A Performance Predictive Model for Emergency Medicine Residents

McMaster University

MSc Thesis

A Performance Predictive Model for Emergency Medicine Residents

Author:
Ali Ariaeinejad

Supervisor:
Dr. Reza Samavi

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

in

eHealth

June - 2017

McMaster University, Master of Science eHealth (2017) Hamilton, Ontario

TITLE: A Performance Predictive Model for Emergency Medicine Residents

AUTHOR: Ali Ariaeinejad

SUPERVISORS:
Dr. Reza Samavi
Dr. Norm Archer
Dr. Thomas E. Doyle
Dr. Teresa M. Chan

NUMBER OF PAGES 52

Abstract

Competency-based medical education (CBME) is a paradigm of assessing resident performance through well-defined tasks, objectives and milestones. A large number of data points are generated during a five-year period as a resident accomplishes the assigned tasks. However, no tool support exists to process this data for early identification of a resident-at-risk failing to achieve future milestones. In this thesis, the implementation of CBME at McMaster's Royal College Emergency Medicine residency program was studied and the development of a machine learning algorithm (MLA) to identify patterns in resident performance was reported. The adaptivity of multiple MLAs to build a tool support for monitoring residents' progress and flagging those who are in most need of assistance in the context of emergency medicine education was evaluated.

Dedication

To my wife, for her constant support and enthusiasm for all of my endeavours.

Acknowledgments

I would first like to thank my thesis supervisor Dr. Reza Samavi. The door to Prof. Samavi office was always open whenever I ran into a trouble spot or had a question about my research or writing.

I would also like to acknowledge Dr. Teresa M. Chan as the second reader of this thesis, and I am gratefully indebted to her for her very valuable comments on this thesis.

I would also like to thank the experts who were helped me in this research project: Dr. Norm Archer, Dr. Thomas E. Doyle, Dr. Margaret Ackerman (Division Director, Division of Emergency Medicine, Dept. of Medicine, McMaster University), Dr. Alim Pardhan (Program Director, Royal College Emergency Medicine Residency Training Program), The McMaster Division of Emergency Medicine administrative staff (Teresa Vallera, Melissa Hymers, and Roxanne Patel) and Sanya Palli for their assistance with this project. Without their passionate support and input, this project could not have been successfully conducted.

Contents

List of Figures

List Of Acronyms

ABSITE - American Board of Surgery In-service Training Examination

AI - Artificial Intelligence

ANN - Artificial Neural Network

AUC - Area Under Curve

CBM - Curriculum-Based Measurement

CBME - Competency Based Medical Education

CC - Competency Committee

EM - Emergency Medicine

kNN - k-Nearest Neighbor

LA - Learning Analytics

McMAP - McMaster Modular Assessment Program

MLA - Machine Learning Algorithm

NBME - National Board Medical Examinations

NCA - Neighborhood Component Analysis

NN - Neural Network

OBE - Outcomes-Based Education

PGY - Post Graduate Year

ROC - Receiver Operating Characteristic

RCPSC -The Royal College of Physicians and Surgeons of Canada

SVM - Support Vector Machine

WBA - Work-Based Assessment

Chapter 1

Introduction

Medical education is transitioning from a time-based system to a competency-based framework (Iobst et al., 2010). Competency-based medical education (CBME) (Frank et al., 2010) is a paradigm of assessing resident performance through well-defined tasks, objectives, and milestones. It uses assessment processes that are more continuous, frequent and work-based. The ultimate goal of CBME is to ensure that residents are fully competent at the end of their training period, and also to support and foster their development throughout training by identifying performance gaps (Holmboe, Sherbino, Long, Swing, & Frank, 2010). In emergency medicine (EM) CBME is even more important because it is a generalist specialty, requiring physicians in this field to be competent with a wide range of fields and skills. The Royal College of Physicians and Surgeons of Canada (RCPSC) has started phasing in CBME as the preferred training method and developed CanMEDS framework (Frank & Danoff, 2007). CanMEDS defines the roles that a competent physician is expected to embody in the practice of medicine. These roles are further refined by each medical specialty into job-specific tasks related to their domain of expertise.

One example of a CBME assessment system is McMaster Modular Assessment Program (McMAP) (Chan & Sherbino, 2015) which collects data from 74 different work-based assessments (WBA) categorized by the CanMEDS roles (Frank & Danoff, 2007). Assessments contain task specific checklists, behaviourally anchored task-specific and global performance ratings, and written comments. McMAP collects approximately 400 data points per resident per year. The current method of identifying residents-at-risk in McMAP is the following: at the end of a certain period of time (usually around 12-16 shifts), data from the daily faculty member observations are compiled into a report card of scores, and based on cut-points that have been defined by the program's competency committee (CC), an administrative assistant compares the average scores a resident receives with a certain threshold. Those who receive a score below the cut point are flagged for review, and considered a "resident-at-risk". Given the importance of these scores, this unwieldy amount of data needs to be aggregated, analyzed, and interpreted more rapidly. Finding a better way to identify and address the needs of individual trainees, to flag areas in need of support or flag those who are at risk for underperforming, or to talent-manage high performing residents would be ideal, but no tool support exists to process this data.

Due to the importance of early identification of residents at risk, we have witnessed significant efforts from research communities to propose and design a model for detecting residents-at-risk. A number of studies reviewed in this paper predict performance of students in medical education (Derderian & Kenkel, 2016; Corrigan, Smeaton, Glynn, & Smyth, 2015; Hamdy et al., 2006; Hayden, Hayden, & Gamst, 2005) and other disciplines (Crawford, Tindal, & Stieber, 2001; Calvo-Flores, Galindo, Jiménez, & Piñeiro, 2006; Lykourentzou, Giannoukos, Mpardis, Nikolopoulos, & Loumos, 2009). Machine learning algorithms (MLA) and regression analysis have been used to analyze undergraduate university students' data to predict their performance (Yost et al., 2015; Crawford et al., 2001; Romero, Ventura, & García, 2008). Our review of the existing protocols shows that the main focus of studies was to find external elements that might predict the competency of learners. To date, there has been no study on the correlation of competencies (scores) with each other, or to study the effects of resident performance during a more continuous longitudinal data set.

To address the gap in the current methods, we aimed to utilize the implementation of CBME at McMaster's Royal College Emergency Medicine residency program to develop a machine learning algorithm (MLA), in hope of identifying patterns in resident performance. The model investigates all fluctuations of residents scores and creates a rich dataset of features. Instead of transferring all of the collected features directly to the MLA, the model performs neighborhood component analysis to find the best combination of features, which increases the accuracy of the model. We subsequently evaluated the adaptivity of multiple MLAs and regression analysis to build a support tool for monitoring residents' progress and flagging those who are in most need of assistance in the context of emergency medicine education.

1.1 Thesis organization

The thesis is structured as follows: Chapter 2 reviews the literature. Chapter 3 provides an overview of the proposed model and the specifications of its four main components (i.e. inputs and data characteristics, preprocessing, machine learning, and output). Chapter 4 presents the experimental procedures and the results of evaluation of the model. We conclude this thesis in Chapter 5 and discuss a number of directions for related future research.

Chapter 2

Literature Review

In this chapter, we review the related work in three areas: predicting performance in medical education, pedagogical prediction using machine learning algorithms and background of machine learning algorithms.

2.1 Predicting performance in medical education

Much of the early work in the area of medical education has been focused on learning analytics that do not seem to hold any predictive capacity. Metro *et al.* (Metro, Talarico, Patel, & Wetmore, 2005) and Brenner *et al.* (Brenner, Mathai, Jain, & Mohl, 2010) have studied the ability of resident selection process to predict the future performance of residents. They could not find statistically significant correlation among the selection committee scores and any of the areas evaluated during their residency. Elfenbein and colleagues studied the correlation between faculty evaluations of resident medical knowledge and resident American Board of Surgery In-service Training Examination (ABSITE) performance. Results of this study showed that "faculty evaluations of resident medical knowledge correlate poorly with resident ABSITE performance, and should not be used as an ongoing predictive tool" (Elfenbein et al., 2015). Meanwhile, other educators have used the

Pearson correlation (Parker, Alford, & Passmore, 2004) to find correlation between residents' predicted performance and their actual performance based on resident's self assessments. Results showed that the ability of residents to predict their performance was poor. They concluded that, self-assessment is not able to predict future performance of residents, which is in line with much of the educational psychology literature on self-assessment in medical education (Eva & Regehr, 2008; Regehr & Eva, 2006; Davis et al., 2006).

There are also a number of studies that found some measures to be more useful for predicting clinical ratings; especially when the measures are more related to actual performance. Wallenstein and colleagues (Wallenstein, Heron, Santen, Shayne, & Ander, 2010) evaluated the ability of an objective structured clinical examination (OSCE) to predict future resident performance. They found statistically significant correlation between overall OSCE scores and overall clinical performance scores. Promisingly, Hamdy and colleagues (Hamdy et al., 2006) studied whether performance scores from medical schools could be useful for predicting future performance in residency. They found mild to moderate correlations between medical school assessment measurements and performance in the residency.

Investigation in this group of related work revealed that identifying residents who are struggling in improvement is immensely important since it can help programs improve the quality of education. There is, however, a gap in this body of literature. The main focus of studies to date was to find external elements that might predict the competency of residents. To date, there has been no study on the correlation of residency assessment items (competencies) with each other or studying the effects of resident performance during a more continuous longitudinal data set. This study has filled this gap in the literature, showing the potential of CBME metrics and how they might be augmented by MLAs to inform teachers and administrators about how best to allocate scarce educational programming and resources.

2.2 Pedagogical prediction using machine learning algorithms

Machine Learning Algorithms (MLAs) are being used in many educational applications in order to help faculty members in finding weaknesses and strengths of students. Calvo *et al.* (Calvo-Flores et al., 2006) were able to predict user exam performance, detecting students who were struggling and in need of additional educational support. They established that features derived from the Moodle (Learning Platform or course management system (CMS)) logs were enough to predict success with a high degree of confidence. They used features like ratio of resources viewed and total resource viewed. Romero *et al.* (Romero et al., 2008) have studied the learning environment data that includes a four-step framework which are: collect data, process data, perform data mining/machine learning steps and deploy results. This study demonstrates different ways of applying data mining and machine learning methods to predict students' performance. Corrigan *et al.* (Corrigan et al., 2015) used time spent on a particular task and past performance to predict student's scores. They successfully predict likely [success rates] on a weekly basis. In another study by Lykourentzou *et al.* (Lykourentzou et al., 2009), the data from several multiple choice tests taken during a year and inputted to predict a student's success on the course's final multiple choice test. Three feed-forward neural networks were used in this method. Results showed that Neural networks have a higher correlation at all prediction stages in comparison with linear regression.

Our review of the literature showed that machine learning algorithms such as artificial neural networks, support vector machines and statistical methods like linear or logistic regression may be useful for predicting future performance of students. Overall, the most important factor for improving the accuracy of prediction is selecting the best combination of features extracted from the students' data.

2.3 Machine learning algorithms (MLAs)

Machine learning is a subset of artificial intelligence (AI) which gives computers the ability to learn without being specifically programmed. It is classified as a computer's ability to independently adapt and learn in order to interpret data including sound, images and text (S. B. Kotsiantis, Zaharakis, & Pintelas, 2007). Example of MLAs are the artificial neural networks (ANN), classification and clustering algorithms (Gomes, 2014). Machine learning has been used in many different aspects of science such as speech recognition (Bahdanau, Chorowski, Serdyuk, Brakel, & Bengio, 2016), image processing (Russ, 2016; Burger & Burge, 2016), phonetic recognition (Shim, Koh, Fister, & Seo, 2016), semantic classification (Dave, Lawrence, & Pennock, 2003), natural language and human writing processing, robotics, and audio processing (Chowdhury, 2003; Shotton et al., 2013; Greenberg et al., 2014).

Generally there are three types of machine learning algorithms: supervised, semi-supervised and unsupervised. The supervised method is used with large amounts of labeled data. Semi-supervised usually interprets a small amount of labeled data with a large amount of unlabeled data. Unsupervised methods are mainly used for clustering the data. One of the drawbacks of supervised learning is the information that the machine needs to learn must be labeled by a human, which is difficult, expensive and labor intensive. Supervised learning requires the algorithm to be provided with pre-existing information about a particular system or a set of problems, which the algorithm uses to learn or solve subsequent problems. Supervised learning is similar to how humans learn; teachers supply us information, providing us with feedback (i.e. identifying the desired response), which results in training that will help us to generate "rules" about the data we have encountered. Figure 2.1 shows a supervised learning block diagram where the environment (information or data) is fed to both the teacher and learning system.

The true error of such an algorithm is the average difference between the desired response and the actual response of the system (error signal) over all possible input-output examples (Haykin, 2009).

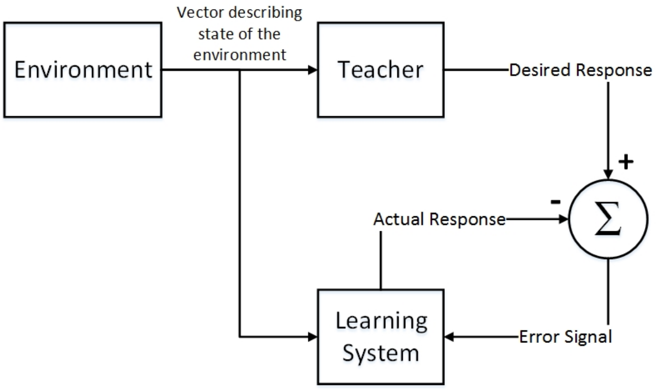


Figure 2.1: Supervised learning block diagram

2.3.1 Potential bias in labeling the data

Given the human labeling requirement in supervised learning, we are potentially facing the labeling bias. Using expert annotators can decrease the labeling bias considerably. Another way to reduce this bias is using an average of different annotations collected for each data-point from non expert annotators. This method is more reasonable because the cost of non expert annotators is relatively lower than the experts (Yamauchi, 2005).

2.3.2 Artificial neural networks (ANN)

Artificial neural networks (ANN) are built with a number of neurons that are connected to each other in different layers. Connections between neurons is a numeric changeable weight which gives the network ability to be adaptive to inputs and capable of learning (Yegnanarayana, 2009). Each layer in the ANN has a number of neurons; there is only one input layer, one output layer and at least one hidden layer. Figure 2.2 is an example of an ANN system with two neurons in the input layer, three neurons in the hidden layer and one neuron in the output layer. All neurons are connected to each other unless the network designer sets the weight of connection between two neurons to zero (which would mean the two items are disconnected). Neural networks are usually used in bioengineering problems such as image processing (Lindblad, Kinser, Lindblad, & Kinser, 1998) and speech recognition (Hinton et al., 2012).

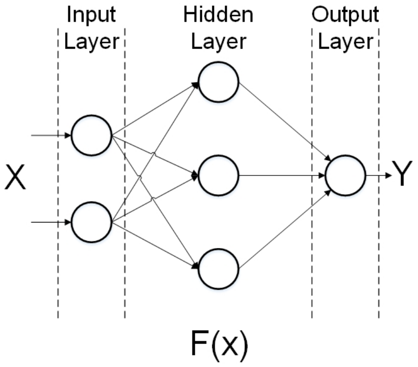


Figure 2.2: Artificial Neural Network

The available data are fed to the input layer and output of each layer fed to the next layer then finally the conclusion is the output of the whole network. The majority of neutral networks use non-linear activation functions. Figure 2.3 shows three popular activation functions used in neural networks (Karlik & Olgac, 2011).

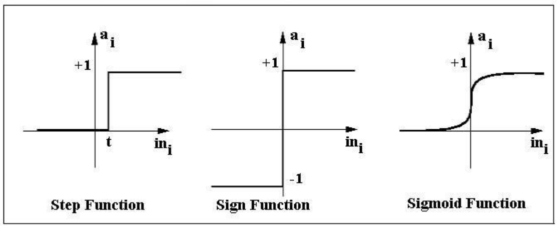


Figure 2.3: Activation Functions

The benefits of neural networks

Neural networks determine their computing power through their parallel conveyed structure and capacity to learn. Moreover, one of the most significant advantages of neural networks is their ability to generalize. They have the capacity to produce reasonable outputs even for inputs that the network was not even trained on. In addition, non-linearity is a positive property of neural networks as it enables neural networks to adapt to non-linear functional relationships (Haykin, 1998).

2.3.3 Support Vector Machines (SVM)

The Support Vector Machine (SVM) is a discriminative classifier formally characterized by an isolating hyperplane. SVM outputs an optimal hyperplane which classifies new examples based on labeled training data (supervised learning) (Joachims, 1998a).

In the Figure 2.4 we can see that there exist different lines that offer an answer to the issue. Are any of them superior to the others? We can intuitively define a criterion to estimate the worth of the lines: A line is not appropriate if it passes too close to the points because the line will be noise sensitive and it will not generalize effectively. Along these lines, the SVM objective is discovering the line that goes quite far from all points.

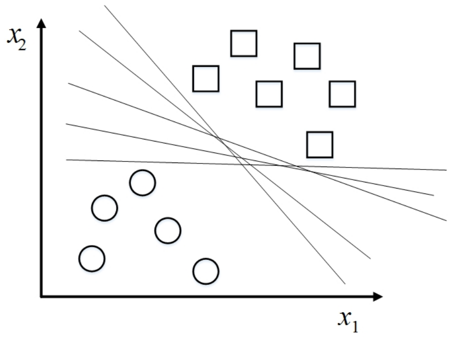


Figure 2.4: Support Vector Machines

The operation of the SVM algorithm depends on finding the hyperplane that gives the biggest least separation to the training examples. In SVM theory, this distance is named the margin. Accordingly, Figure 2.5 shows the ideal isolating hyperplane which maximizes the margin of the training data (Suykens & Vandewalle, 1999).

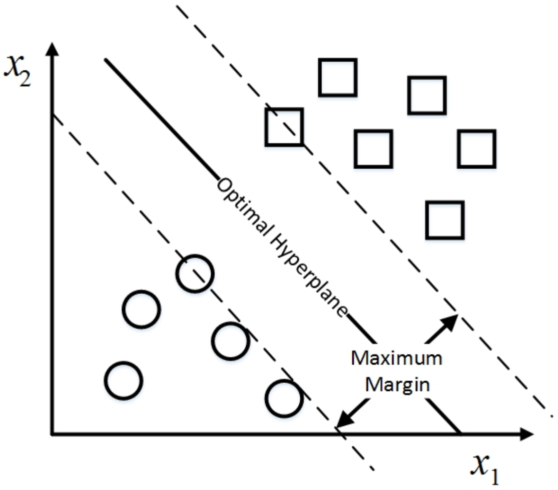


Figure 2.5: Optimal hyperplane

The formula used to formally define a hyperplane is as follows:

$$f(x) = \beta_0 + \beta^t x$$

where β^t is referred to the weight vector and β_0 to the bias. The optimal hyperplane can be represented in an infinite number of different ways by the scaling of β and β_0. Among all representations of the hyper plane, the selected one is

$$\left|\beta_0 + \beta^t x\right| = 1$$

where x symbolizes the training examples nearest to the hyperplane. Generally, the training examples that are nearest to the hyperplane are called support vectors. This representation is known as the canonical hyperplane.

The geometry that gives the distance between a point x and a hyperplane (β,β_0) is:

$$(\beta, \beta_0) : distance = \frac{|\beta_0 + \beta^t x|}{\|\beta\|}$$

Specifically, for the canonical hyperplane, the numerator is equal to one and the distance to the support vectors is:

$$distance_{supportvector} = \frac{|\beta_0 + \beta^t x|}{\|\beta\|} = \frac{1}{\|\beta\|}$$

Note that the margin presented in the previous section, here indicated as M, is twice the distance to the closest examples:

$$M = \frac{2}{\|\beta\|}$$

Finally, the issue of maximizing M is equal to minimizing a function $L(\beta)$ subject to some constraints. The constraints modeling the requirement for the hyperplane to classify correctly all the training examples x_i is:

$$min_{\beta,\beta_0} L(\beta) = \frac{1}{2}\|\beta\|^2 \quad subject \quad y_i(\beta_0 + \beta^t x) \geq 1 \forall_i$$

where y_i represents each of the labels of the training examples. This is an issue of Lagrangian optimization that can be solved using Lagrange multipliers to obtain the weight vector β and the bias β_0 of the ideal hyperplane (Joachims, 1998a, 1998b).

2.3.4 k-Nearest Neighbor (kNN)

kNN analysis classifies a new instance among a number of known examples. As shown in Figure 2.6, examples are pluses and minuses and the red circle is the new sample. kNN classifies the new sample based on a selected number of its nearest neighbors. Generally, we need to know whether the new sample can be a plus or minus (Larose, 2005; Beyer, Goldstein, Ramakrishnan, & Shaft, 1999).

To proceed, let's consider the outcome of kNN based on 1-nearest neighbor. It is clear that in this case kNN will classify the new sample as a plus (since the closest point carries a plus sign). Now let's increase the number of nearest neighbors to 2,

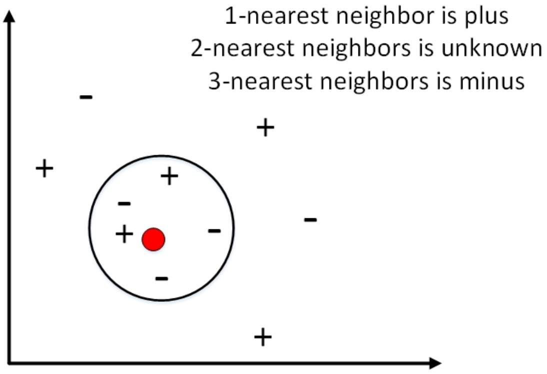

1-nearest neighbor is plus
2-nearest neighbors is unknown
3-nearest neighbors is minus
-
+
+
+
-
+
-
-
-
+

Figure 2.6: K-Nearest Neighbor analysis

this time kNN will not be able to classify the new sample since the second closest point is a minus, and so both the plus and the minus signs achieve the same score. For the next step, let's increase the number of nearest neighbors to 5, which will define a nearest neighbor region, indicated by the circle shown in the Figure 2.6. Since there are 2 plus and 3 minus signs, in this circle kNN will classify the new sample as a minus.

k-Nearest Neighbors (kNN) is a memory-based model characterized by an arrangement of items known as cases, for which the results are known (i.e., the examples are labeled). Every case comprises of an information case having an arrangement of independent qualities labeled by a collection of dependent results. The independent and dependent factors can be either continuous or categorical. For continuous dependent variables, the task is regression; otherwise it is a classification.

The decision of choosing k is very important in building the kNN. k can be considered as the most critical element of the model that can firmly impact the nature of predicts. One approach to choose the best k is to consider it as a smoothing parameter. For any given problem, a small value of k will prompt an expansive fluctuation in predictions. On the other hand, choosing a very large k may prompt a substantial model bias. In this manner, k should be set to a value large enough to minimize the probability of misclassification and small enough with respect to the number of cases in the example sample. Therefore, similar to any smoothing parameter, there is an ideal incentive for k that accomplishes the correct trade-off between the bias and the variance of the model (Beyer et al., 1999).

Chapter 3

The Proposed Model

In this chapter the proposed model as shown in Figure 3.1 is described. Using different machine learning algorithms and regression analysis the model monitors features extracted from resident's assessment data and then classifies the future situation of a resident as "at risk" or "not at risk". The model allows the program to create a more tailored educational curriculum for each resident, while at the same time becoming more active contributors. The model has four main components: input, preprocessing, machine learning and output which are discussed in sections 3.1, 3.2, 3.3, and 3.4 respectively.

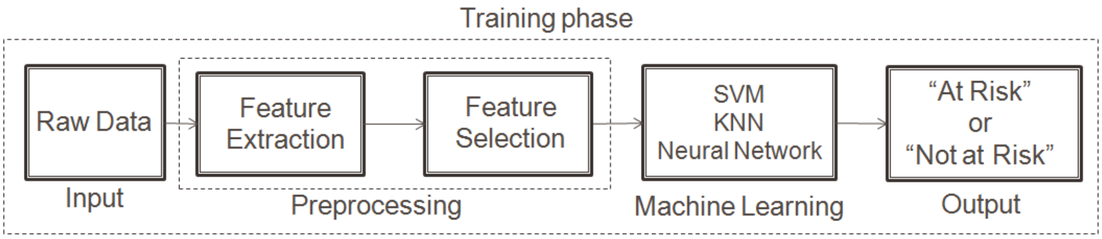


Figure 3.1: The training phase of the proposed model

3.1 Data characteristics and inputs

Data have been extracted from McMAP between years 2012 to 2016. The data include different attributes such as task name and code, task score, global score, post graduate year (PGY), block, group name and date of assessment. As described in Table 3.1, tasks are 74 work-based assessment instruments which are completed by faculty following direct observation of residents during shifts. For example, a faculty member might observe a resident providing discharge instructions to a patient. Each task is coded primarily to one CanMEDS role (Frank & Danoff, 2007). The global rating instrument is completed to capture the resident's global performance of all tasks during a shift. The scoring system is based on a consistent scale of 1 to 7, for both the uniquely anchored task scores and global performance scores. Post graduate year (PGY) shows the level of resident experience which can be 1 to 5 (i.e. their training level). A resident's year is divided into 13 four-week blocks. For each task score we have a group name (CanMEDS label) that categorizes the tasks, which can be professional, leader, manager, communication, collaboration, medical expert, scholar and health advocate.

Table 3.1 displays the descriptive analysis of available attributes. There are 1998 valid task score records with a minimum of 3 and maximum of 7. The mean of task scores is 6.23 and the median of task scores is 6. To investigate more about the distribution of input data we calculated the skewness using Eq. 3.1 where μ is mean, ν is median and σ is standard deviation (Mardia, 1974). The result of this equation was 0.28 which means that the distribution skewed to the right, and as illustrated in Figure 3.2, task score distribution is not symmetric (Boyer, Mitton, & Vorkink, 2009). We repeat the same distribution test on global scores. The mean of global scores is 6.08 and the median is 6. In terms of distribution, standard deviation tells us that most global scores are clustered around 5.4 and 7, which appears similar to task scores.

$$Skewness = (\mu - \nu)/\sigma \tag{3.1}$$

To investigate the correlation of task scores and global scores we conducted the Spearman correlation test since task and global scores are ordinal variables and the Spearman correlation test evaluates the monotonic relationship between two continuous or ordinal variables. Results of the test showed that correlation coefficient R (COR) was 0.67 ($p < 0.0001$). It means that these two variables have a semi-strong linear relationship (Taylor, 1990). This reveals that global scores are substantially correlated with task scores, and that the task assessments are inextricably tied to the global scores. Therefore, we decided to use task scores as predictor value for training the model.

Table 3.1: Descriptive analysis of available data

Attributes	Description	Mean	St. Dev	Valid N	Median	Min	Max
Task score	The resident's level of performance in a specific task	6.23	0.8	1998	6	3	7
Global Score	The resident's global performance of all tasks during the shift	6.08	0.77	2252	6	1	7
PGY	Post graduate year	-	-	2355	-	1	5
Block	A resident's year is divided into 13 four-week blocks	-	-	2355	-	1	13
Date	Date of assessment	-	-	2355	-	2012	2016
Group	Associated CanMEDS role	-	-	1998	-	-	-

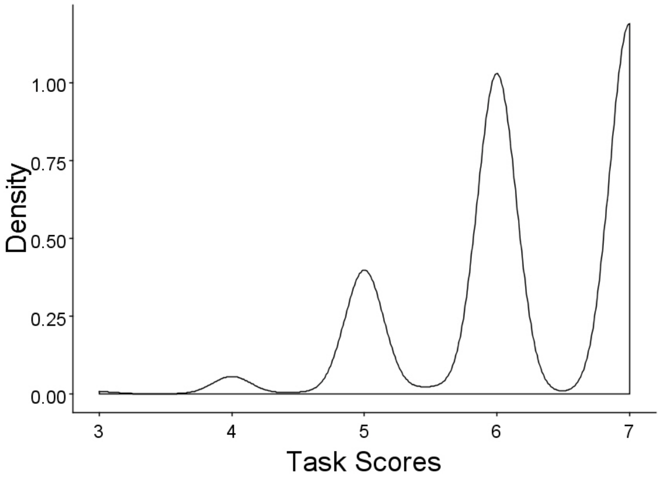


Figure 3.2: Density plot of Task Scores

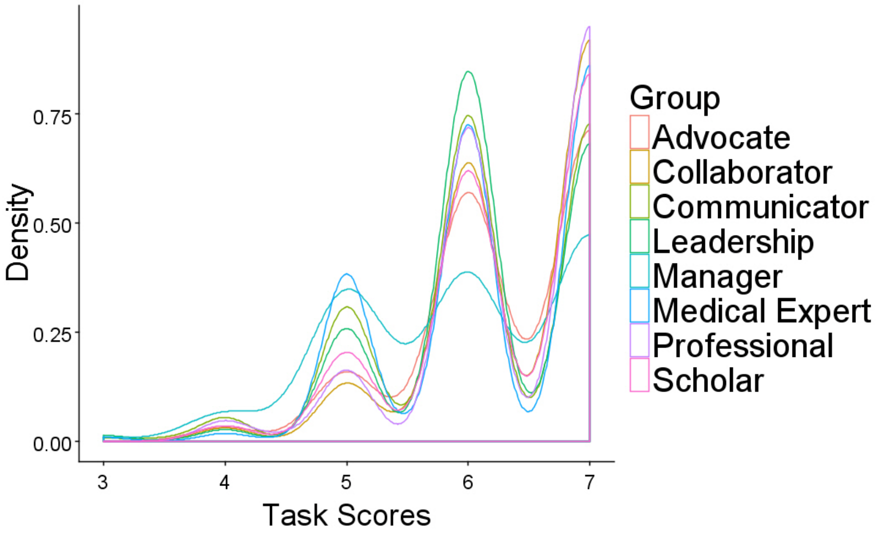


Figure 3.3: Density plot of Task Scores within the groups

The density plot of task scores is illustrated in Figure 3.2. It reveals that the majority of task-scores are 6 and 7. It seems that predicting score under 6 would be a challenge because the incidence of such scores is so low. By investigating task-scores within the groups as shown in Figure 3.3 we found that distribution of scores is similar within the different groups. Investigating global scores as displayed in density plot of global score in Figure 3.4, showed that most residents got 6 for their global performance rating. Figure 3.5 showed the distribution of global scores within the groups, which are distributed the same as the whole distribution of global scores.

3.2 Preprocessing

A key challenge of data preparation in the medical education context is transforming raw data to a set of features. The model's input is a sequence of scores for different tasks that residents received over a certain period of time. The sequence can have different lengths depending on a resident's level of training (i.e. post graduate year). Therefore, we need to extract a fixed set of features from each sequence. Additionally investigating and quantifying all fluctuations of each

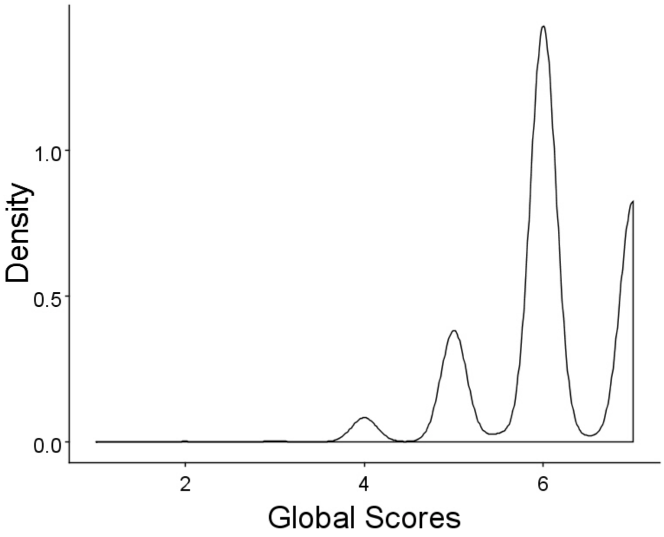


Figure 3.4: Global Scores

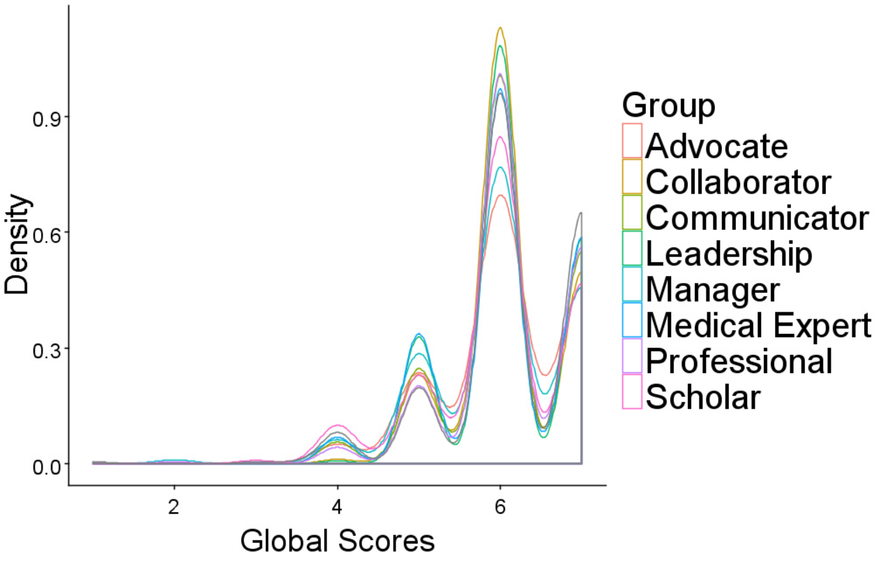


Figure 3.5: Global Scores within the groups

resident's scores and combining all extracted features for residents in one dataset, may give the machine learning algorithms the potential for finding hidden patterns.

In this study, we are looking for 18 different features which are listed below:

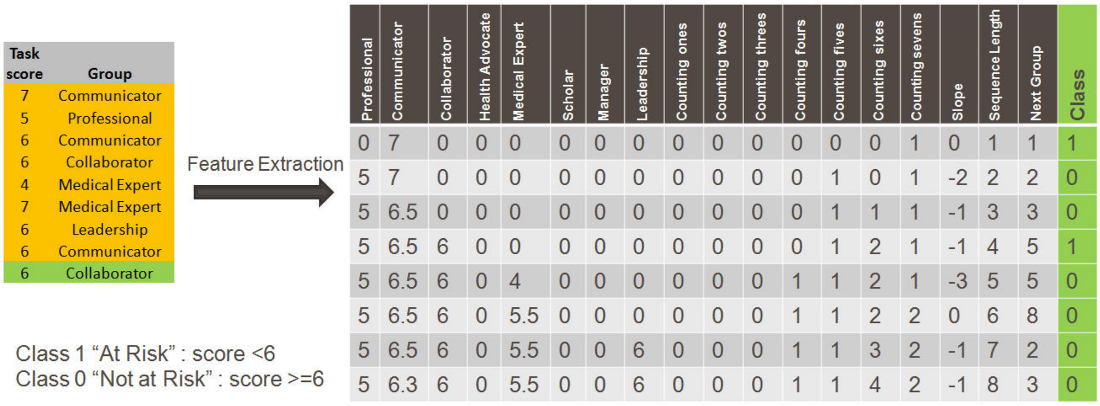
1. Average of professional task scores

2. Average of communicator task scores

3. Average of collaborator task scores
4. Average of health advocate task scores
5. Average of medical expert task scores
6. Average of scholar task scores
7. Average of manager task scores
8. Average of leadership task scores
9. Frequency of ones
10. Frequency of twos
11. Frequency of threes
12. Frequency of fours
13. Frequency of fives
14. Frequency of sixes
15. Frequency of sevens
16. Slope of the sequence
17. Sequence length
18. Next group label (i.e. the CanMEDS role)

In each step of features extraction we used a growing window to read the task scores. In the first step, size of the window was one and in each step we increased the size of window by one. It means that in each step we read one more score. Then we calculated 18 features and the class for each step. If the resident's score was under 6, the class would be 1 which means that the resident is "at risk".

If the score was 6 or 7, the class would be 0, which means resident is "not at risk". We continued the process until read the last score in the sequence. Then we repeated the process for all residents and added all extracted features to the features dataset. Figure 3.6 displays running one complete round of feature extraction for an example sequence of scores. The stepwise feature extraction process is illustrated in Appendix A.



Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Feature Extraction

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0
5	6.5	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	3	3	0
5	6.5	6	0	0	0	0	0	0	0	0	0	1	2	1	-1	4	5	1
5	6.5	6	0	4	0	0	0	0	0	0	1	1	2	1	-3	5	5	0
5	6.5	6	0	5.5	0	0	0	0	0	0	1	1	2	2	0	6	8	0
5	6.5	6	0	5.5	0	0	6	0	0	0	1	1	3	2	-1	7	2	0
5	6.3	6	0	5.5	0	0	6	0	0	0	1	1	4	2	-1	8	3	0

Class 1 "At Risk" : score <6
Class 0 "Not at Risk" : score >=6

Figure 3.6: Features dataset

The next step is normalizing the features. Each feature has its own numeric range which can have a knock-on effect on machine learning algorithm ability to learn and the objective functions will not work properly without normalization (Bolstad, Irizarry, Åstrand, & Speed, 2003). Therefore, each attribute should be normalized by scaling its values so that they fall within a specified range of 0 to 1. There are many methods for data normalization including min-max normalization, z-score normalization and normalization by decimal scaling. In a study by Al Shalabi et al. (Al Shalabi, Shaaban, & Kasasbeh, 2006) they concluded that min-max normalization always has the highest accuracy. Therefore, we used this method for normalizing the features.

Achieving a high precision and accuracy requires an appropriate collection of features. Using too many features leads to over-fitting the model and too few features leads to under-fitting the model. Generally, feature selection restricts the features in the models to those, which are most relevant. Using as few features as possible will also reduce the complexity of the model, which means it needs less time and computational power to run and is easier to understand. There are several ways to identify how much each feature contributes to the model and to restrict the number of selected features such as heteroscedastic discriminant analysis (HLDA), principal components analysis (PCA) and neighborhood component analysis (NCA). The results of a study by Singh-Miller et al. (Singh-Miller, Collins, & Hazen, 2007) showed that NCA has significant improvements in accuracy of the model over PCA and HLDA. Therefore, the feature selection method used in this study was NCA. NCA minimizes the expected leave-one-out classification error under a probabilistic neighborhood assignment (Goldberger & Salakhutdinov, 2005). Figure 3.7 shows the results of running NCA. Features which are weighted zero did not participate in the training or testing phases. NCA selected features were established by computing the average of scores related to professional, communicator, collaborator, medical expert, scholar and manager groups and the counted number of fours in the resident's scoring sequence.

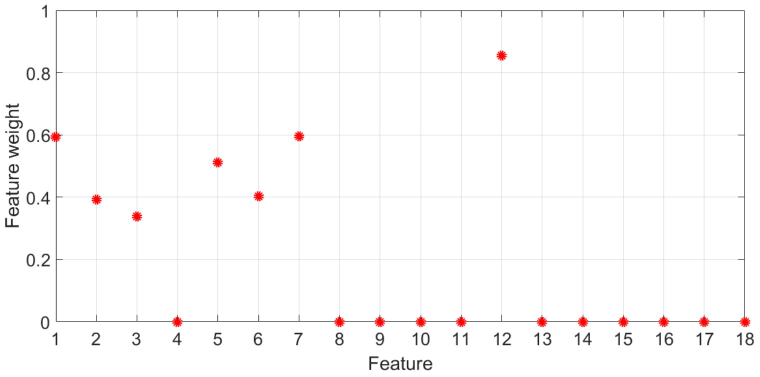


Figure 3.7: Weighted Features by NCA

3.3 Machine Learning

Determining the appropriate machine learning algorithm to accurately predict the future performance of a resident is another challenge that needs to be addressed. There has been little prior work in medical education using machine learning. Therefore, we studied the literature to find the most commonly used MLAs for similar problems. The Support Vector Machine (SVM) has been used by (S. Huang & Fang, 2013; Corrigan et al., 2015) and found useful in predicting student performance. Kotsiantis et al. (S. Kotsiantis, Patriarcheas, & Xenos, 2010) and Romero et al. (Romero, Espejo, Zafra, Romero, & Ventura, 2013) concluded that the k-Nearest Neighbor algorithm can be appropriate for the construction of a software support tool in predicting future performance of students. In different disciplines, like surgical training and e-learning, artificial neural networks showed better results in predicting student success (Yost et al., 2015; Lykourentzou et al., 2009). Regression analysis has been used in a wide variety of studies for predicting future performance of students (Hayden et al., 2005; Hamdy et al., 2006; Wallenstein et al., 2010). This literature review led us to conclude that the present study is the first to develop and compare Support Vector Machine (SVM), Neural Networks, k-Nearest Neighbor and regression analysis to predict the future performance of emergency residents. In the following sections we describe how we configured and used these methods.

3.3.1 Support Vector Machine (SVM)

The SVM uses a kernel function to transform finite input space to higher or infinite spaces (Cortes & Vapnik, 1995). In this study, we investigated multiple kernels such as the radial basis function (RBF), polynomial, linear, and quadratic. The best results were achieved by the polynomial kernel function which defined in Eq. 3.2. The polynomial kernel has two parameters: the penalty constant C and polynomial degree d. In the polynomial kernel function, C is chosen to be 1 by default, and we need to optimize d.

$$k(x, x_i) = (x^T x_i + C)^d \tag{3.2}$$

3.3.2 Neural Network

In order to select an appropriate network topology, different topologies such as multilayer perceptrons, recurrent networks, and time-lagged recurrent networks were considered. Due to the nature of our data, which is static and not sufficiently large to enable the use of complex topologies, the multilayer perceptron was selected (Hornik, Stinchcombe, & White, 1989). The neural network structure selected for this study consists of six input nodes, one hidden layer with ten nodes and one output node. One hidden layer was selected because a large number of hidden layers will progressively slow down the training time.

3.3.3 k-Nearest Neighbor (kNN)

kNN is a non-parametric classification algorithm. The model of the kNN classifier is based on feature vectors and class labels from the training data set. This classifier induces the class of the query vector from the labels of the feature vectors in the training data set to which the query vector is similar. To check similarity in a multidimensional feature space, there are different metrics such as: Euclidean, City-block, Correlation, Minkowski, Chebychev and Jaccard. In this thesis, the best results were achieved using the Euclidean distance metric.

The three MLAs were implemented in MATLAB in order to predict resident performance. The Logistic regression analysis was also used as a separate benchmark as previous work in medical education has shown promising results with correlation and regression (Hamdy et al., 2006). The objective was to determine which algorithm is most appropriate to predict residents' performance accurately, and also in which case it could be useful as an educational supporting tool for instructors.

3.4 Output

The model was trained with data gathered from residents who are "at risk" (Class 1) and "not at risk" (Class 0) for their performance. Class 1 residents are identified as needing further attention from their supervisors in order to become better focused on their work. Class 0 residents are those that show a positive trend in their assessments. The MLAs are used to classify the future performance of residents in a specific group. The MLAs have a binary output (0,1), which is interpreted and displayed to the user.

Chapter 4

Results and Evaluation

In this chapter, we describe the experimental setup and discuss the results of two experiments: 1) task-level performance experiment which requires the model to predict the future performance of a resident in a specific group of tasks, and 2) a block-level performance experiment where the model predicts the next block average of scores for a resident based on the features extracted from the current block.

4.1 Experimental setup

In all stages of this study, a desktop computer with Intel Core2 Quad CPU Q8400 @ 2.66GHz processor and 4GB of installed Memory was used. Operating system was Windows 10 Professional 64bit. Feature extraction, feature selection (Neighborhood Component Analysis) and MLAs were implemented and evaluated with the machine learning and bioinformatics toolbox in MATLAB R2017a. The evaluation method was k-fold cross-validation (Kohavi, 1995). In k-fold cross validation the data set is divided into k subsets, and the training-testing phase is repeated k times. Each time, one of the k subsets is used as the test set and the other $k-1$ subsets are put together to form a training set. Each time, the model is trained using the training set only. Then the model was asked to predict the output values for the data in the testing set. Finally, the average error across all k trials was computed. Diagnostic accuracy relates to the ability of a test to discriminate between the labeled classes and model output. This discriminative potential can be quantified by the measures of diagnostic accuracy such as sensitivity, specificity, and the Receiver Operating Characteristic (ROC) curve, which is usually expressed as the area under the curve (AUC) (Pencina, D'Agostino, & Vasan, 2008).

Sensitivity and specificity are statistical measures of the performance of a binary classification test. In this study, sensitivity (also called the true positive rate) measures the proportion of "at risk" residents that are correctly identified and specificity (also called the true negative rate) measures the proportion of "not at risk" residents that are correctly identified. A receiver operating characteristic curve (ROC) is a graphical plot that illustrates the accuracy of a binary classifier. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR). In this study, the accuracy of models is measured by the area under the ROC curve (AUC). The AUC is used in classification analysis in order to determine which of the used models predicts the classes best. The AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example (J. Huang & Ling, 2005). An area of 1 represents a perfect test; an area of 0.5 represents a worthless test because in a binary classification there is a 50 percent chance of true classification by just randomly selecting the classes (Hanley & McNeil, 1982).

4.2 Task-level performance experiment

In this experiment, the model is asked to predict the future performance of a resident in a specific group of tasks based on the features extracted from the current situation of a resident. Results are displayed in Table 4.1. In order to make sure about the usability of feature selection phase we tested this experiment without feature selection phase and results, as illustrated in Appendix B, showed that when we used all features the total accuracy decreased.

Table 4.1: Results of task-level performance experiment

Model	Sensitivity	Specificity	AUC
SVM	0.54	0.74	0.64
kNN	0.30	0.84	0.57
Neural Network	0.35	0.86	0.61
Logistic regression	0.43	0.24	0.34

In order to further increase the accuracy of the SVM based model we conducted a grid search to find the optimal degree for the polynomial kernel and the best k value for separating data into training and testing subsets. The best result was achieved with $k = 5$ and $degree = 5$. As shown in Figure 4.1 sensitivity was 0.54, specificity was 0.74 and area under curve was 0.64. In the kNN based model, results of running a grid search for finding the optimal combination of the k for k-fold and number of neighbors showed that the best results achieved on 5-fold cross validation with 2-nearest neighbors. As shown in Figure 4.2 sensitivity was 0.30, specificity was 0.84 and AUC was 0.57. Running k-fold cross-validation for neural network showed that the best k was 5. As shown in Figure 4.3 sensitivity was 0.35, specificity was 0.86 and AUC was 0.61. Results of running logistic regression analysis on the available data showed that this method was weaker than machine learning algorithms. As shown in Figure 4.4 for logistic regression, sensitivity was 0.43, specificity was 0.24 and area under curve was 0.34.

4.3 Block-level performance experiment

The current method of identifying residents-at-risk in McMAP is described below. At the end of each block (usually around 28 days), data from the daily faculty member observations are compiled into a report card of scores. A comparison of

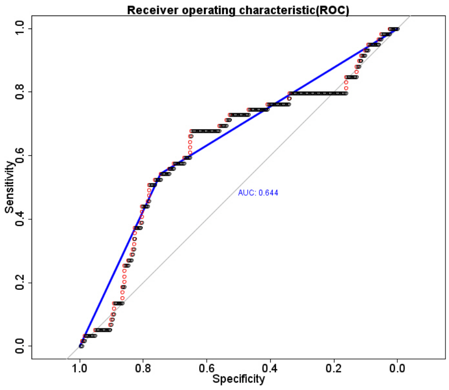


Figure 4.1: Exp1. ROC of SVM

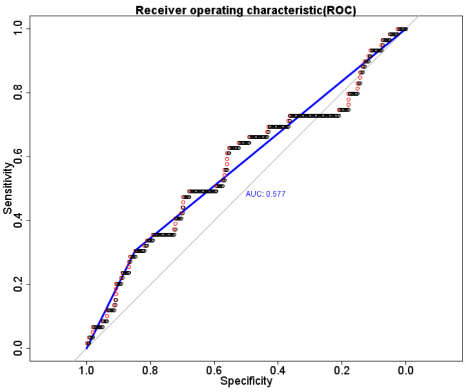


Figure 4.2: Exp1. ROC of kNN

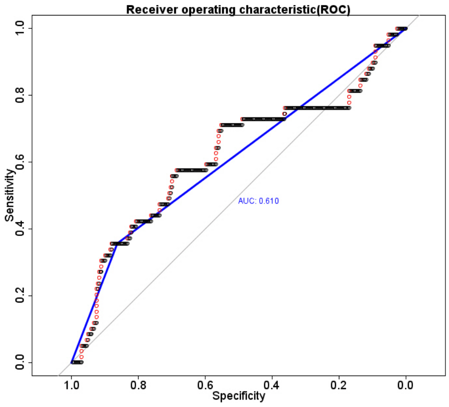


Figure 4.3: Exp1. ROC of Neural Network

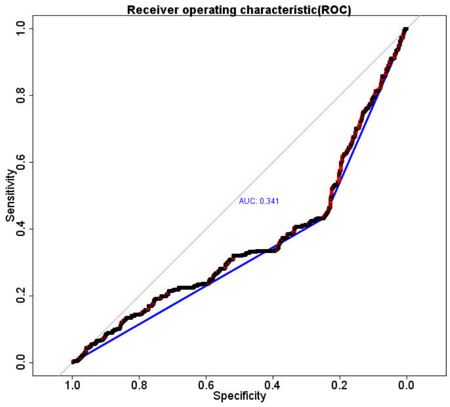


Figure 4.4: Exp1. ROC of Logistic Regression

the average scores a resident receives in a block is made against a threshold defined by the program's competency committee (CC), Those who receive a score below the threshold are flagged for review, and considered a "resident-at-risk". The current threshold in McMaster's emergency residency program is 6.

In this experiment, we evaluated the model against the current method of identifying "resident-at-risk" in McMAP and predicted the next block average class for each resident. To do that, the model trained with features extracted from task scores for each block. The label 1 means that the average of scores in next block is under six and the label 0 means that the average is six or seven. Table 4.2 displays the results of this experiment.

Table 4.2: Results of block-level performance experiment

Model	Sensitivity	Specificity	AUC
SVM	0.50	0.76	0.63
kNN	0.42	0.83	0.61
Neural Network	0.49	0.69	0.54
Logistic Regression	0.17	0.97	0.57

Results of running Neighborhood Component Analysis (NCA) showed that all features are participating in the training and testing phase. Also, results of running k-fold cross-validation showed that the best results were achieved on $k = 5$. Results of running SVM as shown in Figure 4.5 revealed that SVM was the most accurate method in predicting future average class. Tabel 4.2 presents the SVM results with sensitivity of 0.50 and specificity of 0.76 and 0.63 area under curve. Figure 4.6 shows the results of kNN based model with sensitivity of 0.42 and specificity of 0.83 and 0.61 area under curve. Figure 4.7 shows the results of neural network which is 0.49 sensitivity, 0.69 specificity and 0.54 area under the curve. Figure 4.8 shows the results of Logistic regression which is 0.17 sensitivity, 0.97 specificity and 0.57 area under the curve.

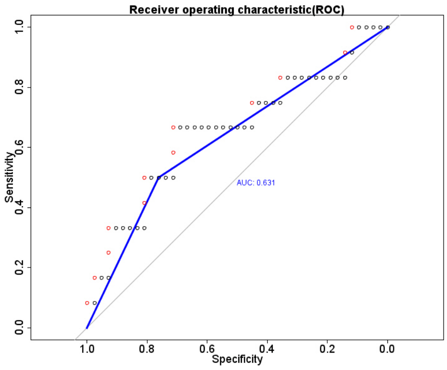


Figure 4.5: Exp2. ROC of SVM

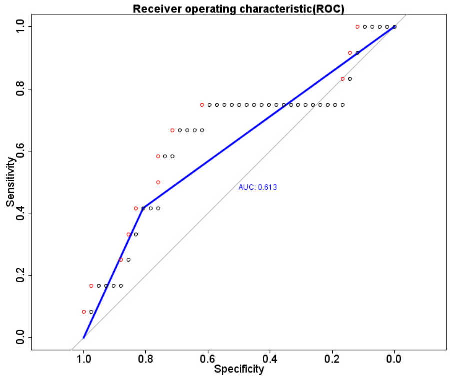


Figure 4.6: Exp2. ROC of kNN

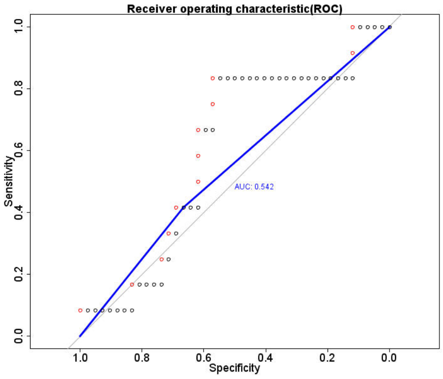


Figure 4.7: Exp2. ROC of Neural Network

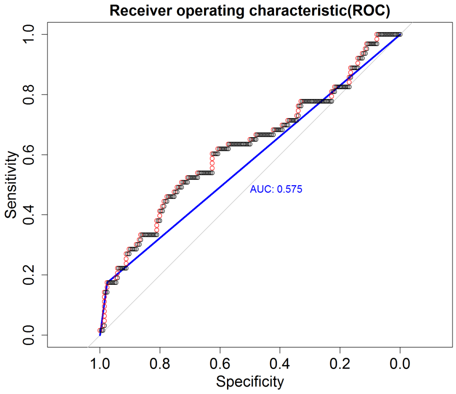


Figure 4.8: Exp2. ROC of Logistic regression

Chapter 5

Conclusions and Future Work

This chapter provides conclusions of the results. Also, a number of directions for future work is suggested.

5.1 Conclusions

In this study, we proposed a model for predicting future performance of emergency residents based on their past performance. We extracted features from different sequences of scores for each resident and combined all of them in one dataset then applied the NCA feature selection algorithm for dimension reduction and selecting the best combination of features which increase the accuracy of the model then trained multiple MLAs for making decision about each set of features. Results showed that the SVM based model was the most accurate method in successfully identifying residents “at risk” in a specific group of tasks with 0.54 sensitivity, 0.74 specificity and 0.64 area under curve. In block-level prediction also the SVM was the most accurate method with 0.50 sensitivity, 0.76 specificity and 0.63 area under curve.

5.2 Future Work

In this study, we were facing different kinds of limitations. One of the most important limitations was inter-rater reliability. It is well known, that when people evaluate someone, their evaluations reflect the person being assessed, and the assessor's built in biases. As human beings, our judgments about many things are affected by our own perceptual ideas. The effects of assessing teacher's perceptions introduces highly subjective factors that make many evaluations unreliable or inconsistent. Rater effect is a major problem in medical education, because there are idiosyncrasies to observations (Gingerich, Regehr, & Eva, 2011; Gingerich, van der Vleuten, Eva, & Regehr, 2014; Gingerich, Kogan, Yeates, Govaerts, & Holmboe, 2014). When faculty members rate residents using scales that are vague or holistic, this results in unreliable data. Undoubtedly, McMAP is subject to these previously described biases. The rater "problem" can be tackled by a number of different methods ranging from increased assessor training to better behavioural anchors, which can align the assessor's thinking to create shared mental models (Gingerich, Kogan, et al., 2014). Recent work in the same group of emergency physician teachers has shown that the assessors themselves have specific rating tendencies (i.e. some are more lenient [doves] and some are harsher [hawks]). An interesting future study might be to add the assessors' personality into a MLA algorithm and see if this provides another data point that can increase the accuracy of the algorithms in predicting resident performance. Just as we can predict resident performance, we anticipate that there are similar patterns in assessor behaviors that can also be detected using MLAs.

Yet another opportunity is the addition of qualitative data. The McMAP system asks the rater to describe the resident's performance or score using free-text comments that may offers qualitative input. Recent work by Ginsburg and colleagues has shown that human readers using qualitative comments can reliably sort through residents (Ginsburg, van der Vleuten, Eva, & Lingard, 2016). Another future direction would be to add qualitative comments as sources of data input for the MLAs.

Finally, multi class MLAs could be tested to identify not only residents at risk but also find the talented residents. Identifying talented residents helps the program to have a better focus on residents who are in need of more challenging educational programs leading a resident to be certified and enter the workforce as an unsupervised doctor earlier.

References

Al Shalabi, L., Shaaban, Z., & Kasasbeh, B. (2006). Data mining: A preprocessing engine. *Journal of Computer Science*, *2*(9), 735–739.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., & Bengio, Y. (2016). End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4945–4949).

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? In *International conference on database theory* (pp. 217–235).

Bolstad, B. M., Irizarry, R. A., Åstrand, M., & Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, *19*(2), 185–193.

Boyer, B., Mitton, T., & Vorkink, K. (2009). Expected idiosyncratic skewness. *The Review of Financial Studies*, *23*(1), 169–202.

Brenner, A. M., Mathai, S., Jain, S., & Mohl, P. C. (2010). Can we predict "problem residents"? *Academic Medicine*, *85*(7), 1147–1151.

Burger, W., & Burge, M. J. (2016). *Digital image processing: an algorithmic introduction using java.* Springer.

Calvo-Flores, M. D., Galindo, E. G., Jiménez, M. P., & Piñeiro, O. P. (2006). Predicting students' marks from moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, *1*(1), 586–590.

Chan, T., & Sherbino, J. (2015). The mcmaster modular assessment program (McMAP): A theoretically grounded work-based assessment system for an emergency medicine residency program. *Academic Medicine*, *90*(7), 900–905.

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, *37*(1), 51–89.

Corrigan, O., Smeaton, A. F., Glynn, M., & Smyth, S. (2015). Using educational analytics to improve test performance. In *Design for teaching and learning in a networked world* (pp. 42–55). Springer.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. doi: 10.1007/BF00994018

Crawford, L., Tindal, G., & Stieber, S. (2001). Using oral reading rate to predict student performance on statewide achievement tests. *Educational Assessment*, *7*(4), 303–323.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on world wide web* (pp. 519–528).

Davis, D. A., Mazmanian, P. E., Fordis, M., Van Harrison, R., Thorpe, K. E., & Perrier, L. (2006). Accuracy of physician self-assessment compared with observed measures of competence: a systematic review. *Journal of the American Medical Association (JAMA)*, *296*(9), 1094–1102.

Derderian, C. A., & Kenkel, J. M. (2016). Remediation as a road to competency: strategies for early identification of the struggling resident and generating the remediation plan. *Journal of Craniofacial Surgery*, *27*(1), 8–12.

Elfenbein, D. M., Sippel, R. S., McDonald, R., Watson, T., Scarborough, J. E., & Migaly, J. (2015). Faculty evaluations of resident medical knowledge: can they be used to predict american board of surgery in-training examination performance? *The American Journal of Surgery*, *209*(6), 1095–1101.

Eva, K. W., & Regehr, G. (2008). "I'll never play professional football" and other fallacies of self-assessment. *Journal of Continuing Education in the Health Professions*, *28*(1), 14–19.

Frank, J. R., & Danoff, D. (2007). The canmeds initiative: implementing an outcomes-based framework of physician competencies. *Medical teacher*, *29*(7), 642–647.

Frank, J. R., Snell, L. S., Cate, O. T., Holmboe, E. S., Carraccio, C., Swing, S. R., Harris, P., Glasgow, N. J., Campbell, C., & Dath, D. (2010). Competency-based medical education: theory to practice. *Medical teacher*, *32*(8), 638–645.

Gingerich, A., Kogan, J., Yeates, P., Govaerts, M., & Holmboe, E. (2014). Seeing the 'black box' differently: assessor cognition from three research perspectives. *Medical education*, *48*(11), 1055–1068.

Gingerich, A., Regehr, G., & Eva, K. W. (2011). Rater-based assessments as social judgments: rethinking the etiology of rater errors. *Academic Medicine*, *86*(10), S1–S7.

Gingerich, A., van der Vleuten, C. P., Eva, K. W., & Regehr, G. (2014). More consensus than idiosyncrasy: Categorizing social judgments to examine variability in mini-cex ratings. *Academic Medicine*, *89*(11), 1510–1519.

Ginsburg, S., van der Vleuten, C., Eva, K. W., & Lingard, L. (2016). Hedging to save face: a linguistic analysis of written comments on in-training evaluation reports. *Advances in Health Sciences Education*, *21*(1), 175–188.

Goldberger, E. H., T. Roweis, & Salakhutdinov, R. (2005). Neighbourhood components analysis. *Advances in Neural Information Processing Systems*, *17*, 513-520.

Gomes, L. (2014, October). *Machine-learning maestro Michael Jordan on the delusions of big data and other huge engineering efforts.* Retrieved from http://spectrum.ieee.org/robotics/artificial-intelligence/machinelearning-maestro-michael-jordan-on-the-delusions-of-big-data-and-other-huge-engineering-efforts

Greenberg, C. S., Bansé, D., Doddington, G. R., Garcia-Romero, D., Godfrey, J. J., Kinnunen, T., Martin, A. F., McCree, A., Przybocki, M., & Reynolds, D. A. (2014). The National Institute of Standards and Technology (NIST) 2014 speaker recognition i-vector machine learning challenge. In *Odyssey: The speaker and language recognition workshop* (pp. 224–230).

Hamdy, H., Prasad, K., Anderson, M. B., Scherpbier, A., Williams, R., Zwierstra, R., & Cuddihy, H. (2006). Beme systematic review: predictive values of measurements obtained in medical schools and future performance in medical practice. *Medical teacher*, *28*(2), 103–116.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29–36.

Hayden, S. R., Hayden, M., & Gamst, A. (2005). What characteristics of applicants to emergency medicine residency programs predict future success as an emergency medicine resident? *Academic emergency medicine*, *12*(3), 206–210.

Haykin, S. (1998). *Neural networks: a comprehensive foundation* (Second ed.). New Jersey, USA: Prentice Hall PTR Upper Saddle River.

Haykin, S. (2009). *Neural networks and learning machines* (Third ed.). New Jersey: Pearson. Retrieved from https://cours.etsmtl.ca/sys843/REFS/Books/ebook_Haykin09.pdf

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., & Sainath, T. N. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Holmboe, E. S., Sherbino, J., Long, D. M., Swing, S. R., & Frank, J. R. (2010). The role of assessment in competency-based medical education. *Medical teacher*, *32*(8), 676–682.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359–366.

Huang, J., & Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, *17*(3), 299–310.

Huang, S., & Fang, N. (2013). Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. *Computers & Education*, *61*, 133–145.

Iobst, W. F., Sherbino, J., Cate, O. T., Richardson, D. L., Dath, D., Swing, S. R., Harris, P., Mungroo, R., Holmboe, E. S., & Frank, J. R. (2010). Competency-based medical education in postgraduate medical education. *Medical teacher*, *32*(8), 651–656.

Joachims, T. (1998a). *Making large-scale svm learning practical* (Tech. Rep.). https://www.econstor.eu/handle/10419/77178: Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

Joachims, T. (1998b). Text categorization with support vector machines: Learning with many relevant features. *European Conference on Machine Learning (ECML)*, 137–142.

Karlik, B., & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, *1*(4), 111–122.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)* (Vol. 14, pp. 1137–1145).

Kotsiantis, S., Patriarcheas, K., & Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, *23*(6), 529–535.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). *Supervised machine learning: A review of classification techniques.*

Larose, D. T. (2005). k-nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, 90–106.

Lindblad, T., Kinser, J. M., Lindblad, T., & Kinser, J. (1998). *Image processing using pulse-coupled neural networks.* Springer.

Lykourentzou, I., Giannoukos, I., Mpardis, G., Nikolopoulos, V., & Loumos, V. (2009). Early and dynamic student achievement prediction in e-learning courses using neural networks. *Journal of the American Society for Information Science and Technology*, *60*(2), 372–380.

Mardia, K. V. (1974). Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies. *Sankhyā: The Indian Journal of Statistics, Series B*, *36*(2), 115–128.

Metro, D. G., Talarico, J. F., Patel, R. M., & Wetmore, A. L. (2005). The resident application process and its correlation to future performance as a resident. *Anesthesia & Analgesia*, *100*(2), 502–505.

Parker, R. W., Alford, C., & Passmore, C. (2004). Can family medicine residents predict their performance on the in-training examination? *Family Medicine Kansas city*, *36*(10), 705–709.

Pencina, M. J., D'Agostino, R. B., & Vasan, R. S. (2008). Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond. *Statistics in medicine*, *27*(2), 157–172.

Regehr, G., & Eva, K. (2006). Self-assessment, self-direction, and the self-regulating professional. *Clinical orthopaedics and related research*, *449*, 34–38.

Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., & Ventura, S. (2013). Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, *21*(1), 135–146.

Romero, C., Ventura, S., & García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, *51*(1), 368–384.

Russ, J. C. (2016). *The image processing handbook.* CRC press.

Shim, J., Koh, J., Fister, S., & Seo, H. (2016). Phonetic analytics technology and big data: real-world cases. *Communications of the ACM*, *59*(2), 84–90.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., & Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*(1), 116–124.

Singh-Miller, N., Collins, M., & Hazen, T. J. (2007). Dimensionality reduction for speech recognition using neighborhood components analysis. In *Eighth annual conference of the international speech communication association.*

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, *9*(3), 293–300.

Taylor, R. (1990). Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*, *6*(1), 35–39.

Wallenstein, J., Heron, S., Santen, S., Shayne, P., & Ander, D. (2010). A core competency–based objective structured clinical examination (osce) can predict future resident performance. *Academic Emergency Medicine*, *17*(s2), S67–S71.

Yamauchi, T. (2005). Labeling bias and categorical induction: generative aspects of category information. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*(3), 538.

Yegnanarayana, B. (2009). *Artificial neural networks.* Prentice Hall of India Private Limited.

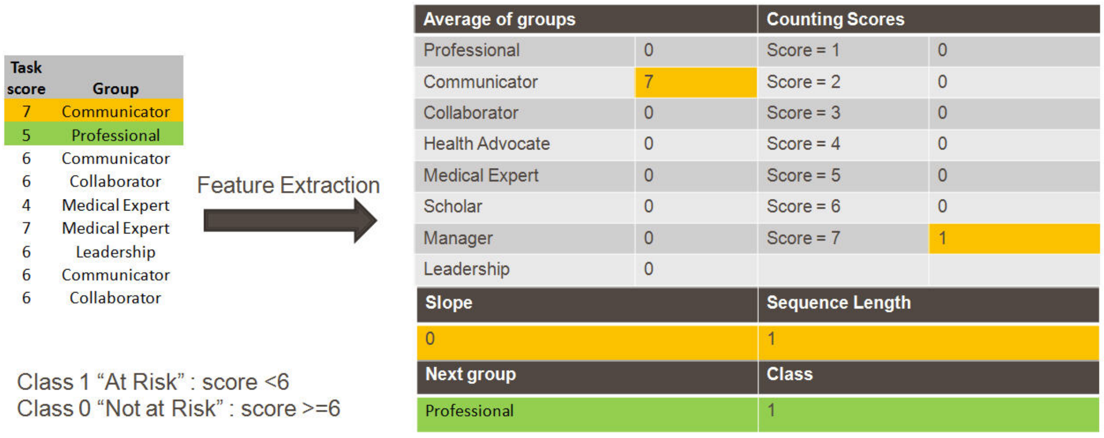
Yost, M. J., Gardner, J., Bell, R. M., Fann, S. A., Lisk, J. R., Cheadle, W. G., Goldman, M. H., Rawn, S., Weigelt, J. A., & Termuhlen, P. M. (2015). Predicting academic performance in surgical training. *Journal of surgical education*, *72*(3), 491–499.

Appendix A

Stepwise feature extraction

Images below show that how feature extraction function step by step extracts the 18 different features from a sample sequence with 9 scores.

In this step, average of communicator task scores is updated because the first score belongs to communicator group. The number of sevens is updated to 1. Slope is 0 and length of sequence is 1. Next task belongs to professional group and the calculated class is 1 because next score is less than 6.



Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Feature Extraction

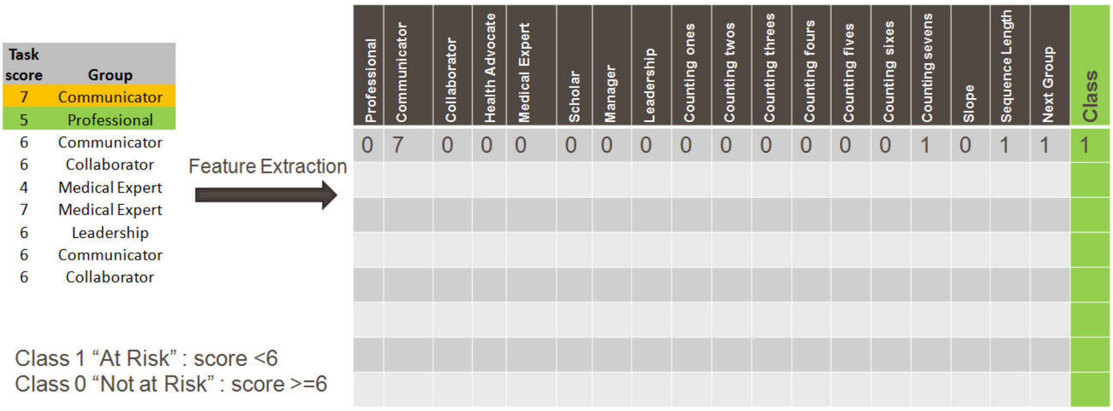
Average of groups		Counting Scores	
Professional	0	Score = 1	0
Communicator	7	Score = 2	0
Collaborator	0	Score = 3	0
Health Advocate	0	Score = 4	0
Medical Expert	0	Score = 5	0
Scholar	0	Score = 6	0
Manager	0	Score = 7	1
Leadership	0		

Slope	Sequence Length
0	1

Next group	Class
Professional	1

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Then, the feature extraction function adds the calculated features to feature dataset.



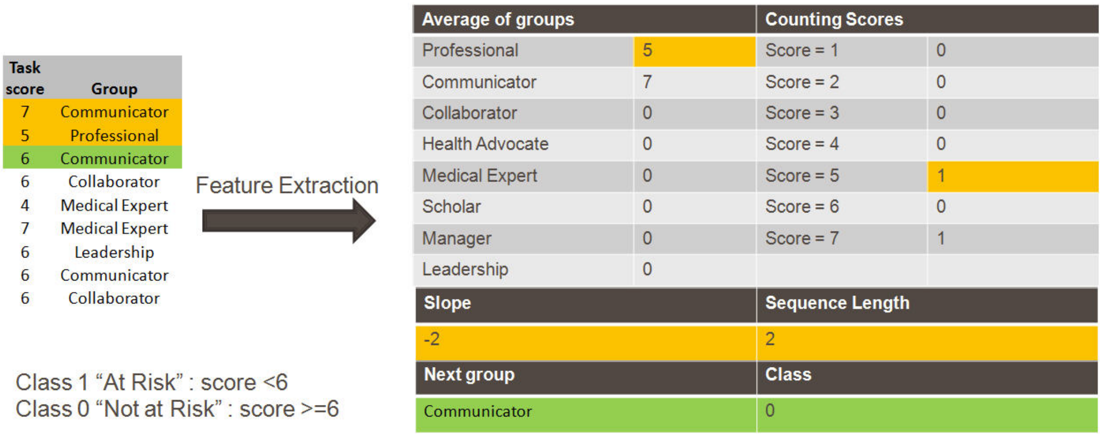
Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Feature Extraction

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

In this step, average of professional task scores is updated to 5. The number of fives is updated to 1. Slope is -2 and length of sequence is 2. Next task belongs to communicator group and the calculated class is 0 because next score is 6.



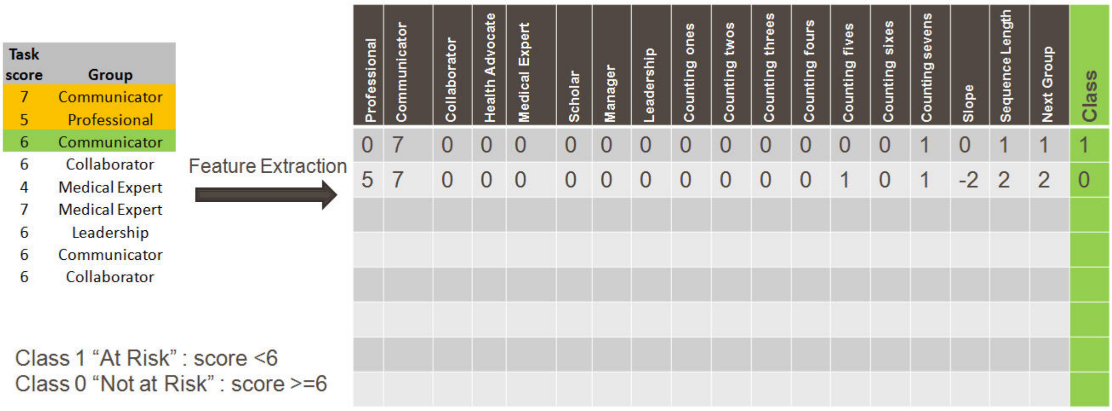
Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Feature Extraction

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Average of groups		Counting Scores	
Professional	5	Score = 1	0
Communicator	7	Score = 2	0
Collaborator	0	Score = 3	0
Health Advocate	0	Score = 4	0
Medical Expert	0	Score = 5	1
Scholar	0	Score = 6	0
Manager	0	Score = 7	1
Leadership	0		
Slope		**Sequence Length**	
-2		2	
Next group		**Class**	
Communicator		0	

Then, the feature extraction function adds the calculated features to feature dataset.



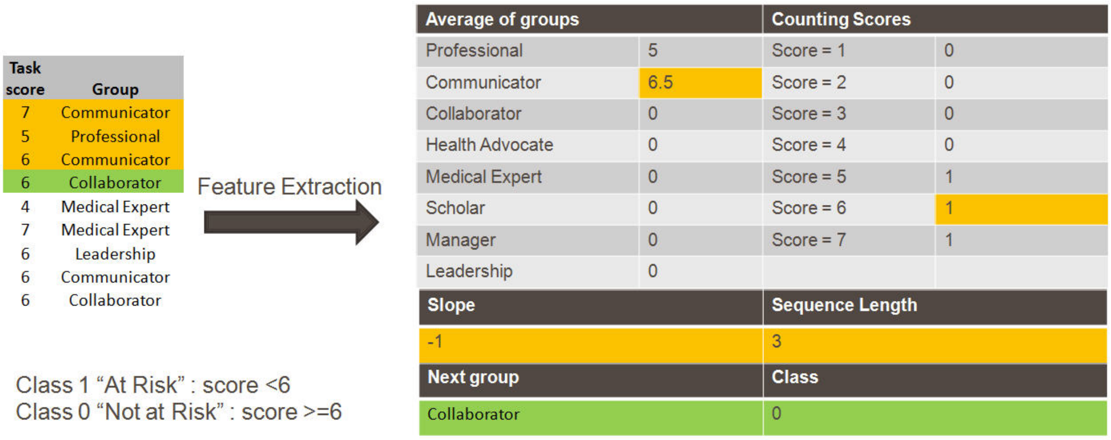
Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Feature Extraction

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0

In this step, average of communicator task scores is updated to 6.5. The number of sixes is updated to 1. Slope is updated to -1 and length of sequence is updated to 3. Next task belongs to collaborator group and the calculated class is 0 because next score is 6.



Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

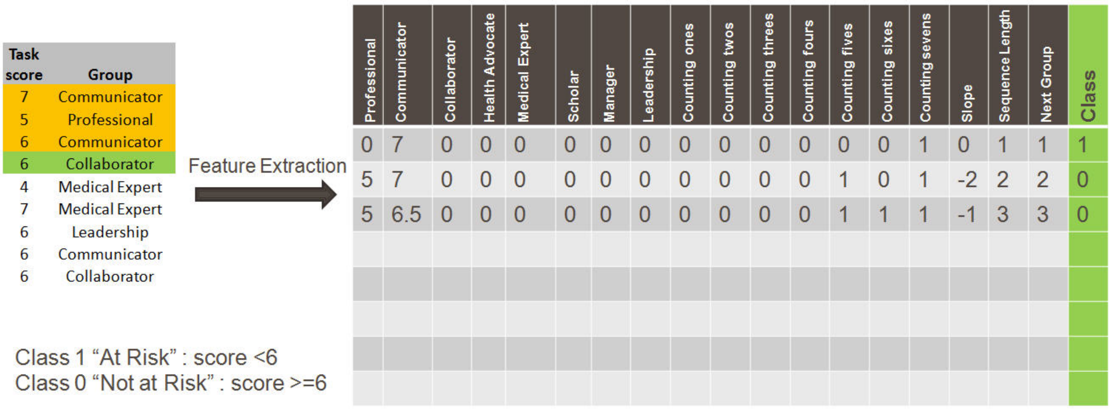
Average of groups		Counting Scores	
Professional	5	Score = 1	0
Communicator	6.5	Score = 2	0
Collaborator	0	Score = 3	0
Health Advocate	0	Score = 4	0
Medical Expert	0	Score = 5	1
Scholar	0	Score = 6	1
Manager	0	Score = 7	1
Leadership	0		

Slope	Sequence Length
-1	3

Next group	Class
Collaborator	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Then, the feature extraction function adds the calculated features to feature dataset.

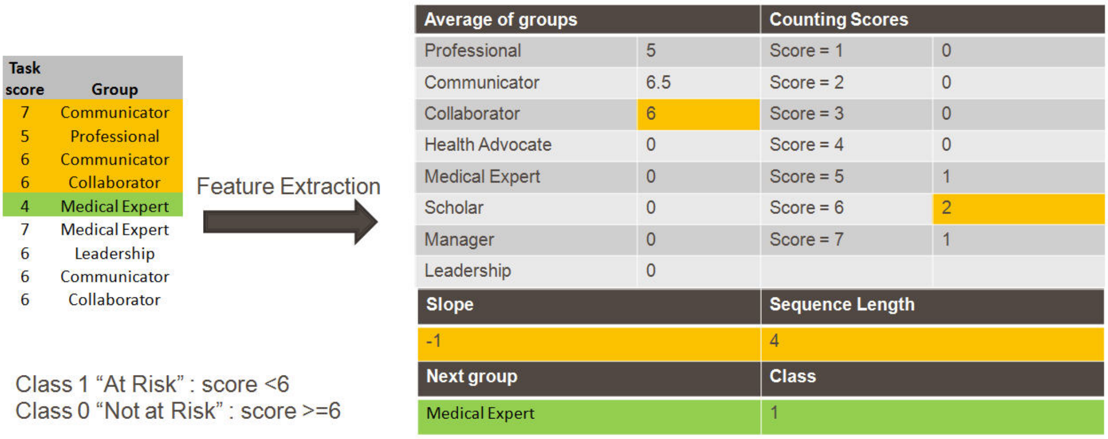


Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0
5	6.5	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	3	3	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

In the next step, average of collaborator task scores is updated to 6. The number of sixes is updated to 2. Slope is same and length of sequence is updated to 4. Next task belongs to medical expert group and the calculated class is 1 because next score is less than 6.

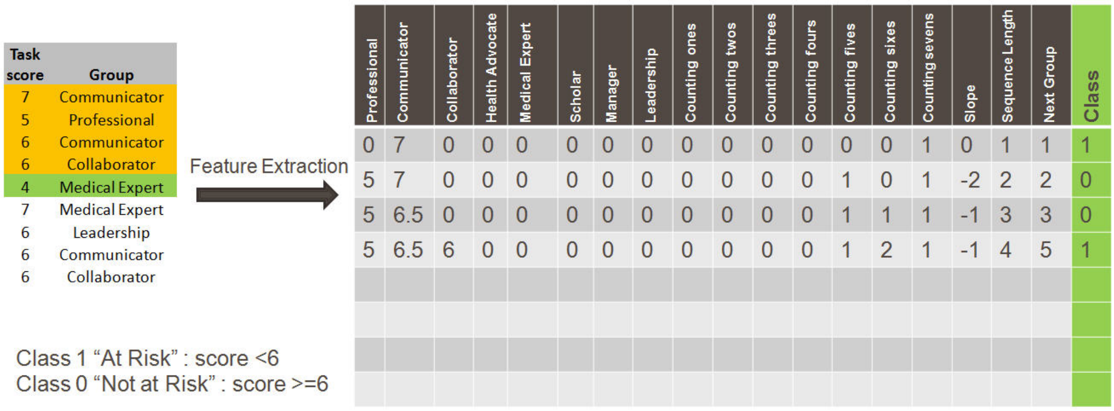


Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Average of groups		Counting Scores	
Professional	5	Score = 1	0
Communicator	6.5	Score = 2	0
Collaborator	6	Score = 3	0
Health Advocate	0	Score = 4	0
Medical Expert	0	Score = 5	1
Scholar	0	Score = 6	2
Manager	0	Score = 7	1
Leadership	0		
Slope		**Sequence Length**	
-1		4	
Next group		**Class**	
Medical Expert		1	

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Then, the feature extraction function adds the calculated features to feature dataset.

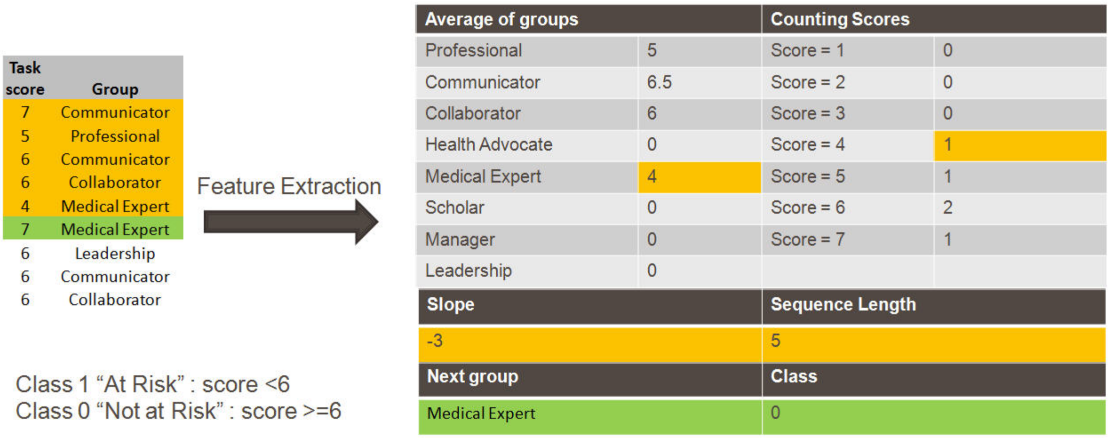


Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0
5	6.5	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	3	3	0
5	6.5	6	0	0	0	0	0	0	0	0	0	1	2	1	-1	4	5	1

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

In the next step, average of medical expert task scores is updated to 4. The number of fours is updated to 1. Slope is updated to -3 and length of sequence is updated to 5. Next task belongs to medical expert group and the calculated class is 0 because next score is 7.



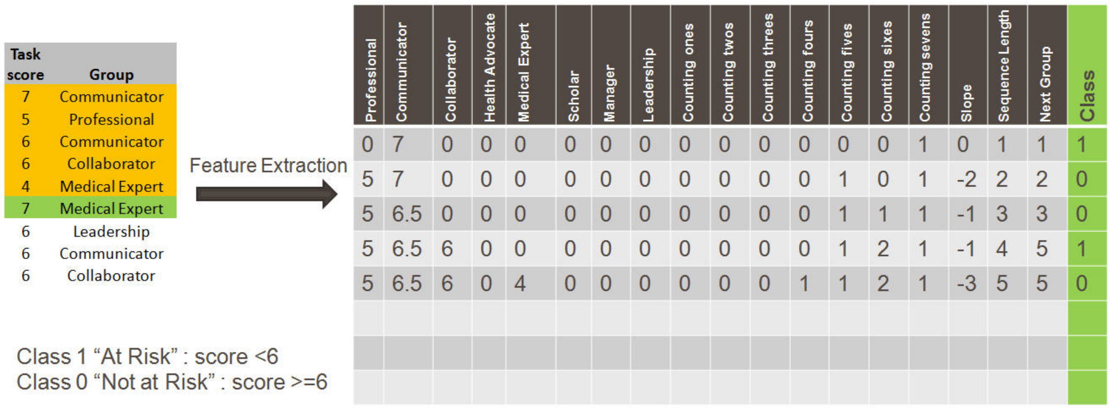
Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Feature Extraction

Average of groups		Counting Scores	
Professional	5	Score = 1	0
Communicator	6.5	Score = 2	0
Collaborator	6	Score = 3	0
Health Advocate	0	Score = 4	1
Medical Expert	4	Score = 5	1
Scholar	0	Score = 6	2
Manager	0	Score = 7	1
Leadership	0		
Slope		**Sequence Length**	
-3		5	
Next group		**Class**	
Medical Expert		0	

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Then, the feature extraction function adds the calculated features to feature dataset.



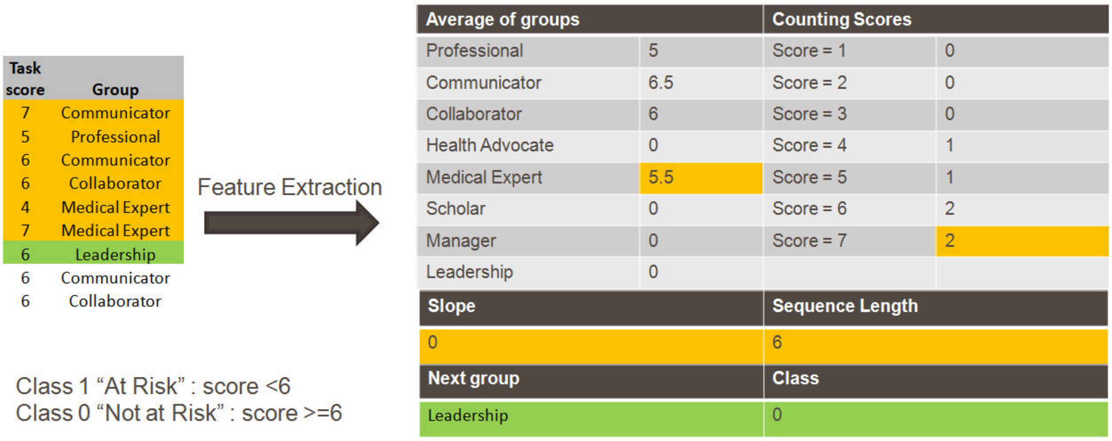
Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Feature Extraction

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0
5	6.5	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	3	3	0
5	6.5	6	0	0	0	0	0	0	0	0	0	1	2	1	-1	4	5	1
5	6.5	6	0	4	0	0	0	0	0	0	1	1	2	1	-3	5	5	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

In the next step, average of medical expert task scores is updated to 5.5. The number of sevens is updated to 2. Slope is updated to 0 and length of sequence is updated to 6. Next task belongs to leadership group and the calculated class is 0 because next score is 6.



Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

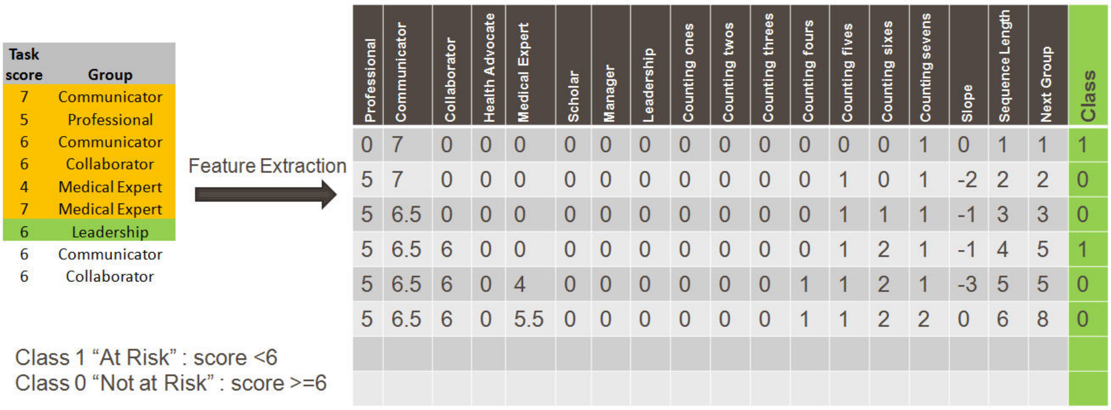
Average of groups		Counting Scores	
Professional	5	Score = 1	0
Communicator	6.5	Score = 2	0
Collaborator	6	Score = 3	0
Health Advocate	0	Score = 4	1
Medical Expert	5.5	Score = 5	1
Scholar	0	Score = 6	2
Manager	0	Score = 7	2
Leadership	0		

Slope	Sequence Length
0	6

Next group	Class
Leadership	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Then, the feature extraction function adds the calculated features to feature dataset.

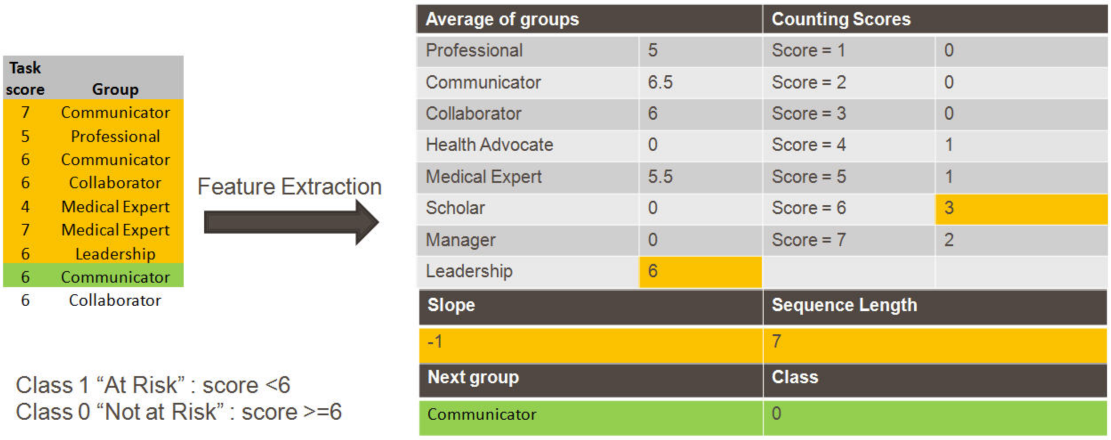


Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0
5	6.5	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	3	3	0
5	6.5	6	0	0	0	0	0	0	0	0	0	1	2	1	-1	4	5	1
5	6.5	6	0	4	0	0	0	0	0	0	1	1	2	1	-3	5	5	0
5	6.5	6	0	5.5	0	0	0	0	0	0	1	1	2	2	0	6	8	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

In the next step, average of leadership task scores is updated to 6. The number of sixes is updated to 3. Slope is updated to -1 and length of sequence is updated to 7. Next task belongs to communicator group and the calculated class is 0 because next score is 6.

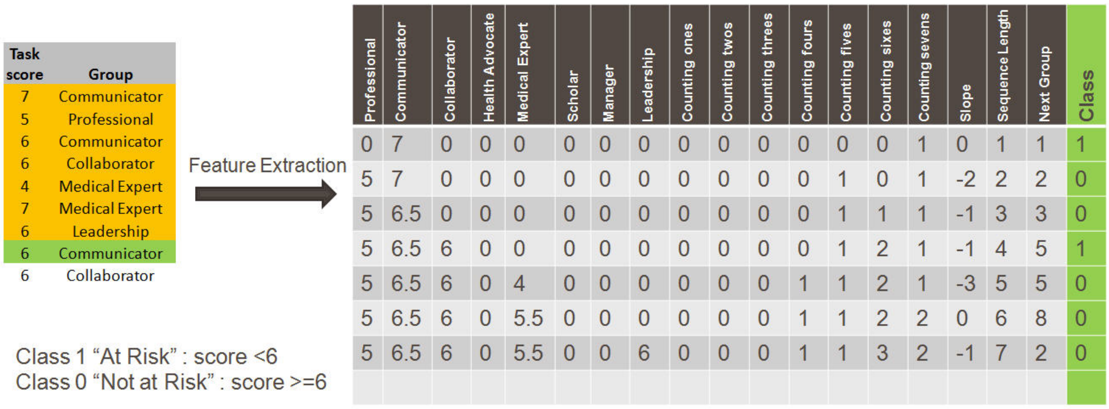


Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Average of groups		Counting Scores	
Professional	5	Score = 1	0
Communicator	6.5	Score = 2	0
Collaborator	6	Score = 3	0
Health Advocate	0	Score = 4	1
Medical Expert	5.5	Score = 5	1
Scholar	0	Score = 6	3
Manager	0	Score = 7	2
Leadership	6		
Slope		**Sequence Length**	
-1		7	
Next group		**Class**	
Communicator		0	

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Then, the feature extraction function adds the calculated features to feature dataset.

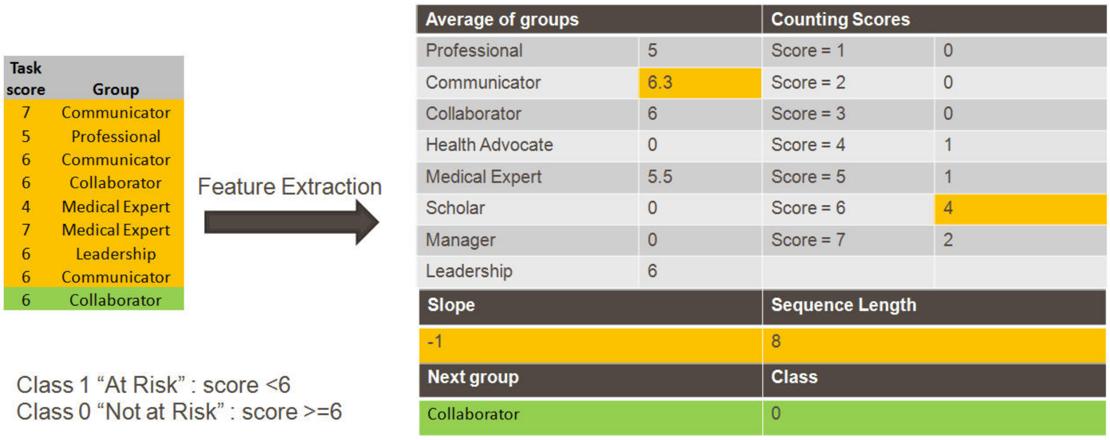


Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0
5	6.5	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	3	3	0
5	6.5	6	0	0	0	0	0	0	0	0	0	1	2	1	-1	4	5	1
5	6.5	6	0	4	0	0	0	0	0	0	1	1	2	1	-3	5	5	0
5	6.5	6	0	5.5	0	0	0	0	0	0	1	1	2	2	0	6	8	0
5	6.5	6	0	5.5	0	0	6	0	0	0	1	1	3	2	-1	7	2	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

In the next step, average of communicator task scores is updated to 6.3. The number of sixes is updated to 4. Slope is updated to -1 and length of sequence is updated to 8. Next task belongs to collaborator group and the calculated class is 0 because next score is 6.



Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

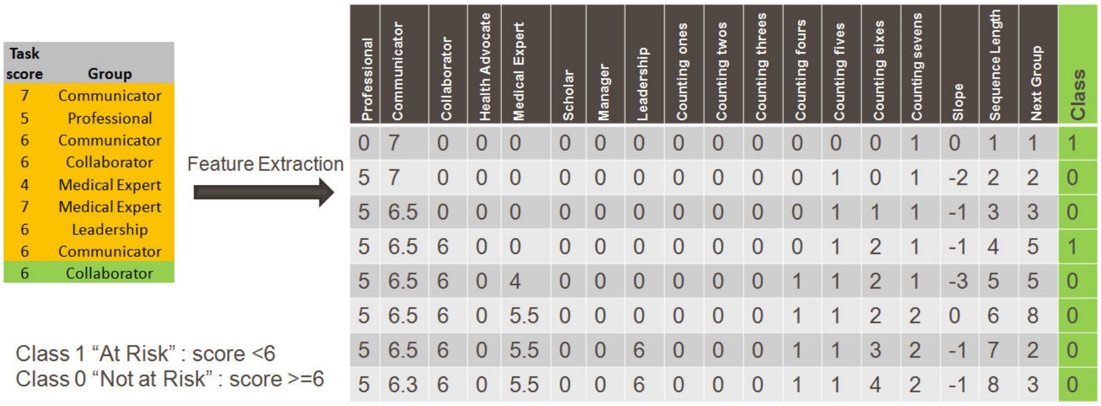
Average of groups		Counting Scores	
Professional	5	Score = 1	0
Communicator	6.3	Score = 2	0
Collaborator	6	Score = 3	0
Health Advocate	0	Score = 4	1
Medical Expert	5.5	Score = 5	1
Scholar	0	Score = 6	4
Manager	0	Score = 7	2
Leadership	6		

Slope	Sequence Length
-1	8

Next group	Class
Collaborator	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Then, the feature extraction function adds the calculated features to feature dataset.



Task score	Group
7	Communicator
5	Professional
6	Communicator
6	Collaborator
4	Medical Expert
7	Medical Expert
6	Leadership
6	Communicator
6	Collaborator

Professional	Communicator	Collaborator	Health Advocate	Medical Expert	Scholar	Manager	Leadership	Counting ones	Counting twos	Counting threes	Counting fours	Counting fives	Counting sixes	Counting sevens	Slope	Sequence Length	Next Group	Class
0	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
5	7	0	0	0	0	0	0	0	0	0	0	1	0	1	-2	2	2	0
5	6.5	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	3	3	0
5	6.5	6	0	0	0	0	0	0	0	0	0	1	2	1	-1	4	5	1
5	6.5	6	0	4	0	0	0	0	0	0	1	1	2	1	-3	5	5	0
5	6.5	6	0	5.5	0	0	0	0	0	0	1	1	2	2	0	6	8	0
5	6.5	6	0	5.5	0	0	6	0	0	0	1	1	3	2	-1	7	2	0
5	6.3	6	0	5.5	0	0	6	0	0	0	1	1	4	2	-1	8	3	0

Class 1 “At Risk” : score <6
Class 0 “Not at Risk” : score >=6

Appendix B

Results of MLAs on predicting performance in specific group of tasks without feature selection showed that SVM based model was the most accurate method with sensitivity of 0.39 and specificity of 0.84 and 0.60 area under curve as shown in Figure B.1. kNN based model as shown in Figure B.2 achieved 0.26 for sensitivity, 0.82 for specificity and 0.55 for area under the curve. Neural network based model as shown in Figure B.3 achieved 0.36 for sensitivity, 0.83 for specificity and 0.58 for area under the curve.

In conclusion, it seems that using Neighborhood Component Analysis (NCA) leads to find the best combination of features to increase the accuracy of all machine learning algorithms.

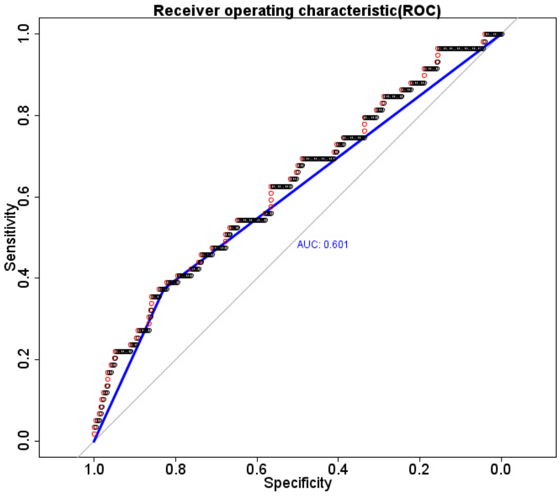


Figure B.1: Receiver Operating Characteristic of SVM

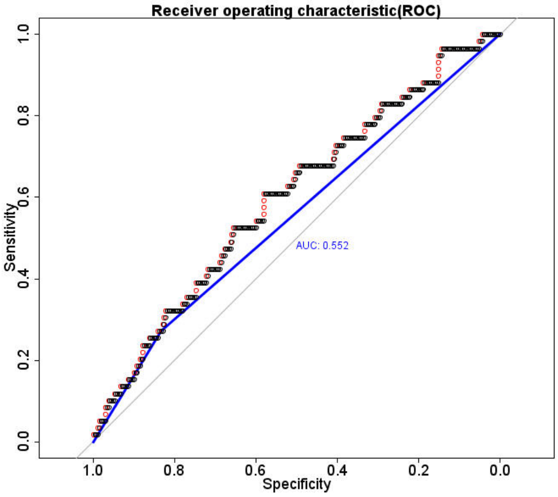


Figure B.2: Receiver Operating Characteristic of KNN

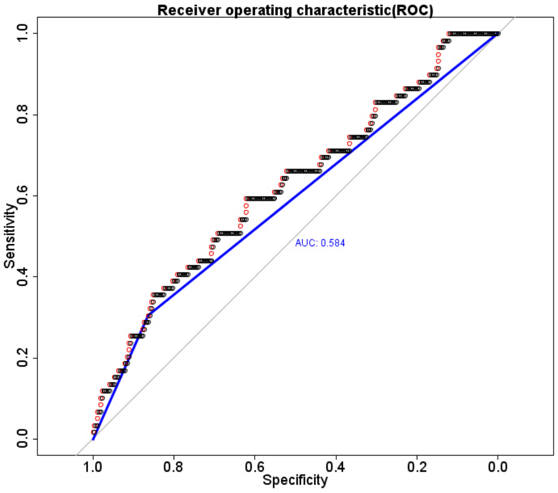


Figure B.3: Receiver Operating Characteristic of Neural Network