# National Poll on Healthy Aging NPHA

Emmanuel Opoku Hinneh
*Department of Data Science*
*Carolina University*
Winston Salem, NC, USA
hinnehe@carolinau.edu

***Abstract*—This paper explores how different machine learning algorithms can be compared using cross-validation techniques. The paper also demonstrates how a machine learning algorithm can be chosen for a problem, and how a better algorithm can be determined from a group of algorithms. The explored models are SVM, Random Forest, and MLP, and the better algorithm for the problem was SVM.**

***Keywords—npha, aarp, hyperparameterization, k-means, svm, random forest, mlp, k-fold, loocv.***

## I. Introduction

The American Association of Retired Persons (AARP), which focuses on issues affecting those over the age of fifty, and Michigan Medicine, which is the University of Michigan's academic medical center, funded the creation of a dataset that intended to gather insights on health, healthcare, and health policy issues affecting Americans ages 50 or older [1]. The University of Michigan aims to educate the public, healthcare practitioners, politicians and advocates about the various aging elements by emphasizing the views of older persons and their caregivers. The senior population's health-related demands and concerns are comprehensively understood by covering health insurance, household composition, sleep disorders, dental care, prescription medications, and caregiving. The data was stripped down to a version that could be shared online for public usage. The public data has been cleaned up and consists of 14 features that are related to health and sleep. This project aims to build a classifier using three different machine learning algorithms, compare their performance, and conduct hyperparameterization. Section II will discuss the EDA, section III will discuss the three chosen machine learning algorithms, section IV will compare the algorithms, and Section V will conclude.

## II. EDA

The initial data consists of 15 features and 714 data points. The features are: **Number_of_Doctors_Visited, Age, Physical_Health, Mental_Health, Dental_Health, Employment, Stress_Keeps_Patient_from_Sleeping, Medication_Keeps_Patient_from_Sleeping, Pain_Keeps_Patient_from_Sleeping, Bathroom_Needs_Keeps_Patient_from_Sleeping, Uknown_Keeps_Patient_from_Sleeping, Trouble_Sleeping, Prescription_Sleep_Medication, Race, and Gender**. Fig 1 and Fig 2 show the bar chart and histogram of the features respectively.

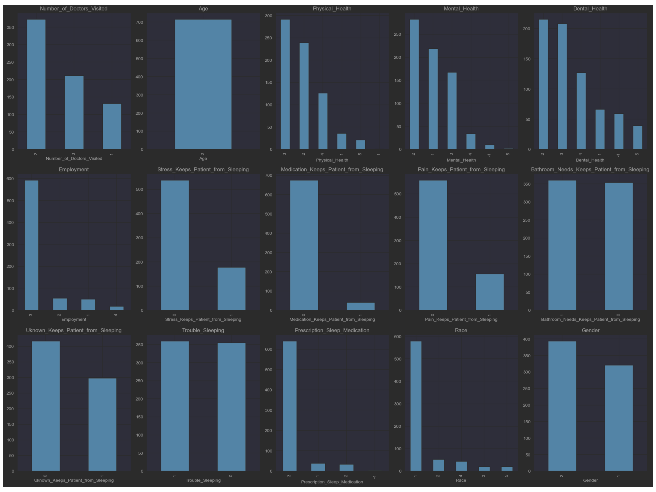

*Fig. 1. Bar chart of features.*

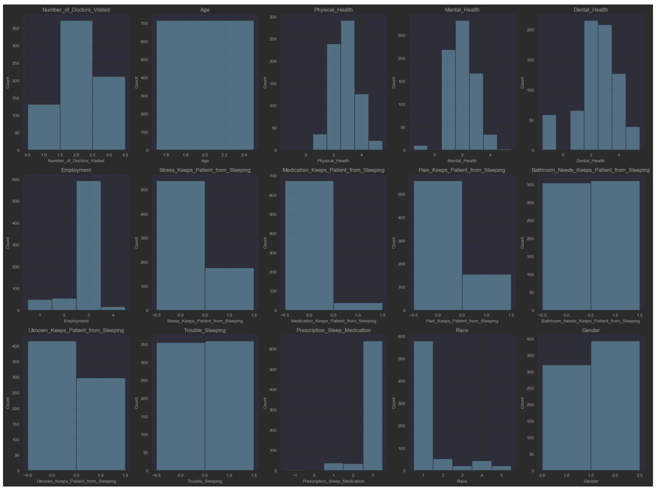

*Fig. 2. Histogram of features.*

In the original data, the target variable is supposed to be " **Number_of_Doctors_Visited",** but a new feature was created using KMeans Clustering. The new feature is "**Risk_Level**" and it consists of three categories; Low, Medium, and High. Fig 3 shows the distribution of the "Risk_Level" feature.

DSC-635-10/24/$00.00 ©2024 IEEE

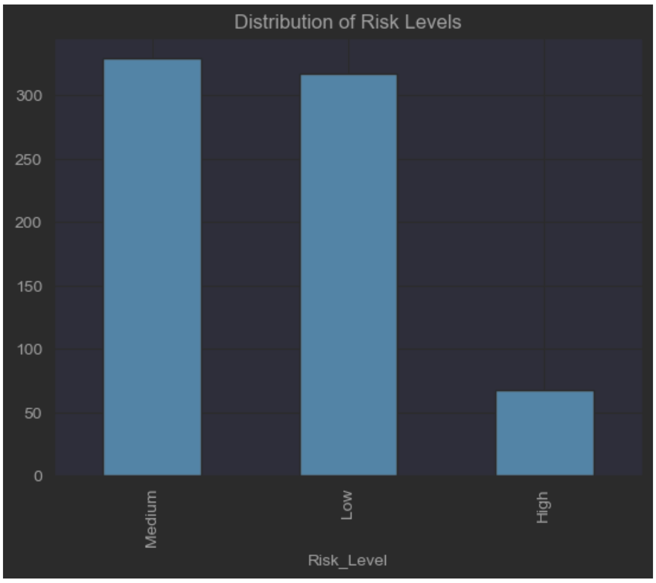

*Fig. 3. Distribution of the Risk_Level feature.*

In K-Means, the data is partitioned into disjoint sets $C_1, \dots, C_k$ where each $C_i$ is represented by a centroid $\mu_i$. The k-means objective function measures the squared distance between each point in X and its cluster's centroid. The centroid of Ci is defined as

$$\mu_i(C_i) = arg\min_{\mu \in X'} \sum_{x \in C_i} d(x,\mu)^2.$$

Then the k-means objective is

$$G_{k-means}\big((X,d),(C_1,\dots,C_k)\big) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x,\mu_i(C_i))^2 \ .$$

## III. Model selection

The model selection is based on the fact that we're trying to train a classifier for predicting if a patient can be classified as high, medium, or low risk. The first model of choice is SVM, the second is a Random Forest Classifier, and the last is a Neural Network with a multi-layer perceptron.

### *A. SVM*

Support Vector Machine helps in sorting data into two or more categories with the help of a boundary to differentiate similar categories. Given some feature vector X in the space of dimension D; $X \in R^D$, we should be able to map it to a complex feature space. This can be represented mathematically as

$$\Phi(x) : \ R^D \to R^M.$$

The equation of the primary separator line is called a hyperplane equation, which can be represented mathematically as

$$H: w^T(x) + b = 0.$$

The distance for each hyperplane in the complex space can be calculated as

$$d_H\big(\phi(x_0)\big) = \frac{\left|w^T\big(\phi(x_0)\big) + b\right|}{\|w\|2}$$

where $\|w\|2$ is the Euclidean norm of the length of w.

After training the classifier, these are the observations:

- Looking at the classification report, the SVM model performed well with an accuracy of 0.94.
- The proportion of true positive predictions of high-risk patients is 1, that of low-risk is 0.94, and that of medium-risk patients is 0.94.
- The proportion of actual positive cases correctly identified is 0.86 for high risk, 0.95 for low risk, and 0.95 for medium risk.
- When we combine the precision and recall, we get the F1-score. The F1-score for high risk is 0.92, low risk is 0.94, and medium risk is 0.95.

The model seems to perform well, with an accuracy of 92%.

### *B. Random Forest Classifier*

Random forest is a machine learning algorithm that combines the output of multiple decision trees to reach a single result. A majority vote over the individual trees' predictions produces the random forest's prediction. The main problem with decision trees is that they are computationally complex to learn; therefore, we described several heuristic procedures for training them.

After training the classifier, these are the observations:

- The Random Forest model performed better than the SVM model with an accuracy of 0.92.
- The proportion of true positive predictions of high-risk patients is 1, that of low-risk is 0.92, and that of medium-risk patients is 0.90.
- The proportion of actual positive cases correctly identified is 0.86 for high risk, 0.89 for low risk, and 0.95 for medium risk.
- The F1-score for high risk is 0.92, low risk is 0.90, and medium risk is 0.93.

The Random Forest model performs worse than the SVM model, with an accuracy of 92%.

### *C. Neural Network with a Multi-Layer Perceptron*

The idea behind neural networks is that many neurons can be joined together by communication links to carry out complex computations. A fully connected multi-layer neural network is called a Multilayer Perceptron. The number of layers and neurons are referred to as hyperparameters of the neural network. We chose to use the "MLPClassifier", a feedforward neural network from Scikit Learn, which trains iteratively since, at each time step, the partial derivatives of the loss function to the model parameters are computed to update the parameters. The activation function chosen is the ReLU activation function which can be represented as:

$$R(z) = \max\,(0, Z)$$

This makes the output fall between the range of zero and infinity. After training our model, these were the observations:

- The MLP model performed better than the Random Forest model with an accuracy of 0.97.

- The proportion of true positive predictions of high-risk patients is 0.91, that of low-risk patients is 1.00, and that of medium-risk patients is 0.96.
- The proportion of actual positive cases correctly identified is 0.93 for high risk, 0.97 for low risk, and 0.98 for medium risk.
- The F1-score for high risk is 0.93, low risk is 0.98, and medium risk is 0.97.

The MLP model performs better than the Random Forest and SVM model, with an overall accuracy of 97%.

## IV. USING K-FOLD AND LOOCV TO ASSESS THE PERFORMANCE OF THE MODELS

Before training any model, the tradition is to split the data into training and testing sets. The splits can contribute to how bad or good the trained model is. In cross-validation, the dataset is not divided into training and testing sets once. Instead, the data is repeatedly partitioned into smaller groups, and the performance in each group is averaged. That way, the impact of partition randomness on the results can be reduced. Two of the many cross-validation techniques were tested in this project. They are k-fold cross-validation and leave-one-out cross-validation (LOOCV).

### *A. K-Fold Cross Validation.*

In k-fold, the dataset is first split into k equal-sized groups. Next, the train-test splitting is repeated k times, using the remaining k-1 subsets as a training set and one of the k subsets as a test set each time. Lastly, the average scores over the k-trials are calculated to estimate the model's performance. In our test, the k was set to 5, and Fig 4 shows a graph of the cross-validation scores of the SVM, Random Forest, and MLP models.

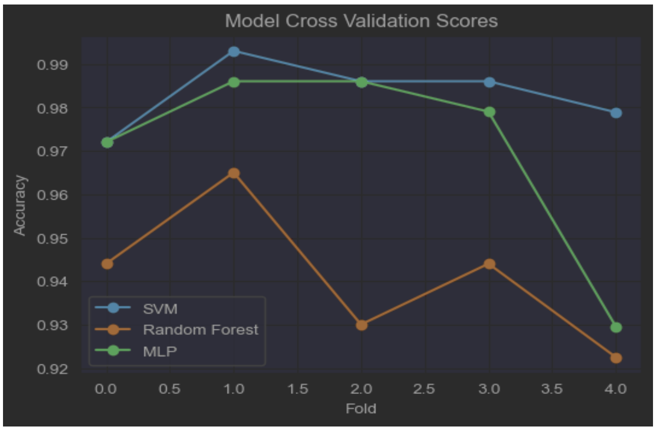

*Fig. 4.CV scores of the models.*

The mean and variance values of the scores are shown in Fig. 5.

| Model | CV mean | CV Variance |
|---|---|---|
| SVM | 0.98 | 5.11e-05 |
| Random Forest | 0.94 | 0.20e-04 |
| MLP | 0.97 | 0.45e-0.4 |

*Fig. 5. Mean and Variance from k-means CV.*

Based on these results, the SVM model performs better than the Random Forest and MLP model, with a mean accuracy of 0.98. The SVM model also has the most negligible variance in its scores compared to the Random Forest and MLP models. This indicates that the SVM model is more stable.

### *B. Leave-One-Out Cross Validation*

In LOOCV, the model is trained n times, where n is the size of the dataset. LOO can be considered as an extreme k-means scenario where k=n. LOOCV can be computationally expensive, mainly if the data size is large. The other problem with LOOCV is that it can be subject to high variance or overfitting as we feed the model almost all the training data to learn and just a single observation to evaluate. Fig 6 shows the mean and variance scores of the various models after performing LOOCV.

| Model | CV mean | CV variance |
|---|---|---|
| SVM | 0.983 | 0.017 |
| Random Forest | 0.962 | 0.036 |
| MLP | 0.980 | 0.019 |

*Fig. 6. Mean and Variance scores from LOOCV*

In the LOOCV, the SVM model proved to be better among the three models. It has the highest mean value of 0.983 and the least variance value of 0.019.

## V. CONCLUSION

Choosing a suitable machine learning algorithm for a project is essential in obtaining a good model. Sometimes, the best way to know the best algorithm is to try different ones, evaluate the results, and use the metrics to decide. Some algorithms are more complex than others, but their complexity does not guarantee they will be the best solution for the job at hand. This was proven by the results of the models we built in this project. Although the MLP is a much more complex algorithm compared to the SVM, the SVM proved to be a better fit for our classification problem. It is also worth knowing that every algorithm has limitations. Although SVM was a better algorithm for our problem, it wouldn't work well if our dataset was large.

To sum up, if you are confident in the hypothesis class and would need high-confidence guarantees on the accuracy of the acquired hypothesis, go with PAC learning. When the hypothesis class is unknown, and you want a more adaptable and durable framework to manage possible discrepancies between the selected class and the actual target idea, go with Agnostic PAC learning. Hyperparameterization should also be considered when comparing multiple algorithms.

## REFERENCES

[1] "UCI Machine Learning Repository." [Online]. Available: https://archive.ics.uci.edu/dataset/936/national+poll+on+healthy+aging+(npha)