**PARKINSONS DISEASE CLASSIFICATION USING VOCAL BIOMARKERS**

**INT510 – INTERNSHIP PROJECT REPORT**

***Submitted by***

**VIDARSHANA G – E0119067**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

Artificial Intelligence and Machine Learning

**Sri Ramachandra Faculty of Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**JUNE, 2023**

**PARKINSONS DISEASE CLASSIFICATION USING VOCAL BIOMARKERS**

**INT510 – INTERNSHIP PROJECT REPORT**

***Submitted by***

**VIDARSHANA G – E0119067**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

Artificial Intelligence and Machine Learning

**Sri Ramachandra Faculty of Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai -600116**

**JUNE,2023**

**BONAFIDE CERTIFICATE**

Certified that this project report **“PARKINSON’S DISEASE CLASSIFICATION USING VOCAL BIOMARKERS”** is the bonafide record of work done by **“VIDARSHANA G – E0119067”** who carried out the internship work under my supervision.

**Signature of the Supervisor Signature of Programme Coordinator**

|  |  |
| --- | --- |
| **Prof.Ragavan**  **Lecturer,**  Department of Artificial Intelligence and Machine Learning  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. | **Dr. Vanitha Magesh**  **Associate Professor,**  Department of Artificial Intelligence and Machine Learning  Sri Ramachandra Faculty of Engineering and Technology,  SRIHER, Porur, Chennai-600 116. |

**Evaluation Date:**

**ACKNOWLEDGEMENT**

I express my sincere gratitude to our Programme Coordinator **Prof. Vanitha Magesh** for their support and for providing the required facilities for carrying out this study.

I wish to thank my faculty supervisor(s), **Prof. Ragavan,** Department of Artificial Intelligence and Machine Learning, Sri Ramachandra faculty of Engineering and Technology for extending help and encouragement throughout the project. Without his/her continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to all the members of Sri Ramachandra Faculty of Engineering and Technology, my beloved parents and friends for extending the support, who helped us to overcome obstacles in the study.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** |  |
|  | **LIST OF TABLES** |  |
|  | **LIST OF FIGURES** |  |
| **1** | **INTRODUCTION** | **1** |
|  | * 1. PARKINSON’S DISEASE | 1 |
|  | * 1. CURRENT DIAGNOSIS METHODS | 2 |
|  | 1.3HOW VOCAL BIOMARKERS A USEFUL METHOD OF DIAGNOSIS? | 3 |
| **2** | **LITERATURE REVIEW** | 4 |
| **3** | **OBJECTIVE** | **6** |
| **4** | **METHODOLOGY** | **7** |
| **5** | **TOOLS AND TECHNIQUES** | 8 |
| **6** | **IMPLEMENTATION**  6.1 ANALYSING DIFFERENT OVERSAMPLING TECHNIQUES WITH MACHINE LEARNING FOR CLASSIFICATION OF PARKINSON’S DISEASE  6.2. DETAILED ANALYSIS WITH HYBRID IMBALANCED TECHNIQUE | 9 |
| **7** | **RESULTS AND DISCUSSIONS** | 18 |
| **8** | **CONCLUSION** | 20 |
|  | **APPENDICIES** | 21 |
|  | Appendix-1: Code – Technical detail | 21 |
|  | Appendix-2: Screenshots | 24 |
|  | **REFERENCES** | 29 |
|  | **WORKLOG** | 30 |
|  |  |  |

**ABSTRACT**

In this study, a novel technique for classifying Parkinson's disease using features gleaned from speech signals is proposed. Early Parkinson's disease detection and diagnosis are crucial for the disease's progression and the course of treatment. The dataset for Parkinson's disease used in this study was found in the machine learning repository at UCI. The dataset for Parkinson's disease (PD dataset) consists of two classes. 147 data are from the diseased class (PD), while 48 data are from healthy (normal) people.

The class distribution in the data set is unbalanced. Different oversampling techniques were used to address the imbalanced data situation before addressing the class imbalance. Following that, a variety of classification models were trained on the data, and SVM and KNN were found to perform the best overall. After that, hyper parameter tuning was done to increase the two models' accuracy. The proposed hybrid method for the following section, which represents a novel approach, entails two steps:

1. Gathering the newly sampled data from the minority class from SMOTE and ADASYN and producing a new subset of data
2. Classification.

There are 23 attributes in the PD dataset. In the classification of the PD dataset, the Support Vector classification rate was 94.89%, while the proposed hybrid method (the combination of SMOTE and ADASYN) achieved classification success of 95.5%. Results obtained indicated that using this hybrid approach, promising results had been obtained in the discrimination of the PD dataset.

**LIST OF FIGURES**

**Figure No. and Figure Name**

Fig. 4.1 Methodology 7

Fig. 6.1 Proposed Methodology 17

Fig. 6.1.1Scatter Plot for SMOTE 10

Fig. 6.1.2 Scatter Plot for ADASYN 11

Fig. 6.1.3 Scatter Plot for Borderline SMOTE 12

Fig.6.1.4 Scatter Plot for SVMSMOTE 12

Fig.6.1.5 Scatter Plot for KmeansSMOTE 13 Fig.6.2 Scatter Plot for SMOTE AND ADASYN 16

**LIST OF TABLES**

**Table No. and Table Name**

Table. 7.1 Results after using different Over Sampling Techniques 18

Table 7.2 Results from Hyper Parameter Tuning 19

Table 7.3 Results from Hybrid Approach 19

**CHAPTER 1**

**INTRODUCTION**

* 1. **PARKINSONS DISEASE**

Parkinson’s disease is a brain disorder that causes unintended or uncontrollable movements, such as shaking, stiffness, and difficulty with balance and coordination.

Symptoms usually begin gradually and worsen over time. As the disease progresses, people may have difficulty walking and talking. They may also have mental and behavioral changes, sleep problems, depression, memory difficulties, and fatigue.

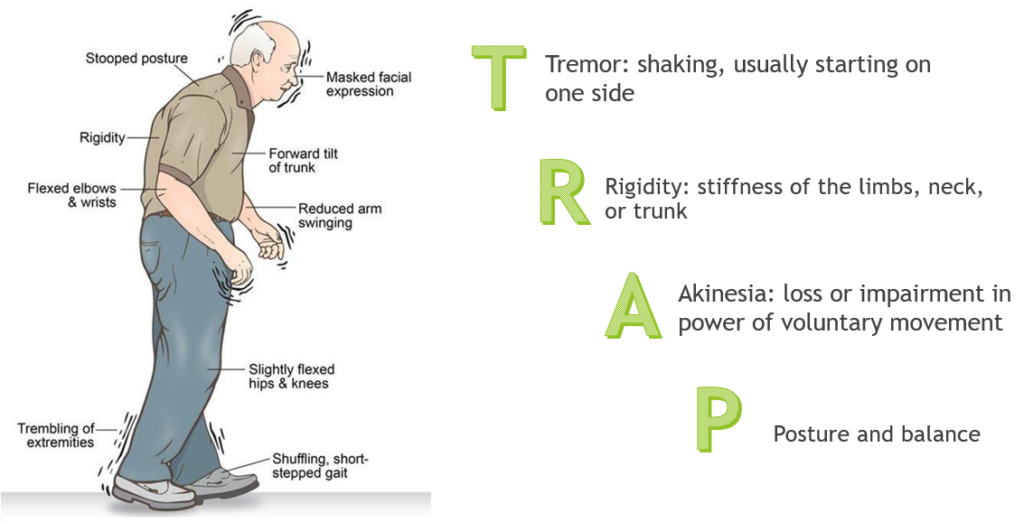
While virtually anyone could be at risk for developing Parkinson’s, some research studies suggest this disease affects more men than women. It’s unclear why, but studies are underway to understand factors that may increase a person’s risk. One clear risk is age: Although most people with Parkinson’s first develop the disease after age 60, about 5% to 10% experience onset before the age of 50. Early-onset forms of Parkinson’s are often, but not always, inherited, and some forms have been linked to specific alterations in genes.

The most prominent signs and symptoms of Parkinson’s disease occur when nerve cells in the basal ganglia, an area of the brain that controls movement, become impaired and/or die. Normally, these nerve cells, or neurons, produce an important brain chemical known as dopamine. When the neurons die or become impaired, they produce less dopamine, which causes movement problems associated with the disease. Scientists still do not know what causes the neurons to die.

* 1. **CURRENT DIAGNOSIS METHODS**

There is no specific lab or imaging test that can diagnose PD. However, certain tests such as magnetic resonance imaging of the brain (MRI brain), a dopamine transporter scan (DaT scan), or blood work can be used to support the diagnosis of PD or to rule out other medical conditions that can mimic PD.

Making an accurate diagnosis of Parkinson’s, particularly in its early **stages**, can be difficult. Often, an internist or family physician is the first to make a diagnosis. Many people may seek an additional opinion from a **movement disorder specialist**. A movement disorder specialist is a neurologist with experience and specific training in the assessment and treatment of PD and related disorders.



To consider a diagnosis of Parkinson’s disease, a person must have bradykinesia (slowness of movement). In addition to bradykinesia, a person must also have one or more of the following:

* Expression and speech are animated
  + Tremor can be observed in your extremities at rest or in action
  + There is stiffness in your extremities or neck
  + There are changes to your walking, step size, and ability to turn
  + You can maintain your balance and examine your posture
  1. **HOW VOCAL BIOMARKERS ARE A USEFUL METHOD OF DIAGNOSIS?**

Vocal biomarkers are important for diagnosing Parkinson's disease due to the following reasons:

1. Non-Invasive and Cost-Effective: Vocal biomarkers provide a non-invasive and affordable way to assess Parkinson's disease.

2. Reflect Motor Control Impairments: Parkinson's disease affects motor control, including vocal patterns, making vocal biomarkers indicative of the neurological changes associated with the disease.

3. Early Detection and Monitoring: Vocal biomarkers can detect Parkinson's disease in its early stages, allowing for timely intervention and management.

4. Objective and Quantifiable Measures: Vocal biomarkers offer objective and quantifiable measurements for accurate diagnosis and monitoring, reducing subjectivity.

5. Potential for Remote Monitoring: Vocal biomarkers enable remote monitoring of Parkinson's disease, facilitating regular assessments without in-person visits.

6. Complementary to Existing Methods: Vocal biomarkers complement other diagnostic methods, enhancing accuracy and reliability.

7. Patient-Friendly and Widely Accessible: Vocal biomarkers are patient-friendly and accessible to a wide range of individuals, including those with limited mobility or cognitive impairments.

**CHAPTER 2**

**LITERATURE REVIEW**

[1] This paper discusses to transform this imbalanced dataset to balanced dataset, SMOTE method is used. Then, Random Forests classification method was used for classification of Parkinson's disease dataset. It is seen from the obtained results that the proposed hybrid model (SMOTE and Random Forests classifier in the classification   
of PD dataset with the 10-fold cross-validation) has achieved good results in the discrimination of Parkinson disease dataset having a class-imbalanced problem.

[2] In this Research Paper, First, the data is collected and the performance is evaluated by data preprocessing -, such as standardization and, normalization to improve the quality of data. Then the data is split into train and test and fed into Parametric (Naïve Bayes, Logistic regression) and Non parametric (KNN and Random forest) models. Nonparametric models using Random Forest and K-Nearest Neighbors produced higher classification accuracy of 87.2% and 90.2% compared to parametric models.

[3]The features in this research paper include fundamental frequencies, harmonicity variants, time-frequency attributes, wavelet and vocal-based features, and many other speech signal data. Dimension reduction techniques, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), were used to reduce the number of features. During classification the procedure, k-fold cross-validation was performed to maintain the class distribution proportion as close as original data-set. Dierent Machine Learning  
 classifiers, such as Support Vector Machine (SVM), Logistic Regression (LR), k  
Nearest Neighbour (k-NN), AdaBoost (AdB), Random Forrest (RF) were used  
 in the experiment. Grid search was used for hyper-parameter tuning and optimize the classifiers performance. For classification on raw data RF showed the best performance. SMV showed the highest overall performance with an accuracy 94.1%. For tree based classifiers (such as : AdB, RF), AdB showed superior accuracy of 90.4%. k-NN showed the lowest peak performance of 86.3%.

# [4] In this paper, minimum redundancy maximum relevance feature selection algorithms is used to select the most important feature among all the features to predict the Parkinson diseases. Here, it is observed that the random forest with 20 number of features selected by minimum redundancy maximum relevance feature selection algorithms provide the overall accuracy 90.3%, precision 90.2%, Mathews correlation coefficient values of 0.73 and ROC values 0.96 which is better in comparison to all other machine learning based approaches such as bagging, boosting, random forest, rotation forest, random subspace, support vector machine, multilayer perceptron, and decision tree based methods.

[5] This paper explores the effectiveness of using supervised classification algorithms, such as deep neural networks, to accurately diagnose individuals with the disease. Our peak accuracy of 85% provided by the machine learning models exceeds the average clinical diagnosis accuracy of non-experts (73.8%) and average accuracy of movement disorder specialists (79.6% without follow-up, 83.9% after follow-up) with pathological post-mortem examination as ground truth. The Gradient Boosted Classifier performed well on nearly every metric. The classifier was able to generate the best overall accuracy scores of 86% for the AVEC features and 82% for the GeMaps features. The gradient boosted classifier also performed the best on the ROC AUC score with .924 and .892 for AVEC and GeMaps respectively. This indicates that this model can produce the best separation of the two classes- PD and control.

**CHAPTER 3**

**OBJECTIVE**

The objective of this project is to develop a machine-learning model for the classification of Parkinson's disease using vocal biomarkers. By leveraging advanced machine learning techniques, the project aims to accurately distinguish between individuals with Parkinson's disease and healthy individuals based on vocal characteristics extracted from speech samples.

Vocal biomarkers have the potential to detect Parkinson's disease at its early stages, even before the emergence of other motor symptoms. By analyzing vocal characteristics, such as changes in pitch, loudness, speech rate, and articulation, it may be possible to identify subtle abnormalities that indicate the presence of Parkinson's disease.

* Analyzing the dataset available and understanding the common range for healthy and Parkinson’s patient.
* Perform Exploratory Data Analysis
* Work with speech dataset and perform detailed analysis using hybrid bala technique to classify the data into Parkinson’s and non-Parkinsons by using the machine learning algorithm.

**CHAPTER 4**

**METHODOLOGY**

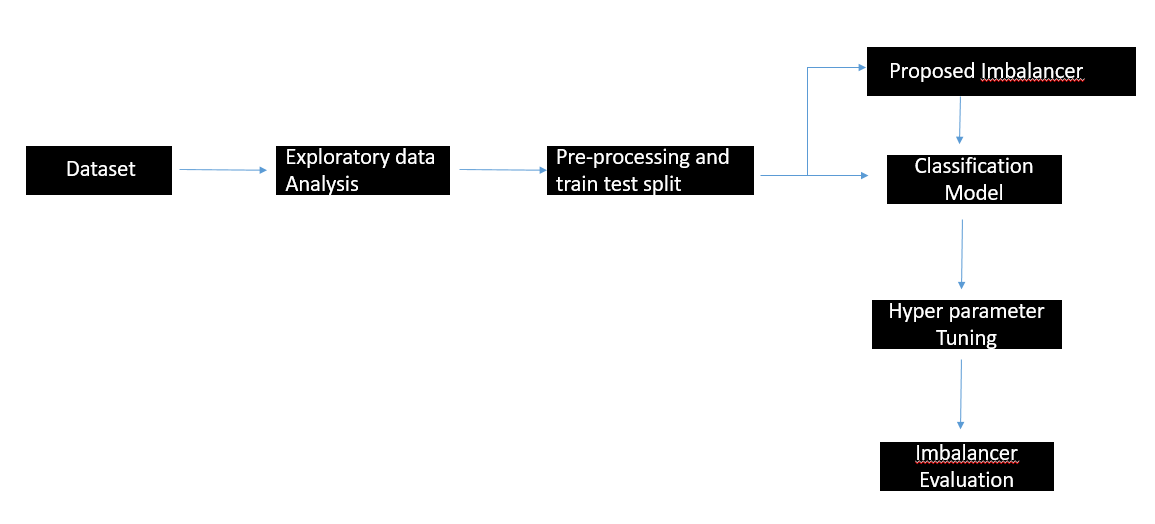
****

Fig 4.1 Methodology

This project aims to classify Vocal biomarkers of individuals using Machine learning techniques The speech data were sourced from UCI machine learning repository which has 23 attributes . They were divided into 2 classes of images namely; Parkinson’s disease and non-Parkinson’s disease.

But the data was imbalanced.ie twenty-five percent data belonged to healthy individuals whereas 75% of data belonged to Parkinson’s patients. Therefore in order to correct the imbalance data situation, we have tried different oversampling techniques to make the number of healthy an Parkinson samples equal models were utilized to predict the images into the 4 different classes.

Once that was done, we perform hyper parameter tuning for the classification models which shows higher accuracy consistently throughout the different up sampling techniques. With the results, we determine the two important oversampling techniques and build a hybrid model.

**CHAPTER 5**

**TOOLS AND TECHNIQUES**

There are several tools commonly used for machine learning classification, depending on the specific application and the level of expertise of the practitioner. Here are some popular tools used for machine learning classification:

* Oversampling Techniques: Oversampling is a technique used in data analysis and machine learning to address the issue of imbalanced datasets, where one class or category is significantly underrepresented compared to others. It involves creating additional samples for the minority class to balance the data distribution. Several oversampling techniques are available, including SMOTE (Synthetic Minority Over-Sampling Technique), which generates synthetic samples along the line segments connecting neighboring instances of the minority class; ADASYN (Adaptive Synthetic Sampling), which focuses on generating samples in regions where the dataset is sparser; and Borderline-SMOTE, which selectively synthesizes samples near the decision boundary. Other techniques include SVM SMOTE, KMEANS SMOTE, and SMOTEEN. These techniques aim to improve the performance and generalizability of machine learning models by providing a more representative and balanced dataset for training.
* Scikit-learn: Scikit-learn is a popular Python library that offers a wide range of machine-learning algorithms and tools for classification tasks. It provides implementations of various algorithms like support vector machines (SVM), random forests, and gradient boosting. TensorFlow and PyTorch are deep learning frameworks that allow for the development of neural network models for more complex tasks.
* Libraries such as NumPy and pandas are essential for data analysis and manipulation. NumPy provides efficient numerical computations, while pandas offers data structures and functions for data preprocessing, feature extraction, and data manipulation tasks.
* MATLAB: It is a programming language and environment that includes many machine learning tools and features, including support for popular machine learning frameworks.

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 Analysing Different Oversampling Techniques With Machine Learning For Classification Of Parkinson’s Disease:**

**6.1.1. ABOUT THE DATASET:**

The Data is hand collected from various websites with each and every labels verified.

**Source** :UCI Machine Learning Repository **Columns** : 23 Columns

**Rows** : 195 entries

**Attribute Information:**

**name** - ASCII subject name and recording number

**MDVP: Fo(Hz)** - Average vocal fundamental frequency

**MDVP: Fhi(Hz)** - Maximum vocal fundamental frequency

**MDVP: Flo(Hz)** - Minimum vocal fundamental frequency

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

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

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

**status** - The health status of the subject (one) - Parkinson's, (zero) - healthy

**RPDE, D2** - Two nonlinear dynamical complexity measures

**DFA - Signal** fractal scaling exponent

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

**6.1.2. PREPROCESSING**

Pre-processing is an important step in machine learning and data analysis. It refers to the steps and techniques applied to the raw data before it is used for training a machine learning model. Preprocessing is an essential part of the machine learning pipeline as it helps to transform the data into a suitable format for effective modeling and analysis. The main objectives of preprocessing are:

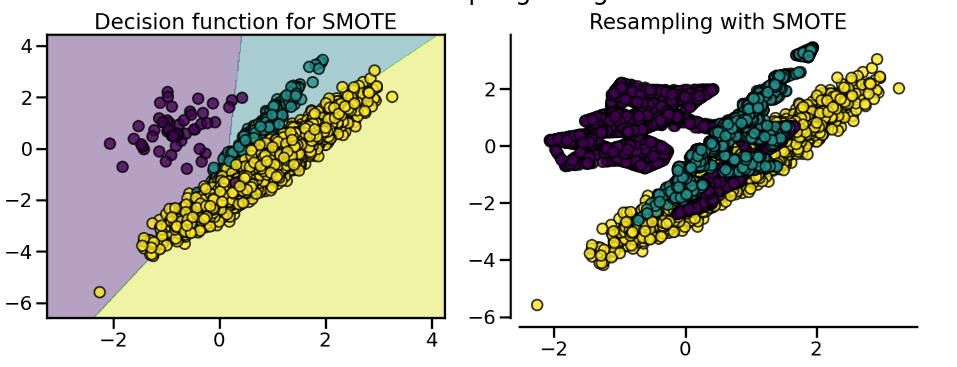
1. *Data Cleaning*: This involves handling missing data, removing outliers, and addressing any inconsistencies or errors in the dataset. Our dataset had columns that were not necessary for our prediction. Therefore that column was dropped

2. *Data Transformation*: Data transformation techniques are used to convert the data into a format that meets the assumptions of the chosen machine learning algorithm. This project involves scaling the features to a specific range (standardization), and transforming skewed distributions using techniques like log transformations or power transformations.

3. *Feature Extraction*: Feature extraction involves selecting or creating relevant features from the available data that can best represent the underlying patterns and relationships. The least correlated column to the target variable is dropped and the correlation score is calculated for each attribute and insights are gained as to which attributes are highly related to the target column.

4.*Handling Imbalanced Data*: Imbalanced data occurs when the classes in the target variable are not represented equally. Preprocessing techniques for imbalanced data include oversampling the minority class (SMOTE, ADASYN, Borderline-SMOTE, SVMSMOTE, KMEANSSMOTE, SMOTENN) to balance the class distribution.

* + 1. **SMOTE (Synthetic Minority Over-sampling Technique):**



Advantages:

- Helps address class imbalance by generating synthetic samples of the minority class.

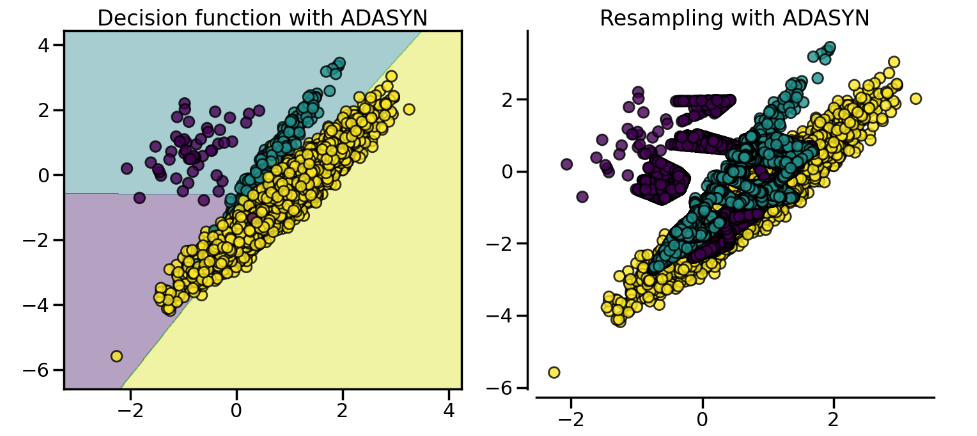
- Can increase the overall predictive performance for the minority class.

Disadvantages:

- Can introduce noisy samples if the synthetic samples are not well aligned with the actual minority class distribution.

- Does not consider the underlying data distribution and may generate samples in regions that are unrealistic.

* + 1. **ADASYN (Adaptive Synthetic Sampling):**



Advantages:

- Addresses the limitation of SMOTE by adapting the generation of synthetic samples based on the local density of minority samples.

- Places more emphasis on generating samples in regions that are harder to learn.

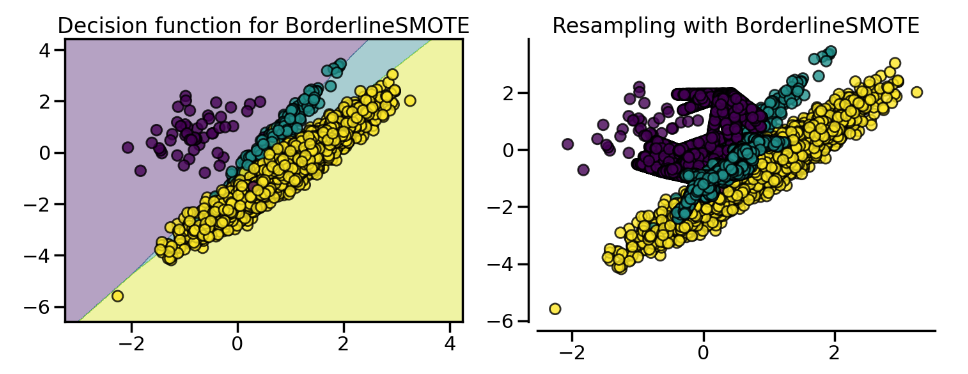
- Can handle datasets with multiple minority classes effectively.

Disadvantages:

- The synthetic samples generated can still be noisy, especially if the local density estimation is not accurate.

- Can be computationally expensive compared to SMOTE due to the additional density estimation step.

* + 1. **Borderline SMOTE:**



Advantages:

- Focuses on the samples near the decision boundary, which are considered more difficult to classify.

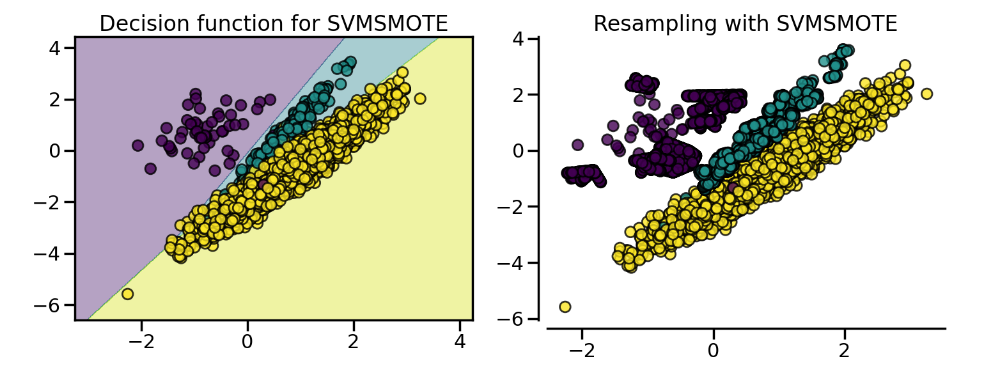
- Generates synthetic samples only in the vicinity of the decision boundary, reducing the chances of creating noisy samples in the majority class region.

Disadvantages:

- May not work well if the decision boundary between classes is not well-defined or if the minority samples are scattered throughout the feature space.

- Requires careful parameter tuning to determine the appropriate distance threshold for identifying borderline samples.

* + 1. **SVMSMOTE (Support Vector Machine Synthetic Minority Over-sampling Technique):**



Advantages:

- Uses support vector machines (SVMs) to identify the potential support vectors in the minority class.

- Generates synthetic samples by interpolating between these support vectors.

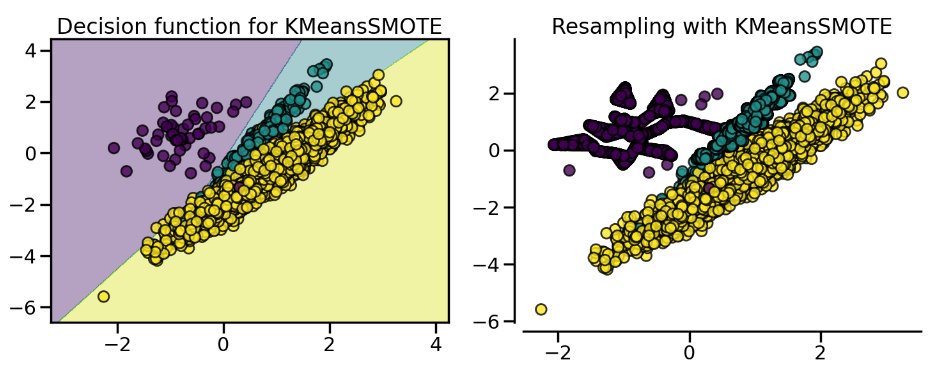
- Can effectively handle datasets with complex decision boundaries.

Disadvantages:

- Can be computationally expensive due to the involvement of SVMs.

- May not work well if the minority samples are not well separated or if the decision boundary is highly non-linear.

* + 1. **K means SMOTE**:



Advantages:

- Combines the concepts of K-means clustering and SMOTE to generate synthetic samples.

- Generates synthetic samples in regions where the minority class is densely located, avoiding oversampling in sparse regions.

- Can handle datasets with overlapping classes more effectively.

Disadvantages:

- Relies on the assumption that the K-means clustering accurately captures the underlying data distribution.

- May not work well if the clusters formed by K-means are not representative of the minority class.

5. *Splitting Data*: The dataset is typically divided into separate training, and testing sets. The training set is used to train the machine learning model, and the testing set is used to evaluate the final model's performance on unseen data. The data was split as, 80 % for training and 20% for testing.

By performing these preprocessing steps, the data is transformed into a format that is more suitable for training machine learning models, improving the model's accuracy, efficiency, and generalization capability

**6.1.3. MODELS**

After having performed data preprocessing, we apply classification models, namely

Logistic Regression

Logistic regression is a statistical model used for binary classification tasks, where the goal is to predict the probability of an instance belonging to a particular class.

It works by fitting a logistic function, also known as the sigmoid function, to the input data. The logistic function maps any real-valued input to a value between 0 and 1, representing the estimated probability of the positive class.

The logistic regression algorithm calculates the coefficients (weights) for the input features that best fit the data by minimizing the error between the predicted probabilities and the actual class labels.

Once the model is trained, it can make predictions by applying the logistic function to new instances and classifying them based on a chosen threshold, typically 0.5. If the predicted probability is above the threshold, the instance is classified as the positive class; otherwise, it is classified as the negative class.

Decision Tree:

* Decision tree is a supervised machine-learning algorithm that is commonly used for classification and regression tasks.
* It is a type of predictive modeling algorithm that makes predictions by learning simple decision rules from the input features of the training data(by recursively splitting training data into subsets based on input features).

Random Forest

* Random Forest is an ensemble learning algorithm that is commonly used for classification and regression tasks in machine learning.
* Random Forest algorithm is based on probability.
* To make a prediction on new data, the random forest algorithm first runs each data point through each of the individual decision trees in the forest, and then takes the majority vote of the predicted class (in classification tasks) or the average prediction (in regression tasks).

Support Vector Machine

* Support Vector Machines (SVMs) are a type of supervised learning algorithm used for classification and regression tasks. The basic idea behind SVMs is to find a hyperplane or a set of hyperplanes in a high-dimensional space that can best separate the data into different classes.
* In the case of classification, SVMs aim to find a decision boundary that maximizes the margin between the classes.

K-Nearest Neighbor

* KNN is a simple yet effective algorithm that does not make any assumptions about the underlying distribution of the data.
* The algorithm works by calculating the distance between the test example and each training example in the feature space. K-Nearest Neighbors (KNN) is a type of supervised machine learning algorithm that is used for both classification and regression tasks.
* The value of K is a hyper parameter that needs to be tuned for each specific problem, and a larger value of K will result in a smoother decision boundary but may also introduce more bias.

Naïve Bayes

* Naive Bayes is a type of probabilistic machine learning algorithm that is commonly used for classification tasks.
* To train a Naive Bayes classifier, the algorithm first calculates the prior probabilities of each class based on the training data.
* Then, for each feature, it calculates the conditional probability of that feature given each class. Finally, it applies Bayes' theorem to calculate the posterior probabilities of each class given the evidence (the features of the instance).

Hyperparameter Tuning

* Hyper parameter tuning is the process of selecting the best values for the parameters that control the behavior of a machine learning algorithm.
* It involves exploring different combinations of hyperparameters and evaluating their impact on the model's performance.
* The goal is to find the optimal configuration that maximizes the model's accuracy or other desired metrics. It is a critical step in optimizing the performance of machine learning models.

**6.1.3. EVALUATION**

Evaluation is a critical step in any deep learning project as it helps to assess the performance of the model and identify areas for improvement. There are several evaluation metrics that are commonly used in deep learning, depending on the task and the type of model being evaluated.

1. Accuracy: Accuracy is the most commonly used evaluation metric for classification tasks. It measures the proportion of correct predictions made by the model.

2. Precision and Recall: Precision and Recall are evaluation metrics used in binary classification tasks. Precision measures the proportion of true positives among all positive predictions, while recall measures the proportion of true positives among all actual positive cases.

3. F1 Score: The F1 score is a weighted average of precision and recall that combines the two metrics into a single score. It is often used in binary classification tasks where precision and recall are both important.

**6.2. DETAILED ANALYSIS WITH HYBRID IMBALANCED TECHNIQUE**

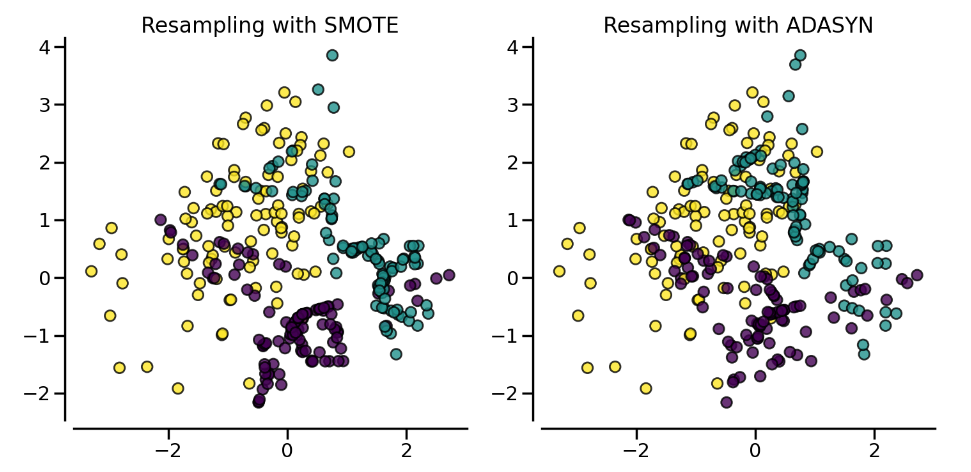
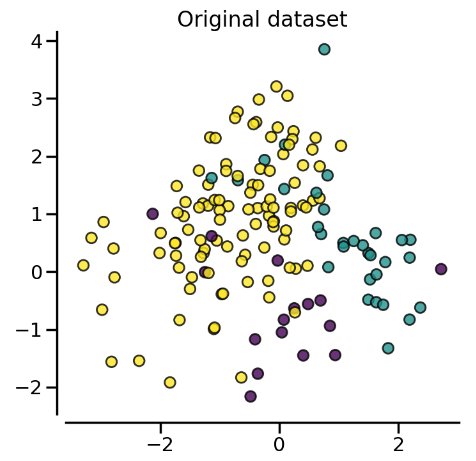


Fig 6.1 Differences in SMOTE and ADAYSYN data

The novel approach that this project experiment is to collect the newly generated samples from the best-performing oversampling techniques which were found out to be SMOTE and ADASYN. Our class imbalance has over 115 data in majority class while only 31 data in the minority class. Therefore the newly generated samples from SMOTE and ADASYN were collected separately and a new subset of minority was created. Then we randomly shuffle the data and get 115 data samples from them. Now we use the 115 data from majority class and 115 from minority class to predict Parkinson’s disease using classification Models.

**PROPOSED METHODOLOGY**

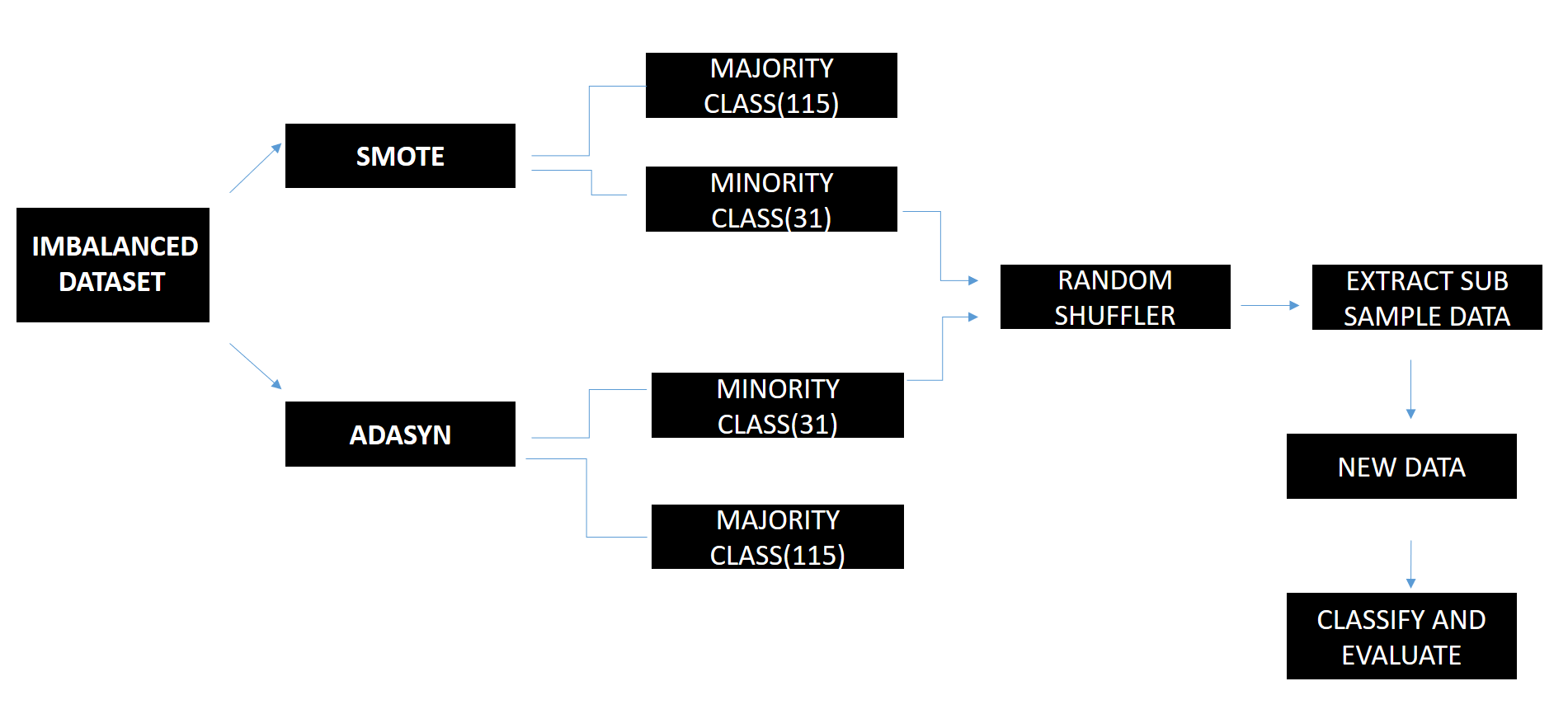


Fig 6.2 Proposed Methodology

According to the results, of earlier classification models, throughout the different models, SVM and KNN have given better accuracy overall for all the different oversampling techniques. Then performing Hyper-parameter tuning for these two models, the SMOTE and ADASYN techniques provided the highest accuracy of 93 and 95% respectively. Therefore the proposed methodology includes getting the newly generated samples from SMOTE and ADASYN and considering that as our new sample data, we perform a classification model. It was found that this gave us results of 96% for SVM and almost 94% for Random Forest, which is promising compared to previous results.

**CHAPTER 7**

**RESULTS AND DISCUSSIONS**

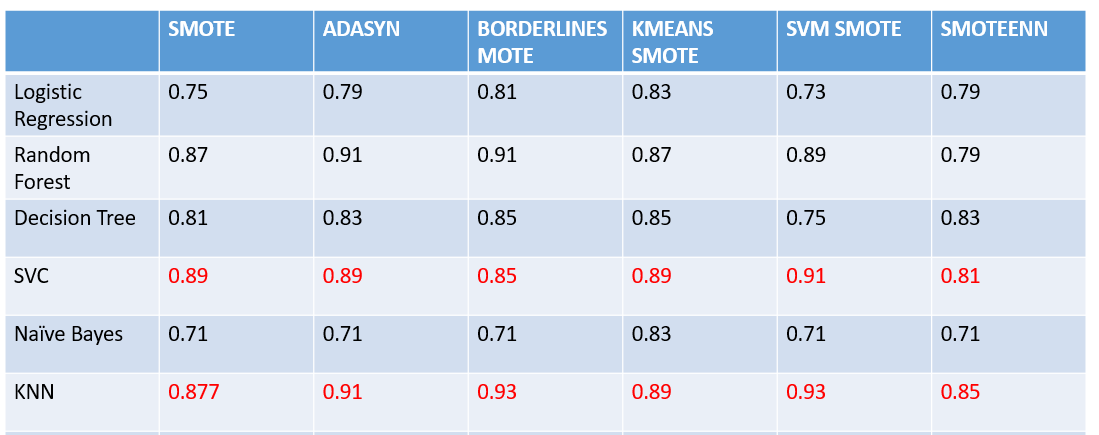


Table 7.1 Results after using different oversampling techniques

With the different oversampling techniques ,different classification models was used. When comparing the accuracies of the different oversampling techniques with the different classification model,it was observed that throughout the different oversampling techniques SVC and KNN has performed well.We can notice that the accuracies lie between 85 to 93 whereas for other models the accuracy wavers between 70-90.

Therefore ,SVC and KNN models were taken up for hyperparameter tuning.The method of hyperparameter chosen for this dataset was gridsearchCV as the no of datas are not very large ,so it will be computationally possible to check for the different parameter combination rather than randomsearch.

Following are the results from hyperparameter tuning.

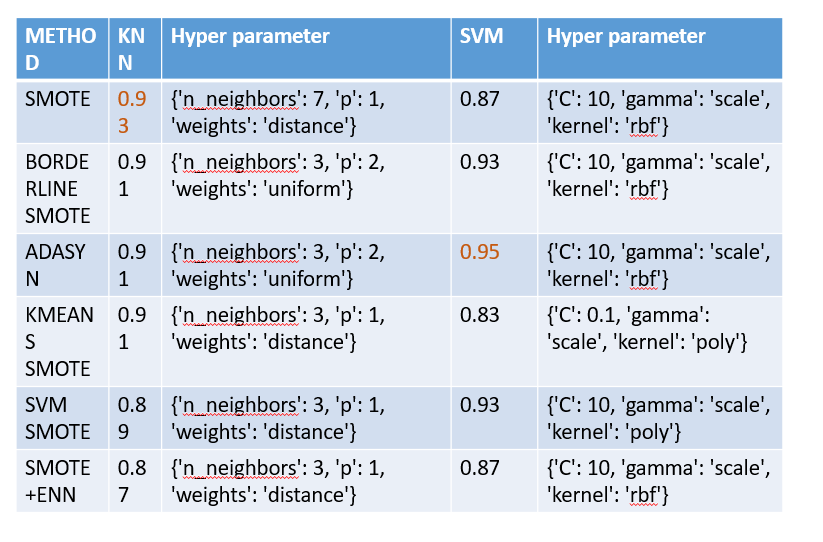


Table 7.2 Results from Hyper Parameter Tuning

Now, the hybrid approach was done and new data was formed,and classification model was built on it. The Results obtained were as follows

|  |  |  |
| --- | --- | --- |
| SVM | RANDOM FOREST | KNN |
| 96% | 93.5 | 94% |

**CHAPTER 8**

**CONCLUSION**

* In conclusion, the classification of Parkinson’s using machine learning shows promising results.
* With both datasets, we can observe that higher results were obtained with the Support Vector machine.
* With the hybrid approach, where the newly generated samples were collected and stored and made into a new dataset, the accuracy obtained was close to 96.5%.
* However, further research is needed to improve the specificity and sensitivity of these classification methods, as well as to evaluate their practicality and applicability in clinical settings.

**APPENDICIES**

**SAMPLE CODE:**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from scipy.stats import norm

from scipy import stats

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification\_report

data=pd.read\_csv('/content/drive/MyDrive/Parkinsson disease.csv')

data.head()

data=data.drop(['name'],axis=1)

data=data.drop\_duplicates()

data.info()

corr=data.corr()

cor\_target = abs(corr["status"])

#Selecting highly correlated features

relevant\_features = cor\_target[cor\_target>0.5]

relevant\_features

import seaborn as sns

sns.countplot(x="status", data=data)

data.corr()['status'][:-1].sort\_values().plot(kind='bar')

import matplotlib.pyplot as plt

%matplotlib inline

plt.figure(figsize=(30,30))

sns.heatmap(data.corr(), annot = True, cmap= "GnBu")

data.skew().sort\_values(ascending=False)

data[['NHR']]=np.sqrt(data[['NHR']])

data[['Jitter:DDP']]=np.sqrt(data[['Jitter:DDP']])

data[['MDVP:RAP']]=np.sqrt(data[['MDVP:RAP']])

data[['MDVP:Jitter(%)']]=np.sqrt(data[['MDVP:Jitter(%)']])

data[['MDVP:PPQ']]=np.sqrt(data[['MDVP:PPQ']])

data[['MDVP:Jitter(Abs)']]=np.sqrt(data[['MDVP:Jitter(Abs)']])

data[['MDVP:APQ']]=np.sqrt(data[['MDVP:APQ']])

data[['MDVP:Fhi(Hz)']]=np.sqrt(data[['MDVP:Fhi(Hz)']])

data.skew().sort\_values(ascending=False)

data.groupby('status').mean()

scaler = MinMaxScaler()

scaled = scaler.fit\_transform(data)

dataset = pd.DataFrame(scaled,columns = data.columns)

X = data.drop(['status'], axis=1)

y = data['status']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,test\_size=0.25, random\_state=45)

from imblearn.over\_sampling import SMOTE

print("Before OverSampling, counts of label '1': {}".format(sum(y\_train==1)))

print("Before OverSampling, counts of label '0': {} \n".format(sum(y\_train==0)))

sm = SMOTE(random\_state=2)

X\_train\_res, y\_train\_res = sm.fit\_resample(X\_train, y\_train.ravel())

print("After OverSampling, counts of label '1': {}".format(sum(y\_train\_res==1)))

print("After OverSampling, counts of label '0': {}".format(sum(y\_train\_res==0)))

from imblearn. over\_sampling import ADASYN

counter=Counter(y\_train)

print ('Before' ,counter)

# oversampling the train dataset using ADASYN

ada = ADASYN (random\_state=138)

X\_train\_ada, y\_train\_ada =ada.fit\_resample(X\_train, y\_train)

counter = Counter(y\_train\_ada)

print("After", counter)

from imblearn.over\_sampling import BorderlineSMOTE

from imblearn.over\_sampling import SMOTENC

smote = BorderlineSMOTE(random\_state=42)

X\_train\_resam, y\_train\_resam = smote.fit\_resample(X\_train, y\_train)

from imblearn.over\_sampling import KMeansSMOTE

kmeans\_smote = KMeansSMOTE(random\_state=42)

X\_train\_kmeans, y\_train\_kmeans = kmeans\_smote.fit\_resample(X\_train, y\_train)

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import GaussianNB

from sklearn.neighbors import KNeighborsClassifier

from sklearn.cluster import KMeans

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

classifiers = {

    'SVC': SVC(),

    'Decision Tree': DecisionTreeClassifier(),

    'Logistic Regression': LogisticRegression(),

    'Naive Bayes': GaussianNB(),

    'KNN': KNeighborsClassifier(),

    'KMeans': KMeans(n\_clusters=2),

    'Random Forest': RandomForestClassifier(),

}

for name, classifier in classifiers.items():

    classifier.fit(X\_train\_res, y\_train\_res)

    y\_pred = classifier.predict(X\_test)

    accuracy = accuracy\_score(y\_test, y\_pred)

    report = classification\_report(y\_test, y\_pred)

    print(f'{name} classification report:')

    print(report)

    print('-' \* 50)

#HYBRID

#shuffle data

# Shuffle the DataFrame

shuffled\_data = dfminority.sample(frac=1, random\_state=42)

# Reset the index of the shuffled DataFrame

shuffled\_data = shuffled\_data.reset\_index(drop=True)

# Select 115 random data points from the shuffled data

random\_samples = shuffled\_data.sample(n=115, random\_state=42)

# Filter the data for label '1' after applying SMOTE

X\_train\_res\_label\_1 = X\_train\_res[y\_train\_res == 1]

y\_train\_res\_label\_1 = y\_train\_res[y\_train\_res == 1]

# Append df2 to df1

appended\_df = pd.concat([random\_samples, df\_majority], axis=0)

appended\_df

new\_hybrid =appended\_df.sample(frac=1, random\_state=442)

# Reset the index of the shuffled DataFrame

new\_hybrid = new\_hybrid.reset\_index(drop=True)

svc = SVC()

# Perform grid search with 5-fold cross-validation

grid\_search = GridSearchCV(svc, param\_grid\_svm, cv=5)

# Fit the grid search to the training data

grid\_search.fit(X\_new, y\_new)

# Print the best parameters and best score

print("Best Parameters: ", grid\_search.best\_params\_)

print("Best Score: ", grid\_search.best\_score\_)

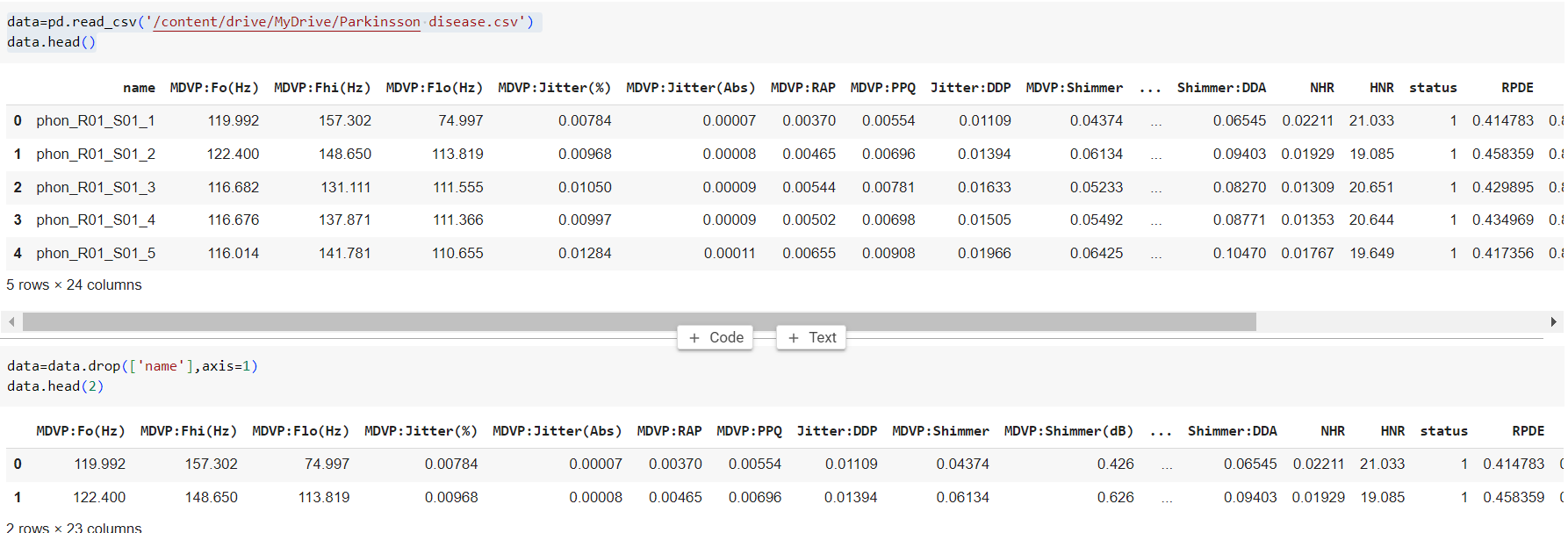
# Evaluate the best model on the test set

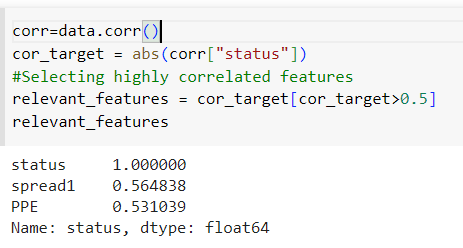
best\_model = grid\_search.best\_estimator\_

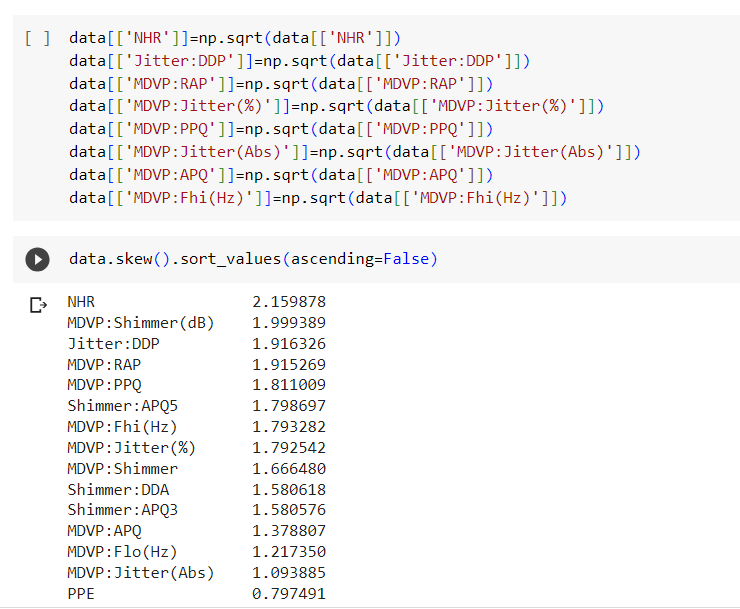
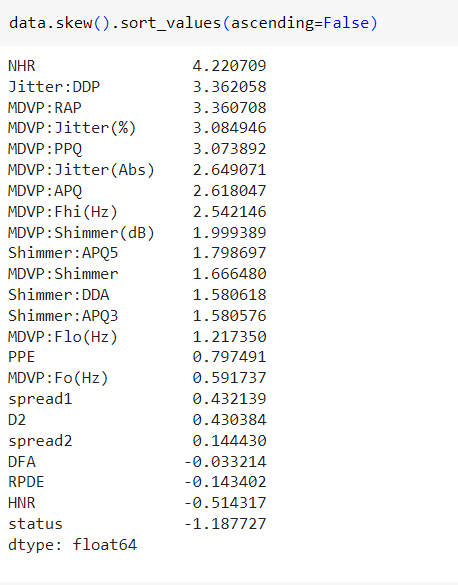
accuracy = best\_model.score(X\_test, y\_test)

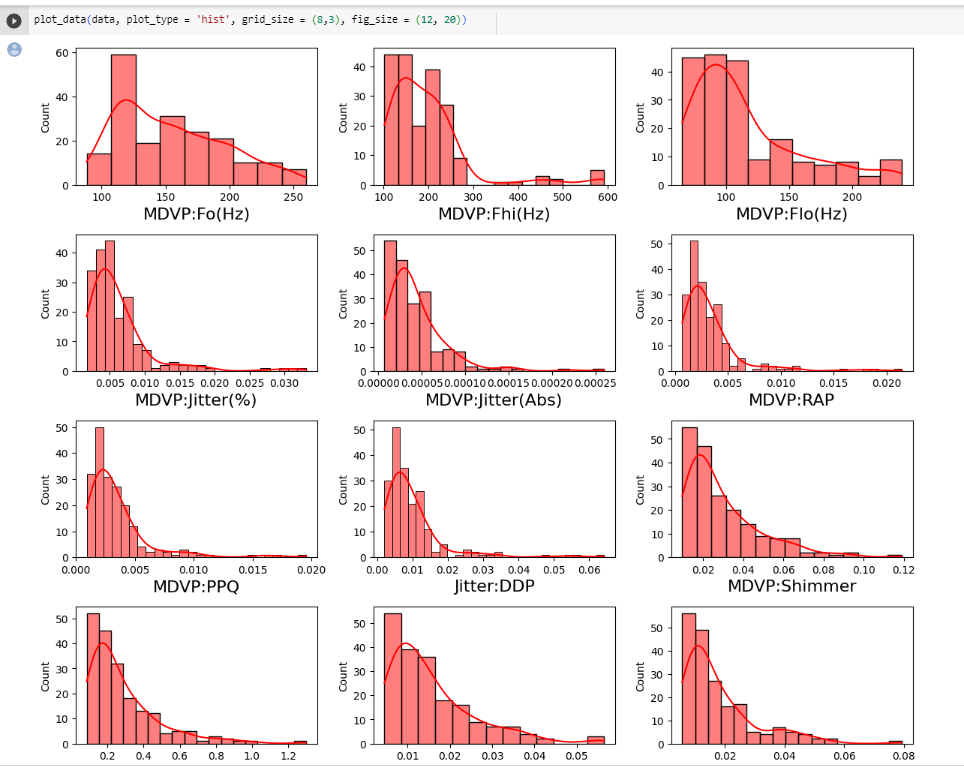
print("Test Accuracy: ", accuracy)

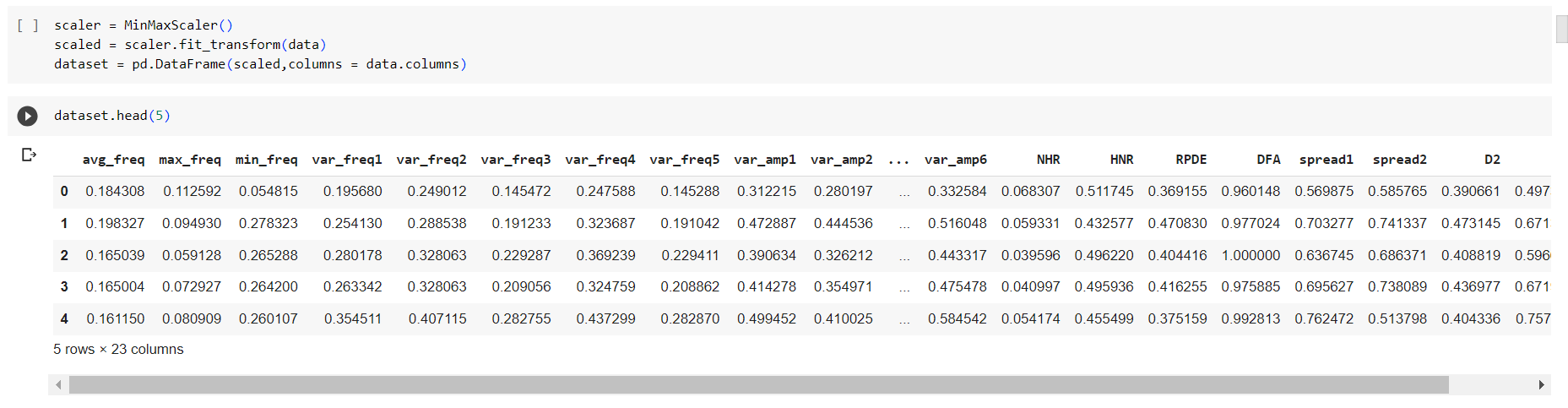
**SCREENSHOTS**

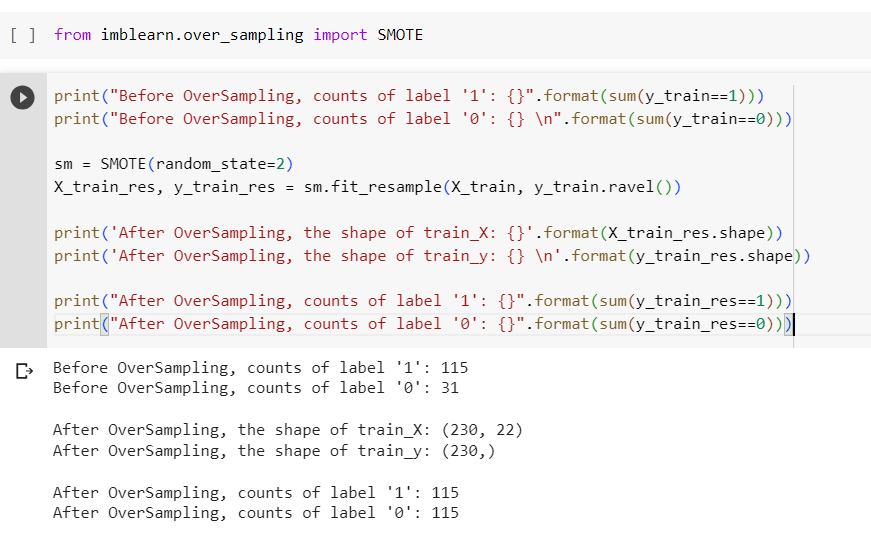
****

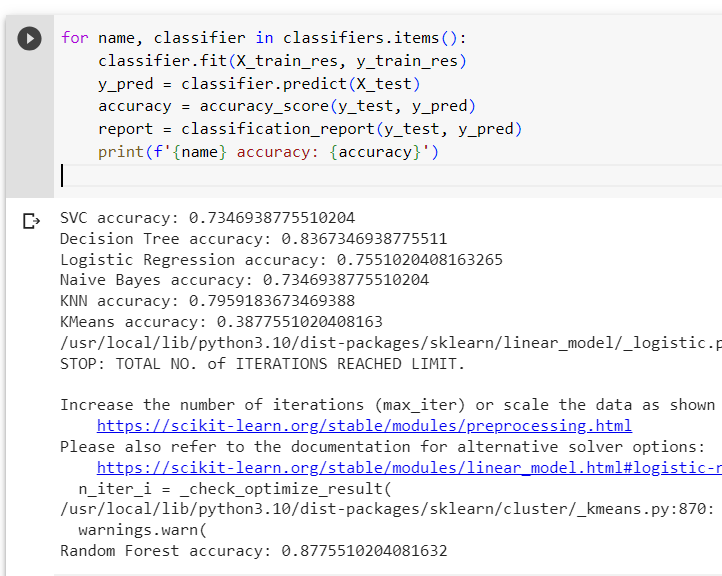


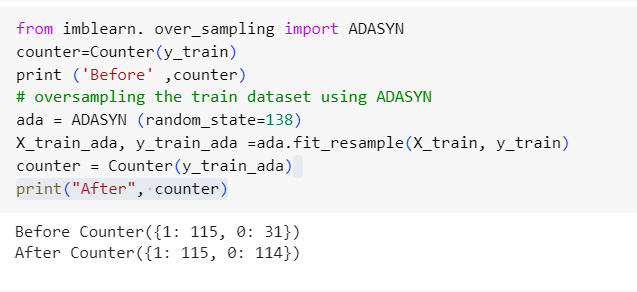


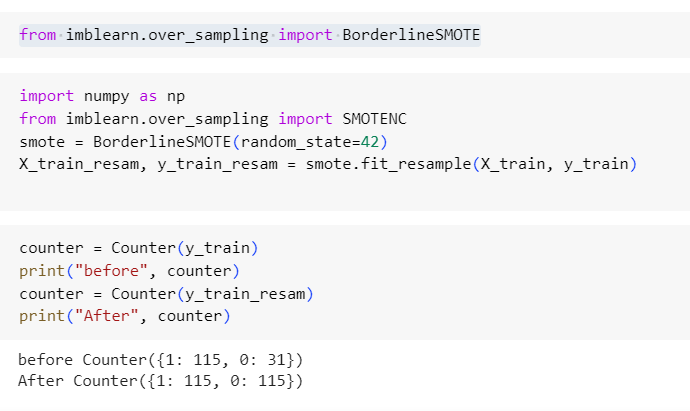


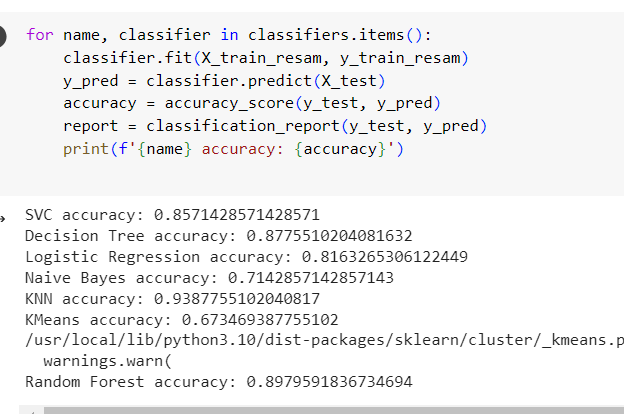


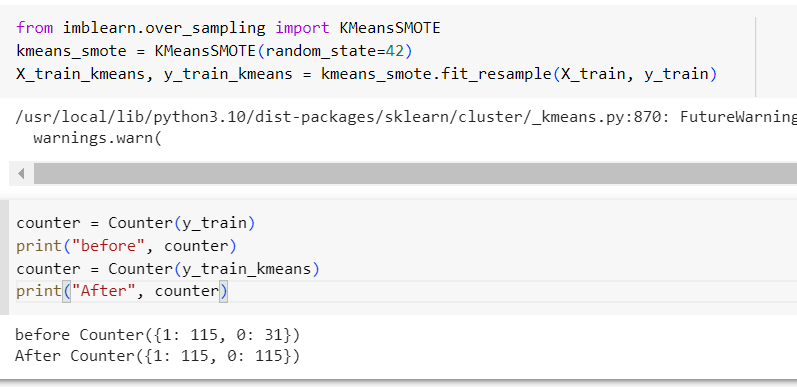


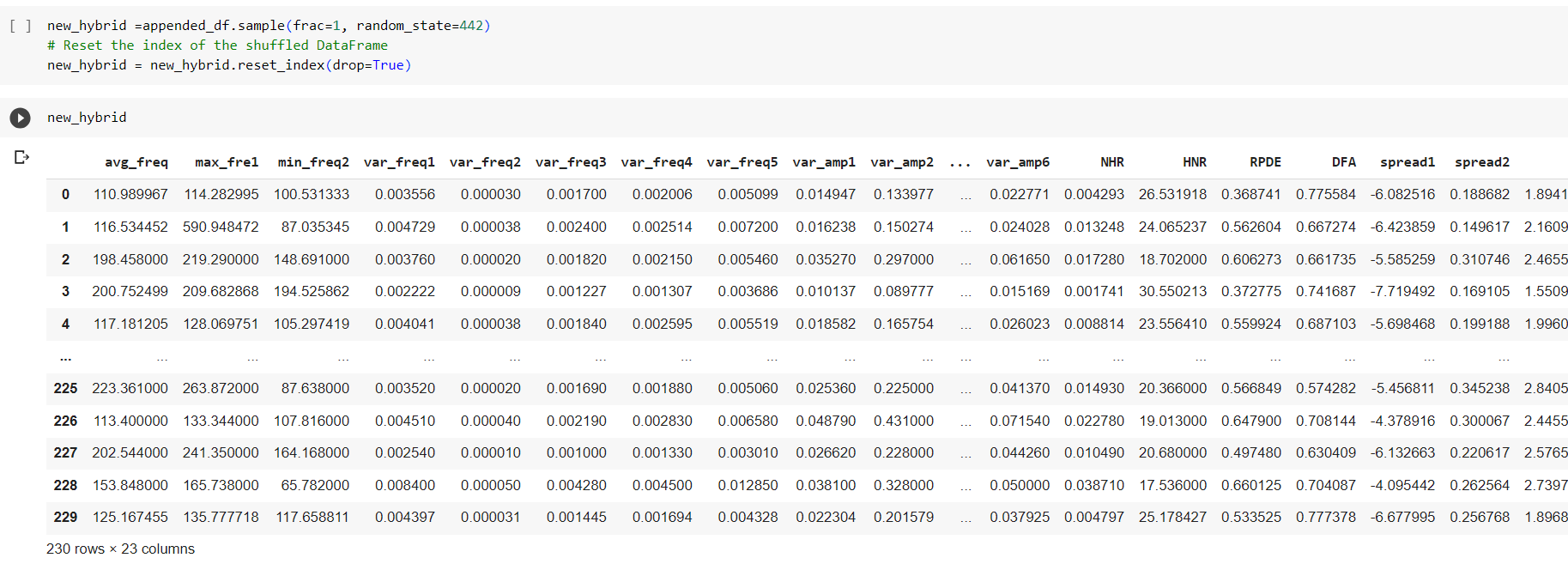


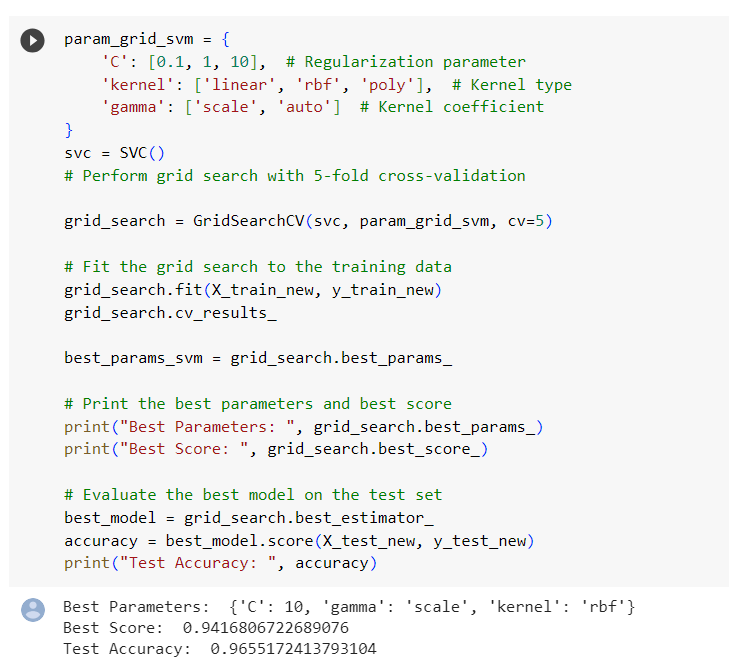












**REFERENCES**

**Journal References:**

[1] K. Polat, "A Hybrid Approach to Parkinson Disease Classification Using Speech Signal: The Combination of SMOTE and Random Forests," 2019

[2] S. Sharanyaa, P. N. Renjith and K. Ramesh, "Classification of Parkinson's Disease using Speech Attributes with Parametric and Nonparametric Machine Learning Techniques,", 2020

[3] [Anik Pramanik](https://link.springer.com/chapter/10.1007/978-981-16-0586-4_36#auth-Anik-Pramanik) & [Amlan Sarker](https://link.springer.com/chapter/10.1007/978-981-16-0586-4_36" \l "auth-Amlan-Sarker) , “Parkinson’s Disease Detection from Voice and Speech Data Using Machine Learning”, 2021

[4] Arvind Kumar Tiwari, “Machine learning based approaches for prediction of Parkinson’s disease”,

[5]Wroge TJ, Özkanca Y, Demiroglu C, Si D, Atkins DC, Ghomi RH. “Parkinson’s disease diagnosis using machine learning and voice”, 2018

**Web References:**

1. SMOTE Variations:<https://www.kdnuggets.com/2023/01/7-smote-variations-oversampling.html>
2. Impact of Over sampling in the dataset: [https://imbalanced-learn.org/stable/over\_sampling.html#smote-variants](https://imbalanced-learn.org/stable/over_sampling.html)
3. Parkinson’s Disease:<https://www.ninds.nih.gov/health-information/disorders/parkinsons-disease>
4. Correlation Coefficients and their impact: [https://www.analyticsvidhya.com/blog/2021/03/comparison-of-pearson-and-spearman-correlation-coefficients/#:~:text=Pearson%20and%20Spearman%20correlation%20coefficients,coefficient%20evaluates%20the%20monotonic%20relationship](https://www.analyticsvidhya.com/blog/2021/03/comparison-of-pearson-and-spearman-correlation-coefficients/)

**CHAPTER 11**

**WORKLOG**

|  |  |  |
| --- | --- | --- |
| **Day** | **Date** | **Task Done** |
| Day 1 | 26/04/2023 | Project domain discussion |
| Day 2 | 27/02/2023 | Domain Research |
| Day 3 | 28/04/2023 | Domain Research |
| Day 4 | 29/04/2023 | Domain Research |
| Day 5 | 2/05/2023 | Project title finalization |
| Day 6 | 3/05/2023 | Collecting Study materials |
| Day 7 | 4/05/2023 | Collecting Study materials |
| Day 8 | 5/05/2023 | Going through research journals |
| Day 9 | 6/05/2023 | Going through research journals |
| Day 10 | 8/05/2023 | Going through research journals |
| Day 11 | 9/05/2023 | Discussion project architecture |
| Day 12 | 10/05/2023 | Discussion project architecture |
| Day 13 | 11/05/2023 | Collecting Parkinson’s Dataset |
| Day 14 | 12/05/2023 | Collecting Parkinson’s Dataset |
| Day 15 | 13/05/2023 | Analyzing the Dataset available |
| Day 16 | 15/05/2023 | Performing EDA on the dataset |
| Day 17 | 16/05/2023 | Pre-processing the dataset |
| Day 18 | 17/05/2023 | Pre-processing the dataset |
| Day 19 | 18/05/2023 | Discussion about the Imbalancer Problem |
| Day 20 | 19/05/2023 | Discussion about the Imbalancer Problem |
| Day 21 | 20/05/2023 | Finalizing the types of Over Sampling techniques |
| Day 22 | 22/05/2023 | Finalizing the types of Over Sampling techniques |
| Day 23 | 23/05/2023 | Learning about the Oversampling techniques |
| Day 24 | 24/05/2023 | Learning about the Oversampling techniques |
| Day 25 | 25/05/2023 | Learning about the Oversampling techniques |
| Day 26 | 26/05/2023 | Learning about the Oversampling techniques |
| Day 27 | 27/05/2023 | Learning about the Oversampling techniques |
| Day 28 | 29/05/2023 | Discussion on the Classification models |
| Day 29 | 30/05/2023 | Discussion on the Classification models |
| Day 30 | 31/05/2023 | Discussion on the Classification models |
| Day 31 | 1/06/2023 | Experimenting models with OverSampling techniques |
| Day 32 | 2/06/2023 | Experimenting models with Over Sampling techniques |
| Day 33 | 3/06/2023 | Experimenting models with Over Sampling techniques |
| Day 34 | 5/06/2023 | Learning about Hyper Parameter Tuning |
| Day 35 | 6/06/2023 | Learning about Hyper Parameter Tuning |
| Day 36 | 7/06/2023 | Learning about Hyper Parameter Tuning |
| Day 37 | 8/06/2023 | Choosing the best Tuning method for the problem |
| Day 38 | 9/06/2023 | Choosing the best Tuning method for the problem |
| Day 39 | 10/06/2023 | Implementing about GridSearchCV |
| Day 40 | 12/06/2023 | Implementing about GridSearchCV |
| Day 41 | 13/06/2023 | Discussion on the Hybrid Approach |
| Day 42 | 14/06/2023 | Discussion on the Hybrid Approach |
| Day 43 | 15/06/2023 | Discussion on the Hybrid Approach |
| Day 44 | 16/06/2023 | Combining SMOTE and ADASYN techniques |
| Day 45 | 17/06/2023 | Combining SMOTE and ADASYN techniques |
| Day 46 | 19/06/2023 | Combining SMOTE and ADASYN techniques |
| Day 47 | 20/06/2023 | Arriving at different results |
| Day 48 | 21/06/2023 | Arriving at different results |
| Day 49 | 22/06/2023 | Working on PPT |
| Day 50 | 23/06/2023 | Working on Report |