Predicting Emotions from Sound Events

Rakshit Gupta[1], Nitish Ranjan[1], Xiaoqian Yang[1],
[1]Computer Science Department, San Jose State University
{*faranak.abri, rakshit.gupta, nitish.ranjan, xiaoqian.yang*}*@sjsu.edu*

I. Introduction

Research on predicting emotions based on sound events or soundscapes is scarce. This study aims to predict emotions based on datasets of sounds using machine learning.
The report is organized as follows: Section II covers related works in emotion predictions using sounds and soundscapes. Section III provides the description of datasets used for the emotion predictions. Section IV describes the preprocessing techniques and the libraries used for preprocessing. Section V illustrates the prediction models and their evaluations. Section VI of the report discusses the feature selection methods used in this project. Computational resources used in this project are outlined in Section VII. Further, different models are compared and analyzed in Section VIII. We conclude with the results of our predictions in Section IX.

II. Related Work

Emotions are associated with sounds in two ways: one is "perceived" emotions, in which listeners recognize the emotions expressed by the source, and the other one is "induced" emotions, in which listeners feel emotions provoked by the source [1].

Two representative datasets are used to study the performance of the prediction of these two types of emotions: Emo-Soundscapes [2] and IADSE [3]. Emo-Soundscapes [2] allows for studying Soundscape emotion recognition and how the mixing of various soundscape recordings influences listeners' perceived emotions. It contains 600 audio clips following Schafer's taxonomy and 613 mixed sounds from freesound. A two-dimensional space (arousal/valence) is used in this dataset. The researchers also provided two protocols and demonstrated a few baseline SVR models to evaluate future models of SER. IADSE [3] allows for studying how soundscape recordings influence listeners' induced emotions. It contains 935 sound recordings from different sources and uses a three-dimensional space(arousal/valence/dominance).

III. Dataset Description

The datasets used are Emo-soundscapes [2] and IADSE [3]. These datasets contain sound samples with annotated emotions. Emo-soundscapes contains 600 samples with 68 features and is tagged with arousal and valence. IADSE contains 935 samples with 68 features and is tagged with arousal, valence, and dominance. After exploring the IADSE dataset, no missing values were found in the Emosoundscape dataset.

https://www.overleaf.com/read/wxfyggzzcfmm

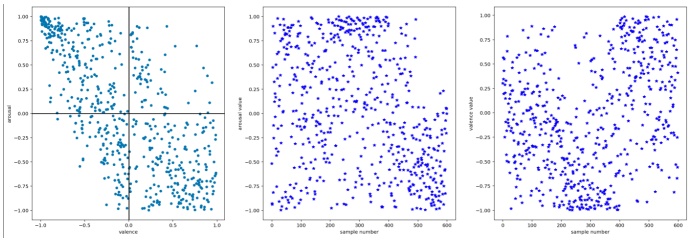

Fig. 1: Data distribution of Arousal and Valence for Emo-Soundscapes

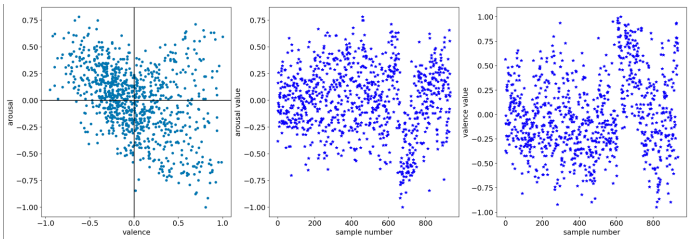

Fig. 2: Data distribution of Arousal and Valence for IADSE

However, we found that there were a total of 8 rows in the IADSE dataset with missing values in the following columns: tonal_keyclarity_mean, tonal_keyclarity_std, tonal_mode_std, tonal_hcdf_std, rhythm_temp_std, rhythm_temp_mean.

The Python library used for creating plots for visualizations is Matplotlib. Figure 1 shows scatter plots of valence versus arousal, arousal values and valence values to better understand the Emo-Soundscapes dataset. Figure 2 plots the same figures for IADSE dataset.

IV. Preprocessing

Data preprocessing is an essential step after collecting and assembling the data. Raw data collected from the real world tends to be inconsistent, missing, duplicated, or noisy. Preprocessing data can help to transform the data into a useful format and improve the overall data quality.

A. Background about preprocessing techniques

In general, there are four primary techniques that can be used for preprocessing data, i.e., data cleaning, reduction, scaling, and transformation [4]. Data cleaning is applied to handle the missing values and noisy data and to remove outliers. Data reduction is used to handle large amounts of data and is usually used to reduce data dimensions. Data scaling aims to normalize or standardize data, transforming the

original data into a smaller range. Data transformation converts the data into appropriate formats, for example, transferring numerical data into categorical data.

B. Preprocessing techniques and libraries

In the preprocessing step, two main libraries were imported, which are Pandas and Numpy. Pandas library was used to read CSV files into dataframes; enabling better visualization and preprocessing of data. The Numpy library was used to select all numerical columns of the datasets.

Eight rows contain null values in the IADSE dataset. Since these entries comprise only about 0.85% of the total data, these rows were dropped from the dataset using the dropna() function.

To implement data scaling for both datasets, min-max normalization was used to normalize all numerical columns. Min-max normalization is a widely used method to normalize data. For every feature, the maximum value is transformed into 1, the minimum value is transformed into 0, and other values are transformed into decimals between 0 and 1. It calculates and subtracts the minimum value for every numerical feature, then divides this value by the subtraction of the maximum value and minimum value, as shown in equation (1).

$$x = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

C. Dataset Labeling

This project aims to predict the four distinct categories of emotions as depicted in Figure 3. Since we are using supervised learning algorithms to make the predictions, each row of the datasets needs to be labeled correctly into one of the four classes. The rows of both datasets can be labeled using the arousal and valence columns. The arousal and valence values were scaled to -1 and 1 before assigning the rows to each class. Then each class was assigned based on the below logic:

- arousal ≥ 0 and valence ≥ 0: Class 1 (High-Arousal, Positive-Valence);
- arousal ≥ 0 and valence < 0: Class 2 (High-Arousal, Negative-Valence);
- arousal < 0 and valence < 0: Class 3 (Low-Arousal, Negative-Valence);
- arousal < 0 and valence ≥ 0: Class 4 (Low-Arousal, Positive-Valence);

After labeling, data distribution was computed for each class. It was observed that both the Emo-Soundscapes and the IADSE datasets are imbalanced.

V. Prediction Models

Two different classification models were used to train the models and predict values - Logistic regression and Random Forest classification.

Two splitting methods were adapted during the train-split phases. The first one is 80% for training and 20% for testing, the second one is 60% for training, 20% for validation 20% for testing. The training dataset is the sample of data used to fit the model; the Validation dataset is the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration; the Test dataset is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. For the linear classification models, datasets were split using the first method. For the Random Forest classification model, datasets were split using the second method.

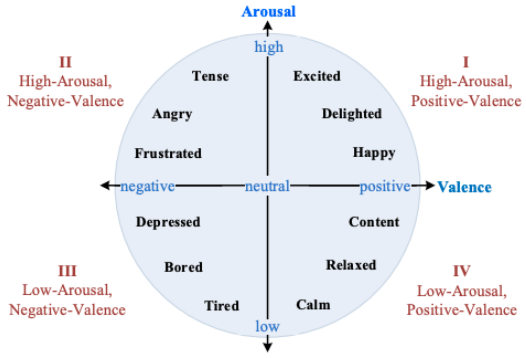

Fig. 3: Two-dimensional valence-arousal space [5]

Linear and non-linear models were trained separately to compare accuracy and errors more efficiently.

A. Evaluation

The F1-score, also called the F-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing.

It is possible to adjust the F-score to give more importance to precision over recall, or vice-versa. Common adjusted F-scores are the F0.5-score and the F2-score, as well as the standard F1-score. The formula for the standard F1-score is the harmonic mean of the precision and recall. A perfect model has an F-score of 1.

$$F_1 = \frac{2}{\frac{1}{recall} \times \frac{1}{precision}} \tag{2}$$

When performing Random Forest classification, the predictions made by the model on training data were 100% accurate, and it had an F1-score of 1, whereas the model performed poorly on the test data and had a low F1-score. These observations indicated that the model was suffering from the problem of overfitting. To deal with the overfitting of the Random Forest Classifier, we tuned its hyper-parameters to the optimal value using a validation set and found the parameter

	Logistic Regression		Random Forest	
	Train	Test	Train	Test
Class 1	0.66	0.34	0.85	0.15
Class 2	0.89	0.74	0.96	0.79
Class 3	0.59	0.38	0.88	0.34
Class 4	0.83	0.79	0.93	0.82
Macro Avg	0.74	0.56	0.90	0.53

Fig. 4: F1-scores - EMO soundscape

	Logistic Regression		Random Forest	
	Train	Test	Train	Test
Class 1	0.61	0.52	0.98	0.34
Class 2	0.71	0.62	0.99	0.70
Class 3	0.57	0.48	0.99	0.43
Class 4	0.72	0.63	0.98	0.68
Macro Avg	0.65	0.56	0.99	0.54

Fig. 5: F1-scores - IADSE

"min_samples_leaf" to be the prime contributor to overfitting. This parameter determines the minimum number of samples required to split the node. So the default value of 1 could mean that each data sample of the training set could have its tree node. Similarly, the "max_depth" hyper-parameter was updated to reduce the complexity of the model and prevent overfitting.

B. Cross Validation

Cross validation is a technique for evaluating an ML model and testing its performance. It helps to compare and select an appropriate model, and it tends to have a lower bias than other methods used to count the model's efficiency scores.

To check the effectiveness of the trained model, five-fold cross-validation was used in the project. The average F-1 score from the validated datasets was compared with the test F-1 score of the trained models. The difference was minute, proving that overfitting was mitigated.

Post selecting the optimum number of features, cross-validation was performed again to fine-tune the hyperparame-

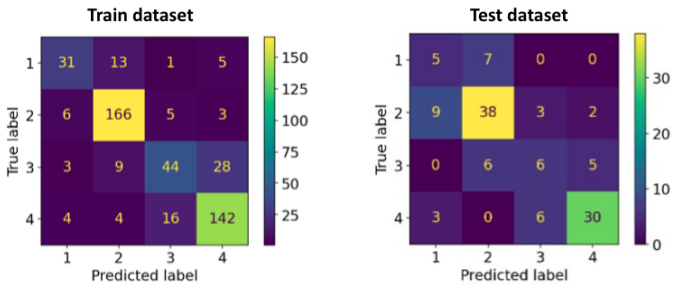

Fig. 6: Confusion Matrices for Logistic Regression - EMO soundscape

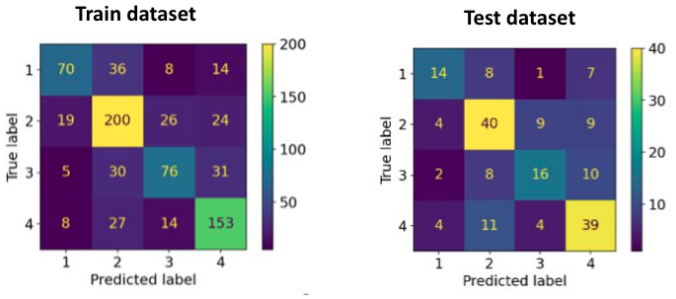

Fig. 7: Confusion Matrices for Logistic Regression - IADSE

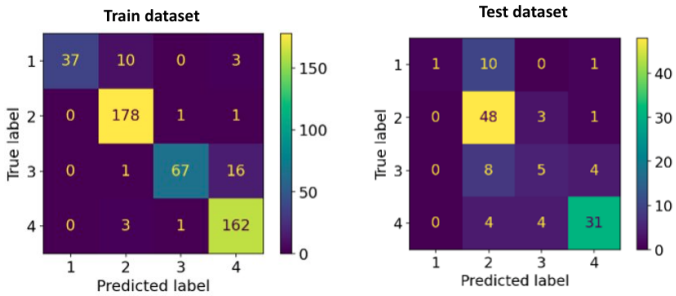

Fig. 8: Confusion Matrices for Random Forest Classifier - EMO soundscape

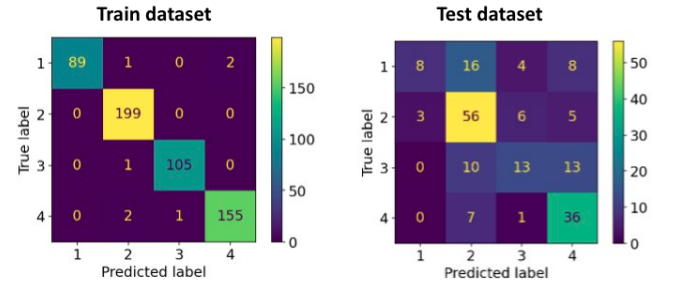

Fig. 9: Confusion Matrices for Random Forest Classifier - IADSE

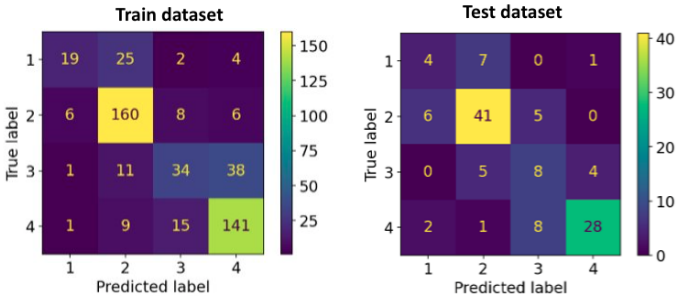

Fig. 10: Confusion Matrices for Logistic Regression with Feature Selection - EMO soundscape

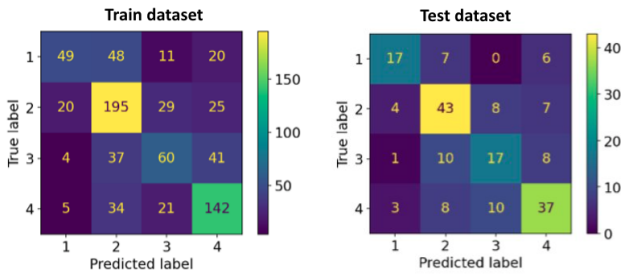

Fig. 11: Confusion Matrices for Logistic Regression with Feature Selection - IADSE

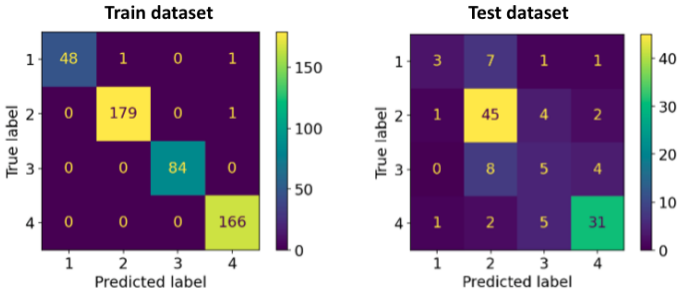

Fig. 12: Confusion Matrices for Random Forest with Feature Selection in EMO soundscape

ters. For this, the GridSearchCV method was used with five-fold validation setting.

C. Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the best set of parameters for a model to obtain optimal results. Grid search is a technique that can be employed to find the optimal parameters of the model through which all combinations of the determined values for parameters are examined. An exhaustive search was performed for hyper-parameter tuning on 4 parameters in order to find the optimal values. Here is the list of parameters that were tuned:

- n_estimators : (50, 100, 150), number of trees in the forests;
- max_depth : (10, 50, 100), the maximum number of levels in each decision tree;

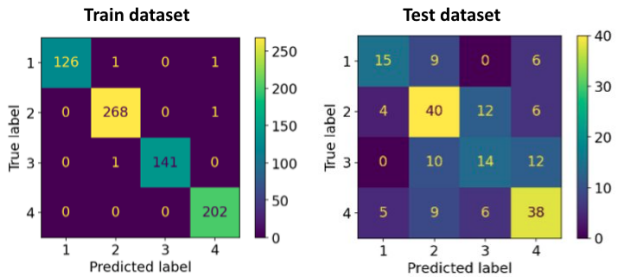

Fig. 13: Confusion Matrices for Random Forest with Feature Selection - IADSE

	Logistic Regression		Random Forest	
	Train	Test	Train	Test
Class 1	0.49	0.33	0.98	0.35
Class 2	0.83	0.77	0.99	0.79
Class 3	0.48	0.42	1.00	0.31
Class 4	0.79	0.78	0.99	0.81
Macro Avg	0.65	0.58	0.99	0.57

Fig. 14: F1-scores with Feature selection (Filter method) - EMO

- min_samples_split : (2, 3, 5), the minimum number of data points placed in a node before the node is split;
- min_samples_leaf : (2, 3, 5), the minimum number of data points allowed in a leaf node;
- criterion : (gini, entropy), the function to measure the quality of a split;

VI. Feature Selection

In general, the datasets contain huge amounts of data. Not all data is useful for our training, for instance, noisy data, irrelevant data, or duplicated data can slow down the process of training models. And this is why choosing important features for the model is very important. Selecting the optimal subset of the original feature set is called feature selection.

Generally, there are two main techniques of feature selection - supervised feature selection and unsupervised feature selection. Commonly used supervised feature selection techniques are - the filter, wrapper, and embedded methods. The project uses filter and wrapper methods.

In the filter method, choosing features is considered a pre-processing step, and constant and quasi-constant columns were first filtered out. Then SelectKBest method was used to choose the optimal number of features for different models. The SelectKBest method is easy to implement, low in computational time, and does not cause overfitting of data. Libraries that were imported for the project include SelectKBest and f_classif (ANOVA) from sklearn.feature_selecton.

In the wrapper method, the features are selected by treating it as a search problem, in which alternative combinations are created, assessed, and compared with other combinations. It trains the algorithm by using the subset of features iteratively. Boruta Algorithm was used in the project. The idea behind Boruta is that a feature is useful only if it can do better than the best randomized feature. The project extracted the best features using 100 iterations of the Boruta algorithm to shuffle and train the data. These extracted features were the best suited based on feature importance. The library that was imported for the project was boruta.BorutaPy.

The most valuable features were selected based on different methods and models. The final number of features are shown in Figure 18.

VII. Computational Resources

In both filter and wrapper methods, two classification models were used for predicting classes. One linear model-

	Logistic Regression		Random Forest	
	Train	Test	Train	Test
Class 1	0.48	0.62	0.99	0.56
Class 2	0.67	0.66	0.99	0.62
Class 3	0.46	0.48	1.00	0.41
Class 4	0.66	0.64	1.00	0.63
Macro Avg	0.57	0.60	0.99	0.55

Fig. 15: F1-scores with Feature selection (Filter method) - IADSE

	Logistic Regression		Random Forest	
	Train	Test	Train	Test
Class 1	0.42	0.45	0.77	0.40
Class 2	0.84	0.76	0.93	0.79
Class 3	0.51	0.29	0.81	0.26
Class 4	0.80	0.76	0.90	0.80
Macro Avg	0.64	0.57	0.85	0.56

Fig. 16: F1-scores with Feature selection (Wrapper method) - EMO

LogisticRegression(), and one non-linear classification model- RandomForestClassifier(). Wrapper feature selection methods are computationally expensive, hence the project was done using a Google Colab notebook with 13GB RAM and a hard disk capacity of 108GB - Python 3 Google Compute Engine.

VIII. Comparison and Analysis

Feature selection results in a more accurate model. The F1-scores reported in the project were slightly higher with feature selection than without feature selection for the Emo-soundscape dataset, whereas performance on the IADSE dataset remained relatively unaffected. The changes in f1-scores after incorporating feature selection are illustrated in figures 14, 15, 16 and 17. For better visualization of the comparison, the train and test F1-scores without feature selection and with feature selection are shown in figures 19 and 20. Poor macro-average F1-scores compared to per-class F1-scores can be attributed to the imbalance in the datasets.

	Logistic Regression		Random Forest	
	Train	Test	Train	Test
Class 1	0.53	0.69	0.98	0.62
Class 2	0.70	0.63	0.99	0.63
Class 3	0.53	0.37	0.98	0.37
Class 4	0.70	0.62	0.99	0.65
Macro Avg	0.62	0.58	0.99	0.57

Fig. 17: F1-scores with Feature selection (Wrapper method) - IADSE

Model	Feature Selection Method	Emo Soundscape	IADSE
Logistic Regression	Filter	40	20
Random Forest	Filter	40	25
Logistic Regression	Boruta	26	31
Random Forest	Boruta	26	31

Fig. 18: Number of features selected

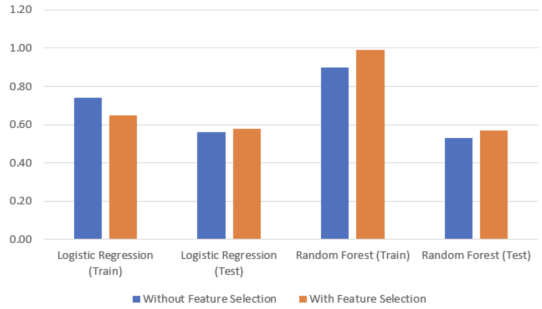

Fig. 19: F1-scores comparison for EMO dataset

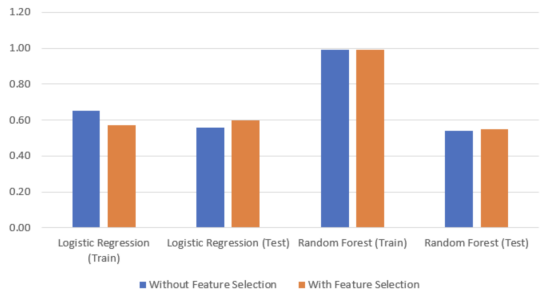

Fig. 20: F1-scores comparison for IADSE dataset

IX. Conclusion

Predicting emotions through sounds and soundscapes is promising, but comes with its own challenges and problems. It is important to understand perceived (expressed emotions) and induced (felt) emotions. Perceived emotions are easier to predict than induced emotions, as is evident from the range of errors in the two datasets of EMO-Soundscape and IADSE. This notion is further bolstered by the fact that induced emotions are subjective and vary highly from individual to individual.

References

[1] F. Abri, L. Gutiérrez, P. Datta, D. Sears, A. Siami Namin, and K. Jones, "A comparative analysis of modeling and predicting perceived and induced emotions in sonification," *Electronics*, vol. 10, p. 2519, 10 2021.

[2] J. Fan, M. Thorogood, and P. Pasquier, "Emo-soundscapes: A dataset for soundscape emotion recognition," in *International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2017.

[3] Y. Wanlu, K. Makita, T. Nakao,, N. Kanayama, M. Machizawa, T. Sasaoka, A. Sugata, R. Kobayashi, H. Ryosuke, and S. Yamawaki, "Affective auditory stimulus database: An expanded version of the international affective digitized sounds(iads-e), doi:10.3758/s13428-018-1027-6." in *Behav. Res. Methods*, 2018.

[4] F. Xiao and C. Fan, "Data mining in building automation system for improving building operational performance," in *Energy and Buildings. 75 (11), 109–118. doi:10.1016/j.enbuild.2014.02.005*, 2014.

[5] L.-C. Yu, L.-H. Lee, S. Hao, J. Wang, Y. He, J. Hu, K. R. Lai, and X. Zhang, "Building Chinese affective resources in valence-arousal dimensions," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 540–545. [Online]. Available: https://aclanthology.org/N16-1066