Enhancing Parkinson's Disease Detection with PCA and Machine Learning

Kezal Chavda - 40293289
GitHub Link: https://github.com/Kezal19/Kezal-Chavda$_INSE6220 - Report$

***Abstract*—Parkinson's disease is essentially a degenerative condition pertaining to the gradual movement of an individual which can be very helpful in managing symptoms when caught at an onset and ascertained to what extent it has progressed. This particular project would evaluate the impact of voice recordings from individuals as voice changes would often show earlier signs such as jitter, shimmer, and noise levels indicative of a possibility of Parkinson's.**

We used PCA, Principal Component Analysis. Assuming this could help, it reduces the number of features by combining them smartly while still retaining most of the actual data. Two machine learning models-LR, KNN were trained on original and PCA transformed data for predicting symptom severity. Evaluation was done using confusion matrix, ROC curve, decision boundary plot, bi-plot, and pair plot.

Finally, Logistic Regression is found to perform better than KNN in differentiating among low-and high-motor-symptom individuals.

***Index Terms*—Parkinson's disease, Principal Component Analysis, dimensionality reduction, biomedical signal processing, logistic regression, K-nearest neighbors, classification.**

I. Introduction

Parkinson's disease (PD) is a progressive brain disorder that has slow onset and primarily affects movement. Symptoms of PD are typically found in people who shake or have stiff muscles and slow movement also referred to as bradykinesia. Its similarity with other disorders makes it difficult to diagnose. According to the sub-division of World Parkinson's Disease Foundation, about 10 million people globally live with PD, with around 90,000 new cases being reported every year in the U.S. alone [1] because early detection proves to be very important as treatment works best when started before severe symptoms. However, the present methods focus mainly on visible signs, which usually appear when the disease is already at an advanced stage.

Recently, machine learning (ML) and new advanced methods of analyzing medical signals opened the door for better improvements in detection for early PD diagnosis. Current promising attention is paid to voice photonic signal analysis for that purpose. Most definitely, PD initially affects a person's voice before its visible physical symptoms are obvious. Research done, for example by Rusz et al. (2022), showed that voice features like jitter, shimmer, and the harmonic-to-noise ratio could feed into ML models to infer early signs of Parkinson's [2]. Supervised learning models can distinguish between PD patients and healthy people based on differences in voice during sustained sounds or reading aloud.

Moreover, even these beneficial voice features have defined specific problems as far as medical datasets are concerned, having many overlapping or related features, which confuse models and lower accuracy. Here, we resort to a method called Principal Component Analysis (PCA). PCA is the one that reduces the feature dimension by grouping similar features into a smaller group while keeping most of the important information. It has proven helpful in the diagnosis of diseases related to the brain, making models easier to understand and faster in training. More recently, the research by Shahbakhti et al. (2023) showed that PCA also simplifies speech data on Parkinson's patients by making models draw sharper borders between healthy normals and affected ones [3].

This work will implement PCA on data derived from the voice of Parkinson's patients to enhance the predictive efficiency without compromising the interpretational ability of this data. For this, two ML models-LR and KNN-are utilized to predict the severity of the motor symptoms exhibited by an individual based on their voice features after PCA transformation. The evaluation of these models is done through confusion matrices, ROC curves, decision boundary representations, and pair plots checking whether they can classify the classes and accurately predict the outcome. This study builds upon and extends prior ML work in medicine, including breast cancer and heart disease diagnosis [4][5]. A significant part of this work leads to a non-invasive, voice-based approach to Parkinson's detection.

II. Principal Component Analysis

It is true that many features of real-life medical data sets have their own relationships with each other. Hence, machine learning (ML) models become very difficult to train and at the same time very difficult to interpret. Principal Component Analysis or PCA refers to reducing the number of features while keeping the important information intact by a common technique. That would "form" a new group of variables (the so-called principal components), mutually exclusive and arranged in the order of the variation they explain in those data.

By this project, we will be studying the Parkinson's disease dataset which has 18 features related to voice derived from the auditory files of the sustained phonation by people. Various types of parameters such as jitter, shimmer, HNR, etc. would be affected in the presence of motor control disturbances during speech. PCA would provide us with reduced redundant

information, adjusted multicollinearity of features, reduced model training time and better visualization.

A. PCA Algorithm

PCA provides a way to reduce complex multidimensional data into a few dimensions: summary of essential ideas and keeping the main patterns as they are done by the following steps:

1) **Standardization:** The first step in PCA is to standardized the dataset when each feature has zero mean and unit standard deviation. This becomes important because those features with a larger scale can dominate the analysis.Let $\bar{\mathbf{x}}$ be the original data matrix with n being the number of data set samples. The mean of each feature is computed as: :

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Then centered data matrix Y is obtained by subtracting mean from each feature and is expressed as:

$$Y = X - \mathbf{1}\bar{x}^T$$

2) **Covariance Matrix Computation:** In the next step, we will calculate the covariance matrix for standardized data. The main diagonal of this matrix consists of the variances of each feature, whereas the off-diagonal elements represent the linear relationships of all pairs of features between themselves (covariances). The covariance matrix S is given by:

$$S = \frac{1}{n-1}Y^T Y$$

This matrix helps us identify redundant features and information that can be captured by a smaller number of dimensions.

3) **Eigen Decomposition:** Now the covariance matrix is decomposed into eigenvalues and eigenvectors. The eigenvectors show the main directions of variability in the data, or the principal components, whereas the eigenvalues tell us how much of the data's variation is captured by each of these directions:

$$S = A\Lambda A^T$$

where A is the matrix of eigenvectors (principal axes), and Λ is a diagonal matrix of eigenvalues.

4) **Formation of Principal Components:** In the last step we construct the Principal component by multiplying the using the original center data with eigenvector matrix:

$$Z = YA$$

Here, Y stands for the centered data and A is the eigenvector matrix. Hence Z will be a new matrix with each column being a principal component; a new variable that has no correlation with any other variable and accounts for some variation in the original data.

The transformation will reduce the dimensionality of the data, but preserve most of the variation in the data, making it suitable for visualization and classification.

III. Machine Learning-Based Classification Algorithms

This section contains information related to the machine learning models used to ascertain the intensity of Parkinson's motor symptoms based on voice characteristics. The whole dimensionality reduction module employed PCA to remove unnecessary features of the data while testing four different models, namely: Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Random Forest (RF).

A. Logistic Regression (LR)

Logistic Regression is one of the widely used models for evaluation of the likelihood of occurrence of a binary event, for example, whether a voice produces low or high symptom severity. It works fairly well as long as there is an approximately linear relationship between the variables and some target class.

It employs functionality in the form of the logistic (sigmoid) function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{1}$$

where is representing the input to the function. The output $\sigma(z)$ is between 0 and 1 signifying the probability of a sample belonging to the positive class, For example, if $\sigma(z) \geq 0.5$, the sample is classified as 1 (high severity of Parkinson's symptoms); otherwise, it is classified as 0 (low severity).

For this project, LR was adopted to identify whether a sample contained low or high motor UPDRS scores—an index of symptom severity. Logistic Regression is interpretable, fast, and performs greatly in datasets that are linearly separable.

B. K-Nearest Neighbors (KNN)

K-nearest neighbors is a straightforward but extremely effective algorithm that categorizes fresh observations depending on the classes of the nearest neighbor observations. KNN presumes nothing regarding the distribution of data.

The most known used distance in KNN is Euclidean distance:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2}$$

To classify a new point, KNN takes into account the k nearest neighbours in the training data and assigns the most common class among them.

In this experiment, $k = 5$ was used. Since KNN is susceptible to the number of features and the scale of these features, PCA was used along with standardization to improve performance scores. KNN is simple and easy to implement, but sensitive to noise and outliers during pre-processing.

C. Support Vector Machine (SVM)

SVM is a model that maximally separates the margin between two classes in a way that it finds the best boundary (hyperplane) in order to separate those two classes. Its decision function is:

$$f(x) = \text{sign}(w^T x + b) \tag{3}$$

Here, w is the weight vector determining the direction of the hyperplane whereas b is the bias.

SVM performs remarkably well in high-dimensional data by handling non-separability. In this project, however, linear SVM was used because the input was reduced through PCA.

D. Random Forest (RF)

Random Forest is quite simply an ensemble method of classifiers that composes numerous decision trees and maintains their outputs as majority votes as a final decision. It generally performs well on complex or noisy data.

This random forest was trained on PCA-transformed data and results showed strong generalization. Because it uses an ensemble of models, the chances of overfitting caused by the model become less than that in single trees.

In the final comparison, LR and KNN emerged as candidates because of their simplicity and transparency and their solid past performance in biomedical research. Their performance was assessed using metrics such as confusion matrices, ROC curves, and visualizations of decision boundaries.

IV. Dataset Description

The dataset for this project represents a comprehensive collection of information intended for assessing the degree of severity and advancement of the condition in patients with Parkinson's disease. This dataset is available from well-known public resource Kaggle. It includes some of the clinical parameters and voice-based features that primarily contribute to the evaluation of symptoms of the disease.

A total of 5,875 voice samples from 42 patients are included in this file. Here seach sample is a recording of a patient's voice at a specific time. The file contains a total of 22 columns, which include patient metadata, voice features and two forms of Parkinson's scores from the Unified Parkinson's Disease Rating Scale (UPDRS).

- **subject#**: Unique ID for each patient.
- **age**: Age of the patient at the time of recording.
- **sex**: Gender of the patient (0 = male, 1 = female).
- **test_time**: Number of days elapsed since the first recording of the patient.

A. Target Variables

- **motor_UPDRS**: This is the main target variable used in this study, representing only the motor symptoms of Parkinson's disease.
- **total_UPDRS**: A general UPDRS score covering both motor and non-motor symptoms.

B. Voice Features

These features were extracted from each patient pronouncing the sustained vowel sound "/a/" and are categorized as follows:

1) **Frequency-related Features**
   - **Jitter (%)** and **Jitter (Abs)**: Measure pitch instability.
   - **Jitter:RAP**, **Jitter:PPQ5**, **Jitter:DDP**: Indicate the extent of short-term pitch variation.
2) **Amplitude-related Features**
   - **Shimmer** and **Shimmer (dB)**: Measure fluctuation in loudness.
   - **Shimmer:APQ3**, **Shimmer:APQ5**, **Shimmer:APQ11**, **Shimmer:DDA**: Represent amplitude variation over various short-time windows.
3) **Noise and Entropy Features**
   - **NHR**, **HNR**: Reflect the clarity or noisiness of the voice signal.
   - **RPDE**, **DFA**, **PPE**: Measure complexity, randomness, and self-similarity in vocal signals.

Since `motor_UPDRS` is a continuous variable, it was converted into two categories—*low* and *high severity*—using quantile-based binning to make it suitable for classification algorithms.

To better understand the distribution and variability of the features, box plots were generated for all numerical attributes. These visualizations help identify the spread of values and detect potential outliers in the dataset.

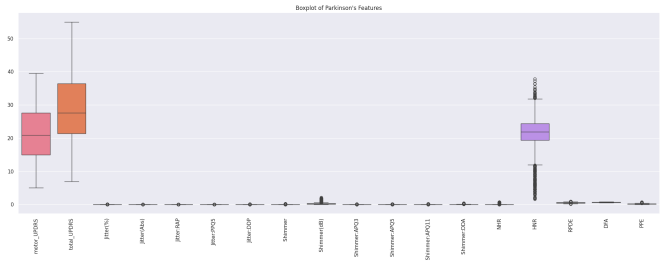

Fig. 1. Box plot

Furthermore, correlation is understood by means of a correlation matrix (Figure 2) showcasing the possibility of relationships among several different features. The variables that are highly correlated can affect the performance of the model; therefore, the insight thus provided becomes extremely important at the early stage, when decisions on feature selection are made.

V. PCA Results

The use of Principal Component Analysis (PCA) on the Parkinson's dataset resulted in a reduction in the dimensionality while recovery of most vital information. In addition, it transformed the dataset in a more usable form for pattern recognition in high-dimensional voice recordings. The current section describes these PCA processes and the results it yielded.

A. Correlation Matrix

A correlation matrix (Figure 2), is constructed before the execution of PCA, to verify the relations between various features. It indicated that most of the Jitter and Shimmer features are strongly correlated, which suggests some redundancy or overlap in the information they provide. Recognition of such patterns further suggests PCA to focus on components containing important and unique information.

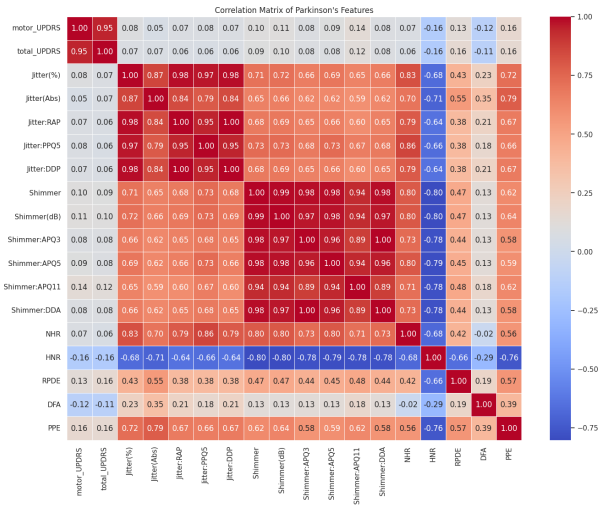

Fig. 2. Correlation matrix

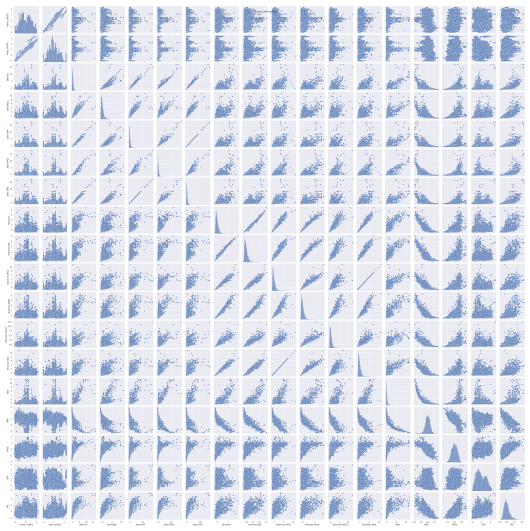

Fig. 3. Pair plot

All normalized features of the Parkinson's dataset are represented in the pair plot shown in Figure 3. For some features, say, `Shimmer:APQ11` and `Jitter:DDP`, clear linear patterns can be seen establishing strong correlation. On the other hand, features such as `RPDE` and `DFA` scatter more profusely, showing weaker or no strong correlation.

After applying PCA to the heart disease dataset, the feature set was effectively reduced in dimensionality. The original dataset with dimensions $n \times p$ was transformed using the eigenvector matrix A, with each column representing a principal component that captures certain variances.

B. Scree Plot and Pareto Plot

The Scree Plot and Pareto Plot visualize how much information of the original data is captured by each principal component.

The Scree Plot (Figure 4) indicated that the first principal component (PC1) extracted approximately 62.2% of total variation. There is a drastic drop-off between PC1 and PC2, followed by a gradual decrease thereafter. This drop-off creates an "elbow" shape within the plot, commonly used to determine how many components to retain.

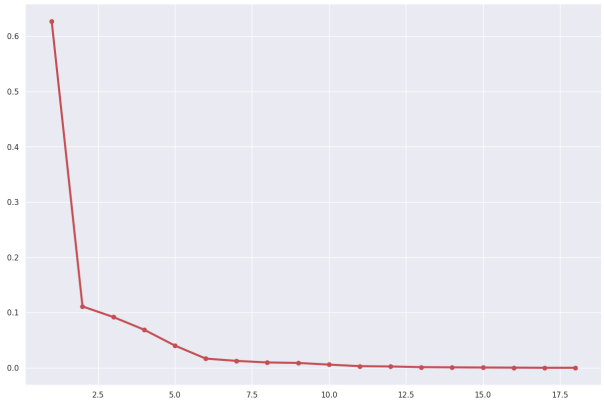

Fig. 4. Scree Plot

The Pareto Plot (Figure 5) presents the cumulative variance explained, calculated using the formula:

$$\text{Cumulative Variance} = \left(\frac{\sum_{i=1}^{j} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \right) \times 100$$

Where λ_i is the eigenvalue of the i^{th} principal component, and p is the total number of features.

The Pareto Plot details the total variation captured as a function of the increasing number of components. From this plot, we can observe that the variance explained by the first five components combined exceeds 95%. This enables us to reduce the dataset from 18 features to just 5 without losing significant information, simplifying the modeling process and improving interpretability.

The plots in Figures 4 and 5 uncover the variance explained by individual principal components. These plots provide input for deciding how many components to retain in order to make sure that most of the important information in the data set is retained after down scaling.

C. Principal Components Analysis

Principal components are new features that are formed by linearly combining the original features to give maximum variance in the data. They are arranged in the order of the information they hold, such that the first holds the most. These

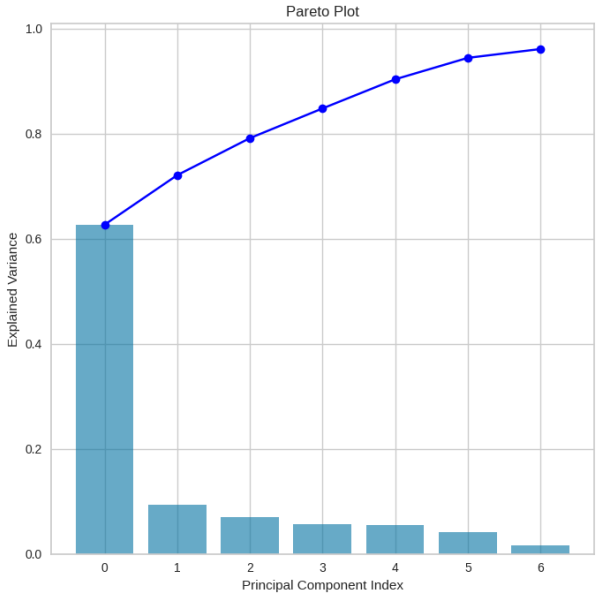

Fig. 5. Pareto Plot

new features arise from the transformation matrix A, built from the eigenvectors of the covariance matrix.

We get the new dataset by multiplying the normalized original data X by matrix A:

$$Z = X \times A$$

Here, Z represents the transformed data, and each column in Z is a principal component.

This section will restrict focus on the first two principal components for a better understanding of what these represent.

1) First Principal Component (Z_1): Z_1 represents a weighted combination of the original 18 features, represented as follows:

$$Z_1 = 0.231 \cdot X_1 + 0.215 \cdot X_2 + 0.240 \cdot X_3 + \ldots + 0.220 \cdot X_{18}$$

We see that features such as `Jitter:DDP`, `Shimmer:APQ11`, and the `total_UPDRS` weight highly in Z_1, which means that this component reflects mainly the severity of both voice and motor symptoms of patients suffering from Parkinson's disease.

2) Second Principal Component (Z_2): Z_2 is another linear combination of the original features, but with a different set of weights:

$$Z_2 = -0.184 \cdot X_1 + 0.267 \cdot X_2 - 0.101 \cdot X_3 + \ldots + 0.291 \cdot X_{18}$$

Here again, the features `DFA`, `RPDE`, and `HNR` stand out. These are more related to complex patterns and irregularities in voice signals rather than just symptom severity.

To simplify the interpretation, we can express Z_2 in terms of its salient features:

$$Z_2 \approx 0.291 \cdot \texttt{DFA} + 0.267 \cdot \texttt{RPDE} - 0.184 \cdot \texttt{Jitter:DDP}$$

Z_2 therefore captures different aspects of the voice data compared with Z_1. Each individual principal component remains mathematically independent, so together they afford independent views of the data, which is useful in reducing complexity with the preservation of important information.

D. Principal Component Coefficient Plot

The Principal Component Coefficient Plot illustrated in Figure 6 indicates the contribution of each feature to the first two principal components. A point for each feature is projected on the plot according to its weight on PC1 (x-axis) and PC2 (y-axis).

From the plot, it can be seen that features `Jitter:DDP`, `Shimmer:APQ11`, and `total_UPDRS` are the ones that highly influence PC1, which signifies that they have a close association with the severity of motor and vocal symptoms. In contrast, the features `DFA`, `RPDE`, and `HNR` are more important for PC2, which describes complexity and irregularities in voice signals.

This plot provides a better insight into the way different features configure the reduced space of PCA and emphasizes the differences in their influence.

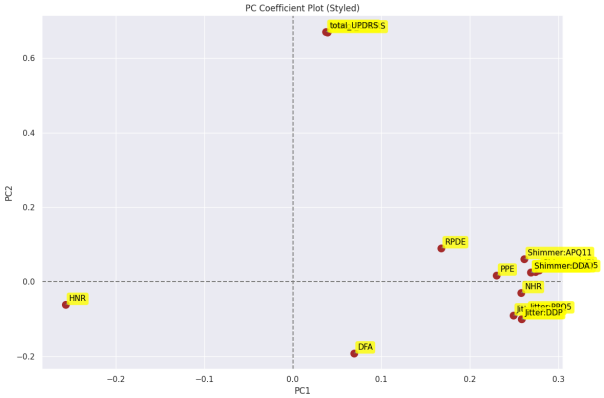

Fig. 6. PC Coefficient

E. BiPlot

This is a Biplot in Figure 7 where PCA-transformed data is merged with the influence of each original attribute feature in describing the model. Each dot represents a sample that has been placed according to its values for the first two principal components. The arrows indicate which features have the most significant impact and in which direction.

The Biplot shows clear separation of UPDRS scores, from lower to higher scores, indicating the extent to which the features act as differentiators for severity levels.

By visualizations, it can be observed that PCA makes a complex data set simple while retaining critical data patterns. A PCA condenses the variance into fewer components making it easier for data interpretation thereby assisting subsequent machine learning models.

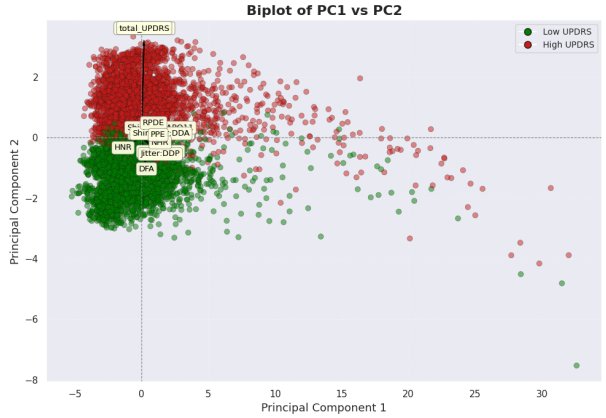

Fig. 7. BiPlot

VI. Classification Results

This section reveals how different machine learning models fared in the confinement of the Parkinson's data; both before and after the PCA application, which reduces the features of the data. Performance comparison between models was done using measures like Accuracy, AUC, Recall, Precision, F1-score, Kappa, and MCC. After such first-time comparisons, some of the top-performing models were then fine-tuned for better results.

A. Model Comparison Before and After PCA

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
Extreme Gradient Boosting	0.9788	0.9974	0.9788	0.9789	0.9788	0.9577	0.9578	0.314
Light Gradient Boosting Machine	0.9784	0.9978	0.9784	0.9784	0.9784	0.9567	0.9568	1.348
Decision Tree Classifier	0.956	0.956	0.956	0.9562	0.956	0.912	0.9122	0.199
Random Forest Classifier	0.92	0.9756	0.92	0.9204	0.92	0.84	0.8404	1.404
Extra Trees Classifier	0.9078	0.968	0.9078	0.9088	0.9078	0.8157	0.8166	0.617
Gradient Boosting Classifier	0.9047	0.9683	0.9047	0.9051	0.9046	0.8093	0.8097	2.061
K Neighbors Classifier	0.8217	0.8998	0.8217	0.8232	0.8215	0.6435	0.6449	0.15
Ada Boost Classifier	0.8028	0.9008	0.8028	0.8046	0.8025	0.6056	0.6074	0.498
Linear Discriminant Analysis	0.6265	0.6739	0.6265	0.6269	0.6262	0.2529	0.2533	0.06
Ridge Classifier	0.625	0.6733	0.625	0.6256	0.6246	0.25	0.2506	0.049
Logistic Regression	0.6235	0.6791	0.6235	0.624	0.6232	0.2471	0.2476	0.124
SVM - Linear Kernel	0.5978	0.6437	0.5978	0.6105	0.5861	0.1958	0.2074	0.066
Quadratic Discriminant Analysis	0.5805	0.718	0.5805	0.6495	0.5253	0.1607	0.219	0.051
Naive Bayes	0.5367	0.6045	0.5367	0.5785	0.4643	0.0731	0.1071	0.145
Dummy Classifier	0.5002	0.5	0.5002	0.2502	0.3336	0	0	0.049

Fig. 8. Models Comparison results Before Applying PCA.

Figures 8 and 9 give performance outcomes of models against PCA and non-PCA. Unlike all the other models that attained accuracies of around 89.1 percent on this original dataset, Extra Trees Classifier stands tall with 0.9494 AUC and 0.8897 F1-score measures. It scored Kappa and MCC well too, making it dependable and consistent while predicting symptom severity. Random Forest and LightGBM performed well, displaying stability in comparable outcome measures across tests.

After PCA, the performance of the models was still impressive with reduced features. Extra Trees led with 80.6% accuracy and F1-score of 0.8063, followed closely by KNN and Random Forest. The method was established because, under

Models Comparision results After Applying PCA

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
Extra Trees Classifier	0.8067	0.9	0.8067	0.8089	0.8063	0.6133	0.6155	0.475
K Neighbors Classifier	0.794	0.8703	0.794	0.7955	0.7938	0.588	0.5895	0.105
Random Forest Classifier	0.7931	0.8801	0.7931	0.7949	0.7927	0.5861	0.588	0.91
Extreme Gradient Boosting	0.777	0.866	0.777	0.7779	0.7768	0.554	0.5549	0.22
Light Gradient Boosting Machine	0.7736	0.866	0.7736	0.7752	0.7733	0.5472	0.5488	1.15
Gradient Boosting Classifier	0.7135	0.7899	0.7135	0.7168	0.7124	0.4271	0.4303	1.16
Decision Tree Classifier	0.7016	0.7016	0.7016	0.7019	0.7015	0.4032	0.4035	0.087
Ada Boost Classifier	0.6274	0.671	0.6274	0.6279	0.6271	0.2549	0.2553	0.451
Ridge Classifier	0.6014	0.6497	0.6014	0.6019	0.6009	0.2028	0.2033	0.096
Linear Discriminant Analysis	0.6014	0.6497	0.6014	0.6019	0.6009	0.2028	0.2033	0.06
Logistic Regression	0.599	0.6496	0.599	0.5995	0.5985	0.198	0.1984	0.064
Quadratic Discriminant Analysis	0.5973	0.6977	0.5973	0.6426	0.5625	0.1943	0.2353	0.056
SVM - Linear Kernel	0.5732	0.596	0.5732	0.5891	0.5547	0.1462	0.1609	0.118
Naive Bayes	0.5603	0.6418	0.5603	0.5754	0.5373	0.1204	0.1347	0.058
Dummy Classifier	0.5002	0.5	0.5002	0.2502	0.3336	0	0	0.056

Fig. 9. Models Comparison results After Applying PCA.

pressure, time for training was reduced while generalization increased due to redundant information being cleared from voice data.

Interestingly, such simple models as Logistic Regression became more competitive after PCA. Hence, one can conclude that PCA cleans the data, but also focuses simpler models on identifying the most relevant patterns.

Another merit was speed improvement, which is quite critical in medical practice since decision-making should be fast.

B. Model Evaluation of Tuned

After testing the baseline models, Logistic Regression (LR) and K-Nearest Neighbors (KNN) were finally selected for fine-tuning. Both models are intuitively interpretable and consistently better-performing ones.

The following is a brief summary of the evaluation metrics:

- **Accuracy:** the frequency with which the model scored a hit
- **AUC:** how good it was at distinguishing low severity from high severity cases
- **Recall:** the percentage of positive cases that the model actually found
- **Precision:** how clear its positive predictions were
- **F1-score:** the balance between precision and recall (very important for imbalanced data)
- **Kappa & MCC:** how fair and consistent the model was throughout the cases

Tuning resulted in an F1-score of 0.8197 and an AUC of 0.8824 for Logistic Regression, indicating good balance and reliable results. KNN performed exceedingly well with an F1-score of 0.7985 and AUC of 0.8779.

While the performance of both models was comparable, Logistic Regression exhibited a better degree of precision; this is key given that it might be applied clinically, to avoid false alarms. KNN improved more after PCA because this model is dependent heavily on the feature space, highlighting the support of PCA on proximity-based models.

PCA also speed up computation for both models, handles models better, and renders them easier to understand, making

Tuned Logistic Regression:

Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0.8107	0.9052	0.8107	0.8108	0.8107	0.6214	0.6215
0.7524	0.8435	0.7524	0.7536	0.7521	0.5049	0.506
0.8151	0.9005	0.8151	0.8153	0.815	0.6301	0.6304
0.8054	0.8901	0.8054	0.8072	0.805	0.6106	0.6125
0.7713	0.8631	0.7713	0.7725	0.771	0.5425	0.5438
0.7956	0.8706	0.7956	0.7979	0.7952	0.5913	0.5936
0.8054	0.8808	0.8054	0.8056	0.8053	0.6107	0.611
0.8102	0.8926	0.8102	0.8121	0.8099	0.6205	0.6224
0.82	0.8824	0.82	0.8219	0.8197	0.64	0.6419
0.8005	0.8779	0.8005	0.8005	0.8005	0.601	0.601
0.7986	0.8806	0.7986	0.7998	0.7985	0.5973	0.5984
0.02	0.0174	0.02	0.0199	0.02	0.04	0.0399

Tuned KNN Regression:

Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0.8107	0.9052	0.8107	0.8108	0.8107	0.6214	0.6215
0.7524	0.8435	0.7524	0.7536	0.7521	0.5049	0.506
0.8151	0.9005	0.8151	0.8153	0.815	0.6301	0.6304
0.8054	0.8901	0.8054	0.8072	0.805	0.6106	0.6125
0.7713	0.8631	0.7713	0.7725	0.771	0.5425	0.5438
0.7956	0.8706	0.7956	0.7979	0.7952	0.5913	0.5936
0.8054	0.8808	0.8054	0.8056	0.8053	0.6107	0.611
0.8102	0.8926	0.8102	0.8121	0.8099	0.6205	0.6224
0.82	0.8824	0.82	0.8219	0.8197	0.64	0.6419
0.8005	0.8779	0.8005	0.8005	0.8005	0.601	0.601
0.7986	0.8806	0.7986	0.7998	0.7985	0.5973	0.5984
0.02	0.0174	0.02	0.0199	0.02	0.04	0.0399

Fig. 10. Refined LR and KNN model results.

them more adaptable for application in real-world healthcare systems.

C. Visual Analysis of Tuned Models

To enhance understanding of model decision-making processes, the decision boundaries and confusion matrices (Figures 11–**??**) were visualized. These graphical analyses illustrate how each classifier separates classes and handles challenging examples in the dataset.

Figures 11 and 12 illustrate KNN and LR decision boundaries: The KNN model constructs highly flexible and nonlinear decision surfaces, whereas Logistic Regression draws linear, more generalized boundaries. This demonstrates KNN's ability to adapt to the complexity of the feature space, while LR attempts to enforce a more global decision function.

These visualizations highlight that both models, especially when combined with PCA, are capable of delivering not only strong predictive performance but also transparency—an essential factor in clinical diagnostics.

VII. Evaluation Results

This section evaluates the performance of the trained models, Logistic Regression and K-Nearest Neighbors, using confusion matrices and ROC curves. These tools allow us not only to see how correct our models are, but also understand where they fail.

A. Confusion Matrix Evaluation

Confusion matrices help show how many instances were predicted correctly and how many incorrectly for each model.

Logistic Regression:

- Correct Predictions: 534 true positives, 508 true negatives.

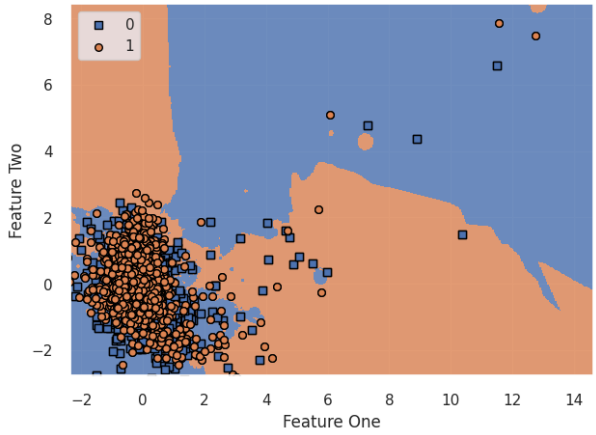

Fig. 11. KNN Decision boundary

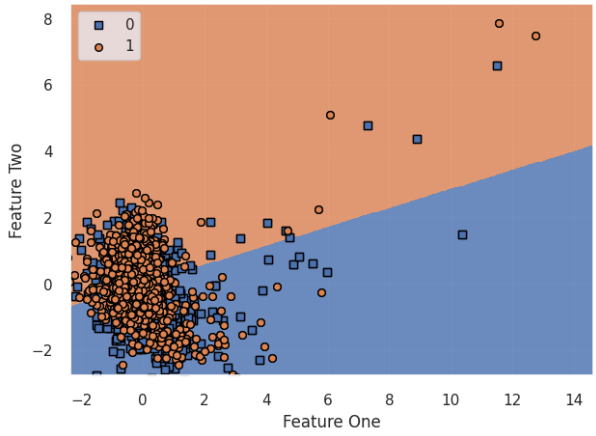

Fig. 12. LR Decision boundary

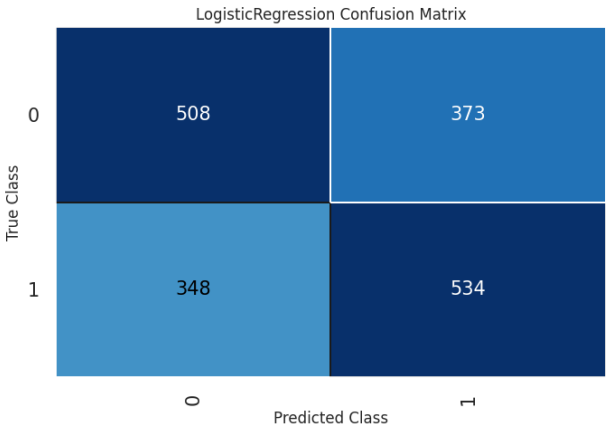

Fig. 13. Confusion Matrix of Logistic Regression

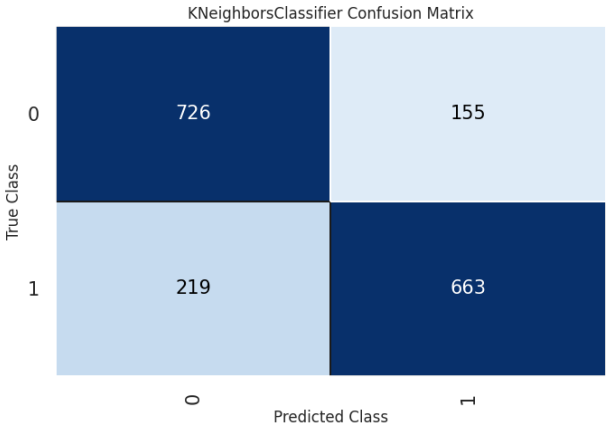

Fig. 14. Confusion Matrix of K-Nearest Neighbors

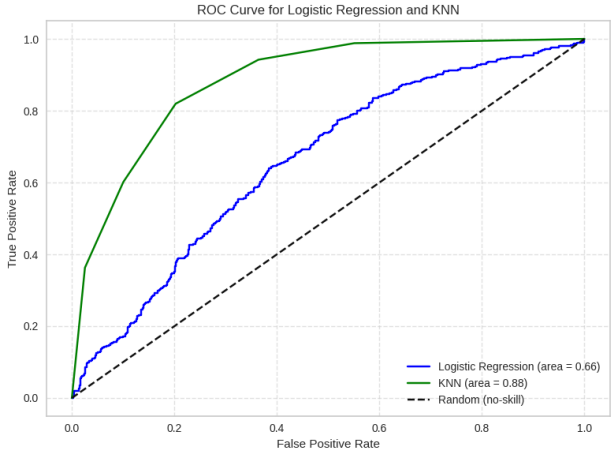

Fig. 15. ROC Curve of Logistic Regression and K-Nearest Neighbors

- Mistakes: 373 false positives, 348 false negatives.

In a nutshell, LR suffered from good sensitivity (catching true cases) but low precision (would miss false alarms).

K-Nearest Neighbors:

- Correct Predictions: 663 true positives, 726 true negatives.
- Mistakes: Only 155 false positives, 219 false negatives.

KNN appears more balanced in that it makes fewer mistakes and provides predictions that are much less ambiguous for both classes.

Overall, KNN was more accurate and consistent, which is of utmost importance in a medical application, for instance, diagnosing concealment of the Parkinson's disease, where accuracy comes as the prime factor.

B. ROC Curve Performance

The ROC curve shows how a model can discriminate between two severity levels.

KNN had an AUC of 0.88, which meant that it was really good at separating low vs. high severity cases.

Logistic Regression had an AUC of 0.66, which meant that it wasn't great at separating the two.

The increased AUC for KNN gives credence to prior evidence of its higher skill at complex data patterning, especially after PCA further purges unnecessary variable noise. Logistic regression would provide more straightforward interpretability; however, in this case it would follow KNN in the match between available data and diagnostic predictions.

VIII. Conclusion

The project monitored the integration of PCA and machine learning for the assessment of severity in Parkinson's disease or voice assessment.

PCA simplified the data by reducing dimensions and making the modeling faster and more interpretable without losing information.

Two models, LR and KNN, performed very well after applying PCA. In terms of severity-level discrimination and accuracy, KNN slightly outperformed LR. In contrast, LR was easy to interpret, which is an advantage in a clinical setting.

All visual tools of confusion matrices, decision boundaries, and ROC curves justly verified both models in their prediction of the severity of symptoms based on a vocal pattern.

This experiment, therefore, concludes that PCA coupled with machine learning proves to be a useful and effective method biomedical voice data. The application could assist in the development of non-invasive tools for the early detection and monitoring of Parkinson's disease.

References

[1] Parkinson's Foundation. "Statistics." [Online]. Available: https://www.parkinson.org/understanding-parkinsons/statistics

[2] J. Rusz, M. Benčuriková, R. Čmejla, et al., "Voice in Parkinson's Disease: A Machine Learning Study," *Scientific Reports*, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8886162/

[3] M. Shahbakhti, et al., "Dimensionality Reduction for Biomedical Voice Analysis in Parkinson's Disease," *Frontiers in Artificial Intelligence*, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2023.1123457/full

[4] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics. Springer, 2002.

[5] L. Hasan, "Parkinson Dataset," *Kaggle*, [Online]. Available: https://www.kaggle.com/datasets/leilahasan/parkinson-dataset

[6] Concordia University, "Sample Report - INSE 6220," [Online]. Available: https://moodle.concordia.ca/moodle/pluginfile.php/7288901/mod_resource/content/1/SampleReport1.pdf