School of Chemistry, Chemical Engineering and Biotechnology

Nanyang Technological University Singapore

BG4104: Machine Learning and Optimisation for Bioengineers

Group 4 Final Report

No.	Name of Members	Matriculation No.
1	Ashwariya Ganeshan	U2121435E
2	Bryan Chang Ding Lun	U2121878B
3	Hairul Nafiz Bin Mohammad Zaid	U2122040H
4	Lian Michelle Andrea	U2123782H
5	Muhammad Haziq Bin Razeli	U2122196G
6	Muhammad Taufiq Bin Mohammad Zainuddin	U2122793E
7	Nurul Khadijah Binte Sanusi **(Team Leader)**	U2121649E
8	Phua Hui Ying	U2121456C
9	Yeo Jie Xavier	U2121319J

Dateline: 22 November 2024, 23:59

Predicting Likelihood of Musculoskeletal Disease in Elderly with Machine Learning

- Done by Group 4 -

***Abstract*—This report explores how the implementation of machine learning detects and mitigates the prevalence of musculoskeletal disease amongst the elderly, through hand grip strength and other relevant biomarkers. After thorough experimentation and analysis, we concluded that our chosen model, k-NN, holds promise for integration into our GRIPNOSIS application - In hopes to overcome our problem statement.**

I. Introduction

As we grow older, our muscles, bones and joints naturally deteriorate in turn increasing the risks of musculoskeletal disease (MSD) that can potentially cause temporary to lifelong limitations in everyday functioning [1], [2].

The problem arises as many remain under diagnosed or detected at later stages globally, when functional impairment is more pronounced [3]. With that being said, how can we increase the detection of potential musculoskeletal disease in the elderly population through the use of supervised learning? We aim to utilize handgrip strength as a marker for musculoskeletal disease detection in elderly through the usage of a handheld dynamometer and GRIPNOSIS, our machine learning application which can be further referred to in Appendix 1 and 2.

Handgrip strength has been found to be a simple, reliable and cost-effective biomarker for the assessment of musculoskeletal health and related functions. Acknowledging handgrip strength as a proxy for overall muscle strength, we can then make deductions on one's musculoskeletal health [4]. Reduction in muscle strength, which can be objectively measured through handgrip strength, often parallels the development of musculoskeletal conditions such as osteoarthritis and sarcopenia.

Given the widespread nature of MSD, there is an increasing need for accessible, cost-effective and reliable methods for early detection which is critical to reduce long-term effects of MSD on mobility, functional ability and overall health [5]. With this, we hypothesize that handgrip strength when combined together with common biomarkers such as age, sex and Body Mass Index (BMI) can serve as a reliable predictor of musculoskeletal disease in elderly people using machine learning models.

II. Overview of Dataset

Our dataset originates from a study conducted by Universiti Kebangsaan Malaysia, focusing on the factors influencing handgrip strength among elderly Malaysians especially in relation to non-communicable diseases [6]. The dataset comprises 1204 participants aged between 60 and 91 years, capturing various demographic, health and lifestyle factors such as locality, age, sex, marital status, education level, employment status, blood pressure (SBP and DBP), BMI, handgrip strength, cognitive assessment scores and a range of disease statuses. For our analysis, we narrowed down the key features to Age, Sex (0 for Male, 1 for Female), BMI, Grip Strength and MSD status (0 for No, 1 for Yes).

Exploratory Data Analysis revealed a clean dataset with no missing values, duplicates

or anomalous entries. However, a significant challenge identified was the class imbalance in the target variable - MSD. Out of the 1204 samples, 84.4% (1017 cases) were negative for MSD while only 15.6% (187 cases) were positive. This imbalance poses a risk of biased model predictions towards the majority class, potentially overlooking cases of MSD. To address this, we applied the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic samples for the minority class by interpolating between existing positive cases and their nearest neighbors. This approach helps to balance the class distribution, thereby improving the model's sensitivity and ability to detect positive MSD cases [7]. In our data, we created synthetic data until there were an equal number of positive and negative cases of MSD.

III. Methodology

Moving forward, the methodology behind our approach involves a comparative analysis between three learning algorithm models: 1) FeedForward Neural Network (FFNN), 2) Support Vector Machines (SVM) and 3) k-Nearest Neighbors (k-NN). This will aid in determining the most effective model for predicting the likelihood of getting MSD.

Introducing the three models - Starting with FFNN, it mimics the working principle of the biological brain, utilizing artificial neurons over three main layers which includes input, hidden and output layers. The hidden layers contribute weights and biases to incoming data while activation functions contribute non-linearity, allowing the neural network to learn complicated patterns. Here, we implemented FFNN which processes data in one direction. Although it is considered a basic form of neural network, we believe that it was still effective in helping us achieve our goal [8]. Next, SVM identifies the optimal hyperplane that maximizes margin between classes. Here, we implemented both linear and non-linear kernel functions, Radial Basis Function (RBF) and sigmoid, for basic data separability and complicated transformations, respectively [9]. This was done to consider and comprehend which classifier was necessary in handling our data better. Finally, k-NN utilizes the closeness of data points for classification, allocating a data point to the class most common among its k closest neighbors. We considered various values for k, weights and metrics to find balance for stability and sensitivity [10].

Tuning hyperparameters for each model were also critical for optimization. "GridSearchCV" allowed an exhaustive search over a range of values on the hyperparameters to obtain the best base model. Hyperparameters for FFNN mainly include size of hidden layers, activation functions, learning rate, solver type, maximum iterations and alpha for regularization. Hyperparameters for SVM mainly include gamma, coef0 and c for regularization. It is important to note that gamma is only used for RBF and sigmoid while coef0 is only used for sigmoid. Hyperparameters for k-NN mainly include weights, metrics and n_neighbors for regularization.

In evaluating the models, we focused on recall for circumstances on misinterpreting a positive case as it maximizes true positives. On top of recall, we also considered F1-Score to balance accuracy, precision and recall. Outputs of the performance metrics and visualization plots were generated. These aid in identifying which metric is the most relevant and providing a comprehensive display of how well the model performs, at the same time pinpointing potential misclassifications.

Another important pointer to note is our dataset splitting. Our dataset was split into training and testing sets with a ratio of 80-20 to assess the model performance. Additionally, we experimented with different ratios like 70-30 and 90-10 to investigate its impact on the model performance. We implemented a

5 fold cross-validation to ensure robustness and minimize overfitting.

IV. Results and Analysis

FFNN

Learning curves help show how the change of different hyperparameters affects the performance of the FFNN model. The model then evaluates the overall value of the training error, testing error, and validation score to select the best-performing hyperparameter.

The learning curve of the hidden layer configuration showed that double layers are generally better than single layers by having a lower training and testing error followed by a higher validation score. Among all the double-layer configurations, (180,90) is determined to be the best configuration by using "GridSearchCV". Increase in the layers and neurons lower the errors, suggesting that the model fits better when the neurons increase. However, it is important to note that more layers or neurons also require more computational cost and time.

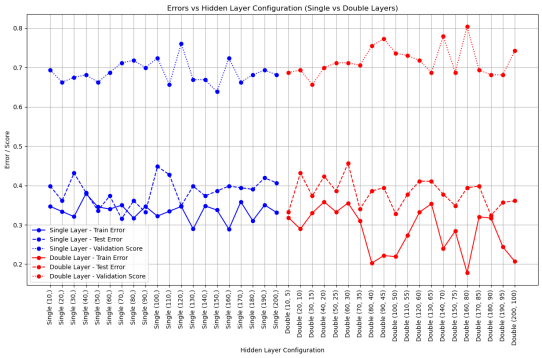


Fig. 1. Learning Curve for Hidden Layer Configuration

The learning curve of the activation functions showed that "ReLU" is more suitable than "Tanh" for this model as it has a lower training error and higher validation score. However, the difference between the two activations is not significant, suggesting that both functions can be applied in this model with "ReLU" demonstrating a relatively better performance.

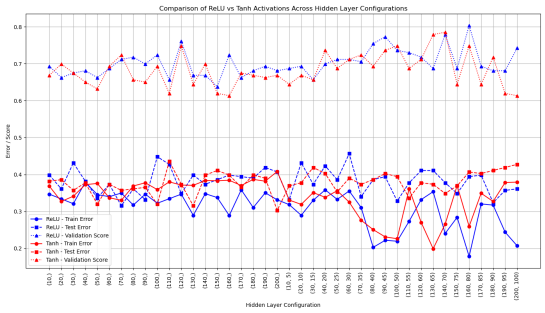


Fig. 2. Learning Curve for Activation Functions

The learning rate curve demonstrates that the optimum learning rate for this model is 0.01. Learning rate below 0.01 showed increase in training and testing error while above 0.01 showed decrease in training error but increase in testing error. As a low learning rate will result in a long processing time, but a high learning rate will cause underfitting.

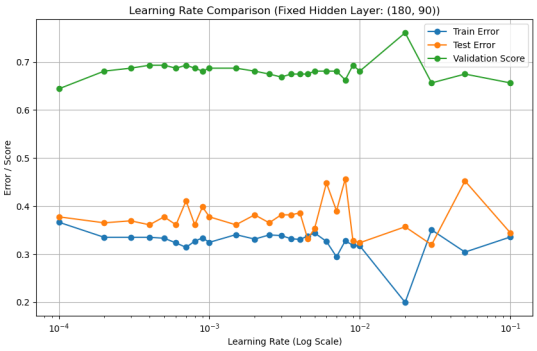


Fig. 3. Learning Curve for Learning Rate

Alpha value is the regularization parameter that prevents the model from overfitting. Using "GridSearchCV", alpha value=0.005 was determined to be the optimal value.

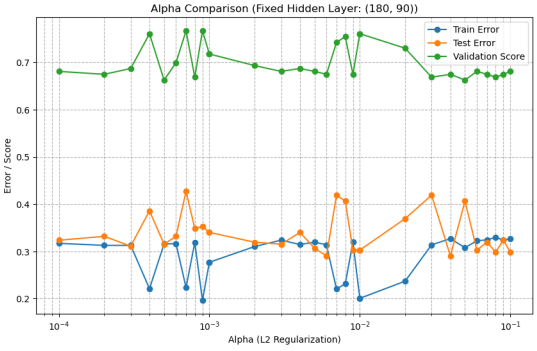


Fig. 4. Learning Curve for Alpha

According to the learning curves of each

hyperparameter analyzed above, the best performing hyperparameters were selected by the model and used to predict the output for the musculoskeletal disease.

SVM

Linear, RBF and sigmoid kernels were employed to determine which performed better with three hyperparameters utilized to optimize each model's performance accordingly. C regulates how much misclassification in the training dataset is acceptable. Gamma regulates how flexible the model is and the extent of overfitting or underfitting. Coef0, applicable only to the sigmoid kernel, adjusts the influence of each individual data point. The learning curves provides a better understanding on how the training error rates, testing error rates and validation score gets affected from the hyperparameters established on the different kernels we explored.

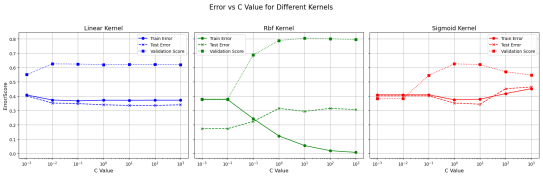


Fig. 5. Learning Curve for C of All Kernels

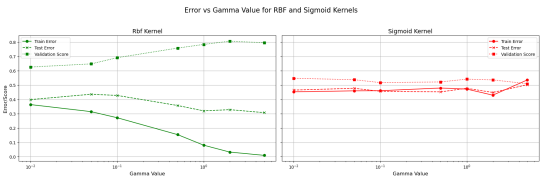


Fig. 6. Learning Curve for Gamma of SVM RBF and Sigmoid

Using "GridSearchCV", C=0.1 was determined to be the optimal value for SVM linear. Although C=0.1 has the highest testing error, it also has the lowest training error and high validation score which in turn provides a balance due to the good generalization and lower likelihood of overfitting.

Using GridSearchCV, the C and gamma values of 10 and 2 respectively were indicated to be optimal for RBF. C=10 shows the highest validation score and a training error that is low but not indicative of overfitting, offering good generalization that balances between underfitting and overfitting. This can also be seen in the gamma value curve. This means that the model is able to fit the training data well and generalizes well to unseen data without overfitting.

However, for the sigmoid curve, C and gamma values seem to have no impact on any of the factors, with the exception of C=0.01 which has a large decrease. This could indicate a degree of underfitting at C=0.01 and an increase in model performance at C=0.1.

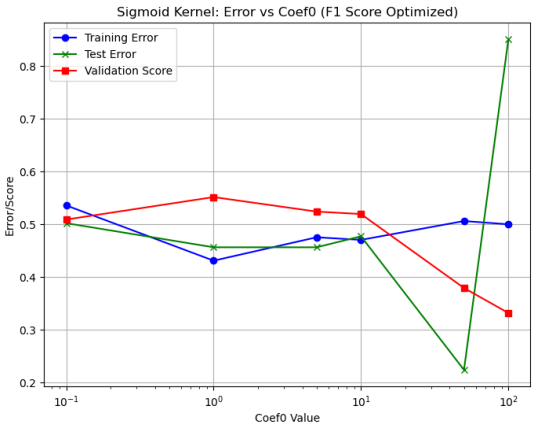


Fig. 7. Learning Curve for Coef0 (SVM Sigmoid)

Non-linear SVM is typically preferred over linear SVM as it is able to handle more complex and non linear relationships in real world data. The additional gamma and coef0 parameters in the RBF and sigmoid SVM provides a more detailed and flexible analysis in optimizing model performance. However, the sigmoid performs better for our particular dataset.

k-NN

The learning curves below demonstrate how training and testing error rates and validation accuracy fluctuate for a KNN model with different hyperparameter values - the number of neighbors, k.

The training and testing error rates were plotted against the training dataset's size for

varying values of k. The training sizes were sampled at different points and error rates were averaged using cross-validation. Additionally, the optimal number of neighbors was determined by plotting validation accuracy against various k values.

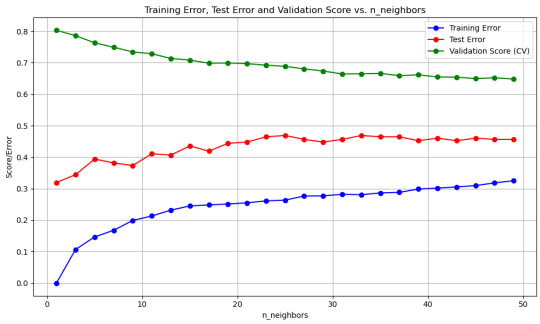


Fig. 8. Learning Curve for k

Referring to Figure 8 above, for smaller k values (k=1), the training error is nearly zero which indicates that there is overfitting. This is because the model memorizes the training data instead of generalizing well. As the k value increases, the training error rises which indicates that there is less model flexibility but greater generalization.

As for the testing errors, they increase with smaller or larger k values which suggests that overfitting at small k and underfitting at very large k. Furthermore, optimal k values (k=3 or 5) balance errors and decrease testing errors.

Validation accuracy peaks at intermediate k values (k=5) and decreases for smaller and larger k values, demonstrating the trade-off between model complexity and generalization. Overfitting occurs at smaller k values where the training error is low but the testing error is high. Meanwhile, underfitting occurs at large k values, where both training and testing errors are high. For intermediate k values, increasing training data size leads to convergence of testing and training errors, indicating that more data no longer significantly improves model performance.

With the increase in the training size, the training error increases as the model concentrates on generalization, with distance weighting showing a slightly slower rise than uniform weighting. This indicates that the model may use new data more efficiently. On the other hand, test error decreases as training size increases, stabilizing at lower levels for distance weighting than uniform weighting indicating greater generalization. Validation scores first rise with more training data but plateau or slightly drop as new data provides diminishing returns. Distance weighting obtains slightly higher validation scores at convergence, which suggests that it is more appropriate for this dataset.

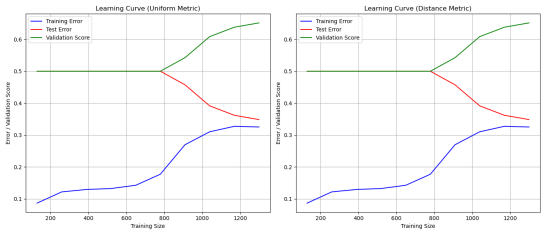


Fig. 9. Learning Curve for k

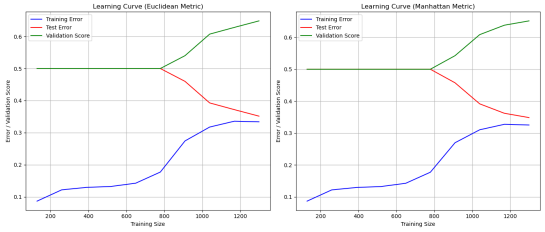


Fig. 10. Learning Curve for k

As the training size increases, the training error increases as the model's capability of memorizing the data declines and stabilizes at a comparable level for both metrics. The Euclidean metric has a more gradual increase in training error than the Manhattan metric, which indicates that it performs better with larger training datasets. The test error decreases as the training size increases, which suggests an increase in generalization. Then, it stabilizes at a specific point, showing that the model has learn as much as possible from the data. At convergence, the Manhattan metric has a slightly higher test error indicating that it generalizes less effectively

than the Euclidean metric. Validation scores first rise with training size, but then level or slightly drop as more data provides diminishing benefits. The Euclidean metric has slightly higher validation scores at convergence, indicating more substantial generalization for this dataset.

Overall Model Performance

Method	Training Ratio	Accuracy	Precision	Recall	F1 Score
FFNN	90-10	0.653	0.810	0.653	0.709
SVM (Linear)		0.686	0.806	0.686	0.732
SVM (RBF)		0.653	0.799	0.653	0.708
k-NN		0.655	0.790	0.661	0.712
FFNN	80-20	0.672	0.755	0.672	0.707
SVM (Linear)		0.651	0.780	0.651	0.697
SVM (RBF)		0.585	0.753	0.585	0.643
k-NN		0.680	0.757	0.680	0.713
FFNN	70-30	0.497	0.778	0.497	0.571
SVM (Linear)		0.663	0.793	0.663	0.711
SVM (RBF)		0.586	0.755	0.586	0.648
k-NN		0.655	0.771	0.655	0.701

Fig. 11. Comparison of Performance Metrics for All Models with Experimented Dataset Splitting Ratios

Figure 9 above illustrates a table with summarized scores for the three different models used. Three different ratios were experimented on to study how it affects the models. The green values highlighted represent the best performing scores, whereas the red values represent the worst performing scores.

With reference to the table in Figure 9, we can see that SVM RBF is the worst performing model to analyze the data for the 80-20 dataset splitting ratio, as the scores are consistently the lowest among the 3 performance metrics used. Meanwhile, k-NN is the best performing model to analyze the data for the 80-20 dataset splitting ratio, as the scores are consistently the highest for 3 out of 4 performance metrics. As for 70-30 dataset splitting ratio, k-NN performs the best in recall and F1-Score, whereas FFNN performs the best in precision and SVM Linear performs best in accuracy. As for 90-10 dataset splitting ratio, FFNN performed the best in precision but SVM Linear performed the best in all the other performance metrics.

Overall, k-NN exhibits better performance than the other models when a 80-20 and 70-30 dataset splitting ratio is utilized. As the training ratio determines how much the model learns from the training set and how well the model performs from the testing set, a more balanced training ratio would be more desirable to reduce bias and inaccauracy when training the model, making it more robust. Moreover, as the dataset we are using is moderately large with a limited number of data points, the established ratio of 80 for training to 20 for testing provides balance and stability. However, if the data quantity increases, a 90 for training to 10 for testing can be considered for very large data sets. However, at this stage, using 90-10 would make the model train too much but will not be able to accurately judge the performance, whereas a 70-30 ratio would incur too much testing without learning adequately.

V. Discussion

FFNN

The advantages of using FFNN as a model for the prediction of musculoskeletal disease are the flexibility and scalability. As mention above, there are various hyperparameters that can be tuned to fit the dataset, where different dataset have different features and FFNN can adjust flexibly to fit. In addition, FFNN is able to handle large and complex dataset with the scalable neurons in the hidden layer, where the more complex the data the more neurons are required.

On the other hand, there are also some limitations for using FFNN. It is relatively complex to build as it have lots of hyperparameters to tune, and each hyperparameter have impact on the overall performance. Moreover, the training of the neural network model is also time-consuming as it need to train the model with different combinations of the hyperparameters.

SVM

There are several advantages to using SVM in our models for predicting muscu-

loskeletal disease based on grip strength. It helps prevent overfitting by maximising the margin between different classes, ensuring reliable predictions for new patients. SVM is also memory efficient, using only support vectors from the training data to construct the decision boundary. Additionally, its versatile kernel function handles both linear and non-linearly data, allowing it to capture complex relationships and improve classification accuracy.

However, there are some limitations in using SVM. It is not suitable for large datasets, as it can become slow and consume a lot of memory. For the current dataset we are working with, this is not a major concern. However, this could become a challenge if we plan to collect new patient data and expand the database.

k-NN: Our Chosen Model

The k-NN model's strength lies in its simple algorithm and ability to produce results with minimal effort, especially when using smaller datasets like the one used in this report. Both FFNN and SVM are more complicated in nature requiring deep hierarchical feature extractions and complex decision boundaries respectively. Moreover, the data used from the dataset is largely structured categorical data works well with the k-NN model that operates in the raw feature space without requiring extensive feature extraction or transformation. In comparison, FFNN is designed for images or spatial data while SVM may perform poorly if the data is not linearly separable or has overlapping classes. Both models require feature engineering to get meaningful data. Thus, these highlight the strength of the k-NN model.

However, the k-NN model still has weaknesses. Moving forward, given an exponential increase in the dataset used, the larger quantity of data will render k-NN less effective as the dataset becomes larger. Moreover, if the data becomes increasingly more complex (i.e. Incorporating more features from the dataset aside from age, sex, BMI and hand grip strength), k-NN may not be able to handle the higher dimensionality and complex relationships between the data. In this case, using other models may increase the performance of training and testing leading to better scores.

In our case, choosing to stick with the k-NN model is still the best option regardless. Aside from exhibiting the best performance among the three models at the 80-20 dataset splitting ratio, the dataset used in this report is not very large hence not requiring a complex model to train and test the data. Moreover, the data used is structured and low-dimensional thus synergizing well with the k-NN model.

VI. Conclusion

Future Improvements

With our findings and discussion thus far, we believe that there are still further room for improvement for the model. We can implement more extensive feature engineering. This involves the creation of new features or transforming the existing features into refined ones that may provide more meaningful data to the algorithm, improving the model's performance overall. The raw data such as age, sex and BMI are not mutually exclusive and are affected by one another in various different ways. As such, the models may be able to derive clearer relationships between these new transformed features. For example, instead of looking solely at hand grip strength, relative hand grip strength based on a patient's weight can be used, which would be the ratio of grip strength to BMI. A study indicates that BMI has a positive correlation with grip strength in the elderly [11]. In addition, our dataset includes the data columns involving lifestyle choices such as alcohol consumption, cigarette smoking and betel chewing. These factors have often been

associated with poorer health. The machine learning model may be able to derive useful insights from this data and could be used to further improve the model.

Another possibility would be to utilise all 3 models we have created in an ensemble. As discussed previously, each of these models have their strengths and limitations for classification problems. Relying on a single model would be unwise as it may not be able to capture the nuances in the data and lead to poor generalisation. Given that we have used 3 different models in this project, they can be used in an ensemble and even have more models added.

Through the use of ensemble learning, the model is able to leverage the strengths of the different models while covering for the limitations of each individual model as the models will be improved one after another based on the errors of the previous models. It would lead to a more accurate and robust model.

Bagging (Bootstrap Aggregating) and boosting are the most common and widely used ensemble methods in ML today. In Python, Bagging can be done using "RandomForestClassifier" while boosting is typically done using AdaBoost or XGBoost.

Practical Implications

In essence, our model has significant potential in diagnosing MSD by leveraging hand grip strength as a marker by incorporating age, sex, and BMI to create highly personalized predictive models. This allows ease of early detection, allowing for timely intervention before significant functional decline occurs. Partnerships with JAMAR electronic hand held dynamometer, as well as a subscription plan to fund the start and running of our GRIPNOSIS application allows our solution to be widely deployed across the elderly care homes in Malaysia, given its projected population growth.

References

[1] CERN, "Ergonomics — HSE Unit at CERN," CERN Accelerating Science . https://hse.cern/content/ergonomics (accessed Nov. 20, 2024).

[2] F. D. Brodkey and D. C. Dugdale, "Aging changes in the bones - muscles - joints," MedlinePlus Medical Encyclopedia, Jul. 21, 2022. https://medlineplus.gov/ency/article/004015.htm (accessed Nov. 20, 2024).

[3] C. Chen et al., "Global years lived with disability for musculoskeletal disorders in adults 70 years and older from 1990 to 2019, and projections to 2040," Heliyon, vol. 10, no. 15, Jul. 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e35026.

[4] R. Vaishya, A. Misra, A. Vaish, N. Ursino, and R. D'Ambrosi, "Hand grip strength as a proposed new vital sign of health: A narrative review of evidences," Journal of Health, Population and Nutrition, vol. 43, no. 1, Jan. 2024, doi: https://doi.org/10.1186/s41043-024-00500-y.

[5] World Health Organization, "Musculoskeletal health," World Health Organization, Jul. 14, 2022. https://www.who.int/news-room/fact-sheets/detail/musculoskeletal-conditions (accessed Nov. 20, 2024).

[6] Z. Mohammad and S. A. Shah, "HGS and NCDs Elderly Malaysia," Mendeley Data, vol. 1, no. 1, Nov. 2020, doi: https://doi.org/10.17632/hsc4k7vtfp.1.

[7] C. Maklin, "Synthetic Minority Over-sampling TEchnique (SMOTE)," Medium, May 15, 2022. https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c (accessed Nov. 20, 2024).

[8] H. Singh, "Deep Learning 101: Beginners Guide to Neural Network," Analytics Vidhya, Jul. 27, 2023. https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/ (accessed Nov. 20, 2024).

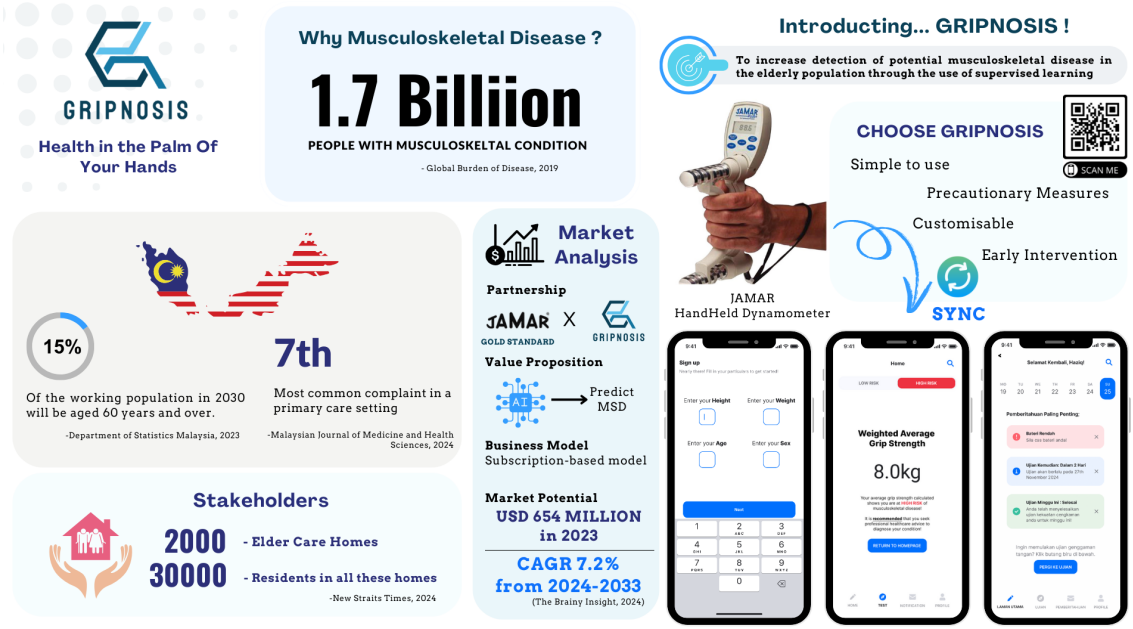
[9] IBM, "What are support vector machines (SVMs)?," IBM, Dec. 27, 2023. https://www.ibm.com/topics/support-vector-machine (accessed Nov. 20, 2024).

[10] P. Huilgol, "Precision and Recall in Machine Learning," Analytics Vidhya, Nov. 18, 2024. https://www.analyticsvidhya.com/articles/precision-and-recall-in-machine-learning/ (accessed Nov. 20, 2024).

[11] N. Soraya and E. Parwanto, "The Controversial Relationship between Body Mass Index and Handgrip Strength in the Elderly: An Overview," Malaysian Journal of Medical Sciences, vol. 30, no. 3, pp. 73–83, Jun. 2023, doi: https://doi.org/10.21315/mjms2023.30.3.6.

Appendix 1 - Pitch

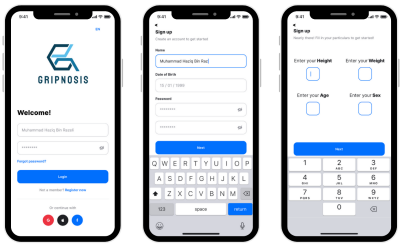
Pitch *Deck*



Appendix 2 - GRIPNOSIS Application Interface

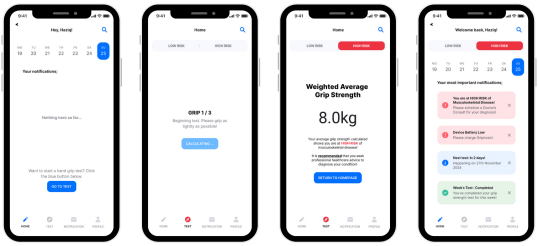
Pitch *Deck*

LOG IN / SIGN UP PAGE



Pitch *Deck*

HAND GRIP TEST



Pitch *Deck*

PROFILE AND LANGUAGE SETTINGS