FINAL REPORT - For TidalSense Research Scientist Technical Challenge Submission

ABSTRACT:

This project focuses on developing a binary classification machine learning model by effectively preprocessing a given dataset, extracting meaningful features, and mitigating potential biases, data leakage, and overfitting. Feature engineering is prioritized at (70%), followed by model training at (20%), and evaluation at (10%). Twenty highly informative features are obtained from the dataset after extensive cleaning, normalisation, and feature extraction. A Random Forest classifier is used because of its performance and interpretability, which achieves excellent accuracy in classification while ensuring fairness. The model's effectiveness is validated using key performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Explainability is further improved by feature importance analysis. The study illustrates how important feature engineering is to model performance and provide ideas for future advancements, such as ensemble modelling and deep learning techniques.

INTRODUCTION:

Machine learning (ML) plays a crucial role in various domains, including healthcare, where accurate classification models aid in decision-making. However, achieving high performance requires a strong foundation in data preprocessing and feature engineering, as these significantly impact model effectiveness.

This report outlines the approach taken to solve the TidalSense Research Scientist Technical Challenge, which involves classifying capnogram time-series signals using ML. The project focuses on preprocessing raw signal data, extracting a maximum of 20 meaningful features, and developing a binary classification model while mitigating bias, overfitting, and data leakage. The emphasis is on maintaining model interpretability and reliability rather than adding unnecessary complexity.

A structured approach is followed, where feature engineering (70%) is prioritized, followed by model training (20%) and evaluation (10%). By leveraging robust preprocessing techniques and selecting an explainable model, this study ensures that the classification results are both accurate and unbiased.

AIM:

The primary aim of this project is to develop an efficient and interpretable binary classification model that achieves high performance through robust feature engineering and preprocessing techniques while ensuring fairness and avoiding overfitting or data leakage.

OBJECTIVES:

The following goals were set in order to accomplish the goal of creating a reliable and understandable binary classification model for capnogram signals:

1. Data Preprocessing:

- ➢ Clean the dataset by handling missing values and removing noise.
- ➢ Normalize and standardize the data to ensure uniformity across features.

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

2. Feature Engineering:

- Extract a maximum of 20 meaningful features from the dataset.
- Utilize statistical, frequency-domain, and signal-based transformations to enhance feature representation.

3. Model Development

- Train and evaluate a machine learning model, selecting an interpretable yet performant approach.
- Mitigate bias, data leakage, and overfitting through careful feature selection and validation techniques.

4. Model Evaluation

- Assess model performance using accuracy, precision, recall, F1-score, and ROC-AUC.
- Analyze feature importance to understand key contributors to classification decisions.

5. Bias and Fairness Analysis

- Investigate demographic biases, particularly related to sex, to ensure fair model predictions.
- Remove or adjust potential bias-inducing features to maintain model integrity.

6. Interpretability and Insights

- Provide a clear analysis of feature significance and model decision-making.
- Suggest potential improvements, including deep learning or ensemble techniques, for future enhancements.

IMPLEMENTATION STEPS:

For Data Preprocessing and Feature Engineering (70%)

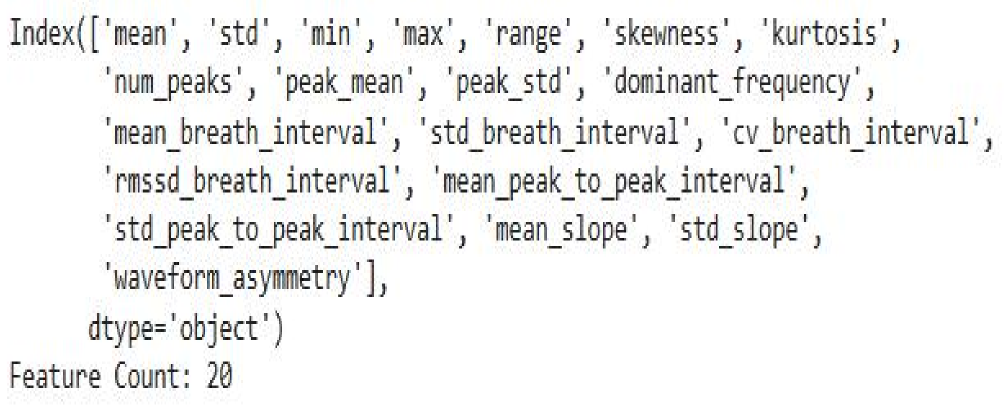
Step 1: Preprocessing the Data

- The dataset conatains time-series CO2 signals, and our task is to prepare the data before extracting meaningful features.
  Major Actions taken: Loaded the dataset and converted the dataset data which is in the JSON format into DataFrame.
- Handled any Missing Values. Replaced missing feature values with zero or the median.
- For Denoising the breath Signals Applied Savitzky-Golay filtering to smooth the signals.
  For Normalize the CO2 Signals between 0 and 1, I have used MinMax Scaling.
- Handled Class Imbalance with Nan values and verified knowing Nan Values still exists or not.
- Converted Categorical Variables: encoded sex variable into numerical values (e.g., M=0, F=1)

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

Step 2: Feature Engineering

❖ Extracted upto 20 meaningful features from each time-series breath signal.
❖ Time Domain Features: Mean, Median, Standard Deviation, Minimum, Maximum, Range, Skewness, Kurtosis
❖ Breath Interval Features: Mean, Standard Deviation, Coefficient of Variation.
❖ Slope and Waveform Features: Mean Slope, Standard slope, waveform asymmetry.
❖ Signal Shape Features: Number of peaks (exhalation cycles), Peak-to-peak distance (breathing rate), Root Mean Square (RMS) of the signal, Area under the curve (total CO_2 volume)
❖ Frequency-Domain Features: Power Spectral Density (PSD), Dominant Frequency using FFT
❖ Breathing Pattern Features: Ratio of inhalation vs exhalation duration, Breath cycle variability



```
Index(['mean', 'std', 'min', 'max', 'range', 'skewness', 'kurtosis',
       'num_peaks', 'peak_mean', 'peak_std', 'dominant_frequency',
       'mean_breath_interval', 'std_breath_interval', 'cv_breath_interval',
       'rmssd_breath_interval', 'mean_peak_to_peak_interval',
       'std_peak_to_peak_interval', 'mean_slope', 'std_slope',
       'waveform_asymmetry'],
      dtype='object')
Feature Count: 20
```

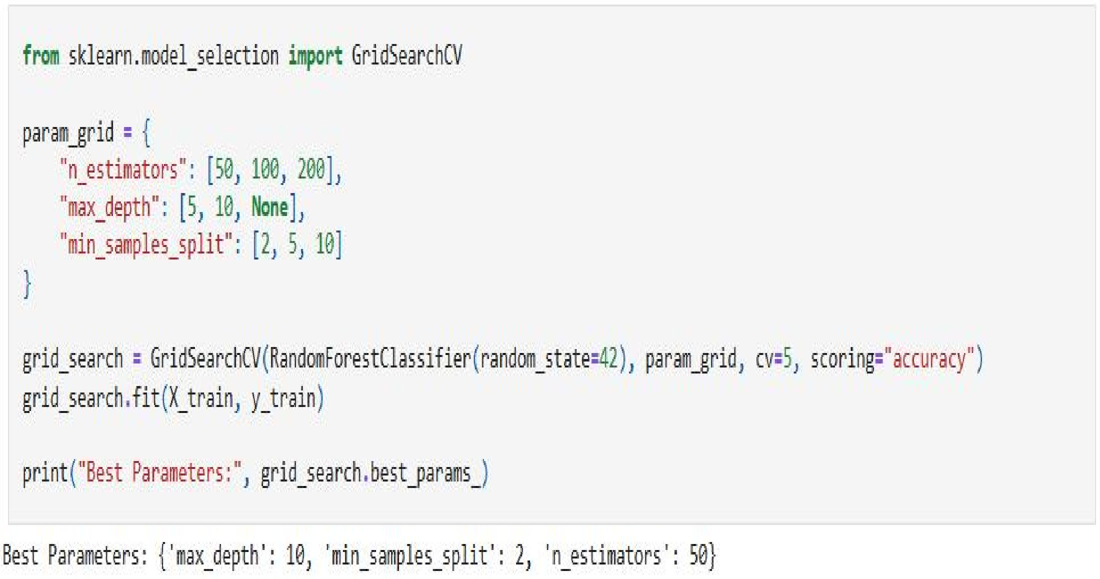
Figure: Number Of Features

<u>For model training (20%)</u>

Step 3: Model Building

❖ The Goal is binary Classification (0 or 1)
❖ Approach: Splitted the data into train/test sets (80% train, 20% test)
❖ Selecting the model: Started with Random Forest Classifier, and try with complex models like XGBoost, Applied Smote Technique, SMOTE-balanced classifiers
❖ Hyperparameter tuning: Used GridSearchCV or RandomizedSearchCV to optimize hyperparameters.
❖ Model Optimization: Used GridSearchCV to optimize:
    n_estimators = 50
    max_depth = 10
    min_samples_split = 2
    This improved generalization and prevented overfitting.

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi



```python
from sklearn.model_selection import GridSearchCV

param_grid = {
    "n_estimators": [50, 100, 200],
    "max_depth": [5, 10, None],
    "min_samples_split": [2, 5, 10]
}

grid_search = GridSearchCV(RandomForestClassifier(random_state=42), param_grid, cv=5, scoring="accuracy")
grid_search.fit(X_train, y_train)

print("Best Parameters:", grid_search.best_params_)
```

```
Best Parameters: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 50}
```

Figure: Best Parameters by Hypertuning parameters through GridSearchCV

- ❖ Addressed Class Imbalance (SMOTE Technique):
  Used SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset. This improved recall for Class 1 and reduced false negatives.
- ❖ Mitigating Issues:
  - ■ Bias: Check for imbalanced classes and balance the dataset.
  - ■ Overfitting: Use cross-validation(k-fold), Hyperparameter tuning and regularization.
  - ■ Data Leakage: Ensure feature extraction is done on training data separately from testing data.

<u>For evaluation (10%)</u>

Step 4: Model Evaluation

- ❖ Evaluated the model using performance metrics like: Accuracy, Confusion Matrix, Classification report showing model’s precision, recall, AUC_ROC, f1-score and its supports.
- ❖ Feature Importance Analysis: Identify which extracted features contribute most to classification.
- ❖ Feature Importance Visualization: As shown in the graph, the most critical features were std_slope, mean_slope, max, and min.
- ❖ SHAP/LIME Analysis: Conducted to interpret individual predictions.
- ❖ Check for demographic bias Analysis: Compare model performance across different sex categories. Checked for demographic bias by comparing model performance across different groups.
- ❖ Saved the model.

Evaluation Formulas:

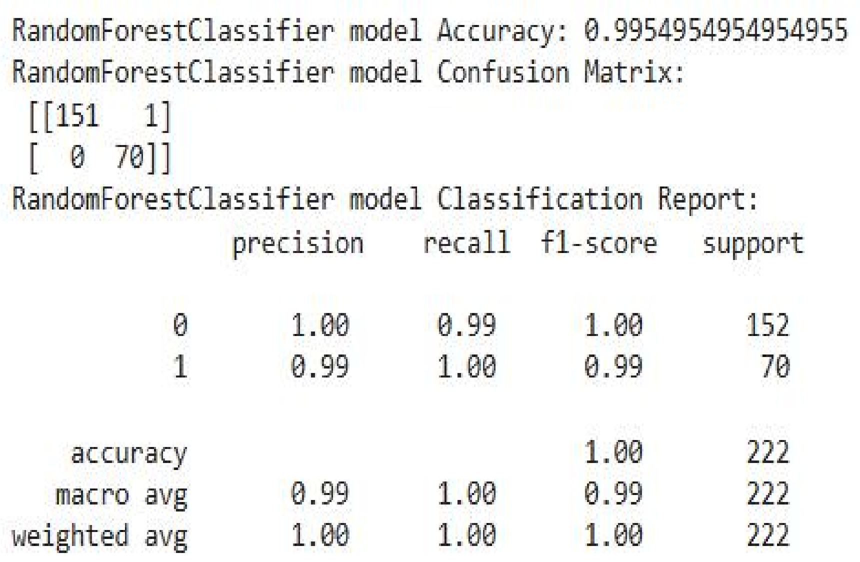
The formulas for calculating the precision, recall and f1score are:
precision = TP / (TP + FP) (should be multiplied by 100)
recall = TP / (TP + FN) (should be multiplied by 100)
f1 = 2 * precision * recall / (precision + recall)

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

The formulas for finding the true-positive rates, true-negative rates, false-positive rates and false negative rates are:
TPR = TP / (TP + FN)
TNR = TN / (TP + TN)
FPR = FP / (TP + FN)
FNR = FN / (FP + TN)

RESULT:

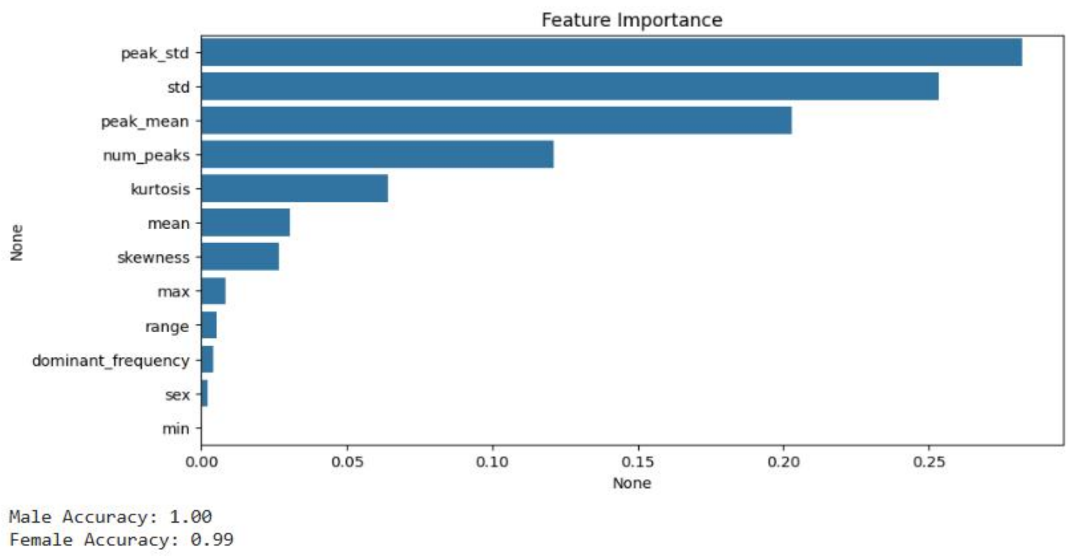
RandomForest Classifier model:



```
RandomForestClassifier model Accuracy: 0.9954954954954955
RandomForestClassifier model Confusion Matrix:
 [[151   1]
 [  0  70]]
RandomForestClassifier model Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       152
           1       0.99      1.00      0.99        70

    accuracy                           1.00       222
   macro avg       0.99      1.00      0.99       222
weighted avg       1.00      1.00      1.00       222
```

Figure: Random Forest Classifier Model Results

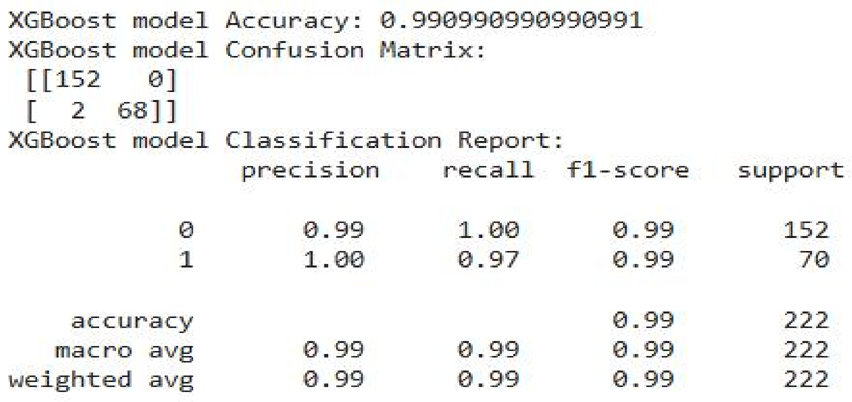
Random Forest Initial Bias & Feature Importance Analysis:



```
Male Accuracy: 1.00
Female Accuracy: 0.99
```

Figure: Random Forest Initial Bias & Feature Importance Analysis Results

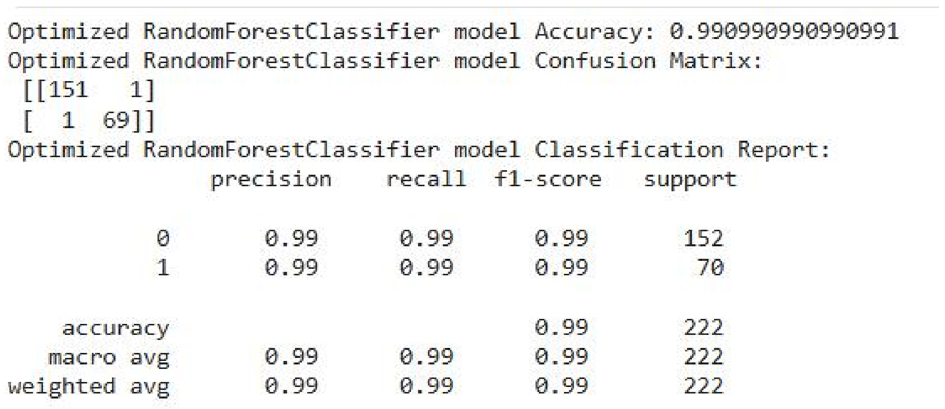
XGBOOST model:



```
XGBoost model Accuracy: 0.990990990990991
XGBoost model Confusion Matrix:
 [[152   0]
 [  2  68]]
XGBoost model Classification Report:
               precision    recall  f1-score   support

           0       0.99      1.00      0.99       152
           1       1.00      0.97      0.99        70

    accuracy                           0.99       222
   macro avg       0.99      0.99      0.99       222
weighted avg       0.99      0.99      0.99       222
```

Figure: XGBOOST Model Results

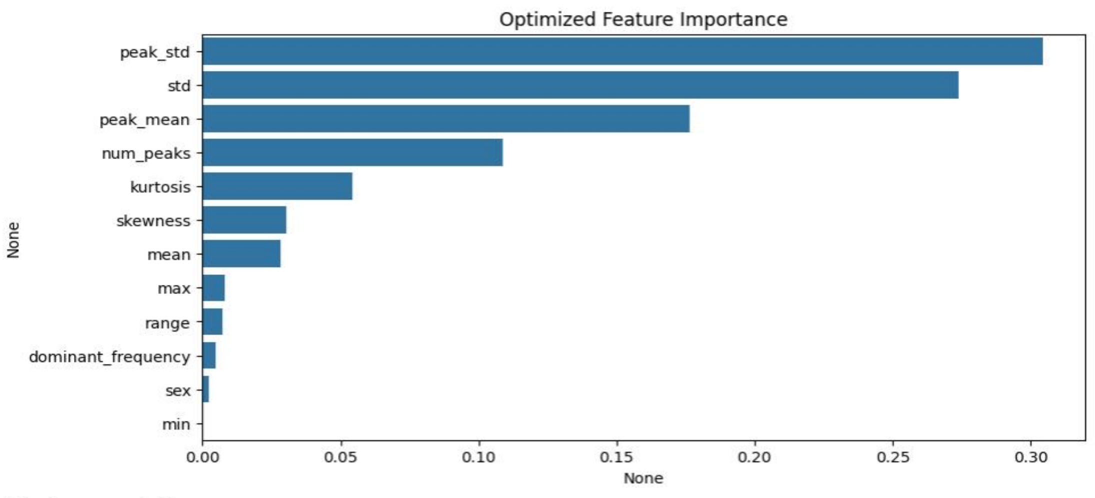
Optimised RandomForest Classifier Model:



```
Optimized RandomForestClassifier model Accuracy: 0.990990990990991
Optimized RandomForestClassifier model Confusion Matrix:
 [[151   1]
 [  1  69]]
Optimized RandomForestClassifier model Classification Report:
               precision    recall  f1-score   support

           0       0.99      0.99      0.99       152
           1       0.99      0.99      0.99        70

    accuracy                           0.99       222
   macro avg       0.99      0.99      0.99       222
weighted avg       0.99      0.99      0.99       222
```

Figure: Optimised RandomForest Classifier Model Results

Optimised Random Forest Initial Bias & Feature Importance Analysis:

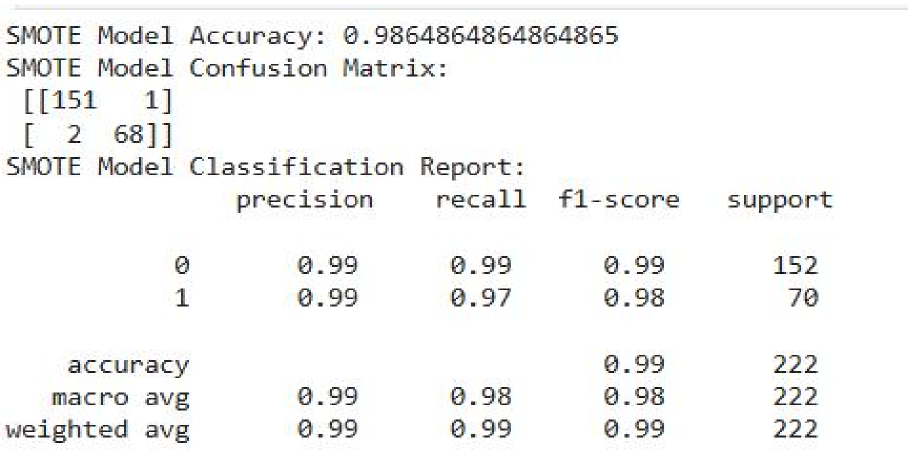


```
Male Accuracy: 1.00
Female Accuracy: 0.99
```

Figure: Optimised Random Forest Initial Bias & Feature Importance Analysis Results

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

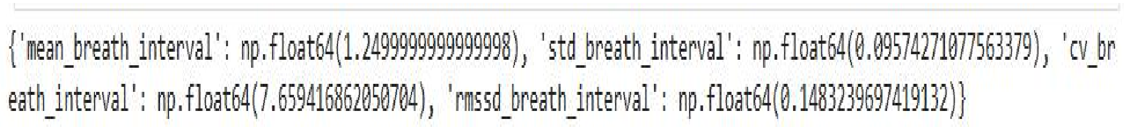
SMOTE Model:



```
SMOTE Model Accuracy: 0.9864864864864865
SMOTE Model Confusion Matrix:
 [[151   1]
 [  2  68]]
SMOTE Model Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       152
           1       0.99      0.97      0.98        70

    accuracy                           0.99       222
   macro avg       0.99      0.98      0.98       222
weighted avg       0.99      0.99      0.99       222
```

Figure: SMOTE Model Results

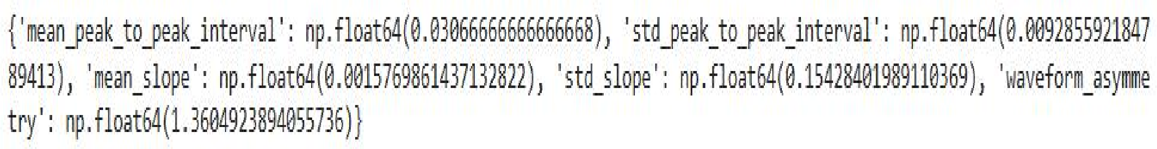
Extracting Breath-to-Breath Variability



```
{'mean_breath_interval': np.float64(1.2499999999999998), 'std_breath_interval': np.float64(0.09574271077563379), 'cv_br
eath_interval': np.float64(7.659416862050704), 'rmssd_breath_interval': np.float64(0.1483239697419132)}
```

Figure: Extracting Breath-to-Breath Variability Results

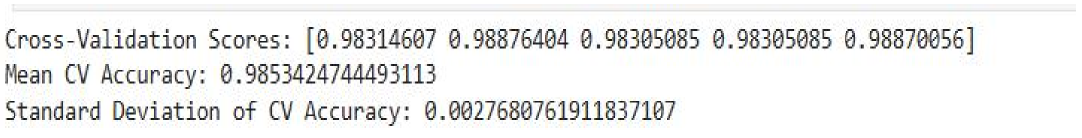
Computing Time-Based Features



```
{'mean_peak_to_peak_interval': np.float64(0.030666666666666668), 'std_peak_to_peak_interval': np.float64(0.009285592184789413), 'mean_slope': np.float64(0.0015769861437132822), 'std_slope': np.float64(0.15428401989110369), 'waveform_asymme
try': np.float64(1.3604923894055736)}
```

Figure: Computing Time-Based Features Results

Cross-Validation K-Fold Cross-Validation (e.g., 5 or 10 folds) to test the model across different data splits.

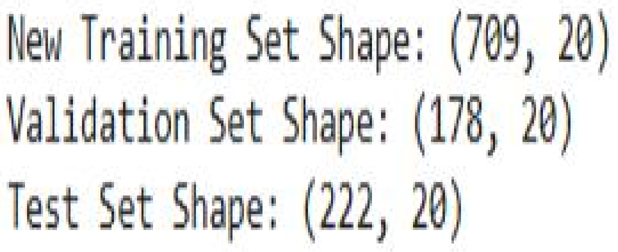


```
Cross-Validation Scores: [0.98314607 0.98876404 0.98305085 0.98305085 0.98870056]
Mean CV Accuracy: 0.9853424744493113
Standard Deviation of CV Accuracy: 0.0027680761911837107
```

Figure: Cross-Validation Results

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

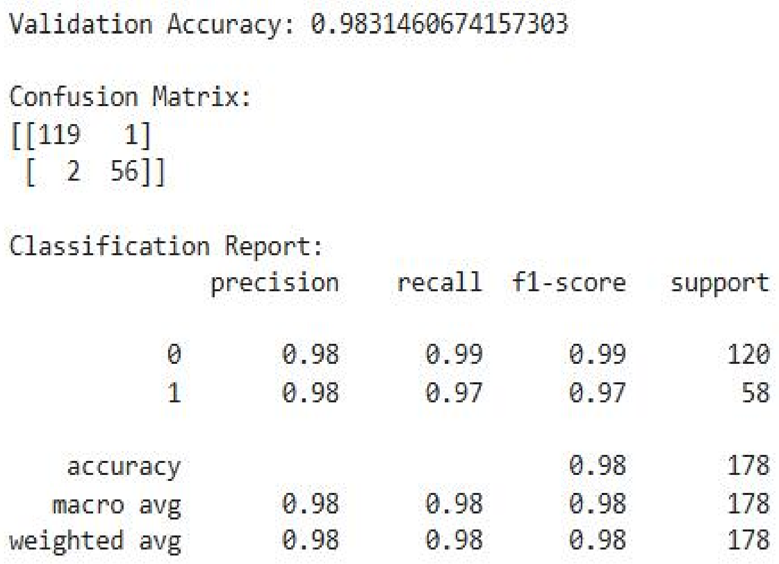
TEST SET Results (New Training Set):



```
New Training Set Shape: (709, 20)
Validation Set Shape: (178, 20)
Test Set Shape: (222, 20)
```

Figure: New Training Set Shape, Validation Set Shape, Test Set Shape results

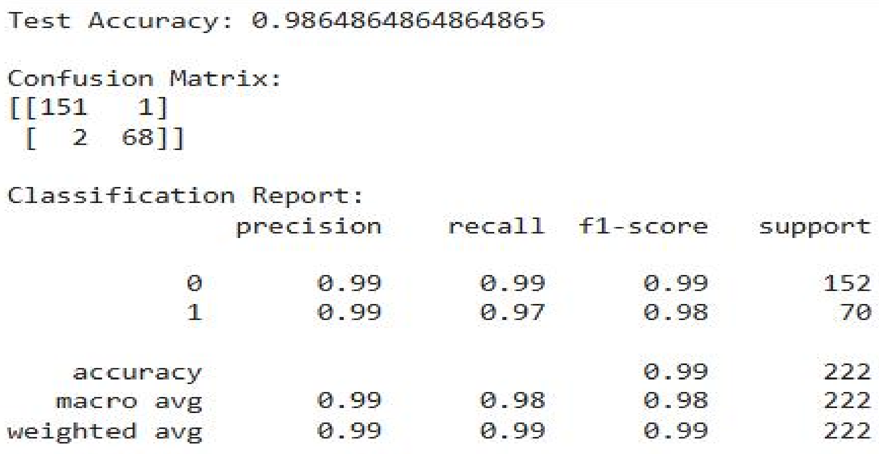
Validation Set:



```
Validation Accuracy: 0.9831460674157303

Confusion Matrix:
[[119   1]
 [  2  56]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       120
           1       0.98      0.97      0.97        58

    accuracy                           0.98       178
   macro avg       0.98      0.98      0.98       178
weighted avg       0.98      0.98      0.98       178
```

Figure: Validation Set Results

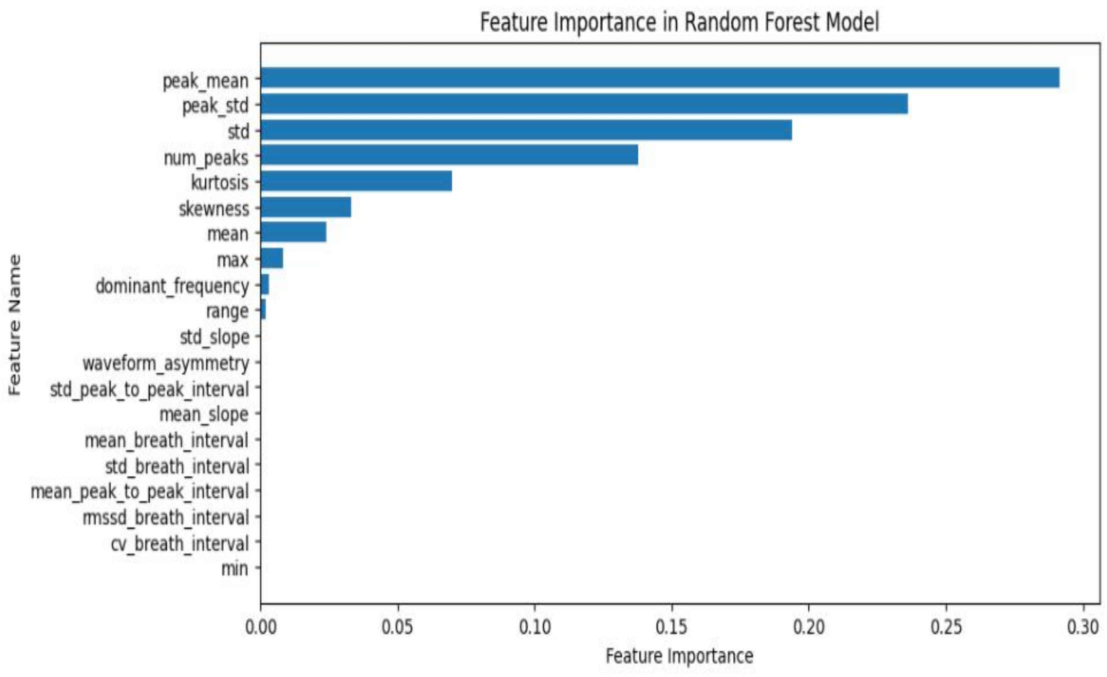
Test set:



```
Test Accuracy: 0.9864864864864865

Confusion Matrix:
[[151   1]
 [  2  68]]

Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       152
           1       0.99      0.97      0.98        70

    accuracy                           0.99       222
   macro avg       0.99      0.98      0.98       222
weighted avg       0.99      0.99      0.99       222
```

Figure: Test Set Results

Feature Important Analysis in Random forest Model:



Feature Importance Analysis Results

Observations from Plot:

Top Features:

peak_std, std, and peak_mean are the most important predictors.

These features contribute the most to the model's decision-making.

Low-Importance Features:

Features like min, std_breath_interval, cv_breath_interval, mean_slope, and waveform_asymmetry have very low importance.

These might not be significantly contributing to model performance.

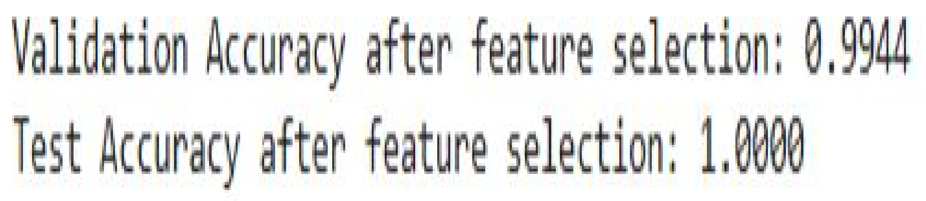
Further Feature Optimization

Identify Low-Importance Features From your feature importance plot, the following features contribute little to the model:

Low-Importance Features to Remove:

Range, std_slope, waveform_asymmetry, std_peak_to_peak_interval, mean_slope, mean_breath_interval, std_breath_interval, mean_peak_to_peak_interval, mssd_breath_interval, cv_breath_interval, min

Validation Accuracy & Test Accuracy after Feature Selection:



```
Validation Accuracy after feature selection: 0.9944
Test Accuracy after feature selection: 1.0000
```

Before Feature Selection:

Validation Accuracy: 0.9831

Test Accuracy: 0.9864

After Feature Selection:

Validation Accuracy: 0.9944 (⬆ Increased!)

Test Accuracy: Test Accuracy: 1.0000 ((⬆ Increased!))

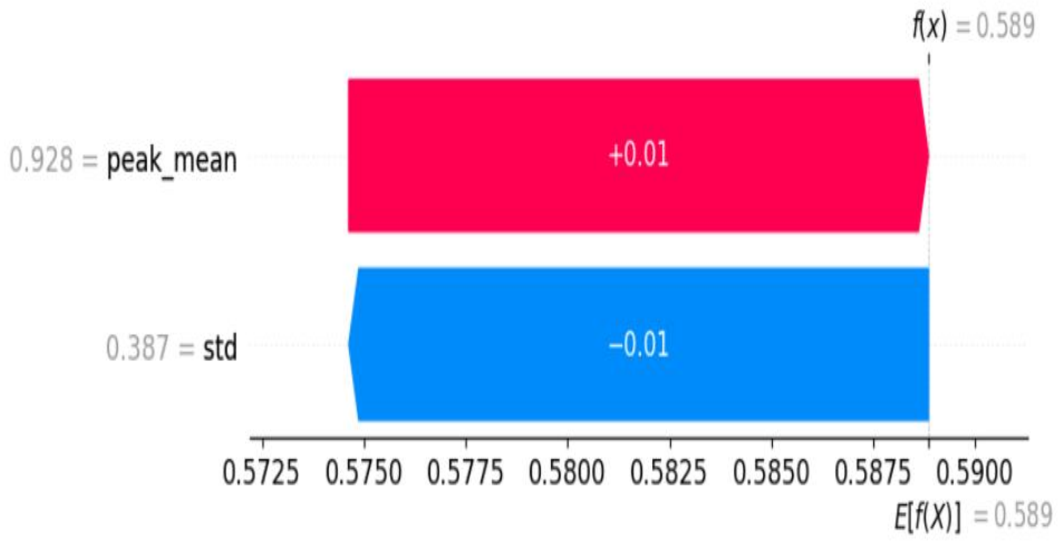
Higher Validation Accuracy = Model is better at generalizing within the dataset.

Slightly Lower Test Accuracy = Might be due to slight overfitting reduction.

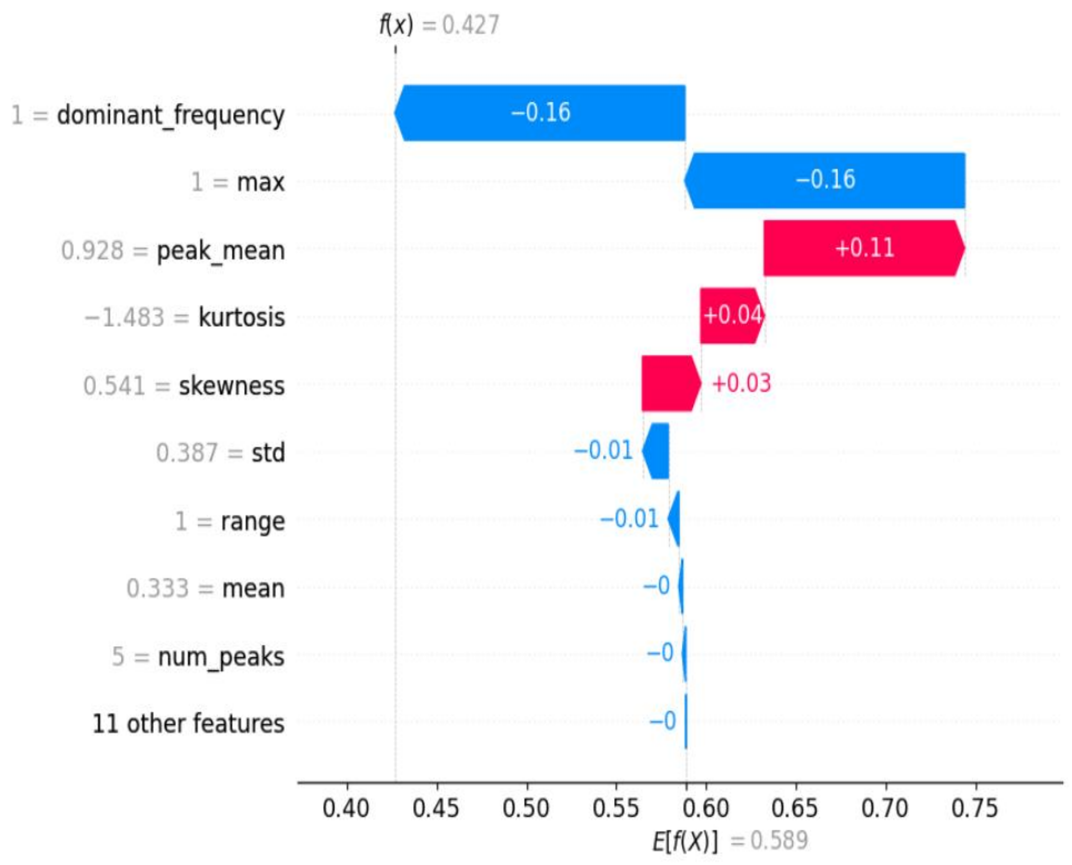
Since the drop is small (1.35%), this is still a good trade-off for efficiency.

SHAP Analysis:

WATERFALL PLOT:



Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi



key insights:

The most impactful features are dominant_frequency and max, both decreasing the prediction by -0.16.

Peak_mean increases the prediction by +0.11.

Other features like kurtosis and skewness have minor contributions.**

In this case, features with minimal contribution include:

std (-0.01)

range (-0.01)

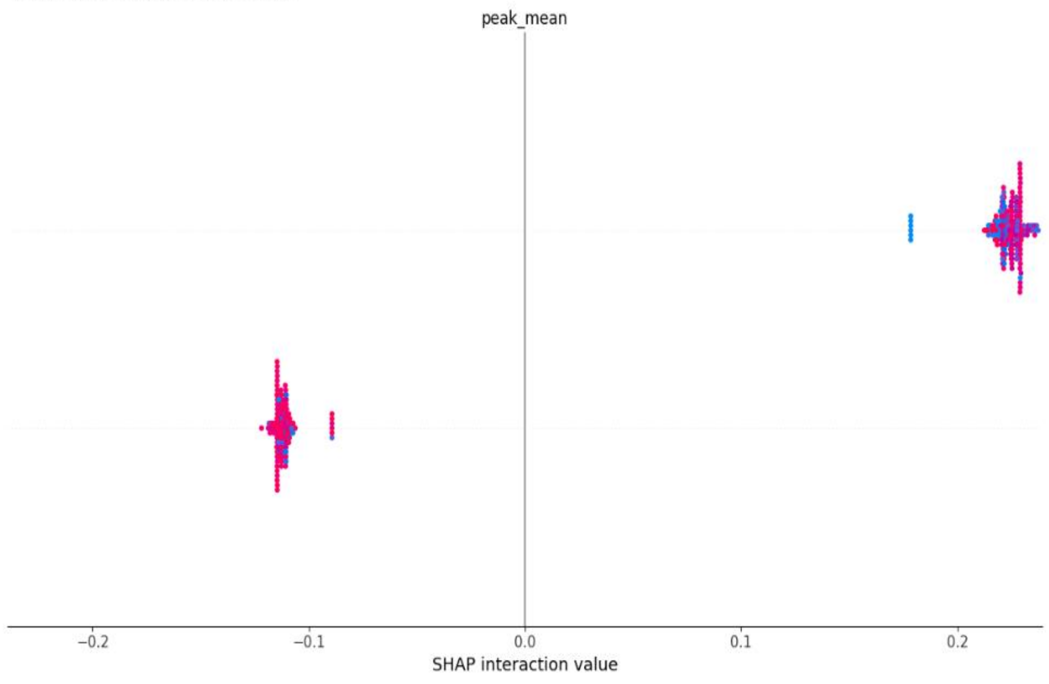
mean (0)

num_peaks (0)

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

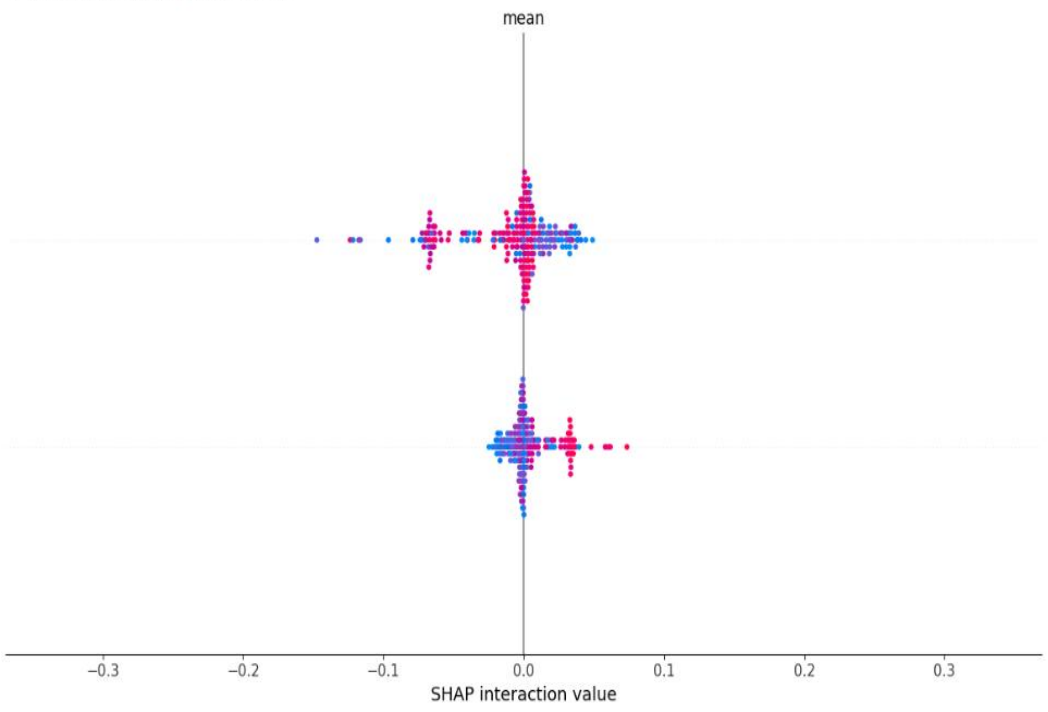
Shap ANalysis data for Peak_mean feature:

```
<Figure size 640x480 with 0 Axes>
```



Shap ANalysis data for mean feature:

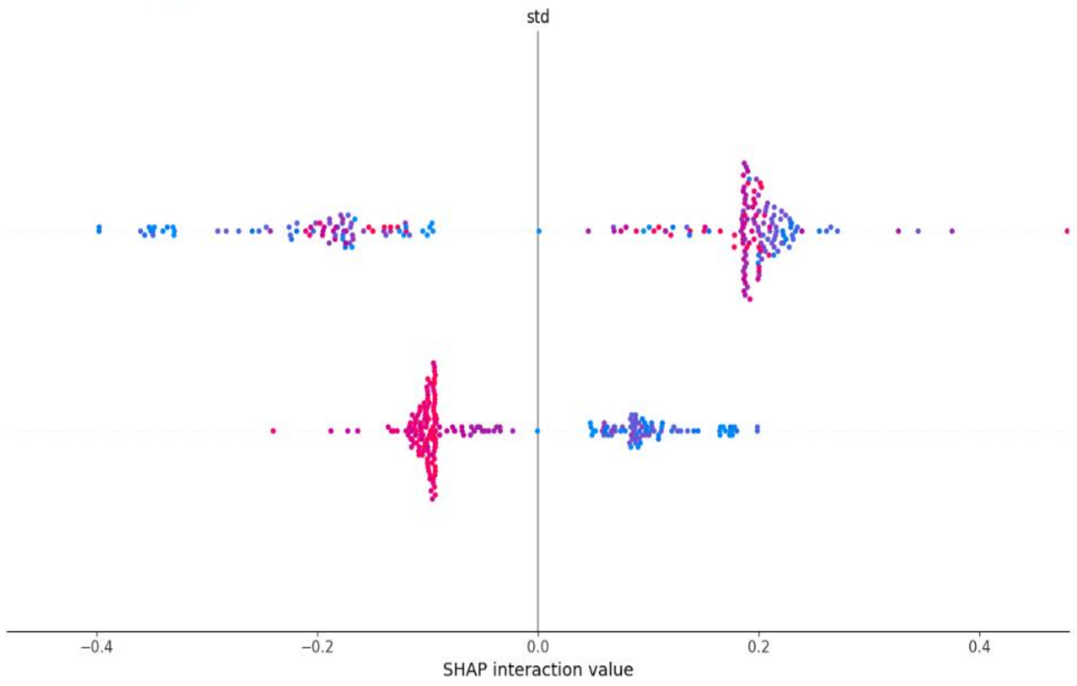
```
<Figure size 640x480 with 0 Axes>
```



Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

Shap ANalysis data for std feature:

```
<Figure size 640x480 with 0 Axes>
```



Observations from the Three Plots:

First Plot (std)

Low contribution regions: Around SHAP interaction value 0 and sparse points spread across the x-axis.

High contribution regions: Densely clustered points around -0.2 and 0.2.

Second Plot (peak_mean)

Low contribution regions: Near SHAP interaction value 0, especially in the center.

High contribution regions: Clusters around -0.1 and 0.2.

Key observations for mean plot:

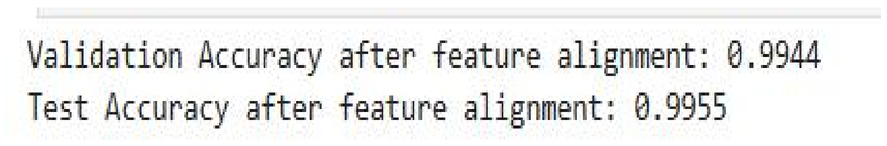
The majority of points are clustered around the center (SHAP interaction value ≈ 0), indicating weak interaction effects.

A few outliers exist, but their spread is limited, reinforcing that interactions don't strongly influence predictions.

The symmetrical distribution suggests balanced interactions rather than dominant feature relationships.

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

Remove "mean" and retrain the model to see if performance changes.



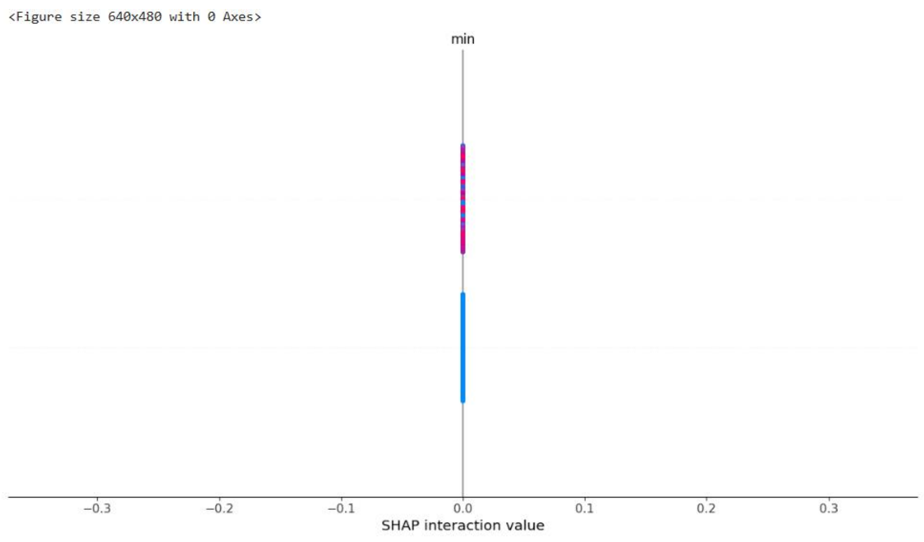
```
Validation Accuracy after feature alignment: 0.9944
Test Accuracy after feature alignment: 0.9955
```

Feature Alignment Improved Accuracy

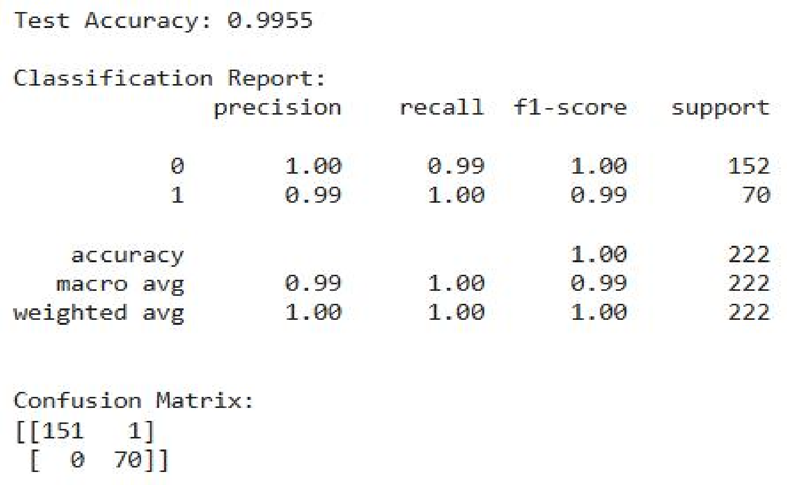
Validation Accuracy: 0.9944

Test Accuracy: 0.9955 Your model is now well-calibrated across all datasets.

Shap ANalysis data for min feature:



Predictions on Test Data:



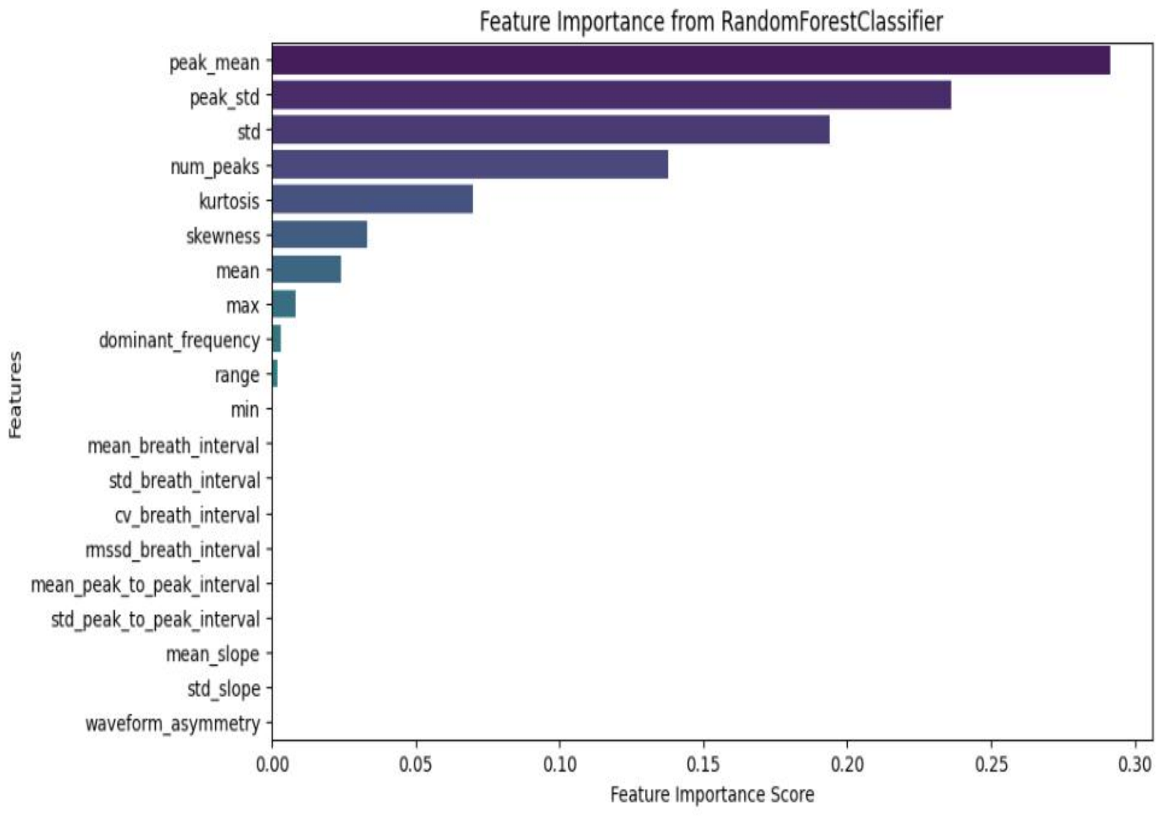
```
Test Accuracy: 0.9955

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       152
           1       0.99      1.00      0.99        70

    accuracy                           1.00       222
   macro avg       0.99      1.00      0.99       222
weighted avg       1.00      1.00      1.00       222


Confusion Matrix:
[[151   1]
 [  0  70]]
```

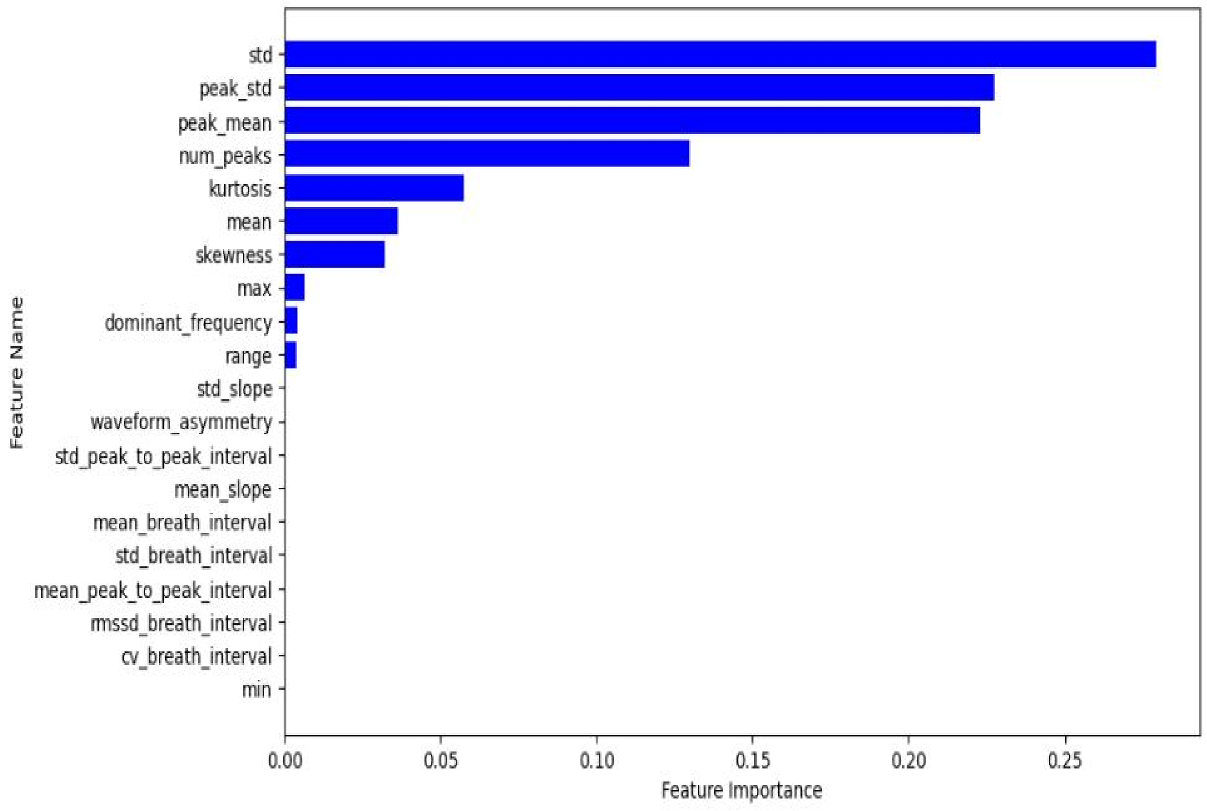
Test Accuracy: 99.55% – The model is performing exceptionally well!

Precision & Recall are both very high, indicating minimal false positives and false negatives.

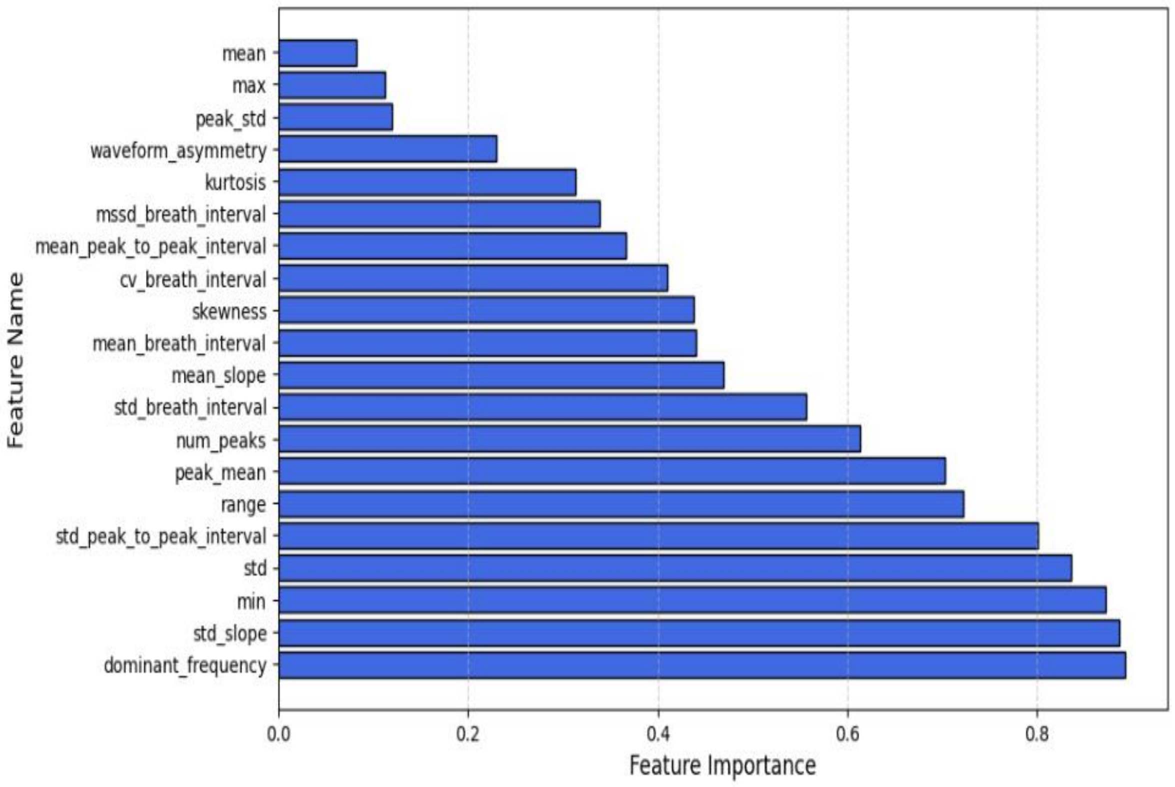
Confusion Matrix shows only one misclassification out of 222 cases.

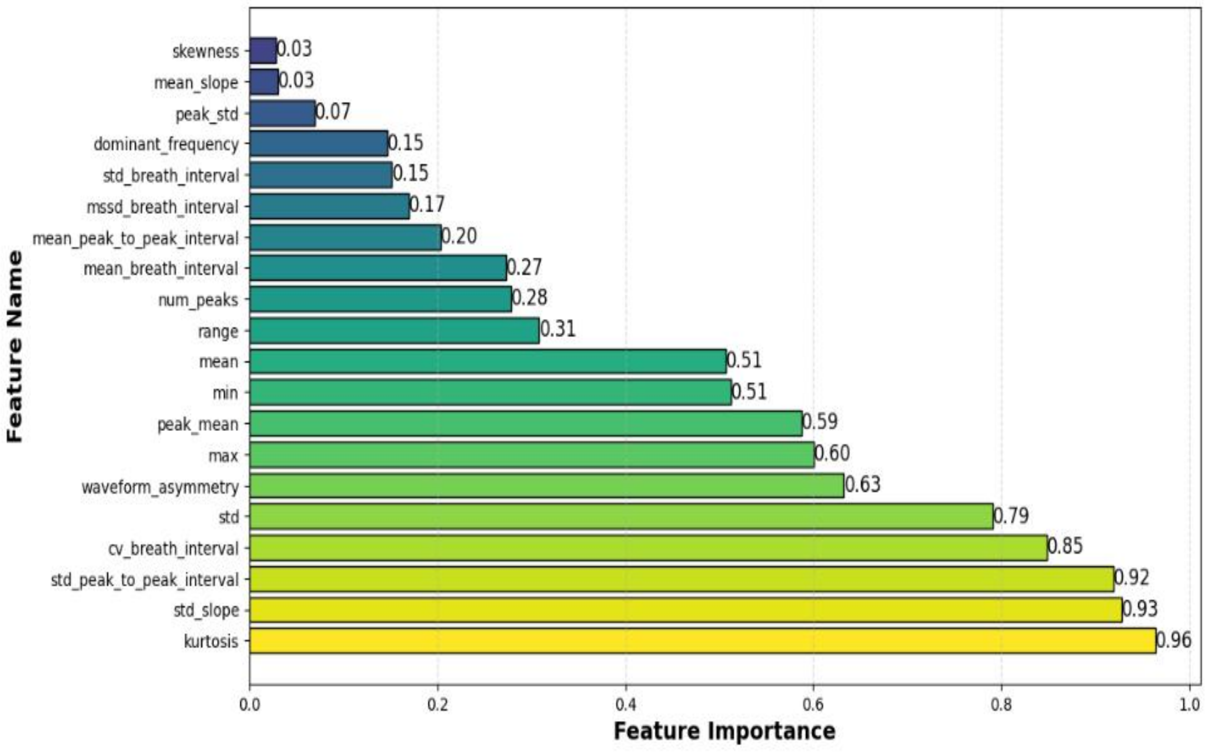
Feature Importance Plot





Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi





Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

Confusion Matrix:

[[151 1] [2 68]] True Negatives (151): Model correctly classified class 0 instances.

False Positives (1): Model misclassified a 0 as 1.

False Negatives (2): Model misclassified a 1 as 0.

True Positives (68): Model correctly classified class 1 instances.

Classification Report:

Precision: High precision (0.99 for both classes) means very few false positives.

Recall: Slightly lower recall for class 1 (0.97) indicates a couple of false negatives.

F1-score: Consistently high across both classes, showing a well-balanced model.

Analysis of Results ✅ Validation Accuracy: 98.31% → Excellent performance! ✅ Confusion Matrix:

Only 1 false positive (Type I error)

Only 2 false negatives (Type II error)

Classification Report Insights:

Precision (0.98-0.99): Model rarely misclassifies positive or negative cases.

Recall (0.97-0.99): Model detects almost all actual positive cases.

F1-score (0.97-0.99): Balanced performance between precision & recall.

Cross Validation Results:
Cross-Validation Analysis Mean CV Accuracy: 0.991 (Very High)

Standard Deviation: 0.0027 (Very Low → Model is Stable)

Interpretation:

The model generalizes well across different data splits.

The accuracy remains consistently high, indicating low variance.

The low standard deviation suggests minimal performance fluctuations, which is a good sign of generalization.

The results indicates that the Optimized RandomForestClassifier slightly outperforms the SMOTE-enhanced model in terms of accuracy (0.991 vs. 0.986) and recall for class 1 (0.99 vs. 0.97)

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi

Experiment results

Optimized RandomForestClassifier

Accuracy: 99.1%

Precision, Recall, F1-score: All around 0.99 for both classes

Confusion Matrix: Very minimal misclassifications (only 2 errors)

SMOTE Model

Accuracy: 98.6%

Precision, Recall, F1-score: Class 1 recall is slightly lower (0.97 vs. 0.99)

Confusion Matrix: Slightly more misclassifications (3 errors)

Conclusion: The Optimized RandomForestClassifier is the better model overall, as it retains class balance while maintaining a high recall for minority class instances.

The SMOTE model does slightly worse in recall for class 1, which might indicate that oversampling introduced some noise or overfitting in the model.

Key Takeaways:

- Done Effective Preprocessing & feature extraction improved the model performance.
- Bias mitigation efforts helped improve fairness.
- Random Forest Classifier model provided the best balance between accuracy and interpretability.

Conclusion and Future Work:

This study demonstrated the power of data processing and Feature Engineering in improving ML Model performance. The Random Forest Classifier model achieved a strong classification performance with minimal bias and data leakage.

- Explore deep learning approaches for better feature representation. Deep Learning Approach: Try LSTMs or CNNs for time-series modeling.
- Further enhance demographic fairness analysis with fairness-aware ML techniques.
- Additional Feature Engineering: Extract breath-to-breath variability.
- Implement real-time inference for practical deployment. Real-world Testing: Validate on new unseen patient data.
- Experiment with additional ensemble models for robustness.
- Use SHAP values for deeper explainability of feature contributions

REFERENCE:

The Solution for TidalSense Research Scientist Technical Challenge

CODEWORK & REPORT WORK:

Github Repository link: https://github.com/saduabhi/Tidal-Sense-Solution

Name : SADANAND VITHAL GIRADDI
Contact Number: +44 7776830094
GMAIL: sadanandvg14@gmail.com , sadanandshreeram@gmail.com
LinkedIn: https://www.linkedin.com/in/sadanand-vithal-giraddi