Classification of Parkinson's Disease Using PCA and Supervised Learning

Manan Mishra
Student ID: 40321141
INSE 6220 Advanced Statistical Approaches to Quality
https://github.com/MananMishra-7/INSE6220

***Abstract*—Parkinson's disease is progressive neurological disease with no definitive cure which affects motor and non motor functions in a human being. This study focuses on the implementation of Principal Component Analysis(PCA) and supervised Machine Learning(ML) techniques over voice data of individuals to aid the early detection of Parkinson's disease. PCA is applied to reduce the number features in the data set and capture only those features which explains the most variance. This technique is highly relevant in today's world as the dimensionality of data increases so does the required computational power and time taken to make a prediction. Also it is easy to visualize and explain results with less dimensions. Later 3 classification ML algorithms, K-Nearest Neighbor, Decision Tree and Logistic Regression are evaluated on the original data set and reduced dimension data set. Here each combinations is then evaluated by Accuracy, Precision, Recall and Area Under Curve(AUC) scores. Based on the comparative analysis K-Nearest Neighbor with reduced dimensions data set outperforms the classification task with highest accuracy and AUC score.**

***Index Terms*—Principal Component Analysis, Machine Learning, Accuracy, Precision, Recall and Area Under Curve, Supervised Learning.**

I. Introduction

Parkinson's disease(PD) is a chronic neurological disorder that causes impaired movement, coordination, and balance in humans. It is considered to be the second most common neurodegenerative disorder after Alzheimer's and is generally diagnosed in people over the age of 50 years. It is more frequently diagnosed in men as compared to women [1]. It significantly disrupts the quality of life of the patient emotionally as well as economically.There is no definitive cure to this as the reason why it is caused is still unclear to the researchers. Medical professionals have tried to diagnose Parkinson's in early stages but very often the symptoms overlap with other neurological disorders. Few studies have shown that the way a person speaks can be a way to identify PD patients.Singh et al. [2] proposed a telemedicine approach to detect PD patients by collecting voices from smartphone data. Features like change in frequency, amplitude and pitch can be captured and analyzed for prediction. Implementation of Supervised Machine Learning techniques can help to make a informed logical data driven decision in diagnosing PD quickly and accurately.

II. Tools Utilized

To perform this study on the Parkinson's dataset, a base jupyter notebook file was provided by Professor A. Ben Hamza [7]. The study was conducted using the Python programming language, and few libraries that were utilized are as follows:

- PyCaret
- Pandas
- NumPy
- Matplotlib
- Scikit-learn
- Seaborn

III. Data Description

The data set used in this study has been taken from UCI Machine Learning Repository [3]. It is a widely used data set on platforms like Kaggle to perform classification algorithms. There are 24 features in the data set with 195 entries in it. As shown in Figure 1, 22 features are floating type integers where as "name" feature is in object type and "status" in int. The "status" feature is our target variable which takes two values 0 and 1. 0 shows that the person is healthy and 1 shows that the person is diagnosed with PD. There are no duplicate values and no null values in the data set.

A. Feature Selection

Selecting key features is a crucial task before implementing Principal Component Analysis(PCA) and Machine Learning algorithms. It reduces redundancy from the data set and becomes computationally less expensive. The features selected are on the basis of explanation of features from Kaggle's Parkinson's Disease description [4].It has grouped together multiple features which are informing about the same things like shimmer, jitter and noise. For PCA only the following features will be considered:

- MDVP:Fo(Hz)
- MDVP:Jitter(Abs)
- MDVP:Shimmer
- status
- HNR
- RPDE
- spread1
- PPE

16 features which are removed were highly correlated which would have resulted into redundant data.To obtain the correlation values a heatmap was generated for 23 features. Removal of highly correlated feature is necessary as features with high

```
RangeIndex: 195 entries, 0 to 194
Data columns (total 24 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   name              195 non-null    object
 1   MDVP:Fo(Hz)       195 non-null    float64
 2   MDVP:Fhi(Hz)      195 non-null    float64
 3   MDVP:Flo(Hz)      195 non-null    float64
 4   MDVP:Jitter(%)    195 non-null    float64
 5   MDVP:Jitter(Abs)  195 non-null    float64
 6   MDVP:RAP          195 non-null    float64
 7   MDVP:PPQ          195 non-null    float64
 8   Jitter:DDP        195 non-null    float64
 9   MDVP:Shimmer      195 non-null    float64
 10  MDVP:Shimmer(dB)  195 non-null    float64
 11  Shimmer:APQ3      195 non-null    float64
 12  Shimmer:APQ5      195 non-null    float64
 13  MDVP:APQ          195 non-null    float64
 14  Shimmer:DDA       195 non-null    float64
 15  NHR               195 non-null    float64
 16  HNR               195 non-null    float64
 17  status            195 non-null    int64
 18  RPDE              195 non-null    float64
 19  DFA               195 non-null    float64
 20  spread1           195 non-null    float64
 21  spread2           195 non-null    float64
 22  D2                195 non-null    float64
 23  PPE               195 non-null    float64
dtypes: float64(22), int64(1), object(1)
memory usage: 36.7+ KB
```

Fig. 1. Data Overview

correlation conveys the same information. Feature ”name” was removed at the start only as it was an object type and had no contribution to the target variable.

B. Exploratory Data Analysis

Exploratory Data Analysis(EDA) helps us to identify the central measures of the data set that can help to identify outliers and assist in data cleaning or explaining predicted results. Figure 2 shows the frequency of 2 instances in the target variable ”status”.Where 48 instances are Healthy and 147 are identified as PD. Figure 3 displays the box plot of other 7 features highlighting the median values of each as well as showcasing outliers. The outlier identification can de done using the following formula's [6]

$$\text{IQR} = Q_3 - Q_1$$
$$\text{LIF (Lower Inner Fence)} = Q_1 - 1.5 \times \text{IQR}$$
$$\text{UIF (Upper Inner Fence)} = Q_3 + 1.5 \times \text{IQR}$$

As this is a medical data set the outlier values can have significance and hence won't be removed or changed. Figure 4 is heatmap generated for 7 independent variables other than the target variable. The values can be calculated using Pearson Corelation. Values closer to 1 show high level corelation where as values closer to -1 shows negative relation among each other. Value of 0 signifies there is no relationship between the features [5].It is important to note that values shown in Figure 3 and Figure 4 are after standardizing the selected data set.

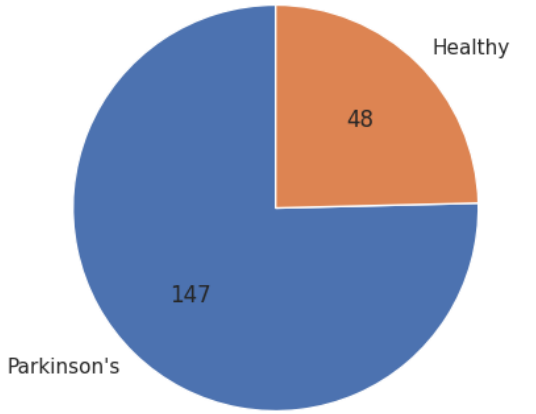

Fig. 2. Target Variable

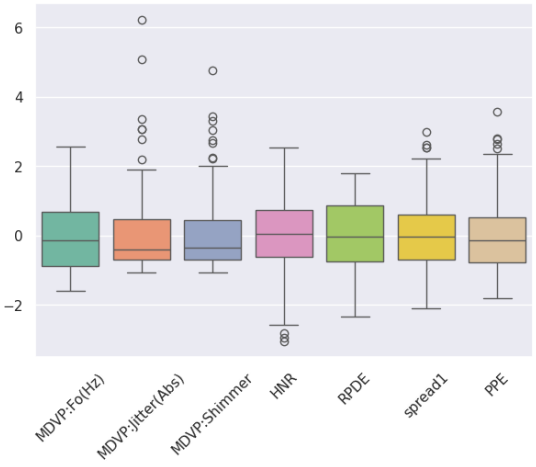

Fig. 3. Feature Box Plot

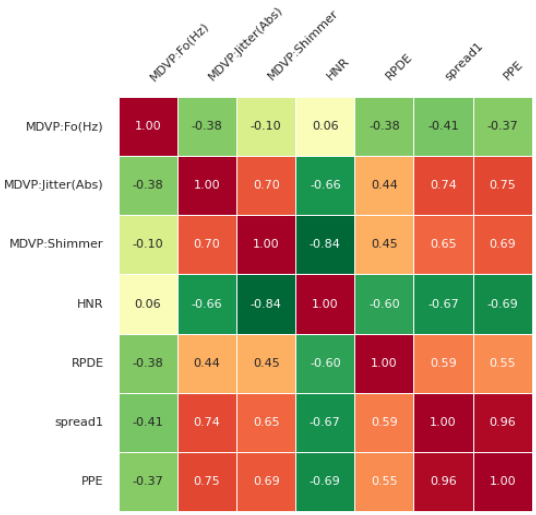

Fig. 4. Heatmap

IV. Principal Component Analysis

PCA is an unsupervised multivariate data reduction technique based on linear correlation [6]. It is unsupervised as the target variable goes under no transformation. PCA is implemented because the original dataset is highly correlated even after removing multiple features from the data set shown in Figure 4. PCA generates a set of new variables and transforms our data set so that it becomes completely uncorrelated. Here the new variables are calculated in such a way that they can explain the variation (information) accounted in the original data set. Hence after PCA we get an reduced dimensions, reduced parameters and reduced complexity.The whole process of PCA can be broken down into following steps:

- **Dropping the target variable "status":** as it is used as the label for prediction and PCA is an unsupervised technique.
- **Centering of Data:** mean of each column is been subtracted with the respective values that column. Mathematically it can be explained as follows:

$$Y_{ij} = X_{ij} - \mu_j$$

where:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}$$

Thus, matrix Y has the same dimensions as X (i.e., $n \times p$
- **Computing Covariance matrix:** S, which will have dimensions of $p \times p$.
- **Eigen Decomposition:** Since S is a square matrix, we can directly perform eigenvalue decomposition instead of singular value decomposition. After this step, we can compute the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and the eigen matrix A, with dimensions of $p \times p$.
- **Principal Components:** A final matrix Z, ideally with dimensions $n \times p$, can be calculated as $Z = YA$, where A is chosen as $n \times r$, with r representing the reduced number of features. The number of principal components will therefore be reduced, resulting in Z having dimensions of $n \times r$ hence reducing the number of variables. The first column of Z explains maximum variability.

A. PCA Results

To interpret PCA results it's important to select a value of r. To choose the value of r Scree plot also known as elbow plot is an good illustration to do so. As shown in Figure 5 when number of components are 2 we can see a sharp change hence making the elbow shape. Via Figure 5 we can also intuitively say that the cumulative explained variance is around 80% but for finding the exact value we can see the Pareto graph in Figure 6. The explained variance l_i corresponding to the i-th principal component is given by:

$$l_i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}$$

where λ_i is the eigenvalue associated with the i-th principal component, and n is the total number of components. Based on

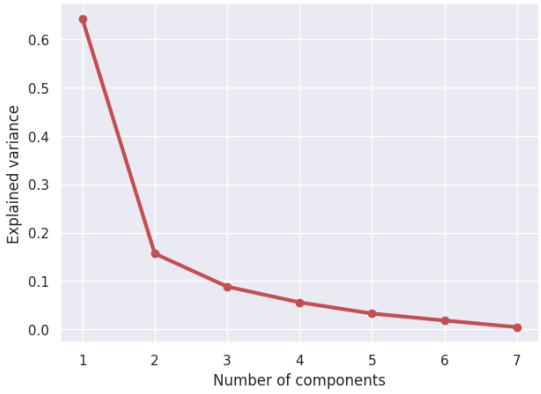

Fig. 5. Scree Plot

Figure 6 we can calculate that the explained variance is 79.8% where PC1 accounts for 64.2% of variability and PC2 accounts for 15.6%. Hence 2 Principal components can account for 79.8% of explained variance

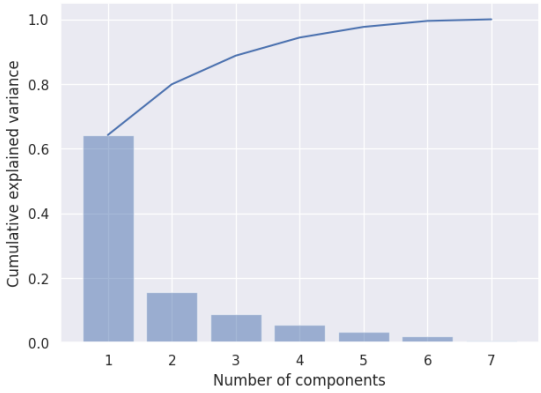

Fig. 6. Pareto Analysis

Hence based on this r=2 which means we are going to use 2 principal components Z_1 and Z_2.The Two principal components can be shown by Figure 7. The reduced dimensions are 195 X 2 instead of 195 X 7. Also to visualize the PCA results in more detail we can also use Biplot shown in Figure 8 which helps us to identify features which contribute the most in PC1 and PC2 respectively. The title shows the cumulative value of first three principal components which results into 88.75% as the counter in python starts from 0. In our case we have identified PC0 as PC1 (First Principal Component) and PC1 as PC2 (Second Principal Component) as shown in Figure 7.

In Figure 8 X axis is PC1 and Y axis is PC2 where as blue dots are the sample points . Each arrow length shows the strength of the particular feature like PPE, spread1, and MDVP:Jitter(Abs) have strong influence. The angle between the feature arrows and axis shows how comparable their variance is, like MDVP:Fo(Hz) has a narrow angle with y

	PC1	PC2
0	1.290182	-0.313418
1	2.777572	0.049517
2	2.178716	-0.273282
3	2.555366	-0.280846
4	3.365263	-0.058580
...	...	...
190	-0.666391	1.035064
191	-0.822074	1.483592
192	-0.549974	0.910047
193	-1.115878	1.227327
194	-1.022321	1.140119

195 rows × 2 columns

Fig. 7. PC1 and PC2

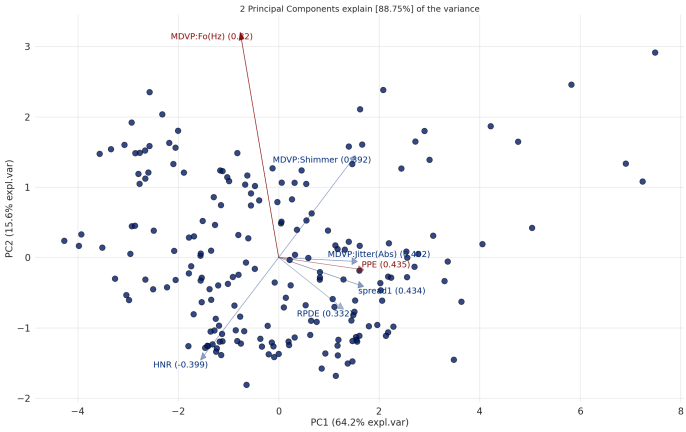

Fig. 8. Biplot

axis than the x axis signifying it has strong corelation with PC2.

V. Modeling Machine Learning Algorithms

After Completing EDA and PCA we can proceed to implement machine learning (ML) techniques on our data set. Here supervised ML algorithms will be used in two separate scenarios.

- **Without PCA:**Original data set with 195 entries and 23 features will be utilized to train and test the model.
- **With PCA:** Transformed dataset with 195 entries and 2 principal components with a cumulative explained variance of 79.8% will be used for training and testing.

A. Train and Test split

The dimensions for the original dataset are (195,23) (”name” feature is not considered in our analysis).Out of this only 90% of the dataset will be used to train and test the ML models but the remaining 10% where never given to make any predictions. Hence Data used for modeling will have the dimensions of (176, 23) and unseen data for inference will have the dimensions of (19, 23).

- **Without PCA:**Original data set considered will have 176 entries and 23 features. Out of this ML models will be trained on 70% of the data which will have dimensions of 123,23 and remaining 53,23 will be used for testing.
- **With PCA:** Transformed data set will be used which has the dimensions of 176,3. Train and test split is similar as before.

B. Model Selection

The following 3 models are used:

- **K Nearest Neighbor(KNN):** A popular robust classification algorithm based in similarity [8]. KNN finds K closest samples using similarity indices like Euclidean, Manhattan or cosine etc. One important hyper parameter is the value of K, if it is too low the results will have high variance.If K is too high the value it will result into a oversimplified model with high bias. Generally in case of binary classification value of k is kept in odd numbers.
- **Decision Tree(DT)** One of the most intuitive models which is used for classification as well as regression tasks. It learns rules from data and splits in to regions using if-else conditions where leaf nodes represent the classes [9].
- **Logistic Regression(LR):**Despite it’s name it is a classification model and not a regression. It works on the basis of probability that an input belongs to which class [**?**]. In case of binary classification if the probability of the input is greater than 0.5 it is classified to the higher value class.

C. Hyperparameter Tuning

Hyperparameter tuning is a important process to find optimal values for the parameters before training like value of K in KNN or the max_depth in DT. It helps in improving model performance like accuracy, F1-score. It is important to note that Hyperparameter are different from parameters as they are set before training where as parameters learned during training. In our analysis we have used PyCaret’s in built library tune_model() for this. Each of our models undergoes this process, and best versions are used for comparisons. Hyperparameter Tuning is done by K fold cross validation. In this approach data set is split into K parts and then train and tests each model on each part and evaluates performance as an average. By this we make sure that no bias is there.

VI. Evaluation

- **Confusion Matrix** This matrix is used to describe the performance of a classification model [12]. Based on this matrix multiple scores can be computed which on later stage can be compared.

Actual/Predicted	**Positive**	**Negative**
Positive	True Positive	False Negative
Negative	False Positive	True Negative

- True Positive(TP): Model has correctly predicted the positive class.
- False Positive(FP): Model has incorrectly predicted the positive whereas it is negative.
- True Negative(TN):Model correctly predicts negative class.
- False Negative(FN): Model incorrectly predicted negative when it is actually positive.

Based on this we can find Accuracy Precision and Recall each of which formula can be written as follows:

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (Sensitivity):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Based on this formulas the Table I can be generated which shows the Accuracy Precision and Recall using the original dataset and transformed dataset after PCA. It is important to note that the values are selected after tuning each model and not every model showed better performance after hyperparameter tuning hence the best values are shown.

TABLE I
EVALUATION METRICS BEFORE AND AFTER PCA

Model	PCA	Accuracy	Precision	Recall
K-Nearest Neighbors	No	0.86	0.89	0.94
K-Nearest Neighbors	Yes	0.90	0.93	0.94
Decision Tree	No	0.87	0.92	0.91
Decision Tree	Yes	0.88	0.93	0.91
Logistic Regression	No	0.88	0.91	0.93
Logistic Regression	Yes	0.85	0.88	0.94

- **Receiver Operating Characteristic (ROC) Curve:** ROC curve establishes a relationship between true positive rate and false positive rate. Ideally true positive rate should be =1, but it generally does not happen. The closer the curve is to the top left corner the better the performance of the model. The following graphs show the ROC curve of each machine learning model before and after PCA it also shows the Area Under Curve (AUC) score which signifies the ability of the model to predict the classes. If AUC=1 the model is perfect but if AUC = 0.5 the model is not able to discriminate between classes [11].

VII. EXPLAINABLE AI WITH SHAPLEY VALUES

Unlike the traditional black box approach of the machine learning and deep learning models Explainable AI tries to explain the model in understandable and interpretable ways to humans. The concept of Shapely values comes from game theory concepts where the aim is to fairly

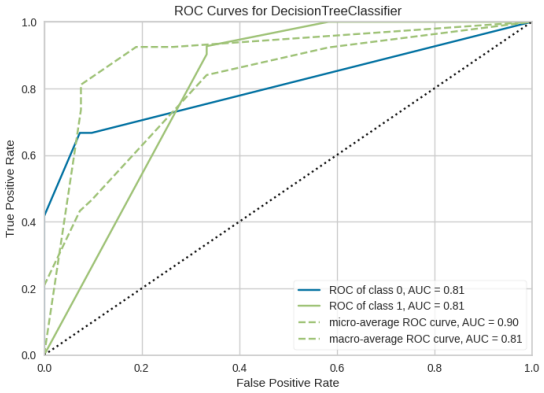

Fig. 9. ROC of DT

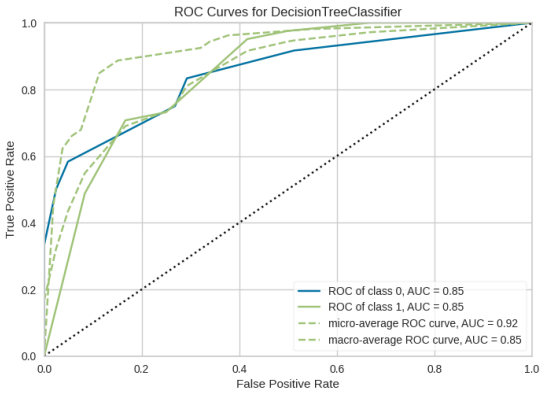

Fig. 10. ROC of PCA + DT

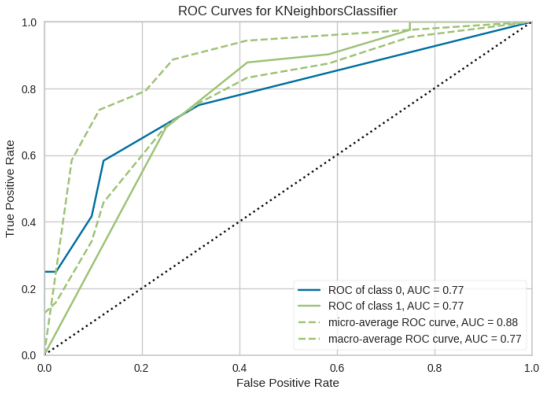

Fig. 11. ROC of KNN

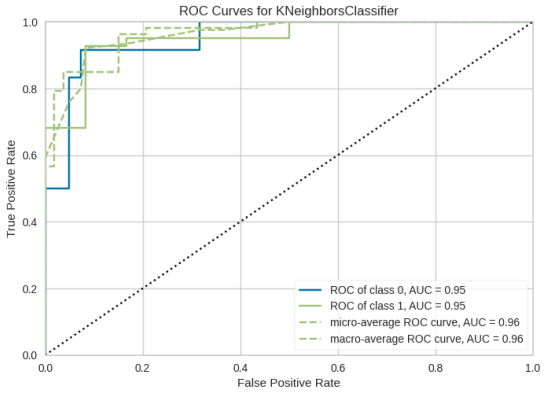

Fig. 12. ROC of PCA + KNN

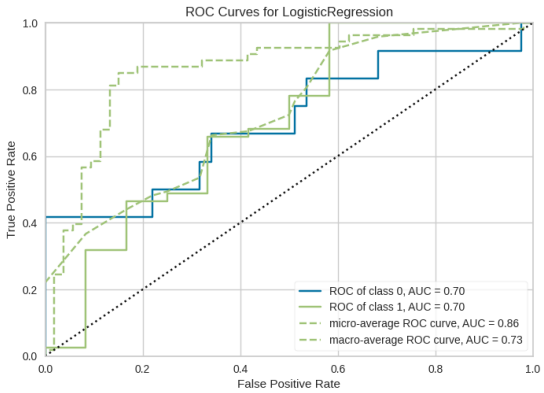

Fig. 13. ROC of LR

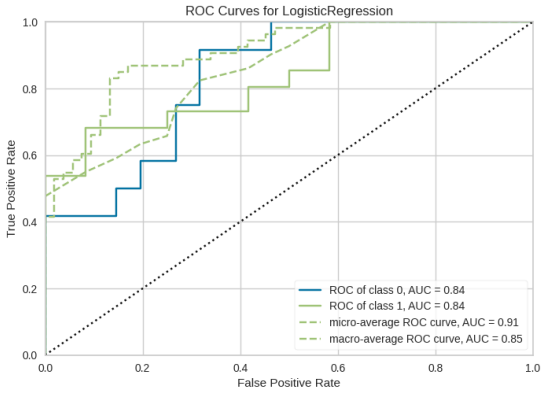

Fig. 14. ROC of PCA + LR

TABLE II
AUC VALUES BEFORE AND AFTER PCA

Model	PCA	AUC
K-Nearest Neighbors	No	0.77
K-Nearest Neighbors	Yes	0.95
Decision Tree	No	0.81
Decision Tree	Yes	0.85
Logistic Regression	No	0.7
Logistic Regression	Yes	0.84

distribute the total gain among players depending on their contribution to the overall result. Here players are an analogy of features and total gain is an analogy of prediction. So the Shapley value of a feature computes how much a particular feature contributes to the final prediction. Here, we have utilized 2 functions:

– **interpret_model(Model,plot='summary')** It shows most influential features across all predictions.The illustration can be shown in Figure 15.Here the model considered is Decision Tree after PCA. Here the X axis shows ShAP values and Y axis shows two principal components.Lighter colors dots shows lower values in a particular feature. As we can see in the Figure 15 for pca0 majority values are below zero indicating that the feature decreases the model output when pca0 is high and lower values of pca0 tend to increase the model output. For pca1 the distribution is more balanced resulting into more mixed impact on prediction.

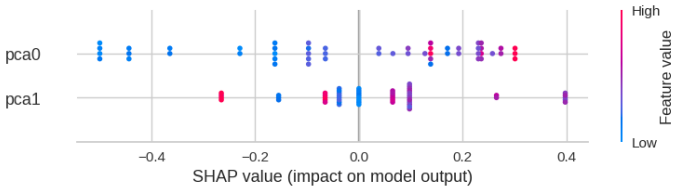

Fig. 15. Impact of PCA Features on Model Output

- **interpret_model(Model,plot='reason')** Here the X axis shows index of data points.Here data points are grouped together based on similar feature values. Y axis shows feature values. For first instance we can say that the pca1 had positive contribution where as pca0 had a negative contribution.

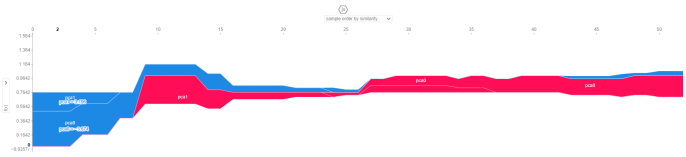

Fig. 16. Force Plot: Feature Contributions to Model Predictions

VIII. Conclusion

In this study, we trained, tested and tuned three popular classification models K Nearest Neighbor, Decision Tree, and Logistic Regression on two datasets, original data set and the other reduced dimension dataset after identifying two principal components. In Table I we were able to compute Accuracy Precision and Recall score for each combination. Overall, without PCA, logistic regression outperformed K-Nearest Neighbors and Decision Tree marginally with an accuracy score of 0.88. After using the transformed dataset K-Nearest Neighbors had the highest accuracy of 0.9. Overall K-Nearest Neighbors had the best accuracy score and with transformed dataset also it performs well on Precision and Recall score.

Table II illustrates the observed values of area under the curve by the three models. Here Decision Tree has the highest value when original data set is used with the value of 0.81. After using transformed data set K-Nearest Neighbors gives the highest AUC score of 0.95.

It is important to note that using transformed data set significantly increases the AUC values of three models where as it is not true in case of Accuracy Precision and Recall. But overall K-Nearest Neighbors outperforms other 2 models in most of the Metric to classify Parkinson's Disease.

References

[1] Parkinson's Foundation, "Statistics," Parkinson's Foundation, [Online]. Available: https://www.parkinson.org/understanding-parkinsons/statistics. [Accessed: 13-Apr-2025].

[2] S. Singh and W. Xu, "Robust detection of Parkinson's disease using harvested smartphone voice data: A telemedicine approach," Telemed. J. E Health, vol. 26, no. 3, pp. 327–334, Mar. 2020, doi: 10.1089/tmj.2018.0271, Epub 2019 Apr. 26, PMID: 31033397; PMCID: PMC7071066.

[3] UCI Machine Learning Repository, "Parkinson's Data Set," [Online]. Available: https://archive.ics.uci.edu/dataset/174/parkinsons. [Accessed: 13-Apr-2025].

[4] V. Ukani, "Parkinson's Disease Data Set," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/vikasukani/parkinsons-disease-data-set. [Accessed: 13-Apr-2025].

[5] P. Schober, C. Boer, and L. A. Schwarte, "Correlation Coefficients: Appropriate Use and Interpretation," *Anesthesia & Analgesia*, vol. 126, no. 5, pp. 1763–1768, May 2018. doi: 10.1213/ANE.0000000000002864.

[6] A. Ben Hamza, "Advanced statistical approaches to quality," Lecture notes, Concordia Institute for Information Systems Engineering, Concordia Univ., Montreal, QC, Canada, n.d.

[7] myConcordia, "INSE6220 - Advanced Statistical Approaches to Quality," GitHub repository, https://github.com/myconcordia/INSE6220, Accessed: Apr. 14,

[8] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, Catania, Sicily, Italy, Nov. 3–7, 2003, pp. 986–996. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-39964-3_63 [Accessed: Apr. 13, 2025].

[9] L. Ying *et al.*, "Decision tree methods: applications for classification and prediction," *Shanghai Archives of Psychiatry*, vol. 27, no. 2, pp. 130, 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/ [Accessed: 13-Apr

[10] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, vol. 24, no. 1, pp. 12–18, 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900058/ [Accessed: 13-Apr-2025]

[11] C. E. Metz, "Basic Principles of ROC Analysis," *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283–298, 1978.

[12] R. Susmaga, "Confusion matrix visualization," in *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM '04 Conference*, Zakopane, Poland, May 17–20, 2004, pp. 107–116. Springer, 2004.