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 talks about the feature selection methods used in this project

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. 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.

https://www.overleaf.com/project/630e97315f164d91aa543c0d

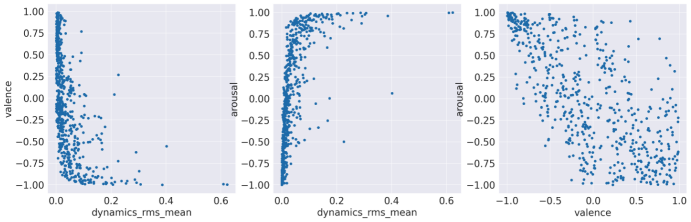

Fig. 1: Scatter plots for Emo-Soundscapes dataset

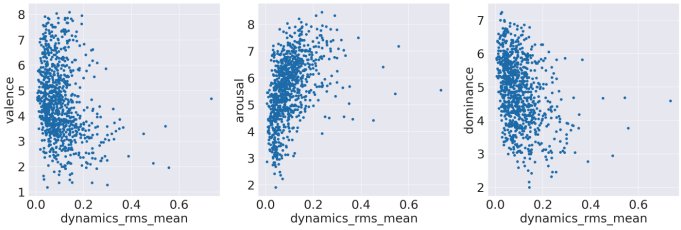

Fig. 2: Scatter plots for IADSE dataset

The Python library used for creating plots for visualizations is Matplotlib. Figure 1 shows scatter plots of dynamics_rms_mean versus valence, dynamics_rms_mean versus arousal, and valence versus arousal to better understand the Emo-Soundscapes dataset. Figure 2 plots dynamics_rms_mean versus valence, dynamics_rms_mean versus arousal, and dynamics_rms_mean versus dominance 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 used in P1.1

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, two methods are used to normalize all numerical columns, which are mean normalization and min-max normalization. Mean normalization is implemented by calculating and subtracting the mean for every numerical feature, and then dividing this value by standard deviation, as shown in equation (1). Min-max normalization is another widely used way 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 (2).

$$x = \frac{x - \mu}{\sigma} \tag{1}$$

$$x = \frac{x - min(x)}{max(x) - min(x)} \tag{2}$$

V. Prediction Models

Three different regression models were used to train the models and predict values - Simple Linear Regression, Polynomial Regression (with degrees 3 and 2), and Random Forest Regression.

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 regression models, datasets were split using the first method. For the non-linear regression model, datasets were split using the second method.

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

A. Evaluation

Root mean square error (RMSE) is one of the most commonly used measurements for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance. The RMSE value is given by the equation:

$$RMSE = \sqrt{\frac{\Sigma(y_i - \hat{y_i})^2}{n}} \tag{3}$$

RMSE values for different target values (valence, arousal, and dominance) are given in Figure 3 and 4.

When performing the Linear Regression model, the training error is small and the gap between the training error and the test error is small as well. When performing Polynomial Regression (degree three) model, the training error was negligible compared to the testing error, which indicates that the overfitting problem exists. One way to solve the overfitting problem is to reduce the complexity of the model. However, overfitting persisted even with a degree of two (decreased complexity) in the Polynomial Regression. Regularization is another way to solve the overfitting problem. Lasso regression (regularization), one of the simple techniques to reduce model complexity and prevent over-fitting, is conducted in the experiment. Regularization removed overfitting and reduced test error. The value used for the regularization parameter (λ) was 0.01. The new error values were comparable to those obtained from the simple linear regression model (degree one). For comparison with non-linear regression models, random forest regression was used.

	Emo Soundscape					
	Arousal			Valence		
Model	Train	Test	Validate	Train	Test	Validate
Linear regression(deg 1)	0.227	0.347	-	0.336	0.414	-
Linear regression(deg 2)	0.000	0.482	-	0.000	0.815	-
Linear regression(deg 3)	0.000	0.449	-	0.000	0.765	-
Regularized model	0.351	0.368	-	0.424	0.404	-
Random Forest model	0.103	0.239	0.237	0.141	0.344	0.422

Fig. 3: Emo-Soundscapes: RMSE of different models without Feature Selection

	IADSE								
	Arousal			Valence			Dominance		
Model	Train	Test	Validate	Train	Test	Validate	Train	Test	Validate
Linear regression(deg 1)	0.749	0.903	-	1.082	1.142	-	0.877	0.772	-
Linear regression(deg 2)	0.000	2.229	-	0.000	5.646	-	0.000	3.778	-
Linear regression(deg 3)	0.000	1.712	-	0.000	3.267	-	0.000	2.782	-
Regularized model	0.658	0.804	-	1.042	1.041	-	0.810	0.847	-
Random Forest model	0.285	0.763	0.811	0.438	1.075	1.164	0.321	0.799	0.867

Fig. 4: IADSE dataset: RMSE of different models without Feature Selection

	Emo Soundscape		IADSE		
Model	Arousal	Valence	Arousal	Valence	Dominance
Linear regression	0.302	0.424	0.853	1.186	0.858
Random Forest model	0.236	0.369	0.236	1.107	0.823

Fig. 5: RMSE values for Cross Validations without Feature Selection

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 resulting RMSE values are shown in Figure 5. The average error from the validated datasets was compared with the test error 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 hyperparameters. For this, the GridSearchCV method was used with five-fold validation.

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;
- min_samples_split : (2, 3, 5, 7), the minimum number of data points placed in a node before the node is split;
- min_samples_lea f : (1, 3, 5), the minimum number of data points allowed in a leaf node;

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_regression 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 method 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. Library that was imported for the project was boruta.BorutaPy.

Number of features					
	Emo Soundscape		IADSE		
	Arousal	Valence	Arousal	Valence	Dominance
Linear Regression (Filter)	18	15	30	16	9
Random Forest (Filter)	15	13	29	10	8
Linear Regression (Boruta)	22	15	21	15	12
Random Forest (Boruta)	22	15	21	15	12

Fig. 6: Number of features selected

Different number of features were selected for arousal and valence based on different methods and models. The final numbers were shown in Figure 6.

VII. Computational Resources

In both filter and wrapper methods, two regression models were used for predicting arousal and valence separately. One linear model-LinearRegression(), and one non-linear regression model- RandomForestRegressor(). 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 error values reported in the project were lower with feature selection than without feature selection (for the linear model). The Random Forest model remains relatively unaffected by feature selection since it has its own tree-based built-in selection, eliminating the need to remove irrelevant features. This is demonstrated by many guides on RF/ML algorithm tuning. For example, the book 'Hands-on Machine Learning' (Géron 2019) provides evidence that Random Forest tuning is sufficient to deal with irrelevant features [5]. The changes in error values after incorporating feature selection are illustrated in figures 7 and 8. For better visualization of the comparison, the train and test RMSE without feature selection and with feature selection are shown in Figure 9, Figure 10, Figure 11, Figure 12,

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 perceived

Filter Method

	Emo Soundscape			
	Arousal		Valence	
Model	Train	Test	Train	Test
Linear regression(deg 1)	0.295	0.259	0.386	0.376
Random Forest model	0.101	0.240	0.142	0.365

Wrapper Method

	Emo Soundscape			
	Arousal		Valence	
Model	Train	Test	Train	Test
Linear regression(deg 1)	0.290	0.341	0.413	0.408
Random Forest model	0.096	0.246	0.144	0.360

Fig. 7: EMO dataset: RMSE of different models with Feature Selection

Filter Method

	IADSE					
	Arousal		Valence		Dominance	
Model	Train	Test	Train	Test	Train	Test
Linear regression(deg 1)	0.809	0.903	1.177	1.113	0.834	0.888
Random Forest model	0.293	0.792	0.495	1.038	0.544	0.843

Wrapper

	IADSE					
	Arousal		Valence		Dominance	
Model	Train	Test	Train	Test	Train	Test
Linear regression(deg 1)	0.815	0.891	1.191	1.133	0.833	0.892
Random Forest model	0.364	0.765	0.513	1.046	0.424	0.832

Fig. 8: IADSE dataset: RMSE of different models with Feature Selection

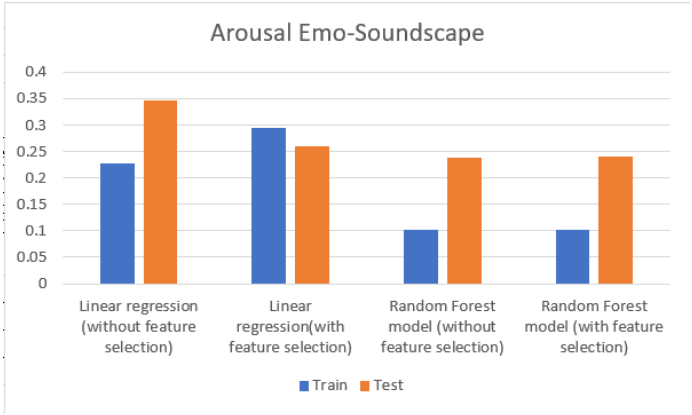

Fig. 9: Emo_soundscape dataset: RMSE of different models for arousal

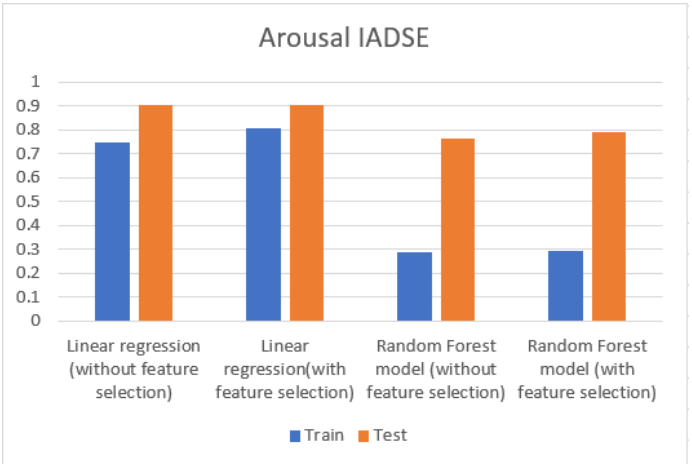

Fig. 10: IADSE dataset: RMSE of different models for arousal

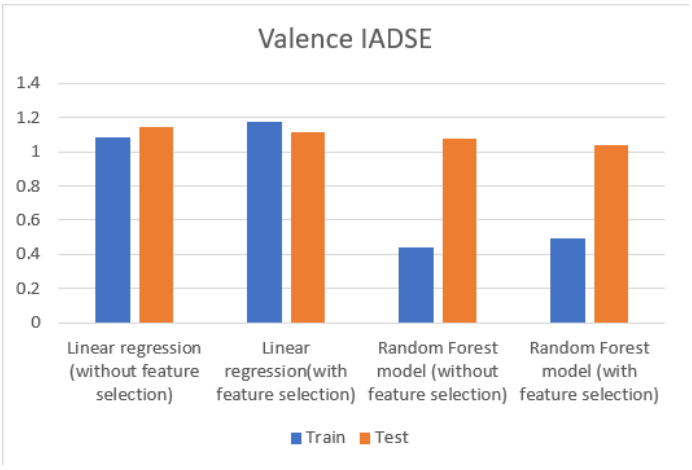

Fig. 11: IADSE dataset: RMSE of different models for valence

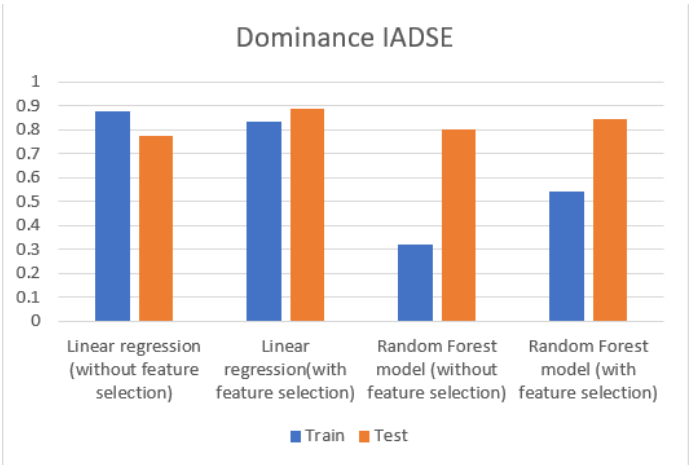

Fig. 12: IADSE dataset: RMSE of different models for dominance

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] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc., 2019. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/