

Name: Nefeli Eleftheria Sextou

Student ID: 503

E-mail: pcs00503@uoi.gr

5 Feb 2024

MY D8: Biomedical Data Processing and Analysis

Brief Project Report



Project Goal:

The goal is to clean, process and analyze fetal heart rate (FHR) signals included in the CTU-CHB Cardiotocography Database and proceed to utilize the produced data as input in three different classifiers and evaluate the outcomes.

The Dataset:

The dataset includes 552 CTGs (cardiotocography data) from the Czech Technical University (CTU, Prague) and the Hospital of Brno (UHB).

It can be found here: <https://physionet.org/content/?topic=ctu>

Information about the data included is found in the paper that accompanies the dataset :

<https://bmcpregnancychildbirth.biomedcentral.com/counter/pdf/10.1186/1471-2393-14-16.pdf>

Implementation Details:

All code has been written in Python on Jupyter Notebooks that accompany the present report in both **.ipynb** and **html** form in folder **biomed_project_code**. The data is in folder **biomed_project_data**.

Data Cleaning and Pre-Processing:

data_loading\ data_loading_interp.ipynb (.html)

Dataframe Preparation:

dataframe_creation\ age_dataframes.ipynb (.html)

dataframe_creation\ weight_dataframes.ipynb (.html)

dataframe_creation\ term_dataframes.ipynb (.html)

dataframe_creation\ ph_dataframes.ipynb (.html)

Classification using different feature selectors:

Linear SVC Feature Selector:

svc_feature_slc_classification\age_lin_svc.ipynb (.html)

svc_feature_slc_classification\weight_lin_svc.ipynb (.html)

svc_feature_slc_classification\term_lin_svc.ipynb (.html)

svc_feature_slc_classification\ph_lin_svc.ipynb (.html)

ExtraTreesClassifier Feature Selector:

tree_feature_slc_classification\age_tree.ipynb (.html)

tree_feature_slc_classification\weight_tree.ipynb (.html)

tree_feature_slc_classification\term_tree.ipynb (.html)

tree_feature_slc_classification\ph_tree.ipynb (.html)

Data Cleaning and Processing:

The first thing that had to be done was the loading of the data. The data is in .dat and .hea form which required the **Python waveform-database (WFDB) package** (<https://wfdb.readthedocs.io/en/latest/>)

After loading all available data onto a Pandas Dataframe it was necessary to:

1. Isolate the FHR Signal
2. Remove Caesarian Birth Entries
 - a. This was achieved by deleting all entries with a value of 2.0 in the field 'Deliv.type'
3. Extract Features From the 'Comments' feature since it contains features that will be necessary.

FHR Split into Stage I and Stage II:

The FHR signal must be split into two parts since it should not be analyzed in its full form since the two parts it will be split into are representative of two stages of labor.

- **Stage I** has been limited by the creators of the database to a maximum of 60 minutes ($60(\text{mins}) * 60(\text{secs}) * 4(\text{fs}) = 14000$ samples) and a minimum of 30 minutes ($30(\text{mins}) * 60(\text{secs}) * 4(\text{fs}) = 7200$ samples)
- **Stage II** has likewise been constrained to a maximum of 30 minutes.

The split was achieved utilizing the provided starting point of stage II in the feature 'Pos.II.st.'.

Removal of Zeros:

The signals have zeros which are the result of noise removal done by the dataset creators. Signals with way too many zeros must be discarded and the zeros must be handled in the remaining signals in the dataset.

The statistics of the number of zeros, as well as the mean zeros to total signal length ratio were examined for each stage separately:

Stage I Zero Statistics and Mean Zero-Signal Length Ratio:

```
mean number of zeros = 2172.4703557312255
median number of zeros = 1927.5
standard deviation of number of zeros = 1702.518355583135
min number of zeros = 0
max number of zeros = 7812
```

```
mean zero to signal I length ratios: [0.09998962 0.10259208 0.10346216 0.11073768 0.11130038 0.11491683
 0.11898148 0.12402677 0.12497569 0.12803906 0.13182191 0.13237415
 0.13245602 0.13385417 0.13386346]
counts: [ 1  1  1  1  1  1  1  1  1  1  1  1  2 491  1]
```

Stage II Zero Statistics and Mean Zero-Signal Length Ratio:

```
mean number of zeros = 960.9762845849802
median number of zeros = 699.5
standard deviation of number of zeros = 986.9752227622885
min number of zeros = 0
max number of zeros = 6650
```

```
mean zero to signal II lenght ratios: [1.33099208e-01 1.33136088e-01 1.33172988e-01 1.33468928e-01
1.59657133e-01 1.59736749e-01 1.60162714e-01 1.99372673e-01
1.99538265e-01 2.00203393e-01 2.65463062e-01 2.65756716e-01
2.66937857e-01 3.97097638e-01 4.00406785e-01 7.87685479e-01
8.00813570e-01 2.00203393e+00 9.60976285e+02]
counts: [ 1  1  1 36  1  1 28  4  1 70  2  1 81  2 159  2 110  1
4]
```

Looking at the statistics and ratios for Stage I signals, it is evident that up to 15% of the signal is made up of zeros, which is not that significant and can be countered with some form of interpolation.

With regard to the entries' Stage II data, there exist signals with a significant amount of zeros with a ratio that would amount to 80% of the signal. Some cases that are concerning include: 110 signals with a ratio equivalent to roughly 80%, 2 signals with an approximate ratio of 78%, 159 signals with a ratio of about 40% and 2 signals with a ratio of about 39%. Another odd result apparent in the data is the 4 counts of 9.60976285e+02.

The entries that correspond to the 'concerning' cases described above are discarded (the entire entry/row, not just the Stage II feature).

It must be noted that the mean was chosen for the ratio given it is sensitive to outliers which *must* be included in this scenario.

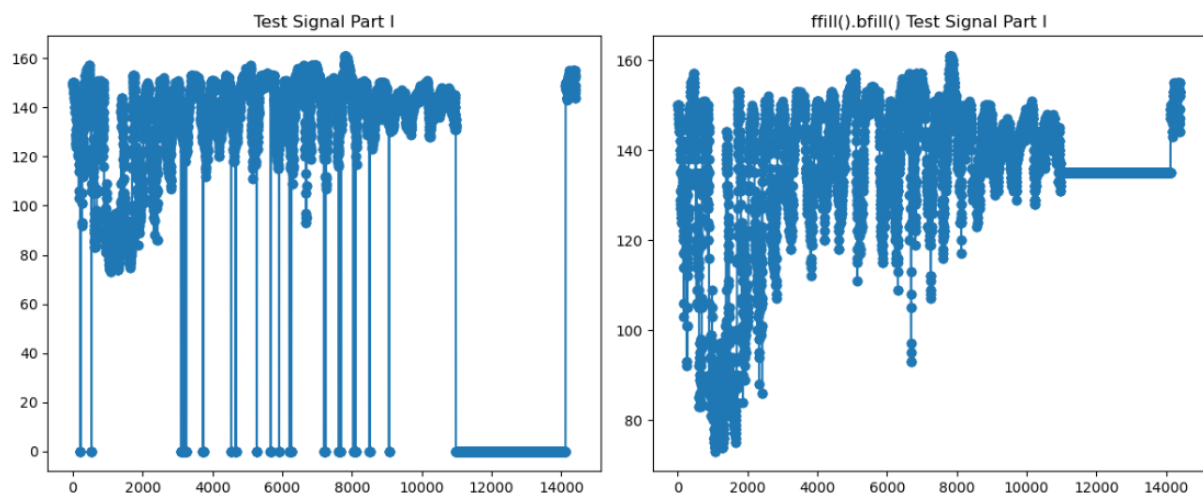
The rest of the remaining entries are interpolated using Forward and Backward filling.

Zeros are transformed into nan values that the Pandas replace() function can locate and treat as "missing", and then a Forward Fill pass is followed by a Backward Fill pass to ensure all values have been filled.

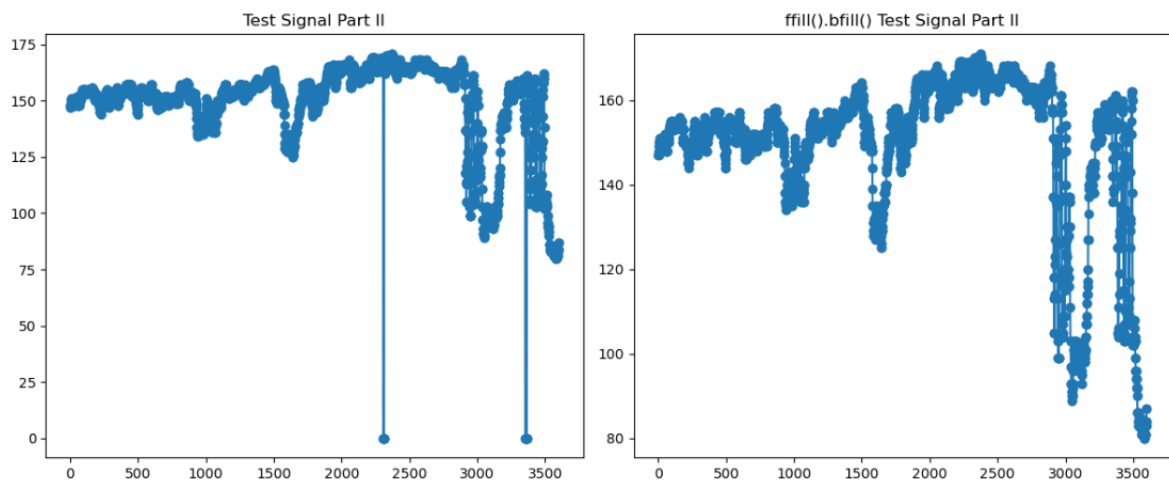
In the Forward Fill pass the "missing" values are replaced with the most recent "non-missing" values before them. In the Backward Fill pass, the "missing" values are replaced with the most recent "non-missing" values after them. The results are very similar to a simple linear interpolation applied on these parts.

Continuous areas of zeros were not removed, they were filled with a corresponding valid value as described above.

Example of the Interpolation of a Stage I Signal:



Example of the Interpolation of a Stage II Signal:



It is now possible to move on to feature extraction from the two parts of the FHR signals independently.

FHR Feature Extraction:

Specific features must be extracted from the FHR signal. The FHR signal contains heart rate variability (HRV) information. There are several types of methods that help extract features from such a signal and they include time domain methods, frequency domain methods, geometric methods, time-frequency methods and non-linear methods.

The features that will be extracted correspond to the frequency domain, time domain, time-frequency domain and some entropy based non-linear methods.

Time Domain Features: mean and median NN intervals (mean_nni, median_nni), SDNN, RMSSD, NNI20, PNNI20

Frequency Domain Features: LF/HF Ratio, Total Power, VLF

Time-Frequency Features: Haar Wavelet St. Deviation, Mean and Shannon Entropy

Entropy Features: Sample Entropy, Bubble Entropy and Shannon Entropy

An experimental set of features was added that quantifies the difference of some of the feature values of the categories above. The choice was arbitrary. If these features offer good information, they will be retained in the feature selection stage.

Difference Features: the absolute difference of nni20, LF/HF and of the Haar Wavelet St.Dev.

Note about the Haar Wavelet: The Haar wavelet is applied with multiple scale settings. The goal is to choose the scale that provides a min median case for all signals in each stage and then extract more information from the detail coefficient vector of that scale. The **optimal scale** is found to be **1** for both stages independently.

The features utilizing the Shannon entropy had to be removed because they resulted in **-inf** in all entries.

After this a “main” dataframe was created by retaining the features in the screenshot on the next page. The Shannon entropy features are commented out.

The 'BDecf', 'pCO2', 'BE' columns contained a few null values. The corresponding entries were removed.

```

: to_keep = ['pH',
'BDecf',
'pCO2',
'BE',
'Apgar1',
'Apgar5',
'Gest.weeks',
'Weight(g)',
'Sex',
'Age',
'Gravidity',
'Parity',
'Diabetes',
'Hypertension',
'Preeclampsia',
'Liq.praecox',
'Pyrexia',
'Meconium',
'Presentation',
'Induced',
'NoProgress',
'CK/KP',
'dbID',
'Rec.type',
'FHR_I_ffill_mean_nni',
'FHR_I_ffill_median_nni',
'FHR_I_ffill_sdn',
'FHR_I_ffill_rmssd',
'FHR_I_ffill_nni20',
'FHR_I_ffill_pnni20',
'FHR_I_ffill_lf_hf_ratio',
'FHR_I_ffill_total_power',
'FHR_I_ffill_vlf',
'FHR_I_ffill_haar_stddev',

```

```

'FHR_I_ffill_haar_mean',
#'FHR_I_ffill_haar_shan_entr',
'FHR_I_ffill_samp_entr',
'FHR_I_ffill_bub_entr',
#'FHR_I_ffill_shan_entr',
'FHR_II_ffill_mean_nni',
'FHR_II_ffill_median_nni',
'FHR_II_ffill_sdn',
'FHR_II_ffill_rmssd',
'FHR_II_ffill_nni20',
'FHR_II_ffill_pnni20',
'FHR_II_ffill_lf_hf_ratio',
'FHR_II_ffill_total_power',
'FHR_II_ffill_vlf',
'FHR_II_ffill_haar_stddev',
'FHR_II_ffill_haar_mean',
#'FHR_II_ffill_haar_shan_entr',
'FHR_II_ffill_samp_entr',
'FHR_II_ffill_bub_entr',
#'FHR_II_ffill_shan_entr',
'diff_nni20',
'diff_lf_hf',
'diff_haar_std']

```

Data Preparation and Class Creation:

There are four target variables for which classification will be applied: the mother's age (feature: 'Age'), the neonate's weight (feature: 'Weight(g)'), the gestational term in weeks (feature: 'Gest.weeks') and the pH (feature: 'pH'). Classes must be created for each of the above.

'Maternal Age' Classes for the Mother's Age:

The classes for the mother's age are, initially, not chosen arbitrarily but based on methodology used on a well cited paper about the relation of maternal age to adverse pregnancy outcomes provided in the link below:

<https://bmcpregnancychildbirth.biomedcentral.com/articles/10.1186/s12884-019-2400-x>

Classes:

A: ≤ 17 years old

B: 18-28 years old

C: 29-39 years old

D: ≥ 40 years old

However, this resulted in an imbalanced dataset so the problem was turned into one of binary classification with classes:

A: ≤ 29 years old

B: > 29 years old

‘Neonate Weight’ Classes for the Neonate’s Weight:

The initial classes were chosen with reference to:

- The following resource that is partnered with some world renowned medical schools (Yale, UCSF, UCSD and more):
<https://emedicine.medscape.com/article/938854-overview?form=fpf>
- The Mayo Clinic : <https://www.mayoclinic.org/diseases-conditions/fetal-macrosomia/symptoms-causes/syc-20372579>
- Penn Medicine Lancaster General Health:
<https://www.lancastergeneralhealth.org/health-hub-home/motherhood/the-first-year/your-newborns-weight-gain>

Neonate Weight is classified as follows:

ELBW: Extremely low birth weight -> weight < 1000 g

VLBW: Very low birth weight -> weight < 1500 g

LBW: Low birth weight -> weight < 2500 g

NORM: Normal birth weight -> weight ≥ 2500 g and ≤ 4000 g

MACRO: Macrosomia -> weight > 4000 g

The ELBW and VLBW did not exist in my dataset, and the remaining three classes were imbalanced. The new classes that were created were the following:

ABNORM: Low birth weight -> weight < 2500 g

(includes abnormal ELBW, VLBW and LBW cases)

NORM: Normal birth weight -> weight ≥ 2500 g and ≤ 4000 g

ABNORM: Macrosomia -> weight > 4000 g

(includes MACRO abnormal cases)

‘Gest Term’ Classes for the Gestational Term in Weeks:

According to The American College of Obstetricians and Gynecologists (ACOG) : <https://www.acog.org/clinical/clinical-guidance/committee-opinion/articles/2013/11/definition-of-term-pregnancy>, pregnancy terms can be classified in weeks as follows:

ET: Early Term -> weeks ≥ 37 and ≤ 38

FT: Full Term -> weeks ≥ 39 and ≤ 40

LT: Late Term -> weeks ≥ 41

P: Postterm -> weeks ≥ 42

While all four classes appeared, there was prominent class imbalance. To find a good midway point between interpretability and class balance, the 'FT' class can remain the same and the rest of the classes can be merged as 'OT' (other). So, the two categories will be full term pregnancies and 'other' pregnancies where some type of pathological phenomenon may exist.

OT: Other -> weeks ≥ 37 and ≤ 38

FT: Full Term -> weeks ≥ 39 and ≤ 40

OT: Other -> weeks ≥ 41

OT: Other -> weeks ≥ 42

‘pH risk Classes for the pH:

In order to create two classes for the pH feature, it is necessary to find a threshold value. Many contesting scientific opinions on the value of this threshold exist and with regard to which ranges of values may indicate what pathological phenomenon or risk indicator. The ideal threshold value is the subject of ongoing research and there is no agreement on a threshold value that may be used to discern between normal and abnormal situations but those proposed are usually smaller than 7.2.

The approach followed here was the following:

The instances of unique pH values were counted:

```
[6.87 6.92 6.93 6.98 6.99 7.    7.02 7.03 7.05 7.07 7.08 7.09 7.1  7.11
 7.12 7.13 7.14 7.15 7.16 7.17 7.18 7.19 7.2  7.21 7.22 7.23 7.24 7.25
 7.26 7.27 7.28 7.29 7.3  7.31 7.32 7.33 7.34 7.35]
[ 1  1  2  2  1  2  1  3  3  3  2  1  4  2  5  9 11  3 15  4 11  8 14  7
 6  7 13 12 15  8  6  9  9  6  7  1  5  3]
```

Then, they were grouped and total sums per group were calculated:

{6.87 6.92 6.93 6.98 6.99 7.} -> total count = 9

{7.02 7.03 7.05 7.07 7.08 7.09 7.1} -> total count = 17

{7.11 7.12 7.13 7.14 7.15 7.16 7.17 7.18 7.19 7.2}-> total count = 82

{7.21 7.22 7.23 7.24 7.25 7.26 7.27 7.28 7.29 7.3} -> total count = 92

{7.31 7.32 7.33 7.34 7.35} -> total count = 22

Looking at the existing pH values and their corresponding count values, there are two good split options that will lead to balanced classes. They can be considered equivalent. The first is using 7.2 as the splitting point for the two classes (class A : 108 elements, class B: 114 elements). The second is using 7.21 as the splitting point for the two classes (class A: 115 elements, class B:107 elements).

Instead of class A and B we can name the classes Risky 'RISK' and Non-Risky 'NORISK' given the fact there may be loose correlation to CTG data and pH values that indicate asphyxiation of the neonate or other pathological phenomena.

Given the dataset information and the information available in the paper that accompanies the initial dataset where a lot of clinical details are briefly explained, 7.2 is chosen as the split point.

Classification:

All implementations mentioned refer to existing python implementations that are well established and widely used.

The three classifiers that will be tried are the:

- K-Nearest Neighbors Classifier
- Random Forest Classifier
- XGBoost XGBClassifier

The features are selected using two different methods:

- Linear SVC based: An application of Linear Support Vector classification with L1 regularization as a penalty term that discourages the use of unnecessary features in the model. Features with non-zero coefficients are considered important in making predictions, while features with zero coefficients are ignored.
- ExtraTreesClassifier based: It builds a set of multiple decision trees through bootstrapping, where each tree is trained on a random subset of the training data. Features are randomly selected for splitting each node in a decision tree and this contributes to feature decorrelation. The results of the tree predictions are aggregated to achieve classification.

Data is split in 80% train and 20% test.

Metrics used to evaluate the results:

- The Confusion Matrix
- Accuracy = $(TP + TN) / (TP + TN + FP + FN)$
- Precision = $(TP) / (TP + FP)$
- Recall = $(TP) / (TP + FN)$
- F1-Score = $2 * ((Precision * Recall) / (Precision + Recall))$
- ROC Curve :
True Positive Rate(Recall) – False Positive Rate($FP / (FP + TN)$) and AUC

*higher values closer to 1.0 are better for all

For XGBClassifier, one extra evaluation method was utilized:

The **SHAP (SHapley Additive exPlanations)** is a Python library designed for interpreting the output of machine learning models. It provides a framework for understanding the contributions of each feature in model predictions.

<https://shap.readthedocs.io/en/latest/index.html>

It is based on Shapely Values which are a concept from cooperative game theory that has found application in machine learning model interpretability. (https://en.wikipedia.org/wiki/Shapley_value)

In brief:

In cooperative game theory, players form coalitions (subsets) to achieve a collective outcome. When used in machine learning, the players are the features, and the coalitions are subsets of features.

The Shapley values calculate each player's/feature's marginal contribution to every possible coalition it could join.

marginal contribution = (the model's prediction with a feature in the coalition) – (the model's prediction with a feature NOT in the coalition)

They then take the average of the marginal contributions over all possible coalitions. In this manner, each feature is attributed a fair share of the total contribution and all possible interactions with other features are considered.

Main lines of interpretation:

Positive Shapley Value: the presence of that feature contributes positively to the model's prediction.

Negative Shapley Value: the absence of that feature contributes positively to the model's prediction.

The magnitude reflects the strength of the contribution. The larger the magnitude, the more significant the feature's presence is for the model's output.

Linear SVC Feature Selection Results:

Maternal Age

```
Index(['BE', 'Weight(g)', 'FHR_I_ffill_median_nni', 'FHR_I_ffill_sdn',  
      'FHR_I_ffill_nni20', 'FHR_I_ffill_total_power', 'FHR_I_ffill_vlf',  
      'FHR_II_ffill_nni20', 'FHR_II_ffill_total_power', 'FHR_II_ffill_vlf',  
      'diff_nni20'],  
      dtype='object')
```

Neonate Weight

```
Index(['FHR_I_ffill_mean_nni', 'FHR_I_ffill_total_power', 'FHR_II_ffill_nni20',  
      'FHR_II_ffill_total_power', 'diff_nni20'],  
      dtype='object')
```

Gest Term

```
Index(['BDecf', 'Weight(g)', 'Age', 'FHR_I_ffill_mean_nni',  
      'FHR_I_ffill_median_nni', 'FHR_I_ffill_total_power',  
      'FHR_II_ffill_median_nni', 'FHR_II_ffill_sdn',  
      'FHR_II_ffill_total_power', 'FHR_II_ffill_vlf'],  
      dtype='object')
```

pH risk

```
Index(['pCO2', 'BE', 'Weight(g)', 'FHR_I_ffill_nni20', 'FHR_II_ffill_mean_nni',  
      'FHR_II_ffill_total_power', 'FHR_II_ffill_vlf', 'diff_nni20'],  
      dtype='object')
```

ExtraTreesClassifier Feature Selection Results:

Maternal Age

```
Index(['Apgar1', 'Gest.weeks', 'Weight(g)', 'Gravidity', 'Parity',  
      'Liq.praecox', 'dbID', 'FHR_I_ffill_mean_nni', 'FHR_I_ffill_median_nni',  
      'FHR_I_ffill_sdnn', 'FHR_I_ffill_rmssd', 'FHR_I_ffill_pnni20',  
      'FHR_I_ffill_lf_hf_ratio', 'FHR_I_ffill_total_power',  
      'FHR_I_ffill_haar_stddev', 'FHR_I_ffill_samp_entr',  
      'FHR_I_ffill_bub_entr', 'FHR_II_ffill_sdnn', 'FHR_II_ffill_nni20',  
      'FHR_II_ffill_lf_hf_ratio', 'FHR_II_ffill_total_power',  
      'FHR_II_ffill_samp_entr', 'FHR_II_ffill_bub_entr', 'diff_nni20',  
      'diff_lf_hf', 'diff_haar_std'],  
      dtype='object')
```

Neonate Weight

```
Index(['pH', 'pCO2', 'BE', 'Apgar5', 'Age', 'Gravidity', 'Pyrexia', 'Induced',  
      'FHR_I_ffill_median_nni', 'FHR_I_ffill_rmssd', 'FHR_I_ffill_pnni20',  
      'FHR_I_ffill_total_power', 'FHR_I_ffill_haar_stddev',  
      'FHR_I_ffill_bub_entr', 'FHR_II_ffill_sdnn', 'FHR_II_ffill_rmssd',  
      'FHR_II_ffill_nni20', 'FHR_II_ffill_pnni20', 'FHR_II_ffill_lf_hf_ratio',  
      'FHR_II_ffill_total_power', 'FHR_II_ffill_haar_stddev',  
      'FHR_II_ffill_samp_entr', 'diff_lf_hf'],  
      dtype='object')
```

Gest Term

```
Index(['pH', 'BDecf', 'pCO2', 'BE', 'Apgar5', 'Weight(g)', 'Age',  
      'Liq.praecox', 'Induced', 'dbID', 'FHR_I_ffill_sdnn',  
      'FHR_I_ffill_rmssd', 'FHR_I_ffill_lf_hf_ratio', 'FHR_I_ffill_haar_mean',  
      'FHR_I_ffill_bub_entr', 'FHR_II_ffill_median_nni', 'FHR_II_ffill_sdnn',  
      'FHR_II_ffill_lf_hf_ratio', 'FHR_II_ffill_total_power',  
      'FHR_II_ffill_vlf', 'FHR_II_ffill_samp_entr', 'diff_nni20'],  
      dtype='object')
```

pH risk

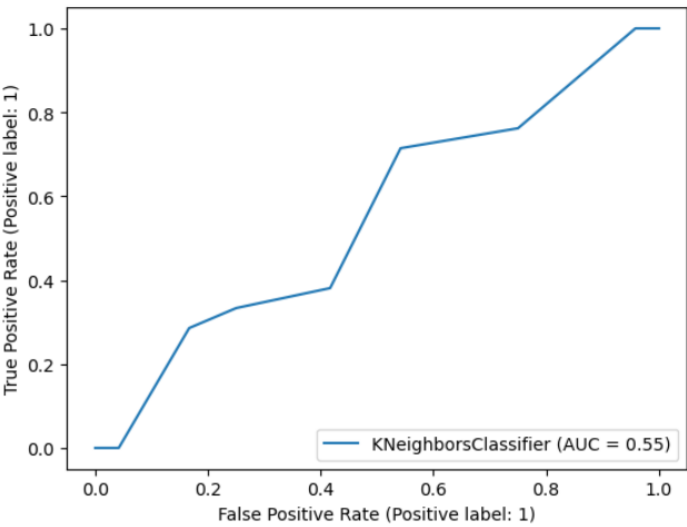
```
Index(['BDecf', 'pCO2', 'BE', 'Apgar1', 'Apgar5', 'Rec.type',  
      'FHR_II_ffill_sdnn'],  
      dtype='object')
```

Maternal Age (Linear SVC) Results

KNN

Accuracy: 0.5555555555555556
Confusion Matrix:
[[18 6]
 [14 7]]
Classification Report:

	precision	recall	f1-score	support
0	0.56	0.75	0.64	24
1	0.54	0.33	0.41	21
accuracy			0.56	45
macro avg	0.55	0.54	0.53	45
weighted avg	0.55	0.56	0.54	45



Accuracy for all folds: [0.6 0.52272727 0.45454545 0.54545455 0.52272727]
Mean Accuracy: 0.5290909090909091
F1 Score for all folds: [0.47058824 0.43243243 0.45454545 0.28571429 0.22222222]
Mean F1 Score: 0.3731005260417025
Precision for all folds: [0.57142857 0.47058824 0.41666667 0.5 0.42857143]
Mean Precision: 0.47745098039215683
Recall for all folds: [0.57142857 0.47058824 0.41666667 0.5 0.42857143]
Mean Recall: 0.47745098039215683

Random Forest

Accuracy: 0.5777777777777777

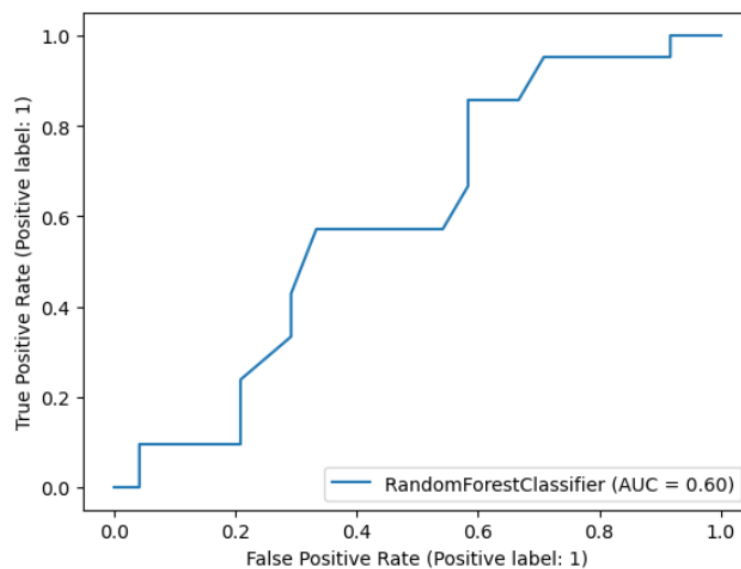
Confusion Matrix:

```
[[17  7]
```

```
 [12  9]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.71	0.64	24
1	0.56	0.43	0.49	21
accuracy			0.58	45
macro avg	0.57	0.57	0.56	45
weighted avg	0.58	0.58	0.57	45



Accuracy for all folds: [0.55555556 0.47727273 0.52272727 0.52272727 0.61363636]

Mean Accuracy: 0.5383838383838384

F1 Score for all folds: [0.44444444 0.41025641 0.4 0.32258065 0.4137931]

Mean F1 Score: 0.39821492066208414

Precision for all folds: [0.5 0.42105263 0.46666667 0.45454545 0.66666667]

Mean Precision: 0.501786283891547

Recall for all folds: [0.5 0.42105263 0.46666667 0.45454545 0.66666667]

Mean Recall: 0.501786283891547

XGBClassifier

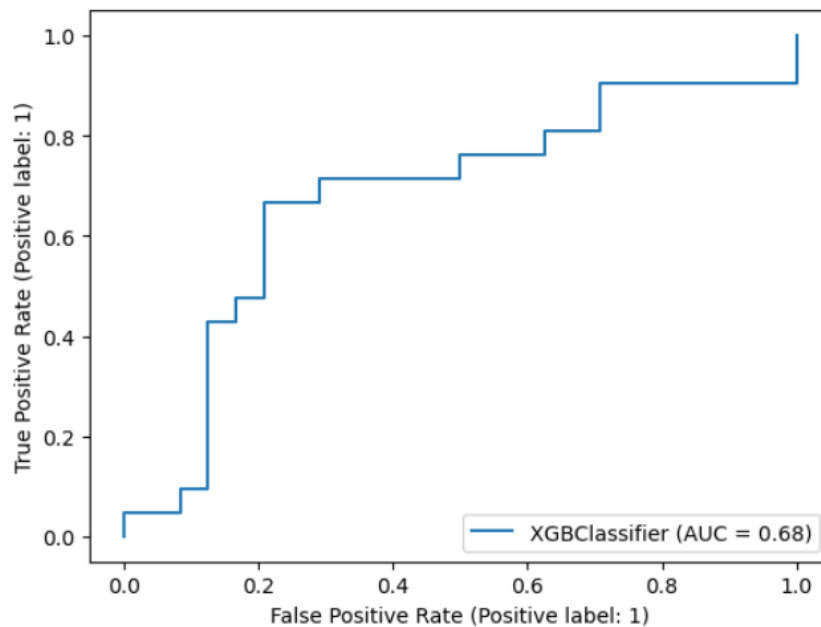
Accuracy: 0.6444444444444445

Confusion Matrix:

```
[[19  5]
 [11 10]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.63	0.79	0.70	24
1	0.67	0.48	0.56	21
accuracy			0.64	45
macro avg	0.65	0.63	0.63	45
weighted avg	0.65	0.64	0.63	45



Accuracy for all folds: [0.48888889 0.43181818 0.47727273 0.56818182 0.56818182]

Mean Accuracy: 0.5068686868686869

F1 Score for all folds: [0.48888889 0.41860465 0.48888889 0.42424242 0.48648649]

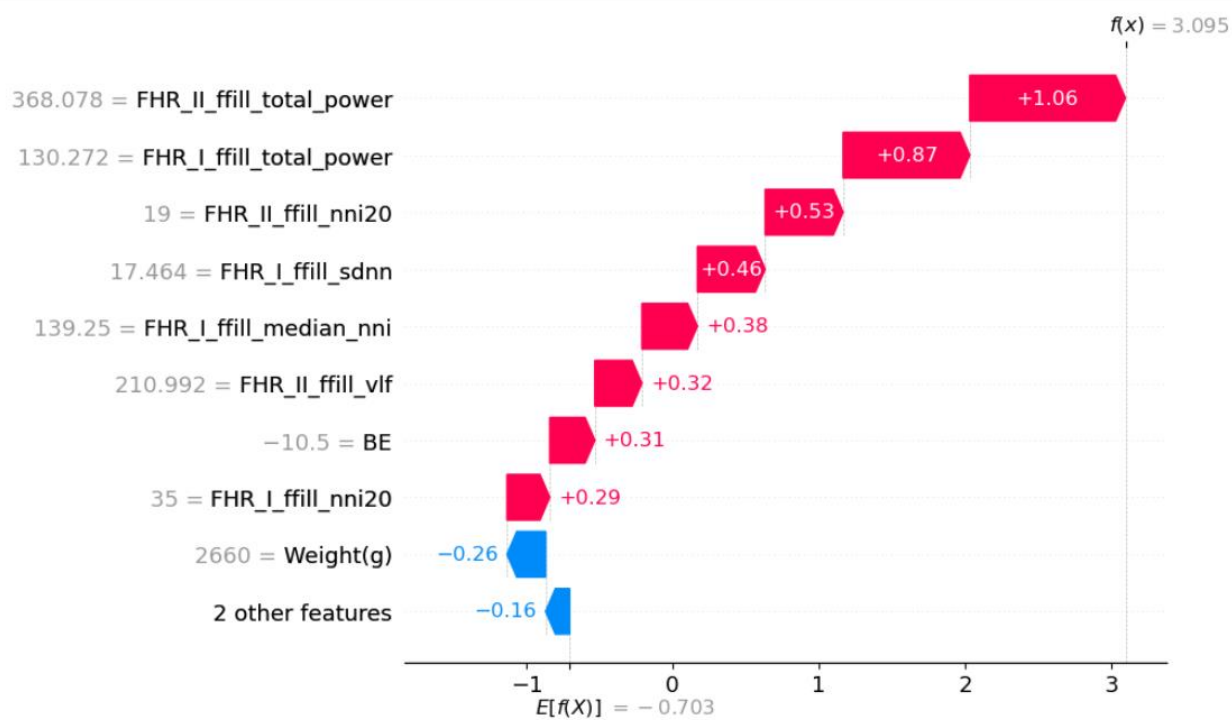
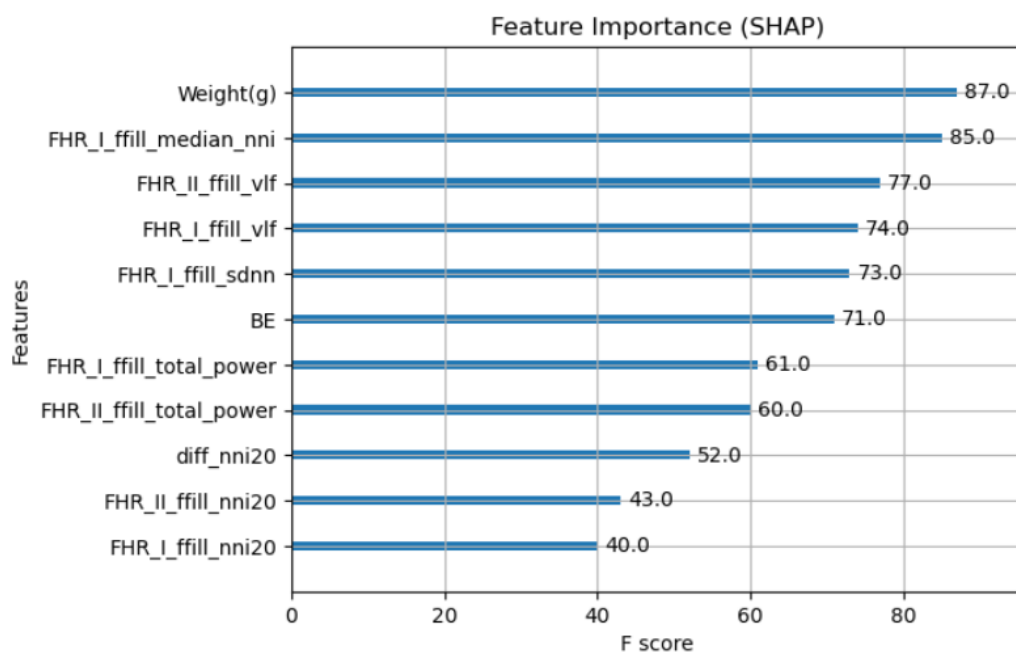
Mean F1 Score: 0.46142226793389585

Precision for all folds: [0.44 0.39130435 0.44 0.53846154 0.52941176]

Mean Precision: 0.46783553019870155

Recall for all folds: [0.44 0.39130435 0.44 0.53846154 0.52941176]

Mean Recall: 0.46783553019870155



Neonate Weight (Linear SVC) Results

KNN

Accuracy: 0.8444444444444444

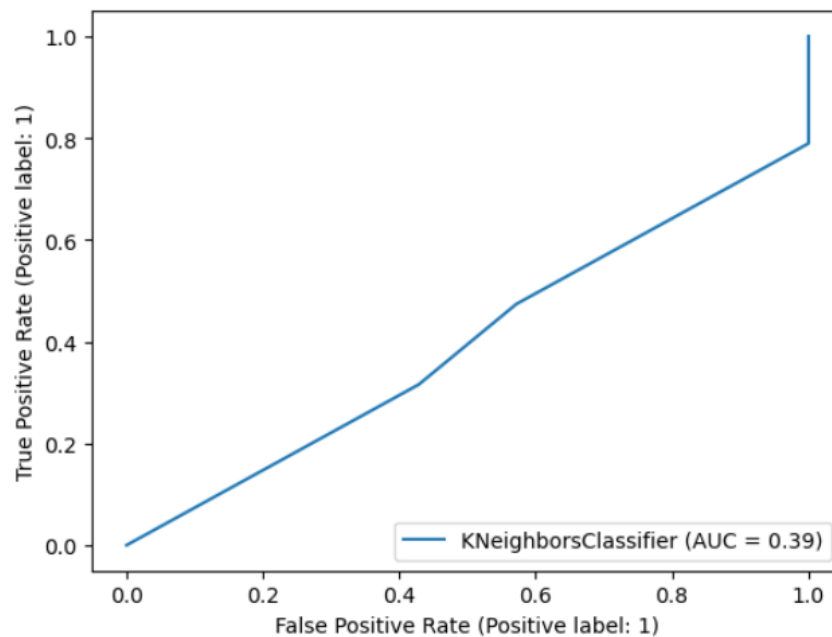
Confusion Matrix:

```
[[ 0  7]
```

```
 [ 0 38]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.84	1.00	0.92	38
accuracy			0.84	45
macro avg	0.42	0.50	0.46	45
weighted avg	0.71	0.84	0.77	45



Accuracy for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90909091]

Mean Accuracy: 0.914040404040404

F1 Score for all folds: [0.95348837 0.96470588 0.95238095 0.95238095 0.95238095]

Mean F1 Score: 0.9550674223177644

Precision for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90909091]

Mean Precision: 0.914040404040404

Recall for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90909091]

Mean Recall: 0.914040404040404

Random Forest

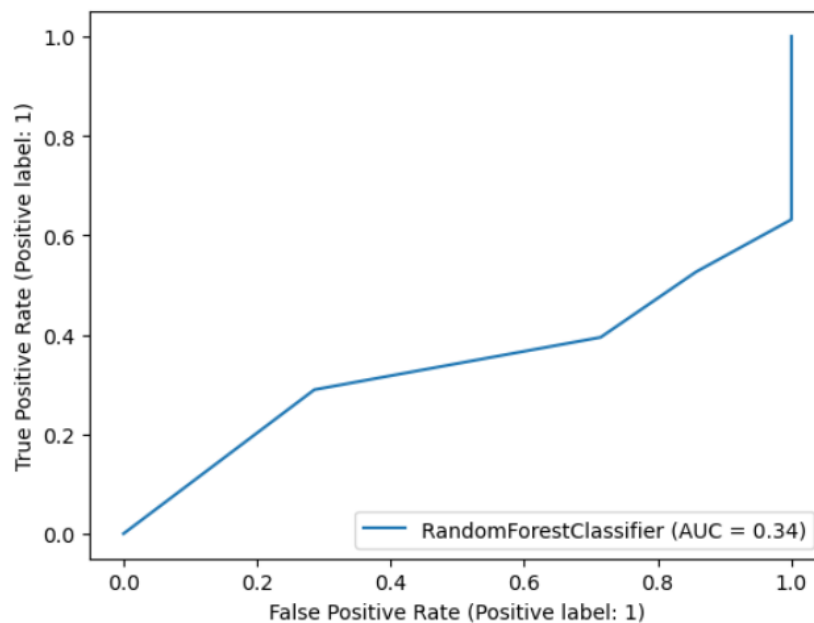
Accuracy: 0.8444444444444444

Confusion Matrix:

```
[[ 0  7]
 [ 0 38]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.84	1.00	0.92	38
accuracy			0.84	45
macro avg	0.42	0.50	0.46	45
weighted avg	0.71	0.84	0.77	45



Accuracy for all folds: [0.88888889 0.93181818 0.90909091 0.90909091 0.90909091]
Mean Accuracy: 0.9095959595959595
F1 Score for all folds: [0.94117647 0.96470588 0.95238095 0.95238095 0.95238095]
Mean F1 Score: 0.9526050420168067
Precision for all folds: [0.90909091 0.93181818 0.90909091 0.90909091 0.90909091]
Mean Precision: 0.9136363636363637
Recall for all folds: [0.90909091 0.93181818 0.90909091 0.90909091 0.90909091]
Mean Recall: 0.9136363636363637

XGBClassifier

Accuracy: 0.8444444444444444

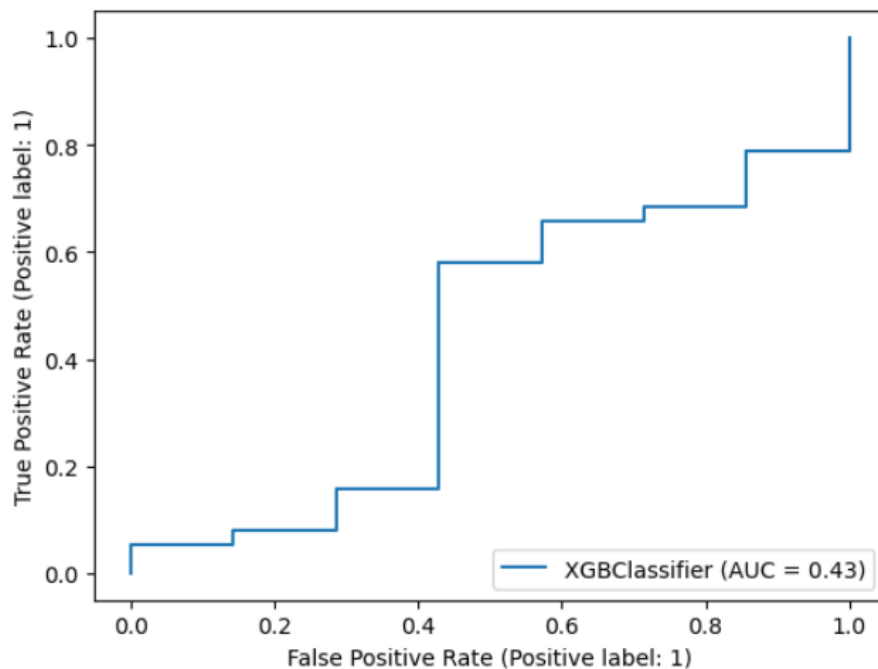
Confusion Matrix:

```
[[ 0  7]
```

```
 [ 0 38]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.84	1.00	0.92	38
accuracy			0.84	45
macro avg	0.42	0.50	0.46	45
weighted avg	0.71	0.84	0.77	45



Accuracy for all folds: [0.86666667 0.90909091 0.86363636 0.90909091 0.84090909]

Mean Accuracy: 0.8778787878787879

F1 Score for all folds: [0.92857143 0.95238095 0.92682927 0.95238095 0.91358025]

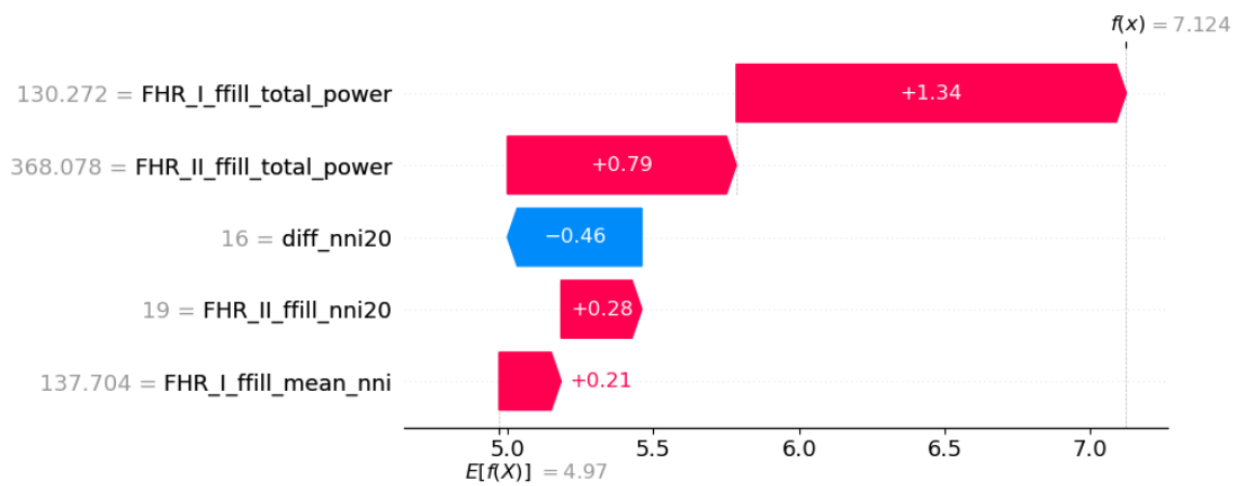
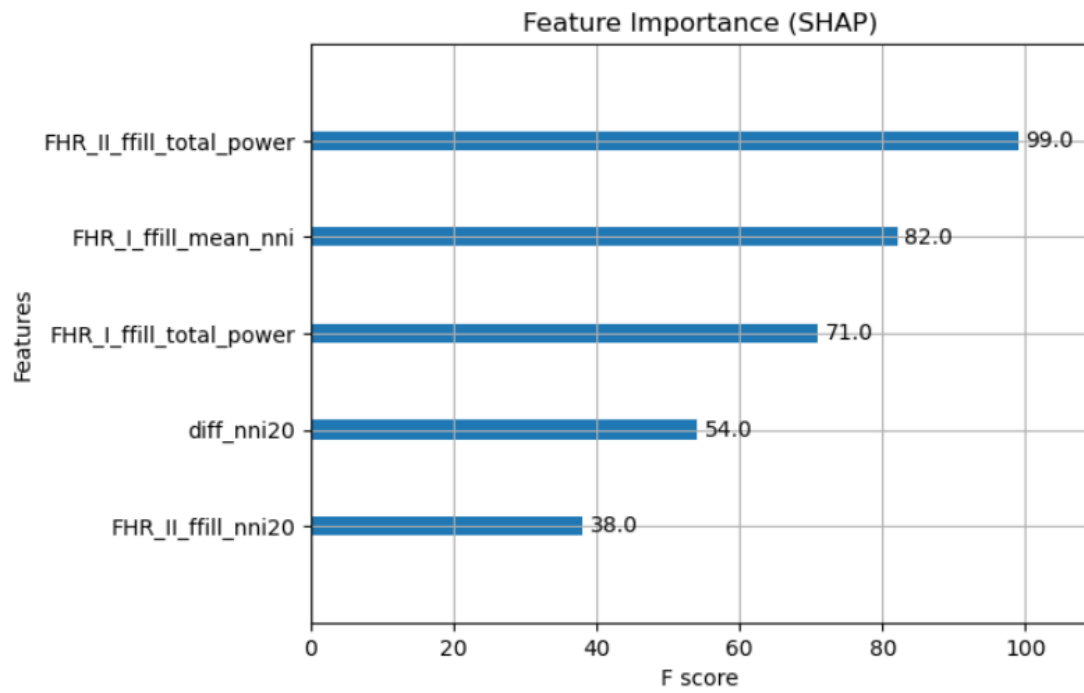
Mean F1 Score: 0.9347485697079192

Precision for all folds: [0.90697674 0.93023256 0.9047619 0.90909091 0.90243902]

Mean Precision: 0.9107002281137279

Recall for all folds: [0.90697674 0.93023256 0.9047619 0.90909091 0.90243902]

Mean Recall: 0.9107002281137279



Gest Term (Linear SVC) Results

KNN

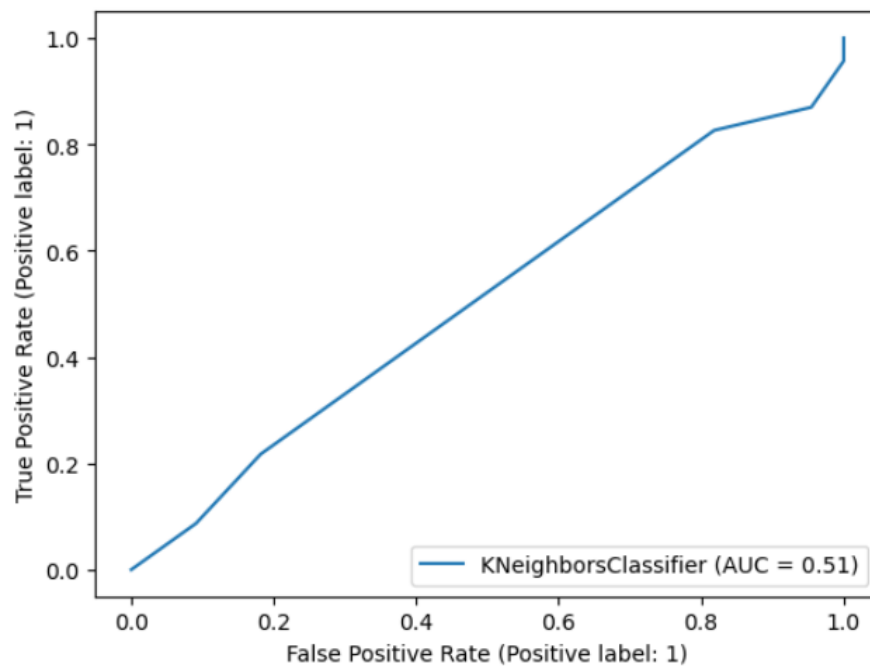
Accuracy: 0.5111111111111111

Confusion Matrix:

```
[[18  4]
 [18  5]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.50	0.82	0.62	22
1	0.56	0.22	0.31	23
accuracy			0.51	45
macro avg	0.53	0.52	0.47	45
weighted avg	0.53	0.51	0.46	45



Accuracy for all folds: [0.53333333 0.54545455 0.63636364 0.5 0.52272727]
Mean Accuracy: 0.5475757575757575
F1 Score for all folds: [0.36363636 0.44444444 0.57894737 0.45 0.4]
Mean F1 Score: 0.44740563530037214
Precision for all folds: [0.5 0.53333333 0.64705882 0.45 0.46666667]
Mean Precision: 0.5194117647058824
Recall for all folds: [0.5 0.53333333 0.64705882 0.45 0.46666667]
Mean Recall: 0.5194117647058824

Random Forest

Accuracy: 0.6222222222222222

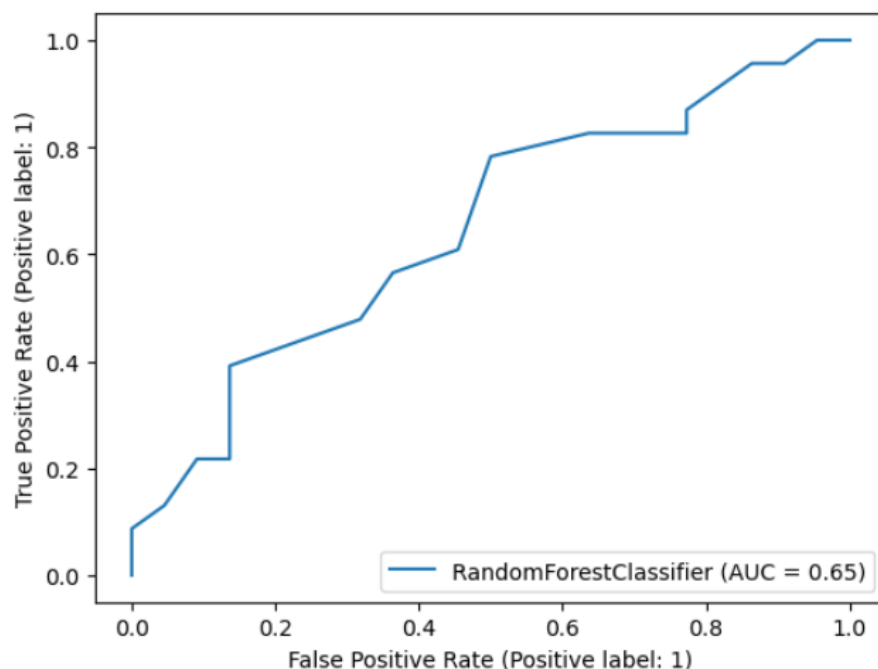
Confusion Matrix:

[[19 3]

[14 9]]

Classification Report:

	precision	recall	f1-score	support
0	0.58	0.86	0.69	22
1	0.75	0.39	0.51	23
accuracy			0.62	45
macro avg	0.66	0.63	0.60	45
weighted avg	0.66	0.62	0.60	45



Accuracy for all folds: [0.51111111 0.47727273 0.56818182 0.59090909 0.5]
Mean Accuracy: 0.5294949494949495
F1 Score for all folds: [0.5 0.37837838 0.51282051 0.59090909 0.42105263]
Mean F1 Score: 0.48063212273738587
Precision for all folds: [0.47826087 0.4375 0.55555556 0.54166667 0.44444444]
Mean Precision: 0.49148550724637674
Recall for all folds: [0.47826087 0.4375 0.55555556 0.54166667 0.44444444]
Mean Recall: 0.49148550724637674

XGBClassifier

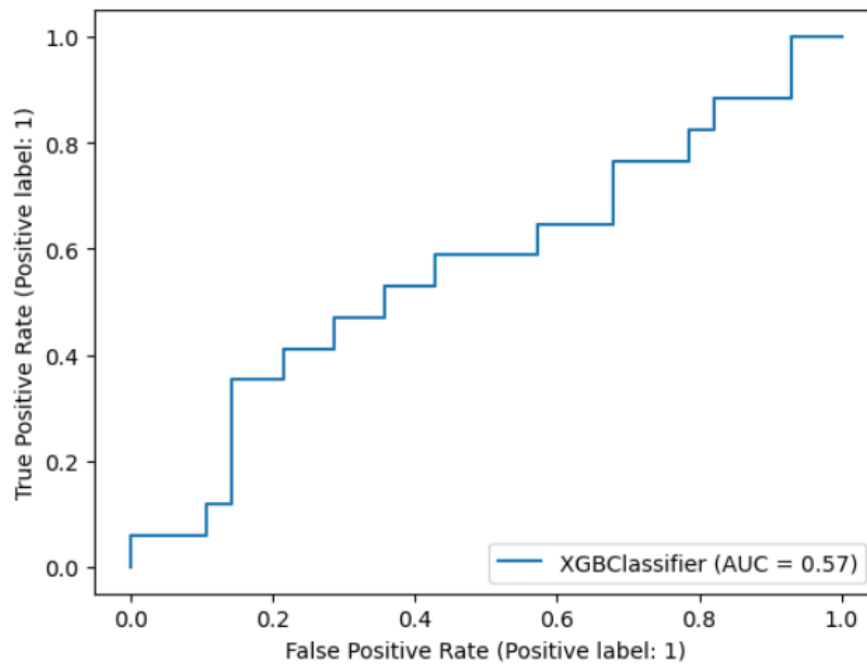
Accuracy: 0.5111111111111111

Confusion Matrix:

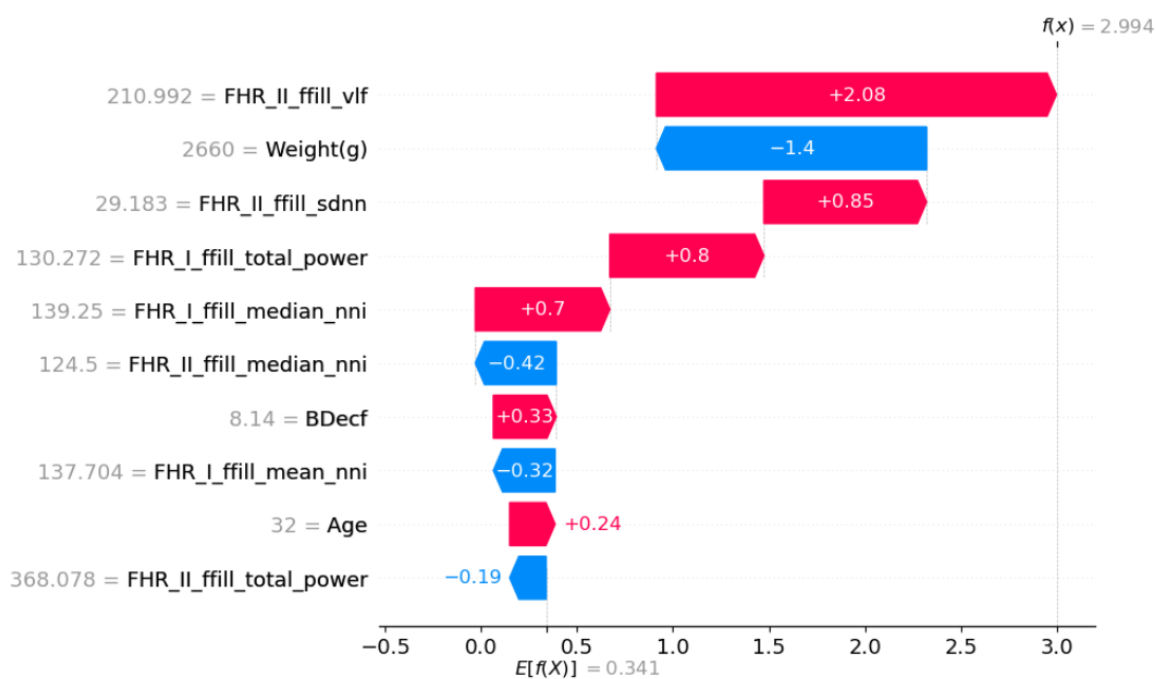
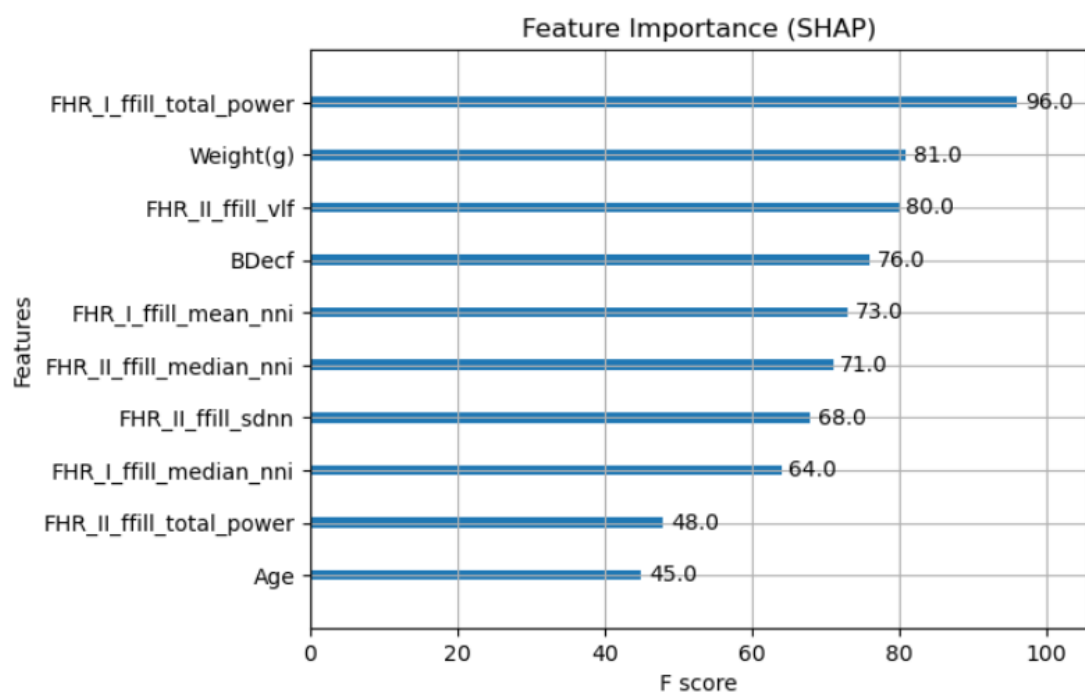
```
[[13 15]
 [ 7 10]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.65	0.46	0.54	28
1	0.40	0.59	0.48	17
accuracy			0.51	45
macro avg	0.53	0.53	0.51	45
weighted avg	0.56	0.51	0.52	45



Accuracy for all folds: [0.48888889 0.56818182 0.59090909 0.45454545 0.52272727]
Mean Accuracy: 0.5250505050505051
F1 Score for all folds: [0.41025641 0.48648649 0.52631579 0.36842105 0.51162791]
Mean F1 Score: 0.4606215291649808
Precision for all folds: [0.44444444 0.5625 0.58823529 0.38888889 0.47826087]
Mean Precision: 0.4924658994032395
Recall for all folds: [0.44444444 0.5625 0.58823529 0.38888889 0.47826087]
Mean Recall: 0.4924658994032395



pH risk (Linear SVC) Results

KNN

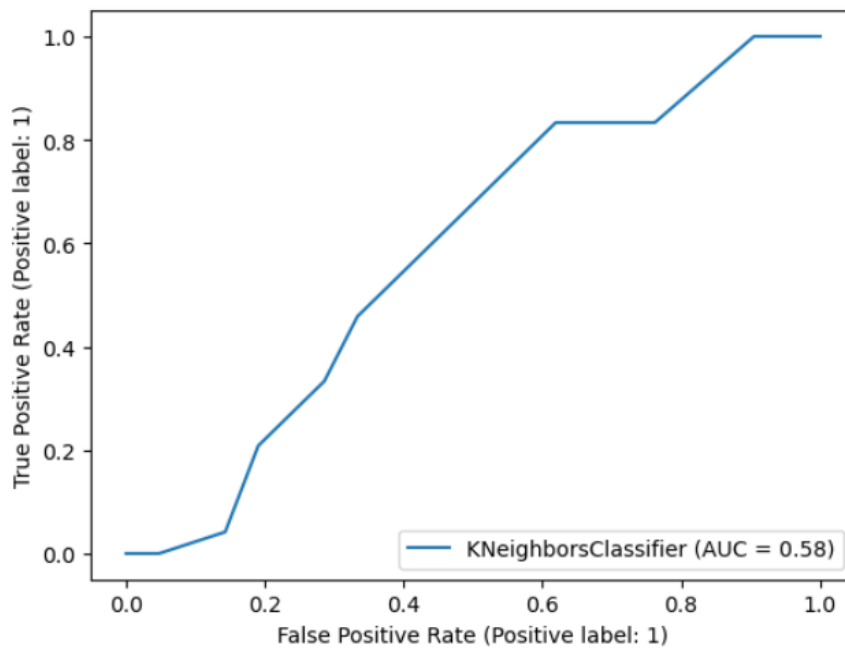
Accuracy: 0.5111111111111111

Confusion Matrix:

```
[[15  6]
 [16  8]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.48	0.71	0.58	21
1	0.57	0.33	0.42	24
accuracy			0.51	45
macro avg	0.53	0.52	0.50	45
weighted avg	0.53	0.51	0.49	45



Accuracy for all folds: [0.42222222 0.54545455 0.52272727 0.45454545 0.5]
Mean Accuracy: 0.48898989898989899
F1 Score for all folds: [0.31578947 0.44444444 0.48780488 0.36842105 0.45]
Mean F1 Score: 0.4132919697618028
Precision for all folds: [0.375 0.57142857 0.5 0.41176471 0.47368421]
Mean Precision: 0.46637549756744806
Recall for all folds: [0.375 0.57142857 0.5 0.41176471 0.47368421]
Mean Recall: 0.46637549756744806

Random Forest

Accuracy: 0.9777777777777777

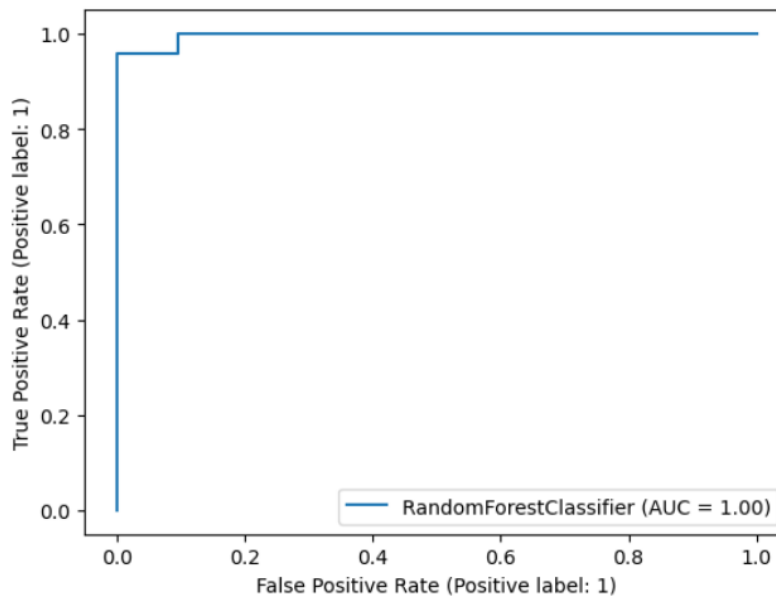
Confusion Matrix:

```
[[21  0]
```

```
 [ 1 23]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.95	1.00	0.98	21
1	1.00	0.96	0.98	24
accuracy			0.98	45
macro avg	0.98	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45



Accuracy for all folds: [0.95555556 0.97727273 0.95454545 0.93181818 0.93181818]

Mean Accuracy: 0.9502020202020202

F1 Score for all folds: [0.95238095 0.97777778 0.95238095 0.93023256 0.92682927]

Mean F1 Score: 0.9479203017943799

Precision for all folds: [1. 0.95652174 0.95238095 0.90909091 0.95]

Mean Precision: 0.9535987201204593

Recall for all folds: [1. 0.95652174 0.95238095 0.90909091 0.95]

Mean Recall: 0.9535987201204593

XGBClassifier

Accuracy: 0.9333333333333333

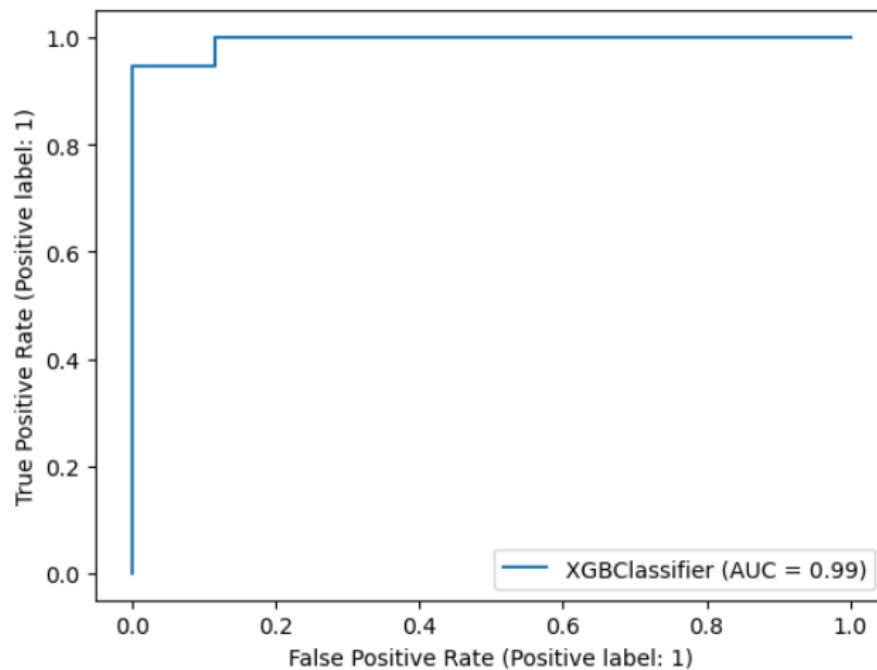
Confusion Matrix:

```
[[23  3]
```

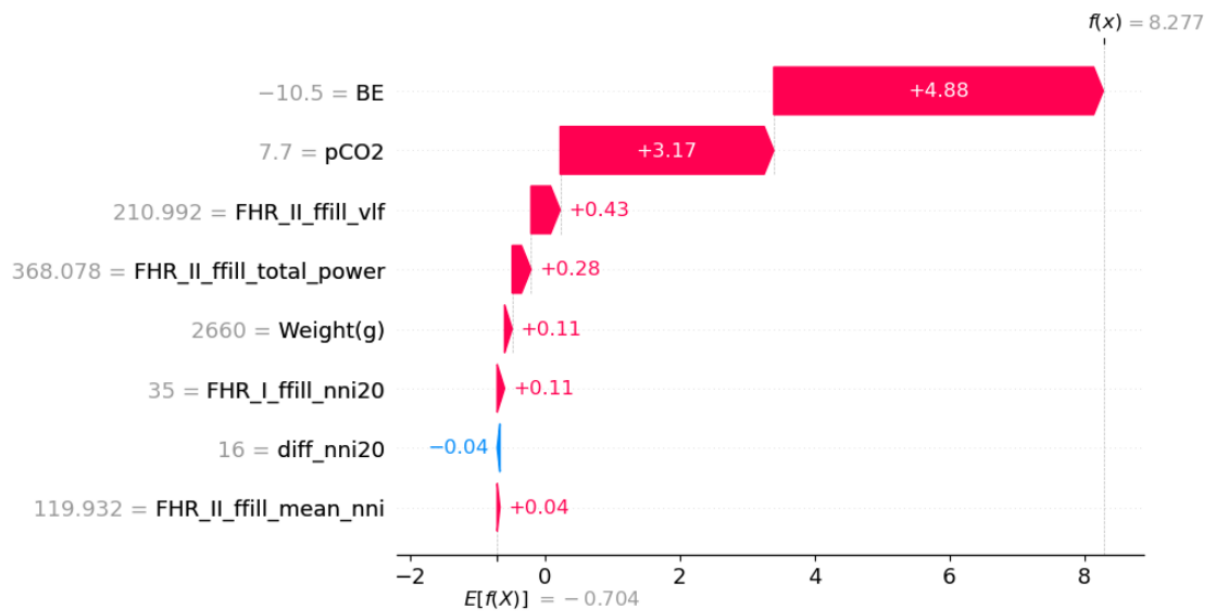
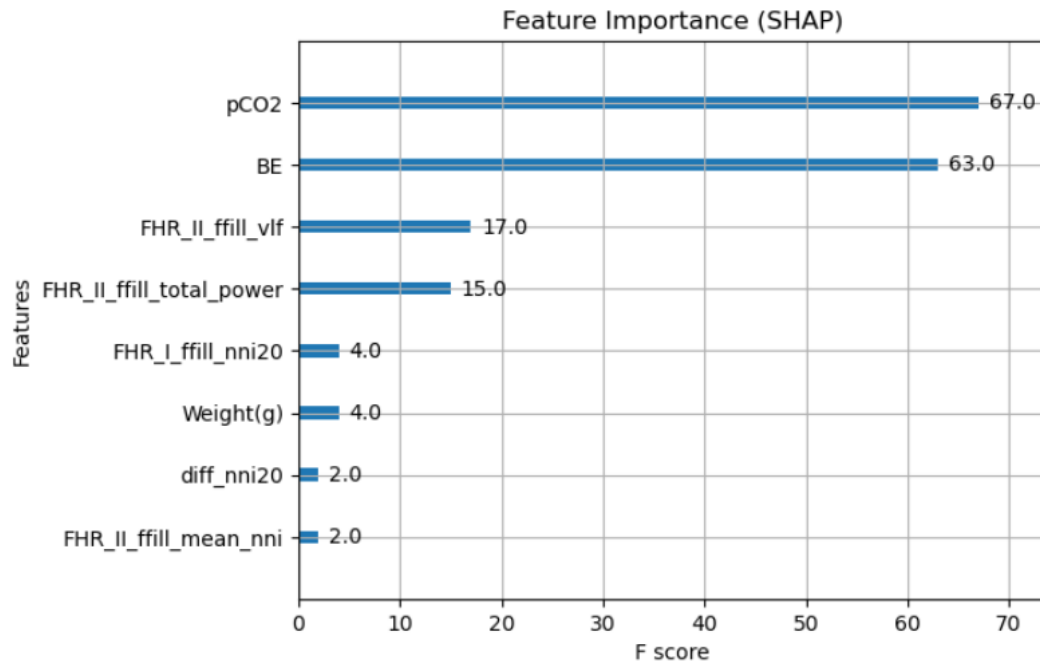
```
 [ 0 19]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.88	0.94	26
1	0.86	1.00	0.93	19
accuracy			0.93	45
macro avg	0.93	0.94	0.93	45
weighted avg	0.94	0.93	0.93	45



Accuracy for all folds: [1. 0.95454545 0.95454545 0.97727273 0.95454545]
Mean Accuracy: 0.9681818181818184
F1 Score for all folds: [1. 0.95652174 0.95238095 0.97674419 0.95238095]
Mean F1 Score: 0.9676055659877703
Precision for all folds: [1. 0.91666667 0.95238095 0.95454545 0.95238095]
Mean Precision: 0.955194805194805
Recall for all folds: [1. 0.91666667 0.95238095 0.95454545 0.95238095]
Mean Recall: 0.955194805194805

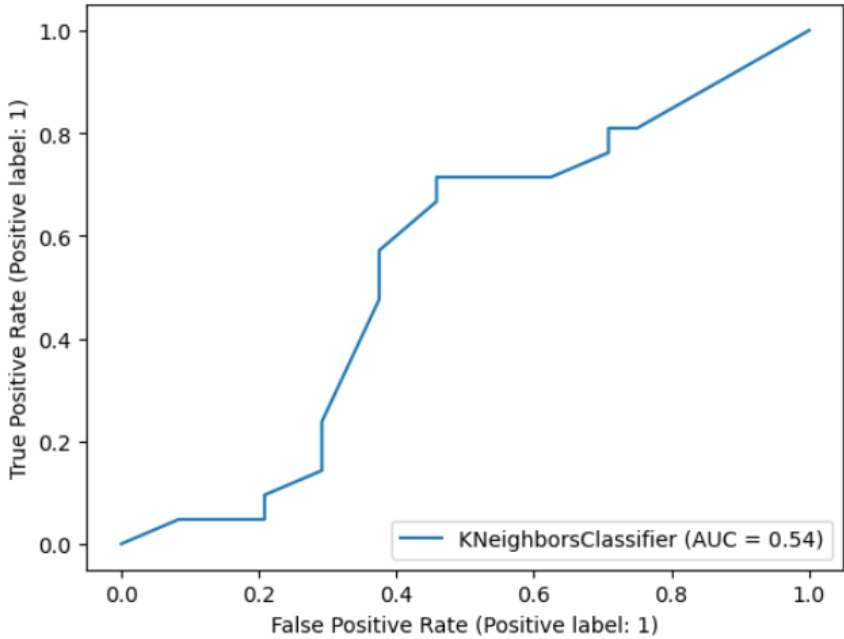


Maternal Age (ExtraTreesClassifier) Results

KNN

Accuracy: 0.6
Confusion Matrix:
[[14 10]
 [8 13]]
Classification Report:

	precision	recall	f1-score	support
0	0.64	0.58	0.61	24
1	0.57	0.62	0.59	21
accuracy			0.60	45
macro avg	0.60	0.60	0.60	45
weighted avg	0.60	0.60	0.60	45



Accuracy for all folds: [0.55555556 0.47727273 0.45454545 0.59090909 0.54545455]
Mean Accuracy: 0.5247474747474747
F1 Score for all folds: [0. 0.63492063 0.625 0.18181818 0.]
Mean F1 Score: 0.28834776334776335

Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.

Precision for all folds: [0. 0.46511628 0.45454545 1. 0.]
Mean Precision: 0.3839323467230444
Recall for all folds: [0. 0.46511628 0.45454545 1. 0.]
Mean Recall: 0.3839323467230444

Random Forest

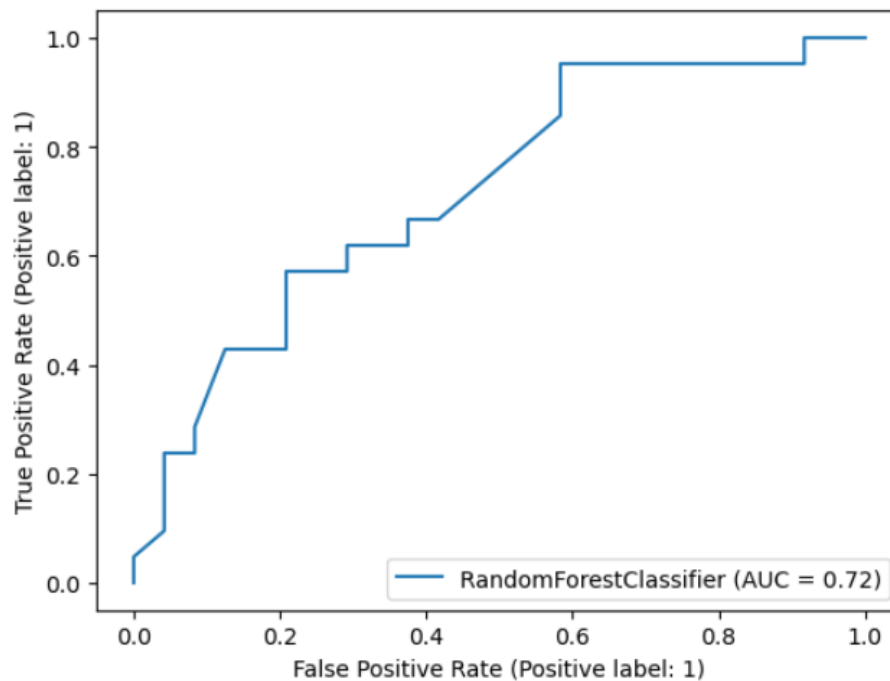
Accuracy: 0.6444444444444445

Confusion Matrix:

```
[[17  7]
 [ 9 12]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.65	0.71	0.68	24
1	0.63	0.57	0.60	21
accuracy			0.64	45
macro avg	0.64	0.64	0.64	45
weighted avg	0.64	0.64	0.64	45



Accuracy for all folds: [0.64444444 0.5 0.56818182 0.59090909 0.56818182]
Mean Accuracy: 0.5743434343434344
F1 Score for all folds: [0.38461538 0.56 0.59574468 0.4375 0.0952381]
Mean F1 Score: 0.41461963214090874
Precision for all folds: [0.83333333 0.46666667 0.51851852 0.58333333 1.]
Mean Precision: 0.6803703703703704
Recall for all folds: [0.83333333 0.46666667 0.51851852 0.58333333 1.]
Mean Recall: 0.6803703703703704

XGBClassifier

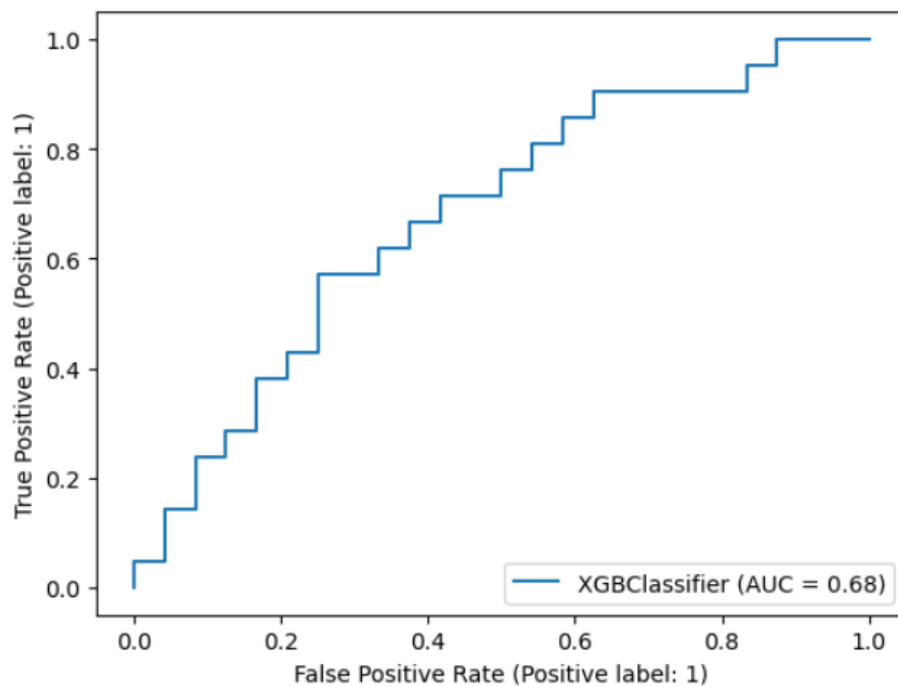
Accuracy: 0.6444444444444445

Confusion Matrix:

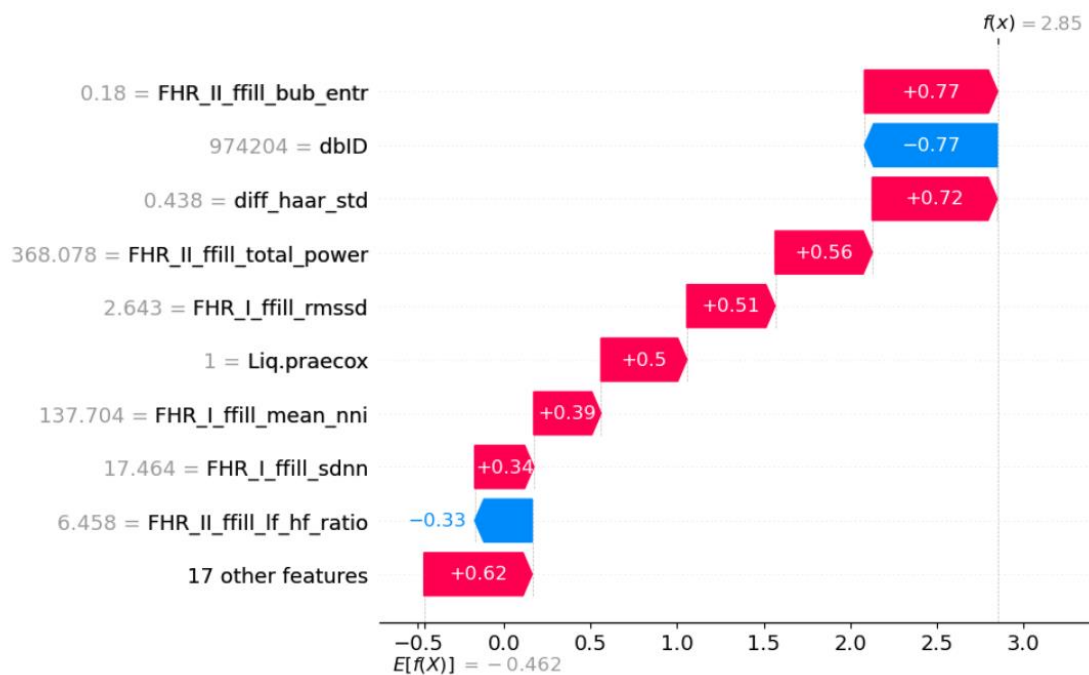
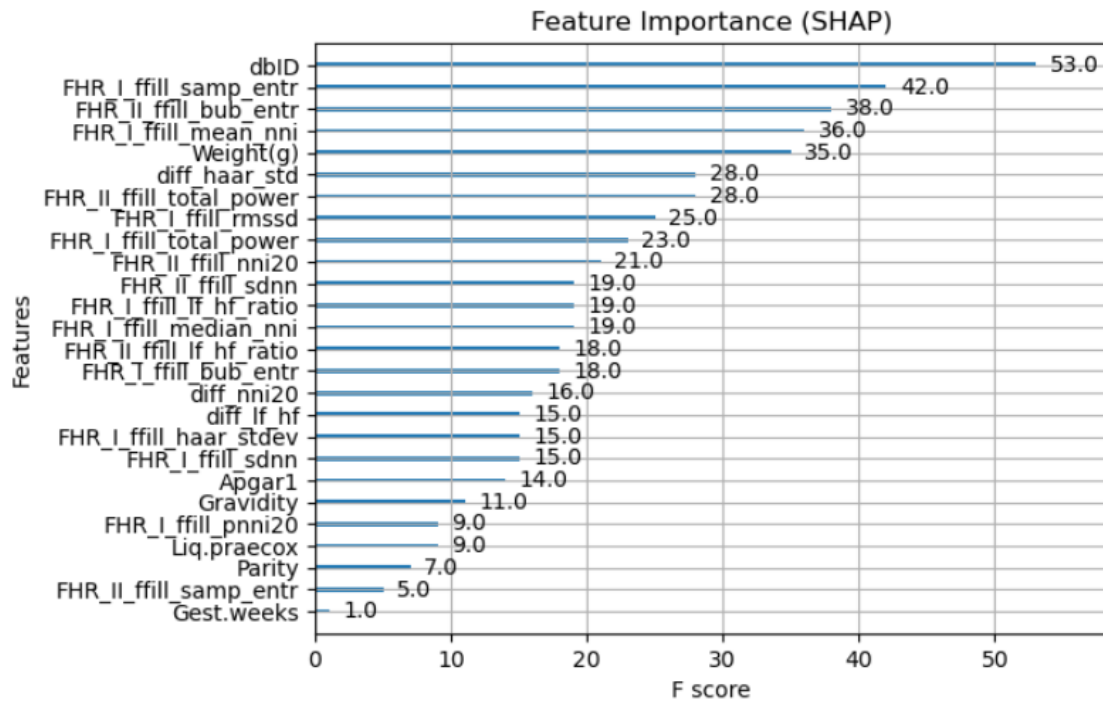
```
[[14 10]
 [ 6 15]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.70	0.58	0.64	24
1	0.60	0.71	0.65	21
accuracy			0.64	45
macro avg	0.65	0.65	0.64	45
weighted avg	0.65	0.64	0.64	45



Accuracy for all folds: [0.64444444 0.45454545 0.56818182 0.52272727 0.59090909]
Mean Accuracy: 0.5561616161616161
F1 Score for all folds: [0.38461538 0.57142857 0.65454545 0.4 0.30769231]
Mean F1 Score: 0.46365634365634356
Precision for all folds: [0.83333333 0.44444444 0.51428571 0.46666667 0.66666667]
Mean Precision: 0.5850793650793651
Recall for all folds: [0.83333333 0.44444444 0.51428571 0.46666667 0.66666667]
Mean Recall: 0.5850793650793651



Neonate Weight (ExtraTreesClassifier) Results

KNN

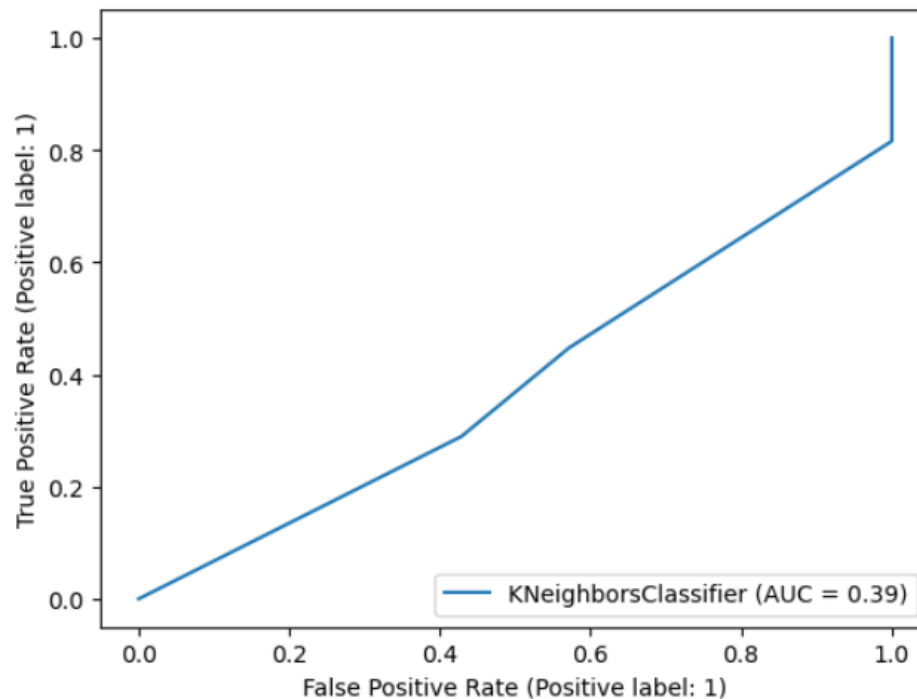
Accuracy: 0.8444444444444444

Confusion Matrix:

```
[[ 0  7]
 [ 0 38]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.84	1.00	0.92	38
accuracy			0.84	45
macro avg	0.42	0.50	0.46	45
weighted avg	0.71	0.84	0.77	45



Accuracy for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90909091]
Mean Accuracy: 0.914040404040404
F1 Score for all folds: [0.95348837 0.96470588 0.95238095 0.95238095 0.95238095]
Mean F1 Score: 0.9550674223177644
Precision for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90909091]
Mean Precision: 0.914040404040404
Recall for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90909091]
Mean Recall: 0.914040404040404

Random Forest

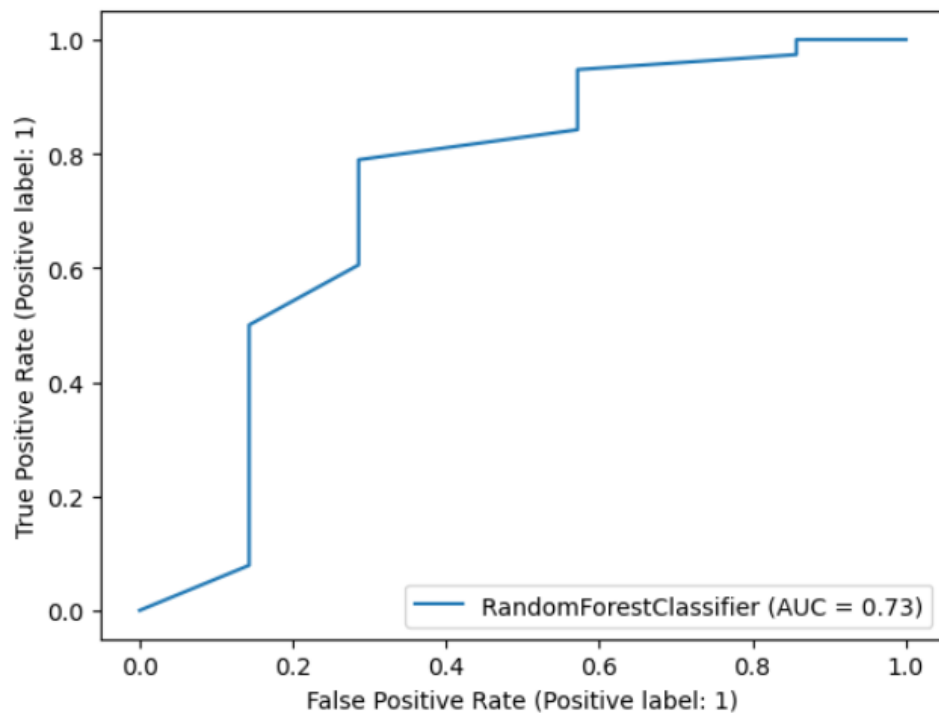
Accuracy: 0.8444444444444444

Confusion Matrix:

```
[[ 0  7]
 [ 0 38]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.84	1.00	0.92	38
accuracy			0.84	45
macro avg	0.42	0.50	0.46	45
weighted avg	0.71	0.84	0.77	45



Accuracy for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.88636364]
Mean Accuracy: 0.9094949494949495
F1 Score for all folds: [0.95348837 0.96470588 0.95238095 0.95238095 0.93975904]
Mean F1 Score: 0.9525430390704896
Precision for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90697674]
Mean Precision: 0.9136175710594315
Recall for all folds: [0.91111111 0.93181818 0.90909091 0.90909091 0.90697674]
Mean Recall: 0.9136175710594315

XGBClassifier

Accuracy: 0.8444444444444444

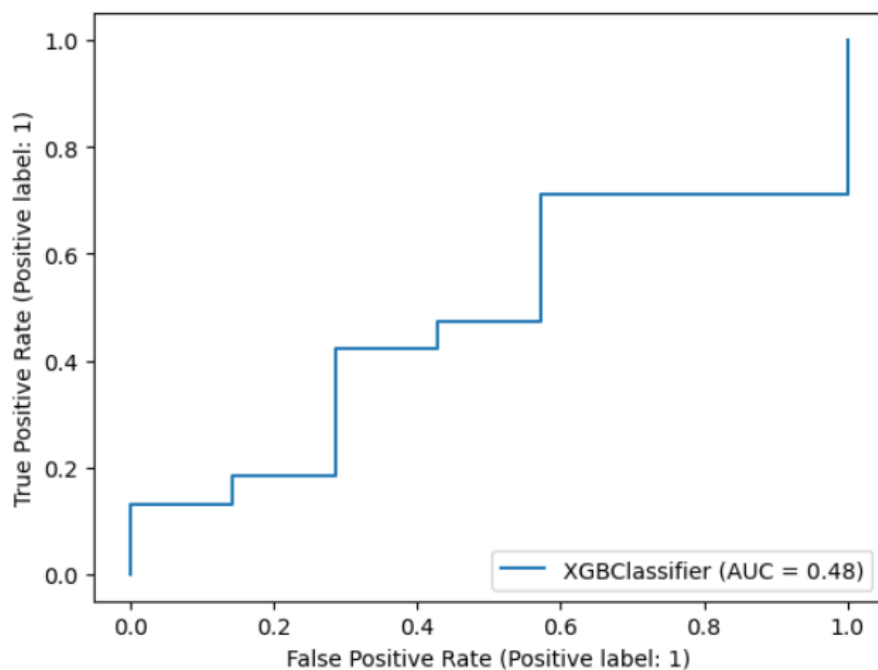
Confusion Matrix:

```
[[ 0  7]
```

```
 [ 0 38]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.84	1.00	0.92	38
accuracy			0.84	45
macro avg	0.42	0.50	0.46	45
weighted avg	0.71	0.84	0.77	45



Accuracy for all folds: [0.91111111 0.93181818 0.88636364 0.90909091 0.86363636]

Mean Accuracy: 0.9004040404040404

F1 Score for all folds: [0.95238095 0.96470588 0.93975904 0.95238095 0.92682927]

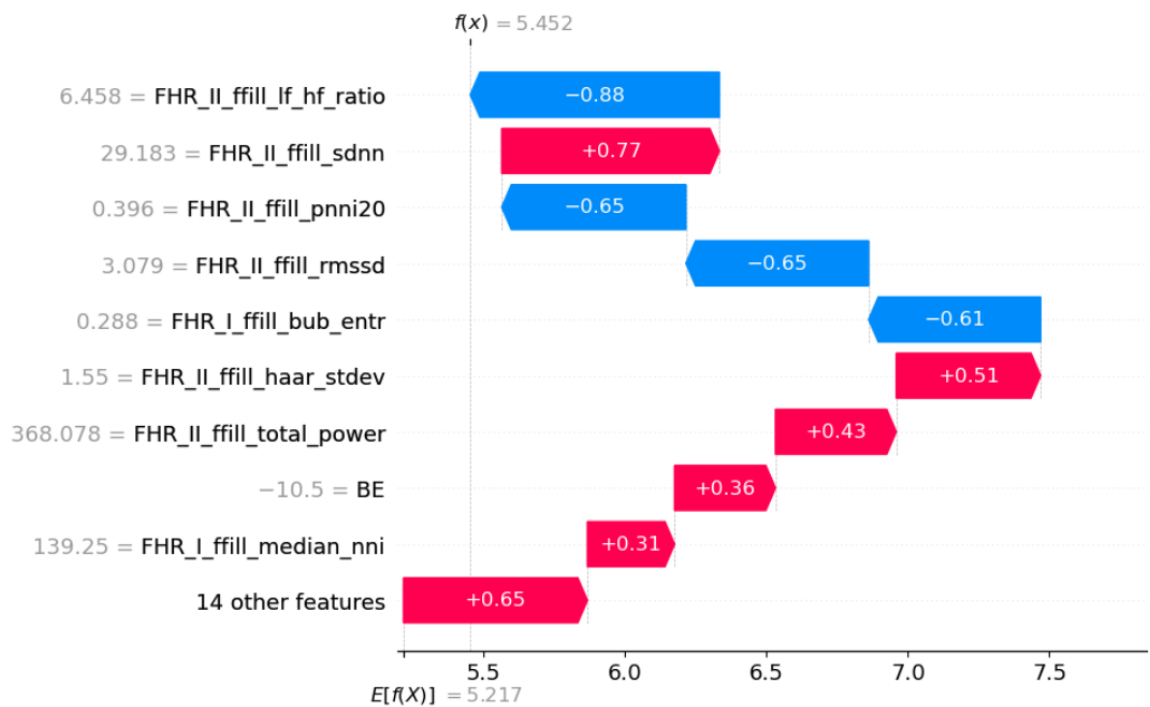
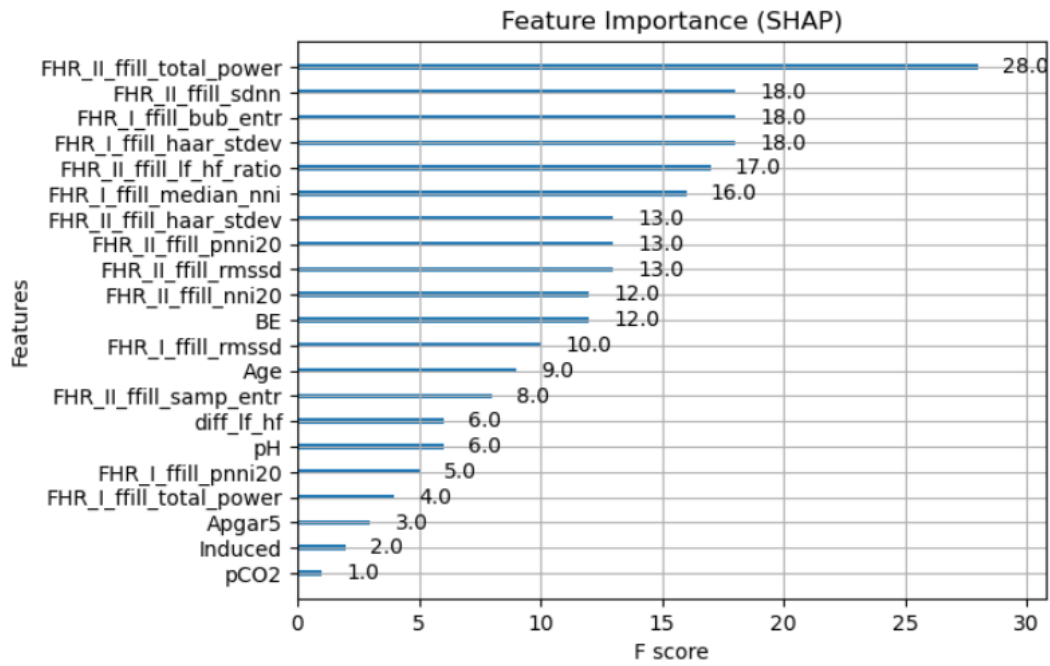
Mean F1 Score: 0.9472112183104213

Precision for all folds: [0.93023256 0.93181818 0.90697674 0.90909091 0.9047619]

Mean Precision: 0.9165760595993154

Recall for all folds: [0.93023256 0.93181818 0.90697674 0.90909091 0.9047619]

Mean Recall: 0.9165760595993154



Gest Term (ExtraTreesClassifier) Results

KNN

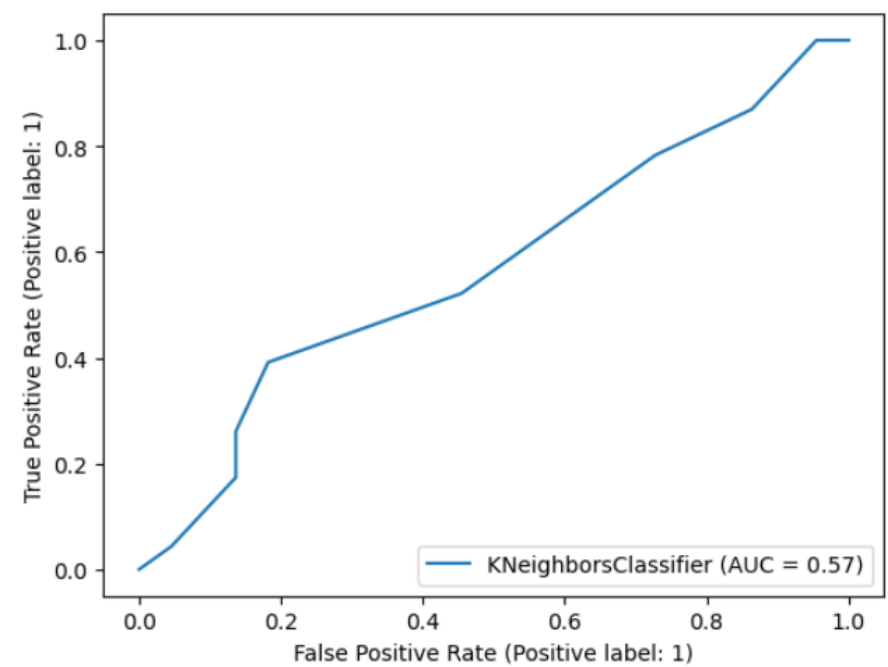
Accuracy: 0.5555555555555556

Confusion Matrix:

```
[[19  3]
 [17  6]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.53	0.86	0.66	22
1	0.67	0.26	0.38	23
accuracy			0.56	45
macro avg	0.60	0.56	0.52	45
weighted avg	0.60	0.56	0.51	45



Accuracy for all folds: [0.46666667 0.34090909 0.29545455 0.54545455 0.54545455]
Mean Accuracy: 0.4387878787878788
F1 Score for all folds: [0.63636364 0.12121212 0.27906977 0. 0.]
Mean F1 Score: 0.20732910500352358

Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.

Precision for all folds: [0.46666667 0.16666667 0.27272727 0. 0.]
Mean Precision: 0.1812121212121212
Recall for all folds: [0.46666667 0.16666667 0.27272727 0. 0.]
Mean Recall: 0.1812121212121212

Random Forest

Accuracy: 0.6222222222222222

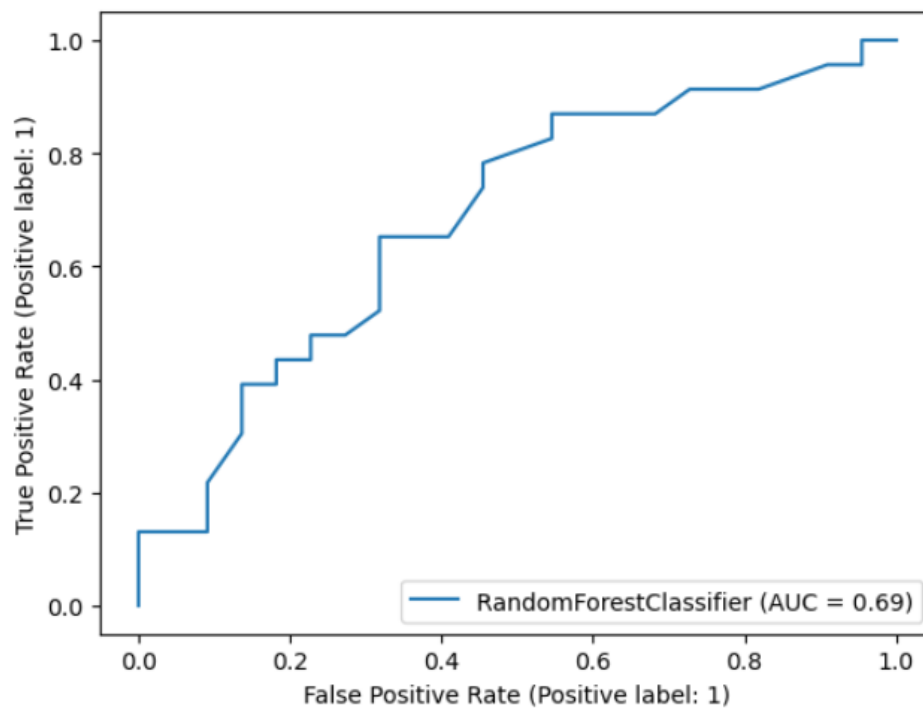
Confusion Matrix:

[[18 4]

[13 10]]

Classification Report:

	precision	recall	f1-score	support
0	0.58	0.82	0.68	22
1	0.71	0.43	0.54	23
accuracy			0.62	45
macro avg	0.65	0.63	0.61	45
weighted avg	0.65	0.62	0.61	45



Accuracy for all folds: [0.48888889 0.43181818 0.54545455 0.54545455 0.47727273]

Mean Accuracy: 0.49777777777777776

F1 Score for all folds: [0.63492063 0.28571429 0.375 0.375 0.25806452]

Mean F1 Score: 0.3857398873527906

Precision for all folds: [0.47619048 0.35714286 0.54545455 0.5 0.36363636]

Mean Precision: 0.44848484848484843

Recall for all folds: [0.47619048 0.35714286 0.54545455 0.5 0.36363636]

Mean Recall: 0.44848484848484843

XGBClassifier

Accuracy: 0.5777777777777777

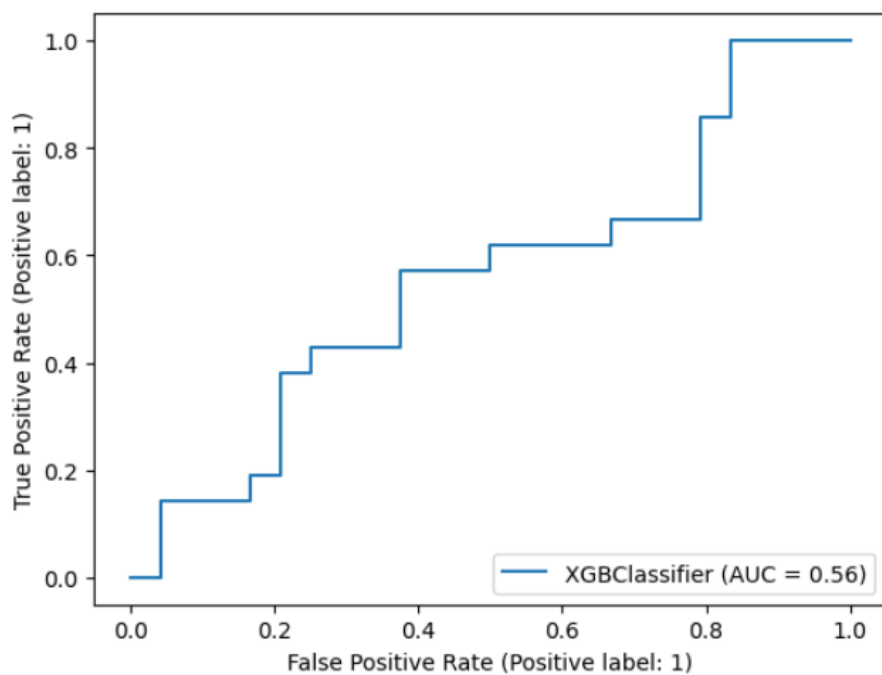
Confusion Matrix:

[[17 7]

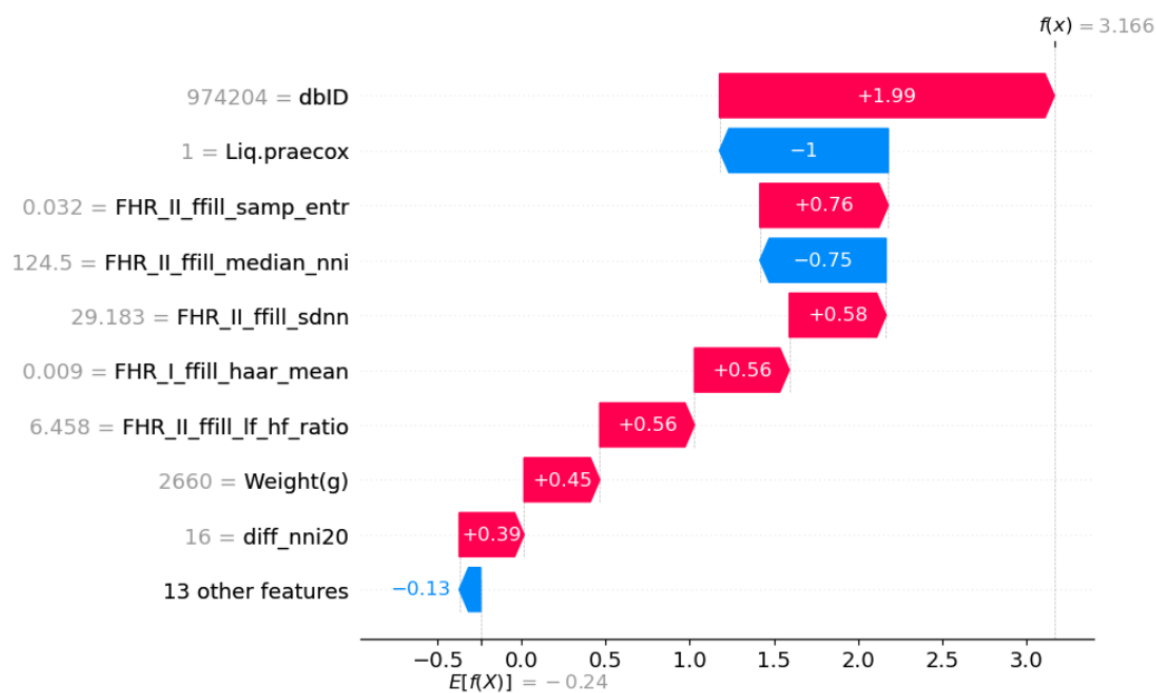
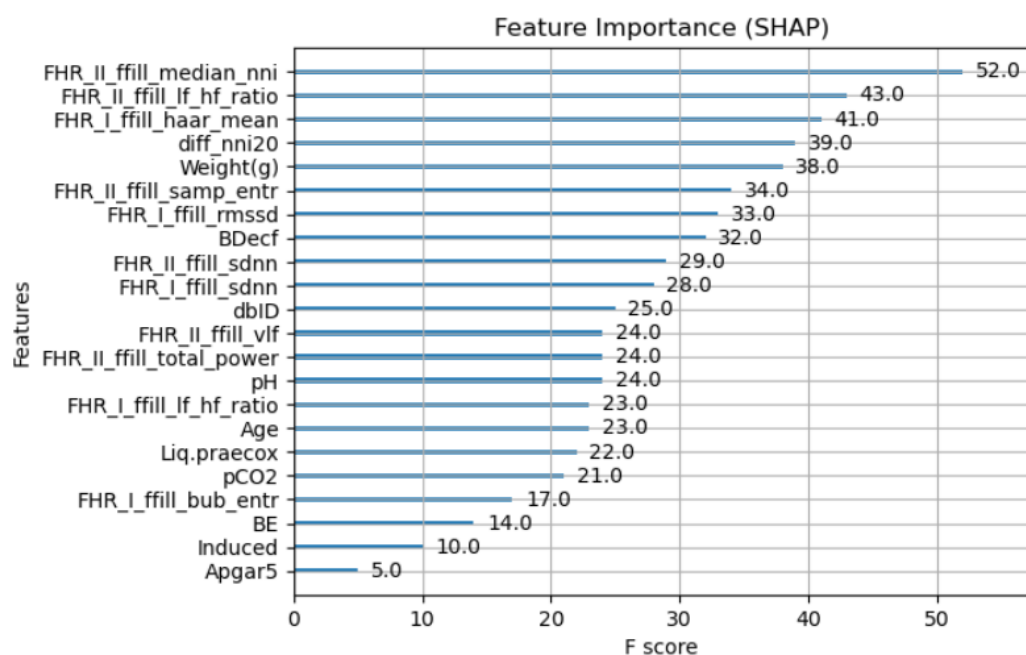
[12 9]]

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.71	0.64	24
1	0.56	0.43	0.49	21
accuracy			0.58	45
macro avg	0.57	0.57	0.56	45
weighted avg	0.58	0.58	0.57	45



Accuracy for all folds: [0.44444444 0.5 0.56818182 0.59090909 0.54545455]
Mean Accuracy: 0.5297979797979797
F1 Score for all folds: [0.61538462 0.45 0.45714286 0.47058824 0.44444444]
Mean F1 Score: 0.48751203045320696
Precision for all folds: [0.45454545 0.47368421 0.57142857 0.57142857 0.5]
Mean Precision: 0.5142173615857827
Recall for all folds: [0.45454545 0.47368421 0.57142857 0.57142857 0.5]
Mean Recall: 0.5142173615857827



pH risk (ExtraTreesClassifier) Results

KNN

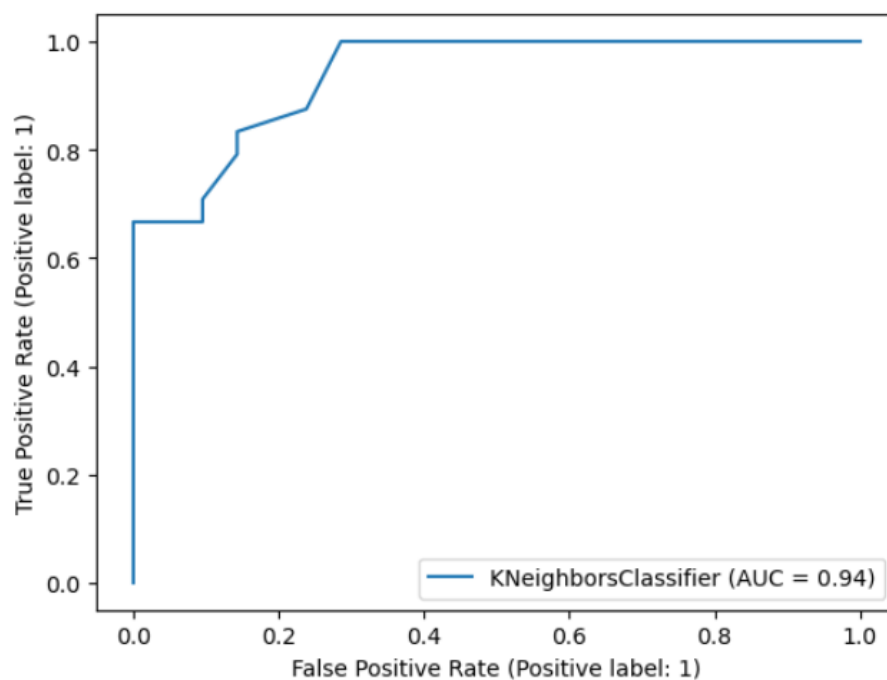
Accuracy: 0.8222222222222222

Confusion Matrix:

```
[[18  3]
 [ 5 19]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.78	0.86	0.82	21
1	0.86	0.79	0.83	24
accuracy			0.82	45
macro avg	0.82	0.82	0.82	45
weighted avg	0.83	0.82	0.82	45



Accuracy for all folds: [0.84444444 0.86363636 0.81818182 0.79545455 0.84090909]

Mean Accuracy: 0.83252525252524

F1 Score for all folds: [0.82051282 0.85 0.78947368 0.76923077 0.82051282]

Mean F1 Score: 0.8099460188933871

Precision for all folds: [0.94117647 0.94444444 0.88235294 0.83333333 0.88888889]

Mean Precision: 0.8980392156862745

Recall for all folds: [0.94117647 0.94444444 0.88235294 0.83333333 0.88888889]

Mean Recall: 0.8980392156862745

Random Forest

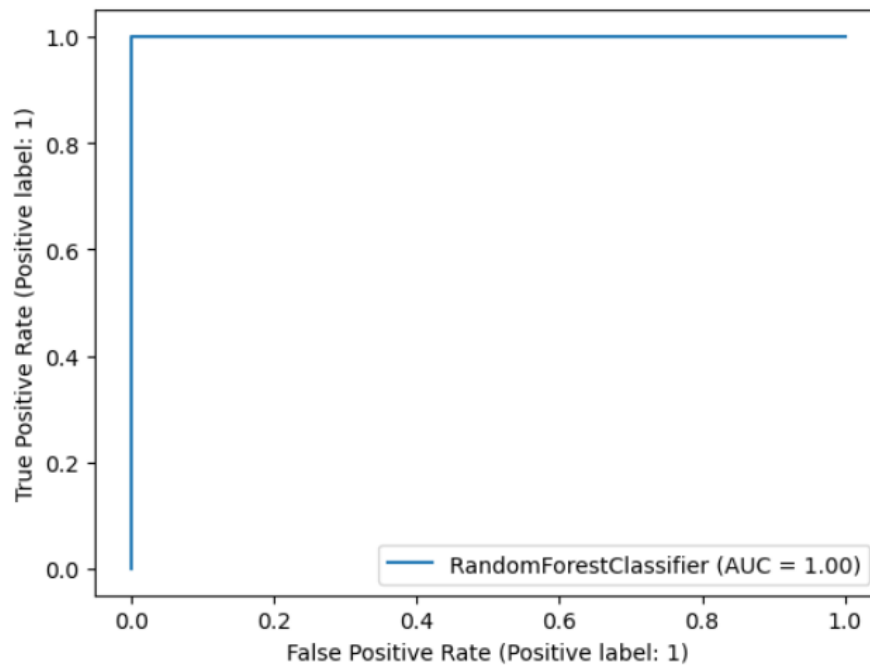
Accuracy: 0.9777777777777777

Confusion Matrix:

```
[[20  1]
 [ 0 24]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.95	0.98	21
1	0.96	1.00	0.98	24
accuracy			0.98	45
macro avg	0.98	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45



Accuracy for all folds: [0.97777778 0.93181818 0.95454545 0.95454545 0.95454545]
Mean Accuracy: 0.9546464646464645
F1 Score for all folds: [0.97674419 0.93617021 0.95238095 0.95238095 0.95238095]
Mean F1 Score: 0.9540114511910653
Precision for all folds: [1. 0.88 0.95238095 0.95238095 0.95238095]
Mean Precision: 0.9474285714285715
Recall for all folds: [1. 0.88 0.95238095 0.95238095 0.95238095]
Mean Recall: 0.9474285714285715

XGBClassifier

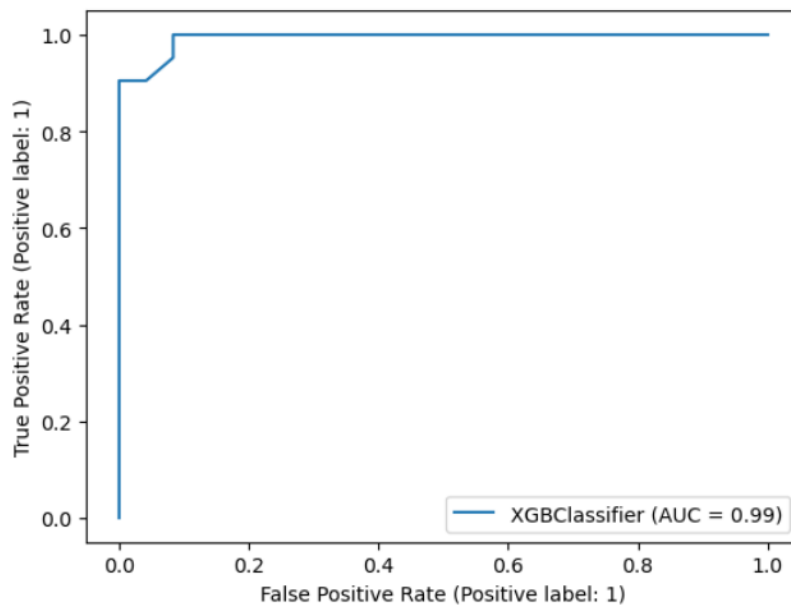
Accuracy: 0.9555555555555556

Confusion Matrix:

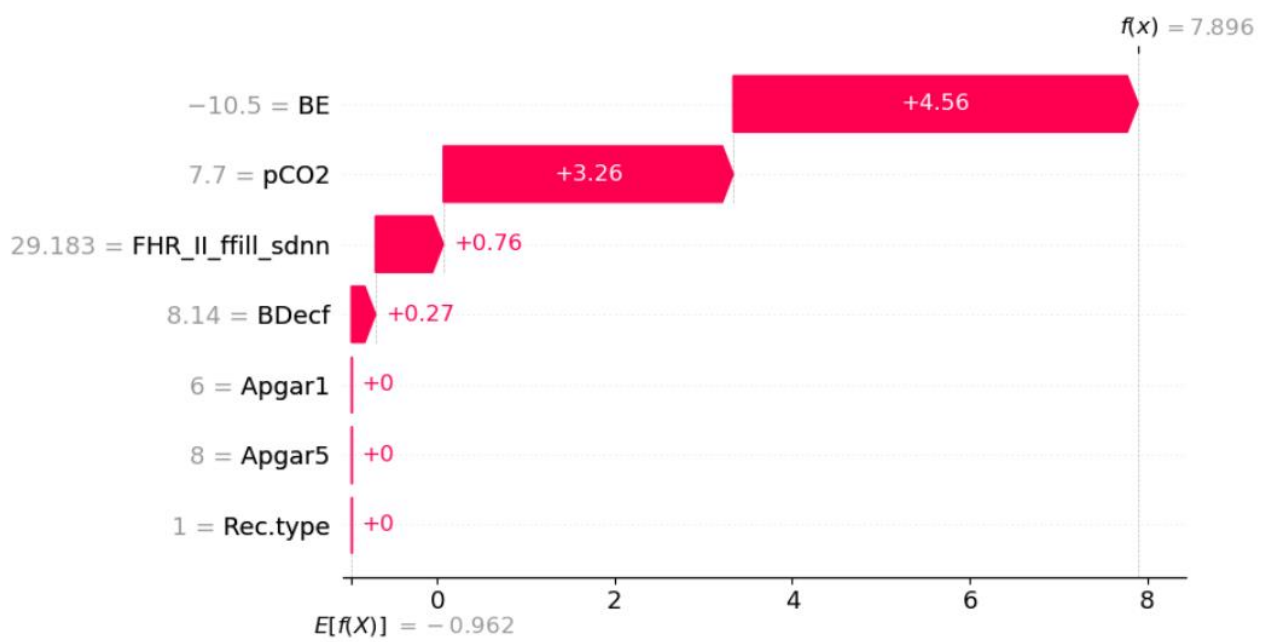
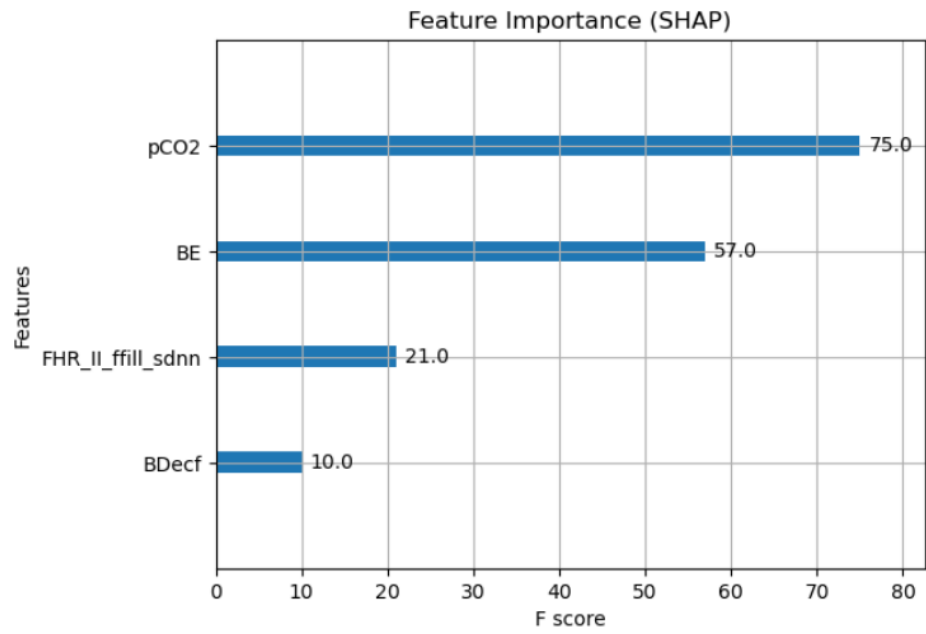
```
[[22  2]
 [ 0 21]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.92	0.96	24
1	0.91	1.00	0.95	21
accuracy			0.96	45
macro avg	0.96	0.96	0.96	45
weighted avg	0.96	0.96	0.96	45



Accuracy for all folds: [1. 0.97727273 0.95454545 0.97727273 0.95454545]
Mean Accuracy: 0.9727272727272727
F1 Score for all folds: [1. 0.97777778 0.95238095 0.97674419 0.95238095]
Mean F1 Score: 0.9718567737172389
Precision for all folds: [1. 0.95652174 0.95238095 0.95454545 0.95238095]
Mean Precision: 0.9631658196875588
Recall for all folds: [1. 0.95652174 0.95238095 0.95454545 0.95238095]
Mean Recall: 0.9631658196875588



Results Summary:

- Excellent performance for target '**ph risk**'
 - Good Performance for target '**Neonate Weight**' (but biased by the **NORM class**)
 - Mediocre to poor performance for targets '**Mother's Age**' and '**Gest Term**'
- ❖ The ExtraTreesClassifier based feature selection method is a better choice since it leads to better outcomes.

Some Possible Problems:

- Signals with useful information in Stage I were discarded due to too many zeros in Stage II
- Mistakes/Misunderstandings in pre-processing or the need for alternative methods to be applied at any part of that process.
- Individual and more refined processing being required that is assisted in part by relevant clinical information
- Lack of meaningful relationships amongst available data, requirement of further data extraction efforts.

Further work:

- Utilize the XGBClassifier data to explore the outcomes of removal or choice of features based on the SHAP information
- Manual feature extraction based on the overall information extracting via SHAP
- If that fails, further exploration of other machine learning models, feature extraction methods and data processing techniques.

