Fall 12-16-2022

Rehabilitation Robotics and Machine Learning for Stroke Severity Classification

Raymond Greenfield

Rehabilitation Robotics and Machine Learning for Stroke Severity Classification

by

Raymond McCoy Greenfield

Under the Direction of Committee Chair's Igor Belykh, Ph.D.

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2022

ABSTRACT

Stroke therapy is essential to reduce impairments and improve motor movements by engaging autogenous neuroplasticity. This study uses supervised learning methods to address an autonomous classification via stroke severity labeled data by a clinician. Thirty-three patients with chronic stroke performed a variety of rehabilitation activities while utilizing the Motus Nova rehabilitation technology to capture upper and lower body motion. Based on the minimum, maximum, and mean of the range of motion and pressure as well as the number of movements, force flexion, and extension for each game and session provided from the sensor data. Supervised learning methods were applied to a harmonized dataset of roughly 32,000 patient sessions based on the maximum score per session per game. With this approach using light gradient boosting methods we achieved an average of 94% accuracy with 10-fold cross-validation to prevent overfitting. This thesis shows objectively-measured rehabilitation training, enabling the identification of the stroke severity class with the hopes to have patients have a less severe class in the future.

Over the last 10 years robotic rehabilitation has been utilized in inpatient therapy. Robotic rehabilitation has been shown to be effective in improving the severity of stroke in some cases. In particular, robotic devices can be used to help stroke survivors regain movement, improve their functional abilities and improve depression (11). These devices can provide a high level of precision and repeatability, allowing patients to perform therapeutic exercises with greater accuracy and consistency (1). Additionally, because robotic devices can be programmed to provide different levels of assistance, they can be tailored to the individual needs of each patient. This allows for a more personalized and effective rehabilitation in-home program (21).

INDEX WORDS: Machine Learning, Artificial Intelligence, Deep Neural Network, Gradient Boosting, Physical Therapy, Neuroplasticity

Rehabilitation Robotics and Machine Learning for Stroke Severity Classification

by

Raymond McCoy Greenfield

Committee Chair: Igor Belykh

Committee: Vladimir Bondarenko

Alexandra Smirnova Digitally signed by Alexandra Smirnova
DN: cn=Alexandra Smirnova, o, ou,
email=asmirnova@gsu.edu, c=US
Date: 2022.12.17 21:19:02 -05'00'

Alexandra Smirnova

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

December 2022

DEDICATION

This thesis is dedicated to the stroke victims of the world in an effort to have a higher quality of life.

ACKNOWLEDGMENTS

Above all, I offer my genuine gratitude to my major advisor Professor Igor Belykh and Dr. Rusell Jeter, both of who inspired me with their support and guidance. Without their advice and help, I could not have developed the knowledge and accomplished this thesis. Dr. Jeter has not only guided me into the statistical learning field but also taught me the way to learn new things which will allow me to continue my work and embrace new knowledge. I am also truly grateful to the members of my thesis committee: Professors Igor Belykh, Alexandra Smirnova, and Vladimir Bondarenko for their time and kind assistance during my thesis preparation.

I would also like to thank Dr. Xiaojing Ye and Dr. Sergey Plis for their courses on optimization and deep learning respectively. I genuinely learned a lot from them in their courses and were honestly some of my favorites. Both pushed me to not only understand the rigorous mathematics that backs the frameworks that are used in modern machine-learning methods. My heartiest appreciation also goes to all the faculties and staff of the department of Mathematics Statistics who contributed in diverse ways to make my experience a token of memory at Georgia State University.

Lastly, I would like to express my thanks to my mother Julie Greenfield RPH for her many manifold blessings and support to me. Without them, I would not have been able to complete the pursuit of my thesis studies.

This work was supported by the NSF under Grant CMMI-1953135.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Background

Stroke is a leading cause of mortality and disability worldwide and the economic costs of treatment and post-stroke care are substantial (5). In 2019, there were 12·2 million incident cases of stroke, 101 million prevalent cases of stroke, 143 million (DALYs) due to stroke, and 6.55 million deaths from stroke (6). The severity of a stroke can range from mild to severe, with severe strokes often leading to long-term disability or even death. Stroke rehabilitation typically involves a team of healthcare professionals, including doctors, nurses, therapists, and other specialists. The specific goals and interventions of stroke rehabilitation will vary depending on the individual's needs and abilities, but may include physical therapy to improve mobility, occupational therapy to improve the ability to perform daily activities, speech therapy to improve communication skills, and cognitive therapy to improve memory, problem-solving, and other cognitive abilities. Stroke rehabilitation typically involves a team of healthcare professionals, including doctors, nurses, therapists, and other specialists. The specific goals and interventions of stroke rehabilitation will vary depending on the individual's needs and abilities but may include physical therapy to improve mobility, occupational therapy to improve the ability to perform daily activities, speech therapy to improve communication skills, and cognitive therapy to improve memory, problem-solving, and other cognitive abilities. Our goal is to transfer this now not only to an in-home setting for the patient but use machine learning to quantify the progress of patient improvement.

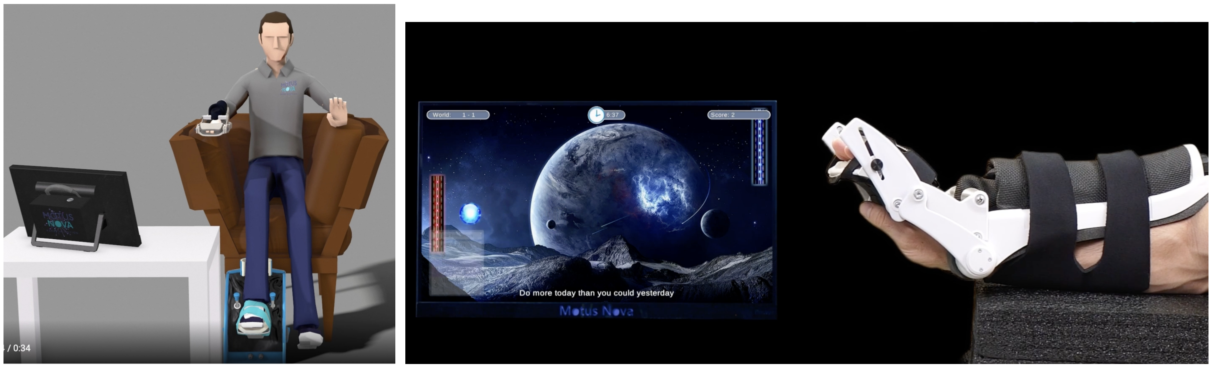

Figure 1.1: Patients do therapy sessions with the Motus Hand or Motus Foot using a pneumatically-driven exogenous robotic device that is worn on the affected hand/arm (depicted in the bottom panel). The peripheral acts as a game controller (through an angle sensor embedded in the wrist joint) that allows users to play therapeutic video games that are able to dynamically adapt to their needs and provide them the requisite assistance/resistance (computer screen in the bottom panel).

Supervised learning is a type of machine learning in which a model is trained on a labeled dataset, where the correct output (or label) is provided for each input. The goal of supervised learning is to build a model that can make accurate predictions on unseen data.

To use supervised learning methods to classify stroke severity, you would need to collect a dataset of stroke patients, where each patient is labeled with their stroke severity (e.g. mild, moderate, severe). This dataset would then be used to train a machine learning model.

There are many different machine learning algorithms that could be used for this task, including decision trees, support vector machines, and neural networks. The choice of algorithm would depend on the specific characteristics of the dataset, such as the number of samples, the number of features, and the distribution of the classes.

Once the model is trained, it can be used to make predictions on new, unseen data. For example, given information about a patient's symptoms and medical history, the model could predict the severity of their stroke. These predictions could then be used to guide treatment and prognosis.

The Motus Hand and Foot can be used without needing help from a clinician or caregiver. Motus technology uses high-dose repetitive task practice to induce neuroplasticity to help stroke survivors improve neuroplasticity in the upper and lower peripherals (13).

Patients take part in video games during a therapy session and would receive a score based on their performance in the game. Using artificial intelligence the Motus Nova robotic Hand or Foot applies pressure to the hand or foot to increase the range of motion during the session (22).

The recent growing interest is due to the increasing complexity and numerosity of available data sets, for which more classical methods do not allow accurate results as shown in this study (3),(18). Here, we provide a unique dataset collected from patient therapy sessions. Machine Learning is the methodology that provides computers with the ability to learn from experience i.e. training a model on a split dataset (3).

Utilizing ML and AI may have a central role in rehabilitation decision-making in determining if patient therapy is improving. ML is the methodology that provides computers with the ability to learn from experience. By constructing and training supervised classifiers to learn decision rules from data, automatic solutions able to make predictions on new data can be exploited.

Like in many healthcare, disease, or machine learning research applied in a clinical setting, labeling of patient data by a clinician is necessary (16). This study applies the same heuristic methodologies.

Figure 1.2 provides a high-level overview of the data collection, analysis, processing, and modeling that produce the final classification results.

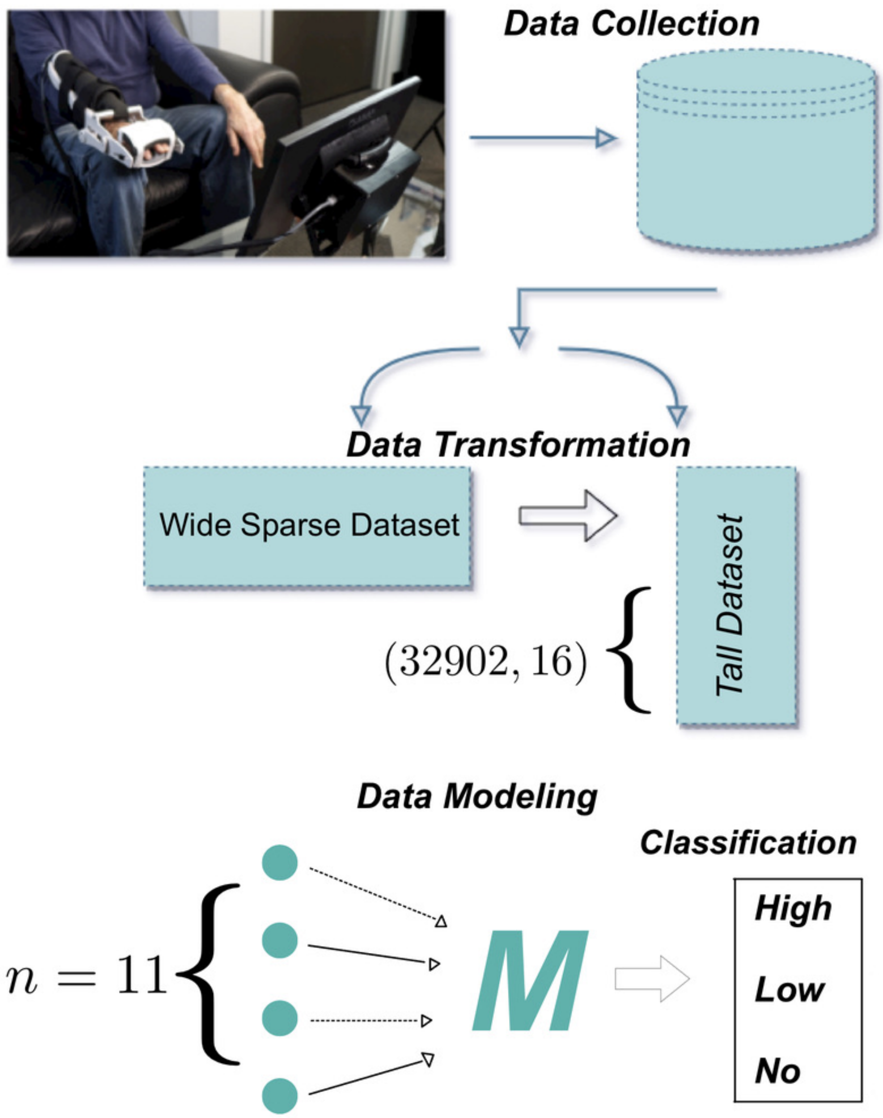

Figure 1.2: Concept diagram of the overall data science process. Data is gathered from therapy sessions performed using the Motus Hand and Motus Foot. The data is processed and used in a supervised machine learning model to classify the stroke severity of the patient. A total of 11 predictor variables attribute to the classification of stroke severity.

1.2 Methods and Procedure

The Motus Hand and the Motus Foot each consist of two major components: a peripheral that the patient attaches to their affected limb, and an interactive console that guides their therapy routine and assessment using a video game interface. The peripherals are equipped with a pneumatic actuator that is able to dynamically provide assistance/resistance by filling an air muscle located in the peripheral that then moves the wrist/ankle joint of the peripheral. The wrist/ankle joint of the peripheral has an embedded angle and pressure sensor that transmits live angle data to the console. This allows the console to give the user immediate visual feedback of their movement through avatars in a video game on the screen. The therapeutic video game activities are able to provide a dynamic feedback loop consisting of in-game goals (ships to shoot, for example) that drive user movements, which correspond to movement on-screen which allow the console to react and set new goals/obstacles. This feedback loop is designed to promote sensory-motor function.

A therapy session with the Motus Hand or Foot consists of a variety of stretching, gross motor control, fine motor control, and endurance exercises depending on the patient's needs. This process is depicted in Figure 1.1 where a Motus Hand user is playing ”Cosmic Tennis,” which is a gross motor control exercise that plays like the classic arcade game Pong 1.1. The user's wrist/ankle movement corresponds to the movement of the paddle on the right-hand side of the screen, and the goal is to hit the ball back and forth to score on the AI-controlled opponent.

In order to use the data collected during a therapy session to autonomously classify a patient's stroke severity, each patient was given a guided assessment with a clinician using the Motus Hand or Foot to classify them as having High Range of Motion (ROM), Low ROM, or No ROM.

Using four machine learning algorithms: light gradient boosting, extra trees classifier, deep neural network (DNN), and logistic regression. A practical model using the most common data measured in each session data based on the maximum score per session per patient. Unsupervised learning methods were applied to the final dataset such as the correlation matrix and principal component analysis to show that all variables collected are relevant to the study. This includes a 10-fold cross-validation on the final dataset of size 16 columns and 32,902 rows (sessions) with the mean and standard deviation of accuracy from each computational experiment. From here the following metrics are used to determine the performance of the model including the accuracy, precision, and recall from the confusion matrix. The macro average f1-score was accounted for this multi-classification problem (4).

In summary, session data is collected, processed, and analyzed sequentially. Based on the performance measures mentioned before the light gradient boosting model best fit this dataset with more than a 50% improvement compared to other classical methods such as logistic regression.

ID	Max	Min	Assessment	Classification
2054	30	3	Passive	No
1495	37	-20	Passive	No
2058	50	-24	Passive	No
2273	21	-16	Passive	No
2085	40	-15	Passive	No
2098	44	-9	Passive	No
1864	28	-12	Passive	No
1859	45	-17	Passive	No
1479	45	-20	Passive	No
1865	30	-15	Passive	No
2128	34	-10	Passive	No
1838	33	-15	Passive	No
2183	41	-16	Passive	No
2040	37	-18	Passive	No
2097	43	-18	Passive	Low
2356	-3	-17	Assisted	Low
2356	-3	-17	Assisted	Low
1688	52	-23	Assisted	Low
1876	54	-12	Passive	Low
2029	46	-20	Passive	Low
1458	30	-18	Passive	Low
1113	33	-20	Assisted	High
2262	38	-13	Assisted	High
1637	10	-12	Assisted	High
2282	8	-16	Assisted	High
1781	39	-15	Assisted	High
2360	10	-18	Assisted	High
2035	41	7	Assisted	High
1799	48	1	Assisted	High
2191	40	-20	Assisted	High
1974	38	-6	Assisted	High
2004	41	-20	Assisted	High
2179	49	12	Assisted	High
1470	20	-16	Assisted	High

Table 1.1: Example patient label table assessed by a clinician using a potentiometer (12). Note that the final label is at the discretion of the clinician and could be based on qualitative factors not accounted for in the test.

CHAPTER 2
Data Collection and Harmonization

2.1 Data Collection

Throughout a therapy session using the Motus Hand or Motus Foot, live angle data (measured in degrees from a natural midpoint in wrist/ankle placement) is collected from the sensor embedded in the wrist or ankle joint at 50 readings per second. This "raw" angle sensor data is then stored in a high resolution time series database (InfluxDB (15)). In addition to the high resolution angle data, pressure readings (measured in PSI) are taken from the pressure management system at 50 measurements per second.

Each therapy session a patient does includes a selection of ≈ 30 activities that focus on different types of motor function including: gross motor control, fine motor control, flexor tone reduction, endurance, reaction time, and tracking.

A patient can participate in more than one video game during a patient session. Once the patient is finished with the video game the score is recorded and stored.

Gender and other biometric data such as age, height and weight are not included in the patient description or the analysis.

2.2 Clinician Labeling

In order to have a supervised learning method for the machine learning algorithm, there has to be human labeling. Quantitative and qualitative factors at the discretion of the clinician go into the labeling of stroke patients. By using the potentiometer (12), clinicians are able to gather data on the patient in Table 1.1. We define a patient as Assisted if the clinician themselves applies pressure to the patient to achieve the minimum and maximum angle. Similarly, patients are Passive meaning that the clinician does not help them achieve this maximum or minimum angle in the labeling process. The low label is mixed between the assisted and passive. Notice that all patients that are classified with a high stroke severity were assisted by the clinician. This is important when noticing that patients 2085 and 1781 (Blue) have a similar total range of motion (Min + Max), but patient 1781 is assisted by the clinician. However, there is ambiguity in some labels. For example, take patient 2356 (red) where it can be argued that the patient should have a high stroke severity given the low total range of motion with the assistance. This is where the clinician has other outside factors that contribute to the final labeled classification of a patient.

2.2.1 Clinician Label Mapping

In efforts to quantify the labels that the clinician provided to the patients. A mapping based on the minimum and maximum angle, the range of motion assessment (Active or Passive), and the peripheral type were used as variables in the analysis. The assessment is done by using a potentiometer (12) This is simply to check if there is consistency in the clinician labeling and for later purposes, as more and more patient labels are collected it can possible to classify more patients without necessarily having the clinician label. In this case, there are 50 patients. The 33 patients are users with the score data. Unfortunately, a strong classification was not achieved with a performance of less than 80% as shown in Figure 2.1. In the future, more patient data is gathered this could become more prominent.

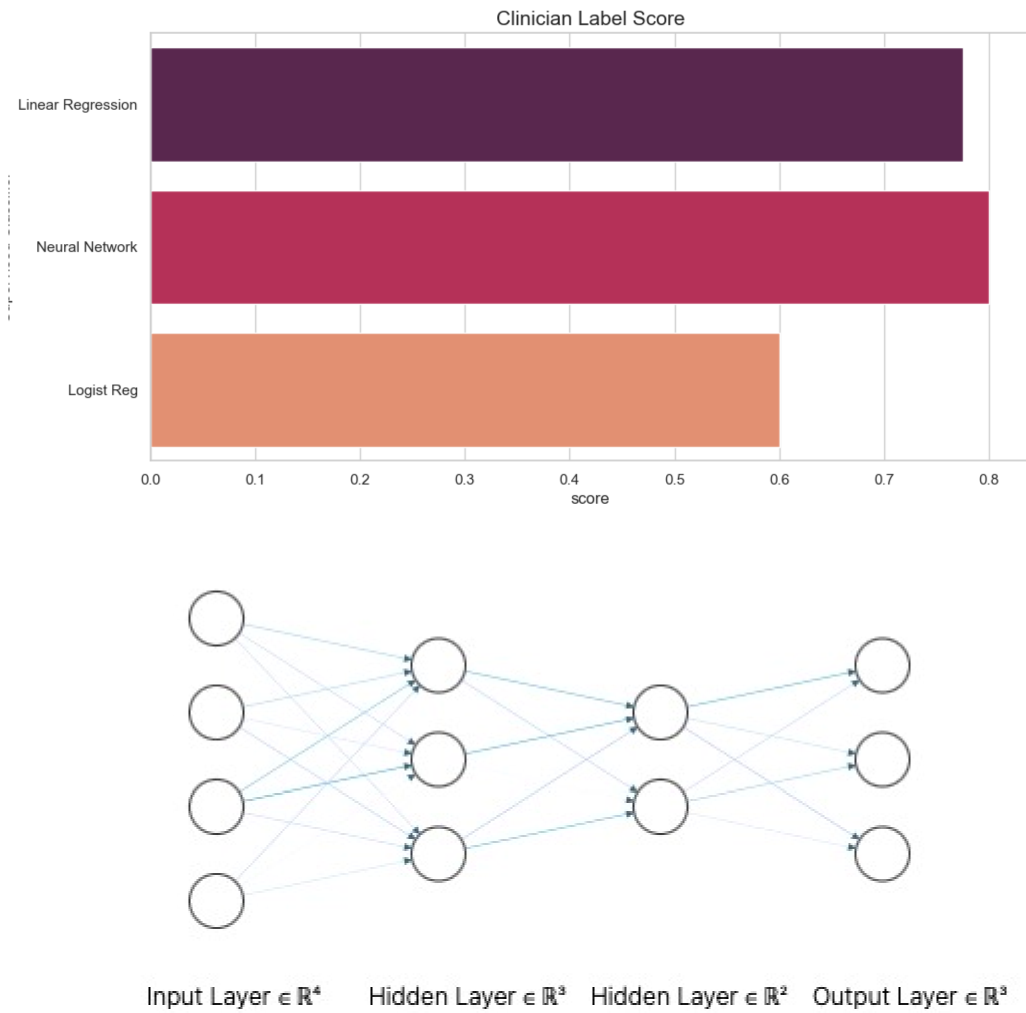

Figure 2.1: Mapping of Clinician Label based on Min, Max, Peripheral Type, and Assessment Variables

Deep learning can be more of a blackbox approach, especially with only 50 data-points. The following is an example of the architecture using in testing for clinician label mapping.

2.3 Data Harmonization

Each game had an original column and associate min, max, and mean range of motion and pressure causing the dataset to be rather wide (72 columns) and sparse since each patient cannot necessarily play every game in every rehab session. In order to combat this, we must perform a data transformation. For example, we have the following six variables for Plinko;

$$\text{Plinko\_rom\_min} \rightarrow game\_rom\_min,$$

$$\text{Plinko\_rom\_max} \rightarrow game\_rom\_max,$$

$$\text{Plinko\_rom\_mean} \rightarrow game\_rom\_mean,$$

$$\text{Plinko\_pressure\_min} \rightarrow game\_pressure\_min,$$

$$\text{Plinko\_pressure\_max} \rightarrow game\_pressure\_max,$$

$$\text{Plinko\_pressure\_mean} \rightarrow game\_pressure\_mean$$

This is then mapped for all thirty games with the associated game_id column with a key value of 1-30. Each game variable was mapped to a row and the dataset then had a general range of motion and pressure variable corresponding to each patient, session, and game. Hence, we go from having a sparse wide dataset to a tall dataset with missing values, but not a sparse matrix.

There were five main cases of missing data or data misrepresentation. The first three cases included the R_{min}, R_{max}, and the t_{game} (game time) variables with missing values. Since we could not necessarily deal with taking averages on the patient level we considered the mean of the column and filled the values accordingly. the game (game time) variable we took the minimum value. Another assumption was the N_{mov} variable (number of movements in a session) where we placed a floor at the minimum number of movements at 3. A movement is considered a full flexion or extension of the hand or foot in a patient session. This is mainly under the assumption that the patient did not actually commit to a therapy session without a certain number of movements. In the last case for some patients, the sensor read a negative value for the minimum pressure value. This, of course, is not physically possible and in these cases, the rows with negative pressure were clipped and set to zero meaning no pressure was applied by the robotic hand or foot during the rehab session.

Finally, aggregating the dataset based on the maximum score per game per session generated a dataset of size 32,902 rows with 16 columns.

For the final dataset the maximum score for each therapy session per game for each patient. In this dataset, we have the following description of the columns: the game patient, and session identification with the start time associated with these. The minimum, maximum, and mean pressure (PSI) and range of motion along with the amount of time the game is played and the score of the game. Each of these patients has an associated peripheral and classification. In the summary statistics of the final dataset, there were a few missing points that needed to be addressed. This includes the matter that there were negative pressure readings from the sensor data. Since this is not actually possible the data was floored and assumed that the negative pressures were reading at zero pressure. The range of motion measurement has negative values. This is from the sensor when gathering data where the range of the angle is -25° to 55°.

CHAPTER 3

Exploratory Data Analysis

The next step for any data science project after collection and harmonization is to explore the data and gather insights. One of the properties to distinguish the distribution of the data is shown in Figure 3.1. These variables have to be normally distributed in order to use a machine learning algorithm because of scaling (10).

Figure 3.1: The minimum pressure variable is not displayed since the majority of the values are zero. This can be seen in the average pressure and maximum pressure variables.

It is noted that the distribution of a random variable needs to be normally distributed to follow the central limit theorem in order to use a machine learning algorithm. Though one can argue the lack of normal distribution of the force flexion and extension variable. Later it is shown to have little effect on the final results of the classification analysis (see chapter 5).

The following Figure 3.2 shows a count of the peripheral types and the number of patients in each class (High, Low, No). As seen many of the patients used the Motus Hand and only 3 used the Foot. In future studies, there will be more usage of the foot. Also, note that there are more patients in the No ROM class. This is at the discretion of the clinician during a patient test with a potentiometer. The patient with the No ROM class all received no assistance during the test, while on the other hand patients with the High ROM class received assistance from the clinician. The bar charts are from table 1.1.

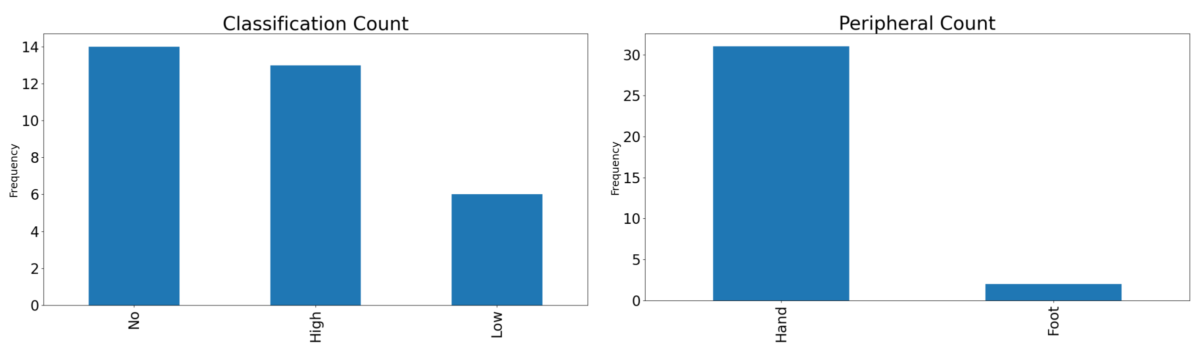

Figure 3.2: Value counts of the peripheral types and classes.

CHAPTER 4

Methodologies

Unsupervised Learning Methods: Correlation Matrix and Principal Component Analysis

One of the goals of this study is to determine if there exists a key indication variable that is a direct indicator of stroke classification. Here we use this unsupervised learning allows us to see if there is a need for a dimensionality reduction in the analysis (8).

The correlation matrix in Figure 4.1 is an indication of dimensionality reduction inherently. We disregard key variables such as Patient ID, Game ID, and Start Time since those are indications of the start of a patient session. If two variables are highly correlated i.e. $Cor(X, Y) > 0.9$, then this is a good indication that one of those variables can be dropped in our analysis (19). In Figure 4.1, we can consider the lower or upper half of the matrix. There exists a strong negative correlation between F_{ext} and F_{flex}, but intuitively it does not seem useful to drop either one of the variables in the analysis individually. The correlation between the game pressure mean and max with the value of 0.80, is an indication that more pressure is being applied by the Motus Hand or Foot on average in each session. Since this correlation does not surpass the threshold of 0.90 we do not drop either variable. In consideration of the case where the t_{game} (game time) variable has a correlation of 0.60, this is an indication that the longer a patient plays a game i.e. the longer the session time this should increase the score received.

Consider Figure 4.2, here we graph each principal component and how much variation that eigenvector contributes to the total variation. Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. Explained variance is a statistical measure of how much variation in a dataset can be attributed to each of the principal components (eigenvectors) generated by the Principal Component Analysis (PCA) method. It tells us how much of the total variance is "explained" by each component. This allows us to rank the components in order of importance, and to focus on the most important ones when interpreting the results of our analysis (9).

From here we can interpret the following, that all 11 variables are needed for the analysis. Hence, we cannot fit a better model (see final results) that has a high accuracy and performance measure.

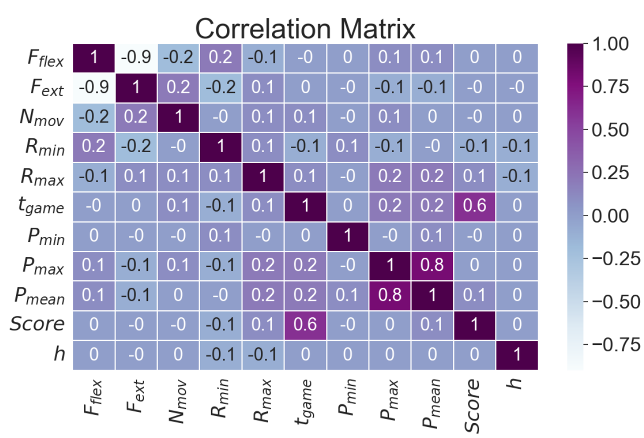

Figure 4.1: The correlation matrix does not include key variables such as Patient ID, Session ID, Game ID, and Start Time. Importance in dimensionality reduction based on a greater than 0.9 threshold. As seen above, F_{flex} and F_{ext} are highly negatively correlated. However, these variables were both used in the analysis.

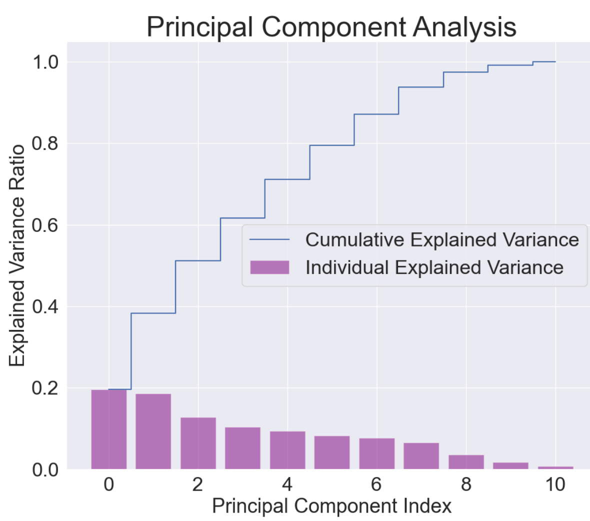

Figure 4.2: Depiction of the Principal Components with the Explained Variance Ratio. As shown 95% of the explained variance is contributed by all principal components (8). As a result, all variables are used in the machine learning model for the analysis.

CHAPTER 5

Computational Experiments

5.1 Model Description

Gradient boosting decision tree (GBDT) is a widely-used machine learning algorithm, due to its efficiency, accuracy, and interpretability (7). In essence, the algorithm uses smaller "weaker classifiers" with a number of leaves. Taking a weighted average of these several "weaker classifiers" we are able to construct a "stronger classifier". This is by the weak learners theorem (17). This will be the key model used in our analysis. From the lightgbm module in python version 3.9.10 we used the LGBMClassifier as well as for the other models. The next four sections will be on the model descriptions used in the python code.

For example, the n-estimators parameter is the number of "weaker" models used in the construction of the model. The construction of the model came from Microsoft GitHub with auto machine learning libraries (14). Hence, the parameters chosen above came from the algorithm in the library FLAML. FLAML stands for "Flexible Large-scale Automated Machine Learning." It is a machine learning library developed by Microsoft that allows users to easily train and deploy machine learning models on large datasets.

5.1.1 Light Gradient Boosting

Below is an general approach to how the algorithm is applied.

Input: training set $\{(x_i, y_i)\}_{i=1}^{n}$, a differentiable loss function $L(y, F(x))$, number of iterations M. Algorithm: 1. Initialize the model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, \gamma\right).$$

2. For $m = 1$ to M : 1. Compute so-called pseudo-residuals: for i=1, ..., n .

$$r_{im} = -\left[\frac{\partial L\left(y_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)}\right]_{F(x)=F_{m-1}(x)}$$

2. Fit a base learner (or weak learner, e.g. tree) closed under scaling $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^{n}$. 3. Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}\left(x_i\right) + \gamma h_m\left(x_i\right)\right).$$

4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

```
from lightgbm import LGBMClassifier
LGBMClassifier(colsample_bytree=0.5332477358865868,
               learning_rate=0.4072454511649998,
```

```
max_bin=255, min_child_samples=2,
n_estimators=139, num_leaves=115,
reg_alpha=0.006958608037974516,
reg_lambda=0.0009765625,
verbose=-1)
```

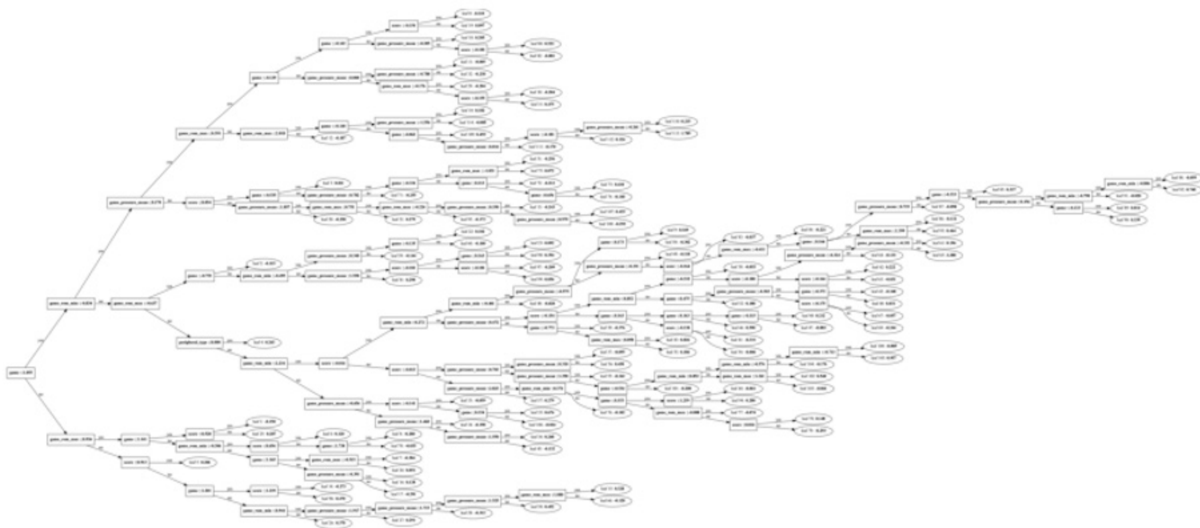

Figure 5.1: LGB classification stump. There are 139 stumps in the model. Due to the complexity of the tree the labels are difficult to read.

5.1.2 Extra Trees Classifier

Extra Trees Classifier:

```
from sklearn.ensemble import ExtraTreesClassifier
ExtraTreesClassifier(criterion='entropy', max_features=1.0,
                    max_leaf_nodes=8717, n_estimators=42, n_jobs=-1
```

5.1.3 Multi-Layer Perceptron (Deep Neural Network)

```
from sklearn.neural_network import MLPClassifier

MLPClassifier(activation = 'relu', alpha = 0.005,
              hidden_layer_sizes = (8,5,8), learning_rate = 'constant',
              solver = 'adam', max_iter=1000)
```

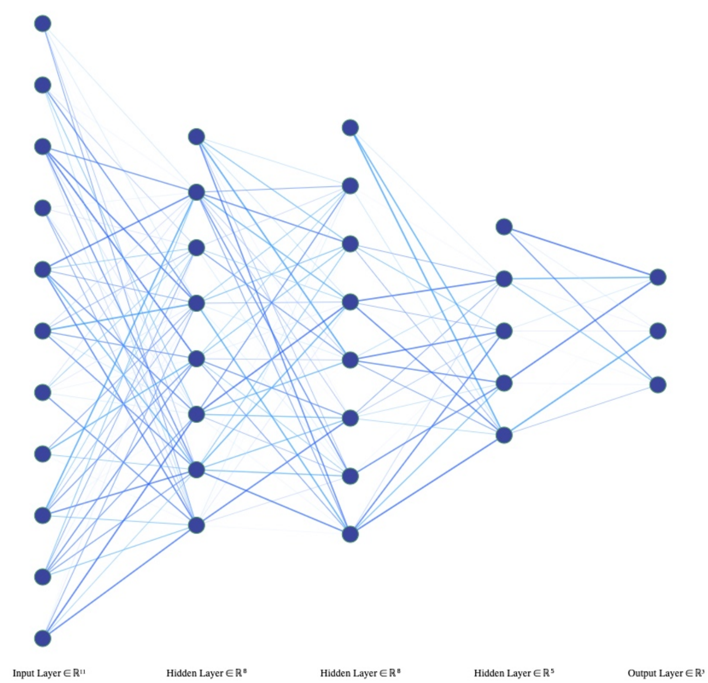

Figure 5.2: Several different architectures of the DNN can be applied. In this case, the dimensionality reduction techniques implied keeping the same number of dimensions for the hidden layers. Here, there are 3 hidden layers with dimensions (8,5,8) respectively.

5.1.4 Logistic Regression

The default architecture was used from the documentation website.

5.2 Results

The goal of our computational experiments was to compare the performance of different machine learning methods for the identification of stroke severity. The original harmonized dataset contained all the scores, minimum and maximum range of motion, and minimum and maximum pressure and we took the maximum score per game per session. Because of the smaller-sized dataset, the training and testing were split on the 80/20 principle where 20% of the data was the testing data (20).

Train Shape: (26321, 11)

Test Shape: (6581, 11)

In Table 5.1, we see a 10-fold cross-validation of each of the machine learning classification algorithms with Figure 6 as a visual display of a single accuracy measure. This was done on the chance that our model of high accuracy is over-fitting our model. This is where the dataset is randomly divided into 10 different subsets with 320 rows each (2). Each "folded" dataset becomes the new training data another becomes the testing data. The same model is used to fit the dataset and then tested on the random testing dataset. We then take the mean and standard deviation of the random sample of the accuracy.

In Figure 5.4, we see the confusion matrix of each of the supervised learning methods. Generally, a confusion matrix is used to visually represent the performance of the algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa. We represent the percentage over the actual numeric number for display purposes. Three metrics of performance come from the confusion matrix precision, recall, and the f1-score. Precision measures the proportion of predicted positives that are truly positive. Recall measures the proportion of predicted negatives that are truly negative. The f1-score is the harmonic mean of the precision and recall (4). In this case, this is macro-averaging (treating all classes as equally important). The interesting portion here is that the accuracy of the Extra Trees classifier is higher than the Light Gradient Boosting Method (LGBM), but when comparing the f1-score the LGBM is (in this case) the better classifier.

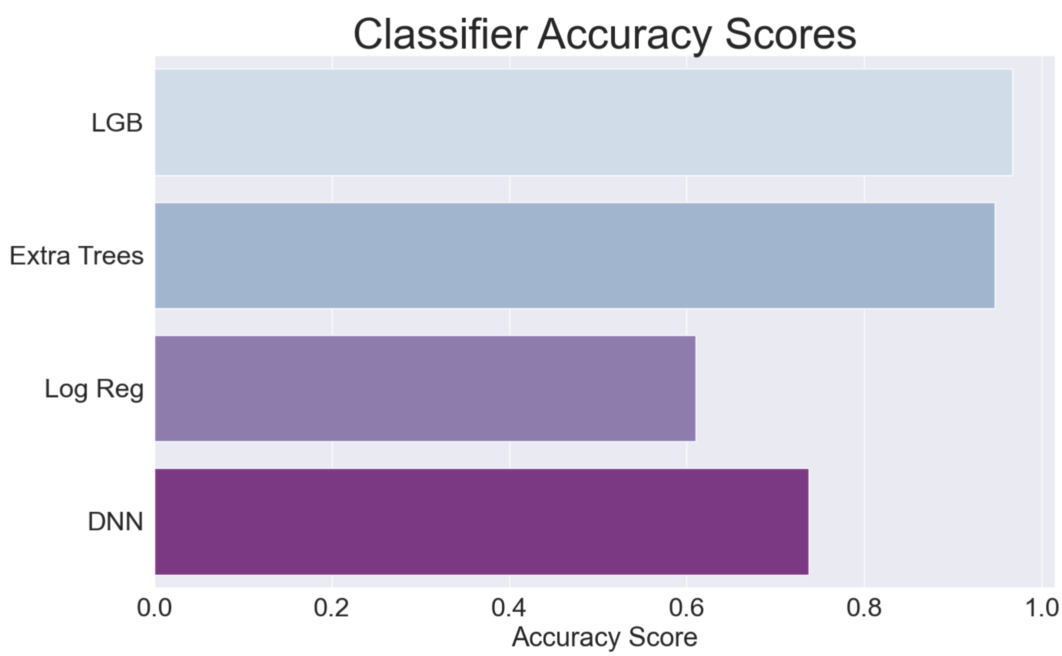

Figure 5.3: Accuracy Score for each Machine Learning Model. As shown, the Light Gradient Boosting (LGB) and Extra Trees have the greatest accuracy score. See Table 5.1 for the depiction of the cross-validation to prevent over-fitting (2). Note that accuracy alone is not the best illustration for the performance of a machine learning model. LGB is chosen as the final model in extension for future work.

Table 5.1: 10-Fold Cross-Validation Mean Scores with Standard Deviation

Classifier	Mean	Std
Extra Trees	96.40%	0.4%
Light Gradient Boosting	94.0%	0.4%
Neural Network	71.70%	0.7%
Logistic Regresssion	61.20%	0.5%

In order to prevent over-fitting 10-fold cross-validation is performed on the dataset (2). This is done by slicing the data into 10 different training and testing datasets and then training the model on each of those 10 ”sliced” datasets. The average score is taken from each of the models to indicate performance.

In the classification report, the f1-score for light gradient boosting receives an accuracy of 96%. This is an indication this is the model of choice. When considering the confusing matrix, the model does a better job of not classifying patients as having No ROM when the patient actually has a High ROM. When a model is wrong there needs to be a way to minimize this. Light gradient boosting does the best out of all the models along with high performance.

5.2.1 Determining Relevant Features

When using a bagging algorithm, the final step is to analyze the features of the model contributing most to the classification. There are two of these methods used: Extra Trees and Light Gradient Boosting. In Figure 5.6, notice that both algorithms have similar results.

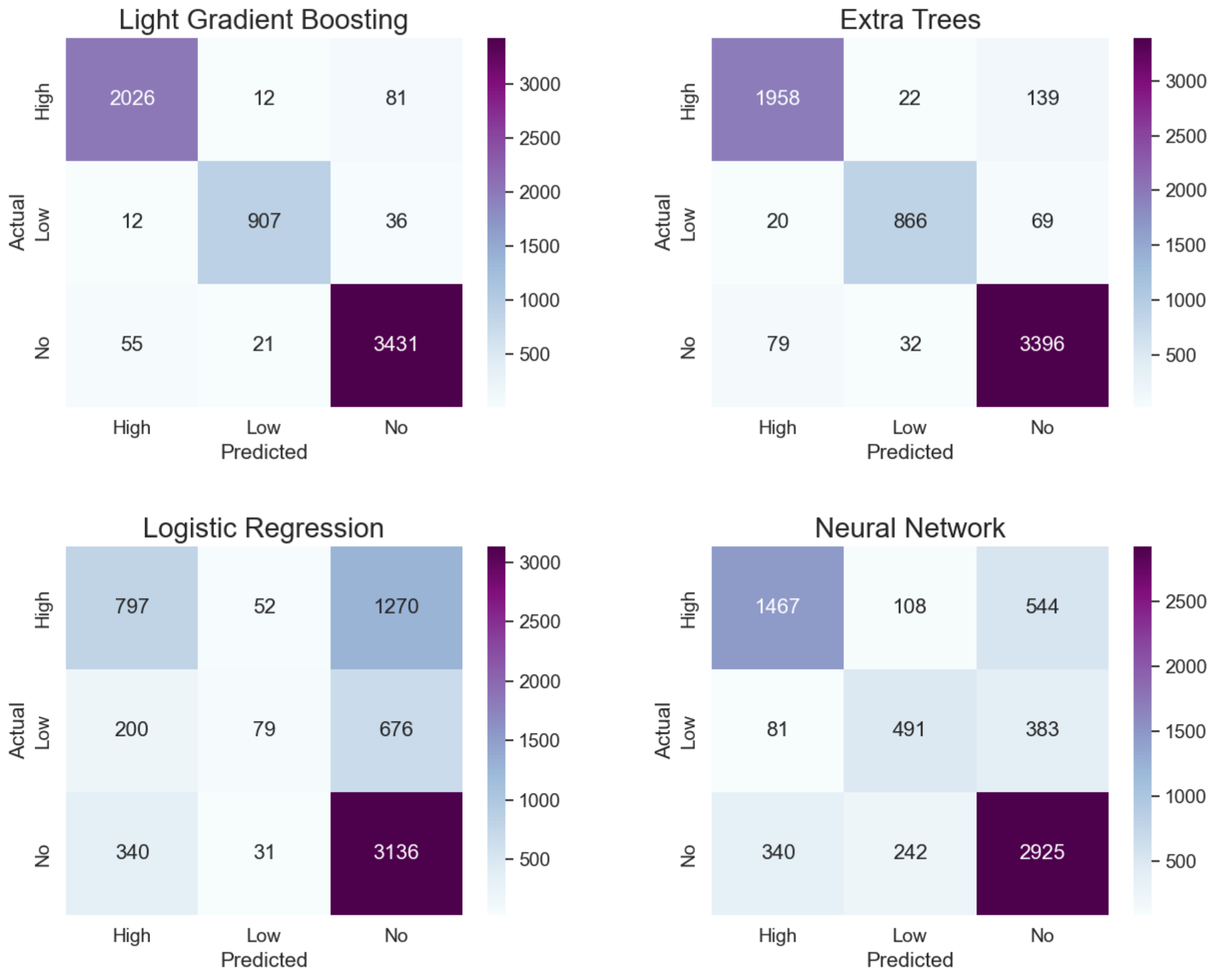

Figure 5.4: Confusion Matrices Light Gradient Boosting and Logistic Regression. Considering the false negative column of the No classification, it is seen that the LGB model greatly improved this classification. This is especially important when classifying a patient as having No stroke severity when they are actually a High severity.

Interestingly enough, the score variable contributed little in comparison to the other variables. This was against most intuition. The peripheral type as well contributed the least, which is most like because most of the patients used the Motus hand in the patient sessions. This is speculation. On the other hand, the number of movements, force extension and flexion have the most contributions to the model. This is very important since this

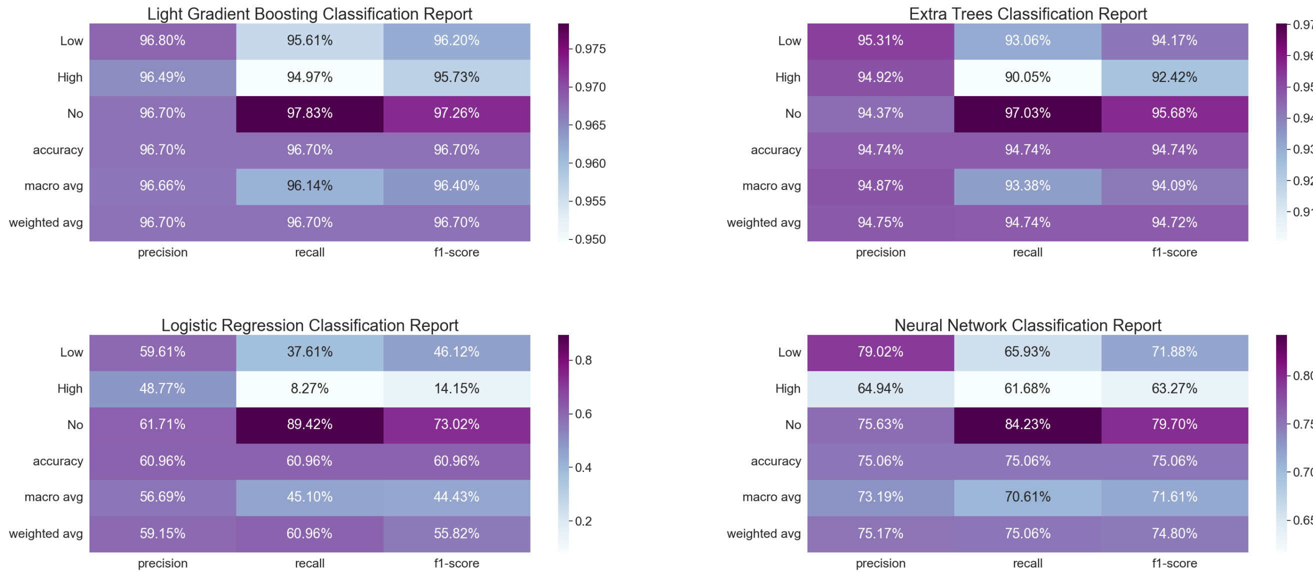

Figure 5.5: Classification Report Extra Trees, Light Gradient Boosting, Logistic Regression, Neural Network

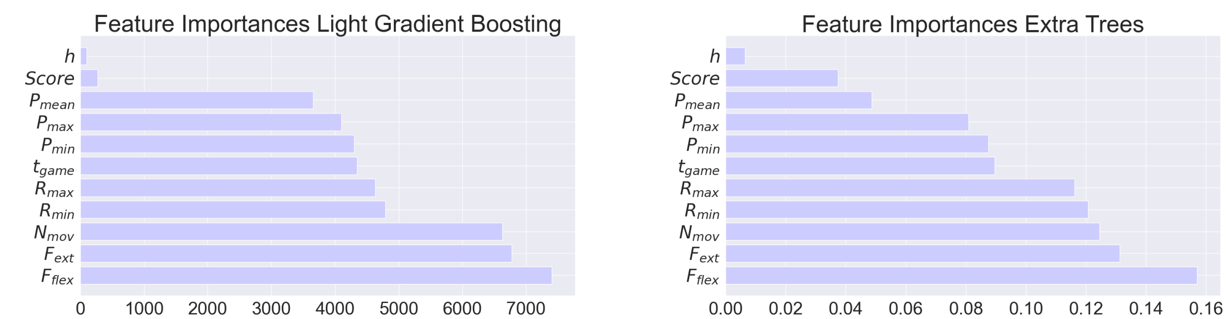

Figure 5.6: Feature Importances of each model displaying which variables make the greatest contributions to the model. As shown the $F_(ext)$ and $F_(flex)$ make the most contributions to the mode. Here we also see that the peripheral type (Hand or Foot) made the least. This could be due to the fact that only two patients used the foot as a lower extremity.

implies that the more the patients move their hand or foot, the greater contribution to their stroke classification! Although this does not imply (in this case) moving their hand decreases the stroke severity level. A time series analysis is needed.

CHAPTER 6
Conclusion

This study uses 4 computational methods via supervised learning methods to classify stroke severity for patients. From this, there not standard methods (logistic regression), black box methods (Deep Neural Network), and bagging methods (light gradient boosting and extra trees). There are many benefits to the bagging methods, one of them being interpretability. This is different from using a neural network, which may or may not be a stronger algorithm in the future. The interpretability of a model (see feature importances 5.6) allows there to be an interpretation of which variables contribute to the classification. This is assuming high accuracy and lack of over-fitting (seeing cross-validation table). For example, as stated before, the score variable has a small contribution although game time is highly contributed (both variables have a correlation of 0.60, interesting.

Robotic in-home patient therapy allows stroke patients to battle their illnesses in a more effective way. Not only do patients lack the ability to do day-to-day tasks like eating or tying a shoe, but they can also simply struggle to see a clinician without a caregiver with the inability to operate a motor vehicle. This can be a financial toll on the patient as well. With the combination of in-home robotic therapy and machine learning, there could be potential to do this in an autonomous way.

Predictive models can be a promising support tool for clinicians. Machine Learning algorithms can be easily deployed for this purpose, due to their capability of handling large cohorts and high dimensional datasets. Once trained, they provide accurate estimates at a low cost. Among these advantages, this kind of solution could stimulate a more data-driven approach in clinical practice, promote a more structured definition of studies and reduce the gap between clinical and research areas. For this reason, more data will be gathered in time, with more analyzes used such as time-series approaches to lead to further embedded decision-support tools for the daily use of not only the clinician but the patient as well.

In another study, there can be a greater quantitative understanding of patient therapy improved through robotic therapy. This can be done by simply having more session data gathered. Potentially taking a group of patients and monitoring the timer series analysis to see when a stroke severity changes from High to Low or Low to No. There could even be a web application for each individual patient to track progress. This could allow more data to be gathered faster, but patient labeling would have to be done separately.

This could allow patients to track their progress in real time on a weekly and monthly basis. Insights with these models can offer a practical guideline for future design for improved treatment. As more and more session data is gathered there can even be a single diagnosis of which a patient should be improving to have more effective treatment on the individual level.

Appendices

A Data Dictionary

Table 1: Data Dictionary Table

Variable	Description	Unit	Example
$Start\ Time$	Start time of the session	Datetime	31:44.2
$Game\ ID$	Unique game identifier assigned the 30 games	0-29	16
$Session\ ID$	Unique identifier for a therapy session	RNG	338057214367
$Patient\ ID$	Unique user identifier preserved across Motus Nova databases	integer	1113
F_{flex}	Maximum centripetal force generated moving in the downward direction (computed from derivatives of angle data)	neutons	-3.047709105
F_{ext}	Maximum centripetal force generated moving in the upward direction (computed from derivatives of angle data)	neutons	3.251405759
N_{mov}	The number of completed movements	integer	10
R_{min}	Absolute minimum angle detected by angle sensor during therapy session	degrees	-25
R_{max}	Absolute maximum angle detected by angle sensor during therapy session	degrees	46.41941
t_{game}	Total time spent performing therapy during a session for that game	seconds	15
P_{min}	Minimum Pressure applied by the sensor in a patient session	psi	-0.04511994
P_{max}	Maximum pressure applied by the sensor in a patient session	psi	10.30989
P_{mean}	Average pressure applied by the sensor in a patient session	psi	3.590553432
$Score$	Score achieve by patient per video game	integer	100
h	Peripheral type variable indicating the Hand or Foot	0, 1	Hand
$Class$	Designate stroke severity label for a patient by a clinician (High, Low, No)	0, 1, 2	High

B Python Code

Displaying all of the code used in the analysis would take over 50 pages of this thesis. Here is a link to the Github repository: https://github.com/rgreenfield/GSU_MotusNova_Data

A TallDataFrame.py

```
def tall_df(data, game_dict):


    #Construct an empty dataframe replacing the game variables
    cols = ['game_id', 'patient_id', 'start_time',
            'session_id', 'active_duration', 'passive_duration', 'duration',
            'rom_min', 'rom_max', 'force_flexion','force_extension',
            'num_movements', 'game_rom_min', 'game_rom_max', 'game',
            'game_pressure_min', 'game_pressure_max', 'game_pressure_mean']

    df = pd.DataFrame(columns=cols)


    for idx, row in data.iterrows():
        for key in game_dict:
            try:
                dict = {
                'game_id':              game_dict[key],
```

```
    'patient_id':          row['patient_id'],
    'start_time':          row['start_time'],
    'session_id':          row['session_id'],
    'active_duration':     row['active_duration'],
    'passive_duration':    row['passive_duration'],
    'duration':            row['duration'],
    'rom_min':             row['rom_min'],
    'rom_max':             row['rom_max'],
    'force_flexion':       row['force_flexion'],
    'force_extension':     row['force_extension'],
    'num_movements':       row['num_movements'],
    'game_rom_min':        row[key + '_rom_min'],
    'game_rom_max':        row[key + '_rom_max'],
    'game':                row[key],
    'game_pressure_min':   row[key + '_pressure_min'],
    'game_pressure_max':   row[key + '_pressure_max'],
    'game_pressure_mean': row[key + '_pressure_mean'],
    }

    df = df.append(dict, ignore_index=True)
except:
```

```
                pass
    return df


def concat_tall_dataframes(df_old, df_new):

    chunks = [df_old, df_new]
    return pd.concat(chunks, ignore_index=True)


def main():

    # I want to add some sort of a list to loop through

    for _ in range(1):

        file_list = [
            'user_session_data_with_pressure.txt',
            'user_session_data_with_pressure_2.txt',
            'user_session_data_with_pressure_dataframe_new.txt',
            'labelled_session_data.txt',]

        path_new = './data/raw_data/' + file_list[_]
```

```
data_wide = pd.read_csv(path_new, delimiter='|',
            index_col=0)

game_dict = {'Console':                0,
             'Thermometer':            1,
             'Strongman':              2,
             'Strongman Up':           3,
             'Space Shooter':          4,
             'Blocks':                 5,
             'Golf':                   6,
             'Balloon Rider':          7,
             'Brick Breaker':          8,
             'Assessment':             26,
             'Cosmic Tennis':          10,
             'Slot Machine':           11,
             'Classic Assessment':     12,
             'Coin Push':              13,
             'Fishing':                14,
             'Symphony Hero':          15,
             'Solitaire':              16,
             'Main Menu':              17,
```

```
    'Sphero':                18,
    'Shape Shooter':         19,
    'Peggling':              20,
    'Plinko':                21,
    'Countdown':             22,
    'Tower':                 23,
    'Motus Chef':            24,
    'Motus Pets':            25,
    'Black Jack':            27,
    'Motus Karts':           28,
    'Candy Swap':            29,
    'Nick Assessment':       30,
    }


#df_new = tall_df(data_wide, game_dict)


#path_old = './data/Tall_DataFrame_Pressure.txt'
#df_old = pd.read_csv(path_old, delimiter='|', index_col=0)


#df_final = concat_tall_dataframes(df_old, df_new)
```

```
        df = tall_df(data_wide, game_dict)

        # Save file in to the unlabeled processed data
        df.to_csv(f'./data/processed_data/tall_unlabeled/
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣Tall_DataFrame_Pressure␣{str(_)}.txt' , sep='|')

if __name__ == '__main__':
    main()
```

B max_score_per_session.py

```
def create_unlabel_dataframe():

    #tall_dataframe_path = "./data/processed_data/tall_unlabeled/
    Tall_DataFrame_Pressure.txt"

␣␣␣␣tall_dataframe_path␣=␣"./data/processed_data/tall_unlabeled/
    Tall_DataFrame_Pressure_old_Jan_8_2022.txt"
␣␣␣␣unlabeled_data␣=␣pd.read_csv(tall_dataframe_path,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣delimiter="|",␣index_col=0)
␣␣␣␣unlabeled_data␣=␣unlabeled_data.drop(
␣␣␣␣␣␣␣␣["active_duration",␣"passive_duration",
```

```
        "rom_min", "rom_max"], axis=1
    )

    return unlabeled_data


def create_label_dataframe():
    labelpath = "./data/raw_data/
    label_data/ROM Labelled Data - Sheet.csv"
    drop_label = [
        "Date of Assesment",
        "Max", "Min", "Assistance",
        "SW Version", "ROM Assessment Type",
    ]

    return (
        pd.read_csv(labelpath)
        .rename(columns={"ID": "patient_id"})
        .drop(drop_label, axis=1))


def create_peripheral_dataframe():
    peripath = "./data/raw_data/label_data/
```

```
            Peripheral Type Lookup - Sheet.csv"
    return pd.read_csv(peripath)


def create_score_dataframe():


    scorepath = "./data/raw_data/user_score_data.txt"
    score_data = pd.read_csv(scorepath,
                             delimiter="|",
                             index_col=0).rename(
                             columns={"Unnamed: 0": "time_stamp"})


    score_drop = ["device_id", "dynamic_rom_max", "dynamic_rom_min",
            "game", "session_index", "isTesting",
            "is_low_rom", "is_no_rom", "pt_pressure_override",
            "set_initial_rom",]


    score_data = score_data.drop(score_drop, axis=1)


    sort_score_columns = ["session_id",
                          "game_id",
                          "patient_id",
```

```
                        "time_stamp",
                        "game_time",
                        "level",
                        "score",]

    score_data = score_data[sort_score_columns]

    return score_data

def create_max_score_dataframe():

    score_data = create_score_dataframe()



    agg_func_max_score = {
                            "score": "max",
                          }

    # merge the df_max_game_score with the score_data to have
    # the "level" and "game_time" variables
```

```
    return (score_data.groupby([
                                "session_id",
                                "game_id",
                                "patient_id",
                                ]).agg(agg_func_max_score
                                ).reset_index())

def create_max_score_game_dataframe():

    score_data = create_score_dataframe()
    max_game_score = create_max_score_dataframe()

    df_max_game_score = max_game_score.merge(score_data,
                                    on=["patient_id",
                                        "game_id",
                                        "session_id",
                                        "score"])

    df_max_game_score = df_max_game_score[[
                                    'session_id',
                                    'game_id',
```

```
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣'patient_id',
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣'time_stamp',
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣'game_time',
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣'level',
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣'score',]]
␣␣␣␣df_max_game_score␣=␣df_max_game_score.drop('time_stamp',␣axis=1)
␣␣␣␣return␣df_max_game_score

def␣create_unlabeled_score():
␣␣␣␣df_unlabeled␣=␣create_unlabel_dataframe()
␣␣␣␣df_score␣=␣create_score_dataframe()
␣␣␣␣ls␣=␣["patient_id",␣"game_id",␣"session_id",]
␣␣␣␣df_unlabeled_score␣=␣df_unlabeled.merge(df_score,␣on=ls)
␣␣␣␣return␣df_unlabeled_score

def␣create_unlabeled_max_score():
␣␣␣␣df_unlabeled␣=␣create_unlabel_dataframe()
␣␣␣␣#df_max_score␣=␣create_score_dataframe()
␣␣␣␣#df_max_score␣=␣create_max_score_game_dataframe()
␣␣␣␣df_max_score␣=␣create_max_score_dataframe()
␣␣␣␣return␣df_unlabeled.merge(
```

```
        df_max_score, on=["patient_id",
                          "game_id",
                          "session_id"])

def create_labeled_score():

    df_unlabeled_score = create_unlabeled_score()
    df_peripheral = create_peripheral_dataframe()

    labeled_data = create_label_dataframe()

    df_label_score = df_unlabeled_score
                .merge(labeled_data, on=["patient_id",])

    return df_label_score.merge(df_peripheral, on="patient_id")
                .drop("Peripheral Type", axis=1)

def label_max_score():
    df = create_labeled_score()
    return df.groupby(["session_id", "patient_id"]).
               agg({"score": "max"}).reset_index()
```

```
def create_labeled_max_score():

    df_unlabeled_max_score = create_unlabeled_max_score()
    df_peripheral = create_peripheral_dataframe()

    labeled_data = create_label_dataframe()

    # merging labeled data with the unlabeled score data
    df_label_max_score = df_unlabeled_max_score
                        .merge(labeled_data, on=["patient_id",])

    return df_label_max_score.merge(df_peripheral, on="patient_id")
                              .drop("Peripheral Type", axis=1)

def max_score_per_session_game():

    df = create_labeled_max_score()

    df["start_time"] = pd.to_datetime(df["start_time"])
```

```
    df = df.sort_values(by="start_time")

    # variable constraints like having negative pressure
    df = cap_data(df)

    score_col_sort = ["game_id", "patient_id", "session_id", "start_time",
                      "force_flexion", "force_extension", "num_movements",
                      "game_rom_min", "game_rom_max", "game",
                      "game_pressure_min", "game_pressure_max",
                      "game_pressure_mean",
                      "score", "peripheral_type",
                      "Classification",]

    df = df[score_col_sort]

    # 4397 drop values (I am going to try to fill this missing data)
    #df = df.dropna()

    # number of dropped duplicate values
    df = df.drop_duplicates()
```

```
    fill_missing_data(df)
    label_encoding(df)

    return df


def label_encoding(df):
    LE = LabelEncoder()
    # {foot: 0, hand: 1}
    df["peripheral_type"] = LE.fit_transform(df["peripheral_type"])
    # {low: 0, high: 1, No: 2}
    df["Classification"] = LE.fit_transform(df["Classification"])


def fill_missing_data(df):


    """
    Missing data: ["game_rom_min", "game_rom_max", "game"]


    fillna(0) for ["game_rom_min", "game_rom_max"]
    """


    missing_data = {"game_rom_min": df['game_rom_min'].mean(),
```

```
                "game_rom_max": df['game_rom_max'].mean(),
                "game": df['game'].min()}

    return df.fillna(missing_data, inplace=True)

df = max_score_per_session_game()
```

REFERENCES

[1] Sahil Bajaj, Stephen N. Housley, David Wu, Mukesh Dhamala, G. A. James, and Andrew J. Butler. Dominance of the unaffected hemisphere motor network and its role in the behavior of chronic stroke survivors. *Frontiers in Human Neuroscience*, 10, 12 2016.

[2] Daniel Berrar. Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 1-3:542–545, 1 2018.

[3] Silvia Campagnini, Chiara Arienti, Michele Patrini, Piergiuseppe Liuzzi, Andrea Mannini, and Maria Chiara Carrozza. Machine learning methods for functional recovery prediction and prognosis in post-stroke rehabilitation: a systematic review, 12 2022.

[4] Ivo Düntsch and Günther Gediga. Confusion matrices and rough set data analysis. *Journal of Physics: Conference Series*, 1229(1):012055, may 2019.

[5] Valery L Feigin, Benjamin A Stark, Catherine Owens Johnson, Gregory A Roth, Catherine Bisignano, Gdiom Gebreheat Abady, Mitra Abbasifard, Mohsen Abbasi-Kangevari, Foad Abd-Allah, Vida Abedi, Ahmed Abualhasan, Niveen ME Abu-Rmeileh, Abdelrahman I Abushouk, Oladimeji M Adebayo, Gina Agarwal, Pradyumna Agasthi, Bright Opoku Ahinkorah, Sohail Ahmad, Sepideh Ahmadi, Yusra Ahmed Salih, Budi Aji, Samaneh Akbarpour, Rufus Olusola Akinyemi, Hanadi Al Hamad, Fares Alahdab, Sheikh Mohammad Alif, Vahid Alipour, Syed Mohamed Aljunid, Sami Almustanyir, Rajaa M Al-Raddadi, Rustam Al-Shahi Salman, Nelson Alvis-Guzman, Robert Ancuceanu, Deanna Anderlini, Jason A Anderson, Adnan Ansar, Ippazio Cosimo Anton-

azzo, Jalal Arabloo, Johan Ärnlöv, Kurnia Dwi Artanti, Zahra Aryan, Samaneh Asgari, Tahira Ashraf, Mohammad Athar, Alok Atreya, Marcel Ausloos, Atif Amin Baig, Ovidiu Constantin Baltatu, Maciej Banach, Miguel A Barboza, Suzanne Lyn Barker-Collo, Till Winfried Bärnighausen, Mark Thomaz Ugliara Barone, Sanjay Basu, Gholamreza Bazmandegan, Ettore Beghi, Mahya Beheshti, Yannick Béjot, Arielle Wilder Bell, Derrick A Bennett, Isabela M Bensenor, Woldesellassie Mequanint Bezabhe, Yihienew Mequanint Bezabih, Akshaya Srikanth Bhagavathula, Pankaj Bhardwaj, Krittika Bhattacharyya, Ali Bijani, Boris Bikbov, Mulugeta M Birhanu, Archith Boloor, Aime Bonny, Michael Brauer, Hermann Brenner, Dana Bryazka, Zahid A Butt, Florentino Luciano Caetano dos Santos, Ismael R Campos-Nonato, Carlos Cantu-Brito, Juan J Carrero, Carlos A Castañeda-Orjuela, Alberico L Catapano, Promit Ananyo Chakraborty, Jaykaran Charan, Sonali Gajanan Choudhari, Enayet Karim Chowdhury, Dinh-Toi Chu, Sheng-Chia Chung, David Colozza, Vera Marisa Costa, Simona Costanzo, Michael H Criqui, Omid Dadras, Baye Dagnew, Xiaochen Dai, Koustuv Dalal, Albertino Antonio Moura Damasceno, Emanuele D'Amico, Lalit Dandona, Rakhi Dandona, Jiregna Darega Gela, Kairat Davletov, Vanessa De la Cruz-Góngora, Rupak Desai, Deepak Dhamnetiya, Samath Dhamminda Dharmaratne, Mandira Lamichhane Dhimal, Meghnath Dhimal, Daniel Diaz, Martin Dichgans, Klara Dokova, Rajkumar Doshi, Abdel Douiri, Bruce B Duncan, Sahar Eftekharzadeh, Michael Ekholuenetale, Nevine El Nahas, Islam Y Elgendy, Muhammed Elhadi, Shaimaa I El-Jaafary, Matthias Endres, Aman Yesuf Endries, Daniel Asfaw Erku, Emerito Jose A Faraon, Umar Farooque, Farshad Farzadfar, Abdullah Hamid Feroze, Irina Filip, Florian Fischer, David Flood, Mohamed M Gad, Shilpa Gaidhane, Reza Ghanei Gheshlagh, Ahmad Ghashghaee, Nermin Ghith, Ghozali Ghozali, Sherief Ghozy, Alessandro Gialluisi, Simona Giampaoli, Syed Amir Gilani, Paramjit Singh Gill, Elena V Gnedovskaya, Mahaveer Golechha, Alessandra C Goulart, Yuming Guo, Rajeev Gupta, Veer Bala Gupta, Vivek Kumar Gupta, Pradip Gyanwali, Nima Hafezi-Nejad, Samer Hamidi, Asif Hanif, Graeme J Hankey, Arief Hargono, Abdiwahab Hashi, Treska S Hassan, Hamid Yimam Hassen, Rasmus J Havmoeller, Simon I Hay, Khezar Hayat, Mohamed I Hegazy, Claudiu Herteliu, Ramesh Holla, Sorin Hostiuc, Mowafa Househ, Junjie Huang, Ayesha Humayun, Bing-Fang Hwang, Licia Iacoviello, Ivo Iavicoli, Segun Emmanuel Ibitoye, Olayinka Stephen Ilesanmi, Irena M Ilic, Milena D Ilic, Usman Iqbal, Seyed Sina Naghibi Irvani, Sheikh Mohammed Shariful Islam, Nahlah Elkudssiah Ismail, Hiroyasu Iso, Gaetano Isola, Masao Iwagami, Louis Jacob, Vardhmaan Jain, Sung-In Jang, Sathish Kumar Jayapal, Shubha Jayaram, Ranil Jayawardena, Panniyammakal Jeemon, Ravi Prakash Jha, Walter D Johnson, Jost B Jonas, Nitin Joseph, Jacek Jerzy Jozwiak, Mikk Jürisson, Rizwan Kalani, Rohollah Kalhor, Yogeshwar Kalkonde, Ashwin Kamath, Zahra Kamiab, Tanuj Kanchan, Himal Kandel, André Karch, Patrick DMC Katoto, Gbenga A Kayode, Pedram Keshavarz, Yousef Saleh Khader, Ejaz Ahmad Khan, Imteyaz A Khan, Maseer Khan, Moien AB Khan, Mahalaqua Nazli Khatib, Jagdish Khubchandani, Gyu Ri Kim, Min Seo Kim, Yun Jin Kim, Adnan Kisa, Sezer Kisa, Mika Kivimäki, Dhaval Kolte, Ali Koolivand, Sindhura Lakshmi Koulmane Laxminarayana, Ai Koyanagi, Kewal Krishan, Vijay Krishnamoorthy, Rita V Krishnamurthi, G Anil Kumar, Dian Kusuma, Carlo La Vecchia, Ben Lacey, Hassan Mehmood Lak, Tea Lallukka, Savita Lasrado, Pablo M Lavados, Matilde Leonardi, Bingyu Li, Shanshan Li, Hualiang Lin, Ro-Ting Lin, Xuefeng Liu, Warren David Lo, Stefan Lorkowski, Giancarlo Lucchetti, Ricardo Lutzky Saute, Hassan Magdy Abd El Razek, Francesca Giulia Magnani, Preetam Bhalchandra Mahajan, Azeem Majeed, Alaa Makki, Reza Malekzadeh, Ahmad Azam Malik, Navid Manafi, Mohammad Ali Mansournia, Lorenzo Giovanni Mantovani, Santi Martini, Giampiero Mazzaglia, Man Mohan Mehndiratta, Ritesh G Menezes, Atte Meretoja, Amanual Getnet Mersha, Junmei Miao Jonasson, Bartosz Miazgowski, Tomasz Miazgowski, Irmina Maria Michalek, Erkin M Mirrakhimov, Yousef Mohammad, Abdollah Mohammadian-Hafshejani, Shafiu Mohammed, Ali H Mokdad, Yaser Mokhayeri, Mariam Molokhia, Mohammad Ali Moni, Ahmed Al Montasir, Rahmatollah Moradzadeh, Lidia Morawska, Jakub Morze, Walter Muruet, Kamarul Imran Musa, Ahamarshan Jayaraman Nagarajan, Mohsen Naghavi, Sreenivas Narasimha Swamy, Bruno Ramos Nascimento, Ruxandra Irina Negoi, Sandhya Neupane Kandel, Trang Huyen Nguyen, Bo Norrving, Jean Jacques Noubiap, Vincent Ebuka Nwatah, Bogdan Oancea, Oluwakemi Ololade Odukoya, Andrew T Olagunju, Hans Orru, Mayowa O Owolabi, Jagadish Rao Padubidri, Adrian Pana, Tarang Parekh, Eun-Cheol Park, Fatemeh Pashazadeh Kan, Mona Pathak, Mario F P Peres, Arokiasamy Perianayagam, Truong-Minh Pham, Michael A Piradov, Vivek Podder, Suzanne Polinder, Maarten J Postma, Akram Pourshams, Amir Radfar, Alireza Rafiei, Alberto Raggi, Fakher Rahim, Vafa Rahimi-Movaghar, Mosiur Rahman, Muhammad Aziz Rahman, Amir Masoud Rahmani, Nazanin Rajai, Priyanga Ranasinghe, Chythra R Rao, Sowmya J Rao, Priya Rathi, David Laith Rawaf, Salman Rawaf, Marissa B Reitsma, Vishnu Renjith, Andre M N Renzaho, Aziz Rezapour, Jefferson Antonio Buendia Rodriguez, Leonardo Roever, Michele Romoli, Andrzej Rynkiewicz, Simona Sacco, Masoumeh Sadeghi, Sahar Saeedi Moghaddam, Amirhossein Sahebkar, KM Saif-Ur-Rahman, Rehab Salah, Mehrnoosh Samaei, Abdallah M Samy, Itamar S Santos, Milena M Santric-Milicevic, Nizal Sarrafzadegan, Brijesh Sathian, Davide Sattin, Silvia Schiavolin, Markus P Schlaich, Maria Inês Schmidt, Aletta Elisabeth Schutte, Sadaf G Sepanlou, Allen Seylani, Feng Sha, Saeed Shahabi, Masood Ali Shaikh, Mohammed Shannawaz, Md Shajedur Rahman Shawon, Aziz Sheikh, Sara Sheikhbahaei, Kenji Shibuya, Soraya Siabani, Diego Augusto Santos Silva, Jasvinder A Singh, Jitendra Kumar Singh, Valentin Yurievich Skryabin, Anna Aleksandrovna Skryabina, Badr Hasan Sobaih, Stefan Stortecky, Saverio Stranges, Eyayou Girma Tadesse, Ingan Ukur Tarigan, Mohamad-Hani Temsah, Yvonne Teuschl, Amanda G Thrift, Marcello Tonelli, Marcos Roberto Tovani-Palone, Bach Xuan Tran, Manjari Tripathi, Gebiyaw Wudie Tsegaye, Anayat Ullah, Brigid Unim, Bhaskaran Unnikrishnan, Alireza Vakilian, Sahel Valadan Tahbaz, Tommi Juhani Vasankari, Narayanaswamy Venketasubramanian, Dominique Vervoort, Bay Vo, Victor Volovici, Kia Vosoughi, Giang Thu Vu, Linh Gia Vu, Hatem A Wafa, Yasir Waheed, Yanzhong Wang, Tissa Wijeratne, Andrea Sylvia Winkler, Charles D A Wolfe, Mark Woodward, Jason H Wu, Sarah Wulf Hanson, Xiaoyue Xu, Lalit Yadav, Ali Yadollah-

pour, Seyed Hossein Yahyazadeh Jabbari, Kazumasa Yamagishi, Hiroshi Yatsuya, Naohiro Yonemoto, Chuanhua Yu, Ismaeel Yunusa, Muhammed Shahriar Zaman, Sojib Bin Zaman, Maryam Zamanian, Ramin Zand, Alireza Zandifar, Mikhail Sergeevich Zastrozhin, Anasthasia Zastrozhina, Yunquan Zhang, Zhi-Jiang Zhang, Chenwen Zhong, Yves Miel H Zuniga, and Christopher J L Murray. Global, regional, and national burden of stroke and its risk factors, 1990–2019: a systematic analysis for the global burden of disease study 2019. *The Lancet Neurology*, 20:795–820, 10 2021.

[6] Catherine Owens Johnson, Minh Nguyen, Gregory A. Roth, Emma Nichols, Tahiya Alam, Degu Abate, Foad Abd-Allah, Ahmed Abdelalim, Haftom Niguse Abraha, Niveen Me Abu-Rmeileh, Oladimeji M. Adebayo, Abiodun Moshood Adeoye, Gina Agarwal, Sutapa Agrawal, Amani Nidhal Aichour, Ibtihel Aichour, Miloud Taki Eddine Aichour, Fares Alahdab, Raghib Ali, Nelson Alvis-Guzman, Nahla Hamed Anber, Mina Anjomshoa, Jalal Arabloo, Antonio Arauz, Johan Ärnlöv, Amit Arora, Ashish Awasthi, Maciej Banach, Miguel A. Barboza, Suzanne Lyn Barker-Collo, Till Winfried Bärnighausen, Sanjay Basu, Abate Bekele Belachew, Yaschilal Muche Belayneh, Derrick A. Bennett, Isabela M. Bensenor, Krittika Bhattacharyya, Belete Biadgo, Ali Bijani, Boris Bikbov, Muhammad Shahdaat Bin Sayeed, Zahid A. Butt, Lucero Cahuana-Hurtado, Juan J. Carrero, Félix Carvalho, Carlos A. Castañeda-Orjuela, Franz Castro, Ferrán Catalá-López, Yazan Chaiah, Peggy Pei Chia Chiang, Jee Young J. Choi, Hanne Christensen, Dinh Toi Chu, Monica Cortinovis, Albertino Antonio Moura Damasceno, Lalit Dandona, Rakhi Dandona, Ahmad Daryani, Kairat Davletov, Barbora De

Courten, Vanessa De la Cruz-Góngora, Meaza Girma Degefa, Samath Dhamminda Dharmaratne, Daniel Diaz, Manisha Dubey, Eyasu Ejeta Duken, Dumessa Edessa, Matthias Endres, Emerito Jose A. Faraon, Farshad Farzadfar, Eduarda Fernandes, Florian Fischer, Luisa Sorio Flor, Morsaleh Ganji, Abadi Kahsu Gebre, Teklu Gebrehiwo Gebremichael, Birhanu Geta, Kebede Embaye Gezae, Paramjit Singh Gill, Elena V. Gnedovskaya, Hector Gómez-Dantés, Alessandra C. Goulart, Giuseppe Grosso, Yuming Guo, Rajeev Gupta, Arvin Haj-Mirzaian, Arya Haj-Mirzaian, Samer Hamidi, Graeme J. Hankey, Hamid Yimam Hassen, Simon I. Hay, Mohamed I. Hegazy, Behnam Heidari, Nabeel A. Herial, Mohammad Ali Hosseini, Sorin Hostiuc, Seyed Sina Naghibi Irvani, Sheikh Mohammed Shariful Islam, Nader Jahanmehr, Mehdi Javanbakht, Ravi Prakash Jha, Jost B. Jonas, Jacek Jerzy Józwiak, Mikk Jürisson, Amaha Kahsay, Rizwan Kalani, Yogeshwar Kalkonde, Teshome Abegaz Kamil, Tanuj Kanchan, André Karch, Narges Karimi, Hamidreza Karimi-Sari, Amir Kasaeian, Tesfaye Dessale Kassa, Hossein Kazemeini, Adane Teshome Kefale, Yousef Saleh Khader, Ibrahim A. Khalil, Ejaz Ahmad Khan, Young Ho Khang, Jagdish Khubchandani, Daniel Kim, Yun Jin Kim, Adnan Kisa, Mika Kivimäki, Ai Koyanagi, Rita K. Krishnamurthi, G. Anil Kumar, Alessandra Lafranconi, Sarah Lewington, Shanshan Li, Warren David Lo, Alan D. Lopez, Stefan Lorkowski, Paulo A. Lotufo, Mark T. Mackay, Marek Majdan, Reza Majdzadeh, Azeem Majeed, Reza Malekzadeh, Navid Manafi, Mohammad Ali Mansournia, Man Mohan Mehndiratta, Varshil Mehta, Getnet Mengistu, Atte Meretoja, Tuomo J. Meretoja, Bartosz Miazgowski, Tomasz Miazgowski, Ted R. Miller, Erkin M. Mirrakhimov, Bahram

Mohajer, Yousef Mohammad, Milad Mohammadoo-Khorasani, Shafiu Mohammed, Farnam Mohebi, Ali H. Mokdad, Yaser Mokhayeri, Ghobad Moradi, Lidia Morawska, Ilais Moreno Velásquez, Seyyed Meysam Mousavi, Oumer Sada S. Muhammed, Walter Muruet, Mehdi Naderi, Mohsen Naghavi, Gurudatta Naik, Bruno Ramos Nascimento, Ruxandra Irina Negoi, Cuong Tat Nguyen, Long Hoang Nguyen, Yirga Legesse Nirayo, Bo Norrving, Jean Jacques Noubiap, Richard Ofori-Asenso, Felix Akpojene Ogbo, Andrew T. Olagunju, Tinuke O. Olagunju, Mayowa Ojo Owolabi, Jeyaraj Durai Pandian, Shanti Patel, Norberto Perico, Michael A. Piradov, Suzanne Polinder, Maarten J. Postma, Hossein Poustchi, V. Prakash, Mostafa Qorbani, Alireza Rafiei, Fakher Rahim, Kazem Rahimi, Vafa Rahimi-Movaghar, Mahfuzar Rahman, Muhammad Aziz Rahman, Cesar Reis, Giuseppe Remuzzi, Andre M.N. Renzaho, Stefano Ricci, Nicholas L.S. Roberts, Stephen R. Robinson, Leonardo Roever, Gholamreza Roshandel, Parisa Sabbagh, Hosein Safari, Saeed Safari, Saeid Safiri, Amirhossein Sahebkar, Saleh Salehi Zahabi, Abdallah M. Samy, Paola Santalucia, Itamar S. Santos, João Vasco Santos, Milena M. Santric Milicevic, Benn Sartorius, Arundhati R. Sawant, Aletta Elisabeth Schutte, Sadaf G. Sepanlou, Azadeh Shafieesabet, Masood Ali Shaikh, Mehran Shams-Beyranvand, Aziz Sheikh, Kevin N. Sheth, Kenji Shibuya, Mika Shigematsu, Min Jeong Shin, Ivy Shiue, Soraya Siabani, Badr Hasan Sobaih, Luciano A. Sposato, Ipsita Sutradhar, P. A. Sylaja, Cassandra E.I. Szoeke, Braden James Te Ao, Mohamad Hani Temsah, Omar Temsah, Amanda G. Thrift, Marcello Tonelli, Roman Topor-Madry, Bach Xuan Tran, Khanh Bao Tran, Thomas Clement Truelsen, Afewerki Gebremeskel

Tsadik, Irfan Ullah, Olalekan A. Uthman, Muthiah Vaduganathan, Pascual R. Valdez, Tommi Juhani Vasankari, Rajagopalan Vasanthan, Narayanaswamy Venketasubramanian, Kia Vosoughi, Giang Thu Vu, Yasir Waheed, Elisabete Weiderpass, Kidu Gidey Weldegwergs, Ronny Westerman, Charles D.A. Wolfe, Dawit Zewdu Wondafrash, Gelin Xu, Ali Yadollahpour, Tomohide Yamada, Hiroshi Yatsuya, Ebrahim M. Yimer, Naohiro Yonemoto, Mahmoud Yousefifard, Chuanhua Yu, Zoubida Zaidi, Mohammad Zamani, Afshin Zarghi, Yunquan Zhang, Sanjay Zodpey, Valery L. Feigin, Theo Vos, and Christopher J.L. Murray. Global, regional, and national burden of stroke, 1990–2016: a systematic analysis for the global burden of disease study 2016. *The Lancet Neurology*, 18:439–458, 5 2019.

[7] Guolin Ke, Qi Meng, Thomas Finely, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NIP 2017)*, December 2017.

[8] Ferath Kherif and Adeliya Latypova. Principal component analysis. pages 209–225. Elsevier, 1 2019.

[9] Ajitesh Kumar. Pca explained variance concepts with python example, 2022.

[10] Sang Gyu Kwak and Jong Hae Kim. Central limit theorem: The cornerstone of modern statistics. *Korean Journal of Anesthesiology*, 70:144–156, 4 2017.

[11] Susan M. Linder, Anson B. Rosenfeldt, R. Curtis Bay, Komal Sahu, Steven L. Wolf, and Jay L. Alberts. Improving quality of life and depression after stroke through telerehabilitation. *American Journal of Occupational Therapy*, 69, 3 2015.

[12] Pablo MacEira-Elvira, Traian Popa, Anne Christine Schmid, and Friedhelm C. Hummel. Wearable technology in stroke rehabilitation: Towards improved diagnosis and treatment of upper-limb motor impairment. *Journal of NeuroEngineering and Rehabilitation*, 16, 11 2019.

[13] Jan Mehrholz, Marcus Pohl, Thomas Platz, Joachim Kugler, and Bernhard Elsner. Electromechanical and robot-assisted arm training for improving activities of daily living, arm function, and arm muscle strength after stroke, 9 2018.

[14] microsoft. Flaml. `https://github.com/microsoft/FLAML`, 2022.

[15] Syeda Noor, Zehra Naqvi, Sofia Yfantidou, Esteban Zimányi, and Zim ́ Zimányi. Time series databases and influxdb, 2017.

[16] Matthew A Reyna, Chris Josef, Salman Seyedi, Russell Jeter, Supreeth P Shashikumar, M Brandon Westover, Ashish Sharma, Shamim Nemati, and Gari D Clifford. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. In *2019 Computing in Cardiology (CinC)*, pages Page 1–Page 4, 2019.

[17] Robert E Schapire. The strength of weak learnability, 1990.

[18] Joeky T. Senders, Patrick C. Staples, Aditya V. Karhade, Mark M. Zaki, William B. Gormley, Marike L.D. Broekman, Timothy R. Smith, and Omar Arnaout. Machine learning and neurosurgical outcome prediction: A systematic review, 1 2018.

[19] Yong Shin and Cheong Park. Analysis of correlation based dimension reduction methods. *Applied Mathematics and Computer Science*, 21:549–558, 09 2011.

[20] Cheng Tang, Di Wang, Ah-Hwee Tan, and Chunyan Miao. Eeg-based emotion recogni-

tion via fast and robust feature smoothing. pages 83–92, 11 2017.

[21] Steven L. Wolf, Komal Sahu, R. Curtis Bay, Sharon Buchanan, Aimee Reiss, Susan Linder, Anson Rosenfeldt, and Jay Alberts. The haapi (home arm assistance progression initiative) trial: A novel robotics delivery approach in stroke rehabilitation. *Neurorehabilitation and Neural Repair*, 29:958–968, 11 2015.

[22] Curt Bay David Wu. Expanding tele-rehabilitation of stroke through in-home robot-assisted therapy. *International Journal of Physical Medicine Rehabilitation*, 02, 2013.