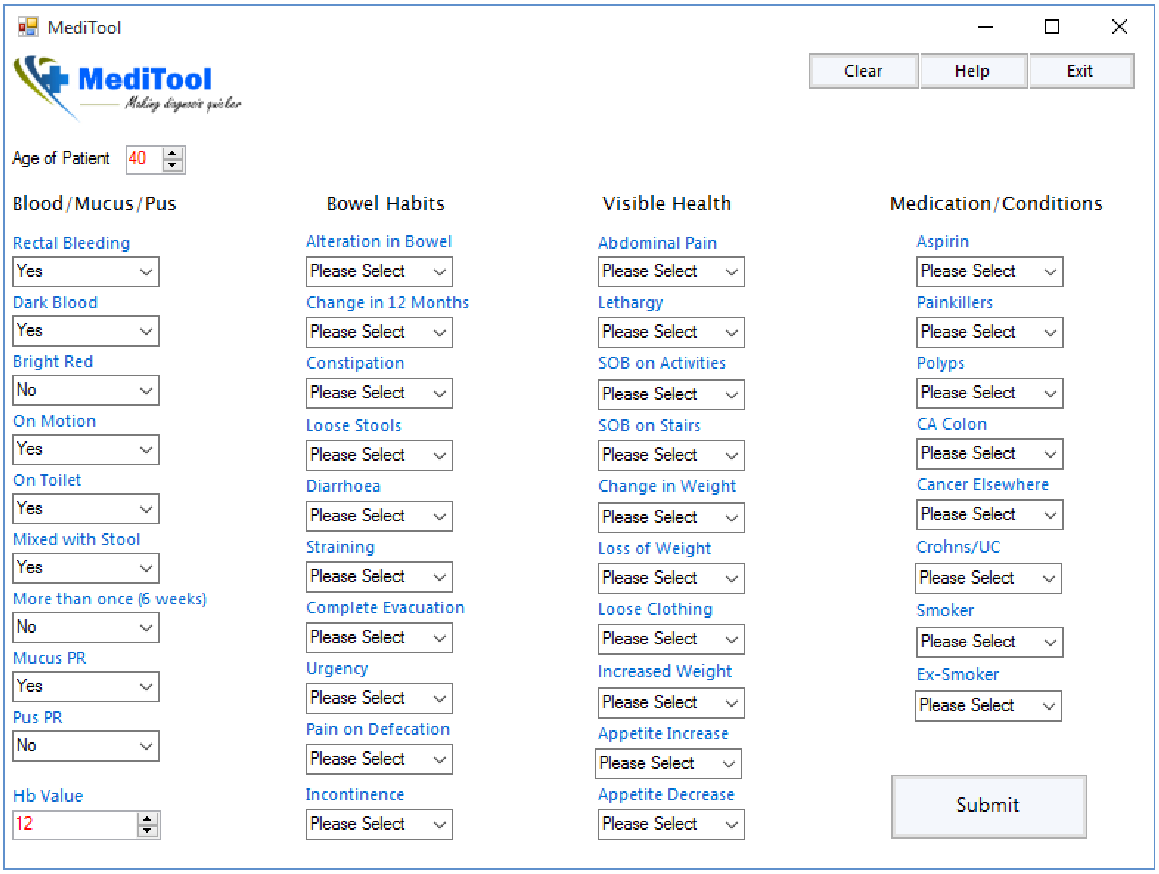
Semester Two

#	Task Name	Duration (Weeks)	W16	W17	W18	W19	W20	W21	W22	W23	W24	W25	W26	W27	W28	W29	W30	W31
		-	16/1	23/1	30/1	6/2	13/2	20/2	27/2	5/3	12/3	19/3	26/3	2/4	9/4	16/4	23/4	30/4
1	Background Research	Ongoing	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	End							
2	Final Report	2 weeks	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	End			
3	Hb Standard Deviation	4 weeks					🟦	🟦	End									
4	Over/Under-Sampling	6 weeks					🟦	🟦	🟦	End								
5	Final GUI Designs	4 weeks									🟦	End						
6	Final Software Designs	5 weeks							🟦	🟦	🟦	🟦	🟦	End				
7	Activity Diagram	6 weeks										🟦	🟦	🟦	🟦	🟦	End	
8	Software Development	Ongoing	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	End			
10	Software Testing	2 weeks															🟦	End

Appendix H – Decision Tree Output

```
J48 pruned tree
------------------
On Motion = No
|   Appetite Dec = No
|   |   Hb <= 12.6: No (373.0/57.0)
|   |   Hb > 12.6
|   |   |   SOB on stairs = No
|   |   |   |   Mixed with stool = No
|   |   |   |   |   Relative Ca Elsewhere = No
|   |   |   |   |   |   Loss of Wt = No
|   |   |   |   |   |   |   Pain on defectation = No
|   |   |   |   |   |   |   |   Complete evacuation = Yes: No (139.0/30.0)
|   |   |   |   |   |   |   |   Complete evacuation = No
|   |   |   |   |   |   |   |   |   PR Bleeding = No: No (28.0/4.0)
|   |   |   |   |   |   |   |   |   PR Bleeding = Yes
|   |   |   |   |   |   |   |   |   |   Family Ca Colon = No: Yes (14.0/4.0)
|   |   |   |   |   |   |   |   |   |   Family Ca Colon = Yes: No (2.0)
|   |   |   |   |   |   |   Pain on defectation = Yes
|   |   |   |   |   |   |   |   abdominal pain = No: Yes (4.0/1.0)
|   |   |   |   |   |   |   |   abdominal pain = Yes: No (9.0)
|   |   |   |   |   |   Loss of Wt = Yes
|   |   |   |   |   |   |   Ex Smoker = No: No (12.0/2.0)
|   |   |   |   |   |   |   Ex Smoker = Yes
|   |   |   |   |   |   |   |   abdominal pain = No
|   |   |   |   |   |   |   |   |   Hb <= 12.8: Yes (3.0)
|   |   |   |   |   |   |   |   |   Hb > 12.8: No (7.0/1.0)
|   |   |   |   |   |   |   |   abdominal pain = Yes: Yes (11.0/3.0)
|   |   |   |   |   Relative Ca Elsewhere = Yes: No (10.0/1.0)
|   |   |   |   Mixed with stool = Yes
|   |   |   |   |   Straining at defecation = No
|   |   |   |   |   |   Smoker = No: Yes (13.0/4.0)
|   |   |   |   |   |   Smoker = Yes: No (7.0/2.0)
|   |   |   |   |   Straining at defecation = Yes: No (3.0)
|   |   |   SOB on stairs = Yes
|   |   |   |   Diarrhoea = No
|   |   |   |   |   Complete evacuation = Yes: No (4.0/1.0)
|   |   |   |   |   Complete evacuation = No: Yes (5.0)
|   |   |   |   Diarrhoea = Yes: No (2.0)
|   Appetite Dec = Yes
|   |   Diarrhoea = No
|   |   |   Family Ca Elsewhere = No
|   |   |   |   Loose stool = Yes
|   |   |   |   |   Complete evacuation = Yes: No (11.0)
|   |   |   |   |   Complete evacuation = No
|   |   |   |   |   |   Smoker = No: Yes (6.0/1.0)
|   |   |   |   |   |   Smoker = Yes: No (4.0/1.0)
|   |   |   |   Loose stool = No
|   |   |   |   |   Bright Red = No: Yes (18.0/4.0)
|   |   |   |   |   Bright Red = Yes: No (6.0/1.0)
|   |   |   Family Ca Elsewhere = Yes: No (10.0/1.0)
|   |   Diarrhoea = Yes: No (39.0/7.0)
On Motion = Yes: No (90.0/12.0)

Number of Leaves  :     26
Size of the tree :      51

Time taken to build model: 0.84 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances        634               76.3855 %
Incorrectly Classified Instances      196               23.6145 %
Kappa statistic                         0.0169

Mean absolute error                     0.3345
Root mean squared error                 0.4349
Relative absolute error                99.5396 %
Root relative squared error           106.1702 %
Total Number of Instances             830
```

Appendix I – Initial Interface Design



Age of Patient 40

Blood/Mucus/Pus

Rectal Bleeding	Yes
Dark Blood	Yes
Bright Red	No
On Motion	Yes
On Toilet	Yes
Mixed with Stool	Yes
More than once (6 weeks)	No
Mucus PR	Yes
Pus PR	No
Hb Value	12

Bowel Habits

Alteration in Bowel	Please Select
Change in 12 Months	Please Select
Constipation	Please Select
Loose Stools	Please Select
Diarrhoea	Please Select
Straining	Please Select
Complete Evacuation	Please Select
Urgency	Please Select
Pain on Defecation	Please Select
Incontinence	Please Select

Visible Health

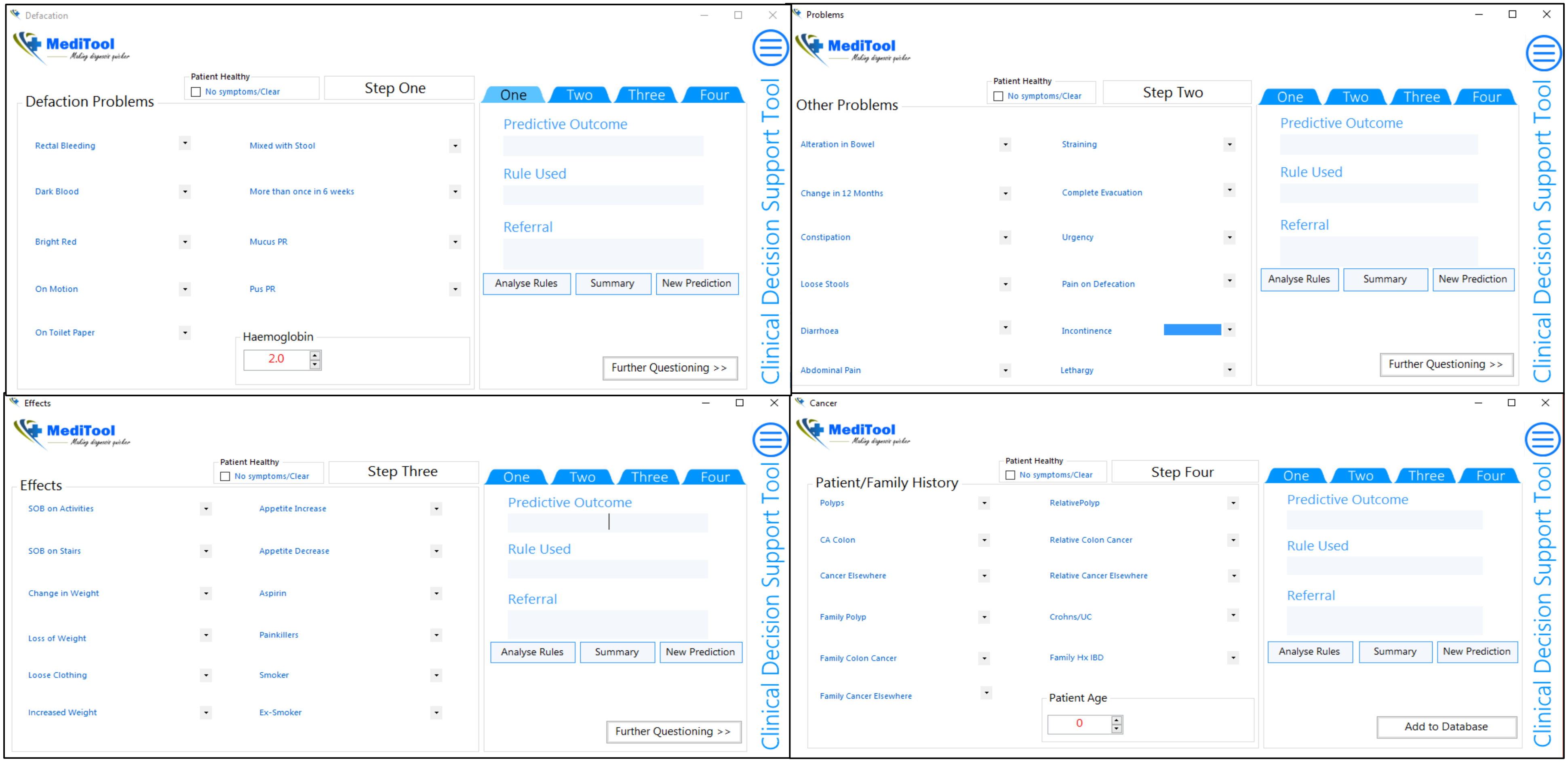
Abdominal Pain	Please Select
Lethargy	Please Select
SOB on Activities	Please Select
SOB on Stairs	Please Select
Change in Weight	Please Select
Loss of Weight	Please Select
Loose Clothing	Please Select
Increased Weight	Please Select
Appetite Increase	Please Select
Appetite Decrease	Please Select

Medication/Conditions

Aspirin	Please Select
Painkillers	Please Select
Polyps	Please Select
CA Colon	Please Select
Cancer Elsewhere	Please Select
Crohns/UC	Please Select
Smoker	Please Select
Ex-Smoker	Please Select

Figure33. Original software design showing one form with multiple decision boxes for the patient symptoms

Appendix J – Alternative Interface



Haemoglobin 2.0

Patient Age 0

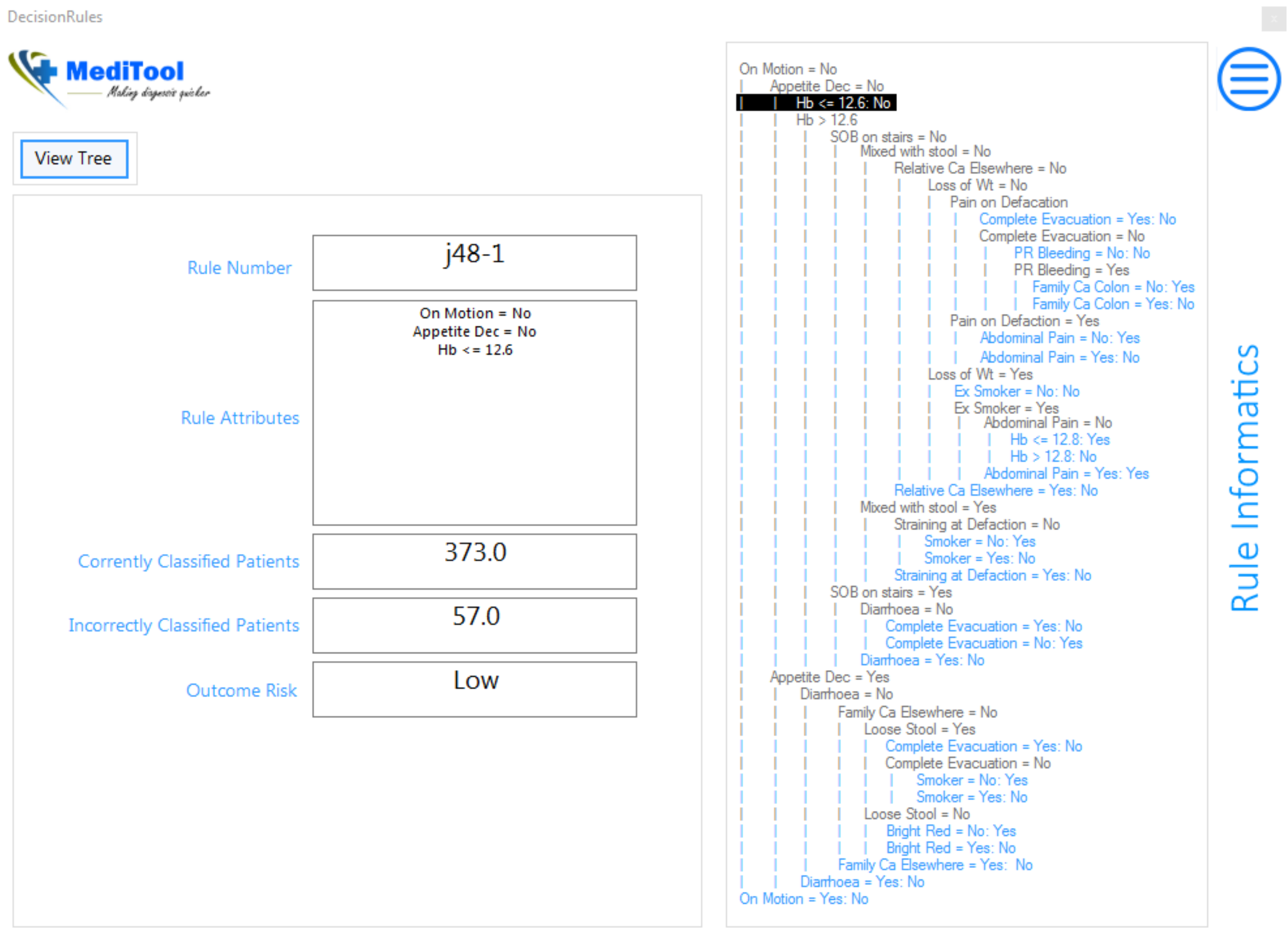
Figure34. Alternative software design showing multiple forms with fewer decision boxes for the patient symptoms for better clarity

Appendix K – Task List

#	Task Name	Description	Duration (days)
1	**Data Cleansing**	Cleaning the data to check for duplicate entries (homonyms)	2
2	**Background Research on Data mining**	Do background research on Data Mining and its core concepts	14
3	**Initial report**	Create the initial report outlining background research and project plan	14
4	**Software Development**	Develop the decision support tool with features outlined in the software designs	120
5	**Hb Standard Deviation**	Find the standard deviation value for both the ‘No’ and ‘Yes’ risks of cancer.	3
6	**Over/Under-Sampling**	Over and under sample the data to find the best classifier output	10
7	**Design an interface theme**	To design what the interface will look like for the user	14
8	**Final software designs**	Produce final software designs for the software development	3
9	**Activity Diagram**	Produce an activity diagram showing the flow of the built software	28
10	**Create Database**	Create a backend database, containing user selections	14
11	**Create ruleset**	Create a ruleset to create decisions on high or low risk patients	21
12	**Implement visual window**	Implement a visual window to show the data to the user	3
13	**Interim report**	Write the interim report deliverable	14
14	**Final Report**	To deliver the final report	120
15	**Software Testing**	Test the software for abnormalities	4

Table4. Table showing the individual tasks and their timescale for completion

Appendix L – Rule Information



Rule Number	j48-1
Rule Attributes	On Motion = No Appetite Dec = No Hb <= 12.6
Corrently Classified Patients	373.0
Incorrectly Classified Patients	57.0
Outcome Risk	Low

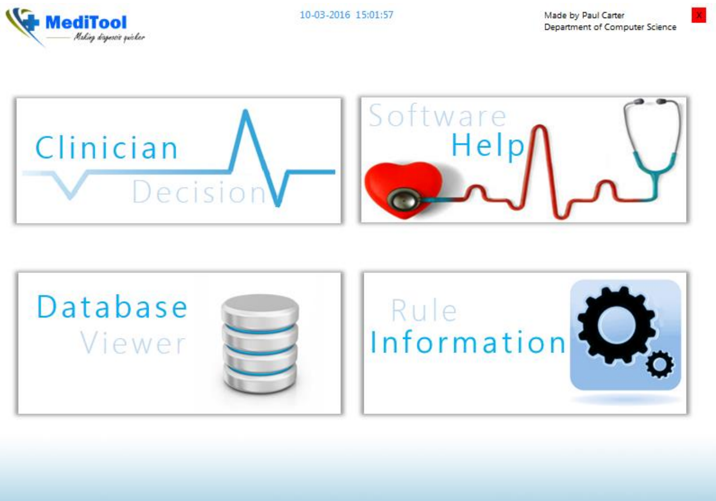
```
On Motion = No
|   Appetite Dec = No
|   |   Hb <= 12.6: No
|   |   Hb > 12.6
|   |   |   SOB on stairs = No
|   |   |   |   Mixed with stool = No
|   |   |   |   |   Relative Ca Elsewhere = No
|   |   |   |   |   |   Loss of Wt = No
|   |   |   |   |   |   |   Pain on Defacation
|   |   |   |   |   |   |   |   Complete Evacuation = Yes: No
|   |   |   |   |   |   |   |   Complete Evacuation = No
|   |   |   |   |   |   |   |   |   PR Bleeding = No: No
|   |   |   |   |   |   |   |   |   PR Bleeding = Yes
|   |   |   |   |   |   |   |   |   |   Family Ca Colon = No: Yes
|   |   |   |   |   |   |   |   |   |   Family Ca Colon = Yes: No
|   |   |   |   |   |   |   Pain on Defaction = Yes
|   |   |   |   |   |   |   |   Abdominal Pain = No: Yes
|   |   |   |   |   |   |   |   Abdominal Pain = Yes: No
|   |   |   |   |   |   Loss of Wt = Yes
|   |   |   |   |   |   |   Ex Smoker = No: No
|   |   |   |   |   |   |   Ex Smoker = Yes
|   |   |   |   |   |   |   |   Abdominal Pain = No
|   |   |   |   |   |   |   |   |   Hb <= 12.8: Yes
|   |   |   |   |   |   |   |   |   Hb > 12.8: No
|   |   |   |   |   |   |   |   Abdominal Pain = Yes: Yes
|   |   |   |   |   Relative Ca Elsewhere = Yes: No
|   |   |   |   Mixed with stool = Yes
|   |   |   |   |   Straining at Defaction = No
|   |   |   |   |   |   Smoker = No: Yes
|   |   |   |   |   |   Smoker = Yes: No
|   |   |   |   |   Straining at Defaction = Yes: No
|   |   |   SOB on stairs = Yes
|   |   |   |   Diarrhoea = No
|   |   |   |   |   Complete Evacuation = Yes: No
|   |   |   |   |   Complete Evacuation = No: Yes
|   |   |   |   Diarrhoea = Yes: No
|   Appetite Dec = Yes
|   |   Diarrhoea = No
|   |   |   Family Ca Elsewhere = No
|   |   |   |   Loose Stool = Yes
|   |   |   |   |   Complete Evacuation = Yes: No
|   |   |   |   |   Complete Evacuation = No
|   |   |   |   |   |   Smoker = No: Yes
|   |   |   |   |   |   Smoker = Yes: No
|   |   |   |   Loose Stool = No
|   |   |   |   |   Bright Red = No: Yes
|   |   |   |   |   Bright Red = Yes: No
|   |   |   Family Ca Elsewhere = Yes:  No
|   |   Diarrhoea = Yes: No
On Motion = Yes: No
```

Figure35. Rule Information form showing the specific rule information for ‘j48-1’ and associated tree

Appendix M – User Guide

Application Welcome Screen

Upon loading the application, a welcome screen will greet the user of the different features that can be used, e.g. the decision support tool, help screen, database viewer and rule information.



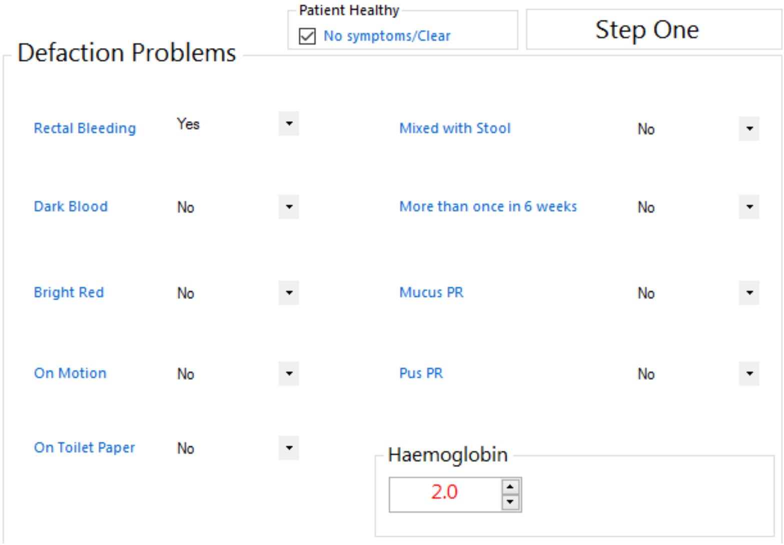
10-03-2016 15:01:57

Made by Paul Carter
Department of Computer Science

Figure36. Screenshot of the welcome screen showing the various options the user can take

Using the decision Support tool

Upon clicking the decision support tool link in the main menu, the tool will load onto the first stage of the support. There will be numerous categorized selections to make on each form.



Patient Healthy
☑ No symptoms/Clear

Defaction Problems

Rectal Bleeding	Yes	Mixed with Stool	No
Dark Blood	No	More than once in 6 weeks	No
Bright Red	No	Mucus PR	No
On Motion	No	Pus PR	No
On Toilet Paper	No		

Haemoglobin
2.0

Figure37. Screenshot of the decision support tool under the form category 'Defecation problems'

Submitting the applicable conditions

When the specific selections have been made, an 'analyse' button can be selected which work out a risk against the conditions which were selected.

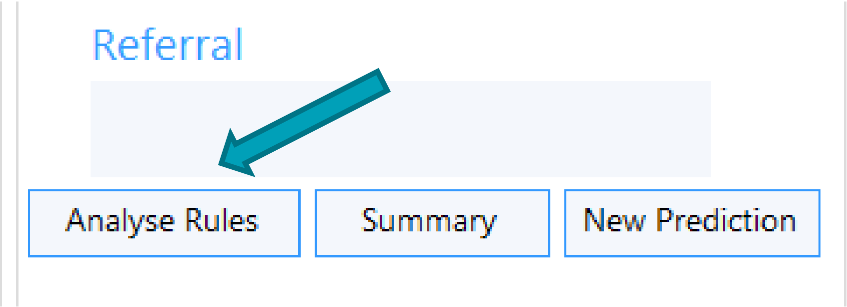
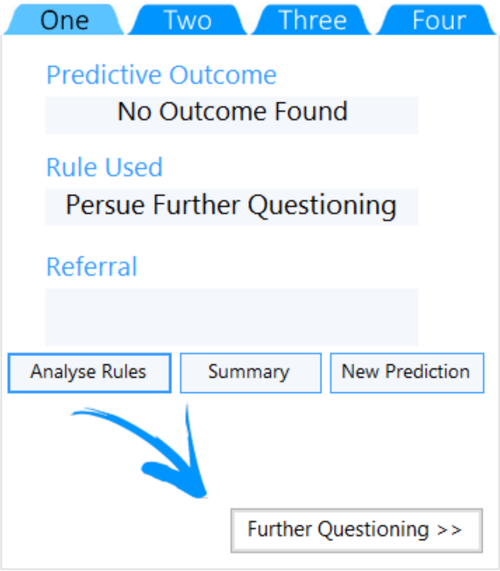


Figure38. Screenshot showing the 'Analyse Rules' button which will tell the user of a risk

The decision support tool output (No Assigned Risk)

If there is no assigned risk to the patient, the form will tell the user that no risk has been assigned. However, further questioning can be made on the next forms by clicking the 'Further Questioning' Button.



Predictive Outcome
No Outcome Found

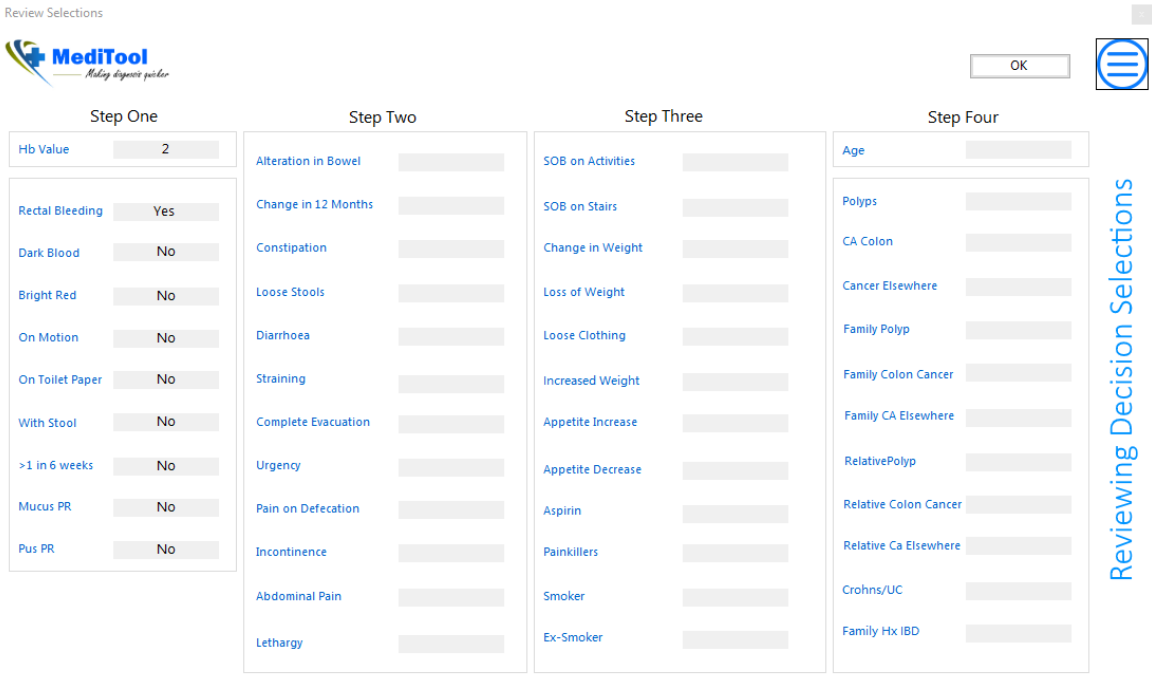
Rule Used
Persue Further Questioning

Referral

Figure39. Screenshot of further question, which will take the user to the next set of symptoms

Summary Form

The ‘Summary’ Button can also be clicked which will show the user which selections have been made. This allows the user to quickly check, without having to slide through each decision form.



Step One

Hb Value	2
Rectal Bleeding	Yes
Dark Blood	No
Bright Red	No
On Motion	No
On Toilet Paper	No
With Stool	No
>1 in 6 weeks	No
Mucus PR	No
Pus PR	No

Step Two

Alteration in Bowel	
Change in 12 Months	
Constipation	
Loose Stools	
Diarrhoea	
Straining	
Complete Evacuation	
Urgency	
Pain on Defecation	
Incontinence	
Abdominal Pain	
Lethargy	

Step Three

SOB on Activities	
SOB on Stairs	
Change in Weight	
Loss of Weight	
Loose Clothing	
Increased Weight	
Appetite Increase	
Appetite Decrease	
Aspirin	
Painkillers	
Smoker	
Ex-Smoker	

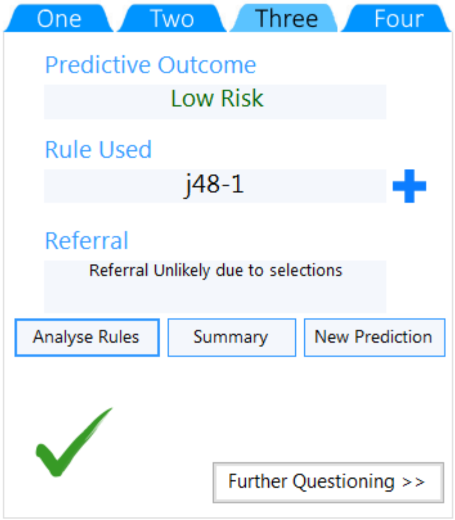
Step Four

Age	
Polyps	
CA Colon	
Cancer Elsewhere	
Family Polyp	
Family Colon Cancer	
Family CA Elsewhere	
RelativePolyp	
Relative Colon Cancer	
Relative Ca Elsewhere	
Crohns/UC	
Family Hx IBD	

Figure40. Screenshot showing the ‘review’ of the selections made in the decision support tool

The decision support tool output (Assigned Risk)

If there is an assigned risk to the patient, the form will tell the user what that risk is, as well as telling them what rule was used and if there needs to be a referral based on the given risk.



Predictive Outcome
Low Risk

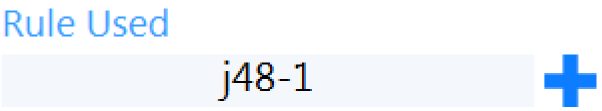
Rule Used
j48-1

Referral
Referral Unlikely due to selections

Figure41. Screenshot showing what happens when there is an assigned risk to a patient

Using the 'Rule Information' feature

When a particular risk is assigned the patient, a button appears which will take the user to a separate form. Clicking this button will open the form and will show the information for the rule used in the decision support tool.

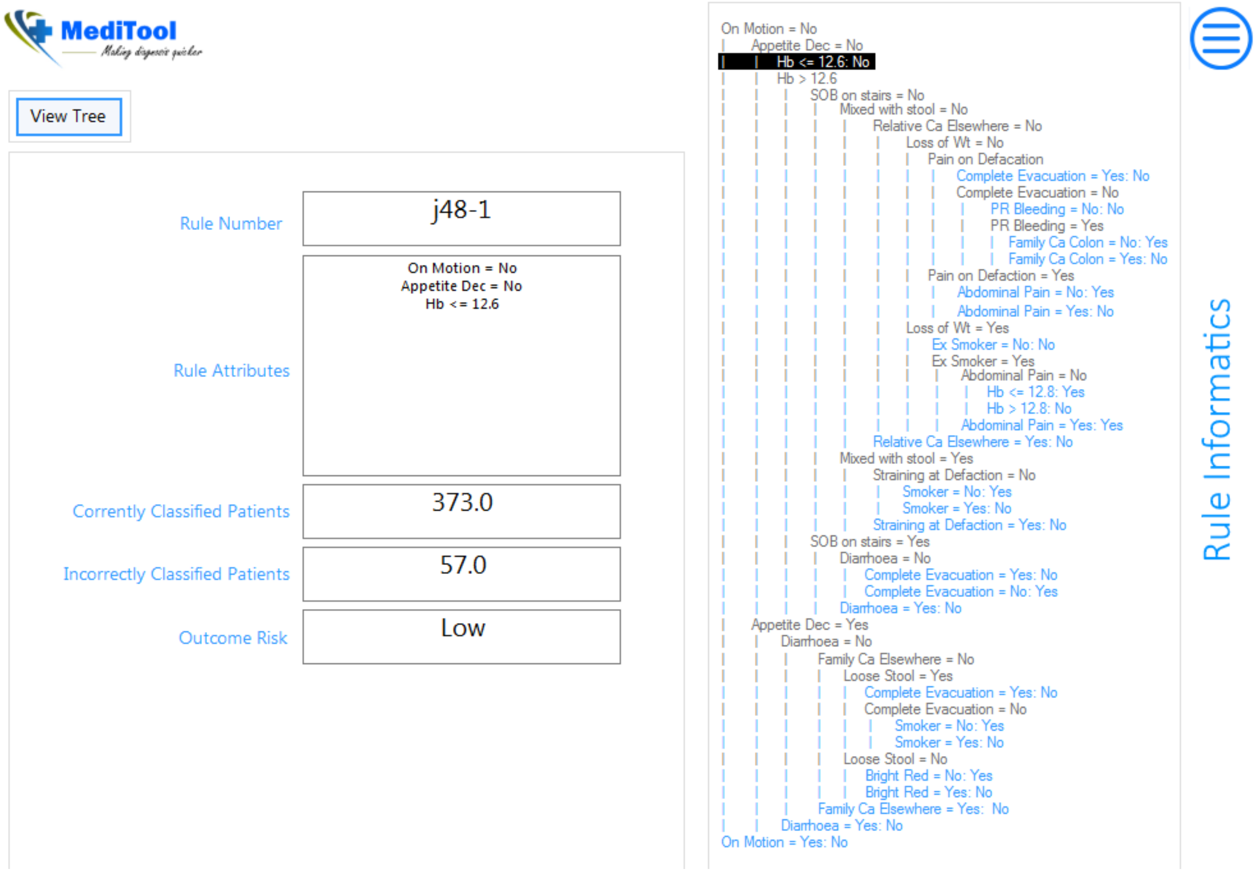


Rule Used

j48-1

Figure42. Screenshot showing the 'plus' button which can be selected when there is an assigned risk

When the button is clicked, this is what the form will look like based on the information for 'Rule1'. This shows the rule, the conditions used, the correctly/incorrectly classified patients and the outcome risk assigned to the rule.



Rule Informatics

Rule Number: j48-1

Rule Attributes:
On Motion = No
Appetite Dec = No
Hb <= 12.6

Corectly Classified Patients: 373.0

Incorrectly Classified Patients: 57.0

Outcome Risk: Low

```
On Motion = No
|   Appetite Dec = No
|   |   Hb <= 12.6: No
|   |   Hb > 12.6
|   |   |   SOB on stairs = No
|   |   |   |   Mixed with stool = No
|   |   |   |   |   Relative Ca Elsewhere = No
|   |   |   |   |   |   Loss of Wt = No
|   |   |   |   |   |   |   Pain on Defacation
|   |   |   |   |   |   |   |   Complete Evacuation = Yes: No
|   |   |   |   |   |   |   |   Complete Evacuation = No
|   |   |   |   |   |   |   |   |   PR Bleeding = No: No
|   |   |   |   |   |   |   |   |   PR Bleeding = Yes
|   |   |   |   |   |   |   |   |   |   Family Ca Colon = No: Yes
|   |   |   |   |   |   |   |   |   |   Family Ca Colon = Yes: No
|   |   |   |   |   |   |   Pain on Defaction = Yes
|   |   |   |   |   |   |   |   Abdominal Pain = No: Yes
|   |   |   |   |   |   |   |   Abdominal Pain = Yes: No
|   |   |   |   |   |   Loss of Wt = Yes
|   |   |   |   |   |   |   Ex Smoker = No: No
|   |   |   |   |   |   |   Ex Smoker = Yes
|   |   |   |   |   |   |   |   Abdominal Pain = No
|   |   |   |   |   |   |   |   |   Hb <= 12.8: Yes
|   |   |   |   |   |   |   |   |   Hb > 12.8: No
|   |   |   |   |   |   |   |   Abdominal Pain = Yes: Yes
|   |   |   |   |   Relative Ca Elsewhere = Yes: No
|   |   |   |   Mixed with stool = Yes
|   |   |   |   |   Straining at Defaction = No
|   |   |   |   |   |   Smoker = No: Yes
|   |   |   |   |   |   Smoker = Yes: No
|   |   |   |   |   Straining at Defaction = Yes: No
|   |   |   SOB on stairs = Yes
|   |   |   |   Diarrhoea = No
|   |   |   |   |   Complete Evacuation = Yes: No
|   |   |   |   |   Complete Evacuation = No: Yes
|   |   |   |   Diarrhoea = Yes: No
|   Appetite Dec = Yes
|   |   Diarrhoea = No
|   |   |   Family Ca Elsewhere = No
|   |   |   |   Loose Stool = Yes
|   |   |   |   |   Complete Evacuation = Yes: No
|   |   |   |   |   Complete Evacuation = No
|   |   |   |   |   |   Smoker = No: Yes
|   |   |   |   |   |   Smoker = Yes: No
|   |   |   |   Loose Stool = No
|   |   |   |   |   Bright Red = No: Yes
|   |   |   |   |   Bright Red = Yes: No
|   |   |   Family Ca Elsewhere = Yes:  No
|   |   Diarrhoea = Yes: No
On Motion = Yes: No
```

Figure43. Screenshot of the rule informatics form showing the rule information for any given rule

Adding all selections to the database

When all the user selections have been made, the fourth and final form in the decision support will contain a button called 'Add to Database'. When this is selected, all the selections are immediately added to the backend database for the user to look at a later data. Once the button is pressed, the database form will load.

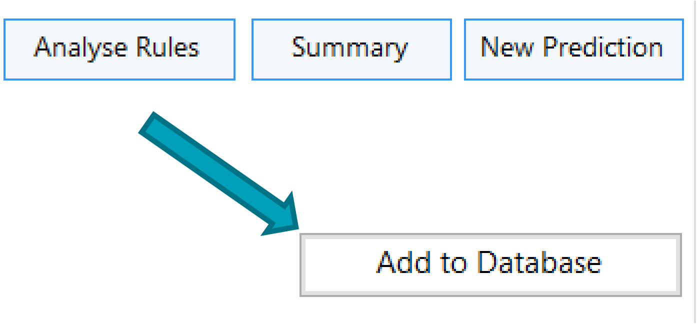


Figure44. Screenshot of the 'add to database' button which can be clicked to store selections

Using the database viewer form

Once the database has loaded, the 'Load Button' can be pressed which will populate the 'DataGridView' with previous patient symptoms.

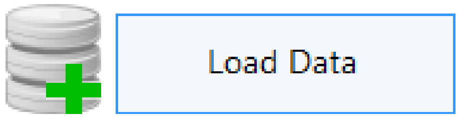
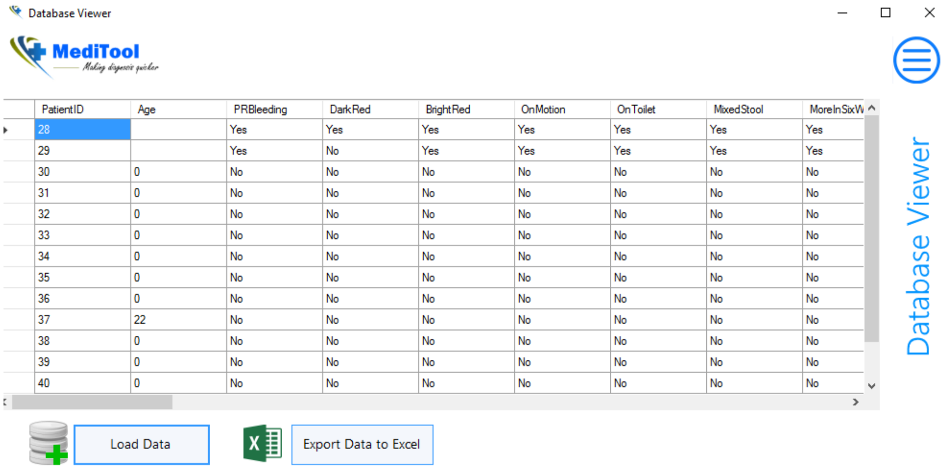


Figure45. Screenshot of the 'Load Data' button which can be clicked to bring up the database



PatientID	Age	PRBleeding	DarkRed	BrightRed	OnMotion	OnToilet	MixedStool	MoreInSixW
28		Yes	Yes	Yes	Yes	Yes	Yes	Yes
29		Yes	No	Yes	Yes	Yes	Yes	Yes
30	0	No	No	No	No	No	No	No
31	0	No	No	No	No	No	No	No
32	0	No	No	No	No	No	No	No
33	0	No	No	No	No	No	No	No
34	0	No	No	No	No	No	No	No
35	0	No	No	No	No	No	No	No
36	0	No	No	No	No	No	No	No
37	22	No	No	No	No	No	No	No
38	0	No	No	No	No	No	No	No
39	0	No	No	No	No	No	No	No
40	0	No	No	No	No	No	No	No

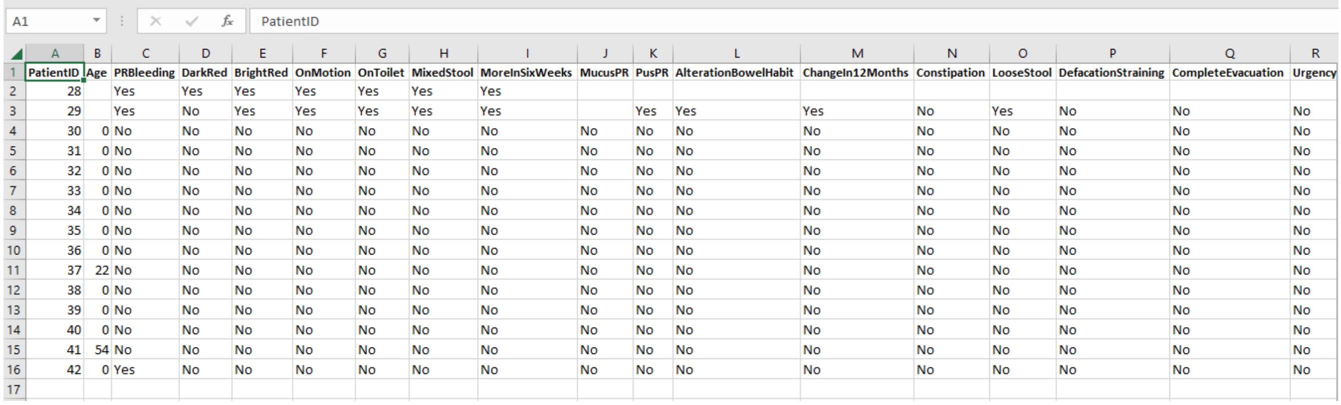
Figure46. Upon clicking the 'Load Data' button, this is what the database looks like

Exporting the data from the application to Excel

The user also has the ability to export all the data from the DataGridView to Excel by clicked the ‘Export Data to Excel’ button.

Figure47. As well as the ‘Load Data’ button, there is an ‘Export Data’ button to export data to Excel

Upon clicking this button, a new excel worksheet will load, populated with the data from the application.



A1 PatientID

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	PatientID	Age	PRBleeding	DarkRed	BrightRed	OnMotion	OnToilet	MixedStool	MoreInSixWeeks	MucusPR	PusPR	AlterationBowelHabit	ChangeIn12Months	Constipation	LooseStool	DefacationStraining	CompleteEvacuation	Urgency
2	28		Yes	Yes	Yes	Yes	Yes	Yes	Yes									
3	29		Yes	No	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	No	Yes	No	No	No
4	30	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
5	31	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
6	32	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
7	33	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
8	34	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
9	35	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
10	36	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
11	37	22	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
12	38	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
13	39	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
14	40	0	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
15	41	54	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
16	42	0	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
17																		

Figure48. Screenshot of exported data inside excel, which has been formatted via code in the DST

Using the ‘Software Help’ form.

Upon clicking the software help form from the main menu, the user will be greeted with a help screen showing the three buttons corresponding to the three other features in the application. Clicking each of these will show the help for that particular selected form.

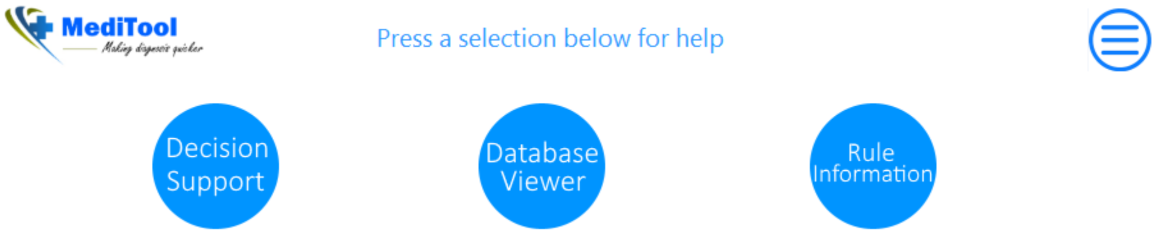
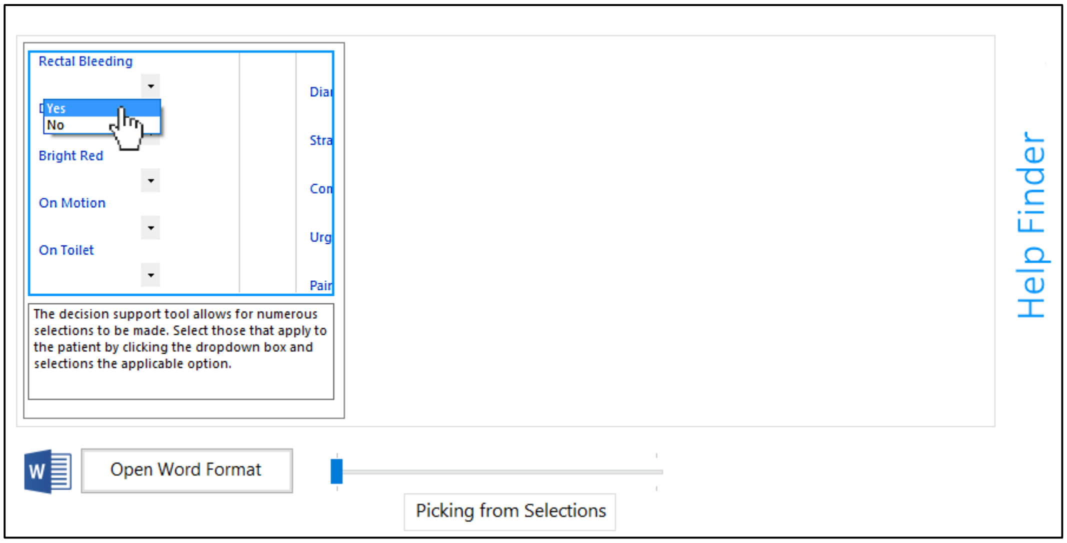


Figure49. Screenshot of the help menu showing the various options the user can take for help



The decision support tool allows for numerous selections to be made. Select those that apply to the patient by clicking the dropdown box and selections the applicable option.

Figure50. Screenshot of the help screen showing the scrollable help bar at the bottom

Using the slider feature at the bottom of the screen allows for easier use of finding help in using the application.

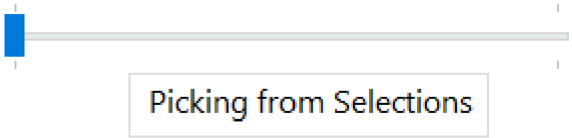
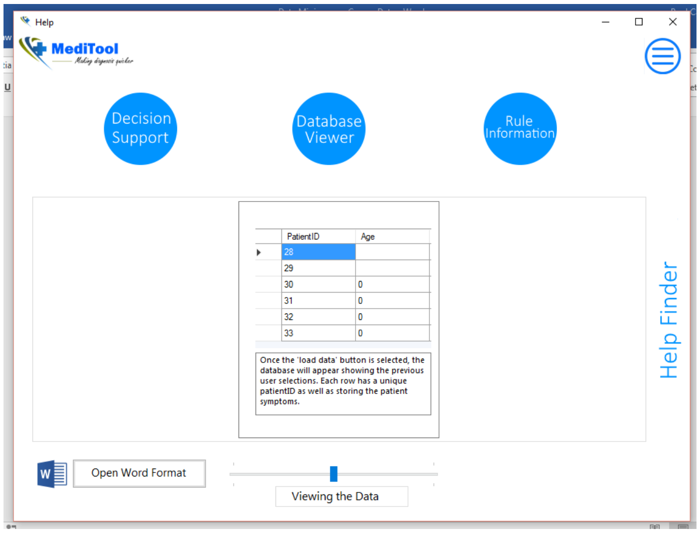


Figure51. Screenshot of the slider bar which can be slid across to look for more help in the DST

This is what the help form will look like when using the slider feature. Sliding across will allow the user to view the help information.



	PatientID	Age
▸	28	
	29	
	30	0
	31	0
	32	0
	33	0

Once the 'load data' button is selected, the database will appear showing the previous user selections. Each row has a unique patientID as well as storing the patient symptoms.

Figure52. Screenshot of the scrolling bar in action which slides through the various help screens

Exiting the Application

In each form there is a menu button which can be used to select other forms and to completely close the application, as well other options to get between the different forms.

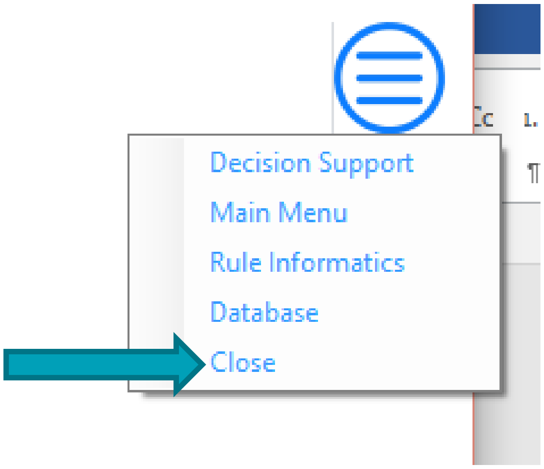


Figure53. Screenshot of the menu button which is available on each form, showing the various options including close