Extended Essay

Topic: Enhancing Parkinson's Disease Diagnosis: Insights from Speech Biomarkers and Machine Learning Algorithms

Research Question: How accurate is the integration of biomarker and machine-learning classification algorithms in detecting Parkinson's disease through voice?

Subject – Computer Science

Word Count – 3839

Table of Contents

1. INTRODUCTION

The practical application of Artificial Intelligence (AI) has been increasingly popular in the role of the healthcare industry, particularly with advancements in technologies designed to identify illnesses through voice analysis. Given the limited accuracy of traditional methods in detecting certain illnesses, AI is progressively replacing these methods. Hence, this research paper will specifically focus on both the use of speech *biomarker extraction* and *machine-learning classification* algorithms for the early detection of Parkinson's disease, a prominent neurological disorder. Specifically, the **primary** focus of this investigation will be on estimating the accuracy of integration between speech biomarker extraction and machine learning classification algorithms to diagnose Parkinson's disease based on speech signals.

2. THEORY

2.1 Parkinson's Disease

Parkinson's disease is one of the most widespread neurodegenerative[1] diseases around the world, and several voice symptoms can be observed. These symptoms can be broadly categorized into three main components: perceptual changes, acoustic changes, and additional factors. The primary voice changes associated with Parkinson's disease entail:

Perceptual Changes:

- **Softer voice:** Reduced volume and intensity.
- **Monotone speech:** Lack of variation in pitch and intonation.
- **Hoarseness and breathiness:** Raspy or airy quality to the voice.

[1] "Involving the nerves gradually stopping working" as defined in Cambridge Dictionary https://dictionary.cambridge.org/dictionary/english/neurodegenerative

- **Tremor in the voice:** Shaky or quivering vocal sound.
- **Slurred speech:** Difficulty pronouncing words clearly.
- **Hesitation and stuttering:** Pauses and disfluencies in speech.

Acoustic Changes:

- **Decreased fundamental frequency (F0):** Lower average pitch of the voice.
- **Increased jitter and shimmer:** Variations in F0 and vocal cord vibration.
- **Reduced harmonics-to-noise ratio (HNR):** Increased breathiness and background noise.
- **Changes in Mel-frequency cepstral coefficients (MFCCs):** Spectral characteristics of the voice.[2]

Additional Factors:

- **Speech rate and prosody:** Slow speech rate and rhythm[3]

In short, in the case of perceptual change, people with Parkinson's disease tend to have a softer voice, monotone speech[4], and hoarseness or breathy voice. At the same time, acoustic

---

[2] "Let's Talk Parkinson's and Speech." *YouTube*, YouTube, 25 July 2023, www.youtube.com/watch?v=CuKY49u2GT4 .

"Parkinson's Disease." *National Institute of Neurological Disorders and Stroke*, U.S. Department of Health and Human Services, www.ninds.nih.gov/health-information/disorders/parkinsons-disease . Accessed 20 Jan. 2023.

[3] Paker, Nurdan, et al. "Gait Speed and Related Factors in Parkinson's Disease." *Journal of Physical Therapy Science*, U.S. National Library of Medicine, Dec. 2015, www.ncbi.nlm.nih.gov/pmc/articles/PMC4713769/#:~:text=In%20previous%20studies%2C%20the%20gait,1.36%20m%2Fs28) .

[4] "a sound or way of speaking in which the tone and volume remain the same and therefore seem boring" as defined in the Oxford Dictionary https://www.oxfordlearnersdictionaries.com/definition/english/monotone_1?q=monotones

changes show an increased jitter and shimmer, reduced harmonics-to-noise ratio (HNR), and changes in Mel-frequency cepstral coefficients (MFCCs). Studying each voice characteristic observed in Parkinson's disease will be essential for further investigations.

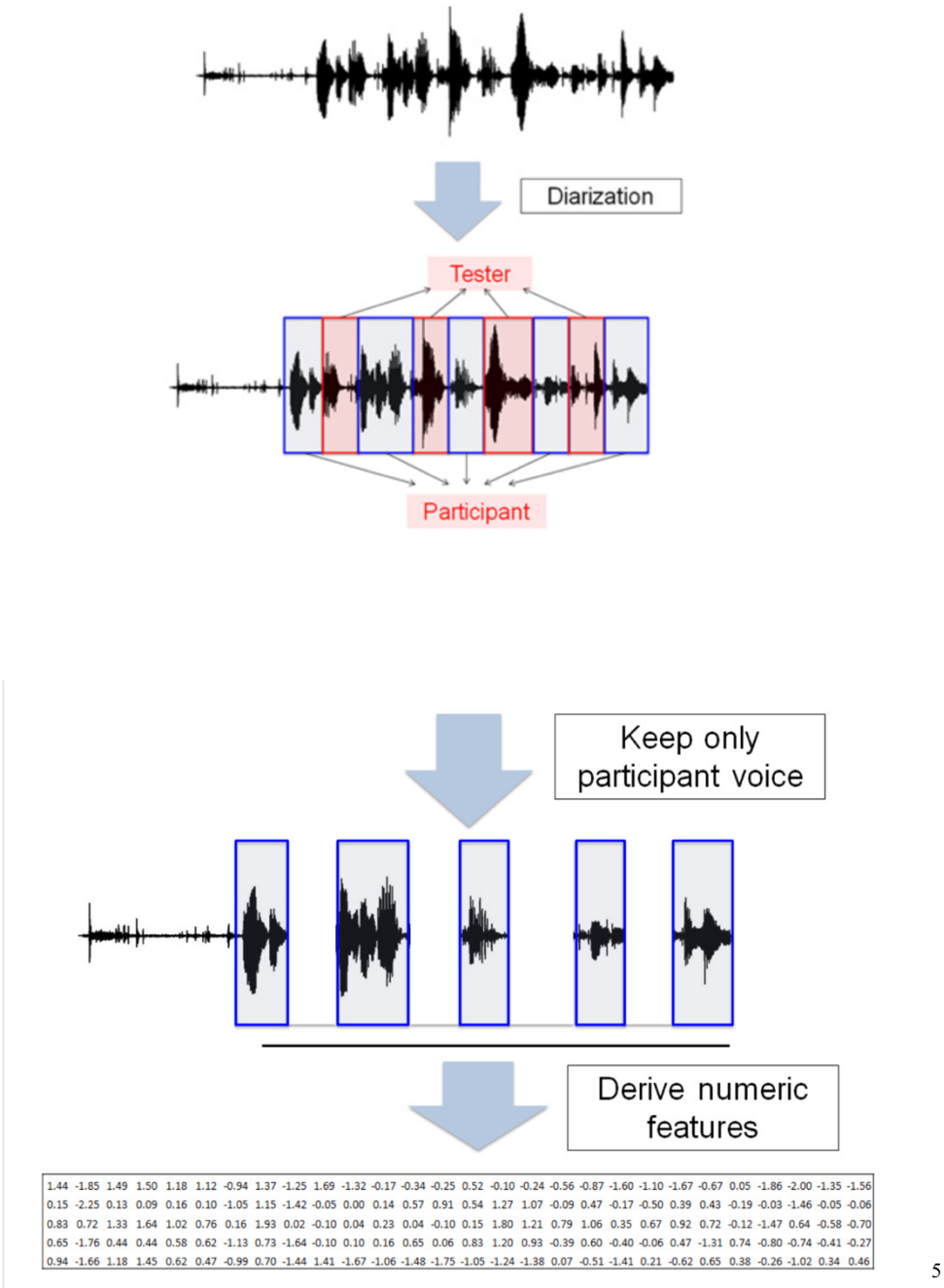
2.2 Speech Biomarker Extraction Algorithm

A speech biomarker extraction algorithm is designed to analyze speech signals and extract **acoustic** features associated with neurological disorders, such as Parkinson's disease in our case. This algorithm works by processing digital recordings of speech and extracting various characteristics of the speaker's voice, such as the fundamental frequency, intensity, duration, and spectral characteristics.

This can be shown in ***Figure 1***, which displays the preprocessing steps of voice data within the workflow, implementing the speech biomarker extraction algorithm using the "OpenSMILE" feature extraction tool to prepare only participant voice data for machine learning classification.

One fascinating feature observed in this process is that using the diarization process to distinguish the voice segments from the tester, and keeps only the voice of the participant. The diarization process divides an audio recording into segments that correspond to different speakers.

Figure 1: A diagram of the speech biomarker extraction algorithm (Data preprocessing)



[5]

If we break down each step in this image of the extraction process:

[5] Lin, Honghuang, et al. "Identification of Digital Voice Biomarkers for Cognitive Health." *Exploration of Medicine*, Open Exploration, 31 Dec. 2020, www.explorationpub.com/Journals/em/Article/100128.

1. Recording:

- The initial step involves recording the voice of both the tester and participant during an interaction.
- This recording captures the complete audio conversation.

2. Diarization:

- The second step utilizes diarization software to separate the voices of the tester and participant within the recording.
- This process identifies and segments the audio portions containing only the participant's voice, eliminating the tester's voice segments.

3. Segmentation:

- After isolating the participant's voice segments, these segments are further divided into smaller, uniform time intervals.
- This segmentation ensures consistent data units for feature extraction in the next step.

4. Feature Extraction with OpenSMILE (Biomarker extraction algorithm process):

- OpenSMILE, an open-source software tool, is employed to extract a set of numerical features from each segmented voice interval.
- These features capture various acoustic properties of the voice, including:
    - Pitch (fundamental frequency)
    - Intensity (loudness)
    - Mel-Frequency Cepstral Coefficients (MFCCs) related to spectral characteristics
    - Jitter and shimmer reflecting variations in pitch and loudness, respectively

- Formant frequencies, resonances of the vocal tract
- Speech rate and articulation measures

5. Feature Normalization:

- The extracted features may vary in their scales and distributions.
- This step involves normalization, bringing all features to a common scale to ensure they contribute equally during subsequent analysis.

6. Feature Selection (Optional):

- Depending on the specific analysis goals, a subset of the extracted features may be chosen for further analysis.
- Feature selection techniques can identify features that are most relevant to the target variable (e.g., cognitive health score) and reduce the dimensionality of the data.

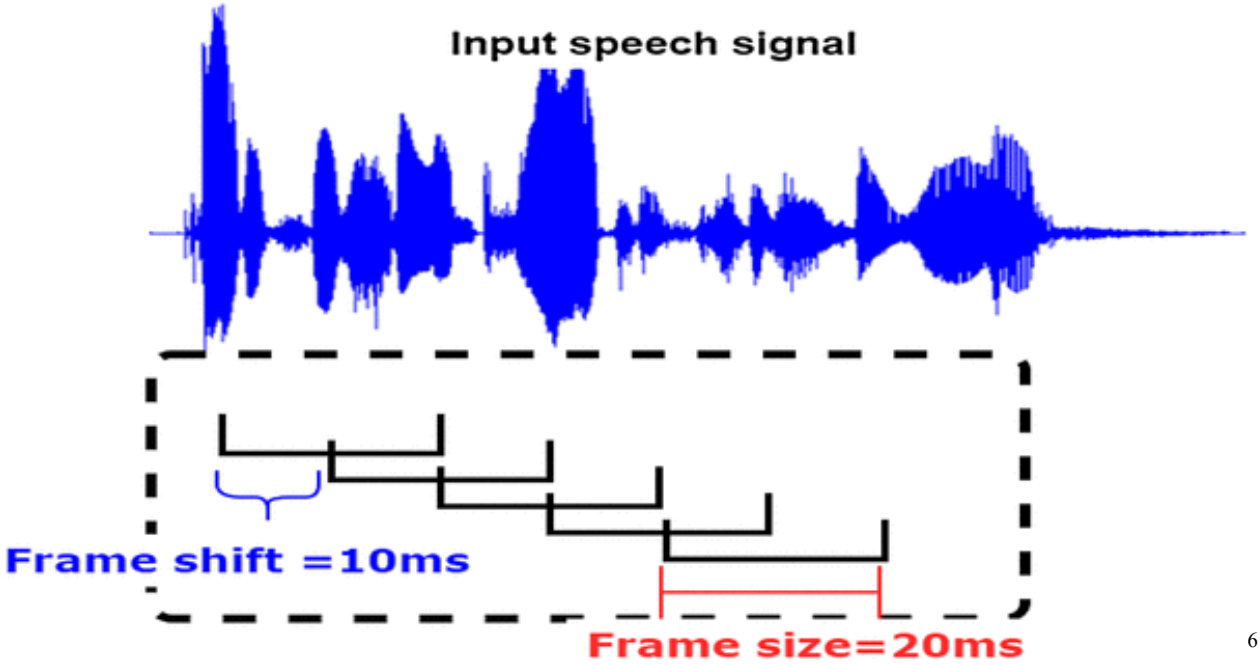
7. Data Preparation for Machine Learning:

- The final step prepares the preprocessed features for use in machine learning algorithms.
- This often involves formatting the data into a suitable format for the chosen algorithms

Therefore, if we sum up all the information above, the speech biomarker extraction algorithm analyzes the speech signals as a waveform and breaks the waveform down into smaller segments, known as frames (please refer to ***Figure 2***). These frames are then analyzed using the "OpenSMILE" (in this case) biomarker extraction algorithm tool to extract features. This

process remains consistent in the context of diagnosing Parkinson's disease, where the output of this algorithm is crucial for transferring the extracted voice data into machine-learning classification algorithms, a topic that will be explored in the next section.

Figure 2: A graphical representation of waveform frames.



[6]

2.3 Machine-Learning Classification Algorithms

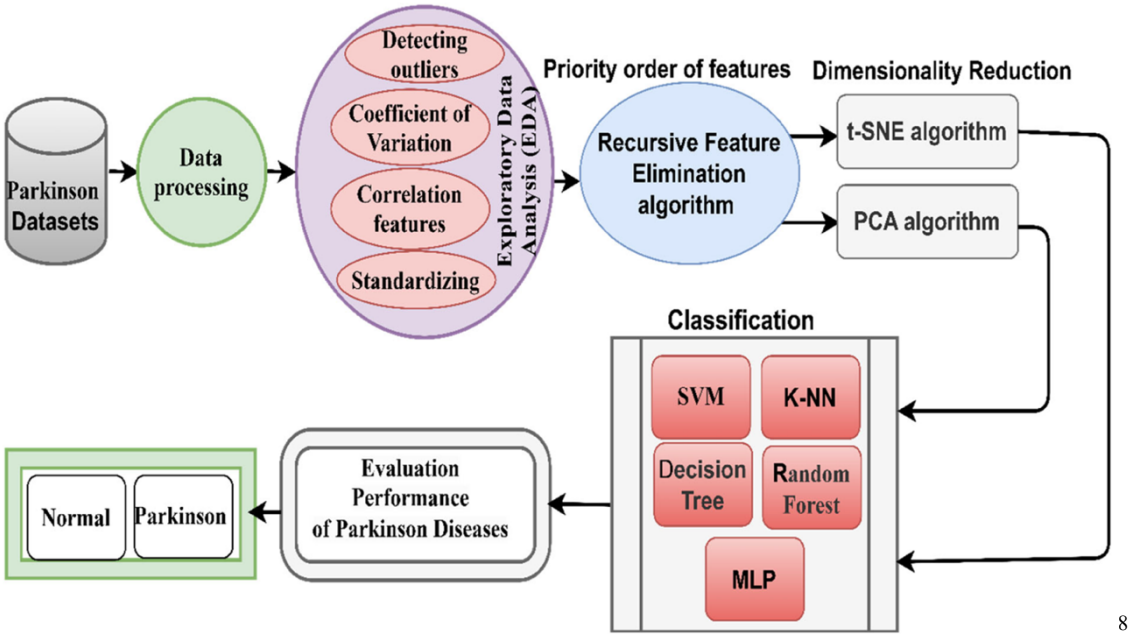
The machine-learning classification algorithms play an important role in the early detection of Parkinson's disease, primarily by using the extracted features from a speech[7] biomarker algorithm to classify speech signals as either indicative of good health or indicative of

[6] *Framing and Windowing of Filtered Signal - Researchgate*, www.researchgate.net/figure/Framing-and-Windowing-of-filtered-signal_fig3_323227273. Accessed 20 Jan. 2023.

[7] Alalayah, Khaled M., et al. "Automatic and Early Detection of Parkinson's Disease by Analyzing Acoustic Signals Using Classification Algorithms Based on Recursive Feature Elimination Method." *MDPI*, Multidisciplinary Digital Publishing Institute, 31 May 2023, www.mdpi.com/2075-4418/13/11/1924?fbclid=IwAR0jS_XHaoku8MRMuMUdYX5EwT3q_M0nQQYc-PmB85jybLOYkXAYTbYEHY0. Accessed 21 Jan 2023.

Parkinson's disease. In simpler terms, it examines the extracted features to determine their association with the disease and employs this information to make final predictions.

Figure 3: The Role of Classification Techniques in Diagnosing Parkinson's Disease



[8]

For example, as shown in ***Figure 3***, the classification technique in machine learning involves several advanced algorithms, including Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Decision Tree, Random Forest, and Multi-Layer Perceptron (MLP). While not all of these algorithms are commonly used to detect Parkinson's disease from voice, the SVM, Random Forest, and MLP algorithms are primarily utilized for this purpose. Here is the breakdown of each algorithm:

1. Support Vector Machine (SVM): Its effectiveness in high-dimensional data and robustness to small datasets make it ideal for analyzing diverse Parkinson's data like speech,

[8] Alalayah, Khaled M., et al. "Automatic and Early Detection of Parkinson's Disease by Analyzing Acoustic Signals Using Classification Algorithms Based on Recursive Feature Elimination Method." *MDPI*, Multidisciplinary Digital Publishing Institute, 31 May 2023, www.mdpi.com/2075-4418/13/11/1924 .

handwriting, and gait patterns. Studies have shown SVMs achieving accuracy rates exceeding 90% in some cases.[9]

2. Random Forest: This ensemble method excels at handling non-linear relationships and complex datasets, a common occurrence in Parkinson's research. It offers several advantages, including identifying crucial features for prediction and achieving high classification accuracy.

3. Multi-Layer Perceptron (MLP): Artificial neural networks like MLPs are gaining traction due to their ability to learn complex patterns from various data sources like speech, handwriting, and gait. While research is ongoing, MLPs have shown promising results in Parkinson's detection, potentially even outperforming other algorithms in specific data settings.[10]

2.4 openSMILE – Audio Extraction Tool

For my further experiment, the data I used (please refer to ***Appendix 1***) is derived from the "openSMILE" extraction tool to transform each participant's voice into numeric features represented in ***Figure 1*** above. In short, this tool which is completely written in C++ programming language called "OpenSMILE" refers to open-source speech and music interpretation by large space extraction; it works as an open source for audio feature extraction along with a classification of the signal from both speech and music signals.[11]

---

[9] Rana, Arti et al. "Imperative Role of Machine Learning Algorithm for Detection of Parkinson's Disease: Review, Challenges and Recommendations." *Diagnostics (Basel, Switzerland)* vol. 12,8 2003. 19 Aug. 2022, doi:10.3390/diagnostics12082003 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9407112/

[10] Author links open overlay panelAditi Govindu a, et al. "Early Detection of Parkinson's Disease Using Machine Learning." *Procedia Computer Science*, Elsevier, 31 Jan. 2023, www.sciencedirect.com/science/article/pii/S1877050923000078.

[11] "OpenSMILE 3.0." *audEERING*, 7 July 2022, www.audeering.com/research/opensmile/.

The working mechanism of this tool is as follows:

The "openSMILE" extraction tool operates in three main phases: the Pre-config phase, the Configuration phase, and the Execution phase. In the Pre-config phase, command-line options are read, the configuration file is parsed, and usage information is displayed. The Configuration phase involves creating the component manager, instantiating all components listed in its configuration array, and configuring each component in three phases: registration, main configuration, and finalization. The tool then enters the Execution phase, where the main execution loop is started, and each component's tick() method is called in series. This loop continues until at least one component indicates data processing, after which end-of-input processing may occur for off-line processing.[12]

2.5 The Relationship between two Algorithms

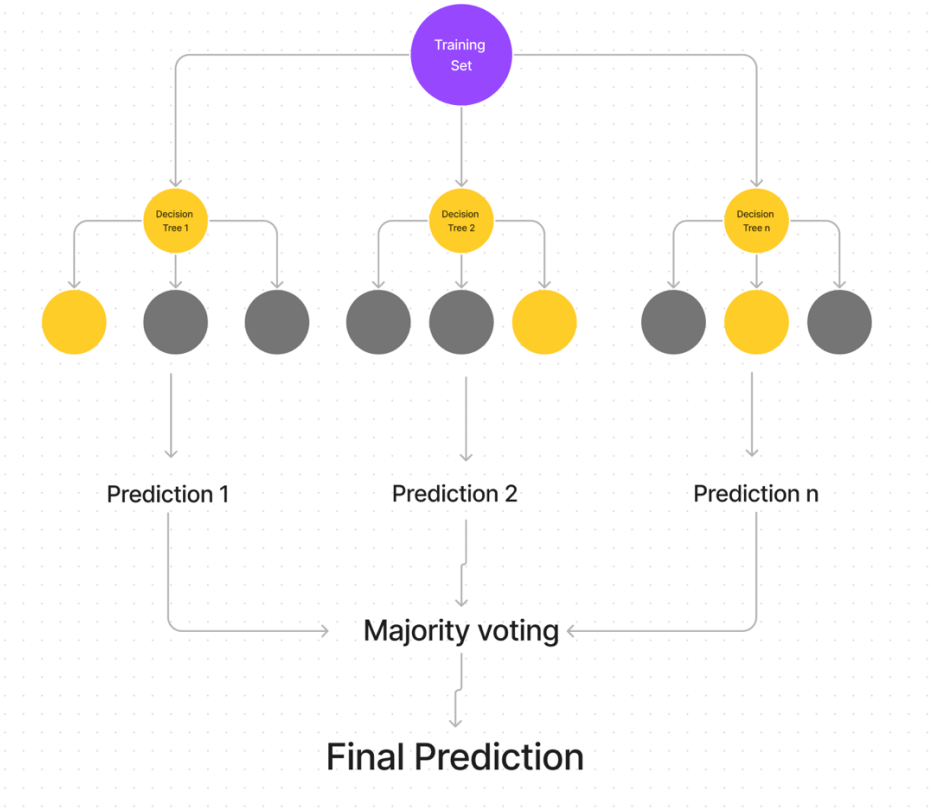
So, what is the relationship between these two algorithms? Generally, these two types of algorithms work together in a complementary manner. In simpler terms, the speech biomarker extraction algorithm prepares the input data for the machine learning classification algorithm, by extracting the relevant acoustic features from the speech signal as shown in ***Figure 1.*** The machine learning classification algorithm then uses these numeric features to predict the presence or absence of Parkinson's disease. This is because, without a biomarker extraction algorithm, the machine learning classification algorithm would not have access to the relevant information needed to make accurate predictions through voice analysis.

[12] "Reference Section¶." *Reference Section - openSMILE Documentation*, https://audeering.github.io/opensmile/reference.html . Accessed 22 Jan. 2023.

2.6 The Random Forest Classifier Algorithm

Random Forest is a supervised machine-learning algorithm that combines multiple decision trees to produce a single accurate result.[13] For diagnosing Parkinson's disease, a random forest classifier method is essential for making an accurate prediction on the input voice data. ***Figure 4***, shows the working mechanism of the random forest machine-learning classifier algorithm, which starts by dividing the training set (extracted data inputs) into multiple trees (from 1 to n as shown). Each tree makes a prediction based on the majority vote of its branches. The final prediction is the majority vote of all the trees in the forest. After all, the model outputs the final result of whether the patient has Parkinson's disease or not.

Figure 4: Random Forest Classifier Architecture



[13] *Random Forest classifier*. Random Forest Classifier - an overview | ScienceDirect Topics. (n.d.). https://www.sciencedirect.com/topics/computer-science/random-forest-classifier#:~:text=A%20random%20forest%20(RF)%20classifier,of%20training%20samples%20and%20variables.

3. HYPOTHESIS

In order to thoroughly investigate the accuracy of this technology, I am conducting an experiment centered around using voice data inputs (please refer to ***Appendix 1***) to measure the accuracy rate. Subsequently, this essay will compare this accuracy rate with the estimated accuracy of Aditi Govindu and Shushila Palwe, scientists in diagnosing Parkinson's disease through voice. They theorized in their "Early detection of Parkinson's disease using machine learning[14]" research paper that the Random Forest model has a detection accuracy of 91.83% and a sensitivity of 0.95. However, my assumption is that the accuracy of the diagnosis of Parkinson's disease may vary based on different voice input data. Therefore, throughout the experiment, my **primary** objectives are to **assess** whether different voice input data can affect the accuracy rate of diagnosing Parkinson's disease, thereby determining the real accuracy rate of the model.

My hypothesis regarding this argument is that even with different voice input data being tested, the accuracy rate might change due to external factors, like dataset size or dataset quality. Since the machine-learning model will compare each input data with healthy and Parkinson's diseased data, the accuracy rate of the Parkinson's disease detector machine-learning model will be close to Aditi Govindu and Shushila Palwe's estimated accuracy, at 91.83%.

[14] Author links open overlay panelAditi Govindu a, a, & AbstractParkinson's disease (PD) is a neurodegenerative disorder affecting 60% of people over the age of 50 years. Patients with Parkinson's (PWP) face mobility challenges and speech difficulties. (2023a, January 31). *Early detection of parkinson's disease using machine learning*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050923000078

4. METHODOLOGY

This experiment will be using a machine-learning model technique[15] implemented through Python source code and extracted voice samples (in CSV format) from the GitHub repository hosted by the user DeepranjanG[16] (Refer to ***Appendix 3*** for permission to use). The Python programming language will be employed to execute the code, and the specific interpretation of the experiment will be provided in this section.

4.1 Independent Variables

In this experiment, the type of voice input will be our only dependent variable. Since I am using data that has already been extracted through the biomarker algorithm, it is not possible to identify the characteristics (e.g., age, gender, length of recording) of the input data. Therefore, this variable may affect the final accuracy percentage results of the machine-learning model for detecting Parkinson's disease from voice.

4.2 Dependent Variables

In this experiment, the only dependent variable will be the accuracy of the machine learning model. The accuracy of this model will be measured through the experimental results of

[15] YouTube. (2022a, September 25). *Parkinson's disease detection using machine learning | machine learning projects 2022 | simplilearn*. YouTube. https://www.youtube.com/watch?v=CQLkX4utdIU&t=518s

[16] DeepranjanG. (n.d.). *DeepranjanG/Parkinson-S-disease-detection*. GitHub. https://github.com/DeepranjanG/Parkinson-s-Disease-Detection

accuracy prediction and is expected to vary depending on the biomarker extraction method that is used to extract features from the voice data.

4.3 Controlled Variables

Variable	Description	Specifications (if applicable)
Computer/Device used	I will be using my own laptop: a MacBook Air	**Chip**: Apple M1 **Memory**: 8GB **Version**: 14.2.1
Integrated Development Environment (IDE) used	I will be using the "Jupyter Notebook" for running my code.	**IDE**: Jupyter Notebook **Memory and disk space**: 1 GB RAM, 1 GB of disk, and 0.5 CPU core per user **Server overhead**: 2-4 GB or 10% system overhead
Data preproccessing steps	Feature extraction of the voice using the biomarker extraction algorithm will be part my data preprocessing section.	
Same type of machine-learning algorithm	Both my experiment and the experiments conducted by Aditi Govindu and Shushila Palwe use the "Random Forest" algorithm.	

4.4 Procedure

The procedure of my experiment will look like this:

1. Use the Python code (please refer to ***Appendix 2***) to prepare the machine-learning model for further detection.

2. Use the comma-separated values (CSV) formatted voice data extracted through the Biomarker extraction algorithm (please refer to ***Appendix 1***) by directly specifying the file path: `input_data = ("path of the voice data")`.

3. Use 195 voice data samples and prepare them to be processed through the machine-learning model. (refer to ***Appendix 1***)

4. Calculate the accuracy using the following formula:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ nuber\ of\ predictions} \times 100\%$$

5. Evaluate the experiment results in diagram/table form.

4.5 Data Preprocessing

The data preprocessing stage of my experiment involves transforming 195 extracted voice samples, with 19 of them displayed in ***Figure 5***, into a format suitable for easy reading and analysis by the machine-learning model. This crucial step aims to enhance data quality, improve model performance, and identify any missing values. For instance, as depicted in ***Figure 6***, checking for missing values in each column using the 'df.isnull().sum()' method and displaying the first 5 rows with 'df.head()' provides a quick overview of the data structure. Subsequent steps are as follows:

Figure 5: Used voice sample data


```
name,MDVP:Fo(Hz),MDVP:Fhi(Hz),MDVP:Flo(Hz),MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP,MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA,NHR,HNR,status,RPDE,DFA,spread1,spread2,D2,PPE
phon_R01_S01_1,119.99200,157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.04374,0.42600,0.02182,0.03130,0.02971,0.06545,0.02211,21.03300,1,0.414783,0.815285,-4.813031,0.266482,2.301442,0.284654
phon_R01_S01_2,122.40000,148.65000,113.81900,0.00968,0.00008,0.00465,0.00696,0.01394,0.06134,0.62600,0.03134,0.04518,0.04368,0.09403,0.0[illegible]00,1,0.458359,0.819521,-4.075192,0.335590,2.486855,0.368674
phon_R01_S01_3,116.68200,131.11100,111.55500,0.01050,0.00009,0.00544,0.00781,0.01633,0.05233,0.48200,0.02757,0.03858,0.03590,0.08270,0.01309,20.65100,1,0.429895,0.825288,-4.443179,0.311173,2.342259,0.332634
phon_R01_S01_4,116.67600,137.87100,111.36600,0.00997,0.00009,0.00502,0.00698,0.01505,0.05492,0.51700,0.02924,0.04005,0.03772,0.08771,0.01353,20.64400,1,0.434969,0.819235,-4.117501,0.334147,2.405554,0.368975
phon_R01_S01_5,116.01400,141.78100,110.65500,0.01284,0.00011,0.00655,0.00908,0.01966,0.06425,0.58400,0.03490,0.04825,0.04465,0.10470,0.01767,19.64900,1,0.417356,0.823484,-3.747787,0.234513,2.332180,0.410335
phon_R01_S01_6,120.55200,131.16200,113.78700,0.00968,0.00008,0.00463,0.00750,0.01388,0.04701,0.45600,0.02328,0.03526,0.03243,0.06985,0.01222,21.37800,1,0.415564,0.825069,-4.242867,0.299111,2.187560,0.357775
phon_R01_S02_1,120.26700,137.24400,114.82000,0.00333,0.00003,0.00155,0.00202,0.00466,0.01608,0.14000,0.00779,0.00937,0.01351,0.02337,0.00607,24.88600,1,0.596040,0.764112,-5.634322,0.257682,1.854785,0.211756
phon_R01_S02_2,107.33200,113.84000,104.31500,0.00290,0.00003,0.00144,0.00182,0.00431,0.01567,0.13400,0.00829,0.00946,0.01256,0.02487,0.00344,26.89200,1,0.637420,0.763262,-6.167603,0.183721,2.064693,0.163755
phon_R01_S02_3,95.73000,132.06800,91.75400,0.00551,0.00006,0.00293,0.00332,0.00880,0.02093,0.19100,0.01073,0.01277,0.01717,0.03218,0.01070,21.81200,1,0.615551,0.773587,-5.498678,0.327769,2.322511,0.231571
phon_R01_S02_4,95.05600,120.10300,91.22600,0.00532,0.00006,0.00268,0.00332,0.00803,0.02838,0.25500,0.01441,0.01725,0.02444,0.04324,0.01022,21.86200,1,0.547037,0.798463,-5.011879,0.325996,2.432792,0.271362
phon_R01_S02_5,88.33300,112.24000,84.07200,0.00505,0.00006,0.00254,0.00330,0.00763,0.02143,0.19700,0.01079,0.01342,0.01892,0.03237,0.01166,21.11800,1,0.611137,0.776156,-5.249770,0.391002,2.407313,0.249740
phon_R01_S02_6,91.90400,115.87100,86.29200,0.00540,0.00006,0.00281,0.00336,0.00844,0.02752,0.24900,0.01424,0.01641,0.02214,0.04272,0.01141,21.41400,1,0.583390,0.792520,-4.960234,0.363566,2.642476,0.275931
phon_R01_S04_1,136.92600,159.86600,131.27600,0.00293,0.00002,0.00118,0.00153,0.00355,0.01259,0.11200,0.00656,0.00717,0.01140,0.01968,0.00581,25.70300,1,0.460600,0.646846,-6.547148,0.152813,2.041277,0.138512
phon_R01_S04_2,139.17300,179.13900,76.55600,0.00390,0.00003,0.00165,0.00208,0.00496,0.01642,0.15400,0.00728,0.00932,0.01797,0.02184,0.01041,24.88900,1,0.430166,0.665833,-5.660217,0.254989,2.519422,0.199889
phon_R01_S04_3,152.84500,163.30500,75.83600,0.00294,0.00002,0.00121,0.00149,0.00364,0.01828,0.15800,0.01064,0.00972,0.01246,0.03191,0.00609,24.92200,1,0.474791,0.654027,-6.105098,0.203653,2.125618,0.170100
phon_R01_S04_4,142.16700,217.45500,83.15900,0.00369,0.00003,0.00157,0.00203,0.00471,0.01503,0.12600,0.00772,0.00888,0.01359,0.02316,0.00839,25.17500,1,0.565924,0.658245,-5.340115,0.210185,2.205546,0.234589
phon_R01_S04_5,144.18800,349.25900,82.76400,0.00544,0.00004,0.00211,0.00292,0.00632,0.02047,0.19200,0.00969,0.01200,0.02074,0.02908,0.01859,22.33300,1,0.567380,0.644692,-5.440040,0.239764,2.264501,0.218164
phon_R01_S04_6,168.77800,232.18100,75.60300,0.00718,0.00004,0.00284,0.00387,0.00853,0.03327,0.34800,0.01441,0.01893,0.03430,0.04322,0.02[illegible]
```

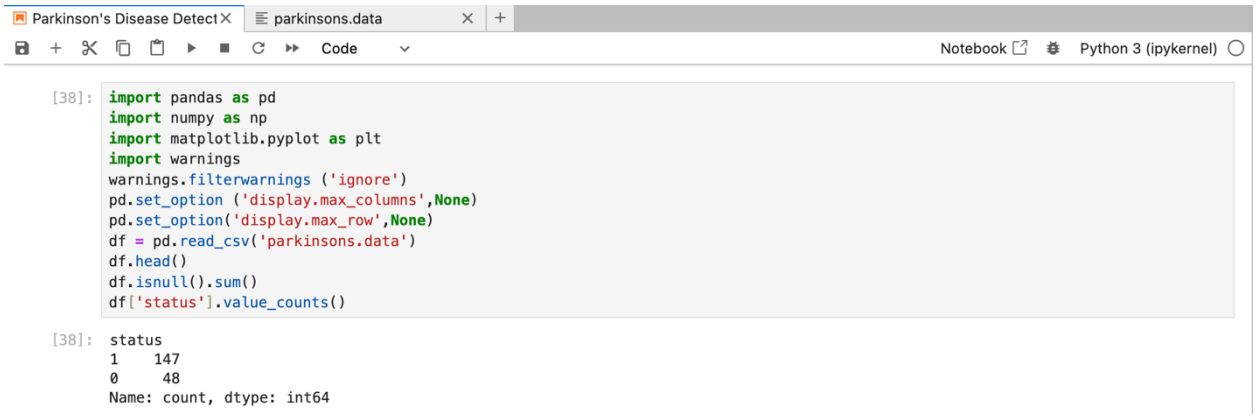
Figure 6: Checking the null values in the dataset (no null values)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings ('ignore')
pd.set_option ('display.max_columns',None)
pd.set_option('display.max_row',None)
df = pd.read_csv('parkinsons.data')
df.head()
df.isnull().sum()
```

```
name              0
MDVP:Fo(Hz)       0
MDVP:Fhi(Hz)      0
MDVP:Flo(Hz)      0
MDVP:Jitter(%)    0
MDVP:Jitter(Abs)  0
MDVP:RAP          0
MDVP:PPQ          0
Jitter:DDP        0
MDVP:Shimmer      0
MDVP:Shimmer(dB)  0
Shimmer:APQ3      0
Shimmer:APQ5      0
MDVP:APQ          0
Shimmer:DDA       0
NHR               0
HNR               0
status            0
RPDE              0
DFA               0
spread1           0
spread2           0
D2                0
PPE               0
dtype: int64
```

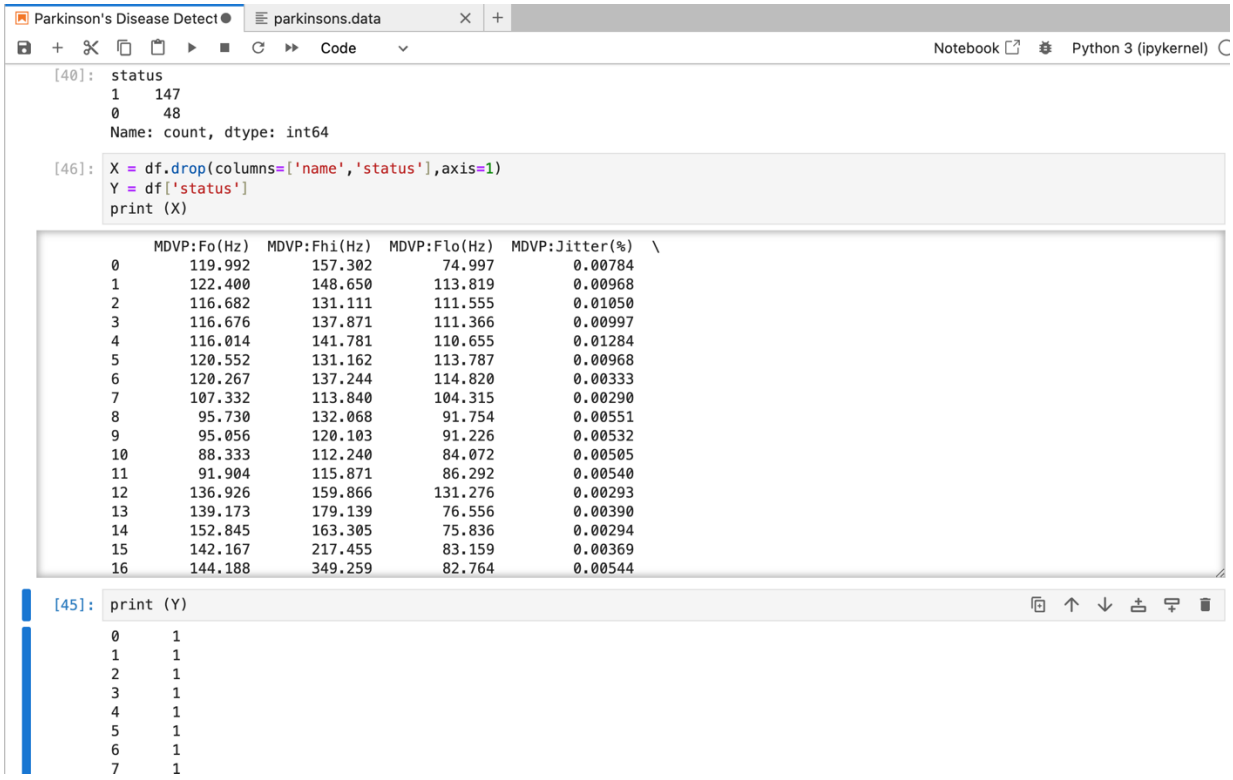
To clarify the distribution of Parkinson's and non-Parkinson's data, crucial for measuring accuracy, we will distribute our Target variable columns. ***Figure 6*** demonstrates how 'df['status'].value_counts()' method specifically deals with the "status" column of the dataset, which contains labels indicating the ratio of Parkinson's-affected and non-affected values. The value 1 means affected data, which is at 147, and 0 means unaffected data, at 48.

Figure 6: Dataset Status Distribution (Affected vs Non-Affected)



Our last step, in ***Figure 7***, is splitting the data into training and testing sets from the “X” input data to the “Y” target variable. By doing so, it will be much more appropriate for the machine-learning model to test the dataset without any trouble. Line 45 in the code represents the resulting two columns of data: the first column aligns with the data indices (from 0 to 195), and the second column indicates whether the corresponding data point is classified as Parkinson’s or not. (1 is with Parkinson’s disease, 0 is without Parkinson’s disease)

Figure 7: Data splitting



4.5 Random Forest Classification

As exhibited by the main functionalities in ***Figure 4***, which represent the working mechanism of the Random Forest classifier, this section will present some machine-learning techniques in the Python programming language for classifying the preprocessed data into the random forest machine-learning classifier (Please refer to ***Appendix 2***).

As shown in ***Figure 8***, the whole process of this line of code represents the final preparation of data by standardizing features and using the Random Forest Classifier for training a model. The first step involves importing the necessary modules from a machine-learning library: scikit-learn. After that, we are going to standardize the features in the training and datasets using the StandardScaler. This preprocessing step of standardization ensures that all features have a mean of 0 and a standard deviation of 1, further preventing certain features from dominating others during model training.

The next step is to create a random forest classifier model with 1000 decision trees. Subsequently, this model will be trained on a dataset using the 'fit' method and used for making predictions.

Figure 8: Feature Standardization and Random Forest Classification

```
•[13]: from sklearn.ensemble import RandomForestClassifier
       from sklearn.preprocessing import StandardScaler

       # Assuming X_train, Y_train, X_test are defined before this point

       # Standardize the features
       ss = StandardScaler()
       ss.fit(X_train)
       X_train = ss.transform(X_train)
       X_test = ss.transform(X_test)

       # Create a Random Forest classifier
       model = RandomForestClassifier(n_estimators=1000, random_state=42)

       # Train the Random Forest model
       model.fit(X_train, Y_train)
```

Figure 9 outlines the final step of the experiment, which is dedicated to scaling a set of input features using a StandardScaler. After that, it will feed them into a machine learning model to determine whether the model predicts the presence of Parkinson's disease based on the provided features. The data transmission process is also implemented in this code, as demonstrated by using the Python NumPy library to convert the features into 'input_data_n'.

Figure 9: Parkinson's Disease Prediction Model Implementation

```
input_data = (157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.04374,0.42600,0.02182,0.03130,0.02971,0.06545,0.022
input_data_np = np.asarray (input_data)
input_data_re = input_data_np.reshapre(1, -1)
s_data = ss.transform(input_data_re)
pred = model.predict(s_data)

print(pred)

if (pred[0]==0):
    print("Negative, No Parkinson's Found")
else:

    print("Positive, Parkinson's Found")
```

From ***Figure 9***, I deliberately chose the second feature from the datasets, as highlighted in ***Figure 5***. The actual characteristic of the feature is Parkinson's disease, and this will be illustrated in the data-splitting process in ***Figure 8***. As expected, ***Figure 10*** shows the final result of the tested feature, which prints a "Positive, Parkinson's Found" message on the screen. Hence, we can conclude that this model has successfully detected Parkinson's disease based on this feature.

Figure 10: The result given by the model

```
Positive, Parkinson's Found
```

5. RESULTS AND DISCUSSION

5.1 Accuracy Calculation

Table 1: A result of the experiment

Experiment	Parkinson's Disease	No Parkinson's Desease
Number of Samples	147	48
Correct Predictions	135	42
Incorrect Predictions	12	6
Accuracy rate (%)	91.8%	87.5%

Out of the 48 non-Parkinsons data points, our machine-learning classifier successfully identified 42, resulting in an accuracy rate of 87.5% using the formula above. In the case of the 147 Parkinson's disease data points, the model correctly detected 135, giving an accuracy percentage of 91.8%.

To obtain the average accuracy rate of our machine-learning model, we add these percentages and divide the sum by 2.

$$Average\ accuracy\ rate = \frac{(91.8\% + 87.5\%)}{2} = 89.6\%$$

Now, we have the precise accuracy rate of the Random Forest mechanism in our machine-learning classifier, which is approximately 89.6%.

5.2 Discussion

My initial hypothesis for this experiment was that the accuracy of the machine-learning model would remain consistent when different voice input data is tested. This hypothesis was

tested using a Random Forest Machine-learning classifier to calculate accuracy, and the results were compared with scientist Aditi Govindu and Shushila Palwe's estimated accuracy of 91.83%. In other words, the experiment aimed to determine whether the accuracy of the machine-learning technique is close to the scientists'. Referring to ***Table 1*** and conducting further calculations, my estimated accuracy for the machine-learning classifier was roughly 89.6%, reflecting a 2.2% difference compared to the scientists' result. Upon seeing this, despite this small difference, the results still suggest a notable alignment between the two outcomes: I can say that this Random Forest classifier-based classifier machine-learning model is authentically accurate for diagnosing Parkinson's disease through voice. However, if we determine the cause of the 2.2% error, the following factors may have affected it:

1. Differences in Data Sets and Biomarkers extraction method differences

The accuracy of the machine-learning classifier is highly dependent on the quality and composition of the data used for testing. Hence, one possible reason for differences in accuracy between this investigation and Aditi Govindu and Shushila Palwe's investigation may lie in the different extractions or preparation methods of the voice data.

2. Data Preprocessing and Handling differences

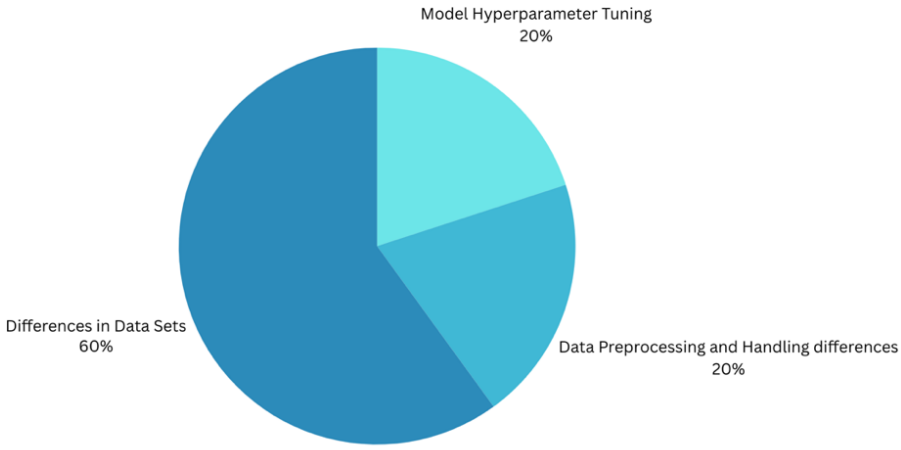
An additional explanation could be that the performance of the machine-learning models can be influenced by the preprocessing of the data or the feature extraction process. Specifically, in the article from Aditi Govindu and Shushila Palwe in which voice data was explored the data underwent careful cleaning from outliers and noise. In the same token, the other experiment was not handled in this manner and thus may score lower accordingly. Additionally, the way data is processed or extracted in the model can significantly affect the efficiency of its performance. This is so because if Aditi Goviudd and Shushila Pawales's

papers employed a different set of features or treated the features in a different manner, the result may not be the same.

3. Model Hyperparameter Tuning

Model hyperparameter tuning refers to the adaptation of underlying machine learning algorithms for a library, namely model hyperparameters. So, if the model was tuned based on its performance on the validation data, there is a risk of overfitting that voice data, which can possibly lead to reduced accuracy of new data. This is because the model may have learned the specific patterns in the validation data too well, and may not generalize well to data that is different.

Figure 11: A pie chart for the contribution of each cause to the 2.2% error



As shown in ***Figure 8***, presented as a pie chart form, the approximate percentage contributions of each factor to the 2.2% error in the estimated accuracy of the machine-learning model are delineated. In particular, a significant portion of this pie chart is attributed to differences in the datasets for both experiments, which can be linked to the distinct methodologies employed in each experiment. The reasons encompassed within this category

involve dataset biases, dataset size, and diversity. These factors may be considered impactful in contributing to the observed discrepancies in the machine-learning model's accuracy compared to other causes.

Upon reflection on the entire experiment, it becomes apparent that despite the close proximity of the accuracy percentage estimated in this research paper to the value reported by Aditi Govindu and Shushila Palwe, a marginal error persists. This error can be attributed to several potential causes, as outlined earlier, with particular emphasis on the disparities within the datasets. Remarkably, issues related to the different methods used to extract the voice data using the Biomarker algorithm emerge as key factors influencing the observed discrepancies. Therefore, recognizing and addressing these contributors are crucial steps in refining the model and further improving its accuracy.

6. CONCLUSION

All in all, the primary objective of the experiment was to assess how the accuracy of the machine-learning model varies when different voice data, extracted through the Biomarker algorithm, is used as input. As expected, the experimental outcome indicates a marginal disparity in accuracy percentages among various voice datasets, as indicated in ***Table 1***. Notably, the accuracy percentages observed in this study (89.6%) were found to be in close proximity to the values reported by Aditi Govindu and Shushila Palwe (91.83%). To take further, this investigation also aimed to analyze the cause of this error, concluding that this error is particularly dependent on the Biomarker extraction methods. Therefore, I am concluding that the machine-learning technique for diagnosing Parkinson's disease through

voice is accurate, and a slight degree of inaccuracy may occur due to the difference in the Biomarker extraction algorithm method used in both experiments.

To answer the research question of this essay, "*How accurate is the integration of biomarker and machine-learning classification algorithms in detecting Parkinson's disease through voice?",* my final conclusion is that the accuracy of the integration between Biomarker and Machine-learning classification algorithms is really accurate for diagnosing Parkinson's disease through voice.

When thinking about further development, it is also important to note some aspects that need to be considered so that this technology can be more effective and ethical.

1. Dataset Diversity and Size:

- Future studies should explore the impact of dataset variance and size on the accuracy of diagnosis of Parkinson's disease. To make the model more reliable and applicable to a greater number of situations, the dataset could be increased in size, and many voices added as well.

2. Feature Engineering:

- Further acoustic and perceptual feature analysis can provide for further betterment of the Biomarker algorithm. It may be, however locating and inserting new features could support the model in correlating to the most insignificant speech pattern variations related to Parkinson's disease.

3. Integration of Multi-Model Data:

- The innovation of integrating various sources, such as genetic information and wearable devices, to input data into the machine-learning model, may enhance diagnostic accuracy.

4. Ethical Considerations:

- Lastly, when seeking to introduce and deploy any kind of medical diagnostic technology, ethical considerations become important elements of interest. Basically, transparency of usage of data and informed consent procedures on social media along with privacy issues should be the first top priorities that researchers need to consider.

By considering such, these implication aims at filling the gap in trust between the old method of basic diagnostics and a new one that emerges with the use of machine learning for Parkinson's disease detection.

BIBLIOGRAPHY

"Involving the nerves gradually stopping working" as defined in Cambridge Dictionary https://dictionary.cambridge.org/dictionary/english/neurodegenerative

"Let's Talk Parkinson's and Speech." *YouTube*, YouTube, 25 July 2023, www.youtube.com/watch?v=CuKY49u2GT4 .

"Parkinson's Disease." *National Institute of Neurological Disorders and Stroke*, U.S. Department of Health and Human Services, www.ninds.nih.gov/health-information/disorders/parkinsons-disease . Accessed 20 Jan. 2023.

Paker, Nurdan, et al. "Gait Speed and Related Factors in Parkinson's Disease." *Journal of Physical Therapy Science*, U.S. National Library of Medicine, Dec. 2015, www.ncbi.nlm.nih.gov/pmc/articles/PMC4713769/#:~:text=In%20previous%20studies%2C%20the%20gait,1.36%20m%2Fs28) .

"a sound or way of speaking in which the tone and volume remain the same and therefore seem boring" as defined in the Oxford Dictionary https://www.oxfordlearnersdictionaries.com/definition/english/monotone_1?q=monotones

Lin, Honghuang, et al. "Identification of Digital Voice Biomarkers for Cognitive Health." *Exploration of Medicine*, Open Exploration, 31 Dec. 2020, www.explorationpub.com/Journals/em/Article/100128.

Framing and Windowing of Filtered Signal - Researchgate, www.researchgate.net/figure/Framing-and-Windowing-of-filtered-signal_fig3_323227273. Accessed 20 Jan. 2023.

Alalayah, Khaled M., et al. “Automatic and Early Detection of Parkinson’s Disease by Analyzing Acoustic Signals Using Classification Algorithms Based on Recursive Feature Elimination Method.” *MDPI*, Multidisciplinary Digital Publishing Institute, 31 May 2023, www.mdpi.com/2075-4418/13/11/1924?fbclid=IwAR0jS_XHaoku8MRMuMUdYX5EwT3q_M0nQQYc-PmB85jybLOYkXAYTbYEHY0. Accessed 21 Jan 2023.

Rana, Arti et al. “Imperative Role of Machine Learning Algorithm for Detection of Parkinson's Disease: Review, Challenges and Recommendations.” *Diagnostics (Basel, Switzerland)* vol. 12,8 2003. 19 Aug. 2022, doi:10.3390/diagnostics12082003 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9407112/

Author links open overlay panelAditi Govindu a, et al. “Early Detection of Parkinson’s Disease Using Machine Learning.” *Procedia Computer Science*, Elsevier, 31 Jan. 2023, www.sciencedirect.com/science/article/pii/S1877050923000078.

“Reference Section¶.” *Reference Section - openSMILE Documentation*, https://audeering.github.io/opensmile/reference.html . Accessed 22 Jan. 2023.

Random Forest classifier. Random Forest Classifier - an overview | ScienceDirect Topics. (n.d.). https://www.sciencedirect.com/topics/computer-science/random-forest-classifier#:~:text=A random forest (RF) classifier,of training samples and variables.

Author links open overlay panelAditi Govindu a, a, & AbstractParkinson’s disease (PD) is a neurodegenerative disorder affecting 60% of people over the age of 50 years. Patients with Parkinson’s (PWP) face mobility challenges and speech difficulties. (2023a, January 31). *Early detection of parkinson’s disease using machine learning*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050923000078

YouTube. (2022a, September 25). *Parkinson's disease detection using machine learning | machine learning projects 2022 | simplilearn*. YouTube. https://www.youtube.com/watch?v=CQLkX4utdIU&t=518s

DeepranjanG. (n.d.). *DeepranjanG/Parkinson-S-disease-detection*. GitHub. https://github.com/DeepranjanG/Parkinson-s-Disease-Detection

APPENDICES

Appendix 1: Voice Data Set Used in the experiment

12 sample voice datasets out of 195:

name,MDVP:Fo(Hz),MDVP:Fhi(Hz),MDVP:Flo(Hz),MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP,MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA,NHR,HNR,status,RPDE,DFA,spread1,spread2,D2,PPE

phon_R01_S01_1,119.99200,157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.04374,0.42600,0.02182,0.03130,0.02971,0.06545,0.02211,21.03300,1,0.414783,0.815285,-4.813031,0.266482,2.301442,0.284654

phon_R01_S01_2,122.40000,148.65000,113.81900,0.00968,0.00008,0.00465,0.00696,0.01394,0.06134,0.62600,0.03134,0.04518,0.04368,0.09403,0.01929,19.08500,1,0.458359,0.819521,-4.075192,0.335590,2.486855,0.368674

phon_R01_S01_3,116.68200,131.11100,111.55500,0.01050,0.00009,0.00544,0.00781,0.01633,0.05233,0.48200,0.02757,0.03858,0.03590,0.08270,0.01309,20.65100,1,0.429895,0.825288,-4.443179,0.311173,2.342259,0.332634

phon_R01_S01_4,116.67600,137.87100,111.36600,0.00997,0.00009,0.00502,0.00698,0.01505,0.05492,0.51700,0.02924,0.04005,0.03772,0.08771,0.01353,20.64400,1,0.434969,0.819235,-4.117501,0.334147,2.405554,0.368975

phon_R01_S01_5,116.01400,141.78100,110.65500,0.01284,0.00011,0.00655,0.00908,0.01966,0.06425,0.58400,0.03490,0.04825,0.04465,0.10470,0.01767,19.64900,1,0.417356,0.823484,-3.747787,0.234513,2.332180,0.410335

phon_R01_S01_6,120.55200,131.16200,113.78700,0.00968,0.00008,0.00463,0.00750,0.013
88,0.04701,0.45600,0.02328,0.03526,0.03243,0.06985,0.01222,21.37800,1,0.415564,0.8250
69,-4.242867,0.299111,2.187560,0.357775

phon_R01_S02_1,120.26700,137.24400,114.82000,0.00333,0.00003,0.00155,0.00202,0.004
66,0.01608,0.14000,0.00779,0.00937,0.01351,0.02337,0.00607,24.88600,1,0.596040,0.7641
12,-5.634322,0.257682,1.854785,0.211756

phon_R01_S02_2,107.33200,113.84000,104.31500,0.00290,0.00003,0.00144,0.00182,0.004
31,0.01567,0.13400,0.00829,0.00946,0.01256,0.02487,0.00344,26.89200,1,0.637420,0.7632
62,-6.167603,0.183721,2.064693,0.163755

phon_R01_S02_3,95.73000,132.06800,91.75400,0.00551,0.00006,0.00293,0.00332,0.00880,
0.02093,0.19100,0.01073,0.01277,0.01717,0.03218,0.01070,21.81200,1,0.615551,0.773587,-
5.498678,0.327769,2.322511,0.231571

phon_R01_S02_4,95.05600,120.10300,91.22600,0.00532,0.00006,0.00268,0.00332,0.00803,
0.02838,0.25500,0.01441,0.01725,0.02444,0.04324,0.01022,21.86200,1,0.547037,0.798463,-
5.011879,0.325996,2.432792,0.271362

phon_R01_S02_5,88.33300,112.24000,84.07200,0.00505,0.00006,0.00254,0.00330,0.00763,
0.02143,0.19700,0.01079,0.01342,0.01892,0.03237,0.01166,21.11800,1,0.611137,0.776156,-
5.249770,0.391002,2.407313,0.249740

17

[17] DeepranjanG. (n.d.). *DeepranjanG/Parkinson-S-disease-detection*. GitHub. https://github.com/DeepranjanG/Parkinson-s-Disease-Detection

Appendix 2: Python Code used for the machine-learning model

```
//data preproccessing

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings ('ignore')

pd.set_option ('display.max_columns',None)

pd.set_option('display.max_row',None)

df = pd.read_csv('parkinsons.data')

df.head()

df.isnull().sum()

df['status'].value_counts()

X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size = 0.2, random_state = 2)

ss = StandardScaler()

ss.fit(X_train)

X_train = ss.transform(X_train)

X_test = ss.transform(X_test)

X = df.drop(columns=['name','status'],axis=1)

Y = df['status']

print (X)



//classification

from sklearn.ensemble import RandomForestClassifier
```

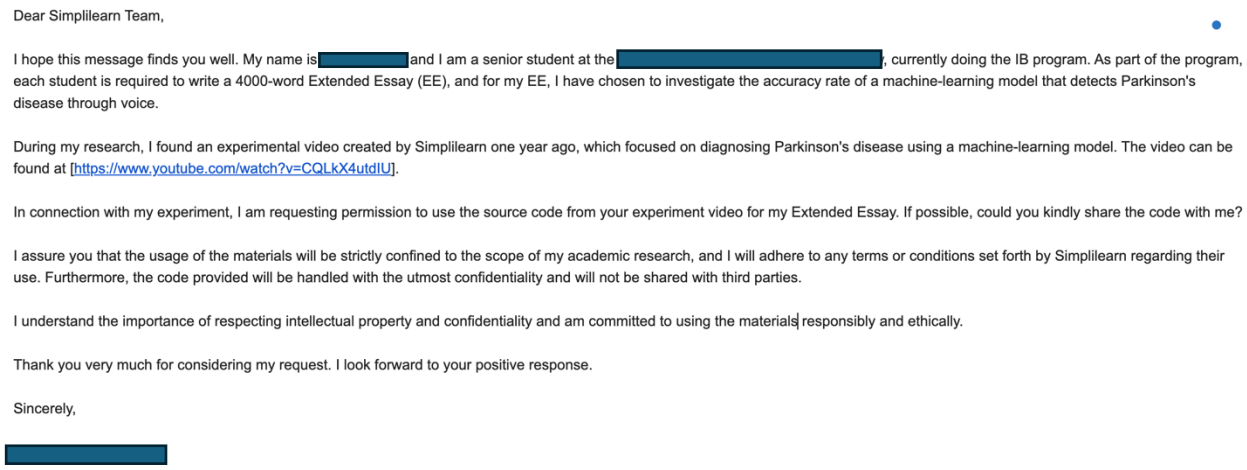
```
from sklearn.preprocessing import StandardScaler

# Assuming X_train, Y_train, X_test are defined before this point

# Standardize the features
ss = StandardScaler()
ss.fit(X_train)
X_train = ss.transform(X_train)
X_test = ss.transform(X_test)

# Create a Random Forest classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the Random Forest model
model.fit(X_train, Y_train)
input_data =
(157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0.01109,0.04374,0.42600,0.02182,0.
03130,0.02971,0.06545,0.02211,21.03300,1,0.414783,0.815285,-
4.813031,0.266482,2.301442,0.284654)
input_data_np = np.asarray (input_data)
input_data_re = input_data_np.reshapre(1, -1)
s_data = ss.transform(input_data_re)
pred = model.predict(s_data)

print(pred)
```

```
if (pred[0]==0):
    print("Negative, No Parkinson's Found")
else:
    print("Positive, Parkinson's Found")
```

[18]

Appendix 3: Permission Letter for the Source Code

3. A: Permission Email was sent to Simplilearn company



Dear Simplilearn Team,

I hope this message finds you well. My name is [redacted] and I am a senior student at the [redacted], currently doing the IB program. As part of the program, each student is required to write a 4000-word Extended Essay (EE), and for my EE, I have chosen to investigate the accuracy rate of a machine-learning model that detects Parkinson's disease through voice.

During my research, I found an experimental video created by Simplilearn one year ago, which focused on diagnosing Parkinson's disease using a machine-learning model. The video can be found at [https://www.youtube.com/watch?v=CQLkX4utdIU].

In connection with my experiment, I am requesting permission to use the source code from your experiment video for my Extended Essay. If possible, could you kindly share the code with me?

I assure you that the usage of the materials will be strictly confined to the scope of my academic research, and I will adhere to any terms or conditions set forth by Simplilearn regarding their use. Furthermore, the code provided will be handled with the utmost confidentiality and will not be shared with third parties.

I understand the importance of respecting intellectual property and confidentiality and am committed to using the materials responsibly and ethically.

Thank you very much for considering my request. I look forward to your positive response.

Sincerely,

[redacted]

[18] YouTube. (2022a, September 25). *Parkinson's disease detection using machine learning | machine learning projects 2022 | simplilearn*. YouTube. https://www.youtube.com/watch?v=CQLkX4utdIU&t=518s

3. B: Replied Email from the Simplilearn Machine-learning Instruction Team

Dear [redacted]

Certainly! We are pleased to provide you with the source code relevant to your research paper. Good Luck!

The Simplilearn Machine Learning Instruction Team

Provided the necessary code I requested…

3. C: A Permission Email was sent to Deepranjan

Dear DeepranjanG,

I trust this message finds you well. My name is [redacted], and I am a senior student at the [redacted] currently enrolled in the IB program. As part of the program requirements, each student is tasked with writing a 4000-word Extended Essay (EE). For my EE, I have chosen to investigate the accuracy rate of a machine-learning model that detects Parkinson's disease through voice.

During my research, I came across voice data on your GitHub repository. In connection with my experiment, I am seeking your permission to utilize the voice data for my Extended Essay.

Recognizing the importance of respecting intellectual property and confidentiality, I am committed to using the materials responsibly and ethically.

Thank you very much for considering my request. I eagerly await your positive response.

Sincerely,

[redacted]

3. D: Replied Email from Deepranjan

Dear [redacted]

Absolutely. Good luck with your experiment!

Regards,

Deepranjan Kumar Gupta