PROJECT REPORT

"EXPLAINABLE AI IN CLASSIFYING NUERODISORDER"

Under the supervision of:

Prof. Manjari Gupta

Under the guidance of:

Mr. Bethany Gosala

DST Centre for Interdisciplinary Mathematical Sciences

Name-Chhaya Ojha

23419CAS010

M.Sc in computational science and applications (Specialization: data Science)

Explainable AI in classifying Nuerodisorder

Chhaya Ojha

DST Centre for Interdisciplinary Mathematical Sciences, Banaras Hindu University, Varanasi, India-221005

Abstract.

The accurate and interpretable classification of neurodisorders poses a critical challenge in the field of medical diagnostics. Conventional machine learning models often achieve high predictive performance but lack transparency, limiting their adoption in clinical settings. This study employs Explainable Artificial Intelligence (XAI) techniques to enhance the interpretability of a binary classification model designed to distinguish between healthy individuals and schizophrenic patients.

This project explores the application of Explainable Artificial Intelligence (XAI) in classifying neurodisorders using machine learning models. The study employs logistic regression and random forest algorithms to develop predictive models for accurately identifying neurodisorders based on EEG data. To ensure transparency and interpretability, the XAI techniques LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) are utilized to explain model predictions.

The results demonstrate the efficacy of integrating XAI with traditional machine learning models in achieving interpretable predictions, providing a pathway for more trustworthy and actionable neurodiagnostic tools.

Keywords: Explainable AI SHAP(Shaply additive values),LIME,eli5,PDP,Integrated gradient,EEG

1.INTRODUCTION

1.1 Schizophrenia

Schizophrenia is a complex and chronic mental disorder that profoundly impacts individuals' lives, characterized by a spectrum of cognitive, behavioral, and emotional dysfunctions. Its main symptoms include delusions, hallucinations, and cognitive impairment, which significantly affect the quality of human life. The disorder is marked by abnormalities in brain structure and function, yet its exact cause remains elusive. The complexity and variability of schizophrenia symptoms make its diagnosis challenging, often relying on the subjectivity and experience of clinicians. This subjectivity underscores the need for more objective and accurate diagnostic methods.

The impact of schizophrenia extends beyond the affected individuals to their families and society. It often leads to emotional distress, social withdrawal, and decreased ability to function in daily life. The economic burden of schizophrenia is considerable, including healthcare costs, lost productivity, and social services. Early and accurate diagnosis, followed by effective treatment, is crucial for improving the prognosis and quality of life for those affected. The search for objective diagnostic methods has led to the exploration of various techniques, including electroencephalography (EEG), which offers promise in identifying neurophysiological markers of the disorder.

The significance of advancing research in schizophrenia diagnosis cannot be overstated. It holds the potential not only to enhance our understanding of this complex disorder but also to revolutionize the way schizophrenia is diagnosed and managed, thereby improving the lives of millions affected worldwide.

Diagnosis of schizophrenia requires a thorough clinical evaluation rather than a definitive medical test. Psychiatrists assess symptoms, medical history, and behavioral patterns to determine if the individual meets the criteria for schizophrenia. Brain imaging and blood tests may be conducted to rule out other medical conditions that mimic schizophrenia symptoms.

Although schizophrenia cannot be cured, effective treatment can help manage symptoms and improve quality of life. Antipsychotic medications are the primary form of treatment, helping regulate brain chemistry and reduce hallucinations and delusions. Cognitive-behavioral therapy (CBT) is also beneficial, equipping patients with coping strategies to manage their condition. Social skills training and rehabilitation programs assist individuals in reintegrating into society and finding employment. Support from family members, friends, and mental health professionals plays a crucial role in long-term stability.

People with schizophrenia can lead fulfilling lives with proper care and early intervention. Establishing a strong support system, maintaining medication adherence, and adopting healthy lifestyle habits—such as regular exercise, balanced nutrition, and stress management—can significantly enhance well-being. Despite the challenges posed by the disorder, many individuals achieve stability and successfully manage their symptoms, proving that schizophrenia does not define a person’s potential or future.

1.2 Electroencephalography (EEG)

Electroencephalography (EEG) is a widely used neurophysiological technique that records the electrical activity of the brain. It is a non-invasive method that involves placing electrodes on the scalp to detect and measure brain wave patterns. EEG is instrumental in diagnosing and studying various neurological conditions, including epilepsy, sleep disorders, brain injuries, and cognitive impairments. The technique has evolved significantly since its inception in 1929 by German scientist Hans Berger, who first demonstrated that electrical activity in the brain could be recorded and analyzed.

The human brain generates electrical impulses that fluctuate rhythmically, forming distinct wave patterns. These waves are classified into different frequency bands: delta (0.5–4 Hz), theta (4–7 Hz), alpha (8–13 Hz), beta (13–30 Hz), and gamma (above 30 Hz). Each of these wave types is associated with different states of consciousness and cognitive functions. For instance, alpha waves are prominent during relaxed wakefulness, while beta waves are linked to active thinking and problem-solving. Delta waves are typically observed during deep sleep, whereas theta waves are associated with drowsiness and meditation. Gamma waves, though less understood, are believed to play a role in higher cognitive functions such as perception and consciousness.

EEG is commonly used in clinical settings to diagnose and monitor neurological disorders. One of its primary applications is in the detection of epilepsy, a condition characterized by abnormal electrical discharges in the brain. EEG can identify epileptic spikes and wave patterns, aiding neurologists in determining the type and severity of seizures. Additionally, EEG is valuable in assessing sleep disorders such as insomnia, narcolepsy, and sleep apnea. By analyzing brain wave activity during different sleep stages, researchers and clinicians can gain insights into sleep quality and disturbances.

Beyond clinical applications, EEG is extensively used in research to study brain function and cognition. Cognitive neuroscience relies on EEG to investigate processes such as attention, memory, language, and decision-making. The technique is particularly useful in event-related potential (ERP) studies, where brain responses to specific stimuli are measured. ERPs provide insights into how the brain processes sensory information and how cognitive functions are affected by various factors, including aging, neurological diseases, and psychological conditions.

The procedure for conducting an EEG is relatively straightforward. Electrodes are placed on the scalp using a conductive gel to ensure proper signal transmission. These electrodes detect voltage fluctuations resulting from neuronal activity, which are then amplified and recorded by an EEG machine. The recorded signals are displayed as waveforms, which can be analyzed visually or through computational methods. Modern EEG systems often incorporate advanced signal processing techniques to enhance data interpretation and reduce artifacts caused by external interference or muscle movements.

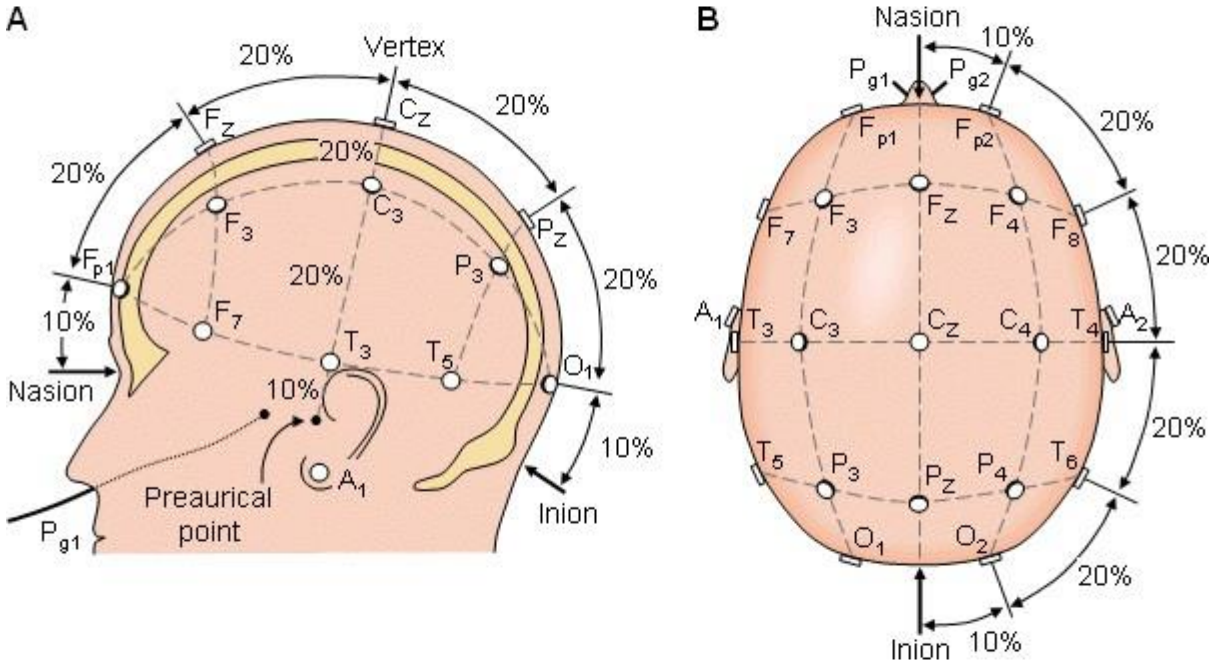


Figure 1: EEG 10-20 Electrode Placement

1.3 Explainable AI

Explainable AI (XAI) is a crucial field within artificial intelligence that focuses on making AI models more transparent, interpretable, and understandable to human users. As AI systems become increasingly complex, the need for explainability has grown, particularly in high-stakes applications such as healthcare, finance, and autonomous systems. Traditional AI models, especially deep learning networks, often function as "black boxes," meaning their decision-making processes are difficult to interpret. This lack of transparency can lead to ethical concerns, biases, and trust issues, making explainability a key requirement for responsible AI development.

The concept of explainable AI emerged as a response to the growing complexity of machine learning models and the need for accountability in AI-driven decisions. Organizations and researchers have developed various techniques to enhance AI interpretability, including model-agnostic methods, intrinsic explainability approaches, and post-hoc explanations. Model-agnostic methods, such as Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), provide insights into AI predictions without altering the underlying model. Intrinsic explainability focuses on designing models that are inherently interpretable, such as decision trees and linear regression. Post-hoc explanations involve analyzing model outputs after predictions have been made, using techniques like feature importance analysis and counterfactual reasoning.

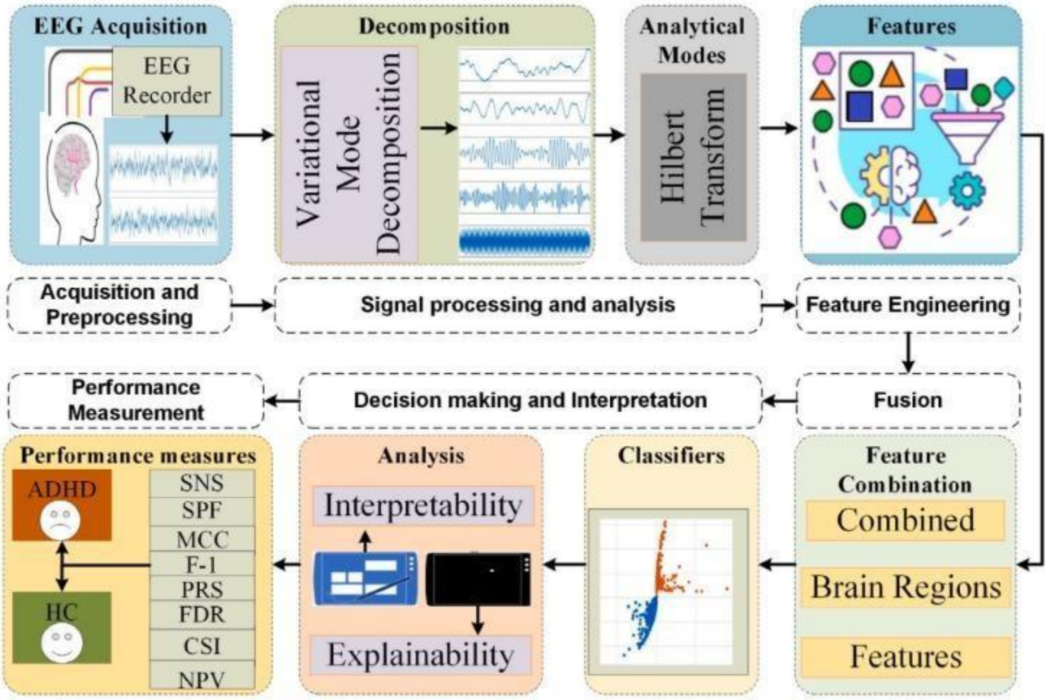
One of the primary motivations for explainable AI is ensuring fairness and mitigating bias in AI systems. AI models trained on biased data can perpetuate discrimination, leading to unfair outcomes in areas such as hiring, lending, and criminal justice. Explainability techniques help identify and address biases by providing insights into how models make decisions. For example, fairness-aware algorithms can adjust predictions to ensure equitable treatment across different demographic groups. Additionally, regulatory frameworks, such as the European Union's General Data Protection Regulation (GDPR), emphasize the importance of AI transparency, requiring organizations to provide explanations for automated decisions that impact individuals.

Explainable AI also plays a vital role in improving user trust and adoption of AI technologies. When users understand how AI models arrive at decisions, they are more likely to trust and accept AI-driven recommendations. This is particularly important in domains like healthcare, where AI-assisted diagnoses must be interpretable for medical professionals to validate and act upon them. In financial services, explainability helps ensure compliance with regulations and enables auditors to assess AI-driven risk assessments. By providing clear and interpretable explanations, AI systems can foster greater confidence among users and stakeholders.

XAI Architecture:

Explainable AI (XAI) architecture is designed to enhance the transparency and interpretability of artificial intelligence models, ensuring that their decision-making processes are understandable to human users. As AI systems become more complex, the need for explainability has grown, particularly in applications where accountability and trust are paramount. XAI architecture integrates various components and methodologies to achieve this goal, including model design, interpretability techniques, and post-hoc explanation methods.

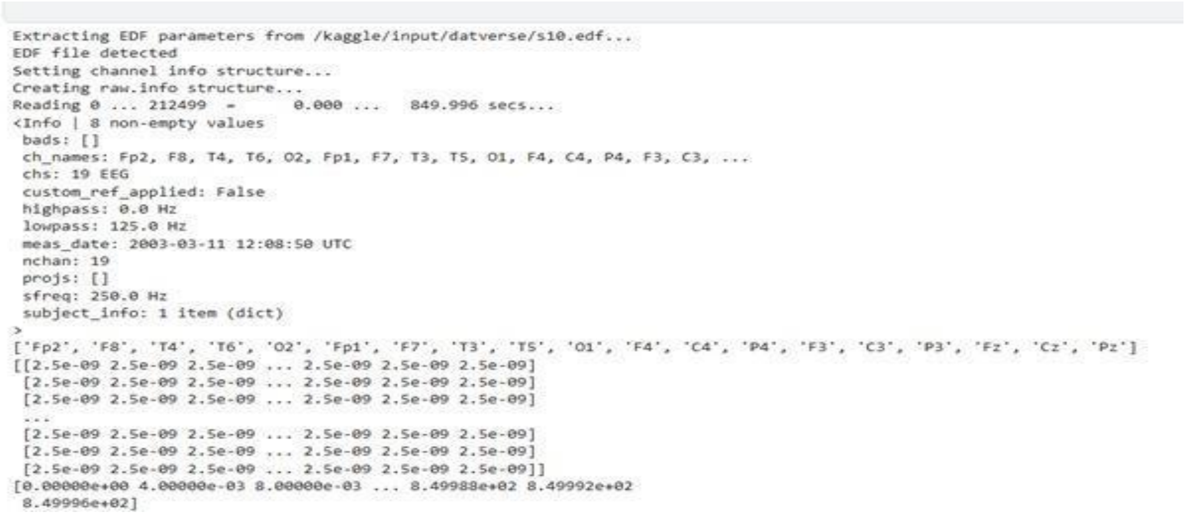
At the core of XAI architecture is the development of inherently interpretable models. Traditional machine learning models, such as decision trees and linear regression, are naturally explainable due to their straightforward structure. However, more complex models, such as deep neural networks, require additional mechanisms to make their predictions interpretable. Researchers have developed hybrid models that combine interpretable components with deep learning architectures, allowing for a balance between accuracy and transparency.



2. Methodology

2.1 DATASET

The EEG data used in this project is sourced from the study titled "Graph-based analysis of brain connectivity in schizophrenia" by Elzbieta Olejarczyk and Wojciech Jernajczyk. This dataset consists of EEG recordings from **28 participants**—14 healthy individuals and 14 diagnosed with schizophrenia. EEG data captures brain activity through electrodes placed on the scalp, providing a high temporal resolution signal that reflects underlying neural processes. Data were acquired with the sampling frequency of 250 Hz using the standard 10-20 EEG montage with 19 EEG channels: Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T6, O1, O2. The reference electrode was placed between electrodes Fz and Cz.



```
Extracting EDF parameters from /kaggle/input/datverse/s10.edf...
EDF file detected
Setting channel info structure...
Creating raw.info structure...
Reading 0 ... 212499  =      0.000 ...   849.996 secs...
<Info | 8 non-empty values
 bads: []
 ch_names: Fp2, F8, T4, T6, O2, Fp1, F7, T3, T5, O1, F4, C4, P4, F3, C3, ...
 chs: 19 EEG
 custom_ref_applied: False
 highpass: 0.0 Hz
 lowpass: 125.0 Hz
 meas_date: 2003-03-11 12:08:50 UTC
 nchan: 19
 projs: []
 sfreq: 250.0 Hz
 subject_info: 1 item (dict)
>
['Fp2', 'F8', 'T4', 'T6', 'O2', 'Fp1', 'F7', 'T3', 'T5', 'O1', 'F4', 'C4', 'P4', 'F3', 'C3', 'P3', 'Fz', 'Cz', 'Pz']
[[2.5e-09 2.5e-09 2.5e-09 ... 2.5e-09 2.5e-09 2.5e-09]
 [2.5e-09 2.5e-09 2.5e-09 ... 2.5e-09 2.5e-09 2.5e-09]
 [2.5e-09 2.5e-09 2.5e-09 ... 2.5e-09 2.5e-09 2.5e-09]
 ...
 [2.5e-09 2.5e-09 2.5e-09 ... 2.5e-09 2.5e-09 2.5e-09]
 [2.5e-09 2.5e-09 2.5e-09 ... 2.5e-09 2.5e-09 2.5e-09]
 [2.5e-09 2.5e-09 2.5e-09 ... 2.5e-09 2.5e-09 2.5e-09]]
[0.00000e+00 4.00000e-03 8.00000e-03 ... 8.49988e+02 8.49992e+02
 8.49996e+02]
```

2.2 Environment

Kaggle: Kaggle is an online community and platform for data science and machine learning practitioners. It provides tools, resources, and opportunities for collaboration and competition in the field.

- **Competitions**: Kaggle hosts machine learning challenges where individuals or teams solve real-world problems using datasets provided by organizations.
- **Datasets**: It offers a vast repository of datasets across various domains, which users can explore, download, and use for analysis and modeling.
- **Kernels**: Kaggle provides an integrated environment (now called "Notebooks") where users can write and execute Python or R code directly in the browser without requiring local setup.
- **Community**: Kaggle has an active community of data scientists who share code, ideas, and insights through forums and discussions.
- **Learning Resources**: It offers free courses on topics like Python, machine learning, data visualization, and deep learning, making it a valuable educational resource.

Anaconda: Anaconda is a popular open-source distribution of Python and R for data science, machine learning, and scientific computing. It simplifies the process of managing packages and environments, making it an essential tool for data practitioners.

2.3 Preprocessing:

To ensure data quality and consistency, we used the **MNE-Python** package, which is a powerful tool for preprocessing, analyzing and visualizing EEG data. Our preprocessing steps included:

- **Filtering:** We applied bandpass filtering to isolate frequency bands of interest.
- **Artifact Removal:** Noise and non-neural artifacts were mitigated to improve data quality.

- **Segmentation:** EEG data was segmented into manageable epochs, allowing for analysis of temporal patterns in brain activity.

2.3.1 Feature Extraction

Using MNE and custom functions, we extracted a set of **statistical features** from the EEG signals, such as mean, variance, skewness, and kurtosis , in total 13 features of each segment. These features were selected to capture both the amplitude and variability of neural activity.

2.3.2 Classification Model

For classification, we employed a **Random Forest (RF)** model, which is well-suited for EEG data due to its ability to handle complex interactions and non-linear relationships between features. The Random Forest model was trained on the extracted EEG features, achieving an accuracy of **93%** in distinguishing between healthy and schizophrenic participants

2.3.3 Explainability through SHAP

To understand the decision-making process of the Random Forest model, we applied **SHAP (SHapley Additive exPlanations)**, an explainable AI technique that provides insights into feature importance for individual predictions:

- **Global Interpretability:** SHAP values allowed us to identify which features, on average, contributed most to the model's decisions. Certain statistical features, such as variance and skewness, were found to play a prominent role in classification.
- **Local Interpretability:** SHAP also provided case-by-case explanations, revealing the specific features that influenced predictions for each individual. This local interpretability is especially valuable in clinical settings, where patient-specific insights can inform personalized treatment plans.

2.4 Black Box Model

2.4.1 Logistic Regression

Logistic Regression is a statistical model used for binary classification tasks, where the goal is to predict the probability of an event occurring, given a set of input features. Unlike linear regression, which predicts continuous values, logistic regression is specifically designed to model the probability of a binary outcome (e.g., yes/no, 1/0). It uses the logistic function (also called the sigmoid function) to map the predicted values to a probability between 0 and 1.

The equation for logistic regression is:

$$P=1/1+e^{-z}$$

Where: p is the probability that the event
occurs. e is the base of the natural
logarithm.
z is the linear combination of input features, defined as:
$z=\beta 0+\beta 1X1+\beta 2X2+\cdots+\beta nXn$

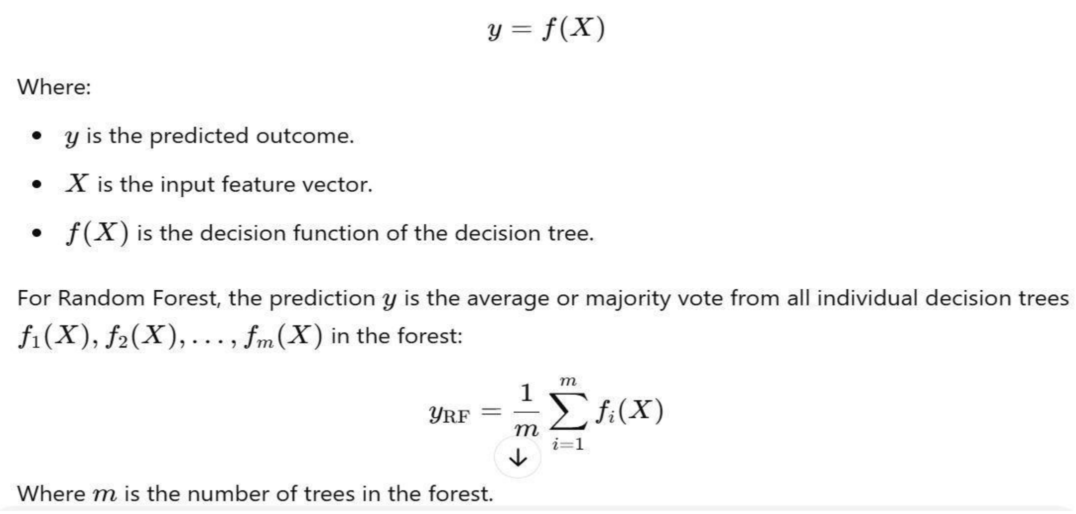
2.4.2 Random Forest

Random Forest is an ensemble learning technique that combines the predictions of multiple decision trees to improve classification accuracy and robustness. It is a type of bagging model that builds many decision trees using different random subsets of the training data, and the final prediction is made based on a majority vote or averaging from all the trees.

Each decision tree in the forest is trained using a random subset of the features, and the prediction is determined by aggregating the outputs of all the trees.

The Random Forest algorithm can be summarized in the following steps:

Bootstrap Sampling: Randomly sample subsets of the training data with replacement to build multiple decision trees.
There isn't a specific equation for Random Forest as it is based on multiple decision trees, but the prediction from a single decision tree is given by:



$$y = f(X)$$

Where:

- y is the predicted outcome.
- X is the input feature vector.
- $f(X)$ is the decision function of the decision tree.

For Random Forest, the prediction y is the average or majority vote from all individual decision trees $f_1(X), f_2(X), \ldots, f_m(X)$ in the forest:

$$y_{\mathrm{RF}} = \frac{1}{m}\sum_{i=1}^{m} f_i(X)$$

Where m is the number of trees in the forest.

Random Feature Selection: For each tree, randomly select a subset of features to split on at each node.

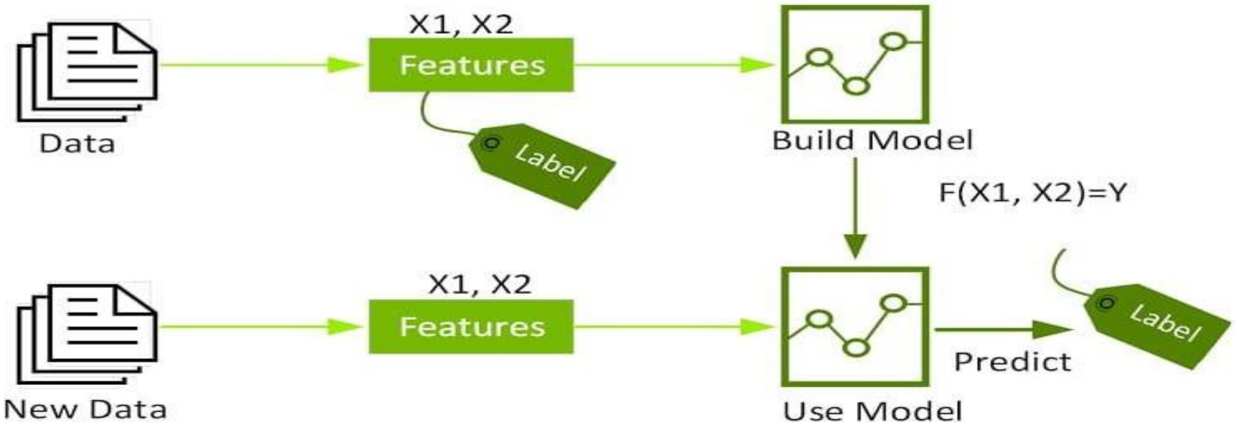
Aggregation: For classification tasks, predict the class that has the majority vote from all the trees. For regression tasks, average the predictions from all trees.

2.4.3 XgBoost(extreme Boosting)

- XGBoost builds decision trees sequentially, with each tree trying to fix mistakes from the previous one.
- It uses parallel and distributed computing to speed up model discovery.
- It works well with large, complex datasets.

Working Mechanism:

- XGBoost (Extreme Gradient Boosting)
- Uses gradient boosting, where each new tree corrects the errors of previous trees.
- Uses a loss function gradient to minimize errors.
- Supports parallel computation for fast execution.
- Includes L1 and L2 regularization to prevent overfitting.
- Handles missing values automatically.



2.4.4 AdBoost(Adaptive Boosting)

AdaBoost, or adaptive boosting, is a machine learning technique that combines multiple weak classifiers to create a strong classifier. It's used to improve the accuracy of a classifier for classification or regression tasks.

- AdaBoost starts with equal weights for all training samples.
- It iteratively adjusts the weights to focus on misclassified data points.
- It combines the output of the weak classifiers into a weighted sum.
- The better-performing trees get more influence in the final

decision. Working Mechanism of AdaBoost (Adaptive Boosting)

AdaBoost is a sequential ensemble learning technique that improves weak classifiers by adjusting their weights iteratively. Here’s how it works step by step:

Step-by-Step Working of AdaBoost:

1. Initialize Weights

Assign equal weights to all training samples (initially 1/N, where N is the number of samples). These weights determine how much influence each sample has on training.

2. Train a Weak Classifier

Train a simple weak model (usually a decision stump, which is a one-level decision tree). Evaluate its performance on the training set.

3. Compute the Error Rate

4. Compute Classifier Weight

5. Update Sample Weights Increase weights of misclassified samples so that the next classifier focuses more on them.

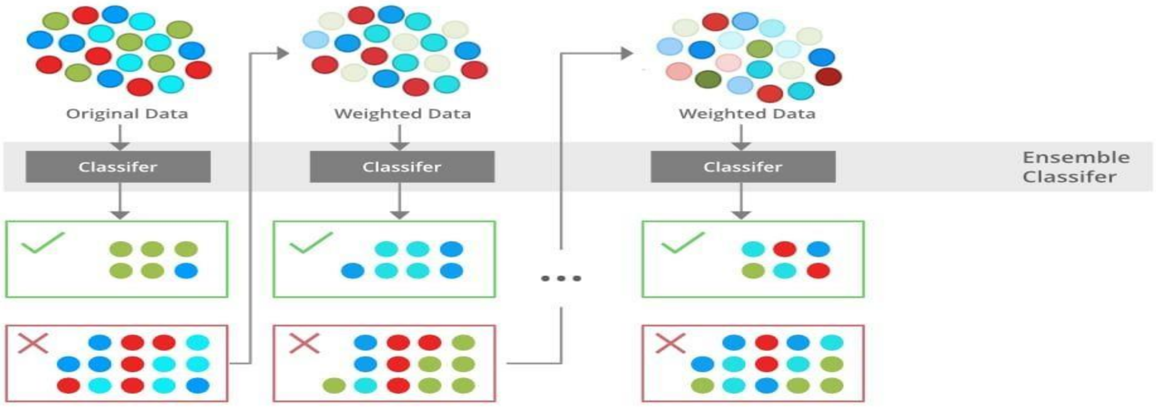
6. Repeat Steps 2-5

Train multiple weak classifiers iteratively.

Each new classifier focuses more on the previously misclassified samples.

7. Final Prediction (Weighted Voting)

Combine the predictions of all weak classifiers using their weights.



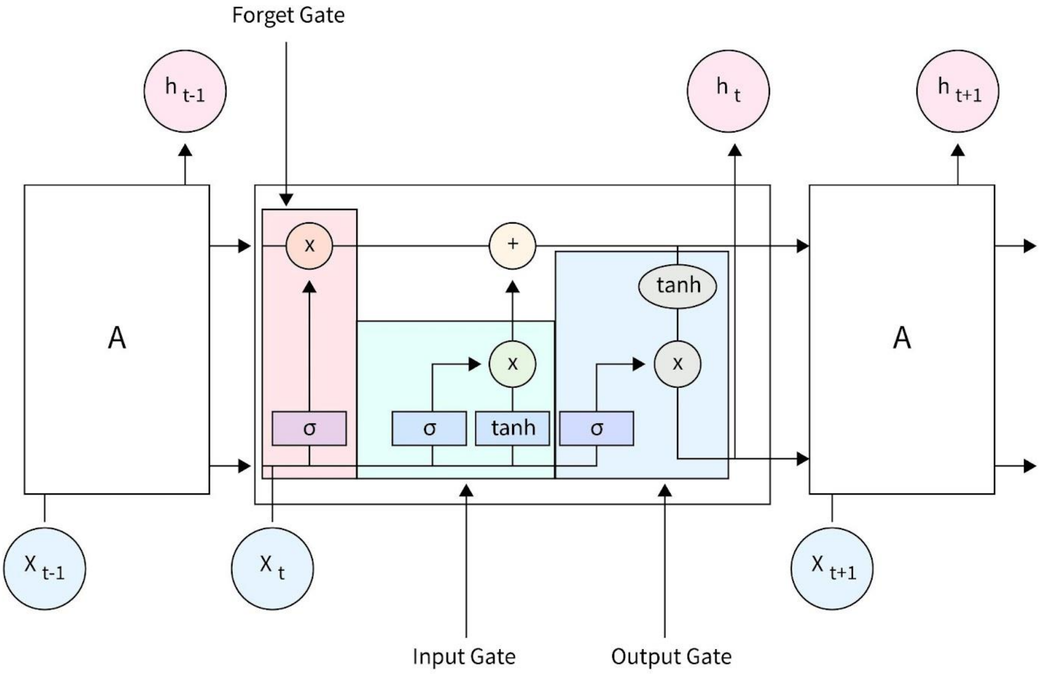
2.4.5 SVM(Support Vector Machine)

Support Vector Machines (SVM) are a powerful and widely used supervised learning algorithm designed for classification, regression, and outlier detection. Introduced by Vladimir Vapnik and his colleagues in the 1990s, SVMs have become a fundamental tool in machine learning due to their ability to handle high-dimensional data and provide robust decision boundaries. The core idea behind SVM is to find an optimal hyperplane that separates different classes in a dataset while maximizing the margin between them. This margin-based approach ensures that the model generalizes well to unseen data, making SVM particularly effective in applications such as image recognition, text classification, and bioinformatics.

The mathematical foundation of SVM revolves around the concept of hyperplanes and support vectors. In a two-dimensional space, a hyperplane is simply a line that divides the data into two categories. In higher dimensions, it becomes a plane or a more complex geometric structure. The goal of SVM is to identify the hyperplane that maximizes the margin between the closest data points from each class, known as support vectors. These support vectors play a crucial role in defining the decision boundary, ensuring that the model is not overly sensitive to individual data points but rather focuses on the overall structure of the dataset.

2.4.6 LSTM(Long Short Term Memory)

Long Short-Term Memory (LSTM) networks are a specialized form of **Recurrent Neural Networks (RNNs)** designed to model sequential and time-series data. Standard RNNs suffer from the **vanishing and exploding gradient problem**, which makes them ineffective in learning long-term dependencies. LSTMs overcome this limitation by introducing a **memory cell** and **gating mechanisms**, enabling them to capture dependencies over longer time intervals.



2.5 Explainable AI models

2.5.1 LIME

Local Interpretable Model Agnostic Explanations, LIME [35] is a method that can make an ML model understandable while remaining model agnostic. Formally, we define an explanation as a model g ∈ G, where G is a class of potentially interpretable models, such as linear models, decision trees, or falling rule lists [27], i.e. a model g ∈ G can be readily presented to the user with visual or textual artifacts. The domain of g is {0, 1} d 0 , i.e. g acts over absence/presence of the interpretable components. As not every g ∈ G may be simple enough to be interpretable - thus we let Ω(g) be a measure of complexity (as opposed to interpretability) of the explanation g ∈ G. For example, for decision trees Ω(g) may be the depth of the tree, while for linear models, Ω(g) may be the number of non-zero weights. Let the model being explained be denoted f : R d → R. In classification, f(x) is the probability (or a binary indicator) that x belongs to a certain class1 . We further use πx(z) as a proximity measure between an instance z to x, so as to define locality around x. Finally, let L(f, g, πx) be a measure of how unfaithful g is in approximating f in the locality defined by πx. In order to ensure both interpretability and local fidelity, we must minimize L(f, g, πx) while having Ω(g) be low enough to be interpretable by humans. The explanation produced by LIME is obtained by the following:

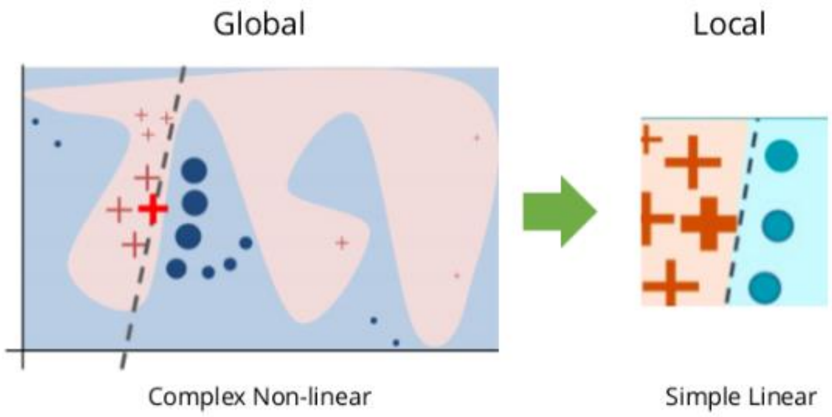
$$\xi(x) = \text{argmin } g{\in}G$$

$$L(f, g, \pi x) + \Omega(g) \ (9)$$

where G is denoted as a set of interpretable models, and the (g) denotes the complexity of the explanation g ∈ G.

The objective is to reduce (g) so that the simpler models can also be interpreted. The L(f, g, πx) represents a measure of how closely the explanation model g matches of the original model's prediction, also known as fidelity. Local fidelity refers to the need for the explanation to accurately represent the classifier's behavior "around" the instance being predicted without peeking into the model, thus the model-agnostic approach. The LIME model generates outputs in visual figures that are divided into three sections:

the feature probabilities on the left, the feature probabilities on the right, and the feature value table on the bottom side of the figure. The predicted values of the probabilities are located on the left. The graph of prediction probabilities illustrates the model's judgment on



For humans to trust AI systems, it is essential for models to be explainable to users. AI interpretability reveals what is happening within these systems and helps identify potential issues such as information leakage, model bias, robustness, and causality. LIME offers a generic framework to uncover black boxes and provides the "why" behind AI-generated predictions or recommendations.

2.5.2 Shap(shaply additive explainations)

As modern machine learning models grow in complexity, understanding why a model makes a certain prediction is critical for trust, transparency, and debugging. To address this, SHAP (SHapley Additive exPlanations) provides a unified framework for interpreting model predictions by assigning feature importance scores based on cooperative game theory.

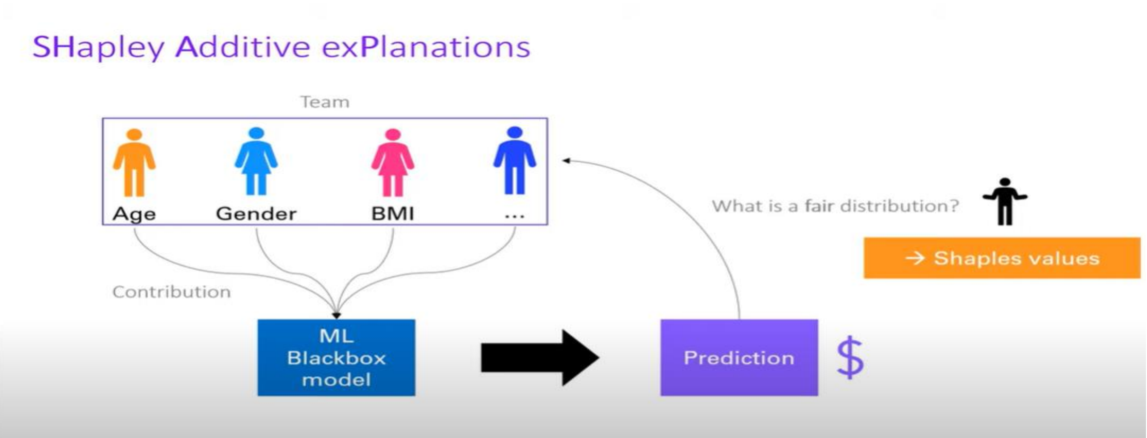
SHAP values are grounded in the Shapley values from game theory, which fairly distribute the "payout" (i.e., model prediction) among features based on their marginal contributions across all possible feature combinations.

Key Features of SHAP:

- Model-Agnostic: Works with any machine learning model, including decision trees, neural networks, and linear models.
- Global and Local Interpretations: Provides explanations for individual predictions (local) and overall feature importance across the model (global).
- Fairness and Consistency: The Shapley value framework ensures that feature contributions are fairly distributed based on their marginal contributions to the model.
- Visualization Tools: SHAP provides powerful visualizations, such as:
  Summary Plots: Show the overall impact of features on
  predictions. Dependence Plots: Illustrate how changes in a feature
  affect the prediction. Force Plots: Provide a detailed explanation
  for a single prediction.

Workflow of SHAP:

- Model Training: Train a machine learning model on the dataset.
- Feature Attribution: Use SHAP to compute the Shapley values for each feature.
- Explanation Generation: Visualize and analyze the results to understand the model's decision-making process.



Given a model f, SHAP explains the prediction $f(x)$ for an instance x by attributing contributions ϕ_i to each feature x_i, such that:

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i$$

- ϕ_0: the expected model output (baseline or reference value)
- ϕ_i: the contribution of feature i to the prediction

This additive form ensures local interpretability, where each feature's effect on the prediction is explicitly represented.

SHAP Value Formula

The SHAP value ϕ_i for feature i is defined as:

$$\phi_i(f, x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right]$$

Where:

- F is the full set of input features.
- $S \subseteq F \setminus \{i\}$ represents all subsets of features excluding i.
- $f_S(x_S)$: model prediction using only the features in subset S.
- The coefficients are Shapley weights ensuring fairness across all permutations.

SHAP is the unique solution among additive feature attribution methods that satisfies the following properties:

1. Local Accuracy: The sum of attributions matches the model's output.
2. Missingness: Features that are not present receive zero attribution.
3. Consistency: If a model changes so that a feature's marginal contribution increases, its attribution should not decrease.

2.5.3 Eli5(Explain I’m Like 5)

Explain Like I’m 5 (ELI5) is a Python library that helps understand the predictions made by machine learning models in a simple and easy-to-understand way. It is useful for debugging models and improving their interpretability. With ELI5, one can get a better understanding of how a model makes its predictions, which features are important for those predictions, and whether the model is making biased or unexpected decisions.

ELI5 offers diverse techniques to explain various types of machine learning models, such as linear and tree-based models, as well as deep learning models. Additionally, ELI5 is compatible with multiple machine learning libraries, including scikit-learn, Keras, XGBoost, and LightGBM.

ELI5 is known for its ability to offer global and local explanations, two types of explanations that give insight into different aspects of machine learning models. Global explanations provide a broad understanding of the model’s behavior and what features are important for making predictions. On the other hand, local explanations provide information about why a specific instance was classified in a particular way and which features influenced that classification. This feature of ELI5 is essential for gaining a comprehensive understanding of the model’s inner workings and identifying any issues that may exist.

Advantages and Disadvantages of ELI5

ELI5 provides several advantages for machine learning practitioners, such as:

1. **Improved Model Performance**: ELI5 can identify the most impactful features on predictions, allowing for the removal of irrelevant or redundant features. This simplifies the model and enhances its accuracy.
2. **Interpretability**: ELI5 can offer visual explanations of how models function, making them easier to interpret. This can aid stakeholders, including decision-makers and customers, in comprehending how the model arrived at its decision.
3. **Debugging**: ELI5 can assist in identifying errors or biases in the model. By analyzing the most important features of each prediction, we can determine whether the

model is making decisions based on relevant information or merely exploiting spurious correlations.

Despite its many benefits, ELI5 has some limitations that users should be aware of. Here are a few:

1. **Interpretation may vary**: ELI5's explanations can vary based on the model, data, and parameters used. Therefore, it is important to carefully evaluate and interpret the results.
2. **Incompatibility with certain packages**: ELI5 may not be compatible with some machine learning packages, such as statsmodels or the surprise package used in recommender systems. This limits its applicability to a few packages.

2.5.4 PDP

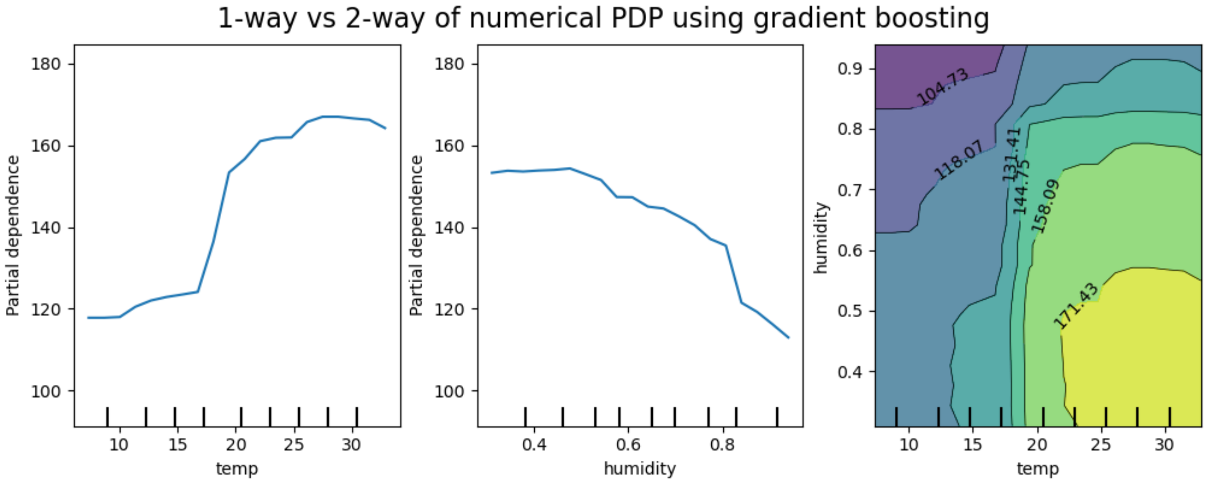
A Partial Dependence Plot (PDP) is a global, model-agnostic interpretability technique that illustrates the marginal effect of one or two input features on the predicted outcome of a machine learning model. By averaging the model's predictions over the distribution of other features, PDPs help in understanding the relationship between specific features and the target variable.

The partial dependence function for regression is defined as:

$$\hat{f}_S(\mathbf{x}_S) = \mathbb{E}_{\mathbf{X}_C}\left[\hat{f}(\mathbf{x}_S, X_C)\right] = \int \hat{f}(\mathbf{x}_S, X_C)d\mathbb{P}(\mathbf{X}_C)$$

The xS are the features for which the partial dependence function should be plotted and XC are the other features used in the machine learning model f^, which are here treated as random variables. Usually, there is only one or two features in the set S. The feature(s) in S are those for which we want to know the effect on the prediction. The feature vectors xS and xC combined make up the total data X. Partial dependence works by marginalizing the machine learning model output over the distribution of the features in set C, so that the function shows the relationship between the features in set S we are interested in and the predicted outcome. By marginalizing over the other features, we get a function that depends only on features in S, with interactions with other features included.

The figure below shows two one-way and one two-way partial dependence plots for the bike sharing dataset, with a Hist Gradient Boosting Regressor:



2.5.5 Integrated Gradients (IG)

Integrated Gradients is an attribution method designed to explain the predictions of deep neural networks. It quantifies the contribution of each input feature to the model's output by integrating the gradients of the output with respect to the inputs along a path from a baseline input to the actual input.

This method is **axiomatically justified** and satisfies two important properties:

Definition 1 (Axiom: Sensitivity) *An attribution method satisfies **Sensitivity** if for every input and baseline that differ in one feature but have different predictions, then the differing feature should be given a non-zero attribution. If the function implemented by the deep network does not depend (mathematically) on some variable, then the attribution to that variable is always zero.*

Definition 2 (Axiom: Implementation Invariance) *Two networks are functionally equivalent if their outputs are equal for all inputs, despite having very different implementations. Attribution methods should satisfy Implementation Invariance, i.e., the attributions are always identical for two functionally equivalent networks.*

Gradient-based explanations suffer from **saturation** and **discontinuities**, which can lead to noisy or misleading attributions. Integrated Gradients addresses this by **accumulating gradients** over many scaled versions of the input.

Given a model $F : \mathbb{R}^n \rightarrow \mathbb{R}$, an input $x \in \mathbb{R}^n$, and a baseline input x' (e.g., all zeros or a reference input), the Integrated Gradient along the i-th dimension is defined as:

$$\mathrm{IG}_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \cdot (x - x'))}{\partial x_i} \, d\alpha$$

Where:

- x: The actual input
- x': The baseline input
- α: Scalar that interpolates between the baseline and the actual input
- $\frac{\partial F}{\partial x_i}$: Gradient of the output with respect to the i-th input

This integral is usually **approximated numerically** using a Riemann sum over m steps:

$$\mathrm{IG}_i(x) \approx (x_i - x_i') \times \frac{1}{m} \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m}(x - x'))}{\partial x_i}$$

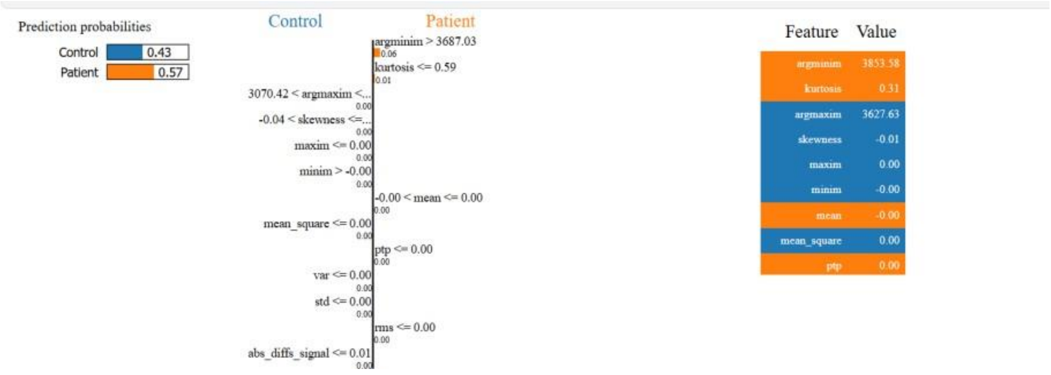
Each IG i(x) tells us **how much feature i** contributed to the difference in output between the actual input and the baseline.

- A **positive value** indicates the feature contributed positively to the prediction.
- A **negative value** means it decreased the prediction.
- A value **close to zero** suggests little influence.

The sensitivity axiom introduces the **baseline** which is an important part of the IG method. A baseline is defined as an **absence of a feature** in an input. This definition is confusing, especially when dealing with complex models, but the baseline could be interpreted as *"input from the input space that produces a neutral prediction"*. A baseline can be treated as an input to produce a counterfactual explanation by checking how the model behaves when moving from baseline to the original image.

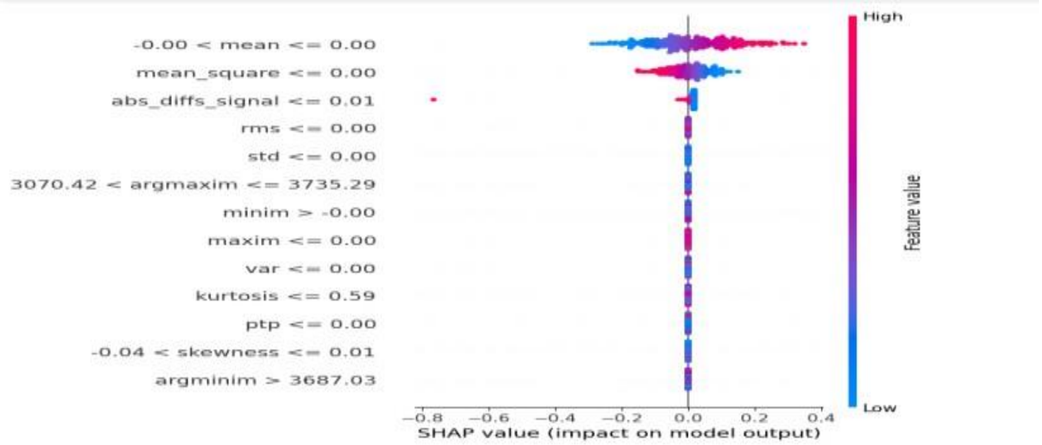
3. Results and Analysis

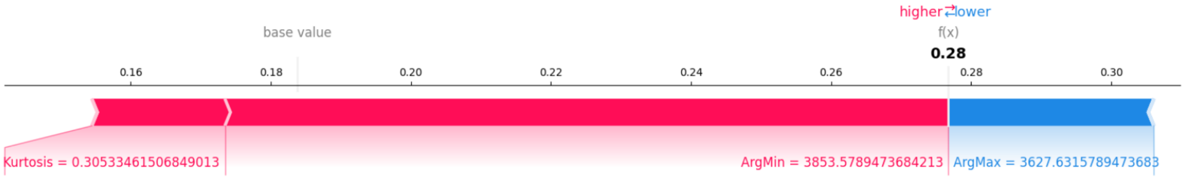
3.1 Applied 3 explainable AI model over logistic regression



3.1.1 LIME Explanation :

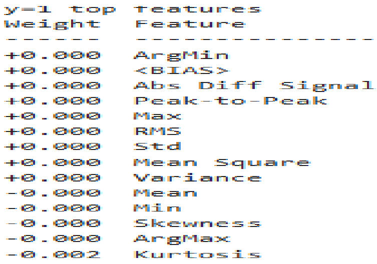
- The image displays a LIME explanation for a single instance.
- The prediction probabilities show a 57% chance that the sample belongs to the Patient class.
- Key contributing features include:
- argminim (Time of Minimum Value in Signal) – This positively impacts the prediction.
- kurtosis (Sharpness of the Signal Distribution) – A lower kurtosis value contributes to the Patient class.
- argmaxim (Time of Maximum Value in Signal) – This feature also played a role in distinguishing between classes.



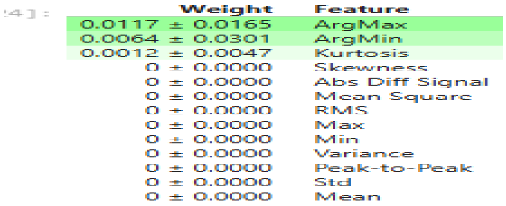


3.1.2 SHAP Analysis :

- Features with a higher absolute SHAP value contribute significantly to predictions.
- Key takeaways : argmaxim, argminim, and kurtosis are among the most influential features.
- Blue indicates lower feature values, while red indicates higher values.
- Features like mean, skewness, and peak-to-peak amplitude have relatively lower impact.



```
y=1 top features
Weight   Feature
------   ----------------
+0.000   ArgMin
+0.000   <BIAS>
+0.000   Abs Diff Signal
+0.000   Peak-to-Peak
+0.000   Max
+0.000   RMS
+0.000   Std
+0.000   Mean Square
+0.000   Variance
-0.000   Mean
-0.000   Min
-0.000   Skewness
-0.000   ArgMax
-0.002   Kurtosis
```

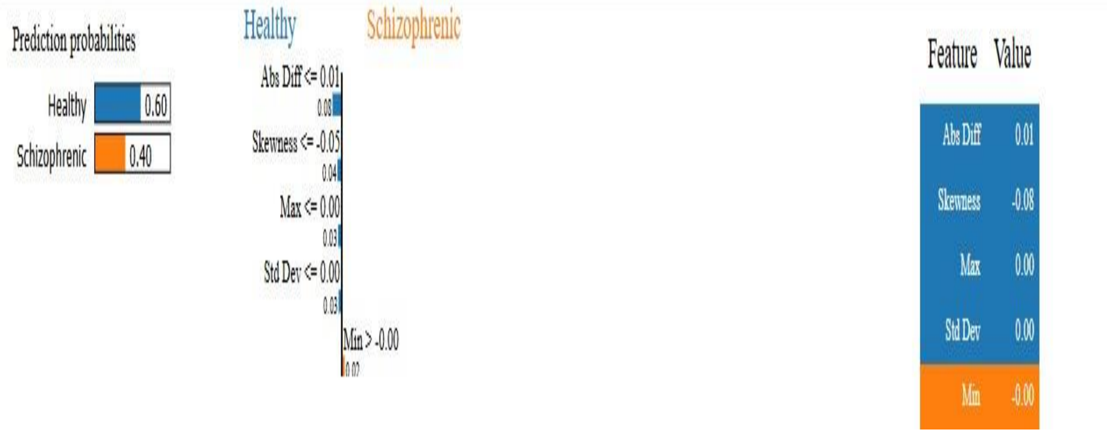


```
[4]:          Weight          Feature
      0.0117 ± 0.0165      ArgMax
      0.0064 ± 0.0301      ArgMin
      0.0012 ± 0.0047      Kurtosis
           0 ± 0.0000      Skewness
           0 ± 0.0000      Abs Diff Signal
           0 ± 0.0000      Mean Square
           0 ± 0.0000      RMS
           0 ± 0.0000      Max
           0 ± 0.0000      Min
           0 ± 0.0000      Variance
           0 ± 0.0000      Peak-to-Peak
           0 ± 0.0000      Std
           0 ± 0.0000      Mean
```

3.1.3 Eli5 Analysis :

The top contributing features are:
argmax (Weight: 0.0117)
argmin (Weight: 0.0064)
kurtosis (Weight: 0.0012)
Other features like mean square, variance, and peak-to-peak signal have negligible importance.

3.2 Applied 4 explainable ai model over Random Forest

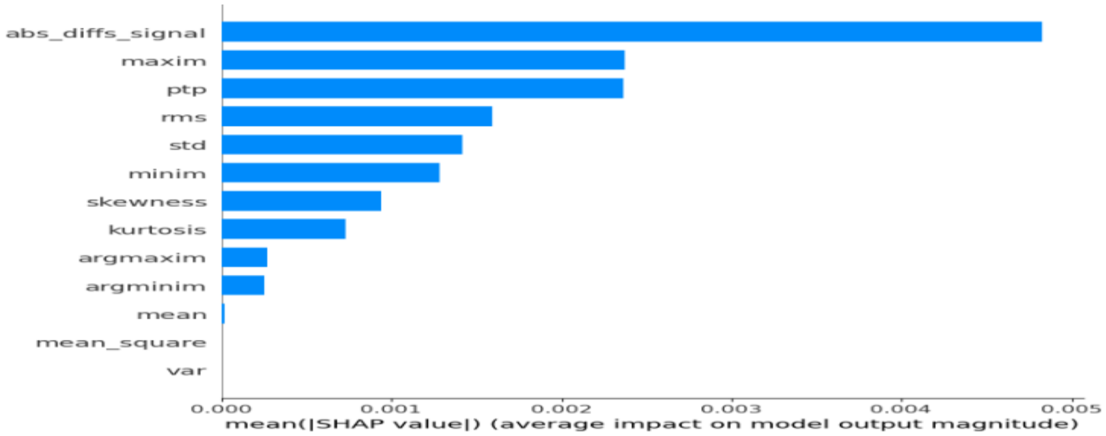


3.2.1 LIME Explanation for a Single Prediction :

Explanation: This visualization shows how individual feature values contribute to predicting whether a subject is healthy or schizophrenic.

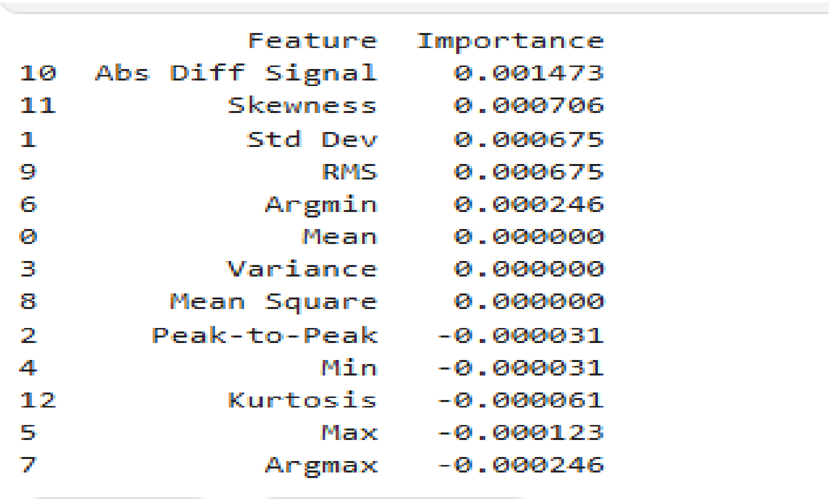
Key Observations : Abs Diff Signal (0.01) and Skewness (-0.08) are the dominant features influencing the prediction.

The prediction probability is 60% Healthy and 40% Schizophrenic



3.2.2 SHAP-Based Feature Importance Ranking :

- Explanation: This bar chart represents the mean absolute SHAP values for different features used in the Random Forest model. The higher the SHAP value, the more impact the feature has on the model’s predictions.
- Key Observations:
  - The most influential feature is abs_diffs_signal, followed by maxim, ptp (peak-to-peak amplitude), and rms.
  - Features like argmaxim, argminim, mean, and variance have minimal impact on the predictions.



```
              Feature  Importance
10    Abs Diff Signal    0.001473
11           Skewness    0.000706
1             Std Dev    0.000675
9                 RMS    0.000675
6              Argmin    0.000246
0                Mean    0.000000
3            Variance    0.000000
8         Mean Square    0.000000
2        Peak-to-Peak   -0.000031
4                 Min   -0.000031
12           Kurtosis   -0.000061
5                 Max   -0.000123
7              Argmax   -0.000246
```

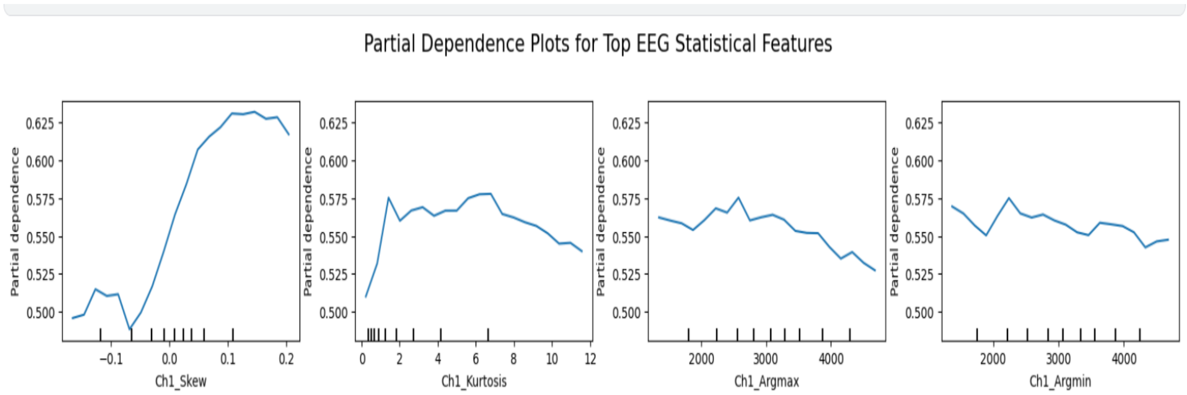
3.2.3 ELI5 Feature Importance Table :

Explanation: This table shows feature importance scores derived from ELI5, another explainability tool.

Key Observations:

The Abs Diff Signal is again identified as the most crucial feature.

Unlike SHAP, Skewness and Standard Deviation have higher importance scores here. Negative importance scores indicate features that decrease the model’s confidence in classification.



1. Ch1_Skew

The PDP for **Ch1_Skew** reveals a strong **positive correlation** with the predicted class probability. As the skewness value increases from approximately -0.1 to 0.2, the model's prediction probability steadily rises from around 0.50 to over 0.62. This suggests that **higher skewness in the EEG signal** from channel 1 is associated with a higher likelihood of classifying a subject as schizophrenic (or whichever positive class the model targets).

2. Ch1_Kurtosis

The PDP for **Ch1_Kurtosis** exhibits a non-linear trend. The prediction probability **initially increases**, peaking around a kurtosis value of 2–4, and then **gradually declines** as the kurtosis increases beyond 6. This indicates that **moderate levels of kurtosis** (i.e., a certain degree of peakiness in the signal) are more indicative of the positive class than either low or very high kurtosis.

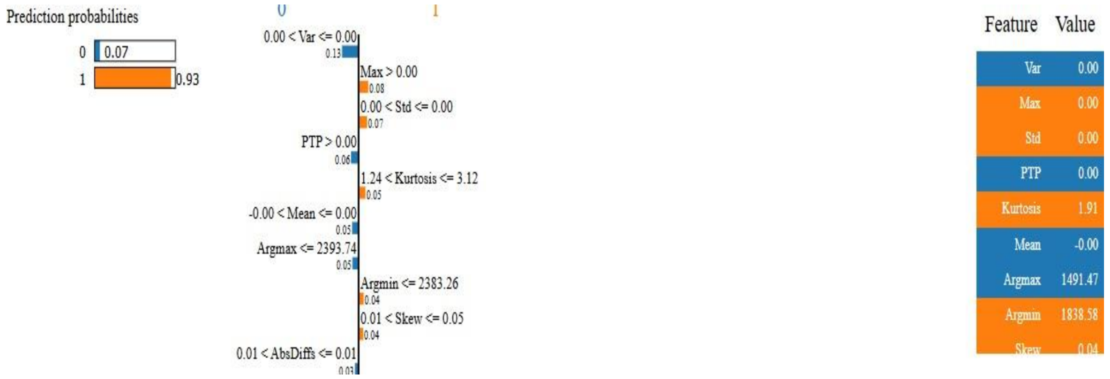
3. Ch1_Argmax

The **Ch1_Argmax** plot, representing the index location of the maximum EEG value in the signal, displays a **slightly negative correlation** with the predicted probability. As the value increases (i.e., the peak of the signal occurs later in time), the model's prediction slightly decreases. This implies that **earlier occurrences of peak activity** in the EEG signal are marginally more associated with the target class.

4. Ch1_Argmin

The **Ch1_Argmin** plot shows a **similar trend to Ch1_Argmax**, where the prediction probability slightly **decreases** as the index of the minimum value increases. This suggests that **early occurrences of negative peaks** (signal dips) may also carry more relevance to the predicted class.

3.3 Applied 4 explainable ai model over XGBoost

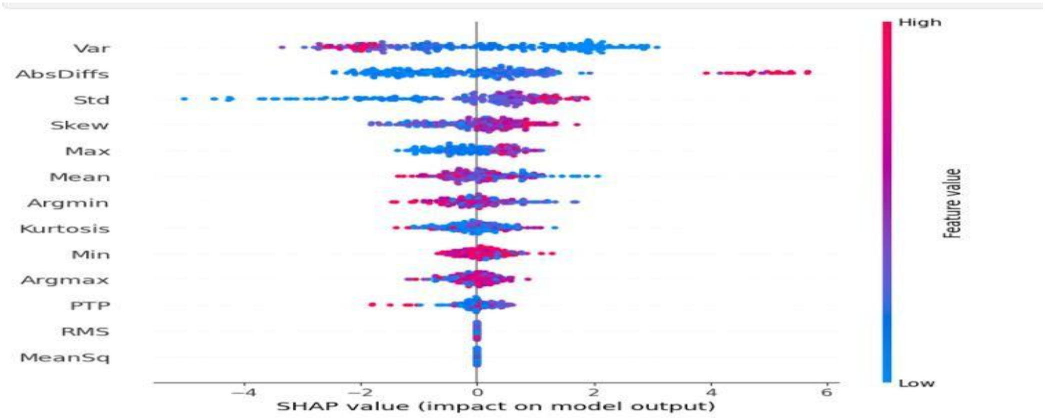


3.3.1 LIME Analysis:

The LIME explanation visualizations show how a particular instance is classified.

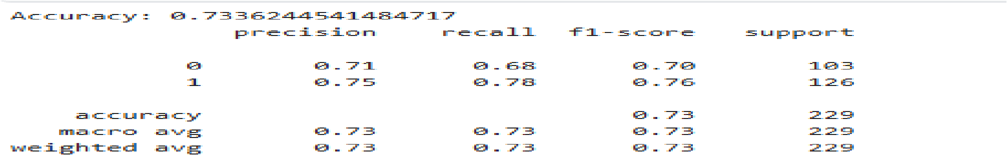
The probability distribution bars indicate the likelihood of a patient being schizophrenic or healthy.

Decision thresholds for key features are displayed, such as Max > 0, Kurtosis > 1.24, and PTP > 0.



3.3.2 SHAP Analysis:

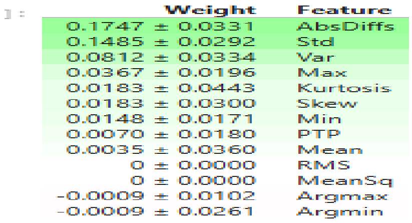
- SHAP summary plots showing how each feature influences the model's prediction.
- Key Features:
- AbsDiffSignal, Max, PTP (Peak-to-Peak), RMS, and Std Dev are the most influential features in both models.
- SHAP values show how these features impact the model’s decision, with red indicating higher feature values and blue indicating lower values.



```
Accuracy: 0.7336244541484717
              precision    recall  f1-score   support

           0       0.71      0.68      0.70       103
           1       0.75      0.78      0.76       126

    accuracy                           0.73       229
   macro avg       0.73      0.73      0.73       229
weighted avg       0.73      0.73      0.73       229
```

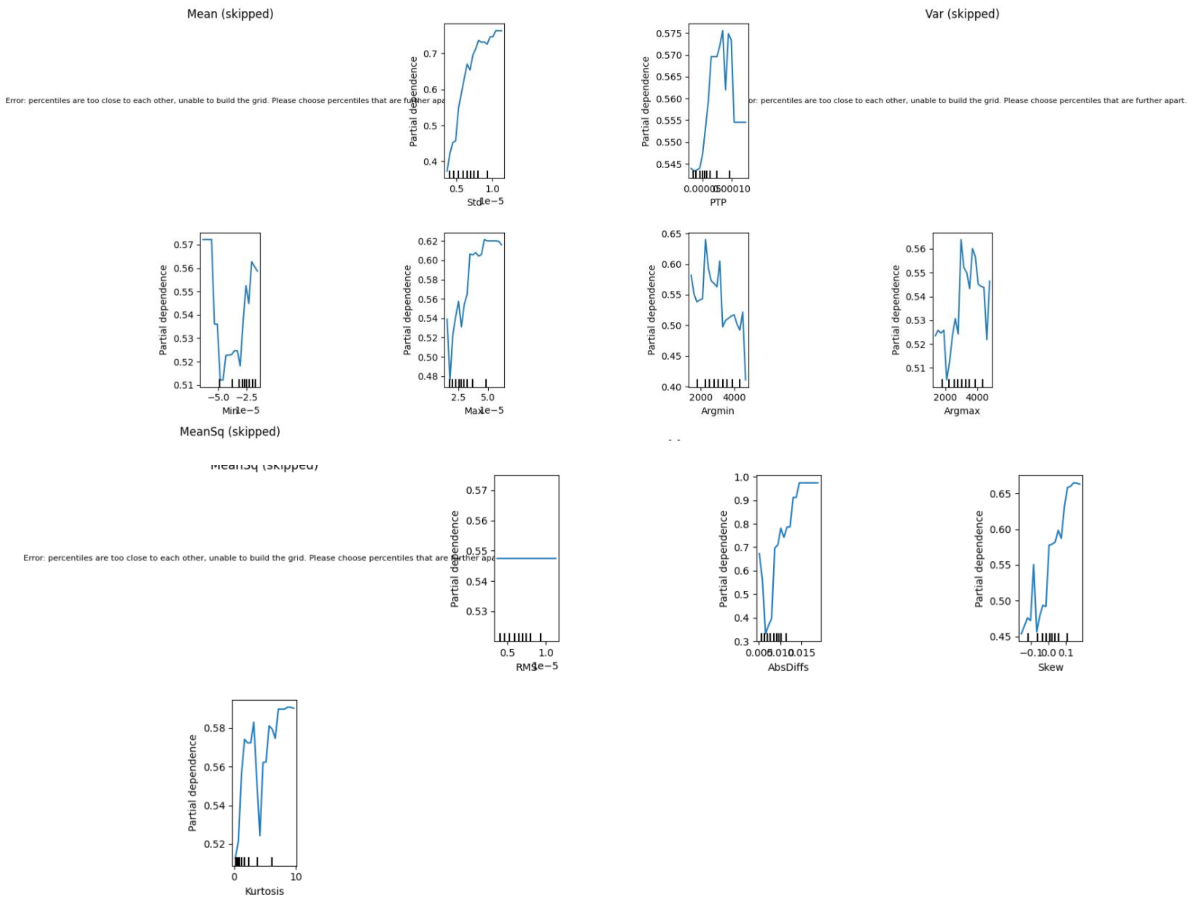


```
         Weight          Feature
0.1747 ± 0.0331          AbsDiffs
0.1485 ± 0.0292          Std
0.0812 ± 0.0334          Var
0.0367 ± 0.0196          Max
0.0183 ± 0.0443          Kurtosis
0.0183 ± 0.0300          Skew
0.0148 ± 0.0171          Min
0.0070 ± 0.0180          PTP
0.0035 ± 0.0360          Mean
     0 ± 0.0000          RMS
     0 ± 0.0000          MeanSq
-0.0009 ± 0.0102         Argmax
-0.0009 ± 0.0261         Argmin
```

3.3.3 Eli5 Analysis:

- The top features (AbsDiffSignal, Std, Var, Max) have high positive weights.
- Some features (like Argmax, MeanSq) have near-zero or negative importance.

Partial Dependence Plots for All 13 EEG Features



3.3.4 PDP Analysis:

Standard Deviation (Std) and **Absolute Differences (AbsDiffs)** show strong positive influence, where higher values lead to significantly higher prediction scores. This suggests that signal variability is a key indicator.

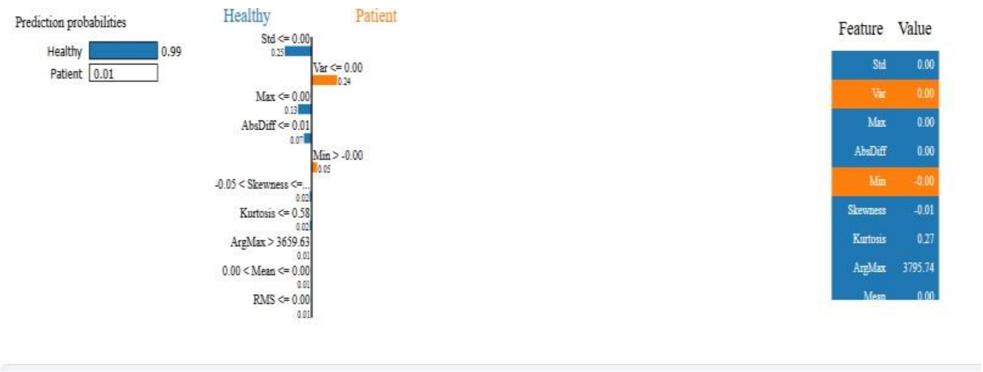
Skewness and **Kurtosis** both demonstrate **non-linear positive trends**, indicating that asymmetry and peakedness in the EEG waveform have diagnostic value.

Argmin/Argmax (positions of min/max in the signal) show a **mild negative trend**, suggesting early signal extrema may be more informative.

PTP (Peak-to-Peak) and **Min/Max amplitude** features also influence predictions, but with mixed or non-monotonic relationships.

Mean, Variance, Mean Square, and RMS were skipped or yielded flat outputs due to insufficient variance or numerical instability during PDP computation.

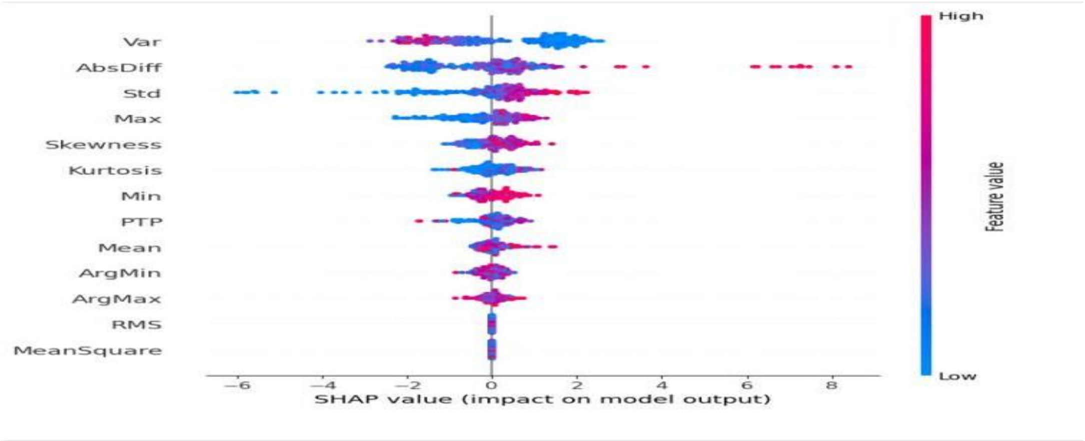
3.4 Applied 4 explainable AI model over AdaBoost



Feature	Value
Std	0.00
Var	0.00
Max	0.00
AbsDiff	0.00
Min	-0.00
Skewness	-0.01
Kurtosis	0.27
ArgMax	3795.74
Mean	0.00

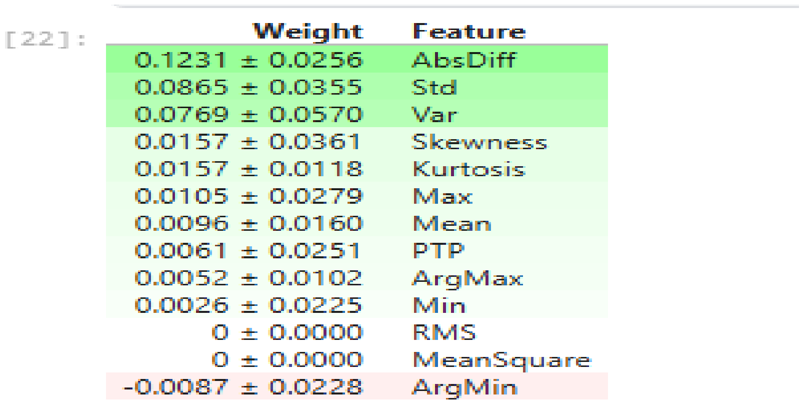
3.4.1 **LIME Analysis:**

- The LIME image shows Healthy classification with 99% probability.
- Key thresholds that influenced the classification include Std, Var, Min, Skewness, and ArgMax.
- The values of these features determined why the model strongly classified the sample as Healthy**.**



3.4.2 SHAP Analysis:

- Similar to XGBoost, the SHAP summary plot shows key features influencing predictions.
- Variance, AbsDiffs, and Std are the most significant features.
- The spread of SHAP values shows that some features have a stronger influence in AdaBoost compared to XGBoost.

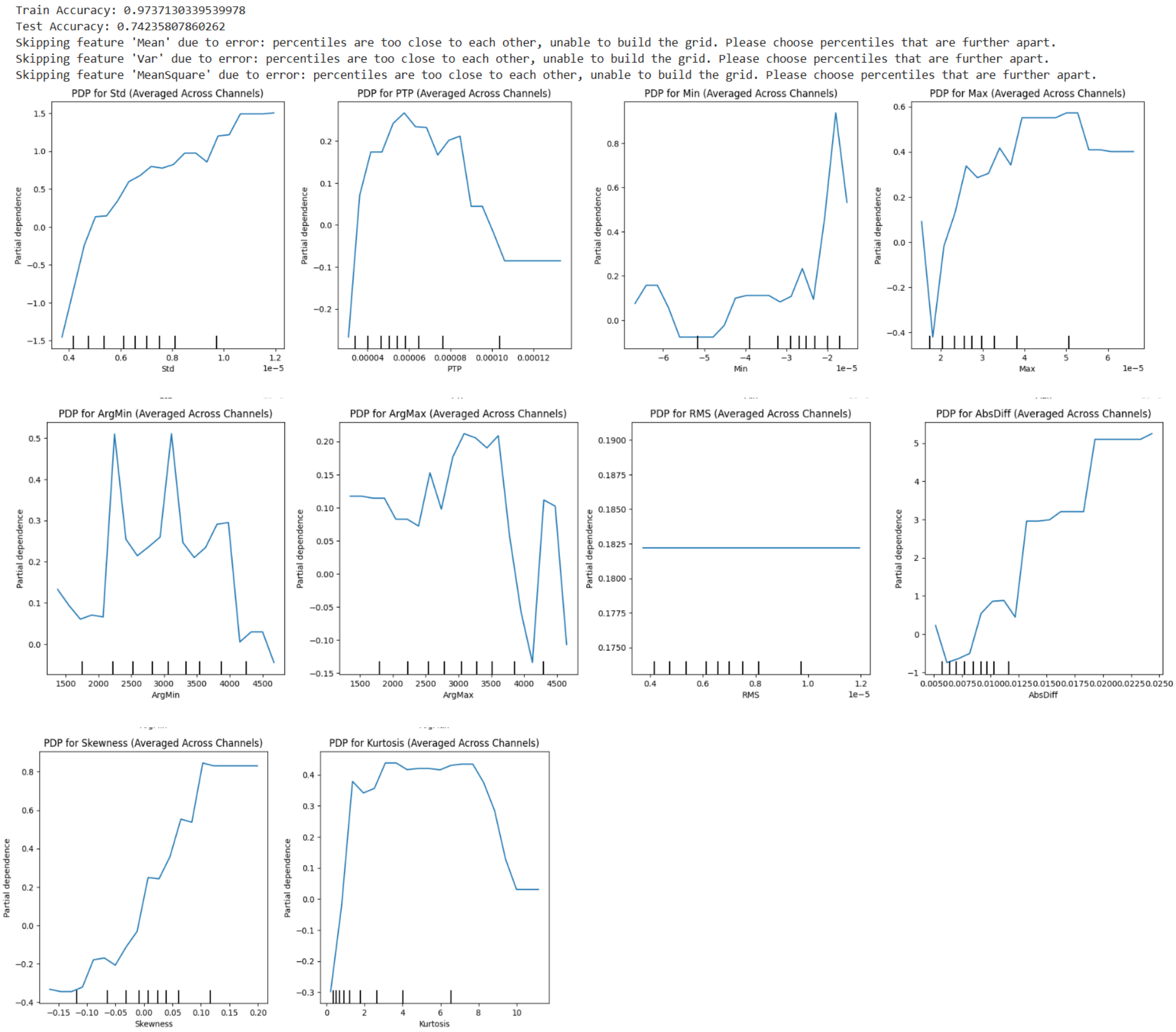


[22]:

Weight	Feature
0.1231 ± 0.0256	AbsDiff
0.0865 ± 0.0355	Std
0.0769 ± 0.0570	Var
0.0157 ± 0.0361	Skewness
0.0157 ± 0.0118	Kurtosis
0.0105 ± 0.0279	Max
0.0096 ± 0.0160	Mean
0.0061 ± 0.0251	PTP
0.0052 ± 0.0102	ArgMax
0.0026 ± 0.0225	Min
0 ± 0.0000	RMS
0 ± 0.0000	MeanSquare
-0.0087 ± 0.0228	ArgMin

3.4.3 ELI5 Analysis:

- AbsDiffs (0.1231), Std (0.0865), and Variance (0.0769) were the most impactful features.
- Unlike XGBoost, in AdaBoost, ArgMin had a slightly negative impact (-0.0087).
- Some features such as MeanSquare and RMS had no weight, meaning they did not contribute significantly.



```
Train Accuracy: 0.9737130339539978
Test Accuracy: 0.74235807860262
Skipping feature 'Mean' due to error: percentiles are too close to each other, unable to build the grid. Please choose percentiles that are further apart.
Skipping feature 'Var' due to error: percentiles are too close to each other, unable to build the grid. Please choose percentiles that are further apart.
Skipping feature 'MeanSquare' due to error: percentiles are too close to each other, unable to build the grid. Please choose percentiles that are further apart.
```

PDP for Std (Averaged Across Channels)

PDP for PTP (Averaged Across Channels)

PDP for Min (Averaged Across Channels)

PDP for Max (Averaged Across Channels)

PDP for ArgMin (Averaged Across Channels)

PDP for ArgMax (Averaged Across Channels)

PDP for RMS (Averaged Across Channels)

PDP for AbsDiff (Averaged Across Channels)

PDP for Skewness (Averaged Across Channels)

PDP for Kurtosis (Averaged Across Channels)

3.4.4 PDP Analysis:

Standard Deviation (Std) and **Absolute Differences (AbsDiff)** show strong **positive correlations**, indicating that greater signal variability leads to higher model confidence in predicting the target class.

Peak-to-Peak (PTP) shows a **non-monotonic effect**: predictions increase up to a point, then decrease—suggesting an optimal range of signal amplitude for class discrimination.

Minimum and Maximum Values reveal non-linear effects, with extremes (especially minimum) contributing more strongly to positive predictions.

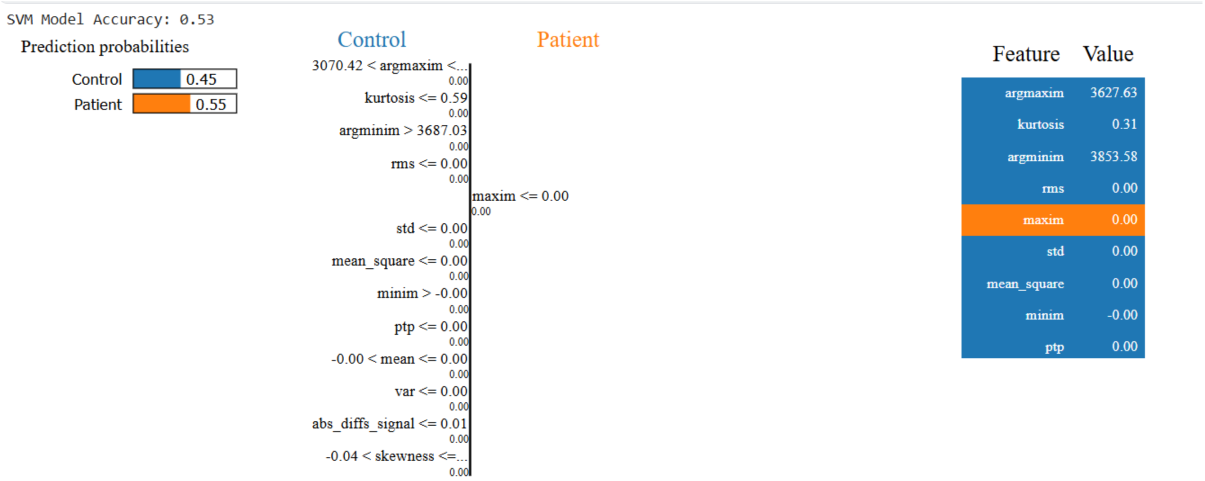
Skewness has a **clear increasing trend**, suggesting that asymmetrical signal shapes are significant indicators.

 Kurtosis shows a **peak plateau**, where moderate kurtosis values are most influential.

 Argmin/Argmax (temporal indices of min/max) show irregular effects but suggest some influence of signal timing.

 RMS and some features like **Mean, Variance, and MeanSquare** were skipped due to insufficient data variation, or yielded flat PDPs indicating minimal impact.

3.5 Applied 2 explainable AI model over SVM



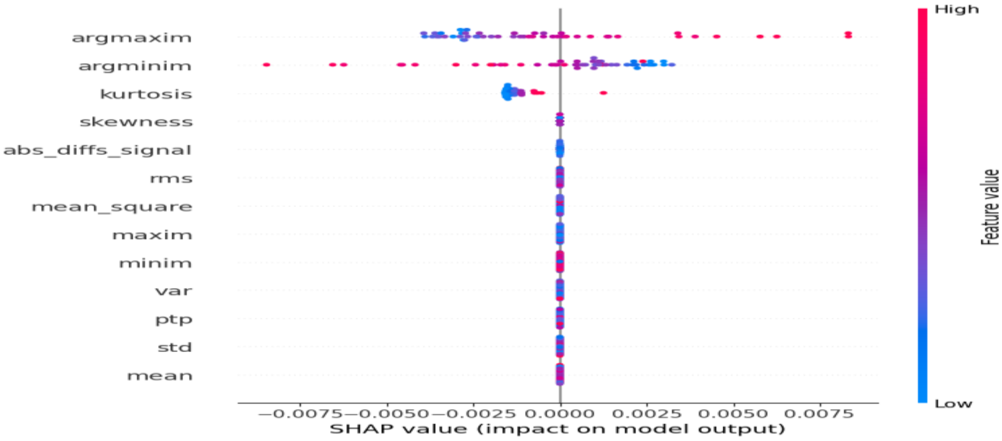
```
SVM Model Accuracy: 0.53
```

Feature	Value
argmaxim	3627.63
kurtosis	0.31
argminim	3853.58
rms	0.00
maxim	0.00
std	0.00
mean_square	0.00
minim	-0.00
ptp	0.00

3.5.1 LIME Analysis:

The vertical rule path shows the **decision path** based on feature thresholds. Each condition (like kurtosis <= 0.59) contributes incrementally toward classifying the instance as either **Control (left)** or **Patient (right)**.

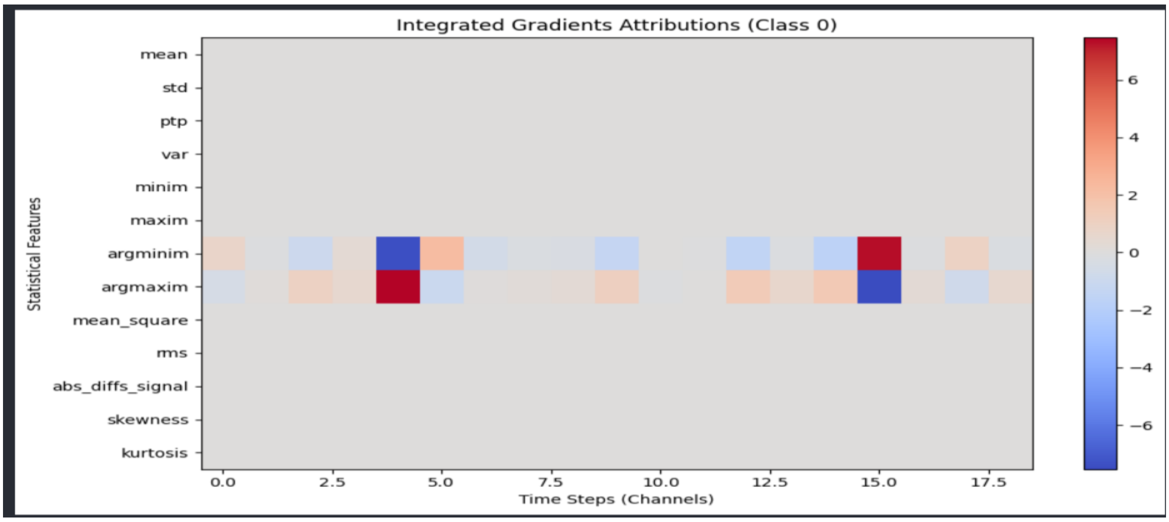
- **Contributing rules toward "Patient" class**:
  - argmaxim < 3070.42
  - kurtosis <= 0.59
  - argminim > 3687.03
  - Many other features like maxim, rms, std, mean_square, etc., are near zero and used as weak thresholds.

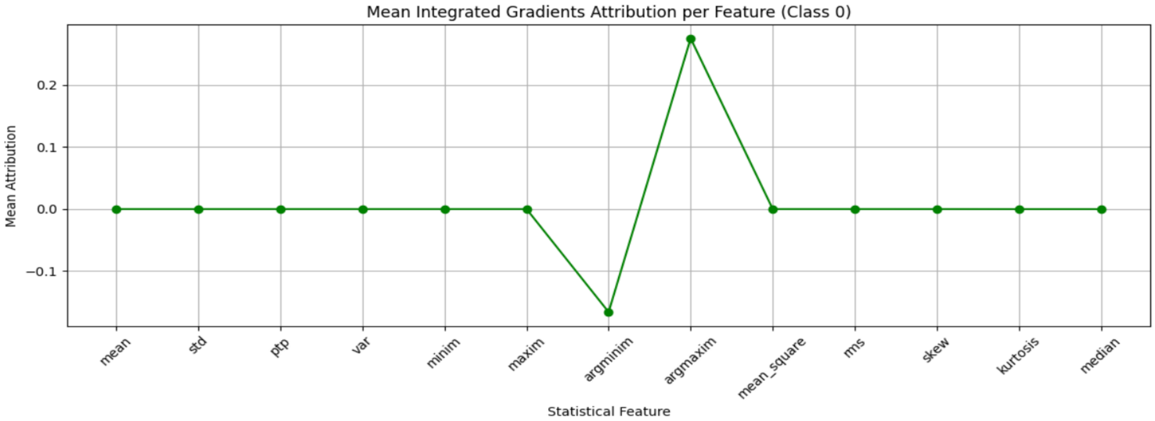


3.5.2 Shap Analysis:

Features like argmax and argmin have the widest spread of SHAP values, indicating they have the most significant impact on the model's predictions. Features like mean and std have narrower spreads, suggesting less influence.

3.6 Applied Integrated Gradients explainable AI model over SVM





- **argmin** and **argmax** show significant attribution, with strong positive (red) and negative (blue) contributions around time steps 5 to 10 and 15 to 17.5. This suggests that the positions of the minimum and maximum values in the time series are highly influential for the LSTM's prediction of Class 0.
- Features like mean, std, ptp, and var show minimal attribution (mostly neutral/light colors), indicating they have less impact on the model's decision for Class 0.

Time Step Relevance:

- The most significant attributions occur around time steps 5 to 10 and 15 to 17.5, where argmin and argmax have strong positive and negative impacts. This suggests the LSTM focuses on these time windows when making predictions for Class 0.
- Other time steps (e.g., 0 to 5 and 10 to 15) show less intense attributions, meaning the model pays less attention to those periods.

Positive vs. Negative Contributions:

- For argmin at time steps 5 to 10, the red color indicates that higher values of argmin (likely the time index of the minimum value in the series) push the model toward predicting Class 0.
- For argmax at time steps 15 to 17.5, the blue color indicates that higher values of argmax (the time index of the maximum value) push the model away from predicting Class 0.

4. Conclusion and Future Works

The analysis of the machine learning model for EEG-based classification of schizophrenia (Class 1) versus healthy controls (Class 0) reveals key insights into feature importance and model behavior. The model achieved an accuracy of 0.7336 with balanced precision, recall, and F1-scores (~0.73 for both classes), indicating reliable performance. SHAP and Integrated Gradients analyses highlight that argmax, argmin, kurtosis, and skewness are the most influential features, with argmax and argmin showing the highest impact on predictions, particularly at specific time steps (5–10 and 15–17.5). These features, reflecting the timing of peak values in EEG signals, significantly drive the model's decision for schizophrenia classification, as seen in patient-specific SHAP values (e.g., argmax = 3735.29, argmin = 3687.03 for a schizophrenia prediction with 0.57 probability). Partial Dependence Plots further confirm that higher skewness and kurtosis values increase the likelihood of schizophrenia prediction, while argmax and argmin exhibit non-linear effects. Feature importance rankings consistently place abs_diff_signal, maxim, and ptp as top contributors to model output magnitude. Collectively, these findings suggest that the model effectively captures temporal and statistical patterns in EEG data, particularly the timing and shape of signal distributions, to differentiate schizophrenia patients from healthy controls, providing a robust foundation for clinical decision support.

In summary, the LSTM outperforms the SVM in EEG-based schizophrenia classification, effectively leveraging temporal and statistical features like argmax, argmin, kurtosis, and skewness to achieve higher accuracy and interpretability. These findings underscore the importance of temporal dynamics in EEG data for schizophrenia detection and provide a foundation for developing reliable clinical diagnostic tools. Future work should focus on optimizing the SVM model or exploring hybrid approaches to enhance performance.

4.1 Future work:

- **Expanding Data Volume:** Focus on acquiring and incorporating more diverse and extensive EEG datasets. Working with larger datasets can significantly improve model robustness and performance, enabling the identification of more nuanced patterns associated with schizophrenia and other disorders.

- **Data Augmentation:** Investigate advanced data augmentation techniques specific to EEG signals to increase the dataset size artificially, improving the model's generalization ability without the need for additional data collection.

- Broader application of XAI methods across various machine learning and deep learning models for improved diagnostic precision:

  different explainable AI models like SHAP and LIME can be applied over different machine learning models like classification, clustering, and different deep learning models like CNN, SVM, Decision Tree to get the specific feature which is contributing to the prediction.

5. REFERENCES

1. Explainable Artificial Intelligence Model for Stroke Prediction Using EEG Signal by Mohammed Saidul Islam1,Iqram Hussain2,3,,Md Mezbaur Rahman1,Se Jin Park4 and Md Azam Hossain
2. https://shap.readthedocs.io/en/latest/index.html
3. https://www.youtube.com/watch?v=OZJ1IgSgP9E&list=PLV8yxwGOxvvovp-j6ztxhF3QcKXT6vORU
4. Ahmad ,Chaddad,1,2,* Yihang Wu,1 Reem Kateb,3 and Ahmed Bouridane4 Chang-Hwan Im, Academic Editor and Yvonne Tran, Academic Editor, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10385593/
5. Graph-based analysis of brain connectivity in schizophrenia
6. Elzbieta Olejarczyk 1, Wojciech Jernajczyk 2
7. https://github.com/talhaanwarch/youtube-tutorials
8. https://www.youtube.com/watch?v=OZJ1IgSgP9E&list=PLV8yxwGOxvvovp-j6ztxhF3QcKXT6vORU
9. Smith K. Kharea, U. Rajendra Acharya,https://www.sciencedirect.com/science/article/pii/S001048252

3001415

10. Smith K. Khare , U. Rajendra Acharya, https://www.sciencedirect.com/science/article/pii/S0950705123006081
11. Application of Explainable Artificial Intelligence in Alzheimer's Disease Classification: A Systematic Review