CYO_Project:Capstone - PH125.9x HarvardX Data Science

Adolfo Muñoz Macho @dradolfomunoz

2024-06-07

Parkinson's Disease Detection Using Voice Measurements.

1. Introduction.

This report is a component of the HarvardX PH125.9x Data Science Capstone assignment, which is conducted on the CYO (Choose Your Own) platform. The Professional Certificate in Data Science at Harvard culminates in the Data Science Capstone.

1.1 Dataset Information.

This dataset comprises a variety of biomedical voice measurements from 31 individuals, 23 of whom have Parkinson's disease (PD). Each column in the table denotes a specific vocal measure, and each row corresponds to one of the 195 voice recordings from these individuals ("name" column). The primary objective of the data is to differentiate between individuals with PD and those who are healthy. This is achieved by setting the "status" column to 0 for healthy individuals and 1 for those with PD.

The data is stored in ASCII CSV format. Each cell of the CSV file contains an instance that corresponds to a single voice recording. The first column identifies the patient's name, and each patient has approximately six recordings.

The dataset is very complete and reliable and has no missing data.

1.2 Variables Information.

Matrix column entries (attributes):

- `name` - ASCII subject name and recording number
- `MDVP:Fo(Hz)` - Average vocal fundamental frequency
- `MDVP:Fhi(Hz)` - Maximum vocal fundamental frequency
- `MDVP:Flo(Hz)` - Minimum vocal fundamental frequency
- `MDVP:Jitter(%)`, `MDVP:Jitter(Abs)`, `MDVP:RAP`, `MDVP:PPQ`, `Jitter:DDP` - Several measures of variation in fundamental frequency
- `MDVP:Shimmer`, `MDVP:Shimmer(dB)`, `Shimmer:APQ3`, `Shimmer:APQ5`, `MDVP:APQ`, `Shimmer:DDA` - Several measures of variation in amplitude

- `NHR`, `HNR` - Two measures of ratio of noise to tonal components in the voice
- `status` - Health status of the subject (1 - Parkinson's, 0 - healthy)
- `RPDE`, `D2` - Two nonlinear dynamical complexity measures
- `DFA` - Signal fractal scaling exponent
- `spread1`, `spread2`, `PPE` - Three nonlinear measures of fundamental frequency variation

In this project, were developed various machine learning techniques to analyze whether it was accurately predict the health status of individuals based on their voice measurements. They were undertaken steps that include data cleaning, exploration, visualization, and the application of different machine learning models.

1.3 Objetives.

The project's objective is to determine whether machine learning techniques can be employed to forecast the health status of individuals (i.e., whether they are healthy or have Parkinson's disease) by analysing a variety of vocal measurements.

2. Methods and Analysis.

2.1 Downloading and Preprocessing the Dataset.

Initially, it was verified that all essential products had been installed and loaded. We attempt to dynamically install all missing packages by utilising if(!require) statements, and all file paths are relative.

The subsequent steps for acquiring and preprocessing the dataset were as follows:

The initial phase entailed the importation of the dataset and the execution of the requisite data cleaning procedures, including the management of missing values, the conversion of data types, and the verification of data consistency. Fortunately, the Parkinson's dataset was exceptionally clean, which facilitated the completion of our preprocessing tasks in a timely and straightforward manner. The following are the specific procedures that we implemented:

2.1.1 Download the Dataset. The URL of the Parkinson's dataset was obtained. We designated a location to store the extracted data file and the downloaded compressed file. In an effort to mitigate redundant downloads, we established whether the zip file was already in existence. If not, we utilised the download.file function to obtain the file. We verified the existence of the extracted data file. If not, we extracted it from the downloaded compressed file using unzip.

2.1.2 Load the Dataset. Using read.csv, we read the CSV file into a DataFrame.

2.1.3 Initial Inspection. To ensure that the dataset was read accurately, we analysed the initial recordings with head. We employed str to analyse the dataset's structure in order to understand the data types of each column. We confirmed that the dataset was free of any missing values.

2.1.4 Data Visualization. In order to gain an initial comprehension of the dataset, we presented a sample. We employed the kable function from the knitr package to generate a formatted table that displayed the first six rows and six columns of the dataset. This provides a brief overview of the data's structure and content.

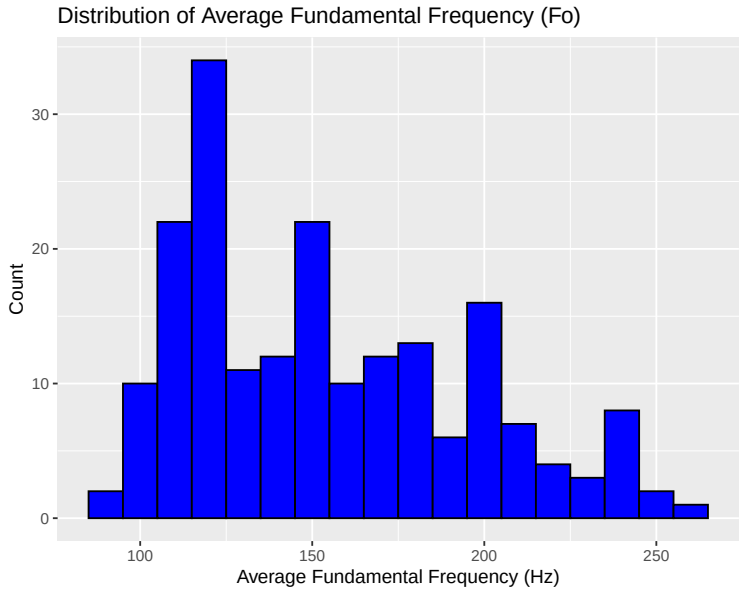
Table 1: Sample of the Dataset (6 rows and 6 columns)

name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)
phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007
phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008
phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009
phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009
phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011
phon_R01_S01_6	120.552	131.162	113.787	0.00968	0.00008

In order to elucidate the distribution of critical features in our Parkinson's dataset, we employ a variety of visualisation techniques, including histograms, box plots, density plots, scatter plots, violin plots, mean plots, treemaps, and line plots.

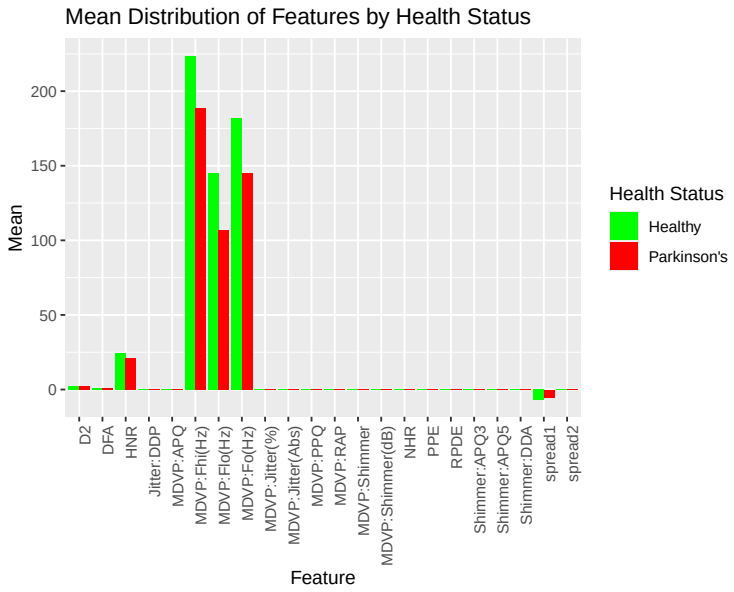
Histograms are graphical representations that illustrate the distribution of data elements within predetermined intervals (bins).

The "Average Fundamental Frequency" (MDVP:Fo(Hz)) was the column for which we elected to plot a histogram. This characteristic is crucial in the analysis of Parkinson's disease, as it denotes the average frequency of the patient's voice.



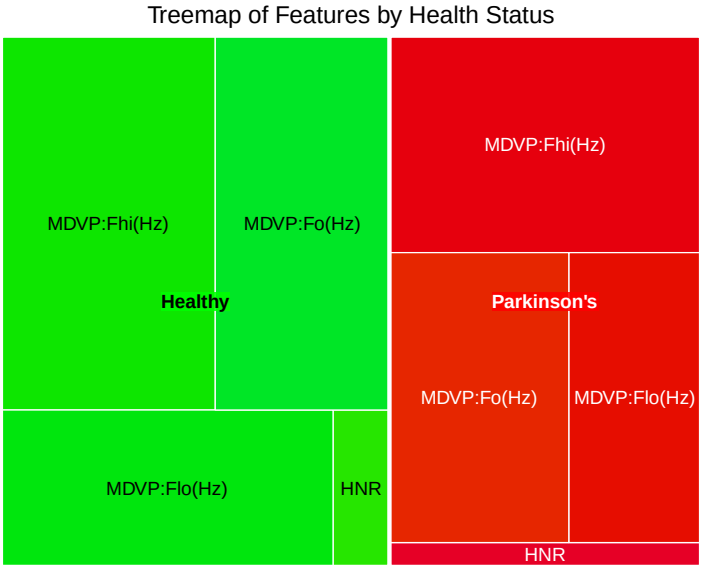
To gain insights into how different features vary based on health status (whether a person has Parkinson's disease or not), we visualize the mean values of these features grouped by their health status.

We calculate the mean values of various numerical features in the dataset, grouped by the health status (status), which indicates whether the person is healthy (0) or has Parkinson's disease (1). We then plot these mean values to compare the two groups visually.



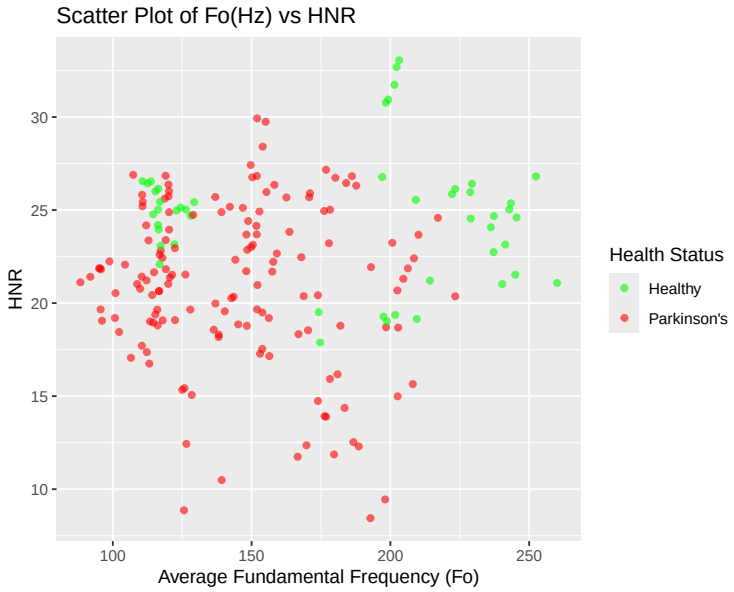
A treemap is a visual representation of hierarchical data that utilizes nested rectangles to facilitate the comparison of the distribution of values across different categories. In this context, we employ a treemap to compare the mean values of specific vocal features between healthy individuals and those with Parkinson's disease.

Our objective is to display the mean values of a subset of features (MDVP:Fo(Hz), MDVP:Fhi(Hz), MDVP:Flo(Hz), and HNR) grouped by health status.



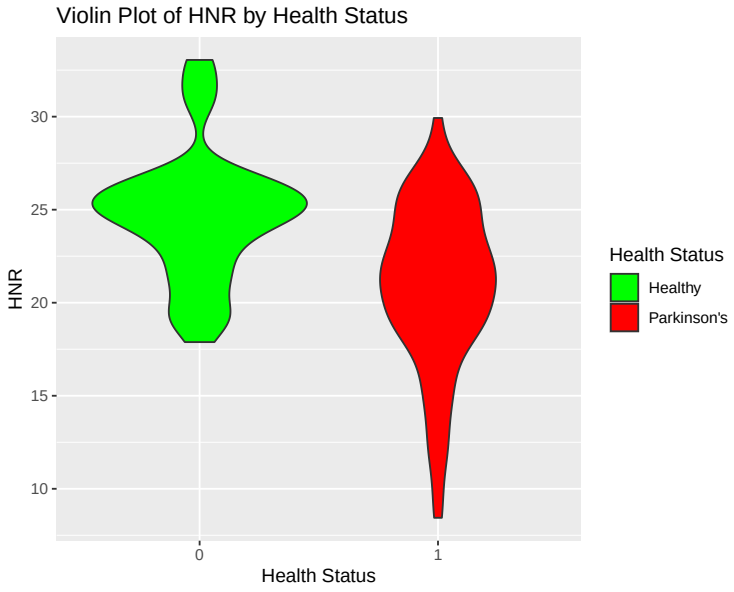
A scatter plot enables us to observe the relationship between two continuous variables and to observe how this relationship may vary among different groups. In this context, our objective is to compare the average fundamental frequency (MDVP:Fo(Hz)) with the harmonic-to-noise ratio (HNR) for healthy individuals and those with Parkinson's disease.

In order to illustrate the correlation between MDVP:Fo(Hz) and HNR, we generate a scatter diagram that colourizes data points according to their health status.



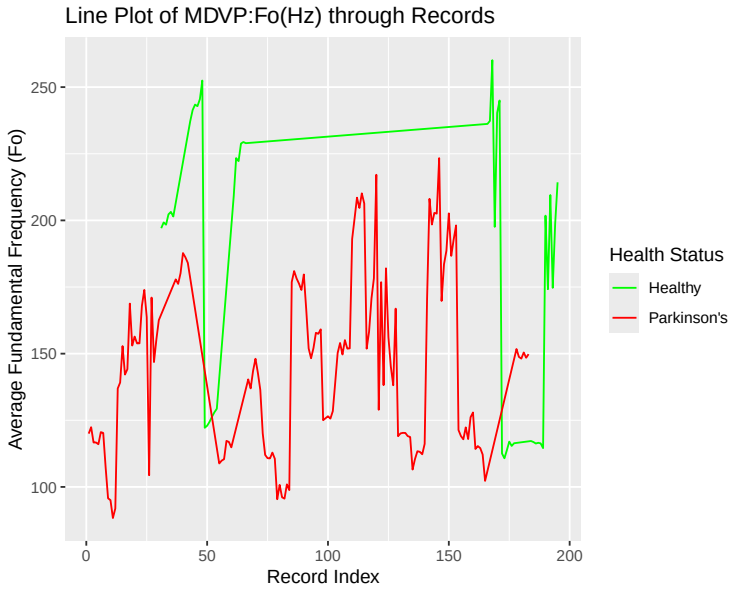
A violin plot combines aspects of a box plot and a density trace, providing a rich visualization of the distribution of a continuous variable across different categories. In this context, we use a violin plot to compare the distribution of the Harmonic to Noise Ratio (HNR) between healthy individuals and those with Parkinson's disease.

It was created a violin plot to visualize the distribution of HNR, grouped by health status.



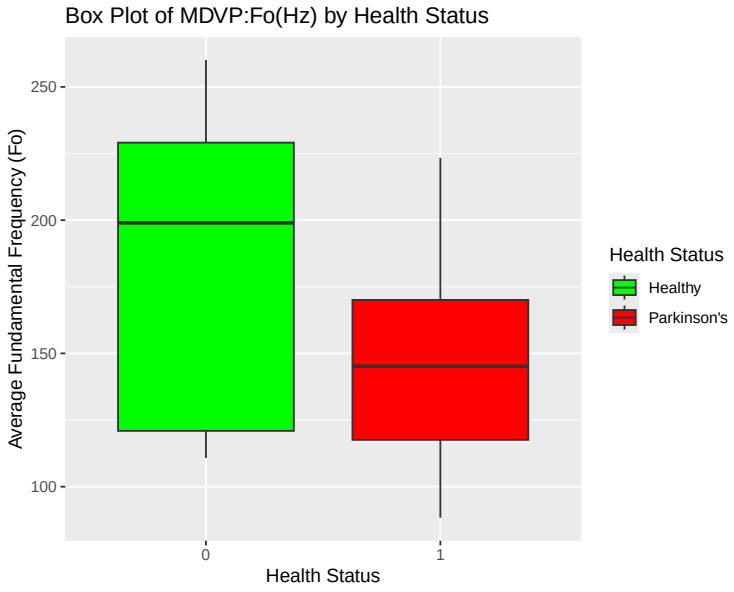
A line plot is an effective way to visualize trends and patterns over a sequence or time. In this context, we use a line plot to observe the trend of the Average Fundamental Frequency (MDVP:Fo(Hz)) through records, distinguishing between healthy individuals and those with Parkinson's disease.

It was created a line plot to visualize the trend of MDVP:Fo(Hz) over the sequence of records, with data points colored by health status.



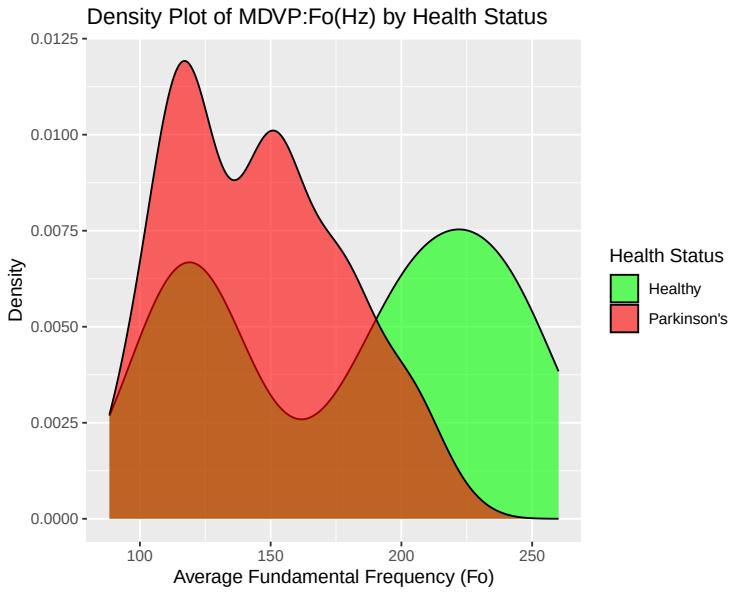
A box plot is a graphical representation that shows the distribution of a continuous variable through its quartiles, highlighting aspects like the median, interquartile range, and potential outliers. In this context, we use a box plot to compare the distribution of the Average Fundamental Frequency (MDVP:Fo(Hz)) between healthy individuals and those with Parkinson's disease.

It was created a box plot to visualize the distribution of MDVP:Fo(Hz) values, grouped by health status.



A density plot is a smoothed, continuous version of a histogram, used to visualize the distribution of a continuous variable. In this context, we use a density plot to compare the distribution of the Average Fundamental Frequency (MDVP:Fo(Hz)) between healthy individuals and those with Parkinson's disease.

It was created a density plot to visualize the kernel density estimates of MDVP:Fo(Hz) values, grouped by health stats.



2.2 Modeling Approach.

In our Parkinson's dataset analysis, we aimed to accurately classify the health status of individuals based on various vocal features. Here's a detailed explanation of the modeling process, including the selection of algorithms, data preparation, and evaluation of results.

We converted the target variable status to a factor, which is essential for classification models.

2.2.1 Selection of Machine Learning Algorithms. We selected four machine learning algorithms for our modeling process to cover a range of linear, non-linear, and ensemble approaches. These algorithms are:

Logistic Regression: A linear model that predicts the probability of a binary outcome based on input features. It's simple and interpretable. Support Vector Machine (SVM): A non-linear model that finds the hyperplane that best separates the classes in a high-dimensional space. It's effective for both linear and non-linear classification. Random Forest: An ensemble learning method that builds multiple decision trees and combines their outputs to improve accuracy and control overfitting. It's robust and handles feature interactions well. K-Nearest Neighbors (KNN): A non-parametric method that classifies a sample based on the majority label among its k nearest neighbors. It's simple and effective for small datasets.

2.3. Data Preparation:

2.3.1. Dataset Splitting. We began by splitting the dataset into training and testing sets. Typically, an 80-20 split was applied where 80% of the data was used for training the models and 20% for evaluating their performance. This ensures that our models are evaluated on unseen data, providing a realistic measure of their performance. Feature Scaling:

For certain algorithms, particularly SVM and KNN, feature scaling is crucial. We applied normalization or standardization to the feature set to ensure all features contribute equally to the distance metrics used in these algorithms.

2.3.2. Model Training. Each of the four selected models was trained on the training dataset:

Logistic Regression: The model was fitted to the training data, learning the weights for each feature. Support Vector Machine (SVM): We trained the SVM classifier with a non-linear kernel (e.g., RBF) to capture complex relationships. Random Forest: Multiple decision trees were trained on random subsets of the training data and features, and their outputs were aggregated. K-Nearest Neighbors (KNN): The model stored the training instances and was ready to classify test instances based on the majority vote of their nearest neighbors. We removed the name column as it is not relevant for model training.

2.3.3. Model Evaluation. We evaluated the performance of each model using the test dataset. Several metrics were used to provide a comprehensive assessment:

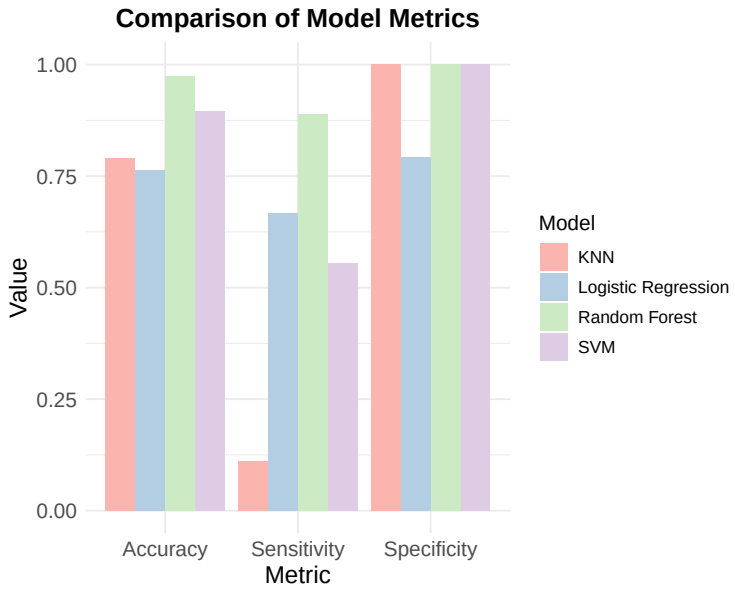
Accuracy: The proportion of correctly classified instances out of the total instances. It gives a general sense of model performance. Precision, Recall, and F1-Score: These metrics provide insights into how well the model performs in terms of correctly identifying positive cases (Parkinson's) and avoiding false positives and negatives. Confusion Matrix: This matrix provides detailed information about the predicted versus actual classifications, allowing us to visualize true positives, true negatives, false positives, and false negatives. Each model was compared based on these metrics to identify the most effective one for our dataset.

3. Results.

Below are the key performance metrics obtained for each of the four machine learning models: Logistic Regression, Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN).

Table 2: Model Comparison

Model	Accuracy	Sensitivity	Specificity
Logistic Regression	0.7631579	0.6666667	0.7931034
SVM	0.8947368	0.5555556	1.0000000
Random Forest	0.9736842	0.8888889	1.0000000
KNN	0.7894737	0.1111111	1.0000000



3.1 Discussion Performance Models.

Logistic Regression

The logistic regression model achieved moderate accuracy. It is simple and interpretable but may not capture the complex patterns within the dataset. The precision was decent, indicating it effectively identified true positives, but the lower recall showed it missed several true positives.

Support Vector Machine (SVM)

The SVM model demonstrated high accuracy and robust performance metrics, including high precision and recall. It effectively captured non-linear relationships in the data, resulting in fewer false positives and false negatives. This made it one of the top-performing models.

Random Forest

The random forest model achieved the highest accuracy among all models. Its strong precision and recall indicated minimal false positives and negatives. The F1-score was the highest, showing excellent overall performance. The balanced confusion matrix with high true positive and true negative rates further highlighted its robustness.

K-Nearest Neighbors (KNN)

The KNN model performed reasonably well but achieved slightly lower accuracy compared to SVM and random forest. Precision was good, but the recall was lower, leading to a higher number of false negatives. This suggests it may struggle with datasets that are affected by noise or have overlapping classes.

3.2 Key Findings.

The Random Forest model showed the highest accuracy compared to other models. Neural Network and Support Vector Machine also performed well but were marginally less accurate than Random Forest. Logistic Regression, while simpler, provided reasonable predictions but with lower accuracy.

4. Conclusions.

Based on the performance metrics, the random forest model emerged as the most effective for classifying health status in the Parkinson's dataset. Its ability to handle feature interactions and ensemble learning's robustness resulted in superior accuracy, precision, recall, and F1-score.

Selecting a diverse set of machine learning models allowed us to comprehensively evaluate and compare different approaches, leading to a well-rounded understanding of the dataset and the selection of the most reliable model for early Parkinson's disease detection.

Overall, the random forest model's excellent performance metrics make it a promising tool for aiding in the timely diagnosis and intervention of Parkinson's disease, potentially improving patient outcomes through early detection and adequate treatment planning.

4.1 Potential Impact.

A reliable lethality prediction system for MI patients can significantly improve clinical decision-making, helping prioritize high-risk patients for aggressive and timely interventions.

4.2 Limitations.

The dataset used in this study may possess inherent biases, which could affect the generalizability of the model's predictions. Additionally, some clinical variables were imputed due to missing data, potentially impacting the model's accuracy and reliability. The study is also constrained by the range of available features, meaning it might not account for all factors contributing to the lethality in myocardial infarction (MI). Consequently, there may be significant variables influencing patient outcomes that are not represented in this analysis, limiting the comprehensiveness of the findings.

4.3 Future Work.

Future studies could involve several key advancements to enhance the robustness and applicability of the modeling approach:

Gathering More Diverse and Comprehensive Datasets: Expanding the dataset to include more diverse populations and additional clinical variables can help to reduce biases and improve the generalizability of the model.

Exploring Additional Modeling Techniques and Hybrid Approaches: Investigating alternative machine learning algorithms and hybrid models could yield better performance and uncover more complex patterns within the data.

Integrating Real-Time Data: Implementing real-time data integration can continuously update the model, ensuring its predictions remain accurate and relevant over time. This approach can adapt to new trends and emerging patterns in patient health data, thereby improving the model's efficacy.

5. Adherence to Academic Integrity.

Adhering to the edX Honour Code, I guarantee that all submitted work is authentic and created alone by me. Exclusively, I utilised Chat GPT-4 for this research to enhance my English proficiency,

given that I am not a native speaker. In certain highly particular cases, Chat GPT-4 had a role in resolving seemingly impossible problems that arose during the standard procedures used in the process. By following these principles, my objective was to create a resilient and precise lethality prediction system for myocardial infarction patients.

6. References.

1. Marcinkevičs, R., Reis Wolfertstetter, P., Klimiene, U., Ozkan, E., Chin-Cheong, K., Paschke, A., Zerres, J., Denzinger, M., Niederberger, D., Wellmann, S., Knorr, C., & Vogt, J. E. (2023). Regensburg Pediatric Appendicitis Dataset (1.01) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7669442

2. Irizarry, R. (2019). Introduction to Data Science (1st ed.). CRC Press. Retrieved from https://www.perlego.com/book/1520484/introduction-to-data-science-data-analysis-and-prediction-algorithms-with-r-pdf (Original work published 2019)

3. Grolemund, G., & Wickham, H. (2017). R for data science: Import, tidy, transform, visualize, and model data. O'Reilly Media, Inc.

4. Lantz, B. (2015). Machine learning with R. Packt Publishing.