Shreya Mishra

shreyamishra414@gmail.com

Report on Parkinson Prediction Model

Table of Contents

[Parkinson’s disease 2](#_Toc145282598)

[Citation for the dataset 2](#_Toc145282599)

[License 2](#_Toc145282600)

[Dataset 2](#_Toc145282601)

[Dataset Description 2](#_Toc145282602)

[Attribute Information 2](#_Toc145282603)

[Target Variable 3](#_Toc145282604)

[Models Used 3](#_Toc145282605)

[Techniques used for Data Preprocessing 3](#_Toc145282606)

[Distribution of Variables among people who have Parkinson vs those who do not 5](#_Toc145282607)

[Feature Selection 7](#_Toc145282608)

[Oversampling using SMOTE 8](#_Toc145282609)

[Standardization of dataset 9](#_Toc145282610)

[PCA 10](#_Toc145282611)

[Analysis of Different Classification Models 12](#_Toc145282612)

[Logistic Regression 12](#_Toc145282613)

[Technique 1: Feature Selection, Data Standardization 12](#_Toc145282614)

[Technique 2: Feature Selection, Data Standardization, Oversampling 12](#_Toc145282615)

[Technique 3: Data Standardization, PCA 13](#_Toc145282616)

[Technique 4: Oversampling, Data Standardization, PCA 13](#_Toc145282617)

[Model Evaluation 13](#_Toc145282618)

Report on Parkinson’s Prediction Model

This project focuses on a classification model. We have applied different Machine Learning models to predict the presence of Parkinson’s disease in a patient.

# Parkinson’s disease

Parkinson’s disease is a progressive disorder that affects the nervous system and the parts of the body controlled by the nerves. Symptoms start slowly. The first symptom may be a barely noticeable tremor in just one hand. Tremors are common, but the disorder may also cause stiffness or slowing of movement. Although Parkinson’s disease can’t be cured, medications might significantly improve your symptoms. Occasionally, your health care provider may suggest surgery to regulate certain regions of your brain and improve your symptoms.

# Citation for the dataset

'Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection',

Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM.

BioMedical Engineering OnLine 2007, 6:23 (26 June 2007)

# License

[GNU Free Documentation License 1.3](http://www.gnu.org/licenses/fdl-1.3.html)

# Dataset

The dataset is available at Kaggle

https://www.kaggle.com/datasets/gargmanas/parkinsonsdataset

# Dataset Description

The dataset consists of 24 columns and 195 records.

The dataset contains 23 attributes and 1 target variable.

# Attribute Information

1. name - ASCII subject name and recording number

2. MDVP:Fo(Hz) - Average vocal fundamental frequency

3. MDVP:Fhi(Hz) - Maximum vocal fundamental frequency

4. MDVP:Flo(Hz) - Minimum vocal fundamental frequency

5. MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP - Several measures of variation in fundamental frequency

6. MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA - Several measures of variation in amplitude

7. NHR, HNR - Two measures of ratio of noise to tonal components in the voice

8. status - Health status of the subject (one) - Parkinson's, (zero) - healthy

9. RPDE, D2 - Two nonlinear dynamical complexity measures

10. DFA - Signal fractal scaling exponent

11. spread1, spread2, PPE - Three nonlinear measures of fundamental frequency variation

12. Tonnetz - The set of pitch classes used to characterize each note

# Target Variable

status - Health status of the subject (one) - Parkinson's, (zero) - healthy

# Models Used

1. Logistic Regression

2. Decision Tree

3. Pruned Decision Tree

4. Random Forest

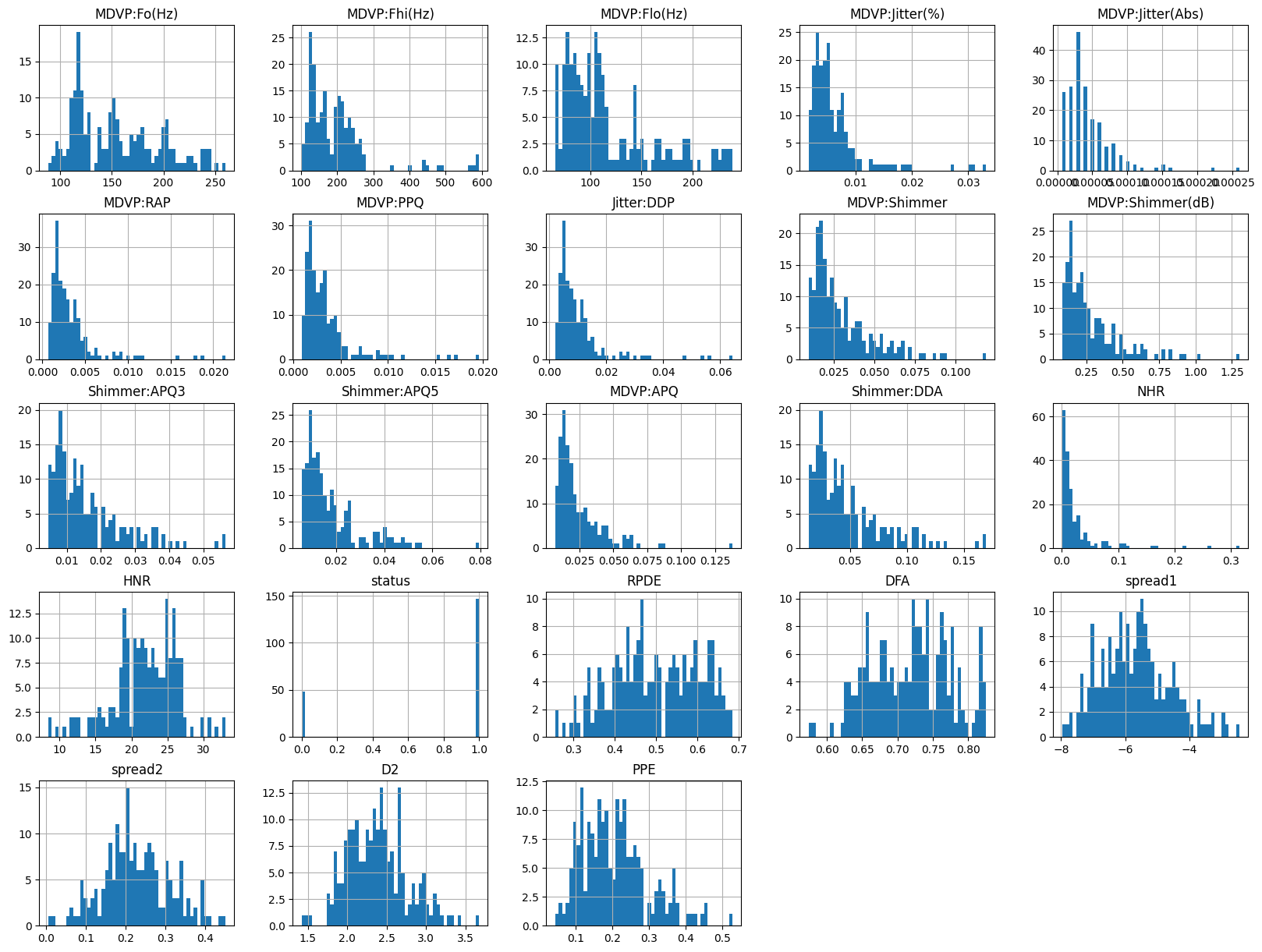
5. XGBClassifier

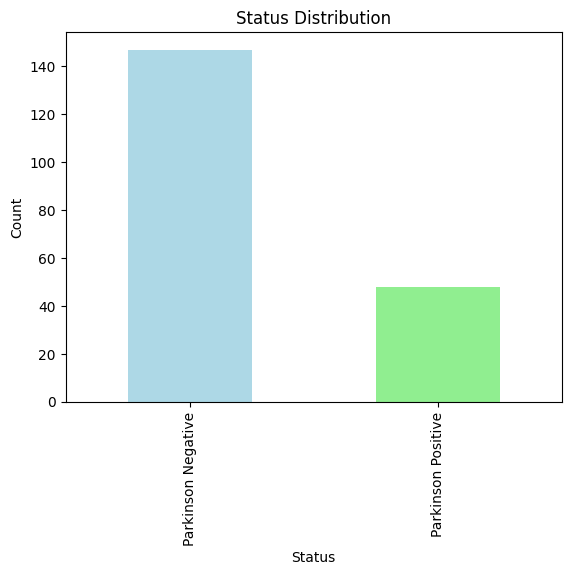
6. Support Vector Machine

# Techniques used for Data Preprocessing

We have 'name', 'MDVP:Fo(Hz)', 'MDVP:Fhi(Hz)', 'MDVP:Flo(Hz)', 'MDVP:Jitter(%)', 'MDVP:Jitter(Abs)', 'MDVP:RAP', 'MDVP:PPQ', 'Jitter:DDP', 'MDVP:Shimmer', 'MDVP:Shimmer(dB)', 'Shimmer:APQ3', 'Shimmer:APQ5', 'MDVP:APQ', 'Shimmer:DDA', 'NHR', 'HNR', 'status', 'RPDE', 'DFA', 'spread1', 'spread2', 'D2', 'PPE' as our columns in parkinsons.csv file. We have dropped the ‘name’ column from the dataframe. We see that all the variables except ‘status’ are continuous numerical variables. The ‘status’ is a categorical variable with values 1 and 0.

1. Parkinson Positive
2. Parkinson Negative





From the above diagram, we can conclude that our dataset is imbalanced. The people with Parkinson positives are only 48 in number out of 195. Thus, when evaluating the model's performance, accuracy alone may not be an appropriate metric.

Imbalanced datasets can lead to a bias in the trained model towards the majority class. In this case, the model may perform well at identifying individuals without Parkinson's disease but poorly at identifying those with the disease, which could be bad.

We might need to employ resampling techniques to address the class imbalance. These techniques include oversampling the minority class, undersampling the majority class, or using more advanced methods like Synthetic Minority Over-sampling Technique (SMOTE) to generate synthetic samples. We have used SMOTE technique later in the model training.

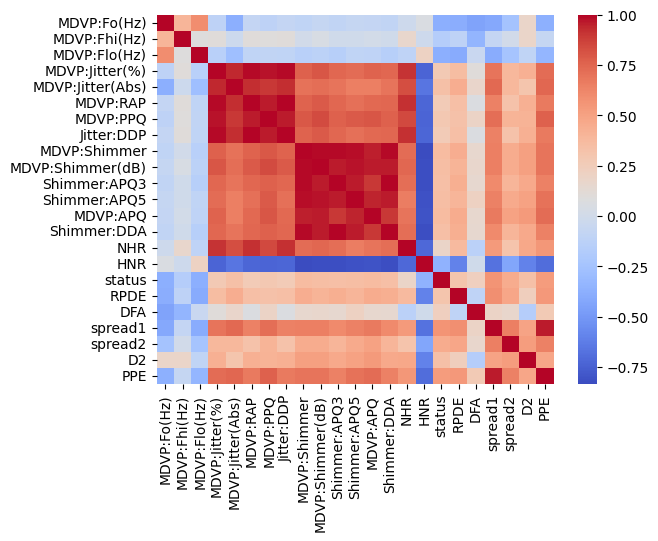
Machine learning algorithms may be more sensitive to class imbalance than others. For instance, decision trees can struggle with imbalanced data, while ensemble methods like Random Forests or Gradient Boosting can handle it better.

## Distribution of Variables among people who have Parkinson vs those who do not

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

From the above visualizations, we can see that the variables are skewed and of different summary statistics when one has Parkinson vs when one does not. The difference in ranges of values such as mean, median, min value, max value, etc can help us determine whether a person has Parkinson or not.

## Feature Selection



The correlation matrix above shows that we have a few variables which are strongly correlated among each other.

Removing highly correlated variables, also known as feature selection or dimensionality reduction, can be beneficial for several reasons in data analysis and modeling:

1. Avoid Multicollinearity
2. Reduced Overfitting
3. Makes data difficult to interpret the impact of individual features on the target variable
4. Reduces Noise
5. Avoids Curse of Dimentionality

We define a threshold for correlation as 0.80. Any variable having correlation greater than 0.80 and less than -0.80 are removed

The columns which are dropped are(Feature Selection): -

HNR  
Jitter:DDP  
MDVP:APQ  
MDVP:Jitter(Abs)  
MDVP:PPQ  
MDVP:RAP  
MDVP:Shimmer(dB)  
NHR  
PPE  
Shimmer:APQ3  
Shimmer:APQ5  
Shimmer:DDA

## Oversampling using SMOTE

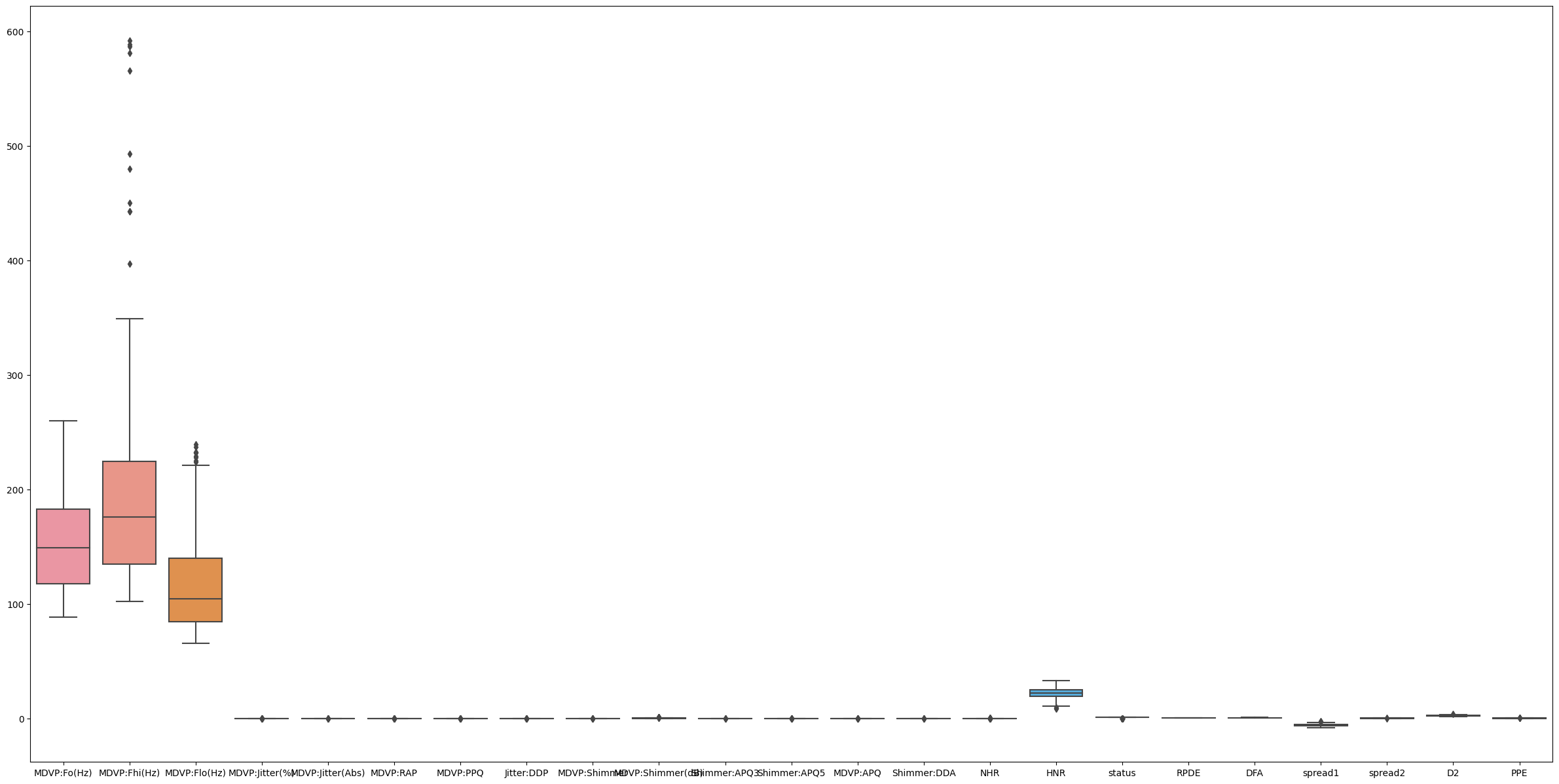
We can see that we have a class imbalance in our dataset. So, the models trained without Oversampling can lead to overfitting of the model. To avoid that, we use SMOTE (Synthetic Minority Over-sampling Technique) technique to generate synthetic samples for the minority class (those who have Parkinson’s) by interpolating between existing samples. This helps balance the class distribution and prevents the model from being biased towards the majority class.

Class distribution after SMOTE:

|  |  |
| --- | --- |
| 1 | 147 |
| 0 | 147 |

In this way, we can balance our class. The other methods to achieve the same are ADASYN (Adaptive Synthetic Sampling), Borderline-SMOTE, Cluster-based Oversampling, etc.

## Standardization of dataset



The boxplot displayed above demonstrates the requirement for standardisation of our data. We used the Standard Scalar Library for our standardisation technique, standardising the independent variables using the Z score.

Machine learning algorithms often make assumptions about the distribution of the data, assuming that features are centred around zero and have a similar scale. If our independent variables have different scales, some features may dominate the learning process, while others may have little influence.

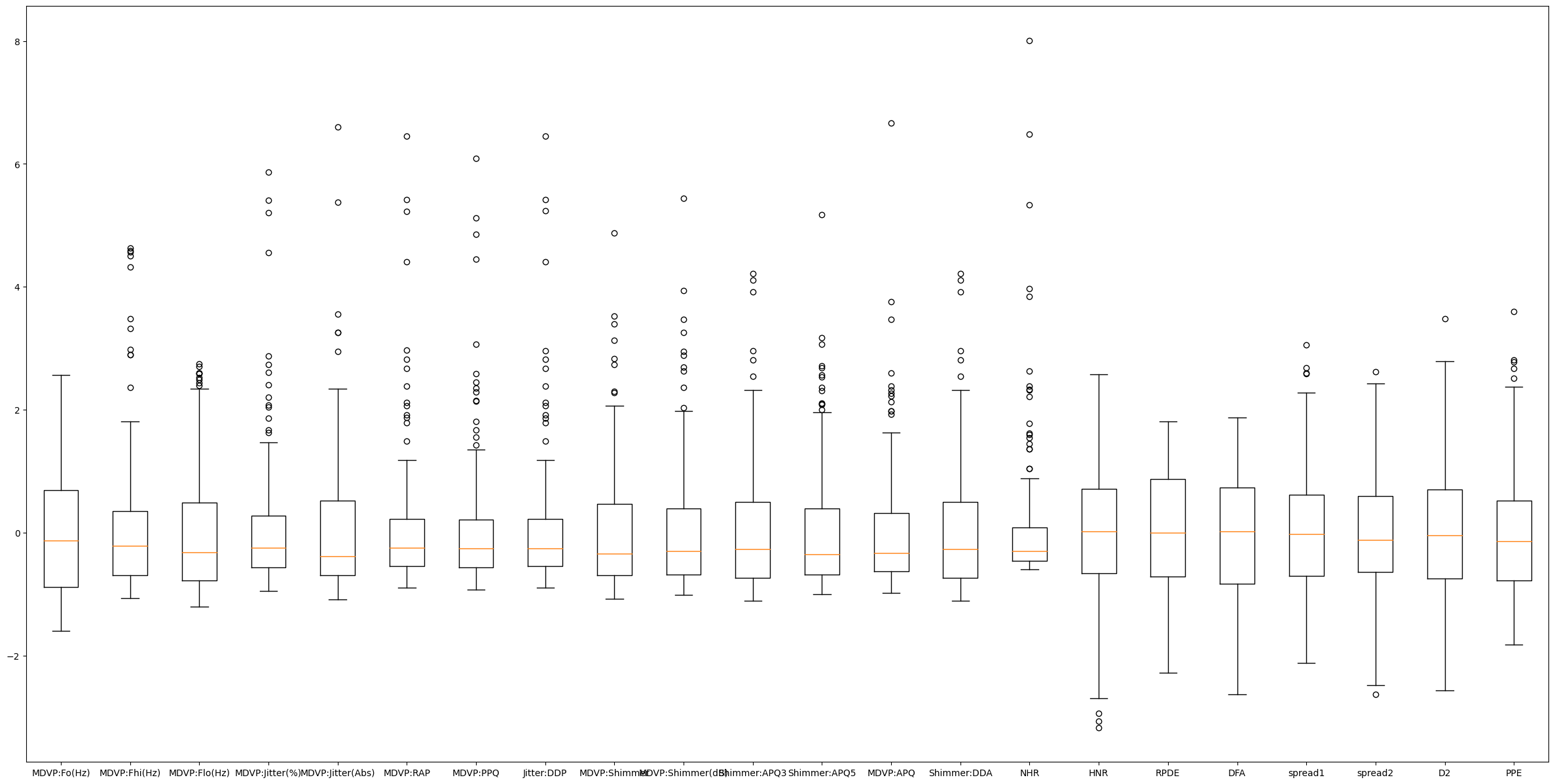
Standardization transforms our data into a standard normal distribution with a mean of 0 and a standard deviation of 1. This makes it easier for machine learning algorithms to work effectively and can lead to better model performance.

The Z score is defined as

z = (x - μ) / σ

where:

**z** is the standardized value.  
**x** is the original value.  
**μ** is the mean of the variable.  
**σ** is the standard deviation of the variable.



The data now looks standardized.

By applying standardization, you ensure that all your independent variables have similar ranges and distributions, which can help your machine learning models converge faster and produce more accurate results.

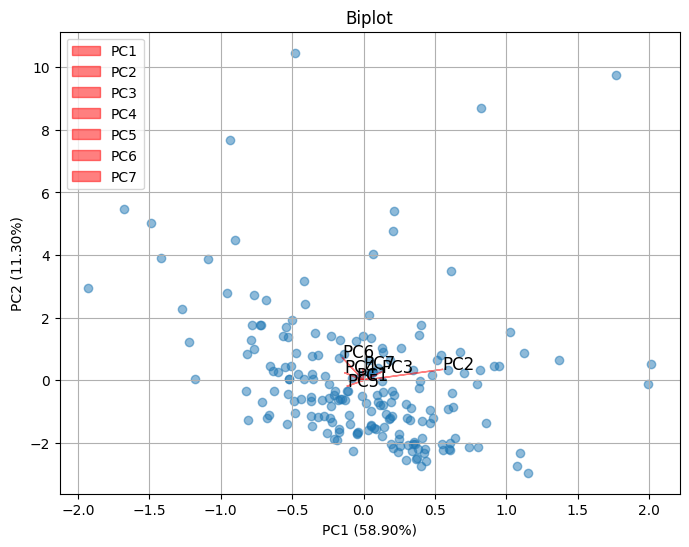
## PCA

We have 22 independent variables in our dataset, but not all of them have been important and our model efficiency decreases due to curse of dimensionality.

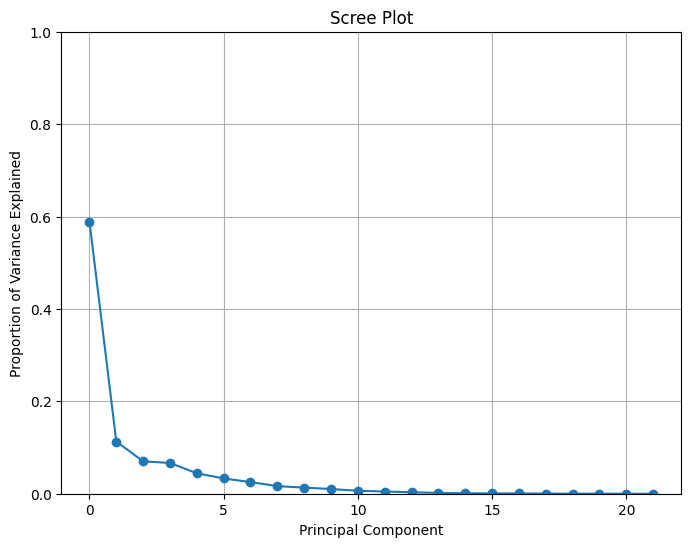
Principal Component Analysis (PCA) is a dimensionality reduction technique that can help address the issue of having too many independent variables (features) in a dataset. It works by transforming the original features into a new set of linearly uncorrelated features called principal components, where the first principal component explains the most variance in the data, the second explains the second most variance, and so on. PCA can be used to reduce the dimensionality of your dataset while preserving as much information as possible.

We initially take all 22 variables and make them undergo dimensionality reduction and transform the data into 22 Principle Components.

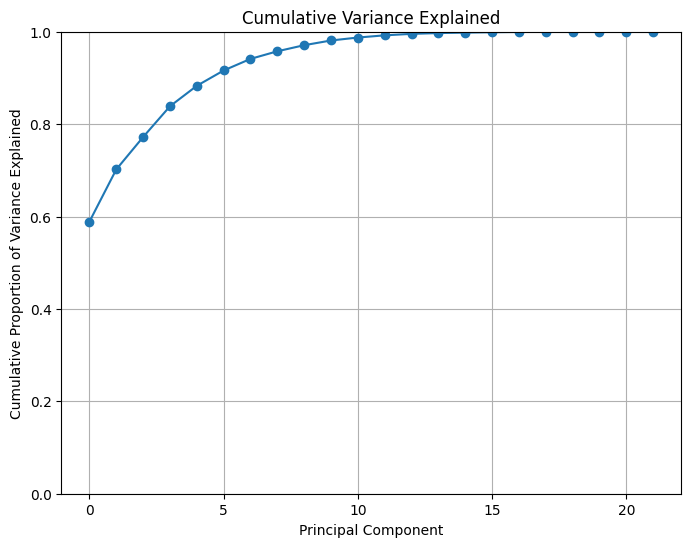
The eigen value for each component is given as: -

Singular Values (Eigenvalues): [5.02676002e+01 2.20169378e+01 1.73405825e+01 1.69018434e+01 1.37809157e+01 1.19237637e+01 1.03772710e+01 8.40646479e+00 7.51787458e+00 6.61094733e+00 5.23547546e+00 4.52151078e+00 3.68764168e+00 2.72808078e+00 2.07178187e+00 1.86240973e+00 1.55852417e+00 1.18606766e+00 8.25730369e-01 4.59963384e-01 8.39986608e-03 2.54148559e-03]  
  
We also visualise a biplot. A biplot is a graphical representation of the results of a Principal Component Analysis (PCA) applied to a dataset. It displays both the principal components (often denoted as PC1 and PC2) and the original variables in a single plot.  


A scree plot is a graphical tool used in Principal Component Analysis (PCA) and factor analysis to help determine the number of principal components or factors to retain. It is a valuable visualization for understanding the variance explained by each component or factor and can assist in making informed decisions about dimensionality reduction.

* In PCA, each principal component represents a linear combination of the original variables and accounts for a certain amount of the total variance in the data.
* The scree plot displays the explained variance by each principal component in descending order.
* The "elbow" point in the scree plot is where the explained variance begins to level off. The number of components before this point is often chosen as the number of components to retain.  
  

Cumulative explained variance represents the cumulative amount of variance explained by a subset of the principal components or factors. Calculating cumulative variance is useful for determining how much of the total variance in the dataset is captured as you retain an increasing number of components.



After all the above results, we go with 7 Principal components for our dataset as after 7 components, the variance explained is near to zero.

# Analysis of Different Classification Models

## Logistic Regression

Logistic regression is a statistical model used for binary classification and sometimes for multiclass classification problem. It uses the logistic function (also known as the sigmoid function) to map the linear combination of input features to a probability between 0 and 1.

The logistic (sigmoid) function is an S-shaped curve that smoothly transitions between 0 and 1. It is used to model the probability that a given input belongs to the positive class. The formula for the logistic function is:

P(Y=1|X) = 1 / (1 + e^(-z))

Where, **P(Y=1|X)** is the probability of the positive class.  
**X** represents the input features.  
**z** is the linear combination of the input features and model parameters.

### Technique 1: Feature Selection, Data Standardization

**Train Data**

True Positive 113   
False Positive 13   
True Negative 23   
False Negative 7   
Accuracy: 0.8717948717948718   
Precision: 0.8968253968253969   
Recall: 0.9416666666666667   
F1\_score: 0.9186991869918699   
confusion matrix [[ 23 13] [ 7 113]]

**Test Data**

True Positive 27   
False Positive 6   
True Negative 6   
False Negative 0   
Accuracy: 0.8461538461538461   
Precision: 0.8181818181818182   
Recall: 1.0   
F1\_score: 0.9   
confusion matrix [[ 6 6] [ 0 27]]

### Technique 2: Feature Selection, Data Standardization, Oversampling

**Train Data**

True Positive 100   
False Positive 24   
True Negative 92   
False Negative 19   
Accuracy: 0.8170212765957446   
Precision: 0.8064516129032258   
Recall: 0.8403361344537815   
F1\_score: 0.8230452674897119   
confusion matrix [[ 92 24] [ 19 100]]

**Test Data**

True Positive 24   
False Positive 5   
True Negative 26   
False Negative 4   
Accuracy: 0.847457627118644   
Precision: 0.8275862068965517   
Recall: 0.8571428571428571   
F1\_score: 0.8421052631578947   
confusion matrix [[26 5] [ 4 24]]

### Technique 3: Data Standardization, PCA

**Train Data**

True Positive 108   
False Positive 17   
True Negative 24   
False Negative 7   
Accuracy: 0.8461538461538461   
Precision: 0.864   
Recall: 0.9391304347826087   
F1\_score: 0.8999999999999999   
confusion matrix [[ 24 17] [ 7 108]]

**Test Data**

True Positive 32   
False Positive 4   
True Negative 3   
False Negative 0   
Accuracy: 0.8974358974358975   
Precision: 0.8888888888888888   
Recall: 1.0   
F1\_score: 0.9411764705882353   
confusion matrix [[ 3 4] [ 0 32]]

### Technique 4: Oversampling, Data Standardization, PCA

**Train Data**

True Positive 96   
False Positive 19   
True Negative 98   
False Negative 22   
Accuracy: 0.825531914893617   
Precision: 0.8347826086956521   
Recall: 0.8135593220338984   
F1\_score: 0.8240343347639484   
confusion matrix [[98 19] [22 96]]

**Test Data**

True Positive 25   
False Positive 5   
True Negative 25   
False Negative 4   
Accuracy: 0.847457627118644   
Precision: 0.8333333333333334   
Recall: 0.8620689655172413   
F1\_score: 0.847457627118644   
confusion matrix [[25 5] [ 4 25]]

### Model Evaluation

Based on both accuracy and recall, it appears that **Technique 3 (Data Standardization, PCA)** performs the best on the test data. It has the highest accuracy (0.8974) and recall (1.0), indicating that it correctly identifies all positive cases without making too many false positives.

**Overfitting or Underfitting:**

* Technique 1 shows good results but might be slightly overfitting on the training data as it has a higher recall on the training data compared to the test data.
* Technique 2 also shows signs of overfitting, as it has a lower recall on the test data compared to the training data.
* Technique 4 seems to be well-balanced between training and test data, indicating a more generalized model.

In summary, based on the provided metrics and considering the balance between training and test performance, **Technique 3 (Data Standardization, PCA)** appears to be the best model among the four. It achieves the highest accuracy and recall on the test data while avoiding signs of overfitting.

## Decision Tree

A Decision Tree is a supervised machine learning algorithm that is used for both classification and regression tasks. It is a graphical representation of decisions and their consequences in the form of a tree-like structure. Decision Trees are widely used because they are easy to understand, interpret, and visualize.

Decision Trees tend to overfit the training data, which means they can capture noise in the data and perform poorly on unseen data. Techniques like pruning, limiting the tree depth, and setting minimum samples per leaf can help prevent overfitting.

Decision Trees can provide a measure of feature importance, indicating which features are more influential in making decisions at higher levels of the tree.

### Technique 1: Feature Selection, Data Standardization

**Train Data**

True Positive 113   
False Positive 13   
True Negative 23   
False Negative 7   
Accuracy: 0.8717948717948718   
Precision: 0.8968253968253969   
Recall: 0.9416666666666667   
F1\_score: 0.9186991869918699   
confusion matrix [[ 23 13] [ 7 113]]

**Test Data**

True Positive 27   
False Positive 6   
True Negative 6   
False Negative 0   
Accuracy: 0.8461538461538461   
Precision: 0.8181818181818182   
Recall: 1.0   
F1\_score: 0.9   
confusion matrix [[ 6 6] [ 0 27]]

### Technique 2: Feature Selection, Data Standardization, Oversampling

**Train Data**

True Positive 100   
False Positive 24   
True Negative 92   
False Negative 19   
Accuracy: 0.8170212765957446   
Precision: 0.8064516129032258   
Recall: 0.8403361344537815   
F1\_score: 0.8230452674897119   
confusion matrix [[ 92 24] [ 19 100]]

**Test Data**

True Positive 24   
False Positive 5   
True Negative 26   
False Negative 4   
Accuracy: 0.847457627118644   
Precision: 0.8275862068965517   
Recall: 0.8571428571428571   
F1\_score: 0.8421052631578947   
confusion matrix [[26 5] [ 4 24]]

### Technique 3: Data Standardization, PCA

**Train Data**

True Positive 108   
False Positive 17   
True Negative 24   
False Negative 7   
Accuracy: 0.8461538461538461   
Precision: 0.864   
Recall: 0.9391304347826087   
F1\_score: 0.8999999999999999   
confusion matrix [[ 24 17] [ 7 108]]

**Test Data**

True Positive 32   
False Positive 4   
True Negative 3   
False Negative 0   
Accuracy: 0.8974358974358975   
Precision: 0.8888888888888888   
Recall: 1.0   
F1\_score: 0.9411764705882353   
confusion matrix [[ 3 4] [ 0 32]]

### Technique 4: Oversampling, Data Standardization, PCA

**Train Data**

True Positive 96   
False Positive 19   
True Negative 98   
False Negative 22   
Accuracy: 0.825531914893617   
Precision: 0.8347826086956521   
Recall: 0.8135593220338984   
F1\_score: 0.8240343347639484   
confusion matrix [[98 19] [22 96]]

**Test Data**

True Positive 25   
False Positive 5   
True Negative 25   
False Negative 4   
Accuracy: 0.847457627118644   
Precision: 0.8333333333333334   
Recall: 0.8620689655172413   
F1\_score: 0.847457627118644   
confusion matrix [[25 5] [ 4 25]]

### Model Evaluation