Leveraging Machine Learning and Ensemble Methods for Accurate Parkinson's Disease Diagnosis: A Study on SMOTE-TomekLinks and SHAP Interpretability

Mohsin Amin

Computer science, ICER VIT Bangalore, WhiteField, 560066, Karnataka, India.

Abstract

Purpose: Parkinson's disease (PD) is a progressive neurodegenerative disorder that affects millions globally, leading to significant impairments in both motor and non-motor functions. The early and accurate diagnosis of PD remains a critical challenge, as existing diagnostic methods often depend on the manifestation of advanced-stage symptoms. This study conducts a comprehensive comparative analysis of machine learning base models, evaluated both with and without the application of SMOTE-TomekLinks to address class imbalance. Additionally, the research integrates SHAP (SHapley Additive Explanations) analysis to ensure model interpretability and employs ensemble stacking techniques that combine the outputs of base models with two meta-models, XGBoost and AdaBoost, to enhance predictive accuracy and reliability.

Methods: A dataset was collected from the UCI repository and preprocessed for normalization and feature selection. Six machine learning models, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors, and Naive Bayes, were trained and evaluated with and without SMOTE-TomekLinks. Ensemble techniques using XGBoost and AdaBoost were employed to enhance predictive accuracy. Model performance was assessed using metrics such as accuracy, F1-score, confusion matrices, and ROC-AUC. SHAP (SHapley Additive exPlanations) analysis was used to interpret feature importance.

Results: SMOTE-TomekLinks significantly improved the performance of all models, with Random Forest achieving the highest accuracy (96.61%) among the base models. Ensemble techniques further enhanced performance, with XGBoost achieving the best results, including an accuracy of 98.30%, an F1-score of 0.98 for both classes, and an ROC-AUC of 0.98. SHAP analysis identified key features such as spread1, spread2, PPE, and MDVP:Fo(Hz) as critical for classification.

Conclusion: The study demonstrates the transformative potential of combining advanced preprocessing, class-balancing techniques, and ensemble methods in diagnosing Parkinson's disease. The findings emphasize the importance of addressing class imbalance to achieve reliable and interpretable diagnostic tools, bridging the gap between computational approaches and clinical applications hence, improving patient outcomes.

Keywords: Parkinson's Disease (PD), Machine Learning, Synthetic Minority Oversampling Technique (SMOTE)-TomekLinks, SHAP (SHapley Additive exPlanations), XGBoost(Extreme Gradient Boosting),

AdaBoost(Adaptive Boosting)

1 Introduction

Parkinson's disease (PD) is a complex neurodegenerative disorder primarily affecting middle-aged and older adults, characterized by both motor and non-motor symptoms. The motor symptoms include bradykinesia, tremor, rigidity, and postural instability, which are primarily due to the degradation and death of dopaminergic neurons in the substantia nigra-striatum [1]. Non-motor symptoms encompass cognitive impairment, depression, anxiety, autonomic dysfunction, and sensory disturbances, which significantly impact the quality of life [2]. PD affects over 6 million people globally, with nearly 1 million cases in the United States alone[3]. In Europe, the prevalence and incidence rates are approximately 108–257 per 100,000 and 11–19 per 100,000 per year, respectively [4]. PD is more common in males, with a male-to-female ratio of 1.5:1.0, and its prevalence increases with age, affecting 1%-2% of those over 60 and up to 3.5% of individuals aged 85-89 [5]. The early diagnosis of Parkinson's disease (PD) is critical for effective intervention and management, and recent research highlights the potential of biomedical voice measurements as a non-invasive diagnostic tool. Traditional methods of diagnosing PD rely heavily on clinical evaluations, which can be subjective and time-consuming [6][7] .Machine learning and deep learning approaches have emerged as promising alternatives, leveraging voice data to detect early signs of PD. Studies have utilized various machine learning models, such as Support Vector Machines (SVM), Random Forest, and Convolutional Neural Networks (CNNs), to analyze voice parameters like pitch, jitter, shimmer, and noise-to-harmonics ratio, which are indicative of vocal impairments associated with PD [6] [8] [9]. These advancements in voice analysis and machine learning not only enhance the precision of PD diagnosis but also pave the way for similar applications in other neurodegenerative disorders, offering a transformative approach to healthcare diagnostics [7].

2 Related Work

Recent advancements in the application of machine learning (ML) to Parkinson's disease (PD) have significantly focused on utilizing biomedical voice measurements and other motor symptoms for early detection and severity assessment. Voice analysis has emerged as a promising non-invasive diagnostic tool, with studies exploring features such as jitter, shimmer, and fundamental frequency. For instance, the integration of advanced vocal features like Vocal Tract Length Normalization and Empirical Mode Decomposition with ML algorithms such as Explainable Boosting Machine has achieved detection accuracies of up to 86.67% [10]. Deep learning models, including Convolutional Neural Networks and Long Short-Term Memory networks, have also been employed to predict PD severity, achieving high accuracy rates of up to 98% [11]. Beyond voice analysis, the integration of handwriting analysis with voice data has shown potential in enhancing diagnostic accuracy, suggesting that a hybrid approach could yield superior results compared to single-modality methods [12]. The application of smartphone-based applications for real-world symptom assessment has been proposed, offering a more objective and patient-centric approach to monitoring PD symptoms[13]. These studies collectively highlight the transformative potential of ML in PD diagnosis and management, emphasizing the importance of early detection through innovative, non-invasive methods that could significantly improve patient outcomes and accessibility to care [14][15]. The integration of multiple features and advanced algorithms, including ensemble learning and deep learning techniques, continues to enhance the precision and efficacy of PD diagnosis, paving the way for more personalized and effective treatment strategies [16].

3 Materials and Methods

This section describes the data preprocessing, training and evaluation of six base models for Parkinson's disease classification, addressing class imbalance using SMOTE-TomekLinks and SHAP analysis with varying importance across models.

3.1 Model Architecture

The primary objective of this study is to develop a predictive model for Parkinson's disease using biomedical voice measurements and a range of machine learning algorithms, including Logistic Regression (LR), Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Naive Bayes. The secondary objective is to conduct a comparative analysis of these base models, both with and without the application of SMOTE-TomekLinks, a technique used to address class imbalance issues present in dataset. Furthermore, ensemble stacking techniques will be employed to improve prediction accuracy by combining the outputs of base models (utilizing SMOTE-TomekLinks) with two meta-models, XGBoost and AdaBoost. To ensure interpretability, SHAP (Shapley Additive Explanations) analysis will be utilized, enabling transparent and comprehensible predictions for clinical decision-making. The overarching goal is to bridge the gap between traditional clinical diagnostic practices and modern computational approaches, ensuring the development of a robust and practical model suitable for real-world application.

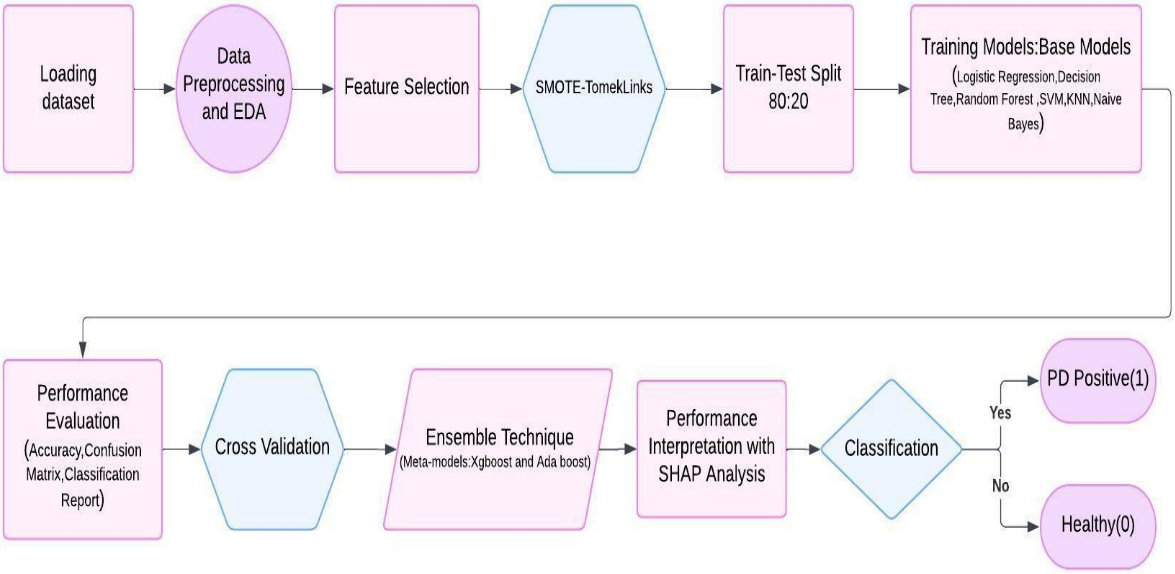
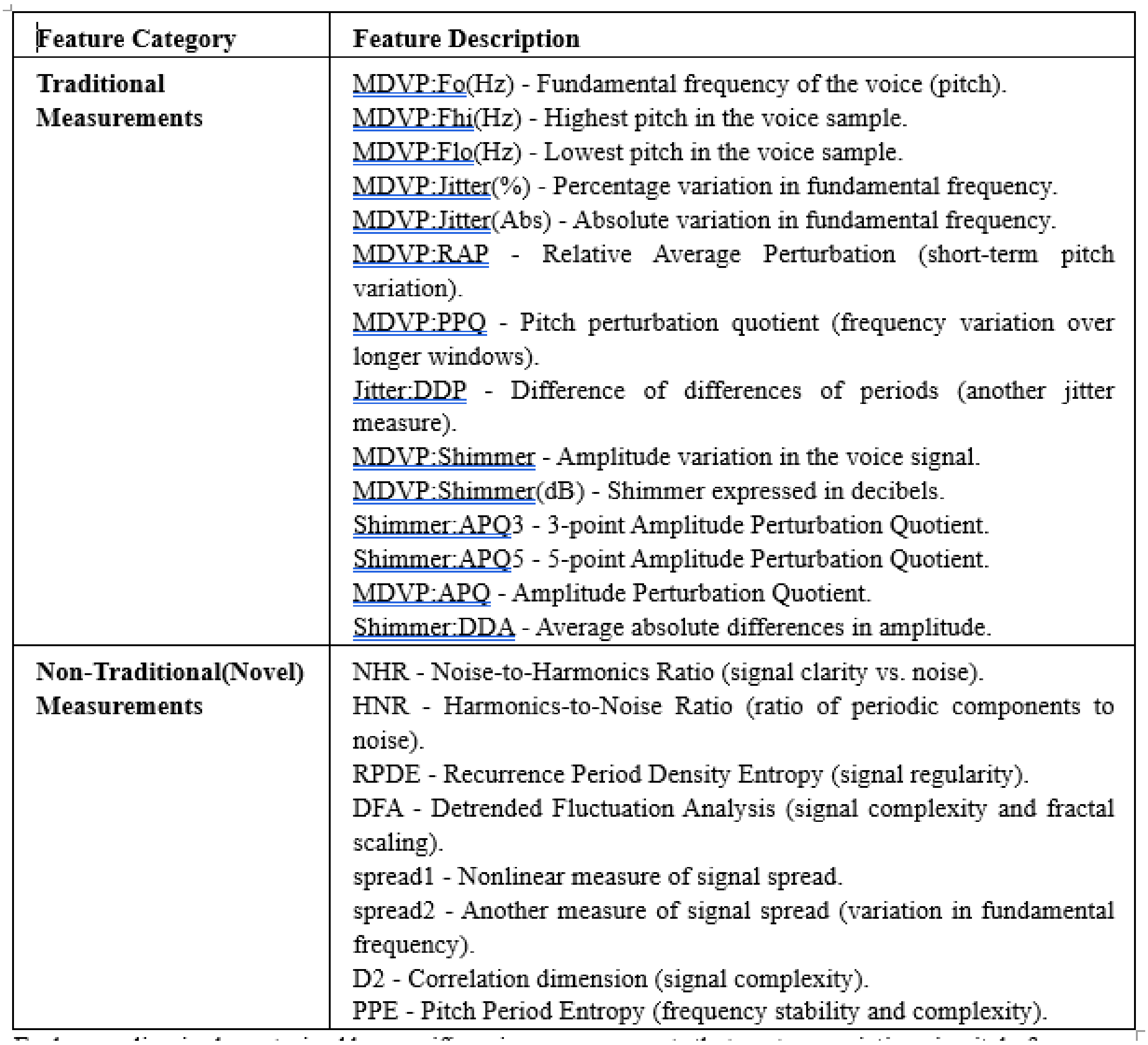

Loading dataset
Data Preprocessing and EDA
Feature Selection
SMOTE-TomekLinks
Train-Test Split 80:20
Training Models:Base Models (Logistic Regression,Decision Tree,Random Forest ,SVM,KNN,Naive Bayes)
Performance Evaluation (Accuracy,Confusion Matrix,Classification Report)
Cross Validation
Ensemble Technique (Meta-models:Xgboost and Ada boost)
Performance Interpretation with SHAP Analysis
Classification
Yes
No
PD Positive(1)
Healthy(0)

Fig. 1 Proposed Methodology Architecture

The dataset utilized in this study is sourced from the UCI Machine Learning Repository, comprising biomedical voice measurements, including features like jitter, shimmer, spread, and PPE (Pitch Period Entropy) etc. The data undergoes preprocessing and Exploratory Data Analysis (EDA) to identify and rectify inconsistencies, detect outliers, and extract meaningful insights. Relevant features are selected based on domain knowledge and statistical techniques to enhance model performance and reduce dimensionality. SMOTE-TomekLinks is applied to the dataset to address class imbalance, ensuring improved model training on underrepresented classes. The dataset is divided into training and testing subsets in an 80:20 ratio to evaluate model performance effectively. The selected machine learning algorithms (LR, DT, RF, SVM, KNN, Naive Bayes) are trained on the preprocessed data to serve as the base models. The base models' performance is evaluated using metrics such as accuracy, confusion matrix, and classification reports. Cross-validation is employed to ensure robust evaluation. To enhance predictive accuracy, ensemble stacking techniques are implemented using XGBoost and AdaBoost as meta-models, leveraging the outputs of the base models trained with SMOTE-TomekLinks. SHAP analysis is employed to interpret the predictions of the developed models, ensuring transparency and alignment with clinical decision-making needs. The final classification identifies whether a patient is Parkinson's disease positive (PD Positive) or healthy, based on the optimized model.

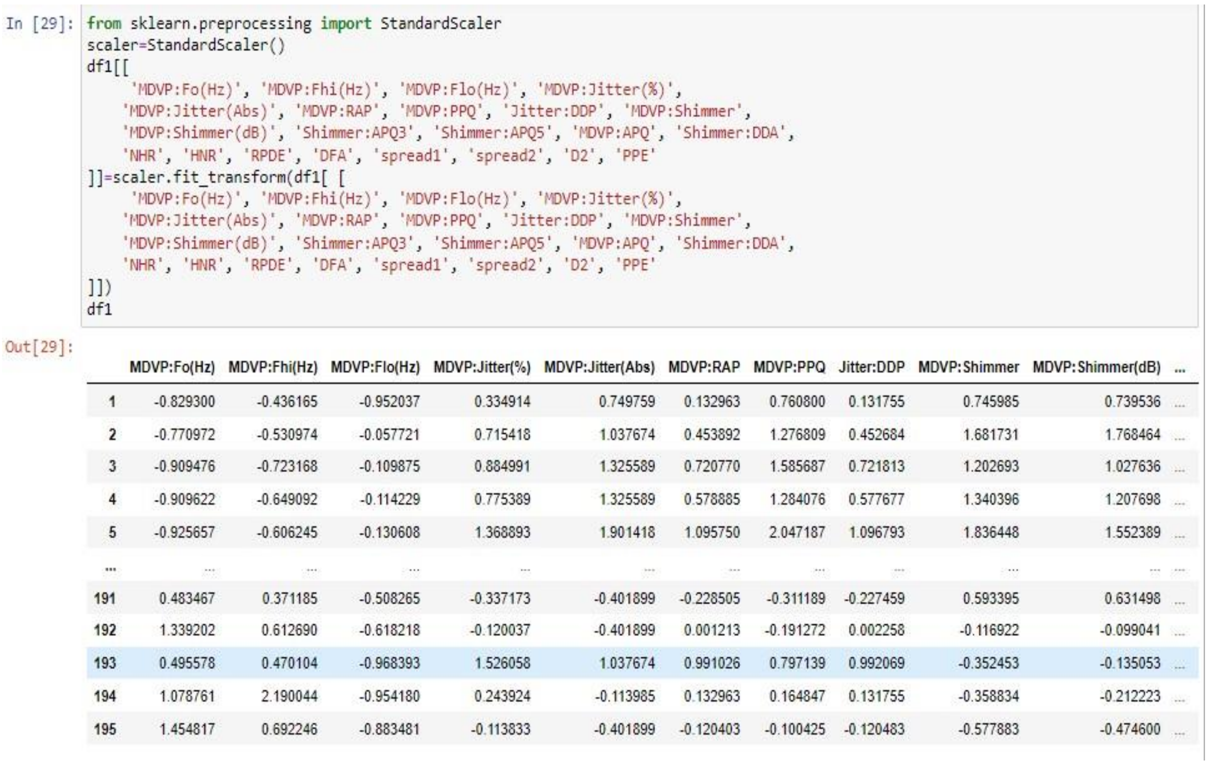
3.2 Data Preprocessing and Exploratory Data Analysis

The dataset used for this research is sourced from UCI Machine learning repository comprises biomedical voice measurements from 31 individuals, of whom 23 are diagnosed with Parkinson's Disease (PD) [17]. Each recording is characterized by specific voice measurements that capture variations in pitch, frequency, and amplitude, among others. With a total of 195 voice recordings (approximately six per patient), here patient identifiers are omitted during preprocessing to focus solely on the features relevant to PD diagnosis. Each row of features is associated with target column(status) which consists of 0 and 1's , where 0 represents healthy individual and 1 PD positve. Features are classified into two categories along with their descriptions and are shown in Table 1.



Feature Category	Feature Description
Traditional Measurements	MDVP:Fo(Hz) - Fundamental frequency of the voice (pitch). MDVP:Fhi(Hz) - Highest pitch in the voice sample. MDVP:Flo(Hz) - Lowest pitch in the voice sample. MDVP:Jitter(%) - Percentage variation in fundamental frequency. MDVP:Jitter(Abs) - Absolute variation in fundamental frequency. MDVP:RAP - Relative Average Perturbation (short-term pitch variation). MDVP:PPQ - Pitch perturbation quotient (frequency variation over longer windows). Jitter:DDP - Difference of differences of periods (another jitter measure). MDVP:Shimmer - Amplitude variation in the voice signal. MDVP:Shimmer(dB) - Shimmer expressed in decibels. Shimmer:APQ3 - 3-point Amplitude Perturbation Quotient. Shimmer:APQ5 - 5-point Amplitude Perturbation Quotient. MDVP:APQ - Amplitude Perturbation Quotient. Shimmer:DDA - Average absolute differences in amplitude.
Non-Traditional(Novel) Measurements	NHR - Noise-to-Harmonics Ratio (signal clarity vs. noise). HNR - Harmonics-to-Noise Ratio (ratio of periodic components to noise). RPDE - Recurrence Period Density Entropy (signal regularity). DFA - Detrended Fluctuation Analysis (signal complexity and fractal scaling). spread1 - Nonlinear measure of signal spread. spread2 - Another measure of signal spread (variation in fundamental frequency). D2 - Correlation dimension (signal complexity). PPE - Pitch Period Entropy (frequency stability and complexity).

Table 1 Classification of Features for Parkinson's Disease Dataset



```
In [29]: from sklearn.preprocessing import StandardScaler
         scaler=StandardScaler()
         df1[[
             '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', 'RPDE', 'DFA', 'spread1', 'spread2', 'D2', 'PPE'
         ]]=scaler.fit_transform(df1[ [
             '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', 'RPDE', 'DFA', 'spread1', 'spread2', 'D2', 'PPE'
         ]])
         df1
```

Out[29]:

	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)	...
1	-0.829300	-0.436165	-0.952037	0.334914	0.749759	0.132963	0.760800	0.131755	0.745985	0.739536	...
2	-0.770972	-0.530974	-0.057721	0.715418	1.037674	0.453892	1.276809	0.452684	1.681731	1.768464	...
3	-0.909476	-0.723168	-0.109875	0.884991	1.325589	0.720770	1.585687	0.721813	1.202693	1.027636	...
4	-0.909622	-0.649092	-0.114229	0.775389	1.325589	0.578885	1.284076	0.577677	1.340396	1.207698	...
5	-0.925657	-0.606245	-0.130608	1.368893	1.901418	1.095750	2.047187	1.096793	1.836448	1.552389	...
...	...	...	...	...	...	...	...	...	...	...	...
191	0.483467	0.371185	-0.508265	-0.337173	-0.401899	-0.228505	-0.311189	-0.227459	0.593395	0.631498	...
192	1.339202	0.612690	-0.618218	-0.120037	-0.401899	0.001213	-0.191272	0.002258	-0.116922	-0.099041	...
193	0.495578	0.470104	-0.968393	1.526058	1.037674	0.991026	0.797139	0.992069	-0.352453	-0.135053	...
194	1.078761	2.190044	-0.954180	0.243924	-0.113985	0.132963	0.164847	0.131755	-0.358834	-0.212223	...
195	1.454817	0.692246	-0.883481	-0.113833	-0.401899	-0.120403	-0.100425	-0.120483	-0.577883	-0.474600	...

Fig. 2 Normalized dataset

Data preprocessing involved several crucial steps to prepare the dataset for predictive modeling. Missing values were handled using imputation techniques such as mean, median, or k-Nearest Neighbors (KNN) imputation to ensure data completeness. Visualizations, including histograms, scatter plots, and correlation heatmaps, were employed to analyze feature distributions and relationships, providing insights into the data's structure. The dataset was normalized (Fig. 2) using the StandardScaler function to ensure that all features have equal influence on the target variable (status). Normalization standardizes the features by removing the mean and scaling to unit variance, thereby improving the comparability of variables with different scales. The boxplot (Fig. 3) shows most features have tightly distributed values with minimal variability, such as Jitter, Shimmer, NHR, and HNR. However, MDVP:Fhi(HZ) has significant outliers, indicating variability in maximum fundamental frequency. MDVP:Flo(HZ) and Spread1 ,Spread2 show broader distributions, suggesting potential distinguishing patterns. The compact distributions in RPDE, DFA, and PPE reflect consistency in these features. Overall, features like MDVP:Fhi(HZ) and Spread1, Spread2, PPE may be key for further analysis. The distribution graph (Fig. 4) shows that ”PD positive” individuals are concentrated in the 100-175 Hz range, while healthy individuals have a broader spread with higher frequencies above 200 Hz. There is significant overlap in the 125-200 Hz range, but lower MDVP: Fo (HZ) values are more common in the “PD positive” group. Healthy individuals show a higher frequency of elevated MDVP:Fo(HZ) values. This suggests MDVP:Fo(HZ) could help differentiate between the two groups. The boxplot of Fundamental Frequency (Fig. 5) shows that individuals with status 1 (PD positive) have lower and less variable MDVP:Fo(HZ) values compared to status 0 (healthy), which has higher and more dispersed values. The median for status 0 is notably higher, reflecting a tendency for healthier individuals to have greater fundamental frequency. There are no outliers in either group, as all values lie within the whiskers' range. This also suggests MDVP:Fo(HZ) could be a key feature for distinguishing between the two groups. Correlation analysis (Fig. 6) was utilized to identify and remove features with high multicollinearity, reducing redundancy and ensuring the model's interpretability and efficiency. This comprehensive preprocessing ensured the dataset's readiness for robust and reliable machine learning applications.

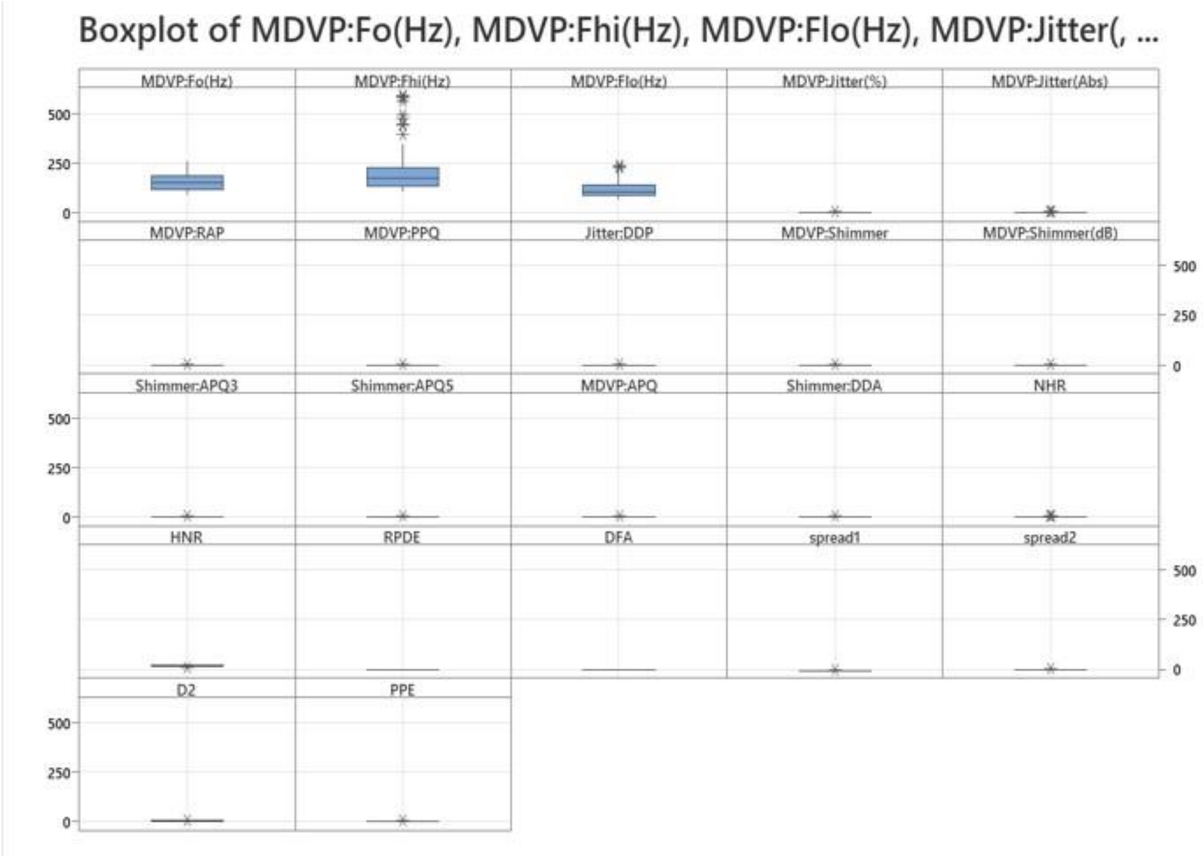
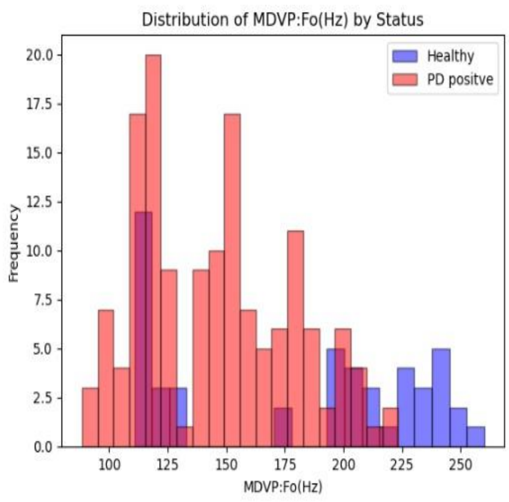


Fig. 3 Boxplot of Features



Distribution of MDVP:Fo(Hz)

Fig. 4

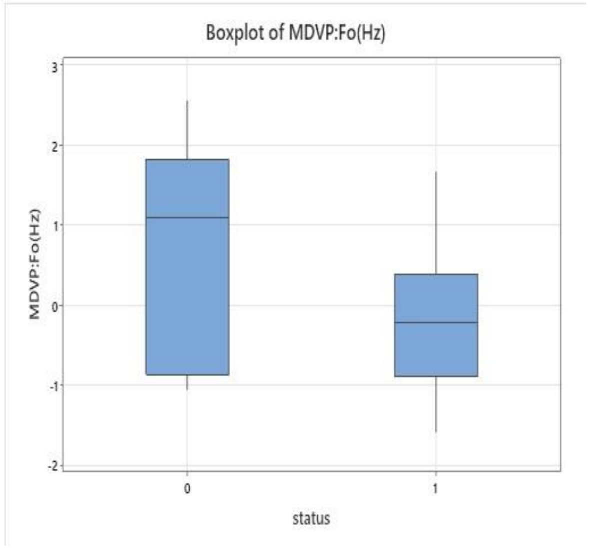


Fig. 5

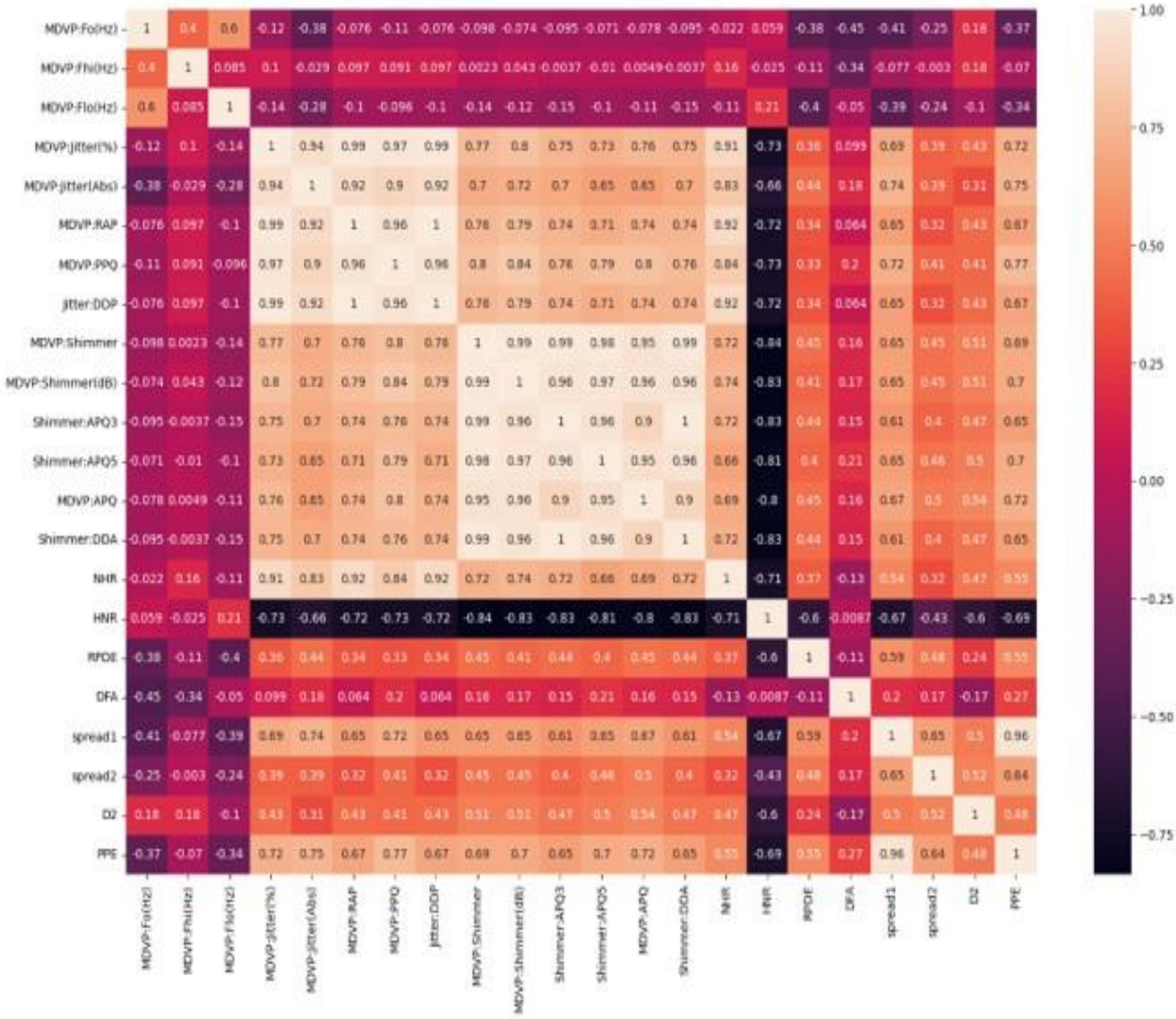


Fig. 6 Correlation Heatmap

3.3 Training Base Models

To build a robust predictive model for Parkinson's disease, six machine learning algorithms were selected as base models: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes (NB). These models were chosen due to their diverse learning approaches, interpretability, and widespread application in medical diagnosis tasks. Each model contributes unique strengths to the analysis, ensuring a comprehensive evaluation of the dataset. An 80:20 train-test split was employed to train and validate these models, ensuring that the evaluation metrics reflected real-world performance on unseen data.

Logistic Regression (LR) was selected for its simplicity and interpretability, making it a reliable baseline model for binary classification problems. This algorithm also provides insights into feature importance and linear separability of the data. Decision Tree (DT), on the other hand, was included for its capability to handle non-linear relationships and its intuitive, interpretable structure. Its ability to focus on the most discriminative features makes it effective in dealing with imbalanced datasets.

Random Forest (RF) was chosen for its ensemble learning approach, which combines multiple decision trees to improve predictive accuracy and robustness against overfitting. Support Vector Machine (SVM) was selected for its effectiveness in handling high-dimensional data and its ability to identify optimal hyperplanes, making it particularly suitable for datasets with complex patterns.

K-Nearest Neighbors (KNN) was included for its simplicity and ability to capture local data structures, which is beneficial in cases with irregular decision boundaries. Finally, Naive Bayes (NB) was selected for its probabilistic approach, which is computationally efficient and particularly advantageous for small datasets. Together, these models provide a comprehensive foundation for evaluating the predictive performance of machine learning techniques on the Parkinson's disease dataset.

Each model was trained on the dataset using the 80% training set, and its performance was evaluated on the 20% test set using accuracy, F1-score, and confusion matrix to account for class imbalance and overall prediction quality. Logistic Regression achieved an accuracy of 74.35%, with F1-scores of 0.55 and 0.82 for class 0 and class 1, respectively. The confusion matrix for LR showed [[6, 5], [5, 23]], indicating challenges in correctly identifying class 0 instances, thus reflecting the class imbalance in the data set.

Decision Tree demonstrated superior performance with an accuracy of 89.74% and F1-scores of 0.80 for class 0 and 0.93 for class 1. Its confusion matrix [[6, 5], [5, 23]] reflected its ability to better balance predictions for both classes but still showed evidence of the underlying class imbalance. The optimal decision tree was plotted (Fig. 7) to demonstrate how the dataset is classified based on key features and their thresholds. At the root node, the feature PPE serves as the primary decision criterion, with a threshold of 0.133993 dividing the data. Samples with lower PPE values tend to belong to Class 0, while those with higher PPE values predominantly fall under Class 1. Subsequent splits further refine the classification based on additional features such as Shimmer:APQ5, MDVP:Fo(Hz), and Shimmer:APQ3. Notably, Class 1 is the majority class, dominating most of the nodes, particularly in Terminal Node 5, where 99% of samples are correctly classified as Class 1. On the other hand, Class 0 is more prevalent in Terminal Nodes 1, 2, and 4. The tree highlights that certain thresholds, such as Shimmer:APQ5 0.012745 or MDVP:Fo(Hz) -0.877878, are critical for distinguishing between the two classes. Overall, the decision tree captures the underlying relationships in the data, with each path from the root to a terminal node representing a sequence of decisions that lead to a final classification.

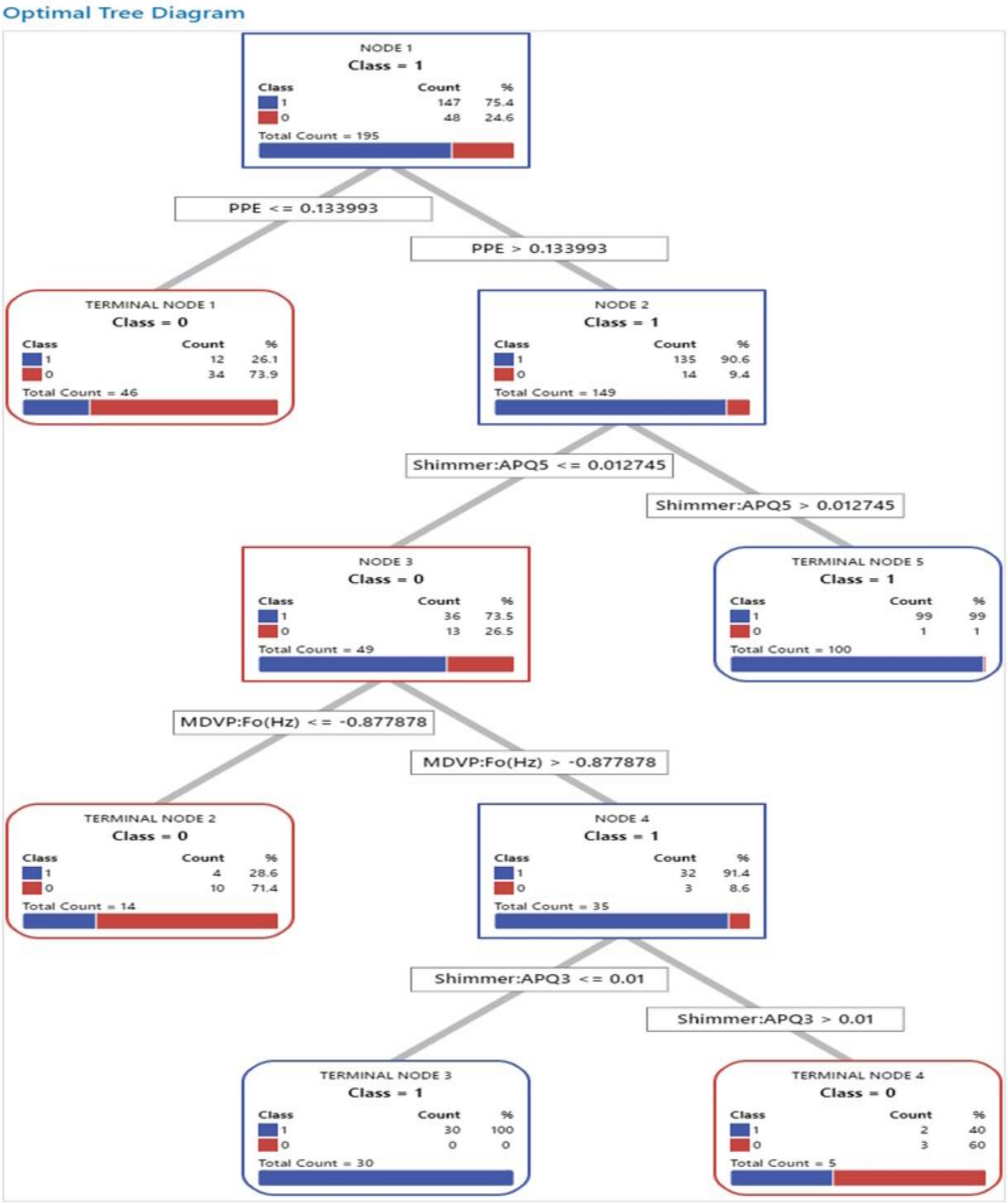
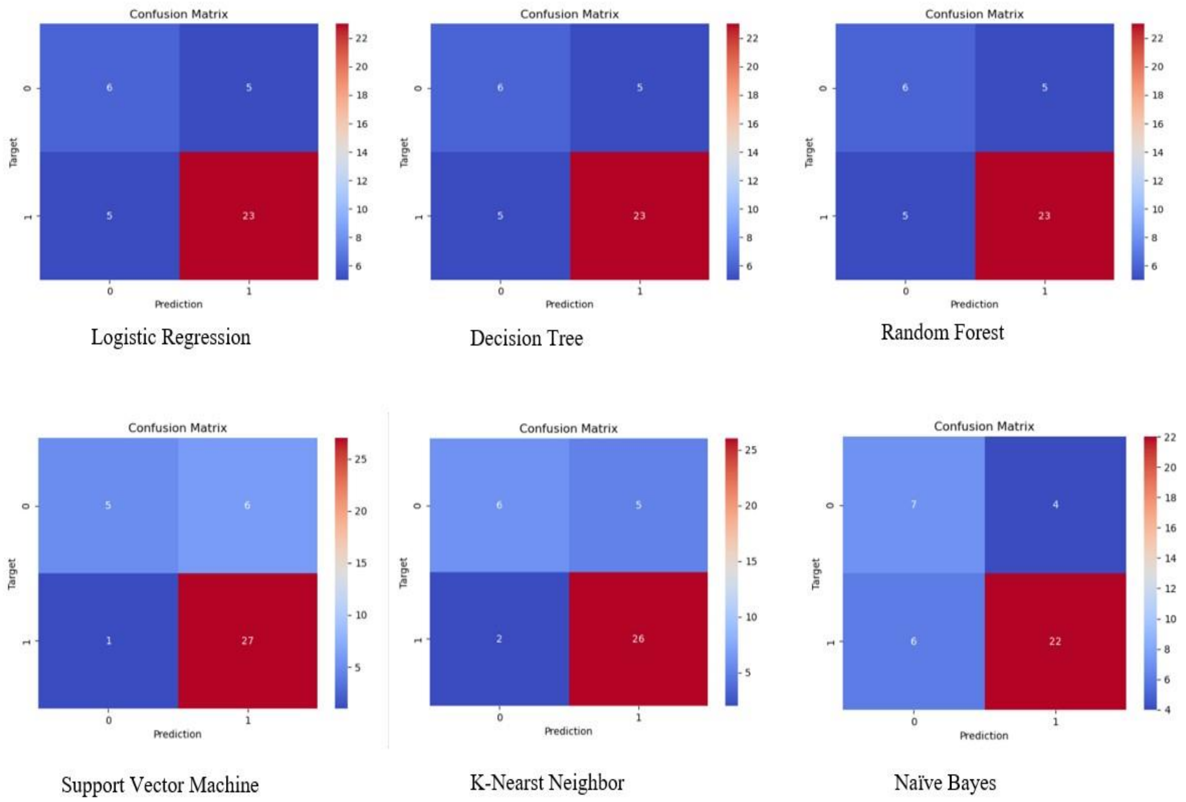


Fig. 7 Optimal Tree Diagram

Similarly, Random Forest achieved high accuracy (87.17%) and F1-scores of 0.71 and 0.92 for class 0 and class 1, respectively, with a confusion matrix identical to that of the Decision Tree, again showing the class imbalance affecting the minority class predictions. Support Vector Machine achieved an accuracy of 82.05% and F1-scores of 0.59 for class 0 and 0.89 for class 1. The confusion matrix [[5, 6], [1, 27]] highlighted its strong performance in identifying class 1 but relatively weaker performance for class 0, further evidencing the class imbalance. K-Nearest Neighbors also recorded an accuracy of 82.05%, with F1-scores of 0.63 for class 0 and 0.88 for class 1. Its confusion matrix [[6, 5], [2, 26]] suggested unbalanced performance ,difficulty in correctly identifying instances of the minority class.

Naive Bayes, the final base model, achieved an accuracy of 74.35%, with F1-scores of 0.58 for class 0 and 0.81 for class 1. Its confusion matrix [[7, 4], [6, 22]] revealed its limitations in correctly classifying instances of class 0 due to the class imbalance.

Overall, the results demonstrated the impact of class imbalance across all models (Fig. 8), with higher F1-scores and better performance for the majority class (class 1) compared to the minority class (class 0). Among the models, Decision Tree and Random Forest emerged as the top performers, achieving higher accuracy and F1scores. Logistic Regression and Naive Bayes, while relatively less accurate, provided essential baseline comparisons. This comprehensive evaluation offers valuable insights into model behavior and identifies potential candidates for further optimization and ensemble techniques, particularly for addressing the class imbalance issue.



Logistic Regression

Decision Tree

Random Forest

Support Vector Machine

K-Nearst Neighbor

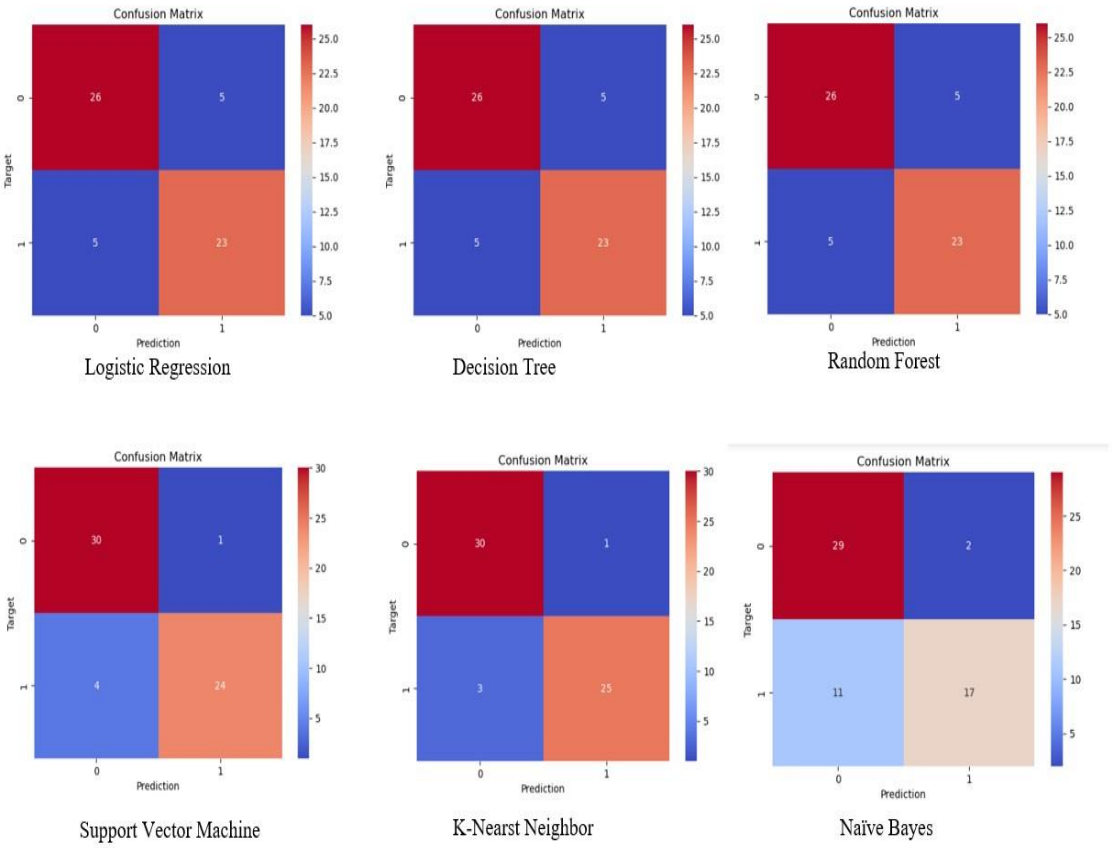
Naïve Bayes

Fig. 8 Imbalanced Confusion matrices of Base Models

3.4 SMOTE-TomekLinks and Class imbalance

Class imbalance is a common issue in medical datasets as was apparent from our case, where one class, such as the absence of Parkinson’s disease (PD) (represented by 0), is underrepresented compared to the other class, such as the presence of Parkinson’s disease (PD) (represented by 1). This imbalance can lead to machine learning models being biased toward the majority class, often resulting in poor predictive performance for the minority class, which is of particular interest in medical diagnostics. To address this issue, SMOTE-TomekLinks was applied in this study. SMOTE-TomekLinks is a hybrid technique that combines two methods: SMOTE and TomekLinks. SMOTE (Synthetic Minority Over-sampling Technique) is an oversampling technique that generates synthetic samples for the minority class by creating new data points based on the existing minority class instances. This helps balance the dataset by increasing the representation of the underrepresented class. On the other hand, TomekLinks is an undersampling technique that identifies and removes pairs of instances from different classes that are closest to each other in the feature space. These instances are often noisy or borderline, and their removal helps improve the overall quality of the dataset by reducing overfitting and enhancing the model's generalization capability.

After applying SMOTE-TomekLinks, the performance of various machine learning models improved. Logistic Regression (LR) achieved an accuracy of 83.05%, with F1scores of 0.84 for Class 0 (absence of PD) and 0.82 for Class 1 (presence of PD). The Decision Tree (DT) model showed an accuracy of 93.22% and F1-scores of 0.94 for Class 0 and 0.93 for Class 1. Random Forest (RF) performed the best, with an accuracy of 96.61% and F1-scores of 0.97 for Class 0 and 0.96 for Class 1. Support Vector Machine (SVM) also performed well, with an accuracy of 91.52% and F1-scores of 0.92 for Class 0 and 0.91 for Class 1. The K-Nearest Neighbors (KNN) model achieved an accuracy of 93.22% and F1-scores of 0.94 for Class 0 and 0.93 for Class 1. The Naive Bayes (NB) model had a lower accuracy of 77.96% and F1-scores of 0.82 for Class 0 and 0.72 for Class 1. Overall, SMOTE-TomekLinks significantly improved the classification performance across all models by addressing class imbalance (Fig. 9) through both oversampling (via SMOTE) and undersampling (via TomekLinks), with Random Forest achieving the highest accuracy and F1-scores, followed by Decision Tree and Support Vector Machine. These results demonstrate the effectiveness of SMOTE-TomekLinks in improving model performance and handling class imbalance in medical datasets



Logistic Regression

Decision Tree

Random Forest

Support Vector Machine

K-Nearst Neighbor

Naïve Bayes

Fig. 9 Balanced Confusion matrices of Base Models After SMOTE-TomekLinks

3.5 SHAP analysis

SHAP (SHapley Additive exPlanations) analysis (Fig.10) was performed to assess the contribution of individual features across various machine learning models in predicting the presence of Parkinson's Disease (PD). The SHAP values provided a comprehensive understanding of feature importance, offering valuable insights into the decision-making processes of Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes models.

In Logistic Regression, features such as spread1 (1.00), spread2 (0.61), RPDE (0.61), and PPE (0.48) emerged as dominant contributors, highlighting their clinical relevance in capturing non-linear dynamics and variability in voice, which are indicative of PD-related impairments. Additionally, jitter-related features, including MDVP:Jitter(%) (0.34) and MDVP:Jitter(Abs) (0.37), reflected sensitivity to pitch irregularities, further underscoring their importance. The Decision Tree model, on the other hand, exhibited a more even reliance on a broader set of features, with MDVP:Fo(Hz) (0.17),MDVP:APQ (0.15) making moderate contributions. This broader feature utilization indicates the Decision Tree's ability to integrate both prominent and less dominant variables for classification.

Random Forest demonstrated a balanced distribution of feature contributions, with spread1 (0.06), PPE (0.09), and spread2 (0.04) standing out as significant features. The model effectively combined jitter- and shimmer-related variables with other features to achieve robust performance. Similarly, SVM strongly relied on spread1 (0.09), spread2 (0.10), and D2 (0.08), emphasizing the importance of capturing variability and distribution in voice data. Interestingly, SVM showed lower reliance on shimmer-related features, suggesting a preference for frequency-based variables in its hyperplane-based decision-making. For KNN, the top contributing features were spread2 (0.06), RPDE (0.04), and DFA (0.04), indicating a focus on broader trends rather than localized variations in voice data. Naive Bayes exhibited a similar pattern, with spread2 (0.06), RPDE (0.04), and DFA (0.04) emerging as significant contributors, reinforcing the importance of frequency-domain features while downplaying jitter and shimmer variations. Across most models, MDVP:Fo(Hz),spread1 and spread2 consistently ranked as top features, particularly in Logistic Regression, SVM, and Random Forest, emphasizing their critical role in quantifying amplitude variations in voice signals and diagnosing PD. RPDE and PPE, which capture non-linear dynamics and pitch variations, also played significant roles, further validating their relevance in distinguishing PD-positive individuals. While jitter- and shimmer-related features were significant in specific models like

This diversity in feature importance underscores the complementary strengths of different machine learning models, suggesting that ensemble or hybrid approaches may offer enhanced diagnostic accuracy for PD. Overall, the SHAP analysis underscores the clinical relevance of acoustic features, validates the interpretability of the models, and provides actionable insights for refining PD diagnostic tools to meet clinical needs.

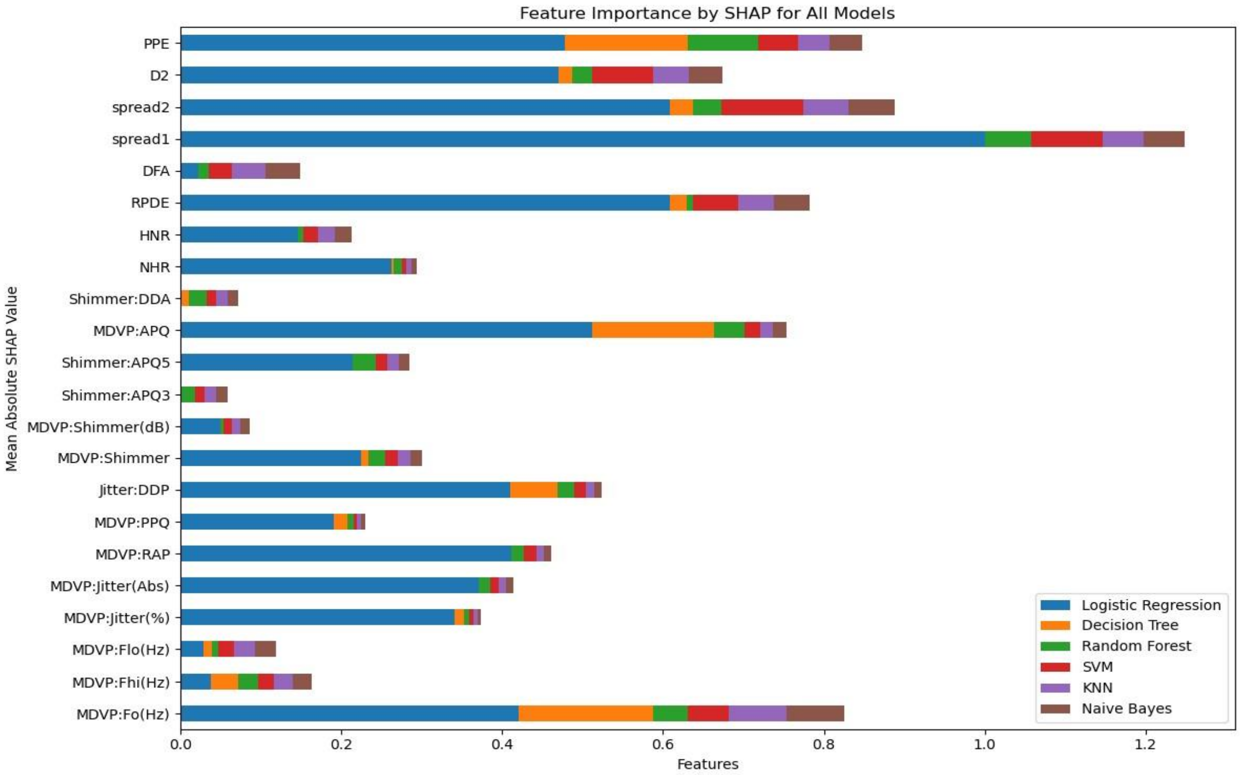


Fig. 10 Feature Importance by SHAP for all Models

4 Results

4.1 Comparative analysis

Table 2. Comparative Analysis of Base Models with and without SMOTE-TomekLinks

Base Models	Without SMOTE-TomekLinks		With SMOTE-TomekLinks	
	Accuracy	F1-Score(Class 0/Class1)	Accuracy	F1-Score(Class 0 /Class 1)
1.Logistic Regression	74.35%	0.55	83.05%	0.84
		0.82		0.82
2.Decision Tree	89.74%	0.80	93.22%	0.94
		0.93		0.93
3.Random Forest	87.19%	0.71	96.61%	0.97
		0.92		0.96
4.Support Vector Machine	82.05%	0.59	91.52%	0.92
		0.89		0.91
5.K-Nearst Neighbors	82.05%	0.63	93.22%	0.94
		0.88		0.93
6.Naive Bayes	74.35%	0.58	77.96%	0.82
		0.81		0.72

To evaluate the impact of addressing class imbalance on predictive performance, a comparative analysis (Table 2) of Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes (NB) was conducted. Key metrics such as accuracy and F1-scores for both classes (Class 0 and Class 1) were assessed. This analysis was integral to the Parkinson's disease project, where accurate identification of the minority class (absence of Parkinson's disease, Class 0) is critical for reliable diagnostic predictions.

Without SMOTE-TomekLinks

Without addressing class imbalance, the models exhibited varied performance, often showing bias toward the majority class (presence of Parkinson's disease, Class 1).

Logistic Regression (LR): Accuracy was 74.35%, with F1-scores of 0.55 (Class 0) and 0.82 (Class 1). This indicates a moderate ability to classify the minority class but struggles to achieve balance between the two classes.

Decision Tree (DT): Achieved the highest accuracy (89.74%) among all models, with F1-scores of 0.80 (Class 0) and 0.93 (Class 1). However, its reliance on data splits led to overfitting in the imbalanced dataset.

Random Forest (RF): Accuracy was 87.19%, and F1-scores were 0.71 (Class 0) and 0.92 (Class 1), highlighting less generalizability than Decision Tree and favoring the majority class.

Support Vector Machine (SVM): Accuracy was 82.05%, with F1-scores of
0.59 (Class 0) and 0.89 (Class 1), indicating limited effectiveness in handling class imbalance.

K-Nearest Neighbors (KNN): With an accuracy of 82.05%, F1-scores were 0.63 (Class 0) and 0.88 (Class 1), reflecting difficulty in distinguishing minority class instances.

Naive Bayes (NB): Accuracy was 74.35%, with F1-scores of 0.58 (Class 0) and 0.81 (Class 1), showing significant limitations in managing imbalanced data distributions.

With SMOTE-TomekLinks

After applying SMOTE-TomekLinks, all models showed improved performance across all metrics, particularly for the minority class (absence of Parkinson's disease, Class0).

Logistic Regression (LR): Accuracy increased to 83.05%, with F1-scores improving to 0.84 (Class 0) and 0.82 (Class 1). This reflects a better balance between the two classes, enhancing its diagnostic utility.

Decision Tree (DT): Accuracy rose to 93.22%, while F1-scores remained high at 0.94 (Class 0) and 0.93 (Class 1), indicating an improved ability to generalize across classes.

Random Forest (RF): Achieved the highest accuracy (96.61%) and F1-scores of 0.97 (Class 0) and 0.96 (Class 1), demonstrating its robustness and suitability for balanced datasets.

Support Vector Machine (SVM): Accuracy increased to 91.52%, with F1scores improving to 0.92 (Class 0) and 0.91 (Class 1), highlighting a more balanced classification performance.

K-Nearest Neighbors (KNN): Accuracy improved to 93.22%, with balanced F1-scores of 0.94 (Class 0) and 0.93 (Class 1), indicating enhanced capability in classifying both classes effectively.

Naive Bayes (NB): Accuracy improved modestly to 77.96%, with F1-scores increasing to 0.82 (Class 0) and 0.72 (Class 1), though it remained the weakest performer among all models.

Comparative Insights

Accuracy Improvements: Across all models, applying SMOTE-TomekLinks resulted in significant accuracy improvements. Random Forest emerged as the most accurate model (96.61%), followed closely by Decision Tree (93.22%) and KNN (93.22%). Logistic Regression and SVM also demonstrated marked gains, emphasizing the effectiveness of addressing class imbalance.

F1-Score Trends: For the minority class (Class 0), Random Forest achieved the highest F1-score (0.97), followed by Decision Tree (0.94) and SVM (0.92). Logistic Regression and KNN showed balanced performance, while Naive Bayes exhibited the least improvement.

Model Robustness: Random Forest consistently outperformed other models in all metrics, showcasing its ability to handle balanced data effectively. SVM and KNN also demonstrated strong performance post-SMOTE-TomekLinks, indicating their suitability for diagnostic tasks.

Naive Bayes Limitations: While Naive Bayes benefitted from SMOTE-TomekLinks, its simplistic assumptions limited its performance compared to other models, highlighting the need for more advanced algorithms in this context

The comparative analysis underscores the pivotal role of SMOTE-TomekLinks in mitigating class imbalance and improving model performance in Parkinson's disease diagnosis. Random Forest emerged as the most reliable model, achieving superior accuracy and balanced F1-scores. These findings reinforce the necessity of class-balancing techniques to ensure equitable predictions, particularly in medical applications where identifying minority class instances is critical. In next section this analysis serves as a foundation for integrating ensemble or hybrid models to further enhance diagnostic accuracy in Parkinson's disease detection.

4.2 Ensemble Technique: XGBoost and AdaBoost

To enhance the predictive performance of the model in diagnosing Parkinson's disease, ensemble techniques employing XGBoost and AdaBoost were utilized as meta-models. Ensemble learning combines the predictions of multiple models to achieve better accuracy and robustness compared to individual classifiers.

XGBoost (Extreme Gradient Boosting) was chosen due to its efficiency in handling complex data and imbalanced datasets. It combines gradient boosting with regularization techniques, which mitigates overfitting and enhances generalization. Its scalability and ability to process imbalanced dataset made it an ideal choice for this study. AdaBoost (Adaptive Boosting), on the other hand, iteratively adjusts the weights of misclassified instances, focusing on difficult samples to improve overall classification performance.

A comparative analysis of XGBoost and AdaBoost is presented in Table 3. XGBoost achieved the highest accuracy of 98.30%, with F1-scores of 0.98 for Class 0 (absence of Parkinson's disease) and 0.98 for Class 1 (presence of Parkinson's disease). It also recorded an ROC-AUC score of 0.98, indicating excellent discriminatory power. The confusion matrix (Fig. 11) for XGBoost highlighted 30 true negatives, 28 true positives, and only 1 misclassification.

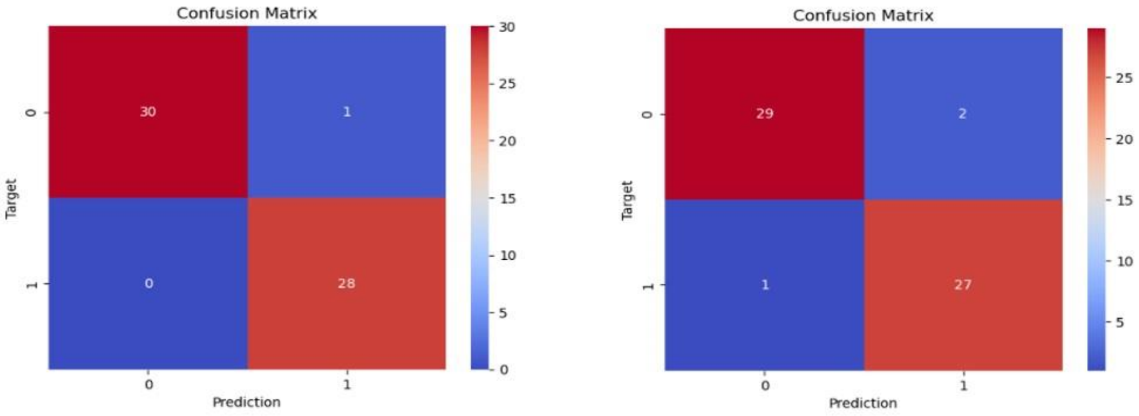
In comparison, AdaBoost achieved an accuracy of 94.91%, with F1-scores of 0.95 for Class 0 and 0.95 for Class 1. Its ROC-AUC score was 0.95, with the confusion matrix showing 29 true negatives, 27 true positives, and 3 misclassifications. While both meta-models benefited from the use of SMOTE-TomekLinks to handle class imbalance, XGBoost emerged as the superior technique for this task.

By integrating ensemble methods with balanced data, the model demonstrated significant improvements in predictive performance. The use of SMOTE-TomekLinks ensured that the minority class (absence of Parkinson's disease) was adequately represented, leading to better generalization and reduced bias. These ensemble techniques effectively addressed the challenges posed by class imbalance, providing a robust and reliable model for clinical decision-making in Parkinson's disease diagnosis.

Table 3 Performance Comparison of Ensemble Techniques

Meta-model	Accuracy (%)	F1-Score (Class 0)	F1-Score (Class 1)	ROC-AUC	Confusion Matrix
XGBoost	98.30	0.98	0.98	0.98	[[30,1],[0,28]]

AdaBoost	94.91	0.95	0.95	0.95	[[29,2],[1,27]]



XgBoost

AdaBoost

Fig. 11 Confusion Matrices of Meta-Models

5 Conclusion

This study aimed to develop a machine learning model to diagnose Parkinson's disease using voice measurements data while addressing the challenges posed by class imbalance and ensuring model interpretability. The research hypothesized that advanced preprocessing methods, class balancing and ensemble techniques could significantly enhance diagnostic accuracy.

The major findings confirmed this hypothesis, as models trained on SMOTE-TomekLinks balanced data demonstrated notable improvements in performance. Among all approaches, ensemble techniques like XGBoost and AdaBoost outperformed base models, achieving accuracies of 98.30% and 94.91%, respectively. XGBoost, in particular, demonstrated superior discriminatory power with an ROC-AUC score of 0.98 and high F1-scores for both classes, highlighting its robustness in handling medical datasets. SHAP analysis further added value by making the model's predictions interpretable for clinical use.

The relevance of this work lies in its ability to bridge the gap between computational methods and clinical diagnostics, providing a reliable framework for early detection of Parkinson's disease. By addressing class imbalance and ensuring interpretability, the study contributes to the development of practical machine learning applications in healthcare.

However, this study has some limitations. The dataset used was relatively small, which may affect the generalizability of the findings. Additionally, the analysis was limited to selected machine learning algorithms and did not explore deep learning approaches or advanced real-time validation techniques.

Future research should focus on using larger, more diverse datasets, exploring real-time clinical validation, and incorporating advanced feature selection methods to improve the reliability and scalability of these models. Furthermore, integrating other biomarkers and longitudinal patient data could enhance the diagnostic accuracy and applicability of the model in real-world clinical settings.

In conclusion, this study provides a strong foundation for using machine learning to diagnose Parkinson's disease, showcasing how ensemble techniques and balanced data can improve accuracy and support clinical decision-making.

6 Program codes

Listing 1 Base Model Building

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
# evaluation:
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
class ModelTrainer:
def __init__(self,x_train,x_test,y_train,y_test):
self.x_train=x_train
self.x_test=x_test
self.y_train=y_train
self.y_test=y_test

def logistic_regression(self):
model=LogisticRegression()
model.fit(self.x_train,self.y_train)
y_pred=model.predict(self.x_test)
return
accuracy_score(self.y_test,y_pred),confusion_matrix(self.y_test,y_pred),classification_report(self.y_test,y_pred)
def decision_tree(self):
model=DecisionTreeClassifier()
model.fit(self.x_train,self.y_train)
y_pred=model.predict(self.x_test)
return
accuracy_score(self.y_test,y_pred),confusion_matrix(self.y_test,y_pred),classification_report(self.y_test,y_pred)
def random_tree_forest(self):
model=RandomForestClassifier()
model.fit(self.x_train,self.y_train)
y_pred=model.predict(self.x_test)
return
accuracy_score(self.y_test,y_pred),confusion_matrix(self.y_test,y_pred),classification_report(self.y_test,y_pred)
def SVM(self):
model=SVC()
model.fit(self.x_train,self.y_train)
y_pred=model.predict(self.x_test)
return
accuracy_score(self.y_test,y_pred),confusion_matrix(self.y_test,y_pred),classification_report(self.y_test,y_pred)
```

```
def KNN(self):
model=KNeighborsClassifier(n_neighbors=5)
model.fit(self.x_train, self.y_train)
y_pred=model.predict(self.x_test)
return
accuracy_score(self.y_test,y_pred),confusion_matrix(self.y_test,y_pred),classification_report(self.y_t
est,y_pred)
def NaiveBayes(self):
model=GaussianNB()
model.fit(self.x_train, self.y_train)
y_pred=model.predict(self.x_test)
return
accuracy_score(self.y_test,y_pred),confusion_matrix(self.y_test,y_pred),classification_report(self.y_t
est,y_pred)

trainer=ModelTrainer(x_train,x_test,y_train,y_test)

accuracy, confusion_mat, class_report = trainer.logistic_regression() # Similarly call other Base models
methods
print("Accuracy:", accuracy)
print('-------------------------')
print("Confusion Matrix:\n", confusion_mat)
print('-------------------------')
print("Classification Report:\n", class_report)
```

Listing 2 SMOTE-TomekLinks

```
import pandas as pd
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import TomekLinks
from imblearn.pipeline import Pipeline
smote=SMOTE(sampling_strategy='auto', random_state=42)
tomek = TomekLinks()
# Create a pipeline that first applies SMOTE, then Tomek Links
pipeline = Pipeline(steps=[('smote', smote), ('tomek', tomek)])
# Fit and resample the training set
X_resampled, y_resampled = pipeline.fit_resample(X_train, y_train)
```

Listing 3 SHAP Analysis

```
# SHAP for LR
lr_explainer = shap.LinearExplainer(lr_model,x_train)
lr_shap_values = lr_explainer.shap_values(x_test)
lr_shap_values.shape
# SHAP for Decision Tree
dt_explainer = shap.TreeExplainer(dt_model)
dt_shap_values = dt_explainer.shap_values(x_test)
dt_shap_values.shape
# Shap values for Random-forest:
rf_explainer = shap.TreeExplainer(rf_model)
rf_shap_values = rf_explainer.shap_values(x_test)
rf_shap_values.shape
# SHAP for SVM (using KernelExplainer)
svm_explainer = shap.KernelExplainer(svm_model.predict_proba, x_train)
svm_shap_values = svm_explainer.shap_values(x_test)
# SHAP for KNN
knn_explainer=shap.KernelExplainer(knn_model.predict_proba,x_train)
knn_shap_values=knn_explainer.shap_values(x_test)
#SHAP for Naive bayes
nb_explainer=shap.KernelExplainer(nb_model.predict_proba,x_train)
nb_shap_values=knn_explainer.shap_values(x_test)
import numpy as np
# For Logistic Regression (already 1D per feature)
lr_feature_importance = np.abs(lr_shap_values).mean(axis=0)

# For multiclass models (aggregate across the class dimension)
dt_feature_importance = np.abs(dt_shap_values).mean(axis=(0, 2))
rf_feature_importance = np.abs(rf_shap_values).mean(axis=(0, 2))
svm_feature_importance = np.abs(svm_shap_values).mean(axis=(0, 2))
knn_feature_importance = np.abs(knn_shap_values).mean(axis=(0, 2))
nb_feature_importance = np.abs(nb_shap_values).mean(axis=(0, 2))
features=x_train.columns
# Combine into a DataFrame
shap_df = pd.DataFrame({
'Logistic Regression': lr_feature_importance,
'Decision Tree': dt_feature_importance,
'Random Forest': rf_feature_importance,
'SVM': svm_feature_importance,
'KNN': knn_feature_importance,
'Naive Bayes': nb_feature_importance
}, index=features)
```

```
print(shap_df)
features=x_train.columns
# Create DataFrame
shap_df = pd.DataFrame({
'Logistic Regression': lr_feature_importance,
'Decision Tree': dt_feature_importance,
'Random Forest': rf_feature_importance,
'SVM': svm_feature_importance,
'KNN': knn_feature_importance,
'Naive Bayes': nb_feature_importance
}, index=features)

# Plot grouped bar chart
fig, ax = plt.subplots(figsize=(12, 8))
shap_df.plot(kind='barh', stacked=True, ax=ax)
plt.title("Feature Importance by SHAP for All Models")
plt.ylabel("Mean Absolute SHAP Value")
plt.xlabel("Features")
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```

Listing 4 Ensemble Models

```
# ENSEMBLE XGBOOST
# from sklearn.ensemble import StackingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
# Split dataset treated with smote-tomeklinks
x_train, x_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2,
random_state=100)

# Define base models
base_models = [
('lr', LogisticRegression()),
. ('dt', DecisionTreeClassifier()),
. ('rf', RandomForestClassifier()),
. ('svm', SVC(probability=True)),
. ('knn', KNeighborsClassifier()),
. ('nb', GaussianNB())
. ]
.
. # Define meta-model
. meta_model = XGBClassifier(use_label_encoder=False, eval_metric='logloss',
random_state=100)
.
. # Create Stacking_model
. stacking_model = StackingClassifier(estimators=base_models, final_estimator=meta_model,
cv=5)
.
. # Train and evaluate
. stacking_model.fit(x_train, y_train)
. y_pred = stacking_model.predict(x_test)
.
. # Accuracy
. print("Stacking Classifier Accuracy:", accuracy_score(y_test, y_pred))
. print("Stacking Classifier Confusion Matrix:""\n', confusion_matrix(y_test, y_pred))
. print("Stacking Classifier Classification Report:""\n', classification_report(y_test, y_pred))
. # ROC-AUC score
. from sklearn.metrics import roc_auc_score
. roc_auc = roc_auc_score(y_test, y_pred)
. print(f"ROC-AUC: {roc_auc:.2f}")
.
.
. # ENSEMBLE AdaBOOST
. from sklearn.ensemble import StackingClassifier
. from sklearn.linear_model import LogisticRegression
. from sklearn.tree import DecisionTreeClassifier
. from sklearn.ensemble import RandomForestClassifier
. from sklearn.svm import SVC
. from sklearn.neighbors import KNeighborsClassifier
. from sklearn.naive_bayes import GaussianNB
. from sklearn.ensemble import AdaBoostClassifier
. from sklearn.metrics import accuracy_score
. from sklearn.model_selection import train_test_split
. # Split dataset treated with smote-tomeklinks
. x_train, x_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2,
random_state=100)
```

```

# Define base models
base_models = [
('lr', LogisticRegression()),
('dt', DecisionTreeClassifier()),
('rf', RandomForestClassifier()),
('svm', SVC(probability=True)),
('knn', KNeighborsClassifier()),
('nb', GaussianNB())
]

# Define meta-model
meta_model = AdaBoostClassifier(n_estimators=50, random_state=100)

# Create Stacking_model
stacking_model = StackingClassifier(estimators=base_models, final_estimator=meta_model, cv=5)

# Train and evaluate
stacking_model.fit(x_train, y_train)
y_pred = stacking_model.predict(x_test)

# Accuracy
print("Stacking Classifier Accuracy:", accuracy_score(y_test, y_pred))
print("Stacking Classifier Confusion Matrix:""\n', confusion_matrix(y_test, y_pred))
print("Stacking Classifier Classification Report:""\n', classification_report(y_test, y_pred))
# ROC-AUC score
from sklearn.metrics import roc_auc_score
roc_auc = roc_auc_score(y_test, y_pred)
print(f"ROC-AUC: {roc_auc:.2f}")
```

Declarations

Software and Resources:

The research was conducted using the following software and resources:

Python 3.8 and libraries including scikit-learn, pandas, numpy, matplotlib, imblearn, and xgboost for data processing, model development, and evaluation.

Jupyter Notebook was used as the integrated development environment (IDE) for model implementation and analysis.

Minitab for Statistical Analysis and Plots

Data Availability:
The dataset used in this study is publicly available from the UCI Machine Learning Repository. The specific dataset is the **Parkinson's Disease Classification dataset**, based on a study by: Max A. Little, E.J.H.L.O.R. Patrick E. McSharry, "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," IEEE Transactions on Biomedical Engineering, 2008.

Use of AI Tools:
I, the author of this manuscript, declare that AI tools have been used to enhance the grammar, spelling, and language readability of the document. Specifically, tools such as grammar-checking software and readability enhancers like Grammarly, Quiltbot were employed to ensure clarity and consistency in the presentation of my research. After using these tools, I reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

1. Nico J. Diederich, C.G.G.: Parkinson's disease: Overstrain focused on basal ganglia and brainstem nuclei. OXFORD ACADEMIC (2024) https://doi.org/10. 1093/med/9780197676592.003.0010
2. Salauddin, M.U.S.S.M.J.N.S.K. Syed Amir Azam Zaidi, Alam, O.: Parkinson's disease: A progressive neurodegenerative disorder and structure-activity relationship of mao inhibitor scaffolds as an important therapeutic regimen. BENTHAM SCIENCE (2024) https://doi.org/10.2174/0118715273324300241010054029
3. Joseph, C.B.: Parkinson disease. Journal of Consumer Health on the Internet (2023) https://doi.org/10.1080/15398285.2023.2212529
4. R. Balestrino, A.H.V.S.: Parkinson disease. European Journal of Neurology (2019) https://doi.org/10.1111/ene.14108
5. Intan Sahara Zein, K.K.: Parkinson's disease. LPKD (2023) https://doi.org/10. 55606/jurrike.v2i2.1701
6. Irin Akter Liza, M.M.H.S.F.H.M.A.A..S.A. Ekramul Hasan: Predictive modeling and early detection of parkinson's disease using machine learning. Journal of Medical and Health Studies. (2024) https://doi.org/10.32996/jmhs.2024.5.4.12
7. Abiona Akeem Adekunle, A.M.O.: Early parkinson's disease detection using machine learning approach. Asian Journal of Research in Computer Science. (2023) https://doi.org/10.9734/ajrcos/2023/v16i2337
8. Ruby Dahiya, D.N.A.L.P.M.E.M. Virendra Kumar Dahiya: Predictive modelling for parkinson's disease diagnosis using biomedical voice measurements. EAI Endorsed Transactions on Pervasive Health and Technology (2024) https: //doi.org/10.4108/eetpht.10.5519
9. Pawan Kumar Badhan, M.K.: Early detection of parkinson disease through biomedical speech and voice analysis. International journal on soft computing, artificial intelligence and applications (2024)
10. Shaharina Shoha, S.A.A.S.M.S.A.D.A.H.S.R.Z.M.A.I.M. Shake Ibna Abir: Enhanced parkinson's disease detection using advanced vocal features and machine learning. Journal of computer science and technology studies (2024) https://doi.org/10.32996/jcsts.2024.6.5.10
11. Anila Raveendran Nambiar, A.R.K.A.: Machine learning based parkinson's disease prediction. QTanalytics (2024) https://doi.org/10.48001/ 978-81-966500-2-5-6

12. Mohamed Ziyadh M, K.S.S.A. Bavisetti Siva Avinash: Integration of voice and handwritten tests analysis for enhanced detection of parkinson's disease using machine learning. IEEE Xplore (2024) https://doi.org/10.1109/I-SMAC61858. 2024.10714637
13. Rosevir Singh, A.K. Sachin Ahuja: Enhancing the parkinson's disease detection through machine learning and feature engineering. IEEE Xplore (2024) https: //doi.org/10.1109/ICCCNT61001.2024.10724102
14. Tian, Y.: The prediction of parkinson's disease based on pearson coefficient feature screening and machine learning. EWA (2024) https://doi.org/10.54254/ 2755-2721/67/20240628
15. SHANKARA GOWDA B, B.G.: Detection of parkinson's disease detection through machine learning. Indian Scientific Journal Of Research In Engineering And Management (2024) https://doi.org/10.55041/IJSREM36505
16. Shaikh Haque Naema, M.B.: Advanced deep learning methodologies in the diagnosis of parkinson's disease: A comprehensive review. Indian Scientific Journal Of Research In Engineering And Management (2024) https://doi.org/10.55041/ IJSREM36005
17. Max A. Little, E.J.H.L.O.R. Patrick E. McSharry: Suitability of dysphonia measurements for telemonitoring of parkinson's disease. IEEE Transactions on Biomedical Engineering (2008)