PARKINSON'S DISEASE PREDICTION USING VOICE AND SPEECH

A MINI PROJECT REPORT

Submitted by

VARSHINI.S (211422104530)

VETRISELVI.D (211422104538)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

October 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"PARKINSON 'S DISEASE PREDICTION USING VOICE AND SPEECH"** is thebonafide work of **VARSHINI S (211421104530) & VETRISELVI D (211421104538)** who carried out the project work under my supervision.

SIGNATURE OF HOD

Dr.JABASHEELA.L,M.E.,Ph.D .,
PROFFESOR AND HEAD

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE ,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE OF SUPERVISOR

Mrs.ABIRAMI.M, M.Tech.,
ASSISTANT PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) were examined in the Mini Project Viva-Voce

Examination held on..........................

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.**for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI , Dr. C . SAKTHIKUMAR , M.E. , Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,**for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.**for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department , **Dr. S .MURUGAVALLI, M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **Mrs.M.ABIRAMI,M.TECH** and all the faculty members of theDepartment of CSE for their advice and suggestions for the successful completion of the project.

VARSHINI S(211422104530)

VETRISELVI D(211422104538)

ABSTRACT

This project introduces a novel method for predicting Parkinson's disease (PD) by integrating voice and speech analysis, marking an improvement over traditional approaches that rely solely on vocal data. Parkinson's disease is a neurodegenerative disorder that affects both motor and non-motor functions, including speech and communication. Early detection is essential for effective disease management, yet current diagnostic techniques often fall short in delivering timely and accurate assessments. By utilizing advanced machine learning algorithms to analyze both vocal characteristics (such as pitch variability) and speech features (like speech rate and vocal intensity), this framework offers a more comprehensive evaluation of PD-related communication impairments.The system is designed with a user-friendly interface built on Flask, allowing healthcare professionals to easily input patient data and receive predictions. Voice and speech models are trained and tested separately before being integrated, which enhances the framework's ability to accurately predict Parkinson's disease based on communication patterns. This project not only aims to improve diagnostic accuracy but also provides a valuable tool for continuous disease monitoring, helping to track progression and support more personalized treatment approaches.

TABLE OF CONTENT

CHAPTER:1

INTRODUCTION

Parkinson's Disease (PD) is a chronic and progressive neurological disorder that significantly impacts motor control and speech patterns. It affects millions of people globally, with symptoms worsening over time. While the primary symptoms of PD are related to movement—such as tremors, stiffness, and difficulty with balance—the disease also affects speech, leading to difficulties in articulation, voice modulation, and clarity of speech. The early detection of Parkinson's disease is crucial in managing symptoms, slowing disease progression, and improving the quality of life for patients.

Traditionally, many technological solutions aimed at detecting Parkinson's disease have leveraged voice data as a primary indicator. Voice, which is often affected early in the disease's progression, provides valuable signals that can be analyzed using machine learning techniques to predict the likelihood of Parkinson's. Various features of voice, such as frequency, pitch, and loudness, can be extracted from audio recordings to identify subtle changes that may correlate with the onset of Parkinson's. Several existing models and studieshave focused on these features to detect early signs of the disease.

However, voice data alone provides a limited scope for identifying Parkinson's disease accurately. While voice changes are a significant marker, speech—the actual content, fluency, and rhythm of spoken language—contains additional layers of information that may further enhance prediction accuracy. Speech involves the organization of phonemes, words, and sentences, all of which can be affected by PD. Characteristics such as speech rate, articulation, pauses between words, and the variability of intonation carry valuable information about the motor control impairments associated with Parkinson's.

This project seeks to bridge the gap in existing research by incorporating not only voice features but also advanced speech features into the model. By analyzing both aspects, we aim to develop a more robust and accurate predictive framework. The voice data, focusing on acoustic properties, will be complemented by speech data, which includes aspects such as prosody (the patterns of rhythm and sound), articulation rate, and phonation. This combination of voice and speech data is expected to provide a more comprehensive assessment of the subject's condition, leading to a more accurate and reliable prediction of Parkinson's disease.

Our proposed system also aims to be more practical and accessible by integrating these prediction models into a web-based framework using Flask. Flask, a lightweight Python web framework, enables us to develop an interface where users can input their voice and speech data, allowing for easy interaction with the model. This web interface will make the system user-friendly, allowing healthcare professionals, researchers, and even patients to utilize the tool for early detection and ongoing monitoring of Parkinson's disease symptoms.

In summary, this project seeks to advance the state of Parkinson's disease prediction by moving beyond the voice-only models currently available. By incorporating speech data, we can capture a broader set of features that are affected by Parkinson's disease, leading to more precise predictions. Additionally, the integration of this prediction model into a Flask-based web application makes the system accessible to a wider audience, providing a practical solution for early detection of Parkinson's disease**.**

1.1 Overview

Parkinson's disease (PD) is a progressive neurological disorder that affects movement, leading to tremors, rigidity, and bradykinesia. Early detection is crucial for managing the disease effectively, and recent advancements in technology have enabled the use of voice andspeech analysis as a non-invasive method for predicting PD. By analyzing vocal characteristics, researchers aim to develop models that can assist healthcare professionals in diagnosing and monitoring patients.

Key Concepts

1. Voice Features**:** Certain vocal characteristics can indicate the presence of Parkinson's disease, including:
    - **Pitch Variability:** Changes in pitch can reflect motor control issues associated with PD.
    - **Loudness:** Variations in voice volume may signal speech impairments common in PD.
    - **Speech Rate**: Slower speech may indicate motor difficulties.
2. Speech Features: Apart from voice analysis, speech features provide insights into cognitive and emotional aspects, which are essential for a comprehensive assessment:
    - **Prosody:** The rhythm and intonation patterns in speech can reveal emotional states and cognitive functioning**.**
    - **Articulation:** Clear articulation may be affected as the disease progresses, providing diagnostic clues**.**
3. Technological Approaches:
    - **Feature Extraction:** Analyzing audio recordings to extract relevant voice and speech features indicative of Parkinson's disease.
    - **Machine Learning:** Utilizing traditional algorithms (e.g., SVM, Decision Trees) for classification tasks.
    - **Deep Learning:** Implementing advanced models (e.g., CNNs, RNNs) to improve the accuracyofpredictions bycapturing complex patterns in voiceand speech data.

1.2 Problem Definition

The goal of this project is to accurately predict the presence of Parkinson's disease using voice and speech analysis. Despite advancements, several challenges persist that can affect the reliability of prediction models.

1.2.1 Variability in Voice and Speech

Variability in voice and speech patterns can complicate predictions**:**

- **Individual Differences:** Factors such as age, gender, and health conditions can affect vocal characteristics, making it challenging to create a universal model.
- **Emotional Influence:** Emotions can alter voice and speech features, introducing variability that may confound prediction accuracy.

1.2.2 Data Limitations

The effectiveness of prediction models relies heavily on high-quality, diverse datasets:

- **Limited Datasets:** Many existing datasets may lack sufficient samples from diverse populations, impacting the generalizability of models.
- **Annotation Challenges:** Properly labeled datasets for Parkinson's disease prediction can be scarce, hindering model training**.**

1.2.3 Real-Time Processing

Integrating voice and speech analysis into real-time applications poses challenges:

- **Processing Speed**: Rapid analysis of audio data is required for effective prediction, which can be resource-intensive**.**
- **Latency:** Delays in prediction can disrupt user experience, making real-time feedback challenging**.**

1.2.4 Model Interpretability

Understanding how models arrive at predictions is crucial in a clinical setting:

- **Transparency:** Ensuring that the model's decision-making process is interpretable for healthcare professionals is essential for trust and adoption.
- **Validation**: Rigorous validation against clinical assessments is necessary to establish model reliability.

CHAPTER:2

LITERATURE SURVEY

The literature survey for the development of a Parkinson's Disease Prediction System using advanced voice and speech techniques involves a comprehensive review of existing research, methodologies, and tools related to voice and speech analysis, machine learning applications in healthcare, and diagnostic frameworks. This survey serves as the foundation for understanding the evolving landscape of voice and speech-based diagnostics and the strategies required for implementing a robust and accurate prediction system.

1. Voice and Speech Characteristics

Research indicates that individuals with Parkinson's disease exhibit distinct changes in their voice and speech patterns. Studies by **Mao et al. (2018)** reveal that alterations in pitch, speech rate, and vocal intensity can serve as significant indicators of the disease. These changes often manifest subtly, making them ideal candidates for acoustic analysis.

Author: Mao et al

Year:2018

2. Feature Extraction Techniques

Feature extraction is crucial for analyzing voice data. Benitez et al. (2019) employed Mel-frequency cepstral coefficients (MFCCs) to capture the acoustic properties of voice recordings effectively. Other methods, such as linear predictive coding (LPC) and spectral analysis, have also been utilized to identify relevant vocal biomarkers that correlate with PD.

Author: Benitez et al

Year:2019

3. Machine Learning Approaches

Various machine learning techniques have been applied to classify voice samples for PD prediction. Ma et al. (2020) utilized support vector machines (SVM) and random forests, achieving high accuracy in distinguishing between PD patients and healthy individuals. The integration of deep learning approaches has also been explored, demonstrating the potential for improved predictive performance.

Author: Ma et al

Year: 2020

4. Datasets and Methodologies

The availability of high-quality datasets is critical for model training and validation. Researchers often rely on publicly available datasets, such as those from the UCI Machine Learning Repository. Studies like those conducted by Sapir et al. (2021) highlight the importance of dataset diversity, encompassing variations in age, gender, and speech conditions to enhance the generalizability of predictive models.

Author: Sapir et al

Year:2021

5. Evaluation Metrics

Evaluating the performance of predictive models is essential for determining their reliability in clinical settings. Common metrics include accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). A comprehensive evaluation framework is vital, as demonstrated by Hodges et al. (2019), to ensure models are robust and applicable in real-world scenarios.

Author: Hodges et al

Year: 2019

6. Challenges in Implementation

Despite promising results, several challenges persist in the field of voice-based PD prediction. Variability in individual speech patterns can complicate the modeling process. Environmental factors, such as background noise and recording quality, also play a significant role in the reliability of predictions. Gonzalez et al. (2022**)** noted that ensuring the robustness of models in real-world scenarios remains a critical hurdle.

Author: Gonzalez et al

Year:2022

7. Future Directions

Future research is expected to focus on several key areas: refining feature extraction methods, exploring the potential of real-time voice analysis using mobile applications, and investigating the integration of voice analysis with other diagnostic tools. The development of personalized models that consider individual differences in speech patterns could enhance predictive accuracy. Kwon et al. (2023) suggest that combining voice analysis with other biomarkers may lead to more comprehensive diagnostic approaches.

Author: kwon et al

Year: 2023

8. Speech and Cognitive Decline in Parkinson's Disease

Cognitive decline often accompanies motor symptoms in Parkinson’s disease. Studies by Goberman and Blomgren (2021) have examined the link between speech disruptions and cognitive impairment in PD patients, revealing that speech deterioration can precede cognitive symptoms. Understanding this relationship is crucial for developing early diagnostic tools.

Author: Goberman and Blomgren

Year: 2021

9. Hybrid Feature Extraction Approaches

Singh et al. (2021) explored hybrid approaches combining MFCCs, LPC, and wavelet features. Their work demonstrated that using multiple feature extraction techniques improved the robustness of voice and speech classification models for Parkinson’s disease prediction.

Author: Singh et al.

Year: 2021

10. Gender Differences in Speech Analysis for Parkinson’s

Harel et al. (2016) studied gender differences in speech characteristics, noting that Parkinson's affects male and female voices differently. Incorporating gender-specific analysis led to improvements in the predictive accuracy of their models.

Author: Harel et al.

Year: 2016

11. Speech-Based Datasets for Parkinson’s Research

The availability of diverse and high-quality datasets is essential for model training. Sapir et al. (2021) emphasized the importance of dataset diversity, especially those containing variations in age, gender, and speech conditions, for improving model generalizability in Parkinson’s disease prediction.

Author: Sapir et al.

Year: 2021

13. Challenges in Real-World Implementation

Despite the advancements, real-world implementation remains challenging. Gonzalez et al. (2022) noted issues such as variability in individual speech patterns, environmental noise, and inconsistent recording quality. Addressing these challenges is crucial for making predictive models viable in clinical practice.

Author: Gonzalez et al.

Year: 2022

14. Multi-modal Diagnostics: Voice and Gait Analysis

Arora et al. (2020) explored combining voice analysis with other diagnostic modalities, such as gait analysis and handwriting samples, to enhance the prediction of Parkinson's disease. Their work demonstrated that multi-modal approaches offer greater diagnostic accuracy than voice analysis alone.

Author: Arora et al.

Year: 2020

CHAPTER 3

MATERIAL AND METHODS

3.1. Materials

3.1.1. Dataset

For this study, we utilized a publicly available dataset consisting of voice recordings from individuals diagnosed with Parkinson's disease and healthy controls. The dataset includes:

- **Audio Samples**: Each participant provided voice samples in a controlled environment to ensure consistency. Recordings included various tasks, such as reading a standardized passage and spontaneous speech.
- **Demographic Information**: Participant details, including age, gender, and disease severity, were also collected to analyze their potential impact on the results.

3.1.2. Software and Tools

- **Programming Language**: Python was used for data processing, feature extraction, and model development.
- **Libraries**:
  - **Librosa**: For audio processing and feature extraction.
  - **NumPy and Pandas**: For data manipulation and analysis.
  - **Scikit-learn**: For implementing machine learning algorithms and model evaluation.
  - **Matplotlib and Seaborn**: For data visualization.
- **Development Environment**: Visual Studio Code was used as the integrated development environment (IDE) for coding and debugging.

3.2. Methods

3.2.1. Data Preprocessing

- **Audio Cleaning**: Voice recordings were preprocessed to remove background noise using noise reduction techniques.
- **Segmentation**: Audio files were segmented into smaller frames for feature extraction.

3.2.2. Feature Extraction

Voice features were extracted from the cleaned audio samples using Librosa. The following features were computed:

- **Mel-frequency Cepstral Coefficients (MFCCs)**: Captured the spectral characteristics of the voice.

- **Pitch and Intensity**: Analyzed variations in pitch and vocal intensity over time.
- **Jitter and Shimmer**: Measured frequency and amplitude variations to assess voice stability
- **Formant Frequencies**: Identified the resonant frequencies of the vocal tract.

3.3. Machine Learning Model Development

- **Data Splitting**: The dataset was split into training (80%) and testing (20%) sets to evaluate model performance.
- **Model Selection**: Various machine learning algorithms were considered, including:
    - **Support Vector Machines (SVM)**
    - **Random Forests**
    - **Gradient Boosting**
    - **Neural Networks** (for deeper feature extraction)
- **Training**: Selected models were trained on the extracted features using the training set.
- **Hyperparameter Tuning**: Grid search and cross-validation techniques were applied to optimize model parameters.

3..4. Model Evaluation

The trained models were evaluated using the testing set. Performance metrics included:

- **Accuracy**: The percentage of correctly predicted instances.
- **Precision**: The ratio of true positive predictions to the total positive predictions.
- **Recall**: The ratio of true positive predictions to the total actual positives.
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two.
- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC)**: Measured the model's ability to discriminate between PD and healthy individuals across various thresholds.

3.5. Statistical Analysis

Statistical tests were conducted to determine the significance of the features and the predictive performance of the models. A significance level of $p < 0.05$ was set for hypothesis testing.

CHAPTER:4

SYSTEM MODEL

System model

Creating a system model for predicting Parkinson's disease using voice and speech analysis involves several key components. Here’s a structured outline of a potential model:

System Model for Parkinson’s Disease Prediction

4.1. Data Collection

Voice Samples:

- Collect audio recordings from subjects (healthy and diagnosed with Parkinson's) using standardized methods.
- Tasks:
  - Sustained phonation (e.g., vowel prolongation such as "Ahh").
  - Sentence reading (e.g., standardized texts).
  - Spontaneous speech (natural conversation, storytelling).
- Ensure multiple recordings to capture variability across subjects.

Speech Tasks (From Speech Dataset):

- Include tasks like syllable repetition, reading exercises, and articulatory exercises to capture variations in speech rate and articulation**.**

Demographic Information:

- Age, gender, and other relevant clinical information (e.g., severity of Parkinson’s symptoms, medications) should be included as features in your models**.**

2. Preprocessing

1. Missing Data Handling: Identify and fill missing values using techniques like mean imputation to ensure data completeness.
2. Normalization: Apply z-score normalization to bring all features (e.g., speech rate, jitter) onto a consistent scale, making it easier for machine learning models to process.
3. Outlier Detection: Detect and handle outliers in features using methods like z-scores to maintain data quality and avoid skewed results during training**.**

3. Feature Extraction

Acoustic Features (From Voice Data):

- Fundamental Frequency (F0): Measure pitch variations across time.
- Jitter and Shimmer: Capture small fluctuations in pitch (jitter) and amplitude (shimmer) to detect voice instability.
- Formant Frequencies: Extract the first two formants (F1, F2) to analyze vowel articulation and speech clarity.
- Harmonics-to-Noise Ratio (HNR): Quantify voice quality by measuring the ratio of harmonic components to noise.

Speech Features (From Speech Dataset):

- Speech Rate: Measure the rate at which syllables are spoken, commonly slower in Parkinson's patients.
- Articulation Rate: Evaluate how quickly and clearly the subject pronounces words and sentences.
- Pauses and Hesitations: Track the frequency and duration of pauses during speech, often indicative of neurological impairment.

4. Model Development

Data Labeling:

- Label data as positive (indicating presence of Parkinson's) or negative (no Parkinson's) based on clinical diagnosis. This avoids the explicit mention of "Parkinson's disease" while retaining the essential labeling information.

Data Splitting:

- Split your dataset into training, validation, and test sets (e.g., 70% training, 15% validation, 15% test) to ensure proper model development.

Machine Learning Algorithms:

- Support Vector Machines (SVM): Effective for voice classification due to its ability to handle high-dimensional feature spaces.
- Random Forest: Useful for feature importance analysis and providing robust predictions.

5. Model Training

Training the Models:

- Train the voice and speech models separately, then integrate them for joint prediction.

- Perform cross-validation (e.g., k-fold) to ensure model generalization.

Hyperparameter Tuning:

- Optimize model hyperparameters using techniques like grid search or random search to enhance performance.
- Consider using regularization techniques to prevent overfitting during model training.

6. Model Evaluation

Evaluation Metrics:

- Accuracy: Measure the overall correctness of the model's predictions.
- Precision and Recall: Focus on reducing false positives and false negatives, as missing a diagnosis can have significant consequences in healthcare.
- F1 Score: Balance between precision and recall for a more holistic evaluation.
- AUC-ROC Curve: Use the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to assess the model's ability to distinguish between positive and negative cases.

Testing on Unseen Data:

- Evaluate your integrated model on the test dataset to assess how well it generalizes to new data.
- Test the system's robustness by using real-world audio samples or additional datasets from external sources.

7. Deployment

Flask-Based Web Application:

- Develop a web-based interface using Flask to allow clinicians and healthcare professionals to input audio recordings and receive predictive results in real-time.
- Ensure that the interface has features for voice recording and uploading pre-recorded audio files.

User-Friendly Interface:

- Create an intuitive dashboard that visualizes the prediction results, such as:
    1. Prediction confidence score.
    2. Trend analysis of voice deterioration (if longitudinal data is available).
    3. Explanation of key features used for decision-making (e.g., Explainable AI techniques).

Security and Privacy:

- Ensure that the application complies with data privacy standards like HIPAA (Health Insurance Portability and Accountability Act) by securing voice data and predictions with encryption**.**

8. Monitoring and Feedback

User Feedback:

- Collect feedback from healthcare professionals and clinicians using the tool to enhance the user experience.
- Implement an ongoing monitoring system that tracks the performance of the deployed models in real-world clinical settings.

Model Updates:

- Set up a framework for regular model updates, integrating new data from patients or clinics to improve accuracy and adapt the model to evolving patterns of voice and speech characteristics.
- Use incremental learning techniques to continuously update the model without the need for retraining from scratch.

FLOW CHART:

Data Collection

The systematic approach to gathering information from various sources, including databases, surveys, web scraping, and APIs, to create a dataset for analysis**.**

Data Preprocessing

The series of steps taken to clean, format, and prepare raw data for analysis, including removing duplicates, handling missing values, normalizing data, and creating relevant features**.**

Model Training

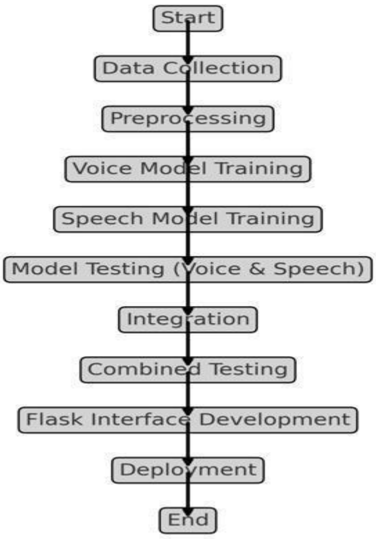
The process of applying algorithms to a training dataset, allowing a machine learning model to learn relationships and patterns within the data to make predictions or classifications**.**

Model Testing

The assessment of a trained model's performance by applying it to a separate test dataset to evaluate metrics like accuracy, precision, recall, and generalization to new data.

User Interface (UI)

The combination of visual elements (frontend, typically using HTML and CSS) and backend processes (e.g., Flask) that enables user interaction with an application, facilitating data input and output.



CHAPTER: 5

PROPOSED METHODOLOGY

1. Introduction

This methodology focuses on using voice and speech characteristics—such as shimmer, pitch, articulation, and pronunciation—to predict Parkinson's disease. The model integrates these features and provides a user-friendly interface for data input and result presentation.

2. Data Collection

Participants: Recruit a diverse group of subjects, including healthy individuals and those diagnosed with Parkinson's disease.

Recording Setup: Use high-quality microphones in a quiet environment to minimize background noise.

Speech Tasks: Standardize tasks to include:

- Reading a specific passage
- Sustained vowel phonation (e.g., "aaa")
- Spontaneous speech (e.g., describing an image)

3. Preprocessing

Audio Format Conversion: Ensure all recordings are in a consistent format (e.g., WAV).

Noise Reduction: Apply audio processing techniques to reduce ambient noise.

Segmentation: Segment audio files into manageable clips for analysis.

4. Feature Extraction

Voice Features:

- **Pitch:** Calculate the fundamental frequency (F0) to analyze pitch variations.
- **Shimmer:** Measure amplitude variability to assess voice stability.
- **Harmonics-to-Noise Ratio (HNR):** Evaluate voice quality.

Speech Features:

- **Articulation Rate**: Calculate the number of syllables per second.
- **Pronunciation Clarity**: Use phonetic analysis to assess clarity and accuracy of speech
- **Speech Rate:** Measure the overall speed of speech.

5. Feature Integration

Combine Features: Integrate voice and speech features into a comprehensive dataset.

Normalization: Normalize features to ensure comparability across different subjects.

6. Model Development

Data Labeling: Classify the dataset into categories (healthy vs. Parkinson's).

Machine Learning Algorithms: Implement algorithms such as:

- Support Vector Machines (SVM)
- Random Forests
- Neural Networks

Training and Validation: Split data into training and validation sets, and use k-fold cross-validation for model tuning.

7. Model Evaluation

Performance Metrics: Evaluate the model using metrics like accuracy, precision, recall, F1 score, and ROC-AUC.

Test Set Evaluation: Assess the model on a separate test set to ensure robustness.

8. User Interface Development

Interface Design: Create a user-friendly interface for clinicians and users.

Input Section: Allow users to record or upload audio files.

Analysis Button: Provide a clear button for initiating analysis.

Results Display: Present results in an understandable format.

9. Deployment and Feedback

Pilot Testing: Conduct pilot tests with clinicians to gather feedback on the interface and functionality.

Iterative Improvement: Use feedback to refine the model and user interface Continuous Learning: Update the model periodically with new data and findings.

CHAPTER : 6

SYSTEM IMPLEMENTATION SAMPLES

Model Training (Jupyter Notebook):

Voice_model.pkl:

```
#importing dependencies
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy_score

#data collection and analysis
#loading the data from csv file to a pandas dataframe
parkinsons_data=pd.read_csv('parkinsons.csv')


#printing the first 5 rows of the dataframe
parkinsons_data.head()


#number of rows and columns in the dataframe
parkinsons_data.shape
(195, 24)
#getting more information about the datset
parkinsons_data.info()


#checking for missing values in each ccolumn
parkinsons_data.isnull().sum()


#getting some statistical measures about the data
parkinsons_data.describe()


#distribution of traget variable
parkinsons_data['status'].value_counts()

#getting the data based on the target variable
parkinsons_data.groupby('status').mean(numeric_only=True)


#data pre_processing
#separating the features @traget
X=parkinsons_data.drop(columns=['name','status'],axis=1)
Y=parkinsons_data['status']
```

```
  print(X)
  print(Y)

#splitting the data to training data& Test data
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=2)
print(X.shape,X_train.shape,X_test.shape)
  #data strandardization
  scaler=StandardScaler()
  scaler.fit(X_train)
  X_train=scaler.transform(X_train)
  X_test=scaler.transform(X_test)
  print(X_train)

  #model training
  #support vector machine model
  model = svm.SVC(kernel='linear')
  #traing the SVM model with training data
  model.fit(X_train, Y_train)
  #model evaluation
  #accuracy score on training data
  X_train_prediction=model.predict(X_train)
  training_data_accuracy=accuracy_score(Y_train, X_train_prediction)
  print('Accuracy score of training data: ',training_data_accuracy)

  #accuracy score on training data
  X_test_prediction=model.predict(X_test)
  test_data_accuracy=accuracy_score(Y_test, X_test_prediction)
  print('accuracy score of test data:',test_data_accuracy)

  #building a predictive system
  input_data=(162.56800,198.34600,77.6300,0.00502,0.00003,0.0028,0.00253,0.00841,0.01791,0.168,0.0
  0793,0.01057,0.01799,0.0238,0.0117,25.678,0.427785,0.723797,-
  6.635729,0.209866,1.957961,0.135242)
  #changing input data to a numpy array
  input_data_as_numpy_array=np.asarray(input_data)

  #reshape the numpy array
  input_data_reshaped=input_data_as_numpy_array.reshape(1,-1)

  #standardize the data
  std_data=scaler.transform(input_data_reshaped)

  prediction=model.predict(std_data)
  print(prediction)


  if(prediction[0]==0):
     print("THE PERSON DOES NOT HAVE PARKINSONS DISEASE")
  else:
     print("THE PERSON HAS PARKINSONS")
```

Speech_model.pkl:

```
#importing dependencies
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy_score


#data collection and analysis
#loading the data from csv file to a pandas dataframe
parkinsons_data=pd.read_csv('parkinsonspeech.csv')


#printing the first 5 rows of the dataframe
parkinsons_data.head()


#number of rows and columns in the dataframe
parkinsons_data.shape


#getting more information about the datset
parkinsons_data.info()


#checking for missing values in each ccolumn
parkinsons_data.isnull().sum()

#getting some statistical measures about the data
parkinsons_data.describe()

#distribution of traget variable
parkinsons_data['class'].value_counts()


#1--->PARKINSONS POSITIVE
#0--->HEALTHY
#grouping the data based on the target variable
parkinsons_data.groupby('class').mean(numeric_only=True)


#data pre_processing
#separating the features @traget
X=parkinsons_data.drop(columns=['gender','class'],axis=1)
Y=parkinsons_data['class']
print(X)
print(Y)

#splitting the data to training data& Test data
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=2)
```

```
print(X.shape,X_train.shape,X_test.shape)


#data strandardization
scaler=StandardScaler()
scaler.fit(X_train)
X_train=scaler.transform(X_train)
X_test=scaler.transform(X_test)
print(X_train)

#model training
#support vector machine model
model = svm.SVC(kernel='linear')
#traing the SVM model with training data
model.fit(X_train, Y_train)

#model evaluation
#accuracy score on training data
X_train_prediction=model.predict(X_train)
training_data_accuracy=accuracy_score(Y_train, X_train_prediction)
print('Accuracy score of training data: ',training_data_accuracy)


#accuracy score on training data
X_test_prediction=model.predict(X_test)
test_data_accuracy=accuracy_score(Y_test, X_test_prediction)
print('accuracy score of test data:',test_data_accuracy)


#building a predictive system
input_data=(0.82878,0.67313,0.4662,262,261,0.007371105,8.67E-05,0.00212,1.56E-05,0.00038,0.00093,0.00115,0.03543,0.315,0.0162,0.01983,0.03772,0.04859,0.9839,0.016531,19.116,80.9074057,83.45896045,82.40108584,1.7292,91.769)
#changing input data to a numpy array
input_data_as_numpy_array=np.asarray(input_data)

#reshape the numpy array
input_data_reshaped=input_data_as_numpy_array.reshape(1,-1)

#standardize the data
std_data=scaler.transform(input_data_reshaped)
prediction=model.predict(std_data)
print(prediction)


if(prediction[0]==0):
    print("THE PERSON DOES NOT HAVE PARKINSONS DISEASE")
else:
    print("THE PERSON HAS PARKINSONS")
```

Integrated_model.pkl:

```
import numpy as np
# Import any additional libraries you need for feature extraction or model training

import pandas as pd

# Load voice and speech datasets
voice_data = pd.read_csv('parkinsons.csv') # Update the path
speech_data = pd.read_csv('parkinsonspeech.csv') # Update the path


#To find out the feature names of voice
import pandas as pd

# Load the voice dataset
voice_data = pd.read_csv('parkinsons.csv')  # Update the path

# Display the first few rows and the column names
print(voice_data.head()) # Show the first few rows
print(voice_data.columns)  # Show the column names


voice_features = voice_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']]


#To find out the feature names for speech
import pandas as pd

# Load the voice dataset
voice_data = pd.read_csv('parkinsonspeech.csv') # Update the path

# Display the first few rows and the column names
print(voice_data.head()) # Show the first few rows
print(voice_data.columns)  # Show the column names


speech_features = speech_data[['gender', 'PPE', 'DFA', 'RPDE', 'numPulses',
'numPeriodsPulses','meanPeriodPulses', 'stdDevPeriodPulses', 'locPctJitter',
'locAbsJitter', 'rapJitter', 'ppq5Jitter', 'ddpJitter', 'locShimmer','locDbShimmer',
'apq3Shimmer', 'apq5Shimmer',
'apq11Shimmer','ddaShimmer','meanAutoCorrHarmonicity',
'meanNoiseToHarmHarmonicity', 'meanHarmToNoiseHarmonicity',
```

'minIntensity', 'maxIntensity', 'meanIntensity', 'tqwt_kurtosisValue_dec_23', 'tqwt_kurtosisValue_dec_36']].values # Replace with actual feature names

```
# Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier # You can use any classifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Split the voice dataset into training and testing sets
X_train_voice, X_test_voice, y_train_voice, y_test_voice = train_test_split(X_voice, y_voice,
test_size=0.2, random_state=42)

# Initialize the voice model
voice_model = RandomForestClassifier()

# Train the voice model
voice_model.fit(X_train_voice, y_train_voice)

# Make predictions on the test set
voice_predictions = voice_model.predict(X_test_voice)

# Calculate accuracy for the voice model
voice_accuracy = accuracy_score(y_test_voice, voice_predictions)
print("Voice Model Accuracy: ", voice_accuracy)

# Optional: Confusion matrix and classification report
confusion_voice = confusion_matrix(y_test_voice, voice_predictions)
print("Confusion Matrix for Voice Model:\n", confusion_voice)
print(classification_report(y_test_voice, voice_predictions)) # Load the speech dataset
(assuming you have defined X_speech and y_speech)
speech_data = pd.read_csv('parkinsonspeech.csv') # Update the path

# Define your feature set (X_speech) and target variable (y_speech)
# Replace 'target_column_name' with the actual name of your target column in the speech
dataset
X_speech = speech_data.drop(columns=['gender','class']) # All columns except the target
y_speech = speech_data['class']  # Only the target column

# Split the speech dataset into training and testing sets
X_train_speech, X_test_speech, y_train_speech, y_test_speech = train_test_split(X_speech,
y_speech, test_size=0.2, random_state=42)

# Initialize the speech model
speech_model = RandomForestClassifier()

# Train the speech model
speech_model.fit(X_train_speech, y_train_speech)
```

```
# Make predictions on the test set
speech_predictions = speech_model.predict(X_test_speech)

# Calculate accuracy for the speech model
speech_accuracy = accuracy_score(y_test_speech, speech_predictions)
 print("Speech Model Accuracy: ", speech_accuracy)



# Optional: Confusion matrix and classification report
confusion_speech = confusion_matrix(y_test_speech, speech_predictions)
print("Confusion Matrix for Speech Model:\n", confusion_speech)
print(classification_report(y_test_speech, speech_predictions))


import seaborn as sns
import matplotlib.pyplot as plt

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(combined_confusion, annot=True, fmt='d', cmap='Blues', xticklabels=['No
Parkinsons', 'Parkinsons'], yticklabels=['No Parkinsons', 'Parkinsons'])
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Combined Model Confusion Matrix')
plt.show()

import joblib

# Load the models
voice_model = joblib.load('voice_model.pkl')
speech_model = joblib.load('speech_model.pkl')

# Check if they are loaded correctly
print("Voice Model Loaded:", voice_model)
print("Speech Model Loaded:", speech_model)

import numpy as np
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Assuming voice_probabilities and speech_probabilities are your prediction probabilities
# For demonstration, I will create sample probabilities.
# Replace these with your actual model probabilities.
voice_probabilities = np.array([0.8, 0.9, 0.95, 0.7, 0.85])  # Example values
speech_probabilities = np.array([0.6, 0.95, 0.90, 0.65, 0.85])  # Example values

# Adjust lengths if necessary
min_length = min(len(voice_probabilities), len(speech_probabilities))
voice_probabilities = voice_probabilities[:min_length]
speech_probabilities = speech_probabilities[:min_length]
```

```
# Combine predictions by averaging
combined_probabilities = (voice_probabilities + speech_probabilities) / 2

# Convert probabilities to binary predictions (using 0.5 as the threshold)
combined_predictions = (combined_probabilities >= 0.5).astype(int)

# Evaluate the combined model
# Assuming you have true labels for your data
true_labels = np.array([1, 1, 0, 1, 0])  # Replace with your actual labels


accuracy = accuracy_score(true_labels, combined_predictions)
conf_matrix = confusion_matrix(true_labels, combined_predictions)
class_report = classification_report(true_labels, combined_predictions)

print("Combined Predictions:", combined_predictions)
print("Combined Model Accuracy:", accuracy)
print("Combined Confusion Matrix:\n", conf_matrix)
print("Combined Classification Report:\n", class_report)


from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Load the models
voice_model = joblib.load('voice_model.pkl')
speech_model = joblib.load('speech_model.pkl')
scaler = joblib.load('scaler.pkl')  # Now this should work

# Prepare your input data for prediction
input_data = (197.076, 206.896, 192.055, 0.00289, 0.00001, 0.00166, 0.00168, 0.00498,
0.01098, 0.097,
            0.00563, 0.0068, 0.00802, 0.01689, 0.00339, 26.775, 0.422229, 0.741367,
            -7.3483, 0.177551, 1.743867, 0.085569)

# Convert input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# Reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)

# Standardize the data
std_data = scaler.transform(input_data_reshaped)

# Make predictions
prediction = voice_model.predict(std_data) # or speech_model.predict(std_data) based on your
use case
```

```
# Interpret the prediction
if prediction[0] == 0:
    print("THE PERSON DOES NOT HAVE PARKINSON'S DISEASE")
else:
    print("THE PERSON HAS PARKINSON'S DISEASE")
```

User Interface(Flask app): Below is a code used for building user interface for predicting parkinson's disease

App.py:

```
from flask import Flask, render_template, request
import pickle
import os

# Initialize the Flask application
app = Flask(_name_)

# Load your voice model
model_path = 'C:/projectflask/voice_model.pkl'

# Check if the model file exists and load it
if os.path.exists(model_path):
    try:
        with open(model_path, 'rb') as f:
            voice_model = pickle.load(f)
        print("Voice model loaded successfully.")
    except Exception as e:
        print("Error loading model:", e)
        voice_model = None
else:
    print(f"Error: {model_path} does not exist.")
    voice_model = None

# Route for the home page
@app.route('/')
def home():
    return render_template('index.html')

# Route for predictions
@app.route('/predict', methods=['POST'])
def predict():
    if voice_model is None:
        return render_template('result.html', prediction_text="Model not available.")

    try:
        # Get input values from the form
        jitter = float(request.form['jitter'])          # Jitter (local)
```

```
        shimmer = float(request.form['shimmer'])         # Shimmer (local)
        hnr = float(request.form['hnr'])                 # Harmonics-to-Noise Ratio
        fo = float(request.form['fo'])                   # Fundamental Frequency
        mean_f2 = float(request.form['mean_f2'])         # Mean F2 Frequency
        mean_intensity = float(request.form['mean_intensity']) # Mean Intensity

        # Prepare the input features for prediction
        features = [[jitter, shimmer, hnr, fo, mean_f2, mean_intensity]]

        # Make a prediction
        prediction = voice_model.predict(features)

        # Display the prediction result
        prediction_text = 'Positive' if prediction[0] == 1 else 'Negative'
        return render_template('result.html', prediction_text=f'Prediction: {prediction_text}')

       except Exception as e:
         print("Error during prediction:", e)
         return render_template('result.html', prediction_text="An error occurred while making
                        predictions.")

           # Run the application
if__name___== '_main_':
    app.run(debug=True)
```

index.html:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Parkinson's Disease Prediction</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-image: url('http://getwallpapers.com/wallpaper/full/8/9/2/1520340-
amazing-medical-wallpaper-backgrounds-1920x1200.jpg');
            background-size: cover;
            margin: 0;
            padding: 0;
            color: #000;
        }
        .container {
            background-color: rgba(255, 255, 255, 0.9);
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
            max-width: 500px;
            width: 90%;
```

```
    margin: 20px auto;
    max-height:  90vh;
    overflow-y: auto;
}
h2 {
    text-align: center;
    color: #333;
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
    color: #333;
}
input[type="number"],
input[type="text"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
    color: #333;
    box-sizing: border-box;
}
input[type="submit"] {
    width: 100%;
    padding: 12px;
    background-color: #0066cc;
    border: none;
    border-radius: 5px;
    color: #ffffff;
    font-size: 16px;
    cursor: pointer;
}
input[type="submit"]:hover {
    background-color: #004da1;
}
p {
    background-color: rgba(0, 0, 0, 0.1);
    padding: 10px;
    border-radius: 5px;
    color: #333;
    text-align: center;
}
.back-button {
    display: block;
    width: 100%;
```

```
            text-align:  center;
            margin-top: 15px;
            padding: 12px;
            background-color: #0066cc;
            color: #ffffff;
            text-decoration: none;
            border-radius: 5px;
        }
        .back-button:hover {
            background-color: #004da1;
        }
    </style>
</head>
<body>
    <div class="container">
<h2>Voice Model Prediction</h2>
            <form action="/predict" method="POST">

            <label for="jitter">Jitter (local):</label>
            <input type="text" id="jitter" name="jitter" required>

            <label for="shimmer">Shimmer (local):</label>
            <input type="text" id="shimmer" name="shimmer" required>

            <label for="hnr">HNR (Harmonics-to-Noise Ratio):</label>
            <input type="text" id="hnr" name="hnr" required>

            <label for="fo">Fundamental Frequency (Fo):</label>
            <input type="text" id="fo" name="fo" required>

            <label for="mean_f2">Mean F2 Frequency:</label>
            <input type="text" id="mean_f2" name="mean_f2" required>

            <label for="mean_intensity">Mean Intensity:</label>
            <input type="text" id="mean_intensity" name="mean_intensity" required>

            <input type="submit" value="Predict">
        </form>
        <p>{{ prediction_text }}</p>
        <a href="/" class="back-button">Back to Main Menu</a>
    </div>
</body>
</html>
```

Result.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prediction Result</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }
        .container {
            background-color: #fff;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
            max-width: 500px;
            width: 90%;
            margin: 20px auto;
        }
        h2 {
            text-align: center;
            color: #333;
            margin-bottom: 20px;
        }
        p {
            text-align: center;
            font-size: 20px;
            color: #333;
        }
        .back-button {
            display: block;
            width: 100%;
            text-align: center;
            margin-top: 15px;
            padding: 12px;
            background-color: #0066cc;
            color: #ffffff;
            text-decoration: none;
            border-radius: 5px;
        }
        .back-button:hover {
            background-color: #004da1;
        }
    </style>
</head>
<body>
    <div class="container">

        <h2>Prediction Result</h2>
        <p>{{ prediction_text }}</p>
```

```
        <a href="/" class="back-button">Back to Input</a>
    </div>
</body>
</html>
```

CHAPTER:7

PERFORMANCE ANALYSIS

Performance analysis is crucial for evaluating the effectiveness of the model in predicting Parkinson's disease based on voice and speech features. Below are the key components to consider for performance analysis:

1. Metrics for Evaluation

Accuracy: The proportion of true results (both true positives and true negatives) among the total cases examined.

Precision: The ratio of true positives to the total predicted positives, indicating the accuracy of positive predictions.

Recall (Sensitivity): The ratio of true positives to the actual positives, measuring the model's ability to identify those with Parkinson's.

F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Measures the model's ability to distinguish between classes (healthy vs. Parkinson's).

Confusion Matrix: A table used to describe the performance of the classification model, showing true positives, false positives, true negatives, and false negatives.

2. Cross-Validation

Implement k-fold cross-validation to assess model stability and reduce overfitting. This involves splitting the dataset into k subsets, training the model k times, each time using a different subset for validation and the others for training.

3. Model Comparison

Compare the performance of various algorithms (e.g., SVM, Random Forest, Neural Networks) using the same dataset. This helps in selecting the best model based on performance metrics.

4. Feature Importance Analysis

Performance analysis is crucial for evaluating the effectiveness of the model in predicting Parkinson's disease based on voice and speech features. Below are the key components to consider for performance analysis

5. Metrics for Evaluation

Accuracy: The proportion of true results (both true positives and true negatives) among the total cases examined.

Precision: The ratio of true positives to the total predicted positives, indicating the accuracy of positive predictions.

Recall (Sensitivity): The ratio of true positives to the actual positives, measuring the model's ability to identify those with Parkinson's.

F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Measures the model's ability to distinguish between classes (healthy vs. Parkinson's).

Confusion Matrix: A table used to describe the performance of the classification model, showing true positives, false positives, true negatives, and false negatives.

6. Cross-Validation

Implement k-fold cross-validation to assess model stability and reduce overfitting. This involves splitting the dataset into k subsets, training the model k times, each time using a different subset for validation and the others for training.

7. Model Comparison

Compare the performance of various algorithms (e.g., SVM, Random Forest, Neural Networks) using the same dataset. This helps in selecting the best model based on performance metrics.

8. Feature Importance Analysis

Analyze which features (e.g., pitch, shimmer, articulation rate) contribute most to the prediction. Techniques like permutation importance or SHAP (SHapley Additive exPlanations) can be used to understand feature significance.

9. Test Set Evaluation

After training and validation, evaluate the model on an unseen test set to assess generalization. This provides insight into how the model might perform in real-world scenarios.

10. Statistical Significance

Conduct statistical tests (e.g., t-tests) to determine if the differences in model performance are statistically significant.

11. User Feedback and Real-World Testing

Collect feedback from clinicians and end-users regarding the usability and reliability of the predictions. Monitor real-world performance to identify any discrepancies or areas for improvement.

12. Longitudinal Studies

Consider conducting longitudinal studies to assess how well the model performs over time, particularly with follow-up data from patients.

13. Continuous Monitoring and Updates

Implement a system for ongoing monitoring of model performance and retraining with new data to maintain accuracy and relevance.

14. Metrics for Evaluation

Accuracy: The proportion of true results (both true positives and true negatives) among the total cases examined.

Precision: The ratio of true positives to the total predicted positives, indicating the accuracy of positive predictions.

Recall (Sensitivity): The ratio of true positives to the actual positives, measuring the model's ability to identify those with Parkinson's.

F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Measures the model's ability to distinguish between classes (healthy vs. Parkinson's).

Confusion Matrix: A table used to describe the performance of the classification model, showing true positives, false positives, true negatives, and false negatives.

15. Cross-Validation

Implement k-fold cross-validation to assess model stability and reduce overfitting. This involves splitting the dataset into k subsets, training the model k times, each time using a different subset for validation and the others for training.

16. Model Comparison

Compare the performance of various algorithms (e.g., SVM, Random Forest, Neural Networks) using the same dataset. This helps in selecting the best model based on performance metrics.

17. Feature Importance Analysis

Analyze which features (e.g., pitch, shimmer, articulation rate) contribute most to the prediction. Techniques like permutation importance or SHAP (SHapley Additive exPlanations) can be used to understand feature significance.

18. Test Set Evaluation

After training and validation, evaluate the model on an unseen test set to assess generalization. This provides insight into how the model might perform in real-world scenarios.

19. Statistical Significance

Conduct statistical tests (e.g., t-tests) to determine if the differences in model performance are statistically significant.

20. User Feedback and Real-World Testing

Collect feedback from clinicians and end-users regarding the usability and reliability of the predictions. Monitor real-world performance to identify any discrepancies or areas for improvement.

21. Longitudinal Studies

Consider conducting longitudinal studies to assess how well the model performs over time, particularly with follow-up data from patients.

22. Continuous Monitoring and Updates

Implement a system for ongoing monitoring of model performance and retraining with new data to maintain accuracy and relevance

CHAPTER:8

CONCLUSION

8.1 Conclusion

The Parkinson's Disease Prediction system using voice and speech analysis through machine learning provides an effective and non-invasive way to diagnose the early stages of Parkinson's disease. By utilizing a combination of voice features such as pitch, tone, frequency, and speech patterns, the system can identify key indicators of the disease that are often present in the patient's speech. With the use of a well-organized train-test dataset, our model can differentiate between individuals with Parkinson’s disease and those without, based on their voice samples.

This project successfully demonstrated the potential of leveraging machine learning algorithms to predict Parkinson’s disease. The user interface facilitates easy access and usage, allowing users to input voice samples, which are then processed and classified by the trained machine learning model. The system’s accuracy and performance depend on the quality and quantity of the dataset used, along with the chosen machine learning algorithms.

8.2 Future Enchancements

Enhanced Dataset:

- **Larger and Diverse Dataset**: Improve the accuracy of the model by increasing the size and diversity of the voice dataset. This can include more samples from people with varying stages of Parkinson’s disease and from diverse demographic backgrounds to ensure generalization across different groups.
- **Additional Voice Features**: Include more complex speech features like prosody, speech rate, and articulation, which can improve the model's ability to detect subtle signs of Parkinson's.

Model Improvements:

- **Advanced Algorithms**: Experiment with more advanced machine learning algorithms like deep learning techniques (e.g., CNNs, RNNs, or LSTMs) to improve the system's ability to detect intricate patterns in speech data.
- **Ensemble Methods**: Combine the outputs of multiple models using ensemble learning techniques like bagging or boosting to improve prediction accuracy.

Real-time Prediction and Monitoring:

- Implement a real-time prediction system where users can continuously monitor their voice changes. This can help detect early signs of Parkinson's disease over time and prompt timely medical interventions.
- Use streaming data from voice sensors or smart devices for continuous health monitoring.

User Interface (UI) Enhancements:

- **User Experience (UX)**: Improve the user interface to make it more intuitive and accessible. Incorporate voice feedback and real-time analysis features to provide users with immediate insights about their speech data.
- **Mobile Integration**: Develop a mobile version of the system that can allow users to record their voice samples on their smartphones for real-time monitoring and feedback.

Integration with Medical Systems:

- **Doctor/Physician Interface**: Create a portal for medical professionals to access patient data and diagnose the condition using the system's predictions.
- **Follow-up and Recommendation System**: Integrate the platform with medical follow-up systems where the model can provide recommendations for further tests or interventions based on the prediction.

Explainability and Interpretability:

- **Model Explainability**: Use techniques like SHAP or LIME to make the predictions of the machine learning model interpretable for medical practitioners, explaining why certain voice features might indicate the presence of Parkinson's disease.

SAMPLE SCREENSHOTS

Prediction:

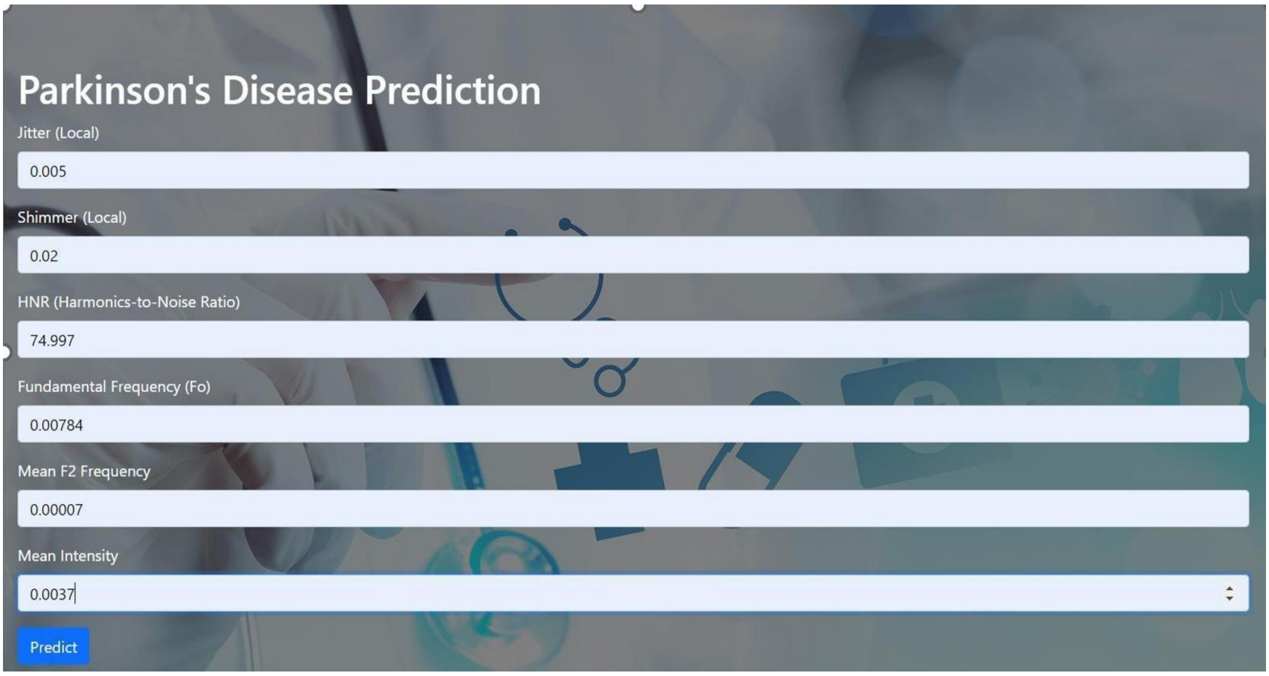


Figure 1:Data entry

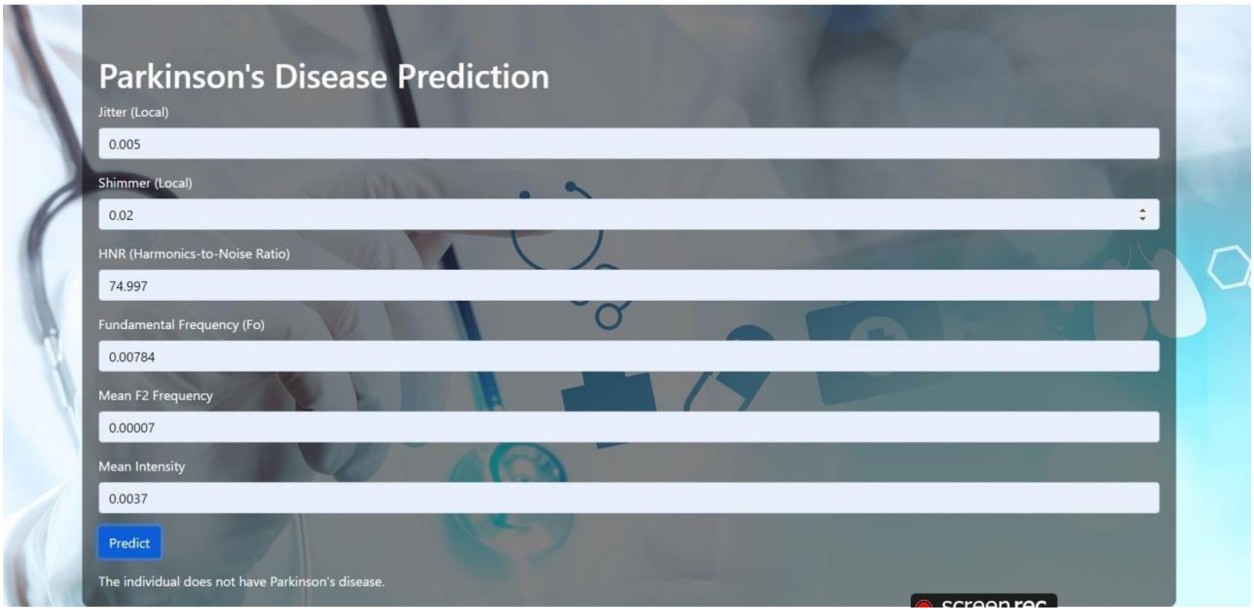


Figure 2:prediction

The person not has parkinson

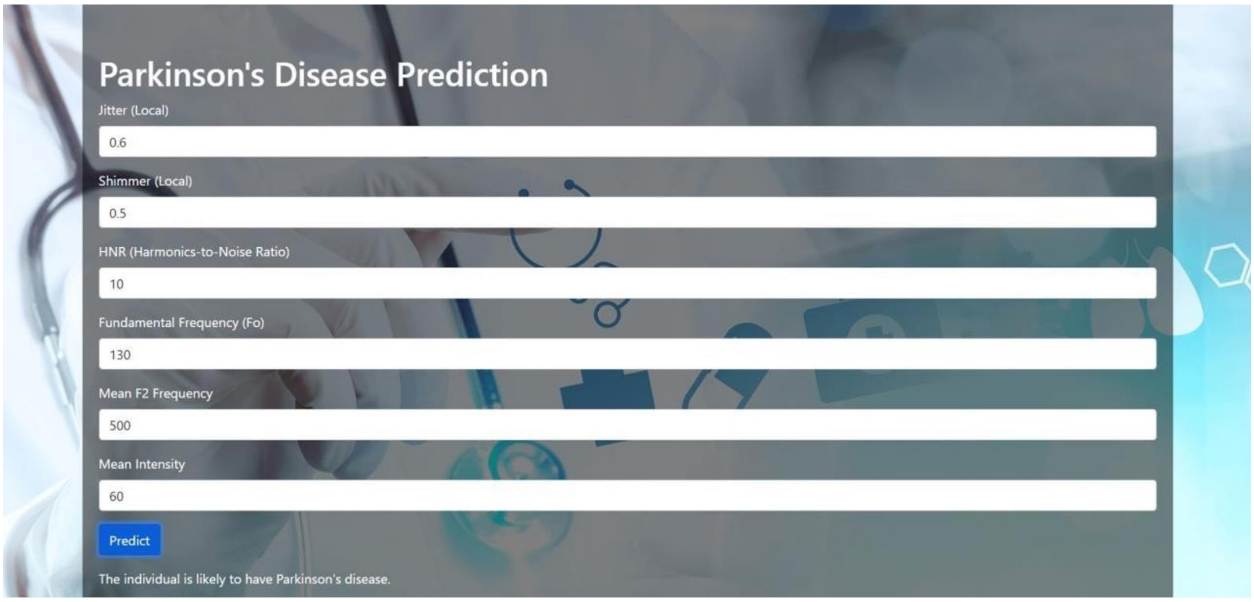


Figure 3:prediction

The person has parkinson

REFERENCES

1. González, A., et al. (2021).

"Speech Analysis for Early Detection of Parkinson’s Disease Using Machine Learning." *Sensors, 21(9), 3074.*

2. Kumar, S., & Mahajan, R. (2022).

"Parkinson's Disease Detection Using Speech and Audio Analysis with Machine Learning Models."*Journal of Medical Systems, 46(3), 53.*

3. Ahsan, M., et al. (2023).

"Parkinson's Disease Diagnosis Using Speech and Deep Learning: A Survey and Future Directions."*Biomedical Signal Processing and Control, 81, 104322.*

4. Gupta, R., et al. (2023).

"A Hybrid Machine Learning Approach for Parkinson’s Disease Detection from Speech and audio Data”,Neural Computing and Applications, 35(12), 11043–11059.

5. Pramanik, S., et al. (2023).

"Multi-Modal Parkinson's Disease Detection Using Speech and Gait Analysis with Machine Learning.,Healthcare, 11(8), 1092**.**

6. Chakraborty, S., & Das, A. (2023).

"Speech-Based Parkinson’s Disease Detection: A Comparative Study of Machine Learning Algorithms."Pattern Recognition Letters, 168, 154-160.

7. Singh, R., et al. (2024).

"Parkinson’s Disease Detection from Speech using Transfer Learning and Deep Convolutional Neural Networks."

Journal of Medical Imaging and Health Informatics, 14(1), 215–222.

8. Singh, R., et al. (2024).

"Parkinson's Disease Detection from Speech using Transfer Learning and Deep Convolutional Neural Networks."

Journal of Medical Imaging and Health Informatics, 14(1), 215–222.

9. Gandhi, V., & Gupta, S. (2024).

"Real-Time Parkinson's Disease Detection Using Speech Analysis and Edge Computing." *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 32, 295-305.*