Classifying Emotions from Sound Events

Shivank Singh Thakur, Rahul Reddy Gangapuram, Prudhvi Sai Raj Dasari, Jathin Shettigar Nagabhushan
[1]Computer Science Department, [2]College of Graduate Studies, [1,2]San Jose State University
{*shivanksingh.thakur, rahulreddy.gangapuram, prudhvisairaj.dasari, jathin.shettigarnagabhushan*}@*sjsu.edu*

[1]

Abstract—**Emotion recognition from sound is one of the most important capabilities of affective computing, enhancing human-computer interaction, entertainment, and therapy. This work assesses the performance of machine learning models to classify emotions based on two perceived dimensions i.e., arousal and valence. The EmoSounds and IADSED datasets are used to predict the perceived emotion. The datasets provide a variety of acoustic features and are preprocessed by handling missing values, feature scaling, and division of data. We implement and compare Logistic Regression, Random Forest and XGBoost models and estimate their performance using accuracy and F1 Score metrics. From the results, XGBoost has the best performance for F1 score and accuracy on the Emosounds dataset and Random Forest has the best performance over the other models with the highest F1 score and accuracy for IADSED dataset.**

Index Terms—**Decision Tree, Emotion Prediction, Logistic Regression, Random Forest, XGBoost**

I. Introduction

Sound-based emotion recognition is an important function in affective computing, which finds applications in human-computer interaction, therapy, and multimedia. These systems, which combine physical sensors with data processing capabilities, heavily rely on sound for communication and emotional recognition. Healthcare monitoring, autonomous vehicles, and human-robot interactions all utilize sound processing for various applications. In this work, two datasets, namely EmoSounds, and IADSED, are used to classify emotions based on arousal and valence values with Logistic Regression, Random Forest and XGBoost models. This study specifically delves into the classification of emotion by arousal and valence annotation of sound databases, with the aim of improving the machine's recognition and response to human emotional states.

A. Key contributions

- Rahul and Jathin worked on the EmoSounds dataset for classifying sounds based on arousal and valence values.
- Prudhvi and Shivank worked on the IADSED dataset for classifying sounds based on arousal and valence values.
- We leveraged Logistic Regression, Random Forest and XGBoost models to classify the sounds into their target class. As the number of features are high, 68 to be precise, we created the preprocessed dataset with the target class based on custom label encoding logic. We then used Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset and Recursive Feature Elimination (RFE) to eliminate less important features and then compared the before and after performance of the models based on accuracy and macro-F1 scores.

B. Organization of the Paper

The paper is organized as follows: Section I introduces the paper, and Section II describes the dataset. Section III explains the preprocessing techniques performed on the dataset. Section IV discusses the various classification models trained. Section V presents the results and evaluation. Section VI concludes the paper with the conclusion and future work.

II. Dataset Description

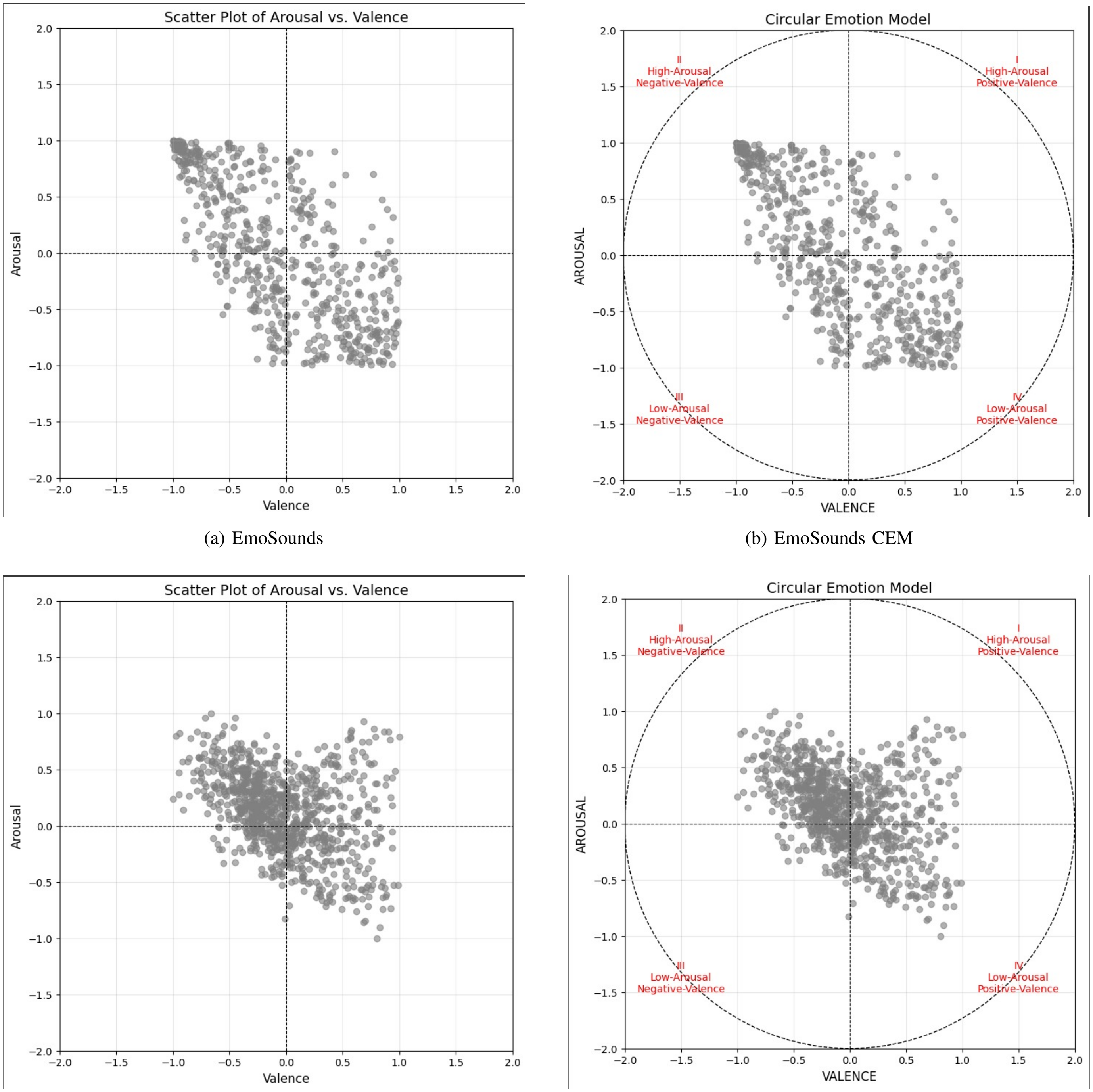
We worked on two datasets -

- **EmoSounds**: Contains numerical features in terms of pitch, rhythm, timbre, and tonal features of the sounds. There are 600 rows and 75 columns in this dataset.
- **IADSED**: Focuses on tonal clarity, mode, and other features related to emotional representation in sounds. There are 935 rows and 76 columns in this dataset.
- Both datasets have 68 features and arousal, valence as target columns but the IADSED dataset had an additional target dominance, all of which are commonly used in emotion analysis. However, for the scope of this research, we consider the two target features i.e, arousal and valence for both the datasets.

III. Preprocessing

1) **Handling Missing Values**: There are no missing NaN or null values in the EmoSounds dataset, whereas there are 27 missing cells and NaN values in the IADSED dataset. For dealing with the missing values, as the missing rows are less compared to the dataset size, we dropped the 8 rows that had missing cells bringing the number of rows to 927.
2) **Transforming data**: The Emosounds dataset is already in the range of [-1,1] hence we need not perform scaling on the Emosounds dataset. However, MinMaxScaler was used for scaling the features in the IADSED dataset. This method adjusts the values so they fit within a range of -1 to 1, including both the input features and target variables (arousal and valence). This helps machine learning models process the data more effectively by keeping everything on a consistent scale, preventing features from dominating others due to larger values. The distribution of the values is shown in Figure 1.

[1] https://www.overleaf.com/read/qqcckqkhtvjf#ac8053



(a) EmoSounds

(b) EmoSounds CEM

(c) IADSED

(d) IADSED CEM

Fig. 1: Distribution of Arousal and Valence in EmoSounds and IADSED Datasets

3) **Creating Target Label**: Based on the polarity of arousal and valence, we assigned classes as below for all the rows in the data and stored the labels in a new column ”class”.

   The quadrants are as below:

   - **Class 0**: Positive Arousal, Positive Valence.
   - **Class 1**: Positive Arousal, Negative Valence.
   - **Class 2**: Negative Arousal, Negative Valence.
   - **Class 3**: Negative Arousal, Positive Valence.

4) **Balancing Dataset**: Based on the new class labels we found that both the datasets were skewed towards positive arousal and negative valence (Class 1). We balanced the dataset using SMOTE which performs oversampling. The Synthetic Minority Over-sampling Technique (SMOTE) is a method used to balance imbalanced datasets, where one class has significantly fewer samples than the others. Instead of just duplicating existing data, SMOTE creates new, realistic samples by looking at similar data points in the minority class and generating new ones between them. This helps machine learning models learn better and avoid being biased toward the majority class. SMOTE is better than simple oversampling because it reduces the risk of overfitting. After analysis, we also found that oversampling gives better results generally than undersampling.
5) **Data Splitting**: Each dataset was split into 60% training, 20% validation and 20% testing sets. The validation dataset is used for tuning the hyperparameters of the classification models.
6) **Shuffling Dataset**: To remove any patterns in the dataset, we shuffle the dataset and help the model generalize well. In order to shuffle the dataset, a new column with random numbers was created, and the data frame was then sorted based on the random numbers. Once the dataset is sorted, the extra column is dropped from the data frame. This ensures that all the rows in the dataset are randomly shuffled while maintaining the attribution to the corresponding target label.
7) **Feature Selection**: We used RFE with Random Forest as an estimator. Recursive Feature Elimination (RFE) is a feature selection technique that iteratively removes the least important features based on model performance. It works by fitting the model multiple times, each time removing the weakest feature(s) and evaluating the model’s accuracy or performance to determine which features contribute most to predicting the target variable. RFE is particularly useful when dealing with high-dimensional datasets, as it helps reduce overfitting and improve model generalization. The process continues until the optimal subset of features is found.

   Figure 2b shows the Scatter Plot of Arousal vs Valence of the EmoSounds dataset, whereas Figure 1b shows the Circular Emotion Model of the EmoSounds dataset. Figure 1c shows the Scatter Plot of Arousal vs Valence of the IADSED dataset, whereas Figure 1d shows the Circular Emotion Model of the IADSED.

IV. Classification Models

We implemented three models:

- **Logistic Regression**: Logistic regression is a simple supervised machine learning algorithm known for its efficiency and low computational cost. It is used for predictive analysis and employs the sigmoid function to estimate the probability of each class.
- **Random Forest:**: Random Forest is a supervised learning algorithm that uses an ensemble of decision trees and bootstrapping on the dataset. It combines the majority vote from multiple trees to make predictions. Random Forests outperform single decision trees as they offer better generalization and are less sensitive to small variations in the training data, thus helping reduce overfitting.
- **eXtreme Gradient Boosting**: eXtreme Gradient Boosting is a powerful supervised ensemble technique that builds multiple decision trees sequentially. Each tree attempts to correct the residual errors from the previous one. The final prediction is the combined result from all the trees, making XGBoost highly effective for complex tasks and improving prediction accuracy.

A. Evaluation

We evaluated the models based on accuracy and macro-F1 scores. Accuracy is a metric that measures the percentage of correct predictions made by a model, calculated as the ratio of correct predictions to the total number of predictions. The Macro F1 score is an evaluation metric that averages the F1 scores of all classes, treating each class equally regardless of its frequency, and providing a balanced measure of a model’s precision and recall.

V. Results

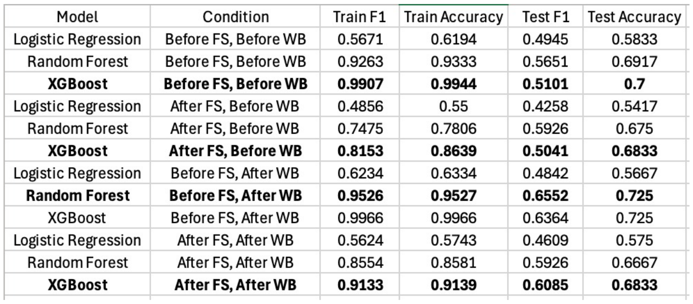
Accuracy and F1 score metric pick up essential information regarding data balancing, feature selection, and to some extent, model behavior. For both the EmoSounds and IADSED datasets, Logistic Regression was worse after feature selection, having F1 scores decreasing from 0.49 to 0.42 in EmoSounds and from 0.6 to 0.59 in IADSED. This suggest that feature selection must have removed essential predictive features upon which the simpler model relied more on.

Random Forest and XGBoost are both extremely robust baseline performances but reportes a slight marginal improvement after feature selection. Random Forest for the EmoSounds dataset registered an improvement after feature selection by a 0.0275 boost in F1 score (from 0.56 to 0.59), whereas XGBoost registered a slight dip of 0.006 (from 0.5101 to 0.5041). On the IADSED data set, both models achieved minor gains post-feature selection: Random Forest by 0.0026 (from 0.59 to 0.60) and XGBoost by 0.0046 (from 0.5459 to 0.5505). This indicates the natural resistance of the tree-based ensemble algorithms like random forest and XGboost against the features and then the fact that some of the features play a vital role in the tree-based model prediction quality.

Data balancing with SMOTE indicated different type of effects among models. Random Forest test F1 measure increased from 0.5651 to 0.6552 in EmoSounds, while XGBoost increased from 0.5101 to 0.6364 under data balancing. Logistic Regression dropped from 0.4945 to 0.4842 for test F1 measure and also from 0.5833 to 0.5667 for test accuracy. In general, before feature selection and data balancing, the XGBoost performed slightly better than the other models in the EmoSounds dataset. Whereas in the IADSED dataset, Random forest performed the best.

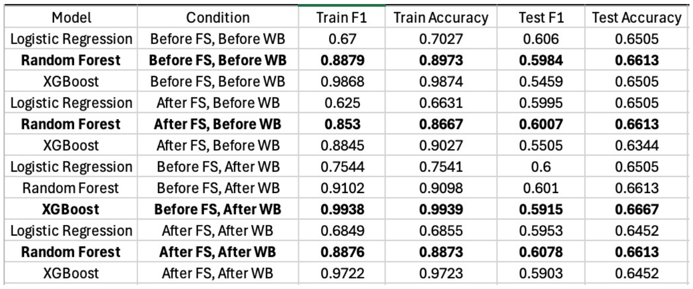
Model comparison on performance, XGBoost tended to perform better than Logistic Regression and Random Forest on the training set, and in a few instances, was near-perfect (e.g., 0.99 on EmoSounds). But such high training accuracy tended to lead to overfitting as can be inferred from the train-test difference. Random Forest might also have a nicely balanced train and test accuracy at times but lag behind XGBoost at times on F1-scores. Simple Logistic Regression had the least train-test gap, meaning that it generalized best despite having lower absolute performance values. For all the best model confusion matrices in each scenario, please refer the code file.

A. Table of results



Model	Condition	Train F1	Train Accuracy	Test F1	Test Accuracy
Logistic Regression	Before FS, Before WB	0.5671	0.6194	0.4945	0.5833
Random Forest	Before FS, Before WB	0.9263	0.9333	0.5651	0.6917
XGBoost	**Before FS, Before WB**	**0.9907**	**0.9944**	**0.5101**	**0.7**
Logistic Regression	After FS, Before WB	0.4856	0.55	0.4258	0.5417
Random Forest	After FS, Before WB	0.7475	0.7806	0.5926	0.675
XGBoost	**After FS, Before WB**	**0.8153**	**0.8639**	**0.5041**	**0.6833**
Logistic Regression	Before FS, After WB	0.6234	0.6334	0.4842	0.5667
Random Forest	**Before FS, After WB**	**0.9526**	**0.9527**	**0.6552**	**0.725**
XGBoost	Before FS, After WB	0.9966	0.9966	0.6364	0.725
Logistic Regression	After FS, After WB	0.5624	0.5743	0.4609	0.575
Random Forest	After FS, After WB	0.8554	0.8581	0.5926	0.6667
XGBoost	**After FS, After WB**	**0.9133**	**0.9139**	**0.6085**	**0.6833**

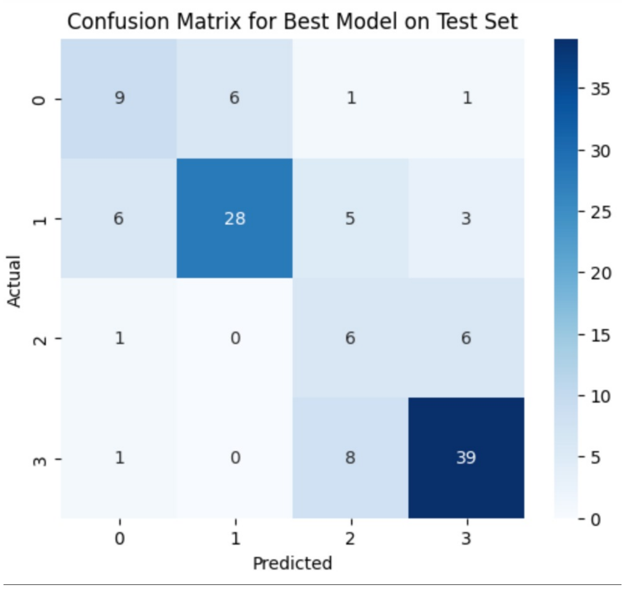
(a) EmoSounds



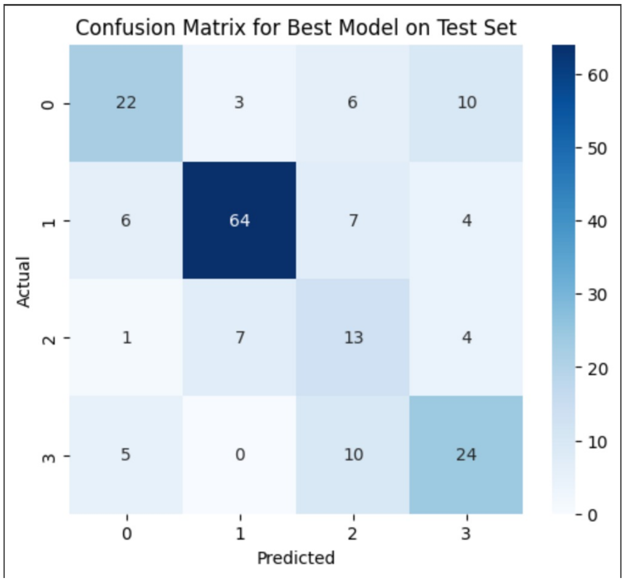
Model	Condition	Train F1	Train Accuracy	Test F1	Test Accuracy
Logistic Regression	Before FS, Before WB	0.67	0.7027	0.606	0.6505
Random Forest	**Before FS, Before WB**	**0.8879**	**0.8973**	**0.5984**	**0.6613**
XGBoost	Before FS, Before WB	0.9868	0.9874	0.5459	0.6505
Logistic Regression	After FS, Before WB	0.625	0.6631	0.5995	0.6505
Random Forest	**After FS, Before WB**	**0.853**	**0.8667**	**0.6007**	**0.6613**
XGBoost	After FS, Before WB	0.8845	0.9027	0.5505	0.6344
Logistic Regression	Before FS, After WB	0.7544	0.7541	0.6	0.6505
Random Forest	Before FS, After WB	0.9102	0.9098	0.601	0.6613
XGBoost	**Before FS, After WB**	**0.9938**	**0.9939**	**0.5915**	**0.6667**
Logistic Regression	After FS, After WB	0.6849	0.6855	0.5953	0.6452
Random Forest	**After FS, After WB**	**0.8876**	**0.8873**	**0.6078**	**0.6613**
XGBoost	After FS, After WB	0.9722	0.9723	0.5903	0.6452

(b) IADSED

B. Confusion Matrix - After Feature Selection on balanced dataset



(a) EmoSounds Confusion Matrix (XGBoost)



(b) IADSED Confusion Matrix (Random Forest)

VI. Conclusion and Future Work

In conclusion, feature selection will enable simpler models such as Logistic Regression to become simpler but intentionally hinder complicated models if valuable features are lost. SMOTE worked extremely well in class balance but had infinitesimal overfitting risks. XGBoost is extremely effective but must be regularized carefully to prevent overfitting, whereas Random Forest offers an exceptionally good tradeoff between complexity and generalizability. Future work will involve hyperparameter optimization, regularization techniques, and ensemble techniques to improve model performance and offer strong generalizability across different datasets.