DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY,
GANDHINAGAR , GUJARAT, INDIA

Classification of Heartbeat Anomalies

Prof. Arpit Rana

202418026 - Krisha Sompura 202418035 - Milan Nagvadiya

202418041- Palak Jain 202418051- Sheetal Jain

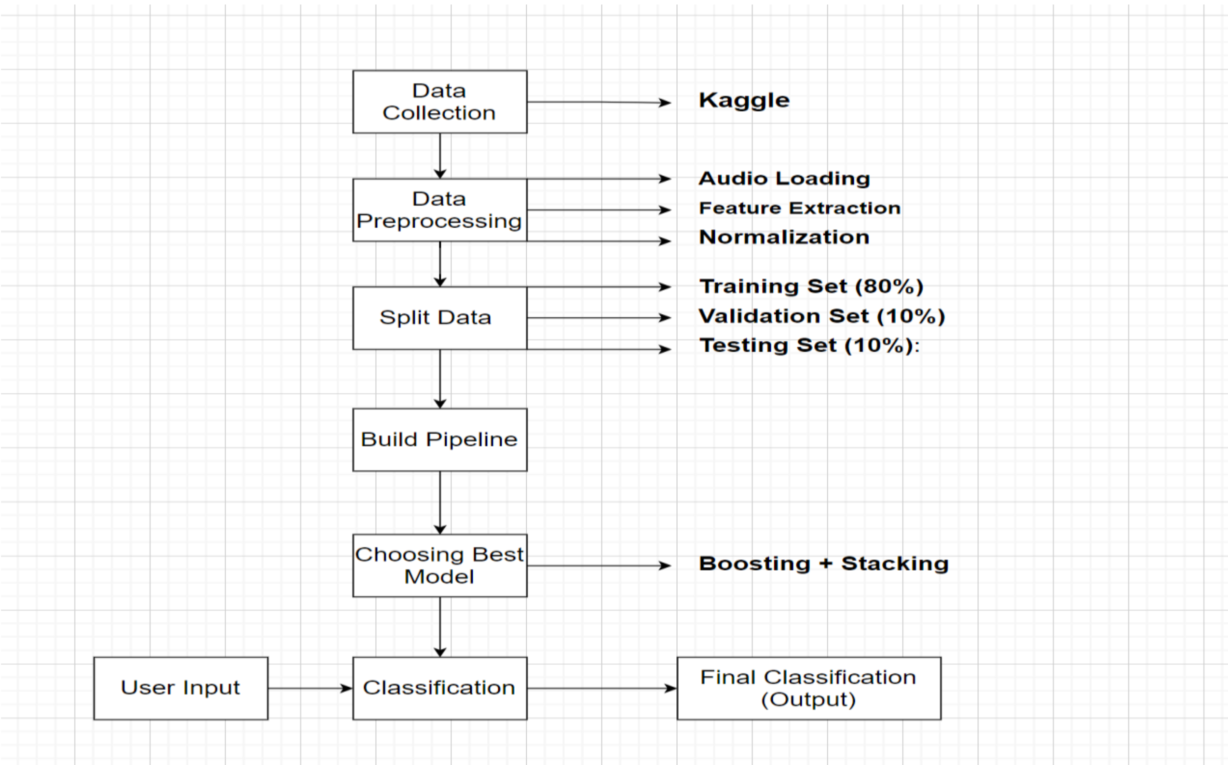
Overview :

This project focuses on using machine learning to classify heartbeat anomalies from audio recordings. By analyzing these recordings, we aim to detect irregular heartbeats that could indicate medical conditions like Murmur , Artifact , Normal , Extrastole and Extrahls.

Problem Statement :

The accurate classification of heartbeat anomalies can assist in early diagnosis and treatment of heart conditions. Traditional diagnosis relies on doctors' experience while machine learning can provide automated and scalable solutions.

Proposed Architecture :



Data Collection :

We have taken the dataset from the datasets from Kaggle.

Link : https://www.kaggle.com/code/brsdincer/heartbeat-sounds-classification-analysis/input

Exploratory Data Analysis :

We have plotted graphs like Original Sound Wave , FFT Spectrum , Liborsa Waveform , MFCCs for each type of heartbeat . Also , we have done comparision for every graph for normal vs other by using librosa library.

Data Preprocessing :

- **Audio Loading**: Loaded the audio files using librosa to handle varying formats and sample rates.
- **Feature Extraction**: Extracted key audio features like **MFCCs (Mel-frequency cepstral coefficients)**, **Spectrograms**, **Chroma**, and **Zero Crossing Rate**, which help capture the rhythmic and spectral properties of the heartbeat sounds.
- **Normalization**: Standardized the extracted features to ensure they are on a similar scale, improving model performance.

Data Splitting

- **Training Set (80%)**: Used for training the model.
- **Validation Set (10%)**: Used for tuning hyperparameters and evaluating the model during training.
- **Testing Set (10%)**: Used for assessing the final performance of the model on unseen data.

Model Evaluation :

Model	precision	recall	f1-score	accuracy
RandomForest Classifier	0.84	0.77	0.75	0.77
SVC	0.78	0.77	0.77	0.77
HistGradientBoosting Classifier	0.82	0.78	0.78	0.78
Staking	0.84	0.78	0.77	0.78
Voting	0.81	0.78	0.77	0.78

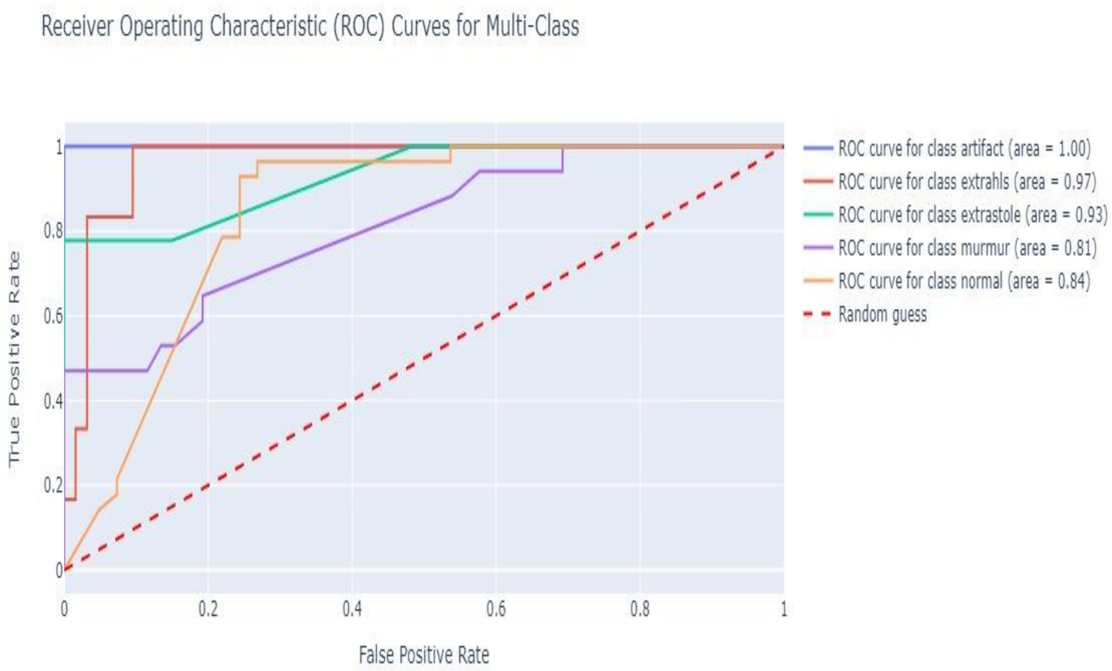
Boosting and Stacking	0.88	0.78	0.76	0.78

Boosting and Stacking :

We applied Boosting to individual models (RandomForest, SVC, and HistGradientBoosting).Then, we used these models as base models for stacking. We chose a OneVsRest logistic regression model as the meta-model.

Best Model	**precision**	**recall**	**f1-score**	**accuracy**
Boosting + Stacking	0.88	0.78	0.76	0.78

ROC Curve



Pseudocode :

Model : Final model(Boosting +Stacking) :

Step 1: Define Base Estimators

Create three base classifiers: HistGradientBoosting, RandomForest, and SVC

Define hist_gradient_boosting with specified parameters

Define random_forest with specified parameters

Define svc with specified parameters (use RBF kernel, set probability=True)

Step 2: Wrap Base Estimators in AdaBoost

Wrap each base classifier in AdaBoost with SAMME algorithm and 50 estimators

Wrap hist_gradient_boosting in AdaBoost -> ada_hist

Wrap random_forest in AdaBoost -> ada_rf

Wrap svc in AdaBoost -> ada_svc

Step 3: Create Stacking Classifier

Combine ada_hist, ada_rf, ada_svc into a StackingClassifier

Use OneVsRestClassifier(SVC with probability=True) as the final estimator

Create stacking_clf using ada_hist, ada_rf, ada_svc as estimators, and OneVsRestClassifier as final_estimator

Step 4: Train the Model

Fit stacking_clf with training data X_train and y_train

stacking_clf.fit(X_train, y_train)

Future Scope :

Model Improvement: Explore deep learning and transfer learning to enhance accuracy

Real-time Monitoring: Apply the model to wearable devices for continuous heartbeat tracking

Expanded Classification: Detect a broader range of heart conditions beyond current anomalies

Clinical Collaboration: Work with cardiologists to validate and refine the model.