HELLENIC REPUBLIC
National and Kapodistrian University of Athens
EST. 1837

Department of Medicine
MSc Medical physics - Radiation physics

Master Thesis

Machine learning techniques for assessment of surgical skills during training in a virtual reality simulator

---

Author: Konstantina Prevezanou

Supervisor: Assoc. Professor Constantinos Loukas

Athens, May 2023

Περίληψη

Τα οφέλη της ελάχιστα επεμβατικής χειρουργικής είναι αποδεδειγμένα για τους ασθενείς (π.χ. μικρότερος χρόνος αποκατάστασης και πιθανότητα μόλυνσης). Ωστόσο, απαιτείται εξειδικευμένη εκπαίδευση για να διασφαλιστεί ότι οι επεμβάσεις αυτές εκτελούνται με ακρίβεια και ασφάλεια. Έτσι, έχουν αναπτυχθεί πλατφόρμες λαπαροσκοπικής εκπαίδευσης και προσομοιωτές εικονικής πραγματικότητας. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να αναλύσουν μεγάλα σύνολα δεδομένων για να αποκαλύψουν προηγουμένως μη αναγνωρισμένα μοτίβα και να διευρύνουν την κατανόηση της τεχνικής δεξιότητας.

Στόχος της διπλωματικής εργασίας είναι η εφαρμογή αλγορίθμων μηχανικής μάθησης για την ταξινόμηση εκπαιδευομένων που βρίσκονται στην αρχή/τελός της εκπαίδευσής τους σε λαπαροσκοπικό προσομοιωτή εικονικής πραγματικότητας (Lap Mentor). Συγκεκριμένα, το σύνολο δεδομένων περιλάμβανε μετρικές απόδοσης για 6 εκπαιδευτικές συνεδρίες (trials) σε 3 λαπαροσκοπικές ασκήσεις, οι οποίες εκτελέστηκαν από 23 φοιτητές ιατρικής (138 συνεδρίες ανά άσκηση). Οι τρεις ασκήσεις που επιλέχθηκαν από τον προσομοιωτή είναι: Clipping and Grasping (Άσκηση 5), Two-Handed Maneuvers (Άσκηση 6) και Cutting (Άσκηση 7). Για κάθε άσκηση, οι 3 πρώτες/τελευταίες συνεδρίες αντιστοιχούν στην αρχή/τέλος της εξάσκησης δεξιοτήτων (Start/End of Training), αντίστοιχα. Οι αλγόριθμοι που εφαρμόστηκαν για την αναγνώριση δεξιοτήτων (Start vs. End of Training) είναι: K-Nearest Neighbors, Logistic Regression, Naive Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Random Forest και Support Vector Machines. Επιπλέον, για κάθε αλγόριθμο διερευνήθηκε η εφαρμογή δύο τεχνικών dimensionality reduction: Principal Component Analysis (PCA) και Fisher's score. Η ανάλυση βασίστηκε σε δύο ξεχωριστά πειραματικά σχήματα: trial-based (η εκπαίδευση των αλγορίθμων έγινε σε επίπεδο συνεδρίας) και subject-based (η εκπαίδευση των αλγορίθμων έγινε σε επίπεδο εκπαιδευομένου).

Η ανάλυση έδειξε ότι το υψηλότερο ποσοστό ακρίβειας επιτεύχθηκε με τη χρήση Support Vector Machine (SVM) με γραμμικό kernel, με ποσοστό ακρίβειας: 97,08% για την άσκηση 5, 97,29% για την άσκηση 6 και 76,43% για την άσκηση 7, με χρήση του PCA. Επιπλέον, η μελέτη προσδιορίζει, μέσω του Fisher's score, τα έξι καλύτερα χαρακτηριστικά (μετρικές απόδοσης) για κάθε άσκηση και τις διαφορές στα ποσοστά ακρίβειας μεταξύ των δύο πειραματικών σχημάτων (διαφορά από 1 έως 3%). Συμπερασματικά, η χρήση αλγορίθμων μηχανικής μάθησης μπορεί να συνεισφέρει σημαντικά στην αντικειμενική αξιολόγηση χειρουργικών δεξιοτήτων σε προσομοιωτές εικονικής πραγματικότητας.

Abstract

The benefits of minimally invasive surgery (MIS) are well-established for patients (e.g. shorter recovery time and less chance of infection). However, specialised training is required to ensure that these procedures are performed accurately and safely. Thus, laparoscopic training platforms and virtual reality simulators have been developed. Machine learning algorithms can analyse large datasets to reveal previously unrecognised patterns and expand understanding of technical skill.

The objective of this thesis is to apply machine learning algorithms to classify trainees at the beginning/end of their training in a virtual reality laparoscopic simulator (Lap Mentor). Specifically, the dataset included performance metrics for 6 training sessions (trials) in 3 laparoscopic tasks performed by 23 medical students (138 sessions per task). The three tasks selected are Clipping and Grasping (Task 5), Two-Handed Maneuvers (Task 6), and Cutting (Task 7). For each task, the first/last 3 sessions correspond to the start/end of skill practice (Start/End of Training), respectively. The algorithms applied for skill assessment (Start vs. End of Training) are K-Nearest Neighbors, Logistic Regression, Naive Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Random Forest and Support Vector Machines. In addition, for each algorithm, the application of two dimensionality reduction techniques were investigated: Principal Component Analysis (PCA) and Fisher's score. The analysis was based on two separate experimental schemes: trial-based (algorithm training was performed at the session level) and subject-based (algorithm training was performed at the trainee level).

The analysis indicates that the the highest accuracy rate was achieved using Support Vector Machine (SVM) with linear kernel, with an accuracy rate of 97.08% for task 5, 97.29% for task 6 and 76.43% for task 7, using PCA. In addition, the study identifies, through Fisher's score, the six best features (performance metrics) for each task and the differences in accuracy rates between the two experimental schemes (difference from 1 to 3%). In conclusion, the use of machine learning algorithms can make a significant contribution to the objective evaluation of surgical skills in virtual reality simulators.

Acknowledgements

Prior to the analysis, I would like to express my heartfelt appreciation to some individuals who played a crucial role in my thesis.

First and foremost, I am deeply grateful to my supervisor, Associate Professor Constantinos Loukas, for granting me the opportunity to undertake this project and for his unwavering assistance and confidence in my abilities.

I also owe a tremendous debt of gratitude to Prof. Emmanouil Pikoulis (Head of the Surgical Simulation Center AKISA-Attikon Center of the University of Athens Medical School) and Dr. Panagis Lykoydis for their invaluable support and the provision of the dataset. Without their contribution, this thesis would not have been possible.

Lastly, I extend a special thank you to Christos Mermigkas and George Vretinaris for their valuable assistance and insightful recommendations throughout the task.

To all of you, I express my sincere appreciation. Without your aid, I would not have been able to write this paper.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Machine learning (ML)

The field of machine learning involves creating computer algorithms that can imitate human intelligence. It is an interdisciplinary field that incorporates concepts from various domains, including artificial intelligence, probability and statistics, computer science, information theory, psychology, control theory, and philosophy. Machine learning has been successfully employed in diverse areas, such as pattern recognition, computer vision, spacecraft engineering, finance, entertainment, ecology, computational biology, and biomedical and medical applications. The key characteristic of these algorithms is their capacity to learn from input data, either with or without a teacher, about the surrounding environment. [10]

Machine learning has a long history, dating back to the seventeenth century when Pascal and Leibniz developed machines that could perform basic arithmetic operations. In the modern era, the term "machine learning" was coined by Arthur Samuel of IBM in 1956, who showed that computers could be programmed to learn to play checkers. One of the earliest neural network architectures, the perceptron, was developed by Rosenblatt in 1958. However, enthusiasm for the perceptron was dampened by Minsky's observation that it could only solve linearly separable problems, not nonlinear ones.

A major breakthrough occurred in 1975 when Werbos developed the multilayer perceptron (MLP). Decision trees were introduced by Quinlan in 1986, followed by support vector machines by Cortes and Vapnik. Ensemble machine learning algorithms such as Adaboost and random forests were subsequently proposed. Recently, deep learning has emerged as a powerful technique for learning good representations of data, enabling more effective classification and prediction. [10]

1.1.1 Machine learning - AI - Deep learning

Machine learning, AI, and deep learning are three related concepts that are frequently mentioned in the tech industry. Although the terms are often used interchangeably, they represent distinct fields of study and practice. AI, or artificial intelligence, is a broad field that encompasses the development of intelligent machines that can simulate human thinking and decision-making processes. Machine learning is a subfield of AI that focuses on teaching machines to learn and improve their performance over time without being explicitly programmed to do so. Deep learning is a subset of machine learning that uses neural networks with many layers to learn from complex data.

Machine learning algorithms use statistical models to analyze and learn from data. They allow machines to identify patterns, make predictions, and take actions based on the knowledge acquired through that learning. This process is often referred to as "learning from experience."

Deep learning algorithms, on the other hand, use neural networks with many layers to learn from complex data. They are designed to recognize patterns in data that are too complex for traditional machine learning algorithms to handle. As a result, deep learning has been highly effective in areas such as computer vision, natural language processing, and speech recognition.

Overall, deep learning is a subset of machine learning, and machine learning is a subset of AI (Fig. 1.1). Without machine learning, AI would not be possible, and without deep learning, many of the complex tasks that we associate with AI would not be possible. As the field of AI continues to evolve, it is likely that deep learning and machine learning will play increasingly important roles in the development of intelligent machines.

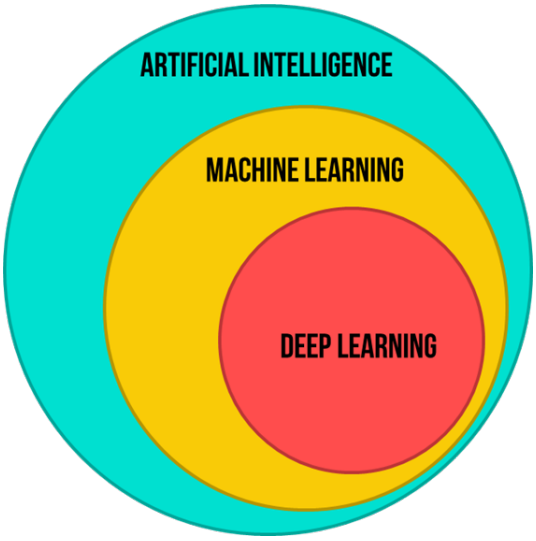


Figure 1.1: The relationship between AI, ML and Deep Learning.

1.1.2 ML algorithms

In order to construct an ML program, one might take a variety of ways. ML methods are typically divided into three broad categories:

1. Supervised learning: a "teacher" provides the computer with sample inputs and the desired outputs in order to teach it a general rule for mapping inputs to outputs. Supervised learning can be used for both classification and regression problems.
2. Unsupervised learning: the learning system is not given any labels or guidance. Instead, it is left to its own devices to identify structure in the data. The goal of unsupervised learning may be to find hidden patterns in the data for their own sake or to aid in feature learning.
3. Reinforcement learning: the computer program interacts with a dynamic environment, such as driving a vehicle or playing a game against an opponent. The software receives feedback that can be thought of as incentives as it navigates through the problem space and seeks to maximize those rewards. [5]

Figure 1.2 depicts a range of algorithms that fall within the category of supervised learning algorithms.

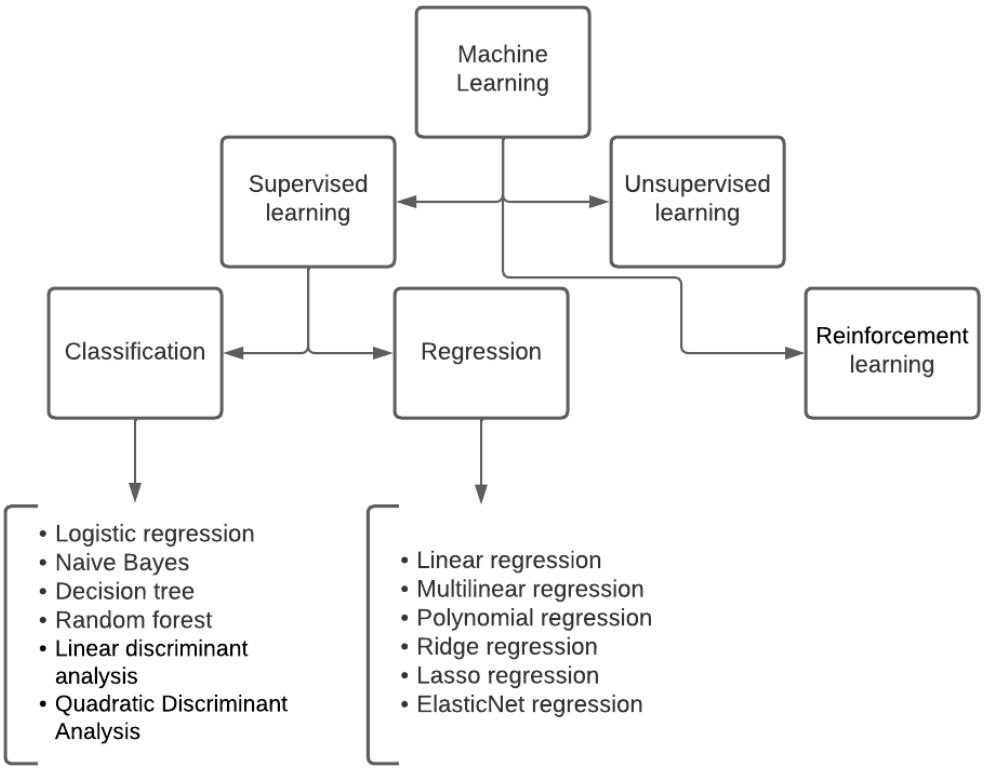


Figure 1.2: Some supervised learning algorithms.

This paper utilized various supervised learning algorithms to categorize data into two distinct groups. These algorithms are as follows:

1. **K-Nearest Neighbors**.
   The K nearest neighbors (kNN) algorithm is a straightforward approach that retains all available cases and categorizes new cases based on a similarity metric, such as distance functions. The classification of a case is based on the majority vote of its neighbors, and the case is allocated to the class that is most common among its K nearest neighbors as measured by a distance function (see Fig. 1.3). When K = 1, the case is directly assigned to the class of its nearest neighbor. [11]

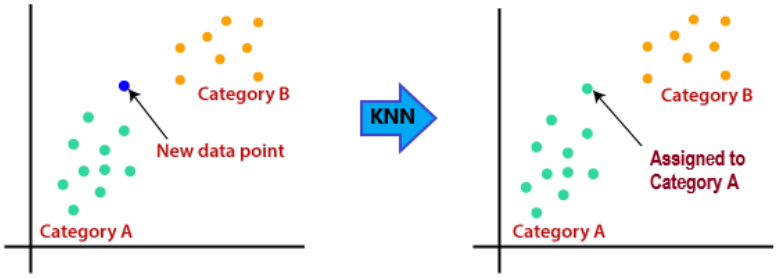


Figure 1.3: K-Nearest Neighbors algorithm.

Two distance measures (Eq. 1.1-1.2) can be utilized for continuous variables:

(a) Euclidean:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{1.1}$$

(b) Minkowski:

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q} \tag{1.2}$$

When dealing with categorical variables, the appropriate distance measure to use is the Hamming distance (Eq. 1.3).

$$D_H = \sum_{i=1}^{k}|x_i - y_i| \tag{1.3}$$

where:

$$x = y \Rightarrow D = 0$$

or

$$x \neq y \Rightarrow D = 1$$

2. **Linear Discriminant Analysis**.
   The objective of the Linear Discriminant Analysis (LDA) classification algorithm is to identify linear discriminant functions that can effectively differentiate between two or more classes based on their characteristics. These functions are created by combining predictors to create a new latent variable for each function.

   The LDA classifier calculates the discriminant function for each class under the assumption that the covariance matrices of the features for both classes are the same (i.e., $\Sigma 1 = \Sigma 2$). As a consequence, the discriminant functions become simpler and can be expressed as:

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^t \Sigma^{-1}(x - \mu_i) + \log P(\omega_i) \Rightarrow g_i(x) = {w_i}^t x + w_{i0} \tag{1.4}$$

   where x is a feature vector, $P(\omega_i)$ is the prior probability, μ_i is the mean vector and w_i and w_{i0} are constants that depend on the mean vectors and the prior probabilities of the classes.[3]

3. **Quadratic Discriminant Analysis**.
   The Quadratic Discriminant Analysis (QDA) algorithm is closely related to linear discriminant analysis, but unlike LDA, it does not assume that the covariance matrix of each class is identical. As a consequence, the discriminant functions are expressed as: [3]

$$g_i(x) = -\frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(x - \mu_i)^t {\Sigma_i}^{-1}(x - \mu_i) + \log P(\omega_i) \tag{1.5}$$

4. **Naive Bayes**.
   The Naive Bayes Classifier (NB) is a probabilistic classification algorithm that uses Bayes' theorem to predict the class of an observation based on its features. It is called "naive" because it assumes that the features are conditionally independent given the class, which means that the presence or absence of one feature does not affect the probability of any other feature.

   The Naive Bayes Classifier works by calculating the posterior probability of each class given the observation and its features, and then choosing the class with the highest probability (see Fig. 1.4). Bayes' theorem states that:

$$P(y|x) = P(x|y) \cdot P(y)/P(x) \tag{1.6}$$

where $P(x|y)$ is the conditional probability of x given y, P(y) is the prior probability of y, and P(x) is the marginal probability of x. [3]

The Naive Bayes Classifier can be represented mathematically as follows:

(a) Calculate the prior probability of each class:

$$P(y) = N(y)/N \tag{1.7}$$

where N(y) is the number of observations in class y, and N is the total number of observations.

(b) For each feature j and each class y, calculate the likelihood:

$$P(x_j|y) = N(x_j, y)/N(y) \tag{1.8}$$

where N(x_j,y) is the number of observations in class y that have feature j.

(c) For a new observation x, calculate the posterior probability of each class.

(d) Choose the class with the highest posterior probability.

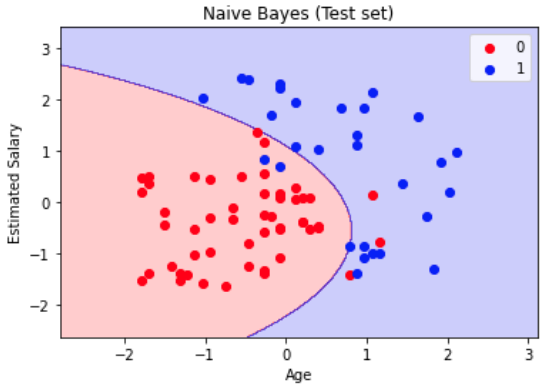


Figure 1.4: Gaussian Naive Bayes algorithm.

5. **Logistic Regression**.
The Logistic Regression algorithm models the probability of a binary outcome based on one or more predictor variables. It assumes that the relationship between the predictor variables and the probability of the binary outcome can be modeled using a logistic function. In other words, logistic regression predicts the probability of an event occurring given the values of one or more independent variables.

To estimate the coefficients for the logistic regression model, maximum likelihood estimation is used. This involves finding the values of the coefficients that maximize the likelihood of observing the given data. The log-likelihood function is represented by the following equation:

$$l(\beta) = \sum_{i=1}^{n} y_i \beta x_i - \log(1 + e^{\beta x_i}) \tag{1.9}$$

where x_i represents the feature vector for the ith sample.

Once the coefficients of the logistic regression model have been estimated, they can be used to predict the probability of the binary outcome for new observations. Specifically, the logistic regression classifier predicts the probability of the dependent variable (y) being equal to 1 for new observations using the following equation:

$$p(y = 1|x) = \frac{1}{1 + e^{-z}} \tag{1.10}$$

where z is the linear combination of the independent variables (x) and their estimated coefficients:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n * x_n \tag{1.11}$$

The logistic regression classifier then classifies the new observations based on a threshold probability, which is typically set to 0.5. If the predicted probability of the dependent variable (y) being equal to 1 is greater than the threshold probability, the observation is classified as belonging to class 1. Otherwise, the observation is classified as belonging to class 0.

6. **Random Forest Classifier**.
The Random Forest Classifier is an ensemble learning method that employs multiple decision trees to classify observations into two or more classes. Each decision tree is constructed using a random subset of the available features, and the final prediction is generated by combining the outcomes of all the individual trees. Decision trees serve as the building blocks of the random forest algorithm, which uses a tree-like model of decisions and their possible outcomes to classify observations. The tree consists of nodes that represent decisions based on the values of one of the input features, as well as leaves that represent the final decision or classification of the observation. [5]

   The decision tree algorithm used to build each individual tree can be represented as follows:

   (a) Let X be a matrix of n observations and p features, where each observation has a label y.

   (b) Starting at the root node, select the best feature j and threshold t_j to split the data into

two subsets S1 and S2, based on some criterion.

(c) Recursively apply the splitting process to each of the resulting subsets S1 and S2, until a stopping criterion is met.

(d) Assign the label y to each leaf node based on the majority class of the observations in that node.

The Random Forest algorithm creates an ensemble of decision trees, where each tree is trained on a random subset of the training data and a random subset of the available features. To classify a new observation, the algorithm obtains the predictions of all the individual trees and combines them to obtain the final prediction. Majority voting is the most common method used to combine the results of the individual trees, where the class that receives the most votes is chosen as the final prediction. Figure 1.5 provides a visual representation of this method.

The Random Forest Classifier can be mathematically represented as follows:

(a) Let X be a matrix of n observations and p features, where each observation has a label y.

(b) Let T be the number of decision trees to include in the forest, and let m be the number of features to consider at each split.

(c) For each t = 1,2,...,T, randomly select a subset of the observations and features to create a training set X_t.

(d) Train a decision tree model f_t on the training set X_t, using a stopping criterion.

(e) To classify a new observation x, obtain the predictions of all the individual trees f_t and combine them using majority voting to obtain the final prediction.

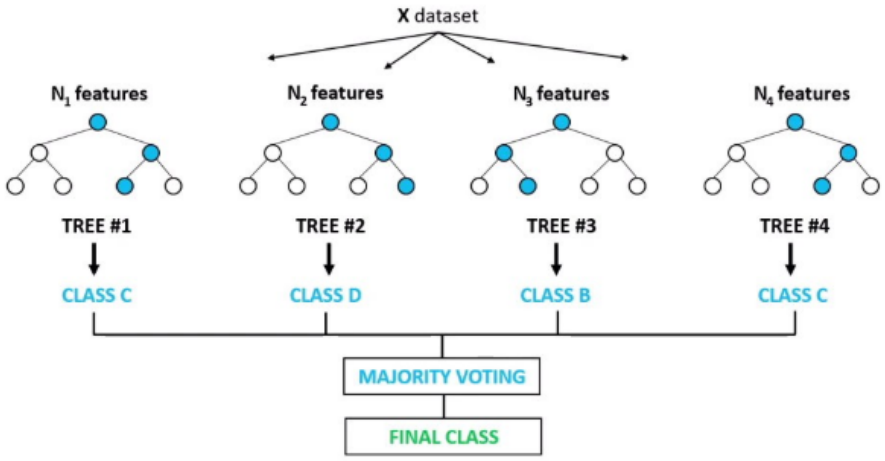


Figure 1.5: Random Forest algorithm.

7. **Support Vector Machine**.
   Support Vector Machine (SVM) is a powerful and widely used classification algorithm that works by finding the hyperplane that separates the data into two classes in the highest-dimensional space. The algorithm tries to maximize the margin between the classes, which is the distance between the hyperplane and the closest data points from each class. The decision boundary of the SVM is determined by a subset of the training data, called support vectors, that lie closest to the hyperplane.

   To classify new data, the SVM projects it into the same high-dimensional space as the training data and assigns it to the class on the side of the hyperplane where it falls.

   The SVM can be used for both linearly separable and non-linearly separable data by using different types of kernel functions to map the data into a higher-dimensional space where it can be linearly separable (see Fig. 1.6). There are four different kernels that can be used within the SVM algorithm:

   (a) Linear kernel

   $$\langle x, x' \rangle \tag{1.12}$$

   (b) Polynomial kernel

   $$(\gamma \langle x, x' \rangle + r)^d \tag{1.13}$$

   (c) RBF kernel

   $$\exp(-\gamma \|x - x'\|^2) \tag{1.14}$$

   (d) Sigmoid kernel

   $$tanh(\gamma \langle x, x' \rangle + r) \tag{1.15}$$

   The SVM classifier can be mathematically represented as follows:

   (a) Let X be a matrix of n observations and p features, where each observation has a label y.

   (b) The SVM finds the hyperplane that maximizes the margin between the classes by solving the following optimization problem:

   $$min_{w,b} \ \frac{1}{2} \|w\|^2 \tag{1.16}$$

subject to

$$y_i[(w \cdot x_i) + b] \geq 1 \tag{1.17}$$

where y_i is the label of the ith observation, x_i is the vector of its p features, w is the vector of weights, b is the bias term, λ is the regularization parameter, and $\|w\|^2$ is the squared L2 norm of the weight vector.

(c) The solution to this optimization problem is a hyperplane defined by the vector w and the scalar b that separates the data into two classes. To classify a new observation x, the SVM computes the sign of the function

$$w \cdot x + b \tag{1.18}$$

and assigns it to the positive class if the result is greater than or equal to zero, and to the negative class otherwise. [13]

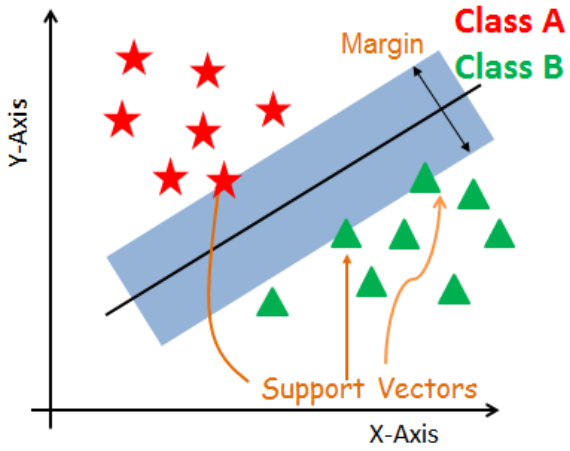


Figure 1.6: Support Vector Machine algorithm.

1.1.3 Dimensionality reduction

Large datasets often contain numerous irrelevant features, making it challenging to analyze and visualize data to identify patterns, and train machine learning models. This problem, known as the "Curse of Dimensionality", is typically addressed using dimensionality reduction techniques. These techniques involve transforming high-dimensional data into a lower-dimensional space while retaining meaningful properties. Dimensionality reduction is used to simplify models, reduce training times, and encode symmetries present in the input space.

Principal Component Analysis (PCA) is a common method for reducing the dimensions of large datasets. This involves transforming the data into a new coordinate system with fewer dimensions that can describe the variation in the original data. Specifically, PCA identifies the axis with the highest variance in the training set and finds a second axis orthogonal to it with the highest remaining variance. For higher-dimensional datasets, PCA can find additional axes, one for each dimension. The steps to find these axes are:

1. Compute the covariance matrix.

   The covariance matrix is a symmetric p x p matrix (where p is the number of dimensions), whose entries represent the covariances between all possible pairs of the initial variables. As a variable's covariance with itself is its variance (Cov(a,a) = Var(a)), the diagonal of the matrix (top left to bottom right) actually represents the variances of the initial variables.

$$cov(X, Y) = \frac{1}{n} \sum_{i=1}^{n} (x - \overline{x})(y - \overline{y}) \tag{1.19}$$

   where:
   x, y - members of X and Y variables,
   $\overline{x}, \overline{y}$ - mean of X and Y variables,
   n - number of members.

2. Compute Eigenvectors and corresponding Eigenvalues.

   Eigenvalues (λ):

$$\det(\lambda I - C) = 0 \tag{1.20}$$

   Eigenvectors (X) for each λ:

$$(\lambda I - C)X = 0 \tag{1.21}$$

3. Rank eigenvectors from highest to lowest and choose the first k you need.
   Eigenvectors of the Covariance matrix indicate the directions of the axes with the highest amount of variance, also known as Principal Components. Eigenvalues are the coefficients that are associated with the eigenvectors and represent the amount of variance carried in each Principal Component. To obtain the Principal Components in order of significance, you need to rank the eigenvectors based on their corresponding eigenvalues, from highest to lowest.

Feature selection is another approach to reducing the number of features in a dataset. This involves selecting a subset of relevant features for use in model construction. Fisher's score is a supervised feature selection method that returns the ranks of variables based on the Fisher criterion (Eq. 1.22), allowing the variables to be selected in descending order of relevance.

$$F = \frac{\sum_{j=1}^{k} p_j(\mu_j - \mu)^2}{\sum_{j=1}^{k} p_j {\sigma_j}^2} \tag{1.22}$$

where:
μ_j - mean of the data points belonging to class j for particular feature,
σ_j - standard deviation of the data points belonging to class j for particular feature,
p_j - the fraction of data points belonging to class j,
μ - the global mean of the data on the feature.

1.2 Minimally invasive surgery

Minimally invasive surgery (MIS), also known as minimal access or endoscopic surgery, has gained wide acceptance over the past three decades as a viable alternative to traditional surgery for various medical procedures. Today, laparoscopy, which involves inserting a small endoscope into the patient's abdominal cavity, is one of the most common types of minimally invasive surgery. During the procedure, the surgeon uses specialized instruments and views the patient's abdomen on a display monitor, allowing for precise manipulation of the instruments. After the procedure, any remaining carbon dioxide is expelled from the abdomen and the small incisions are closed with a minimal number of stitches.

MIS offers a number of benefits for patients, including reduced risk of blood loss and post-operative bleeding, less need for pain relief medication, decreased exposure of internal organs to external contaminants, and smaller scars that reduce the likelihood of post-operative infections.

In addition to the advantages for patients, healthcare providers also benefit from improved patient care and reduced medical risks, which can lead to greater efficiency in healthcare delivery. Shorter rehabilitation times mean that more surgeries can be performed each year, which ultimately translates to higher revenues and the potential for greater investments in the healthcare system. While patient safety remains the primary concern, the financial benefits of MIS can also contribute to long-term improvements in healthcare. [7]

1.2.1 Laparoscopy training

Laparoscopic surgery, as mentioned before, is a minimally invasive procedure that involves inserting a rigid endoscope into the patient's abdomen through a small incision. Surgeons must rely on a 2D view of the operating area displayed on a monitor positioned in front of them, which can make it challenging to operate without direct visual feedback. The limited depth perception, combined with the use of long-shaped instruments inserted through trocars, increases the difficulty of the operation. Surgeons must rely on other cues, such as touch and the interpretation of lights and shadows, to enhance their sense of depth and manipulate the instruments with greater precision.

The long shape of laparoscopic instruments also presents several challenges, including amplifying tremors and reducing the degrees of freedom of movement. Poor ergonomic design of the instruments' handles can make them difficult to manipulate, and the abdominal wall can act as a fulcrum that creates opposing instrument movements with hand movements. These factors require surgeons to adapt to new techniques and perform actions in a different way than with traditional laparotomy.

The traditional methods of surgical skill acquisition may not be sufficient to develop the specialized skills required for MIS. However, with specialized training and practice, surgeons can overcome these challenges and perform laparoscopic surgeries with precision and safety. [7]

Box trainers

As new surgical techniques were introduced, the medical community sought out alternative methods for surgeons to gain the necessary skills outside of the operating room to ensure safe and effective implementation of these techniques. One such method was the development of laparoscopic training platforms, or "box-trainers" (Fig. 1.7), which feature a simple design with holes for trocar insertion to simulate an insufflated abdominal cavity. Surgeons can use real laparoscopic instruments and a camera that simulates an endoscope to practice their skills, manipulating objects such as pegs and inanimate models of human organs. [7]

Figure 1.7: Box trainer.

To systematize the training and evaluation of both cognitive and psychomotor skills necessary for performing minimally invasive surgery (MIS), the Fundamentals of Laparoscopic Surgery (FLS) program was developed using the McGill Inanimate System for Training and Evaluation of Laparoscopic Skills (MISTELS) commercial training system. The FLS program includes a cognitive and manual (psychomotor) component, as well as task-specific metrics for subjective performance assessment and evaluation.

These first-generation box-trainers have since evolved into more sophisticated training platforms known as Physical Reality (PR) surgical simulators (Fig. 1.8), which require trainees to stand up and operate within the confines of simulated anatomical structures such as the pelvis and upper abdomen. [7]

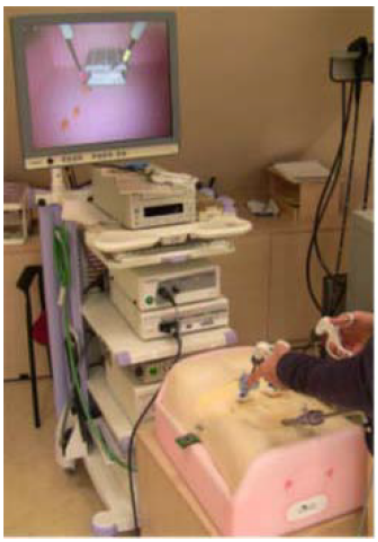


Figure 1.8: Physical Reality (PR) surgical simulator.

Virtual reality simulator

The need for an automated and assessable laparoscopic training and assessment curricula led to the development of computer-based laparoscopic simulations. With the advancement of computer science and technology, virtual reality (VR) has become an ideal solution for laparoscopic training. The first commercially available VR laparoscopic simulator was MIST-VR (Fig. 1.9), introduced in 1997, which provided a realistic and assessable VR environment for laparoscopic cholecystectomy training. [2, 15]

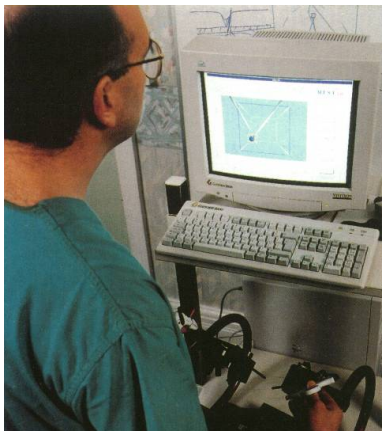


Figure 1.9: MIST-VR. [15]

Today, high-end VR laparoscopic simulators (see Fig. 1.10) combine cognitive and motor skills training into an integrated VR learning experience, providing a unique training opportunity in a highly realistic, purely virtual environment. In VR trainers, surgeons are individually guided through a series of training scenarios of progressive difficulty and complexity. VR trainers allow

for smooth skill development and transition of skills from training to clinical practice, as a large set of basic procedures can be performed, including endoscope navigation, cutting and suturing, needle driving, diathermy, and other essential exercises. Furthermore, VR trainers provide automated objective assessment of trainees' performance based on specific metrics, such as the task completion time, the number of errors committed, and the instruments' path length, to help improve their individual psychomotor and cognitive skills required for performing a real laparoscopic surgery. High-end VR trainers are also equipped with mechanical feedback devices, which provide real-time haptic feedback during training, enhancing the overall sense of simulation realism. [7]

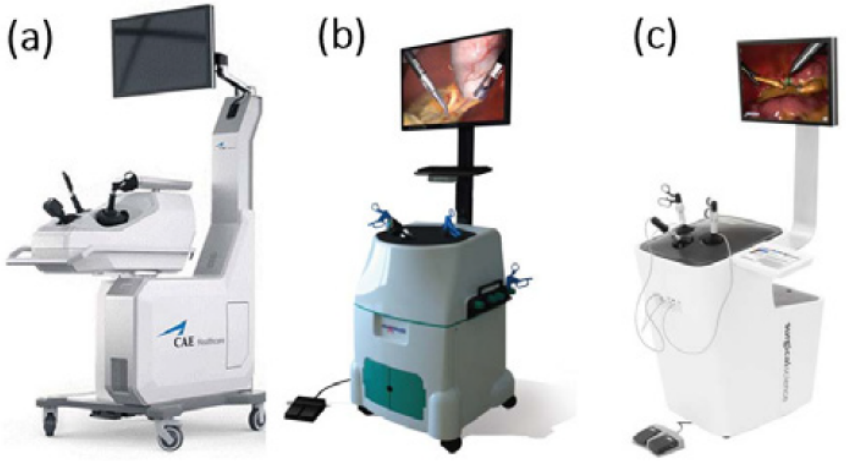


Figure 1.10: Commercially available laparoscopic VR trainers: (a) LapVR (CAE), (b) LapMentor (Simbionix), (c) LapSim (Surgical Science).

1.3 Assessment

1.3.1 Traditional assessment

Surgical residency programs place great emphasis on learning the art of surgery, making it important to have a formal system in place to assess the technical skills of every student and track their progress. There are three main features that can be used to assess technical skills: time, path length and errors. It is crucial for any method used to assess technical skills to be reliable and valid. Currently, preceptor ratings heavily influence the assessment of trainees' technical skills, with a single global rating being the norm. However, this rating is often unreliable and not sufficient for formative feedback or promotion decisions.

To address these issues, the Objective Structured Assessment of Technical Skill (OSATS) was developed in 1997. This assessment method involves observing surgical residents performing various structured operative tasks and rating them on the scale in Figure 1.11.

Please rate the candidate's performance on the following scale:

	1	2	3	4	5
Respect for tissue	Frequently used unnecessary force on tissue or caused damage by inappropriate use of instruments.		Careful handling of tissue but occasionally caused inadvertent damage.		Consistently handled tissues appropriately with minimal damage.
Time and motion	Many unnecessary moves.		Efficient time/motion but some unnecessary moves.		Economy of movement and maximum efficiency.
Instrument handling	Repeatedly makes tentative or awkward moves with instruments.		Competent use of instruments although occasionally appeared stiff or awkward.		Fluid moves with instruments and no awkwardness.
Knowledge of instruments	Frequently asked for the wrong instrument or used an inappropriate instrument.		Knew the names of most instruments and used appropriate instrument for the task.		Obviously familiar with the instruments required and their names.
Use of assistants	Consistently placed assistants poorly or failed to use assistants.		Good use of assistants most of the time.		Strategically used assistant to the best advantage at all times.
Flow of operation and forward planning	Frequently stopped operating or needed to discuss next move.		Demonstrated ability for forward planning with steady progression of operative procedure.		Obviously planned course of operation with effortless flow from one move to the next.
Knowledge of specific procedure	Deficient knowledge. Needed specific instruction at most operative steps.		Knew all important aspects of the operation.		Demonstrated familiarity with all aspects of the operation.

Overall, on this task, should this candidate: ☐**Pass** ☐ **Fail ?**

Figure 1.11: OSATS checklist.

Testing specific operative skills in surgical trainees, the OSATS has been found to be a reliable and valid method. [9]

With the development of the field of surgery, it became necessary to create an appropriate tool for assessing minimally invasive surgery. As a solution, the Global Operative Assessment of Laparoscopic Skills (GOALS) was developed as a global assessment tool (Fig. 1.12). It has been found to be feasible, reliable, and valid for evaluating the technical skills of residents in minimally invasive surgery, providing them with valuable feedback. Furthermore, it can be used to measure the effectiveness of simulator training in improving surgical performance. [14]

Global rating scale component of the intraoperative assessment tool*

Depth perception
1. Constantly overshoots target, wide swings, slow to correct
2.
3. Some overshooting or missing of target, but quick to correct
4.
5. Accurately directs instruments in the correct plane to target

Bimanual dexterity
1. Uses only one hand, ignores nondominant hand, poor coordination between hands
2.
3. Uses both hands, but does not optimize interaction between hands
4.
5. Expertly uses both hands in a complimentary manner to provide optimal exposure

Efficiency
1. Uncertain, inefficient efforts; many tentative movements; constantly changing focus or persisting without progress
2.
3. Slow, but planned movements are reasonably organized
4.
5. Confident, efficient and safe conduct, maintains focus on task until it is better performed by way of an alternative approach

Tissue handling
1. Rough movements, tears tissue, injures adjacent structures, poor grasper control, grasper frequently slips
2.
3. Handles tissues reasonably well, minor trauma to adjacent tissue (ie, occasional unnecessary bleeding or slipping of the grasper)
4.
5. Handles tissues well, applies appropriate traction, negligible injury to adjacent structures

Autonomy
1. Unable to complete entire task, even with verbal guidance
2.
3. Able to complete task safely with moderate guidance
4.
5. Able to complete task independently without prompting

* The descriptors shown are the "anchor" descriptors for scores 1, 3, and 5.

Figure 1.12: GOALS checklist.

1.3.2 AI assessment

The standard method of evaluating surgical proficiency involves skilled assessors observing a surgery or training exercise and evaluating the trainee's performance using global or procedure-specific checklists. However, this method is costly, subjective, and time-consuming, which can delay feedback for trainees and impede their learning. [6]

Virtual reality simulators provide opportunities for formative and summative assessments by gen-

erating multiple performance metrics. [8]

In 2011, Loukas and Georgiou [8] tested an alternative approach using multivariate autoregressive models (MAR) trained with a variational Bayesian algorithm against the already used hidden Markov models. The results showed that the MAR approach outperformed the traditional approach.

Robotic surgery technology has created opportunities for automated objective skill assessment and prompt feedback. The da Vinci surgical device records motion and video data, enabling the development of computational models to analyze surgical skills. New features to quantify surgical flow were introduced in 2017, which can evaluate a surgeon's skills and provide feedback to trainees by comparing their surgical skills with other surgeons' using a comprehensive dataset. [4]

In 2019, the Myo armband was introduced (presented in Fig. 1.13), which enables intraoperative assessment of hand and forearm motion parameters. It contains an inertial measurement unit and eight electromyographic sensors, which in combination with machine learning can distinguish skill level and recognize the phases of a laparoscopic suturing and knot-tying task. [6]

Figure 1.13: Myo armband.

In the same year, a machine learning algorithm was used to accurately classify participants by level of expertise in a virtual reality surgical procedure. [17]

To enable researchers from computer science, medicine, and education to develop a shared understanding of the emerging field of machine learning-assisted surgical education, a standardized reporting approach was developed in 2019. The Machine Learning to Assess Surgical Expertise (MLASE) checklist provides clear subsections and a total score for authors and reviewers to evaluate the overall quality and specific weaknesses of a manuscript. The MLASE checklist can be observed in Figure 1.14. [16]

Section	Element	Yes?
Study design (5 points)	1. Is relevant literature on the use of artificial intelligence in simulation provided?	
	2. Is the sample size clearly stated (including number of groups and number of participants in each group)?	
	3. Is a definition of each group of expertise provided?	
	4. Is the simulator described?	
	5. Are the surgical tasks to be performed outlined?	
Data structure (6 points)	6. Is raw data acquisition described?	
	7. Is feature extraction mentioned?	
	8. Is an effort made to normalize the data?	
	9. Is feature selection mentioned?	
	10. Is the count of features used by the algorithm clearly stated?	
	11. Are the final selected features clearly described?	
Supervised machine learning (5 points)	12. Is the type of the classifier used mentioned and justified (either by comparing multiple classifiers or citing relevant literature)?	
	13. Is the mechanism of the classifier explained or is a relevant source provided?	
	14. Is an effort made to clearly describe the methods used to train and test the algorithm?	
	15. Is the accuracy of the classifier mentioned?	
	16. Is the sensitivity and specificity mentioned?	
Discussion quality (4 points)	17. Are efforts made to explain the educational rationale of the features used by the algorithm?	
	18. Is the educational application of classifiers in the context of surgical simulation stated, specifically its use as a summative or formative assessment tool?	
	19. Are methodological limitations discussed, including those pertaining to any above-points?	
	20. Are the future directions discussed?	
Total Score = _____/20		

The checklist contains 20 elements, separated into 4 sections. A point is awarded for every element completed in the article. The total score is calculated by adding the total number of elements checked.

Figure 1.14: MLASE checklist.

1.4 Aim of this study

The objective of this thesis is to leverage the data gathered from laparoscopy training in a virtual reality simulator utilized by medical students, with the aim of creating diverse scripts through the implementation of machine learning algorithms. This approach will facilitate the classification of students into two distinct categories depending on whether they start or finish their training, namely "*Start of training" (ST)* and "*End of Training" (ET)*.

By utilizing this cutting-edge technology, medical students can receive comprehensive feedback on their training progress, and educators can tailor their instructional techniques to meet individual student needs, ultimately improving the overall quality of medical education.

Chapter 2

Methodology

2.1 Study participants

The input data, provided from Surgical Simulation Center of the University of Athens Medical School (AKISA), located in the Attikon General Hospital, was collected from simulation tasks performed by the 23 medical students. These trials were designed to test their skills and knowledge in various medical tasks, and the records contained the features and feature values from each of these trials.

2.2 Study design

The VR simulator utilized in this study was the Lap Mentor from Simbionix. This advanced platform offers an extensive library of modules that offer structured training programs with varying levels of difficulty for basic laparoscopic tasks and skills, as well as complete procedure training for general, gynecological, urologic, bariatric, colorectal, and thoracic surgery. With 19 training modules and over 80 different tasks and cases, the Lap Mentor provides a comprehensive and versatile learning experience for surgeons and surgical trainees.

The Basic Laparoscopic Skills for Undergraduate Students module is a comprehensive training program that consists of nine tasks (presented in Table 2.1), each focused on a different laparoscopic skill. [1]

Task 1	Camera manipulation - 0 degrees	Task 6	Two-handed maneuvres
Task 2	Camera manipulation - 30 degrees	Task 7	Cutting
Task 3	Eye-hand coordination	Task 8	Electrocautery
Task 4	Clip applying	Task 9	Translocation of objects
Task 5	Clipping and grasping		

Table 2.1: Basic Laparoscopic Skills for Undergraduate Students

Based on the educational curriculum developed in the AKISA Surgical Simulation Center, a student must successfully complete Task 1 before moving on to Task 2 and so on. If the student fails to complete a task, they must start the module from the beginning. This process continues until the student has completed all nine tasks and achieved at least three Passes on each one. This criteria is crucial in ensuring that the students have a solid understanding of the basic laparoscopic skills and are ready to move on to more challenging modules. Upon successfully completing the Basic Laparoscopic Skills module, the students can then move on to more advanced modules to perfect their skills and gain further expertise in laparoscopic surgery.

For the purpose of our analysis, we selected three specific tasks that were deemed most relevant and representative of the students' abilities. These tasks are:

- *Task 5, Clipping and Grasping*
  - Safely grasp and clip leaking ducts within the specified segments. Red segments will appear on the ducts at the beginning of the task. The segment will turn green only when grasped properly. Grasp the leaking duct and use the clipper to place a clip within the green segment only to stop leakage. Complete the task before the pool overflows. A screenshot of the task is shown in Figure 2.1.
  - Main goals:
    1. Basic principles of safe clipping.
    2. Clip applier manipulation.
    3. Tissue handling skills.
    4. Bimanual skills.
    5. Laparoscopic orientation.
    6. Eye-hand coordination.

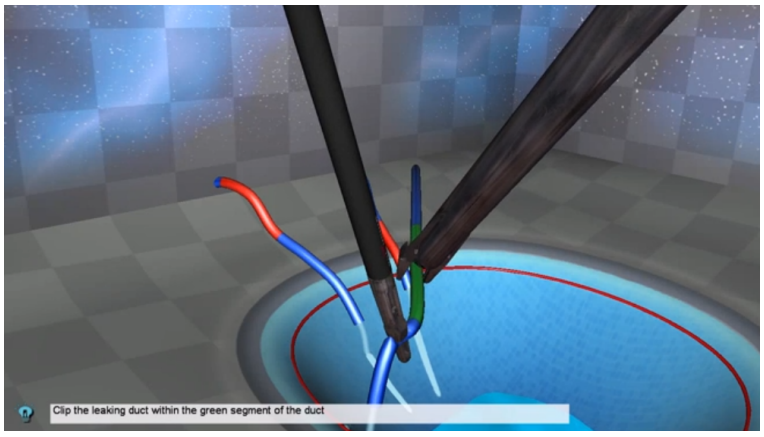


Figure 2.1: Screenshot from task 5.

- *Task 6: Two Handed Maneuvers*
  - Use two graspers. Locate the jelly mass and with one of the graspers move part of the jelly aside to expose a ball. Notice the colot change when the balls are exposed. While holding the jelly aside, use the other tool to grasp the green ball and place it in the Endobag. Make sure to release the balls above the Endobag. A screenshot of the task is shown in Figure 2.2.
  - Main goals:
    1. Advanced bimanual skills.
    2. Laparoscopic instrument manipulation.
    3. Eye-handed coordination.
    4. Tissue handling skills.

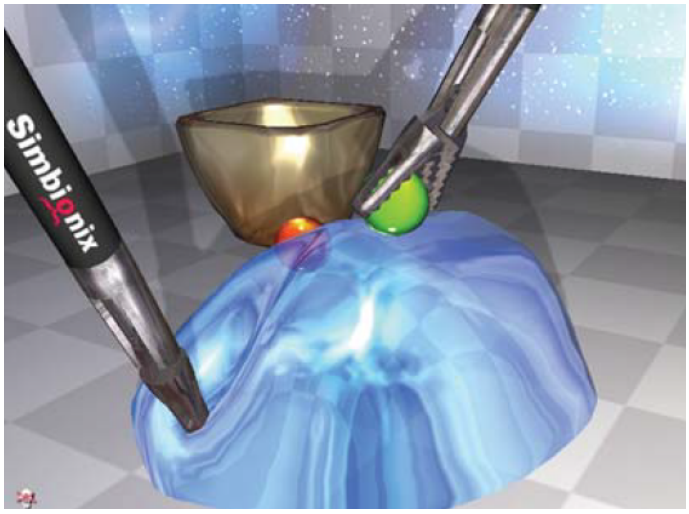


Figure 2.2: Screenshot from task 6.

- *Task 7: Cutting*
  - Safe cutting and separate a circular form. Use one tool to retract the form exposing a safe cutting area. Cut accurately with scissors. A screenshot of the task is shown in Figure 2.3.
  - Main goals:
    1. Applying traction and cutting accurately using scissors.
    2. Bimanual skills.
    3. Eye-handed coordination.

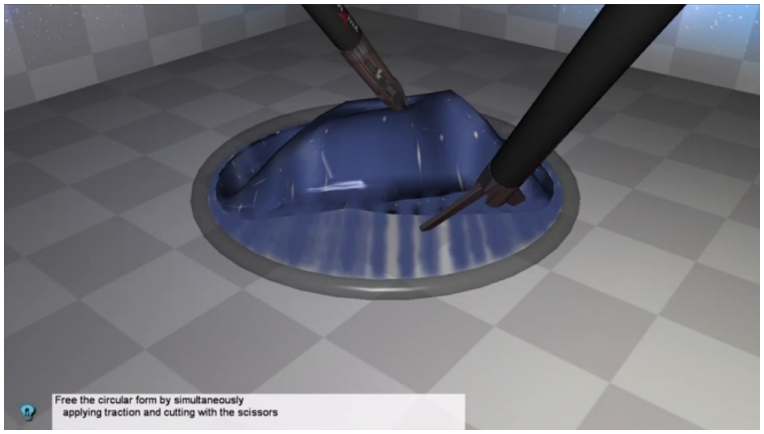


Figure 2.3: Screenshot from task 7.

Table 2.2 shows the thresholds (defined by the educators-experts) required to achieve a Pass in tasks 5, 6 and 7.

Task 5	
Feature	**Value**
Total number of movements (n)	≤ 70
Number of clipped ducts	≥ 9
Number of lost clips	<1
Task 6	
Feature	**Value**
Total number of movements	≤ 50
Number of lost balls which miss the basket	<1
Task 7	
Feature	**Value**
Total Number of cutting maneuvers	≤ 20

Table 2.2: Thresholds to achieve a Pass.

2.3 Class definition

Before embarking on script writing, the dataset had to be compiled. Our selection criteria were the first three attempts - categorized as *"Start of Training" (ST)* - and last three attempts - categorized as *"End of Training" (ET)* - for each task by each student. This resulted in a dataset consisting of 138 rows or samples (23 x 6) for each task.

In instances where a student prematurely discontinued a trial, the simulator did not record any feature values. In such cases, we opted for the subsequent attempt.

2.4 Experimental schemes

As mentioned earlier, when dealing with machine learning scripts, it is necessary to divide the initial dataset into a training set to train the algorithm and a testing set for the algorithm to make predictions. This study utilized two experimental schemes to split the dataset:

1. The first approach, called the *"Trial-based"* split, utilized the ShuffleSplit function from the sklearn.model-selection library. The data was shuffled and divided into 20 different training and testing sets.
2. The second approach, known as the *"Subject-based"* split, used the GroupShuffleSplit function from the same library. In this case, the data was shuffled based on the subject number, resulting in training sets that excluded trials from a specific subject. For instance, one training set was created from trials of subjects 1 to 13, while the testing set consisted of trials from subjects 14 to 23. [12]

To accommodate each scenario, a separate script was written.

2.5 Dimensionality reduction

Two methods were employed to enhance efficiency by reducing the number of available features for each task.

- The first approach was to apply Principal Component Analysis (PCA) for dimensionality reduction. It transforms the data into a new coordinate system in a linear manner, resulting in fewer dimensions while preserving 95% of the variance.
- The second method employed was Fisher's score for feature selection. This process calculates the Fisher's score for each feature, and the features are ranked in descending order based on their score. The best six features are then selected.

To accommodate each scenario, a separate script was written.

2.6 Scripts outline

All scripts involving machine learning algorithms follow a standard structure. The initial step involves examining the data for any issues, such as missing or NaN values. Then, the data is divided into training and testing sets. Subsequently, feature extraction or selection is performed, followed by training the selected algorithm and making predictions. This same sequence was applied in writing the scripts for this paper and is presented in Figure 2.4.

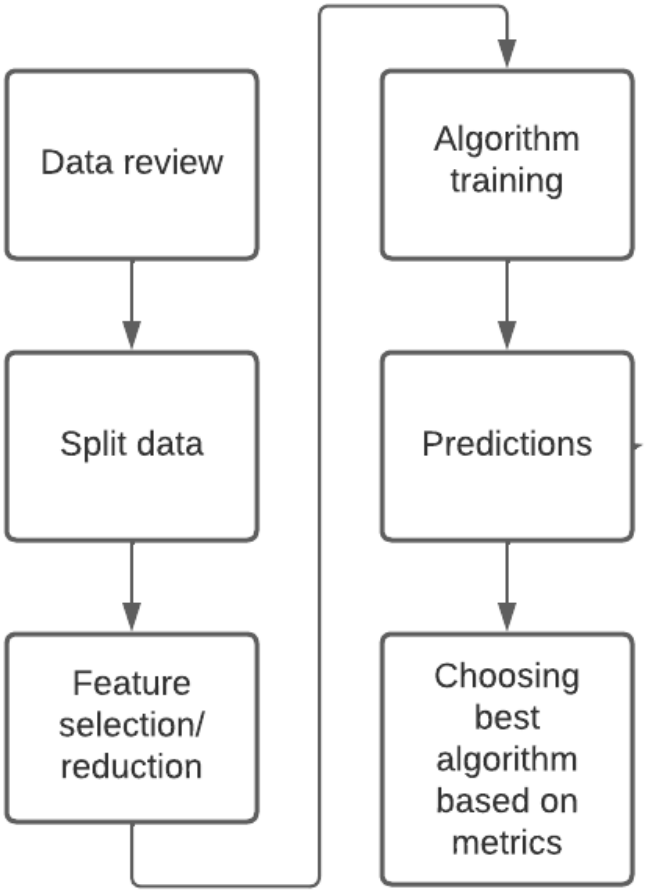


Figure 2.4: Flowchart of scripts.

Firstly, any blank columns (features) were removed from the dataset and any missing values were addressed. To ensure accurate predictions, the data was then divided into separate training and testing sets, with dimensionality reduction performed to improve the algorithms' efficiency. Once these steps were completed, various machine learning algorithms were trained using the prepared data. Finally, these algorithms made their predictions.

2.7 Performance measures

To evaluate the performance of each machine learning algorithm, the models were trained and tested 20 times using the different train-test sets created by the shuffle split. The final evaluation was based on the average results of various metrics, including:

- A confusion matrix: a tabular visualization of the model predictions versus the ground-truth labels (Fig. 2.5).

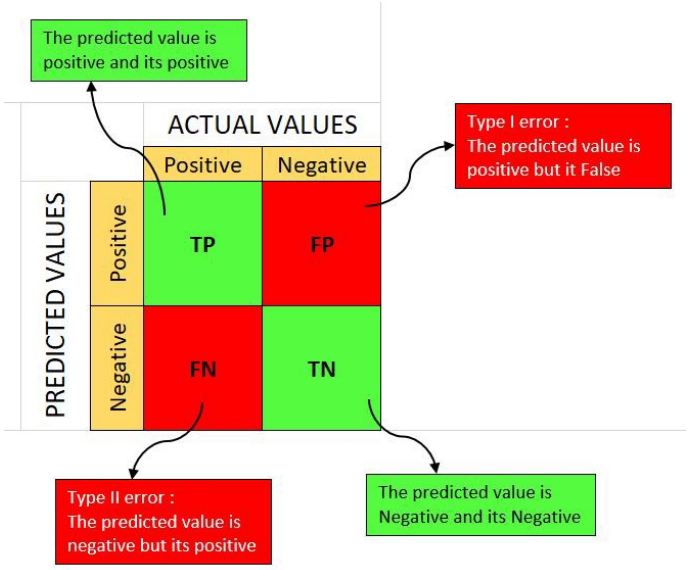


Figure 2.5: Example confusion matrix.

It is important to note that:

  - True Positive: the algorithm predicted yes and it is true.
  - True Negative: the algorithm predicted no and it is true.
  - False Positive: the algorithm predicted yes and it is false.
  - False Negative: the algorithm predicted no and it is false.
  - The sum of each row represents 100% of the elements in a particular class.
  - The sum of all elements represents all data.

Specifically in this study:

  - Positive corresponds to *"Start of Training"* and
  - Negative corresponds to *"End of Training"*.

- $\text{Accuracy} = \dfrac{\text{True positive} + \text{True negative}}{\text{True positive} + \text{False positive} + \text{True negative} + \text{False negative}} \cdot 100\%$

- Precision = $\dfrac{\text{True positive}}{\text{True positive + False positive}} \cdot 100\%$

- Specificity = $\dfrac{\text{True negative}}{\text{True negative + False positive}} \cdot 100\%$

- Sensitivity = $\dfrac{\text{True positive}}{\text{True positive + False negative}} \cdot 100\%$

- $\text{F1} - \text{Score} = \dfrac{2 \cdot \text{Precision} \cdot \text{Sensitivity}}{\text{Precision + Sensitivity}} \cdot 100\%$

These metrics helped to assess the performance of each algorithm in the prediction task, and allowed for a comparison between the different algorithms.

Chapter 3

Results

The results of the scripts were divided based on the task and the method of data splitting, and are presented as follows:

- Results for each task and each method of data splitting are presented in tables, including the evaluation metrics mentioned previously.
- Only the confusion matrix of the algorithm with the highest accuracy is presented in the main text, while the confusion matrices of the other algorithms could be found in the Appendix: Confusion matrices. When looking at the confusion matrices:
  - 0 denotes "*Start of Training*" and 1 denotes "*End of Training*",
  - they are obtained by combining the confusion matrices from the 20 individual runs and each number represents a percentage after normalizing the results (ex. 94% TP: 94% of the samples were categorized as *"Start of Training"* and they were actually at the *"Start of Training"*).
- MatLab (The MathWorks, Inc., R2022b) was used to perform statistical comparison of the accuracy results for each algorithm across all 20 runs. A script was written, which consisted of two parts.
  - Firstly, an Anova1 test was performed using the Matlab command [p,t,stats] = anova1(y, groups). This test aimed to determine whether the samples in y were drawn from populations with the same mean, or whether the population means were not all the same. The graph produced by this command (in Statistical comparison figure (a) on all cases) is a boxplot where on each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the "+" marker symbol.
  - Secondly, a multiple comparison test was conducted using the information contained in the stats structure, by executing the Matlab command [c,m,h,gnames] = multcom-

pare(stats). The output of this test is an interactive graph that indicated which pairs were different (in Statistical comparison figure (b) on all cases). In more detail, each group mean is represented by a symbol, and the interval is represented by a line extending out from the symbol. Two group means are significantly different if their intervals are disjoint; they are not significantly different if their intervals overlap. If you use your mouse to select any group, then the graph will highlight all other groups that are significantly different, if any.

This organization of results allows for a clear comparison between the different algorithms and the different methods of data splitting, making it easier to understand the findings and draw conclusions from the analysis.

3.1 Two class classification (Dim. Reduction: PCA)

3.1.1 Task 5

Trial based: Results

- PCA: Dimensionality was reduced to 6 - 7 components from 25 original features (95% variance threshold).
- Table 3.1 presents the algorithms' performance.

Algorithms	**Accuracy (%)**	**F1 (%)**	**Specificity (%)**
KNN	94.14	94.10	95.11
LDA	92.14	91.48	85.87
QDA	93.86	93.99	**97.55**
Logistic regression	95.71	95.63	95.21
Naive Bayes	95.14	94.95	92.87
Random forest	92.00	92.05	94.18
SVM Linear	**96.14**	**96.16**	94.31
SVM RBF	96.00	96.05	95.78
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	95.06	93.16	
LDA	85.76	**98.01**	
QDA	97.67	90.57	
Logistic regression	95.35	95.91	
Naive Bayes	93.02	96.97	
Random forest	94.19	90.00	
SVM Linear	**98.26**	94.15	
SVM RBF	96.59	95.51	

Table 3.1: Algorithms' performance for Task 5 (trial-based scheme) using PCA.

- Figure 3.1 shows the confusion matrix for SVM Linear.

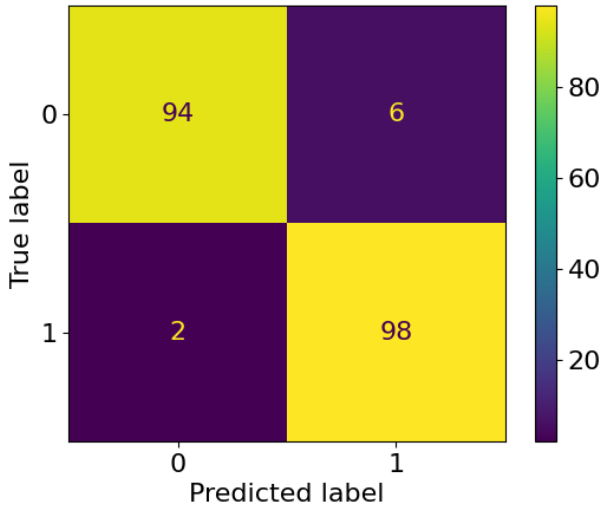
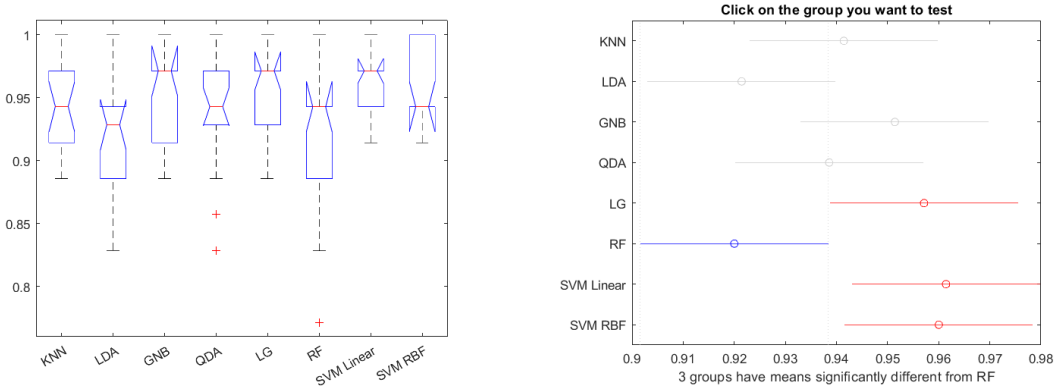


Figure 3.1: Confusion matrix (100%) for SVM Linear (Task 5; trial-based; PCA).

Trial based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.2, which reveals that:

- LDA has a different mean from SVM (linear and RBF),
- RF has a different mean from SVM (linear and RBF) and LG.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.2: Statistical comparison (Task 5; trial-based; PCA).

Subject based: Results

- PCA: Dimensionality was reduced to 6 - 7 components from 25 original features (95% variance threshold).
- Table 3.2 presents the algorithms' performance.

Algorithms	**Accuracy (%)**	**F1 (%)**	**Specificity (%)**
KNN	95.14	95.15	95.28
LDA	89.17	87.96	79.17
QDA	94.31	94.51	**98.06**
Logistic regression	94.44	94.38	93.33
Naive Bayes	93.19	93.05	91.11
Random forest	93.06	93.00	92.22
SVM Linear	**97.08**	**97.04**	95.56
SVM RBF	94.72	94.54	91.39
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	95.28	95.01	
LDA	79.17	**98.96**	
QDA	98.06	91.21	
Logistic regression	93.33	95.45	
Naive Bayes	91.11	95.07	
Random forest	92.22	93.79	
SVM Linear	**98.57**	95.56	
SVM RBF	97.92	91.39	

Table 3.2: Algorithms' performance for Task 5 (subject-based scheme) using PCA.

- Figure 3.3 shows the confusion matrix for SVM Linear.

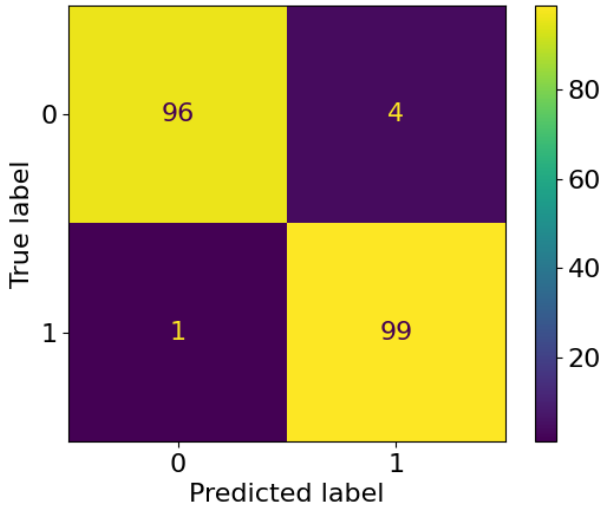
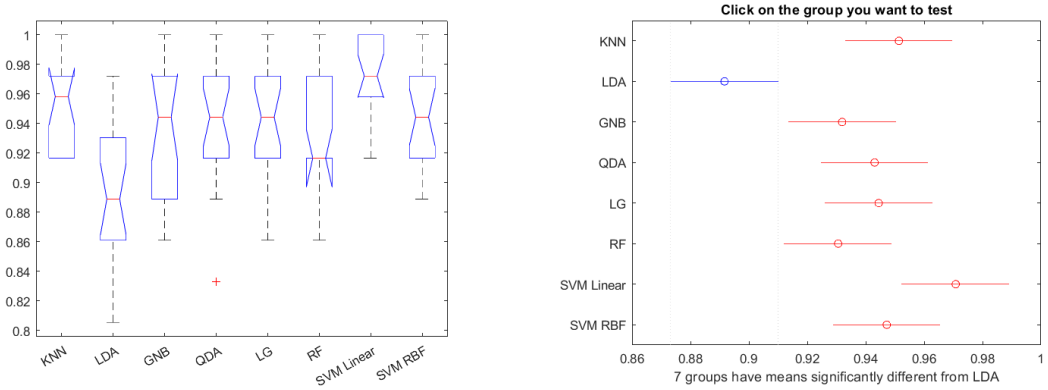


Figure 3.3: Confusion matrix (100%) for SVM Linear (Task 5; subject-based; PCA).

Subject based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.4, which reveals that:

- LDA has a different mean from all other algorithms,
- GNB has a different mean from LDA and SVM Linear,
- RF has a different mean from LDA and SVM Linear.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.4: Statistical comparison (Task 5; subject-based; PCA).

3.1.2 Task 6

Trial based: Results

- PCA: Dimensionality was reduced to 6 - 7 components from 17 original features (95% variance threshold).
- Table 3.3 presents the algorithms' performance.

Algorithms	**Accuracy (%)**	**F1 (%)**	**Specificity (%)**
KNN	95.71	95.77	93.78
LDA	93.00	93.13	91.71
QDA	94.43	94.66	95.59
Logistic regression	95.86	95.93	94.39
Naive Bayes	91.29	91.59	91.78
Random forest	94.71	94.95	95.91
SVM Linear	**97.29**	**97.34**	**96.49**
SVM RBF	96.29	96.20	94.47
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	93.66	**97.98**	
LDA	91.46	94.86	
QDA	95.32	94.02	
Logistic regression	94.21	97.71	
Naive Bayes	91.46	91.71	
Random forest	95.87	94.05	
SVM Linear	**98.30**	96.39	
SVM RBF	98.21	94.27	

Table 3.3: Algorithms' performance for Task 6 (trial-based scheme) using PCA.

- Figure 3.5 shows the confusion matrix for SVM Linear.

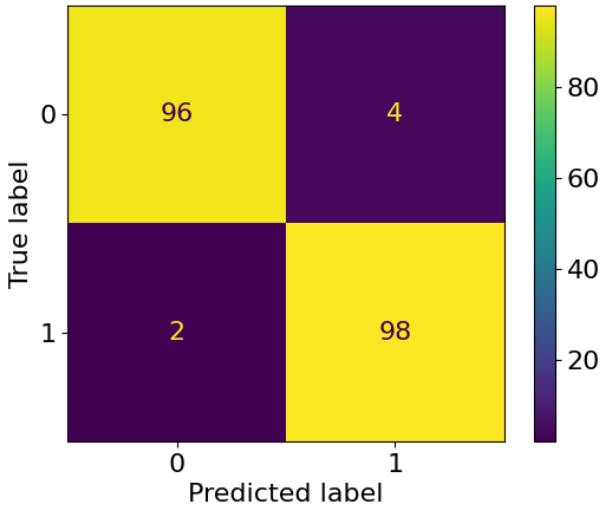
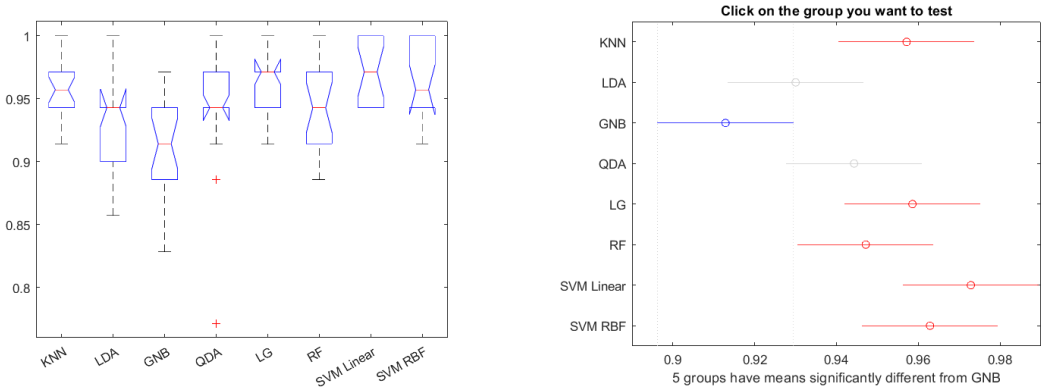


Figure 3.5: Confusion matrix (100%) for SVM Linear (Task 6; trial-based; PCA).

Trial based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.6, which reveals that:

- GNB has a different mean from KNN, LG, RF, SVM Linear and SVM RBF,
- LDA has a different mean from SVM Linear.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.6: Statistical comparison (Task 6; trial-based; PCA).

Subject based: Results

- PCA: Dimensionality was reduced to 6 - 7 components from 17 original features (95% variance threshold).
- Table 3.4 presents the algorithms' performance.

Algorithms	**Accuracy (%)**	**F1 (%)**	**Specificity (%)**
KNN	92.64	92.73	93.89
LDA	94.58	94.40	91.39
QDA	95.14	95.15	**95.28**
Logistic regression	95.42	95.33	93.61
Naive Bayes	92.50	92.50	92.50
Random forest	95.56	95.54	**95.28**
SVM Linear	**96.94**	**96.88**	95.00
SVM RBF	95.00	94.93	93.61
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	93.89	91.60	
LDA	91.39	**97.63**	
QDA	95.28	95.01	
Logistic regression	93.61	97.12	
Naive Bayes	92.50	92.50	
Random forest	95.28	95.81	
SVM Linear	**98.84**	95.00	
SVM RBF	96.29	93.61	

Table 3.4: Algorithms' performance for Task 6 (subject-based scheme) using PCA.

- Figure 3.7 shows the confusion matrix for SVM Linear.

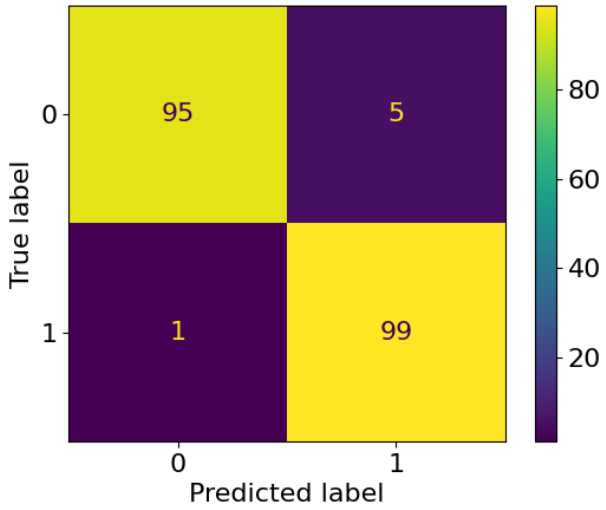
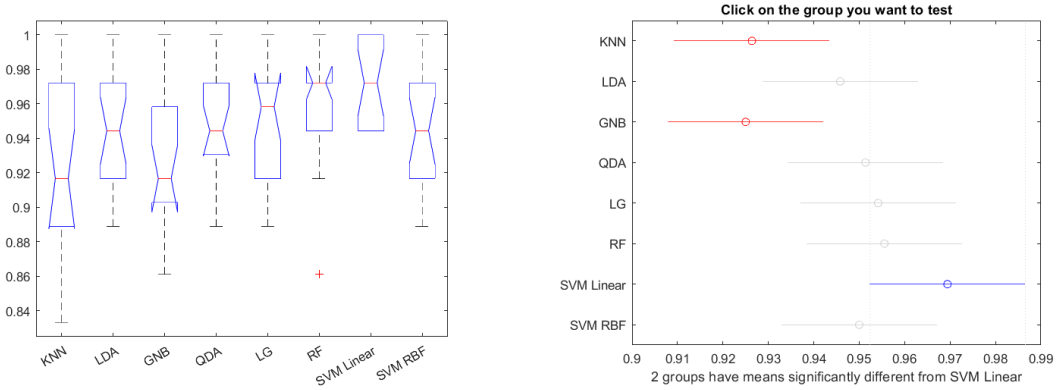


Figure 3.7: Confusion matrix (100%) for SVM (Task 6; subject-based; PCA).

Subject based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.8, which reveals that KNN has a different mean from GNB.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.8: Statistical comparison (Task 6; subject-based; PCA).

3.1.3 Task 7

Trial based: Results

- PCA: Dimensionality was reduced to 6 - 7 components from 15 original features (95% variance threshold).
- Table 3.5 presents the algorithms' performance.

Algorithms	**Accuracy (%)**	**F1 (%)**	**Specificity (%)**
KNN	73.71	71.95	66.86
LDA	74.43	71.36	64.05
QDA	73.43	72.29	**81.81**
Logistic regression	68.57	74.08	72.26
Naive Bayes	74.71	70.35	60.36
Random forest	72.14	70.32	66.95
SVM Linear	**76.43**	**74.10**	69.26
SVM RBF	74.14	71.13	65.81
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	66.67	78.15	
LDA	62.99	82.29	
QDA	**81.07**	65.23	
Logistic regression	71.47	76.90	
Naive Bayes	60.87	**83.33**	
Random forest	65.25	76.24	
SVM Linear	80.27	68.80	
SVM RBF	79.08	64.64	

Table 3.5: Algorithms' performance for Task 7 (trial-based scheme) using PCA.

- Figure 3.9 shows the confusion matrix for SVM Linear.

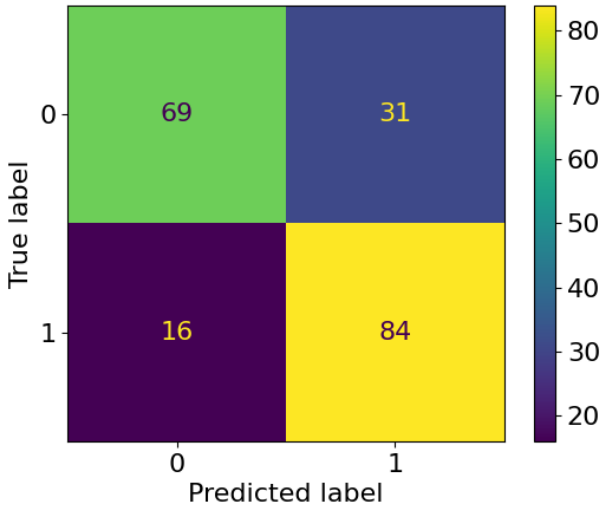
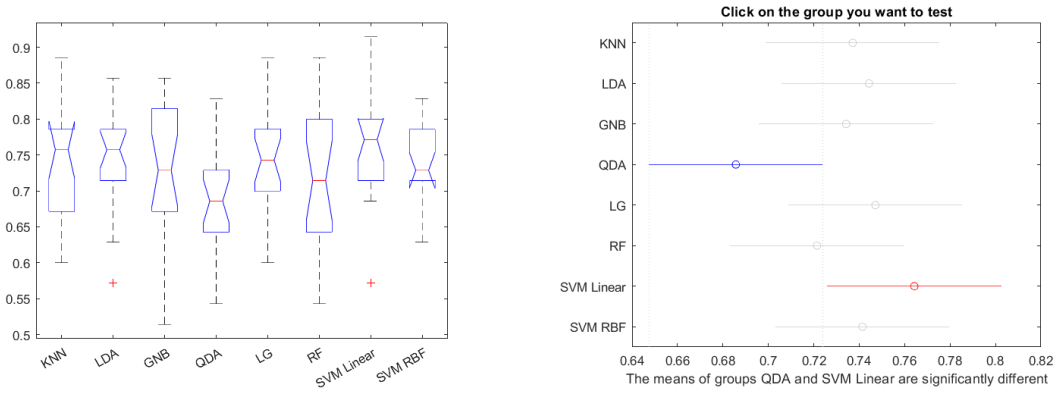


Figure 3.9: Confusion matrix (100%) for SVM Linear (Task 7; trial-based; PCA).

Trial based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.10, which reveals that the means of groups QDA and SVM Linear are significantly different.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.10: Statistical comparison (Task 7; trial-based; PCA).

Subject based: Results

- PCA: Dimensionality was reduced to 6 - 7 components from 15 original features (95% variance threshold).
- Table 3.6 presents the algorithms' performance.

Algorithms	**Accuracy (%)**	**F1 (%)**	**Specificity (%)**
KNN	67.92	65.47	60.83
LDA	**75.69**	72.35	63.61
QDA	73.33	69.03	**81.11**
Logistic regression	63.61	72.92	68.06
Naive Bayes	74.72	69.52	60.83
Random forest	73.61	70.40	62.78
SVM Linear	73.33	**72.41**	70.00
SVM RBF	73.47	71.70	67.22
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	60.83	70.87	
LDA	63.61	**83.88**	
QDA	**81.11**	60.08	
Logistic regression	68.06	78.53	
Naive Bayes	60.83	81.11	
Random forest	62.78	80.14	
SVM Linear	75.00	70.00	
SVM RBF	76.83	67.22	

Table 3.6: Algorithms' performance for Task 7 (subject-based scheme) using PCA.

- Figure 3.11 shows the confusion matrix for LDA.

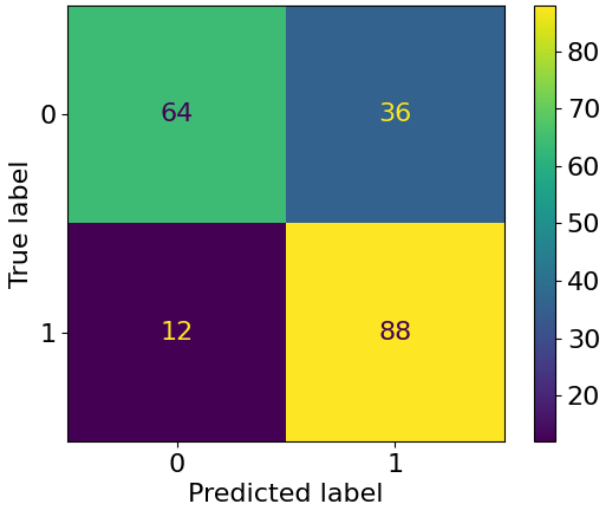
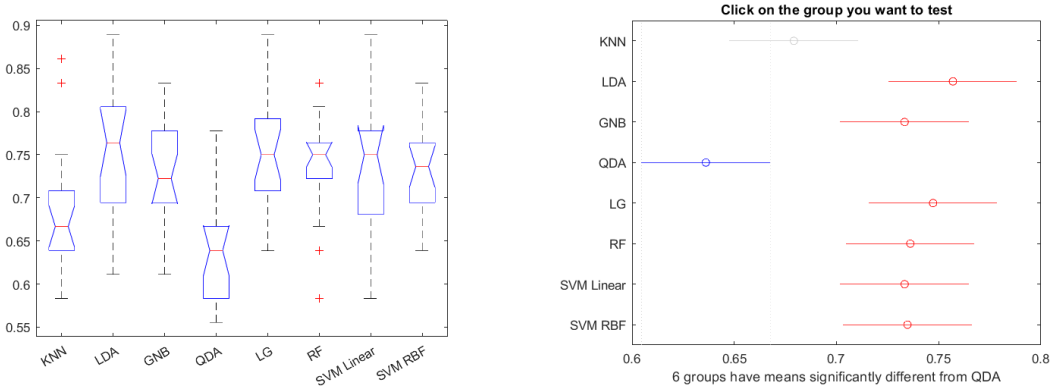


Figure 3.11: Confusion matrix (100%) for LDA (Task 7; subject-based; PCA).

Subject based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.12, which reveals that:

- KNN has a different mean from LDA and LG,
- QDA has a different mean from LDA, GNB, LG, RF, SVM Linear and SVM RBF.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.12: Statistical comparison (Task 7; subject-based; PCA).

3.2 Two class classification (Dim. Reduction: Fisher's Score)

3.2.1 Task 5

Trial based: Results

- Table 3.7 presents the best 6 features. The % denotes how many times each feature was within the top 6 features (selected by Fisher's score) across the 20 classification runs performed.

Feature	%
Average speed of right instrument movement (cm/sec)	90.00
Number of movements of left instrument	70.00
Economy of movement -grasper (%)	65.00
Ideal path length of clipper (cm)	65.00
Number of movements of right instrument	65.00
Relevant path length - clipper(cm)	50.00

Table 3.7: Best 6 features (Task 5; trial-based; Fisher's score).

- Table 3.8 presents the algorithms' performance.

Algorithms	Accuracy (%)	F1 (%)	Specificity (%)
KNN	93.29	93.08	93.42
LDA	92.57	91.98	88.20
QDA	86.86	86.71	89.62
Logistic regression	94.00	93.91	95.96
Naive Bayes	93.86	93.57	92.73
Random forest	94.00	93.77	93.54
SVM Linear	**94.43**	**94.29**	**95.46**
SVM RBF	93.43	93.01	90.77
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	92.67	93.49	
LDA	**96.13**	88.17	
QDA	84.75	88.76	
Logistic regression	92.05	**95.86**	
Naive Bayes	94.56	92.60	
Random forest	94.05	93.49	
SVM Linear	93.33	95.27	
SVM RBF	95.63	90.53	

Table 3.8: Algorithms' performance for Task 5 (trial-based scheme) using Fisher's score.

- Figure 3.13 shows the confusion matrix for SVM Linear.

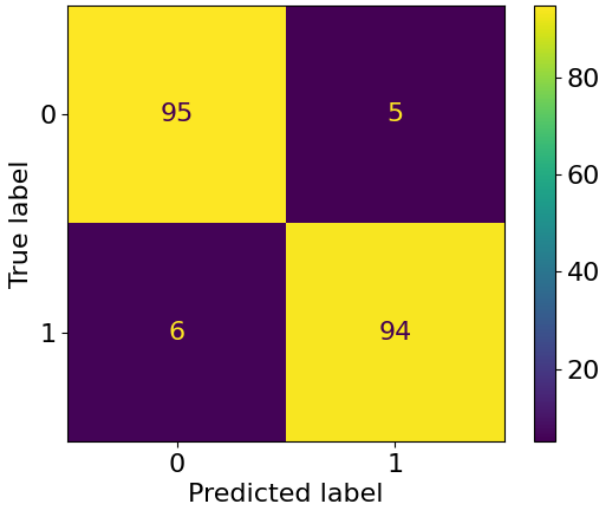
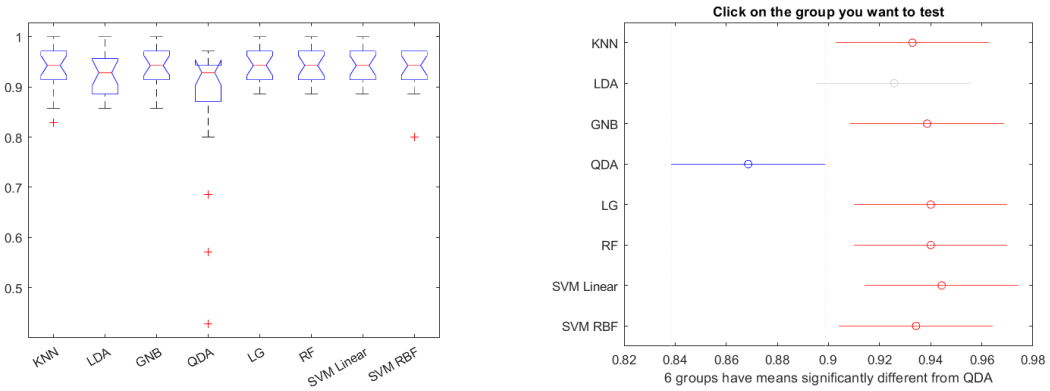


Figure 3.13: Confusion matrix (100%) for SVM Linear (Task 5; trial-based; Fisher's score).

Trial based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.14, which reveals that QDA has a different mean from KNN, GNB, LG, RF, SVM Linear and SVM RBF.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.14: Statistical comparison (Task 5; trial-based; Fisher's score).

Subject based: Results

- Table 3.9 presents the best 6 features. The % denotes how many times each feature was within the top 6 features (selected by Fisher's score) across the 20 classification runs performed.

Feature	%
Number of movements of left instrument	90.00
Ideal path length of clipper (cm)	65.00
Relevant path length - clipper(cm)	65.00
Average speed of right instrument movement (cm/sec)	55.00
Number of clipped ducts	50.00
**Economy of movement -grasper (%)	45.00
**Ideal path length of grasper (cm)	45.00
***NB this feature appeared in the top-6 list < 50% of the experimental runs performed.*	

Table 3.9: Best 6 features (Task 5; subject-based; Fisher's score).

- Table 3.10 presents the algorithms' performance.

Algorithms	Accuracy (%)	F1 (%)	Specificity (%)
KNN	90.69	90.78	91.67
LDA	91.39	91.04	87.50
QDA	83.19	85.04	**95.56**
Logistic regression	**93.61**	**93.73**	**95.56**
Naive Bayes	91.94	91.85	90.83
Random forest	92.64	92.57	91.67
SVM Linear	92.78	92.90	94.44
SVM RBF	92.78	92.59	90.28
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	89.92	91.67	
LDA	94.88	87.50	
QDA	76.61	**95.56**	
Logistic regression	91.98	**95.56**	
Naive Bayes	92.90	90.83	
Random forest	93.48	91.67	
SVM Linear	91.40	94.44	
SVM RBF	**95.03**	90.28	

Table 3.10: Algorithms' performance for Task 5 (subject-based scheme) using Fisher's score.

- Figure 3.15 shows the confusion matrix for Logistic regression.

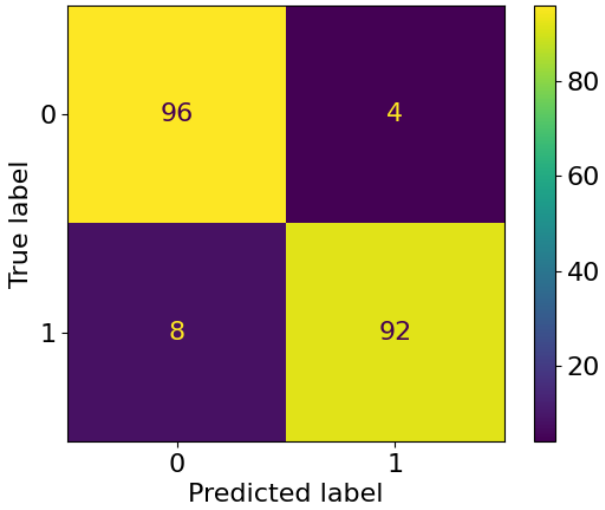
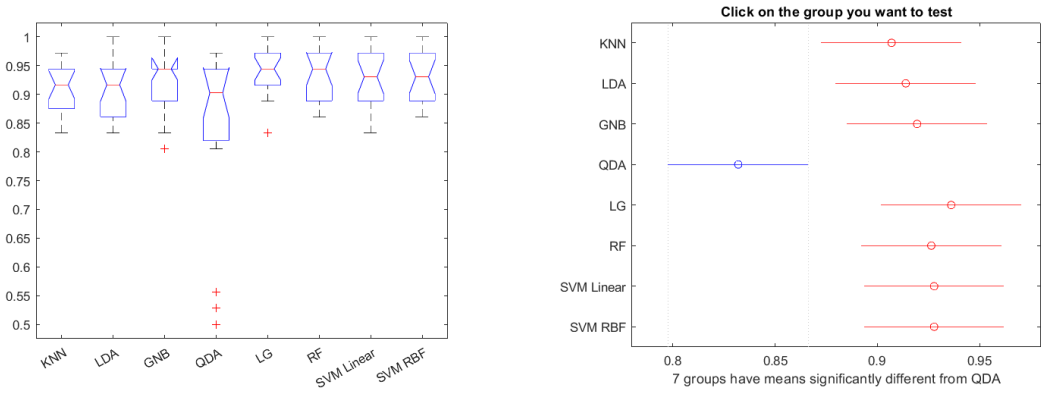


Figure 3.15: Confusion matrix (100%) for Logistic regression (Task 5; subject-based; Fisher's score).

Subject based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.16, which reveals that QDA has a different mean from KNN, LDA, GNB, LG, RF, SVM Linear and SVM RBF.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.16: Statistical comparison (Task 5; subject-based; Fisher's score).

3.2.2 Task 6

Trial based: Results

- Table 3.11 presents the best 6 features. The % denotes how many times each feature was within the top 6 features (selected by Fisher's score) across the 20 classification runs performed.

Feature	%
Number of exposed green balls that are collected	100.00
Average speed of right instrument movement (cm/sec)	95.00
Number of lost balls which miss the basket	90.00
Economy of movement - left instrument (%)	70.00
Number of movements of right instrument	65.00
Ideal path length of left instrument (cm)	50.00

Table 3.11: Best 6 features (Task 6; trial-based; Fisher's score).

- Table 3.12 presents the algorithms' performance.

Algorithms	Accuracy (%)	F1 (%)	Specificity (%)
KNN	93.57	93.71	91.55
LDA	94.00	94.12	91.80
QDA	81.14	80.59	74.03
Logistic regression	95.57	95.72	94.90
Naive Bayes	91.71	91.92	90.14
Random forest	95.29	95.46	**94.93**
SVM Linear	**95.86**	**95.99**	94.79
SVM RBF	95.57	95.68	93.75
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	95.99	91.53	
LDA	96.55	91.80	
QDA	87.26	74.86	
Logistic regression	96.66	94.81	
Naive Bayes	93.75	90.16	
Random forest	96.12	**94.81**	
SVM Linear	97.20	**94.81**	
SVM RBF	**97.72**	93.72	

Table 3.12: Algorithms' performance for Task 6 (trial-based scheme) using Fisher's score.

- Figure 3.17 shows the confusion matrix for SVM Linear.

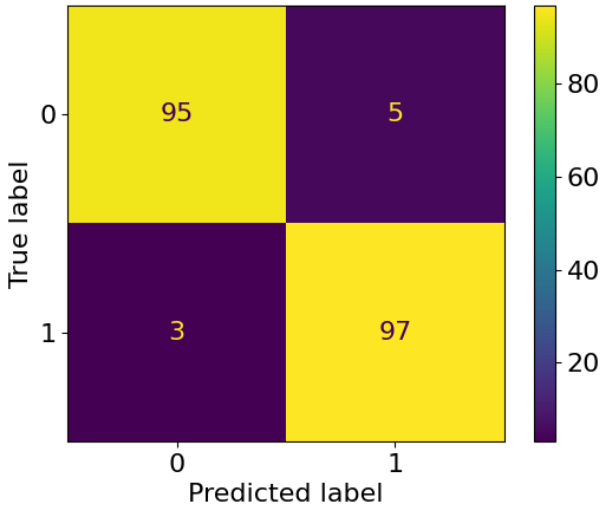
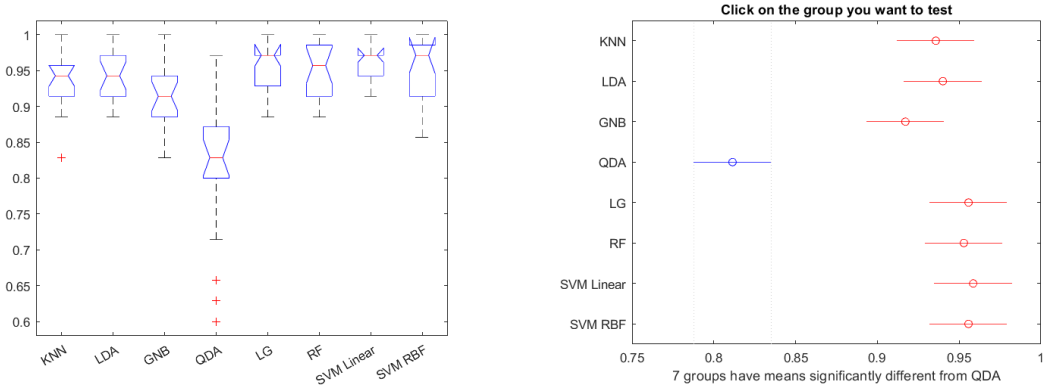


Figure 3.17: Confusion matrix (100%) for SVM Linear (Task 6; trial-based; Fisher's score).

Trial based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.18, which reveals that QDA has a different mean from KNN, LDA, GNB, LG, RF, SVM Linear and SVM RBF.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.18: Statistical comparison (Task 6; trial-based; Fisher's score).

Subject based: Results

- Table 3.13 presents the best 6 features. The % denotes how many times each feature was within the top 6 features (selected by Fisher's score) across the 20 classification runs performed.

Feature	%
Number of lost balls which miss the basket	95.00
Number of exposed green balls that are collected	90.00
Economy of movement - left instrument (%)	85.00
Average speed of right instrument movement (cm/sec)	80.00
Number of movements of right instrument	70.00
Ideal path length of left instrument (cm)	55.00

Table 3.13: Best 6 features (Task 6; subject-based; Fisher's score).

- Table 3.14 presents the algorithms' performance.

Algorithms	Accuracy (%)	F1 (%)	Specificity (%)
KNN	93.89	93.97	95.28
LDA	92.78	92.76	92.50
QDA	82.22	81.82	80.00
Logistic regression	95.14	95.19	96.11
Naive Bayes	91.81	91.95	93.61
Random forest	94.44	94.54	96.11
SVM Linear	94.58	94.68	**96.39**
SVM RBF	**95.28**	**95.29**	95.56
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	92.70	95.28	
LDA	93.02	92.50	
QDA	83.72	80.00	
Logistic regression	94.28	96.11	
Naive Bayes	90.35	93.61	
Random forest	93.01	96.11	
SVM Linear	93.03	**96.39**	
SVM RBF	**95.03**	95.56	

Table 3.14: Algorithms' performance for Task 6 (subject-based scheme) using Fisher's score.

- Figure 3.19 shows the confusion matrix for SVM RBF.

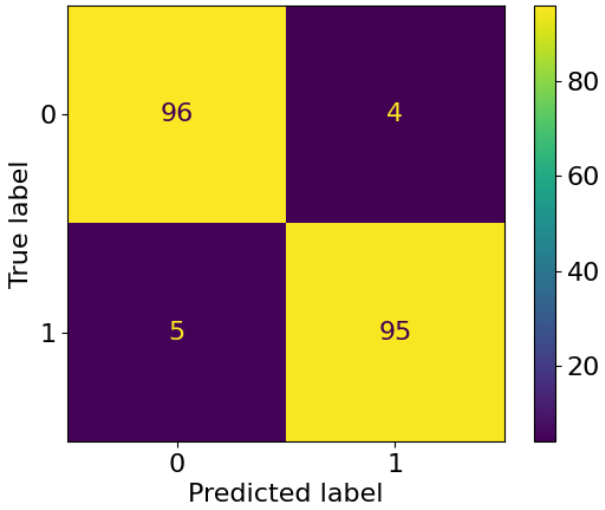
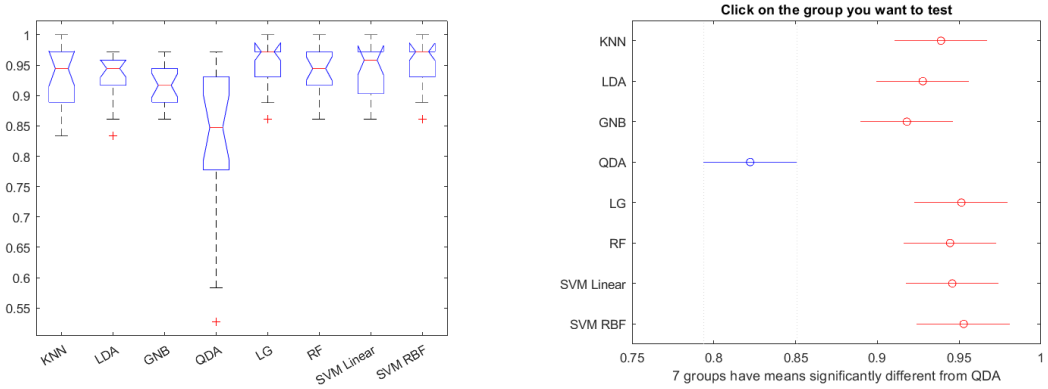


Figure 3.19: Confusion matrix (100%) for SVM RBF (Task 6; subject-based; Fisher's score).

Subject based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.20, which reveals that QDA has a different mean from KNN, LDA, GNB, LG, RF, SVM Linear and SVM RBF.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.20: Statistical comparison (Task 6; subject-based; Fisher's score).

3.2.3 Task 7

Trial based: Results

- Table 3.15 presents the best 6 features. The % denotes how many times each feature was within the top 6 features (selected by Fisher's score) across the 20 classification runs performed.

Feature	%
Average speed of left instrument movement (cm/sec)	100.00
Average speed of right instrument movement (cm/sec)	85.00
Number of cutting maneuvers performed without causing injury	80.00
Safe retraction - overstretch (%)	75.00
Accuracy rate - cuts without injury (%)	60.00
Total time	50.00

Table 3.15: Best 6 features (Task 7; trial-based; Fisher's score).

- Table 3.16 presents the algorithms' performance.

Algorithms	Accuracy (%)	F1 (%)	Specificity (%)
KNN	63.43	63.22	64.01
LDA	71.14	67.21	59.57
QDA	61.57	68.90	**86.11**
Logistic regression	**71.86**	**69.55**	64.95
Naive Bayes	59.43	65.70	78.90
Random forest	69.29	66.77	62.54
SVM Linear	**71.86**	68.58	61.90
SVM RBF	65.86	62.36	58.05
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	62.86	63.58	
LDA	**76.67**	59.83	
QDA	57.42	**86.13**	
Logistic regression	74.75	65.03	
Naive Bayes	56.43	78.61	
Random forest	71.76	62.43	
SVM Linear	76.51	62.14	
SVM RBF	68.51	57.23	

Table 3.16: Algorithms' performance for Task 7 (trial-based scheme) using Fisher's score.

- Figure 3.21 shows the confusion matrix for Logistic regression.

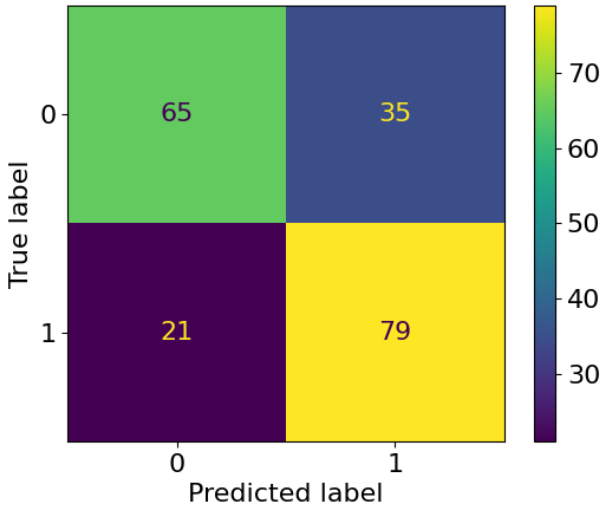
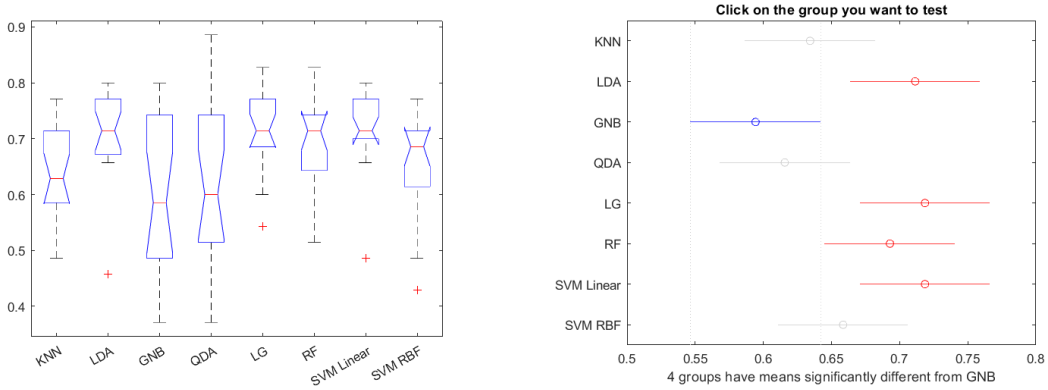


Figure 3.21: Confusion matrix (100%) for Logistic regression (Task 7; trial-based; Fisher's score).

Trial based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.22, which reveals that:

- GNB has a different mean from LDA, LG, RF and SVM Linear,
- QDA has a different mean from LDA, LG and SVM Linear.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.22: Statistical comparison (Task 7; trial-based; Fisher's score).

Subject based: Results

- Table 3.17 presents the best 6 features. The % denotes how many times each feature was within the top 6 features (selected by Fisher's score) across the 20 classification runs performed.

Feature	%
Accuracy rate - cuts without injury (%)	90.00
Average speed of left instrument movement (cm/sec)	85.00
Average speed of right instrument movement (cm/sec)	75.00
Safe retraction - overstretch (%)	70.00
Number of cutting maneuvers performed without causing injury	65.00
Number of movements of left instrument	50.00

Table 3.17: Best 6 features (Task 7; subject-based; Fisher's score).

- Table 3.18 presents the algorithms' performance

Algorithms	Accuracy (%)	F1 (%)	Specificity (%)
KNN	63.33	62.07	60.00
LDA	70.83	67.08	59.44
QDA	52.92	67.31	**96.94**
Logistic regression	72.08	**69.87**	64.72
Naive Bayes	55.00	66.80	90.56
Random forest	68.47	66.76	63.33
SVM Linear	**71.94**	69.21	63.06
SVM RBF	67.92	63.39	55.56
	Sensitivity (Recall) (%)	**Precision (%)**	
KNN	64.29	60.00	
LDA	**76.98**	59.44	
QDA	51.55	**96.94**	
Logistic regression	75.90	64.72	
Naive Bayes	52.92	90.56	
Random forest	70.59	63.33	
SVM Linear	76.69	63.06	
SVM RBF	73.80	55.56	

Table 3.18: Algorithms' performance for Task 7 (subject-based scheme) using Fisher's score.

- Figure 3.23 shows the confusion matrix for SVM Linear.

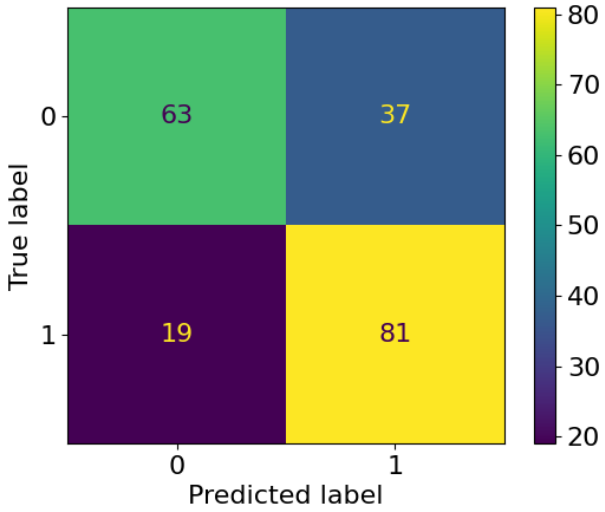
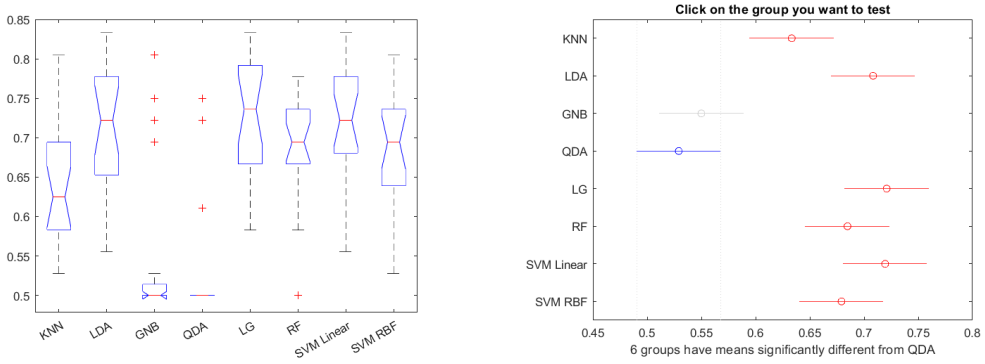


Figure 3.23: Confusion matrix (100%) for SVM Linear (Task 7; subject-based; Fisher's score).

Subject based: Statistical comparison

The findings of the statistical comparison are presented in Figure 3.24, which reveals that:

- KNN has a different mean from GNB, QDA, LG and SVM Linear,
- GNB has a different mean from KNN, LDA, LG, RF, SVM Linear and SVM RBF,
- QDA has a different mean from KNN, LDA, GNB, LG, RF, SVM Linear and SVM RBF.



(a) Anova boxplot.

(b) Multiple comparison graph.

Figure 3.24: Statistical comparison (Task 7; subject-based; Fisher's score).

Chapter 4

Discussion

Upon analyzing all the collected results, several conclusions can be drawn.

Firstly, it is evident that the accuracy percentages of Task 7 is lower than that of the other tasks, regardless of whether the data was split by trial or by subject and whether PCA or Fisher's score was used (Table 4.1 shows these differences). This can be attributed to the fact that by the time Task 7 is reached, the student has already gained experience with the different tasks in the module. Therefore, there is little variation in the feature values between the first and last attempts, resulting in reduced accuracy. Additionally, since the Pass threshold for Task 7 is only one feature (i.e., the total number of cutting maneuvers), the feature is affected by the presence of other features, further reducing the accuracy.

	Task 5	**Task 6**	**Task 7**
	Accuracy	*Accuracy*	*Accuracy*
Trial based (PCA)	96.14	97.29	76.43
Subject based (PCA)	97.08	96.94	73.33
Trial based (Fisher's score)	94.43	95.86	71.86
Subject based (Fisher's score)	92.78	94.58	71.94

Table 4.1: Example of accuracy percentages for Task 5, 6 and 7 for the SVM Linear algorithm.

Furthermore, the accuracy percentage for subject-based scripts, in most cases, was slightly lower than that of trial-based scripts (see Table 4.2). Although the difference was not significant, it was noticeable, as the algorithm had not been exposed to a specific student's trials during the training phase. As a result, the algorithm was presented with entirely new data during the prediction task, potentially impacting the accuracy percentage.

	Accuracy
Trial based (PCA)	92.14
Subject based (PCA)	89.17
Trial based (Fisher's score)	92.57
Subject based (Fisher's score)	91.39

Table 4.2: Example of accuracy percentages for Task 5 for the LDA algorithm.

Table 4.3 was created by consolidating all the available data, and it showcases the most effective algorithms for each task, as determined by their accuracy percentages.

	Task 5			
	PCA		*Fisher's score*	
	Accuracy (%)	*Algorithm*	*Accuracy (%)*	*Algorithm*
Trial based	96.14	SVM Linear	94.43	SVM Linear
Subject based	97.08	SVM Linear	93.61	Logistic regression
	Task 6			
	PCA		*Fisher's score*	
	Accuracy (%)	*Algorithm*	*Accuracy (%)*	*Algorithm*
Trial based	97.29	SVM Linear	95.86	SVM Linear
Subject based	96.94	SVM Linear	95.28	SVM RBF
	Task 7			
	PCA		*Fisher's score*	
	Accuracy (%)	*Algorithm*	*Accuracy (%)*	*Algorithm*
Trial based	76.43	SVM Linear	71.86	SVM Linear, Logistic regression
Subject based	75.69	LDA	72.08	Logistic regression

Table 4.3: Most effective algorithms for each task.

It is evident that the Support Vector Machine with a linear kernel outperformed the other algorithms, achieving the highest overall accuracy percentage.

Regarding the features of each task, the analysis revealed that there was a unanimous agreement on the top six features for Task 6. However, for Tasks 5 and 7, there was no clear consensus on the sixth best feature. Table 4.4 presents the best six features for each task.

Task 5	
Feature	*%*
Average speed of right instrument movement (cm/sec)	72.50
Number of movements of left instrument	80.00
Ideal path length of clipper (cm)	65.00
Relevant path length - clipper(cm)	57.50
Economy of movement -grasper (%)	55.00
*Number of movements of right instrument	65.00
*Number of clipped ducts	50.00
Task 6	
Feature	*%*
Number of exposed green balls that are collected	95.00
Average speed of right instrument movement (cm/sec)	87.50
Number of lost balls which miss the basket	92.50
Economy of movement - left instrument (%)	77.50
Number of movements of right instrument	67.50
Ideal path length of left instrument (cm)	52.50
Task 7	
Feature	*%*
Average speed of left instrument movement (cm/sec)	92.50
Average speed of right instrument movement (cm/sec)	87.50
Number of cutting maneuvers performed without causing injury	72.50
Safe retraction - overstretch (%)	72.50
Accuracy rate - cuts without injury (%)	75.00
*Total time	50.00
*Number of movements of left instrument	50.00
**Appeared only in trial or subject based script.*	

Table 4.4: The six best features for each task.

Chapter 5

Conclusion

In summary, this thesis highlights the potential benefits of using virtual reality simulators and machine learning algorithms to evaluate and enhance laparoscopy training for medical students. By utilizing the Lap Mentor VR simulator, students can receive customized feedback on their progress, while educators can adjust their teaching to cater to the specific needs of each student. The study concentrated on three particular tasks and used the collected data to create scripts that classify students into "*Start of training*" and "*End of training*" categories.

Although the accuracy of the algorithms varied depending on the task and the method of data splitting used, Task 7 had lower accuracy percentages than the other tasks, mainly due to the reduced variation in feature values between the first and last attempts. Additionally, *subject-based* scripts had a slightly lower accuracy percentage than *trial-based* scripts, indicating that the algorithm's familiarity with a particular student's trials during the training phase could positively affect the accuracy percentage during the prediction task.

Moreover, the results reveal that the Support Vector Machine with a linear kernel outperformed the other algorithms in terms of overall accuracy percentage. On the other hand, the Quadratic Discriminant Analysis algorithm achieved the highest specificity percentage, while the Support Vector Machine with a linear kernel algorithm had the best sensitivity percentage.

Furthermore, the analysis of the features for each task indicated a unanimous agreement on the top six features (performance metrics) for Task 6, while Tasks 5 and 7 had no clear consensus on the sixth best feature.

By adopting this technology, educators can deliver personalized feedback to students based on their specific needs, thereby improving the overall quality of medical education. This approach could also be applied to other medical fields, leading to a more comprehensive and efficient training process for healthcare professionals.

Although this study provides valuable insights, there is still room for further research and improvement. Future studies could concentrate on expanding the sample size and incorporating more diverse tasks to achieve a more comprehensive analysis of laparoscopy training.

In conclusion, this thesis highlights the potential for innovative technologies and machine learning algorithms to revolutionize medical education and enhance the training of healthcare professionals.

Bibliography

[1] R Aggarwal, P Crochet, A Dias, A Misra, P Ziprin, and A Darzi. Development of a virtual reality training curriculum for laparoscopic cholecystectomy. *Journal of British Surgery*, 96(9):1086–1093, 2009.

[2] R Aggarwal, K Moorthy, and A Darzi. Laparoscopic skills training and assessment. *Journal of British Surgery*, 91(12):1549–1558, 2004.

[3] Richard O Duda, Peter E Hart, et al. *Pattern classification*. John Wiley & Sons, 2006.

[4] Mahtab J Fard, Sattar Ameri, R Darin Ellis, Ratna B Chinnam, Abhilash K Pandya, and Michael D Klein. Automated robot-assisted surgical skill evaluation: Predictive analytics approach. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 14(1):e1850, 2018.

[5] A. Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Incorporated, 2019.

[6] Karl-Friedrich Kowalewski, Carly R Garrow, Mona W Schmidt, Laura Benner, Beat P Müller-Stich, and Felix Nickel. Sensor-based machine learning for workflow detection and as key to detect expert level in laparoscopic suturing and knot-tying. *Surgical endoscopy*, 33:3732–3740, 2019.

[7] Vasilelios Lahanas, Evangelos Georgiou, and Constantinos Loukas. Surgical simulation training systems: box trainers, virtual reality and augmented reality simulators. *International Journal of Advanced Robotics and Automation*, 1(2):1–9, 2016.

[8] C. Loukas and E. Georgiou. Multivariate autoregressive modeling of hand kinematics for laparoscopic skills assessment of surgical trainees. *IEEE Transactions on Biomedical Engineering*, 58(11):3289–3297, November 2011.

[9] JA Martin, Glenn Regehr, Richard Reznick, Helen Macrae, John Murnaghan, Carol Hutchison, and M Brown. Objective structured assessment of technical skill (osats) for surgical residents. *British journal of surgery*, 84(2):273–278, 1997.

[10] Issam El Naqa, Ruijiang Li, and Martin J. Murphy, editors. *Machine Learning in Radiation Oncology*. Springer International Publishing, 2015.

[11] Simon Parsons. *Introduction to Machine Learning, Second Editon by Ethem Alpaydin, MIT Press, 584 pp., ISBN 978-0-262-01243-0*, volume 25. Cambridge University Press, 2010.

[12] 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.

[13] Armin Shmilovici. *Support vector machines*. Springer, 2005.

[14] Melina C Vassiliou, Liane S Feldman, Christopher G Andrew, Simon Bergman, Karen Leffondré, Donna Stanbridge, and Gerald M Fried. A global assessment tool for evaluation of intraoperative laparoscopic skills. *The American journal of surgery*, 190(1):107–113, 2005.

[15] MS Wilson, A Middlebrook, C Sutton, R Stone, and RF McCloy. Mist vr: a virtual reality trainer for laparoscopic surgery assesses performance. *Annals of the Royal College of Surgeons of England*, 79(6):403, 1997.

[16] Alexander Winkler-Schwartz, Vincent Bissonnette, Nykan Mirchi, Nirros Ponnudurai, Recai Yilmaz, Nicole Ledwos, Samaneh Siyar, Hamed Azarnoush, Bekir Karlik, and Rolando F Del Maestro. Artificial intelligence in medical education: best practices using machine learning to assess surgical expertise in virtual reality simulation. *Journal of surgical education*, 76(6):1681–1690, 2019.

[17] Alexander Winkler-Schwartz, Recai Yilmaz, Nykan Mirchi, Vincent Bissonnette, Nicole Ledwos, Samaneh Siyar, Hamed Azarnoush, Bekir Karlik, and Rolando Del Maestro. Machine learning identification of surgical and operative factors associated with surgical expertise in virtual reality simulation. *JAMA network open*, 2(8):e198363–e198363, 2019.

Acronyms-Abbreviations

AI: Artificial intelligence

ET: End of Training

FLS: Fundamentals of Laparoscopic Surgery

GOALS: Global Operative Assessment of Laparoscopic Skills

kNN: k-Nearest Neighbors

LDA: Linear Discriminant Analysis

MIS: Minimally invasive surgery

MISTELS: McGill Inanimate System for Training and Evaluation of Laparoscopic Skills

ML: Machine learning

MLASE: Machine Learning to Assess Surgical Expertise

MLP: Multilayer perceptron

NB: Naive Bayes Classifier

OSATS: Objective Structured Assessment of Technical Skill

PCA: Principal Component Analysis

PR: Physical Reality

QDA: Quadratic Discriminant Analysis

ST: Start of training

SVM: Support Vector Machine

VR: Virtual reality

Appendix

5.1 Confusion matrices

5.1.1 Two class classification (Dim. Reduction: PCA)

Task 5: Trial based

- KNN

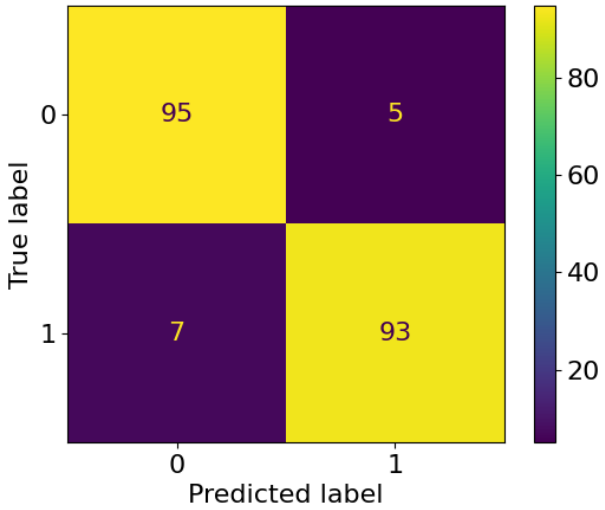


Figure 5.1: Confusion matrix (100%) for KNN (Task 5; trial-based; PCA).

- LDA

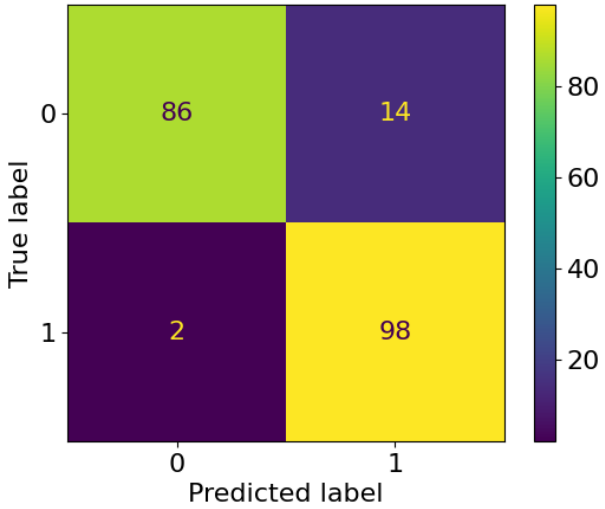


Figure 5.2: Confusion matrix (100%) for LDA (Task 5; trial-based; PCA).

- QDA

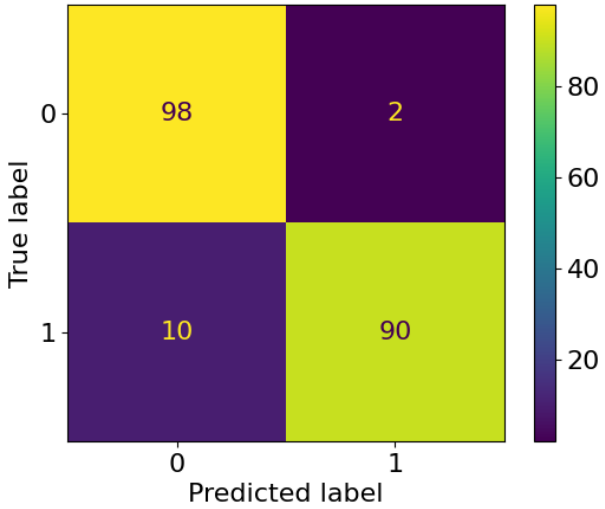


Figure 5.3: Confusion matrix (100%) for QDA (Task 5; trial-based; PCA).

- Logistic regression

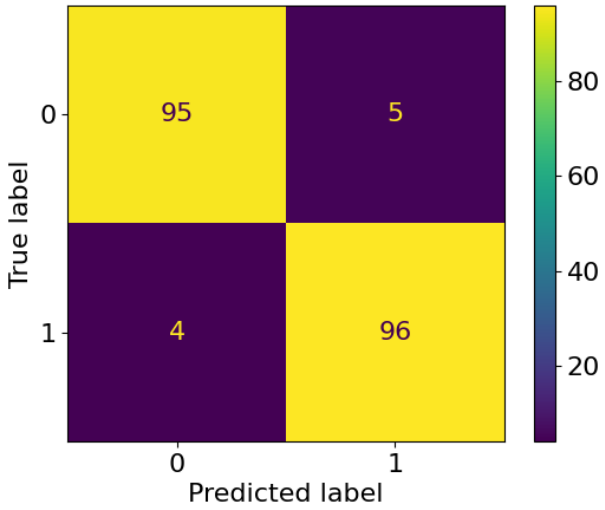


Figure 5.4: Confusion matrix (100%) for Logistic regression (Task 5; trial-based; PCA).

- Naive Bayes

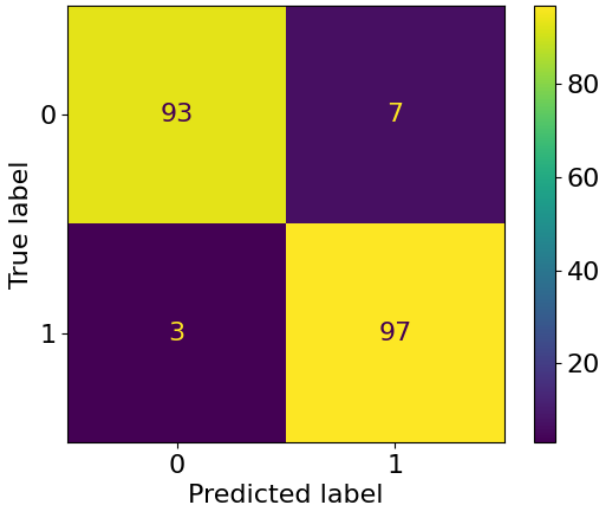


Figure 5.5: Confusion matrix (100%) for Naïve Bayes (Task 5; trial-based; PCA).

- Random forest

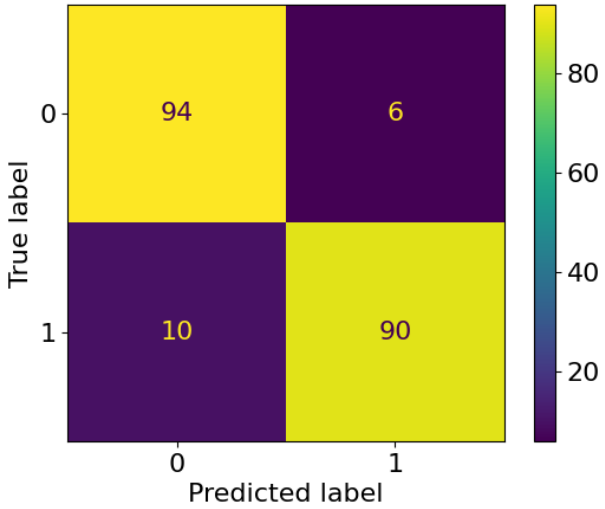


Figure 5.6: Confusion matrix (100%) for Random forest (Task 5; trial-based; PCA).

- SVM RBF

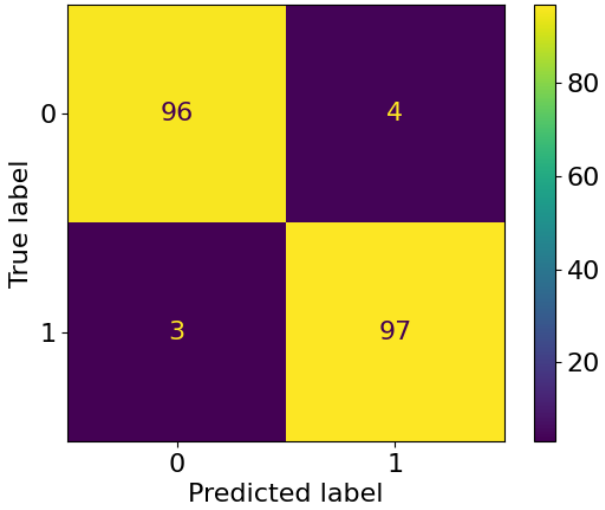


Figure 5.7: Confusion matrix (100%) for SVM RBF (Task 5; trial-based; PCA).

Task 5: Subject based

- KNN

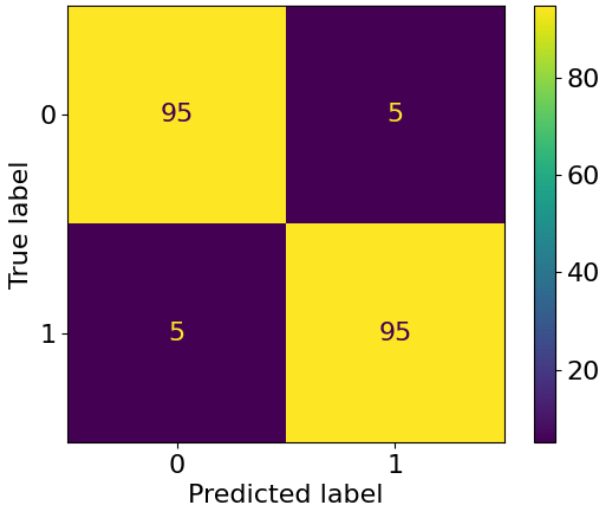


Figure 5.8: Confusion matrix (100%) for KNN (Task 5; subject-based; PCA).

- LDA

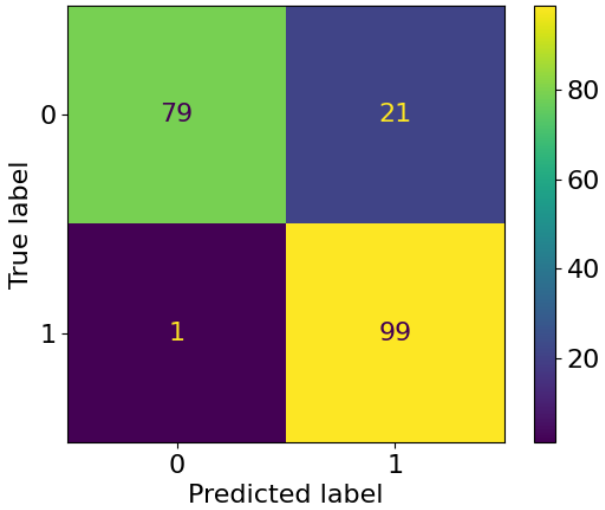


Figure 5.9: Confusion matrix (100%) for LDA (Task 5; subject-based; PCA).

- QDA

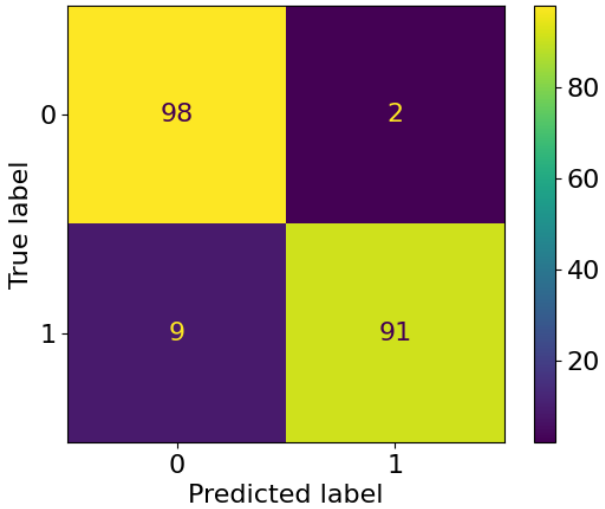


Figure 5.10: Confusion matrix (100%) for QDA (Task 5; subject-based; PCA).

- Logistic regression

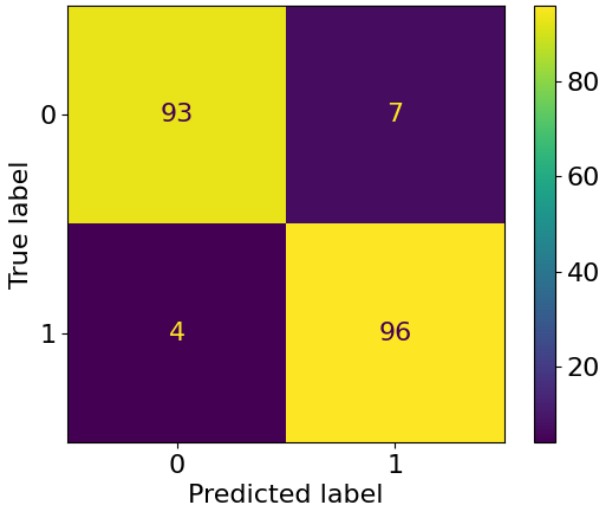


Figure 5.11: Confusion matrix (100%) for Logistic regression (Task 5; subject-based; PCA).

- Naive Bayes

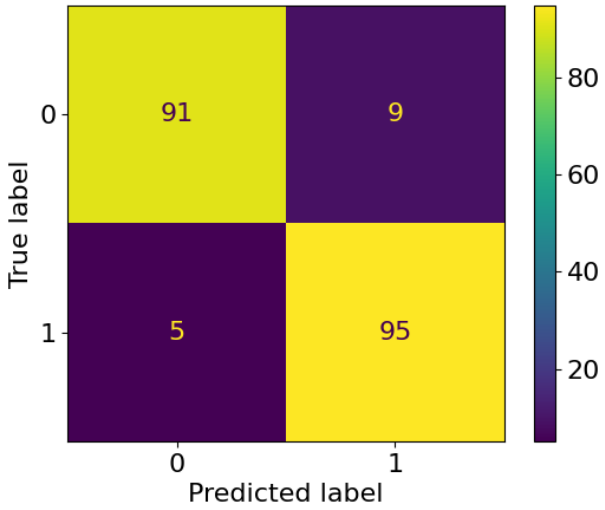


Figure 5.12: Confusion matrix (100%) for Naïve Bayes (Task 5; subject-based; PCA).

- Random forest

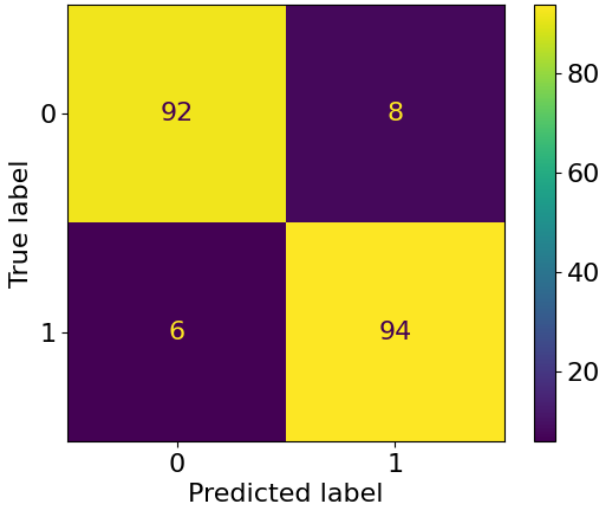


Figure 5.13: Confusion matrix (100%) for Random forest (Task 5; subject-based; PCA).

- SVM RBF

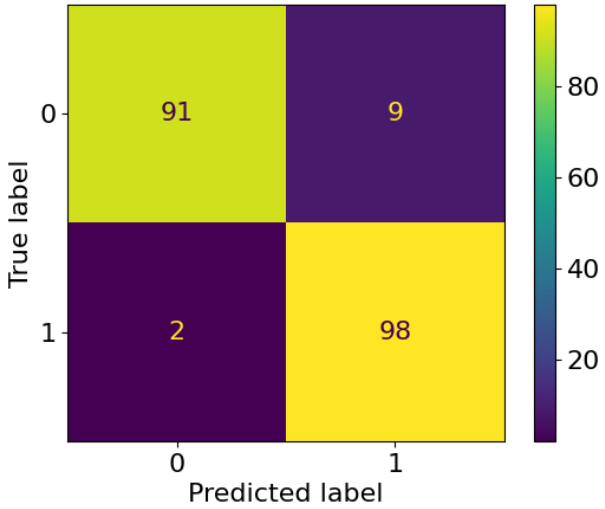


Figure 5.14: Confusion matrix (100%) for SVM RBF (Task 5; subject-based; PCA).

Task 6: Trial based

- KNN

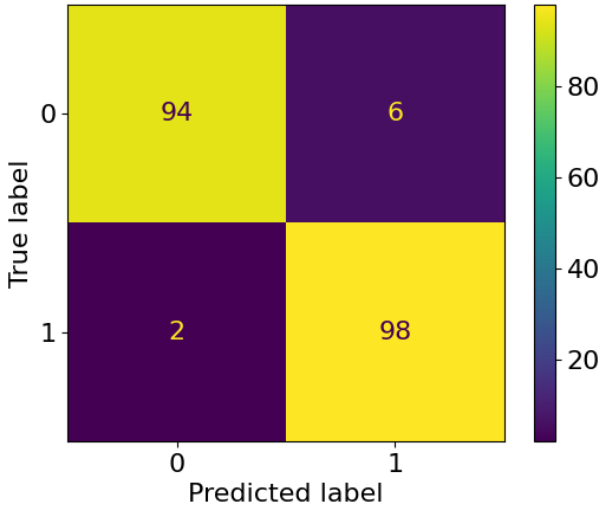


Figure 5.15: Confusion matrix (100%) for KNN (Task 6; trial-based; PCA).

- LDA

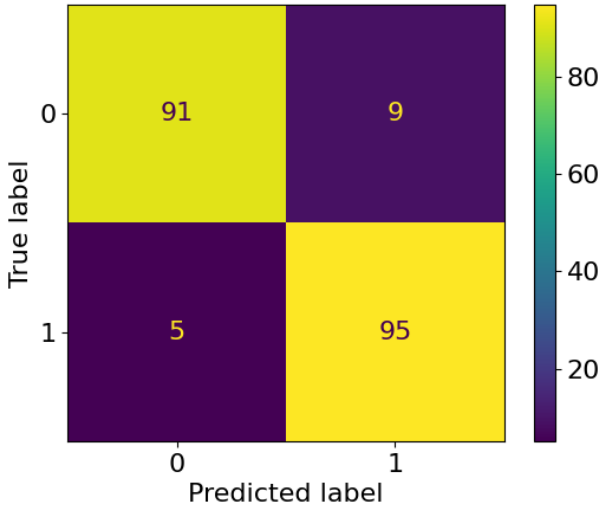


Figure 5.16: Confusion matrix (100%) for LDA (Task 6; trial-based; PCA).

- QDA

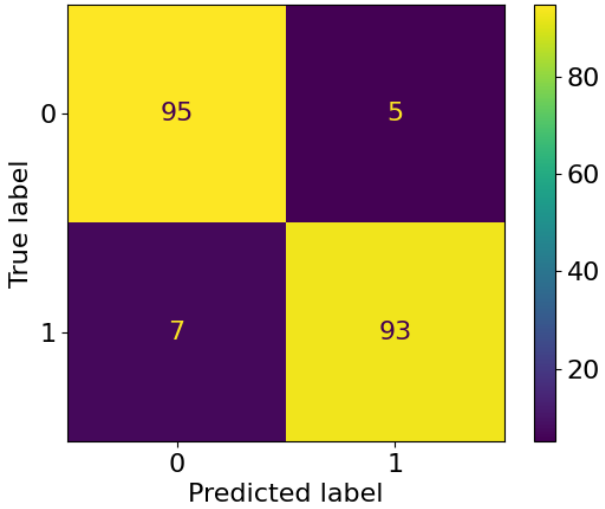


Figure 5.17: Confusion matrix (100%) for QDA (Task 6; trial-based; PCA).

- Logistic regression

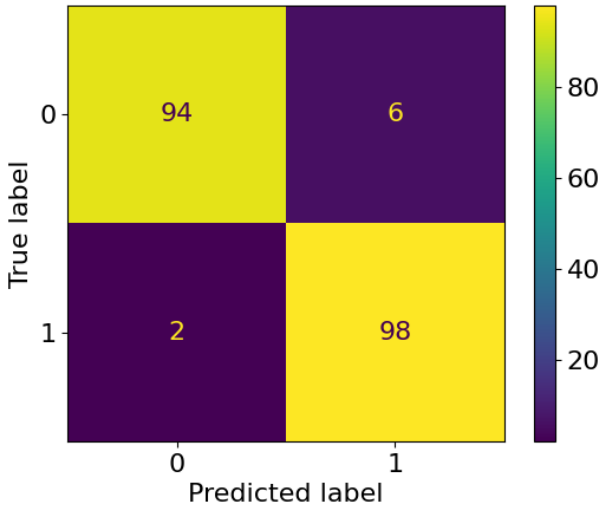


Figure 5.18: Confusion matrix (100%) for Logistic regression (Task 6; trial-based; PCA).

- Naive Bayes

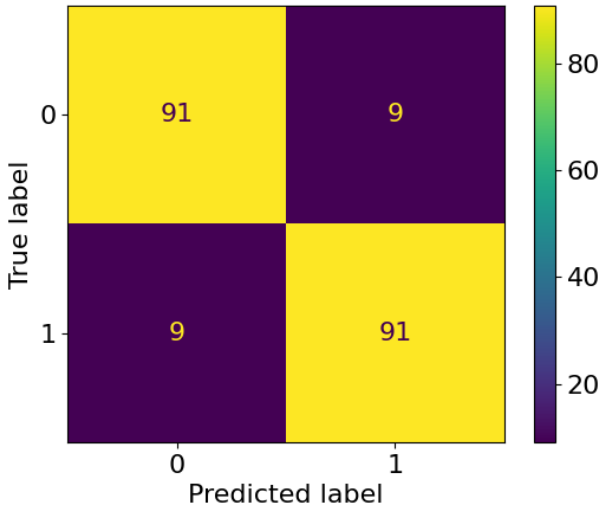


Figure 5.19: Confusion matrix (100%) for Naïve Bayes (Task 6; trial-based; PCA).

- Random forest

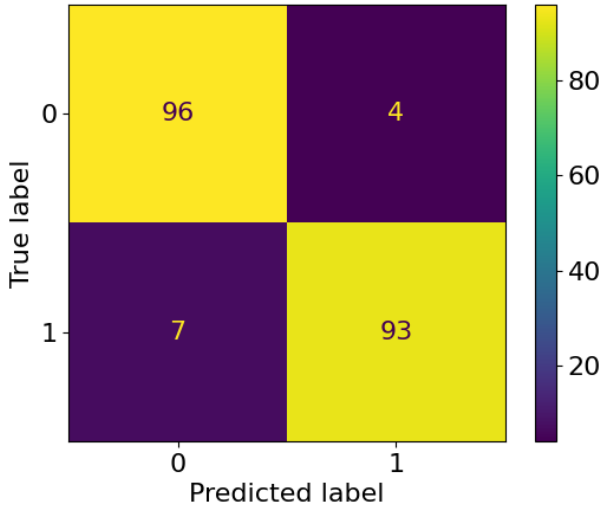


Figure 5.20: Confusion matrix (100%) for Random forest (Task 6; trial-based; PCA).

- SVM RBF

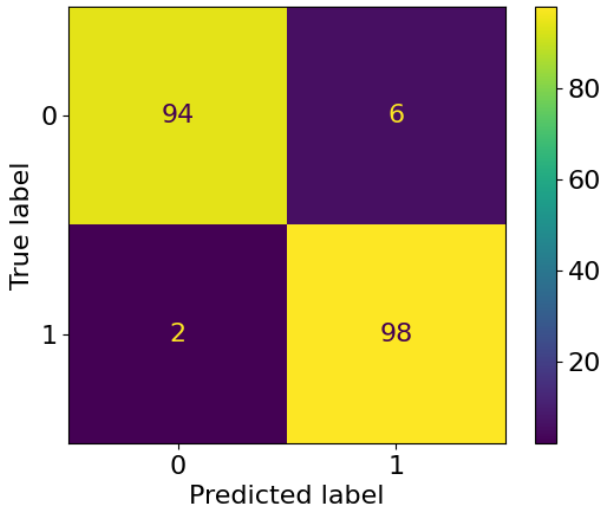


Figure 5.21: Confusion matrix (100%) for SVM RBF (Task 6; trial-based; PCA).

Task 6: Subject based

- KNN

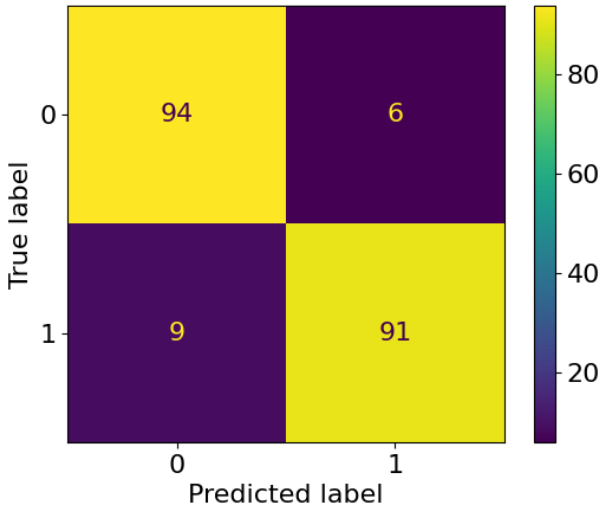


Figure 5.22: Confusion matrix (100%) for KNN (Task 6; subject-based; PCA).

- LDA

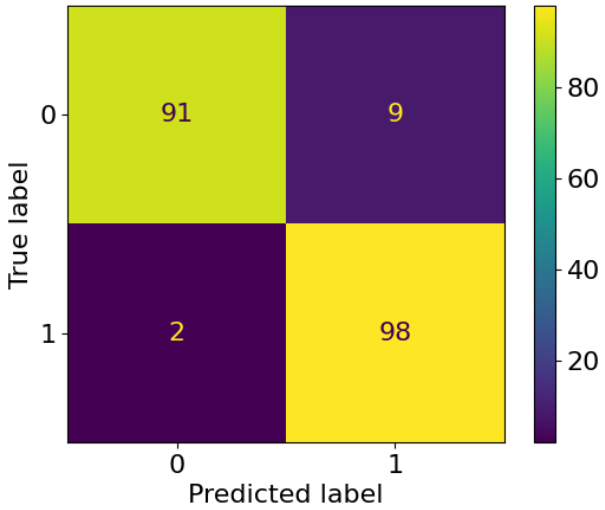


Figure 5.23: Confusion matrix (100%) for LDA (Task 6; subject-based; PCA).

- QDA

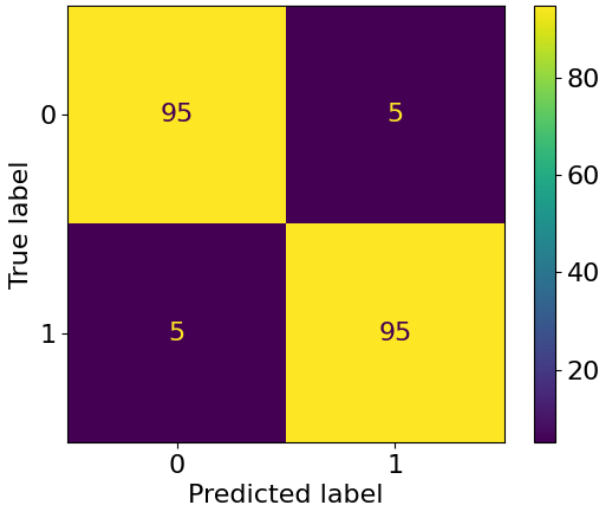


Figure 5.24: Confusion matrix (100%) for QDA (Task 6; subject-based; PCA).

- Logistic regression

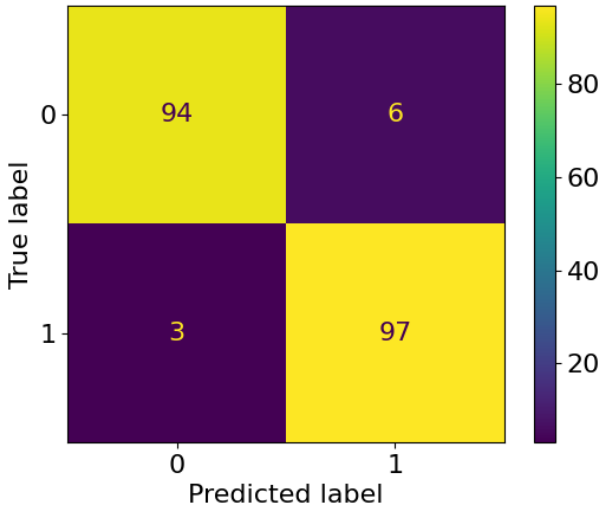


Figure 5.25: Confusion matrix (100%) for Logistic regression (Task 6; subject-based; PCA).

- Naive Bayes

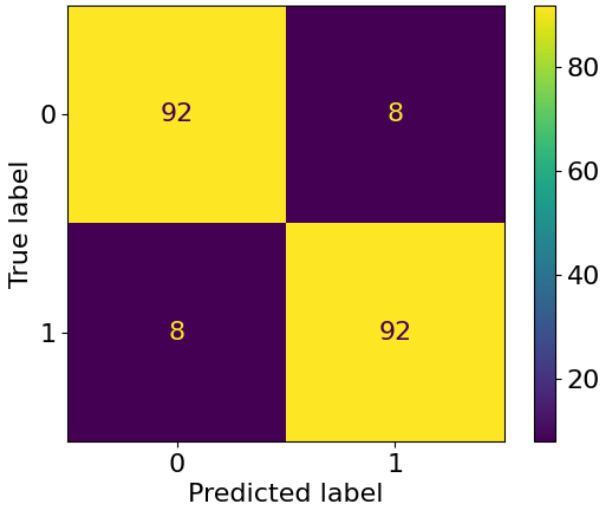


Figure 5.26: Confusion matrix (100%) for Naïve Bayes (Task 6; subject-based; PCA).

- Random forest

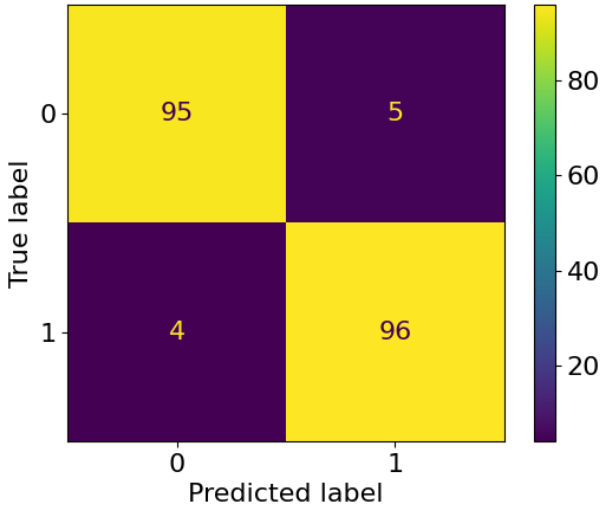


Figure 5.27: Confusion matrix (100%) for Random forest (Task 6; subject-based; PCA).

- SVM RBF

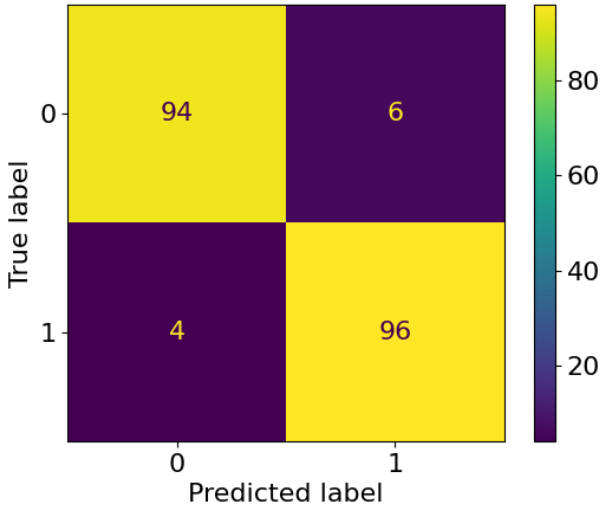


Figure 5.28: Confusion matrix (100%) for SVM RBF (Task 6; subject-based; PCA).

Task 7: Trial based

- KNN

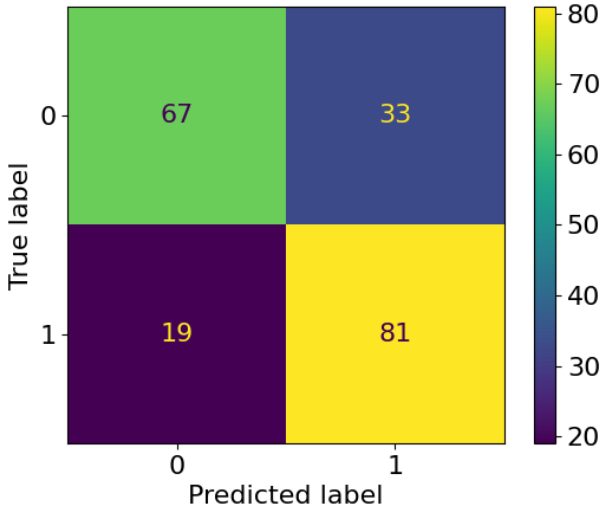


Figure 5.29: Confusion matrix (100%) for KNN (Task 7; trial-based; PCA).

- LDA

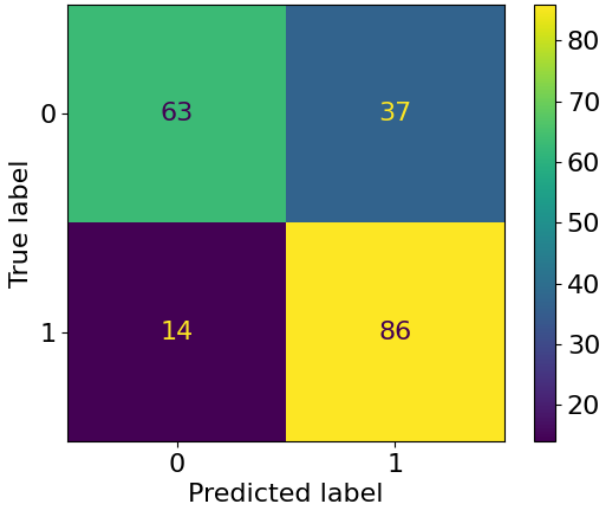


Figure 5.30: Confusion matrix (100%) for LDA (Task 7; trial-based; PCA).

- QDA

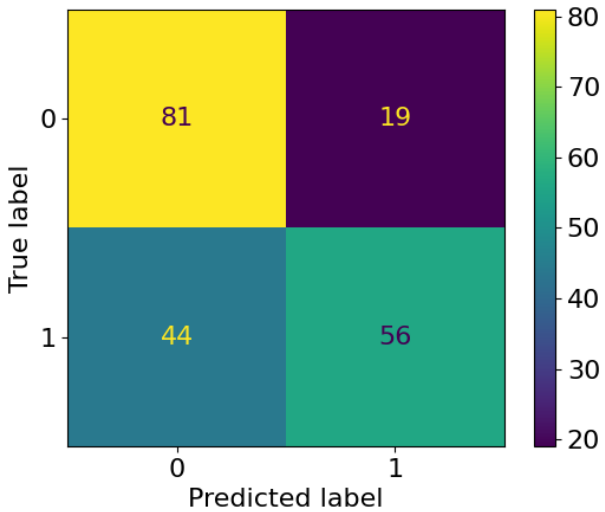


Figure 5.31: Confusion matrix (100%) for QDA (Task 7; trial-based; PCA).

- Logistic regression

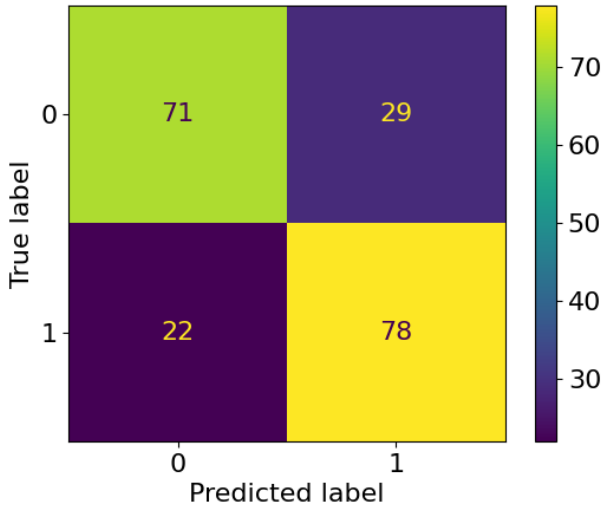


Figure 5.32: Confusion matrix (100%) for Logistic regression (Task 7; trial-based; PCA).

- Naive Bayes

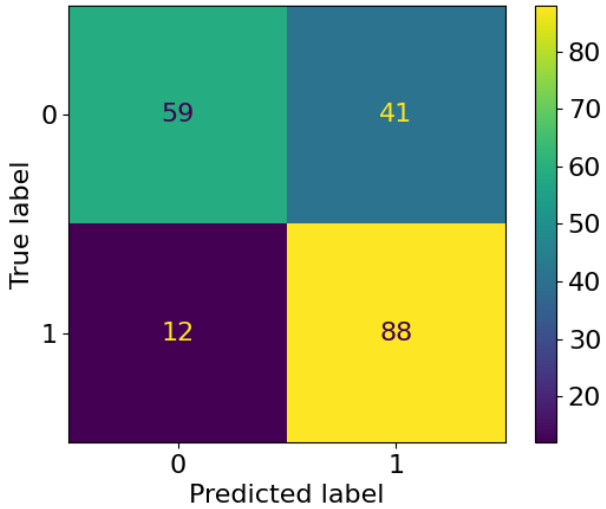


Figure 5.33: Confusion matrix (100%) for Naïve Bayes (Task 7; trial-based; PCA).

- Random forest

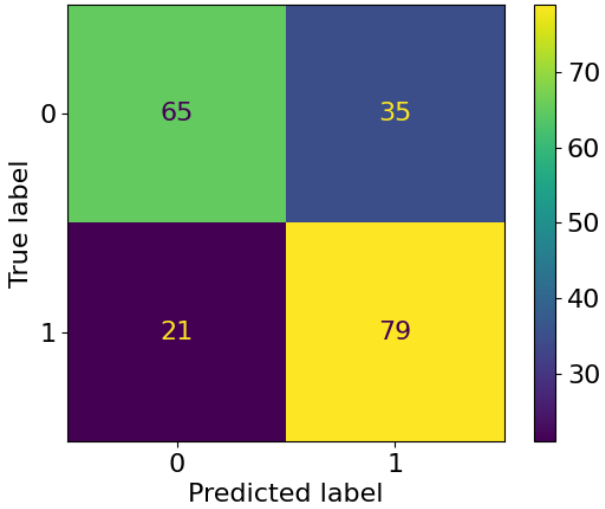


Figure 5.34: Confusion matrix (100%) for Random forest (Task 7; trial-based; PCA).

- SVM RBF

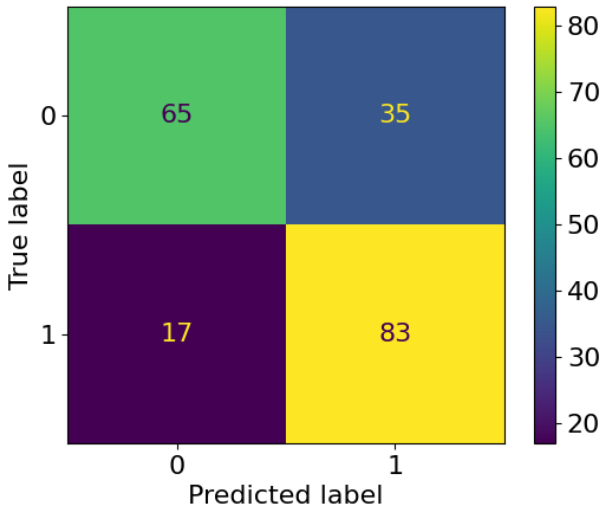


Figure 5.35: Confusion matrix (100%) for SVM RBF (Task 7; trial-based; PCA).

Task 7: Subject based

- KNN

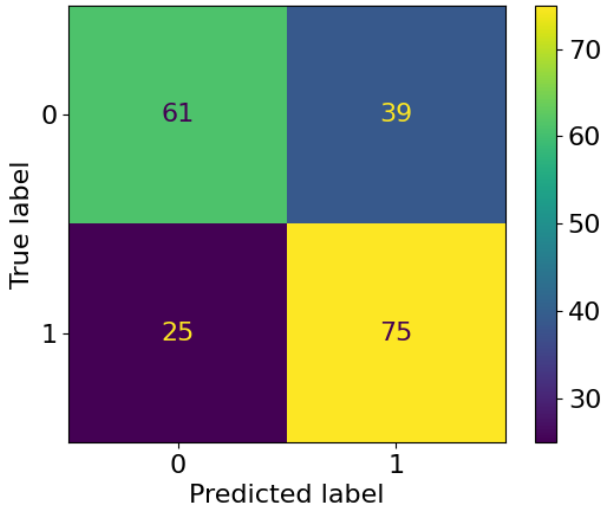


Figure 5.36: Confusion matrix (100%) for KNN (Task 7; subject-based; PCA).

- QDA

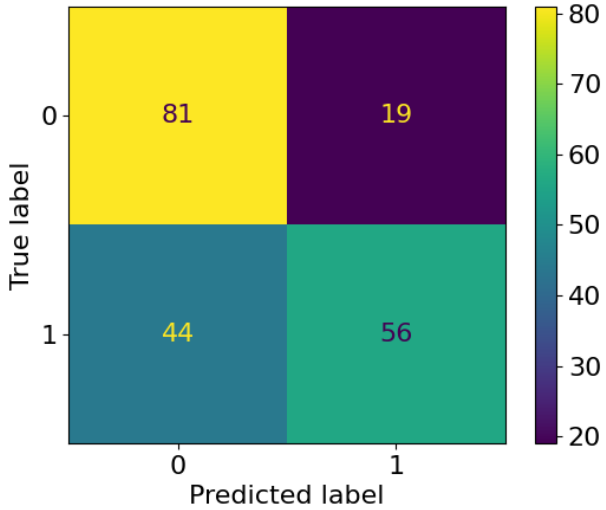


Figure 5.37: Confusion matrix (100%) for QDA (Task 7; subject-based; PCA).

- Logistic regression

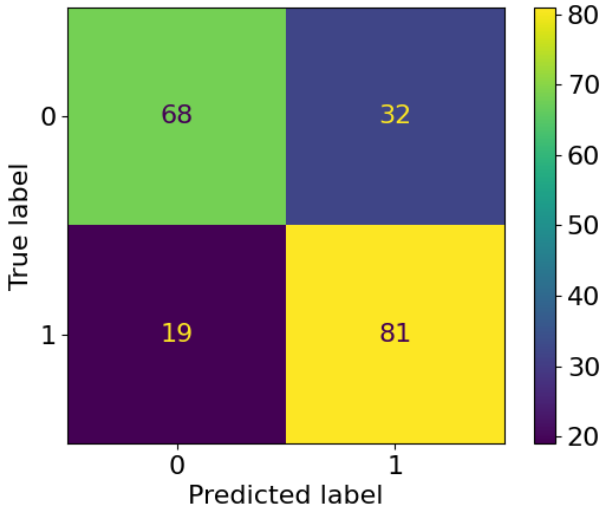


Figure 5.38: Confusion matrix (100%) for Logistic regression (Task 7; subject-based; PCA).

- Naive Bayes

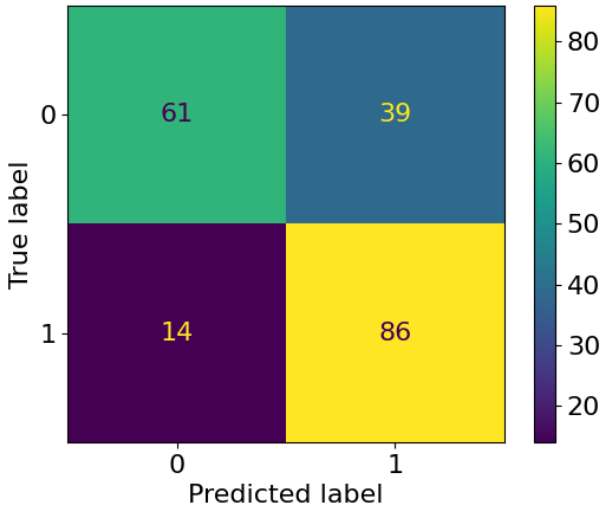


Figure 5.39: Confusion matrix (100%) for Naïve Bayes (Task 7; subject-based; PCA).

- Random forest

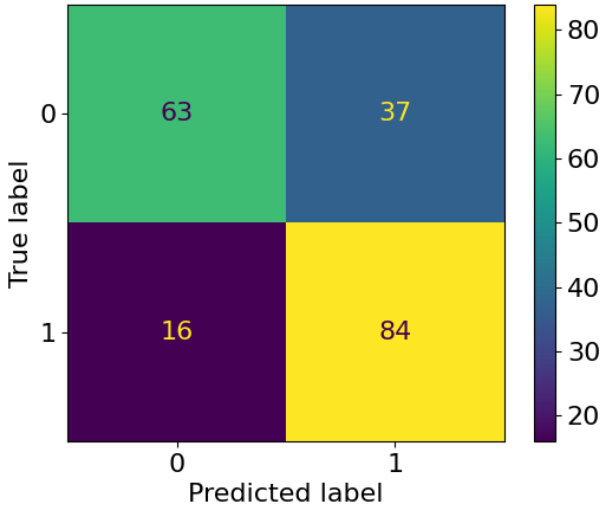


Figure 5.40: Confusion matrix (100%) for Random forest (Task 7; subject-based; PCA).

- SVM Linear

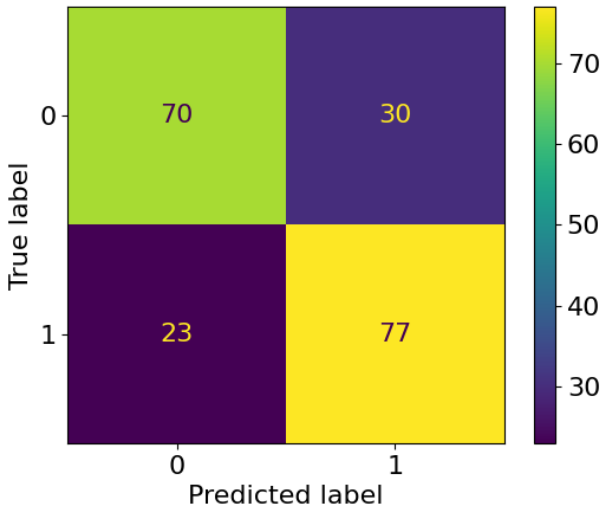


Figure 5.41: Confusion matrix (100%) for SVM Linear (Task 7; subject-based; PCA).

- SVM RBF

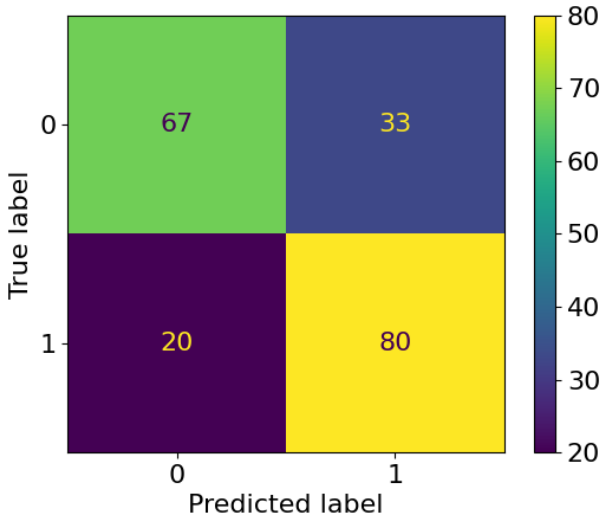


Figure 5.42: Confusion matrix (100%) for SVM RBF (Task 7; subject-based; PCA).

5.1.2 Two class classification (Dim. Reduction: Fisher's Score)

Task 5: Trial based

- KNN

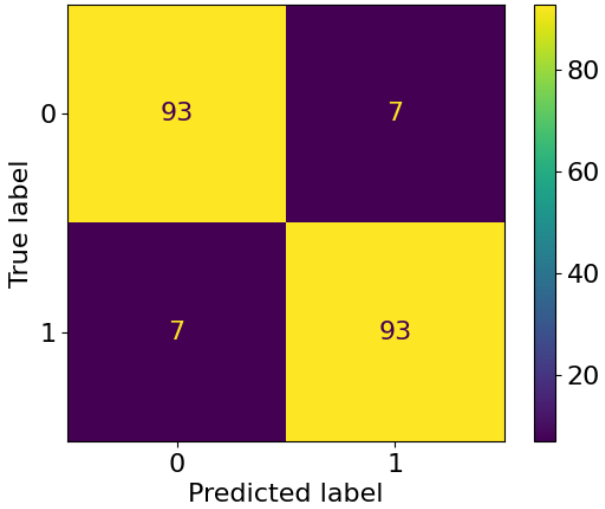


Figure 5.43: Confusion matrix (100%) for KNN (Task 5; trial-based; Fisher's score).

- LDA

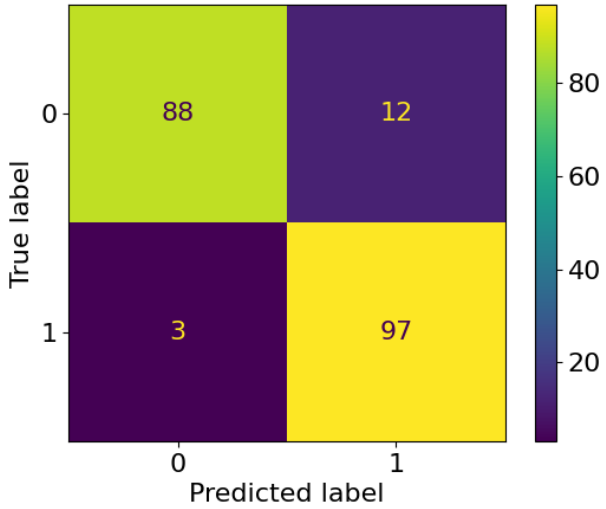


Figure 5.44: Confusion matrix (100%) for LDA (Task 5; trial-based; Fisher's score).

- QDA

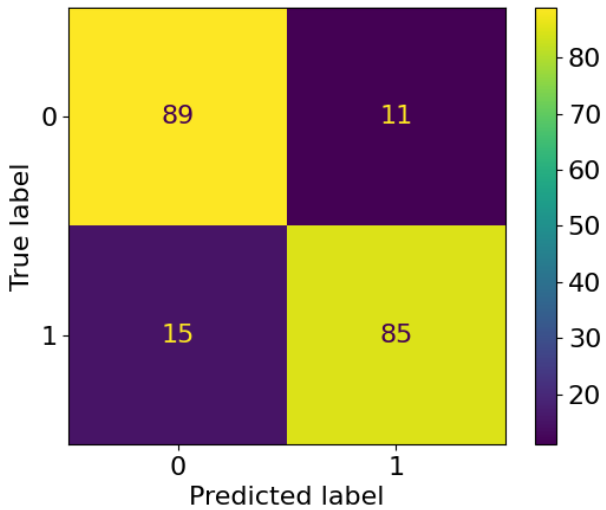


Figure 5.45: Confusion matrix (100%) for QDA (Task 5; trial-based; Fisher's score).

- Logistic regression

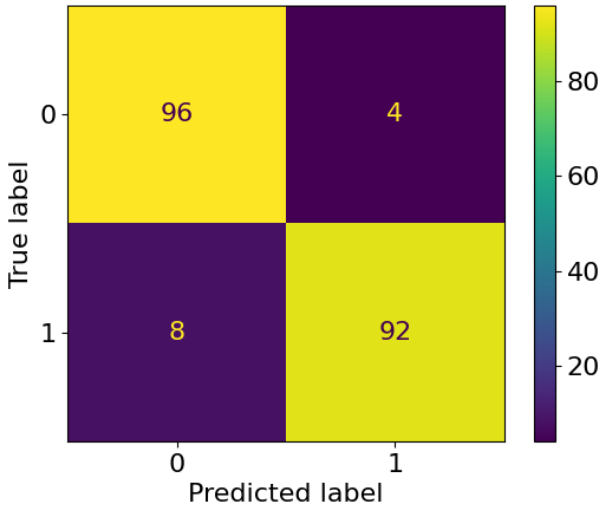


Figure 5.46: Confusion matrix (100%) for Logistic regression (Task 5; trial-based; Fisher's score).

- Random forest

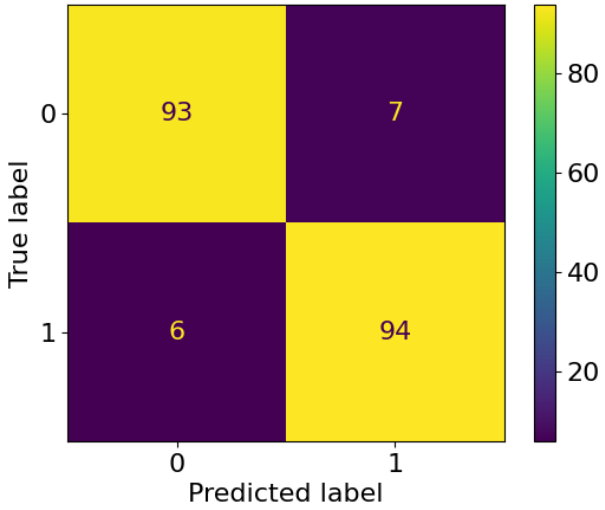


Figure 5.47: Confusion matrix (100%) for Random forest (Task 5; trial-based; Fisher's score).

- Naive Bayes

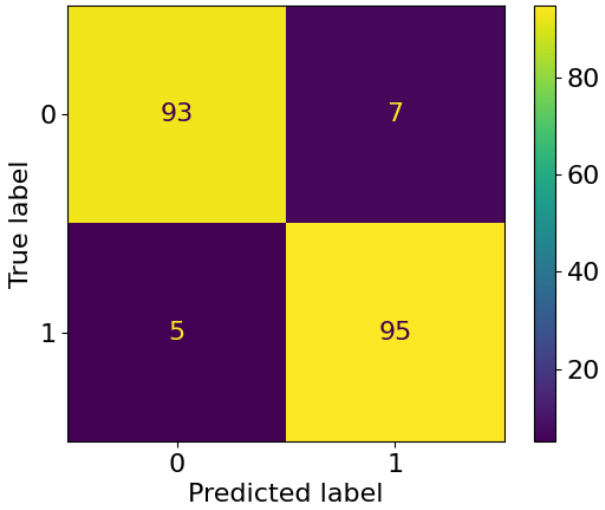


Figure 5.48: Confusion matrix (100%) for Naive Bayes (Task 5; trial-based; Fisher's score).

- SVM RBF

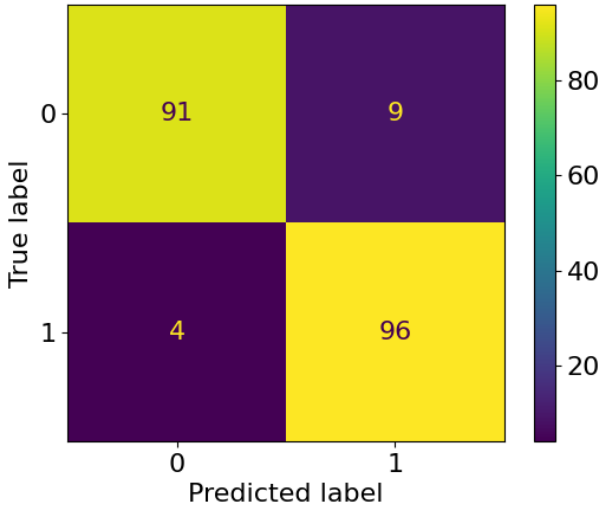


Figure 5.49: Confusion matrix (100%) for SVM RBF (Task 5; trial-based; Fisher's score).

Task 5: Subject based

- KNN

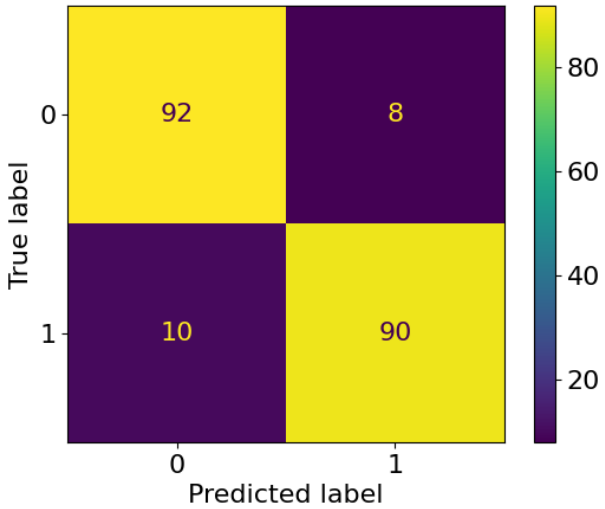


Figure 5.50: Confusion matrix (100%) for KNN (Task 5; subject-based; Fisher's score).

- LDA

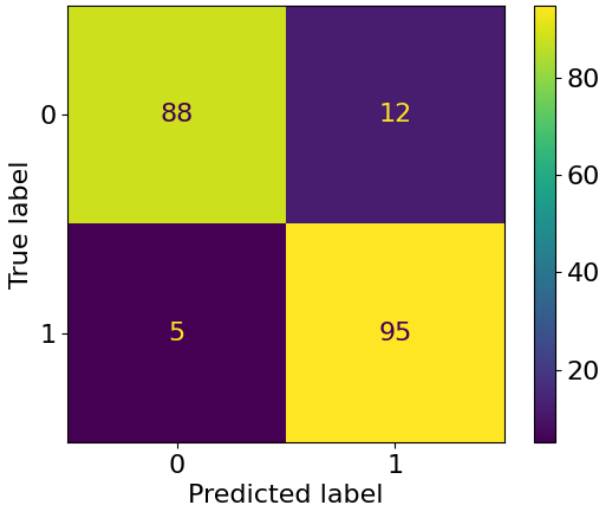


Figure 5.51: Confusion matrix (100%) for LDA (Task 5; subject-based; Fisher's score).

- QDA

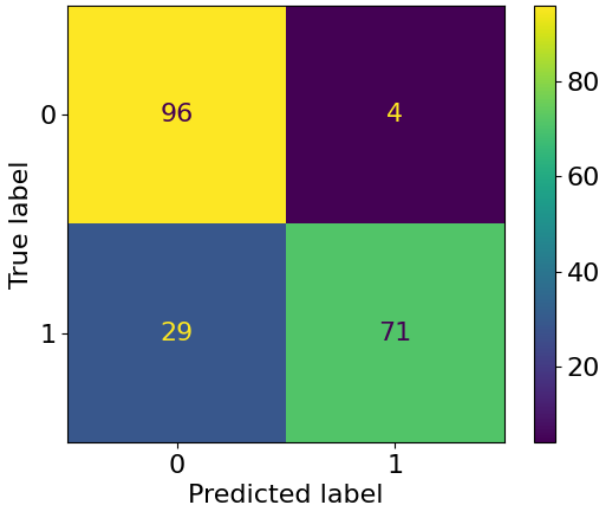


Figure 5.52: Confusion matrix (100%) for QDA (Task 5; subject-based; Fisher's score).

- Naive Bayes

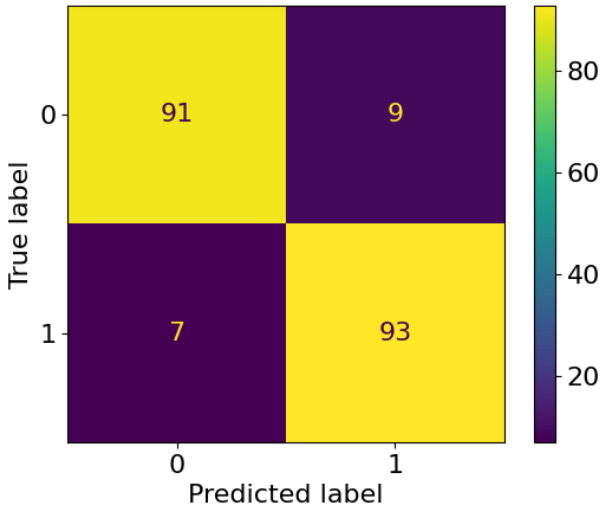


Figure 5.53: Confusion matrix (100%) for Naïve Bayes (Task 5; subject-based; Fisher's score).

- Random forest

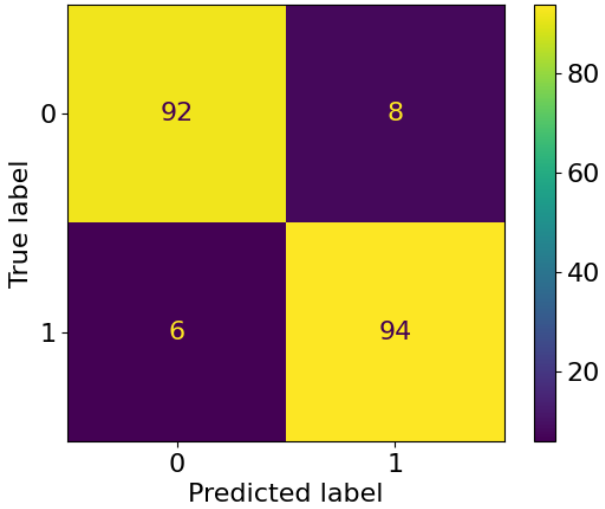


Figure 5.54: Confusion matrix (100%) for Random forest (Task 5; subject-based; Fisher's score).

- SVM Linear

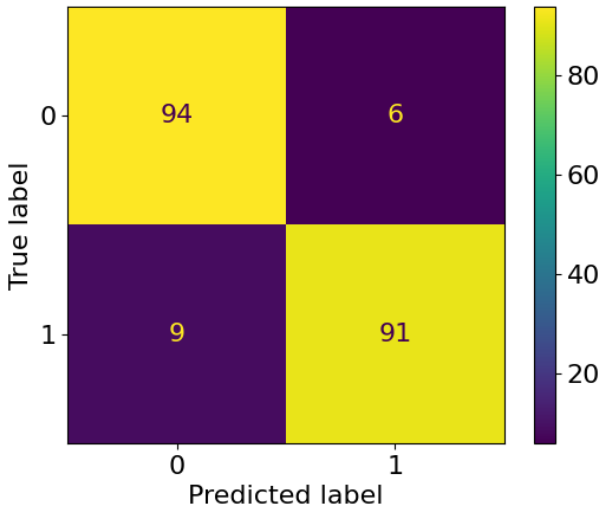


Figure 5.55: Confusion matrix (100%) for SVM Linear (Task 5; subject-based; Fisher's score).

- SVM RBF

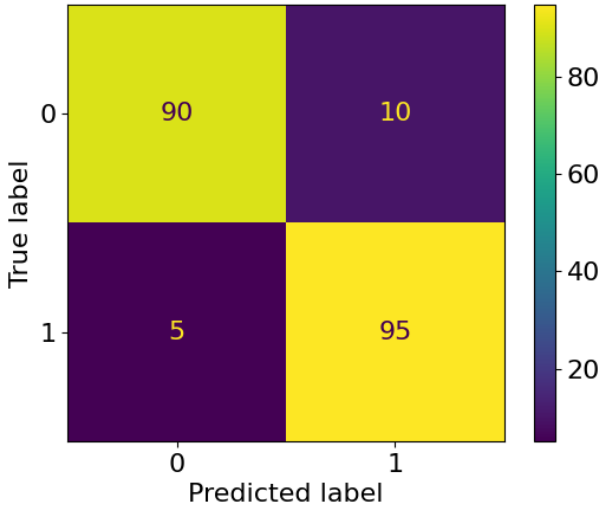


Figure 5.56: Confusion matrix (100%) for SVM RBF (Task 5; subject-based; Fisher's score).

Task 6: Trial based

- KNN

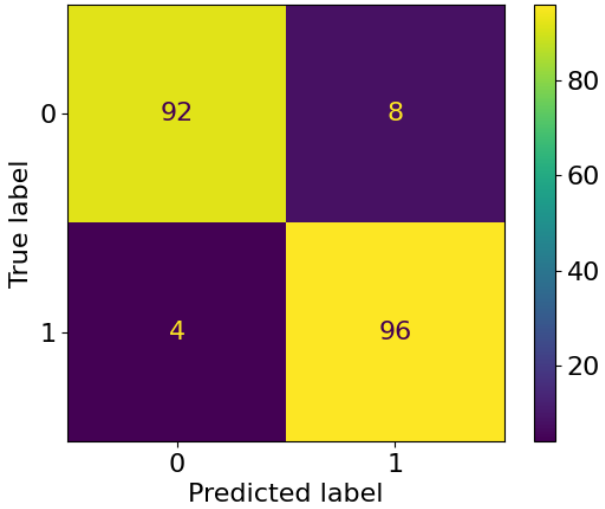


Figure 5.57: Confusion matrix (100%) for KNN (Task 6; trial-based; Fisher's score).

- LDA

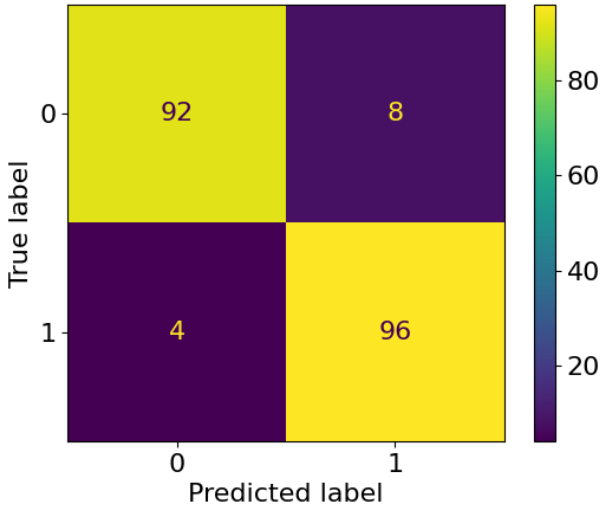


Figure 5.58: Confusion matrix (100%) for LDA (Task 6; trial-based; Fisher's score).

- QDA

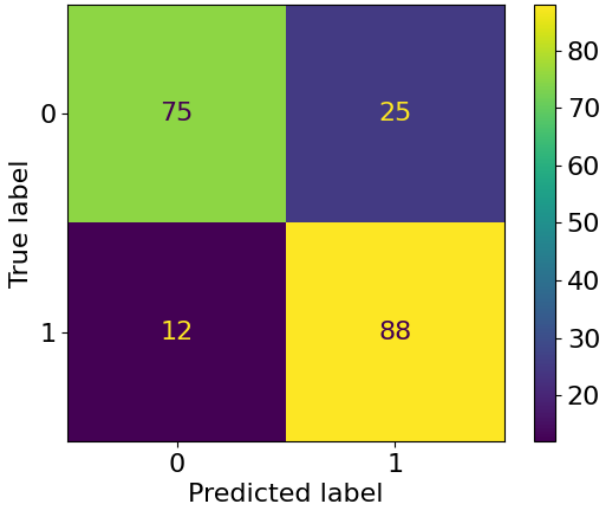


Figure 5.59: Confusion matrix (100%) for QDA (Task 6; trial-based; Fisher's score).

- Logistic regression

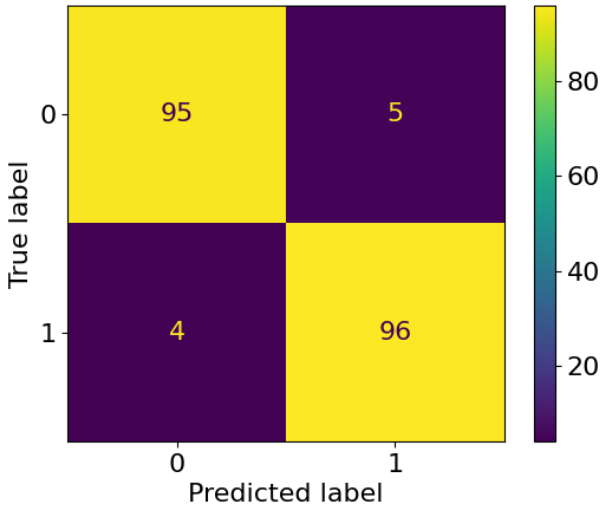


Figure 5.60: Confusion matrix (100%) for Logistic regression (Task 6; trial-based; Fisher's score).

- Naive Bayes

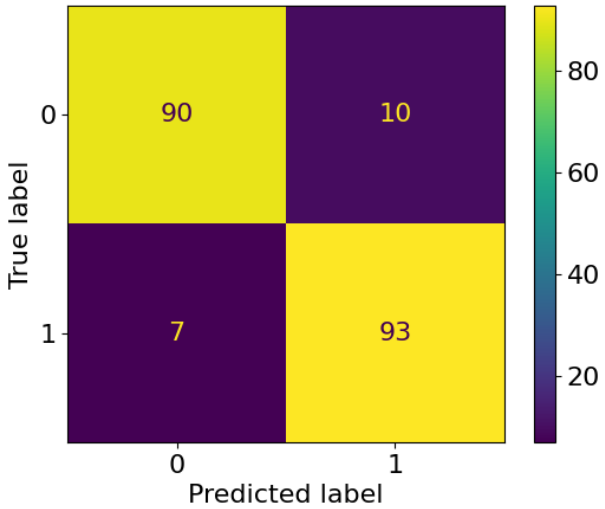


Figure 5.61: Confusion matrix (100%) for Naïve Bayes (Task 6; trial-based; Fisher's score).

- Random forest

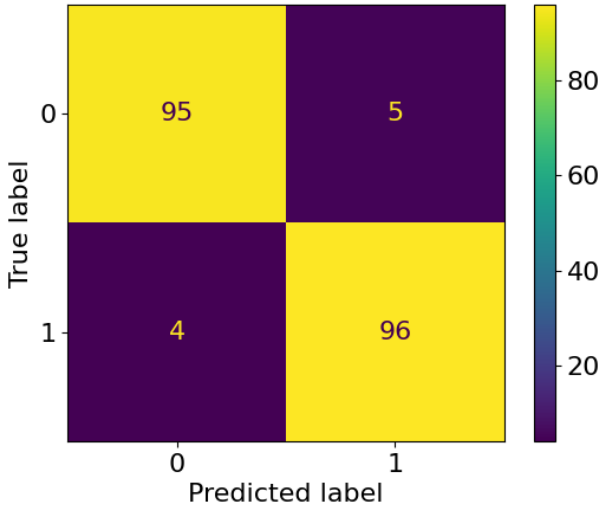


Figure 5.62: Confusion matrix (100%) for Random forest (Task 6; trial-based; Fisher's score).

- SVM RBF

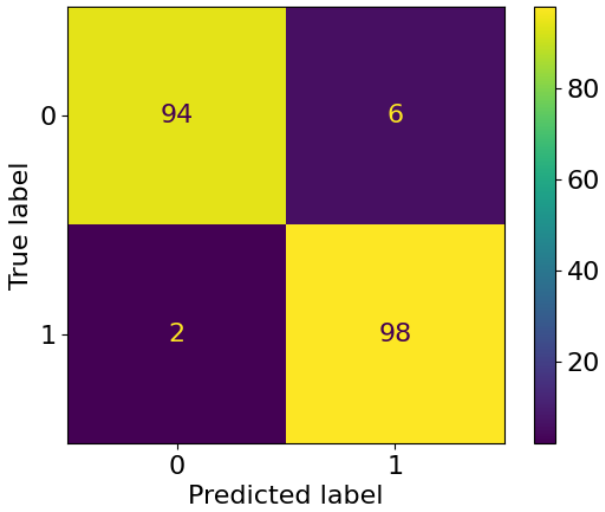


Figure 5.63: Confusion matrix (100%) for SVM RBF (Task 6; trial-based; Fisher's score).

Task 6: Subject based

- KNN

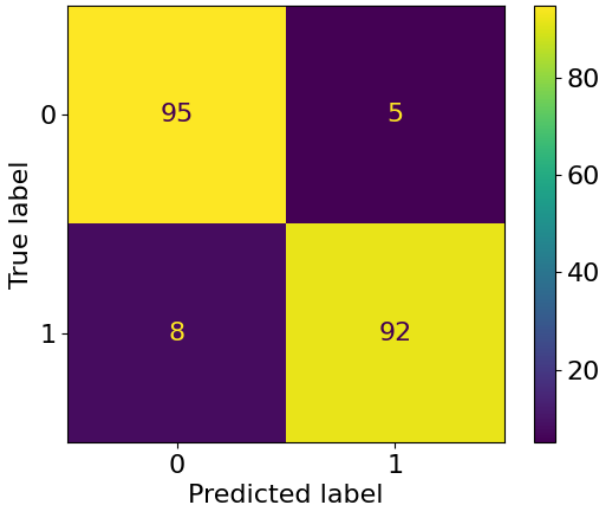


Figure 5.64: Confusion matrix (100%) for KNN (Task 6; subject-based; Fisher's score).

- LDA

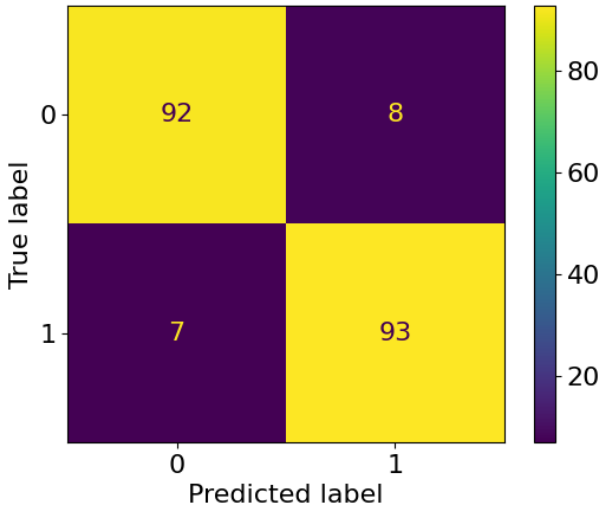


Figure 5.65: Confusion matrix (100%) for LDA (Task 6; subject-based; Fisher's score).

- QDA

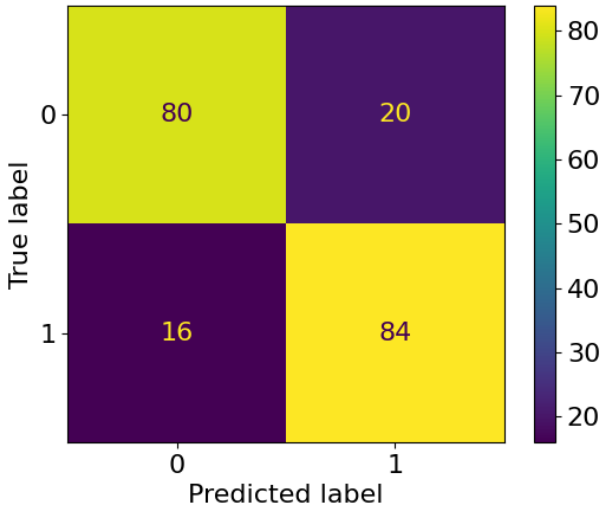


Figure 5.66: Confusion matrix (100%) for QDA (Task 6; subject-based; Fisher's score).

- Naive Bayes

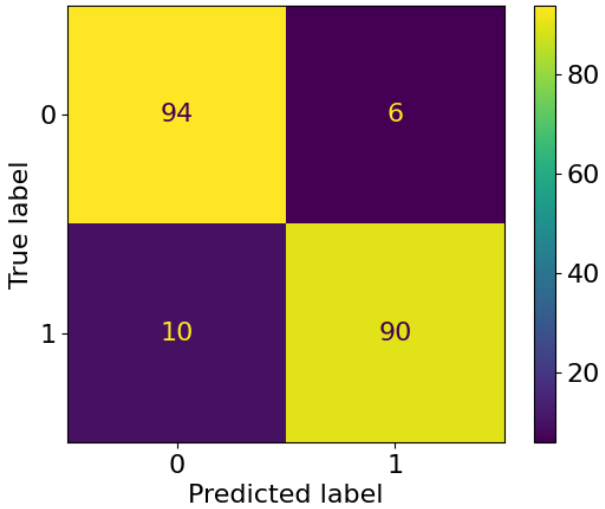


Figure 5.67: Confusion matrix (100%) for Naïve Bayes (Task 6; subject-based; Fisher's score).

- Random forest

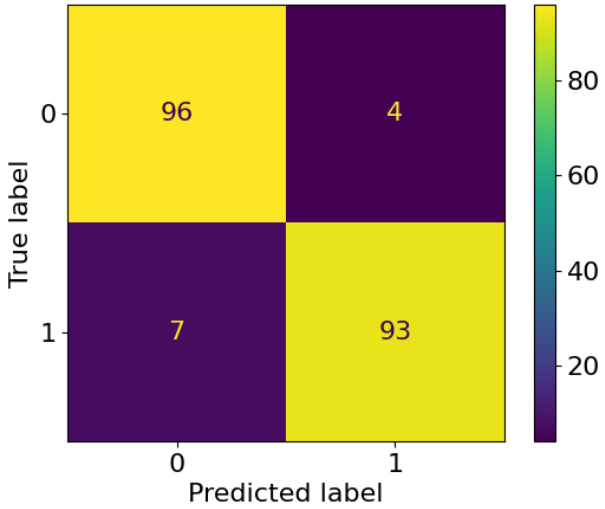


Figure 5.68: Confusion matrix (100%) for Random forest (Task 6; subject-based; Fisher's score).

- SVM Linear

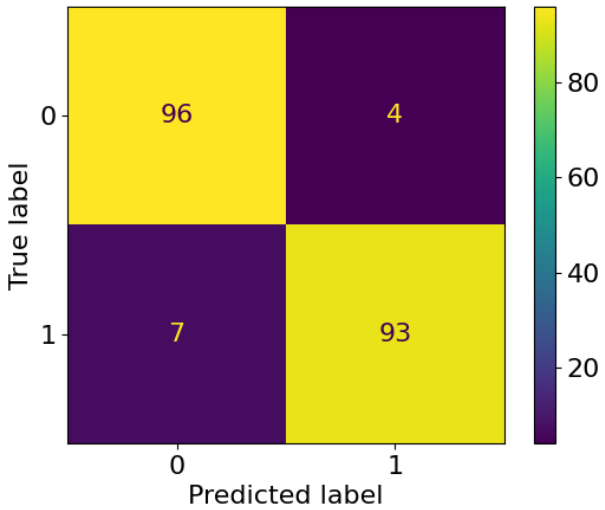


Figure 5.69: Confusion matrix (100%) for SVM Linear (Task 6; subject-based; Fisher's score).

- Logistic regression

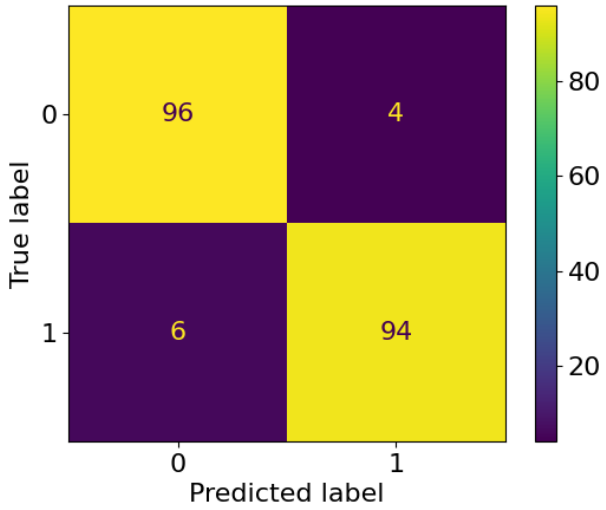


Figure 5.70: Confusion matrix (100%) for Logistic regression (Task 6; subject-based; Fisher's score).

Task 7: Trial based

- KNN

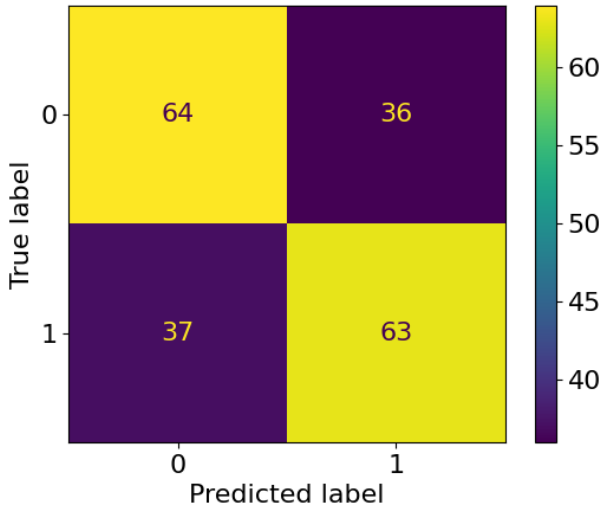


Figure 5.71: Confusion matrix (100%) for KNN (Task 7; trial-based; Fisher's score).

- LDA

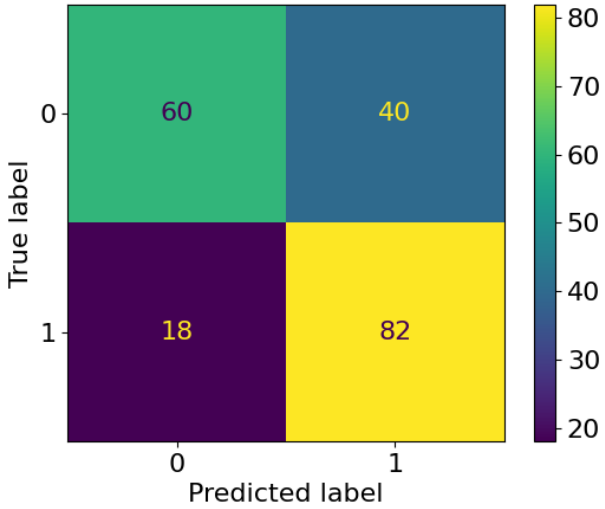


Figure 5.72: Confusion matrix (100%) for LDA (Task 7; trial-based; Fisher's score).

- QDA

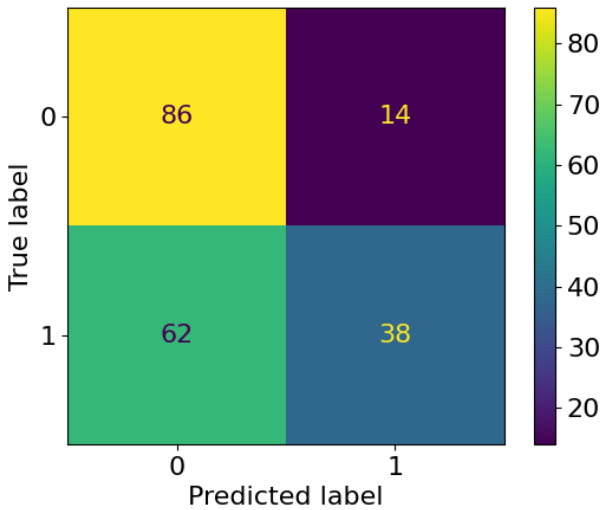


Figure 5.73: Confusion matrix (100%) for QDA (Task 7; trial-based; Fisher's score).

- Naive Bayes

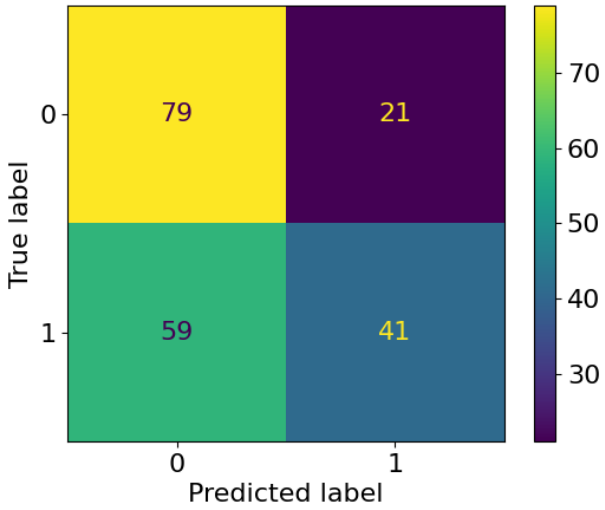


Figure 5.74: Confusion matrix (100%) for Naïve Bayes (Task 7; trial-based; Fisher's score).

- Random forest

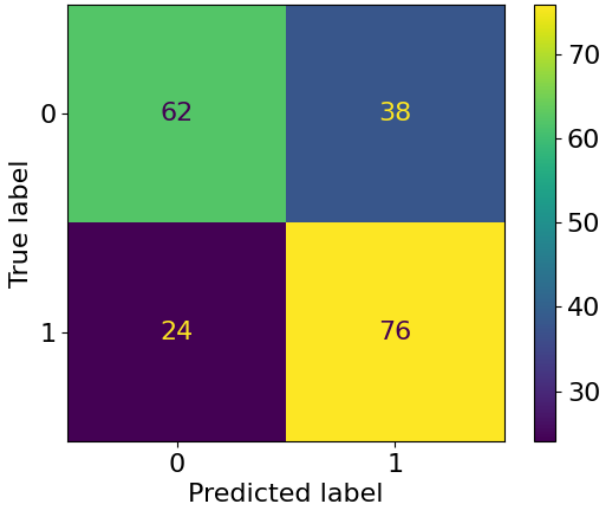


Figure 5.75: Confusion matrix (100%) for Random forest (Task 7; trial-based; Fisher's score).

- SVM Linear

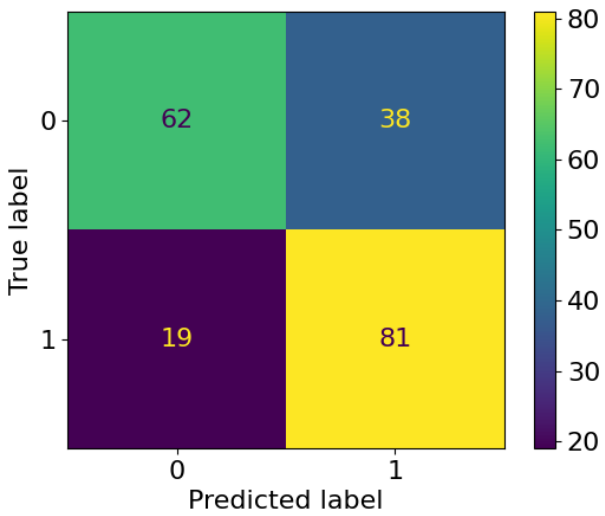


Figure 5.76: Confusion matrix (100%) for SVM Linear (Task 7; trial-based; Fisher's score).

- SVM RBF

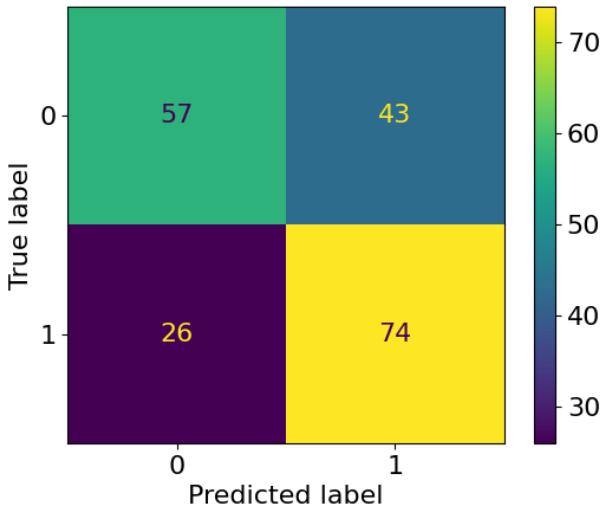


Figure 5.77: Confusion matrix (100%) for SVM RBF (Task 7; trial-based; Fisher's score).

Task 7: Subject based

- KNN

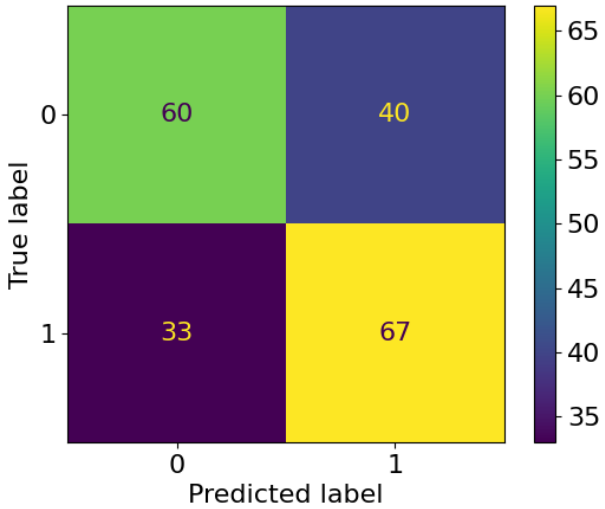


Figure 5.78: Confusion matrix (100%) for KNN (Task 7; subject-based; Fisher's score).

- LDA

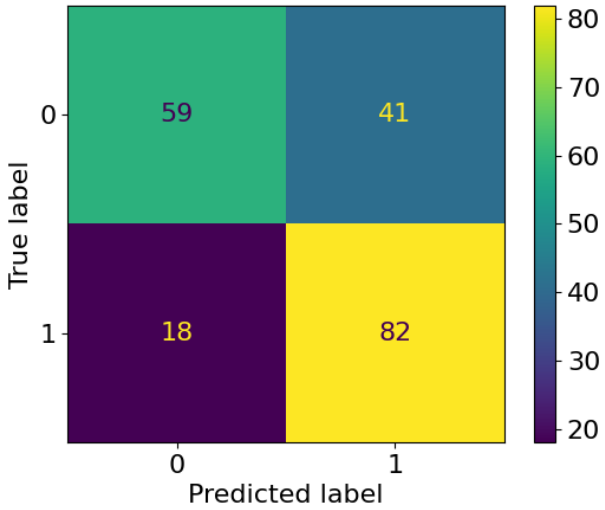


Figure 5.79: Confusion matrix (100%) for LDA (Task 7; subject-based; Fisher's score).

- QDA

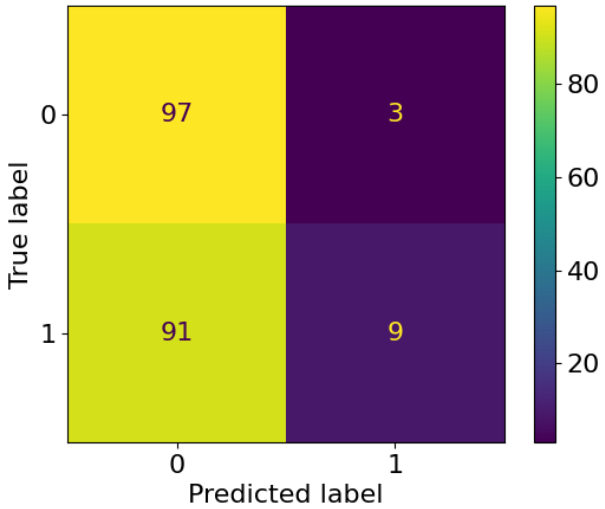


Figure 5.80: Confusion matrix (100%) for QDA (Task 7; subject-based; Fisher's score).

- Naive Bayes

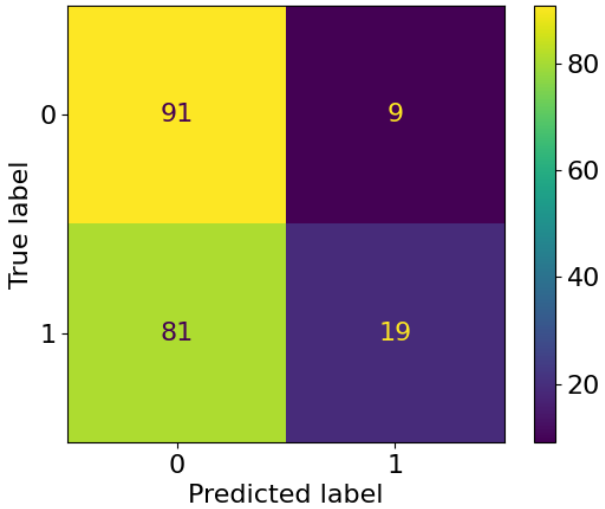


Figure 5.81: Confusion matrix (100%) for Naïve Bayes (Task 7; subject-based; Fisher's score).

- Random forest

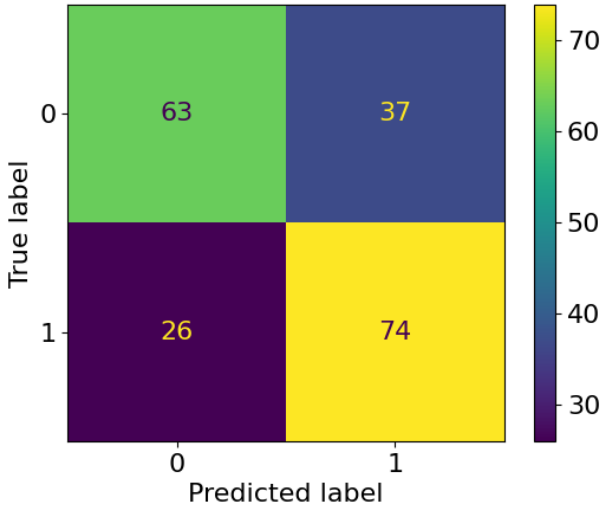


Figure 5.82: Confusion matrix (100%) for Random forest (Task 7; subject-based; Fisher's score).

- Logistic regression

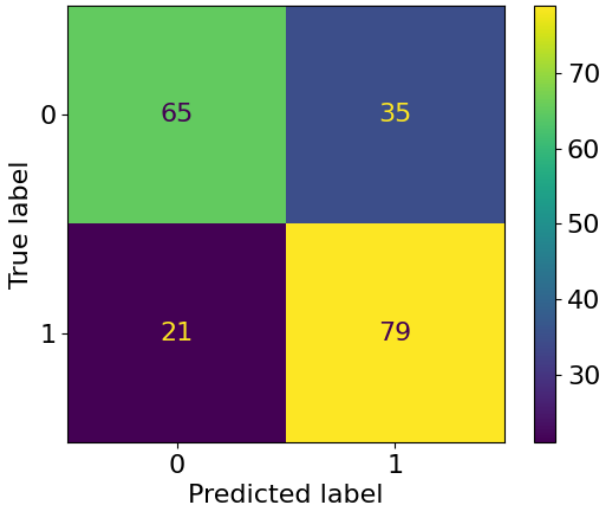


Figure 5.83: Confusion matrix (100%) for Logistic regression (Task 7; subject-based; Fisher's score).

- SVM RBF

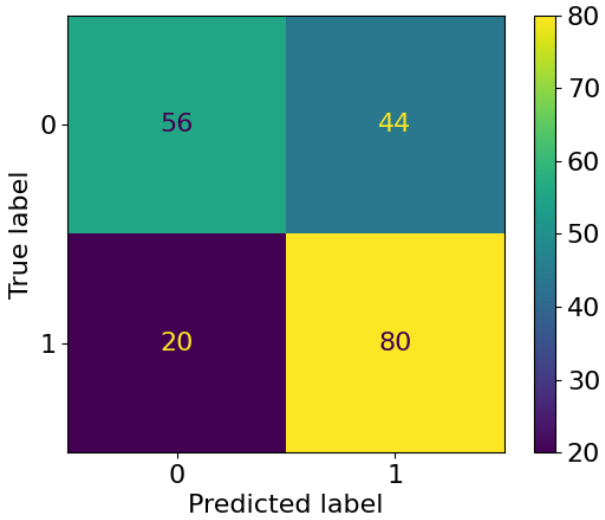


Figure 5.84: Confusion matrix (100%) for SVM RBF (Task 7; subject-based; Fisher's score).