UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

UNIVERSITAT DE
BARCELONA

UNIVERSITAT
ROVIRA I VIRGILI

Master in Artificial Intelligence
Facultat d'informàtica de Barcelona (FIB)
Facultat de matemàtiques (UB)
Escola Tècnica superior d'enginyeria (URV)

Master Thesis

Artificial intelligence techniques to support cognitive rehabilitation

Sara Hoeksma

Advisors
Jesús Cerquides Bueno
Departament de Matemàtica Aplicada y Anàlisi (UB)
Josep Lluís Arcos
Research Scientist (IIIA)

June 2018

Acknowledgements

This dissertation originated in cooperation with the Institute for Artificial Intelligence Research (IIIA), an artificial intelligence research laboratory in Spain that belongs to the Spanish National Research Council (CSIC).

Special thanks to Josep Lluis Arcos for giving me the opportunity to cooperate on their project and guiding me through from beginning to end. Despite an unfortunate accident, I'd also like to thank Jesús Cerquides for being the bridge between the MAI and IIIA and introducing me to the project.

Abstract

In recent years, the Guttmann Institute has incorporated an intelligent assistant as a predicted and personalized decision support system (PPDSS). This PPDSS helps plan rehabilitation sessions for patients suffering from acquired brain injury (ABI). Results show questionable planning when comparing patient profiles and their assigned tasks. The distribution of percentage of effort does not perfectly match the distribution of the cognitive profile. This paper provides a thorough analysis of the patient profiles, showing that a patient's initial profile and the task execution scores during their first few sessions can be used to better predict their final improvement, to a certain degree of accuracy. Furthermore, results show that more executions of tasks does not automatically lead to improvement. Practice does not seem to make perfect. The proposed technique involves the incorporation of task-weights in the new scheduler.

Keywords: ***machine learning; acquired brain injury; decision support; classifiers; cognitive rehabilitation.***

Contents

List of Figures

1 | Introduction

1.1 Introduction

This research has been conducted under the supervision of the Institut d'Investigació en Intelligència Artificial (IIIA). Together with a number of partners, the IIIA is collaborating on the Innobrain project. The project works to apply artificial intelligence techniques to improve the Predicted and Personalized Decision Support Systems (PPDSS). This project is done in close cooperation with the Guttmann Institute.

The Guttmann Institute is a hospital for cognitive rehabilitation, and treats patients suffering from Acquired Brain Injury (ABI). ABI is one of the main causes of disability in the world, as a consequence, patients frequently suffer from a variety of impairments. A combination of physical and mental conditions can limit the daily abilities of a patient.

The neuroscientists at the Guttmann Institute have accumulated a dataset of patients suffering from ABI treated at their hospital. The dataset contains data of patients specifically suffering from Traumatic Brain Injury (TBI) and stroke, the two main causes of ABI. The patients have gone through a number of 'pre-evaluation' tests in order to determine and score their diagnoses. Moreover, the dataset includes the treatment plan, and treatment results. More specifically, the data provided includes the following:

- Patient data: demographic and clinical data. This includes a pre and post treatment analysis.
- Tasks: a list of available tasks and their respective suitability for certain cognitive functions.
- Executed schedule: rehabilitation sessions the patients have executed, including the tasks completed and the accompanying task execution score.

1.1.1 Patient Data

Figure 1.2 shows the demographic data available per patient in the left-most column. The middle column shows the 17 different pre-evaluation tests. Each of the tests result in a score between 0 and 4. Consequently, the test results translate into a pre-evaluation diagnosis, also scored between 0 and 4. These are referred to as (cognitive) impairments or cognitive functions. Where 0 indicates no impairment, and 4 is the most severe. The translation between test results and diagnosis is determined by the neuroscientists and done internally. Not all patients complete all

Demographic data		Pre-evaluation tests ([0,4])		Pre-evaluation diagnosis ([0,4])	
Gender	{'male', 'female'}	1	Digit span forward WAIS	Spec. 1	Categorization
Studies	{'no studies', 'primary', 'secondary', 'degree'}	2	Trail marking test, part A	Spec. 2	Divided attention
		3	Stroop word	Spec. 3	Flexibility
Age at injury	$[17, 76]$; $\upsilon = 40.6 \pm 14.5$	4	Stroop color	Spec. 4	Inhibition
Age treatment	$[17, 76]$; $\upsilon = 41.2 \pm 14.5$	5	Stroop word-color	Spec. 5	Planning
Delay treatment	$[0, 31]$; $\upsilon = 1.1 \pm 2.8$	6	Digit symbol WAIS	Spec. 6	Sequencing
Treatment weeks	$[1, 77]$; $\upsilon = 17.5 \pm 12.8$	7	Block design WAIS	Spec. 7	Selective attention
Sessions per week	$[1, 5]$; $\upsilon = 3.0 \pm 1.3$	8	Digit span backward WAIS	Spec. 8	Sustained attention
Etiology (specific)	{'TBI', 'multiple sclerosis', 'hemorrhagic stroke', 'ischemic-thrombotic stroke', 'ischemic-embolic stroke', 'ischemic undetermined stroke', 'other non-TBI', 'other'}	9	Letter-number sequencing WAIS	Spec. 9	Working memory
		10	RAVLT short-term memory	Spec. 10	Verbal memory
		11	RAVLT long-term memory	Spec. 11	Visual memory
		12	RAVLT recognition	Gen. 1	Attention
		13	Trail marking test, part B	Gen. 2	Executive functions
		14	WCST categories	Gen. 3	Memory
Etiology (general)	{'stroke', 'TBI', 'other'}	15	WCST perseverative errors		
		16	Stroop interference		
		17	PMR maximally produce words		

Figure 1.1: List of tests and impairments scored between [0,4] [1]

pre-evaluation tests. A patient only completes the tests which are assigned by the medical professional. If a patient has not been tested for certain impairments the patient does not receive a score for the untested impairment.

In addition to this pre-evaluation diagnosis, once the patient completes the treatment, a post-evaluation diagnosis is done. The results of the post-evaluation tests are once again translated to scores in the range [0,4] for the corresponding impairments.

Moreover, it is important to note that there are cognitive functions, and cognitive sub-functions. For instance, divided attention is a sub-function of attention. This hierarchical data regarding functions and sub-functions is available in the provided dataset, but for the purpose of this research each cognitive function is treated independently. The relation between functions is taken into consideration for the interpretation of the results.

1.1.2 Tasks

The dataset includes a number of neuro-rehabilitation tasks, each task has a suitability for certain cognitive impairments. The suitability is scored between 0 and 4, the higher the score, the more suitable it is for this particular cognitive function.

For example, task X is suitable for impairment A with a score of 4, and impairment B with a score of 1. This indicates that game X is almost entirely dedicated to improving impairment A, but it is also slightly helpful for impairment B. If a patient executes task X, it is assumed he/she is dedicating 80% ($\frac{4}{5}$) of his effort to impairment A, and 20% ($\frac{1}{5}$) to impairment B.

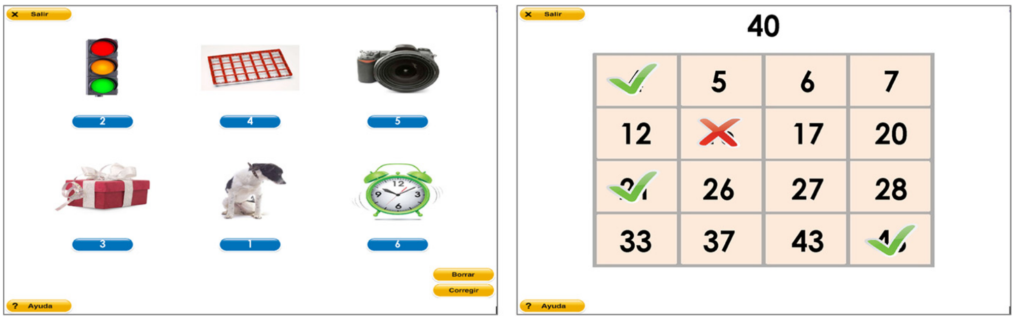

Figure 1.2: Neuro-rehabilitation tasks examples. The figure shows two examples of rehabilitation tasks used in GNPT, for treating working memory (left), and sustained attention (right) [2]

1.1.3 Executed schedule

The provided dataset details the schedule (tasks executed) per patient. The number of executed tasks per patient is variable. Each executed task per patient includes a task execution score between 0% and 100%. Moreover, each execution also includes the planned date and the execution date.

1.2 Problem Understanding

Currently, an 'intelligent assistant' is used in the cognitive rehabilitation planning of patients at the Guttmann Institute. Details on the current approach will be elaborated on at a later stage. However, it is already clear that there appears to be an imbalance between the patient profiles and the effort distribution of the tasks assigned to them.

From previous research by the IIIA regarding the implemented rehabilitation scheduler, an unbalance between efforts was discovered. It would be expected that the distribution of the severity of the impairments should be similar to the distribution effort put in during the rehabilitation process. Figure 1.3 shows that this is not the case. The left axis represents the severity of the cognitive function scored between [0,4], and the right hand side represents the effort put in per cognitive function. The effort was calculated using the task suitability provided. Most effort was put into a cognitive impairments with low scores, clearly something is amiss.

The primary motivation is that there is a lack of understanding from a professional standpoint. The doctors are offering a rehabilitation schedule set-up in a way that is difficult to comprehend. There is no available answer (yet) as to why the impairments and task efforts do not have the same, or similar, distribution. It is crucial to gain a better understanding of the current mechanisms of the scheduler, or to come up with an alternative.

Ultimately, the motivation of the work lies with being able to provide patients with the best and most efficient care possible. As well as being beneficial for the patient, the economic benefits are also a strong incentive as the doctors and rehabilitation process are costly.

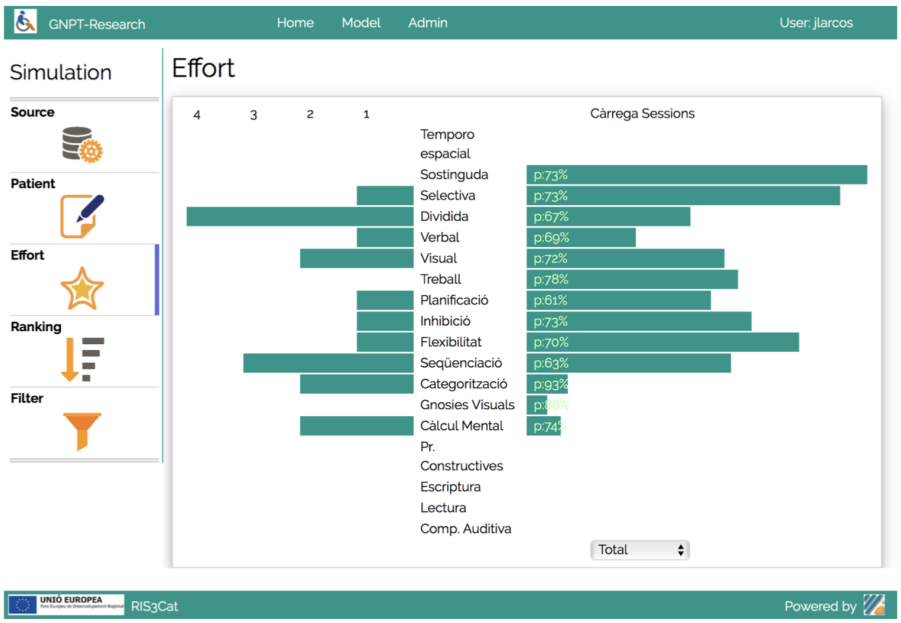

Figure 1.3: Impairment versus task efforts in current assistant [3]

1.3 Project goals

The ultimate goal would be to maximize the efficiency of the treatment program and the collaboration of the doctors and their patients. In order to get closer to this goal a number of sub-goals are defined:

- Better understanding of the dataset: do patients who have a similar impairment profile perform similarly?
- Is it possible to distinguish what patients that do not improve have in common?
- How can a better treatment plan be proposed?

The optimal outcome would be to propose a new decision support scheduler to replace or improve the current 'intelligent assistant'.

1.4 Project plan

The problem at hand will be treated as a data mining problem. The dataset contains many different types of data relevant to different parts of the rehabilitation process. The first step will be to identify the input features, which data to use in order to create an actionable dataset. Secondly a 'measure' or output has to be defined. Either of these are not set in stone, they are subject to adjustment according to the results.

The first phase will take the clinical data of each patient as the input features, using the overall improvement as a 'success' measure. Different machine learning algorithms will be applied, keeping in mind that visualization is a key aspect of the research. The analysis has to be understandable in for an interdisciplinary audience.

Rough outline of the project plan:

1. Use models to analyze the relationship between the initial cognitive profile and the overall improvement of a patient
2. Analyze results and possibly adjust input or output dataset
3. Deliver suggestions for improvements of the decision support scheduler

2 | CRISP-DM

CRoss **I**ndustry **Standard Process** for **D**ata **M**ining (CRISP-DM) is a neutral methodology for tackling data mining projects, independent of the industry and technology [5].

The CRISP-DM method distinguishes efficiently between generic and specialized models, thanks to which it is suitable for both experienced data miners and people with fewer skills and time limitations. This standardized process provides the structure and guidance necessary to approach almost any kind of data mining problem.

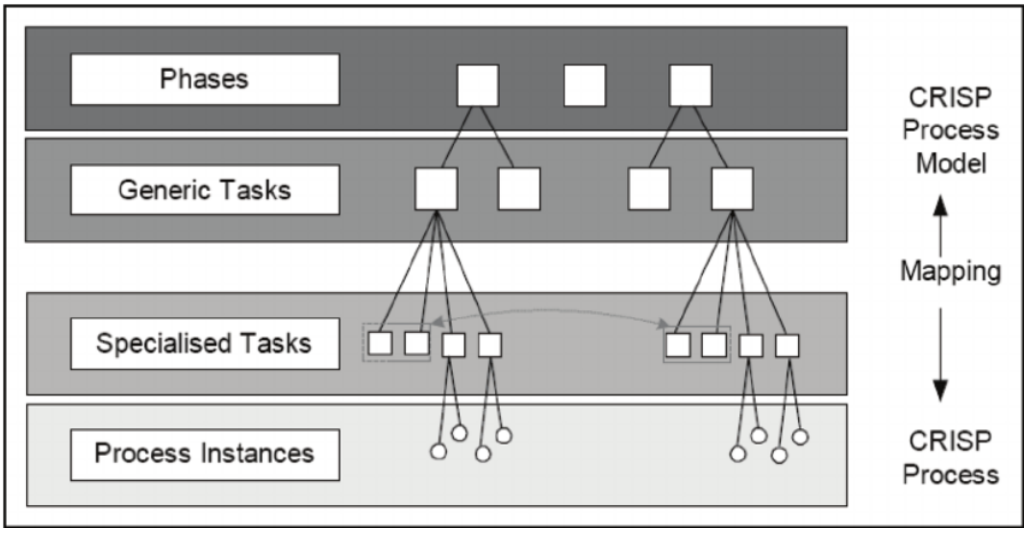

Figure 2.1: Hierarchical structure of CRISP methodology [4]

Figure 2.1 shows the four abstractions from the most general *Phases* to *Process Instances*. Each phase consists of a small number of generic tasks, these are meant to be generic enough to be applicable to all data mining problems. Consequently, the specialized tasks specify how the generic tasks should be executed in the specific problem at hand. The final level is a record of the process, representing the results and the path that was taken to get there.

This hierarchical structure serves as a useful model, however it is not a one way street. It is often necessary to jump up and down between the levels. This is more clearly demonstrated in the life-cycle of a data mining process a shown in figure 2.2.

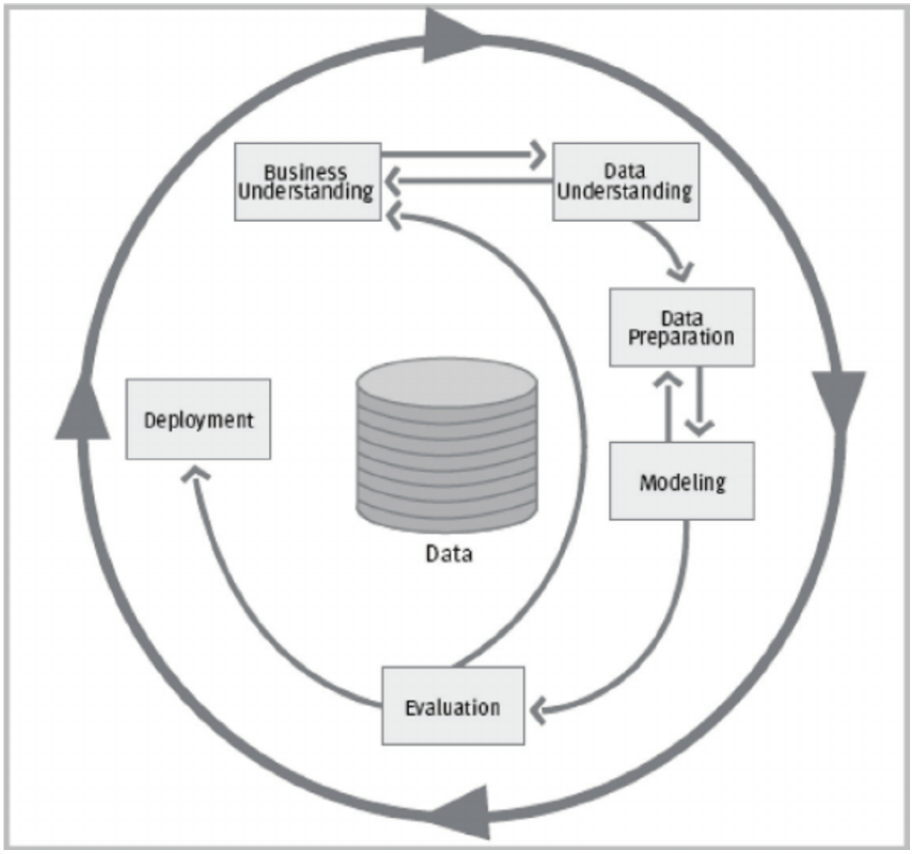

Figure 2.2: Life cycle of a data mining project [4]

The outer circle in figure 2.2 represents the continuous nature of a data mining project. Once a solution is reached, it does not mean the project has concluded; solutions and new discoveries can often lead to new ideas and different approaches to be taken. A key idea in the model is that the experience of going through each phase helps takes the next step, whether this is taking a step back to revise the previous phase, or moving onto the next phase. This is emphasized by arrows in both directions between phases such as *Data Preparation* and *Modeling*. Figure 2.2 merely shows the most common dependencies between the phases, whereas in reality these dependencies and their order will vary per project.

This research project will follow the six phases, as mentioned previously, this will not always happen in order.

Figure 2.3: Project flowchart

Each of these phases can consist of a number of guiding tasks shown in figure 2.4. Note that certain terminology is used interchangeably, for instance *Problem Understanding* and *Business Understanding* refer to the same phase.

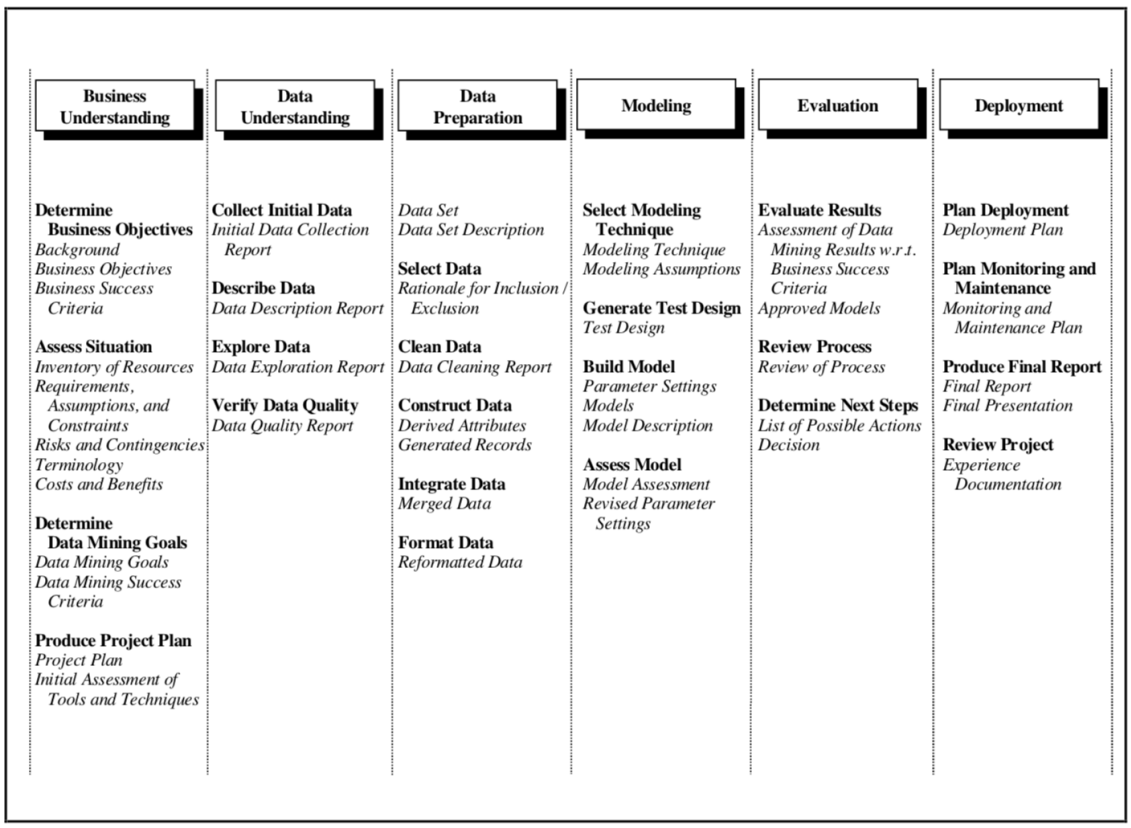

Figure 2.4: Overview of the CRISP-DM tasks and their outputs [5]

2.1 Problem Understanding

In the first phase, *Problem Understanding*, it is crucial to transform the question at hand into a data mining problem by analyzing the targets and requirements.

The problem at hand is in a medical domain. The first step will be to determine the objectives and success criteria from the perspective of the Guttmann Institute. Secondly, assessing the situation by creating an inventory of resources and the assumptions that are currently in place by creating an initial data description report.

In the simplest sense, data mining attempts to automate the process of extracting meaningful behavior, trends, and patterns from datasets, such that these can provide valuable insights. Therefore, the goal is to process the data in a way that can easily be repeated with a new batch of patients.

Using the aforementioned points, a preliminary project plan will serve to represent the first formulation of the task as a data mining problem.

2.2 Data Understanding

The first step to *Data Understanding* will consist of a preliminary statistical summary of the unaltered raw data delivered by the institute. These first statistics will provide some insights, after which a more elaborate analysis can be done to fully describe the data. This will already provide insights into which possible subsets could provide additional interesting information.

It is important to note these phases are not executed in isolation. Evidently the *Problem Understanding* already requires a certain level of *Data Understanding*, and so forth.

2.3 Data Preparation

The *Data Preparation* phase covers all steps taken to create the final dataset that is fed into the model. Most likely this process will be exploratory, and therefore repeated multiple times. Not only does the input dataset have to be constructed, but an output feature also has to be determined. What is the data trying to predict?

One of the key elements of this phase is the creation of new features to add to the dataset, as well as the reformatting of existing features.

It is crucial to provide a rationale for each decision to either include or exclude features.

2.4 Modeling

In this phase, a number of suitable modeling techniques are applied. This includes the adjustment of parameters and other alterations to optimize the model.

The experiments are performed in Python using a Jupyter notebook. Each of the models outlined below is available in the Scikit-learn python package [7]. This library provides a number of packages to easily run the chosen classification algorithms and adjust the variables accordingly.

It is to be expected that during the modeling phase new insights for constructing data occur which will lead to revisiting the previous phase.

2.4.1 K-means clustering

K-means clustering is a general-purpose method to partition an N-dimensional dataset into k subsets [8]. The algorithm initializes by picking k random data points and assigning them as cluster centers. Every other data point is assigned to its closest cluster center by means of a specified distance measure, most commonly Euclidean. Consequently, for each of the three subsets, the cluster center is updated to the mean of its corresponding group. This process is repeated until there is no more change in the assignment of data points to clusters.

K-means clustering can be done in an N-dimensional space, but this is hard to visualize. When combining K-means with principal component analysis (PCA) the input can be reduced to a two-dimensional space and visualized on a regular axis. K-means is a simple and well-working algorithm for basic problems. One of the main disadvantages is having to choose the value of k.

2.4.2 Principal Component Analysis

PCA is a common method in multivariate data analysis [9]. The objective is to reduce the data matrix by representing it by new orthogonal variables: the principal components. The idea is that these new variables represent the maximum variance of the original, inter-correlated data.

The main goals of PCA are to extract the most relevant information from the data, and reduce the size of the dataset by focusing on this 'most relevant information'.

2.4.3 Decision tree classification

To construct a tree, features in each node are selected from top to bottom by calculating the information gain of features, which reduces the entropy by separating instances. The branches represent specific feature observations of a data point and the leaves provide a class label. For new unlabeled instances, the prediction is made by tracing a path from the root to a final leaf node according to features properties of a new instance. The class of new instance is labeled when reaching the final leaf node.

A main advantage of decision trees is that they can be represented visually and concretely demonstrate the steps taken in the decision making process. This makes it easy for non-experts to interpret the results. It can be used for both descriptive purposes as well as predictive.

A regular top-down decision tree will be applied for this project. The first node, the root of the tree is determined by the feature that best splits the data, and consequently the feature in each node is calculated to be the 'best-split'. This implies that the higher up in the tree, the more important the feature. The calculation of 'best' can be done in a number of ways, in this project the Gini impurity will be used as a measure. The Gini impurity calculates the probability a randomly chosen data point would be classified incorrectly with respect to the data in the current subset. For instance, if a node only has data from one class, the Gini impurity will be 0. Mathematically formulated, for a set with J classes, where $i \in 1, 2, 3...J$, and p_i is the proportion of items with label i, the Gini impurity is calculated as follows.

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2$$

Advantages:

- Simple to visualize and understand
- Little data preparation is needed
- Similar to human-like decision making
- Performs well with large datasets

Disadvantages:

- Not the most accurate in comparison to other methods
- Not very robust to changes, small changes can have a big effect
- Can create overly complicated trees that do not generalize well
- Can create biased trees if classes are not balanced
- Achieving an optimal decision tree is an NP-complete problem

2.4.4 Support Vector Machines

Support vector machines (SVM) have proven to be one of the most successful algorithms with respect to the other well-known methods [10]. Some of its main advantages include the fact that SVM requires very little training samples and works consistently well regardless of increased dimensionality.

In the simplest SVM scenario: imagine a two-class classification dataset. An SVM finds a linear classification function, a hyperplane $f(x)$ that divides the two classes and maximizes the margin. The margin is the shortest distance between two data points of different classes, as defined by the hyperplane. Therefore, maximizing the margin maximizes the distance between the two classes; the SVM finds the best separating hyperplane. Once the function of the hyperplane $f(x)$ is found it can be used to classify new data points.

Consequently, SVM can be extended for non-linearly separable datasets using kernel transformations. One of the advantages of a linear SVM is that it returns the coefficients of the function defining the hyperplane, this serves as a strong indicator of feature-importance and is valuable for feature analysis. For this reason both linear and non-linear SVM will be applied. Non-linear SVM uses a non-linear kernel to find the maximum-margin hyperplane. Evidently it is expected that the linear SVM will perform worse as it is less flexible and suffers more from increased dimensionality.

2.4.5 Neural Networks

An overwhelming amount of research in neural classification has shown that neural networks are a favorable option in comparison to a number of classic classification methods [10]. The process in a neural network is often described as a 'black-box' due to the fact that they are self-adaptive. Neural networks are data driven models that independently modify themselves to best fit the data without any in-depth specification.

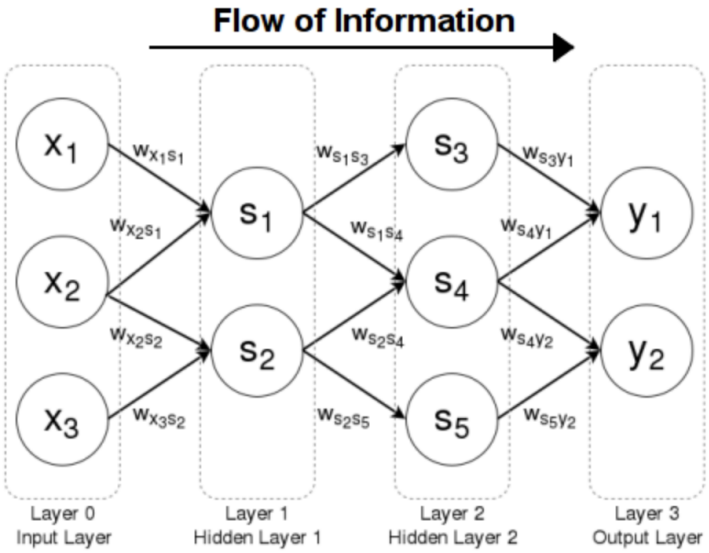

Figure 2.5: Neural network model [6]

A neural network takes one or multiple inputs, processes them, and returns one

or multiple outputs. The network itself consists of a number of small units belonging to a number of layers. Figure 2.5 shows a network with three input units, two hidden layers, and two output units. Each unit is connected to the units in the next layer through weighted connections. w_{x1s1} represents the weight from input unit x_1 in the input layer, to unit s_1 in hidden layer 1. This weight is a real number. The input values are multiplied by their respective weights, for instance $x_1 \times w_{x1s1}$. The result is put into an activation function which transforms the value into the value of the next unit, s_1. The process is repeated for each node as it travels forward in the network. The information is thus propagated through the network. The complexity of neural networks lies in the determination of the correct weights. There are many techniques to determine these weights, among which machine learning.

2.4.6 Gaussian Naive Bayes

The Naive Bayes classifier is based on Bayes' theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Informally, this equation calculates the 'after' probability given a known 'before' probability, or vice versa. The Naive Bayes classifier calculates the probability of every class given the input features, and chooses the outcome with the highest probability. The reason the classifier is 'naive' is due to the fact that it assumes independence between features.

Gaussian Naives Bayes applies when dealing with continuous data, and assumes that the continuous data per class is distributed according to a Gaussian distribution.

2.5 Evaluation

During the *Evaluation* phase, the process as a whole must be evaluated. There should be one or more models that satisfy the minimum criteria from a data-mining point of view. These models, as well as the steps taken to reach them, must be reviewed in order to determine whether it meets the data mining and company objectives set initially. The evaluation should determine whether there are any key points that have not been addressed sufficiently. Finally, the phase should conclude by highlighting the use of the achieved results.

2.6 Implementation

The implementation phase requires the results to be formatted in an 'easy-to-use' manner. Together with Guttmann Institute a protocol has to be defined for the use of the results.

The goal of this research is to provide useful information in order to improve the proposed schedule (the task-planner) for patient rehabilitation. Essentially: improving task allocation within a given time constraint. In order to do this, it is proposed to use an optimization library (Gurobi) in order to focus the effort in the formulation of the problem. The details of this are further explained in chapter 8.

2.7 Current situation at the Guttmann Institute

The current system has used a clustering algorithm to group patients with comparable features. The executed data mining and clustering algorithm makes use of the Expectation Maximization (EM) clustering technique [11]. 'In the end, this clustering process allows the system to group patients with similar characteristics, in order to automatically determine which rehabilitation tasks work better for each cognitive profile, taking into account all previous results and improvements done by similar patients in the past' [11].

An Intelligent Therapy Assistant (ITA) generates a schedule by selecting the most appropriate tasks for each patient. The ITA receives a patient profile and a number of sessions to be filled with tasks. The goal of the ITA is to fill in these tasks. It has been trained and tested during 18 months on 582 patients [2].

ITA forms part of the Guttmann Neuro Personal Trainer (GNPT), which is a tele-rehabilitation platform that has already been incorporated in the clinical routine since 2011 in multiple rehabilitation centres [11]. The ITA is the part of the GNPT that generates the schedule, by matching tasks to patients.

3 | Preliminary Analysis

First, a general analysis of the dataset will be conducted to gather some general statistics. This includes a preliminary analysis of the cognitive functions. These facts could be of help in further development of a PPDSS.

3.1 Summary statistics

- There are 28 cognitive functions
- There are 385 patients
- There are 139 tasks
- The average age at the time of the lesion is 48.74
- The average age at the time of treatment is 49.19
- The average time between lesion and treatment is 0.46 years
- There are 72.7 % Men
- There are 27.3 % Women
- The average education level is 3.76

The education-level ranges between 0 and 5.

3.2 Cognitive function statistics

In total there are 28 cognitive functions in the dataset. The cognitive functions have a number id, and a name. On average, 15.28 cognitive functions are tested per patient. Only 17 different cognitive functions are tested in this dataset.

The most frequent combinations of cognitive functions and scores were calculated:

Cognitive functions 87 and 86 resulting in a 0 score are the most frequently occurring, these are the temporospatial and orientation functions, respectively. Function

Impairment	Score	Frequency
87	0	291
86	0	291
84	0	197
85	0	197
15	4	187

Table 3.1: The 5 most frequently occurring combinations of impairment and score

87 is a sub-function of 86. 84 and 85, gnosia, related to perception and recognition, and it's sub-function visual gnosis follow, also with a score of zero. The first function to occur most frequently with a score greater than zero is function 15, categorization.

These cognitive functions are defined previously by those responsible for setting up the GNPT. They have done this by referring to the International Classification of Functioning, Disability and Health of the WHO [12].

Impairment	Result	Frequency
15	4	187
11	4	166
13	4	162
2	1	139
10	1	131

Table 3.2: The 5 most frequently occurring combinations of impairment and score, excluding score 0

The cognitive impairments which do not result in zero, from most frequent to least are: categorization, planning, flexibility, memory, and working-memory (a sub-function of memory). Thus we can see that based on our dataset, the most frequent high-scoring impairments are categorization, planning, and flexibility.

Another interesting combination to look at are the impairments which patients often suffer together. The impairments tested and resulting in zero have been filtered out for this comparison.

Impairment 1	Impairment 2	Frequency
11	3	349
2	8	327
3	2	325
3	8	325
13	15	323

Table 3.3: The 5 most frequently occurring combinations of two impairments in one patient, excluding those that score 0

The five most frequent combinations translated from function ID's to names, in order:

1. Planning & FFEE
2. Memory & Written-memory

3. FFEE & Memory
4. FFEE & Written-memory
5. Flexibility & Categorization

There appears to be a strong relationship between functions 2, 3, and 8 occurring simultaneously in a patient.

3.3 Data understanding

The provided data consists of four main dictionaries: the cognitive functions, the patients, the tasks, and the executed schedule of tasks per patient. The data indicates that not all cognitive functions are tested in a patient, nor are all tasks assigned. The key elements of the provided data which will be the focus of this research are: the patient profiles and the patient schedules. As mentioned previously, the patient profiles cover a total of 17 cognitive functions, meaning that the remaining 11 were not tested during the creation of this dataset and will therefore be ignored.

As part of the schedule, each executed task has a number of settings, including an assigned difficulty level. It is assumed that this difficulty is set to be at the correct level for the patient. A task should not be too hard because then the patient would get frustrated and would not learn, this is referred to as the infra-therapeutic range: when the score is below 65% [11]. Vice-versa, if the task is too easy the patient will not learn and might even get bored, this is the supra-therapeutic range: a score above 85% [11]. The optimal range is defined by a score in the therapeutic range: between 65% and 85%. The task settings are not incorporated in the dataset, but these are important to keep in mind.

Another important aspect to recall is that the schedule structure varies per patient. Meaning, each patient has completed a different number of sessions, with a varying number of tasks per session.

4 | Constructing the dataset

4.1 Input features

The previous chapter showed that there are 17 unique cognitive functions that are tested throughout the dataset for a total of 385 patients. The input dataset is based on a matrix of dimensions [385,17]. Each patient has 17 features, the value of each of these features is the score of the patient during the pre-evaluation. If the feature (cognitive function) was not tested, the score is assumed to be 0, no deficit.

4.2 Improvement

Each patient undergoes a post-assessment after they have completed their treatment. It is possible that a patient is tested for a certain cognitive impairment in their prior assessment, but not in their post-assessment. In order to be able to assess the improvement of a patient, only the impairments tested both prior and post the rehabilitation are taken into consideration. Each patient has a total improvement value, $Improvement_p$, calculated by summing up all the improvements for the n impairments tested in both pre and post evaluation.

$$Improvement_p = \sum_{i=1}^{n} Post_i - Pre_i$$

Evidently it is also possible that the improvement is negative if a patient worsens overall. The first approach taken is to investigate to what extend patient profiles can be an indicator of overall improvement. This is done by building classification models through different learning algorithms in order to classify the improvement. The $Improvement_p$ value is translated into three classes:

$$Improvement_p < 0 : Worsen$$

$$Improvement_p = 0 : Neutral$$

$$Improvement_p > 0 : Improve$$

5 | Patient improvement based on cognitive profile

5.1 K-means clustering

Every patient undergoes an assessment process prior to treatment. Based on this prior cognitive profile the patients are clustered into three clusters using k-means clustering. The idea is to gain insight into the question whether similar profiles also improve similarly. According to the current GNPT this should hypothetically be the case.

In figures 5.1 and 5.2 the patient's clinical data is taken as the range of input features. Using principal component analysis (PCA) the data is reduced to a two-dimensional feature space. Consequently, the data is clustered using k-means clustering, with $k = 3$. The white crosses mark the centroids of the clusters, the background colors mark the divisions of the clusters. In figure 5.1 the data points are colored according to a color-scale based on the improvement. The darker-red the data point, the more this patient has improved. In figure 5.2 the improvement has been divided into three classes, no improvement, positive improvement, and negative improvement (worsened). The data points in figure 5.2 are colored according to these three classes.

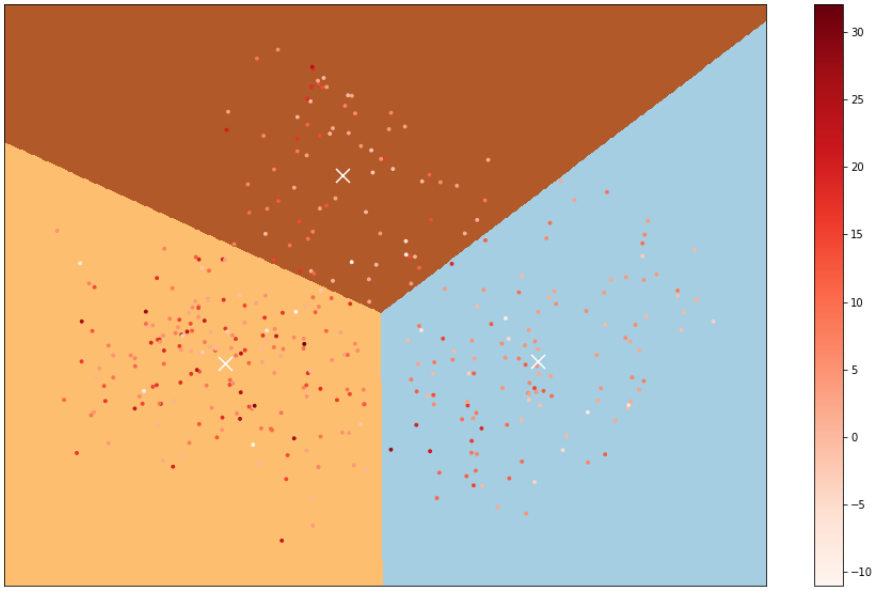

Figure 5.1: Patients clustered by prior assessment and colored according to improvement

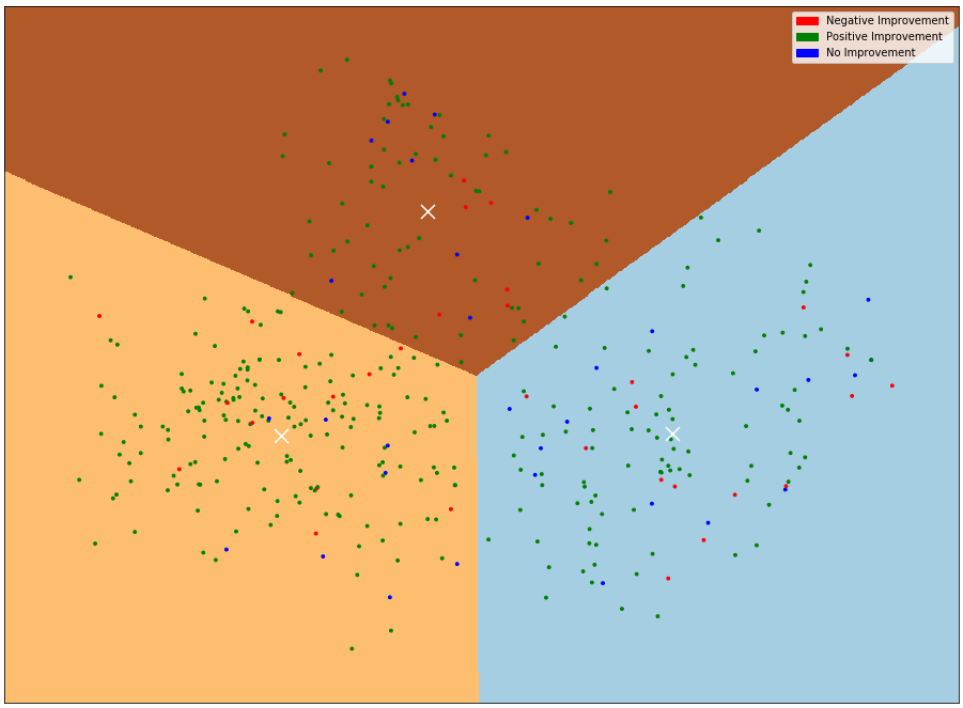

Figure 5.2: Patients clustered by prior assessment and colored according to improvement: either improve, worsen, or neutral

Though the three clusters do not show a clear distinction between them, there do seem to be some regions where patients are more likely to improve, and others where there are more negative or neutral improvements. As there does not appear to be any clear division of clusters, there is no further experimentation done with other values of k.

However, it is interesting to consider whether patients would be better grouped together if the data regarding the performance during the first three sessions would

be included. If a patient has gone through less than three sessions, then the total number of sessions completed is taken into consideration.

Logically speaking it would seem that already knowing whether a patient is performing well in the first few sessions should give an indication of whether he/she will improve overall. The input data is extended to a [385,20] dimensional matrix. The three extra features are calculated per patient using the scores of all tasks executed during the first three sessions. If a patient has not completed at least three sessions, the tasks executed in the total number of sessions is taken into consideration. The three features are the mean, standard deviation, and median of the scores in the first three sessions for each patient. This is once again reduced to two dimensions using PCA and clustered using k-means, $k = 3$.

Figures 5.3 and 5.4 show the adjusted input feature space clustered using k-means.

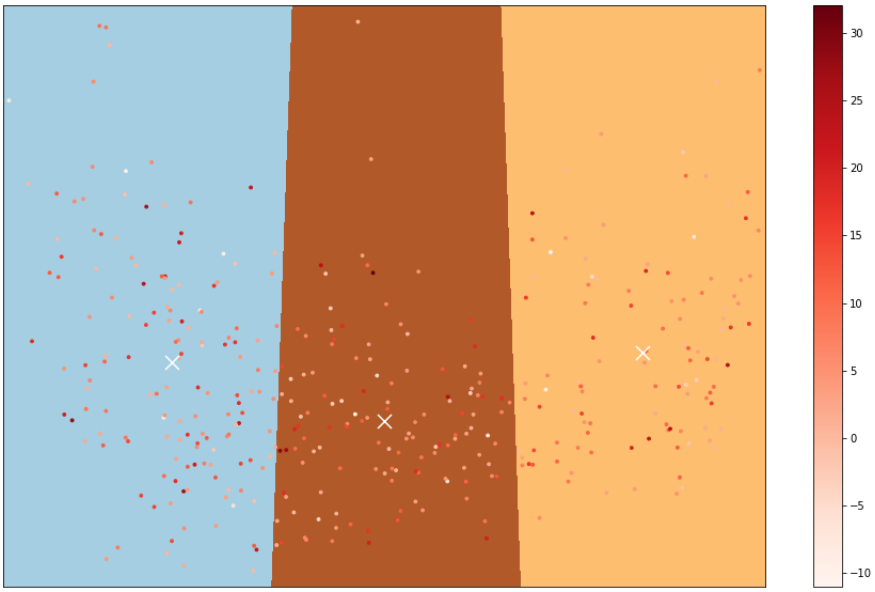

Figure 5.3: Clustered patients by prior assessment and score in the first three session, color scale by improvement

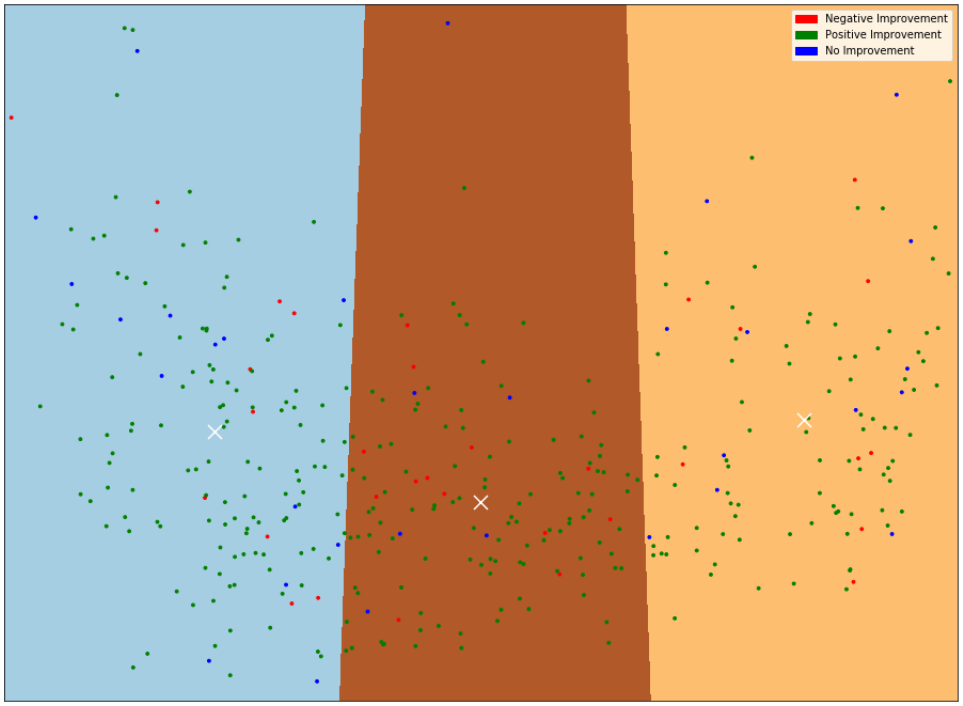

Figure 5.4: Patients clustered by prior assessment and colored according to improvement: either improve, worsen, or neutralt

Results show that the clusters are indeed formed differently, however it is still difficult to distinguish clear regions of similarity. There do appear to be smaller patches of similar patients that perform similarly.

5.2 Decision Tree Classification

One of the goals is to improve the insight the doctors have in the rehabilitation process they are recommending. A decision tree is a useful and visual tool that can help provide an interesting viewpoint. The same input data is taken as for the k-means clustering: an input feature space based on the patient's clinical data plus the mean, standard deviation, and median of the scores in the first three sessions. The goal is to classify between the three improvement classes described previously: *Improve*, *Worsen*, and *Neutral*.

As the majority of patients improve, the dataset is unbalanced. The classifier has been restricted to deal with a balanced weight class, such that each class is equally represented. In addition, there is a cut-off restriction of five values per leaf. Otherwise, almost each patient will result in its own leaf.

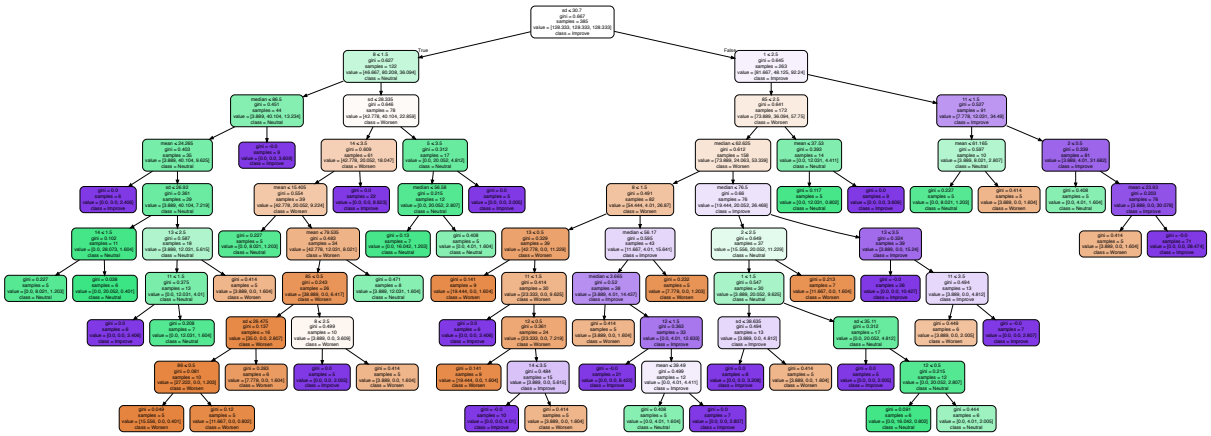

Figure 5.5: Decision Tree based on prior analysis and score first three sessions

Figure 5.5 shows the resulting decision tree, with the orange leafs classified as worsening, purple as improving, and green as neutral. The stronger the color the more 'certain' the classification label.

The tree shows that the data of the first three sessions is very important as they are present at the top roots of the tree. There are some interesting 'orange' (worsening) paths we can trace to possibly see which combination of impairments and performance result in worsening patients.

In addition to this, the decision tree can be used to train a model based on a training set. This was done for a training set based on two thirds of the data, with the remaining third used as the test set.

Where **Dataset** is the input features only including the clinical data (pre-evaluation results), and **Dataset+3** is the clinical data plus the mean, median and standard deviation of the scores in the first three sessions.

Train	Accuracy
Dataset	44.60%
Dataset+3	50.59%

Table 5.1: Decision tree model comparison of datasets, average of 100 runs

As is expected from Figure 5.5, the results from the first three sessions are very important. This additional information leads to an 11% increase in the prediction accuracy. Thus it would indicate that the performance of a patient during the first three sessions helps to predict whether there will be an overall improvement.

5.2.1 Adjusting the feature space

As the first three sessions appear to be an indicator of overall improvement, the feature space is now adjusted to include the clinical data, and the mean, median, and standard deviation of the first six sessions. If a patient has gone through less than

six sessions, then the total number of sessions completed is taken into consideration. This is done in order to test whether more performance information provides better results.

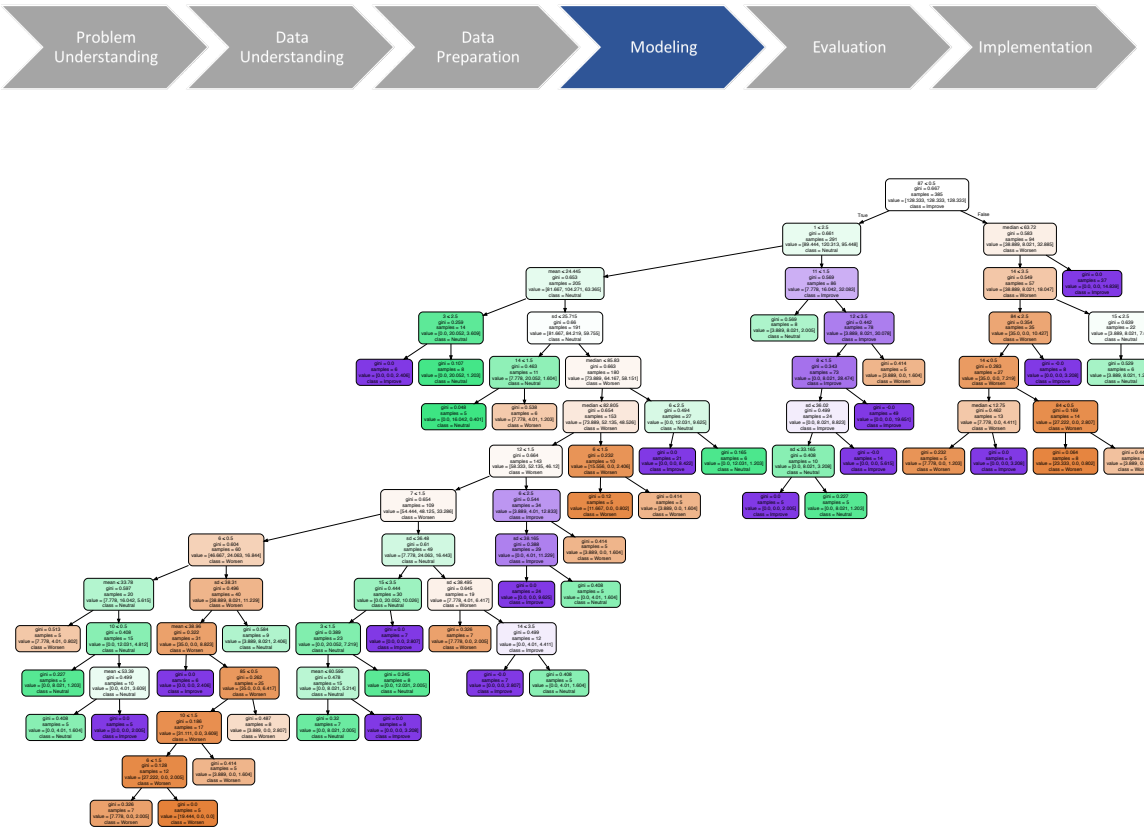

Figure 5.6: Decision Tree based on prior analysis and score first six sessions

Figure 5.6 shows impairment 87 at the root of the tree, this is the temperospatial cognitive function. Whereas in figure 5.5 the standard deviation of the scores in the first three sessions was at the root. This could possibly be due to the fact that after a few sessions, the doctors adjust the settings of the tasks. The adjustment of the task difficulty could be the cause of the scores of more than three sessions losing its predictive power.

Once again using a two-thirds to one-third split of the training and test set, respectively, the following accuracy results are obtained:

Train	Accuracy
Dataset	44.60%
Dataset+3	50.59%
Dataset+6	48.55%

Table 5.2: Decision tree model comparison of datasets. Average of 100 runs.

Where **Dataset+6** is the clinical data plus the mean, median and standard deviation of the scores in the first six sessions. The prediction accuracy of the latest dataset actually worsens. This corroborates the figure 5.6, showing that the score data is no longer at the root. Adding six sessions worsens the predictive accuracy in comparison to three sessions, but still better with respect to the initial **Dataset**.

5.2.2 Adjusting the classes

The worst-case scenario of a patient going through a rehabilitation process would be to actually get worse. It would be interesting to be able to identify possible indicators of worsening. For this reason the classes will be adjusted to *Worsen*, where the improvement is negative, and *Other*.

$$Improvement_p < 0 : Worsen$$

$$Improvement_p \geq 0 : Other$$

As the model presented in the previous section with six sessions did not prove to provide any additional value, the following model is based on the feature space including the clinical data, the mean, median, and standard deviation of only the first three sessions.

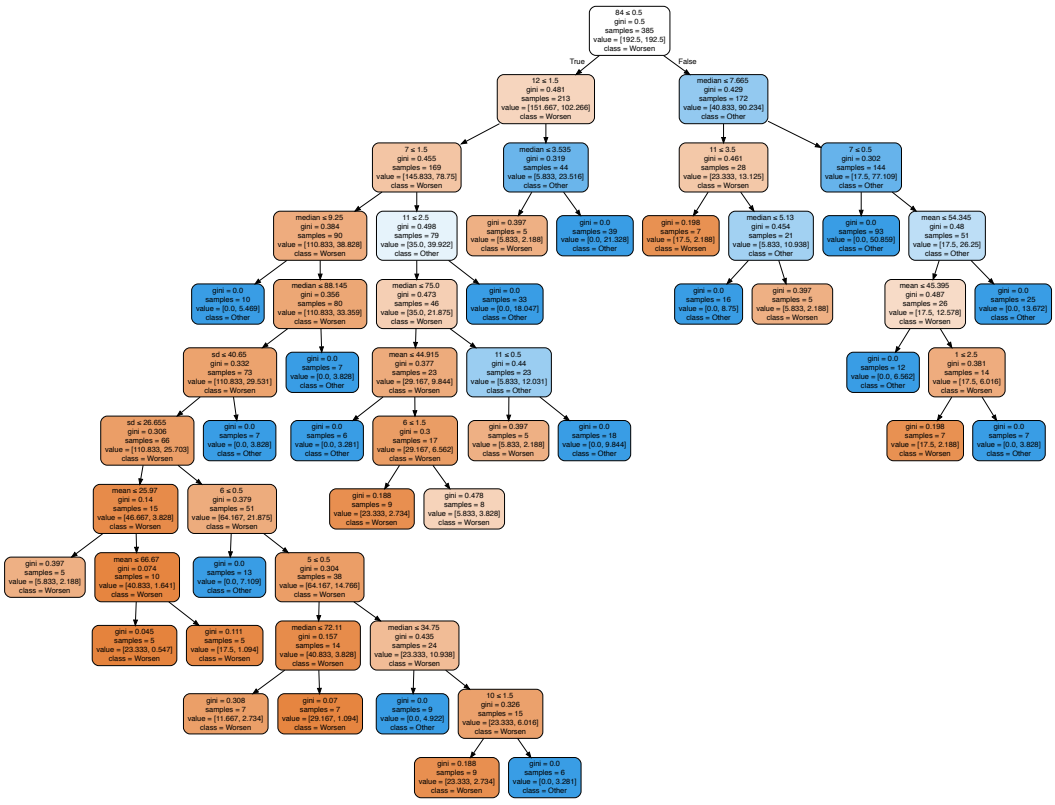

Figure 5.7: Decision Tree based on prior analysis and score first three sessions, 2 classes of improvement

Train	Accuracy
Dataset-WvsO	69.74%
Dataset+3-WvsO	72.25%

Table 5.3: Decision tree model predicting between 2 classes. Average of 100 runs.

The dataset **Dataset-WvsO** in table 5.3 represents all patients, who either worsen or not, with as input features only the clinical data. **Dataset-WvsO+3** includes the clinical data and the 3-session score related data.

The accuracies are significantly higher for this model. This could indicate that the distinction between the classes *Worsen* and *Other* is not as difficult. Perhaps the difference in accuracy between the model with three classes in table 5.1, and this model in table 5.3, is due to higher difficulty in distinguishing between the *Neutral* and *Improvement* classes. This leads to a new question: is it easier to distinguish between *Worsen* and *Other* than between *Neutral* and *Improvement*?

In order to test this theory a new dataset is created by taking a subset of the original; removing all the patients that worsen. The dataset contains 33 patients that worsen. This leaves a dataset of dimensions [352,20], 352 patients with their 17 cognitive function features and 3 score-related features. Evidently the classes are now limited to *Neutral* and *Improvement*.

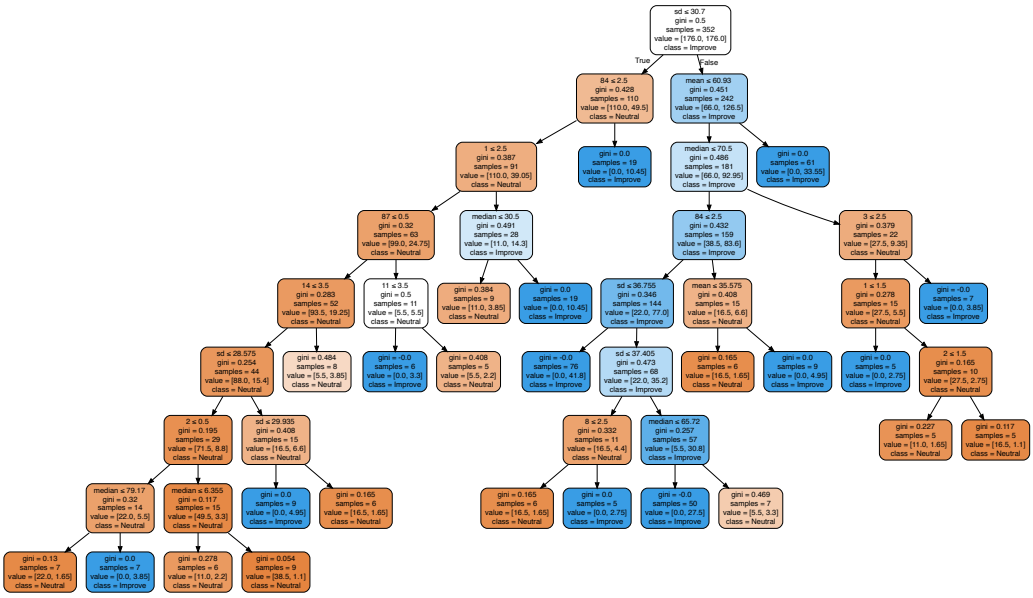

Figure 5.8: Decision Tree based on prior analysis and score first three sessions, 2 classes: Improve or Neutral

The dataset **Dataset-NvsI** in table 5.4 represents a subset of patients who either improve or stay neutral with as input features only the clinical data. **Dataset+3-NvsI** is this same subset in addition to the scores of the first three sessions.

Train	Accuracy
Dataset-NvsI	68.46%
Dataset+3-NvsI	72.15%

Table 5.4: Decision tree model predicting between 2 classes: *Improve* and *Neutral*. Average of 100 runs.

The results in tables 5.3 and 5.4 are quite similar. This does not support the theory that the differentiating between worsening and other patients is less difficult than differentiating between neutral and improving patients. The improved accuracy is most likely due to the simple fact that it is easier to classify correctly between two classes than between three classes.

5.3 Other classification algorithms

Decision trees build a helpful model as they provide a visually comprehensive result. For an analysis in the medical domain for which the results have to be interpreted by medical professionals, this is a good model to use. However, there are a number of other classification algorithms that could also be of use for modeling this problem. The Scikit-learn python package provides a number of packages to easily run the remaining classification algorithms [7]. The exact parameter settings for each can be found in the appendix.

The variations of datasets used are described below:

- **Dataset**: Input features include all patients and their clinical data. Classifying between *Improve*, *Neutral*, and *Worsen*.
- **Dataset+3**: Input features include all patients and their clinical data, in addition to the mean, median, and standard deviation of the scores of all executed tasks during first three sessions. Classifying between *Improve*, *Neutral*, and *Worsen*.
- **Dataset+6**: Input features include all patients and their clinical data, in addition to the mean, median, and standard deviation of the scores of all executed tasks during the first six sessions. Classifying between *Improve*, *Neutral*, and *Worsen*.
- **Dataset-WvsO**: Input features include all patients and their clinical data. Classifying between *Worsen* and *Other*.
- **Dataset+3-WvsO**: Input features include all patients and their clinical data, in addition to the mean, median, and standard deviation of the scores of all executed tasks during the first three sessions. Classifying between *Worsen* and *Other*.
- **Dataset-NvsI**: Input features include a subset of patients who do not worsen and their clinical data. Classifying between *Neutral* and *Improve*.

- **Dataset+3NvsI**: Input features include a subset of patients who do not worsen and their clinical data, in addition to the mean, median, and standard deviation of the scores of all executed tasks during the first three sessions. Classifying between *Neutral* and *Improve*.

	DT	SVM	SVM-Lin	NN	GNB	Average
Dataset	44,60%	**83,20%**	**83,03%**	79,81%	61,17%	70,36%
Dataset+3	50,59%	**82,81%**	73,48%	**82,34%**	61,78%	70,20%
Dataset+6	48,55%	**83,01%**	71,41%	**82,02%**	62,66%	69,53%
Dataset-WvsO	69,74%	**91,33%**	**91,14%**	**90,27%**	76,73%	83,84%
Dataset+3-WvsO	72,25%	**91,52%**	83,46%	**90,39%**	77,45%	83,01%
Dataset-NvsI	68,46%	**90,96%**	**91,33%**	87,22%	65,73%	80,74%
Dataset+3-NvsI	72,15%	**91,03%**	85,72%	**89,13%**	71,68%	81,94%

Table 5.5: Accuracy comparison of different classification algorithms on different datasets. Accuracies are averages of 100 runs.

Table 5.5 demonstrates that the conclusions drawn from the decision tree model are consistent with the other classification algorithms. As expected, the support vector machines and neural networks models perform better than the others. The most accurate models per dataset are shown in bold. The right-most column shows the average performance of each dataset.

It is interesting to note that for SVM and linear SVM in **Dataset+3** the accuracy actually goes down for adding the data regarding three sessions, whereas for decision trees, neural networks, and Gaussian Naive Bayes it does improve. However, for the other datasets the additional features of the performance in the first three sessions consistently provides a better accuracy (excluding the linear SVM).

Furthermore, the data shows that on average the models are able to distinguish better between the classes *Worsen* and *Other* than between *Neutral* and *Improve*.

Overall the results demonstrate that the data chosen is indeed of informative regarding the problem to be analyzed. Patient clinical profiles are able to predict improvement to a certain degree.

5.3.1 Linear SVM

The linear SVM returns feature weights for the hyperplane it creates as it constructs the model. The datasets classifying between two classes will be selected for comparison; more than 2 classes creates a more complex combination of hyperplanes that will not be analyzed at this stage.

The final weights are calculated as the average weights for each feature of the 100 runs corresponding to the experiment. Consequently, these final weights are squared in order to be able to rank the relevance of the attributes.

	D-WvsO	D+3-WvsO	D-NvsI	D+3-NvsI	**Average Rank**
1	0	0	5	5	**2,5**
2	6	10	7	2	**6,25**
3	12	7	10	10	**9,75**
6	3	1	14	12	**7,5**
7	1	2	16	15	**8,5**
8	15	16	6	1	**9,5**
10	13	5	9	11	**9,5**
11	7	3	0	0	**2,5**
12	2	11	4	7	**6**
13	14	6	15	9	**11**
14	11	13	8	8	**10**
15	10	4	1	6	**5,25**
84	5	9	11	13	**9,5**
85	4	8	12	14	**9,5**
86	8	14	2	4	**7**
87	9	15	3	3	**7,5**

Table 5.6: Feature ranking based on linear SVM weights per dataset

From each of the datasets shown in table 5.6, the top 10 ranking features are selected. The left most column shows all the features, cognitive function ID's, that are present in at least one of the top 10 rankings. The position of the ranking of each of the features per dataset is represented accordingly. For instance, cognitive function 1 has rank 0, first place, for both **Dataset-WvsO** and **Dataset+3-WvsO**. This would indicate that for these two datasets it is the most important feature according to the linear SVM. The right-most column represents the average ranking per feature. 16 out of the 17 cognitive functions appear in the top 10 at least once, whereas the mean, median and standard deviation of the first three session scores never appear in the top 10. Note that 0 is the highest possible ranking.

For each feature that ranks within the top five in any of the datasets, the average value per improvement cluster is calculated. Table 5.6 shows the ranks within top 5 in red. The results are shown in table 5.7.

	Dataset+3			Dataset+3WvsO		Dataset+3NvsI	
Feature	W	N	I	W	O	N	I
1	1,55	1,41	2,00	1,55	1,95	1,41	2,00
2	1,48	1,28	1,63	1,48	1,59	1,28	1,63
6	1,76	1,34	1,74	1,76	1,70	1,34	1,74
7	0,73	0,84	1,02	0,73	1,00	0,84	1,02
8	1,85	1,50	1,94	1,85	1,90	1,50	1,94
11	2,45	2,56	2,90	2,45	2,87	2,56	2,90
12	0,55	0,66	0,99	0,55	0,96	0,66	0,99
15	2,30	2,09	2,81	2,30	2,75	2,09	2,81
85	0,33	0,69	0,86	0,33	0,84	0,69	0,86
86	0,33	0,09	0,33	0,33	0,31	0,09	0,33
87	0,33	0,09	0,33	0,33	0,31	0,09	0,33

Table 5.7: Average cluster values per feature

Table 5.7 shows the average values per cognitive ID, per dataset, per class. Note that all columns corresponding to the same class are identical.

For instance, all worsening patients have an average score of 1.55 for cognitive function 1.

Interestingly enough, the average value for worsening patients is not necessarily higher than for improving patients. This discards the idea that patients who worsen are the more severely impaired. For all but one, cognitive function ID 6, the improving patients score worse than the worsening patients. Perhaps this indicates that patients who are worse-off at first have more room for improvement, therefore they generally do improve more than the less-severe patients.

6 | Patient improvement based on assigned tasks

An analysis parallel to the previous chapter will be explored, and the input features will be replaced by the executed tasks. This will allow the exploration of the idea whether the execution of certain tasks could be indicative of improvement. If this were the case, then it would seem that the tasks leading to improvement should be assigned with a higher priority.

6.1 Task distribution analysis

As stated previously, there are 139 tasks described in the dataset. It is possible that some of these tasks are never assigned, and it would be counterproductive to make our feature space unnecessarily large. The frequency histograms in figures 6.1 and 6.2 show the number of executions per task overall, and the number of patients that executed a task, respectively.

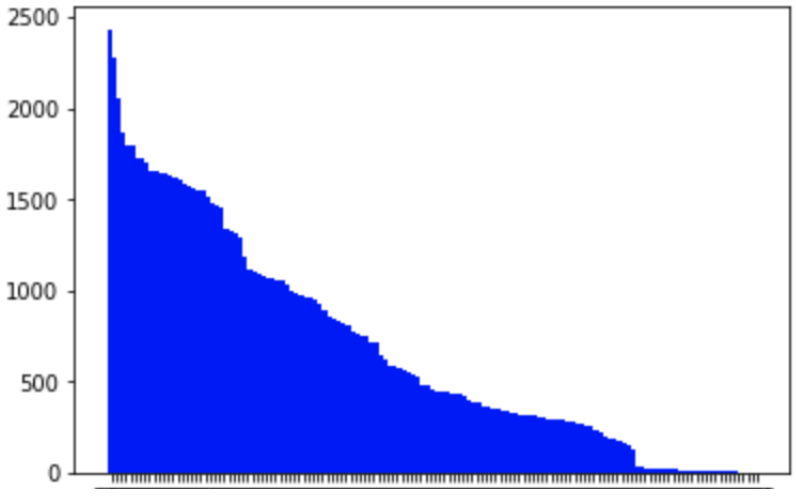

Figure 6.1: Frequency histogram of executed tasks in dataset

Figure 6.1 shows that some tasks are executed almost 2500 times in total, in the entire dataset. There are also a number of task that are executed a few times or not

at all.

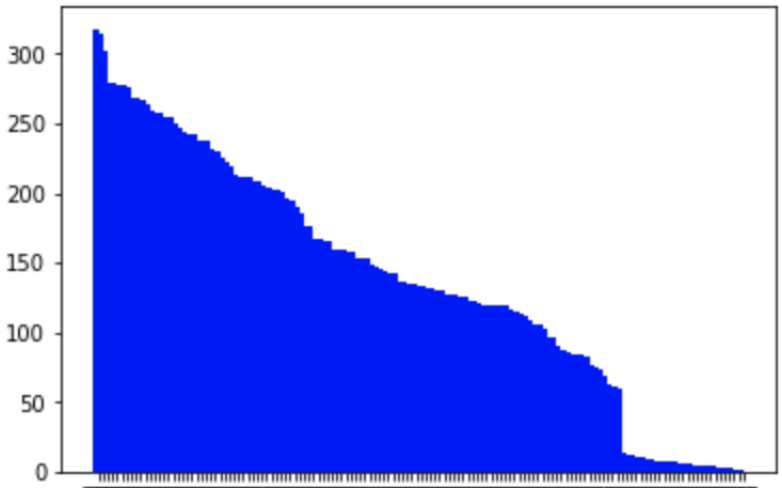

Figure 6.2: Frequency histogram of number of patients that executed tasks

Figure 6.2 appears to have a similar shape to figure 6.1. The maximum is now slightly above 300, meaning that there are certain tasks that almost everyone has executed at least once. Figure 6.2 does not take into consideration the number of times a patient does a task, but merely how many patients have done each tasks at least once.

The top 20 most frequent tasks from figure 6.2 will be considered for the input features to create 'task-profiles' per patient. These represent the 20 most assigned tasks by the GNPT scheduler. The new input dataset will have dimensions [385,20]; each of the 385 patients has 20 features, the value of each of these features indicates how often that patient has executed the respective task in the first three sessions. The task execution is limited to three sessions due to the varying length of sessions in the patient data.

6.2 Decision tree classification

The k-means clustering was not as useful as expected, therefore, for the analysis of task execution with regards to improvement, the k-means clustering will not be repeated. Instead, the first step will be decision tree classification based on merely the task features. Additionally the score-data of the first three sessions is also applied. Previous results showed better accuracy with three sessions rather than six, therefore the six-session scores will not be repeated either.

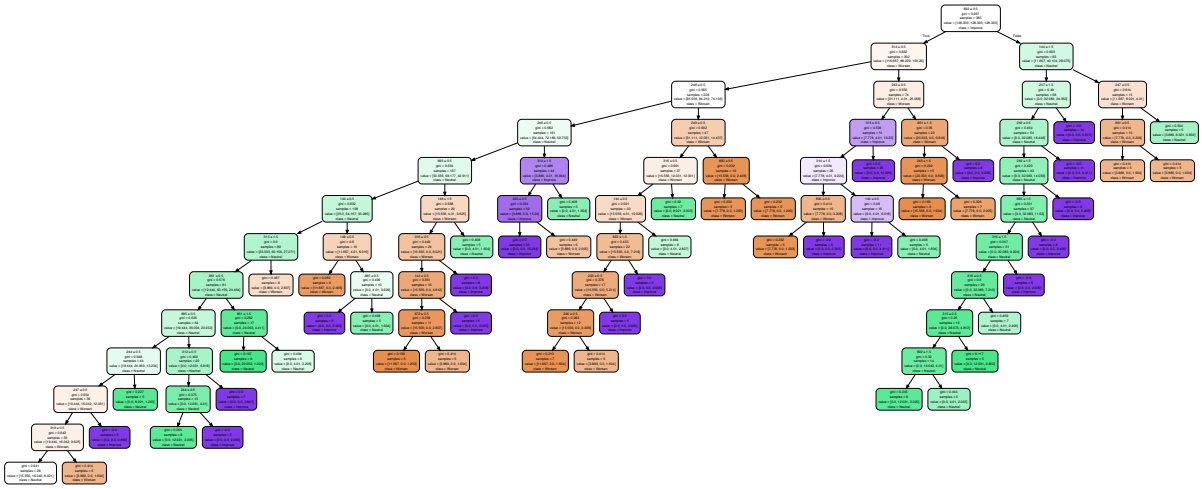

Figure 6.3: Decision tree based on task execution

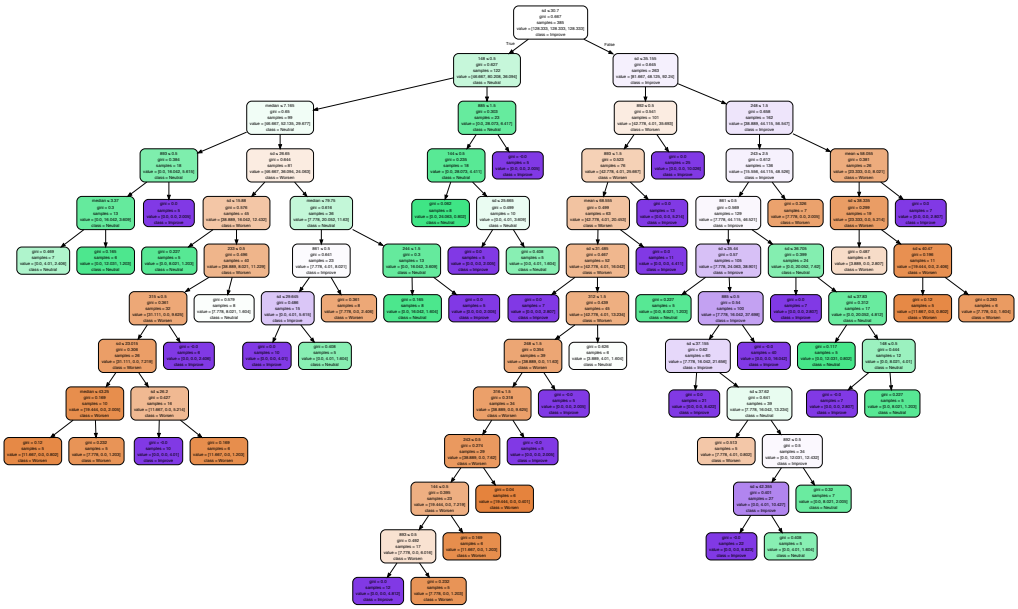

Figure 6.4: Decision tree based on task execution + 3 session scores

Similar to the cognitive profile input features, once again the score-related data: the mean, median and standard deviation of the scores during the first three sessions, appears high-up in the tree in figure 6.4. This indicates that these are strong indicative factors for determining improvement in this feature space as well.

Both trees show interesting patterns, demonstrating paths that almost entirely lead to one specific class. There are some clearly distinguishable branches of classes.

6.3 Further analysis

Table 6.1 includes the comparison with a number of different classification algorithms, as well as variations of the feature space and classes.

	DT	SVM	SVM-Lin	NN	GNB	Average
Dataset	42,34%	**82,59%**	**82,33%**	75,81%	52,72%	67,16%
Dataset+3	47,83%	**83,30%**	70,43%	**82,85%**	54,10%	67,70%
Dataset+6	47,23%	**83,84%**	69,49%	**83,31%**	53,23%	67,42%
Dataset-WvsO	65,89%	**91,45%**	**91,10%**	85,97%	74,11%	81,70%
Dataset+3-WvsO	72,18%	**91,59%**	84,08%	**91,00%**	73,58%	82,51%
Dataset-NvsI	65,80%	**90,84%**	**90,63%**	84,04%	66,13%	79,49%
Dataset+3-NvsI	70,40%	**91,06%**	82,83%	**90,10%**	68,96%	80,67%

Table 6.1: Accuracy comparison of different classification algorithms on different datasets based on task input. Accuracies are averages of 100 runs.

Overall, the datasets including the features regarding 3-session score data show an improved accuracy, and this accuracy decreases once again for 6-session score data. Furthermore, the data corroborates the fact that it is easier to predict between the classes *Worse* and *Other* than between *Neutral* and *Improve*, even with a completely different feature space.

Overall the accuracies are lower for the task-execution feature space in comparison to the cognitive profile feature space, though this difference is only a few percent. However, this does indicate that the initial profile of a patient is a stronger indicator of improvement as opposed to the tasks executed.

6.3.1 Linear SVM

The linear SVM feature weights are used for feature analysis. The left column represents task ID's.

	D-WvsO	D+3-WvsO	D-NvsI	D+3-NvsI	**Average Rank**
885	2	14	8	6	**7,5**
247	9	16	19	12	**14**
893	16	9	1	4	**7,5**
245	17	13	3	3	**9**
315	12	6	7	5	**7,5**
244	6	7	0	0	**3,25**
246	8	11	6	16	**10,25**
249	11	5	17	13	**11,5**
144	5	3	14	8	**7,5**
872	18	10	12	9	**12,25**
243	15	8	15	17	**13,75**
248	1	1	16	15	**8,25**
312	19	22	11	7	**14,75**
313	3	4	9	11	**6,75**
892	0	0	5	1	**1,5**
314	7	15	2	2	**6,5**
148	4	2	10	10	**6,5**
316	10	18	4	20	**13**

Table 6.2: Task ranking according to linear SVM weights

Out of the 20 most frequent tasks selected based on figure 6.2, 18 different tasks occur in the top 10 shown in table 6.2. This means that each of the 18 tasks shows up at least once in the 10 most important features in one of the datasets, according to the linear SVM weights. The '3 session' score-related features, the standard deviation, mean and median, do not occur in the top 10. The features which belong to the top 5 in any dataset (rank 0 to 4) are shown in red. Consequently, these features are selected for further analysis. Table 6.3 shows the average value per feature for each of the three datasets, and each corresponding class.

	Dataset+3			Dataset+3WvsO		Dataset+3NVSI	
Feature	W	N	I	W	O	N	I
885	0,242	0,438	0,569	0,242	0,557	0,438	0,569
893	0,485	0,219	0,628	0,485	0,591	0,219	0,628
245	0,364	0,156	0,394	0,364	0,372	0,156	0,394
244	0,242	0,156	0,522	0,242	0,489	0,156	0,522
144	0,636	0,375	0,413	0,636	0,409	0,375	0,413
248	0,697	0,281	0,338	0,697	0,332	0,281	0,338
313	0,697	0,375	0,375	0,697	0,375	0,375	0,375
892	0,152	0,438	0,331	0,152	0,341	0,438	0,331
314	0,333	0,250	0,453	0,333	0,435	0,250	0,453
148	0,242	0,469	0,409	0,242	0,415	0,469	0,409
316	0,273	0,344	0,275	0,273	0,281	0,344	0,275

Table 6.3: Average feature values per dataset per class

Once again the columns of the same class are identical. For instance, task 885 has been executed on average 0.242 times during the first three sessions for all worsening patients.

Figures 6.5, 6.6, 6.7 show the difference in average task frequency from table 6.3 in a frequency histogram.

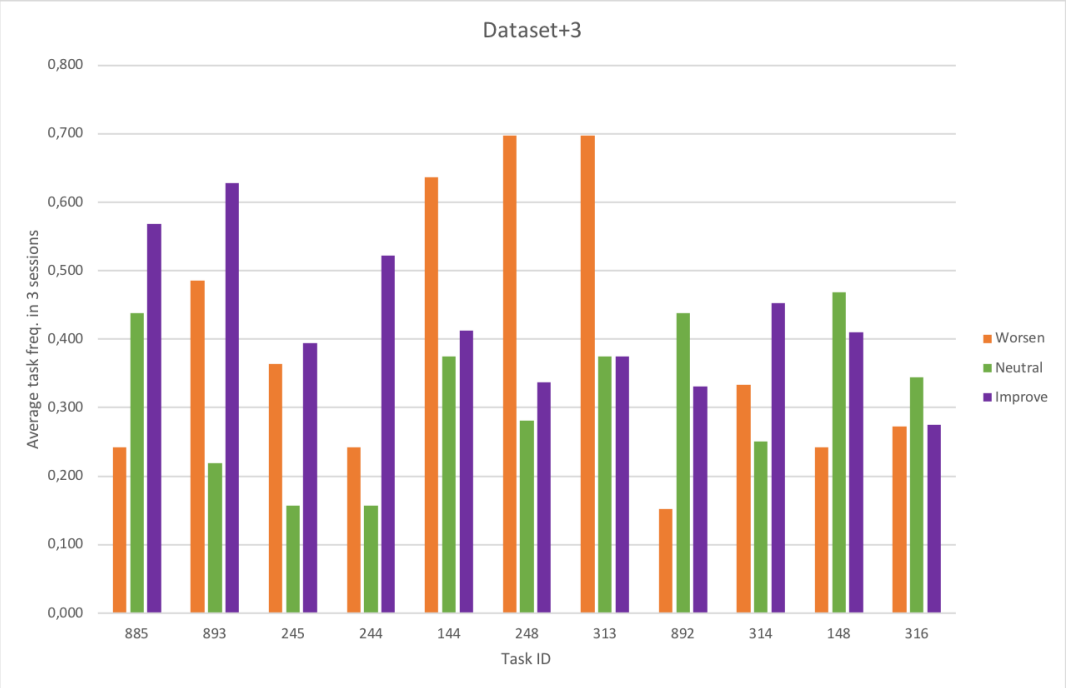

Figure 6.5: Comparing task frequency per class based on table 6.3

Figure 6.6: Comparing task frequency per class based on table 6.3

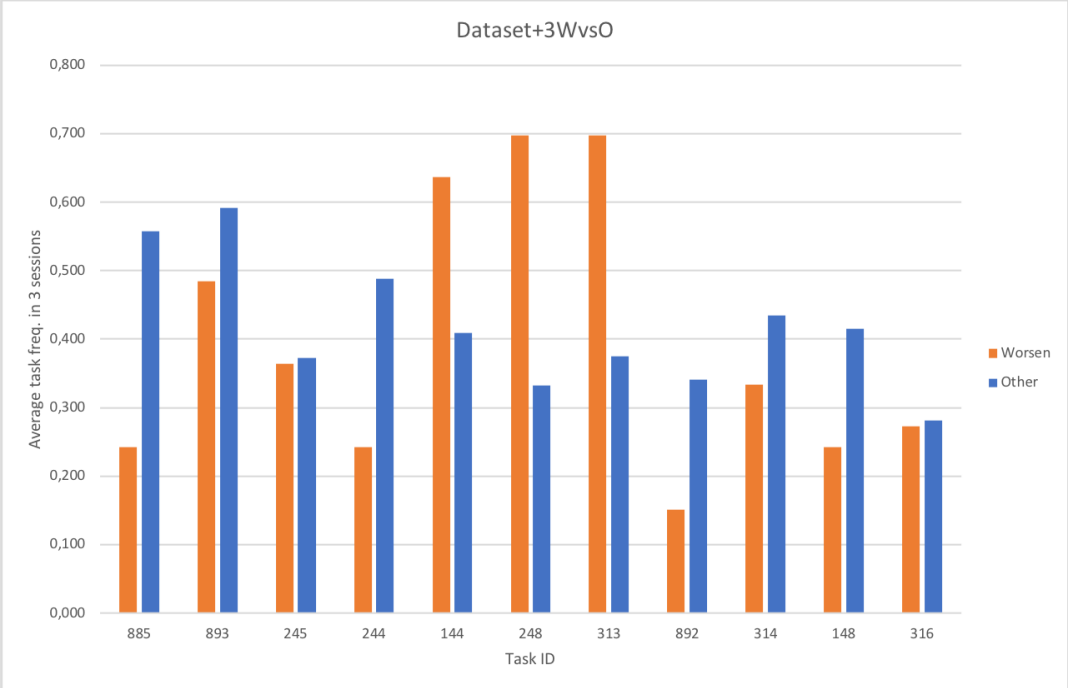

Figure 6.7: Comparing task frequency per class based on table 6.3

The three figures show very interesting results. The y-axis represents the average (for all patients) number of times a task has been executed in the first three sessions. It appears that more executions of tasks 144, 248, and 313 are quite clearly associated with worsening patients. This is a surprising result, as one would expect rehabilitative exercises to have a positive effect regardless.

These results can be implemented in the current proposal for a new scheduler, helping to attach weights to tasks according to the results from this research. This will be further discussed in the section regarding implementation of the results.

7 | Evaluation of the research

7.1 Assumptions

Some assumptions have been made during the process, which may or may not be valid. The main assumptions are outlined below.

- The improvement is based only on the cognitive functions that are evaluated both prior and post rehabilitation. It is possible that a patient was so severely impaired in a certain area that this was impossible to test during the pre-evaluation. If the condition has improved it possibly was tested post-rehabilitation but this improvement will not be counted in the designed improvement function. Vice versa, if a patient has not been tested for something prior because there appeared to be no problem, however the process revealed some unexpected issues and post-evaluation results show there is indeed a problem, this is also not taken into consideration.
- When expanding the input features with the scores of the first few sessions, the settings of the task are disregarded. A task can be set at different difficulties per patient. This has been disregarded by making the assumption that the task is set-up at the correct level for each patient. If a patient performs either too well (it is too easy) or very badly (too difficult), then after a few sessions this will be adjusted. As only the first few sessions are taken as input, the assumption is made that no additional adjustments for difficulty have been made yet. The difference in accuracy between 3-session scores and 6-session scores indicates that this assumption might not be entirely correct. It appears that during the first three sessions the settings were not 'ideal' yet, after which they are adjusted.

In addition to this, the resulting task-weights also have their limitations. Currently, the suggested weights are only applicable for a limited number of tasks, 11 tasks to be precise. Due to the relatively small dataset, the number of tasks used to build a task profile was limited to the 20 most frequent. For instance, if a task has only been executed once and that particular patient did not improve, then the task would be weighted accordingly without sufficient data to support this decision. This

would be an unreliable result, which is why only the most frequent tasks are currently incorporated. This leaves many of the tasks at a default weight. Ideally, this problem would automatically be solved with time as more data becomes available.

7.2 Objectives

Overall, the research project has successfully visualized the available data in a number of ways; it very clear to understand for non-data-mining experts.

The k-means clustering models were visual and comprehensible, though not surprisingly, it appears that the patients could not clearly be divided into a certain number of clusters. The decision trees provided a more thorough understanding of the data, demonstrating which features are important and which paths lead to positive or negative results. Visually, these models are very self-explanatory and can be used by the Guttmann Institute to better understand their data and results. Finally, the task-profile analysis in combination with the linear SVM weights provides a different take on the dataset. Interestingly, the feature-importance ranking appears different than in the decision tree. These weights can help further improve the scheduler by focusing on providing the best schedule for each patient with the greatest likelihood of improvement.

The objective of the research was to gain further insight into the current approach, and contribute to a new possible solution. Both of these key points have been addressed.

One element that is missing however, is to check how the weighted tasks will affect the difference in distributions between effort and impairment, one of the 'red-flags' in the current scheduler. However, due the the setup of the Gurobi solver, which will be explained in the following chapter, it optimizes exactly this similarity, so it is expected for this not to cause any problems.

8 | Implementation

The objective is, given a patient with a cognitive impairment, a collection of tasks and a number of sessions which to plan: find the optimal task allocation per session that 'minimizes the difference' between impairment and effort.

8.1 Gurobi Optimization

The Gurobi solver provides a new approach to tackle the visible problem of mismatched distributions. The Gurobi optimizer is able to go through billions, or even trillions of solutions and return the best one [13].

As mentioned previously, the optimization problem is set up to minimize the difference between the distribution of cognitive impairments and effort. Figure 8.1 shows the similarity between the distribution of effort versus impairment-severity (level of cognitive damage).

The proposed model works with a matrix S_{ijt}, a three dimensional matrix. The first dimension represents the sessions, the second the time-slots, and the third the tasks. Each position indicates that during session i, in time-slot j, task t is executed. For instance, if we find that the value at S_{123} is 1. This indicates that in session 1, during time slot 2, task 3 is executed. All other tasks along the t dimension, S_{12t}, will have value 0 as these are not executed at this point in time. This is one of the constraints: only one task per time-slot. Evidently, a number of constraints need to be applied, these also include: sessions are maximum 45min long, tasks are not repeated within the same session, and in the total schedule, a patient should not repeat a task more than three times. The mathematical formulation of these constraints will follow shortly.

Each task has an assigned suitability, this is used as a measure for effort. The variable P_k incorporates this notion.

$$x_k = \text{number of minutes dedicated to impairment } k$$

$$P_k = \frac{x_k}{\text{number of total minutes}}$$

$$x_k = \sum_i \sum_j \sum_t S_{ijt}.task.profile[t][k]$$

P_k is the variable on the y-axis in figure 8.1.
The mathematical formulation of the restrictions in place:
Only one task per time-slot:

$$\forall i \forall j \sum_{t} S_{ijt} \leq 1$$

No session exceeds 45 minutes:

$$\forall i \sum_{j} \sum_{t} S_{ijt} time[t] \leq 45$$

Tasks are not repeated within the same session:

$$\forall i \forall j \sum_{t} S_{ijt} \leq 1$$

In the overall schedule, a task should not be executed more than three times:

$$\forall t \sum_{j} \sum_{i} S_{ijt} \leq 3$$

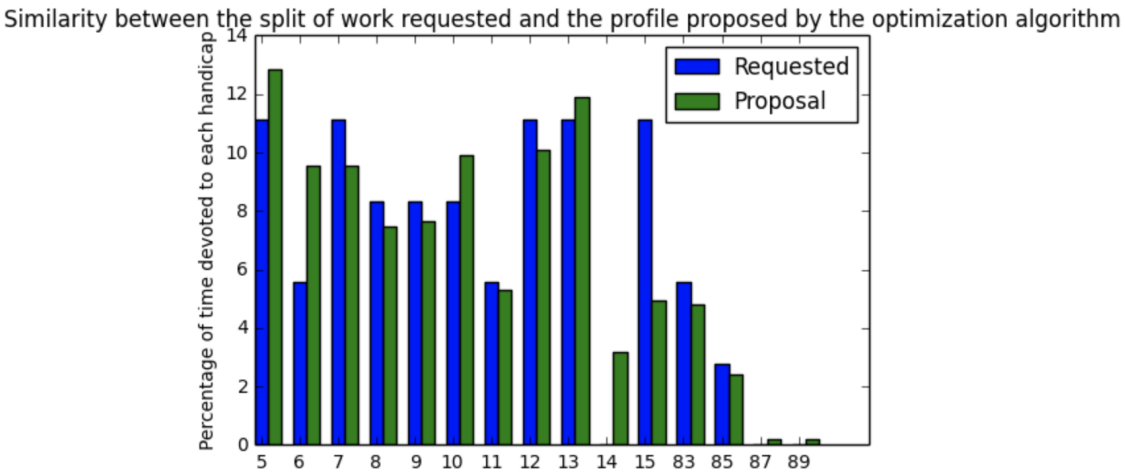

Figure 8.1: Preliminary results of proposed solver

Note that in figure 8.2 the number of tasks per session varies due to the fact that different tasks take a varying amount of time. These results do not yet include the incorporation of task weights.

To summarize, the proposed program goes through the following steps:

1. Set up model
2. Function objectives
3. Constraints
4. Optimization
5. Obtain solution

This research project allows possible adjustments to be made in step 3. The results of the experiments could provide additional insights to better constrain the results and thus provide a more appropriate, and better, solution.

```
Session 0
  Perform task 1239
  Perform task 243
  Perform task 1243
  Perform task 1244
  Perform task 249
  Perform task 159
  Perform task 287
  Perform task 144
  Perform task 216
  Perform task 245
Session 1
  Perform task 1240
  Perform task 1241
  Perform task 1249
  Perform task 236
  Perform task 237
  Perform task 182
  Perform task 222
  Perform task 244
  Perform task 245
Session 2
  Perform task 216
  Perform task 312
  Perform task 159
  Perform task 892
  Perform task 182
  Perform task 253
  Perform task 254
```

Figure 8.2: Sample output of Gurobi solver

8.2 Deployment plan

Overall, the most actionable results are the ones shown in the frequency histograms in figures 6.5, 6.6, 6.7. Taking the values from figures 6.5 and 6.7 for instance, these can be translated into weights for the respective tasks.

Task	**Improve - Worsen**	**Weight**
885	0,326	1,326
893	0,143	1,143
245	0,030	1,030
244	0,279	1,279
144	-0,224	0,776
248	-0,359	0,641
313	-0,322	0,678
892	0,180	1,180
314	0,120	1,120
148	0,167	1,167
316	0,002	1,002

Table 8.1: Task weights based on figure 6.5

The middle column in tables 8.1 and 8.2 is the *Improvement* score minus the *Worsen* score, and the *Other* score minus the *Worsen* score, respectively. Each of these scores can be found in table 6.3. When this number is negative it indicates that the task is possibly detrimental to the patients progress, as it is associated with bad results. The current Gurobi implementation does not implement any task weights. Thus assuming a default weight of 1 for each task, the right most column is 1 + the middle column. These can now be taken as weights for the tasks, where a value greater than 1 indicates it is more important, and a value less than 1 indicates

the task is penalized.

Task	Other - Worsen	Weight
885	0,314	1,314
893	0,106	1,106
245	0,009	1,009
244	0,246	1,246
144	-0,227	0,773
248	-0,365	0,635
313	-0,322	0,678
892	0,189	1,189
314	0,101	1,101
148	0,172	1,172
316	0,009	1,009

Table 8.2: Task weights based on figure 6.7

In addition to the implementation of weights, the visualizations produced during this research can also be valuable to the Guttmann Institute. Medical professionals will be able to better interpret the results and determine meaningful conclusions.

8.3 Monitoring & maintenance

The new weights can be implemented in the Gurobi solver and this will yield an adjusted schedule. First and foremost, the theoretical results should be reviewed and approved by medical professionals. Once these have been approved, the results can be implemented in actual patient schedules. The only way to properly evaluate the results is to test the new schedules on current patients.

These resulting weights should be seen as dynamic, the results of patients should continually be incorporated to adjust the weights accordingly.

9 | Conclusion

9.1 Summary

This work represents a thorough analysis of a dataset provided by the Guttmann Institute regarding patients suffering from ABI that have gone through rehabilitation therapy. The dataset is dissected in detail, determining its possible weaker spots where strong assumptions have to be made in its place. First off, a statistical analysis is done to summarize the main aspects of the datasets. Simple calculations show the average demographic profile of a patient, as well as the most common impairments, and the most common combinations of impairments.

Secondly, a k-means clustering model is used to cluster the patients according their clinical profile. These clusters, reduced to two dimensions using PCA, are visualized together with the final improvement of the patients in order to determine whether there is a link between the initial assessment of a patient and their final improvement. The improvement is visualized as a score as well as a category. There appear to be small clusters of similar patients that do improve similarly. The input dataset is then extended to the initial clinical profile plus the mean, median and standard deviation of the score in the first three sessions. The clusters do appear very different, however it does not seem to provide any additional insights.

The third step includes modeling the data using decision trees. The hierarchical nature of the trees show which features are stronger indicators of improvement. There do appear to be distinguishable paths between the different classes of improvement. The decision trees are built using three different input datasets, the regular initial clinical profile, and adding the scores of the first three and six sessions. Using a $\frac{2}{3}$ training, and $\frac{1}{3}$ test set split, the results show improved accuracy for adding the data of the first three sessions, but this decreases again for the data of the first six sessions. The current hypothesis is that the doctors adjust the difficulty after a certain number of sessions, reducing the indicative power of the scores.

The classes are adjusted in order to gain more clarity into which are possibly easier to predict. Naturally, the accuracy scores are higher when predicting between two classes rather than three. The data shows that on average it is easier to distinguish between worsening and other patients, rather than between neutral and improving patients.

The final step of this stage compares a number of machine learning algorithms and seven different variations of the input data and classes. The linear SVM feature weights are used as an indicator of feature importance, which leads to a new path of analysis. In the decision trees the data regarding the scores tended to be close to the root if not the root itself. However, the linear SVM weights show quite the contrary, none of the three score-related features appear in the top 10 ranked features.

After concluding this thorough research into the possibly predictive nature of the initial cognitive profile of a patient with respect to final improvement, a different approach is taken. Instead of focusing on the initial profile, task execution is now analyzed. Each patient now has a 'task-profile', indicating that out of the overall 20 most frequently executed tasks, how many times each one has been executed in the first three sessions. The decision tree models are repeated as in the previous stage, as well as the comparison with other machine learning algorithms. The result of the additional six-score data being worse than three-score data is consistent.

Finally the linear SVM coefficient analysis leads to a clear table showing the difference in improvement between the number of task executions. It appears that certain tasks that are repeated often are related to worsening results, whereas others are more likely to lead to improvement.

The clinical profile input data lead to higher accuracies when predicting improvement than the task execution input data. However, the task execution feature space was very useful as it lead to actionable results and a new implementation proposal.

9.2 Future Work

Considering the final weights analysis of the tasks, it would be very interesting to apply this into the Gurobi model and actually test the results with patients at the Guttmann Institute. Understandably, it first has to be approved and perhaps simulated thoroughly and analysis by the medical professionals in charge.

Another approach would be to create an interactive scheduler. Neuroscientists are expensive; in stead of a patient having their schedule and progress reviewed every so often, a real-time AI assistant could adjust the schedule by monitoring the patient's progress on current tasks. This could include adjusting the difficulty automatically. This would avoid the patient spending 3 sessions on either 'too easy' or 'too hard' task settings, as well as changing the exercises around completely. Especially seeing that repetition does not always lead to more positive results. Of course this would still need to be supervised by an expert.

References

[1] Joan Serrà, Josep Ll. Arcos, Alejandro Garcia-Rudolph, Alberto García-Molina, Teresa Roig Rovira, and Jose Tormos. Cognitive prognosis of acquired brain injury patients using machine learning techniques. In *The Fifth International Conference on Advanced Cognitive Technologies and Applications*, 05 2013.

[2] Javier Solana Sanchez, Cesar Caceres, Alberto Garcia-Molina, Paloma Chausa, Eloy Opisso, Teresa Roig-Rovira, Ernestina Menasalvas, Jose Tormos, and Enrique Gomez Aguilera. Intelligent therapy assistant (ita) for cognitive rehabilitation in patients with acquired brain injury. *BMC medical informatics and decision making*, 14:58, 07 2014.

[3] Guttmann neuro-personal trainer. https://www.guttmann.com/es/tags-siidon/gnpt. Accessed: 2017-12-10.

[4] Mouhib Alnoukari and Asim El Sheikh. *Knowledge Discovery Process Models: From Traditional to Agile Modeling*. Business Science Reference (an imprint of IGI Global), 01 2012.

[5] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer, 2000.

[6] Feedforward neural networks. https://brilliant.org/wiki/feedforward-neural-networks/.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[9] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[10] Nidhi H Ruparel, Nitin M Shahane, and Devyani P Bhamare. Learning from small data set to build classification model: A survey. In *Proceedings IJCA*

International Conference On Recent Trends In Engineering and Technology (ICRTET), pages 23–26, 2013.

[11] Javier Solana Sanchez, Cesar Caceres, Alberto Garcia-Molina, Eloy Opisso, Teresa Roig, Jose Tormos, and Enrique Gomez Aguilera. Improving brain injury cognitive rehabilitation by personalized telerehabilitation services: Guttmann neuropersonal trainer. *IEEE journal of biomedical and health informatics*, 19, 09 2014.

[12] Alarcos Cieza, Thomas Brockow, Thomas Ewert, Edda Amman, Barbara Kollerits, Somnath Chatterji, T Berdihan Ustun, and Gerold Stucki. Linking health-status measurements to the international classification of functioning, disability and health. *Journal of Rehabilitation Medicine*, 34(5):205–210, 2002.

[13] Gurobi Optimization. Gurobi optimizer 5.0. Gurobi: http://www. gurobi. com, 2013.

Appendix

Classification algorithm settings

All algorithms made use of the python scikit learn package. The details of the parameters for the experiments can be found below.

Decision Tree

```
sklearn.tree.DecisionTreeClassifier(criterion='gini',
splitter='best',max_depth=None, min_samples_split=2,
min_samples_leaf=5, min_weight_fraction_leaf=0.0,
max_features=None, random_state=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
class_weight='balanced', presort=False)
```

SVM

```
sklearn.svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='auto',
coef0=0.0, shrinking=True, probability=False, tol=0.001,
cache_size=200, class_weight=None, verbose=False, max_iter=-1,
decision_function_shape='ovr', random_state=None))
```

Linear SVM

```
sklearn.svm.LinearSVC(penalty='l2', loss='squared_hinge',
dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True,
intercept_scaling=1, class_weight=None, verbose=0, random_state=0,
max_iter=1000)
```

Neural Network

```
sklearn.neural_network.MLPClassifier(hidden_layer_sizes=(5,2),
activation='relu', solver='lbfgs', alpha=1e-5,
batch_size='auto', learning_rate='constant',
learning_rate_init=0.001, power_t=0.5, max_iter=200,
shuffle=True, random_state=1, tol=0.0001, verbose=False,
warm_start=False, momentum=0.9, nesterovs_momentum=True,
early_stopping=False, validation_fraction=0.1,
beta_1=0.9, beta_2=0.999, epsilon=1e-08)
```

GaussianNB

```
sklearn.naive_bayes.GaussianNB(priors=None)
```