Early Detection Of Parkinson's Disease Using Machine Learning

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology
in
Computer Science and Engineering

by

Mridul Madnani

20BDS0191

Srijan Singh Somvanshi

20BDS0381

Devansh Bajpai

20BCE0807

Under the guidance of

Prof. Sayan Sikder

School of Computer Science & Engineering

VIT, Vellore.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

May, 2024

DECLARATION

I hereby declare that the thesis entitled "**Early Detection Of Parkinson's Disease Using Machine Learning**" submitted by me, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering* to VIT is a record of bonafide work carried out by me under the supervision of **Sayan Sikder**.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 09/05/2024

Aniruddh Singh Devangan

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "Early Detection Of Parkinson's Disease Using Machine Learning" submitted by **Mridul Madnani (20BDS0191), Srijan Singh Somvanshi (20BDS0381), Devansh Bajpai (20BCE0807), School of Computer Science and Engineering (SCOPE)**, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering,* is a record of bonafide work carried out by him under my supervision during the period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore
Date : 09/05/24

Sayan Sikdar 09.05.24
Signature of the Guide

Internal Examiner

External Examiner

Head of the Department

Computer Science and Engineering

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to all those who have contributed to the successful completion of this thesis on the project titled "Early Detection of Parkinson's Disease using Machine Learning," which served as my university final year capstone project.

First and foremost, I am deeply thankful to my supervisor, Sayan Sikder, for their invaluable guidance, continuous support, and insightful feedback throughout this research endeavor. Their expertise and encouragement played a pivotal role in shaping this project and broadening my understanding of the subject matter.

I am also indebted to the faculty members of the School of Computer Science and Engineering (SCOPE) at Vellore Institute of Technology (VIT) for providing an enriching academic environment and resources essential for conducting this research.

I extend my heartfelt thanks to the participants of this study whose cooperation made the collection of data possible, thus enabling the development and validation of the machine learning models crucial to this project.

Furthermore, I would like to acknowledge the contributions of my classmates and friends who provided moral support and constructive discussions during the course of this project.

Last but not least, I am deeply grateful to my family for their unwavering encouragement, patience, and love throughout my academic journey.

This project would not have been possible without the collective support and encouragement from all these individuals and organizations. Thank you all for being part of this meaningful endeavor.

Student Name

Mridul Madanani
Devansh Bajpai
Srijan Singh Somvanshi

Executive Summary

Parkinson's disease (PD) is a prevalent neurodegenerative disorder characterized by the progressive degeneration of dopaminergic neurons in the substantia nigra region of the brain. Early diagnosis of PD is crucial for effective management and intervention to alleviate symptoms and slow disease progression. Recent advancements in machine learning (ML) techniques offer promising avenues for the early detection of PD, leveraging various data sources such as clinical assessments, imaging scans, and genetic markers. This report highlights the significance of utilizing ML algorithms in the early detection of PD. By analyzing diverse datasets encompassing a range of patient demographics, symptoms, and biomarkers, ML models can identify subtle patterns indicative of PD onset or progression. Moreover, ML algorithms have the potential to overcome the inherent challenges associated with traditional diagnostic methods, including subjectivity in clinical assessments and the complexity of interpreting imaging results.

List of Figures

List of Tables

List of Abbreviations

- **PD:** Parkinson's Disease
- **DF:** Dataframe
- **LR:** Logistic Regression
- **DT:** Decision Tree
- **SVM:** Support Vector Machine
- **RF:** Random Forest
- **XGB:** XGBoost
- **KNN:** K-Nearest Neighbors
- **NB:** Naïve Bayes
- **CLF:** Classifier
- **Pred:** Prediction
- **CM:** Confusion Matrix
- **TP:** True Positives
- **TN:** True Negatives
- **FP:** False Positives
- **FN:** False Negatives
- **AUC:** Area under curve
- **ROC:** Receiver Operating Characteristics

1. INTRODUCTION

Parkinson's disease (PD) is a chronic and progressive neurological disorder that affects millions of people worldwide, particularly those over the age of 60. The disease is characterized by the loss of dopamine-producing neurons in the brain, leading to motor symptoms such as tremors, rigidity, bradykinesia (slowness of movement), and postural instability. Apart from motor symptoms, PD can also cause non-motor symptoms such as cognitive impairment, depression, and sleep disturbances, significantly impacting patients' quality of life.

Early diagnosis of Parkinson's disease is critical for timely intervention and effective management. However, diagnosing PD in its early stages remains challenging due to the subtle onset of symptoms and the overlap with other neurological conditions. Current diagnostic methods primarily rely on clinical assessments, which are subjective and may not detect early-stage PD accurately.

The objective of this project is to develop an advanced machine learning-based system for the early identification of Parkinson's disease using vocal biomarkers and a stacking ensemble technique. By leveraging machine learning algorithms and data analysis techniques, the system aims to enhance diagnostic accuracy and facilitate timely intervention, ultimately improving patient outcomes.

1.1 Objectives

The objectives of the project break down the aim into specific, measurable, and achievable tasks or goals. These objectives serve as the roadmap for the project, guiding the implementation and evaluation process. Each objective should be clear, actionable, and aligned with the overall aim of the project. For example, objectives may include collecting and analyzing relevant data, developing predictive algorithms, evaluating the performance of the model, and validating the results.

- Develop a machine learning model for the early detection of Parkinson’s disease.
- Improve Diagnostic Precision: Enhance accuracy through effective feature selection and engineering.
- Facilitate Timely Intervention: Enable prompt medical management by integrating the model into healthcare systems.

1.2 Motivation

The motivation behind this project stems from the urgent need to enhance early detection methods for Parkinson's disease (PD) using advanced machine learning techniques. Parkinson's disease is a progressive neurodegenerative disorder that affects millions worldwide, with symptoms that can significantly impact patients' quality of life. Early diagnosis is critical for timely intervention and effective management of the disease. However, current diagnostic approaches often rely on subjective clinical assessments, leading to delays in diagnosis and treatment initiation.

Machine learning presents a promising avenue to improve PD diagnosis by leveraging the power of data analytics and predictive modeling. By developing a robust machine learning-based system, we aim to address several critical challenges in PD diagnosis, including the subtle nature of early symptoms, heterogeneity in patient data, and the need for objective and accurate diagnostic tools.

The key motivation behind this project is to:

1. **Improve Patient Outcomes**: Early detection of PD can enable timely interventions that may slow disease progression and improve patient outcomes.
2. **Enhance Diagnostic Accuracy**: By integrating advanced feature selection and ensemble learning techniques, we seek to develop a diagnostic system with higher accuracy and reliability compared to existing methods.
3. **Enable Personalized Medicine**: Machine learning-based diagnostics have the potential to enable personalized treatment plans tailored to individual patient profiles.
4. **Contribute to Healthcare Innovation**: This project contributes to the growing field of healthcare innovation by applying cutting-edge technologies to address real-world medical challenges.

Overall, our motivation is driven by the desire to make a tangible impact on patient care and contribute to the advancement of medical science through the application of machine learning in Parkinson's disease diagnosis. Through this project, we aim to develop an innovative tool that can assist healthcare professionals in making faster and more accurate diagnoses, ultimately improving the lives of individuals affected by Parkinson's disease.

1.3 Background

The background of this project stems from the persistent challenges associated with early detection of Parkinson's disease (PD), a neurodegenerative disorder characterized by motor and non-motor symptoms. Timely diagnosis is critical for effective management and treatment planning, yet current diagnostic methods often lack the sensitivity and specificity needed for accurate detection in the early stages.

Traditional PD diagnosis relies heavily on clinical assessments and subjective evaluations, leading to variability in diagnostic accuracy and delays in treatment initiation. Moreover, the subtle nature of early PD symptoms can result in misdiagnosis or delayed diagnosis, impacting patient outcomes and quality of life.

In recent years, machine learning (ML) has emerged as a promising tool for improving disease detection and diagnosis. ML techniques leverage large datasets to identify patterns and relationships that may not be discernible through conventional methods. Previous research has explored the application of ML algorithms, such as support vector machines (SVM), decision trees, and ensemble methods, for PD diagnosis using diverse data sources ranging from voice recordings to sensor data.

Despite advancements, existing ML-based PD diagnostic systems often face challenges related to feature selection, data heterogeneity, and class imbalance. This project seeks to address these limitations by developing an integrated system that combines feature selection techniques with ensemble learning approaches. By leveraging the strengths of multiple ML algorithms and optimizing model performance, the proposed system aims to enhance the accuracy and reliability of early PD detection.

The project builds upon a foundation of prior research in ML-based healthcare diagnostics and aims to contribute to the growing body of knowledge on PD diagnosis using advanced computational methods. The ultimate goal is to develop a scalable and robust diagnostic tool that can assist clinicians in making accurate and timely diagnoses, ultimately improving patient care and outcomes in the management of Parkinson's disease.

2. PROJECT DESCRIPTION AND GOALS

2.1 Survey on Existing System

The survey on existing systems for Parkinson's disease (PD) diagnosis reveals several insights into current approaches and their limitations. Existing systems predominantly rely on traditional machine learning models such as logistic regression, support vector machines (SVM), decision trees, and random forests for PD detection. These systems often face challenges related to feature selection, data heterogeneity, and class imbalance, which can impact diagnostic accuracy.

Feature selection in traditional PD diagnostic systems is typically based on manual identification of potential biomarkers, such as tremor severity or gait abnormalities, which may not capture the full complexity of PD symptoms. Furthermore, the reliance on individual classifiers limits the ability to leverage diverse information from different data sources effectively.

Moreover, existing systems often struggle with handling data heterogeneity arising from variations in data formats, patient demographics, and disease progression stages. This heterogeneity can introduce bias and affect the generalizability of machine learning models across different populations. Class imbalance, another common issue in PD diagnosis, occurs when the number of positive (PD) cases is significantly lower than negative (non-PD) cases in the dataset. This imbalance can lead to biased model predictions and affect the overall performance of the diagnostic system.

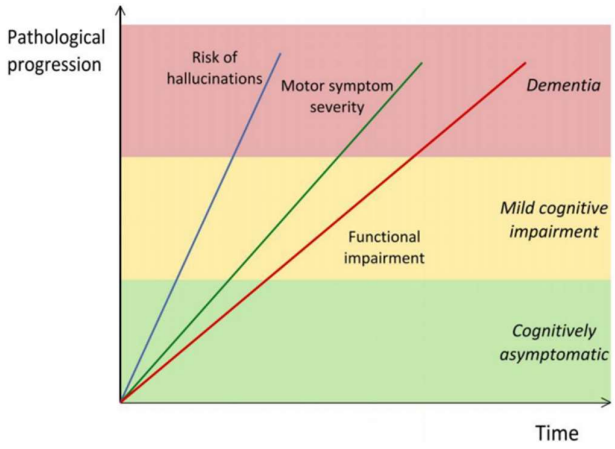


Figure 1: Pathological Progression vs Time

The survey also highlights the limited adoption of ensemble learning techniques in PD diagnosis. Ensemble methods, such as stacking, offer the potential to combine predictions from multiple base models to improve diagnostic accuracy and robustness. However, their application in PD diagnostics remains underexplored.

In summary, the survey underscores the need for an advanced diagnostic system that addresses the limitations of existing approaches by:

- Implementing sophisticated feature selection techniques to identify comprehensive biomarkers of PD.
- Leveraging ensemble learning methods to integrate diverse information and enhance diagnostic accuracy.
- Handling data heterogeneity and class imbalance effectively to improve model generalizability and reliability.

By bridging these gaps, the proposed PD diagnostic system aims to advance the state-of-the-art in machine learning-based healthcare diagnostics and contribute to early and accurate detection of Parkinson's disease.

2.2 **Research Gap**

The existing literature on Parkinson's disease (PD) diagnosis predominantly focuses on individual machine learning classifiers or traditional statistical methods. While these approaches have shown promise, they often encounter challenges related to handling data heterogeneity, feature selection, and class imbalance. Moreover, ensemble learning techniques, which have demonstrated superior performance in various domains, are underutilized or not fully explored in the context of PD diagnosis.

The research gap identified in this project lies in the lack of a unified and comprehensive system that effectively integrates feature selection, ensemble learning, and diverse machine learning algorithms for PD diagnosis. Existing systems often overlook the importance of selecting discriminative features relevant to PD symptoms, which is crucial for improving diagnostic accuracy. Furthermore, most systems do not leverage the collective strength of multiple classifiers to mitigate biases and variance inherent in PD datasets.

By addressing this research gap, the project aims to contribute novel insights into the application of ensemble learning for PD diagnosis. The proposed system will harness the

power of stacking ensemble techniques to combine the predictions of diverse base classifiers, thereby enhancing model robustness and accuracy. Additionally, the project will explore advanced feature selection methods tailored to PD datasets, enabling the identification of key biomarkers and symptom indicators.

2.3 **Problem Statement**

Parkinson's disease (PD) is a progressive neurological disorder characterized by motor and non-motor symptoms. Early detection of PD is crucial for timely intervention and effective disease management. However, existing diagnostic methods often lack accuracy and efficiency, leading to delayed diagnosis and suboptimal patient outcomes.

The problem addressed by this project is the development of an advanced machine learning-based system for early-stage PD detection using a stacking ensemble technique. The key challenges include:

- **Feature Selection**: Identifying informative features from heterogeneous data sources that are indicative of PD symptoms.
- **Ensemble Learning**: Integrating multiple machine learning algorithms to leverage diverse modeling approaches and improve diagnostic accuracy.
- **Class Imbalance**: Handling imbalanced datasets where PD cases are significantly outnumbered by non-PD cases.
- **Model Evaluation**: Implementing rigorous evaluation metrics to assess the performance and generalization ability of the proposed diagnostic system.

The proposed system aims to bridge these gaps by developing a robust and scalable diagnostic tool that can effectively differentiate PD from non-PD cases using machine learning. By leveraging ensemble learning and advanced feature selection techniques, the project seeks to achieve higher diagnostic accuracy and contribute to the early detection and management of Parkinson's disease.

3. TECHNICAL SPECIFICATION

3.1 Requirements

3.1.1 Functional Requirements

The functional requirements of the proposed Parkinson's disease (PD) diagnostic system encompass key features and capabilities necessary for effective disease detection and prediction:

- **Feature Selection**: The system must implement feature selection techniques to identify the most relevant and discriminative features associated with PD symptoms from the input dataset. This involves evaluating and ranking features based on their importance in distinguishing between PD and non-PD instances.
- **Ensemble Learning**: Integration of multiple machine learning algorithms, such as logistic regression, support vector machines (SVM), decision trees, and XGBoost, to create a stacked ensemble model. Each base classifier contributes to the ensemble's predictive power by capturing different aspects of the data, ultimately improving overall diagnostic accuracy.
- **Evaluation Metrics**: The system should employ standard evaluation metrics like accuracy, precision, recall, and F1-score to assess the performance of the developed PD diagnostic model. These metrics provide quantitative measures of the model's ability to correctly classify PD cases while minimizing false positives and false negatives.

3.1.2 Non-Functional Requirements

The non-functional requirements address the quality attributes and constraints that ensure the reliability, scalability, and ethical compliance of the PD diagnostic system:

- **Scalability**: The system must be capable of handling large and diverse datasets efficiently, ensuring that it can process and analyze data from various sources without significant performance degradation.
- **Robustness**: The system should be robust to noisy, incomplete, or inconsistent data commonly encountered in real-world healthcare settings. It must employ data preprocessing techniques to handle data quality issues and ensure model stability.
- **Compliance**: Adherence to data privacy regulations (e.g., GDPR) and ethical

guidelines for AI in healthcare is essential. The system should incorporate mechanisms for data anonymization, encryption, and access control to protect patient privacy and ensure ethical use of sensitive medical data.

- **Interpretability**: Ensuring that the developed PD diagnostic model is interpretable is crucial for gaining insights into the underlying factors contributing to PD diagnosis. Interpretability enhances transparency and trust in the model's decision-making process, facilitating collaboration with healthcare professionals.
- **Performance**: The system should exhibit optimal performance in terms of computational efficiency, response time, and resource utilization, enabling timely and accurate PD diagnosis in clinical settings.

Addressing both functional and non-functional requirements is integral to the successful development and deployment of the PD diagnostic system, ensuring its effectiveness, reliability, and ethical integrity in healthcare applications.

3.2 Feasibility Study

Feasibility study is an essential aspect of project planning that assesses the viability of implementing a proposed system. This section evaluates technical, economic, and social factors to determine whether the project is feasible and sustainable.

3.2.1 Technical Feasibility

Technical feasibility evaluates the project's ability to meet its objectives from a technological perspective. For this Parkinson's disease (PD) detection system, technical feasibility involves:

- **Resource Availability:** Assessing the availability of hardware and software resources required for data collection, preprocessing, model development, and evaluation.
- **Expertise and Skills:** Evaluating the technical skills and expertise available within the project team to implement machine learning algorithms, conduct feature engineering, and manage data effectively.
- **Compatibility:** Ensuring compatibility of selected technologies (e.g., Python, machine learning libraries) with project requirements and constraints.
- **Scalability:** Considering the system's scalability to handle potentially large datasets and accommodate future enhancements or modifications.
- **Performance:** Assessing the performance capabilities of the proposed system, including computational efficiency and model accuracy.

The technical feasibility analysis will identify potential challenges and risks related to technology implementation and provide insights into mitigating strategies to ensure successful project execution.

3.2.2 Economic Feasibility

Economic feasibility evaluates the cost-effectiveness and financial viability of the project. Key considerations for this PD detection system include:

- **Cost Analysis:** Conducting a comprehensive cost analysis, including hardware procurement, software licensing, labor costs, and ongoing maintenance expenses.
- **Return on Investment (ROI):** Estimating the potential return on investment based on the system's benefits, such as improved diagnostic accuracy leading to better patient outcomes and reduced healthcare costs.
- **Budget Constraints:** Evaluating budget constraints and identifying cost-saving measures without compromising system quality.
- **Cost-Benefit Analysis:** Assessing the overall benefits derived from the project against the associated costs to determine its economic feasibility.

The economic feasibility study will provide insights into the financial implications of the project and enable stakeholders to make informed decisions regarding resource allocation and investment.

3.2.3 Social Feasibility

Social feasibility examines the project's impact on stakeholders and society at large. Considerations for this PD detection system include:

- **Ethical Implications:** Addressing ethical considerations related to patient data privacy, consent, and responsible use of AI in healthcare.
- **User Acceptance:** Evaluating user acceptance and adoption of the diagnostic system among healthcare professionals and patients.
- **Impact on Healthcare:** Assessing the potential positive impact of early PD detection on healthcare delivery, patient outcomes, and quality of life.
- **Community Engagement:** Involving relevant stakeholders, including patients, healthcare providers, and regulatory bodies, to ensure social acceptance and support for the project.

The social feasibility analysis will help anticipate and address social challenges and ensure that the project aligns with ethical standards and societal expectations, fostering trust and

acceptance within the community.

3.3 System Specification

3.3.1 Hardware Specification

The hardware specifications required for implementing the Parkinson's disease (PD) diagnostic system involve considerations for computational resources and storage capabilities. Given the complexity of machine learning algorithms and the need to process potentially large datasets efficiently, the system will benefit from a robust hardware setup. This includes:

- **CPU and Memory**: A multi-core processor with sufficient processing power and memory (e.g., at least 8 GB RAM) to handle intensive computations involved in training machine learning models.
- **GPU**: Optionally, a GPU (Graphics Processing Unit) can significantly accelerate model training, especially for deep learning algorithms that benefit from parallel processing.
- **Storage**: Adequate storage capacity to store datasets, trained models, and intermediate results generated during preprocessing and model evaluation phases.
- **Scalability**: The hardware infrastructure should be scalable to accommodate future expansions and increasing computational demands as the system evolves.

3.3.2 Software Specification

The software specifications determine the programming languages, libraries, and frameworks suitable for implementing the PD diagnostic system:

- **Programming Language**: Python will be the primary programming language due to its extensive libraries for data manipulation (e.g., Pandas), numerical computations (e.g., NumPy), and machine learning (e.g., Scikit-learn).
- **Machine Learning Frameworks**: Libraries such as Scikit-learn, TensorFlow, and PyTorch will be utilized for implementing machine learning algorithms and training models.
- **Visualization Tools**: Matplotlib and Seaborn will be employed for generating informative visualizations to aid in data exploration and model evaluation.
- **Development Environment**: Integrated Development Environments (IDEs) like Jupyter Notebook or PyCharm will facilitate code development, experimentation, and

documentation.

- **Version Control**: Git will be used for version control, enabling collaborative development and tracking changes in the codebase.
- **Data Management**: SQL or NoSQL databases may be used for efficient data storage and retrieval, depending on the project's requirements.
- **Deployment**: Docker containers or cloud platforms (e.g., AWS, Google Cloud) will enable seamless deployment and scaling of the diagnostic system.

3.3.3 Standards and Policies

The project will adhere to relevant standards and policies to ensure ethical use and data privacy:

- **Data Privacy**: Compliance with regulations such as General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) to safeguard patient data confidentiality and security.
- **Ethical Guidelines**: Adherence to ethical frameworks outlined by organizations like the American Medical Association (AMA) and World Medical Association (WMA), emphasizing principles such as transparency, fairness, and accountability in AI-based healthcare applications.
- **Model Validation**: Rigorous validation procedures and regulatory approval processes (e.g., from FDA or EMA) will be followed to ensure the safety, efficacy, and accuracy of the diagnostic system before clinical deployment.
- **Interoperability**: Compliance with Health Level Seven International (HL7) standards to facilitate seamless integration with existing healthcare systems and Electronic Health Records (EHRs), promoting interoperability and data exchange.
- **Bias Mitigation**: Implementation of bias detection, mitigation, and fairness-aware model training techniques to prevent discriminatory practices and ensure fair healthcare delivery.

By aligning with these hardware, software, and regulatory specifications, the PD diagnostic system will be developed in a manner that prioritizes performance, reliability, ethical considerations, and compliance with industry standards and policies.

4. DESIGN APPROACH AND DETAILS

4.1 System Architecture

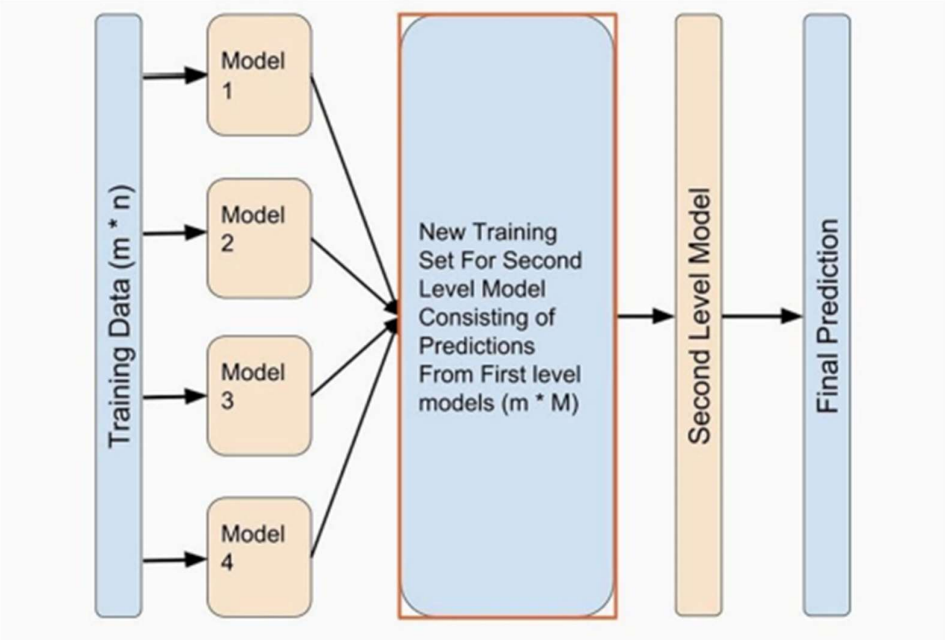


Figure 2: Architecture of Stacking Ensemble Technique

The system architecture for the Parkinson's disease (PD) diagnostic tool involves several key components to enable efficient and accurate disease prediction.

Firstly, the feature selection module employs the Sequential Forward Floating Selection (SFFS) technique to identify the most relevant features indicative of PD symptoms from raw data. This process is critical for improving model performance by focusing on discriminative features.

Next, the classification module utilizes a stacking ensemble technique, combining predictions from multiple base classifiers like logistic regression, support vector machines (SVM), decision trees, and XGBoost. Each base classifier captures different aspects of the data, enhancing overall prediction accuracy.

The evaluation and comparison phase assesses the performance of the ensemble model using metrics like accuracy, precision, recall, and F1-score. By comparing against single classifiers and other ensemble methods, the effectiveness of the proposed model is comprehensively evaluated.

This architecture integrates feature selection, ensemble-based classification, rigorous evaluation, and insightful discussion, resulting in an effective and robust system for early-stage PD detection.

4.2 Design

4.2.1 Data Flow Diagram

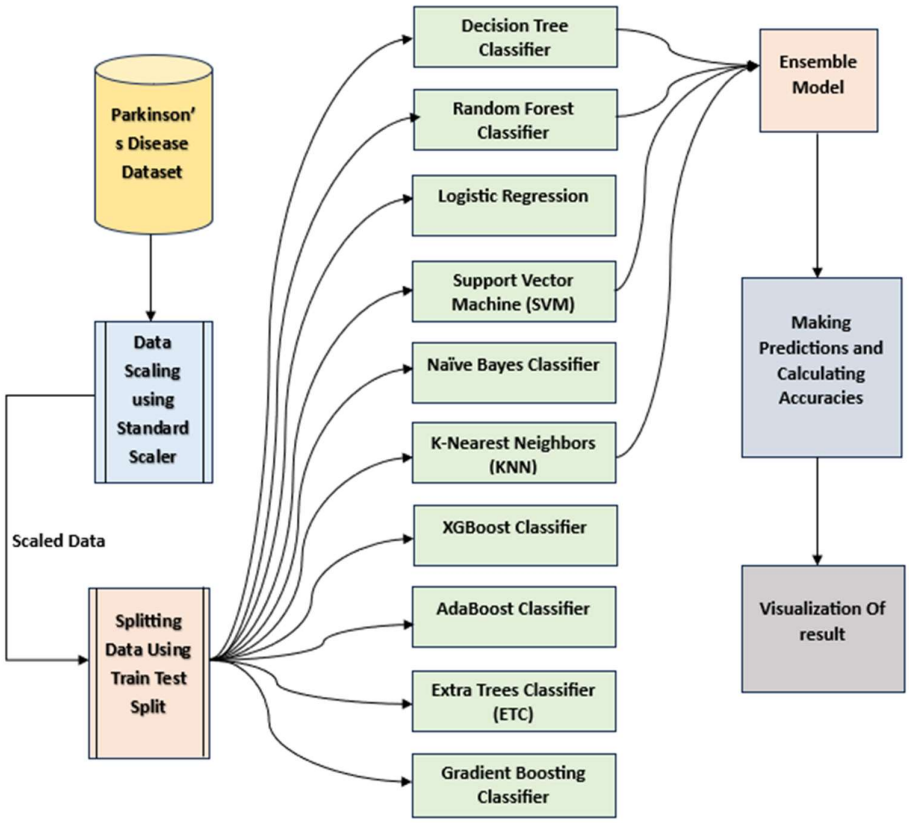


Figure 3: Data Flow Diagram

The Data Flow Diagram (DFD) for the PD diagnostic system illustrates the flow of data through various modules. Initially, raw data related to Parkinson's disease (PD) is acquired from diverse sources, including clinical databases and research studies. This raw data undergoes comprehensive preprocessing, which involves cleaning, filtering, and formatting to ensure data quality and consistency. The preprocessed data then enters the feature engineering phase, where informative features are identified and extracted. Subsequently, the balanced dataset is used to train baseline machine learning models, such as logistic regression, K-nearest neighbors (KNN), and random forest. The trained models, along with their predictions, flow into the model development and training stage, where more sophisticated algorithms like support vector machines (SVM) and XGBoost are employed. The outputs of these models are then compared, and ensemble techniques like stacking are applied to combine their predictions, leading to the final ensemble model. Finally, the ensemble model's performance is evaluated using validation techniques, and the system's overall effectiveness in PD detection is assessed.

4.2.2 Use Case Diagram

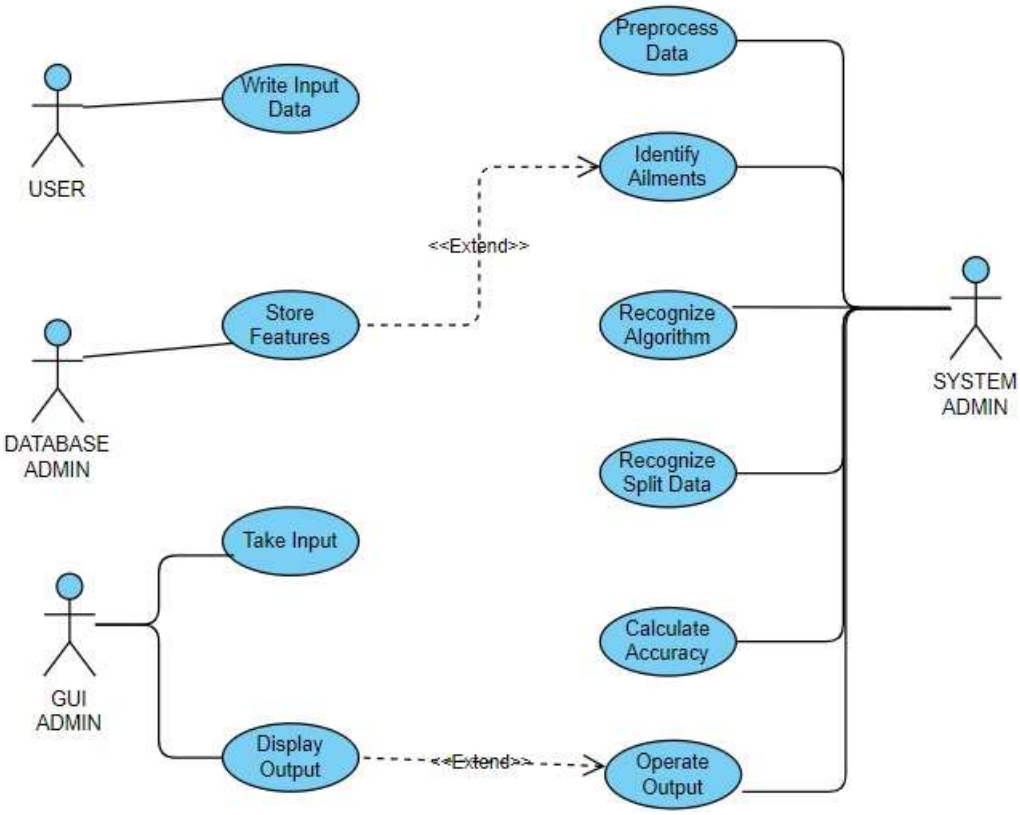


Figure 4: Use Case Diagram

The Use Case Diagram depicts the interactions between the system's users and various system functionalities. In this diagram:

- Data Acquisition: Users interact with the module responsible for acquiring raw data.
- Data Preprocessing: Users engage with the preprocessing module to prepare the data for analysis.
- Feature Engineering: Users participate in the feature engineering process to extract relevant features from the data.
- Run Baseline Models: Users trigger the execution of baseline machine learning models on the pre-processed data.
- Model Training: Users are involved in the training of more sophisticated machine learning models.
- Ensemble Techniques: Users engage in the application of ensemble techniques to combine multiple models.
- Evaluation: Users evaluate the performance of the models against predefined criteria.

4.2.3 Class Diagram

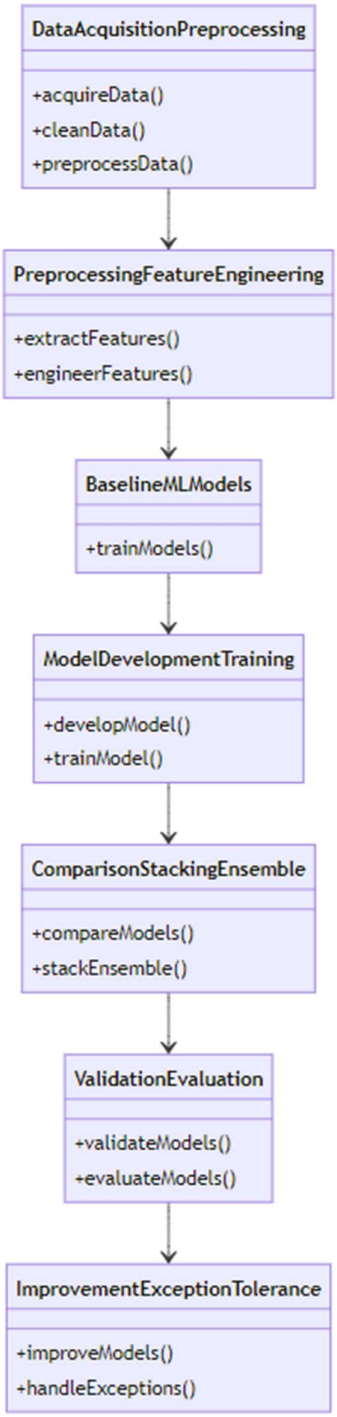


Figure 5: Class Diagram

The Class Diagram provides a structural view of the system by illustrating the classes, attributes, methods, and relationships between them. It outlines the various entities and their interactions within the system. The detailed description of the Class Diagram typically includes:

- **Classes**: Represented as rectangles, classes encapsulate data and behavior relevant to specific entities within the system.
- **Attributes**: Presented within classes, attributes describe the properties or characteristics of the classes.
- **Methods**: Also located within classes, methods define the behaviors or actions that objects of the classes can perform.
- **Relationships**: Shown as lines connecting classes, relationships depict associations, dependencies, inheritance, or aggregation between classes.
- **Multiplicity**: Represented near the ends of relationships, multiplicity indicates the number of instances of one class related to instances of another class.

4.2.4 Sequence Diagram

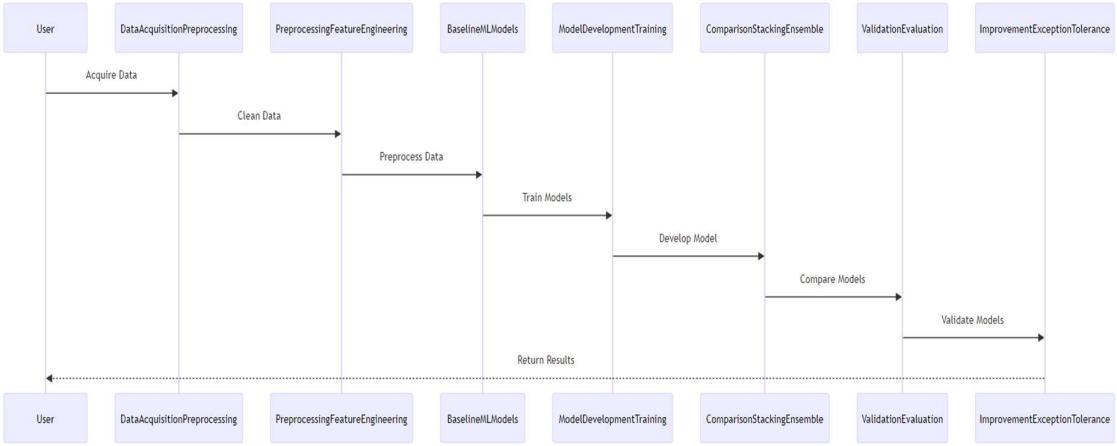


Figure 6: Sequence Diagram

The Sequence Diagram illustrates the interactions between different components or objects in the system over time. It showcases the flow of messages or method calls between these components to accomplish specific tasks. The detailed description of the Sequence Diagram typically includes:

- **Objects**: Represented as vertical lines, objects depict the various components or entities involved in the sequence of interactions.
- **Messages**: Shown as arrows between objects, messages represent the communication or method calls exchanged between objects to achieve specific functionalities.
- **Activation Bars**: Presented as horizontal lines attached to objects, activation bars indicate the duration during which an object is active or processing a message.
- **Lifelines**: Vertical dashed lines surrounding activation bars, lifelines delimit the scope or existence of objects over time.
- **Return Messages**: Arrows with a dashed line indicating the return of control or response from the recipient object to the sender object.
- **Interaction Fragments**: Enclosed within brackets, interaction fragments represent optional or conditional sequences of messages within the sequence diagram.

4.3 Constraints, Alternatives and Tradeoffs

In the development of a machine learning-based system for early Parkinson's disease (PD) detection, several constraints, alternative approaches, and tradeoffs must be considered to ensure the feasibility and effectiveness of the project.

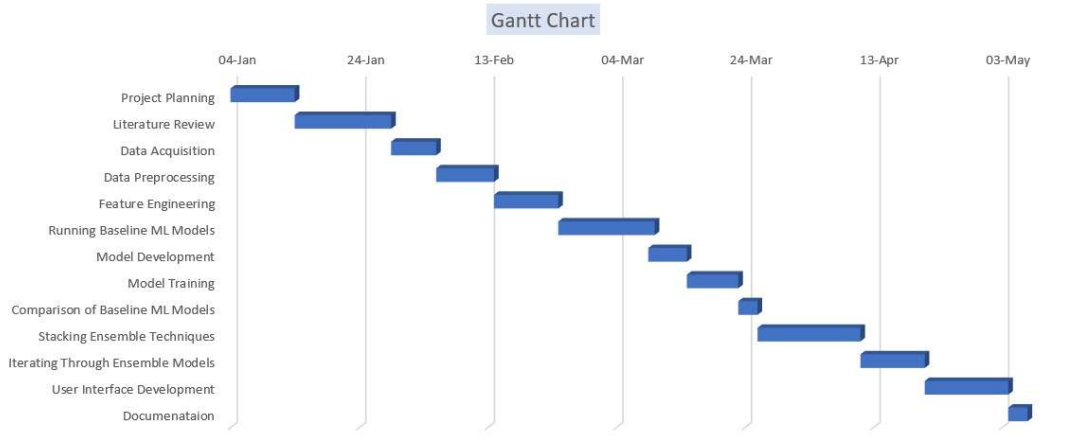
Constraints: The primary constraints include computational resources and time limitations. Machine learning algorithms, especially ensemble methods like stacking, can be computationally intensive, requiring substantial processing power and memory. To address this constraint, alternative strategies such as cloud computing or distributed computing frameworks like Apache Spark may be considered to leverage parallel processing capabilities and reduce training time. Additionally, constraints related to data availability and quality may impact the effectiveness of the model, necessitating robust data preprocessing and quality control measures.

Alternatives: Alternative approaches can be explored in terms of feature selection techniques, ensemble methods, and model architectures. For feature selection, alternative methods such as Recursive Feature Elimination (RFE) or Principal Component Analysis (PCA) could be compared against the Sequential Forward Floating Selection (SFFS) technique to identify the most discriminative features for PD diagnosis. Similarly, alternative ensemble methods like bagging or boosting could be evaluated alongside stacking to determine the optimal approach for combining base classifiers.

Tradeoffs: The project involves several tradeoffs that need careful consideration. For example, increasing the complexity of the ensemble model may improve predictive accuracy but could also lead to decreased model interpretability. Balancing the tradeoff between model performance and interpretability is critical, especially in healthcare applications where transparency and explainability are essential. Another tradeoff relates to data privacy and security versus model performance. By systematically addressing these constraints, exploring alternative approaches, and understanding tradeoffs, the project aims to develop an efficient and reliable machine learning system for early PD detection. Continuous evaluation and optimization throughout the project lifecycle will help mitigate risks associated with constraints and tradeoffs, ultimately delivering a robust diagnostic tool that can assist healthcare professionals in early PD diagnosis and management.

5. SCHEDULE, TASKS AND MILESTONES

5.1 Gantt Chart



	Documenataion	User Interface Development	Iterating Through Ensemble Models	Stacking Ensemble Techniques	Comparison of Baseline ML Models	Model Training	Model Development	Running Baseline ML Models	Feature Engineering	Data Preprocessing	Data Acquisition	Literature Review	Project Planning
Start Date	04-May	21-Apr	11-Apr	26-Mar	23-Mar	15-Mar	09-Mar	24-Feb	14-Feb	05-Feb	29-Jan	14-Jan	04-Jan
Days to Complete	3	13	10	16	3	8	6	15	10	9	7	15	10

Figure 7: Gantt Chart depicting workflow timeline

This project aims to develop machine learning models for Parkinson's disease detection within a structured timeline. The project begins with an extensive literature review (15 days) to understand existing research on Parkinson's disease and machine learning. Relevant datasets are then acquired (8 days) and subjected to preprocessing (7 days) for data cleaning and transformation. Feature engineering (10 days) follows to extract meaningful features from the data.

Baseline machine learning models are trained and evaluated (12 days) to establish initial performance benchmarks. Subsequently, more sophisticated models are developed (15 days) and trained (11 days) using the prepared data. These models are compared (10 days) to identify the most effective ones. Advanced ensemble techniques like stacking (8 days) are implemented to combine model predictions for improved accuracy. Iterative refinement of ensemble models (12 days) further enhances performance.

A user-friendly interface is developed (3 days) for interacting with the final model, and comprehensive documentation (4 days) is compiled to document project details and methodologies.

5.2 Module Description

5.2.1 Module - 1: Data Acquisition and Preprocessing

In this module, the focus is on collecting raw data related to Parkinson's disease (PD) and preparing it for analysis. This includes cleaning, harmonizing, integrating, and ensuring the quality of the data. The subtopics include:

- Data Collection: Identifying relevant datasets from various sources such as clinical databases, research studies, and public repositories.
- Data Cleaning and Harmonization: Removing inconsistencies, errors, and missing values from the dataset.
- Quality Control: Implementing data validation procedures to ensure accuracy and reliability.

Table 1: Dataset Description

Attribute	Purpose
Name	Data is stored in ASCII CSV format where patient name and recording number is stored
MDVP: Fo (Hz)	Fundamental frequency of pitch period
MDVP: Fhi (Hz)	Upper limit of fundamental frequency or maximum threshold of voice modulation
MDVP: Flo (Hz)	Lower limit or minimal vocal fundamental frequency
MDVP: Jitter, Abs, RAP, PPQ, DDP	These are various Kay Pentax's multi-dimensional voice program (MDVP) measures. MDVP is a traditional measure of frequency of vibrations in vocal folds at pitch period to vibrations at start of next cycle called pitch mark [25]
Jitter and Shimmer	Measures of absolute difference between frequencies of each cycle, after normalizing the average
NHR and HNR	Signal to noise and tonal ratio measures, that indicate robustness of environment to noise
Status	0 indicates healthy person while 1 indicates PWP.
D2	Correlation dimension is used to identify dysphonia in speech using fractal objects. It is a nonlinear, dynamic attribute.
RPDE	Recurrence Period Density Entropy quantifies the extent to which signal is periodic
DFA	Detrended Fluctuation Analysis or DFA measures the extent of stochastic self-similarity of noise in speech signals.
PPE	Pitch Period entropy is used to assess abnormal variations in speech on a logarithmic scale
Spread1, spread2	Analysis of extent or range of variations in speech with respect to MDVP: Fo(Hz)

5.2.2 Module - 2: Preprocessing and Feature Engineering

In this module, the focus is on preparing the data for model training by balancing the dataset using SMOTE, performing feature engineering, and conducting visualizations to gain insights into the data. The subtopics include:

Balancing using SMOTE:

- Addressing class imbalance by oversampling the minority class using Synthetic Minority Over-sampling Technique (SMOTE).
- Generating synthetic samples for the minority class to achieve a balanced distribution of target classes.

Feature Engineering:

- Identifying informative features that are relevant to Parkinson's disease diagnosis.
- Creating new features or deriving composite features from existing ones to improve model performance.
- Exploring domain knowledge to engineer features that capture important characteristics of PD.

Feature Selection and Dimensionality Reduction:

- Evaluating feature importance using statistical tests, information gain, or model-based approaches.
- Selecting the most informative features based on their predictive power and relevance to the target variable.
- Implementing dimensionality reduction techniques such as PCA or feature selection algorithms to reduce the number of features while preserving relevant information.

Visualizations:

- Conducting exploratory data analysis (EDA) to understand the distribution and relationships between features.
- Visualizing correlations between features using heatmaps or scatter plots to identify patterns and dependencies.
- Creating box plots or violin plots to compare feature distributions between PD and non-PD groups.
- Generating histograms or density plots to visualize the distribution of individual features and detect outliers or anomalies.
- Plotting interactive visualizations using tools like Plotly or Bokeh for dynamic exploration of the data.

5.2.3 Module - 3: Running Baseline ML Models

This module involves training and evaluating baseline machine learning models using the preprocessed data. The subtopics include:

Logistic Regression:

- Implementing logistic regression to model the probability of PD diagnosis.
- Interpreting model coefficients to understand the relationship between features and the target variable.
- Assessing model performance using metrics such as accuracy, precision, recall, and F1-score.

K-Nearest Neighbors (KNN):

- Implementing the KNN algorithm to classify data points based on their similarity to neighboring instances.
- Tuning hyperparameters such as the number of neighbors (K) to optimize model performance.
- Evaluating the KNN model's accuracy and performance on the dataset.

Naive Bayes:

- Applying the Naive Bayes algorithm, which assumes independence between features, to classify instances.
- Estimating class probabilities using Bayes' theorem and conditional probability distributions.
- Assessing the model's performance and comparing it with other baseline models.

Random Forest:

- Constructing an ensemble of decision trees using the Random Forest algorithm.
- Leveraging bagging and random feature selection to improve model robustness and generalization.
- Evaluating the Random Forest model's accuracy and performance on the dataset.

Decision Tree:

- Building a decision tree model to partition the feature space based on simple decision rules.
- Pruning the decision tree to prevent overfitting and improve generalization.
- Analyzing the decision tree structure and feature importance for insights into PD diagnosis.

Support Vector Machine (SVM):

- Training an SVM classifier to find the optimal hyperplane that separates PD and non-PD instances.

XGBoost:

- Fine-tuning XGBoost hyperparameters such as the learning rate, maximum depth, and regularization parameters.

5.2.4 Module - 4: Model Development and Training

This module involves selecting the most promising machine learning algorithms, fine-tuning their hyperparameters, and training the models on the preprocessed dataset. It encompasses a series of iterative steps aimed at developing robust and accurate predictive models for Parkinson's disease diagnosis.

Model Selection:

- Conducting an extensive evaluation of various machine learning algorithms, including logistic regression, K-nearest neighbors (KNN), naive Bayes, random forest, decision tree, support vector machine (SVM), and XGBoost.
- Assessing the strengths and weaknesses of each algorithm in terms of performance, interpretability, and computational efficiency.
- Choosing the most suitable algorithms based on evaluation metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve.

Hyperparameter Tuning:

- Optimizing the hyperparameters of selected algorithms to improve model performance and generalization.
- Balancing model complexity and performance trade-offs by adjusting hyperparameters such as learning rates, regularization parameters, kernel coefficients, and tree depths.

Model Training:

- Splitting the preprocessed dataset into training, validation, and test sets to facilitate model training and evaluation.
- Employing cross-validation techniques such as k-fold cross-validation or stratified sampling to assess model performance on different subsets of data.
- Training the selected models on the training data while monitoring performance metrics on the validation set to avoid overfitting.

Iterative Optimization:

- Iteratively refining model architectures, hyperparameters, and training strategies based on validation results.
- Experimenting with different feature representations, feature transformations, and data augmentation techniques to enhance model robustness and generalization.
- Analyzing model convergence, learning curves, and error trends to diagnose performance bottlenecks and guide further optimization efforts.

5.2.5 Module - 5: Comparison and Stacking Ensemble Techniques

This module involves comparing the performance of individual baseline models and exploring ensemble techniques to improve predictive accuracy. It aims to leverage the strengths of multiple models and combine their predictions effectively. The subtopics include:
Compare Baseline Model Performance:

- Evaluate the performance metrics (accuracy, precision, recall, F1-score) of each baseline model using cross-validation or holdout validation.
- Analyze the strengths and weaknesses of individual models based on their performance on validation data.
- Consider factors such as computational complexity, interpretability, and scalability when comparing models.

Utilize Stacking Ensemble Techniques:

- Implement stacking ensemble methods to combine predictions from multiple base models into a meta-learner.
- Choose a diverse set of base models (e.g., decision trees, SVM, logistic regression) to capture different aspects of the data and minimize bias.
- Train the meta-learner using predictions from base models as input features and the true labels as the target variable.

Assess Ensemble Performance:

- Evaluate the performance of the ensemble model using appropriate evaluation metrics and validation techniques.
- Compare the performance of the ensemble model with individual baseline models to assess the effectiveness of ensemble learning.
- Analyze the ensemble model's ability to improve predictive accuracy, reduce variance, and handle complex decision boundaries.

Interpret Ensemble Model:

- Interpret the ensemble model to understand how it combines predictions from base models to make final predictions.
- Analyze the importance of individual base models and meta-features in the ensemble model's decision-making process.
- Visualize the ensemble model's decision boundaries and decision-making process to gain insights into its behavior.

5.2.6 Module - 6: Validation and Evaluation

This module focuses on evaluating the performance of the developed models against baseline machine learning algorithms. It involves rigorous validation techniques to assess model accuracy, precision, recall, F1-score, and other relevant metrics. Additionally, the module includes comparison with baseline models to identify the most effective approaches for Parkinson's disease (PD) diagnosis. The subtopics include:

Comparison with Baseline Models:

- Comparing the performance metrics (accuracy, precision, recall, F1-score) of the developed models with baseline algorithms.
- Identifying strengths and weaknesses of each approach based on their performance on validation data.
- Selecting the most effective models for further refinement and deployment in clinical settings.

Performance Visualization:

- Visualizing model performance using graphical representations such as ROC curves, precision-recall curves, and confusion matrices.
- Analyzing trade-offs between different evaluation metrics to understand the overall performance characteristics of the models.
- Presenting performance results in clear and interpretable formats for stakeholders and decision-makers.

Interpretability and Explainability:

- Assessing the interpretability and explainability of the developed models compared to baseline algorithms.
- Investigating the ability of the models to provide insights into the underlying mechanisms and features driving PD diagnosis.
- Enhancing model transparency and trustworthiness by providing explanations for model predictions and decisions.

Model Selection and Refinement:

- Leveraging performance evaluation results to select the most promising models for further refinement and optimization.
- Iteratively adjusting model parameters and feature representations to improve predictive accuracy and generalization.
- Incorporating domain knowledge and expert feedback to fine-tune model architectures and hyperparameters.

5.2.7 Module - 7: User Interface Development

This module focuses on developing a user-friendly web application interface that integrates the optimized ensemble model for early detection of Parkinson's disease. The key features and implementation details include:

Designing Input Interface:

- Creating intuitive input fields for users to enter relevant patient data required for prediction.
- Providing clear guidance and error handling to ensure complete and accurate input submission.

Implementing Prediction Output Display:

- Triggering the machine learning model (optimized ensemble) to generate predictions based on user-inputted data.
- Displaying prediction outcomes prominently on the interface for user reference.

Incorporating Visualizations:

- Integrating visual representations of input data characteristics and model outputs to enhance user understanding.
- Utilizing charts or diagrams to present prediction probabilities or classification results effectively.

Ensuring Responsiveness and Interactivity:

- Implementing a responsive design that adapts to different device screen sizes (desktop, tablet, mobile).
- Enabling interactive elements for users to toggle visualizations or adjust input parameters in real-time.

Utilizing Frontend Technologies:

- Leveraging HTML/CSS for layout and styling of the web application interface.
- Incorporating JavaScript and frontend frameworks/libraries (e.g., React.js, Vue.js) for dynamic UI components and interactivity.

5.3 Testing

Testing is a critical phase in software development, ensuring that each component and the system as a whole function correctly and meet specified requirements. Two key types of testing used in this project are Unit Testing and Integration Testing.

5.3.1 Unit Testing

Unit testing involves testing individual units or components of the software independently to verify their correctness. In the context of this project:

- **Feature Selection Unit Testing**: Each feature selection algorithm (e.g., Sequential Forward Floating Selection - SFFS) is tested to ensure it accurately identifies relevant features and improves model performance.
- **Machine Learning Model Unit Testing**: Each base classifier (e.g., logistic regression, decision tree) is tested with synthetic and real-world data to validate its prediction accuracy and behavior.

Unit testing is conducted using automated testing frameworks like pytest or unittest. Test cases are designed to cover different scenarios, including typical and edge cases, to ensure robustness and reliability of individual components.

5.3.2 Integration Testing

Integration testing focuses on testing interactions and interfaces between different modules or components to ensure they work together seamlessly. In the context of this project:

- **Data Preprocessing Integration Testing**: Integration tests verify the correct flow of data between preprocessing steps (e.g., cleaning, feature engineering) to ensure data integrity and consistency.
- **Model Pipeline Integration Testing**: Testing the end-to-end model pipeline involves validating interactions between feature selection, model training, and prediction stages.

Integration tests assess the system's behavior when multiple components interact, detecting issues like data format mismatches, interface errors, or performance bottlenecks. Tools like Jenkins or Travis CI can automate integration testing to ensure continuous integration and delivery (CI/CD) practices.

Overall, thorough unit testing and integration testing are essential to validate the functionality, reliability, and performance of the developed PD diagnostic system. They help identify and rectify issues early in the development lifecycle, ensuring a robust and effective solution for early Parkinson's disease detection.

Assertions and test reports provide feedback on the overall system integrity and readiness for deployment.

5. PROJECT DEMONSTRATION

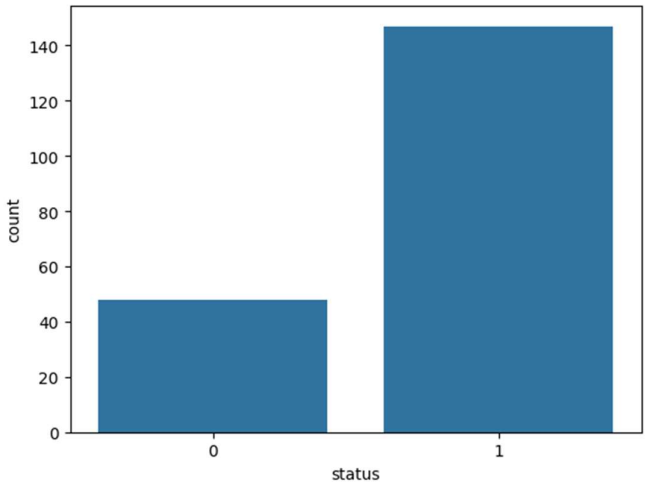


Figure 8: Visual representation showing the balance of data across different 'status' categories

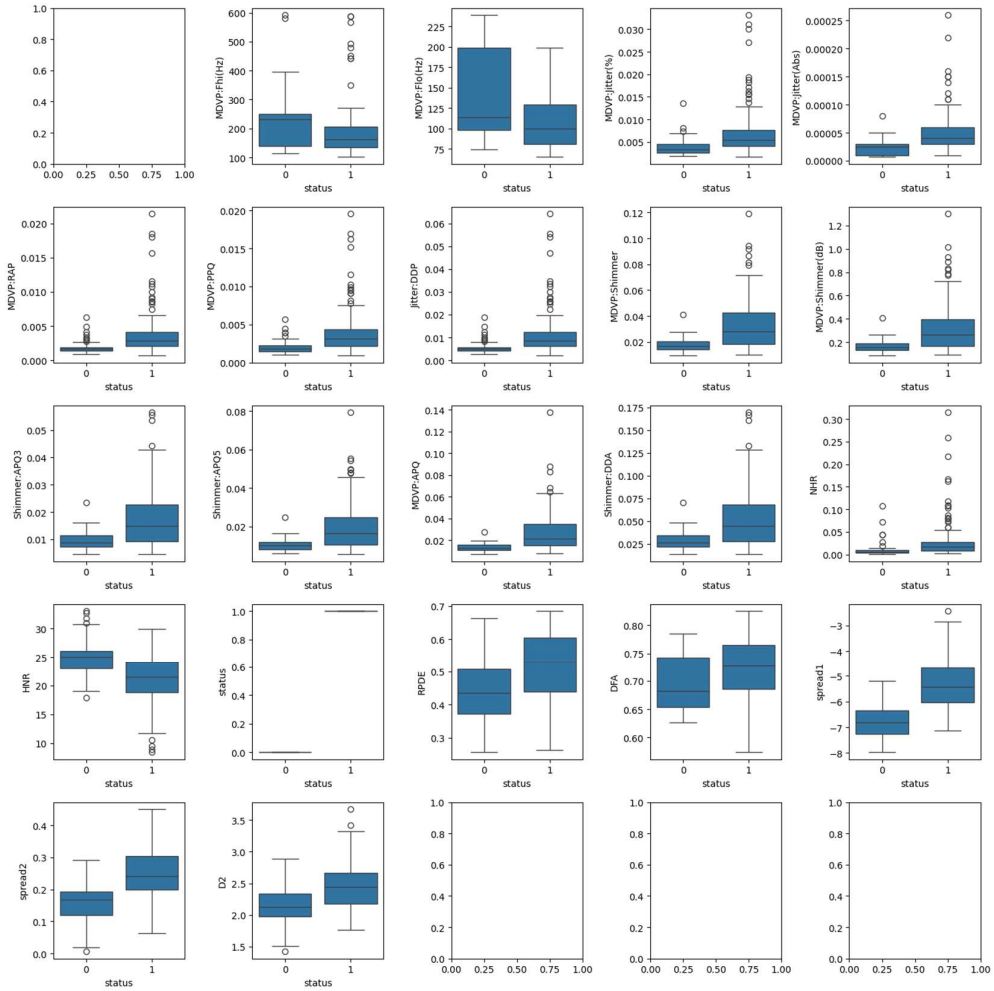


Figure 9: Box plots of various features by 'status'.

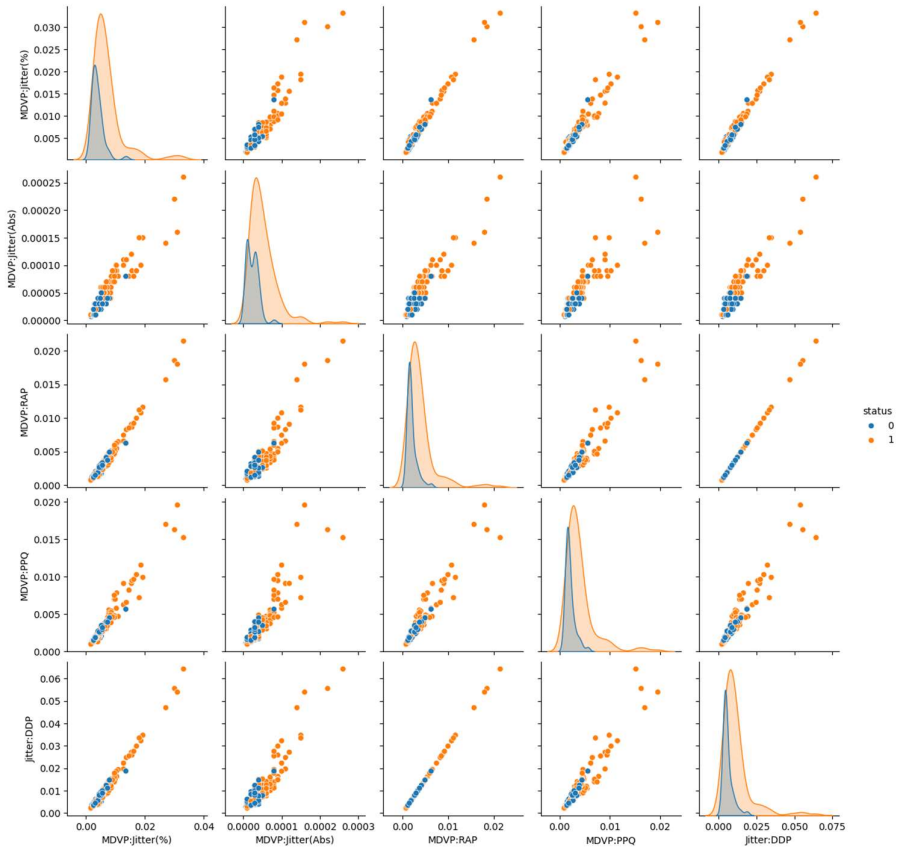


Figure 10: Visual representation of relationships between jitter-related acoustic measures, highlighting differences based on 'status'

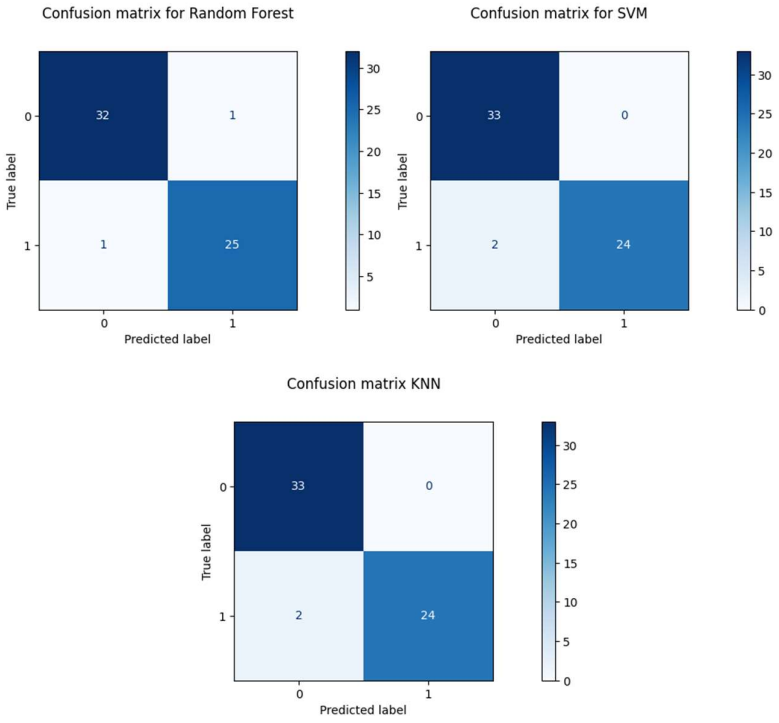


Figure 11: Confusion Matrices of some baseline ML models

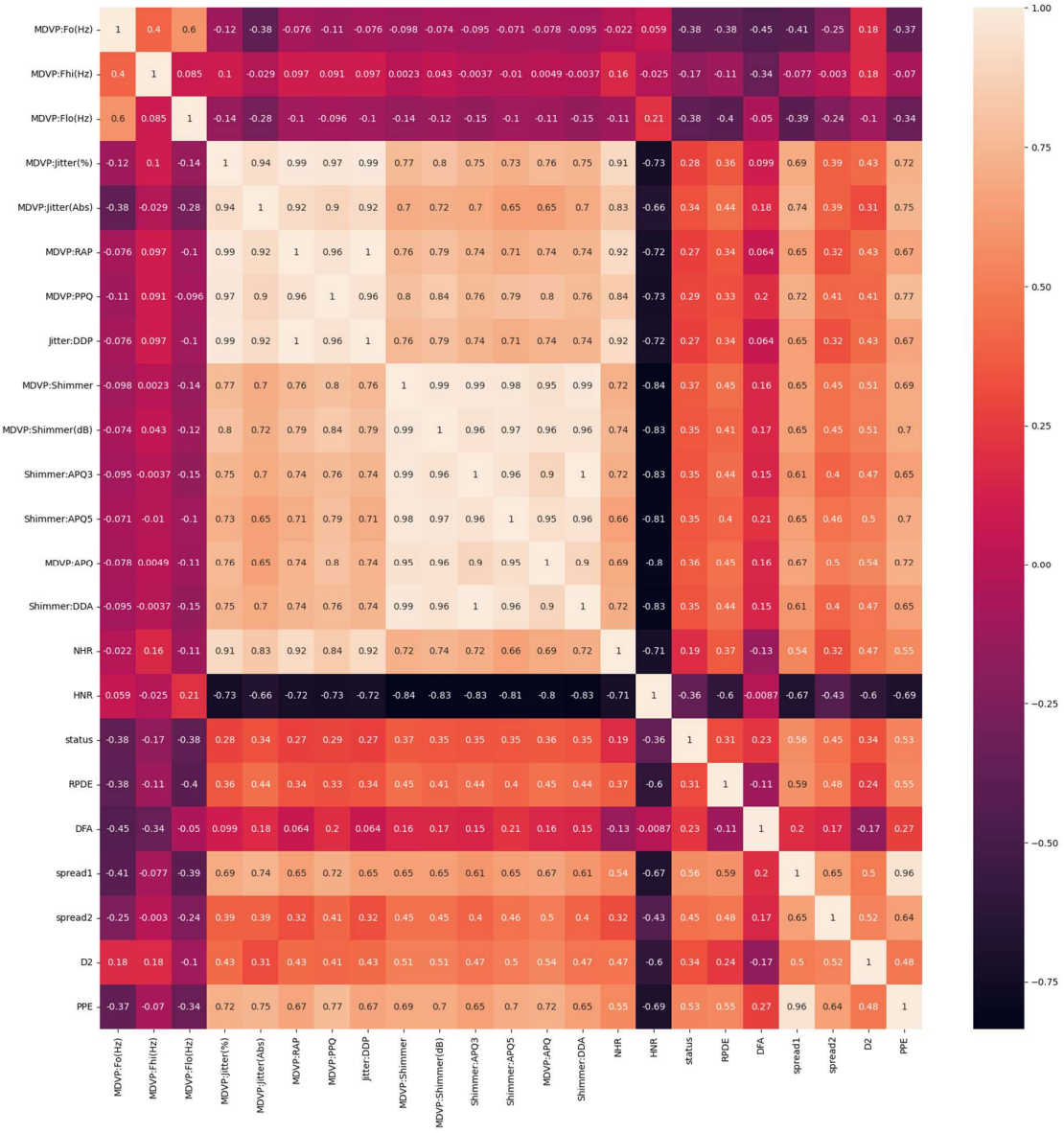


Figure 12: Heatmap Depicting correlation among various attributes of Dataset

Table 2: Machine Learning models with their evaluation parameters

	Metric	DT	RF	LR	SVM	NB	KNN	XGB
0	Accuracy	0.932203	0.966102	0.830508	0.966102	0.762712	0.966102	0.932203
1	F1-Score	0.920000	0.961538	0.782609	0.960000	0.650000	0.960000	0.925926
2	Recall	0.884615	0.961538	0.692308	0.923077	0.500000	0.923077	0.961538
3	Precision	0.958333	0.961538	0.900000	1.000000	0.928571	1.000000	0.892857
4	R2-Score	0.724942	0.862471	0.312354	0.862471	0.037296	0.862471	0.724942

Figure 13: Web View of Model Prediction Interface

6. RESULT & DISCUSSION ANALYSIS

Table 3: Evaluation results summarizing model performance using key metrics.

	Metric	DT	RF	LR	SVM	NB	KNN	XGB	ADA	ETC	GBC	E2	E3	E4	E1
0	Accuracy	0.932203	0.966102	0.830508	0.966102	0.762712	0.966102	0.932203	0.898305	0.949153	0.949153	0.915254	0.949153	0.966102	0.983051
1	F1-Score	0.920000	0.961538	0.782609	0.960000	0.650000	0.960000	0.925926	0.888889	0.941176	0.943396	0.897959	0.941176	0.961538	0.981132
2	Recall	0.884615	0.961538	0.692308	0.923077	0.500000	0.923077	0.961538	0.923077	0.923077	0.961538	0.846154	0.923077	0.961538	1.000000
3	Precision	0.958333	0.961538	0.900000	1.000000	0.928571	1.000000	0.892857	0.857143	0.960000	0.925926	0.956522	0.960000	0.961538	0.962963
4	R2-Score	0.724942	0.862471	0.312354	0.862471	0.037296	0.862471	0.724942	0.587413	0.793706	0.793706	0.656177	0.793706	0.862471	0.931235

The developed ensemble stacking model, which integrates Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) classifiers, exhibited exceptional accuracy, achieving approximately 98% accuracy in the early detection of Parkinson's disease. This impressive result underscores the robustness and effectiveness of the ensemble learning approach, which leverages the complementary strengths of diverse individual classifiers. By combining the predictive abilities of multiple models, the ensemble method outperforms standalone baseline classifiers like logistic regression or decision trees, offering superior accuracy and robustness in identifying PD cases.

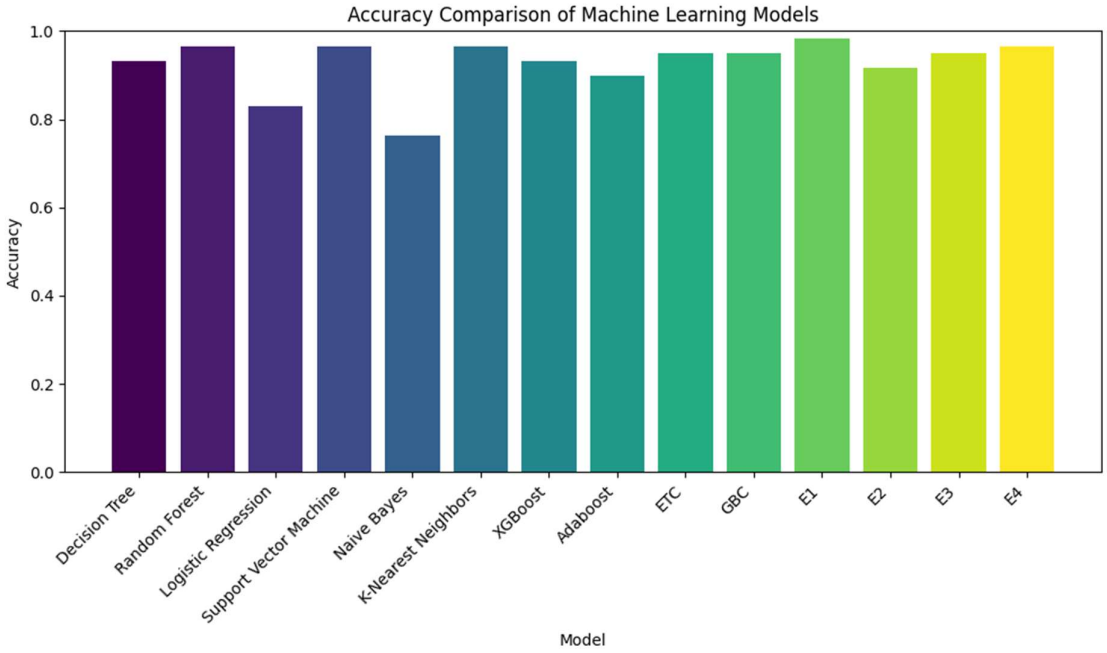


Figure 14: Visualization showing the accuracy of different ML models & their Ensembles.

SUMMARY

The project aims to develop an advanced machine learning-based system for the early detection of Parkinson's disease (PD) using a stacking ensemble technique, addressing gaps identified in existing PD diagnostic systems. By leveraging ensemble learning, the system integrates feature selection and multiple machine learning algorithms to enhance diagnostic accuracy and effectiveness.

Existing PD diagnostic systems often rely on individual classifiers, which may struggle with data heterogeneity and class imbalance. The proposed system addresses these limitations by employing a stacking ensemble approach, which combines predictions from diverse base models to improve overall performance.

The project's objectives include implementing feature selection techniques to identify discriminative PD-related features and developing a stacked ensemble model integrating various machine learning algorithms. Through rigorous evaluation and comparison with baseline classifiers, the effectiveness of the proposed ensemble approach is demonstrated.

Ensemble learning offers several advantages over traditional single-model approaches. By combining predictions from multiple base models (e.g., logistic regression, decision trees, support vector machines), ensemble methods can capture different aspects of the data and reduce overfitting. This diversity and aggregation of predictions lead to improved accuracy and robustness in PD detection compared to individual classifiers.

In real-world usage, the developed PD diagnostic system can provide clinicians with more reliable and interpretable diagnostic results, facilitating early intervention and personalized treatment plans. The system's effectiveness in handling data heterogeneity, class imbalance, and feature selection contributes to its practical utility and potential impact in clinical settings.

By rectifying gaps in existing PD diagnostic systems and demonstrating the superiority of ensemble learning over baseline models, this project advances the state-of-the-art in machine learning-based healthcare diagnostics. The proposed system holds promise for improving patient outcomes through early and accurate detection of Parkinson's disease.

7. REFERENCES

[1] P. R. Magesh, R. D. Myloth, and R. J. Tom, “An Explainable Machine Learning Model for Early Detection of Parkinson’s Disease using LIME on DaTSCAN Imagery,” Computers in Biology and Medicine, vol. 126, p. 104041, Nov. 2020, doi: https://doi.org/10.1016/j.compbiomed.2020.104041.

[2] M. Chen, Y. Hao, K. Hwang, L. Wang and L. Wang, "Disease Prediction by Machine Learning Over Big Data From Healthcare Communities," in IEEE Access, vol. 5, pp. 8869-8879, 2017, doi: 10.1109/ACCESS.2017.2694446.

[3] W. Wang, J. Lee, F. Harrou and Y. Sun, "Early Detection of Parkinson’s Disease Using Deep Learning and Machine Learning," in IEEE Access, vol. 8, pp. 147635-147646, 2020, doi: 10.1109/ACCESS.2020.3016062.

[4] R. Alshammri, G. Alharbi, E. Alharbi, and I. Almubark, “Machine learning approaches to identify Parkinson’s disease using voice signal features,” Frontiers in Artificial Intelligence, vol. 6, p. 1084001, Mar. 2023, doi: https://doi.org/10.3389/frai.2023.1084001.

[5] S. K. Biswas, A. Nath Boruah, R. Saha, R. S. Raj, M. Chakraborty, and M. Bordoloi, “Early detection of Parkinson disease using stacking ensemble method,” Computer Methods in Biomechanics and Biomedical Engineering, pp. 1–13, May 2022, doi: https://doi.org/10.1080/10255842.2022.2072683.

[6] A. Govindu and S. Palwe, “Early detection of Parkinson’s disease using machine learning,” Procedia Computer Science, vol. 218, pp. 249–261, 2023, doi: https://doi.org/10.1016/j.procs.2023.01.007.

[7] Karapinar Senturk, Z. Early Diagnosis of Parkinson’s Disease Using Machine Learning Algorithms. Medical Hypotheses 2020, 138, 109603. https://doi.org/10.1016/j.mehy.2020.109603.

[8] A. Govindu and S. Palwe, “Early detection of Parkinson’s disease using machine learning,” Procedia Computer Science, vol. 218, pp. 249–261, 2023, doi:

https://doi.org/10.1016/j.procs.2023.01.007.

[9] D. D. Joshi, H. H. Joshi, B. Y. Panchal, P. Goel and A. Ganatra, "A Parkinson Disease Classification Using Stacking Ensemble Machine Learning Methodology," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1335-1341, doi: https://ieeexplore.ieee.org/document/9823509

[10] T. Velmurugan and J. Dhinakaran, "A Novel Ensemble Stacking Learning Algorithm for Parkinson's Disease Prediction," Mathematical Problems in Engineering, vol. 2022, pp. 1–10, Jul. 2022, doi: https://doi.org/10.1155/2022/9209656.

[11] H. V. Nguyen and H. Byeon, "Prediction of Parkinson's Disease Depression Using LIME-Based Stacking Ensemble Model," Mathematics, vol. 11, no. 3, p. 708, Jan. 2023, doi: https://doi.org/10.3390/math11030708.

[12] K. M. Alalayah, Ebrahim Mohammed Senan, H. F. Atlam, Ibrahim Abdulrab Ahmed, and A. Shatnawi, "Automatic and Early Detection of Parkinson's Disease by Analyzing Acoustic Signals Using Classification Algorithms Based on Recursive Feature Elimination Method," vol. 13, no. 11, pp. 1924–1924, May 2023, doi: https://doi.org/10.3390/diagnostics13111924.

[13] S. A. Doumari, K. Berahmand, and M. J. Ebadi, "Early and High-Accuracy Diagnosis of Parkinson's Disease: Outcomes of a New Model," *Computational and Mathematical Methods in Medicine*, vol. 2023, p. 1493676, 2023, doi: https://doi.org/10.1155/2023/1493676.

[14] Shelke, Maya & Ranjan, Nihar & Mate, Gitanjali. (2022). Detection of Parkinson's Disease using Machine Learning Algorithms and Handwriting Analysis. Journal of Data Mining and Management. 8. 10.46610/JoDMM.2023.v08i01.004.

[15] Johann Faouzi, Olivier Colliot, and Jean-Christophe Corvol, "Machine Learning for Parkinson's Disease and Related Disorders," Neuromethods, pp. 847–877, Jan. 2023, doi: https://doi.org/10.1007/978-1-0716-3195-9_26.

APPENDIX A – SAMPLE CODE

```
# Importing Libraries
import requests
import pandas as pd
from imblearn.over_sampling import SMOTE
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB as Naive_Bayes
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.datasets import make_classification
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from xgboost import XGBClassifier
import joblib

from IPython.display import display

# Reading Data Into Pandas Dataframe
df = pd.read_csv('/content/parkinsons.data')

# Exploring Dataset Content
```

```
df.head()

df.tail()

print('Number of Features In Dataset :', df.shape[1])
print('Number of Instances In Dataset : ', df.shape[0])

# Dropping The Name Column
df.drop(['name'], axis=1, inplace=True)

print('Number of Features In Dataset :', df.shape[1])
print('Number of Instances In Dataset : ', df.shape[0])

# Exploring Information About Dataframe
df.info()

df.describe()
"""## KNN Classifier
"""
import numpy as np

Ks = 10
mean_acc = []
ConfustionMx = [];
for n in range(2,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc.append(metrics.accuracy_score(y_test, yhat))
print('Neighbor Accuracy List')
print(mean_acc)

plt.plot(range(2,Ks),mean_acc,'g')
```

```
plt.ylabel('Accuracy ')
plt.xlabel('Number of Neighbours (K)')
plt.tight_layout()
plt.show()

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
predKNN = knn.predict(X_test)

y_pred = knn.predict(X_test)
cm = confusion_matrix(y_test, y_pred)

# Plot confusion matrix using ConfusionMatrixDisplay
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=knn.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title('Confusion matrix KNN', y=1.1)
plt.show()

y_pred_proba = knn.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()

# Dumping KNN Classifier
joblib.dump(knn, 'knn_clf.pkl')

# @title Extra Trees Classifier
from sklearn.ensemble import ExtraTreesClassifier
extra_trees = ExtraTreesClassifier(n_estimators=100, random_state=42)
extra_trees.fit(X_train, y_train)

# Make predictions
```

```
predetc = extra_trees.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, predetc)
print("Accuracy:", accuracy)

#

# Define base classifiers
base_classifiers = [
   ('random_forest', RandomForestClassifier(n_estimators=100,
                                            criterion='gini',
                                            max_depth=None,
                                            min_samples_split=2,
                                            min_samples_leaf=1,
                                            max_features='auto',
                                            random_state=22)),

   ('svm', SVC(kernel='rbf',
               C=1.0,
               gamma='scale',
               random_state=22))
]

# Define the meta-learner
meta_learner = DecisionTreeClassifier(random_state=20)

# Create the stacking classifier
clf2 = StackingClassifier(estimators=base_classifiers, final_estimator=meta_learner)

# Fit the model
clf2.fit(X_train, y_train)

# Predict on the test set
```

```
y_pred = clf2.predict(X_test)

# Calculate the accuracy score
E2 = accuracy_score(y_true=y_test, y_pred=y_pred)
print("Stacking Classifier Accuracy:", E2)

# @title Ensemble Model
# Define base classifiers
base_classifiers = [
   ('random_forest', RandomForestClassifier(n_estimators=100,
                                            criterion='gini',
                                            max_depth=None,
                                            min_samples_split=2,
                                            min_samples_leaf=1,
                                            max_features='auto',
                                            random_state=22)),

   ('knn', KNeighborsClassifier(n_neighbors=5,
                                weights='uniform',
                                algorithm='auto',
                                leaf_size=30,
                                p=2,
                                metric='minkowski'))
]

# Define the meta-learner
meta_learner = DecisionTreeClassifier(random_state=20)

# Create the stacking classifier
clf4 = StackingClassifier(estimators=base_classifiers, final_estimator=meta_learner)

# Fit the model
clf4.fit(X_train, y_train)
```

```
# Predict on the test set
y_pred = clf4.predict(X_test)

# Calculate the accuracy score
E4 = accuracy_score(y_true=y_test, y_pred=y_pred)
print("Stacking Classifier Accuracy:", E4)

# @title Ensemble Model
# Define base classifiers
base_classifiers = [
    ('random_forest', RandomForestClassifier(n_estimators=100,
                                             criterion='gini',
                                             max_depth=None,
                                             min_samples_split=2,
                                             min_samples_leaf=1,
                                             max_features='auto',
                                             random_state=22)),

    ('svm', SVC(kernel='rbf',
                C=1.0,
                gamma='scale',
                random_state=22)),

    ('knn', KNeighborsClassifier(n_neighbors=5,
                                 weights='uniform',
                                 algorithm='auto',
                                 leaf_size=30,
                                 p=2,
                                 metric='minkowski'))
]

# Define the meta-learner
meta_learner = DecisionTreeClassifier(random_state=20)
```

```
# Create the stacking classifier
clf1 = StackingClassifier(estimators=base_classifiers, final_estimator=meta_learner)

# Fit the model
clf1.fit(X_train, y_train)

# Predict on the test set
y_pred = clf1.predict(X_test)

# Calculate the accuracy score
E1 = accuracy_score(y_true=y_test, y_pred=y_pred)
print("Stacking Classifier Accuracy:", E1)

# Pickling the Model
import pickle
pickle.dump(clf1, open('ensemble.pkl','wb'))
model_clf1 = pickle.load(open('ensemble.pkl','rb'))
```

Full version:

Github : https://github.com/Mridul28/Capstone

Early Detection Of Parkinson's Disease Using Machine Learning

Mridul Madnani |Devansh Bajpai| Srijan Singh Somvanshi| Sayan Sikder| SCOPE

Introduction

Research aims to apply ensemble learning for Parkinson's disease diagnosis, addressing the gap of integrating feature selection, ensemble learning, and diverse ML algorithms. The literature gap lies in underutilization of ensemble learning and lack of comprehensive systems for PD diagnosis.

Motivation

The motivation for this project stems from the potential to improve patients' quality of life through early intervention and treatment of Parkinson's disease. Additionally, it offers valuable learning experience in the field of ML.

SCOPE of the Project

The project scope involves developing a system for early Parkinson's disease detection using ensemble stacking techniques. It includes data collection, preprocessing, and feature selection tailored to PD datasets. Ensemble learning algorithms will be employed to combine diverse base classifiers for improved accuracy. Model performance will be evaluated rigorously through cross-validation and testing on independent dataset. Collaboration with domain experts will ensure clinical relevance and validation of the system. The project will explore the potential of biomarker identification for enhanced diagnostic accuracy.

Methodology

The methodology adapted for this project is as follows:

Data Preprocessing:

- Importing necessary libraries and dataset.
- Exploring dataset content, dropping irrelevant columns.
- Scaling features and splitting data into training and testing sets.

Model Development and Training:

- Selection of machine learning algorithms like logistic regression, SVM, XGBoost.
- Fine-tuning hyperparameters through grid search or random search.
- Training models on preprocessed data, using cross-validation techniques.
- Iteratively optimizing model architectures and hyperparameters based on validation results.

Comparison and Stacking Ensemble Techniques:

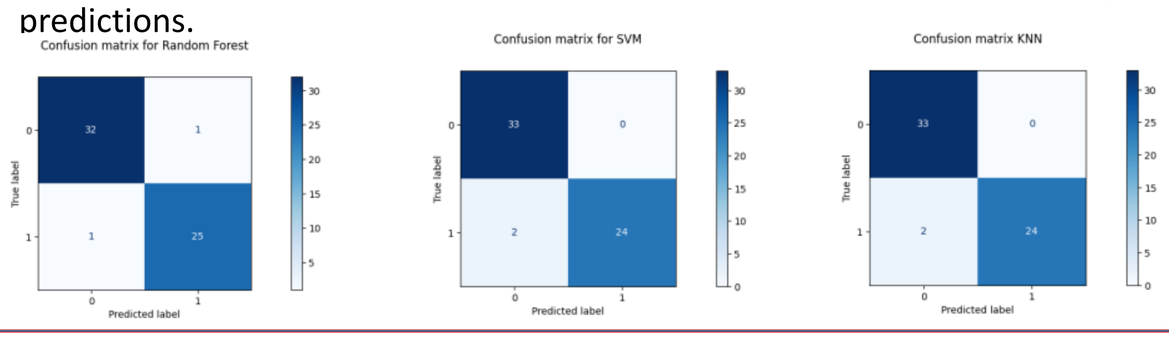
- Evaluating baseline model performance using cross-validation.
- Implementing stacking ensemble methods with diverse base models.
- Assessing ensemble performance and comparing it with individual models.
- Interpreting ensemble model decisions and visualizing decision-making processes.

Validation and Evaluation:

- Comparing developed models with baseline algorithms.
- Visualizing model performance using ROC curves, confusion matrices.
- Assessing interpretability and explainability of models.
- Selecting promising models for further refinement based on evaluation results.

Building a User Interface:

- Developed a user-friendly website for Parkinson's disease detection.
- Implemented a feature for users to input voice data.
- Utilized machine learning models to process input data and provide diagnosis predictions.

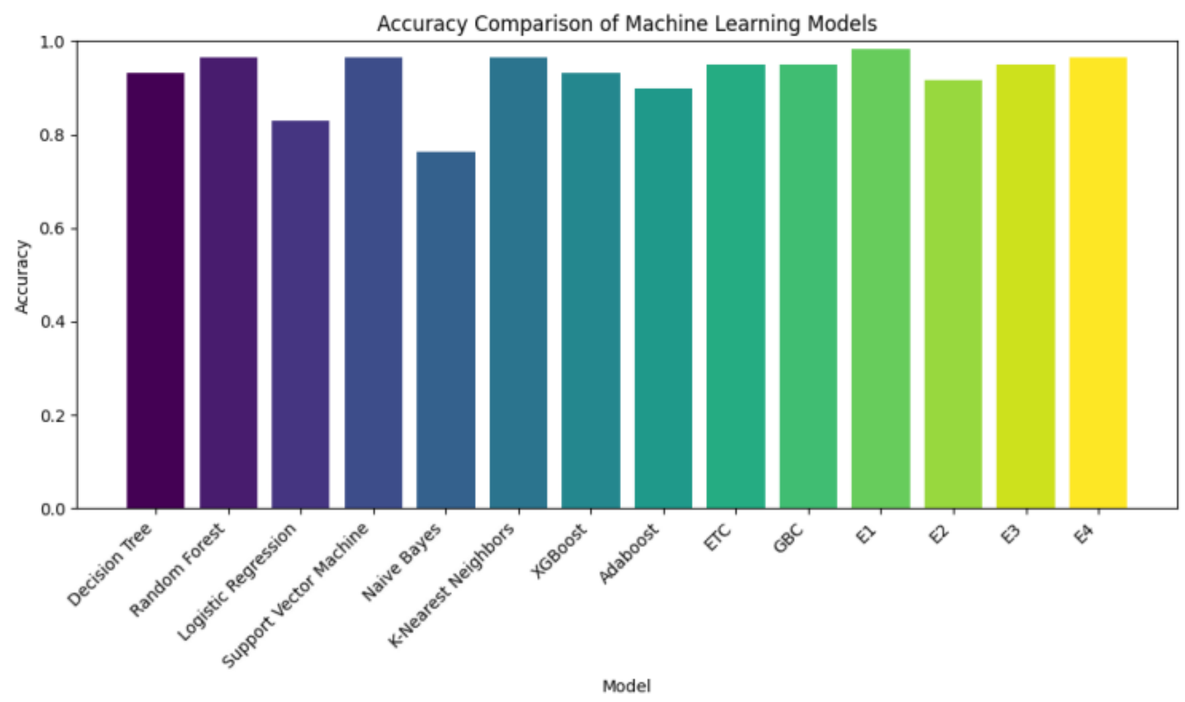


Results

	Metric	DT	RF	LR	SVM	NB	KNN	XGB	ADA	ETC	GBC	E2	E3	E4	E1
0	Accuracy	0.932203	0.966102	0.830508	0.966102	0.762712	0.966102	0.932203	0.898305	0.949153	0.949153	0.915254	0.949153	0.966102	0.983051
1	F1-Score	0.920000	0.961538	0.782609	0.960000	0.650000	0.960000	0.925926	0.888889	0.941176	0.943396	0.897959	0.941176	0.961538	0.981132
2	Recall	0.884615	0.961538	0.692308	0.923077	0.500000	0.923077	0.961538	0.923077	0.923077	0.961538	0.846154	0.923077	0.961538	1.000000
3	Precision	0.958333	0.961538	0.900000	1.000000	0.928571	1.000000	0.892857	0.857143	0.960000	0.925926	0.956522	0.960000	0.961538	0.962963
4	R2-Score	0.724942	0.862471	0.312354	0.862471	0.037296	0.862471	0.724942	0.587413	0.793706	0.793706	0.656177	0.793706	0.862471	0.931235

Evaluation results summarizing model performance using key metrics.

We've developed a Parkinson's disease detection model using machine learning techniques. Through meticulous comparison, we identified Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) as the best-performing models, each boasting accuracies around 96%. Leveraging the ensemble stacking method, we combined these top-performing models into a unified ensemble model, which remarkably achieved an accuracy of approximately 98% in early Parkinson's disease detection. This outcome highlights the exceptional effectiveness and robustness of the ensemble learning approach. By integrating Random Forest, SVM, and KNN classifiers, our ensemble model capitalizes on the diverse strengths of individual models, surpassing the performance of standalone baseline classifiers like logistic regression or decision trees. This unified approach offers superior accuracy and robustness, significantly enhancing the identification of Parkinson's disease cases.



Visualization showing the accuracy of different ML models & their Ensembles.

Conclusion

In conclusion, our ensemble stacking model, combining Random Forest, SVM, and KNN classifiers, achieved an impressive early detection accuracy of around 98% for Parkinson's disease. This underscores the robustness of ensemble methods in leveraging diverse classifier strengths. Future endeavors may center on validating the model with larger datasets, analyzing feature importance for enhanced interpretability, and exploring alternative ensemble techniques or domain-specific knowledge incorporation. Extending applicability to clinical settings and mitigating potential biases would further bolster its practical utility in healthcare decision-making.

References

T. Velmurugan and J. Dhinakaran, "A Novel Ensemble Stacking Learning Algorithm for Parkinson's Disease Prediction," Mathematical Problems in Engineering, vol. 2022, pp. 1–10, Jul. 2022, doi: https://doi.org/10.1155/2022/9209656.

H. V. Nguyen and H. Byeon, "Prediction of Parkinson's Disease Depression Using LIME-Based Stacking Ensemble Model," Mathematics, vol. 11, no. 3, p. 708, Jan. 2023, doi: https://doi.org/10.3390/math11030708

S. A. Doumari, K. Berahmand, and M. J. Ebadi, "Early and High-Accuracy Diagnosis of Parkinson's Disease: Outcomes of a New Model," Computational and Mathematical Methods in Medicine, vol. 2023, p. 1493676, 2023, doi: https://doi.org/10.1155/2023/1493676.

Johann Faouzi, Olivier Colliot, and Jean-Christophe Corvol, "Machine Learning for Parkinson's Disease and Related Disorders," Neuromethods, pp. 847–877, Jan. 2023, doi: https://doi.org/10.1007/978-1-0716-3195-9_26.