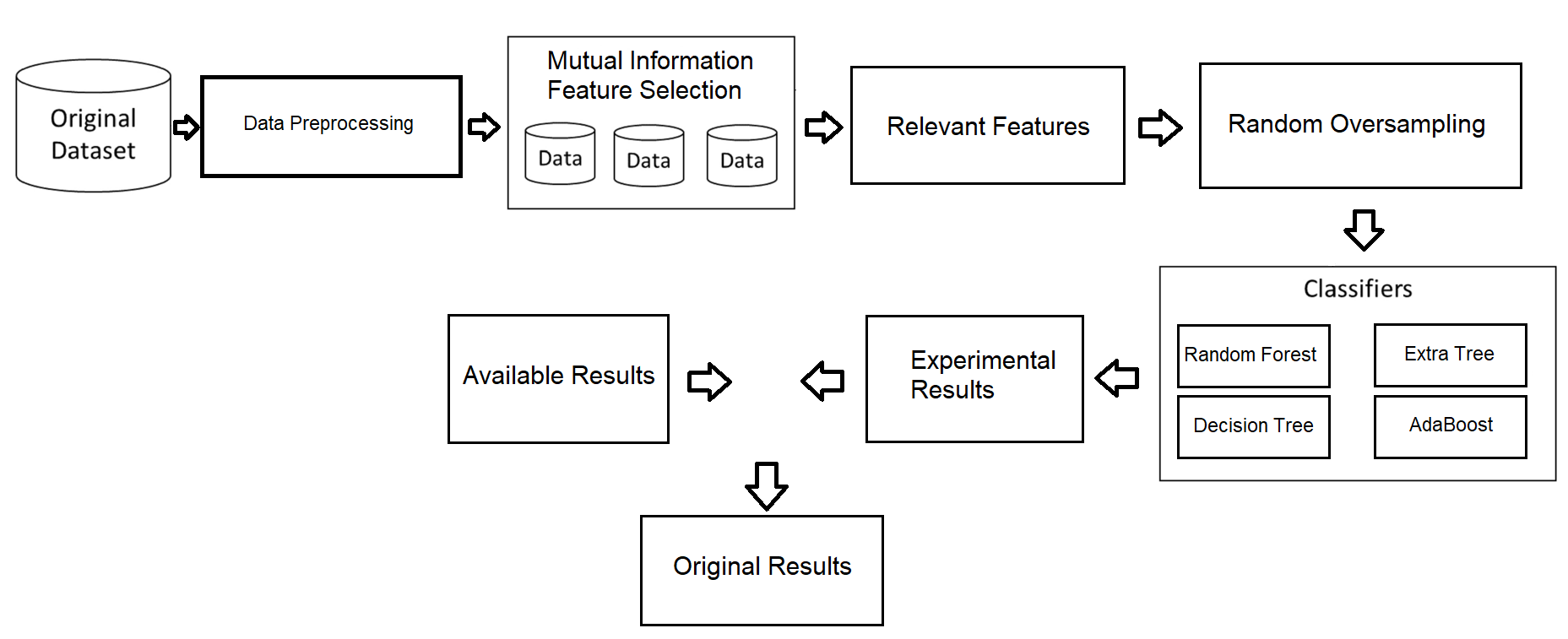
**Using Machine Learning to detect Parkinson’s disease**

**Methods used:**

**Mutual Information Feature Selection**

Effective feature engineering and the process of feature selection serve as the foundation for achieving increased accuracy in prediction and classification within any machine learning model. The dataset, comprising 197 instances, is designed to categorize patients as either positive or negative for Parkinson's. The features encompass various factors, such as gender, PPE, DFE, and several others. Mutual Information Feature Selection aims to identify and retain features that have high mutual information with the target variable (the variable to be predicted) while discarding those with lower information content. By selecting features with high mutual information, this technique helps improve the efficiency of machine learning models by focusing on the most relevant and informative features for making accurate predictions [1].



**Random Forest**

The Random Forest algorithm operates within the framework of supervised learning, serving as an integral part of a larger ensemble. This ensemble utilizes various basic models to achieve the most accurate prediction of loan default risk. Within the Random Forest classifier, multiple decision trees are employed, and the average accuracy of these trees contributes to enhancing the overall dataset accuracy. In contrast to relying on a single decision tree, Random Forest aggregates predictions from each tree and determines the final outcome based on the majority vote [2].

For a classification problem, the prediction (Y-hat-RF) is determined by a majority vote among all the individual decision trees in the Random Forest. Each tree votes for a class, and the class with the most votes become the predicted class.

ŶRF = argmaxj ( (Ŷi = j))

Where:

* (ŶRF) is the predicted class by the Random Forest model.
* *N-*trees​ is the total number of decision trees in the Random Forest.
* (Ŷi) is the predicted class by the *i*-th decision tree.
* argmax *j*​ returns the class *j* that maximizes the expression.

In regression as well as classification tasks, the Random Forest consolidates the predictions made by individual trees, harnessing their diversity to enhance generalization performance and robustness. The efficacy of the Random Forest algorithm is bolstered by the introduction of randomness in the training process of each tree, achieved through bootstrapped sampling and random feature selection. This deliberate injection of variability significantly contributes to the overall effectiveness of the Random Forest model.

**Decision Tree**

Within the realm of supervised learning, the Decision Tree belongs to the family of algorithms. It finds widespread application in tasks such as loan classification, prediction, and various problem-solving techniques. Tree-based learning algorithms are commonly employed to achieve high precision in prediction models utilizing supervised learning approaches. Decision trees are versatile, capable of handling both categorical and numerical data. They play a significant role in reaching conclusions or aiding decision-making processes [2].

The Accuracy of Decision trees can be represented by the following expression:

**Accuracy =**

In this expression, TP represents the “True Positives”, FP represents “False Positives”, TN represents “True Negatives” and FN represents “False Negatives”.

**AdaBoost**

Boosting transforms weak training into robust training by enhancing the predictions of each strategy instruction. AdaBoost, a contemporary type of boosting, employs an array structure for this purpose. The objective of boosting is to effectively train weak classifiers to rectify previous prediction errors. In this approach, the model initially tries to predict on the base dataset and then fits multiple iterations of the exact model on the same dataset. The weights of the items are adjusted during the training process based on the most recent prediction inaccuracies. Consequently, the resulting model focuses more on challenging instances [3].

**Extra Tree**

An extra tree acquires knowledge from a parent dataset by dividing significant data into multiple subgroups (child samples) and making individual predictions for each subgroup [3]. The ultimate prediction is derived from the summation of all subgroup predictions, employing averaging to improve prediction accuracy while mitigating overfitting. The extra tree methodology differs from other tree-based ensembles in its utilization of random split points from distinct subgroups, the ability to randomly separate these subgroups, and the capacity to construct trees using the entire learning sample.

The behavior of Extra Tree algorithm is represented by the following mathematical expression.

ŶET

Where:

* The (ŶET) is the predicted value of the classification by the Extra Tree model.
* () is the predicted value or class by the i-th tree.
* N-trees is the total number of the trees in the Extra Tree Ensemble.
* argmax(j) returns the class (j) that maximizes the expression.

The key distinction of Extra Trees lies in the construction process of each tree within the ensemble. Instead of using the conventional approach seen in traditional decision trees and Random Forests, Extra Trees introduces added randomness in two significant ways. Firstly, a random subset of features is selected for each tree. Secondly, random splitting thresholds are independently chosen for each feature within the selected subset. This unique combination of feature selection and threshold determination contributes to the distinctiveness of Extra Trees and sets it apart from other ensemble learning methods.

**Class Balancing Techniques**

Given the notable imbalance among the classes in all four outputs, it becomes crucial to employ specific class balancing techniques. This research incorporates one of the most important class balancing techniques to identify the most effective one for addressing this particular issue. In every instance, following the implementation of class balancing, the resulting dataset exhibits a class distribution with a ratio approximately equal to 50:50.

1. **Random Oversampling**

One fundamental class balancing methods is random oversampling, which involves addressing class imbalance by randomly selecting and duplicating samples from the minority class [4]. This process is designed to increase the representation of the minority class, aligning it with the number of samples present in the majority class. In essence, random oversampling amplifies the minority class instances, contributing to a more balanced dataset.

#### ADAptive SYNthetic Algorithm

#### ADASYN, short for ADAptive SYNthetic algorithm, is an oversampling technique that can be outlined through a series of steps [4]. The initial step involves computing the ratio between the count of minority and majority class instances. This ratio can be expressed as:

The fraction T is employed to initialize the ADASYN algorithm. Following this initialization, the calculation of the necessary synthetic data is ascertained by:

G = (Nmaj - Nmin)β

where 𝛽 represents the desired ratio after the data generation. In the neighborhood of each minority sample, a dominance of the majority sample (𝑟𝑖) must be defined as:

r1 =

Here, k denotes the count of nearest neighbors. The subsequent step involves establishing the quantity of synthetic samples, defined as:

G = G ∩ r1

where ∩𝑟𝑖 represents the normalization of 𝑟𝑖 using:

σ ∩ r1 =

Each synthetic sample (𝑠𝑖) from the minority class is subsequently determined by randomly selecting one of the K-nearest neighbors for each actual minority sample, and utilizing the:

S1 = χi + (χzi – χi)λ

Here, (𝑥𝑧𝑖−𝑥𝑖) denotes the difference vector, and 𝜆 represents a randomly generated number within the range of 0 to 1.

**Training Algorithms and Techniques**

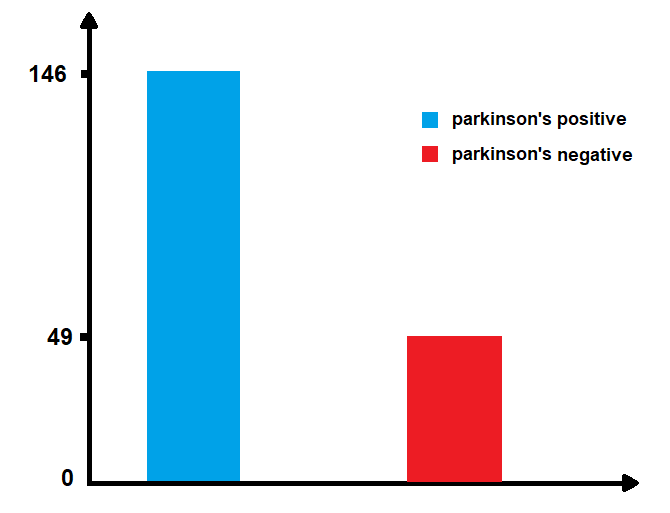
In the realm of machine learning, the term "classification" pertains to the predictive aspect of the field [5]. This challenge revolves around the categorization of independent variables (input data) into various groups, with the number of groupings ranging from two to numerous. Various strategies are employed to classify both structured and unstructured data. The term "classifier" denotes an algorithm designed to assign incoming data to a specific class or category. Three distinct classification types exist: binary classification, multi-label classification, and multi-class classification, with binary classification being the most straightforward form. One of the most effective classification algorithms is XGBoost.

**XGBoost**

Gradient Boosting (XGBoost) serves as the foundation for constructing this classifier, and in numerous competitions, the model derived from its application often emerges as a clear victor. Within the XGBoost algorithm, weights assume a crucial role. A decision tree is formulated, wherein specific independent factors are assigned weights and subsequently fed into the tree to predict outcomes. XG Boost offers several advantages, including the mitigation of overfitting, earning it the designation of a regularized boosting strategy. Additionally, XGBoost adeptly handles missing values in the data and incorporates a built-in cross-validation mechanism that operates at each step [5].

**Experimental Procedure**

Data exploration involves uncovering insights from data through the application of diverse visualization methods, constituting the preliminary phase in the data analysis process [5]. The pre-processing stage assumes significance in machine learning, as it elevates the data's quality, facilitating the extraction of pertinent knowledge from extensive datasets more effectively. Moreover, the data becomes sufficiently prepared for modeling purposes.



**Count of both classes**

The primary stage in the development or deployment of a machine learning model involves data pre-processing. This step entails cleaning and organizing data that may be incomplete, contain errors, possess missing values, or exhibit inconsistencies. Proceeding with this phase, an evaluation of the target characteristic reveals a significant imbalance, with the proportion of class 0 being notably lower than that of class 1. Consequently, the data is highly unbalanced, potentially leading to biased outcomes.

The histogram depicted in the figure illustrates the distribution count of each class category in the data. Observably, there is an imbalance in the data, highlighting the necessity of applying advanced balancing techniques to enhance overall equilibrium.

In the depicted figure, the data exhibits a significant imbalance, potentially resulting in biased outcomes and suboptimal model performance. This is evident as class 1 constitutes approximately 76 percent of the data distribution, making it the majority class. To address this imbalance, a sampling approach was applied, specifically employing random oversampling to balance the data format. This oversampling technique was specifically applied to class 0, as illustrated in the subsequent figure.

* **Feature Selection**: One instance of this is excluding independent variables and utilizing only pertinent data for model training. Consequently, data can be more efficiently condensed using this approach. In the case of such machine learning algorithms, the crucial characteristic is automatically selected, leading to improved efficiency in our model. While there are various methods to enhance the model, both data preparation and feature engineering play pivotal roles.
* **Data Pre-processing**: One instance of this is excluding independent variables and utilizing only pertinent data for model training. Consequently, data can be more efficiently condensed using this approach. In the case of such machine learning algorithms, the crucial characteristic is automatically selected, leading to improved efficiency in our model. While there are various methods to enhance the model, both data preparation and feature engineering play pivotal roles.
* **Removing Outliers**: The dataset exhibits clear signs of outliers, which could have arisen due to errors in data entry. Eliminating these outliers has the potential to enhance the performance of our predictive model [6].
* **Modeling**: From the dataset, a training dataset comprising 80% and a testing dataset comprising 20% are established. The training dataset is utilized to train a model, and its performance is subsequently evaluated using the testing dataset. Various classifiers, including the decision tree classifier, random forest classifier, AdaBoost, XTree, and XGBoost, are employed on the clustered dataset to gauge their respective performances. The evaluation of each classifier is conducted based on metrics such as accuracy and precision [6].

Algorithm modelling is depicted in a simplified architecture/workflow in a previous figure.

**Evaluation metrics of performance**

The assessment process serves to gauge the validity and precision of the constructed models [7], relying on the classification counters outlined below:

True Positive (TP) signifies the count of accurately labeled positive samples.

False Positive (FP) represents the instances of negative samples that were incorrectly classified.

True Negative (TN) corresponds to the number of positive samples incorrectly classified as negative.

False Negative (FN) indicates the number of negative samples that were incorrectly classified.

To appraise classifier outcomes, four evaluation measures are employed, each contingent on the aforementioned classification counters:

1. **Accuracy**: Accuracy: Utilized to ascertain the classifier's effectiveness in achieving the correct expected percentage, as depicted in the equation.
2. **Sensitivity**: also known as the True Positive (TP) rate, measures the percentage of positively classified samples with PD (Probability of Detection) that are accurately identified. Its calculation is depicted in Equation.
3. **Specificity**: also referred to as the True Negative (TN) rate (TNR), signifies the percentage of negative samples without any accurately classified Probability of Detection (PD)
4. **Precision:**  serves as a crucial metric for accuracy assessment, defining the percentage of cases identified as positive by the classifier in relation to the total predictive positive states

The effectiveness of learners within an unbalanced dataset cannot be adequately assessed with these experimental measures. Accuracy, as an evaluation metric, can be misleading for the majority class and often fails to predict parameters belonging to the minority class. In the context of classification problems with an uneven class distribution, a comprehensive evaluation is sought by considering various aspects that measure the classifier's ability to strike a balance between two classes, treating them as similar [30]. In our experiments, the Area Under the Curve (AUC) and Geometric Mean (G-mean) are employed because they demonstrate robustness in handling unbalanced data distribution.

G-mean, a function utilized to discern the optimal classification between two classes, offers the advantage of minimizing the negative category as false-positive results increase. Additionally, the arithmetic mean holds significance for determining sensitivity and specificity [34]. Sensitivity is defined as (TP / (TP + FN)), and specificity as (TN / (TN + FP)). Consequently, G-mean is represented by the following Equation.

G – mean =

AUC is specifically applied in binary classification to identify the optimal model for category prediction. The AUC score involves the use of a threshold to calculate the ratio of the True Positive (TP) rate to the False Positive (FP) rate.

**Table of Features**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Attributes** | **Meaning** | **Gain** |
| 1 | MDVP: Fo (Hz) | Multi-dimensional voice program fundamental frequency | 0.412 |
| 2 | MDVP: APQ | Multi-dimensional amplitude perturbation | 0.267 |
| 3 | MDVP:Jitter(Abs) | Variation measurement in vocal frequency | 0.3125 |
| 4 | PPE | Fundamental frequency measure by pitch | 0.2343 |
| 5 | NHR | Ratio of noise in the voice | 0.1089 |
| 6 | MDVP: RAP | Ratio of noise in amplitude | 0.1203 |
| 7 | DFA | Detrended fluctuation analysis | 0.8756 |
| 8 | Spread 1 | Non-linear measure 1 in fundamental frequency | 0.2143 |
| 9 | Spread 2 | Non-linear measure 2 in Fundamental frequency | 0.2312 |
| 10 | Class | No disease, disease | 0.2032 |
| 11 | PPQ | Period perturbation quotient | 0.1865 |
| 12 | Shimmer | Variance measures in amplitude | 0.2645 |
| 13 | D2 | Measure dynamic complexity | 0.0986 |
| 14 | MDVP: Fhi (Hz) | Fundamental frequency—maximum | 0.2165 |
| 15 | Shimmer APQ3 | 3 Measures in amplitude | 0.1805 |
| 16 | Shimmer DDA | Difference divided by amplitude | 0.1803 |
| 17 | HNR | Ratio of noise in total component | 0.1088 |
| 18 | Shimmer dB | Measures in decibels | 0.1608 |
| 19 | Shimmer APQ5 | 5 measures in amplitude | 0.1607 |
| 20 | Status | Status of healthy subjects | 0.1078 |
| 21 | RPDE | Recurrence period density entropy | 0.0967 |
| 22 | RAP | Relative average perturbation | 0.1943 |
| 23 | DDP | Difference divided by average period | 0.1765 |

Table 1 presents the dataset characteristics, specifically highlighting the vocal features of the subjects with Parkinson's disease (PD). In this study, three sets of attributes are employed, each assessed using the information gain (IG) value as a metric for attribute selection. The first set comprises 11 attributes, the second set comprises 15 attributes, and the third set comprises 18 attributes, with corresponding IG values of 0.13, 0.14, and 0.14, respectively [8].

# References

|  |  |
| --- | --- |
| [1] | D. o. C. S. a. E. S. o. C. S. I. o. S. a. T. K. C. 6. I. S. Iniyan & R. Jebakumar, "Mutual Information Feature Selection (MIFS) Based Crop Yield Prediction on Corn and Soybean Crops Using Multilayer Stacked Ensemble Regression (MSER)," *Springer Link,* vol. 126, pp. 1935-1964, 2021. |
| [2] | M. D. S. S. Asit Subudhi, "Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier," *Science Direct,* vol. 40, pp. 277-289, 2020. |
| [3] | R. M. O. A. E. I. A. Hayder Mohammedqassem, "Diagnosing Coronary Artery Disease on the Basis of Hard Ensemble Voting Optimization," *MDPI,* 2022. |
| [4] | A. L. N. A. I. L. Matko Glucina, "Cervical Cancer Diagnostics Using Machine Learning Algorithms and Class Balancing Techniques," *MDPI,* 2023. |
| [5] | A. V. M. R. K. R. A. M. S. S. A. Palak Gupta, "Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Technique," *Science Direct,* pp. 2575-2584, 2023. |
| [6] | P. P. T. G. P. L. M. Chintan M. Bhatt, "Effective Heart Disease Prediction Using Machine," *MDPI,* 2023. |
| [7] | O. A. ,. M. A. A. ,. M. N. A. S. A. M. A. Hayder Mohammed Qasim, "Hybrid Feature Selection Framework for the Parkinson," *MDPI,* 2021. |
| [8] | G. A. &. M. A. G. K. Kamalakannan, "Performance analysis of attributes selection and discretization of Parkinson’s disease dataset using machine learning techniques: a comprehensive approach," *Springer Link,* vol. 14, pp. 1523-1529, 2023. |